



Cisco MDS 9000 シリーズ リリース 9.x SAN Analytics/SAN Telemetry Steaming 構成ガイド

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255 (フリーコール、携帯・PHS含む)

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（ www.cisco.com/jp/go/safety_warning/ ）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2021–2022 Cisco Systems, Inc. All rights reserved.



目次

Full Cisco Trademarks with Software License ?

はじめに :

はじめに	vii
対象読者	vii
表記法	vii
関連資料	viii
通信、サービス、およびその他の情報	ix

第 1 章

新機能および変更された機能に関する情報	1
変更点	1

第 2 章

SAN アナリティクス ソリューション	5
SAN Analytics ソリューションの概要	5

第 3 章

SAN アナリティクスの設定	7
SAN Analytics の構成の機能履歴	7
SAN アナリティクスの概要	10
SAN アナリティクスのハードウェア要件	11
SAN Analytics の注意事項と制約事項	12
コマンドの変更	15
SAN アナリティクスについて	16
VMID Analytics	21
ポート サンプリング	22
分析エンジン ポート セットのマッピング	24

展開モード	25
SAN アナリティクスの設定	31
SAN アナリティクスの有効化	32
SAN アナリティクスの無効化	32
インターフェイスでの SAN Analytics の有効化	33
インターフェイスでの SAN アナリティクスの無効化	33
VMID Analytics の有効化	34
VMID Analytics の無効化	35
ポート サンプリングの有効化	35
ポート サンプリングの無効化	36
例：SAN Analyticsの設定	36
スイッチでのメトリクスのクエリ	37
メトリックのクエリ用のスキーマ	37
クエリ構文	37
クエリ ルール	38
ビュー	39
サポートされているビュー タイプの一覧	39
ビュー タイプの表示	43
例：クエリ構文の設定	55
クエリの作成と使用	59
インストールされているプッシュ クエリの表示	59
プッシュ クエリの結果の表示	60
プル クエリの実行	60
プッシュ クエリの設定	60
設定されているプッシュ クエリの削除	60
メトリックのクリア	61
ビューの消去	61
設定されているプッシュ クエリの結果の表示	62
例：クエリの作成と使用	62
ShowAnalytics オーバーレイ CLI の使用	78
例：ShowAnalytics オーバーレイ CLI の使用	79

フローごとの輻輳ドロップの表示	98
例：フローごとの輻輳ドロップの表示	98
SAN アナリティクスの確認	99
SAN Analytics のトラブルシューティング	108

 第 4 章

SAN Telemetry Streaming の構成	111
SAN Telemetry Streaming の設定の機能履歴	111
SAN Telemetry Streaming の概要	112
インターフェイスの統計情報のストリーミング	113
トランシーバパラメータ ストリーミング	113
SAN Telemetry Streaming の注意事項と制約事項	115
gRPC エラーの動作	116
SAN テレメトリ ストリーミングのエンコーディング	117
SAN テレメトリ ストリーミングの設定	118
例：SAN テレメトリ ストリーミングの設定	121
SAN テレメトリ ストリーミングの設定と統計情報の表示	124
SAN テレメトリ ストリーミングのトラブルシューティング	130

 付録 A :

付録	133
フローメトリック	133
List of Supported Flow Metrics	134
ポート ビューインスタンス (ポート)	134
論理ポート ビューインスタンス (logical_port)	146
アプリケーション ビューインスタンス (app)	158
ターゲット ビューインスタンス (scsi_target および nvme_target)	159
イニシエーター ビューインスタンス (scsi_initiator および nvme_initiator)	171
ターゲット アプリケーション ビューインスタンス (scsi_target_app および nvme_target_app)	182
イニシエーター アプリケーション ビューインスタンス (scsi_initiator_app および nvme_initiator_app)	183
ターゲット IT フロー ビューインスタンス (scsi_target_it_flow および nvme_target_it_flow)	184

イニシエーター IT フロー ビュー インスタンス (scsi_initiator_it_flow および nvme_initiator_it_flow)	195
ターゲット TL フロー ビュー インスタンス (scsi_target_tl_flow)	205
ターゲット TN フロー ビュー インスタンス (nvme_target_tn_flow)	216
イニシエーター ITL フロー ビュー インスタンス (scsi_initiator_itl_flow)	226
イニシエーター ITN フロー ビュー インスタンス (nvme_initiator_itn_flow)	236
ターゲット ITL フロー ビュー インスタンス (scsi_target_itl_flow)	246
ターゲット ITN フロー ビュー インスタンス (nvme_target_itn_flow)	256
イニシエーター IO フロー ビュー インスタンス (scsi_initiator_io および nvme_initiator_io)	266
ターゲット IO フロー ビュー インスタンス (scsi_target_io および nvme_target_io)	269
インターフェイス カウンタ	271
SAN テレメトリ ストリーミング Proto ファイル	277



はじめに

ここでは、『Cisco MDS 9000 Series Configuration Guide』を使用している対象読者、構成、および表記法について説明します。また、関連資料の入手方法の情報を説明し、次の章にも続きます。

- [対象読者](#) (vii ページ)
- [表記法](#) (vii ページ)
- [関連資料](#) (viii ページ)
- [通信、サービス、およびその他の情報](#) (ix ページ)

対象読者

このインストレーションガイドは、電子回路および配線手順に関する知識を持つ電子または電気機器の技術者を対象にしています。

表記法

このマニュアルでは、次の表記法を使用しています。



(注) 「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。



注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

警告は、次のように表しています。



警告 「危険」の意味です。人身事故を予防するための注意事項が記述されています。装置の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止策に留意してください。各警告の最後に記載されているステートメント番号を基に、装置に付属の安全についての警告を参照してください。ステートメント 1071。

関連資料

Cisco MDS 9000 シリーズ スイッチのドキュメンテーションには、次のマニュアルが含まれます。

Release Notes

<http://www.cisco.com/c/en/us/support/storage-networking/mds-9000-nx-os-san-os-software/products-release-notes-list.html>

『Regulatory Compliance and Safety Information』

<http://www.cisco.com/c/en/us/td/docs/switches/datacenter/mds9000/hw/regulatory/compliance/RCSI.html>

互換性に関する情報

<http://www.cisco.com/c/en/us/support/storage-networking/mds-9000-nx-os-san-os-software/products-device-support-tables-list.html>

インストールおよびアップグレード

<http://www.cisco.com/c/en/us/support/storage-networking/mds-9000-nx-os-san-os-software/products-installation-guides-list.html>

Configuration

<http://www.cisco.com/c/en/us/support/storage-networking/mds-9000-nx-os-san-os-software/products-installation-and-configuration-guides-list.html>

CLI

<http://www.cisco.com/c/en/us/support/storage-networking/mds-9000-nx-os-san-os-software/products-command-reference-list.html>

トラブルシューティングおよび参考資料

<http://www.cisco.com/c/en/us/support/storage-networking/mds-9000-nx-os-san-os-software/tsd-products-support-troubleshoot-and-alerts.html>

オンラインでドキュメントを検索するには、次の Web サイトにある Cisco MDS NX-OS Documentation Locator を使用してください。

http://www.cisco.com/c/en/us/td/docs/storage/san_switches/mds9000/roadmaps/doclocator.html

通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、[Cisco Profile Manager](#) でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、[シスコサービス](#)にアクセスしてください。
- サービス リクエストを送信するには、[シスコサポート](#)にアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、[Cisco Marketplace](#) にアクセスしてください。
- 一般的なネットワーク、トレーニング、認定関連の出版物を入手するには、[Cisco Press](#) にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、[Cisco Warranty Finder](#) にアクセスしてください。

Cisco バグ検索ツール

[Cisco バグ検索ツール](#) (BST) は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理する Cisco バグ追跡システムへのゲートウェイとして機能する、Web ベースのツールです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。



第 1 章

新機能および変更された機能に関する情報

- [変更点, on page 1](#)

変更点

次の表に、このマニュアルにおける新機能および変更された機能の要約、および各機能がサポートされているリリースに関する情報を示します。

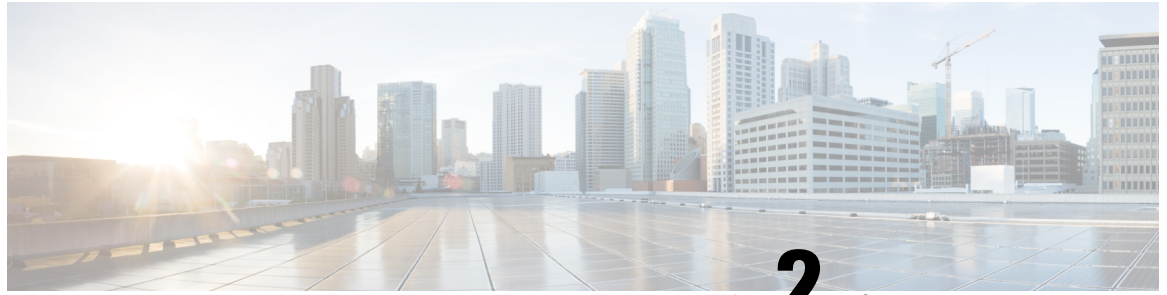
ご使用のソフトウェアリリースで、本書で説明されているすべての機能がサポートされているとは限りません。最新の警告および機能情報については、<https://tools.cisco.com/bugsearch/> のバグ検索ツール、およびご使用のソフトウェア リリースに関するリリース ノートを参照してください。

Table 1: 新機能および変更された機能

機能名	説明	リリース	参照先
AMC をリセット	AMC をリセットすることによる分析の中断のないリカバリのサポートが追加されました。	9.3(1)	SAN Analytics のトラブルシューティング, on page 108
Virtual Machine Identifier (VMID) Analytics	VMID Analytics 機能は、VM レベルでパフォーマンスの問題を監視、分析、特定、およびトラブルシューティングするために導入されています。	8.5(1)	SAN アナリティクスの設定, on page 7
SAN Analytics	Non-Volatile Memory Express (NVMe) 分析タイプのサポートが追加されました。	8.4(1)	SAN アナリティクスの設定, on page 7

機能名	説明	リリース	参照先
SAN Telemetry Streaming	NVMe フローメトリックを使用して、 <i>fabric_telemetry.proto</i> ファイルを更新しました。	8.4(1)	SAN Telemetry Streaming の構成, on page 111
Cisco MDS 9396T 32 Gbps 96 ポート ファイバチャネル ファブリックスイッチおよび Cisco MDS 9148T 32 Gbps 48 ポート ファイバチャネル ファブリックスイッチの SAN 分析サポート	SAN Analytics および SAN Telemetry Streaming 機能は、Cisco MDS 9396T 32 Gbps 96 ポート ファイバチャネル ファブリックスイッチおよび Cisco MDS 9148T 32 Gbps 48 ポート ファイバチャネル ファブリックスイッチでサポートされています。	8.4(1)	SAN アナリティクスの設定, on page 7
クエリ構文	昇順または降順でのメトリックおよびメタデータ フィールドのソートのサポートが追加されています。	8.3(2)	SAN アナリティクスの設定, on page 7
SAN Telemetry Streaming	コンパクト Google Protocol Buffers (GPB-Compact) エンコーディングのサポートが追加されています。	8.3(2)	SAN Telemetry Streaming の構成, on page 111
SAN Telemetry Streaming	SAN Telemetry Streaming 機能は、ストリーム分析やインターフェイスの統計情報の機能を Cisco Data Center Network Manager (DCNM) などの受信者に提供します。	8.3(1)	SAN Telemetry Streaming の構成, on page 111

機能名	説明	リリース	参照先
Cisco MDS 9132T 32-Gbps 32-Port Fibre Channel Switch の SAN Analytics のサポート	SAN Analytics および SAN Telemetry Streaming 機能は、Cisco MDS 9132T 32-Gbps 32-Port Fibre Channel Switch でサポートされています。	8.3(1)	SAN アナリティクスの設定, on page 7
Cisco N-Port Virtualizer (Cisco NPV) スイッチの SAN アナリティクスのサポート	SAN Analytics および SAN Telemetry Streaming 機能は、Cisco NPV モードで動作する Cisco MDS 9132T 32-Gbps 32-Port Fibre Channel Switch でサポートされています。	8.3(1)	SAN アナリティクスの設定, on page 7
Cisco MDS 9700 48-Port 32-Gbps Fibre Channel Switching Module の SAN アナリティクスのサポート	SAN アナリティクス機能は、Cisco MDS 9700 48-Port 32-Gbps Fibre Channel Switching Module でサポートされています。	8.2(1)	SAN アナリティクスの設定, on page 7
SAN アナリティクス	SAN アナリティクス機能を使用すると、サポートされている Cisco MDS スイッチにおけるパフォーマンスの問題を監視、分析、特定、およびトラブルシューティングできます。	8.2(1)	SAN アナリティクスの設定, on page 7



第 2 章

SAN アナリティクス ソリューション

- [SAN Analytics ソリューションの概要 \(5 ページ\)](#)

SAN Analytics ソリューションの概要

SAN Analytics ソリューションは、パフォーマンスの問題の監視、分析、特定、およびトラブルシューティングを可能にすることで、ユーザーのファブリックに関する情報を提供します。このソリューションは次のコンポーネントから構成されています。

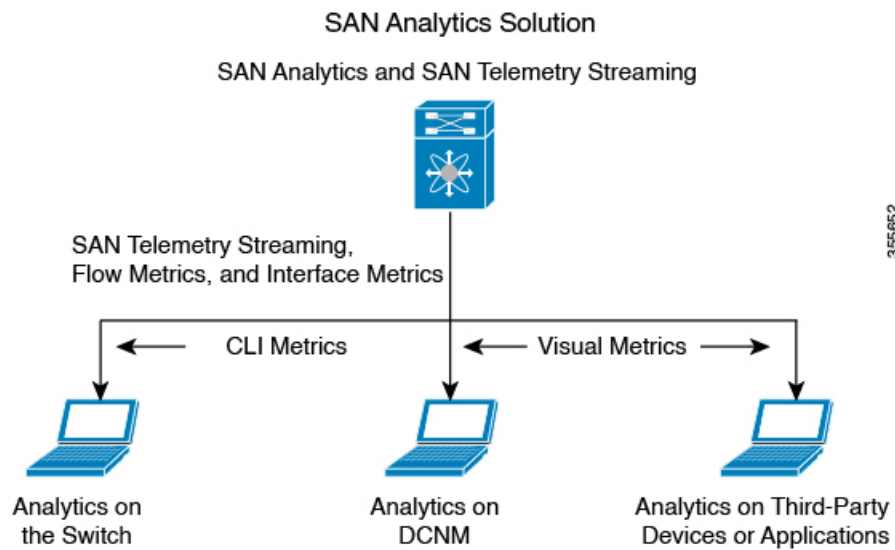
- **SAN Analytics** : SAN Analytics 機能は、スイッチポート上のデータフレームを検査して、パフォーマンスおよびエラーのメトリックを収集するために使用されます。また、SAN Analytics CLI を介してこれらのメトリックをスイッチ上で表示することもできます。
- **SAN Telemetry Streaming (STS)** : SAN Telemetry Streaming 機能は、Cisco Data Center Network Manager (DCNM) などの 1 つ以上のレシーバに、分析する特定のデータをストリーミングするために使用されます。

現在、次の 2 種類のデータのストリーミングがサポートされています。

- **フローメトリック - Small Computer System Interface (SCSI) および Non-Volatile Memory Express (NVMe) フローメトリック**。ファイバーチャネル交換の主要コンポーネントで構成されます。
- **インターフェイスメトリック - インターフェイスの統計情報**。
- **Cisco DCNM SAN Insights** : Cisco DCNM SAN Insights 機能は、分析する特定のデータを視覚的に表します。詳細については、[Cisco DCNM SAN 管理ユーザーガイド \[英語\]](#) を参照してください。
- **サードパーティ製のデバイスまたはアプリケーション** : 特定のデータを、サポートされているサードパーティ製のデバイス (Virtual Instruments 社の VirtualWisdom など) やアプリケーションで視覚的にストリーミングおよび分析することもできます。

次の図は、SAN Analytics ソリューションのワークフローを示しています。

図 1: SAN アナリティクス ソリューション





第 3 章

SAN アナリティクス の設定

この章では、SAN アナリティクス機能とその設定方法について説明します。

- [SAN Analytics の構成の機能履歴 \(7 ページ\)](#)
- [SAN アナリティクスの概要 \(10 ページ\)](#)
- [SAN アナリティクスのハードウェア要件 \(11 ページ\)](#)
- [SAN Analytics の注意事項と制約事項 \(12 ページ\)](#)
- [コマンドの変更 \(15 ページ\)](#)
- [SAN アナリティクスについて \(16 ページ\)](#)
- [SAN アナリティクスの設定 \(31 ページ\)](#)
- [スイッチでのメトリクスのクエリ \(37 ページ\)](#)
- [クエリの作成と使用 \(59 ページ\)](#)
- [ShowAnalytics オーバーレイ CLI の使用 \(78 ページ\)](#)
- [フローごとの輻輳ドロップの表示 \(98 ページ\)](#)
- [SAN アナリティクスの確認 \(99 ページ\)](#)
- [SAN Analytics のトラブルシューティング \(108 ページ\)](#)

SAN Analytics の構成の機能履歴

表 2: SAN Analytics の構成の機能履歴

機能名	リリース	機能情報
AMC をリセット	9.3(1)	AMC をリセットすることによる分析の中断のないリカバリのサポートが追加されました。
SAN Analytics	9.2(2)	Cisco MDS 9700 48-Port 64-Gbps Fibre Channel Switching Module がサポートされているハードウェアの一覧に追加されました。一部のフローメトリックが追加され、一部のフローメトリックが非推奨になりました。詳細については、 付録 (133 ページ) を参照してください。

機能名	リリース	機能情報
Virtual Machine Identifier (VMID) Analytics	8.5(1)	VMID Analytics 機能は、VM レベルでパフォーマンスの問題を監視、分析、特定、およびトラブルシューティングするために導入されました。 analytics vm-tag veid コマンドが導入されました。
SAN Analytics	8.5(1)	NVMe トラフィックの分析は、IO フレームのみをカウントするように変更されました。以前は、管理フレームも含まれていました。
ShowAnalytics オーバーレイ CLI の使用	8.5(1)	ShowAnalytics コマンドの --appendfile および --outfile オプションが追加されました。 ShowAnalytics --help コマンドの出力が変更されました。
ShowAnalytics オーバーレイ CLI の使用	8.4(2)	ShowAnalytics コマンドとそのオプションのコマンドキーワードと変数を一覧表示するオプションが追加されました。 ShowAnalytics コマンドに Non-Volatile Memory Express (NVMe) メトリックのサポートが追加されました。
ShowAnalytics オーバーレイ CLI の使用	8.4(1a)	ShowAnalytics コマンドの --top オプションの --alias 引数を追加しました。
SAN Analytics	8.4(1)	NVMe 分析タイプのサポートが追加されました。 新しい NVMe ビューインスタンスとフローメトリックが追加されました。詳細については、 フローメトリック (133 ページ) を参照してください。 次のコマンドが変更されました。 <ul style="list-style-type: none"> • fc-all および fc-nvme キーワードが [no] analytics type {fc-all fc-nvme fc-scsi} コマンドに追加されました。 • type fc-scsi キーワードが show analytics flow congestion-drops [vsan number] [module number port number] コマンドから削除されました。 • ShowAnalytics コマンドに --erroronly、--evaluate-npload、--minmax、--outstanding-io、--top、--vsan-thput、--alias、--limit、--key、--module、--progress、および --refresh オプションを追加しました。 SCSI および NVMe 分析タイプのスキーマを表示するために、 show analytics schema {fc-nvme fc-scsi} {view-instance instance-name views} コマンドが導入されました。

機能名	リリース	機能情報
クエリ構文	8.4(1)	NVMe 分析タイプのサポートが追加されました。 次のクエリ構文は、 <i>fc-nvme</i> 分析タイプをサポートしています。 select all column1[, column2, column3, ...] from analytics_type.view_type [where filter_list1 [and filter_list2 ...]] [sort column [asc desc]] [limit number]
SAN Analytics	8.4(1)	次のコマンドの出力は変更されました。 <ul style="list-style-type: none"> • show analytics port-sampling module number • show analytics system-load • ShowAnalytics
SAN Analytics	8.4(1)	サポートされているハードウェアのリストに Cisco MDS 9396T 32 Gbps 96 ポート ファイバチャネル ファブリック スイッチおよび Cisco MDS 9148T 32 Gbps 48 ポート ファイバチャネル ファブリック スイッチを追加しました。
クエリ構文	8.3(2)	昇順または降順でのメトリックおよびメタデータ フィールドのソートのサポートが追加されました。 asc オプションと desc オプションがクエリ構文に追加されました。 select all column1[, column2, column3, ...] from analytics_type.view_type [where filter_list1 [and filter_list2 ...]] [sort column [asc desc]] [limit number] show analytics system-load コマンドが導入されました。
SAN Analytics	8.3(1)	次のコマンドが導入されました。 no analytics name query_name Cisco MDS NX-OS リリース 8.2(1) から Cisco MDS NX-OS リリース 8.3(1) で変更されたコマンドについては、 表 4: コマンドの変更 (16 ページ) を参照してください。
ポート サンプリング	8.3(1)	ポート サンプリング機能を使用すると、監視対象モジュール内のポートのサブセットからデータを収集し、ポートの複数のサブセットを循環し、定期的なポート サンプリング間隔でそれらのポートからのデータをストリーミングできます。 次のコマンドが導入されました。 <ul style="list-style-type: none"> • analytics port-sampling module number size number interval seconds • show analytics port-sampling module number

機能名	リリース	機能情報
SAN Analytics	8.3(1)	いくつかのフローメトリックが導入されました。詳細については、「 フローメトリック (133ページ) 」を参照してください。
Cisco MDS 9132T 32-Gbps 32-Port Fibre Channel Switch の SAN Analytics のサポート	8.3(1)	Cisco MDS 9132T 32-Gbps 32-Port Fibre Channel Switch がサポートされているハードウェアの一覧に追加されました。
Cisco N-Port Virtualizer (Cisco NPV) スイッチの SAN Analytics のサポート	8.3(1)	Cisco NPV スイッチで SAN Analytics 機能を使用する際の注意事項と制約事項が追加されました。
SAN Analytics	8.2(1)	Cisco MDS 9700 48-Port 32-Gbps Fibre Channel Switching Module がサポートされているハードウェアの一覧に追加されました。
SAN Analytics	8.2(1)	<p>SAN Analytics 機能を使用すると、Cisco MDS 9000 シリーズ マルチレイヤ スイッチにおけるパフォーマンスの問題を監視、分析、特定、およびトラブルシューティングできます。</p> <p>次のコマンドが導入されました。</p> <ul style="list-style-type: none"> • analytics type fc-scsi • analytics query “query_string” type timer timer_val • clear analytics “query_string” • feature analytics • purge analytics “query_string” • ShowAnalytics • show analytics {query {“query_string” id result} type fc-scsi flow congestion-drops [vsan number] [module number port number]}

SAN アナリティクスの概要



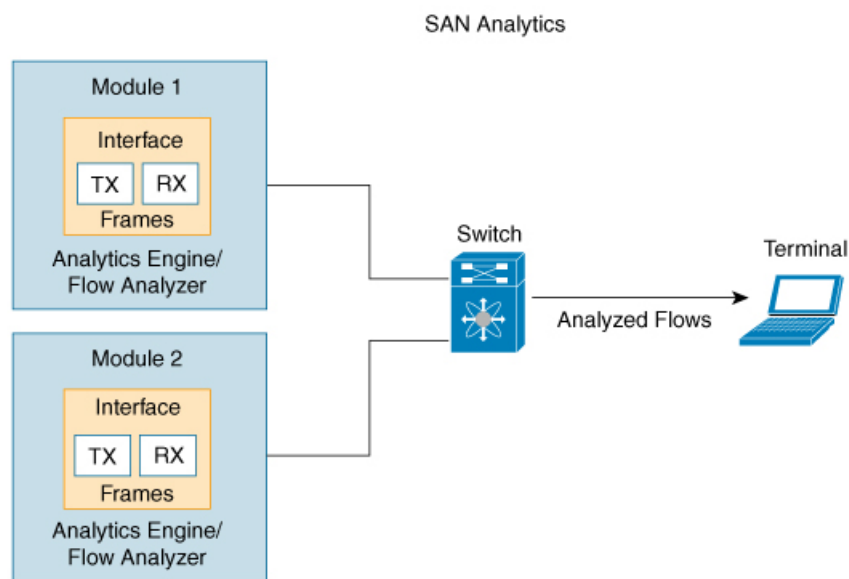
(注) Cisco MDS NX-OS リリース 8.3(1) またはそれ以降のリリースでは、SAN Analytics 機能を使用することをお勧めします。

SANアナリティクス機能を使用すると、Cisco MDS スイッチにおけるパフォーマンスの問題を監視、分析、特定、およびトラブルシューティングできます。サポートされているスイッチの一覧については、[SAN アナリティクスのハードウェア要件 \(11 ページ\)](#) を参照してください。

ファイバチャネル SAN 環境では、すべてのデバイスのパフォーマンスをプロビジョニングおよび監視して、それらのデバイスのパフォーマンスの妨げになる問題を解決可能にすることが重要です。SANアナリティクス機能は、フローを双方向で監視し、モジュールまたは個々のスイッチ内でネットワーク プロセッサ ユニット (NPU) のフローを関連付けて、完全に分析されたネットワーク データをユーザーに提供します。

次の図は、SAN アナリティクス機能の各機能を示しています。

図 2: SAN アナリティクスの概要



SAN アナリティクスのハードウェア要件

次の表は、SAN アナリティクス機能をサポートする Cisco MDS ハードウェアの一覧です。

表 3: サポートされているハードウェアの一覧

スイッチ	モジュール
Cisco MDS 9700 シリーズ マルチレイヤ ディレクタ	<ul style="list-style-type: none"> • Cisco MDS 9700 48-Port 32-Gbps Fibre Channel Switching Module (DS-X9648-1536K9) • Cisco MDS 9700 48-Port 64-Gbps Fibre Channel Switching Module (DS-X9748-3072K9)

Cisco MDS 9396T 32-Gbps 96-Port Fibre Channel Fabric Switch	<ul style="list-style-type: none"> • 96 x 32-Gbps 固定ポート • 32-Gbps Fibre Channel Expansion Module (M9XT-FC1632)
Cisco MDS 9148T 32-Gbps 48-Port Fibre Channel Fabric Switch	<ul style="list-style-type: none"> • 96 x 32-Gbps 固定ポート
Cisco MDS 9132T 32-Gbps 32-Port Fibre Channel Fabric Switch	<ul style="list-style-type: none"> • 32 Gbps 固定ポート X 16 • 16-Port 32-Gbps Fibre Channel Expansion Module (M9XT-FC1632)

SAN Analytics の注意事項と制約事項

- 次のような場合、この機能は VSAN ではサポートされません。
 - default zone permit が設定されている場合。
 - VSAN 間ルーティング (IVR) または Cisco MDS 9000 Input/Output Accelerator (IOA) 機能が有効になっている場合。
 - 相互運用性モードが有効になっている場合。
 - インオーダー デリバリ (IOD) が有効になっている場合。
- この機能には、次の制約事項があります。
 - プッシュ クエリの最大数は 8 です。プッシュ クエリの詳細については、[SAN アナリティクスについて \(16 ページ\)](#) を参照してください。
 - 個々のメトリックのクリアや消去はサポートされていません。メトリックのクリアおよび消去の詳細については、[SAN アナリティクスについて \(16 ページ\)](#) を参照してください。
 - クエリ構文の where 条件には、等号 (=) 演算子のみ使用できます。詳細については、「[クエリ構文 \(37 ページ\)](#)」を参照してください。
- Cisco Nexus スイッチおよび Cisco UCS ファブリック インターコネクト (SAN ポート チャネル) に接続されているポート チャネルのメンバーであるポートでは、**analytics type** コマンドを設定しないことをお勧めします。これにより、欠落した誤ったメトリックが表示されるのを回避できます。
- Cisco NPV モードで動作しているスイッチの場合、NX-OS による自動ロード バランシングまたはユーザによる手動リバランシングのいずれかによって、サーバログインがあるアップリンクから別のアップリンクに移動すると、**show analytics system-load** コマンド出力にそのスイッチの誤った ITL カウントが表示される場合があります。これは、自動負分散デバイスが別のアップストリーム リンク経由で再度ログインする必要がある場合に発

生じます。その場合、新しい FCID が割り当てられます。古い分析デバイスの FCID メトリックは自動的に削除されないため、これらの古いエントリにより、追加の ITL カウントが発生します。 **show analytics system-load** コマンドを使用して正しいデータを取得する前に、最初に **purge analytics** 「*query_string*」 コマンドを使用してメトリックをパーージする必要があります。

- VMID 分析機能を最初に有効にした後、**show analytics system-load** コマンド出力に誤った ITL カウントが表示されます。正しい ITL カウントを取得するには、**purge analytics “select all from fc-scsi.port”** コマンドを使用して正しいデータを取得する前に、まず **show analytics system-load** コマンドを使用してメトリックをパーージする必要があります。
- Cisco MDS 48 ポート 64 Gbps ファイバチャネル スイッチング モジュール (DS-X9748-3072K9) で、分析エンジンポートセットの ITL レコードの総数が 4000 を超えると、不完全な分析データが報告されることがあります。ポートセット内のフロー数を減らしてこの問題を回避する方法については、[分析エンジンポートセットのマッピング \(24 ページ\)](#) セクションを参照してください。

この制限を超えると、次の syslog が表示されます。

```
%ANALYTICS_LC_MGR-SLOT1-4-ANALYTICS_LC_MGR_4K_ITL_LIMIT_HIT: Analytics data may be incomplete on few ports : Affected ports are fc1/5,fc1/1,fc1/7,fc1/3
```

- ポートチャネルの一部であるインターフェイスでの分析サポートは、全体的な分析スケールの数値に影響を与える可能性があります。
- クエリ構文の **select all** オプションは、VMID メトリックを表示しません。VMID メトリックを表示するには、クエリ文字列で 1 つ以上の個別のメトリックを指定し、*vmid* キーを含める必要があります。たとえば、**show analytics query "select port,vsan,app_id,vmid,target_id,initiator_id,lun, active_io_read_count,active_io_write_count from fc-scsi.scsi_initiator_itl_flow"** のようになります。
- この機能を DCNM (またはサードパーティ製のデバイスやアプリケーション) とともに使用する場合は、Network Time Protocol (NTP) を同期させる必要があります。NTP の詳細については、[Cisco MDS 9000 シリーズ基本コンフィギュレーションガイド \[英語\]](#) の「Configuring NTP」セクションを参照してください。
- この機能は、SD ポートと呼ばれる Switched Port Analyzer (SPAN; スイッチドポートアナライザ宛先ポート、および NP ポートではサポートされていません。この機能のあるインターフェイスの範囲で有効にする場合は、そのインターフェイスの範囲に SD ポートや NP ポートが存在しないことを確認してください。存在する場合、この機能はいずれのインターフェイスでも有効になりません。
- この機能は、標準ベースのコマンドを含むフレームのみを分析します。Cisco MDS NX-OS リリース 8.2(x) およびリリース 8.3(x) では、ファイバチャネルプロトコル (FCP) SCSI 読み取りおよび書き込みコマンドがサポートされています。Cisco MDS NX-OS リリース 8.4(1) 以降、ファイバチャネル SCSI とファイバチャネル Non-Volatile Memory Express (NVMe) の両方の読み取りおよび書き込みコマンドがサポートされます。この機能は、独自のコマンドを含むフレームを分析しません。これらは通常、ストレージレプリケーションテクノロジーで使用されます。

- **feature analytics** コマンドが Cisco MDS NX-OS リリース 8.2(1) またはリリース 8.3(1) で有効になっている場合、Cisco MDS NX-OS リリース 8.2(1) とリリース 8.3(1) 間でのアップグレードやダウングレードは、アップグレードやダウングレードの前に **no feature analytics** コマンドを使用してこの機能を無効にし、その後 **feature analytics** コマンドを使用してこの機能を再度、有効にした場合にのみサポートされます。

Cisco MDS NX-OS リリース 8.3(1) 以降のリリースからリリース 8.2(1) にダウングレードした場合、この機能は、不具合 [CSCvm19337](#) に記載されている回避策を実行した後にのみ機能します。

- アップグレード、ダウングレード、スイッチのリロード、またはモジュールのリロード後は、すべてのフローメトリックが消去されます。
- スイッチがソフトゾーニングモードの場合、この機能はサポートされません。
- ストリーミング サンプル間隔 (**snshr-grp id sample-interval interval**)、ポート サンプリング間隔 (**analytics port-sampling module number size number interval seconds**)、およびプッシュクエリ間隔 (**analytics query "query_string" name query_name type periodic [interval seconds] [clear] [differential]**) は、同じ値に設定することをお勧めします。また、最初にプッシュクエリ間隔、次にポート サンプリング間隔、最後にストリーミング サンプル間隔を変更または設定することをお勧めします。



注意

- ストリーミング サンプル間隔、ポート サンプリング間隔、およびプッシュクエリ間隔は、最小推奨値の 30 秒以上に設定することをお勧めします。最小値未満の間隔を設定すると、望ましくないシステム動作が発生する可能性があります。
- モジュールごとのサポートされる Initiator-Target-LUN (ITL) の最大数については、[Cisco MDS NX-OS の設定の制限、リリース 8.x](#) の文書を参照してください。

アクティブな ITL 数が記載されている制限を超えると、syslog メッセージが記録されます。制限を長時間超えている場合、スイッチの安定性に影響が出る可能性があります。 **show analytics system-load** コマンドを使用して、ITL 数と NPU の負荷を確認します。詳細については、[Cisco MDS 9000 ファミリーおよび Nexus 7000 シリーズ NX-OS システム メッセージリファレンス ガイド \[英語\]](#) および [Cisco MDS NX-OS の設定の制限、リリース 8.x](#) の文書を参照してください。

- ネットワーク プロセッサ ユニット (NPU) のキャパシティの超過および超過に伴う結果を回避するためには、ポート サンプリング機能を使用してフローメトリックを分析します。詳細については、[ポート サンプリング \(22 ページ\)](#) を参照してください。

- ビュー インスタンスとその関連メトリックを消去後は、数秒待機してから、プル クエリを実行することをお勧めします。これは、消去操作が完了するまでは、フローメトリックの一部のフィールドに無関係な値が含まれていることがあるためです。
- NVMe 分析は、Fibre Channel Non-Volatile Memory Express - 1 (FC-NVMe-1) および FC-NVMe-2 標準と互換性があります。
- この機能は、ポート単位ですべてのフローメトリックを追跡します。フローの要求と応答が1台のスイッチの異なる物理ポートにまたがっている場合、一部のフローメトリックが正確に計算されない可能性があります。この条件は特に、Inter-Switch Link (ISL) ポート (E ポート) でこの機能が有効になっている場合に発生します。

以下に、要求の応答を異なる ISL ポートで確認できるシナリオを示します。

- ユーザーが **vsan ID loadbalancing src-dst-id** コマンドを使用して、ロードバランシング方式を Source ID (SID) -Destination ID (DID) に変更している場合。
 - ユーザーが **switchport trunk mode off** コマンドを使用して、ISL (E ポート) を非トランキングモードに設定している場合。
 - ポートチャネルの一部である ISL (E ポート) とポートチャネルが **no channel mode active** コマンドを使用してアクティブモードに設定されていない場合。
 - この機能は、非トランク ISL またはポートチャネルでは機能しません。この機能を E ポートで機能させるには、E ポートのトランクモードをオンにする必要があります。
 - ISL は、ポートチャネルの一部になるようにバンドルされていません。つまり、ECMP ISL および ECMP ポートチャネルはサポートされません。
 - Cisco MDS 9250i マルチサービス ファブリック スイッチまたは Cisco MDS 9148S 16-G マルチレイヤ ファブリック スイッチ、および Cisco MDS 9700 48-Port 32-Gbps Fibre Channel Switching Module (DS-X9648-1536K9) 間にポートチャネルが存在する場合。
- この機能は、FICON 対応の Cisco MDS 9000 スイッチではサポートされていません。

コマンドの変更

一部のコマンドは、Cisco MDS NX-OS リリース 8.3(1) で変更されました。このマニュアルでは、Cisco MDS NX-OS リリース 8.3(1) で導入または変更されたコマンドを表示しています。Cisco MDS NX-OS リリース 8.2(1) で使用されている同等のコマンドについては、[表 4: コマンドの変更 \(16 ページ\)](#) を参照してください。

Cisco MDS NX-OS リリース 8.3(1) 以降のリリースでは、SAN Analytics 機能を使用することをお勧めします。

[表 4: コマンドの変更 \(16 ページ\)](#) に、Cisco MDS NX-OS リリース 8.3(1) のコマンドに加えられた変更を示します。

表 4: コマンドの変更

Cisco MDS NX-OS リリース 8.2(1)	Cisco MDS NX-OS リリース 8.3(1)
<code>analytics query "query_string" type timer timer_val</code>	<code>analytics query "query_string" name query_name type periodic [interval seconds] [clear] [differential]</code>
<code>clear analytics "query_string"</code>	<code>clear analytics query "query_string"</code>
<code>purge analytics "query_string"</code>	<code>purge analytics query "query_string"</code>
<code>show analytics query {"query_string" id result}</code>	<code>show analytics query {"query_string" [clear] [differential] all name query_name result}</code>

SAN アナリティクスについて

SAN アナリティクス機能は、データ分析のために特定のフレームを使用してフロー メトリックを収集します。対象には次のコンポーネントが含まれます。

- データ収集：フロー データは NPU から収集され、最終的にスイッチのスーパーバイザに送信されて保存されます。表示されるデータはデータのリアルタイムビューで、履歴データは表示されません。
- オンボードクエリ：プルクエリ、プッシュクエリ、またはオーバーレイ CLI を使用して、データベースに保存されているデータを抽出できます。クエリは、データベースから特定のフローメトリックを抽出するために使用されます。特定のフレームは、スイッチのパフォーマンスの問題を監視、分析、およびトラブルシューティングするために使用されます。詳細については、「[クエリの作成と使用 \(59 ページ\)](#)」を参照してください。

以下は、データベースのさまざまなクエリ方法です。

- プルクエリとは、クエリの実行時にデータベースに保存されているフロー情報を抽出するために使用されるワンタイムクエリです。出力は JSON 形式です。プルクエリは NX-API に準拠しています。

オーバーレイ CLI `ShowAnalytics` コマンドは、フローメトリックが使いやすい表形式で表示される定義済みのプルクエリを発行する python スクリプトです。これは Python で記述されている CLI ラッパーで、実行に備えてブートフラッシュに保存されています。

Cisco MDS NX-OS リリース 8.3(1) 以降は、次のオプションがプルクエリでサポートされています。

- [クリア (Clear)]：すべての最小、最大、およびピークフローのメトリックをクリアします。
- [差分 (Differential)]：前回のストリーミング間隔と現在のストリーミング間隔の間に更新された ITL または ITN フローメトリックのみの絶対値を返します。

スイッチのスケール値を向上させるために、差分プッシュクエリを使用することをお勧めします。

- [プッシュクエリ (Push query)] : データベースに保存されているフローメトリックを定期的に抽出し、宛先に送信するためにインストールされている繰り返しクエリ。出力は JSON 形式です。

Cisco MDS NX-OS リリース 8.3(1)以降は、次のオプションがプッシュクエリでサポートされています。

- [クリア (Clear)] : すべての最小、最大、およびピークフローのメトリックをクリアします。
- [差分 (Differential)] : 前回のストリーミング間隔と現在のストリーミング間隔の間に更新された ITL または ITN フローメトリックのみの絶対値を返します。スイッチのスケール値を向上させるために、差分プッシュクエリを使用することをお勧めします。

プッシュクエリでは、次のフローメトリックの抽出モードがサポートされています。

- 連続モード : データは、すべての分析対応ポートで継続的に収集されます。
- サンプルングモード : データは、設定されているポート サンプルング間隔ごとに分析対応ポートのサブセットで収集され、その後、データ収集メカニズムが次のポートのサブセットを循環します。たとえば、データは、30 秒のポート サンプルング間隔で、24 の分析対応ポート内の 6 ポートのグループで収集されます。詳細については、[ポート サンプルング \(22 ページ\)](#) を参照してください。

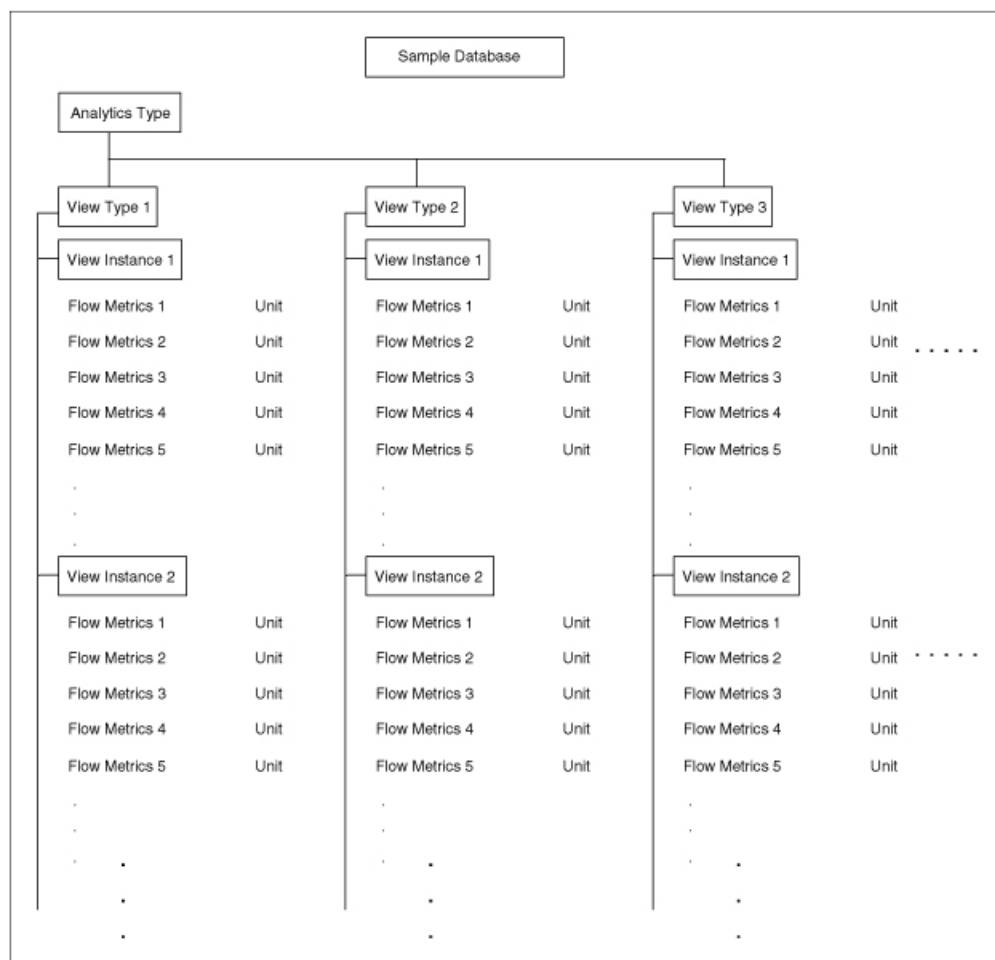
フローメトリックの保存に使用されるデータベースは、次の階層に従って編成されます。

- 分析タイプ : 分析するプロトコルタイプ。 *fc-scsi* 分析タイプは、Cisco MDS NX-OS リリース 8.2(x) および Cisco MDS NX-OS リリース 8.3(x) でサポートされています。 *fc-scsi* および *fc-nvme* 分析タイプは、Cisco MDS NX-OS リリース 8.4(1) からサポートされています。
- ビュー : ビューは、ポート、VSAN、イニシエータ、ターゲット、LUN、およびネームスペース ID パラメータの有効な組み合わせによって定義された、データベース内のフローメトリックの選択です。
- ビュータイプ : ビューは、フローを構成するコンポーネント (ポートビュー、 *initiator_IT* ビュー、 *target_ITL* ビューなど) に基づいて定義されます。クエリ構文は、あるビュータイプでクエリを実行するために使用されます。構文は、1つのビュータイプで1つのクエリのみサポートします。サポートされているビュータイプの一覧については、[サポートされているビュータイプの一覧 \(39 ページ\)](#) を参照してください。
- ビューインスタンス : 特定のビュータイプのインスタンスを指定します。ビューインスタンスには独自のフローメトリックがあります。たとえば、ポートビュータイプの場合、 *fc1/1* が1つのインスタンスで、 *fc1/2* が別のインスタンスのようになります。

- フローメトリック：分析に使用されるフローメトリックを指定します。Cisco MDS NX-OS 8.5(1)以降、NVMeトラフィックメトリックには、NVMeフレームの[カテゴリ (Category)] フィールドによって分類された IO フレームのみが含まれます。このリリースより前は、IO フレームと管理フレームの両方が含まれていました。サポートされているフローメトリックのリストについては、付録のフローメトリック (133 ページ) セクションのビュープロファイルを参照してください。

次の図は、サンプルデータベースのさまざまなコンポーネントを示しています。

図 3: サンプル データベース



クエリ構文の設定例については、例：クエリ構文の設定 (55 ページ) を参照してください。

次に、フローデータの収集ワークフローを示します。

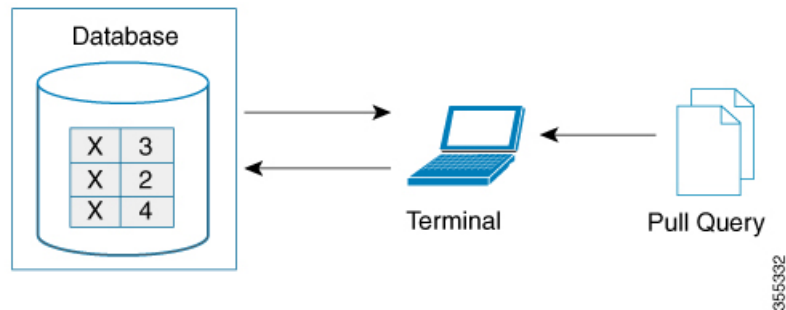
1. 機能の有効化：フローメトリックを分析する必要があるスイッチの SAN アナリティクス機能を有効にします。

2. インターフェイスの有効化：インターフェイスでのフローメトリックの収集を有効にします。ホストインターフェイスのSAN分析機能を有効にすることをお勧めします（[展開モード（25 ページ）](#) の図を参照）。
3. クエリの実行とインストール：データベースからフローメトリックを取得するために、次のクエリが使用されます。
 - プルクエリ：スイッチの問題を直接トラブルシューティングするために、ほぼリアルタイムのフローメトリックを提供します。プルクエリからのデータは、クエリに回答したときにデータベースから抽出されます。プルクエリは、CLI または NX-API を使用して実行できます。Cisco DCNM は NX-API を使用して、可視化用のデータを収集できます。

オーバーレイ CLI：フローメトリックが使いやすい表形式で表示される定義済みのプルクエリ。スイッチの問題を直接トラブルシューティングするために、ほぼリアルタイムのフローメトリックを提供します。

次の図は、プルクエリの機能を示しています。

図 4: プルクエリ



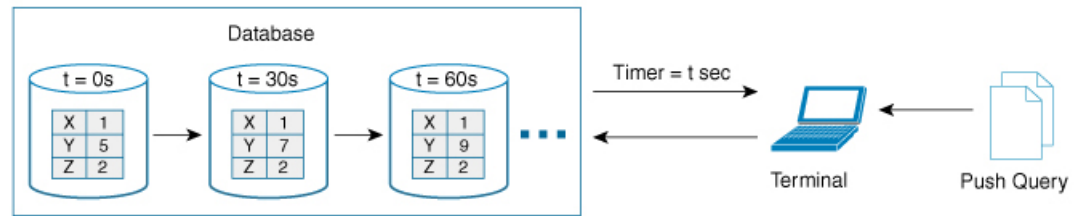
- プッシュクエリ：フローメトリックを定期的に提供します。時間間隔は秒単位で指定できます。指定された時間間隔を過ぎると、ユーザーに関する特定のフローメトリックが更新されて、データベースからプッシュされます。複数のクエリがインストールされている場合、各プッシュクエリがお互いに無関係にフローメトリックをプッシュします。これは想定されている動作です。



- (注)
- プルクエリ、プッシュクエリ、およびオーバーレイ CLI は、SAN アナリティクス機能が有効になっているインターフェイスでのみ適用されます。
 - プッシュクエリ タイマーは NPU からフローメトリックを取得し、指定されたプッシュクエリ間隔でスーパーバイザ上のデータベースに保存します。

次の図は、一定のメトリックだけが特定の間隔で更新されるように設定されているプッシュクエリの機能を示しています。

図 5: プッシュクエリ

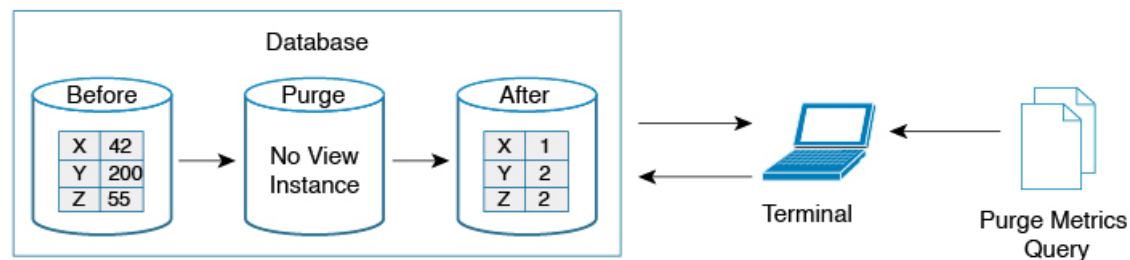


4. メトリックのクリアとリセット：次の機能を使用すると、データベース内の収集したフローメトリックをクリアまたはリセットできます。

- [ページ (Purge)]：指定されたビュー インスタンス、およびそのビュー インスタンスと関連付けられているすべてのメトリックを削除します。ビュー インスタンスは新しい IO ですぐに再構築され、すべてのビュー メトリックはゼロからカウントを開始します。このオプションを使用して、イニシエーターまたはターゲットがアクティブでないか存在しなくなった場合など、古いメトリックをビュー からフラッシュします。

次の図は、消去メトリック クエリの機能を示しています。

図 6: 消去メトリック クエリ



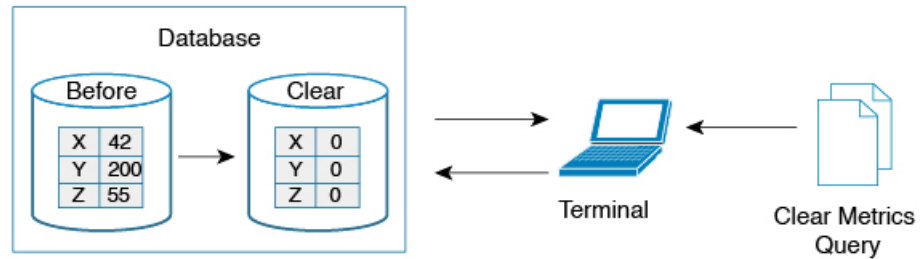
- クリア — タイプ キーのフローメトリックを除き、指定されたクエリ文字列に一致するすべてのメトリックをゼロにリセットします。データベースをクリアした後で、データベースは指定されたクエリのフローメトリックの収集を続けます。



- (注) **clear analytics query** コマンドは、プッシュクエリで使用されている **clear** オプションとは異なります。**clear analytics query** コマンドは、クエリ構文を満たすすべてのメトリックをリセットします。プッシュクエリで使用される **clear** オプションは、最小、最大、およびピークフローメトリックをリセットします。

次の図は、クリアメトリッククエリの機能を示しています。

図 7: クリア メトリック クエリ



355334

VMID Analytics



- (注) VMID Analytics 機能は現在、実稼働環境以外でのみ使用するためのベータ ステータスです。この機能を有効にする前に、アカウントチームまたは Cisco MDS マーケティング チームに連絡して、ユース ケースを理解してください。このベータ ステータスと制限は、今後のリリースで通常の製品ステータスに変更されます。

SAN Analytics 機能は、デバイス (FCID ごと) レベルでファイバチャネルトラフィック情報を提供します。ただし、エンドデバイスは複数の仮想エンティティ (仮想マシン [VM]) をホストでき、各 VM はファイバチャネルファブリックにさまざまな負荷を与える可能性があります。したがって、各 VM のファイバチャネルパフォーマンスを監視することが重要になります。VMID 分析機能を使用すると、VM レベルでファイバチャネルのパフォーマンスの問題を監視、分析、識別、およびトラブルシューティングできます。

特定のデバイス内の個々の VM は、SCSI および NVMe IO 交換に同じ FCID を使用します。NX-OS Virtual Machine Identifier (VMID) サーバー機能により、FCID ごとのデバイス レベルから個々の VM レベルまでのトラフィック ソースを解決できます。この機能の詳細については、『Cisco MDS 9000 Series Fabric Configuration Guide, Release 8.x』の「Managing FLOGI, Name Server, FDMI, and RSCN Databases」の章の「VMID」セクションを参照してください。

VMID サーバー機能を有効にした後、VMID Analytics 機能を有効にして、イニシエータのパフォーマンス メトリックを解決できます。有効にすると、イニシエーターレベルのメトリックを報告するために使用された分析ビューは、VMID レベルのメトリックも報告します。*scsi-initiator-id* または *nvme-initiator-id* キーを含むビュータイプのみが監視されます。これらのビュータイプでは、追加の *vmid* キーがサポートされています。VMID 固有の分析を収集するには、クエリ構文でイニシエーター ID とともに「選択されたフィールド」リストの一部として *vmid* キーを指定する必要があります。「選択されたフィールド」リストで VMID が指定されておらず、イニシエータ ID のみが指定されている場合、集約されたメトリックがイニシエータに対して収集されます。

VMID サーバー機能を無効にすると、接続されたデバイスは VMID 情報をファイバチャネルフレームに挿入しなくなります。また、VMID Analytics 機能が無効になっている場合、フレームは VMID ではなくソース FCID に対してカウントされます。ただし、Analytics データベース

は、以前に収集された VMID ごとのメトリックを引き続き保持します。データベースをリセットするには、メトリックを消去するか、無停止モジュールアップグレードを実行する必要があります。メトリックをバージしない場合、差分オプションを使用した場合と使用しない場合のプルまたはプッシュクエリの出力は次のようになります。

- VMID Analytics 機能を無効にした後にプルまたはプッシュクエリで差分オプションを使用すると、最初のプルまたはプッシュクエリにのみ、古い VMID ごとのメトリックが含まれます。
- VMID Analytics 機能を無効にした後にプルまたはプッシュクエリで差分オプションを使用しない場合、すべてのプルまたはプッシュクエリは古い VMID ごとのメトリックをフェッチします。

VMID Analytics 機能は、Cisco MDS NX-OS リリース 8.5(1) で導入されました。

ポート サンプリング

Cisco MDS NX-OS リリース 8.3(1) で導入されたポート サンプリング機能を使用すると、監視対象モジュール内のポートのサブセットからデータを収集し、ポートのさまざま n サブセットを循環し、定期的なポートサンプリング間隔でそれらのポートからのデータをストリーミングできます。

この機能は、NPU の負荷が高く、モジュール上の監視対象ポートの数を削減できない場合に便利です。そのような状況では、指定されたポートサンプリング間隔で監視対象ポートのサブセットをサンプリングすることで NPU の負荷を削減できます。 **show analytics system-load** コマンドを使用して、NPU の負荷を確認します。

Cisco MDS NX-OS リリース 8.3(2) では、NPU の負荷が高い場合、ITL 数がモジュールの制限を超えている場合、ITL 数がシステムの制限を超えている場合、NPU から分析データに関する応答がない場合にアラートを表示するシステム メッセージが導入されています。詳細については、[Cisco MDS 9000 ファミリーおよび Nexus 7000 シリーズ NX-OS システム メッセージ リファレンス](#) の文書を参照してください。

サンプリングされていない監視対象ポートで発生する I/O およびエラーは表示されず、分析データには含まれません。

この機能で使用されているポート サンプリング間隔は、ストリーミング サンプル間隔とは無関係です。ストリーミング サンプル間隔、ポート サンプリング間隔、およびプッシュクエリ間隔は、最小推奨値の 30 秒以上に設定することをお勧めします。

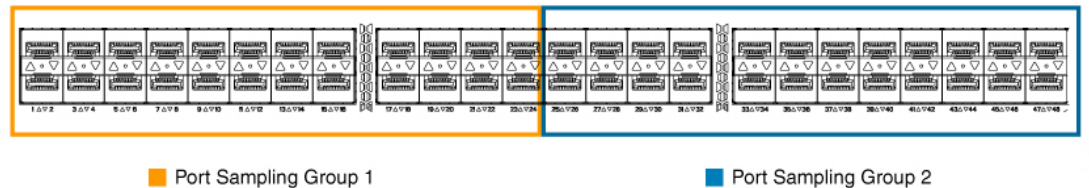


- (注) モジュールでこの機能が有効になっていて、後にそのモジュールの新しいポートで SAN Analytics 機能が有効になった場合、新しいポートのポート サンプリング データは次のポート サンプリング間隔後に初めてストリーミングされます。

ポート サンプリングのシナリオ

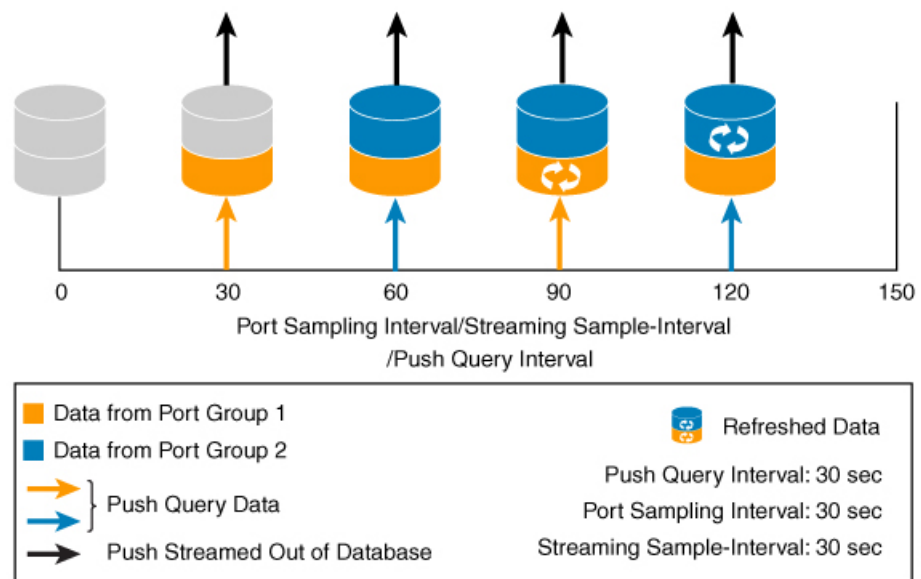
48 ポートで構成されているモジュールを、2つの24ポートのサブセットにグループ化するとします。それらのポートのサブセットに設定されているポートサンプリング間隔および設定されているストリーミングサンプル間隔に応じて、フローメトリックは異なる間隔でキャプチャされます（次の例を参照）。

図 8: ポート サンプリング グループ



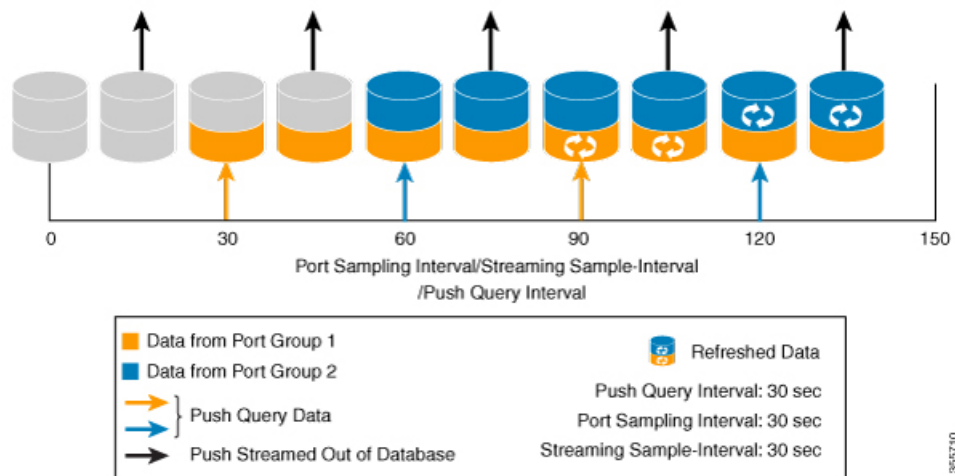
- ポート サンプリング間隔とストリーミング サンプル間隔が同時に開始される場合 :

図 9: 同時刻に始まるポート サンプリング間隔とストリーミング サンプル間隔



- ポート サンプリング間隔とストリーミング サンプル間隔の開始時刻が異なる場合 :

図 10:異なる時刻に始まるポートサンプリング間隔とストリーミングサンプル間隔



分析エンジンポートセットのマッピング

64 Gbps 対応ポート以降、モジュール上の分析データは、ポート ASIC によってポートのセットとして管理されます。各ポートセットからのデータは、専用のメモリブロックに保存されます。分析データの損失を回避するには、各ポートセットを介して監視される ITL フローの総数が、関連するブロックの容量を超えないようにする必要があります。

デバイス	監視間隔ごとの最大 ITL フロー
DS-X9748-3072K9	ポートセットあたり 4000 フロー

ポートセットごとの ITL フローの数が関連するブロックの容量を超える場合は、ポートセットの 1 つ以上のポートで分析を無効にするか、トラフィックを別のポートセットの物理ポートに移動することによって削減できます。

表 5: 分析エンジンポートセットのマッピング (25 ページ) はポートを分析エンジンのポートセットに分割します。

表 5: 分析エンジン ポート セットのマッピング

デバイス	分析エンジン ポート セット	前面パネルポート番号
DS-X9748-3072K9	1	9、11、13、15
	2	25、27、29、31
	3	10、12、14、16
	4	26、28、30、32
	5	1、3、5、7
	6	33、35、37、39
	7	2、4、6、8
	8	34、36、38、40
	9	17、19、21、23
	10	41、43、45、47
	11	18、20、22、24
	12	42、44、46、48

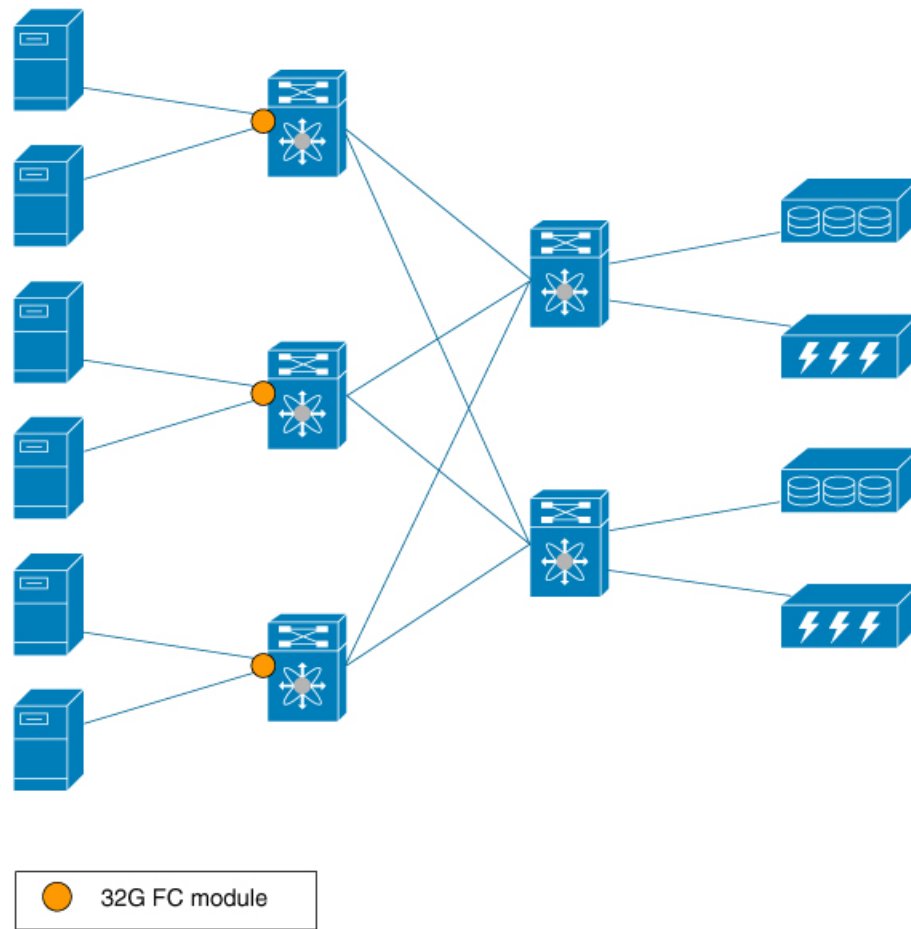
展開モード

SAN アナリティクス機能をサポートしているスイッチが SAN ファブリックで展開されている場所に応じて、次の展開モードが考えられます。

ホスト エッジ展開モード

SAN アナリティクス機能は、すべての Cisco MDS コア スイッチ、およびホストに接続されているインターフェイスで有効になっています。

図 11: ホストエッジ展開モード

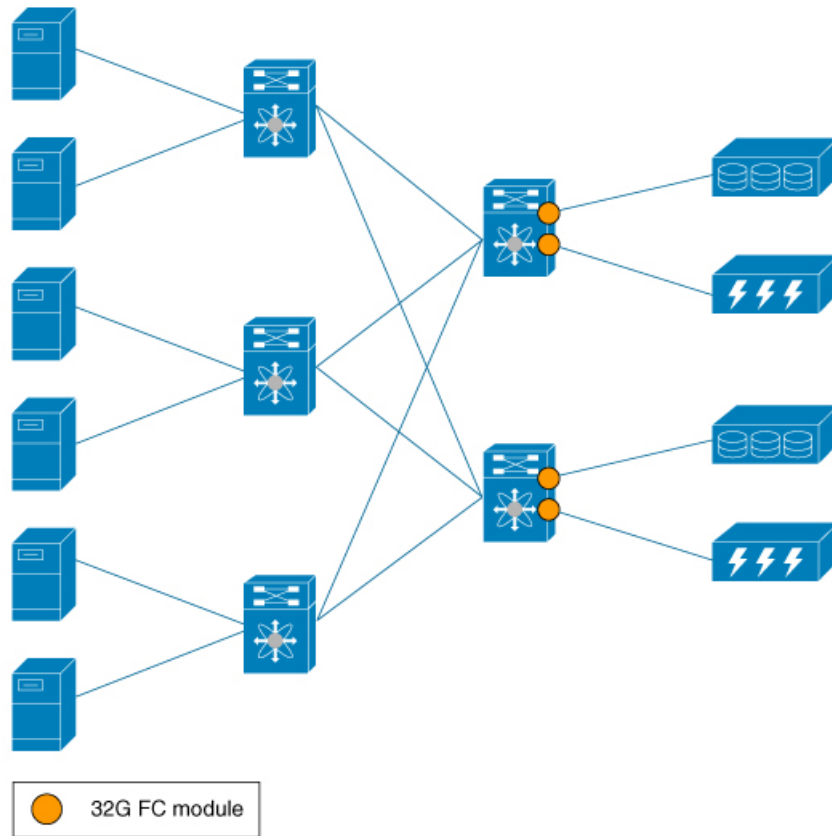


355745

ストレージエッジ展開モード

SAN アナリティクス機能は、すべての Cisco MDS コア スイッチ、およびストレージアレイに接続されているインターフェイスで有効になっています。

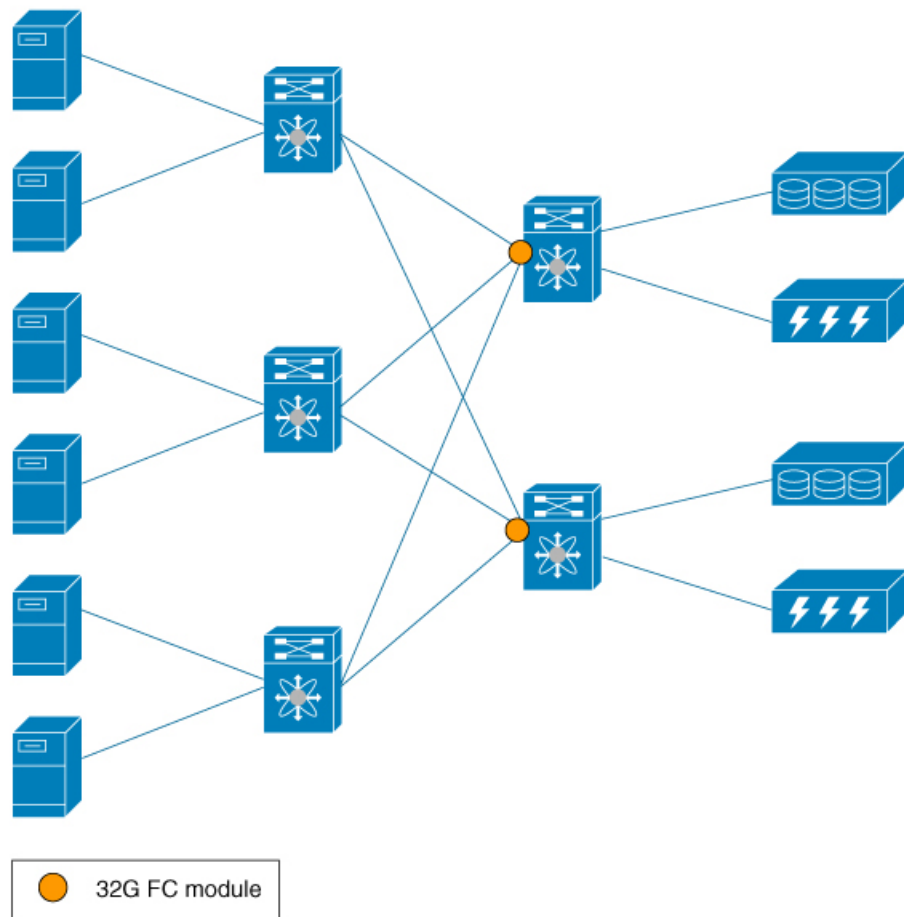
図 12:ストレージエッジ展開モード



ISL 展開モード

SAN アナリティクス機能は、すべての Cisco MDS コア スイッチ、および ISL のいずれかのサイドにあるインターフェイスで有効になっています。

図 13: ISL 展開モード

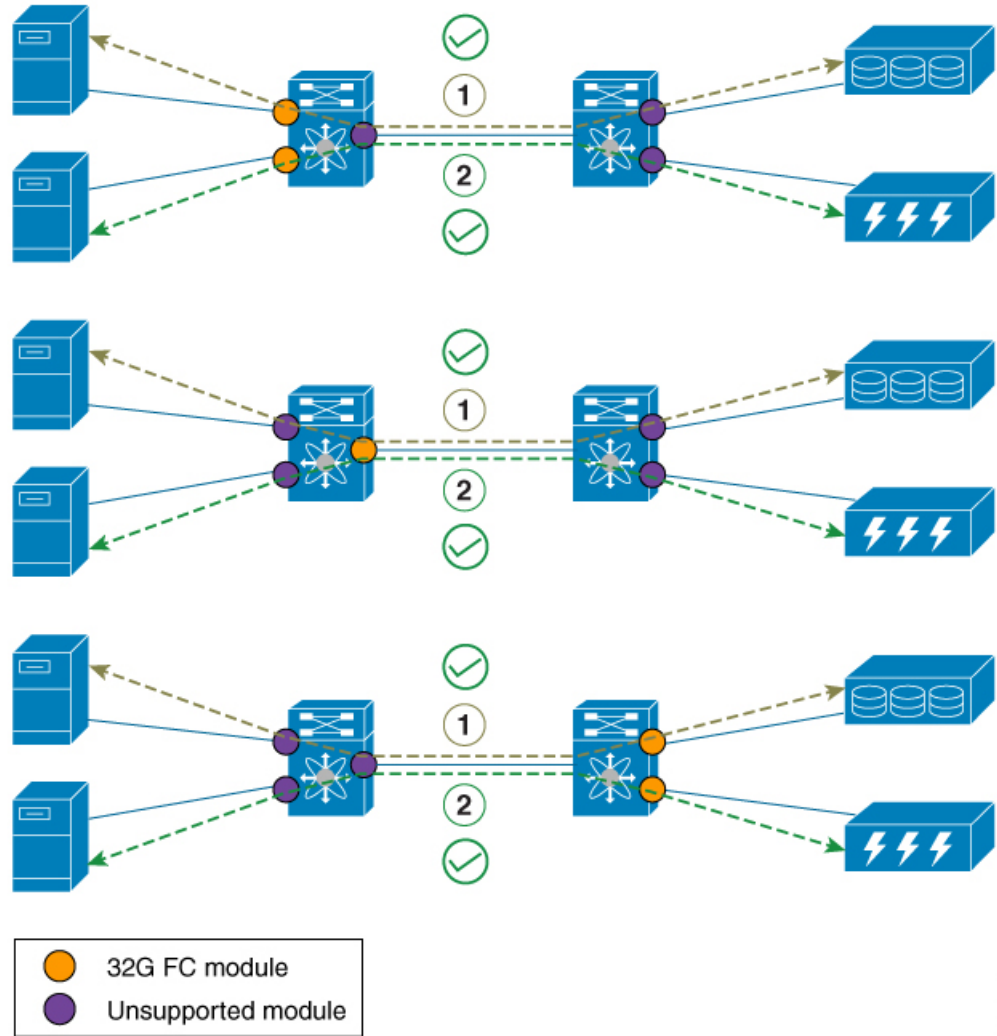


次の図は、サポート対象モジュールとサポート対象外モジュール（16 Gbps ファイバ チャネル、Cisco MDS 9700 40-Gbps 24-Port FCoE Module (DS-X9824-960K9)、Cisco MDS 24/10-Port SAN Extension Module (DS-X9334-K9) など）が SAN で使用されている場合の SAN アナリティクス機能の各機能を示しています。

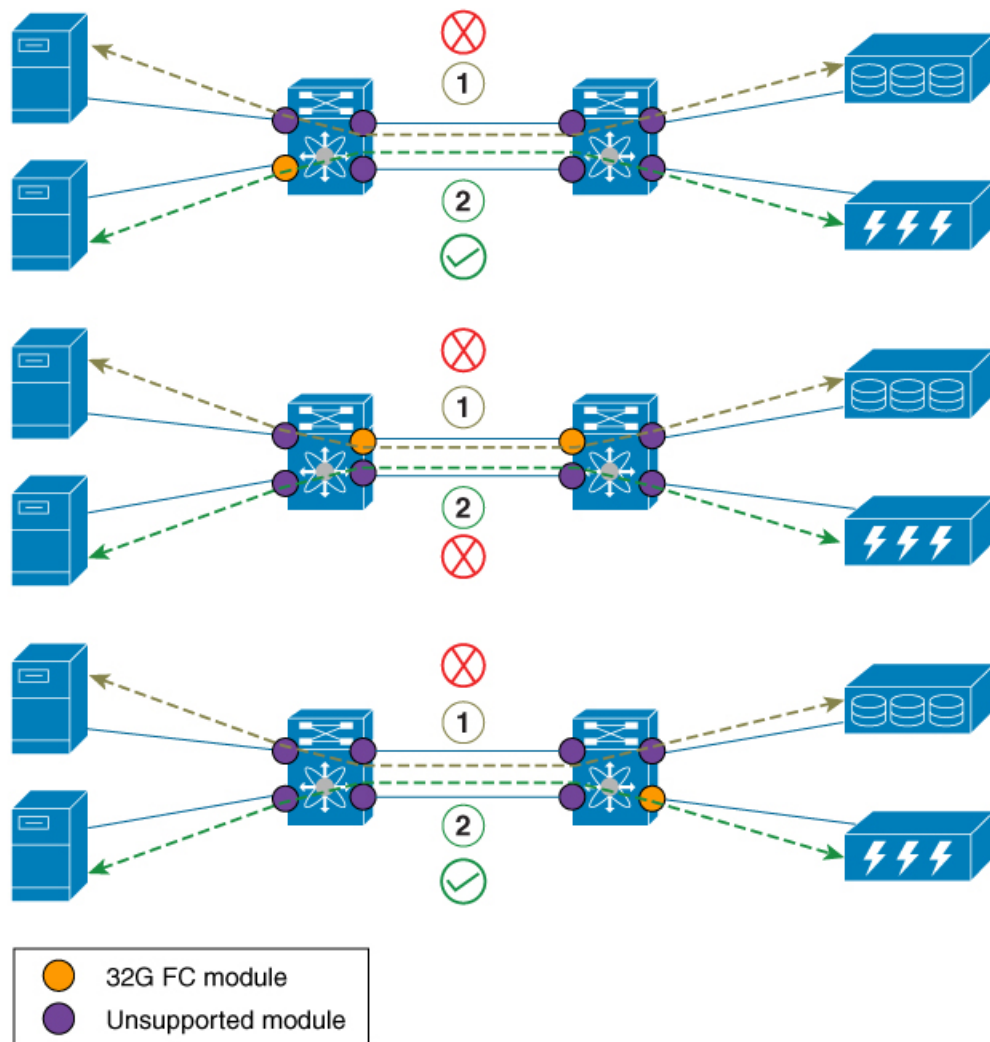


(注) 図 14: サポート対象モジュールとサポート対象外モジュールが使用されている場合の SAN アナリティクス機能の各機能の番号 1 と 2 は、それぞれイニシエータからターゲットに向かう 2 つの異なるフローを表しています。

図 14: サポート対象モジュールとサポート対象外モジュールが使用されている場合の SAN アナリティクス機能の各機能



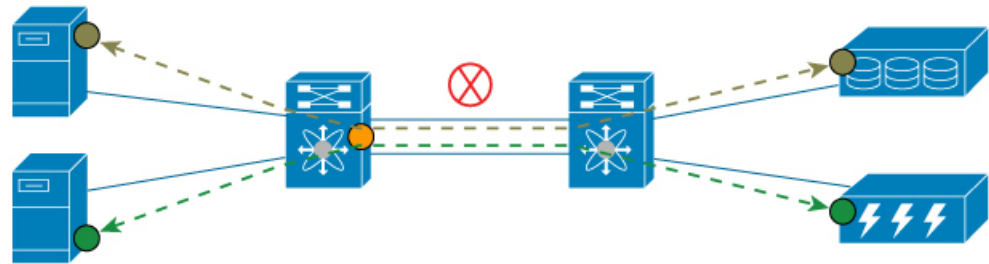
355339



355340

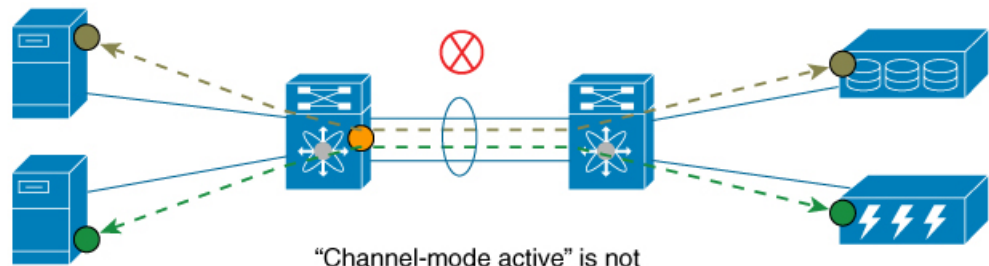


- (注)
- 前述の ISL モードのシナリオでは、要求の応答はポートチャネルの異なるメンバーで確認できます。
 - ISL でサポート対象モジュールとサポート対象外モジュールが使用されている場合、その ISL で分析された分析データは正確ではない可能性があります。そのため、サポート対象モジュールとサポート対象外モジュールが使用されている ISL ではデータを分析しないことをお勧めします。



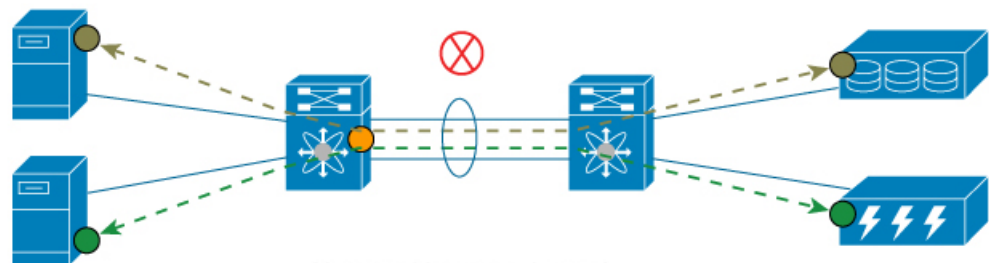
Not a port channel

355354



"Channel-mode active" is not configured on port channel

355355



Non-trunking port channel

355356

SAN アナリティクスの設定

インターフェイスからのフローメトリックの収集を有効にするには、スイッチとそのインターフェイスの両方で SAN アナリティクス機能を有効にします。



- (注)
- SAN アナリティクス機能を使用するには、**install license** コマンドを使用して適切なライセンス パッケージをインストールする必要があります。詳細については、[Cisco MDS 9000 シリーズ ライセンス ガイド \[英語\]](#) を参照してください。
 - Cisco DCNM SAN Insights を使用している場合は、Cisco DCNM SAN Insights で SAN アナリティクス機能を設定できます。スイッチでこの機能を設定する必要はありません。詳細については、[Cisco DCNM SAN 管理コンフィギュレーションガイド \[英語\]](#) の「Configuring SAN Insights」セクションを参照してください。

SAN アナリティクスの有効化



- (注)
- SAN アナリティクス機能は、デフォルトでは無効になっています。
 - アクティブな ITL 数が記載されている制限を超えると、Syslog メッセージが記録されません。

スイッチの SAN アナリティクス機能を有効にするには、次の手順を実行します。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 スwitchの SAN アナリティクス機能を有効にします。

```
switch(config)# feature analytics
```

SAN アナリティクスの無効化

スイッチの SAN アナリティクス機能を無効にするには、次の手順を実行します。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 スwitchの SAN アナリティクス機能を無効にします。

```
switch(config)# no feature analytics
```

インターフェイスでの SAN Analytics の有効化

インターフェイスの SAN Analytics 機能を有効にするには、次の手順を実行します。

始める前に



(注) SAN Analytics 機能は、デフォルトではすべてのインターフェイスで無効になっています。

- スイッチの SAN Analytics 機能を有効にします。 [SAN アナリティクスの有効化 \(32 ページ\)](#) の項を参照してください。
- ポート チャネルで、すべてのインターフェイスで SAN Analytics 機能を有効にします。

手順

ステップ 1 グローバル構成モードを開始します。

```
switch# configure terminal
```

ステップ 2 ファイバチャネル インターフェイスまたはインターフェイスの範囲を選択し、インターフェイス構成サブモードを開始します。

```
switch(config)# interface fc slot number/port number
```

(注) **interface fc slot number/port number - port number**、**fc slot number/port number - port number** コマンドを使用して、インターフェイスの範囲を指定することもできます。ダッシュ (-) とカンマ (,) の前後にはスペースが必要です。

ステップ 3 選択されたインターフェイスの SAN Analytics 機能を有効にします。

```
switch(config-if)# analytics type {fc-all | fc-nvme | fc-scsi}
```

(注) Cisco MDS NX-OS リリース 8.2(1) および Cisco MDS NX-OS リリース 8.3(1) では、**fc-scsi** 分析タイプのみサポートされています。Cisco MDS NX-OS リリース 8.4(1) 以降、**fc-scsi**、**fc-nvme**、および **fc-all** 分析タイプがサポートされます。

インターフェイスでの SAN アナリティクスの無効化

インターフェイスの SAN アナリティクス機能を無効にするには、次の手順を実行します。

始める前に

ポート チャネルで、すべてのインターフェイスの SAN アナリティクス機能を無効にします。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 ファイバ チャネル インターフェイスまたはインターフェイスの範囲を選択し、インターフェイス コンフィギュレーション サブモードを開始します。

```
switch(config)# interface fc slot number/port number
```

(注) **interface fc slot number/port number - port number**、**fc slot number/port number - port number** コマンドを使用して、インターフェイスの範囲を指定することもできます。ダッシュ (-) とカンマ (,) の前後にはスペースが必要です。

ステップ 3 選択されたインターフェイスで SAN アナリティクス機能を無効にします。

```
switch(config-if)# no analytics type {fc-all | fc-nvme | fc-scsi}
```

VMID Analytics の有効化

スイッチの VMID Analytics 機能を有効にするには、次の手順を実行してください。

始める前に

1. 接続されている HBA に VMID 機能をサポートするファームウェアがあり、その機能が HBA で有効になっていることを確認します。
2. スwitchの SAN アナリティクス機能を有効にします。[SAN アナリティクスの有効化 \(32 ページ\)](#) の項を参照してください。
3. インターフェイスで SAN Analytics を有効にします。「[インターフェイスでの SAN Analytics の有効化 \(33 ページ\)](#)」の項を参照してください。
4. VMID Server 機能を有効にします。『[Cisco MDS 9000 Series Fabric Configuration Guide, Release 8.x](#)』の「Managing FLOGI、Name Server、FDMI、および RSCN Databases」の章の「Enabling the VMID Server」の項を参照してください。

手順

ステップ 1 グローバル構成モードを開始します。

```
switch# configure terminal
```

ステップ2 スイッチの VMID Analytics 機能を有効にします。

```
switch(config)# analytics vm-tag veid
```

VMID Analytics の無効化

スイッチの VMID Analytics 機能を無効にするには、次の手順を実行してください。

手順

ステップ1 グローバル構成モードを開始します。

```
switch# configure terminal
```

ステップ2 スイッチの VMID Analytics 機能を無効にします。

```
switch(config)# no analytics vm-tag veid
```

ポート サンプリングの有効化



- (注)
- ポート サンプリングは、Cisco MDS NX-OS リリース 8.3(1) 以降のリリースでのみサポートされています。
 - ポート サンプリングはデフォルトでは無効になっており、継続的な監視はすべての分析対応ポートで有効になっています。ポート サンプリングの詳細については、[ポート サンプリング \(22 ページ\)](#) を参照してください。
-

モジュールのポート サンプリングを有効にするには、次の手順を実行します。

手順

ステップ1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ2 モジュールのポート サンプリングを有効にします。

```
switch# analytics port-sampling module number size number interval seconds
```

ポート サンプリングの無効化

モジュールのポート サンプリングを無効にするには、次の手順を実行します。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 モジュールのポート サンプリングを無効にして、設定済みのストリーミング サンプル間隔を指定して、すべての分析対応ポートの監視をデフォルト モードに戻します。

```
switch# no analytics port-sampling module number
```

例 : SAN Analytics の設定

次に、スイッチの SAN Analytics 機能を有効にする例を示します。

```
switch# configure terminal  
switch(config)# feature analytics
```

次に、スイッチの SAN Analytics 機能を無効にする例を示します。

```
switch# configure terminal  
switch(config)# no feature analytics
```

この例は、NVMe 分析タイプがすでに有効になっている場合に、SCSI 分析タイプのインターフェイスで SAN Analytics 機能を有効にする方法を示しています。

- この例は、NVMe 分析タイプがすでに有効になっていることを示しています。

```
switch# show running-config analytics  
  
!Command: show running-config analytics  
!Running configuration last done at: Wed Mar 13 09:01:56 2019  
!Time: Wed Mar 13 09:02:52 2019  
  
version 8.4(1)  
feature analytics  
  
interface fcl/1  
  analytics type fc-nvme
```

- 次の例は、単一のポートで SCSI 分析タイプを有効にする方法を示しています。

```
switch# configure terminal
```

```
switch(config)# interface fc 1/1
switch(config-if)# analytics type fc-scsi
```

- この例は、SCSI 分析タイプが有効になっていることを示しています。

```
switch# show running-config analytics

!Command: show running-config analytics
!Running configuration last done at: Wed Mar 13 09:01:56 2019
!Time: Wed Mar 13 09:02:52 2019

version 8.4(1)
feature analytics

interface fc1/1
  analytics type fc-scsi
  analytics type fc-nvme
```

スイッチでのメトリクスのクエリ

プルクエリ CLI を実行すると、指定されたメトリックがモジュールの NPU から収集され、スーパーバイザのメトリック データベースに保存されてから、ユーザー セッションに表示されます。

メトリックのクエリ用のスキーマ

スキーマは、データベースに保存されている特定のデータをユーザーに表示するために使用されます。スキーマの詳細については、**show analytics schema** コマンドを使用してください。メトリックは、さまざまなビューインスタンスの形式でデータベース内に保持されています。これらのビューインスタンスは、クエリを使用して取得できます。詳細については、「[ビュー \(39 ページ\)](#)」を参照してください。

クエリ構文

プッシュクエリ、プルクエリ、メトリックのクリア、ビューの消去で使用するクエリ構文を次に示します。

```
select all | column1 [, column2, column3, ...] from analytics_type.view_type [ where filter_list1 [ and filter_list2 ...]] [ sort column [asc | desc]] [ limit number]
```

クエリ構文の要素は次のとおりです。

- *analytics_type* : 分析のタイプを指定します。Cisco MDS NX-OS リリース 8.2(1) および Cisco MDS NX-OS リリース 8.3(1) では、*fc-scsi* タイプのみサポートされています。Cisco MDS NX-OS リリース 8.4(1) から、*fc-nvme* 分析タイプがサポートされます。

- **view_type** : メトリックデータベースのビュータイプを指定します。この構文はそのビュータイプでクエリを実行するために使用されます。構文は、1つのビュータイプで1つのクエリのみサポートします。サポートされているビュータイプの一覧とそれぞれの説明については、[サポートされているビュータイプの一覧 \(39 ページ\)](#) を参照してください。
- **column** : フロー メトリックを指定します。ビュー インスタンスには複数の列が含まれています。
- **filter_list** : ビュー インスタンスの特定のメトリックを抽出するフィルタを指定します。フィルタ条件は、タイプが **key** 値のフロー メトリック列、またはビュー インスタンス列で使用できます。フィルタリングに **AND** 演算子を使用することもできます。サポートされているビュータイプの一覧については、[サポートされているビュータイプの一覧 \(39 ページ\)](#) を参照してください。
- **sort** : 列内の結果をソートすることを指定します。ソートは**limit**操作の実行前に行われず。
- **asc** : 列の結果を昇順でソートします。デフォルトでは、順序が指定されていない場合、ソートは昇順で行われます。
- **desc** : 列の結果を降順でソートします。
- **limit** : 結果で返されるメトリクスの数を制限します。

クエリ構文の構成例については、[例：クエリ構文の設定 \(55 ページ\)](#) を参照してください。



- (注)
- "**query_string**" の **limit** および **where** オプションは、**key** フィールドでのみ使用できます。
 - Cisco MDS NX-OS リリース 8.3(2) より前は、"**query_string**" の **sort** オプションは **key** フィールドでのみ使用でき、メトリックは昇順でのみソートされていました。Cisco MDS NX-OS リリース 8.3(2) からは、"**query_string**" の **sort** オプションは、すべての **metrics** および **metadata** フィールドで使用でき、**asc** または **desc** オプションをそれぞれ使用して、昇順または降順でソートできます。デフォルトでは、順序が指定されていない場合ソートは昇順で行われます。
- sort asc** または **sort desc** オプションを指定してプッシュ クエリを設定している場合は、Cisco MDS NX-OS リリース 8.3(2) から Cisco MDS NX-OS リリース 8.3(1) または以前のリリースにダウングレードする前に、それらの **sort** オプションを削除してください。

クエリルール

以下は、クエリの構築に関するルールです。

- **select**、**from**、**where**、**sort**、および **limit** 条件は、[クエリ構文 \(37 ページ\)](#) の説明に従い、同じ順序で使用する必要があります。

- **select** 条件の下にある列の一覧は、**from** 条件の下にある *view_type* に対応するスキーマに属している必要があります。
- **where** 条件は、タイプが *key* 値のフローメトリックフィールドでのみ使用できます。タイプが *key* 値のフローメトリックフィールドの詳細については、[サポートされているビュータイプの一覧 \(39 ページ\)](#) を参照してください。
- Cisco MDS NX-OS、リリース 8.3(2) より前は、**sort** 条件は *metric* フィールドであり、**select** 条件の下に一覧表示されている列の中に存在している必要がありました。Cisco MDS NX-OS、リリース 8.3(2) からは、**sort** 条件は *metric* または *metadata* フィールドであり、**select** 条件の下に一覧表示されている列の中に存在している必要があります。

ビュー

ビューは、ポート、イニシエータ、ターゲット、LUN、またはこれらの有効な組み合わせに関するフローメトリックを表すものです。各ビュータイプは特定のフローメトリックをサポートしています。リソース使用率を最適化するために、オンボードクエリにはフローメトリックの長い名前が使用され、SAN Telemetry Streaming には短い名前が使用されます。詳細については、[フローメトリック \(133 ページ\)](#) を参照してください。

サポートされているビュータイプの一覧

次の表に、サポートされているビュータイプを示します。

表 6: サポートされているビュータイプ

ビュータイプ	説明	キー
port	ポートビューには、スイッチ上のポートのメタデータと IO メトリックが含まれています。	port
logical_port	論理ポートビューには、スイッチ上のポートに設定されている VSAN のメタデータと IO メトリックが含まれています。	port と vsan
app	アプリケーションビューには、IO 操作を実行しているさまざまなポートの背後でホストされている、関連するアプリケーションのメタデータと IO メトリックが含まれています。	ポートとアプリ ID

ビュータイプ	説明	キー
scsi_target	ターゲットビューには、IO 操作を実行するスイッチ上のさまざまなポートの背後に展開されている SCSI ターゲットのメタデータと IO メトリックが含まれています。	ポート、vsan、および scsi-target-id
nvme_target	ターゲットビューには、IO 操作を実行するスイッチ上のさまざまなポートの背後に展開されている NVMe ターゲットのメタデータと IO メトリックが含まれています。	ポート、vsan、および nvme-target-id
scsi_initiator	イニシエータ ビューには、IO 操作を開始するスイッチ上のさまざまなポートの背後に展開されているイニシエータのメタデータと IO メトリックが含まれています。	ポート、vsan、scsi-initiator-id、および vmid
nvme_initiator	イニシエータ ビューには、IO 操作を開始するスイッチ上のさまざまなポートの背後に展開されているイニシエータのメタデータと IO メトリックが含まれています。	ポート、vsan、nvme-initiator-id、および vmid
scsi_target_app	ターゲットアプリ ビューには、データがさまざまなターゲットでホストされているアプリケーションのメタデータと IO メトリックが含まれています。	port、vsan、scsi-target-fc-id、および app-id
nvme_target_app	ターゲットアプリ ビューには、データがさまざまなターゲットでホストされているアプリケーションのメタデータと IO メトリックが含まれています。	ポート、vsan、nvme-target-id、および app-id

ビュータイプ	説明	キー
scsi_initiator_app	イニシエータアプリ ビューには、イニシエータがIO操作を開始するアプリケーションのメタデータとIOメトリックが含まれています。	ポート、vsan、scsi-initiator-id、app-id、および vmid
nvme_initiator_app	イニシエータアプリ ビューには、イニシエータがIO操作を開始するアプリケーションのメタデータとIOメトリックが含まれています。	ポート、vsan、nvme-initiator-id、app-id、および vmid
scsi_target_it_flow	ターゲット initiator-target (IT) フロー ビューには、さまざまなターゲットと関連付けられている IT フローのメタデータとIOメトリックが含まれています。	ポート、vsan、scsi-target-id、scsi-initiator-id、および vmid
nvme_target_it_flow	ターゲット initiator-target (IT) フロー ビューには、さまざまなターゲットと関連付けられている IT フローのメタデータとIOメトリックが含まれています。	ポート、vsan、nvme-target-id、nvme-initiator-id、および vmid
scsi_initiator_it_flow	イニシエータ IT フロー ビューには、さまざまなイニシエータと関連付けられている IT フローのメタデータとIOメトリックが含まれています。	ポート、vsan、scsi-initiator-id、scsi-target-id、および vmid
nvme_initiator_it_flow	イニシエータ IT フロー ビューには、さまざまなイニシエータと関連付けられている IT フローのメタデータとIOメトリックが含まれています。	ポート、vsan、nvme-initiator-id、nvme-target-id、および vmid
scsi_target_tl_flow	ターゲット target-LUN (TL) フロー ビューには、さまざまな SCSI ターゲットと関連付けられている LUN のメタデータとIOメトリックが含まれています。	ポート、vsan、scsi-target-fc-id、および lun-id

ビュータイプ	説明	キー
nvme_target_tn_flow	ターゲット target-namespace ID (TN) フロービューには、さまざまな NVMeI ターゲットと関連付けられている namespace ID のメタデータと IO メトリックが含まれています。	ポート、vsan、nvme-target-id、および namespace-id
scsi_target_itl_flow	ターゲット initiator-target-LUN (ITL) フロービューには、さまざまな SCSI ターゲットと関連付けられている IT フローのメタデータと IO メトリックが含まれています。	ポート、vsan、scsi-target-id、scsi-initiator-id、lun-id、および vmid
nvme_target_itn_flow	ターゲット initiator-target-namespace ID (ITN) フロービューには、さまざまな NVMeI ターゲットと関連付けられている ITN フローのメタデータと IO メトリックが含まれています。	ポート、vsan、nvme-target-id、nvme-initiator-id、namespace-id、および vmid
scsi_initiator_itl_flow	イニシエータ ITL フロービューには、さまざまな SCSI イニシエータと関連付けられている ITL フローのメタデータと IO メトリックが含まれています。	ポート、vsan、scsi-initiator-id、scsi-target-id、lun-id、および vmid
nvme_initiator_itn_flow	イニシエータ ITN フロービューには、さまざまな NVMe イニシエータと関連付けられている ITN フローのメタデータと IO メトリックが含まれています。	ポート、vsan、nvme-initiator-id、nvme-target-id、namespace-id、および vmid
scsi_target_io	ターゲット IO ビューには、さまざまなターゲットが実行するアクティブな IO の IO トランザクションの詳細が含まれています。	ポート、vsan、scsi-target-id、scsi-initiator-id、および vmid

ビュータイプ	説明	キー
nvme_target_io	ターゲット IO ビューには、さまざまなターゲットが実行するアクティブな IO の IO トランザクションの詳細が含まれています。	ポート、vsan、nvme-target-id、nvme-initiator-id、ox-id、および vmid
scsi_initiator_io	イニシエータ IO ビューには、さまざまなイニシエータが開始するアクティブな IO の IO トランザクションの詳細が記録されています。	ポート、vsan、scsi-initiator-id、scsi-target-id、ox-id および vmid
nvme_initiator_io	イニシエータ IO ビューには、さまざまなイニシエータが開始するアクティブな IO の IO トランザクションの詳細が記録されています。	ポート、vsan、nvme-initiator-id、nvme-target-id、ox-id、および vmid

ビュータイプの表示



(注) このセクションで提供される例は、SCSI 分析タイプ用であり、NVMe 分析タイプにも拡張できます。

シスコでは、さまざまなビュータイプを説明するためのサンプルトポロジを考えました。以下は、次の図の説明です。

- イニシエータ 1 とイニシエータ 2 は VSAN 1 に設定されていて、ゾーン 1 にあるターゲット 1、ターゲット 2、LUN 1、および LUN 2 と通信しています。
 - イニシエータ 1 は、ターゲット 1 に対して 125 の読み取り IO を生成し、ターゲット 2 に対して 75 の読み取り IO を生成しています。
 - イニシエータ 2 は、ターゲット 1 とターゲット 2 のそれぞれに対して 50 の読み取り IO を生成しています。

- イニシエータ 3 は VSAN 2 に設定されていて、ゾーン 2 にあるターゲット 3、LUN 3、および LUN 4 と通信しています。

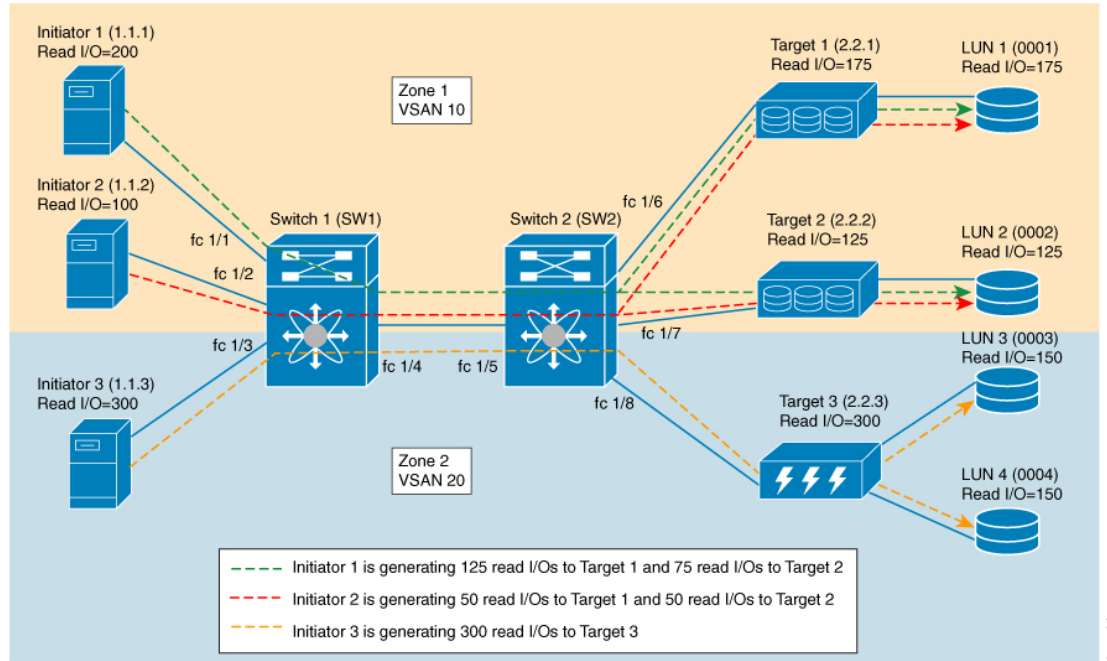
イニシエータ 3 は、ターゲット 3 に対して 300 の読み取り IO を生成しています。ターゲット 3 は、LUN 3 と LUN 4 のそれぞれに対して 150 の読み取り IO を生成しています。



(注) 次の図のカッコ内に記載されている情報はデバイスのファイバチャネル ID (FCID) です。

サポートされているビュータイプの一覧とそれぞれの説明については、[サポートされているビュータイプの一覧 \(39 ページ\)](#) を参照してください。

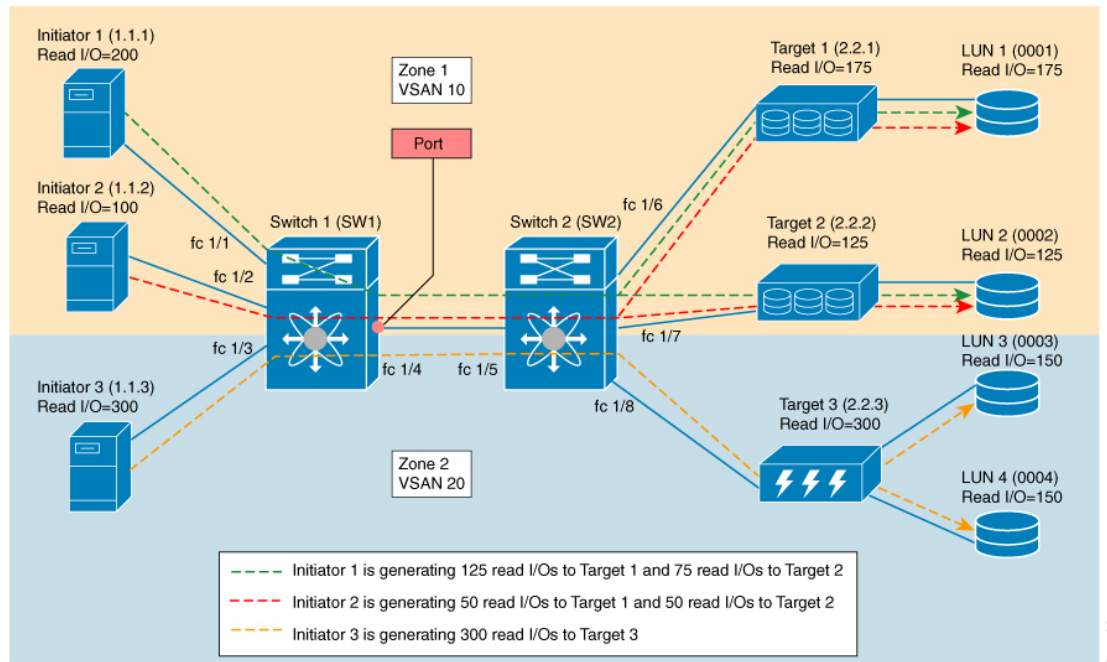
図 15: ビュータイプの表示のサンプルトポロジ



355762

次の図は、ポートビュータイプから見たフローメトリックを示しています。

図 16: ポートビュータイプ



355763

表 7:ポート ビュータイプ

ポートビュー	フローメトリック
ポートビュー、ポート = fc 1/4	total_read_io_count=600 (ポート上に表示されているすべてのイニシエータの読み取り I/O)

次の図は、論理ポートビュータイプから見たフローメトリックを示しています。

図 17:論理ポートビュータイプ

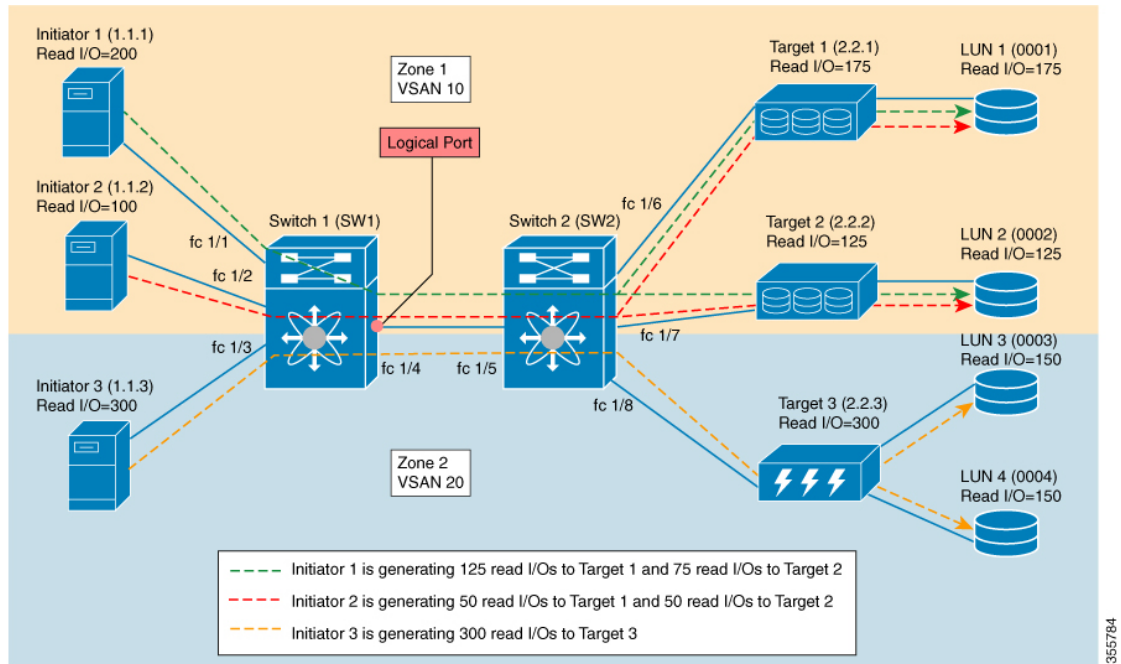
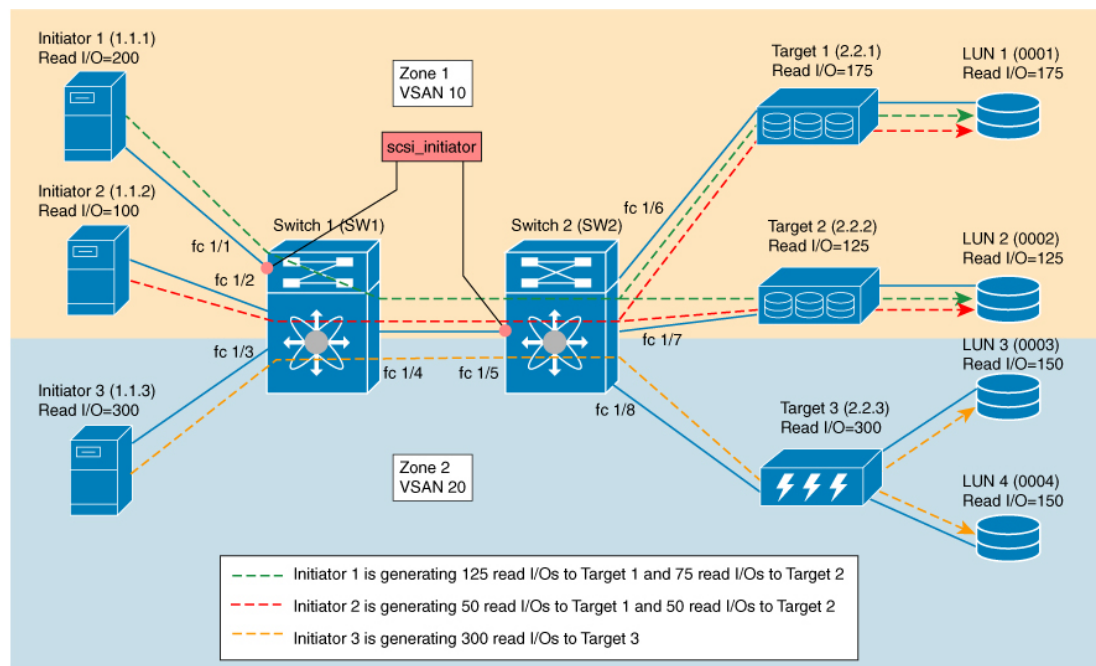


表 8:論理ポートビュータイプ

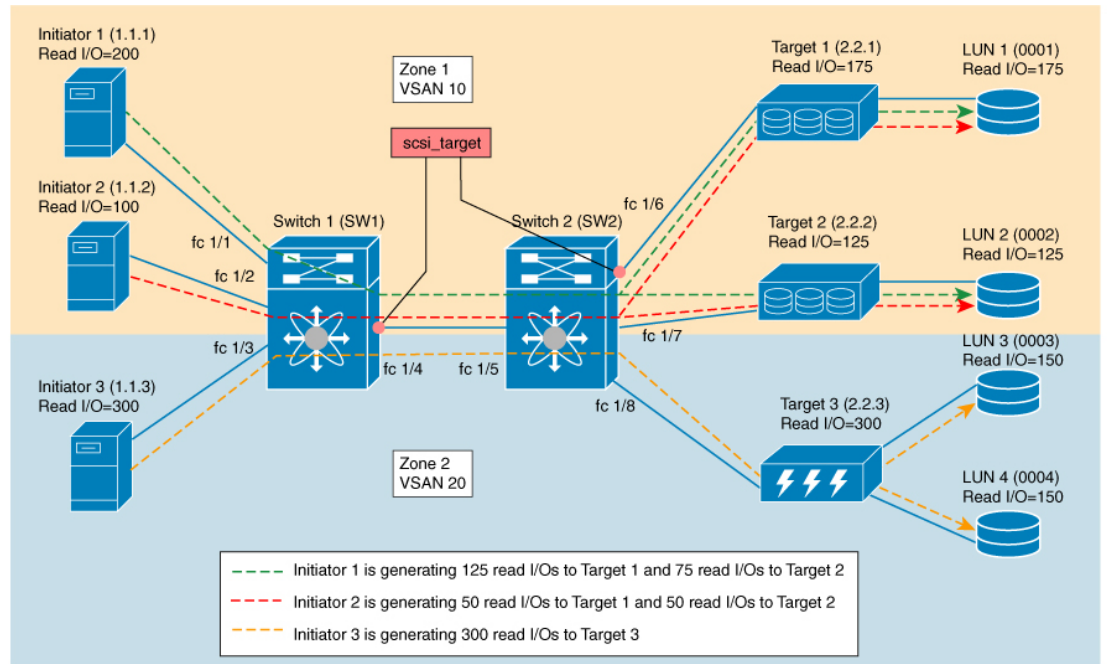
論理ポートビュータイプ	フローメトリック
論理ポートでポート = fc 1/4、VSAN = 1	total_read_io_count = 300 (VSAN 1 内のすべてのイニシエータの読み取り I/O)

次の図は、scsi_initiator ビュータイプから見たフローメトリックを示しています。

図 18: `scsi_initiator` ビュータイプ表 9: `scsi_initiator` ビュータイプ

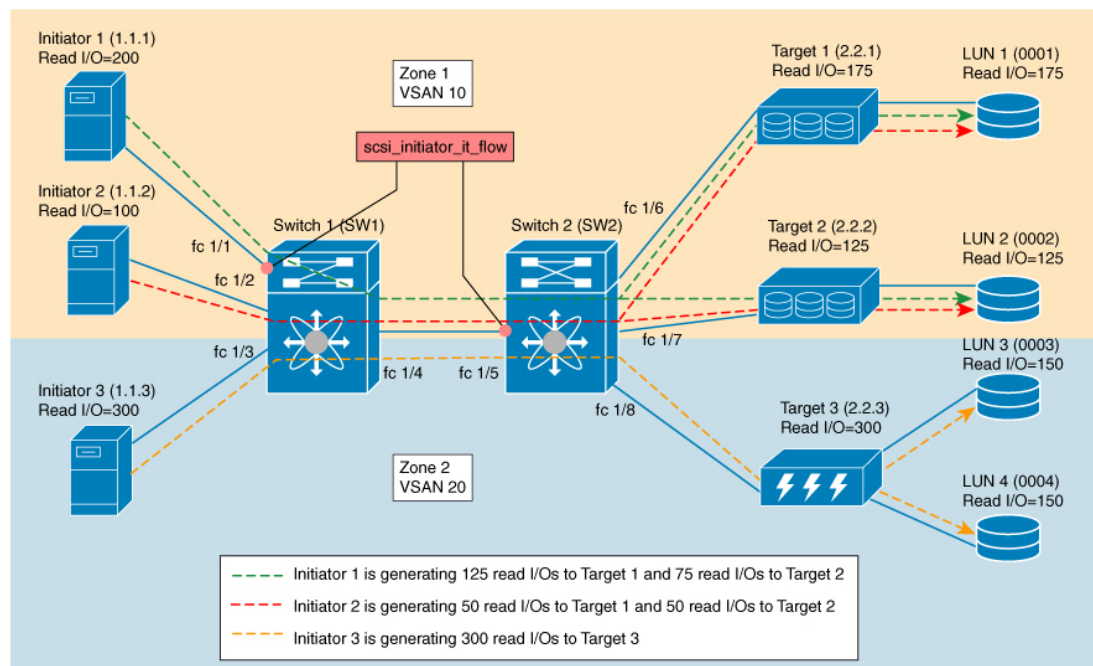
<code>scsi_initiator</code> ビュータイプ	フローメトリック
<code>scsi_initiator</code> ビューでポート = fc 1/1、VSAN = 1、イニシエータ ID = 1.1.1 ポート = fc 1/5、VSAN = 1、イニシエータ ID = 1.1.1 の <code>scsi_initiator</code> ビュー	total_read_io_count=200 (イニシエータ ID 1.1.1 の読み取り I/O)
<code>scsi_initiator</code> ビューでポート = fc 1/5、VSAN = 1、イニシエータ ID = 1.1.2	total_read_io_count=100 (イニシエータ ID 1.1.2 の読み取り I/O)
<code>scsi_initiator</code> ビューでポート = fc 1/5、VSAN = 2、イニシエータ ID = 1.1.3	total_read_io_count=300 (イニシエータ ID 1.1.3 の読み取り I/O)

次の図は、`scsi_target` ビュータイプから見たフローメトリックを示しています。

図 19: `scsi_target` ビュータイプ表 10: `scsi_target` ビュータイプ

scsi_target ビュー	フロー メトリック
scsi_target ビューでポート = fc 1/6、VSAN = 1、ターゲット ID = 2.2.1 scsi_target ビューでポート = fc 1/4、VSAN = 1、ターゲット ID = 2.2.1	total_read_io_count = 175 (ターゲット ID 2.2.1 の読み取り I/O)
scsi_target ビューでポート = fc 1/4、VSAN = 1、ターゲット ID = 2.2.2	total_read_io_count = 125 (ターゲット ID 2.2.2 の読み取り I/O)
scsi_target ビューでポート = fc 1/4、VSAN = 2、ターゲット ID = 2.2.3	total_read_io_count = 300 (ターゲット ID 2.2.3 の読み取り I/O)

次の図は、`scsi_initiator_it_flow` ビュータイプから見たフロー メトリックを示しています。

図 20: `scsi_initiator_it_flow` ビュータイプ表 11: `scsi_initiator_it_flow` ビュータイプ

<code>scsi_initiator_it_flow</code> ビュー	フロー メトリック
<code>scsi_initiator_it_flow</code> ビューで、ポート = fc 1/1、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1 <code>scsi_initiator_it_flow</code> ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1	total_read_io_count=125 (イニシエータ ID 1.1.1 とターゲット ID 2.2.1 間だけの読み取り I/O)
<code>scsi_initiator_it_flow</code> ビューで、ポート = fc 1/1、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.2 <code>scsi_initiator_it_flow</code> ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.2	total_read_io_count=75 (イニシエータ ID 1.1.1 とターゲット ID 2.2.2 間だけの読み取り I/O)
<code>scsi_initiator_it_flow</code> ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.1	total_read_io_count=50 (イニシエータ ID 1.1.2 とターゲット ID 2.2.1 間だけの読み取り I/O)
<code>scsi_initiator_it_flow</code> ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.2	total_read_io_count=50 (イニシエータ ID 1.1.2 とターゲット ID 2.2.2 間だけの読み取り I/O)

scsi_initiator_it_flow ビューで、ポート = fc 1/5、VSAN=2、イニシエータ ID=1.1.3、ターゲット ID = 2.2.3	total_read_io_count=300 (イニシエータ ID 1.1.3 とターゲット ID 2.2.3 間だけの読み取り I/O)
--	--

次の図は、scsi_target_it_flow ビュータイプから見たフローメトリックを示しています。

図 21: scsi_target_it_flow ビュータイプ

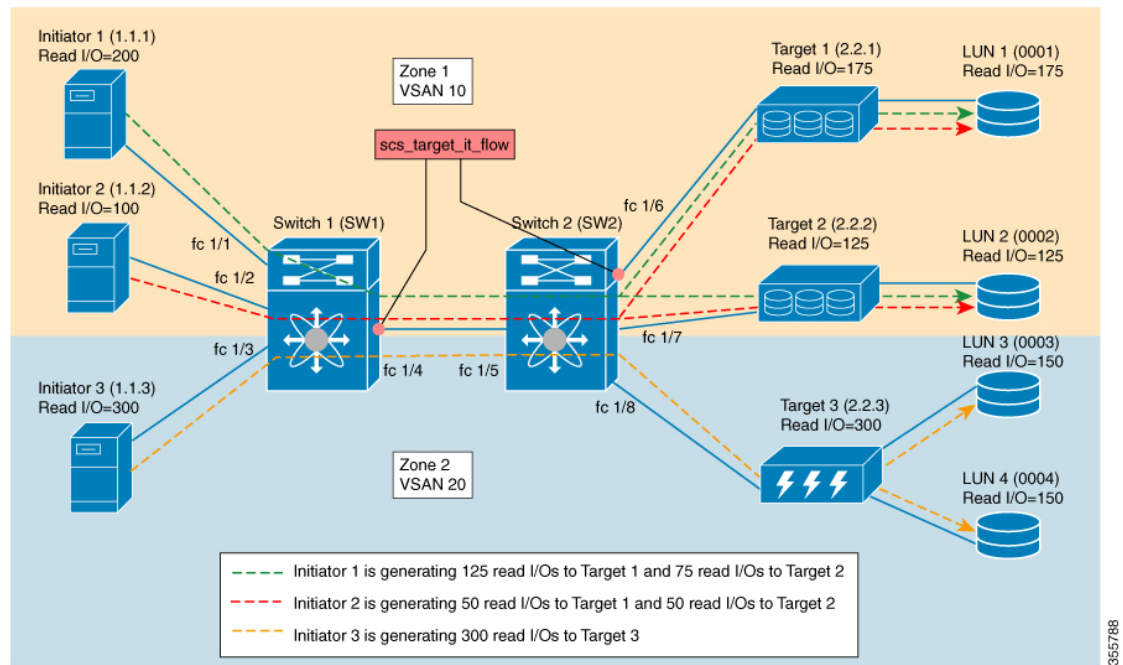


表 12: scsi_target_it_flow ビュータイプ

scsi_target_it_flow ビュー	フローメトリック
scsi_target_it_flow ビューで、ポート = fc 1/6、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1	total_read_io_count=125 (イニシエータ ID 1.1.1 とターゲット ID 2.2.1 間だけの読み取り I/O)
scsi_target_it_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1	
scsi_target_it_flow ビューで、ポート = fc 1/6、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.1	total_read_io_count=50 (イニシエータ ID 1.1.2 とターゲット ID 2.2.1 間だけの読み取り I/O)
scsi_target_it_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.1	

scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.2	total_read_io_count=75 (イニシエータ ID 1.1.1 とターゲット ID 2.2.2 間だけの読み取り I/O)
scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.2	total_read_io_count=50 (イニシエータ ID 1.1.2 とターゲット ID 2.2.2 間だけの読み取り I/O)
scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=2、イニシエータ ID=1.1.3、ターゲット ID = 2.2.3	total_read_io_count=300 (イニシエータ ID 1.1.3 とターゲット ID 2.2.3 間だけの読み取り I/O)

次の図は、scsi_initiator_itl_flow ビュータイプから見たフローメトリックを示しています。

図 22: scsi_initiator_itl_flow ビュータイプ

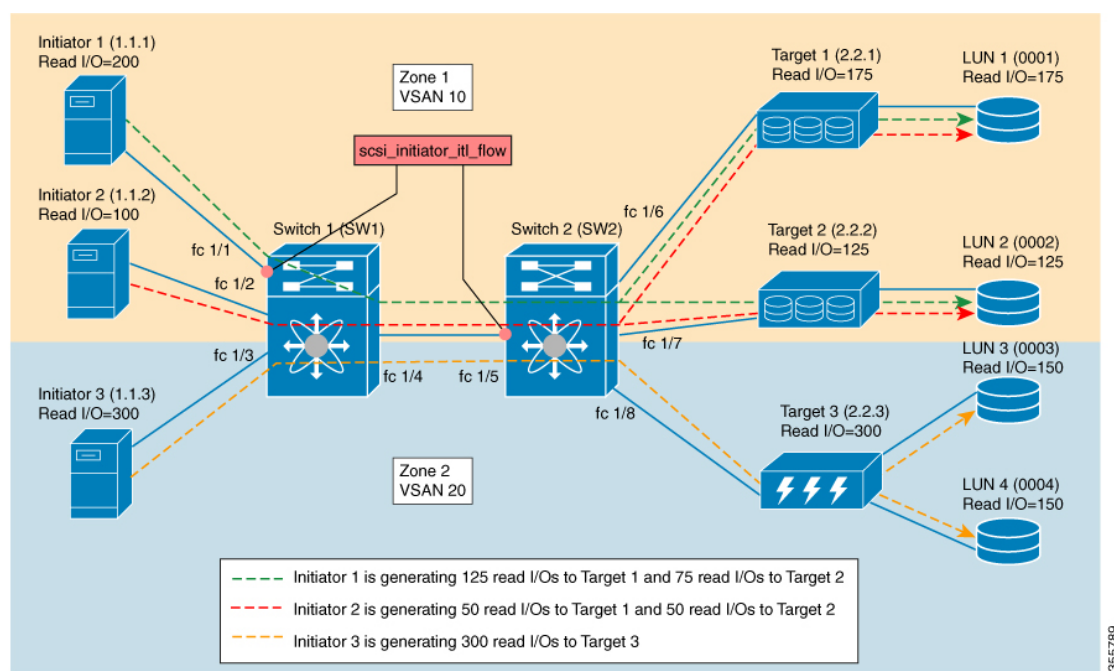


表 13: scsi_initiator_itl_flow ビュータイプ

scsi_initiator_itl_flow ビュー	フローメトリック
scsi_initiator_itl_flow ビューで、ポート = fc 1/1、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1、LUN ID = 0001	total_read_io_count = 125 (イニシエータ ID 1.1.1、ターゲット ID 2.2.1、および LUN ID 0001 間だけの読み取り I/O)
scsi_initiator_itl_flow ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1、LUN ID = 0001	

<p>scsi_initiator_itl_flow ビューで、ポート = fc 1/1、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.2、LUN ID = 0002</p> <p>scsi_initiator_itl_flow ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.2、LUN ID = 0002</p>	<p>total_read_io_count = 75 (イニシエータ ID 1.1.1、ターゲット ID 2.2.2、および LUN ID 0002 間だけの読み取り I/O)</p>
<p>scsi_initiator_itl_flow ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.1、LUN ID = 0001</p> <p>scsi_initiator_itl_flow ビューで、ポート = fc 1/5、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.2、LUN ID = 0002</p>	<p>total_read_io_count = 50 (イニシエータ ID 1.1.2、ターゲット ID 2.2.1、および LUN ID 0001 とイニシエータ ID 1.1.2、ターゲット ID 2.2.2、および LUN ID 0002 間だけの読み取り I/O)</p>
<p>scsi_initiator_itl_flow ビューで、ポート = fc 1/5、VSAN=2、イニシエータ ID=1.1.3、ターゲット ID = 2.2.3、LUN ID = 0003</p> <p>scsi_initiator_itl_flow ビューで、ポート = fc 1/5、VSAN=2、イニシエータ ID=1.1.3、ターゲット ID = 2.2.3、LUN ID = 0004</p>	<p>total_read_io_count = 150 (イニシエータ ID 1.1.3、ターゲット ID 2.2.3、および LUN ID 0003 とイニシエータ ID 1.1.3、ターゲット ID 2.2.3、および LUN ID 0004 間だけの読み取り I/O)</p>

次の図は、scsi_target_itl_flow ビュータイプから見たフローメトリックを示しています。

図 23: scsi_target_itl_flow ビュータイプ

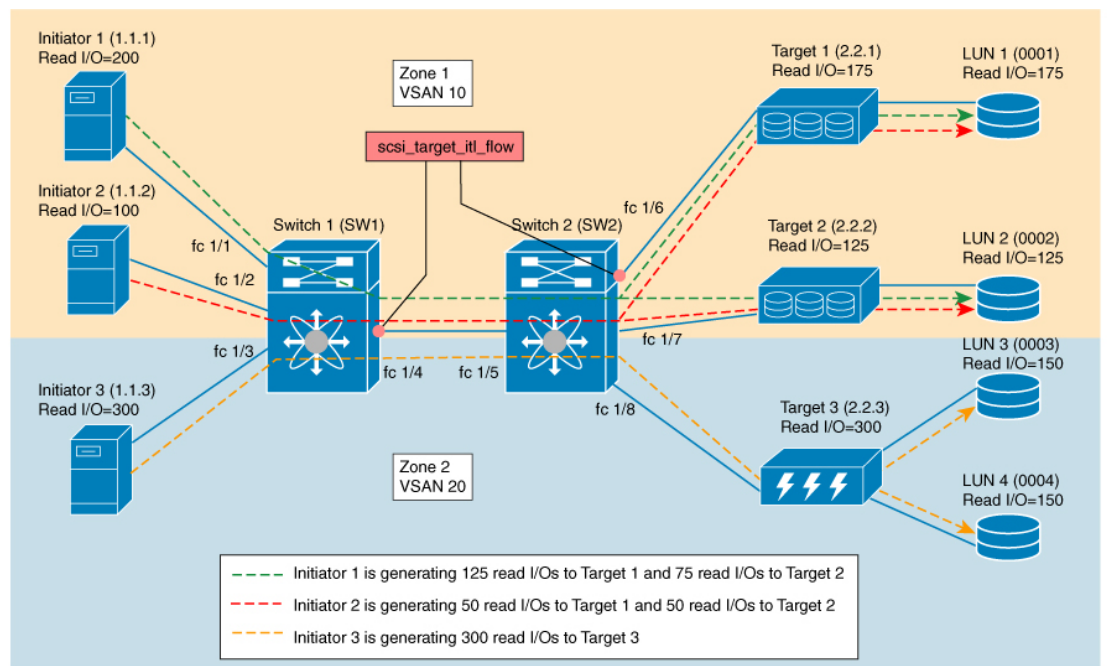
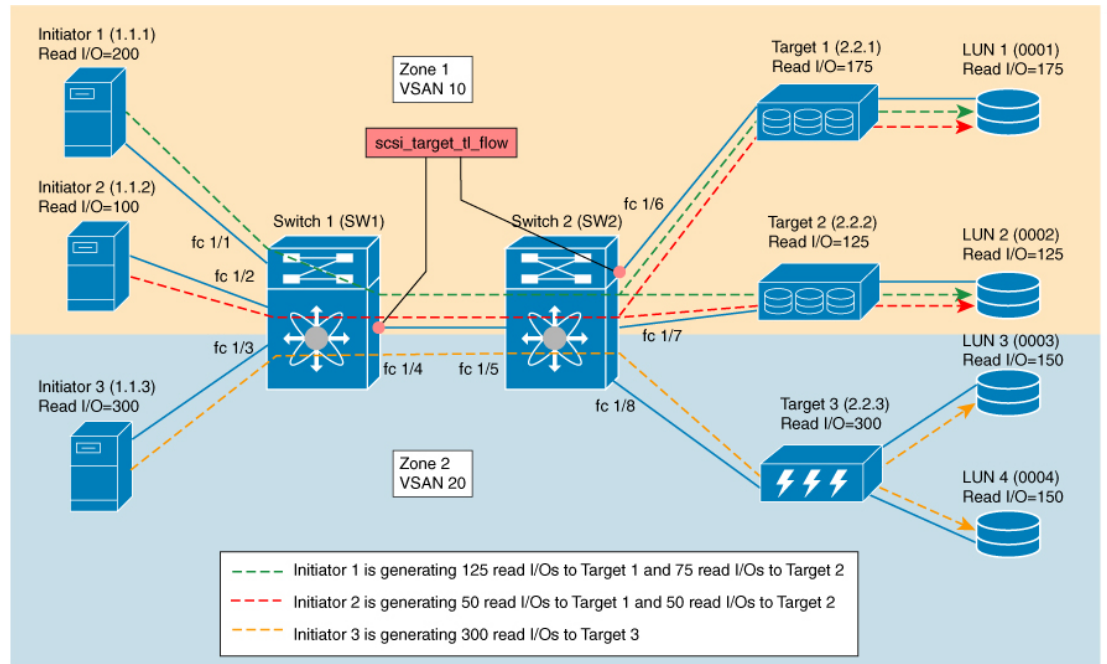


表 14: *scsi_target_itl_flow* ビュータイプ

scsi_target_itl_flow ビュー	フロー メトリック
scsi_target_itl_flow ビューで、ポート = fc 1/6、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1、LUN ID = 0001 scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.1、LUN ID = 0001	total_read_io_count = 125 (イニシエータ ID 1.1.1、ターゲット ID 2.2.1、および LUN ID 0001 間だけの読み取り I/O)
scsi_target_itl_flow ビューで、ポート = fc 1/6、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.1、LUN ID = 0001 scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.1、LUN ID = 0001	total_read_io_count = 50 (イニシエータ ID 1.1.2、ターゲット ID 2.2.1、および LUN ID 0001 間だけの読み取り I/O)
scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.1、ターゲット ID = 2.2.2、LUN ID = 0002	total_read_io_count = 75 (イニシエータ ID 1.1.1、ターゲット ID 2.2.2、および LUN ID 0002 間だけの読み取り I/O)
scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=1、イニシエータ ID=1.1.2、ターゲット ID = 2.2.2、LUN ID = 0002	total_read_io_count = 50 (イニシエータ ID 1.1.2、ターゲット ID 2.2.2、および LUN ID 0002 間だけの読み取り I/O)
scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=2、イニシエータ ID=1.1.3、ターゲット ID = 2.2.3、LUN ID = 0003 scsi_target_itl_flow ビューで、ポート = fc 1/4、VSAN=2、イニシエータ ID=1.1.3、ターゲット ID = 2.2.3、LUN ID = 0004	total_read_io_count = 150 (イニシエータ ID 1.1.3、ターゲット ID 2.2.3、および LUN ID 0003 とイニシエータ ID 1.1.3、ターゲット ID 2.2.3、および LUN ID 0004 間だけの読み取り I/O)

次の図は、scsi_target_tl_flow ビュータイプから見たフローメトリックを示しています。

図 24: `scsi_target_tl_flow` ビュータイプ

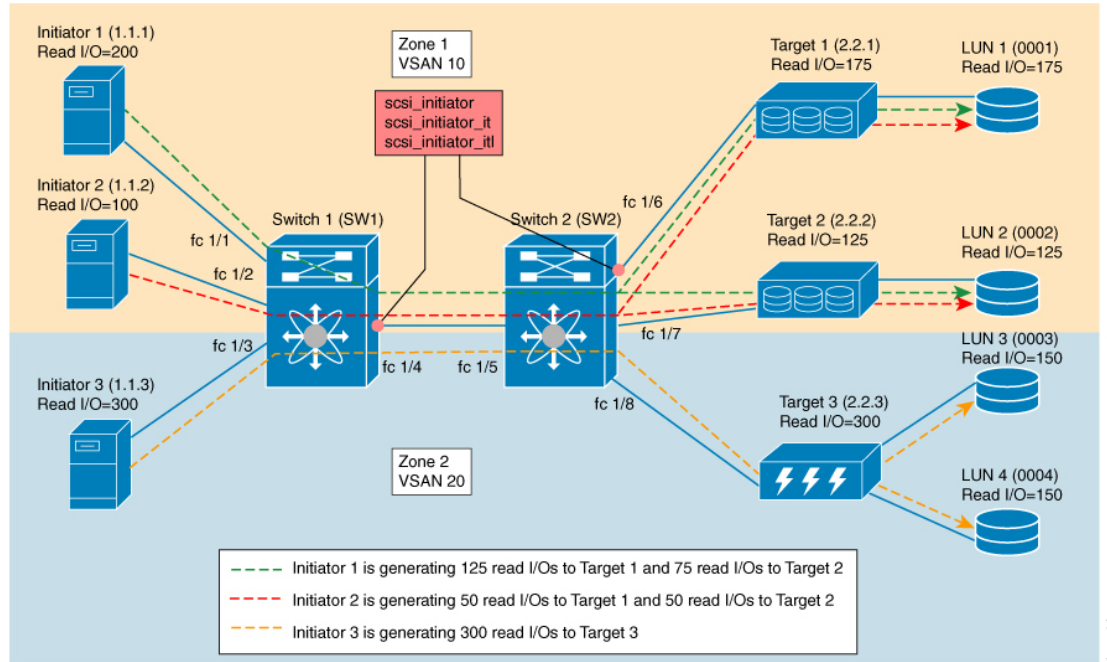
355791

表 15: `scsi_target_tl_flow` ビュータイプ

scsi_target_tl_flow ビュー	フロー メトリック
scsi_target_tl_flow ビューで、ポート = fc 1/6、VSAN = 1、ターゲット ID = 2.2.1、LUN ID = 0001	total_read_io_count = 175 (ターゲット ID 2.2.1 と LUN ID 0001 間だけの読み取り I/O)
scsi_target_tl_flow ビューで、ポート = fc 1/4、VSAN = 1、ターゲット ID = 2.2.1、LUN ID = 0001	
scsi_target_tl_flow ビューで、ポート = fc 1/4、VSAN = 1、ターゲット ID = 2.2.2、LUN ID = 0002	total_read_io_count = 125 (ターゲット ID 2.2.2 と LUN ID 0002 間だけの読み取り I/O)
scsi_target_tl_flow ビューで、ポート = fc 1/4、VSAN = 2、ターゲット ID = 2.2.3、LUN ID = 0003	total_read_io_count = 150 (ターゲット ID 2.2.3 と LUN ID 0003 およびターゲット ID 2.2.3 と LUN ID 0004 間だけの読み取り I/O)
scsi_target_tl_flow ビューで、ポート = fc 1/4、VSAN = 2、ターゲット ID = 2.2.3、LUN ID = 0004	

次の図は、total_read_io_count が 0 のイニシエータ ビューを示しています。

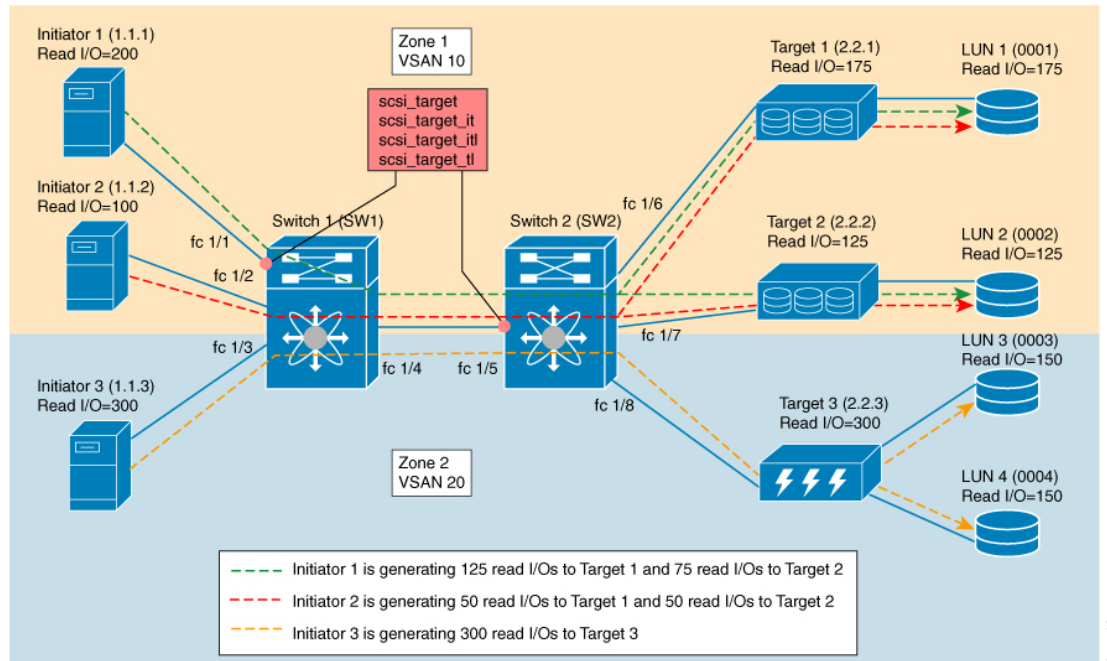
図 25: total_read_io_count が 0 のイニシエータ ビュー



355792

次の図は、total_read_io_count が 0 のターゲット ビューを示しています。

図 26: total_read_io_count が 0 のターゲット ビュー

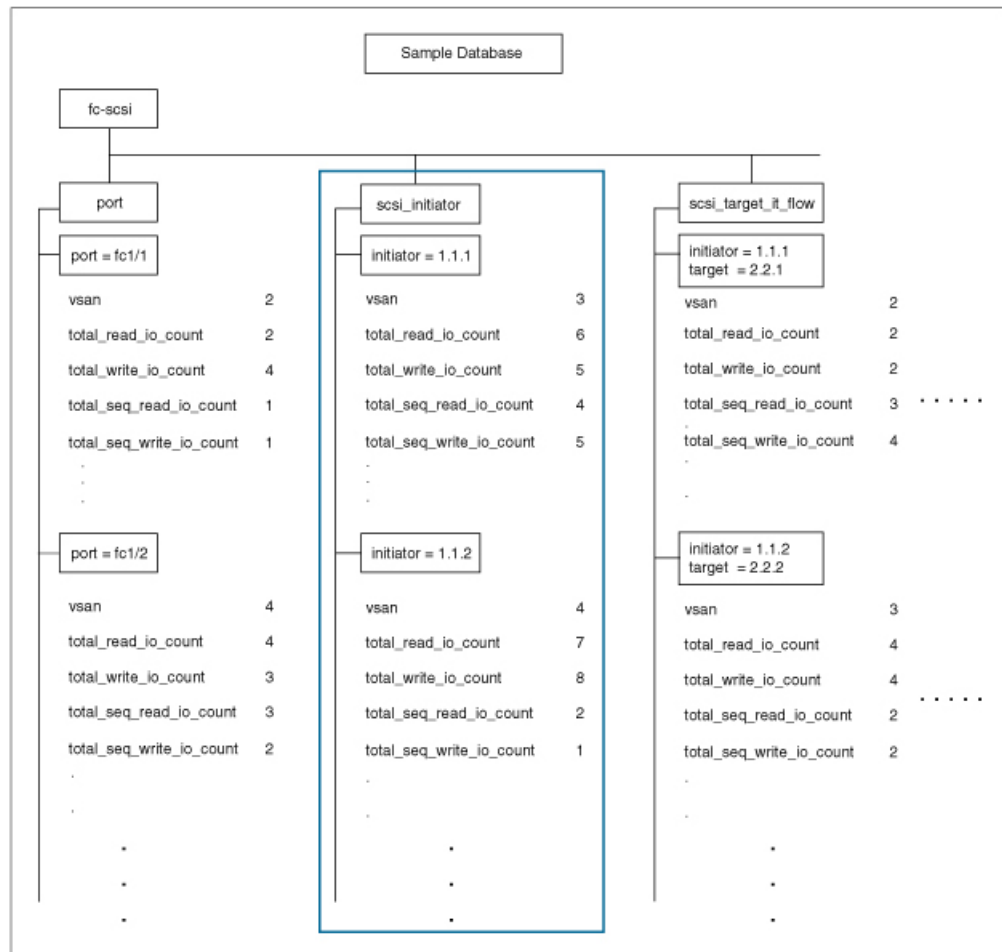


355793

例：クエリ構文の設定

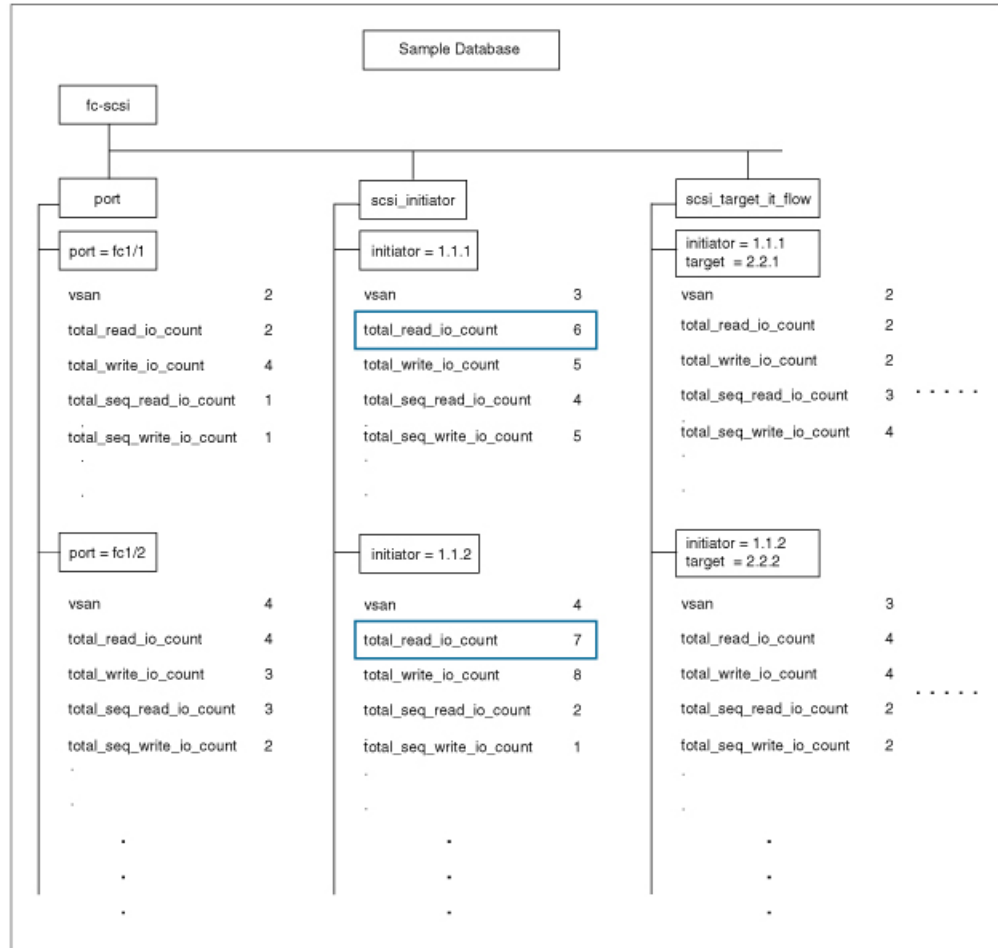
`show analytics query 'select all from fc-scsi.scsi_initiator'` コマンドを実行すると、次の図に示されているサンプルデータベースで確認できるように、すべてのイニシエータのフローメトリックが出力されます。

図 27: すべてのイニシエータのフローメトリック



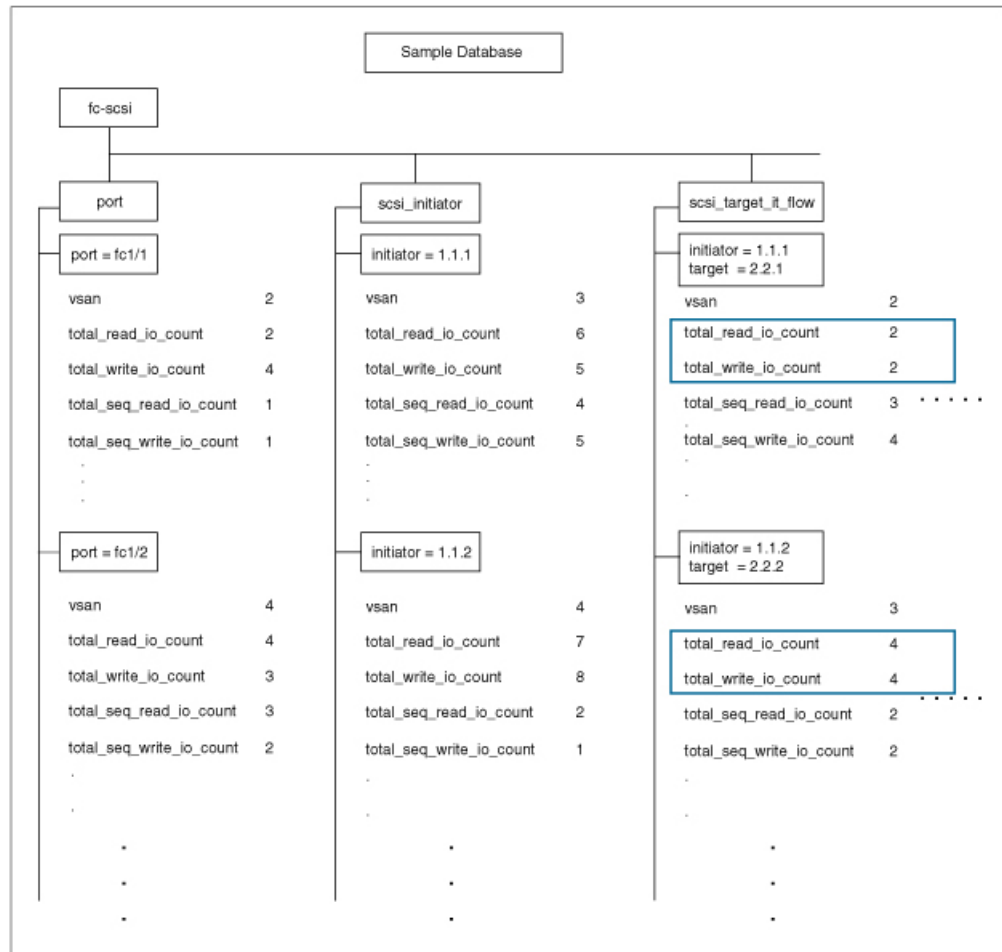
`show analytics query 'select total_read_io_count from fc-scsi.scsi_initiator'` コマンドを実行すると、次の図に示されているサンプルデータベースで確認できるように、ターゲットの `total_read_io_count` フローメトリックが出力されます。

図 28: ターゲットの読み取り IO 総数のフロー メトリック



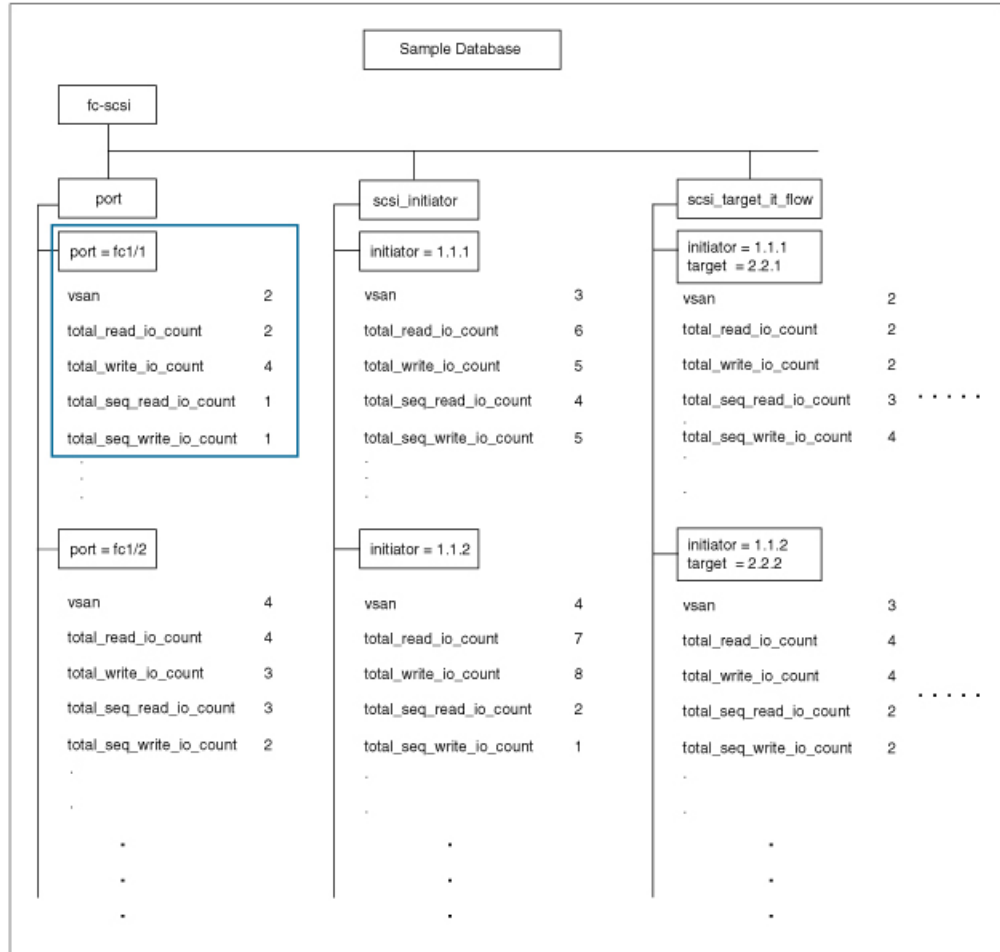
`show analytics query 'select total_read_io_count,total_write_io_count from fc-scsi.scsi_target_it_flow'` コマンドを実行すると、次の図に示されているサンプルデータベースで確認できるように、ターゲットから見たイニシエータとターゲットの `total_read_io_count` および `total_write_io_count` フローメトリックが出力されます。

図 29: イニシエータとターゲットの読み取り IO 総数と書き込み IO 総数のフローメトリック



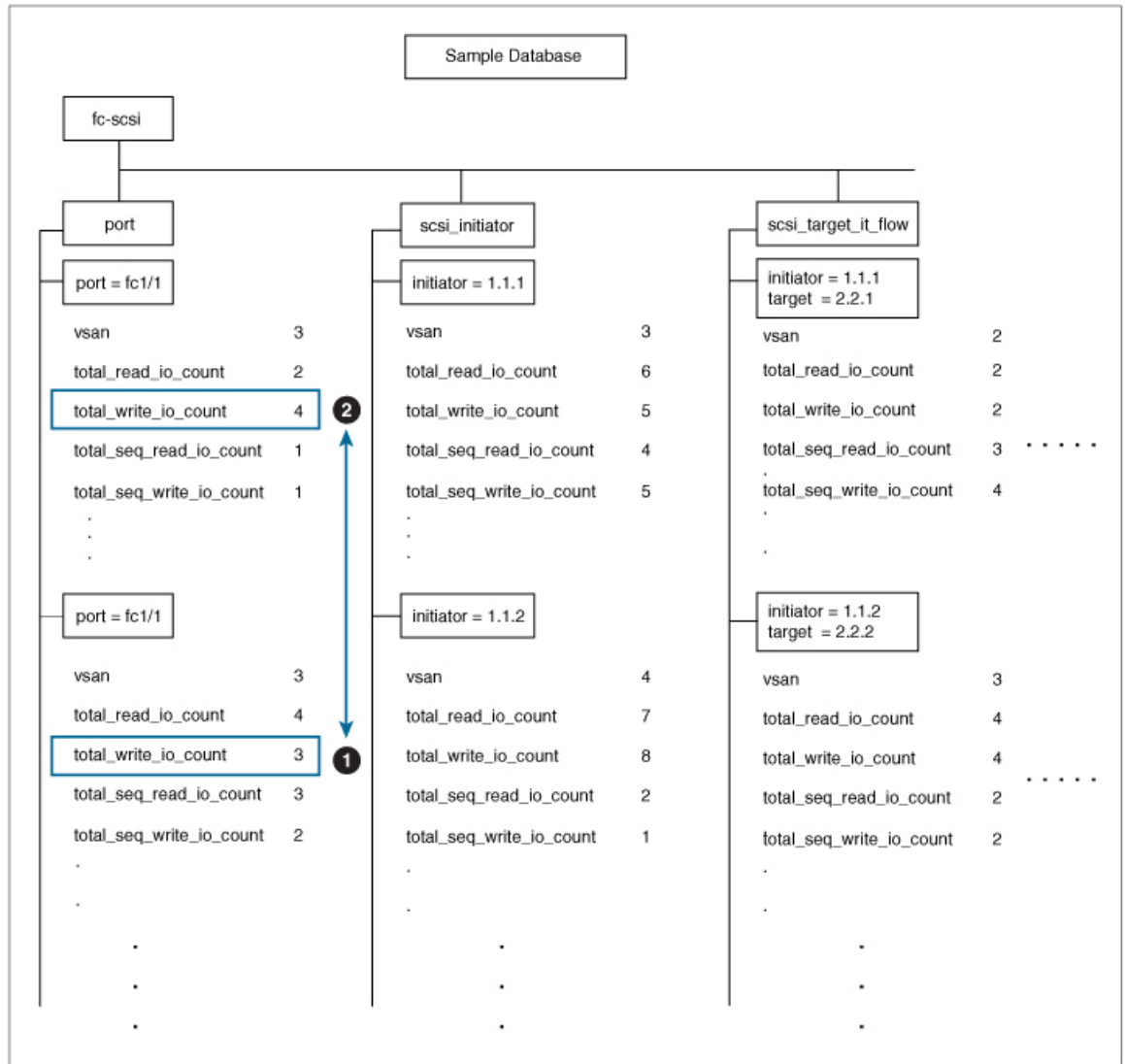
`show analytics query 'select all from fc-scsi.port where port=fc1/1 and vsan=2 limit 1'` コマンドを実行すると、次の図に示されているサンプルデータベースで確認できるように、レコードの数が 1 に制限されているポート fc1/1、VSAN 2 の一部であるポートのフローメトリックが出力されます。

図 30: レコードの数が 1 に制限されている VSAN 2 に属しているポート FC 1/1 のフロー メトリック



show analytics query 'select all from fc-scsi.scsi_initiator where port=fc1/1 and vsan=3 sort total_write_io_count' コマンドを実行すると、次の図のサンプルデータベースで確認できるように、ポート fc1/1 および VSAN 3 の一部であるイニシエータの total_write_io_count フローメトリックが出力されます。出力はソートされています。

図 31: ポート FC1/1 と VSAN 3 に属しているイニシエータの書き込み IO 総数のフローメトリック (出力はソート済み)



355363

クエリの作成と使用

フローメトリックは、クエリ構文の形式である `query_string` を使用して分析されます。

インストールされているプッシュクエリの表示

インストールされているプッシュクエリを表示するには、次のコマンドを実行します。

```
switch# show analytics query {all | name query_name}
```

プッシュクエリの結果の表示

プッシュクエリの結果を表示するには、次のコマンドを実行します。

```
switch# show analytics query name query_name result
```

プルクエリの実行

プルクエリを実行するには、次のコマンドを実行します。

```
switch# show analytics query "query_string" [clear] [differential]
```



(注) "*query_string*" を使用して、"*select all from fc-scsi.port*" のように、**select**、**table**、**limit** などのクエリセマンティックを指定します。

プッシュクエリの設定

プッシュクエリを設定するには、次の手順を実行します。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 特定の間隔で表示されるフローメトリックのクエリ文字列とタイマー値を指定します。

```
switch(config)# analytics query "query_string" name query_name type periodic [interval seconds] [clear] [differential]
```

"*query_string*" を使用したプッシュクエリは、一度に1つのみ許可されます。重複するプッシュクエリ名を設定しようとする、現在の設定が重複していることを示すメッセージが表示されます。

(注) プルクエリ、プッシュクエリ、およびオーバーレイ CLI は、SAN アナリティクス機能が有効になっているインターフェイスにのみ適用されます。

設定されているプッシュクエリの削除

設定されているプッシュクエリを削除するには、次の手順を実行します。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 設定されているプッシュ クエリを削除します。

```
switch(config)# no analytics name query_name
```

メトリックのクリア

クエリ文字列に一致するビュー インスタンスのすべてのフロー メトリックをリセットするには、次のコマンドを実行します。

```
switch# clear analytics query "query_string"
```



- (注)
- "*query_string*" は "*select all from <view-name>*" の形式である必要があります。
 - フロー メトリックは、プッシュ クエリをインストールしなくてもクリアできます。
 - **clear analytics query** コマンドは、プッシュ クエリで使用されている **clear** オプションとは異なります。 **clear analytics query** コマンドは、クエリ構文を満たすすべてのメトリックをリセットします。プッシュ クエリで使用される **clear** オプションは、最小、最大、およびピーク フロー メトリックをリセットします。

ビューの消去

特定のビュー インスタンスとその関連メトリックを削除するには、次のコマンドを実行します。

```
switch# purge analytics query "query_string"
```



- (注)
- "*query_string*" は "*select all from <view-name>*" の形式である必要があります。
 - フロー メトリックは、プッシュ クエリをインストールしなくてもクリアできます。
 - 消去クエリの **where** 句には、*port* キー フィールドのみ指定できます。

設定されているプッシュクエリの結果の表示

show analytics query name *query_name* result コマンドを使用して表示されているフローメトリックは、このコマンドの実行時の時間間隔で更新されたメトリックです。たとえば、プッシュクエリが 30 秒間隔で更新するように設定されている場合、**show analytics query name *query_name* result** コマンドは 35 秒後に実行され、プッシュクエリでは、時間間隔が 30 秒のときに更新されたフローメトリックが表示されます。

設定されているプッシュクエリのフローメトリックを表示するには、次のコマンドを実行します。

```
switch# show analytics query name query_name result
```

例：クエリの作成と使用



- (注)
- 出力の “values” の後にある番号は、対応するレコードの番号を示しています。
 - Cisco MDS NX-OS リリース 8.3(1) では新しいメトリックが追加されています。これは、Cisco MDS NX-OS リリース 8.3(1) 以降のリリースと Cisco MDS NX-OS リリース 8.2(1) の間でクエリの結果が少し異なる可能性があるためです。

次に、SCSI イニシエータ ITL フロー ビュー インスタンスのすべてのフローメトリックの出力の例を示します。

```
switch# show analytics query 'select all from fc-scsi.scsi_initiator_itl_flow'
{ "values": {
  "1": {
    "port": "fc1/1",
    "vsan": "10",
    "app_id": "255",
    "initiator_id": "0xe80041",
    "target_id": "0xd60200",
    "lun": "0000-0000-0000-0000",
    "active_io_read_count": "0",
    "active_io_write_count": "1",
    "total_read_io_count": "0",
    "total_write_io_count": "1162370362",
    "total_seq_read_io_count": "0",
    "total_seq_write_io_count": "1",
    "total_read_io_time": "0",
    "total_write_io_time": "116204704658",
    "total_read_io_initiation_time": "0",
    "total_write_io_initiation_time": "43996934029",
    "total_read_io_bytes": "0",
    "total_write_io_bytes": "595133625344",
    "total_read_io_inter_gap_time": "0",
    "total_write_io_inter_gap_time": "41139462314556",
    "total_time_metric_based_read_io_count": "0",
    "total_time_metric_based_write_io_count": "1162370358",
    "total_time_metric_based_read_io_bytes": "0",
    "total_time_metric_based_write_io_bytes": "595133623296",
    "read_io_rate": "0",
```



```

    "peak_read_io_rate": "0",
    "write_io_rate": "7250",
    "peak_write_io_rate": "7304",
    "read_io_bandwidth": "0",
    "peak_read_io_bandwidth": "0",
    "write_io_bandwidth": "3712384",
    "peak_write_io_bandwidth": "3739904",
    "read_io_size_min": "0",
    "read_io_size_max": "0",
    "write_io_size_min": "512",
    "write_io_size_max": "512",
    "read_io_completion_time_min": "0",
    "read_io_completion_time_max": "0",
    "write_io_completion_time_min": "89",
    "write_io_completion_time_max": "416",
    "read_io_initiation_time_min": "0",
    "read_io_initiation_time_max": "0",
    "write_io_initiation_time_min": "34",
    "write_io_initiation_time_max": "116",
    "read_io_inter_gap_time_min": "0",
    "read_io_inter_gap_time_max": "0",
    "write_io_inter_gap_time_min": "31400",
    "write_io_inter_gap_time_max": "118222",
    "peak_active_io_read_count": "0",
    "peak_active_io_write_count": "5",
    "read_io_aborts": "0",
    "write_io_aborts": "0",
    "read_io_failures": "0",
    "write_io_failures": "0",
    "read_io_scsi_check_condition_count": "0",
    "write_io_scsi_check_condition_count": "0",
    "read_io_scsi_busy_count": "0",
    "write_io_scsi_busy_count": "0",
    "read_io_scsi_reservation_conflict_count": "0",
    "write_io_scsi_reservation_conflict_count": "0",
    "read_io_scsi_queue_full_count": "0",
    "write_io_scsi_queue_full_count": "0",
    "sampling_start_time": "1528535447",
    "sampling_end_time": "1528697457"
  },
  .
  .
  .
  "5": {
    "port": "fc1/8",
    "vsan": "10",
    "app_id": "255",
    "initiator_id": "0xe80001",
    "target_id": "0xe800a1",
    "lun": "0000-0000-0000-0000",
    "active_io_read_count": "0",
    "active_io_write_count": "1",
    "total_read_io_count": "0",
    "total_write_io_count": "1138738309",
    "total_seq_read_io_count": "0",
    "total_seq_write_io_count": "1",
    "total_read_io_time": "0",
    "total_write_io_time": "109792480881",
    "total_read_io_initiation_time": "0",
    "total_write_io_initiation_time": "39239145641",
    "total_read_io_bytes": "0",
    "total_write_io_bytes": "583034014208",
    "total_read_io_inter_gap_time": "0",
    "total_write_io_inter_gap_time": "41479779998852",
  }
}

```

```

"total_time_metric_based_read_io_count": "0",
"total_time_metric_based_write_io_count": "1138738307",
"total_time_metric_based_read_io_bytes": "0",
"total_time_metric_based_write_io_bytes": "583034013184",
"read_io_rate": "0",
"peak_read_io_rate": "0",
"write_io_rate": "7074",
"peak_write_io_rate": "7903",
"read_io_bandwidth": "0",
"peak_read_io_bandwidth": "0",
"write_io_bandwidth": "3622144",
"peak_write_io_bandwidth": "4046336",
"read_io_size_min": "0",
"read_io_size_max": "0",
"write_io_size_min": "512",
"write_io_size_max": "512",
"read_io_completion_time_min": "0",
"read_io_completion_time_max": "0",
"write_io_completion_time_min": "71",
"write_io_completion_time_max": "3352",
"read_io_initiation_time_min": "0",
"read_io_initiation_time_max": "0",
"write_io_initiation_time_min": "26",
"write_io_initiation_time_max": "2427",
"read_io_inter_gap_time_min": "0",
"read_io_inter_gap_time_max": "0",
"write_io_inter_gap_time_min": "25988",
"write_io_inter_gap_time_max": "868452",
"peak_active_io_read_count": "0",
"peak_active_io_write_count": "5",
"read_io_aborts": "0",
"write_io_aborts": "0",
"read_io_failures": "0",
"write_io_failures": "0",
"read_io_scsi_check_condition_count": "0",
"write_io_scsi_check_condition_count": "0",
"read_io_scsi_busy_count": "0",
"write_io_scsi_busy_count": "0",
"read_io_scsi_reservation_conflict_count": "0",
"write_io_scsi_reservation_conflict_count": "0",
"read_io_scsi_queue_full_count": "0",
"write_io_scsi_queue_full_count": "0",
"sampling_start_time": "1528535447",
"sampling_end_time": "1528697457"
}
}}

```

次に、NVMe イニシエータ ITN フロー ビュー インスタンスのすべてのフローメトリックの出力の例を示します。

```

switch# show analytics query 'select all from fc-nvme.nvme_initiator_itn_flow'
{ "values": {
  "1": {
    "port": "fc1/9",
    "vsan": "5",
    "app_id": "255",
    "initiator_id": "0xa40160",
    "target_id": "0xa4018c",
    "connection_id": "0000-0000-0000-0000",
    "namespace_id": "1",
    "active_io_read_count": "0",
    "active_io_write_count": "0",

```

```

"total_read_io_count": "414106348",
"total_write_io_count": "0",
"total_seq_read_io_count": "0",
"total_seq_write_io_count": "0",
"total_read_io_time": "204490863437",
"total_write_io_time": "0",
"total_read_io_initiation_time": "132775579977",
"total_write_io_initiation_time": "0",
"total_read_io_bytes": "16226866588672",
"total_write_io_bytes": "0",
"total_read_io_inter_gap_time": "19198018763772",
"total_write_io_inter_gap_time": "0",
"total_time_metric_based_read_io_count": "414106244",
"total_time_metric_based_write_io_count": "0",
"total_time_metric_based_read_io_bytes": "16226860198912",
"total_time_metric_based_write_io_bytes": "0",
"read_io_rate": "0",
"peak_read_io_rate": "16826",
"write_io_rate": "0",
"peak_write_io_rate": "0",
"read_io_bandwidth": "0",
"peak_read_io_bandwidth": "656438400",
"write_io_bandwidth": "0",
"peak_write_io_bandwidth": "0",
"read_io_size_min": "1024",
"read_io_size_max": "262144",
"write_io_size_min": "0",
"write_io_size_max": "0",
"read_io_completion_time_min": "16",
"read_io_completion_time_max": "7057",
"write_io_completion_time_min": "0",
"write_io_completion_time_max": "0",
"read_io_initiation_time_min": "16",
"read_io_initiation_time_max": "5338",
"write_io_initiation_time_min": "0",
"write_io_initiation_time_max": "0",
"read_io_inter_gap_time_min": "32",
"read_io_inter_gap_time_max": "83725169",
"write_io_inter_gap_time_min": "0",
"write_io_inter_gap_time_max": "0",
"peak_active_io_read_count": "11",
"peak_active_io_write_count": "0",
"read_io_aborts": "24",
"write_io_aborts": "0",
"read_io_failures": "80",
"write_io_failures": "0",
"read_io_timeouts": "0",
"write_io_timeouts": "0",
"read_io_nvme_lba_out_of_range_count": "0",
"write_io_nvme_lba_out_of_range_count": "0",
"read_io_nvme_ns_not_ready_count": "0",
"write_io_nvme_ns_not_ready_count": "0",
"read_io_nvme_reservation_conflict_count": "0",
"write_io_nvme_reservation_conflict_count": "0",
"read_io_nvme_capacity_exceeded_count": "0",
"write_io_nvme_capacity_exceeded_count": "0",
"sampling_start_time": "1512847422",
"sampling_end_time": "1513166516"
},
.
.
.
"5": {
    "port": "fc1/9",

```

```
"vsan": "5",
"app_id": "255",
"initiator_id": "0xa40165",
"target_id": "0xa40190",
"connection_id": "0000-0000-0000-0000",
"namespace_id": "1",
"active_io_read_count": "0",
"active_io_write_count": "0",
"total_read_io_count": "33391955",
"total_write_io_count": "643169087",
"total_seq_read_io_count": "0",
"total_seq_write_io_count": "0",
"total_read_io_time": "13005795783",
"total_write_io_time": "131521212441",
"total_read_io_initiation_time": "5696099596",
"total_write_io_initiation_time": "71938348902",
"total_read_io_bytes": "1309083368448",
"total_write_io_bytes": "329302572544",
"total_read_io_inter_gap_time": "19175084866843",
"total_write_io_inter_gap_time": "19182318062480",
"total_time_metric_based_read_io_count": "33391919",
"total_time_metric_based_write_io_count": "643168808",
"total_time_metric_based_read_io_bytes": "1309074355200",
"total_time_metric_based_write_io_bytes": "329302429696",
"read_io_rate": "0",
"peak_read_io_rate": "574",
"write_io_rate": "0",
"peak_write_io_rate": "9344",
"read_io_bandwidth": "0",
"peak_read_io_bandwidth": "19122176",
"write_io_bandwidth": "0",
"peak_write_io_bandwidth": "4784384",
"read_io_size_min": "1024",
"read_io_size_max": "262144",
"write_io_size_min": "512",
"write_io_size_max": "512",
"read_io_completion_time_min": "16",
"read_io_completion_time_max": "5123",
"write_io_completion_time_min": "27",
"write_io_completion_time_max": "2254",
"read_io_initiation_time_min": "16",
"read_io_initiation_time_max": "3650",
"write_io_initiation_time_min": "12",
"write_io_initiation_time_max": "1377",
"read_io_inter_gap_time_min": "32",
"read_io_inter_gap_time_max": "3234375975",
"write_io_inter_gap_time_min": "32",
"write_io_inter_gap_time_max": "38886219",
"peak_active_io_read_count": "6",
"peak_active_io_write_count": "16",
"read_io_aborts": "6",
"write_io_aborts": "18",
"read_io_failures": "30",
"write_io_failures": "261",
"read_io_timeouts": "0",
"write_io_timeouts": "0",
"read_io_nvme_lba_out_of_range_count": "0",
"write_io_nvme_lba_out_of_range_count": "0",
"read_io_nvme_ns_not_ready_count": "0",
"write_io_nvme_ns_not_ready_count": "0",
"read_io_nvme_reservation_conflict_count": "0",
"write_io_nvme_reservation_conflict_count": "0",
"read_io_nvme_capacity_exceeded_count": "0",
"write_io_nvme_capacity_exceeded_count": "0",
```

```

        "sampling_start_time": "1512847422",
        "sampling_end_time": "1513166516"
    }
}}

```

次の例は、イニシエータ ITL フロー ビュー インスタンスの特定のイニシエータ ID に対する特定のフロー メトリックの出力を示しています。

```

switch# show analytics query 'select
port,initiator_id,target_id,lun,total_read_io_count,total_write_io_count,read_io_rate,write_io_rate
from fc-scsi.scsi_initiator_itl_flow where initiator_id=0xe80001'
{ "values": {
    "1": {
        "port": "fc1/8",
        "initiator_id": "0xe80001",
        "target_id": "0xe800a1",
        "lun": "0000-0000-0000-0000",
        "total_read_io_count": "0",
        "total_write_io_count": "1139010960",
        "read_io_rate": "0",
        "write_io_rate": "7071",
        "sampling_start_time": "1528535447",
        "sampling_end_time": "1528697495"
    }
}}

```

次の例は、イニシエータ ITL フロー ビュー インスタンスの特定のイニシエータ ID と LUN に対する特定のフロー メトリックの出力を示しています。

```

switch# show analytics query 'select
port,initiator_id,target_id,lun,total_read_io_count,total_write_io_count,read_io_rate,write_io_rate
from fc-scsi.scsi_initiator_itl_flow where initiator_id=0xe80001 and
lun=0000-0000-0000-0000'
{ "values": {
    "1": {
        "port": "fc1/8",
        "initiator_id": "0xe80001",
        "target_id": "0xe800a1",
        "lun": "0000-0000-0000-0000",
        "total_read_io_count": "0",
        "total_write_io_count": "1139453979",
        "read_io_rate": "0",
        "write_io_rate": "7070",
        "sampling_start_time": "1528535447",
        "sampling_end_time": "1528697559"
    }
}}

```

次に、特定の LUN の特定のフロー メトリックの出力とターゲット ITL フロー ビュー インスタンスの write_io_rate メトリックのソートされた出力の例を示します。

```

switch# show analytics query 'select
port,initiator_id,target_id,lun,total_read_io_count,total_write_io_count,read_io_rate,write_io_rate
from fc-scsi.scsi_target_itl_flow where lun=0000-0000-0000-0000 sort write_io_rate'
{ "values": {
    "1": {
        "port": "fc1/6",

```


次に、出力が 3 レコードに制限されている、特定の LUN の特定のフローメトリックの出力とイニシエータ ITL フロービューインスタンスの write_io_rate メトリックのソートされた出力の例を示します。

```
switch# show analytics query 'select
port,initiator_id,target_id,lun,total_read_io_count,total_write_io_count,read_io_rate,write_io_rate
from fc-scsi.scsi_initiator_itl_flow where lun=0000-0000-0000-0000 sort write_io_rate
limit 3'
{ "values": {
  "1": {
    "port": "fc1/6",
    "initiator_id": "0xe80020",
    "target_id": "0xd60040",
    "lun": "0000-0000-0000-0000",
    "total_read_io_count": "0",
    "total_write_io_count": "1103901828",
    "read_io_rate": "0",
    "write_io_rate": "6885",
    "sampling_start_time": "1528535447",
    "sampling_end_time": "1528697704"
  },
  "2": {
    "port": "fc1/8",
    "initiator_id": "0xe80000",
    "target_id": "0xe80042",
    "lun": "0000-0000-0000-0000",
    "total_read_io_count": "0",
    "total_write_io_count": "1120018575",
    "read_io_rate": "0",
    "write_io_rate": "6940",
    "sampling_start_time": "1528535447",
    "sampling_end_time": "1528697704"
  },
  "3": {
    "port": "fc1/6",
    "initiator_id": "0xe80021",
    "target_id": "0xe80056",
    "lun": "0000-0000-0000-0000",
    "total_read_io_count": "0",
    "total_write_io_count": "1119711583",
    "read_io_rate": "0",
    "write_io_rate": "6942",
    "sampling_start_time": "1528535447",
    "sampling_end_time": "1528697704"
  }
}
}}
```

次に、イニシエータ ITL フロービューインスタンスの特定の LUN とターゲット ID に関する特定のフローメトリックの出力の例を示します。

```
switch# show analytics query 'select
port,initiator_id,target_id,lun,total_read_io_count,total_write_io_count,read_io_rate,write_io_rate
from fc-scsi.scsi_initiator_itl_flow where lun=0000-0000-0000-0000 and target_id=0xe800a1'
{ "values": {
  "1": {
    "port": "fc1/8",
    "initiator_id": "0xe80001",
    "target_id": "0xe800a1",
    "lun": "0000-0000-0000-0000",
    "total_read_io_count": "0",
```

```

        "total_write_io_count": "1139010960",
        "read_io_rate": "0",
        "write_io_rate": "7071"
        "sampling_start_time": "1528535447",
        "sampling_end_time": "1528697495"
    }
}

```

次の例は、イニシエータ ITL フロー ビュー インスタンスの VMID 4 およびイニシエータ ID 0x0900e0 の特定のフロー メトリックの出力を示しています。

```

switch# show analytics query "select port,vsan,initiator_id,vmid from
fc-scsi.scsi_initiator_itl_flow where initiator_id=0x0900e0 and vmid=4"
{ "values": {
    "1": {
        "port": "fc2/9",
        "vsan": "1",
        "initiator_id": "0x0900e0",
        "vmid": "4",
        "sampling_start_time": "1589269530",
    }
}
}

```

次の例は、フロー メトリックの更新期間がデフォルトの期間の 30 秒に設定されている場合のプッシュクエリを設定を示します。

```

switch# configure terminal
switch(config)# analytics query 'select all from fc-scsi.scsi_initiator_itl_flow' name
initiator_itl_flow type periodic
switch(config)# show analytics query name initiator_itl_flow result
{ "values": {
    "1": {
        "port": "fcl/1",
        "vsan": "10",
        "app_id": "255",
        "initiator_id": "0xe80041",
        "target_id": "0xd60200",
        "lun": "0000-0000-0000-0000",
        "active_io_read_count": "0",
        "active_io_write_count": "1",
        "total_read_io_count": "0",
        "total_write_io_count": "1162370362",
        "total_seq_read_io_count": "0",
        "total_seq_write_io_count": "1",
        "total_read_io_time": "0",
        "total_write_io_time": "116204704658",
        "total_read_io_initiation_time": "0",
        "total_write_io_initiation_time": "43996934029",
        "total_read_io_bytes": "0",
        "total_write_io_bytes": "595133625344",
        "total_read_io_inter_gap_time": "0",
        "total_write_io_inter_gap_time": "41139462314556",
        "total_time_metric_based_read_io_count": "0",
        "total_time_metric_based_write_io_count": "1162370358",
        "total_time_metric_based_read_io_bytes": "0",
        "total_time_metric_based_write_io_bytes": "595133623296",
        "read_io_rate": "0",
        "peak_read_io_rate": "0",
        "write_io_rate": "7250",
        "peak_write_io_rate": "7304",
    }
}
}

```



```

"read_io_bandwidth": "0",
"peak_read_io_bandwidth": "0",
"write_io_bandwidth": "3712384",
"peak_write_io_bandwidth": "3739904",
"read_io_size_min": "0",
"read_io_size_max": "0",
"write_io_size_min": "512",
"write_io_size_max": "512",
"read_io_completion_time_min": "0",
"read_io_completion_time_max": "0",
"write_io_completion_time_min": "89",
"write_io_completion_time_max": "416",
"read_io_initiation_time_min": "0",
"read_io_initiation_time_max": "0",
"write_io_initiation_time_min": "34",
"write_io_initiation_time_max": "116",
"read_io_inter_gap_time_min": "0",
"read_io_inter_gap_time_max": "0",
"write_io_inter_gap_time_min": "31400",
"write_io_inter_gap_time_max": "118222",
"peak_active_io_read_count": "0",
"peak_active_io_write_count": "5",
"read_io_aborts": "0",
"write_io_aborts": "0",
"read_io_failures": "0",
"write_io_failures": "0",
"read_io_scsi_check_condition_count": "0",
"write_io_scsi_check_condition_count": "0",
"read_io_scsi_busy_count": "0",
"write_io_scsi_busy_count": "0",
"read_io_scsi_reservation_conflict_count": "0",
"write_io_scsi_reservation_conflict_count": "0",
"read_io_scsi_queue_full_count": "0",
"write_io_scsi_queue_full_count": "0",
"sampling_start_time": "1528535447",
"sampling_end_time": "1528697457"
},
.
.
.
"5": {
  "port": "fc1/8",
  "vsan": "10",
  "app_id": "255",
  "initiator_id": "0xe80001",
  "target_id": "0xe800a1",
  "lun": "0000-0000-0000-0000",
  "active_io_read_count": "0",
  "active_io_write_count": "1",
  "total_read_io_count": "0",
  "total_write_io_count": "1138738309",
  "total_seq_read_io_count": "0",
  "total_seq_write_io_count": "1",
  "total_read_io_time": "0",
  "total_write_io_time": "109792480881",
  "total_read_io_initiation_time": "0",
  "total_write_io_initiation_time": "39239145641",
  "total_read_io_bytes": "0",
  "total_write_io_bytes": "583034014208",
  "total_read_io_inter_gap_time": "0",
  "total_write_io_inter_gap_time": "41479779998852",
  "total_time_metric_based_read_io_count": "0",
  "total_time_metric_based_write_io_count": "1138738307",
  "total_time_metric_based_read_io_bytes": "0",

```



```
"total_write_io_count": "84701",
"total_seq_read_io_count": "0",
"total_seq_write_io_count": "1",
"total_read_io_time": "0",
"total_write_io_time": "7007132",
"total_read_io_initiation_time": "0",
"total_write_io_initiation_time": "2421756",
"total_read_io_bytes": "0",
"total_write_io_bytes": "86733824",
"total_read_io_inter_gap_time": "0",
"total_write_io_inter_gap_time": "2508109021",
"total_time_metric_based_read_io_count": "0",
"total_time_metric_based_write_io_count": "84701",
"total_time_metric_based_read_io_bytes": "0",
"total_time_metric_based_write_io_bytes": "86733824",
"read_io_rate": "0",
"peak_read_io_rate": "0",
"write_io_rate": "8711",
"peak_write_io_rate": "8711",
"read_io_bandwidth": "0",
"peak_read_io_bandwidth": "0",
"write_io_bandwidth": "8920576",
"peak_write_io_bandwidth": "8920576",
"read_io_size_min": "0",
"read_io_size_max": "0",
"write_io_size_min": "1024",
"write_io_size_max": "1024",
"read_io_completion_time_min": "0",
"read_io_completion_time_max": "0",
"write_io_completion_time_min": "74",
"write_io_completion_time_max": "844",
"read_io_initiation_time_min": "0",
"read_io_initiation_time_max": "0",
"write_io_initiation_time_min": "24",
"write_io_initiation_time_max": "775",
"read_io_inter_gap_time_min": "0",
"read_io_inter_gap_time_max": "0",
"write_io_inter_gap_time_min": "26903",
"write_io_inter_gap_time_max": "287888",
"peak_active_io_read_count": "0",
"peak_active_io_write_count": "3",
"read_io_aborts": "0",
"write_io_aborts": "0",
"read_io_failures": "0",
"write_io_failures": "0",
"read_io_scsi_check_condition_count": "0",
"write_io_scsi_check_condition_count": "0",
"read_io_scsi_busy_count": "0",
"write_io_scsi_busy_count": "0",
"read_io_scsi_reservation_conflict_count": "0",
"write_io_scsi_reservation_conflict_count": "0",
"read_io_scsi_queue_full_count": "0",
"write_io_scsi_queue_full_count": "0",
"sampling_start_time": "1530683133",
"sampling_end_time": "1530684301"
},
}}
```



(注) **show analytics query "query_string" clear** コマンドは、clear-on-push または clear-on-pull コマンドです。そのため、このコマンドの初回実行時にはこのコマンドは適用されません。

- 次の例は、最小、最大、およびピーク フロー メトリックをすべてクリアした後の出力を示します。クリアされたメトリックは出力で強調表示されています。

```
switch# show analytics query "select all from fc-scsi.scsi_target_itl_flow where
port=fcl/17" clear
{ "values": {
  "1": {
    "port": "fcl/17",
    "vsan": "1",
    "app_id": "255",
    "target_id": "0xef0040",
    "initiator_id": "0xef0000",
    "lun": "0000-0000-0000-0000",
    "active_io_read_count": "0",
    "active_io_write_count": "0",
    "total_read_io_count": "0",
    "total_write_io_count": "800615",
    "total_seq_read_io_count": "0",
    "total_seq_write_io_count": "1",
    "total_read_io_time": "0",
    "total_write_io_time": "66090290",
    "total_read_io_initiation_time": "0",
    "total_write_io_initiation_time": "22793874",
    "total_read_io_bytes": "0",
    "total_write_io_bytes": "819829760",
    "total_read_io_inter_gap_time": "0",
    "total_write_io_inter_gap_time": "23702347887",
    "total_time_metric_based_read_io_count": "0",
    "total_time_metric_based_write_io_count": "800615",
    "total_time_metric_based_read_io_bytes": "0",
    "total_time_metric_based_write_io_bytes": "819829760",
    "read_io_rate": "0",
    "peak_read_io_rate": "0",
    "write_io_rate": "0",
    "peak_write_io_rate": "0",
    "read_io_bandwidth": "0",
    "peak_read_io_bandwidth": "0",
    "write_io_bandwidth": "0",
    "peak_write_io_bandwidth": "0",
    "read_io_size_min": "0",
    "read_io_size_max": "0",
    "write_io_size_min": "0",
    "write_io_size_max": "0",
    "read_io_completion_time_min": "0",
    "read_io_completion_time_max": "0",
    "write_io_completion_time_min": "0",
    "write_io_completion_time_max": "0",
    "read_io_initiation_time_min": "0",
    "read_io_initiation_time_max": "0",
    "write_io_initiation_time_min": "0",
    "write_io_initiation_time_max": "0",
    "read_io_inter_gap_time_min": "0",
    "read_io_inter_gap_time_max": "0",
    "write_io_inter_gap_time_min": "0",
```

```

        "write_io_inter_gap_time_max": "0",
        "peak_active_io_read_count": "0",
        "peak_active_io_write_count": "0",
        "read_io_aborts": "0",
        "write_io_aborts": "0",
        "read_io_failures": "0",
        "write_io_failures": "0",
        "read_io_scsi_check_condition_count": "0",
        "write_io_scsi_check_condition_count": "0",
        "read_io_scsi_busy_count": "0",
        "write_io_scsi_busy_count": "0",
        "read_io_scsi_reservation_conflict_count": "0",
        "write_io_scsi_reservation_conflict_count": "0",
        "read_io_scsi_queue_full_count": "0",
        "write_io_scsi_queue_full_count": "0",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530684428"
    },
}
}

```

次の例は、ストリーミング サンプル間隔の間に変更された ITL フローメトリックのみのストリーミングを示します。

- 次の例は、[差分 (Differential)] オプションを使用する前の出力を示します。

```

switch# show analytics query "select port, target_id,
initiator_id,lun,total_write_io_count from fc-scsi.scsi_target_itl_flow where
port=fc1/17" differential
{ "values": {
    "1": {
        "port": "fc1/17",
        "target_id": "0xef0040",
        "initiator_id": "0xef0000",
        "lun": "0001-0000-0000-0000",
        "total_write_io_count": "1515601",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530683484"
    },
    "2": {
        "port": "fc1/17",
        "target_id": "0xef0040",
        "initiator_id": "0xef0020",
        "lun": "0000-0000-0000-0000",
        "total_write_io_count": "1515601",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530683484"
    },
    "3": {
        "port": "fc1/17",
        "target_id": "0xef0040",
        "initiator_id": "0xef0020",
        "lun": "0001-0000-0000-0000",
        "total_write_io_count": "1515600",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530683484"
    },
    "4": {
        "port": "fc1/17",
        "target_id": "0xef0040",
        "initiator_id": "0xef0000",
        "lun": "0000-0000-0000-0000",

```

```

        "total_write_io_count": "1515600",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530683484"
    }
}

```

- 次の例は、[差分 (Differential)] オプションを使用した出力、および変更されたレコードのみの出力を示します。

```

switch# show analytics query "select port, target_id,
initiator_id,lun,total_write_io_count from fc-scsi.scsi_target_itl_flow where
port=fc1/17" differential
{ "values": {
    "1": {
        "port": "fc1/17",
        "target_id": "0xef0040",
        "initiator_id": "0xef0000",
        "lun": "0001-0000-0000-0000",
        "total_write_io_count": "1892021",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530683534"
    },
    "2": {
        "port": "fc1/17",
        "target_id": "0xef0040",
        "initiator_id": "0xef0020",
        "lun": "0000-0000-0000-0000",
        "total_write_io_count": "1892021",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530683534"
    },
    "3": {
        "port": "fc1/17",
        "target_id": "0xef0040",
        "initiator_id": "0xef0000",
        "lun": "0000-0000-0000-0000",
        "total_write_io_count": "1892021",
        "sampling_start_time": "1530683133",
        "sampling_end_time": "1530683534"
    }
}
}

```

次の例は、インストールされたクエリ名の削除方法を示します。

```
switch(config)# no analytics name initiator_itl_flow
```

次の例は、フローメトリックのクリア方法を示します。

1. 次に、フローメトリックをクリアする前の出力の例を示します。

```

switch# show analytics query "select port,target_id,total_write_io_count,
total_write_io_bytes,total_time_metric_based_write_io_count,write_io_rate,
peak_write_io_rate,write_io_bandwidth,peak write_io_bandwidth,
write_io_size_min,write_io_size_max,write_io_completion_time_min,
write_io_completion_time_max,write_io_initiation_time_min,
write_io_initiation_time_max,write_io_inter_gap_time_min,write_io_inter_gap_time_max
from fc-scsi.scsi_target where

```

```
target_id=0x650060"
{ "values": {
  "1": {
    "port": "fc3/17",
    "target_id": "0x650060",
    "total_write_io_count": "67350021",
    "total_write_io_bytes": "17655403905024",
    "total_time_metric_based_write_io_count": "67349761",
    "write_io_rate": "0",
    "peak_write_io_rate": "6300",
    "write_io_bandwidth": "0",
    "peak_write_io_bandwidth": "1651572736",
    "write_io_size_min": "262144",
    "write_io_size_max": "262144",
    "write_io_completion_time_min": "192",
    "write_io_completion_time_max": "9434",
    "write_io_initiation_time_min": "21",
    "write_io_initiation_time_max": "199",
    "write_io_inter_gap_time_min": "2553",
    "write_io_inter_gap_time_max": "358500",
    "sampling_start_time": "1531204359",
    "sampling_end_time": "1531215327"
  }
}
```

2. 次に、フローメトリックをクリアする例を示します。



- (注) メトリックのクリアは、ビューインスタンスでのみ可能で、個々のフローメトリックではできません。

```
switch# clear analytics query "select all from fc-scsi.scsi_target where
target_id=0x650060"
```

3. 次に、フローメトリックをクリアした後の出力の例を示します。

```
switch# show analytics query "select port,target_id,total_write_io_count,
total_write_io_bytes,total_time_metric_based_write_io_count,write_io_rate,
peak_write_io_rate,write_io_bandwidth,peak_write_io_bandwidth,
write_io_size_min,write_io_size_max,write_io_completion_time_min,
write_io_completion_time_max,write_io_initiation_time_min,
write_io_initiation_time_max,write_io_inter_gap_time_min,write_io_inter_gap_time_max

from fc-scsi.scsi_target where target_id=0x650060"
{ "values": {
  "1": {
    "port": "fc3/17",
    "target_id": "0x650060",
    "total_write_io_count": "0",
    "total_write_io_bytes": "0",
    "total_time_metric_based_write_io_count": "0",
    "write_io_rate": "0",
    "peak_write_io_rate": "0",
    "write_io_bandwidth": "0",
    "peak_write_io_bandwidth": "0",
    "write_io_size_min": "0",
    "write_io_size_max": "0",
    "write_io_completion_time_min": "0",
    "write_io_completion_time_max": "0",
```

```

        "write_io_initiation_time_min": "0",
        "write_io_initiation_time_max": "0",
        "write_io_inter_gap_time_min": "0",
        "write_io_inter_gap_time_max": "0",
        "sampling_start_time": "1531204359",
        "sampling_end_time": "1531215464"
    }

```

次に、フロー メトリックを消去した後の出力の例を示します。



(注) `port` キー値は、消去メトリックの **where** 句でのみ使用できます。

```

switch# purge analytics query "select all from fc-scsi.scsi_target where port=fc3/17"
switch# show analytics query "select all from fc-scsi.scsi_target where port=fc3/17"
Table is empty for query "select all from fc-scsi.scsi_target where port=fc3/17"

```

ShowAnalytics オーバーレイ CLI の使用

ShowAnalytics オーバーレイ CLI は、JSON 形式の分析データを使いやすい表形式に変換するために使用されます。ShowAnalytics オーバーレイ CLI には「Linux と同等」の構文があり、組み込みの NX-OS Python インタープリタを使用して、プルクエリの JSON 出力を表形式に変換するスクリプトを実行します。現時点では、フローメトリックの小さなサブセットのみ表示されます。



- (注)
- **ShowAnalytics** オーバーレイ コマンド では、Exchange Completion Time (ECT) for the `--initiator-itl` and `--target-itloptions` under the `--info` オプションの累積データが表示されます。ただし、レートおよび帯域幅メトリックについては瞬間的なデータが表示されます。
 - アクティブな ITL 数が記載されている制限を超えている場合、**ShowAnalytics** オーバーレイ コマンドは警告を表示して終了します。ITL 数の制限の詳細については、[Cisco MDS NX-OS の設定の制限、リリース 8.x](#) ドキュメントを参照してください。
 - Virtual Instruments 社や Cisco DCNM によって推奨されている **clear** キーワードを指定してプッシュクエリを設定する場合、最小と最大のフローメトリックの値は不正確になります。
 - ShowAnalytics コマンドの `--evaluate-npload` オプションは、SCSI 分析タイプのみをサポートします。ただし、SCSI 分析タイプと NVMe 分析タイプの両方をサポートするオプションは除きます。
 - インターフェイスで分析タイプを構成する前に、`--evaluate-npload` オプションを実行します。モジュールのインターフェイスの1つが分析タイプで構成されている場合でも、この `--evaluate-npload` オプションはモジュールでは機能しません。
 - この `--outstanding-io` オプションは F ポートでのみ機能します。

分析情報を表形式で表示するには、次のコマンドを実行します。

```
switch# ShowAnalytics -help
```

詳細については、『[Cisco MDS 9000 シリーズ コマンド リファレンス 8.x](#)』を参照してください。

例 : ShowAnalytics オーバーレイ CLI の使用

次の例は、オーバーレイ CLI のオプションを示します：



- (注) オーバーレイ CLI で使用可能なキーワードと変数を表示するオプションと、Cisco MDS NX-OS リリース 8.4 (2) 以降から追加されたそのオプション。

```
switch# ShowAnalytics ?
ShowAnalytics           Aliased to 'source sys/analytcs.py'
ShowAnalyticsConsistency Aliased to 'source sys/analytcs_pss_consistency_checker.py'

--errors                To display errors stats in all IT(L/N) pairs
--erroronly             To display IT(L/N) flows with errors
--evaluate-npload       To evaluate npload on system
--help                  To display help and exit
--info                  To display information about IT(L/N) flows
--minmax                To display min max and peak info about IT(L/N) flows
--outstanding-io        To display outstanding io for an interface
```

例: ShowAnalytics オーバーレイ CLI の使用

```

--top                To display top 10 IT(L/N) Flow
--version            To display version of utility and exit
--vsan-thput         To display per vsan throughput for interface

```

次に、オーバーレイ CLI バージョンの表示例を示します。

```

switch# ShowAnalytics --version
ShowAnalytics 4.0.0

```

次に、NVMe のイニシエータのフロー メトリックの表示方法の例を示します：

```

switch# ShowAnalytics --info --initiator-itn
2021-02-09 09:01:39.714290

Interface fcl6/12

```

VSan Size	Initiator	Target	Namespace	Avg IOPS		Avg Throughput		Avg ECT		Avg DAL		Avg IO
				Read	Write	Read	Write	Read	Write	Read	Write	
3300	0xc80002	0xed0002	1	2531	2478	158.2 MB/s	154.9 MB/s	781.0 us	2.0 ms	636.0 us	633.0 us	64.0 KB
64.0 KB		690.0 us		562.0 us								
3300	0xc80007	0xed0007	1	2508	2497	156.8 MB/s	156.1 MB/s	764.0 us	2.0 ms	622.0 us	630.0 us	64.0 KB
64.0 KB		683.0 us		572.0 us								
3300	0xc80005	0xed0005	1	2421	2548	151.3 MB/s	159.3 MB/s	785.0 us	2.0 ms	640.0 us	625.0 us	64.0 KB
64.0 KB		686.0 us		561.0 us								
3300	0xc80001	0xed0001	1	2060	2149	128.8 MB/s	134.3 MB/s	764.0 us	1.9 ms	621.0 us	596.0 us	64.0 KB
64.0 KB		649.0 us		537.0 us								
3300	0xc80000	0xed0000	1	333	364	20.8 MB/s	22.8 MB/s	14.8 ms	16.1 ms	14.6 ms	15.3 ms	64.0 KB
64.0 KB		190.0 us		521.0 us								
3300	0xc80008	0xed0008	1	2483	2503	155.2 MB/s	156.4 MB/s	771.0 us	2.0 ms	626.0 us	639.0 us	64.0 KB
64.0 KB		685.0 us		571.0 us								
3300	0xc80009	0xed0009	1	2545	2474	159.1 MB/s	154.6 MB/s	786.0 us	2.0 ms	641.0 us	627.0 us	64.0 KB
64.0 KB		683.0 us		570.0 us								
3300	0xc80004	0xed0004	1	2506	2498	156.6 MB/s	156.1 MB/s	769.0 us	2.0 ms	625.0 us	642.0 us	64.0 KB
64.0 KB		680.0 us		575.0 us								
3300	0xc80006	0xed0006	1	2456	2512	153.5 MB/s	157.0 MB/s	793.0 us	2.0 ms	650.0 us	624.0 us	64.0 KB
64.0 KB		696.0 us		558.0 us								
3300	0xc80000	0xed0001	1	1926	1848	120.4 MB/s	115.5 MB/s	734.0 us	1.8 ms	593.0 us	572.0 us	64.0 KB
64.0 KB		533.0 us		512.0 us								
3300	0xc80003	0xed0003	1	2553	2472	159.6 MB/s	154.5 MB/s	786.0 us	2.0 ms	641.0 us	622.0 us	64.0 KB
64.0 KB		691.0 us		560.0 us								

Total number of ITNs: 11

次に、SCSI のイニシエータのフロー メトリックの表示方法の例を示します：

```

switch# ShowAnalytics --info --initiator-itl
2021-02-09 09:01:39.714290

Interface fc2/22

```

VSan	Initiator	VMID	Target	LUN	Avg IOPS		Avg Throughput		Avg ECT		Avg DAL	
					Read	Write	Read	Write	Read	Write	Read	Write
2200	0xe80ee0	-	0xc809a0	0001-0000-0000-0000	0	0	0 B/s	0 B/s	0 ns	0 ns	0 ns	0
ns	0 B	0 B		0 ns								
2200	0xe80ee0	-	0xc80622	0007-0000-0000-0000	0	0	0 B/s	0 B/s	0 ns	0 ns	0 ns	0
ns	0 B	0 B		0 ns								
2200	0xe80ee0	-	0xc809a0	0002-0000-0000-0000	0	0	0 B/s	0 B/s	0 ns	0 ns	0 ns	0
ns	0 B	0 B		0 ns								
2200	0xe80ee0	-	0xc809a0	0003-0000-0000-0000	0	0	0 B/s	0 B/s	0 ns	0 ns	0 ns	0
ns	0 B	0 B		0 ns								
2200	0xe80ee0	-	0xc80622	0002-0000-0000-0000	0	0	0 B/s	0 B/s	0 ns	0 ns	0 ns	0
ns	0 B	0 B		0 ns								
2200	0xe80ee0	18	0xc809a0	0003-0000-0000-0000	0	5	0 B/s	21.0 KB/s	0 ns	702.0 us	0 ns	251.0 us
ns	0 B	4.2 KB		7.0 us								441.0 us
2200	0xe80ee0	-	0xc80623	0004-0000-0000-0000	0	0	0 B/s	0 B/s	0 ns	0 ns	0 ns	0
ns	0 B	0 B		0 ns								
2200	0xe80ee0	-	0xc80622	0000-0000-0000-0000	0	0	0 B/s	0 B/s	0 ns	0 ns	0 ns	0
ns	0 B	0 B		0 ns								

Total number of ITLs: 8

次に、NVMe のターゲットのフロー メトリックの表示方法の例を示します：

```

switch# ShowAnalytics --info --target-itn
2021-02-09 12:14:59.285397

Interface fc3/15

```

VSAN Size	Initiator	Target	Namespace	Avg IOPS		Avg Throughput		Avg ECT		Avg DAL		Avg IO
				Read	Write	Read	Write	Read	Write	Read	Write	
	Write	Write	Write									
13300	0xc80005	0xed0005	1	2545	2457	159.1 MB/s	153.6 MB/s	112.0 us	1.5 ms	44.0 us	40.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80000	0xed0001	1	2036	2026	127.3 MB/s	126.6 MB/s	110.0 us	1.3 ms	44.0 us	39.0 us	64.0 KB
64.0 KB	1.1 ms	5.0 us										
13300	0xc80004	0xed0004	1	2464	2492	154.0 MB/s	155.8 MB/s	113.0 us	1.5 ms	45.0 us	40.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80001	0xed0001	1	2036	2020	127.2 MB/s	126.2 MB/s	112.0 us	1.3 ms	44.0 us	40.0 us	64.0 KB
64.0 KB	1.1 ms	5.0 us										
13300	0xc80003	0xed0003	1	2460	2491	153.8 MB/s	155.7 MB/s	114.0 us	1.5 ms	45.0 us	39.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80000	0xed0000	1	335	360	20.9 MB/s	22.5 MB/s	14.1 ms	15.6 ms	14.1 ms	14.7 ms	64.0 KB
64.0 KB	784.0 us	5.0 us										
13300	0xc80007	0xed0007	1	2476	2488	154.8 MB/s	155.5 MB/s	114.0 us	1.5 ms	46.0 us	39.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80008	0xed0008	1	2484	2489	155.3 MB/s	155.6 MB/s	114.0 us	1.5 ms	46.0 us	40.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80002	0xed0002	1	2472	2490	154.5 MB/s	155.6 MB/s	112.0 us	1.5 ms	45.0 us	40.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80006	0xed0006	1	2449	2507	153.1 MB/s	156.7 MB/s	116.0 us	1.5 ms	46.0 us	39.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80009	0xed0009	1	2471	2485	154.4 MB/s	155.3 MB/s	114.0 us	1.5 ms	45.0 us	40.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										

Total number of ITNs: 11

次に、SCSI のターゲットのフロー メトリックの表示方法の例を示します：

```
switch# ShowAnalytics --info --target-itl
2021-02-09 12:14:59.285397
Interface fc5/21
```

VSAN	Initiator	VMID	Target	LUN	Avg IOPS		Avg Throughput		Avg ECT		Avg DAL	
					Read	Write	Read	Write	Read	Write	Read	Write
					Read	Write	Read	Write	Read	Write	Read	Write
12200	0xe902e0	-	0xe805a0	0002-0000-0000-0000	0	9236	0 B/s	4.5 MB/s	0 ns	75.0 us	0 ns	25.0 us
	0 B	512.0 B	0 ns	0 ns								
12200	0xe902e0	-	0xe805a0	0003-0000-0000-0000	0	9235	0 B/s	4.5 MB/s	0 ns	75.0 us	0 ns	25.0 us
	0 B	511.0 B	0 ns	0 ns								
12200	0xe902e0	-	0xe805a0	0001-0000-0000-0000	0	9235	0 B/s	4.5 MB/s	0 ns	75.0 us	0 ns	25.0 us
	0 B	512.0 B	0 ns	0 ns								

Total number of ITLs: 3

次の例は、すべてのターゲット ITN を表示し、出力を NVMe の 5 つのランダム レコードに制限する方法を示しています。

```
switch# ShowAnalytics --info --target-itn --interface fc8/15 --limit 5
2019-04-09 11:11:24.652190
Interface fc3/15
```

VSAN	Initiator	Target	Namespace	Avg IOPS		Avg Throughput		Avg ECT		Avg DAL		Avg IO
				Read	Write	Read	Write	Read	Write	Read	Write	
	Write	Write	Write									
13300	0xc80005	0xed0005	1	2396	2473	149.8 MB/s	154.6 MB/s	111.0 us	1.5 ms	45.0 us	40.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80000	0xed0001	1	2180	2250	136.3 MB/s	140.7 MB/s	110.0 us	1.2 ms	43.0 us	39.0 us	64.0 KB
64.0 KB	1.1 ms	5.0 us										
13300	0xc80004	0xed0004	1	2424	2463	151.5 MB/s	154.0 MB/s	114.0 us	1.5 ms	46.0 us	39.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										
13300	0xc80001	0xed0001	1	2129	2202	133.1 MB/s	137.6 MB/s	110.0 us	1.2 ms	43.0 us	37.0 us	64.0 KB
64.0 KB	992.0 us	5.0 us										
13300	0xc80003	0xed0003	1	2457	2462	153.6 MB/s	153.9 MB/s	114.0 us	1.5 ms	46.0 us	38.0 us	64.0 KB
64.0 KB	1.3 ms	5.0 us										

Total number of ITNs: 5

次の例は、すべてのターゲット ITL を表示し、SCSI の出力を 10 のランダム レコードに制限する方法を示しています。

```
switch# ShowAnalytics --info --target-itl --interface fc8/15 --limit 10
2019-04-09 11:11:24.652190
Interface fc5/21
```

VSAN	Initiator	VMID	Target	LUN	Avg IOPS		Avg Throughput		Avg ECT		Avg DAL	
					Read	Write	Read	Write	Read	Write	Read	Write
					Read	Write	Read	Write	Read	Write	Read	Write

例 : ShowAnalytics オーバーレイ CLI の使用

	Read	Write	Write	Write	Read	Write	Read	Write	Read	Write	Read	Write
2200	0xe902e0	-	0xe805a0	0002-0000-0000-0000	0	9235	0 B/s	4.5 MB/s	0 ns	75.0 us	0 ns	25.0
us	0 B	512.0 B	0 ns	0 ns								
2200	0xe902e0	-	0xe805a0	0003-0000-0000-0000	0	9235	0 B/s	4.5 MB/s	0 ns	75.0 us	0 ns	25.0
us	0 B	512.0 B	0 ns	0 ns								
2200	0xe902e0	-	0xe805a0	0001-0000-0000-0000	0	9235	0 B/s	4.5 MB/s	0 ns	75.0 us	0 ns	25.0
us	0 B	512.0 B	0 ns	0 ns								

Total number of ITLs: 3

次に、NVMe のイニシエータ ITN の VSAN 3300 に関するフローメトリックの表示方法の例を示します：

```
switch# ShowAnalytics --info --initiator-itn --vsan 3300
2019-04-08 11:26:23.074904
```

Interface fcl6/12

VSAN Size	Initiator Avg Host Delay	Target Avg Array Delay	Namespace	Avg IOPS	Avg Throughput	Avg ECT	Avg DAL	Avg IO
				Read	Write	Read	Write	Read
3300	0xc80002	0xd00002	1	2466	2458	154.2 MB/s	153.6 MB/s	782.0 us
64.0 KB	714.0 us	567.0 us		2466	2470	154.1 MB/s	154.4 MB/s	786.0 us
3300	0xc80007	0xd00007	1	2466	2470	154.1 MB/s	154.4 MB/s	786.0 us
64.0 KB	712.0 us	561.0 us		2432	2484	152.0 MB/s	155.3 MB/s	775.0 us
3300	0xc80005	0xd00005	1	2432	2484	152.0 MB/s	155.3 MB/s	775.0 us
64.0 KB	714.0 us	564.0 us		2066	2031	129.2 MB/s	126.9 MB/s	723.0 us
3300	0xc80001	0xd00001	1	2066	2031	129.2 MB/s	126.9 MB/s	723.0 us
64.0 KB	470.0 us	507.0 us		339	347	21.2 MB/s	21.7 MB/s	15.3 ms
3300	0xc80000	0xd00000	1	339	347	21.2 MB/s	21.7 MB/s	15.3 ms
64.0 KB	190.0 us	518.0 us		2436	2480	152.2 MB/s	155.0 MB/s	777.0 us
3300	0xc80008	0xd00008	1	2436	2480	152.2 MB/s	155.0 MB/s	777.0 us
64.0 KB	708.0 us	563.0 us		2475	2459	154.7 MB/s	153.7 MB/s	772.0 us
3300	0xc80009	0xd00009	1	2475	2459	154.7 MB/s	153.7 MB/s	772.0 us
64.0 KB	700.0 us	569.0 us		2508	2448	156.8 MB/s	153.0 MB/s	775.0 us
3300	0xc80004	0xd00004	1	2508	2448	156.8 MB/s	153.0 MB/s	775.0 us
64.0 KB	704.0 us	568.0 us		2427	2485	151.7 MB/s	155.3 MB/s	778.0 us
3300	0xc80006	0xd00006	1	2427	2485	151.7 MB/s	155.3 MB/s	778.0 us
64.0 KB	713.0 us	561.0 us		2246	2218	140.4 MB/s	138.7 MB/s	744.0 us
3300	0xc80000	0xd00001	1	2246	2218	140.4 MB/s	138.7 MB/s	744.0 us
64.0 KB	561.0 us	530.0 us		2439	2478	152.4 MB/s	154.9 MB/s	776.0 us
3300	0xc80003	0xd00003	1	2439	2478	152.4 MB/s	154.9 MB/s	776.0 us
64.0 KB	711.0 us	564.0 us						

Total number of ITNs: 11

次に、SCSI のイニシエータ ITL の VSAN 2200 に関するフローメトリックの表示方法の例を示します：

```
switch# ShowAnalytics --info --initiator-itl --vsan 2200
2019-04-08 11:26:23.074904
```

Interface fc2/22

VSAN Size	Initiator Avg Host Delay	VMID	Target Avg Array Delay	LUN	Avg IOPS	Avg Throughput	Avg ECT	Avg DAL
					Read	Write	Read	Write
2200	0xe80ee0	-	0xe80622	0007-0000-0000-0000	0	0	0 B/s	0 B/s
ns	0 B	0 B	0 ns	0 ns			0 ns	0 ns
2200	0xe80ee0	-	0xc809a0	0003-0000-0000-0000	0	0	0 B/s	0 B/s
ns	0 B	0 B	0 ns	0 ns			0 ns	0 ns
2200	0xe80ee0	-	0xe80622	0002-0000-0000-0000	0	0	0 B/s	0 B/s
ns	0 B	0 B	0 ns	0 ns			0 ns	0 ns
2200	0xe80ee0	18	0xc809a0	0003-0000-0000-0000	0	0	0 B/s	2.0 KB/s
us	0 B	4.0 KB	7.0 us	656.0 us			0 ns	843.0 us
2200	0xe80ee0	-	0xe80622	0000-0000-0000-0000	0	0	0 B/s	0 B/s
ns	0 B	0 B	0 ns	0 ns			0 ns	0 ns

Total number of ITLs: 5

次に、NVMe のターゲット ITN のインターフェイス fc3/15 に関するフローメトリックの表示方法の例を示します：

```
switch# ShowAnalytics --info --target-itn --interface fc3/15
2019-04-09 11:11:17.974991
```

Interface fc3/15

VSAN Size	Initiator Avg Host Delay	Target Avg Array Delay	Namespace	Avg IOPS	Avg Throughput	Avg ECT	Avg DAL	Avg IO
-----------	--------------------------	------------------------	-----------	----------	----------------	---------	---------	--------

```

| Write | Write | Write | Read | Write | Read | Write | Read | Write | Read | Write | Read | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3300 | 0xc80005 | 0xc80005 | 1 | 2475 | 2531 | 154.7 MB/s | 158.2 MB/s | 112.0 us | 1.5 ms | 45.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80000 | 0xc80001 | 1 | 2137 | 2158 | 133.6 MB/s | 134.9 MB/s | 112.0 us | 1.4 ms | 46.0 us | 39.0 us | 64.0 KB |
| 64.0 KB | 1.2 ms | 5.0 us |
| 3300 | 0xc80004 | 0xc80004 | 1 | 2465 | 2530 | 154.1 MB/s | 158.2 MB/s | 115.0 us | 1.5 ms | 46.0 us | 39.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80001 | 0xc80001 | 1 | 1785 | 1796 | 111.6 MB/s | 112.2 MB/s | 112.0 us | 1.3 ms | 45.0 us | 38.0 us | 64.0 KB |
| 64.0 KB | 1.1 ms | 5.0 us |
| 3300 | 0xc80003 | 0xc80003 | 1 | 2512 | 2506 | 157.0 MB/s | 156.6 MB/s | 113.0 us | 1.5 ms | 45.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80000 | 0xc80000 | 1 | 355 | 329 | 22.2 MB/s | 20.6 MB/s | 14.8 ms | 15.5 ms | 14.8 ms | 14.6 ms | 64.0 KB |
| 64.0 KB | 753.0 us | 5.0 us |
| 3300 | 0xc80007 | 0xc80007 | 1 | 2465 | 2532 | 154.1 MB/s | 158.2 MB/s | 115.0 us | 1.5 ms | 47.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80008 | 0xc80008 | 1 | 2488 | 2520 | 155.5 MB/s | 157.5 MB/s | 115.0 us | 1.5 ms | 47.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80002 | 0xc80002 | 1 | 2548 | 2497 | 159.3 MB/s | 156.1 MB/s | 113.0 us | 1.5 ms | 46.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80006 | 0xc80006 | 1 | 2476 | 2523 | 154.8 MB/s | 157.7 MB/s | 113.0 us | 1.5 ms | 46.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80009 | 0xc80009 | 1 | 2487 | 2525 | 155.4 MB/s | 157.8 MB/s | 114.0 us | 1.5 ms | 46.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us |
-----
Total number of ITNs: 11

```

次に、SCSI のターゲット ITL のインターフェイス fc3/15 に関するフローメトリックの表示方法の例を示します：

```

switch# ShowAnalytics --info --target-itl --interface fc5/21
2019-04-09 11:11:17.974991

Interface fc5/21
-----
|VSAN | Initiator | VMID | Target | LUN | Avg IOPS | Avg Throughput | Avg ECT | Avg DAL | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg IO Size | Avg Host Delay | Avg Array Delay | Read | Write | Read | Write | Read | Write | Read | Write |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2200 | 0xe902e0 | - | 0xe805a0 | 0002-0000-0000-0000 | 0 | 9231 | 0 B/s | 4.5 MB/s | 0 ns | 75.0 us | 0 ns | 25.0 us |
| 0 B | 512.0 B | 0 ns | 0 ns |
| 2200 | 0xe902e0 | - | 0xe805a0 | 0003-0000-0000-0000 | 0 | 9231 | 0 B/s | 4.5 MB/s | 0 ns | 75.0 us | 0 ns | 25.0 us |
| 0 B | 512.0 B | 0 ns | 0 ns |
| 2200 | 0xe902e0 | - | 0xe805a0 | 0001-0000-0000-0000 | 0 | 9230 | 0 B/s | 4.5 MB/s | 0 ns | 75.0 us | 0 ns | 25.0 us |
| 0 B | 512.0 B | 0 ns | 0 ns |
-----
Total number of ITLs: 3

```

次に、NVMe のターゲット ITN のインターフェイス fc3/15 に関するフローメトリックとデバイス エイリアス情報と NVMe の出力を 10 のランダム レコードに制限するの表示方法の例を示します：

```

switch# ShowAnalytics --info --target-itn --alias --interface fc3/15 --limit 10
2019-04-09 12:04:07.032501

Interface fc3/15
-----
|VSAN | Initiator | Target | Namespace | Avg IOPS | Avg Throughput | Avg ECT | Avg |
|DAL | Avg IO Size | Avg Host Delay | Avg Array Delay | Read | Write | Read | Write | Read | Write | Read |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3300 | 0xc80005 | 0xc80005 | 1 | 2488 | 2514 | 155.5 MB/s | 157.1 MB/s | 113.0 us | 1.5 ms | 46.0 us |
| 39.0 us | 64.0 KB | 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80000 | 0xc80001 | 1 | 2122 | 2154 | 132.6 MB/s | 134.7 MB/s | 111.0 us | 1.4 ms | 45.0 us |
| 40.0 us | 64.0 KB | 64.0 KB | 1.2 ms | 5.0 us |
| 3300 | 0xc80004 | 0xc80004 | 1 | 2492 | 2509 | 155.8 MB/s | 156.8 MB/s | 113.0 us | 1.5 ms | 46.0 us |
| 40.0 us | 64.0 KB | 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80001 | 0xc80001 | 1 | 1847 | 1752 | 115.4 MB/s | 109.5 MB/s | 112.0 us | 1.3 ms | 45.0 us |
| 39.0 us | 64.0 KB | 64.0 KB | 1.1 ms | 5.0 us |
| 3300 | 0xc80003 | 0xc80003 | 1 | 2523 | 2495 | 157.7 MB/s | 155.9 MB/s | 114.0 us | 1.5 ms | 46.0 us |
| 41.0 us | 64.0 KB | 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80000 | 0xc80000 | 1 | 340 | 355 | 21.3 MB/s | 22.2 MB/s | 14.3 ms | 15.3 ms | 14.2 ms |
| 14.4 ms | 64.0 KB | 64.0 KB | 801.0 us | 5.0 us |
| 3300 | 0xc80007 | 0xc80007 | 1 | 2495 | 2510 | 156.0 MB/s | 156.9 MB/s | 114.0 us | 1.5 ms | 47.0 us |
| 40.0 us | 64.0 KB | 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80008 | 0xc80008 | 1 | 2515 | 2496 | 157.2 MB/s | 156.0 MB/s | 114.0 us | 1.5 ms | 47.0 us |
| 40.0 us | 64.0 KB | 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80002 | 0xc80002 | 1 | 2537 | 2484 | 158.6 MB/s | 155.3 MB/s | 114.0 us | 1.5 ms | 46.0 us |
| 41.0 us | 64.0 KB | 64.0 KB | 1.3 ms | 5.0 us |
| 3300 | 0xc80006 | 0xc80006 | 1 | 2502 | 2510 | 156.4 MB/s | 156.9 MB/s | 113.0 us | 1.5 ms | 46.0 us |
| 41.0 us | 64.0 KB | 64.0 KB | 1.3 ms | 5.0 us |
-----
Total number of ITNs: 10

```

例 : ShowAnalytics オーバーレイ CLI の使用

次に、SCSI のターゲット ITL のインターフェイス fc5/21 に関するフロー メトリックとデバイス エイリアス情報と SCSI の出力を 10 のランダム レコードに制限するの表示方法の例を示します :

```
switch# ShowAnalytics --info --target-itl --alias --interface fc5/21 --limit 10
2019-04-09 12:04:07.032501

Interface fc5/21
-----
|VSAN | Initiator | VMID | Target | LUN | Avg IOPS | Avg Throughput | Avg ECT |
| | Avg DAL | Avg IO Size | Avg Host Delay | Avg Array Delay | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Read | Write | Write | Write | Read | Write | Read | Write | Read |
|-----|-----|-----|-----|-----|-----|-----|-----|
|2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21 | 0002-0000-0000-0000 | 0 | 5796 | 0 B/s | 2.8 MB/s | 0 ns | 84.0
us | 0 ns | 29.0 us | 0 B | 512.0 B | 0 ns | 0 ns |
|2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21 | 0003-0000-0000-0000 | 0 | 5797 | 0 B/s | 2.8 MB/s | 0 ns | 84.0
us | 0 ns | 29.0 us | 0 B | 512.0 B | 0 ns | 0 ns |
|2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21 | 0001-0000-0000-0000 | 0 | 5797 | 0 B/s | 2.8 MB/s | 0 ns | 84.0
us | 0 ns | 29.0 us | 0 B | 512.0 B | 0 ns | 0 ns |
|2200 | 0xe90440 | - | Tgt_9706_206_fc5_21 | 0001-0000-0000-0000 | 0 | 5797 | 0 B/s | 2.8 MB/s | 0 ns | 122.0
us | 0 ns | 44.0 us | 0 B | 512.0 B | 0 ns | 0 ns |
|2200 | 0xe90440 | - | Tgt_9706_206_fc5_21 | 0002-0000-0000-0000 | 0 | 5796 | 0 B/s | 2.8 MB/s | 0 ns | 124.0
us | 0 ns | 44.0 us | 0 B | 512.0 B | 0 ns | 0 ns |
|2200 | 0xe906c0 | - | Tgt_9706_206_fc5_21 | 0001-0000-0000-0000 | 0 | 5797 | 0 B/s | 2.8 MB/s | 0 ns | 130.0
us | 0 ns | 47.0 us | 0 B | 512.0 B | 0 ns | 0 ns |
|2200 | 0xe906c0 | - | Tgt_9706_206_fc5_21 | 0002-0000-0000-0000 | 0 | 5796 | 0 B/s | 2.8 MB/s | 0 ns | 131.0
us | 0 ns | 48.0 us | 0 B | 512.0 B | 0 ns | 0 ns |
-----
Total number of ITLs: 7
```

次に、NVMe のターゲット ITN のターゲット識別子 0xed0001 に関するフロー メトリックの表示方法の例を示します :

```
switch# ShowAnalytics --info --target-itn --target 0xed0001
2019-04-09 11:16:26.246741

Interface fc3/15
-----
|VSAN | Initiator | Target | Namespace | Avg IOPS | Avg Throughput | Avg ECT | Avg DAL | Avg IO
Size | Avg Host Delay | Avg Array Delay | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | Write | Write | Write | Read | Write | Read | Write | Read | Write | Read | Write | Read |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|13300 | 0xc80000 | 0xed0001 | 1 | 2100 | 2173 | 131.2 MB/s | 135.8 MB/s | 110.0 us | 1.4 ms | 44.0 us | 38.0 us | 64.0 KB |
64.0 KB | 1.2 ms | 5.0 us |
|13300 | 0xc80001 | 0xed0001 | 1 | 1964 | 1943 | 122.8 MB/s | 121.4 MB/s | 109.0 us | 1.2 ms | 43.0 us | 38.0 us | 64.0 KB |
64.0 KB | 1.0 ms | 5.0 us |
-----
Total number of ITNs: 2
```

次に、SCSI のターゲット ITL のターゲット識別子 0xe80b40 に関するフロー メトリックの表示方法の例を示します :

```
switch# ShowAnalytics --info --target-itl --target 0xe80b40
2019-04-09 11:16:26.246741

Interface fc5/21
-----
|VSAN | Initiator | VMID | Target | LUN | Avg IOPS | Avg Throughput | Avg ECT | Avg DAL |
| | Avg IO Size | Avg Host Delay | Avg Array Delay | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Read | Write | Write | Write | Read | Write | Read | Write | Read | Write | Read | Write |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|2200 | 0xe90440 | - | 0xe80b40 | 0001-0000-0000-0000 | 0 | 5809 | 0 B/s | 2.8 MB/s | 0 ns | 128.0 us | 0 ns | 48.0
us | 0 B | 512.0 B | 0 ns | 0 ns |
|2200 | 0xe90440 | - | 0xe80b40 | 0002-0000-0000-0000 | 0 | 5809 | 0 B/s | 2.8 MB/s | 0 ns | 132.0 us | 0 ns | 48.0
us | 0 B | 511.0 B | 0 ns | 0 ns |
-----
Total number of ITLs: 2
```

次に、ターゲット NVMe の ITN のイニシエータ識別子 0xc80004、ターゲット識別子 0xed0004 と名前空間 1 のフロー メトリックの表示方法の例を示します :

```
switch# ShowAnalytics --info --target-itn --initiator 0xc80004 --target 0xed0004 --namespace 1
2019-04-09 11:17:24.643292

B: Bytes, s: Seconds, Avg: Average, Acc: Accumulative,
ns: Nano Seconds, ms: Milli Seconds, us: Micro Seconds,
GB: Giga Bytes, MB: Mega Bytes, KB: Killo Bytes,
ECT: Exchange Completion Time, DAL: Data Access Latency
```

```

Interface : fc3/15
+-----+-----+-----+
| Metric | Min | Max | Avg |
+-----+-----+-----+
| Read IOPS (4sec Avg) | NA | NA | 2391 |
| Write IOPS (4sec Avg) | NA | NA | 2517 |
| Read Throughput (4sec Avg) | NA | NA | 149.5 MB/s |
| Write Throughput (4sec Avg) | NA | NA | 157.3 MB/s |
| Read Size (Acc Avg) | 65536 B | 65536 B | 65536 B |
| Write Size (Acc Avg) | 65536 B | 65536 B | 65536 B |
| Read DAL (Acc Avg) | 12.0 us | 1.6 ms | 46.0 us |
| Write DAL (Acc Avg) | 10.0 us | 407.0 us | 40.1 us |
| Read ECT (Acc Avg) | 39.0 us | 1.9 ms | 113.8 us |
| Write ECT (Acc Avg) | 123.0 us | 3.6 ms | 1.5 ms |
| Write Host Delay (Acc Avg) | 51.0 us | 3.5 ms | 1.3 ms |
| Write Array Delay (Acc Avg) | NA | 31.0 us | 5.6 us |
| Write IO Seq count (Acc Avg) | 0 | 0 | 1 |
+-----+-----+-----+

```

次に、ターゲット ITL のイニシエータ識別子 0xe90440、ターゲット識別子 0xe80b40 と LUN 識別子 0001-0000-0000-0000 のフロー メトリックの表示方法の例を示します :

```

switch# ShowAnalytics --info --target-itl --initiator 0xe90440 --target 0xe80b40 --lun 0001-0000-0000-0000
2019-04-09 11:17:24.643292

B: Bytes, s: Seconds, Avg: Average, Acc: Accumulative,
ns: Nano Seconds, ms: Milli Seconds, us: Micro Seconds,
GB: Giga Bytes, MB: Mega Bytes, KB: Killo Bytes,
ECT: Exchange Completion Time, DAL: Data Access Latency

```

```

Interface : fc5/21
+-----+-----+-----+
| Metric | Min | Max | Avg |
+-----+-----+-----+
| Read IOPS (4sec Avg) | NA | NA | 0 |
| Write IOPS (4sec Avg) | NA | NA | 4112 |
| Read Throughput (4sec Avg) | NA | NA | 0 |
| Write Throughput (4sec Avg) | NA | NA | 2.0 MB/s |
| Read Size (Acc Avg) | 0 | 0 | 0 |
| Write Size (Acc Avg) | 512 B | 512 B | 512 B |
| Read DAL (Acc Avg) | 0 ns | 0 ns | 0 ns |
| Write DAL (Acc Avg) | 22.0 us | 2.4 ms | 46.1 us |
| Read ECT (Acc Avg) | 0 ns | 0 ns | 0 ns |
| Write ECT (Acc Avg) | 68.0 us | 2.5 ms | 126.6 us |
| Write Host Delay (Acc Avg) | 0 ns | 0 ns | 0 ns |
| Write Array Delay (Acc Avg) | NA | 0 ns | 0 ns |
| Write IO Seq count (Acc Avg) | 0 | 0 | 0 |
+-----+-----+-----+

```

次に、ターゲット NVMe の ITN のイニシエータ識別子 0xc80005 と名前空間 1 のフロー メトリックの表示方法の例を示します :

```

switch# ShowAnalytics --info --target-itn --initiator 0xc80005 --namespace 1
2019-04-09 11:18:40.132828

Interface fc3/15
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| VSAN | Initiator | Target | Namespace | Avg IOPS | Avg Throughput | Avg ECT | Avg DAL | Avg IO |
| Size | Avg Host Delay | Avg Array Delay | | Read | Write | Read | Write | Read | Write | Read | Write | Read |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| | | | | | | | | | | | | | |
| 3300 | 0xc80005 | 0xed0005 | 1 | 2451 | 2478 | 153.2 MB/s | 154.9 MB/s | 114.0 us | 1.5 ms | 45.0 us | 40.0 us | 64.0 KB |
| 64.0 KB | 1.3 ms | 5.0 us | | | | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Total number of ITNs: 1

```

次に、SCSI のターゲット ITL のイニシエータ識別子 0xe90440 と LUN 識別子 0001-0000-0000-0000 のフロー メトリックの表示方法の例を示します :

```

switch# ShowAnalytics --info --target-itl --initiator 0xe90440 --lun 0001-0000-0000-0000
2019-04-09 11:18:40.132828

Interface fc5/21
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| VSAN | Initiator | VMID | Target | LUN | Avg IOPS | Avg Throughput | Avg ECT | Avg DAL |
| | Avg IO Size | | Avg Host Delay | Avg Array Delay | | Read | Write | Read | Write | Read | Write | Read | Write |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| | | | | | | | | | | | | | |
| 2200 | 0xe90440 | - | 0xe80b40 | 0001-0000-0000-0000 | 0 | 5816 | 0 B/s | 2.8 MB/s | 0 ns | 131.0 us | 0 ns | 48.0 |
| us | 0 B | 512.0 B | 0 ns | 0 ns | | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Total number of ITLs: 1

```

例: ShowAnalytics オーバーレイ CLI の使用

フローメトリックの詳細については、[フローメトリック \(133 ページ\)](#) を参照してください。
次の例は、NVMe の 1 秒あたりの I/O 操作数 (IOPS) の上位 ITN を表示する方法を表示します:

```
switch# ShowAnalytics --top --nvme
2019-06-13 10:56:49.099069
```

PORT	VSAN	Initiator	Target	Namespace	Avg IOPS	
					Read	Write
fc3/15	3300	0xc80004	0xed0004	1	2547	2474
fc3/15	3300	0xc80002	0xed0002	1	2521	2486
fc3/15	3300	0xc80008	0xed0008	1	2506	2499
fc3/15	3300	0xc80009	0xed0009	1	2516	2483
fc3/15	3300	0xc80006	0xed0006	1	2516	2482
fc3/15	3300	0xc80007	0xed0007	1	2508	2484
fc3/15	3300	0xc80005	0xed0005	1	2481	2505
fc3/15	3300	0xc80003	0xed0003	1	2469	2517
fc3/15	3300	0xc80000	0xed0001	1	2057	2021
fc3/15	3300	0xc80001	0xed0001	1	1893	1953

次の例は、1 秒あたりの I/O 操作数 (IOPS) の上位 ITL を表示する方法を表示します:

```
switch# ShowAnalytics --top
2019-06-13 10:56:49.099069
```

PORT	VSAN	Initiator	Target	LUN	Avg IOPS	
					Read	Write
fc8/10	5	0xed04b2	0xef0680	0001-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	0003-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	0002-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	0005-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	0006-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	0007-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	0008-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	0009-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	000a-0000-0000-0000	118	0
fc8/10	5	0xed04b2	0xef0680	000b-0000-0000-0000	118	0

次の例は、I/O サイズの上位 ITL を表示する方法を示しています:

```
switch# ShowAnalytics --top --key IOSIZE
Data collected at : Tue, 07 Jun 2022 12:16:09 +0530
```

PORT	VSAN	Initiator	Target	LUN	Avg IO Size	
					Read	Write
fc2/2	2200	0xc80760	0xee0000	0003-0000-0000-0000	0 B	5.8 KB
fc2/19	2200	0xee024b	0xe80441	000c-0000-0000-0000	0 B	4.0 KB
fc2/19	2200	0xee0252	0xe80926	0018-0000-0000-0000	0 B	4.0 KB
fc2/19	2200	0xee024c	0xe80920	002f-0000-0000-0000	0 B	4.0 KB
fc2/20	2200	0xee0253	0xe80927	0051-0000-0000-0000	0 B	4.0 KB
fc2/20	2200	0xee0253	0xe80927	000f-0000-0000-0000	0 B	4.0 KB
fc2/19	2200	0xee024c	0xe80920	0006-0000-0000-0000	0 B	4.0 KB
fc2/20	2200	0xee024c	0xe80920	0049-0000-0000-0000	0 B	4.0 KB
fc2/19	2200	0xee0250	0xe80924	0029-0000-0000-0000	0 B	4.0 KB
fc2/19	2200	0xee0251	0xe80925	0034-0000-0000-0000	0 B	4.0 KB

次に、ITL のイニシエータ フローの表示方法の例を示します:

```
switch# ShowAnalytics --top --initiator-flow
Data collected at : Tue, 07 Jun 2022 12:20:28 +0530
```

PORT	VSAN	Initiator	Avg IOPS	
			Read	Write
fc1/29	2200	0xc803e0	0	29037
fc1/29	2200	0xc803e1	0	19919
fc2/2	2200	0xc80760	0	31


```

| fc12/17 | 2200 | 0xc80600 | | 0 | 0 |
| fc2/20 | 2200 | 0xee01cc | | 0 | 0 |
| fc2/20 | 2200 | 0xee006e | | 0 | 0 |
| fc2/19 | 2200 | 0xee0272 | | 0 | 0 |
| fc2/20 | 2200 | 0xee02b2 | | 0 | 0 |
| fc2/20 | 2200 | 0xee02d1 | | 0 | 0 |
| fc2/19 | 2200 | 0xee02b3 | | 0 | 0 |
+-----+

```

次に、ITL のターゲット フローの表示方法の例を示します :

```

switch# ShowAnalytics --top --target-flow

Data collected at : Tue, 07 Jun 2022 12:20:42 +0530

+-----+-----+-----+-----+
| PORT | VSAN | Target | Avg IOPS |
+-----+-----+-----+-----+
|      |      |      | Read | Write |
| fc1/22 | 2200 | 0xc80329 | 0 | 20269 |
| fc1/23 | 2200 | 0xc80349 | 0 | 20262 |
| fc1/24 | 2200 | 0xc80369 | 0 | 20196 |
| fc1/34 | 2200 | 0xc804a9 | 0 | 20177 |
| fc1/36 | 2200 | 0xc804c9 | 0 | 20165 |
| fc1/35 | 2200 | 0xc80589 | 0 | 20095 |
| fc1/33 | 2200 | 0xc80469 | 0 | 20042 |
| fc1/1 | 2200 | 0xc80029 | 0 | 18684 |
| fc1/2 | 2200 | 0xc80069 | 0 | 18663 |
| fc1/15 | 2200 | 0xc80249 | 0 | 18654 |
+-----+-----+-----+-----+

```

次に、ITL のフローの表示方法の例を示します :

```

switch# ShowAnalytics --top --it-flow

Data collected at : Tue, 07 Jun 2022 12:21:58 +0530

+-----+-----+-----+-----+-----+
| PORT | VSAN | Initiator | Target | Avg IOPS |
+-----+-----+-----+-----+-----+
|      |      |      |      | Read | Write |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0 | 28321 |
| fc1/22 | 2200 | 0xc809e9 | 0xc80329 | 0 | 20274 |
| fc1/24 | 2200 | 0xc80a29 | 0xc80369 | 0 | 20244 |
| fc1/23 | 2200 | 0xc80a09 | 0xc80349 | 0 | 20244 |
| fc1/34 | 2200 | 0xc80b49 | 0xc804a9 | 0 | 20181 |
| fc1/36 | 2200 | 0xc80b89 | 0xc804c9 | 0 | 20173 |
| fc1/35 | 2200 | 0xc80b69 | 0xc80589 | 0 | 20054 |
| fc1/33 | 2200 | 0xc80b29 | 0xc80469 | 0 | 20019 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 0 | 19425 |
| fc1/1 | 2200 | 0xc80ac9 | 0xc80029 | 0 | 18659 |
+-----+-----+-----+-----+-----+

```

次に、ITL のイニシエータ、ターゲットと LUN フローの表示方法の例を示します :

```

switch# ShowAnalytics --top --noclear

Data collected at : Tue, 07 Jun 2022 12:27:38 +0530

```

例 : ShowAnalytics オーバーレイ CLI の使用

```

+-----+-----+-----+-----+-----+-----+
| PORT | VSAN | Initiator | Target | LUN | Avg IOPS |
|      |      |           |        |     |          |
+-----+-----+-----+-----+-----+-----+
|      |      |           |        |     | Read |
Write |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0064-0000-0000-0000 | 0 |
567 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 003b-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 004e-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0043-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 0038-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 0040-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0061-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0014-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 001e-0000-0000-0000 | 0 |
283 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 001d-0000-0000-0000 | 0 |
283 |
+-----+-----+-----+-----+-----+-----+

```

Data collected at : Tue, 07 Jun 2022 12:27:45 +0530

```

+-----+-----+-----+-----+-----+-----+
| PORT | VSAN | Initiator | Target | LUN | Avg IOPS |
|      |      |           |        |     |          |
+-----+-----+-----+-----+-----+-----+
|      |      |           |        |     | Read |
Write |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0064-0000-0000-0000 | 0 |
554 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 003b-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 004e-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0043-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 0038-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 0040-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0061-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 0014-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e1 | 0xc80701 | 001e-0000-0000-0000 | 0 |
277 |
| fc1/29 | 2200 | 0xc803e0 | 0xc80700 | 001d-0000-0000-0000 | 0 |
277 |
+-----+-----+-----+-----+-----+-----+

```

277

次の例は、NVMe のスループットの上位 ITN を段階的に表示する方法を示しています：

```
switch# ShowAnalytics --top --key thput --progress --nvme
2019-06-13 10:58:16.015546
```

PORT	VSAN	Initiator	Target	Namespace	Avg Throughput	
					Read	Write
fc3/15	3300	0xc80003	0xed0003	1	159.1 MB/s	154.6 MB/s
fc3/15	3300	0xc80002	0xed0002	1	157.4 MB/s	155.0 MB/s
fc3/15	3300	0xc80006	0xed0006	1	157.7 MB/s	154.3 MB/s
fc3/15	3300	0xc80004	0xed0004	1	157.1 MB/s	154.8 MB/s
fc3/15	3300	0xc80007	0xed0007	1	155.5 MB/s	155.4 MB/s
fc3/15	3300	0xc80009	0xed0009	1	153.8 MB/s	156.6 MB/s
fc3/15	3300	0xc80008	0xed0008	1	152.2 MB/s	157.1 MB/s
fc3/15	3300	0xc80005	0xed0005	1	150.9 MB/s	158.1 MB/s
fc3/15	3300	0xc80000	0xed0001	1	133.7 MB/s	133.3 MB/s
fc3/15	3300	0xc80001	0xed0001	1	118.4 MB/s	120.2 MB/s

次の例は、スループットの上位 ITL を段階的に表示する方法を示しています：

```
switch# ShowAnalytics --top --key thput --progress
2019-06-13 10:58:16.015546
```

PORT	VSAN	Initiator	Target	LUN	Avg THROUGHPUT	
					Read	Write
fc8/10	5	0xed04b2	0xef0680	000f-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b3	0xef0681	000a-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b3	0xef0681	0014-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b4	0xef0682	000f-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b5	0xef0683	000a-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b5	0xef0683	000f-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b5	0xef0683	0013-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b6	0xef0684	0013-0000-0000-0000	133.8 KB/s	0
fc8/10	5	0xed04b2	0xef0680	0004-0000-0000-0000	133.5 KB/s	0
fc8/10	5	0xed04b3	0xef0681	0009-0000-0000-0000	133.5 KB/s	0

この例は、NVMe の 1 秒あたりの I/O 操作 (IOPS) が最も高い ITN を表示する方法を示しています。オプション **--alias** により、イニシエータおよびターゲット デバイスのエイリアス情報が表示されます。

```
switch# ShowAnalytics --top --alias --nvme
2021-02-09 09:15:25.445815
```

PORT	VSAN	Initiator	Target	Namespace	Avg IOPS	
					Read	Write
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2518	2459
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2499	2470
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2491	2472
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2491	2471
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2457	2487
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2445	2496
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2440	2495
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2434	2499
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	2197	2199
fc3/15	3300	sanblaze-147-port7-p	sanblaze-147-port6-p	1	1987	1982

この例は、SCSI の 1 秒あたりの I/O 操作 (IOPS) が最も高い ITL を表示する方法を示しています。オプション **--alias** により、イニシエータおよびターゲット デバイスのエイリアス情報が表示されます。

```
switch# ShowAnalytics --top --alias
2021-02-09 09:15:25.445815
```

PORT	VSAN	Initiator	VMID	Target	LUN	Avg IOPS	
						Read	Write
fc5/22	2200	0xe90460	-	0xe80b60	0002-0000-0000-0000	0	9124
fc5/22	2200	0xe90460	-	0xe80b60	0003-0000-0000-0000	0	9124
fc5/22	2200	0xe90460	-	0xe80b60	0001-0000-0000-0000	0	9123

例 : ShowAnalytics オーバーレイ CLI の使用

```

| fc5/21 | 2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21 | 0003-0000-0000-0000 | 0 | 5718 |
| fc5/21 | 2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21 | 0001-0000-0000-0000 | 0 | 5718 |
| fc5/21 | 2200 | 0xe906c0 | - | Tgt_9706_206_fc5_21 | 0002-0000-0000-0000 | 0 | 5718 |
| fc5/21 | 2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21 | 0002-0000-0000-0000 | 0 | 5717 |
| fc5/21 | 2200 | 0xe90440 | - | Tgt_9706_206_fc5_21 | 0001-0000-0000-0000 | 0 | 5717 |
| fc5/21 | 2200 | 0xe90440 | - | Tgt_9706_206_fc5_21 | 0002-0000-0000-0000 | 0 | 5717 |
| fc5/21 | 2200 | 0xe906c0 | - | Tgt_9706_206_fc5_21 | 0001-0000-0000-0000 | 0 | 5717 |

```

この例は、1秒あたりのI/O操作（IOPS）が最も高いITLを表示する方法を示しています。オプション `--alias` により、イニシエータおよびターゲットデバイスのエイリアス情報が表示されます。

```

switch# ShowAnalytics --top --alias
2021-02-09 09:15:25.445815

```

PORT	VSAN	Initiator	VMID	Target	LUN	Avg IOPS	
						Read	Write
fc1/2	20	tie-2000012341newdev	89	tie-2000012341newdev	0000-0000-0000-0000	0	1769
fc1/1	20	tie-2000012341newdev	89	tie-2000012341newdev	0000-0000-0000-0000	0	1769

次の例は、すべてのターゲットITNのエラーを表示し、NVMeの出力を10のランダムレコードに制限する方法を示しています：

```

switch# ShowAnalytics --errors --target-itn --limit 10
2019-05-23 11:28:34.926267

```

```

Interface fc3/15

```

VSAN	Initiator	Target	Namespace	Total NVMe Failures		Total FC Aborts	
				Read	Write	Read	Write
3300	0xc80005	0xed0005	1	0	0	0	0
3300	0xc80000	0xed0001	1	0	0	0	0
3300	0xc80004	0xed0004	1	0	0	0	0
3300	0xc80001	0xed0001	1	0	0	0	0
3300	0xc80003	0xed0003	1	0	0	0	0
3300	0xc80000	0xed0000	1	0	0	1260	1210
3300	0xc80007	0xed0007	1	0	0	0	0
3300	0xc80008	0xed0008	1	0	0	0	0
3300	0xc80002	0xed0002	1	0	0	0	0
3300	0xc80006	0xed0006	1	0	0	0	0

次の例は、すべてのターゲットITLのエラーを表示し、出力を10のランダムレコードに制限する方法を示しています：

```

switch# ShowAnalytics --errors --target-itl --limit 10
2019-05-23 11:28:34.926267

```

```

Interface fc8/7

```

VSAN	Initiator	Target	LUN	Total SCSI Failures		Total FC Aborts	
				Read	Write	Read	Write
510xed0332	0xef0592	000f-0000-0000-0000		0	0	0	0
510xed0342	0xef05a2	000a-0000-0000-0000		0	0	0	0
510xed0332	0xef0592	0008-0000-0000-0000		0	0	0	0
510xed0340	0xef05a0	0010-0000-0000-0000		0	0	0	0
510xed0322	0xef0582	0008-0000-0000-0000		0	0	0	0
510xed032c	0xef058c	0014-0000-0000-0000		0	0	0	0
510xed033a	0xef059a	000d-0000-0000-0000		0	0	0	0
510xed034a	0xef05aa	0005-0000-0000-0000		0	0	0	0
510xed033a	0xef059a	0007-0000-0000-0000		0	0	0	0
510xed034a	0xef05aa	0013-0000-0000-0000		0	0	0	0

次の例では、NVMe障害がゼロ以外のすべてのITNを表示し、カウントを元に戻す方法を示します：

```

switch# ShowAnalytics --errorsonly --initiator-itn
2019-04-09 11:27:42.496294

```

```

Interface fc16/12

```

VSAN	Initiator	Target	Namespace	Total NVMe Failures		Total FC Aborts	
				Read	Write	Read	Write
3300	0xc80000	0xed0000	1	0	0	1635	1631

次の例は、SCSI 障害がゼロ以外のすべての ITL を表示し、カウントを戻す方法を示しています：

```
switch# ShowAnalytics --erroronly --initiator-itl
2019-04-09 11:27:42.496294

Interface fc8/27
-----
| VSAN|Initiator|Target|LUN | Total SCSI Failures | Total FC Aborts | | |
|---|---|---|---|---|---|---|---|
| | | | | Read | Write | Read | Write |
|-----|-----|-----|-----|-----|-----|
| 311|0x900000|0xc90000|0000-0000-0000-0000 | 0 | 42 | 0 | 0 |
|-----|-----|-----|-----|-----|-----|
```

この例は、SCSI 障害がゼロ以外の 10 のランダム ITL を表示し、カウントを戻す方法を示しています。デバイスエイリアス（存在する場合）は、イニシエータとターゲットの両方に含まれます。

```
switch# ShowAnalytics --erroronly --initiator-itl --alias --limit 10
2019-04-09 12:06:19.847350

Interface fcl/26
-----
| VSAN|Initiator|Target|LUN | Total SCSI Failures|Total FC Aborts|Initiator Device alias|Target Device alias| | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Read | Write | Read | Write | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 108|0xee0467|0x70039b|0001-0000-0000-0000 | 0 | 1 | 0 | 0 | | SB_112_port_T_18_7 |
| 108|0xee0401|0xbc092b|0002-0000-0000-0000 | 10 | 16 | 0 | 0 | | SB_112_port_T_0_1 |
| 108|0xee0441|0xbc092b|0003-0000-0000-0000 | 3 | 13 | 0 | 0 | SB_112_port_I_7_1 | SB_112_port_T_0_1 |
| 108|0xee0401|0xbc0996|0001-0000-0000-0000 | 3 | 0 | 0 | 0 | | SB_112_port_T_0_1 |
| 108|0xee0441|0xbc0996|0002-0000-0000-0000 | 0 | 3 | 0 | 0 | SB_112_port_I_7_1 |
| 108|0xee0481|0xbc0996|0004-0000-0000-0000 | 0 | 4 | 0 | 0 | | SB_112_port_T_0_3 |
| 108|0xee0403|0xbc092d|0000-0000-0000-0000 | 12 | 8 | 0 | 0 | | SB_112_port_T_0_3 |
| 108|0xee0443|0xbc092d|0001-0000-0000-0000 | 3 | 12 | 0 | 0 | SB_112_port_I_7_3 | SB_112_port_T_0_3 |
| 108|0xee0443|0xbc0998|0000-0000-0000-0000 | 1 | 0 | 0 | 0 | SB_112_port_I_7_3 |
|-----|-----|-----|-----|-----|-----|-----|-----|
```

次の例は、NVMe のターゲット ITN のターゲット識別子 0xed0000 の最小、最大、およびピークフローメトリックを表示する方法を示しています：

```
switch# ShowAnalytics --minmax --target-itn --target 0xed0000
2019-04-09 11:22:08.652598
Interface fc3/15
-----
| VSAN | Initiator | Target | Namespace | Peak IOPS* | Peak Throughput* | Read ECT* | Write ECT* |
| Host Delay* | Array Delay* | Write IO sequence* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Max | Min | Max | Min | Read | Write | Read | Write | Min | Max | Min | Max | Min |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 3300 | 0xc80000 | 0xed0000 | 1 | 383 | 379 | 24.0 MB/s | 23.7 MB/s | 2.6 ms | 26.7 ms | 3.5 ms | 28.7 ms | 12.0 |
| us | 3.1 ms | NA | 1.4 ms | 0 | 0 | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
Total number of ITNs: 1
*These values are calculated since the metrics were last cleared.
```

次の例は、SCSI のターゲット ITL のターゲット識別子 0xe80b40 の最小、最大、およびピークフローメトリックを表示する方法を示しています：

```
switch# ShowAnalytics --minmax --target-itl --target 0xe80b40
2019-04-09 11:22:08.652598

Interface fc5/21
-----
| VSAN | Initiator | VMID | Target | LUN | Peak IOPS* | Peak Throughput* | Read ECT* | Write ECT* |
| Host Delay* | Array Delay* | Write IO sequence* | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Min | Max | Min | Max | Min | Max | Read | Write | Min | Max | Min | Max |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 2200 | 0xe90440 | - | 0xe80b40 | 0001-0000-0000-0000 | 0 | 8361 | 0 B/s | 4.1 MB/s | 0 ns | 0 ns | 68.0 us | 2.6 |
| ms | 0 ns | 0 ns | NA | 0 ns | 0 | 0 | | | | | | |
| 2200 | 0xe90440 | - | 0xe80b40 | 0002-0000-0000-0000 | 0 | 7814 | 0 B/s | 3.8 MB/s | 0 ns | 0 ns | 69.0 us | 2.6 |
| ms | 0 ns | 0 ns | NA | 0 ns | 0 | 0 | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
Total number of ITLs: 2
*These values are calculated since the metrics were last cleared.
```

例 : ShowAnalytics オーバーレイ CLI の使用

次の例は、ターゲット ITN のインターフェイス fc3/15 のデバイス エイリアス情報、最小、最大、およびピーク フロー メトリックを表示し、NVMe の出力を 10 のランダム レコードに制限する方法を示しています：

```
switch# ShowAnalytics --minmax --target-itn --alias --interface fc3/15 --limit 10
2019-04-09 12:01:40.609197

Interface fc3/15
-----
|VSAN | Initiator | Target | Namespace | Peak IOPS* | Peak Throughput* | Read ECT* | Write ECT* | Host Delay*
| | Array Delay* | Write IO sequence* | | | | | | |
| | Min | Max | Min | Max | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2674 | 2595 | 167.1 MB/s | 162.2 MB/s | 38.0 us | 2.3 ms | 69.0 us | 3.9 ms | 12.0 us | 3.7
ms | NA | 36.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 10199 | 10163 | 637.4 MB/s | 635.2 MB/s | 9.0 us | 2.4 ms | 65.0 us | 3.9 ms | 12.0 us | 3.7
ms | NA | 32.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2618 | 2587 | 163.6 MB/s | 161.7 MB/s | 39.0 us | 2.4 ms | 69.0 us | 3.8 ms | 12.0 us | 3.6
ms | NA | 34.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2288 | 2287 | 143.0 MB/s | 143.0 MB/s | 37.0 us | 2.4 ms | 69.0 us | 4.0 ms | 12.0 us | 3.7
ms | NA | 35.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2624 | 2583 | 164.0 MB/s | 161.4 MB/s | 38.0 us | 2.5 ms | 108.0 us | 3.6 ms | 12.0 us | 3.4
ms | NA | 33.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 383 | 379 | 24.0 MB/s | 23.7 MB/s | 2.6 ms | 27.0 ms | 3.5 ms | 28.7 ms | 12.0 us | 3.1
ms | NA | 1.4 ms | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2624 | 2587 | 164.0 MB/s | 161.7 MB/s | 38.0 us | 2.4 ms | 69.0 us | 3.7 ms | 12.0 us | 3.5
ms | NA | 39.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2621 | 2597 | 163.8 MB/s | 162.3 MB/s | 38.0 us | 2.4 ms | 77.0 us | 3.9 ms | 12.0 us | 3.5
ms | NA | 31.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2646 | 2590 | 165.4 MB/s | 161.9 MB/s | 38.0 us | 2.6 ms | 69.0 us | 3.8 ms | 12.0 us | 3.6
ms | NA | 33.0 us | 0 | 0 |
|3300 | sanblaze-147-port7-p | sanblaze-147-port6-p | 1 | 2651 | 2594 | 165.7 MB/s | 162.2 MB/s | 39.0 us | 2.6 ms | 69.0 us | 3.6 ms | 12.0 us | 3.5
ms | NA | 32.0 us | 0 | 0 |
-----
Total number of ITNs: 10
*These values are calculated since the metrics were last cleared.
```

次に、ターゲット ITL のインターフェイス fc5/21 のデバイス エイリアス情報、最小、最大、およびピーク フロー メトリックを表示し、SCSI の出力を 10 のランダム レコードに制限する例を示します：

```
switch# ShowAnalytics --minmax --target-itl --alias --interface fc5/21 --limit 10
2019-04-09 12:01:40.609197

Interface fc5/21
-----
|VSAN | Initiator | VMID | Target | LUN | Peak IOPS* | Peak Throughput* | Read ECT* | Write ECT* | Host
Delay* | Array Delay* | Write IO sequence* | | | | | | | |
| | Max | Min | Max | Min | Max | | Read | Write | Read | Write | Min | Max | Min | Max | Min |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21_ | 0002-0000-0000-0000 | 0 | 9242 | 0 B/s | 4.5 MB/s | 0 ns | 0 ns | 66.0 us | 2.6 ms | 0 ns |
0 ns | NA | 0 ns | 0 | 0 |
|2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21_ | 0003-0000-0000-0000 | 0 | 9243 | 0 B/s | 4.5 MB/s | 0 ns | 0 ns | 66.0 us | 2.6 ms | 0 ns |
0 ns | NA | 0 ns | 0 | 0 |
|2200 | 0xe902e0 | - | Tgt_9706_206_fc5_21_ | 0001-0000-0000-0000 | 0 | 9242 | 0 B/s | 4.5 MB/s | 0 ns | 0 ns | 66.0 us | 2.6 ms | 0 ns |
0 ns | NA | 0 ns | 0 | 0 |
|2200 | 0xe90440 | - | Tgt_9706_206_fc5_21_ | 0001-0000-0000-0000 | 0 | 8361 | 0 B/s | 4.1 MB/s | 0 ns | 0 ns | 68.0 us | 2.6 ms | 0 ns |
0 ns | NA | 0 ns | 0 | 0 |
|2200 | 0xe90440 | - | Tgt_9706_206_fc5_21_ | 0002-0000-0000-0000 | 0 | 7814 | 0 B/s | 3.8 MB/s | 0 ns | 0 ns | 69.0 us | 2.6 ms | 0 ns |
0 ns | NA | 0 ns | 0 | 0 |
|2200 | 0xe906c0 | - | Tgt_9706_206_fc5_21_ | 0001-0000-0000-0000 | 0 | 7779 | 0 B/s | 3.8 MB/s | 0 ns | 0 ns | 69.0 us | 2.7 ms | 0 ns |
0 ns | NA | 0 ns | 0 | 0 |
|2200 | 0xe906c0 | - | Tgt_9706_206_fc5_21_ | 0002-0000-0000-0000 | 0 | 7779 | 0 B/s | 3.8 MB/s | 0 ns | 0 ns | 69.0 us | 2.6 ms | 0 ns |
0 ns | NA | 0 ns | 0 | 0 |
-----
Total number of ITLs: 7
*These values are calculated since the metrics were last cleared.
```

次の例は、インターフェイスの範囲の NPU 負荷を表示する方法を示しています：

```
switch# ShowAnalytics --evaluate-npload --interface fc8/7-8
2019-05-09 10:56:54.021234
There are 2 interfaces to be evaluated. Expected time is 2 minutes 0 seconds
Do you want to continue [Yes|No]? [n]y
-----
| Interface | ITL/N Count | NPU Load % | Analysis | Analysis |
| | SCSI | NVMe | Total | SCSI | NVMe | Total | Start Time | End Time |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| fc8/7 | 1000 | 0 | 1000 | 8.1 | 0.0 | 8.1 | 10:57:20 | 10:57:52 |
| fc8/8 | 1000 | 0 | 1000 | 8.1 | 0.0 | 8.1 | 10:58:20 | 10:58:51 |
| *Total | 2000 | 0 | 2000 | 16.2 | 0.0 | 16.2 | | |
-----
* This total is an indicative reference based on evaluated ports
```



- (注) NPU 負荷の評価には時間がかかります。評価プロセス中にスイッチへの接続が失われた場合、プロセスは完了するまでバックグラウンドで実行され続け、出力はファイルに保存されます。プロセスが完了すると、syslog メッセージが生成され、ファイル名と出力が保存されるファイルの場所が示されます。

次に、ブートフラッシュの *output.txt* という名前のファイルに出力を複製する例を示します：



- (注) **--outfile** オプションをすべての **ShowAnalytics** コマンド オプションとともに使用して、コマンド出力をファイルに複製できます。

```
switch# ShowAnalytics --evaluate-npload --outfile output.txt
2020-11-24 13:42:19.510351
There are 4 interfaces to be evaluated. Expected time is 4 minutes 0 seconds
Do you want to continue [Yes|No]? [n]y
Module 1
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Interface | Type | ITL/N Count | NPU Load % | Analysis | Analysis |
|           |     | SCASI | NVMe | Total | SCASI | NVMe | Total | Start Time | End Time |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| fcl/1 | Target | 1 | 0 | 1 | 0.6 | 0.0 | 0.6 | 13:42:40 | 13:43:11 |
| fcl/2 | Initiator | 1 | 0 | 1 | 0.6 | 0.0 | 0.6 | 13:43:40 | 13:44:11 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| *Total | | 2 | 0 | 2 | 1.2 | 0.0 | 1.2 | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Recommended port sampling size: 48

* This total is an indicative reference based on evaluated ports

Errors:
-----

Traffic is not running on port fcl/47
Traffic is not running on port fcl/48
```

次の例は、出力を *bootflash :* の *output.txt* という名前のファイルに追加する方法を示しています。このファイルには、すでいくつかの出力が含まれています：

```
switch# ShowAnalytics --evaluate-npload --appendfile output.txt
2020-11-24 13:45:07.535440
There are 4 interfaces to be evaluated. Expected time is 4 minutes 0 seconds
Do you want to continue [Yes|No]? [n]y
Module 1
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Interface | Type | ITL/N Count | NPU Load % | Analysis | Analysis |
|           |     | SCASI | NVMe | Total | SCASI | NVMe | Total | Start Time | End Time |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| fcl/1 | Target | 1 | 0 | 1 | 0.6 | 0.0 | 0.6 | 13:45:40 | 13:46:11 |
| fcl/2 | Initiator | 1 | 0 | 1 | 0.6 | 0.0 | 0.6 | 13:46:40 | 13:47:11 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| *Total | | 2 | 0 | 2 | 1.2 | 0.0 | 1.2 | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Recommended port sampling size: 48

* This total is an indicative reference based on evaluated ports

Errors:
-----

Traffic is not running on port fcl/47
Traffic is not running on port fcl/48
```

次に、NVMe の VSAN スループット情報を表示する例を示します：

```
switch# ShowAnalytics --vsan-thput --nvme
2019-05-09 14:02:07.940600

Interface fcl6/12
-----+-----+-----+-----+-----+
| VSAN | Throughput (4s avg) | | |
|      | Read | Write | Total |
|      | (Mbps) | (Mbps) | (Mbps) |
+-----+-----+-----+-----+-----+
| 3300 | 1605.8 | 1626.8 | 3232.6 |
```

例 : ShowAnalytics オーバーレイ CLI の使用

```
-----+
Note: This data is only for NVMe
```

次に、SCSI の VSAN スループット情報を表示する例を示します :

```
switch# ShowAnalytics --vsan-thput
2019-05-09 14:02:07.940600

Interface fc8/17
-----+
| VSAN | Throughput (4s avg) | |
| Read | Write | Total |
| (Mbps) | (Mbps) | (Mbps) |
-----+
| 5 | 0.0 | 0.0 | 0.0 |
-----+

Interface fc8/18
-----+
| VSAN | Throughput (4s avg) | |
| Read | Write | Total |
| (Mbps) | (Mbps) | (Mbps) |
-----+
| 5 | 0.0 | 0.0 | 0.0 |
-----+

Interface fc8/19
-----+
| VSAN | Throughput (4s avg) | |
| Read | Write | Total |
| (Mbps) | (Mbps) | (Mbps) |
-----+
| 5 | 0.0 | 0.0 | 0.0 |
-----+

Interface fc8/20
-----+
| VSAN | Throughput (4s avg) | |
| Read | Write | Total |
| (Mbps) | (Mbps) | (Mbps) |
-----+
| 5 | 0.0 | 0.0 | 0.0 |
-----+

Interface fc8/21
-----+
| VSAN | Throughput (4s avg) | |
| Read | Write | Total |
| (Mbps) | (Mbps) | (Mbps) |
-----+
| 3500 | 301.9 | 302.8 | 604.7 |
-----+

Interface fc8/22
-----+
| VSAN | Throughput (4s avg) | |
| Read | Write | Total |
| (Mbps) | (Mbps) | (Mbps) |
-----+
| 3500 | 302.7 | 304.8 | 607.5 |
-----+

Note: This data is only for SCSI
```

次に、ポートチャンネルの VSAN スループット情報を表示する例を示します :

```
switch# ShowAnalytics --vsan-thput --interface port-channel108
2019-05-09 15:01:32.538121

Interface port-channel108
-----+
| VSAN | Throughput (4s avg) | |
| Read | Write | Total |
| (Mbps) | (Mbps) | (Mbps) |
-----+
| 1 | 0.0 | 0.0 | 0.0 |
| 5 | 145.9 | 0.0 | 145.9 |
| 3500 | 561.9 | 558.6 | 1120.5 |
-----+

Note: This data is only for SCSI
```

次の例は、NVMe のインターフェイスの ITN ごとの未処理の IO を表示する方法を示しています :

```
switch# ShowAnalytics --outstanding-io --interface fc16/12 --nvme
2019-05-20 11:59:48.306396
Interface : fc16/12 VSAN : 3300 FCNS_type : Initiator

-----+
| Initiator | Target | Namespace | Outstanding IO |
| | | | Read | Write |
-----+
| 0xc80002 | 0xed0002 | 1 | 3 | 6 |
-----+
```



```

| 0xc80007 | 0xed0007 | 1 | 5 | 5 |
| 0xc80005 | 0xed0005 | 1 | 1 | 10 |
| 0xc80001 | 0xed0001 | 1 | 2 | 7 |
| 0xc80000 | 0xed0000 | 1 | 6 | 5 |
| 0xc80008 | 0xed0008 | 1 | 1 | 7 |
| 0xc80009 | 0xed0009 | 1 | 3 | 4 |
| 0xc80004 | 0xed0004 | 1 | 3 | 6 |
| 0xc80006 | 0xed0006 | 1 | 2 | 5 |
| 0xc80000 | 0xed0001 | 1 | 3 | 4 |
| 0xc80003 | 0xed0003 | 1 | 4 | 4 |
+-----+
Instantaneous Qdepth : 96

```

次の例は、SCSI のインターフェイスの ITL ごとの未処理の IO を表示する方法を示しています：

```

switch# ShowAnalytics --outstanding-io --interface fc8/7
2019-05-20 11:59:48.306396

Interface : fc8/7 VSAN : 5 FCNS_type : Target

+-----+
| Initiator|Target|LUN | Outstanding IO |
+-----+
| | | | Read | Write |
+-----+
| 0xed0320|0xef0580|0001-0000-0000-0000 | 2 | 0 |
| 0xed0320|0xef0580|0002-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0003-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0004-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0005-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0006-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0007-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0008-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0009-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|000a-0000-0000-0000 | 1 | 0 |
+-----+
Instantaneous Qdepth : 11

```



(注) 出力の *Instantaneous Qdepth* 値は、指定されたインターフェイスで現在現用系な IO の数を表します。

次の例は、インターフェイスの ITN ごとの未処理の IO を表示し、出力を 10 レコードに制限し、NVMe のデータを定期的に更新する方法を示しています：

```

switch# ShowAnalytics --outstanding-io --interface fc8/7 --limit 10 --refresh --nvme
2019-05-20 12:00:21.028228
Interface : fc16/12 VSAN : 3300 FCNS_type : Initiator

+-----+
| Initiator | Target | Namespace | Outstanding IO |
+-----+
| | | | Read | Write |
+-----+
| 0xc80002 | 0xed0002 | 1 | 2 | 7 |
| 0xc80007 | 0xed0007 | 1 | 3 | 5 |
| 0xc80005 | 0xed0005 | 1 | 1 | 8 |
| 0xc80001 | 0xed0001 | 1 | 1 | 0 |
| 0xc80000 | 0xed0000 | 1 | 5 | 6 |
+-----+

```

次の例は、インターフェイスの ITL ごとの未処理の IO を表示し、出力を 10 レコードに制限し、SCSI のデータを定期的に更新する方法を示しています：

```

switch# ShowAnalytics --outstanding-io --interface fc8/7 --limit 10 --refresh
2019-05-20 12:00:21.028228

Interface : fc8/7 VSAN : 5 FCNS_type : Target

+-----+
| Initiator|Target|LUN | Outstanding IO |
+-----+
| | | | Read | Write |
+-----+
| 0xed0320|0xef0580|0001-0000-0000-0000 | 0 | 0 |
| 0xed0320|0xef0580|0002-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0003-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0004-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0005-0000-0000-0000 | 0 | 0 |
| 0xed0320|0xef0580|0006-0000-0000-0000 | 0 | 0 |
| 0xed0320|0xef0580|0007-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|0008-0000-0000-0000 | 0 | 0 |
| 0xed0320|0xef0580|0009-0000-0000-0000 | 1 | 0 |
| 0xed0320|0xef0580|000a-0000-0000-0000 | 1 | 0 |
+-----+

```

例: ShowAnalytics オーバーレイ CLI の使用

```
-----+
Estimated Qdepth : 6
```

この例では、ターゲット ITL のイニシエータ識別子 0xee008e、ターゲット識別子 0xe80b22、および LUN 識別子 0060-0000-0000-0000 のヒストグラムを表示する方法を示します

```
switch# ShowAnalytics --histogram --initiator-itl --initiator 0xee008e --target 0xe80b22 --lun 0060-0000-0000-0000
```

```
Starting histogram monitor session
Session ID: 15789
```

```
-----+
|                | 25-05-2022 |
| Metric          | 15:29:30   |
|-----+-----|
| IOPS Read       | 0          |
| IOPS Write      | 11         |
| ECT Read        | 0 ns       |
| ECT Write       | 28.1 ms    |
| DAL Read        | 0 ns       |
| DAL Write       | 13.7 ms    |
| FAILURES Read   | 0          |
| FAILURES Write  | 0          |
| ABORTS Read     | 0          |
| ABORTS Write    | 0          |
|-----+-----|
```

```
Histogram data will get updated every 5 mins
```

この例では、ターゲット ITL のイニシエータ識別子 0xee008e およびターゲット識別子 0xe80b22 のヒストグラムを表示する方法を示します。

```
switch# ShowAnalytics --histogram --initiator-it --initiator 0xee008e --target 0xe80b22
```

```
Starting histogram monitor session
Session ID: 16205
```

```
-----+
|                | 25-05-2022 |
| Metric          | 15:30:13   |
|-----+-----|
| IOPS Read       | 0          |
| IOPS Write      | 106        |
| ECT Read        | 0 ns       |
| ECT Write       | 28.1 ms    |
| DAL Read        | 0 ns       |
| DAL Write       | 13.7 ms    |
| FAILURES Read   | 0          |
| FAILURES Write  | 0          |
| ABORTS Read     | 0          |
| ABORTS Write    | 0          |
|-----+-----|
```

```
Histogram data will get updated every 5 mins
```

この例は、すべてのセッションのヒストグラムを表示する方法を示しています。

```
switch# ShowAnalytics --histogram --show-sessions
```

```
-----+
| Session ID | Arguments |
|-----+-----|
| 15789      | --initiator-itl --initiator 0xee008e --target 0xe80b22 --lun 0060-0000-0000-0000 --interval 5 --metric IOPS,ECT,DAL,ERRORS |
| 16205      | --initiator-it --initiator 0xee008e --target 0xe80b22 --interval 5 --metric IOPS,ECT,DAL,ERRORS |
| 20924      | --target-itl --initiator 0xc80ba3 --target 0xc804e3 --lun 0002-0000-0000-0000 --interval 5 --metric IOPS,ECT,DAL,ERRORS |
|-----+-----|
```

```
Analytic-scale184# ShowAnalytics --histogram --sessionId 16205
```

```
-----+
|                | 25-05-2022 | 25-05-2022 | 25-05-2022 |
| Metric          | 15:40:15   | 15:35:14   | 15:30:13   |
|-----+-----+-----+-----|
| IOPS Read       | 0          | 0          | 0          |
| IOPS Write      | 95         | 142        | 106        |
| ECT Read        | 0 ns       | 0 ns       | 0 ns       |
| ECT Write       | 28.2 ms    | 27.7 ms    | 28.1 ms    |
| DAL Read        | 0 ns       | 0 ns       | 0 ns       |
| DAL Write       | 13.7 ms    | 13.6 ms    | 13.7 ms    |
| FAILURES Read   | 0          | 0          | 0          |
| FAILURES Write  | 0          | 0          | 0          |
| ABORTS Read     | 0          | 0          | 0          |
| ABORTS Write    | 0          | 0          | 0          |
|-----+-----+-----+-----|
```

この例は、停止した特定のセッション 15789 のヒストグラムを表示する方法を示しています。

```
switch# ShowAnalytics --histogram --stop-session --sessionId 15789
```

```
Stopping session id: 15789
```

```
Analytic-scale184# ShowAnalytics --histogram --initiator-itn --initiator 0xc80960 --target 0xe80641 --namespace 3
Starting histogram monitor session
Session ID: 27792
```

```

+-----+
|          | 25-05-2022 |
| Metric   | 15:47:11   |
+-----+
| IOPS Read | 0          |
| IOPS Write| 0          |
| ECT Read  | 433.0 us   |
| ECT Write | 1.0 ms     |
| DAL Read  | 421.0 us   |
| DAL Write | 339.0 us   |
| FAILURES Read | 0        |
| FAILURES Write | 0        |
| ABORTS Read | 0          |
| ABORTS Write | 0        |
+-----+
Histogram data will get updated every 5 mins

```

この例は、5分ごとにイニシエータ識別子 0xee008e とターゲット識別子 0xe80b22 の IOPS、ECT、DAL、エラーなどのメトリックの詳細を含むヒストグラムを表示する方法を示しています。

```

switch# ShowAnalytics --histogram --initiator-it --initiator 0xee008e --target 0xe80b22 --interval 5 --metric IOPS,ECT,DAL,ERRORS
Data collected at : Wed, 25 May 2022 16:20:12 +0530
+-----+
|          | 25-05-2022 | 25-05-2022 | 25-05-2022 | 25-05-2022 | 25-05-2022 | 25-05-2022 | 25-05-2022 | 25-05-2022 | 25-05-2022 |
| 25-05-2022 | 16:15:22 | 16:10:21 | 16:05:19 | 16:00:18 | 15:55:18 | 15:50:17 | 15:45:16 | 15:40:15 | 15:35:14 |
| 15:30:13 |
+-----+
| IOPS Read  | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| IOPS Write | 138        | 104        | 50         | 135        | 68         | 74         | 89         | 95         | 142        |
| ECT Read   | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       |
| ECT Write  | 28.2 ms   | 27.8 ms   | 28.3 ms   | 28.0 ms   | 28.0 ms   | 28.0 ms   | 27.9 ms   | 28.2 ms   | 27.7 ms   |
| DAL Read   | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       | 0 ns       |
| DAL Write  | 13.7 ms   | 13.6 ms   | 13.8 ms   | 13.7 ms   | 13.7 ms   | 13.7 ms   | 13.7 ms   | 13.7 ms   | 13.6 ms   |
| FAILURES Read | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| FAILURES Write | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| ABORTS Read | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| ABORTS Write | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
+-----+

```

この例は、イニシエータ識別子 0xee008e およびターゲット識別子 0xe80b22 のヒストグラムを、リフレッシュ時間 120 分で表示する方法を示しています。

```

switch# ShowAnalytics --histogram --initiator-it --initiator 0xee008e --target 0xe80b22 --interval 120
Starting histogram monitor session
Session ID: 21352
+-----+
|          | 25-05-2022 |
| Metric   | 16:21:29   |
+-----+
| IOPS Read | 0          |
| IOPS Write| 84         |
| ECT Read  | 0 ns       |
| ECT Write | 28.1 ms    |
| DAL Read  | 0 ns       |
| DAL Write | 13.7 ms    |
| FAILURES Read | 0        |
| FAILURES Write | 0        |
| ABORTS Read | 0          |
| ABORTS Write | 0        |
+-----+
Histogram data will get updated every 120 mins

```

この例は、イニシエータ識別子 0xee008e およびターゲット識別子 0xe80b22 の ECT や DAL などのメトリックの詳細を含むヒストグラムを 5 分ごとに表示する方法を示しています。

```

switch# ShowAnalytics --histogram --initiator-it --initiator 0xee008e --target 0xe80b22 --metric ECT,DAL
Starting histogram monitor session
Session ID: 22073
+-----+
|          | 25-05-2022 |
| Metric   | 16:22:35   |
+-----+
| ECT Read  | 0 ns       |
| ECT Write | 28.1 ms    |
| DAL Read  | 0 ns       |

```

フローごとの輻輳ドロップの表示

```
| DAL Write | 13.7 ms |
+-----+
Histogram data will get updated every 5 mins
```

この例は、イニシエータ識別子 0xee008e のヒストグラムを 5 分ごとに表示する方法を示しています。

```
switch# ShowAnalytics --histogram --initiator 0xee008e

Starting histogram monitor session
Session ID: 23254
+-----+
| Metric | 25-05-2022 |
|         | 16:24:10 |
+-----+
| IOPS Read | 0 |
| IOPS Write | 124 |
| ECT Read | 0 ns |
| ECT Write | 28.2 ms |
| DAL Read | 0 ns |
| DAL Write | 13.8 ms |
| FAILURES Read | 0 |
| FAILURES Write | 0 |
| ABORTS Read | 0 |
| ABORTS Write | 0 |
+-----+
Histogram data will get updated every 5 mins
```

この例は、ターゲット 0xc804e3 のヒストグラムを 5 分ごとに表示する方法を示しています。

```
switch# ShowAnalytics --histogram --target 0xc804e3

Starting histogram monitor session
Session ID: 24003
+-----+
| Metric | 25-05-2022 |
|         | 16:25:07 |
+-----+
| IOPS Read | 0 |
| IOPS Write | 3939 |
| ECT Read | 0 ns |
| ECT Write | 23.3 ms |
| DAL Read | 0 ns |
| DAL Write | 10.7 ms |
| FAILURES Read | 0 |
| FAILURES Write | 30429 |
| ABORTS Read | 0 |
| ABORTS Write | 0 |
+-----+
Histogram data will get updated every 5 mins
```

フローごとの輻輳ドロップの表示

SAN アナリティクス機能では、フローごとのパケット タイムアウト ドロップ数が表示されます。ポートのタイムスタンプとともにドロップされたパケットの数が表示されます。

フローごとのパケット ドロップ数を表示するには、次のコマンドを実行します。

```
switch# show analytics flow congestion-drops
```

例：フローごとの輻輳ドロップの表示

この例は、輻輳が原因でフレームがドロップされるフローを示しています。送信元と宛先の FCID、IT ペアの差分フレーム ドロップ数、ドロップのタイムスタンプが表示されます。

```
switch# show analytics flow congestion-drops
```

```
=====
| Source      | Destination      | Congestion      | Timestamp      |
| INTF | VSAN | FCID | FCID | Drops(delta) |
=====
| fc2/13| 0002 | 0x9900E1 | 0x640000 | 00000105 | 1. 09/13/17 11:09:48.762 |
=====
```

```

| fc2/13| 0002 | 0x9900E1 | 0x640000 | 00000002 | 2. 09/13/17 09:05:39.527 |
| fc2/13| 0002 | 0x990000 | 0x640020 | 00000002 | 3. 09/13/17 09:05:39.527 |
=====
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000084 | 1. 09/12/17 08:17:11.905 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000076 | 2. 09/12/17 05:50:37.721 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000067 | 3. 09/12/17 03:24:03.319 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000088 | 4. 09/12/17 00:57:28.019 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000088 | 5. 09/11/17 22:30:53.723 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000086 | 6. 09/11/17 20:04:18.001 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000026 | 7. 09/11/17 17:37:24.273 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000076 | 8. 09/11/17 15:10:50.240 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000074 | 9. 09/11/17 12:44:15.866 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000087 |10. 09/11/17 10:17:41.402 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000086 |11. 09/11/17 07:51:10.412 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000084 |12. 09/11/17 05:24:35.981 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000083 |13. 09/11/17 02:58:01.067 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000086 |14. 09/11/17 00:31:26.709 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000079 |15. 09/10/17 22:04:51.399 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000084 |16. 09/10/17 19:38:17.217 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000082 |17. 09/10/17 17:11:42.594 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000086 |18. 09/10/17 14:44:52.786 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000089 |19. 09/10/17 12:18:18.394 |
| fc2/31| 0002 | 0x640000 | 0x9900E1 | 00000087 |20. 09/10/17 09:51:44.067 |
=====

```

SAN アナリティクスの確認

次に、SAN アナリティクス機能が有効になっているインターフェイスのリストの例を示します。

```

switch# show running-config analytics

!Command: show running-config analytics
!Running configuration last done at: Mon Apr 1 05:27:54 2019
!Time: Mon Apr 1 05:28:42 2019

version 8.4(0)SK(1)
feature analytics
analytics port-sampling module 4 size 12 interval 30

analytics query "select all from fc-scsi.scsi_target_itl_flow" name VI_scsi type periodic
interval 30 differential clear
analytics query "select all from fc-nvme.nvme_target_itn_flow" name nvme-184 type periodic
interval 30 differential clear

interface fc4/25
  analytics type fc-scsi

interface fc4/26
  analytics type fc-nvme

interface fc12/44
  analytics type fc-scsi
  analytics type fc-nvme

```

次に、スイッチにインストールされている設定済みのプッシュクエリのリストの例を示します。

```
switch# show analytics query all
Total queries:2
=====
Query Name      :VI_scsi
Query String    :select all from fc-scsi.scsi_target_itl_flow
Query Type     :periodic, interval 30
Query Options   :differential clear

Query Name      :nvme-184
Query String    :select all from fc-nvme.nvme_target_itn_flow
Query Type     :periodic, interval 30
Query Options   :differential clear
```

次に、モジュールごとのNPUの負荷、ITL、およびITNカウントを表示する例を示します。

```
switch# show analytics system-load
n/a - not applicable
-----
| Module | NPU Load (in %) | ITLs | Analytics System Load Info | Targets |
|         | SCSi NVMe Total | SCSi | ITNs Both | Hosts | Total | SCSi NVMe Total | SCSi NVMe Total |
-----
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 64 | 0 | 64 | 20743 | 0 | 20743 | 0 | 0 | 0 | 346 | 0 | 346 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 12 | 12 | 0 | 300 | 300 | 0 | 0 | 0 | 0 | 40 | 40 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 13 | 13 | 1 | 1 | 2 | 1 | 1 | 2 | 0 | 0 | 0 |
| Total | n/a | n/a | n/a | 20744 | 301 | 21045 | 1 | 1 | 2 | 346 | 40 | 386 |
-----
As of Mon Apr 1 05:31:10 2019
```



(注) **show analytics system-load** コマンドは、アクティブおよび非アクティブのITLカウントを含む、すべてのITLカウントに基づいたシステム負荷情報を提供します。したがって、**purge analytics query** 「*query_string*」コマンドを使用して非アクティブなITLカウントを削除してから、このコマンドを実行してアクティブなITLカウントを取得することをお勧めします。

この例では、すべてのアクティブモジュールのNPU負荷、ITL、およびITNを表示します。

```
switch# ShowAnalytics --systemload-active

This will run differential query on scsi_initiator_itl_flow, scsi_target_itl_flow,
nvme_initiator_itn_flow, nvme_target_itn_flow, scsi_initiator, scsi_target,
nvme_initiator and nvme_target or use the result of installed query if present
Do you want to continue [Yes|No]? [n]y

Data collected at : Wed, 25 May 2022 16:29:24 +0530

Using result of installed queries: dcnmtgtITN,dcnmtgtITL
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Module | ITL/N Count | Initiators | Targets |
|         | SCSi NVMe Total | SCSi NVMe Total | SCSi NVMe Total |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 5571 | 0 | 5571 | 2 | 0 | 2 | 55 | 0 | 55 |
| 2 | 14904 | 1 | 14905 | 191 | 1 | 192 | 191 | 0 | 191 |
| 3 | 7588 | 0 | 7588 | 128 | 0 | 128 | 128 | 0 | 128 |
| 5 | 0 | 0 | 0 | 56 | 0 | 56 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 28063 | 1 | 28064 | 377 | 1 | 378 | 374 | 1 | 375 |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

この例では、特定のアクティブ モジュールの NPU 負荷、ITL、および ITN の詳細を表示します。

```
switch# ShowAnalytics --systemload-active --module 1 --detail
```

```
This will run differential query on scsi_initiator_itl_flow, scsi_target_itl_flow,
nvme_initiator_itn_flow, nvme_target_itn_flow, scsi_initiator, scsi_target,
nvme_initiator and nvme_target or use the result of installed query if present
Do you want to continue [Yes|No]? [n]y
```

```
Data collected at : Wed, 25 May 2022 16:35:35 +0530
```

```
Using result of installed queries: dcnmtgtITN,dcnmtgtITL
```

Module	ITL/N Count			Initiators			Targets		
	SCSI	NVMe	Total	SCSI	NVMe	Total	SCSI	NVMe	Total
1	5571	0	5571	2	0	2	55	0	55
Total	5571	0	5571	2	0	2	55	0	55

```
Detailed output for DS-X9748-3072K9 modules
```

```
Module : 1
```

Ports	ITL/N Count			Initiators			Targets		
	SCSI	NVMe	Total	SCSI	NVMe	Total	SCSI	NVMe	Total
fc1/1,fc1/3,fc1/5,fc1/7	186	0	186	0	0	0	2	0	2
fc1/2,fc1/4,fc1/6,fc1/8	186	0	186	0	0	0	2	0	2
fc1/9,fc1/11,fc1/13,fc1/15	185	0	185	0	0	0	2	0	2
fc1/10,fc1/12,fc1/14,fc1/16	93	0	93	0	0	0	1	0	1
fc1/17,fc1/19,fc1/21,fc1/23	186	0	186	0	0	0	2	0	2
fc1/18,fc1/20,fc1/22,fc1/24	186	0	186	0	0	0	2	0	2
fc1/25,fc1/27,fc1/29,fc1/31	171	0	171	2	0	2	0	0	0
fc1/33,fc1/35,fc1/37,fc1/39	2188	0	2188	0	0	0	22	0	22
fc1/34,fc1/36,fc1/38,fc1/40	2190	0	2190	0	0	0	22	0	22
Total	5571	0	5571	2	0	2	55	0	55

次に、ポート サンプリング ステータスと瞬間的な NPU の負荷を確認する例を示します。

```
switch# show analytics port-sampling module 1
```

```
Sampling Window Size: 12
```

```
Rotation Interval: 30
```

```
NPU LOAD : 64% [SCSI 64%, NVMe 0%]
```

Port	Monitored Start Time	Monitored End Time
fc4/25	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/26	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/27	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/28	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/29	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/30	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/31	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/32	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/33	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/34	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/35	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/36	04/01/19 - 05:25:29	04/01/19 - 05:25:59
fc4/37*	04/01/19 - 05:25:59	-
fc4/38*	04/01/19 - 05:25:59	-
fc4/39*	04/01/19 - 05:25:59	-
fc4/40*	04/01/19 - 05:25:59	-
fc4/41*	04/01/19 - 05:25:59	-
fc4/42*	04/01/19 - 05:25:59	-
fc4/43*	04/01/19 - 05:25:59	-
fc4/44*	04/01/19 - 05:25:59	-
fc4/45*	04/01/19 - 05:25:59	-
fc4/46*	04/01/19 - 05:25:59	-

```

fc4/47*           04/01/19 - 05:25:59      -
fc4/48*           04/01/19 - 05:25:59      -
=====
! - Denotes port is link down but analytics enabled.
* - Denotes port in active analytics port sampling window.

```

ポートの横にあるアスタリスク記号 (*) は、そのポートが現在サンプリングされていることを示しています。

次に、すでに設定されているプッシュ クエリの出力の例を示します。

```

switch# show analytics query name iniitl result
{ "values": {
  "1": {
    "port": "fc1/6",
    "vsan": "10",
    "app_id": "255",
    "initiator_id": "0xe800a0",
    "target_id": "0xd601e0",
    "lun": "0000-0000-0000-0000",
    "active_io_read_count": "0",
    "active_io_write_count": "7",
    "total_read_io_count": "0",
    "total_write_io_count": "1008608573",
    "total_seq_read_io_count": "0",
    "total_seq_write_io_count": "1",
    "total_read_io_time": "0",
    "total_write_io_time": "370765952314",
    "total_read_io_initiation_time": "0",
    "total_write_io_initiation_time": "52084968152",
    "total_read_io_bytes": "0",
    "total_write_io_bytes": "2065630357504",
    "total_read_io_inter_gap_time": "0",
    "total_write_io_inter_gap_time": "16171468343166",
    "total_time_metric_based_read_io_count": "0",
    "total_time_metric_based_write_io_count": "1008608566",
    "total_time_metric_based_read_io_bytes": "0",
    "total_time_metric_based_write_io_bytes": "2065630343168",
    "read_io_rate": "0",
    "peak_read_io_rate": "0",
    "write_io_rate": "16070",
    "peak_write_io_rate": "32468",
    "read_io_bandwidth": "0",
    "peak_read_io_bandwidth": "0",
    "write_io_bandwidth": "32912384",
    "peak_write_io_bandwidth": "66494976",
    "read_io_size_min": "0",
    "read_io_size_max": "0",
    "write_io_size_min": "2048",
    "write_io_size_max": "2048",
    "read_io_completion_time_min": "0",
    "read_io_completion_time_max": "0",
    "write_io_completion_time_min": "111",
    "write_io_completion_time_max": "9166",
    "read_io_initiation_time_min": "0",
    "read_io_initiation_time_max": "0",
    "write_io_initiation_time_min": "36",
    "write_io_initiation_time_max": "3265",
    "read_io_inter_gap_time_min": "0",
    "read_io_inter_gap_time_max": "0",
    "write_io_inter_gap_time_min": "100",

```



```

"write_io_inter_gap_time_max": "1094718",
"peak_active_io_read_count": "0",
"peak_active_io_write_count": "23",
"read_io_aborts": "0",
"write_io_aborts": "0",
"read_io_failures": "0",
"write_io_failures": "0",
"read_io_scsi_check_condition_count": "0",
"write_io_scsi_check_condition_count": "0",
"read_io_scsi_busy_count": "0",
"write_io_scsi_busy_count": "0",
"read_io_scsi_reservation_conflict_count": "0",
"write_io_scsi_reservation_conflict_count": "0",
"read_io_scsi_queue_full_count": "0",
"write_io_scsi_queue_full_count": "0",
"sampling_start_time": "1529993232",
"sampling_end_time": "1529993260"
},
"2": {
  "port": "fc1/6",
  "vsan": "10",
  "app_id": "255",
  "initiator_id": "0xe800a1",
  "target_id": "0xd601e1",
  "lun": "0000-0000-0000-0000",
  "active_io_read_count": "0",
  "active_io_write_count": "8",
  "total_read_io_count": "0",
  "total_write_io_count": "1004271260",
  "total_seq_read_io_count": "0",
  "total_seq_write_io_count": "1",
  "total_read_io_time": "0",
  "total_write_io_time": "370004164726",
  "total_read_io_initiation_time": "0",
  "total_write_io_initiation_time": "51858511487",
  "total_read_io_bytes": "0",
  "total_write_io_bytes": "2056747540480",
  "total_read_io_inter_gap_time": "0",
  "total_write_io_inter_gap_time": "16136686881766",
  "total_time_metric_based_read_io_count": "0",
  "total_time_metric_based_write_io_count": "1004271252",
  "total_time_metric_based_read_io_bytes": "0",
  "total_time_metric_based_write_io_bytes": "2056747524096",
  "read_io_rate": "0",
  "peak_read_io_rate": "0",
  "write_io_rate": "16065",
  "peak_write_io_rate": "16194",
  "read_io_bandwidth": "0",
  "peak_read_io_bandwidth": "0",
  "write_io_bandwidth": "32901632",
  "peak_write_io_bandwidth": "33165824",
  "read_io_size_min": "0",
  "read_io_size_max": "0",
  "write_io_size_min": "2048",
  "write_io_size_max": "2048",
  "read_io_completion_time_min": "0",
  "read_io_completion_time_max": "0",
  "write_io_completion_time_min": "114",
  "write_io_completion_time_max": "9019",
  "read_io_initiation_time_min": "0",
  "read_io_initiation_time_max": "0",
  "write_io_initiation_time_min": "37",
  "write_io_initiation_time_max": "3158",
  "read_io_inter_gap_time_min": "0",

```

```

        "read_io_inter_gap_time_max": "0",
        "write_io_inter_gap_time_min": "101",
        "write_io_inter_gap_time_max": "869035",
        "peak_active_io_read_count": "0",
        "peak_active_io_write_count": "19",
        "read_io_aborts": "0",
        "write_io_aborts": "0",
        "read_io_failures": "0",
        "write_io_failures": "0",
        "read_io_scsi_check_condition_count": "0",
        "write_io_scsi_check_condition_count": "0",
        "read_io_scsi_busy_count": "0",
        "write_io_scsi_busy_count": "0",
        "read_io_scsi_reservation_conflict_count": "0",
        "write_io_scsi_reservation_conflict_count": "0",
        "read_io_scsi_queue_full_count": "0",
        "write_io_scsi_queue_full_count": "0",
        "sampling_start_time": "1529993232",
        "sampling_end_time": "1529993260"
    }
}

```



(注) これらのクエリの出力は JSON 形式です。

この例は、*fc-scsi* 分析タイプでサポートされているビューインスタンスのリストを示しています。

```

switch# show analytics schema fc-scsi views

fc-scsi db schema tables:
port
logical_port
app
scsi_target
scsi_initiator
scsi_target_app
scsi_initiator_app
scsi_target_tl_flow
scsi_target_it_flow
scsi_initiator_it_flow
scsi_target_itl_flow
scsi_initiator_itl_flow
scsi_target_io
scsi_initiator_io

```

この例は、*fc-nvme* 分析タイプでサポートされているビューインスタンスのリストを示しています。

```

switch# show analytics schema fc-nvme views

fc-nvme db schema tables:
port
logical_port
app
nvme_target

```

```

nvme_initiator
nvme_target_app
nvme_initiator_app
nvme_target_tn_flow
nvme_target_it_flow
nvme_initiator_it_flow
nvme_target_itn_flow
nvme_initiator_itn_flow
nvme_target_io
nvme_initiator_io

```

この例は、*fc-scsi.port* ビューインスタンスでサポートされているフローメトリックのリストを示しています。



- (注) 出力の *exceed_count* カウンタは、将来の Cisco MDS NX-OS リリースでサポートされる予定です。

```
switch# show analytics schema fc-scsi view-instance port
```

```

fc-scsi.port table schema columns:
*port
scsi_target_count
scsi_initiator_count
io_app_count
logical_port_count
scsi_target_app_count
scsi_initiator_app_count
active_io_read_count
active_io_write_count
scsi_target_it_flow_count
scsi_initiator_it_flow_count
scsi_target_itl_flow_count
scsi_initiator_itl_flow_count
scsi_target_tl_flow_count
total_abts_count
total_read_io_count
total_write_io_count
total_seq_read_io_count
total_seq_write_io_count
total_read_io_time
total_write_io_time
total_read_io_initiation_time
total_write_io_initiation_time
total_read_io_bytes
total_write_io_bytes
total_read_io_inter_gap_time
total_write_io_inter_gap_time
total_time_metric_based_read_io_count
total_time_metric_based_write_io_count
total_time_metric_based_read_io_bytes
total_time_metric_based_write_io_bytes
read_io_rate
peak_read_io_rate
write_io_rate
peak_write_io_rate
read_io_bandwidth
peak_read_io_bandwidth

```

```

write_io_bandwidth
peak_write_io_bandwidth
read_io_size_min
read_io_size_max
write_io_size_min
write_io_size_max
read_io_completion_time_min
read_io_completion_time_max
write_io_completion_time_min
write_io_completion_time_max
read_io_initiation_time_min
read_io_initiation_time_max
write_io_initiation_time_min
write_io_initiation_time_max
read_io_inter_gap_time_min
read_io_inter_gap_time_max
write_io_inter_gap_time_min
write_io_inter_gap_time_max
peak_active_io_read_count
peak_active_io_write_count
read_io_aborts
write_io_aborts
read_io_failures
write_io_failures
read_io_timeouts
write_io_timeouts
read_io_scsi_check_condition_count
write_io_scsi_check_condition_count
read_io_scsi_busy_count
write_io_scsi_busy_count
read_io_scsi_reservation_conflict_count
write_io_scsi_reservation_conflict_count
read_io_scsi_queue_full_count
write_io_scsi_queue_full_count
read_io_rate_exceed_count
write_io_rate_exceed_count
read_io_bandwidth_exceed_count
write_io_bandwidth_exceed_count
read_io_size_min_exceed_count
read_io_size_max_exceed_count
write_io_size_min_exceed_count
write_io_size_max_exceed_count
read_io_initiation_time_min_exceed_count
read_io_initiation_time_max_exceed_count
write_io_initiation_time_min_exceed_count
write_io_initiation_time_max_exceed_count
read_io_completion_time_min_exceed_count
read_io_completion_time_max_exceed_count
write_io_completion_time_min_exceed_count
write_io_completion_time_max_exceed_count
read_io_inter_gap_time_min_exceed_count
read_io_inter_gap_time_max_exceed_count
write_io_inter_gap_time_min_exceed_count
write_io_inter_gap_time_max_exceed_count
read_io_abort_exceed_count
write_io_abort_exceed_count
read_io_failure_exceed_count
write_io_failure_exceed_count
sampling_start_time
sampling_end_time

```

(* - indicates the metric is a 'key' for the table)

この例は、*fc-nvme.port* ビュー インスタンスでサポートされているフロー メトリックのリストを示しています。



- (注) 出力の *exceed_count* カウンタは、将来の Cisco MDS NX-OS リリースでサポートされる予定です。

```
switch# show analytics schema fc-nvme view-instance port
```

```
fc-nvme.port table schema columns:
*port
nvme_target_count
nvme_initiator_count
io_app_count
logical_port_count
nvme_target_app_count
nvme_initiator_app_count
active_io_read_count
active_io_write_count
nvme_target_it_flow_count
nvme_initiator_it_flow_count
nvme_target_itn_flow_count
nvme_initiator_itn_flow_count
nvme_target_tn_flow_count
total_abts_count
total_read_io_count
total_write_io_count
total_seq_read_io_count
total_seq_write_io_count
total_read_io_time
total_write_io_time
total_read_io_initiation_time
total_write_io_initiation_time
total_read_io_bytes
total_write_io_bytes
total_read_io_inter_gap_time
total_write_io_inter_gap_time
total_time_metric_based_read_io_count
total_time_metric_based_write_io_count
total_time_metric_based_read_io_bytes
total_time_metric_based_write_io_bytes
read_io_rate
peak_read_io_rate
write_io_rate
peak_write_io_rate
read_io_bandwidth
peak_read_io_bandwidth
write_io_bandwidth
peak_write_io_bandwidth
read_io_size_min
read_io_size_max
write_io_size_min
write_io_size_max
read_io_completion_time_min
read_io_completion_time_max
write_io_completion_time_min
write_io_completion_time_max
read_io_initiation_time_min
read_io_initiation_time_max
write_io_initiation_time_min
```

```

write_io_initiation_time_max
read_io_inter_gap_time_min
read_io_inter_gap_time_max
write_io_inter_gap_time_min
write_io_inter_gap_time_max
peak_active_io_read_count
peak_active_io_write_count
read_io_aborts
write_io_aborts
read_io_failures
write_io_failures
read_io_timeouts
write_io_timeouts
read_io_nvme_lba_out_of_range_count
write_io_nvme_lba_out_of_range_count
read_io_nvme_ns_not_ready_count
write_io_nvme_ns_not_ready_count
read_io_nvme_reservation_conflict_count
write_io_nvme_reservation_conflict_count
read_io_nvme_capacity_exceeded_count
write_io_nvme_capacity_exceeded_count
read_io_rate_exceed_count
write_io_rate_exceed_count
read_io_bandwidth_exceed_count
write_io_bandwidth_exceed_count
read_io_size_min_exceed_count
read_io_size_max_exceed_count
write_io_size_min_exceed_count
write_io_size_max_exceed_count
read_io_initiation_time_min_exceed_count
read_io_initiation_time_max_exceed_count
write_io_initiation_time_min_exceed_count
write_io_initiation_time_max_exceed_count
read_io_completion_time_min_exceed_count
read_io_completion_time_max_exceed_count
write_io_completion_time_min_exceed_count
write_io_completion_time_max_exceed_count
read_io_inter_gap_time_min_exceed_count
read_io_inter_gap_time_max_exceed_count
write_io_inter_gap_time_min_exceed_count
write_io_inter_gap_time_max_exceed_count
read_io_abort_exceed_count
write_io_abort_exceed_count
read_io_failure_exceed_count
write_io_failure_exceed_count
sampling_start_time
sampling_end_time

```

(* - indicates the metric is a 'key' for the table)

SAN Analytics のトラブルシューティング

ASICの問題により、交換への応答が別のリンクで受信された場合、ITOテーブルがフラッシュされない可能性があります（ポートチャネルフラップまたはそのようなまれなケースのため）。このイベント自体は分析に影響しません。ただし、これが多数のITLで発生し、ファブリックに多くのチャーンがある場合（ITOテーブルヒットを持つITLが静かになり、新しいITLセットがファブリックでアクティブになっているなど）、スケールは影響を受ける可能性

があります。スケール制限を超えると、AMC でエラーが発生する可能性があります。64G モジュールおよびスイッチでは、AlertMgrCollector (AMC) を介して分析が収集されます。

AMC リセット機能は、ASIC 分析のみをリセットすることにより、分析の中断のない回復を提供します。 **analytics reset module <module-number>** コマンドを使用して、ラインカードの AMC をリセットできます。スケールの制限の詳細については、[Cisco MDS NX-OS の構成の制限、リリース 9.x](#) を参照してください。

このコマンドは、AMC モジュールのみをリセットし、テーブル内のすべてのエントリをフラッシュし、AMC を ITO_HIT_ON_CMD から回復します。

例：

```
switch # analytics reset module 6
switch # 2022 Jun 15 12:24:48 sw184-9706
%ANALYTICS_LC_MGR-SLOT6-5-ANALYTICS_LC_MGR_RESET_SUCCESS:
Analytics reset successful on module 6
```

リセットが成功すると、次の syslog が表示されます。

```
switch# 2022 Mar 13 22:35:54 switch
%ANALYTICS_LC_MGR-SLOT6-5-ANALYTICS_LC_MGR_RESET_SUCCESS: Reset of Analytics
engine
succeeded.
```

リセットに失敗すると、次の syslog が表示されます。

```
switch# 2022 Mar 13 22:35:54 switch
%ANALYTICS_LC_MGR-SLOT6-3-ANALYTICS_LC_MGR_RESET_FAILURE: Reset of Analytics
engine
failed
```

失敗した syslog が表示された場合は、テクニカル サポートを収集し、回復のためにモジュールをリロードします。



第 4 章

SAN Telemetry Streaming の構成

この章では、SAN Telemetry Streaming 機能とその構成方法について説明します。

- [SAN Telemetry Streaming の設定の機能履歴](#) (111 ページ)
- [SAN Telemetry Streaming の概要](#) (112 ページ)
- [SAN Telemetry Streaming の注意事項と制約事項](#) (115 ページ)
- [gRPC エラーの動作](#) (116 ページ)
- [SAN テレメトリ ストリーミングのエンコーディング](#) (117 ページ)
- [SAN テレメトリ ストリーミングの設定](#) (118 ページ)
- [例 : SAN テレメトリ ストリーミングの設定](#) (121 ページ)
- [SAN テレメトリ ストリーミングの設定と統計情報の表示](#) (124 ページ)
- [SAN テレメトリ ストリーミングのトラブルシューティング](#) (130 ページ)

SAN Telemetry Streaming の設定の機能履歴

表 16 : SAN Telemetry Streaming の設定の機能履歴

機能名	リリース	機能情報
トランシーバー パラメータのストリーミング	9.2(2)	FC トランシーバーパラメータ ストリーミングのサポートが追加されました。
SAN Telemetry Streaming	8.4(1)	NVMe フローメトリックを使用して、 <i>fabric_telemetry.proto</i> ファイルを更新しました。
SAN Telemetry Streaming	8.3(2)	コンパクト Google Protocol Buffers (GPB) エンコーディングのサポートが追加されています。

機能名	リリース	機能情報
SAN Telemetry Streaming	8.3(1)	<p>ストリーミング アナリティクスやインターフェイスの統計情報の機能を DCNM などの受信者に提供します。</p> <p>次に、追加されたコマンドを示します。</p> <ul style="list-style-type: none"> • certificate <i>certificate_path host_name</i> • destination-group <i>id</i> • destination-profile • dst-grp <i>id</i> • feature telemetry • {ip ipv6} <i>address address port number [protocol procedural-protocol encoding encoding-protocol]</i> • path <i>sensor_path</i> • sensor-group <i>id</i> • show run telemetry • show telemetry {control {database [destination-groups destinations sensor-groups sensor-paths subscriptions] stats} data collector {brief details} pipeline stats transport <i>session_id</i> [errors stats]} • snsr-grp <i>id sample-interval interval</i> • subscription <i>id</i> • telemetry • use-retry size <i>buffer_size</i>
インターフェイスの統計情報	8.3(1)	<p>ファイバ チャネル インターフェイスからトラフィックおよびエラー カウンタのデータをストリーミングできます。</p>

SAN Telemetry Streaming の概要

Cisco NX-OS には、ネットワークからデータを収集するための複数のメカニズム（Simple Network Management Protocol (SNMP)、CLI、Syslog など）があります。SAN Telemetry Streaming 機能は、DCNM などの1つ以上のアップストリーム レシーバに、分析する特定のデータをストリーミングするために使用されます。SAN Analytics で使用されるプル モデルは、クライアントから要求された場合にのみサーバーからデータを送信するために使用されます。

一般的には、クライアントに継続的にデータをストリーミングするために使用されるプッシュ（フェッチ）モデルを使用してスイッチからデータが収集されます。SAN Telemetry Streaming はプッシュ モデルを有効にし、モニタリング データにほぼリアルタイムでアクセスできるようにします。

収集されたトラフィックおよびエラーカウンタのデータは、SAN Telemetry Streaming の構成でセンサー グループにセンサー パスを追加することで、DCNM、サードパーティ製デバイス、またはアプリケーションにストリーミングできます。詳細については、[SAN テレメトリ ストリーミングの設定（118 ページ）](#) を参照してください。



(注) Cisco MDS NX-OS リリース 8.3(1) では、テレメトリ ペイロードに追加されたバージョン番号は 1.0.0.1 です。

インターフェイスの統計情報のストリーミング

インターフェイスの統計情報のストリーミングを使用すると、ファイバチャネルインターフェイスからトラフィックおよびエラーカウンタのデータをストリーミングできます。トラフィックおよびエラーカウンタの収集はデフォルトで有効になっているため、設定または無効化することはできません。65 を超えるインターフェイスの統計情報カウンタを使用できます。インターフェイスの統計情報をサポートしているモジュールの詳細については、[SAN アナリティクス のハードウェア要件（11 ページ）](#) を参照してください。

サポートされているインターフェイス カウンタの一覧については、[インターフェイス カウンタ（271 ページ）](#) を参照してください。

トランシーバパラメータ ストリーミング

トランシーバパラメータ ストリーミングは、トランシーバに関する情報を定期的に収集し、それをレシーバにストリーミングします。この情報は、動作中の DOM (Diagnostic Optical Monitoring) データと、監視対象の各トランシーバのベンダー名、モデル番号、シリアル番号、およびスイッチのタイムスタンプに関する静的データの両方で構成されます。これにより、ローカル NX-OS オンスイッチ トランシーバパラメータのしきい値モニタリングを介した集中型および拡張トランシーバモニタリングが可能になります。

トランシーバ DOM の動作パラメータを経時的に分析することで、トランシーバのパフォーマンスの問題を特定できます。たとえば、ビット エラーやフレーム CRC などのインターフェイスエラーをトランシーバの受信電力レベルと関連付けることで、ケーブルの断続的な問題を特定することができます。タイムスタンプは、時間のシーケンシングや、他のデータやログとの関連付けに使用できます。

トランシーバパラメータストリーミングセンサーは、ローカルスイッチ トランシーバ データのみを収集するか、ローカルおよびピア トランシーバ データの両方を収集するように定義できます。



(注) ピア トランシーバー データを監視するには、ピア デバイスがインバンド FC 読み取り診断パラメータ (RDP) ELS 要求をサポートしている必要があります。

この機能は次のコンポーネントで構成されています。

- スイッチでの収集：表 17: ストリーミング トランシーバパラメータ (114 ページ) にリストされているトランシーバのパラメータは定期的に収集されます。これらは、トランシーバパラメータストリーミングとは関係なく、スイッチ上の NX-OS によってローカルに監視されます。
- レシーバへのストリーミング：テレメトリ構成コマンドを使用して、ストリーミングするインターフェイスの範囲と、トランシーバパラメータのストリーミングインターバルを指定します。ストリーミングは、古いデータがストリーミングされるのを避けるために、トランシーバーが動作してから 10 分後に開始されます。次に、レシーバはデータを監視および分析することができます。



(注) トランシーバパラメータのストリーミングは、ファイバチャネルポートでのみサポートされます。

表 17: ストリーミング トランシーバパラメータ (114 ページ) は、ストリーミングされるトランシーバパラメータのリストを表示します。

表 17: ストリーミング トランシーバパラメータ

トランシーバパラメータ	ユニット
温度	摂氏 (C)
電圧	ボルト (V)
電流	ミリアンペア (mA)
Tx Power	デシベル ミリワット (dBm)
Rx 電力	デシベル ミリワット (dBm)
ベンダー名 (Vendor Name)	—
モデル番号	—
シリアル番号	—
スイッチ タイムスタンプ	—

SAN Telemetry Streaming の注意事項と制約事項

- **feature telemetry** コマンドが有効になっている場合は、Cisco MDS NX-OS リリース 8.3(1) よりも前のリリースにダウングレードする前に、**no feature telemetry** コマンドを使用してこの機能を無効にします。
- Cisco MDS NX-OS リリース 8.3(2) の前は、SAN Telemetry Streaming は Google リモートプロシージャコール (gRPC) 転送を介した Google Protocol Buffers (GPB) エンコーディングのみサポートしていましたが、Cisco MDS NX-OS リリース 8.3(2) からは、コンパクト GPB エンコーディングのサポートが追加されています。宛先グループの下にあるすべての宛先とサブスクリプションの下にあるすべての宛先グループが同じエンコーディングタイプであることを確認します。



(注) GPB キー値エンコーディングは、単純に GPB として参照されています。GPB は、configuration および show コマンドで GPB キー値の代わりに使用されています。

- Cisco DCNM SAN Insights を使用している場合は、Cisco DCNM SAN Insights で SAN Telemetry Streaming 機能を構成できます。スイッチでこの機能を構成する必要はありません。詳細については、[Cisco DCNM SAN 管理コンフィギュレーションガイド \[英語\]](#) の「Configuring SAN Insights」セクションを参照してください。
- ストリーミング サンプル間隔 (**snsr-grp id sample-interval interval**)、ポート サンプリング間隔 (**analytics port-sampling module number size number interval seconds**)、およびプッシュクエリ間隔 (**analytics query “query_string” name query_name type periodic [interval seconds] [clear] [differential]**) は、同じ値に設定することをお勧めします。また、最初にプッシュクエリ間隔、次にポート サンプリング間隔、最後にストリーミング サンプル間隔を変更または設定することをお勧めします。
- サポートされている最小のストリーミング サンプル間隔は 30 秒です。プッシュクエリ間隔、ポート サンプリング間隔、およびストリーミング サンプル間隔は、最小推奨値の 30 秒以上にし、同じ値に設定することをお勧めします。最小値未満の間隔を設定すると、望ましくないシステム動作が発生する可能性があります。
- インターフェイスの統計情報のストリーミングは、Cisco NPV モードで動作している Cisco MDS 9132T スイッチではサポートされていません。
- 最大 2 つの管理レシーバ (宛先) がサポートされます。ただし、最適なパフォーマンスを実現するために、レシーバは 1 つだけ設定することをお勧めします。
- 複数のレシーバ (Cisco DCNM またはサードパーティ製デバイスやアプリ) を構成する場合は、同じ宛先グループの下に構成することをお勧めします。複数の Cisco DCNM レシーバがある場合は、それらのレシーバを手動で同じ宛先グループに設定する必要があります。

- SAN Telemetry Streaming レシーバが機能を停止すると、他のレシーバでデータフローが中断されます。失敗したレシーバを再起動します。レシーバの再起動方法については、レシーバのマニュアルを参照してください。

レシーバが遅延なく稼働していて、管理ポートでパケットドロップが発生していない場合、テレメトリ データ ストリーミングは均一です。レシーバまたはネットワークの遅さが原因で gRPC 転送に遅延が発生している場合、データ収集が中断され、システムメモリの制限によりスイッチ側でデータがドロップされる可能性があります。この問題が発生するかどうかは、ストリーム出力する ITL 数の程度、およびネットワークの遅延または速度低下の程度によって決まります。 **show telemetry control database sensor-groups**、**show telemetry transport session_id errors**、およびテレメトリ **syslog** コマンドを使用して、センサー グループ レベルでのドロップ数とすべての転送遅延がある場合はその転送ステータスを確認します。詳細については、[SAN テレメトリ ストリーミングのトラブルシューティング \(130 ページ\)](#) を参照してください。



- (注) ネットワークの速度低下が解決しない場合、または分析データの転送やストリーミング速度を 25 時間以上遅延させている継続的なネットワークの低下が発生している場合、転送セッションが永久的に無効化されて Syslog メッセージが生成されます。この問題が解決した後は、対応する宛先グループの下の IP アドレスを削除して構成することでストリーミングを再開できます。設定の詳細については、「[SAN テレメトリ ストリーミングの設定 \(118 ページ\)](#)」を参照してください。

gRPC エラーの動作

gRPC レシーバが 20 個のエラー (gRPC エラーの 1 つまたは両方) をスイッチに送信した後、スイッチクライアントは gRPC レシーバへの接続を無効にします。レシーバからの応答の受信に 30 秒以上かかり、この状態が 25 時間継続している場合、各転送セッションは「無効」とマークされます。gRPC レシーバ有効にするには、宛先グループの下の宛先 IP アドレスの設定を解除して再設定する必要があります。生成されたエラーを表示するには、**show telemetry transport session_id errors** コマンドを使用します。設定の詳細については、[SAN テレメトリ ストリーミングの設定 \(118 ページ\)](#) を参照、エラーについては、[SAN テレメトリ ストリーミングのトラブルシューティング \(130 ページ\)](#) を参照してください。

以下は gRPC エラーです。

- gRPC クライアントがセキュアな接続に対して誤った証明書を送信する。
- gRPC レシーバでのクライアントメッセージの処理に時間がかかりすぎて、タイムアウトが発生する。別のメッセージ処理スレッドを使用してメッセージを処理することで、タイムアウトを回避している。

SAN テレメトリ ストリーミングのエンコーディング

SAN テレメトリ ストリーミングでは、次のエンコーディングが使用されます。

- **GPB キー値** : Cisco MDS NX-OS リリース 8.3(2) の前は、GPB キー値が唯一サポートされているエンコーディングでした。このエンコーディングで使用されるキーは文字列で、自己記述型です。ただし、このエンコーディングで使用されるデータのサイズは、コンパクト GPB エンコーディングよりも大きくなります。このタイプのエンコーディングでは、中間プロセスなしでデータを簡単に分析できます。[key フィールドの詳細については、フローメトリック \(133 ページ\)](#) を参照してください。
- **コンパクト GPB** : Cisco MDS NX-OS リリース 8.3(2) からは、コンパクト GPB エンコーディングのサポートが追加されています。このエンコーディングで使用されるキーは整数です。そのため、このエンコーディングで使用されるデータのサイズは GPB-KV エンコーディングよりも小さくなります。ただし、整数をそれぞれのメトリックに復号化するには復号化テーブルが必要です。コンパクト GPB の復号化テーブルは `.proto` ファイルです。コンパクト GPB を使用する場合は、すべての `path analytics: query_name` クエリに対して `telemetry_bis.proto` ファイルを使用し、データ ストリームを解析するためにコレクタにこのファイルをアップロードする必要があります。



- (注) インターフェイスの統計情報のストリーミング (`path show_stats`) の場合は、GPB-KV エンコーディングのみサポートされます。

次に、コンパクト GPB `.proto` ファイルで使用されるテレメトリ フィールドのスニペットの表示例を示します。

```
message Telemetry {
  ...
  repeated TelemetryField data_gpbkv = 11;
  TelemetryGPBTable data_gpb = 12;
  ...
  message TelemetryGPBTable {
    repeated TelemetryRowGPB row = 1;
  }
  message TelemetryRowGPB {
    uint64 timestamp = 1;
    bytes keys = 10;
    bytes content = 11;
  }
}
```

この例では、コンパクト GPB の `.proto` ファイルで使用されているフィールドは、`data_gpb` フィールドの下に含まれています。TelemetryRowGPB メッセージ構造の `key` フィールドは、`.proto` ファイル名 (`fabric_telemetry`) を伝送し、`content` フィールドは `.proto` ファイルからのフィールドを伝送します。

コンパクト GPB で使用される `.proto` ファイルの詳細については、[SAN テレメトリ ストリーミング Proto ファイル \(277 ページ\)](#) を参照してください。

SAN テレメトリ ストリーミングの設定



- (注) Cisco DCNM SAN Insights を使用している場合は、Cisco DCNM SAN Insights で SAN Telemetry Streaming 機能を構成できます。スイッチでこの機能を構成する必要はありません。詳細については、[Cisco DCNM SAN 管理コンフィギュレーションガイド \[英語\]](#) の「Configuring SAN Insights」セクションを参照してください。

後続の図は、センサーと宛先グループのさまざまな設定方法を示しています。

図 32: 同じ宛先グループにマッピングされているセンサーグループ

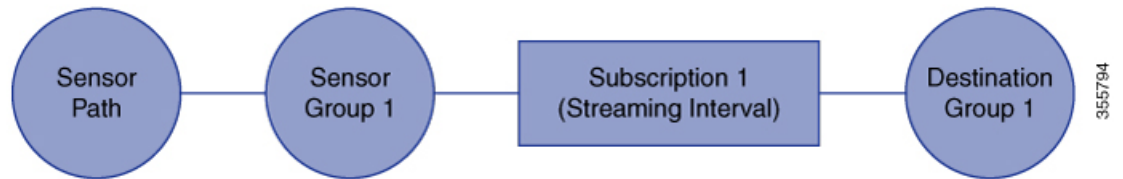


図 33: 別の宛先グループにマッピングされているセンサーグループ

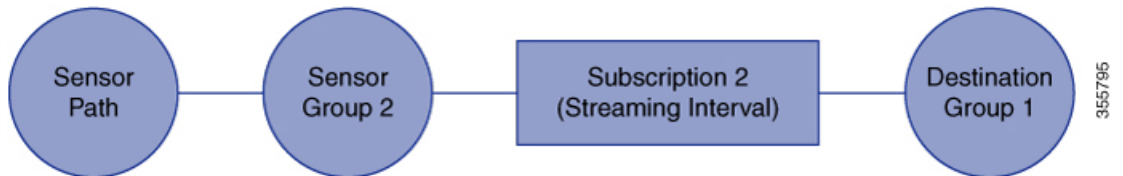


図 34: 複数の宛先グループにマッピングされている 1つのセンサーグループ

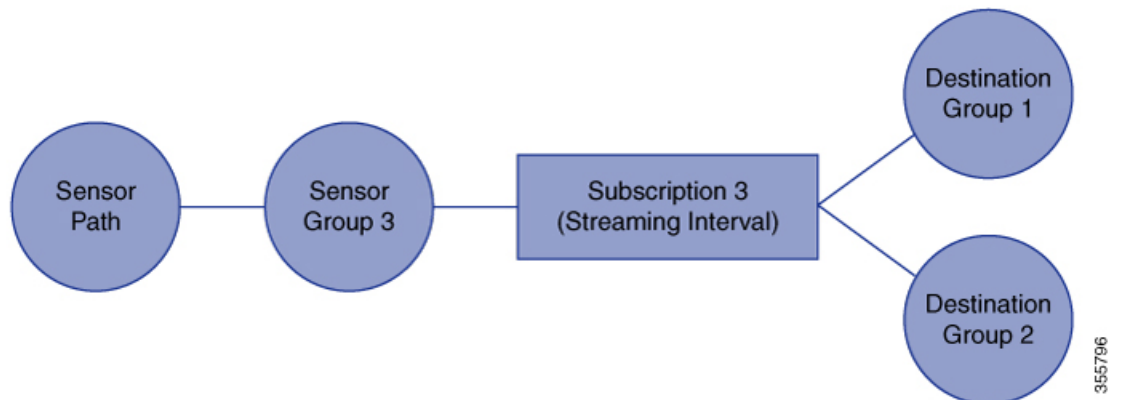
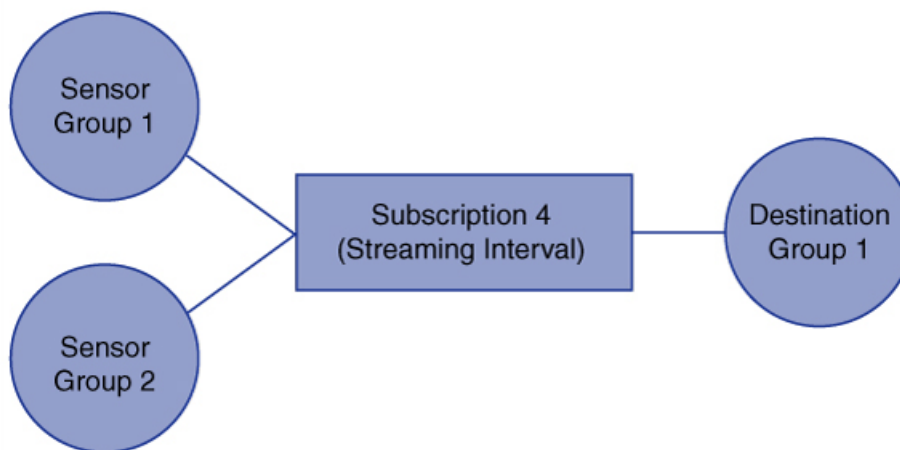


図 35: 単一の宛先グループにマッピングされている複数のセンサーグループ



355797

SAN Telemetry Streaming を構成するには、次の手順を実行します。

始める前に

- スイッチが Cisco MDS NX-OS リリース 8.3(1) 以降のリリースを実行していることを確認します。
- SAN アナリティクス機能を有効にします。「[SAN アナリティクスの有効化 \(32 ページ\)](#)」を参照してください。
- テレメトリ ソーススイッチのタイムゾーンが **clock** 構成コマンドで正しく設定されていることを確認します。そうしないと、SAN テレメトリ レシーバーは、受信した分析タイムスタンプを関連付けることができません。このコマンドの詳細については、『[Cisco MDS 9000 シリーズ コマンド リファレンス](#)』を参照してください。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 SAN Telemetry Streaming 機能を有効にします。

```
switch(config)# feature telemetry
```

ステップ 3 SAN Telemetry Streaming の構成モードに入ります。

```
switch(config)# telemetry
```

ステップ 4 (オプション) 既存の SSL または TLS 証明書を使用します。

```
switch(config-telemetry)# certificate certificate_path host_name
```

(注) Cisco MDS 9700 シリーズ スイッチでは、セキュアなテレメトリ構成のために、アクティブ スーパーバイザとスタンバイ スーパーバイザの両方でクライアント証明書を使用できることを確認してください。そうしないと、アップグレードまたはダウングレード後に SAN Telemetry Streaming が失敗します。 **copy bootflash:<client certificate file> bootflash://sup-standby/<client certificate file>** コマンドを使用して、アクティブ スーパーバイザからスタンバイ スーパーバイザにクライアント証明書をコピーします。

ステップ 5 (オプション) 宛先プロファイル構成モードを開始し、gRPC 転送プロトコルの送信の再試行の詳細を指定します。

1. switch(config-telemetry)# **destination-profile**
2. switch(conf-tm-dest-profile)# **use-retry size buffer_size**

宛先プロファイルには、たとえば、すべての宛先に固有のトランスポート リトライ バッファ サイズなどのパラメータを構成できます。

(注) バッファ サイズは MB 単位で、範囲は 10 ~ 1500 です。

ステップ 6 ID を持つセンサー グループを作成し、センサー グループ コンフィギュレーション モードを開始します。

```
switch(conf-tm-dest-profile)# sensor-group id
```

センサー グループは 1 つ以上のセンサー パスの集合です。

現在は、数字のセンサー グループ ID 値のみサポートされています。センサー グループでは、テレメトリ レポートの監視対象ノードを定義します。

ステップ 7 センサー グループにセンサー パスを追加します。

```
switch(conf-tm-sensor)# path sensor_path
```

sensor_path では、ストリーミングされる特定のインターフェイスの統計情報とプッシュ クエリが指定されます。1 つのセンサー グループに複数のセンサー パスを設定できます。テレメトリのセンサー パスは **path analytics: query_name**、インターフェイスの統計情報ストリーミングのセンサー パスは **path show_stats_fc slot/port** です。

(注) センサー パスの構文は設定時には検証されません。センサー パスが正しくないと、データ ストリーミングが失敗する可能性があります。

ステップ 8 宛先グループを作成して、宛先グループ コンフィギュレーション モードを開始します。

```
switch(conf-tm-sensor)# destination-group id
```

現在は、宛先グループ ID は数字の ID 値のみサポートしています。

(注) 宛先グループは 1 つ以上の宛先の集合です。

ステップ 9 発信データの宛先プロファイルを作成します。

```
switch(conf-tm-dest)# {ip | ipv6} address address port number [ protocol procedural-protocol encoding encoding-protocol]
```

(注) Cisco MDS NX-OS リリース 8.3(2) の時点では、gRPC がサポートされている唯一の転送プロトコルです。また、GPB とコンパクト GPB エンコーディングだけがサポートされています。

宛先グループがサブスクリプション ノードにリンクされている場合、テレメトリ データは、宛先プロファイルで指定されている IP アドレスとポートに送信されます。

ステップ 10 ID を持つサブスクリプション ノードを作成し、サブスクリプション 構成モードを開始します。

```
switch(conf-tm-dest)# subscription id
```

サブスクリプションは、センサー グループを宛先グループにマッピングします。

現在、サブスクリプション ID は数字の ID 値のみサポートしています。

ステップ 11 ID を持つセンサー グループをこのサブスクリプション ノードにリンクして、データ ストリーミングのサンプル間隔（ミリ秒単位）を設定します。

```
switch(conf-tm-sub)# snsr-grp id sample-interval interval
```

(注) 推奨の最小サンプル間隔は 30000 です。

現在、センサー グループ ID は数字の ID 値のみサポートしています。ストリーミング サンプル間隔の値を指定します。値はミリ秒単位である必要があります。サポートされているストリーミング サンプル間隔の最小値は 30000 ミリ秒です。最小値より大きい間隔値の場合、テレメトリ データが指定された間隔で定期的に送信される頻度に基づいたサブスクリプションが作成されます。

ステップ 12 ID を持つ宛先グループをこのサブスクリプションにリンクします。

```
switch(conf-tm-sub)# dst-grp id
```

現在は、宛先グループ ID は数字の ID 値のみサポートしています。

例：SAN テレメトリ ストリーミングの設定

この例では、ファイバチャネル インターフェイス 3/1 と 4/1 のデータを 30 秒ごとに、宛先 IP 1.2.3.4 ポート 50003 および IP 1:1::1:1 ポート 50009 にストリーミングし、test.pem を使用して検証された GPB エンコーディングを使用してストリームを暗号化するサブスクリプションの作成方法を示します。

```
switch# configure terminal
switch(config)# telemetry
switch(config-telemetry)# certificate /bootflash/test.pem foo.test.google.fr

switch(conf-tm-telemetry)# destination-group 100
switch(conf-tm-dest)# ip address 1.2.3.4 port 50003 protocol gRPC encoding GPB
```

例 : SAN テレメトリ ストリーミングの設定

```

switch(config-tm-dest)# destination-group 1
switch(config-tm-dest)# ipv6 address 1:1::1:1 port 50009 protocol gRPC encoding GPB-compact

switch(config-dest)# sensor-group 100
switch(config-tm-sensor)# path show_stats_fc3/1
switch(config-tm-sensor)# subscription 100
switch(config-tm-sub)# snsr-grp 100 sample-interval 30000
switch(config-tm-sub)# dst-grp 100

switch(config-dest)# sensor-group 1
switch(config-tm-sensor)# path show_stats_fc4/1
switch(config-tm-sensor)# subscription 1
switch(config-tm-sub)# snsr-grp 1 sample-interval 30000
switch(config-tm-sub)# dst-grp 1

```

この例では、30 秒ごとに **show** コマンドデータを収集し、レシーバ 1.2.3.4 と 1.1::1.1 に送信する定期的な収集の作成方法を示します。

```

switch# configure terminal
switch(config)# telemetry

switch(config-telemetry)# destination-group 100
switch(config-tm-dest)# ip address 1.2.3.4 port 60001 protocol gRPC encoding GPB

switch(config-tm-sensor)# destination-group 1
switch(config-tm-dest)# ipv6 address 1:1::1:1 port 60009 protocol gRPC encoding GPB-compact

switch(config-dest)# sensor-group 100
switch(config-tm-sensor)# subscription 100
switch(config-tm-sub)# snsr-grp 100 sample-interval 30000
switch(config-tm-sub)# dst-grp 100

switch(config-tm-dest)# sensor-group 1
switch(config-tm-sensor)# subscription 1
switch(config-tm-dest)# snsr-grp 1 sample-interval 30000
switch(config-tm-sub)# dst-grp 1

```

次に、センサーグループに複数のパスを含め、宛先グループに複数の宛先プロファイルを含め、サブスクリプションを複数のセンサーグループと宛先グループにリンクできる例を示します。

```

switch# configure terminal
switch(config)# telemetry

switch(config-telemetry)# sensor-group 100
switch(config-tm-sensor)# path analytics:init
switch(config-tm-sensor)# path analytics:inititl

switch(config-telemetry)# sensor-group 200
switch(config-tm-sensor)# path analytics:inititl

switch(config-tm-sensor)# destination-group 100
switch(config-tm-dest)# ip address 1.2.3.4 port 50004
switch(config-tm-dest)# ipv6 address 5:6::7:8 port 50005

switch(config-tm-dest)# destination-group 200
switch(config-tm-dest)# ip address 5.6.7.8 port 50001

```

```

switch(conf-tm-dest)# subscription 600
switch(conf-tm-sub)# snsr-grp 100 sample-interval 30000
switch(conf-tm-sub)# snsr-grp 200 sample-interval 30000
switch(conf-tm-sub)# dst-grp 100
switch(conf-tm-sub)# dst-grp 200

switch(conf-tm-dest)# subscription 900
switch(conf-tm-sub)# snsr-grp 200 sample-interval 30000
switch(conf-tm-sub)# dst-grp 100

```



- (注) *sensor_path* は、ストリーミングされる特定のインターフェイスの統計情報とプッシュクエリが指定される場所です。1つのセンサーグループに複数のセンサーパスを設定できます。テレメトリ ストリーミングのセンサーパスは **path analytics: query_name** であり、インターフェイスの統計情報ストリーミングの場合、センサーパスは **path show_stats_fc slot/port** です。センサーパスで指定されているクエリ名「init」、「initit」、および「inititl」は、SAN Analytics 機能で構成されています。詳細については、[プッシュクエリの設定 \(60 ページ\)](#) を参照してください。

次の例に、トランシーバストリーミングのサンプル構成を示します。

```

switch# configure terminal
switch(config)# telemetry

switch(config-telemetry)# sensor-group 200
switch(conf-tm-sensor)# path transceiver:fc1/1
switch(conf-tm-sensor)# path transceiver:fc13/1-48

switch(conf-tm-sensor)# show telemetry data collector details
-----
Row ID           Successful      Failed          Skipped          Sensor Path(GroupId)
-----
1                 398             14              0                show_stats_fc3/1-48(100)
2                 30488           0                1                analytics:dcnmtgtITL(2)
3                 395             0                0                show_stats_fc5/1-48(100)
4                 0               0                0                transceiver:fc1/1(200)
5                 0               0                0                transceiver:fc13/1-48(200)
6                 0               0                0                analytics:dcnmtgtITN(1)

```

この例は、サンプル構成と、SAN Telemetry Streaming 構成を確認する方法を示しています。SAN Telemetry Streaming 構成を確認するために、**show telemetry data collector details** および **show telemetry transport session_id stats** コマンドの出力を確認することもできます。詳細については、[SAN テレメトリ ストリーミングの設定と統計情報の表示 \(124 ページ\)](#) を参照してください。

```

switch# configure terminal
switch(config)# telemetry

switch(config-telemetry)# destination-group 100
switch(conf-tm-dest)# ip address 1.2.3.4 port 50003 protocol gRPC encoding GPB
switch(conf-tm-dest)# ip address 1.2.3.4 port 50004 protocol gRPC encoding GPB

```

```

switch(config-telemetry)# destination-group 1
switch(conf-tm-dest)# ipv6 address 1:1::1:1 port 50008 protocol gRPC encoding GPB-compact
switch(conf-tm-dest)# ipv6 address 1:2::3:4 port 50009 protocol gRPC encoding GPB-compact

switch(conf-tm-dest)# end

switch# show running-config telemetry
!Command: show running-config telemetry
!Running configuration last done at: Thu Jun 14 08:14:24 2018
!Time: Thu Jun 14 08:14:40 2018
version 8.3(1)
feature telemetry
telemetry
destination-group 1
  ipv6 address 1:2::3:4 port 50008 protocol gRPC encoding GPB-compact
  ipv6 address 1:1::1:1 port 50009 protocol gRPC encoding GPB-compact
destination-group 100
  ip address 1.2.3.4 port 50003 protocol gRPC encoding GPB
  ip address 1.2.3.4 port 50004 protocol gRPC encoding GPB

```



(注) NPU の負荷は、アクティブな ITL と非アクティブな ITL の数を含む、すべての ITL に基づいています。したがって、NPU の負荷を確認する前に、クエリをクリアまたはページすることを勧めます。

SAN テレメトリ ストリーミングの設定と統計情報の表示

次の Cisco NX-OS CLI **show** コマンドを使用して、SAN Telemetry Streaming 構成、統計情報、エラー、およびセッション情報を表示します。

この例は、SAN Telemetry Streaming の構成を反映している内部データベースを示します。

```

switch# show telemetry control database
Subscription Database size = 1
-----
Subscription ID      Data Collector Type
-----
100                  SDB

Sensor Group Database size = 1
-----
Row ID  Sensor Group ID  Sensor Group type  Sampling interval(ms)  Linked subscriptions
SubID
-----
1       100              Timer /SDB          30000 /Running          1
100
Collection Time in ms (Cur/Min/Max): 53/9/81
Encoding Time in ms (Cur/Min/Max): 21/6/33
Transport Time in ms (Cur/Min/Max): 10470/1349/11036
Streaming Time in ms (Cur/Min/Max): 10546/9/11112

Collection Statistics:
  collection_id_dropped      = 0
  last_collection_id_dropped = 0
  drop_count                 = 0

```

Sensor Path Database size = 4

```
-----
Row ID  Subscribed  Linked  Sec  Retrieve  Path  Query:  Filter
          Groups  Groups level  (GroupId):
-----
1      No      1      0      Self      analytics:inititl(100): NA :  NA
GPB Encoded Data size in bytes (Cur/Min/Max): 162310/162014/162320
JSON Encoded Data size in bytes (Cur/Min/Max): 0/0/0

2      No      1      0      Self      show_stats_fcl/3(100): NA :  NA
GPB Encoded Data size in bytes (Cur/Min/Max): 2390/2390/2390
JSON Encoded Data size in bytes (Cur/Min/Max): 0/0/0

3      No      1      0      Self      analytics:initit(100): NA :  NA
GPB Encoded Data size in bytes (Cur/Min/Max): 158070/157444/158082
JSON Encoded Data size in bytes (Cur/Min/Max): 0/0/0

4      No      1      0      Self      analytics:init(100): NA :  NA
GPB Encoded Data size in bytes (Cur/Min/Max): 159200/158905/159212
JSON Encoded Data size in bytes (Cur/Min/Max): 0/0/0

```

Destination Group Database size = 1
> use-vrf : default

```
-----
Destination Group ID  Refcount
-----
```

```
100 1
```

Destination Database size = 3

```
-----
Dst IP Addr  Dst Port  Encoding  Transport  Count
-----
10.30.217.80  50009     GPB       gRPC       1
2001:420:301:2005:3::11
60003     GPB       gRPC       1
2001:420:54ff:a4::230:e5
50013     GPB       gRPC       1

```

switch(conf-tm-dest)# **show telemetry control database sensor-groups**
Sensor Group Database size = 1

```
-----
Row ID Sensor Group ID Sensor Group type  Sampling interval(ms)  Linked subscriptions
SubID
-----
```

```
1      100      Timer  /SDB      30000     /Running     1
100
```

Collection Time in ms (Cur/Min/Max): 53/9/81

Encoding Time in ms (Cur/Min/Max): 21/21/33

Transport Time in ms (Cur/Min/Max): 10304/461/15643

Streaming Time in ms (Cur/Min/Max): 10380/9/15720

Collection Statistics:

collection_id_dropped = 0

last_collection_id_dropped = 0

drop_count = 0



- (注) コマンド出力内の SDB は、SAN データ コレクターのタイプです。テレメトリは、他のサポート対象プラットフォームの DME、NX-API、および YANG データ ソースもサポートしていません。

この例は、SAN Telemetry Streaming の構成の内部データベースの統計を示します。

```
switch# show telemetry control stats
show telemetry control stats entered
```

Error Description	Error Count
Chunk allocation failures	0
Sensor path Database chunk creation failures	0
Sensor Group Database chunk creation failures	0
Destination Database chunk creation failures	0
Destination Group Database chunk creation failures	0
Subscription Database chunk creation failures	0
Sensor path Database creation failures	0
Sensor Group Database creation failures	0
Destination Database creation failures	0
Destination Group Database creation failures	0
Subscription Database creation failures	0
Sensor path Database insert failures	0
Sensor Group Database insert failures	0
Destination Database insert failures	0
Destination Group Database insert failures	0
Subscription insert to Subscription Database failures	0
Sensor path Database delete failures	0
Sensor Group Database delete failures	0
Destination Database delete failures	0
Destination Group Database delete failures	0
Delete Subscription from Subscription Database failures	0
Sensor path delete in use	0
Sensor Group delete in use	0
Destination delete in use	0
Destination Group delete in use	0
Delete destination(in use) failure count	0
Sensor path Sensor Group list creation failures	0
Sensor path prop list creation failures	0
Sensor path sec Sensor path list creation failures	0
Sensor path sec Sensor Group list creation failures	0
Sensor Group Sensor path list creation failures	0
Sensor Group Sensor subs list creation failures	0
Destination Group subs list creation failures	0
Destination Group Destinations list creation failures	0
Destination Destination Groups list creation failures	0
Subscription Sensor Group list creation failures	0
Subscription Destination Groups list creation failures	0
Sensor Group Sensor path list delete failures	0
Sensor Group Subscriptions list delete failures	0
Sensor Group Subscriptions unsupported data-source failures	0
Destination Group Subscriptions list delete failures	0
Destination Group Destinations list delete failures	0
Subscription Sensor Groups list delete failures	0
Subscription Destination Groups list delete failures	0
Destination Destination Groups list delete failures	0
Failed to delete Destination from Destination Group	0
Failed to delete Destination Group from Subscription	0


```
Failed to delete Sensor Group from Subscription      0
Failed to delete Sensor path from Sensor Group     0
Failed to get encode callback                      0
Failed to get transport callback                   0
```

次に、データ収集に関する簡単な統計情報の例を示します。

```
switch# show telemetry data collector brief
```

Row ID	Collector Type	Successful	Failed	Skipped
1	NX-API	0	0	0
2	SDB	1513	902	0

この例では、すべてのセンサーパスの詳細を含む、データ収集に関する詳細な統計情報を示します。

```
switch# show telemetry data collector details
```

Row ID	Successful	Failed	Skipped	Sensor Path(GroupId)
1	496	305	0	analytics:inititl(100)
2	16	0	0	show_stats_fcl/3(100)
3	507	294	0	analytics:initit(100)
4	498	303	0	analytics:init(100)



(注) 出力でスキップされた数は、ゼロ差分レコードがフェッチされた回数を示します。

この例では、SAN Telemetry Streaming パイプラインの統計情報を表示します。SAN Telemetry Streaming パイプラインからは、収集キューと転送キューの統計情報（キューサイズ、キューのドロップ数など）が得られます。

```
switch# show telemetry pipeline stats
Main Statistics:
  Timers:
    Errors:
      Start Fail      =    0

  Data Collector:
    Errors:
      Node Create Fail =    0

  Event Collector:
    Errors:
      Node Create Fail =    0   Node Add Fail   =    0
      Invalid Data     =    0

  Memory:
    Allowed Memory Limit = 838860800 bytes
    Occupied Memory      = 53399552 bytes

Queue Statistics:
```

```

Request Queue:
  High Priority Queue:
    Info:
      Actual Size      = 50    Current Size      = 0
      Max Size         = 0     Full Count        = 0

    Errors:
      Enqueue Error    = 0     Dequeue Error     = 0

  Low Priority Queue:
    Info:
      Actual Size      = 50    Current Size      = 0
      Max Size         = 0     Full Count        = 0

    Errors:
      Enqueue Error    = 0     Dequeue Error     = 0

Data Queue:
  High Priority Queue:
    Info:
      Actual Size      = 160000  Current Size      = 0
      Max Size         = 0     Full Count        = 0

    Errors:
      Enqueue Error    = 0     Dequeue Error     = 0

  Low Priority Queue:
    Info:
      Actual Size      = 2     Current Size      = 0
      Max Size         = 0     Full Count        = 0

    Errors:
      Enqueue Error    = 0     Dequeue Error     = 0

```

この例は、構成されているすべての転送セッションを示します。

```
switch# show telemetry transport
```

Session Id	IP Address	Port	Encoding	Transport	Status
2	10.30.217.80	50009	GPB	gRPC	Connected
0	2001:420:301:2005:3::11	60003	GPB	gRPC	Connected
1	2001:420:54ff:a4::230:e5	50013	GPB	gRPC	Transmit Error

```

Retry buffer Size:          10485760
Event Retry Messages (Bytes): 0
Timer Retry Messages (Bytes): 10272300
Total Retries sent:         0
Total Retries Dropped:      5377

```

次に、特定の転送セッションの詳細なセッション情報の例を示します。

```
switch# show telemetry transport 0
```

```

Session Id:          2
IP Address:Port      10.30.217.80:50009
Transport:           GRPC
Status:              Connected

```

```

Last Connected:      Fri Jun 22 07:07:12.735 UTC
Last Disconnected:  Never
Tx Error Count:      0
Last Tx Error:       None
Event Retry Queue Bytes:  0
Event Retry Queue Size:  0
Timer Retry Queue Bytes:  0
Timer Retry Queue Size:  0
Sent Retry Messages:  0
Dropped Retry Messages:  0

```

次に、特定の転送セッションの詳細の例を示します。

```

switch# show telemetry transport 2 stats

Session Id:          2
Connection Stats
  Connection Count    2
  Last Connected:     Fri Jun 22 07:07:12.735 UTC
  Disconnect Count    0
  Last Disconnected:  Never
Transmission Stats
  Compression:        disabled
  Source Interface:   not set()
  Transmit Count:     44
  Last TX time:       Fri Jun 22 07:14:16.533 UTC
  Min Tx Time:        227 ms
  Max Tx Time:        3511 ms
  Avg Tx Time:        1664 ms
  Cur Tx Time:        227 ms

```

次のコマンドでは、特定の転送セッションの詳細なエラーの統計情報が表示されます。

```

switch# show telemetry transport 2 errors

Session Id:          1
Connection Errors
  Connection Error Count:  0
Transmission Errors
  Tx Error Count:         1746
  Last Tx Error:          Fri Jun 22 07:15:07.970 UTC
  Last Tx Return Code:    UNAVAILABLE

```



(注) 以下は、**show telemetry transport errors** コマンド出力の戻りコードの説明です。

- OK : エラーは検出されませんでした。
- UNAVAILABLE : 設定された IP アドレスまたはポートに到達できません。設定をチェックして、正しい IP アドレスまたはポートが設定されていることを確認します。
- DEADLINE_EXCEEDED : レシーバからの応答が 30 秒以上ないか、ネットワークの遅延が発生しています。

SAN テレメトリ ストリーミングのトラブルシューティング

show tech-support telemetry コマンドを使用して、トラブルシューティング用のテレメトリ データを収集します。エラーを発見した場合は、[SAN テレメトリ ストリーミングの設定 \(118 ページ\)](#) をチェックして構成を確認します。

テレメトリ ステータスのトラブルシューティングには、次の情報を使用します。

1. **show analytics system-load** コマンドを使用して、NPU の負荷を確認します。NPU の負荷が高い場合は、一部のポートでの分析を無効にします。

```
switch# show analytics system-load
n/a - not applicable
----- Analytics System Load Info -----
| Module | NPU Load (in %) | I7Ls | ITNs | Both | Hosts | Targets |
|         | SCSI NVMe Total | SCSI | NVMe | Total | SCSI | NVMe | Total | SCSI | NVMe | Total |
-----
| 1       | 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4       | 64 0 64 | 20743 | 0 | 20743 | 0 | 0 | 346 | 0 | 346 |
| 5       | 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8       | 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12      | 0 12 12 | 0 | 300 | 300 | 0 | 0 | 40 | 40 | 40 |
| 13      | 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18      | 0 13 13 | 1 | 1 | 2 | 1 | 2 | 0 | 0 | 0 | 0 |
| Total   | n/a n/a n/a | 20744 | 301 | 21045 | 1 | 1 | 346 | 40 | 386 |

As of Mon Apr 1 05:31:10 2019
```

2. **show telemetry control database sensor-groups** コマンドを使用し、コマンド出力でサンプル インターバルタイマーが動作しているかどうかを確認します。タイマーが動作していない場合は、タイマーが正しく設定されているかどうかを確認します。

```
switch# show telemetry control database sensor-groups
Sensor Group Database size = 3
-----
Row ID      Sensor Group ID  Sensor Group type  Sampling interval(ms)  Linked
subscriptions  SubID
-----
1           100              Timer /SDB           5000 /Running           1
100
Collection Time in ms (Cur/Min/Max): 0/0/1
Encoding Time in ms (Cur/Min/Max): 0/0/0
Transport Time in ms (Cur/Min/Max): 0/0/0
Streaming Time in ms (Cur/Min/Max): 1/1/4753

Collection Statistics:
  collection_id_dropped      = 0
  last_collection_id_dropped = 0
  drop_count                 = 0

2           1              Timer /SDB           30000 /Running           1
1
Collection Time in ms (Cur/Min/Max): 5/4/16
Encoding Time in ms (Cur/Min/Max): 2/2/11
Transport Time in ms (Cur/Min/Max): 644/635/1589
Streaming Time in ms (Cur/Min/Max): 3223/3168/4964

Collection Statistics:
  collection_id_dropped      = 0
```

```
last_collection_id_dropped = 0
drop_count                  = 0
```

3. **show telemetry data collector details** コマンド使用し、コマンド出力でデータの収集にエラーがあるかどうかを確認します。エラーを発見した場合は、SAN Telemetry Streaming の構成時に指定した *sensor_path* が誤っているため修正する必要があります。

```
switch# show telemetry data collector details
```

```
-----
Row ID          Successful  Failed Skipped  Sensor Path(GroupId)
-----
1                0           2994    0         analytics:panup(1)
2              2994         0        0         show_stats_fc2/2(1)
3                0           2994    0         analytics:port(1)
4              2994         0        0         show_stats_fc2/6(1)
5              2994         0        0         show_stats_fc2/1(1)
```

4. **show logging logfile | grep -i telemetry** コマンドを使用して、Syslog メッセージ内のエラーを確認します。

```
switch# show logging logfile | grep -i telemetry
2018 Jun 28 16:26:17 switch %TELEMETRY-4-TRANSPORT_SEND_ERROR: GRPC send to
172.20.30.129:60002 failed. (DEADLINE_EXCEEDED(len:2876013))
```

5. **ステップ 1**、**ステップ 2**、および**ステップ 3** で使用して問題が見つからなかった場合、問題は転送プロトコルにあることが考えられます。**show telemetry transport 0 errors** コマンドを使用して、コマンド出力で転送プロトコルにエラーがあるかどうかを確認します。

次の理由が原因で転送プロトコルエラーが発生することがあります。

- 宛先プロファイルまたはサブスクリプションで、間違った IP アドレスまたはポートを設定している。宛先プロファイルまたはサブスクリプションの IP アドレスまたはポートを修正します。
- レシーバが起動していない。レシーバがアクティブになっていて、gRPC ポートをリスンしていることを確認します。
- レシーバは起動しているが、メッセージを処理していない。レシーバアプリケーションのエラーを確認します。
- 管理 IP の問題がある。**telnet** コマンドを使用して、IP アドレスとポートに到達可能かどうかをテストします。

```
switch# show telemetry transport 1 errors
```

```
Session Id:                1
Connection Errors
  Connection Error Count:   0
Transmission Errors
  Tx Error Count:          0
  Last Tx Error:           None
```

Last Tx Return Code: OK



付録 **A**

付録

- [フローメトリック \(133 ページ\)](#)
- [インターフェイス カウンタ \(271 ページ\)](#)
- [SAN テレメトリ ストリーミング Proto ファイル \(277 ページ\)](#)

フローメトリック

このセクションでは、各フローメトリックに関する詳細情報について説明します。SAN Analytics にはフローメトリックの長い名前が使用され、SAN Telemetry Streaming の目的には短い名前が使用されます。



- (注)
- `total_abts_count` フローメトリックは、SCSI 分析タイプについてのみ更新されます。
 - Cisco MDS NX-OS リリース 9.2(2) から、次のビューインスタンスが非推奨になりました。
 - アプリケーション ビュー インスタンス (`app`)
 - ターゲット アプリケーション ビュー インスタンス (`scsi_target_app` および `nvme_target_app`)
 - イニシエーター アプリケーション ビュー インスタンス (`scsi_initiator_app` および `nvme_initiator_app`)
 - Cisco MDS NX-OS リリース 9.2(2) から、次のメトリックが廃止されました。
 - `total_seq_read_io_count`
 - `total_seq_write_io_count`
 - `read_io_inter_gap_time_min`
 - `read_io_inter_gap_time_max`
 - `write_io_inter_gap_time_min`
 - `write_io_inter_gap_time_max`
-

以下は、サポートされているビューの一覧です。

- ポート ビュー インスタンス (ポート)
- 論理ポート ビュー インスタンス (logical_port)
- アプリケーション ビュー インスタンス (app)
- ターゲット ビュー インスタンス (scsi_target および nvme_target)
- イニシエーター ビュー インスタンス (scsi_initiator および nvme_initiator)
- ターゲット アプリケーション ビュー インスタンス (scsi_target_app および nvme_target_app)
- イニシエーター アプリケーション ビュー インスタンス (scsi_initiator_app および nvme_initiator_app)
- ターゲット IT フロー ビュー インスタンス (scsi_target_it_flow および nvme_target_it_flow)
- イニシエーター IT フロー ビュー インスタンス (scsi_initiator_it_flow および nvme_initiator_it_flow)
- ターゲット TL フロー ビュー インスタンス (scsi_target_tl_flow)
- ターゲット TN フロー ビュー インスタンス (nvme_target_tn_flow)
- イニシエーター ITL フロー ビュー インスタンス (scsi_initiator_itl_flow)
- イニシエーター ITN フロー ビュー インスタンス (nvme_initiator_itn_flow)
- ターゲット ITL フロー ビュー インスタンス (scsi_target_itl_flow)
- ターゲット ITN フロー ビュー インスタンス (nvme_target_itn_flow)
- イニシエーター IO フロー ビュー インスタンス (scsi_initiator_io および nvme_initiator_io)
- ターゲット IO フロー ビュー インスタンス (scsi_target_io および nvme_target_io)

List of Supported Flow Metrics

ポート ビュー インスタンス (ポート)

表 18: ポート ビュー インスタンスのフローメトリック

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
scsi_target_count	stc	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部の固有の SCSI ターゲット FCID の数。
nvme_target_count	ntc	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部の固有の NVMe ターゲット FCID の数。
scsi_initiator_count	sic	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部のイニシエータの数。
nvme_initiator_count	nic	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部のイニシエータの数。
io_app_count	IOac	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの背後でホストされているアプリケーションの数。
logical_port_count	lpc	メタデータ	数	×	メトリックが最後にクリアされてから、IOを使用してスイッチポートに構成された VSAN の数。
scsi_target_app_count	stac	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションの数。
nvme_target_app_count	ntac	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションの数。
scsi_initiator_app_count	siac	メタデータ	数	×	スイッチポートの外部にあるイニシエータによってデータが要求されているアプリケーションの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
nvme_initiator_app_count	niac	メタデータ	数	×	スイッチポートの外部にあるイニシエータによってデータが要求されているアプリケーションの数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチポートと関連付けられている未処理の write コマンドカウントの数。
scsi_target_it_flow_count	stITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている IT フローの数。
nvme_target_it_flow_count	ntITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている IT フローの数。
scsi_initiator_it_flow_count	siITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている initiator-target (IT) フローの数。
nvme_initiator_it_flow_count	niITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている initiator-target (IT) フローの数。
scsi_target_itl_flow_count	stITLfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている ITL フローの数。
nvme_target_itn_flow_count	ntITNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている ITN フローの数。
scsi_initiator_itl_flow_count	siITLfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている ITL フローの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
nvme_initiator_itn_flow_count	niITNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている ITN フローの数。
scsi_target_tl_flow_count	sfTLfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている LUN の数。
nvme_target_tn_flow_count	ntTNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている ネームスペース ID の数。
total_abts_count	totAbts	メトリック	数	○	観測した中止の数。
total_read_io_count	rtIO	メトリック	数	○	スイッチポートの外部で観測した read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	スイッチポートの外部で観測した write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	スイッチポートの外部で観測した逐次の read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	スイッチポートの外部で観測した逐次の write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	スイッチポートの外部で確認された read の完了時間の累計。 この情報を使用して、読み取り IO の平均完了時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	<p>スイッチポートの外部で確認された write コマンドの完了時間の累計。</p> <p>この情報を使用して、write コマンドの平均完了時間を計算できます。</p>
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	<p>スイッチポートの外部で確認された read コマンドの開始時間の累計（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、data access latency と呼ばれることもあります。</p> <p>この情報を使用して、読み取りIOの平均開始時間を計算できます。</p>
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	<p>スイッチポートの外部で確認された write コマンドの開始時間の累計（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、data access latency と呼ばれることもあります。</p> <p>この情報を使用して、write コマンドの平均開始時間を計算できます。</p>
total_read_io_bytes	rtIOb	メトリック	バイト	○	<p>スイッチポートの外部で観測した read コマンドデータの合計。</p>

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_bytes	wtIOb	メトリック	バイト	○	スイッチポートの外部で観測した write コマンドデータの合計。
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部で確認された read コマンドのインターギャップ時間の累計。 この情報を使用して、読み取りIOの平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部で確認された write コマンドのインターギャップ時間の累計。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	スイッチポートの外部で観測した完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	スイッチポートの外部で観測した完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	スイッチポートの外部で観測した完了した read コマンドデータの合計（バイト）。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	スイッチポートの外部で観測した完了した write コマンドデータの合計（バイト）。
read_io_rate	rIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部で観察された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部で観察された read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部で観察された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部で観察された write コマンドのピークレート。
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部で観測した read コマンド帯域幅の合計。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部で観測したピーク read コマンド帯域幅の合計。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部で観測した write コマンド帯域幅の合計。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部で観測したピーク write コマンド帯域幅の合計。
read_io_size_min	rIOsMi	メトリック	バイト	○	スイッチポートの外部で観測した read コマンドサイズの最小値。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_size_max	rIOsMa	メトリック	バイト	○	スイッチポートの外部で観測した read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	スイッチポートの外部で観測した write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で観測した write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read コマンド完了時間の最大値。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部で観測した write コマンド完了時間の最小値。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部で観測した write コマンド完了時間の最大値。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最小の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最大の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最小の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最大の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続したIOコマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read コマンドインターギャップ時間の最大値。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部で観測した write コマンドインターギャップ時間の最小値。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部で観測した write コマンドインターギャップ時間の最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	スイッチポートの外部で観測した read コマンド中止の合計数。
write_io_aborts	wIOa	メトリック	数	○	スイッチポートの外部でホストされているアプリケーションの外部で確認された、中止された write コマンドの数。
read_io_failures	rIOf	メトリック	数	○	スイッチポートの外部で観測した read コマンド失敗の合計数。
write_io_failures	wIOf	メトリック	数	○	スイッチポートの外部で観測した write コマンド失敗の合計数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	スイッチポートの外部で確認された read コマンドのチェック条件の数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	スイッチポートの外部で確認された write コマンドのチェック条件の数。
read_io_scsi_busy_count	rIOsbc	メトリック	数	×	スイッチポートの外部で確認された read コマンドのビジー状態の数。
write_io_scsi_busy_count	wIOsbc	メトリック	数	×	スイッチポートの外部で確認された write コマンドのビジー状態の数。
read_io_nvme_lba_out_of_range_count	rIONLbaoort	メトリック	数	×	表示された read コマンド <i>lba out of range</i> エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaoort	メトリック	数	×	表示された write コマンド <i>lba out of range</i> エラーの数。
read_io_nvme_ns_not_ready_count	rIONnsnrc	メトリック	数	×	表示された read コマンドの <i>namespace not ready</i> エラーの数。
write_io_nvme_ns_not_ready_count	wIONnsnrc	メトリック	数	×	表示された write コマンドの <i>namespace not ready</i> エラーの数。
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	スイッチポートの外部で確認された read コマンドの予約競合の数。
read_io_nvme_reservation_conflict_count	rIONrect	メトリック	数	×	スイッチポートの外部で確認された read コマンドの予約競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	スイッチポートの外部で確認された write コマンドの予約競合の数。
write_io_nvme_reservation_conflict_count	wIONrect	メトリック	数	×	スイッチポートの外部で確認された write コマンドの予約競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	スイッチポートの外部で確認された read コマンドのキューが満杯状態の数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	スイッチポートの外部で確認された write コマンドのキューが満杯状態の数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。

論理ポートビューインスタンス (logical_port)

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

論理ポートビューインスタンス (logical_port)

表 19: 論理ポートビューインスタンスのフローメトリック

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN分析機能が有効になっているスイッチのポート。
vsan	vsan	キー	番号	×	メトリックの最後のクリア以降に IO を使用してスイッチポートに設定された VSAN。
scsi_target_count	stc	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部のターゲットの数。
nvme_target_count	ntc	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部のターゲットの数。
scsi_initiator_count	sic	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部のイニシエータの数。
nvme_initiator_count	nic	メタデータ	数	×	メトリックの最後のクリア以降の、IOのあるスイッチポートの外部のイニシエータの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
scsi_target_app_count	stac	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションの数。
nvme_target_app_count	ntac	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションの数。
scsi_initiator_app_count	siac	メタデータ	数	×	スイッチポートの外部にあるイニシエータによってデータが要求されているアプリケーションの数。
nvme_initiator_app_count	niac	メタデータ	数	×	スイッチポートの外部にあるイニシエータによってデータが要求されているアプリケーションの数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチポートと関連付けられている未処理の write コマンドカウントの数。
scsi_target_it_flow_count	stITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている IT フローの数。
nvme_target_it_flow_count	ntITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている IT フローの数。
scsi_initiator_it_flow_count	siITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている initiator-target (IT) フローの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
nvme_initiator_it_flow_count	niITfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている initiator-target (IT) フローの数。
scsi_target_itl_flow_count	stITLfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている ITL フローの数。
nvme_target_itn_flow_count	ntITNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている ITN フローの数。
scsi_initiator_itl_flow_count	siITLfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている ITL フローの数。
nvme_initiator_itn_flow_count	niITNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられている ITN フローの数。
scsi_target_tl_flow_count	stTLfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている LUN の数。
nvme_target_tn_flow_count	ntTNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられている ネームスペース ID の数。
total_abts_count	totAbts	メトリック	数	○	観測した中止の数。
total_read_io_count	rtIO	メトリック	数	○	スイッチポートの外部で観測した read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	スイッチポートの外部で観測した write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	スイッチポートの外部で観測した逐次な read コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_seq_write_io_count	wrstIOc	メトリック	数	×	スイッチポートの外部で観測した逐次の write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	<p>スイッチポートの外部で確認された read コマンドの完了時間の累計。</p> <p>この情報を使用して、読み取りIOの平均完了時間を計算できます。</p>
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	<p>スイッチポートの外部で確認された write コマンドの完了時間の累計。</p> <p>この情報を使用して、write コマンドの平均完了時間を計算できます。</p>
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	<p>スイッチポートの外部で確認された read コマンドの開始時間の累計（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、data access latency とも呼ばれることもあります。</p> <p>この情報を使用して、読み取りIOの平均開始時間を計算できます。</p>

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	スイッチポートの外部で確認された write コマンドの開始時間の累計（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。 この情報を使用して、writeコマンドの平均開始時間を計算できます。
total_read_io_bytes	rtIOb	メトリック	バイト	○	スイッチポートの外部で観測した read コマンドデータの合計。
total_write_io_bytes	wtIOb	メトリック	バイト	○	スイッチポートの外部で観測した write コマンドデータの合計。
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部で確認された read コマンドのインターギャップ時間の累計。 この情報を使用して、読み取りIOの平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部で確認された write コマンドのインターギャップ時間の累計。 この情報を使用して、writeコマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	スイッチポートの外部で観測した完了した read コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	スイッチポートの外部で観測した完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	スイッチポートの外部で観測した完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	スイッチポートの外部で観測した完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部で観察された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットで観察された、LUN の外部の read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部で観察された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットで観察された、LUN の外部の write コマンドのピークレート。
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部で観測した read コマンド帯域幅の合計。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ノート可能	説明
正式名称	回線のテキストラベル				
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	logical-unit-number (LUN) の外部、スイッチポートの外部のターゲットで観察された read コマンドのピーク帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部で観測した write コマンド帯域幅の合計。このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	LUN の外部、スイッチポートの外部のターゲットで観察された write コマンドのピーク帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	スイッチポートの外部で観測した read コマンドサイズの最小値。
read_io_size_max	rIOsMa	メトリック	バイト	○	スイッチポートの外部で観測した read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	スイッチポートの外部で観測した write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で観測した write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read-command-completion 時間の最大値。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最小の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEの txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最大の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEの txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最小の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEの txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	スイッチのポートの外部で確認された最大の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続したIOコマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部で観測した read コマンドインターギャップ時間の最大値。 read_io_inter_gap_time_max は連続したIOコマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部で観測した write コマンドインターギャップ時間の最小値。 write_io_inter_gap_time_min は連続したIOコマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部で観測した write コマンドインターギャップ時間の最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	スイッチポートの外部で観測した read コマンド中止の合計数。
write_io_aborts	wIOa	メトリック	数	○	スイッチポートの背後でホストされているアプリケーションの外部で確認された、中止された write コマンドの数。
read_io_failures	rIOf	メトリック	数	○	スイッチポートの外部で観測した read コマンド失敗の合計数。
write_io_failures	wIOf	メトリック	数	○	スイッチポートの外部で観測した write コマンド失敗の合計数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	スイッチポートの外部で確認された read コマンドのチェック条件の数。
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	スイッチポートの外部で確認された write コマンドのチェック条件の数。
read_io_scsi_busy_count	rIOsbc	メトリック	数	×	スイッチポートの外部で確認された read コマンドのビジー状態の数。
write_io_scsi_busy_count	wIOsbc	メトリック	数	×	スイッチポートの外部で確認された write コマンドのビジー状態の数。
read_io_nvme_lba_out_of_range_count	rIONLbaoort	メトリック	数	×	表示された read コマンド <i>lba</i> 範囲外エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaoort	メトリック	数	×	表示された write コマンド <i>lba</i> 範囲外エラーの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_nvme_ns_not_ready_count	rIONsnrc	メトリック	数	×	表示された read コマンド名前空間準備中エラーの数。
write_io_nvme_ns_not_ready_count	wIONsnrc	メトリック	数	×	表示された write コマンド名前空間準備中エラーの数。
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	スイッチポートの外部で確認された read コマンドの予約競合の数。
read_io_nvme_reservation_conflict_count	rIONrect	メトリック	数	×	スイッチポートの外部で確認された read コマンドの予約競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	スイッチポートの外部で確認された write コマンドの予約競合の数。
write_io_nvme_reservation_conflict_count	wIONrect	メトリック	数	×	スイッチポートの外部で確認された write コマンドの予約競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	スイッチポートの外部で確認された read コマンドのキューが満杯状態の数。
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	スイッチポートの外部で確認された write コマンドのキューが満杯状態の数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

アプリケーションビューインスタンス (app)

表 20: アプリケーションビューインスタンスのフローメトリック

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics機能が有効になっているスイッチのポート。
app_id	app_id	キー	数	×	スイッチポート外部のアプリケーションのアプリケーション識別子。
scsi_target_itl_flow_count	stITLfc	メタデータ	数	×	スイッチポートの外部のアプリケーションと関連付けられているターゲットITLフローの数。
nvme_target_itn_flow_count	ntITNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなターゲットと関連付けられているITNフローの数。
scsi_initiator_itl_flow_count	siITLfc	メタデータ	数	×	スイッチポートの外部のアプリケーションと関連付けられているイニシエータITLフローの数。
nvme_initiator_itn_flow_count	niITNfc	メタデータ	数	×	スイッチポートの外部にあるさまざまなイニシエータと関連付けられているITNフローの数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートの外部のアプリケーションと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチポートの外部のアプリケーションと関連付けられている未処理の write コマンドカウントの数。
scsi_target_app_count	stac	メタデータ	数	×	スイッチポートの外部のアプリケーションのデータをホストするターゲットの数。
nvme_target_app_count	ntac	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションの数。
scsi_initiator_app_count	siac	メタデータ	数	×	スイッチポートの外部のアプリケーションからのデータにアクセスするイニシエータの数。

フロー メトリック		タイプ	単位	ソート 可能	説明
正式名称	回線のテキ ストラベル				
nvme_initiator_app_count	niac	メタデー タ	数	×	スイッチ ポートの外部にあるイニシエータ によってデータが要求されているアプリケー ションの数。
scsi_target_tl_flow_count	stTLfc	メタデー タ	数	×	スイッチ ポートの外部のアプリケーション と関連付けられている LUN の数。
nvme_target_tn_flow_count	ntTNfc	メタデー タ	数	×	スイッチポートの外部にあるさまざまなター ゲットと関連付けられている ネームスパー ス ID の数。
sampling_end_time	samEtm	メトリッ ク	UNIX 時 間	×	サンプリング時間間隔の終了時間。
sampling_start_time	samStm	メトリッ ク	UNIX 時 間	×	サンプリング時間間隔の開始時間。

ターゲット ビュー インスタンス (scsi_target および nvme_target)

表 21: ターゲット ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソ ト 可 能	説明
正式名称	回線のテキ ストラベル				
port	port	キー	テキス ト	い い え	SAN Analytics 機能が有効になっ ているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされ てから、IO を使用してスイッチ ポートに構成された VSAN。
target_id	did	キー	テキス ト	い い え	メトリックが最後にクリアされ て以降、IO があるスイッチポ ートの外部にあるターゲット ファ イバチャネル 識別子。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
scsi_target_app_count	stac	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションの数。
nvme_target_app_count	ntac	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションの数。
scsi_target_lun_count	stLc	メタデータ	数	×	スイッチポートの外部のターゲット上に表示される LUN の数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートの外部にあるターゲットの関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチポートの外部にあるターゲットの関連付けられている未処理の write コマンドカウントの数。
scsi_target_entity_it_flow_count	stITfc	メタデータ	数	×	スイッチポートの外部にあるターゲットと関連付けられている ITL フローの数。
nvme_target_entity_it_flow_count	ntITfc	メタデータ	数	×	スイッチポートの外部にあるターゲットと関連付けられている ITL フローの数。
scsi_target_entity_itl_flow_count	stITLfc	メタデータ	数	×	スイッチポートの外部にあるターゲットと関連付けられている ITL フローの数。
nvme_target_entity_itn_flow_count	ntITNfc	メタデータ	数	×	スイッチポートの外部にあるターゲットと関連付けられている ITN フローの数。
total_abts_count	totAbts	メトリック	数	○	観測した中止の数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_count	rtIO	メトリック	数	○	スイッチポートの外部のターゲットの外部で見られた read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	スイッチポートの外部のターゲットの外部で見られた write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	スイッチポートの外部のターゲットの外部で見られたシーケンシャル read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	スイッチポートの外部のターゲットの外部で見られたシーケンシャル write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	スイッチポートの外部のターゲットの外部で見られた累積 read コマンド完了時間。 この情報を使用して、読み取りIOの平均完了時間を計算できます。
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	スイッチポートの外部のターゲットで見られた write コマンド完了時間の累積した合計。 この情報を使用して、 write コマンドの平均完了時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	累積合計 read コマンド開始時間 (IO コマンドとストレージからの最初の応答の間の時間ギャップ。最初の応答は、 READ コマンドの場合は最初のデータフレーム、 WRITE コマンドの場合は最初の txfr_rdy の場合がある) スイッチポートの外部のターゲットの外部で観測されます。開始時間は、 data access latency とも呼ばれることもあります。 この情報を使用して、読み取り IO の平均開始時間を計算できます。
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	累積合計 write コマンド開始時間 (時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります) スイッチポートの外部のターゲットの外部で観測されます。開始時間は、 data access latency とも呼ばれることもあります。 この情報を使用して、 write コマンドの平均開始時間を計算できます。
total_read_io_bytes	rtIOb	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で観測された read コマンドデータの合計。
total_write_io_bytes	wtIOb	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で見られた write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部のターゲットの外部で見られた累積 read コマンドインターギャップ時間。 この情報を使用して、読み取りIOの平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部のターゲットの外部で見られた累積 write コマンドインターギャップデータ時間。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	スイッチポートの外部のターゲットの外部で観測された、完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwrtIOc	メトリック	数	×	スイッチポートの外部のターゲットの外部で観測された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	スイッチポートの外部のターゲットの外部で観測された、完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwrtIOb	メトリック	数	×	スイッチポートの外部のターゲットの外部で観測された、完了した write コマンドデータの合計 (バイト)。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_rate	rIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部のターゲットの外部で観察された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットの外部で観察された read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部のターゲットの外部で観察された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットの外部で観察された write コマンドのピークレート。
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のターゲットの外部で観測された read コマンドの帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部のターゲットの外部で観測された read コマンドのピーク帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のターゲットの外部で観測された write コマンドの帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部のターゲットの外部で観察された write コマンドのピーク帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で観察される最小 read コマンドサイズ。
read_io_size_max	rIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で観察される最大 read コマンドサイズ。
write_io_size_min	wIOsMi	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で観察される最小 write コマンドサイズ。
write_io_size_max	wIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲットの外部で観測した write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最小 read コマンド完了時間。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最大 read コマンド完了時間。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最小 write コマンド完了時間。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最大 write コマンド完了時間。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のターゲットの外部で確認された最小の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のターゲットの外部で確認された最大の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のターゲットの外部で確認された最小の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のターゲットの外部で確認された最大の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最小 read コマンドインターギャップ時間。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最大 read コマンドインターギャップ時間。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最小 write コマンドインターギャップ時間。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ノート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲットの外部で観察される最大 write コマンドインターギャップ時間。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	スイッチポートの外部のターゲットの外部で観察された read コマンドの中止の数。
write_io_aborts	wIOa	メトリック	数	○	スイッチポートの外部のターゲットの外部で観察された write コマンドの中止の数。
read_io_failures	rIOf	メトリック	数	○	スイッチポートの外部のターゲットの外部で確認された read コマンドの失敗の数。
write_io_failures	wIOf	メトリック	数	○	スイッチポートの外部のターゲットの外部で観察された write コマンドの失敗の数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された read コマンドチェック条件の数。
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された write コマンドチェック条件の数。
read_io_scsi_busy_count	rIOsbc	メトリック	数	×	スイッチポート外部のターゲットの外部で確認された read コマンドのビジー状態の数。
write_io_scsi_busy_count	wIOsbc	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された write コマンド ビジー ステータスの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_nvme_lba_out_of_range_count	rIONLbaorct	メトリック	数	×	表示された read コマンド <i>lba out of range</i> エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaorct	メトリック	数	×	表示された write コマンド <i>lba out of range</i> エラーの数。
read_io_nvme_ns_not_ready_count	rIONNsnrc	メトリック	数	×	表示された read コマンドの <i>namespace not ready</i> エラーの数。
write_io_nvme_ns_not_ready_count	wIONNsnrc	メトリック	数	×	表示された write コマンドの <i>namespace not ready</i> エラーの数。
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された read コマンド予約の競合の数。
read_io_nvme_reservation_conflict_count	rIONrect	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された read コマンド予約の競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された write コマンド予約の競合の数。
write_io_nvme_reservation_conflict_count	wIONrect	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された write コマンド予約の競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された read コマンドキューのフルステータスの数。
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	スイッチポートの外部のターゲットの外部で確認された write コマンドキューのフルステータスの数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

イニシエータービューインスタンス (scsi_initiator および nvme_initiator)

表 22: イニシエータービューインスタンスのフローメトリック

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
initiator_id	sid	キー	テキスト	いいえ	IO トランザクションが確認されたスイッチのポートの背後に展開されているイニシエーターファイバチャンネルの識別子。
scsi_initiator_app_count	siac	メタデータ	数	×	スイッチポートの外部にあるイニシエーターでデータがホストされているアプリケーションの数。
nvme_initiator_app_count	niac	メタデータ	数	×	スイッチポートの外部にあるイニシエーターによってデータが要求されているアプリケーションの数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートの外部にあるイニシエーターの関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチポートの外部にあるイニシエーターの関連付けられている未処理の write コマンドカウントの数。
scsi_initiator_entity_it_flow_count	siITfc	メタデータ	数	×	スイッチポートの外部にあるイニシエーターと関連付けられている IT フローの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
nvme_initiator_entity_it_flow_count	niITfc	メタデータ	数	×	スイッチポートの外部にあるイニシエータと関連付けられている IT フローの数。
scsi_initiator_entity_itl_flow_count	siITLfc	メタデータ	数	×	スイッチポートの外部にあるイニシエータと関連付けられている ITL フローの数。
nvme_initiator_entity_itn_flow_count	niITNfc	メタデータ	数	×	スイッチポートの外部にあるイニシエータと関連付けられている ITN フローの数。
total_abts_count	totAbts	メトリック	数	○	観測した中止の数。
total_read_io_count	rtIO	メトリック	数	○	スイッチポートの外部のイニシエータの外部で見られた read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	スイッチポートの外部のイニシエータの外部で見られた write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	スイッチポートの外部のイニシエータの外部で見られたシーケンシャル read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	スイッチポートの外部のイニシエータの外部で見られたシーケンシャル write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	スイッチポートの外部のイニシエータの外部で見られた累積 read コマンド完了時間。 この情報を使用して、読み取り IO の平均完了時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	スイッチポートの外部のイニシエータの外部で見られた累積 write コマンド完了時間。 この情報を使用して、 write コマンドの平均完了時間を計算できます。
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	累積合計 read コマンド開始時間（IO コマンドとストレージからの最初の応答の間の時間ギャップ。最初の応答は、 READ コマンドの場合は最初のデータフレーム、 WRITE コマンドの場合は最初の txfr_rdy の場合がある）スイッチポートの外部のイニシエータの外部で観測されます。開始時間は、 data access latency と呼ばれることもあります。 この情報を使用して、読み取り IO の平均開始時間を計算できます。
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	累積合計 write コマンド開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）スイッチポートの外部のイニシエータの外部で観測されます。開始時間は、 data access latency と呼ばれることもあります。 この情報を使用して、 write コマンドの平均開始時間を計算できます。
total_read_io_bytes	rtIOb	メトリック	バイト	○	スイッチポートの外部のイニシエータの外部で見られた read コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_bytes	wtIOb	メトリック	バイト	○	スイッチポートの外部のインシエータの外部で見られた write コマンドデータの合計。
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部のインシエータの外部で見られた累積 read コマンドインターギャップ時間。 この情報を使用して、読み取り IO の平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部のインシエータの外部で見られた累積 write コマンドインターギャップ時間データ。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	スイッチポートの外部のインシエータの外部で見られた完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	スイッチポートの外部のインシエータの外部で見られた完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	スイッチポートの外部のインシエータの外部で見られた完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	スイッチポートの外部のインシエータの外部で見られた完了した write コマンドデータの合計 (バイト)。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_rate	rIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部のインシエータの外部で観察された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のインシエータの外部で観察された read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部のインシエータの外部で観察された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のインシエータの外部で観察された write コマンドのピークレート。
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のインシエータの外部で観測された read コマンドの帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部のインシエータの外部で観測されたピーク read コマンドの帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のインシエータの外部で観測された write コマンドの帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部のインシエータの外部で観測されたピーク write コマンドの帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	スイッチポートの外部のインシエータの外部で観察される最小 read コマンドサイズ。
read_io_size_max	rIOsMa	メトリック	バイト	○	スイッチポートの外部のインシエータの外部で観察される最大 read コマンドサイズ。
write_io_size_min	wIOsMi	メトリック	バイト	○	スイッチポートの外部のインシエータの外部で観察される最小 write コマンドサイズ。
write_io_size_max	wIOsMa	メトリック	バイト	○	スイッチポートの外部のインシエータの外部で観察される最大 write コマンドサイズ。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最小 read コマンド完了時間。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最大 read コマンド完了時間。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最小 write コマンド完了時間。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最大 write コマンド完了時間。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のイニシエータの外部で確認された最小の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEの txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のイニシエータの外部で確認された最大の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEの txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のイニシエータの外部で確認された最小の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEの txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のインシエータの外部で確認された最大の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEの txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最小 read コマンドインターギャップ時間。 read_io_inter_gap_time_min は連続したIOコマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最大 read コマンドインターギャップ時間。 read_io_inter_gap_time_max は連続したIOコマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最小 write コマンドインターギャップ時間。 write_io_inter_gap_time_min は連続したIOコマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のインシエータの外部で観察される最大 write コマンドインターギャップ時間。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	スイッチポートの外部のインシエータの外部で観察された read コマンドの中止の数。
write_io_aborts	wIOa	メトリック	数	○	スイッチポートの外部のインシエータの外部で観察された write コマンドの中止の数。
read_io_failures	rIOf	メトリック	数	○	スイッチポートの外部のインシエータの外部で観察された read コマンドの失敗の数。
write_io_failures	wIOf	メトリック	数	○	スイッチポートの外部のインシエータの外部で観察された write コマンドの失敗の数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	スイッチポートの外部のインシエータの外部で確認された read コマンドチェック条件の数。
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	スイッチポートの外部のインシエータの外部で確認された write コマンドチェック条件の数。
read_io_scsi_busy_count	rIOSbc	メトリック	数	×	スイッチポート外部のインシエータの外部で確認された read コマンドのビジー状態の数。
write_io_scsi_busy_count	wIOSbc	メトリック	数	×	スイッチポート外部のインシエータの外部で確認された write コマンドのビジー状態の数。

フローメトリック		タイプ	単位	ノート可能	説明
正式名称	回線のテキストラベル				
read_io_nvme_lba_out_of_range_count	rIONLbaorct	メトリック	数	×	表示された read コマンド <i>lba out of range</i> エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaorct	メトリック	数	×	表示された write コマンド <i>lba out of range</i> エラーの数。
read_io_nvme_ns_not_ready_count	rIONnsnrc	メトリック	数	×	表示された read コマンドの <i>namespace not ready</i> エラーの数。
write_io_nvme_ns_not_ready_count	wIONnsnrc	メトリック	数	×	表示された write コマンドの <i>namespace not ready</i> エラーの数。
read_io_scsi_reservation_conflict_count	rIOSreect	メトリック	数	×	スイッチポートの外部のイニシエータの外部で確認された read コマンド予約の競合の数。
read_io_nvme_reservation_conflict_count	rIONreect	メトリック	数	×	スイッチポートの外部のイニシエータの外部で確認された read コマンド予約の競合の数。
write_io_scsi_reservation_conflict_count	wIOSreect	メトリック	数	×	スイッチポートの外部のイニシエータの外部で確認された write コマンド予約の競合の数。
write_io_nvme_reservation_conflict_count	wIONreect	メトリック	数	×	スイッチポートの外部のイニシエータの外部で確認された write コマンド予約の競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	スイッチポートの外部のイニシエータの外部で確認された read コマンドキューのフルステータスの数。
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	スイッチポートの外部のイニシエータの外部で確認された write コマンドキューのフルステータスの数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

ターゲットアプリケーションビューインスタンス (scsi_target_app および nvme_target_app)

表 23: ターゲットアプリケーションビューインスタンスのフローメトリック

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	×	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。
target_id	did	キー	テキスト	×	メトリックが最後にクリアされて以降、IO があるスイッチポートの外部にあるターゲットファイバチャネル識別子。
scsi_target_entity_itl_flow_count	stITLfc	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションと関連付けられている ITL フローの数。
nvme_target_entity_itn_flow_count	ntITNfc	メタデータ	数	×	スイッチポートの外部にあるターゲットでデータがホストされているアプリケーションと関連付けられている ITN フローの数。
scsi_target_lun_count	stLc	メタデータ	数	×	スイッチポートの外部のターゲット上のアプリケーションの外部と見なされる LUN の数。
nvme_target_namespace_count	ntNc	メタデータ	数	×	スイッチポートの外部のターゲット上のアプリケーションの外部で認識される namespace ID の数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートの外部にあるターゲットの外部のアプリケーションと関連付けられている未処理の read コマンドカウントの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
active_io_write_count	waIO	メタデータ	数	○	スイッチポートの外部にあるターゲットの外部のアプリケーションと関連付けられている未処理の write コマンドカウントの数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。

イニシエーターアプリケーションビューインスタンス (scsi_initiator_app および nvme_initiator_app)

表 24: イニシエーターアプリケーションビューインスタンスのフローメトリック

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	×	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IOを使用してスイッチポートに構成された VSAN。
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。
initiator_id	sid	キー	テキスト	×	IO トランザクションが確認されたスイッチのポートの背後に展開されている Initiator Fibre Channel の ID。
scsi_initiator_entity_itl_flow_count	siITLfc	メタデータ	数	×	スイッチのポートの外部にあるイニシエーターによってデータがアクセスされているアプリケーションと関連付けられている ITL フローの数。

ターゲット IT フロー ビュー インスタンス (scsi_target_it_flow および nvme_target_it_flow)

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
nvme_initiator_entity_itn_flow_count	niITNfc	メタデータ	数	×	スイッチのポートの外部にあるイニシエータによってデータがアクセスされているアプリケーションと関連付けられている ITN フローの数。
active_io_read_count	raIO	メタデータ	数	○	スイッチのポートの外部にあるイニシエータによってデータがアクセスされているアプリケーションと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチのポートの外部にあるイニシエータによってデータがアクセスされているアプリケーションと関連付けられている未処理の write コマンドカウントの数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。

ターゲット IT フロー ビュー インスタンス (scsi_target_it_flow および nvme_target_it_flow)

表 25: ターゲット IT フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	×	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
target_id	did	キー	テキスト	いいえ	前回のメトリックのクリア以降の IO を持つスイッチポートの外部の Target Fibre Channel ID。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
initiator_id	sid	キー	テキスト	×	スイッチのポートの背後で展開されているイニシエータで IO トランザクションが実行されている Initiator Fibre Channel の ID。
active_io_read_count	raIO	メタデータ	数	○	target-IT-flow レコードと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	target-IT-flow レコードと関連付けられている未処理の write コマンドカウントの数。
scsi_target_entity_itl_flow_count	stITLfc	メタデータ	数	×	target-IT-flow レコードと関連付けられている ITL フローの数。
nvme_target_entity_itn_flow_count	ntITNfc	メタデータ	数	×	target-IT-flow レコードと関連付けられている ITN フローの数。
total_abts_count	totAbts	メトリック	数	○	観測した中止の数。
total_read_io_count	rtIO	メトリック	数	○	target-IT-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	target-IT-flow レコードの外部で確認された write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	target-IT-flow レコードの外部で確認された逐次な read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	target-IT-flow レコードの外部で確認された逐次な write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	target-IT-flow レコードの外部で確認された read コマンドの完了時間の累計。 この情報を使用して、読み取り IO の平均完了時間を計算できます。
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	target-IT-flow レコードの外部で確認された write コマンドの完了時間の累計。 この情報を使用して、 write コマンドの平均完了時間を計算できます。
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	target-IT-flow レコードの外部で確認された read コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。 この情報を使用して、読み取り IO の平均開始時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	target-IT-flow レコードの外部で確認された write コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。 この情報を使用して、 write コマンドの平均開始時間を計算できます。
total_read_io_bytes	rtIOb	メトリック	バイト	○	target-IT-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_bytes	wtIOb	メトリック	バイト	○	target-IT-flow レコードの外部で確認された write コマンドデータの合計。
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	target-IT-flow レコードの外部で確認された read コマンドのインターギャップ時間の累計。 この情報を使用して、読み取り IO の平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	target-IT-flow レコードの外部で確認された write コマンドのインターギャップ時間データの累計。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	target-IT-flow レコードの外部で確認された完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	target-IT-flow レコードの外部で確認された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	target-IT-flow レコードの外部で確認された完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	target-IT-flow レコードの外部で確認された完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1 秒あたりの IO 数	○	target-IT-flow レコード上の外部で確認された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1 秒あたりの IO 数	×	target-IT-flow レコード上の外部で確認された read コマンドのピーク レート。
write_io_rate	wIOr	メトリック	1 秒あたりの IO 数	○	target-IT-flow レコード上の外部で確認された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1 秒あたりの IO 数	×	target-IT-flow レコード上の外部で確認されたピーク write コマンドのレート。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	target-IT-flow レコードの外部で確認された read コマンド帯域幅。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	target-IT-flow レコードの外部で確認されたピーク read コマンド帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	target-IT-flow レコードの外部で確認された write コマンド帯域幅。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	target-IT-flow レコードの外部で確認されたピーク write コマンド帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	target-IT-flow レコードの外部で確認された read コマンドサイズの最小値。
read_io_size_max	rIOsMa	メトリック	バイト	○	target-IT-flow レコードの外部で確認された read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	target-IT-flow レコードの外部で確認された write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	target-IT-flow レコードの外部で確認された write コマンドサイズの最大値。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された read コマンド完了時間の最大値。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された read コマンド完了時間の最大値。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された read コマンド完了時間の最大値。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最小の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最大の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最小の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最大の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された write コマンドインターギャップ時間の最小値。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	target-IT-flow レコードの外部で確認された write コマンドインターギャップ時間の最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、中止された read コマンドの数。
write_io_aborts	wIOa	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、中止された write コマンドの数。
read_io_failures	rIOf	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、失敗した read コマンドの数。
write_io_failures	wIOf	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、失敗した write コマンドの数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	target-IT-flow レコードの外部で確認された read コマンドのチェック条件の数。
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	target-IT-flow レコードの外部で確認された write コマンドのチェック条件の数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_scsi_busy_count	rIOsbc	メトリック	数	×	target-IT-flow レコードの外部で確認された read コマンドのビジーステータスの数。
write_io_scsi_busy_count	wIOsbc	メトリック	数	×	target-IT-flow レコードの外部で確認された write コマンドのビジーステータスの数。
read_io_nvme_lba_out_of_range_count	rIONLbaorct	メトリック	数	×	表示された read コマンド <i>lba out of range</i> エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaorct	メトリック	数	×	表示された write コマンド <i>lba out of range</i> エラーの数。
read_io_nvme_ns_not_ready_count	rIONNsnrc	メトリック	数	×	表示された read コマンドの <i>namespace not ready</i> エラーの数。
write_io_nvme_ns_not_ready_count	wIONNsnrc	メトリック	数	×	表示された write コマンドの <i>namespace not ready</i> エラーの数。
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	target-IT-flow レコードの外部で確認された read コマンドの予約競合の数。
read_io_nvme_reservation_conflict_count	rIONrect	メトリック	数	×	target-IT-flow レコードの外部で確認された read コマンドの予約競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	target-IT-flow レコードの外部で確認された write コマンドの予約競合の数。
write_io_nvme_reservation_conflict_count	wIONrect	メトリック	数	×	target-IT-flow レコードの外部で確認された write コマンドの予約競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	target-IT-flow レコードの外部で確認された read コマンドキューのフルステータスの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	target-IT-flow レコードの外部で確認された write コマンドキューのフルステータスの数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビュー インスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

イニシエーター IT フロー ビュー インスタンス (scsi_initiator_it_flow および nvme_initiator_it_flow)

表 26: イニシエーター IT フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
initiator_id	sid	キー	テキスト	いいえ	IO トランザクションが確認されたスイッチのポートの背後に展開されている Initiator Fibre Channel の ID。
target_id	did	キー	テキスト	いいえ	スイッチポートの外部にあるイニシエーターによって開始されている IO トランザクションを実行しているターゲットファイバチャネルの識別子。
active_io_read_count	raIO	メタデータ	数	○	initiator-IT-flow レコードと関連付けられている未処理の read コマンドカウントの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
active_io_write_count	waIO	メタデータ	数	○	initiator-IT-flow レコードと関連付けられている未処理の write コマンドカウントの数。
scsi_initiator_entity_itl_flow_count	siITLfc	メタデータ	数	×	initiator-IT-flow レコードと関連付けられている ITL フローの数。
nvme_initiator_entity_itn_flow_count	niITNfc	メタデータ	数	×	initiator-IT-flow レコードと関連付けられている ITN フローの数。
total_abts_count	totAbts	メトリック	数	○	観測した中止の数。
total_read_io_count	rtIO	メトリック	数	○	initiator-IT-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	initiator-IT-flow レコードとの外部で確認された write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	initiator-IT-flow レコードの外部で確認された逐次の read コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	initiator-IT-flow レコードの外部で確認された read コマンドの完了時間の累計。 この情報を使用して、読み取り IO の平均完了時間を計算できません。
total_seq_write_io_count	wrstIOc	メトリック	数	×	initiator-IT-flow レコードの外部で確認された逐次の write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	<p>initiator-IT-flow レコードの外部で確認された write コマンドの完了時間の累計。</p> <p>この情報を使用して、write コマンドの平均完了時間を計算できます。</p>
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	<p>initiator-IT-flow レコードの外部で確認された read コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、data access latency と呼ばれることもあります。</p> <p>この情報を使用して、読み取り IO の平均開始時間を計算できます。</p>
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	<p>initiator-IT-flow レコードの外部で確認された write コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、data access latency と呼ばれることもあります。</p> <p>この情報を使用して、write コマンドの平均開始時間を計算できます。</p>
total_read_io_bytes	rtIOb	メトリック	バイト	○	<p>initiator-IT-flow レコードの外部で確認された read コマンドデータの合計。</p>

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_bytes	wtIOb	メトリック	バイト	○	initiator-IT-flow レコードとの外部で確認された write コマンドデータの合計。
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	initiator-IT-flow レコードの外部で確認された read コマンドのインターギャップ時間の累計。 この情報を使用して、読み取りIOの平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	initiator-IT-flow レコードの外部で確認された write コマンドのインターギャップ時間データの累計。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	initiator-IT-flow レコードの外部で確認された完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	initiator-IT-flow レコードの外部で確認された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	initiator-IT-flow レコードの外部で確認された完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	initiator-IT-flow レコードの外部で確認された完了した write コマンドデータの合計 (バイト)。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_rate	rIOr	メトリック	1秒あたりのIO数	○	initiator-IT-flow レコード上の外部で確認された read コマンドのレート。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	initiator-IT-flow レコードの外部で観察される read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	initiator-IT-flow レコード上の外部で確認された write コマンドのレート。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	initiator-IT-flow レコード上の外部で確認された write コマンドのピークレート。
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	initiator-IT-flow レコードの外部で確認された read コマンド帯域幅。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	initiator-IT-flow レコードの外部で確認されたピーク read コマンド帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	initiator-IT-flow レコードの外部で確認された write コマンド帯域幅。 このメトリックは、NPU から4秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	initiator-IT-flow レコードの外部で確認されたピーク write コマンド帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	initiator-IT-flow レコードの外部で確認された read コマンドサイズの最小値。
read_io_size_max	rIOsMa	メトリック	バイト	○	initiator-IT-flow レコードの外部で確認された read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	initiator-IT-flow レコードの外部で確認された write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	initiator-IT-flow レコードの外部で確認された write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された read コマンド完了時間の最大値。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された write コマンド完了時間の最小値。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された write コマンド完了時間の最大値。

フローメトリック		タイプ	単位	ソース可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最小の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最大の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最小の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソース可能	説明
正式名称	回線のテキストラベル				
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された最大の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された read コマンドインターギャップの最大値。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された write コマンドインターギャップの最小値。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	initiator-IT-flow レコードの外部で確認された write コマンドインターギャップの最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、中止された read コマンドの数。
write_io_aborts	wIOa	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、中止された write コマンドの数。
read_io_failures	rIOf	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、失敗した read コマンドの数。
write_io_failures	wIOf	メトリック	数	○	initiator-IT-flow レコードの外部で確認された、失敗した write コマンドの数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	initiator-IT-flow レコードの外部で確認された read コマンドのチェック条件の数
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	initiator-IT-flow レコードの外部で確認された write コマンドのチェック条件の数
read_io_scsi_busy_count	rIOsbc	メトリック	数	×	initiator-IT-flow レコードの外部で確認された read コマンドのビジーステータスの数
write_io_scsi_busy_count	wIOsbc	メトリック	数	×	initiator-IT-flow レコードの外部で確認された write コマンドビジーステータスの数
read_io_nvme_lba_out_of_range_count	rIONLbaoorct	メトリック	数	×	表示された read コマンド <i>lba out of range</i> エラーの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_nvme_lba_out_of_range_count	wIONLbaorct	メトリック	数	×	表示された write コマンド <i>lba out of range</i> エラーの数。
read_io_nvme_ns_not_ready_count	rIONNsnrc	メトリック	数	×	表示された read コマンドの <i>namespace not ready</i> エラーの数。
write_io_nvme_ns_not_ready_count	wIONNsnrc	メトリック	数	×	表示された write コマンドの <i>namespace not ready</i> エラーの数。
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	initiator-IT-flow レコードの外部で確認された read コマンドの予約競合の数。
read_io_nvme_reservation_conflict_count	rIONrect	メトリック	数	×	initiator-IT-flow レコードの外部で確認された read コマンドの予約競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	initiator-IT-flow レコードの外部で確認された write コマンドの予約競合の数。
write_io_nvme_reservation_conflict_count	wIONrect	メトリック	数	×	initiator-IT-flow レコードの外部で確認された write コマンドの予約競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	initiator-IT-flow レコードの外部で確認された read コマンドキューのフルステータスの数
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	initiator-IT-flow レコードの外部で確認された write コマンドキューのフルステータスの数
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

ターゲット TL フロー ビュー インスタンス (scsi_target_tl_flow)



(注) ターゲット TL フロー ビュー インスタンスのフロー メトリックは、SCSI 分析タイプにのみ適用されます。

表 27: ターゲット TL フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
target_id	did	キー	テキスト	いいえ	メトリックが最後にクリアされて以降、IO があるスイッチポートの外部にあるターゲットファイバチャネル 識別子。
lun	lun	キー	単位	×	IO が実行されるターゲットと関連付けられている論理ユニット番号 (LUN) 。
scsi_target_entity_itl_flow_count	stTLfc	メタデータ	数	×	スイッチポートの外部にあるターゲット上の LUN と関連付けられている ITL フローの数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートの外部にあるターゲット上の LUN と関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチポートの外部にあるターゲット上の LUN と関連付けられている未処理の write コマンドカウントの数。
total_read_io_count	rtIO	メトリック	数	○	スイッチポートの外部のターゲットで観察された、LUN の外部の read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	スイッチポートの外部のターゲットで観察された、LUN の外部の write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_seq_read_io_count	rstIOc	メトリック	数	×	スイッチポートの外部のターゲットで観察された、LUNの外部の逐次な read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	スイッチポートの外部のターゲットで観察された、LUNの外部の逐次な write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	<p>スイッチポートの外部のターゲットで観察された、LUNの外部の read コマンド完了時間の累積した合計。</p> <p>この情報を使用して、読み取りIOの平均完了時間を計算できます。</p>
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	<p>スイッチポートの外部のターゲットで観察された、LUNの外部の write コマンド完了時間の累積した合計。</p> <p>この情報を使用して、write コマンドの平均完了時間を計算できます。</p>

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	<p>累積合計 read コマンド開始時間（IO コマンドとストレージからの最初の応答の間の時間ギャップ。最初の応答は、READ コマンドの場合は最初のデータフレーム、WRITE コマンドの場合は最初の txfr_rdy の場合がある）スイッチポートの外部のターゲット上の LUN の外部で観測されます。開始時間は、data access latency と呼ばれることもあります。</p> <p>この情報を使用して、読み取り IO の平均開始時間を計算できます。</p>
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	<p>累積合計 write コマンド開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）スイッチポートの外部のターゲット上の LUN の外部で観測されます。開始時間は、data access latency と呼ばれることもあります。</p> <p>この情報を使用して、write コマンドの平均開始時間を計算できます。</p>
total_read_io_bytes	rtIOb	メトリック	バイト	○	<p>スイッチポートの外部のターゲット上の LUN の外部で観測された read コマンドデータの合計。</p>
total_write_io_bytes	wtIOb	メトリック	バイト	○	<p>スイッチポートの外部のターゲットで観察された、LUN の外部の write コマンドデータの合計。</p>

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部のターゲット上の LUN の外部で見られた累積 read コマンドインターギャップ時間。 この情報を使用して、読み取り IO の平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	スイッチポートの外部のターゲット上の LUN の外部で見られた累積 write コマンドインターギャップデータ時間。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で観測された、完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwrtIOc	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で観測された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で観測された、完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwrtIOb	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で観測された、完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1 秒あたりの IO 数	○	スイッチポートの外部のターゲット上の LUN の外部で観測された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットで観察された、LUNの外部の read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部のターゲットで観察された、LUNの外部の write コマンドのレート。 このメトリックは、NPUから4秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットで観察された、LUNの外部の write コマンドのピークレート。
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のターゲット上のLUNの外部で観測された read コマンドの帯域幅。 このメトリックは、NPUから4秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部のターゲット上のLUNの外部で観測された read コマンドのピーク帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のターゲット上のLUNの外部で観測された write コマンドの帯域幅。 このメトリックは、NPUから4秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	LUNの外部、スイッチポートの外部のターゲットで観察された write コマンドのピーク帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最小 read コマンドサイズ。

フローメトリック		タイプ	単位	ソース可能	説明
正式名称	回線のテキストラベル				
read_io_size_max	rIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最大 read コマンドサイズ。
write_io_size_min	wIOsMi	メトリック	バイト	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最小 write コマンドサイズ。
write_io_size_max	wIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲット上の LUN の外部で観測した write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最小 read コマンド完了時間。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最大 read コマンド完了時間。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最小 write コマンド完了時間。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最大 write コマンド完了時間。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上の LUN の外部で確認された最小の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上の LUN の外部で確認された最大の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上の LUN の外部で確認された最小の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上の LUN の外部で確認された最大の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。

フローメトリック		タイプ	単位	ソース可能	説明
正式名称	回線のテキストラベル				
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最小 read コマンドインターギャップ時間。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最大 read コマンドインターギャップ時間。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最小 write コマンドインターギャップ時間。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上の LUN の外部で観察される最大 write コマンドインターギャップ時間。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	スイッチポートの外部のターゲット上の LUN の外部で観察された read コマンドの中止の数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_aborts	wIOa	メトリック	数	○	スイッチポートの外部のターゲット上の LUN の外部で観察された write コマンドの中止の数。
read_io_failures	rIOf	メトリック	数	○	スイッチポートの外部のターゲット上の LUN の外部で観察された read コマンドの失敗の数。
write_io_failures	wIOf	メトリック	数	○	スイッチポートの外部のターゲット上の LUN の外部で観察された write コマンドの失敗の数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で確認された read コマンドチェック条件の数。
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で確認された write コマンドチェック条件の数。
read_io_scsi_busy_count	rIOSbc	メトリック	数	×	スイッチポート外部のターゲット上の LUN の外部で確認された read コマンドのビジー状態の数。
write_io_scsi_busy_count	wIOSbc	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で確認された write コマンドビジーステータスの数。
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で確認された read コマンド予約の競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で確認された write コマンド予約の競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で確認された read コマンドキューのフルステータスの数。

フローメトリック		タイプ	単位	ソース可能	説明
正式名称	回線のテキストラベル				
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	スイッチポートの外部のターゲット上の LUN の外部で確認された write コマンドキューのフルステータスの数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。

ターゲット TN フロー ビュー インスタンス (nvme_target_tn_flow)

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

ターゲット TN フロー ビュー インスタンス (nvme_target_tn_flow)



(注) ターゲット TN フロー ビュー インスタンスのフロー メトリックは、NVMe 分析タイプにのみ適用されます。

表 28: ターゲット TN フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
target_id	did	キー	テキスト	いいえ	メトリックが最後にクリアされて以降、IO があるスイッチポートの外部にあるターゲットファイバチャネル識別子。
connection_id	ci	キー	数	×	メトリックが最後にクリアされて以降、IO を備えたスイッチポートの外部にある NVMe 接続識別子。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
namespace_id	ni	キー	数	×	名前空間識別子は、名前空間の NVMe コントローラの一意の識別子であり、1～255 の値に設定できます。これは、SCSI の論理ユニット番号 (LUN) に似ています。
nvme_target_entity_itn_flow_count	ntfITNfc	メタデータ	数	×	スイッチポートの外部のターゲット上の名前空間識別子に関連付けられた ITN フローの数。
active_io_read_count	raIO	メタデータ	数	○	スイッチポートの外部にあるターゲット上の LUN と関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	スイッチポートの外部にあるターゲット上の LUN と関連付けられている未処理の write コマンドカウントの数。
total_read_io_count	rtIO	メトリック	数	○	スイッチポートの外部のターゲットで観察された、LUN の外部の read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	スイッチポートの外部のターゲットで観察された、LUN の外部の write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	スイッチポートの外部のターゲットで観察された、LUN の外部の逐次な read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	スイッチポートの外部のターゲットで観察された、LUN の外部の逐次な write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	<p>スイッチポートの外部のターゲットで観察された、LUNの外部の read コマンド完了時間の累積した合計。</p> <p>この情報を使用して、読み取りIOの平均完了時間を計算できます。</p>
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	<p>スイッチポートの外部のターゲットで観察された、LUNの外部の write コマンド完了時間の累積した合計。</p> <p>この情報を使用して、write コマンドの平均完了時間を計算できます。</p>
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	<p>累積合計 read コマンド開始時間 (IOコマンドとストレージからの最初の応答の間の時間ギャップ。最初の応答は、READ コマンドの場合は最初のデータフレーム、WRITE コマンドの場合は最初の txfr_rdy の場合がある) スwitchポートの外部のターゲット上のLUNの外部で観測されます。開始時間は、data access latency と呼ばれることもあります。</p> <p>この情報を使用して、読み取りIOの平均開始時間を計算できます。</p>

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	<p>累積合計 write コマンド開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、またはWRITE のtxfr_rdyである可能性があります）スイッチポートの外部のターゲット上のLUNの外部で観測されます。開始時間は、data access latency とも呼ばれることもあります。</p> <p>この情報を使用して、write コマンドの平均開始時間を計算できます。</p>
total_read_io_bytes	rtIOb	メトリック	バイト	○	<p>スイッチポートの外部のターゲット上のLUNの外部で観測された read コマンドデータの合計。</p>
total_write_io_bytes	wtIOb	メトリック	バイト	○	<p>スイッチポートの外部のターゲットで観察された、LUNの外部の write コマンドデータの合計。</p>
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	<p>スイッチポートの外部のターゲット上のLUNの外部で見られた累積 read コマンドインターギャップ時間。</p> <p>この情報を使用して、読み取りIOの平均インターギャップ時間を計算できます。</p>
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	<p>スイッチポートの外部のターゲット上のLUNの外部で見られた累積 write コマンドインターギャップデータ時間。</p> <p>この情報を使用して、write コマンドの平均インターギャップ時間を計算できます。</p>

フローメトリック		タイプ	単位	ノート可能	説明
正式名称	回線のテキストラベル				
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	スイッチポートの外部のターゲット上のLUNの外部で観測された、完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	スイッチポートの外部のターゲット上のLUNの外部で観測された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	スイッチポートの外部のターゲット上のLUNの外部で観測された、完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	スイッチポートの外部のターゲット上のLUNの外部で観測された、完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部のターゲットで観測された、LUNの外部の read コマンドのレート。 このメトリックは、NPUから4秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットで観測された、LUNの外部の read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	スイッチポートの外部のターゲットで観測された、LUNの外部の write コマンドの読み取り。 このメトリックは、NPUから4秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	スイッチポートの外部のターゲットで観測された、LUNの外部の write コマンドのピークレート。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のターゲット上のLUNの外部で観測された read コマンドの帯域幅。 このメトリックは、NPUから4秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	スイッチポートの外部のターゲット上のLUNの外部で観測された read コマンドのピーク帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	スイッチポートの外部のターゲット上のLUNの外部で観測された write コマンドの帯域幅。 このメトリックは、NPUから4秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	LUNの外部、スイッチポートの外部のターゲットで観測された write コマンドのピーク帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	スイッチポートの外部のターゲット上のLUNの外部で観測される最小 read コマンドサイズ。
read_io_size_max	rIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲット上のLUNの外部で観測される最大 read コマンドサイズ。
write_io_size_min	wIOsMi	メトリック	バイト	○	スイッチポートの外部のターゲット上のLUNの外部で観測される最小 write コマンドサイズ。
write_io_size_max	wIOsMa	メトリック	バイト	○	スイッチポートの外部のターゲット上のLUNの外部で観測した write コマンドサイズの最大値。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最小 read コマンド完了時間。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最大 read コマンド完了時間。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最小 write コマンド完了時間。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最大 write コマンド完了時間。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上のLUNの外部で確認された最小の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency と呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上のLUNの外部で確認された最大の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上のLUNの外部で確認された最小の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	スイッチポート外部のターゲット上のLUNの外部で確認された最大の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READの最初のデータフレーム、またはWRITEのtxfr_rdyである可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最小 read コマンドインターギャップ時間。 read_io_inter_gap_time_min は連続したIOコマンド間の期間で、マイクロ秒の1/256で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最大 read コマンドインターギャップ時間。 read_io_inter_gap_time_max は連続したIOコマンド間の期間で、マイクロ秒の1/256で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最小 write コマンドインターギャップ時間。 write_io_inter_gap_time_min は連続したIOコマンド間の期間で、マイクロ秒の1/256で測定されます。
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	スイッチポートの外部のターゲット上のLUNの外部で観察される最大 write コマンドインターギャップ時間。 write_io_inter_gap_time_max は連続したIOコマンド間の期間で、マイクロ秒の1/256で測定されます。
read_io_aborts	rIOa	メトリック	数	○	スイッチポートの外部のターゲット上のLUNの外部で観察された read コマンドの中止の数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_aborts	wIOa	メトリック	数	○	スイッチポートの外部のターゲット上のLUNの外部で観察された write コマンドの中止の数。
read_io_failures	rIOf	メトリック	数	○	スイッチポートの外部のターゲット上のLUNの外部で観察された read コマンドの失敗の数。
write_io_failures	wIOf	メトリック	数	○	スイッチポートの外部のターゲット上のLUNの外部で観察された write コマンドの失敗の数。
read_io_nvme_lba_out_of_range_count	rIONLbaoort	メトリック	数	×	表示された read コマンド <i>lba out of range</i> エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaoort	メトリック	数	×	表示された write コマンド <i>lba out of range</i> エラーの数。
read_io_nvme_ns_not_ready_count	rIONNsnrc	メトリック	数	×	表示された read コマンドの <i>namespace not ready</i> エラーの数。
write_io_nvme_ns_not_ready_count	wIONNsnrc	メトリック	数	×	表示された write コマンドの <i>namespace not ready</i> エラーの数。
read_io_nvme_reservation_conflict_count	rIONrecct	メトリック	数	×	スイッチポートの外部のターゲット上の名前空間識別子の外部で確認された read コマンド予約の競合の数。
write_io_nvme_reservation_conflict_count	wIONrecct	メトリック	数	×	スイッチポートの外部のターゲット上の名前空間識別子の外部で確認された write コマンド予約の競合の数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチ ポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチ ポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチ ポートの外部で確認された write コマンド ホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチ ポートの外部で確認された write コマンド シーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド ホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド ホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチ ポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチ ポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

イニシエーター ITL フロー ビュー インスタンス (scsi_initiator_itl_flow)



(注) イニシエーター *ITL* フロー ビュー インスタンスのフローメトリックは、SCSI 分析タイプにのみ適用されます。

表 29: イニシエータ *ITL* フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。
initiator_id	sid	キー	テキスト	いいえ	IO トランザクションが確認されたスイッチのポートの背後に展開されているイニシエータ ファイバチャネルの識別子。
target_id	did	キー	テキスト	いいえ	スイッチ ポートの外部にあるイニシエータによって開始された IO トランザクションを実行しているターゲットファイバチャネルの識別子。
lun	lun	キー	数	×	IO が実行されるイニシエータと関連付けられている論理ユニット番号 (LUN)。
active_io_read_count	raIO	メタデータ	数	○	initiator-ITL-flow レコードと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	initiator-ITL-flow レコードと関連付けられている未処理の write コマンドカウントの数。
total_read_io_count	rtIO	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された read コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_count	wtIO	メトリック	数	○	initiator-ITL-flow レコードとの外部で確認された write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された逐次な read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された逐次な write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	initiator-ITL-flow レコードの外部で確認された read コマンドの完了時間の累計。 この情報を使用して、読み取り IO の平均完了時間を計算できます。
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	initiator-ITL-flow レコードの外部で確認された write コマンドの完了時間の累計。 この情報を使用して、 write コマンドの平均完了時間を計算できます。
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	initiator-ITL-flow レコードの外部で確認された read コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。 この情報を使用して、読み取り IO の平均開始時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	<p>initiator-ITL-flow レコードの外部で確認された write コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、data access latency とも呼ばれることもあります。</p> <p>この情報を使用して、write コマンドの平均開始時間を計算できます。</p>
total_read_io_bytes	rtIOb	メトリック	バイト	○	<p>initiator-ITL-flow レコードの外部で確認された read コマンドデータの合計。</p>
total_write_io_bytes	wtIOb	メトリック	バイト	○	<p>initiator-ITL-flow レコードと外部で確認された write コマンドデータの合計。</p>
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	<p>initiator-ITL-flow レコードの外部で確認された read コマンドのインターギャップ時間の累計。</p> <p>この情報を使用して、読み取り IO の平均インターギャップ時間を計算できます。</p>
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	<p>initiator-ITL-flow レコードの外部で確認された write コマンドのインターギャップ時間データの累計。</p> <p>この情報を使用して、write コマンドの平均インターギャップ時間を計算できます。</p>
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	<p>initiator-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計。</p>

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1 秒あたりの IO 数	○	initiator-ITL-flow レコード上の外部で確認された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1 秒あたりの IO 数	×	initiator-ITL-flow レコード上の外部で確認された read コマンドのピーク レート。
write_io_rate	wIOr	メトリック	1 秒あたりの IO 数	○	initiator-ITL-flow レコード上の外部で確認された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1 秒あたりの IO 数	×	initiator-ITL-flow レコード上の外部で確認された write コマンドのピーク レート。
read_io_bandwidth	rIObw	メトリック	1 秒あたりの バイト数	○	initiator-ITL-flow レコードの外部で確認された read コマンド帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	initiator-ITL-flow レコードの外部で確認されたピーク read コマンド帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	initiator-ITL-flow レコードの外部で確認された write コマンド帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	initiator-ITL-flow レコードの外部で確認されたピーク write コマンド帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された read コマンドサイズの最小値。
read_io_size_max	rIOsMa	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンド完了時間の最大値。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンド完了時間の最小値。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンド完了時間の最大値。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソース ト 可 能	説明
正式名称	回線のテキスト ストラベル				
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンドインターギャップの最大値。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンドインターギャップの最小値。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンドインターギャップの最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された read コマンドの数。
write_io_aborts	wIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された write コマンドの数。
read_io_failures	rIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した read コマンドの数。
write_io_failures	wIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した write コマンドの数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された read コマンドのチェック条件の数
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された write コマンドのチェック条件の数
read_io_scsi_busy_count	rIOsbc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された read コマンドのビジーステータスの数
write_io_scsi_busy_count	wIOsbc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された write コマンドビジーステータスの数

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された read コマンドの予約競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された write コマンドの予約競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された read コマンドキューのフルステータスの数
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された write コマンドキューのフルステータスの数
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。

■ イニシエータ ITN フロー ビュー インスタンス (nvme_initiator_itn_flow)

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド ホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド ホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチ ポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチ ポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

イニシエータ ITN フロー ビュー インスタンス (nvme_initiator_itn_flow)



(注) イニシエータ ITN フロー ビュー インスタンスのフロー メトリックは、NVMe 分析タイプにのみ適用されます。

表 30: イニシエータ ITN フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチ ポートに構成された VSAN。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。
initiator_id	sid	キー	テキスト	いいえ	IO トランザクションが確認されたスイッチのポートの背後に展開されているイニシエータファイバチャンネルの識別子。
connection_id	ci	キー	数	×	メトリックが最後にクリアされて以降、IO を備えたスイッチポートの外部にある NVMe 接続 ID。
namespace_id	ni	キー	数	×	名前空間 ID は、名前空間の NVMe コントローラの一意的識別子であり、1 ~ 255 の値に設定できます。これは、SCSI の論理ユニット番号 (LUN) に似ています。
target_id	did	キー	テキスト	いいえ	スイッチポートの外部にあるイニシエータによって開始された IO トランザクションを実行しているターゲットファイバチャンネルの識別子。
connection_id	ci	キー	数	×	メトリックが最後にクリアされて以降、IO を備えたスイッチポートの外部にある NVMe 接続識別子。
namespace_id	ni	キー	数	×	名前空間 ID は、名前空間の NVMe コントローラの一意的識別子であり、1 ~ 255 の値に設定できます。これは、SCSI の論理ユニット番号 (LUN) に似ています。
active_io_read_count	raIO	メタデータ	数	○	initiator-ITL-flow レコードと関連付けられている未処理の read コマンドカウントの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
active_io_write_count	waIO	メタデータ	数	○	initiator-ITL-flow レコードと関連付けられている未処理の write コマンドカウントの数。
total_read_io_count	rtIO	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された write コマンドデータの合計。
total_seq_read_io_count	rstIOc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された逐次な read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された逐次な write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	initiator-ITL-flow レコードの外部で確認された read コマンドの完了時間の累計。 この情報を使用して、読み取りIOの平均完了時間を計算できます。
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	initiator-ITL-flow レコードの外部で確認された write コマンドの完了時間の累計。 この情報を使用して、 write コマンドの平均完了時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	<p>initiator-ITL-flow レコードの外部で確認された read コマンドの開始時間の累計（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、data access latency とも呼ばれることもあります。</p> <p>この情報を使用して、読み取りIOの平均開始時間を計算できます。</p>
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	<p>initiator-ITL-flow レコードの外部で確認された write コマンドの開始時間の累計（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、data access latency とも呼ばれることもあります。</p> <p>この情報を使用して、write コマンドの平均開始時間を計算できます。</p>
total_read_io_bytes	rtIOb	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_bytes	wtIOb	メトリック	バイト	○	initiator-ITL-flow レコードと外部で確認された write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	initiator-ITL-flow レコードの外部で確認された read コマンドのインターギャップ時間の累計。 この情報を使用して、読み取り IO の平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	initiator-ITL-flow レコードの外部で確認された write コマンドのインターギャップ時間データの累計。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	initiator-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1 秒あたりの IO 数	○	initiator-ITL-flow レコード上の外部で確認された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
peak_read_io_rate	prIOr	メトリック	1秒あたりのIO数	×	initiator-ITL-flow レコード上の外部で確認された read コマンドのピークレート。
write_io_rate	wIOr	メトリック	1秒あたりのIO数	○	initiator-ITL-flow レコード上の外部で確認された write コマンドのレート。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1秒あたりのIO数	×	initiator-ITL-flow レコード上の外部で確認された write コマンドのピークレート。
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	initiator-ITL-flow レコードの外部で確認された read コマンド帯域幅。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	initiator-ITL-flow レコードの外部で確認されたピーク read コマンド帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	initiator-ITL-flow レコードの外部で確認された write コマンド帯域幅。 このメトリックは、NPU から4秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	initiator-ITL-flow レコードの外部で確認されたピーク write コマンド帯域幅。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_size_min	rIOsMi	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された read コマンドサイズの最小値。
read_io_size_max	rIOsMa	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	initiator-ITL-flow レコードの外部で確認された write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンド完了時間の最大値。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンド完了時間の最小値。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンド完了時間の最大値。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、またはWRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の read コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の write コマンドの開始時間（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された read コマンドインターギャップの最大値。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンドインターギャップの最小値。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された write コマンドインターギャップの最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された read コマンドの数。
write_io_aborts	wIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された write コマンドの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_failures	rIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した read コマンドの数。
write_io_failures	wIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した write コマンドの数。
read_io_nvme_lba_out_of_range_count	rIONLbaorct	メトリック	数	×	表示された read コマンド <i>lba</i> 範囲外エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaorct	メトリック	数	×	表示された write コマンド <i>lba</i> 範囲外エラーの数。
read_io_nvme_ns_not_ready_count	rIONNsnrc	メトリック	数	×	表示された read コマンド名前空間準備中エラーの数。
write_io_nvme_ns_not_ready_count	wIONNsnrc	メトリック	数	×	表示された write コマンド名前空間準備中エラーの数。
read_io_nvme_reservation_conflict_count	rIONrecct	メトリック	数	×	initiator-ITN-flow レコードの外部で確認された read コマンドの予約競合の数。
write_io_nvme_reservation_conflict_count	wIONrecct	メトリック	数	×	initiator-ITN-flow レコードの外部で確認された write コマンドの予約競合の数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。

ターゲット ITL フロー ビュー インスタンス (scsi_target_itl_flow)

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド ホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンド シーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンド ホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド ホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド 配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンド マルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンド マルチシーケンス交換シーケンス。

ターゲット ITL フロー ビュー インスタンス (scsi_target_itl_flow)



(注) ターゲット *ITL* フロー ビュー インスタンスのフロー メトリックは、SCSI 分析タイプにのみ適用されます。

表 31: ターゲット ITL フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。
target_id	did	キー	テキスト	いいえ	メトリックが最後にクリアされて以降、IO があるスイッチポートの外部にあるターゲットファイバチャネル識別子。
initiator_id	sid	キー	テキスト	いいえ	スイッチのポートの背後で展開されているイニシエータで IO トランザクションが実行されている Initiator Fibre Channel の ID。
lun	lun	キー	単位	×	IO が実行されるターゲットと関連付けられている論理ユニット番号 (LUN)。
active_io_read_count	raIO	メタデータ	数	○	target-ITL-flow レコードと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	target-ITL-flow レコードと関連付けられている未処理の write コマンドカウントの数。
total_read_io_count	rtIO	メトリック	数	○	target-ITL-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	target-ITL-flow レコードの外部で確認された write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_seq_read_io_count	rstIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された逐次な read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された逐次な write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された read コマンドの完了時間の累計。 この情報を使用して、読み取り IO の平均完了時間を計算できます。
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された write コマンドの完了時間の累計。 この情報を使用して、 write コマンドの平均完了時間を計算できます。
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された read コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。 この情報を使用して、読み取り IO の平均開始時間を計算できます。

フローメトリック		タイプ	単位	ノート可能	説明
正式名称	回線のテキストラベル				
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された write コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。 この情報を使用して、 write コマンドの平均開始時間を計算できます。
total_read_io_bytes	rtIOb	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_bytes	wtIOb	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された write コマンドデータの合計。
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された read コマンドのインターギャップ時間の累計。 この情報を使用して、読み取り IO の平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された write コマンドのインターギャップ時間データの累計。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計。

フローメトリック		タイプ	単位	ソース可能	説明
正式名称	回線のテキストラベル				
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1 秒あたりの IO 数	○	target-ITL-flow レコード上の LUN の外部で確認された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1 秒あたりの IO 数	×	target-ITL-flow レコード上の外部で確認された read コマンドのピーク レート。
write_io_rate	wIOr	メトリック	1 秒あたりの IO 数	○	target-ITL-flow レコード上の外部で確認された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1 秒あたりの IO 数	×	target-ITL-flow レコード上の外部で確認された write コマンドのピーク レート。
read_io_bandwidth	rIObw	メトリック	1 秒あたりのバイト数	○	target-ITL-flow レコードの外部で確認された read コマンド帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。

フローメトリック		タイプ	単位	ノート可能	説明
正式名称	回線のテキストラベル				
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	target-ITL-flow レコードの外部で確認されたピーク read コマンド帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	target-ITL-flow レコードの外部で確認された write コマンド帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	target-ITL-flow レコードの外部で確認されたピーク write コマンド帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された read コマンドサイズの最小値。
read_io_size_max	rIOsMa	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された write コマンドサイズの最大値。
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最大値。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最大値。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最大値。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された write コマンドインターギャップ時間の最小値。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された write コマンドインターギャップ時間の最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された read コマンドの数。
write_io_aborts	wIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された write コマンドの数。
read_io_failures	rIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した read コマンドの数。
write_io_failures	wIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した write コマンドの数。
read_io_scsi_check_condition_count	rIOSchcoct	メトリック	数	×	target-ITL-flow レコードの外部で確認された read コマンドのチェック条件の数。
write_io_scsi_check_condition_count	wIOSchcoct	メトリック	数	×	target-ITL-flow レコードの外部で確認された write コマンドのチェック条件の数。
read_io_scsi_busy_count	rIOsbc	メトリック	数	×	target-ITL-flow レコードの外部で確認された read コマンドのビジーステータスの数。
write_io_scsi_busy_count	wIOsbc	メトリック	数	×	target-ITL-flow レコードの外部で確認された write コマンドのビジーステータスの数。

フローメトリック		タイプ	単位	ノート可能	説明
正式名称	回線のテキストラベル				
read_io_scsi_reservation_conflict_count	rIOSrect	メトリック	数	×	target-ITL-flow レコードの外部で確認された read コマンドの予約競合の数。
write_io_scsi_reservation_conflict_count	wIOSrect	メトリック	数	×	target-ITL-flow レコードの外部で確認された write コマンドの予約競合の数。
read_io_scsi_queue_full_count	rIOSQfct	メトリック	数	×	target-ITL-flow レコードの外部で確認された read コマンドキューのフルステータスの数。
write_io_scsi_queue_full_count	wIOSQfct	メトリック	数	×	target-ITL-flow レコードの外部で確認された write コマンドキューのフルステータスの数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。

ターゲット ITN フロー ビュー インスタンス (nvme_target_itn_flow)

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド ホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド ホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチ ポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチ ポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチ ポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

ターゲット ITN フロー ビュー インスタンス (nvme_target_itn_flow)



(注) ターゲット *ITN* フロー ビュー インスタンスのフロー メトリックは、NVMe 分析タイプにのみ適用されます。

表 32: ターゲット *ITN* フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチ ポートに構成された VSAN。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。
target_id	did	キー	テキスト	いいえ	前回のメトリックのクリア以降の IO を持つスイッチポートの外部の Target Fibre Channel ID。
initiator_id	sid	キー	テキスト	いいえ	スイッチのポートの背後で展開されているイニシエータで IO トランザクションが実行されている Initiator Fibre Channel の ID。
connection_id	ci	キー	数	×	メトリックが最後にクリアされた以降の IO を備えたスイッチポートの外部の NVMe 接続 ID。
namespace_id	ni	キー	数	×	名前空間 ID は、名前空間の NVMe コントローラの一意的識別子であり、1 ~ 255 の値に設定できます。これは、SCSI の論理ユニット番号 (LUN) に似ています。
active_io_read_count	raIO	メタデータ	数	○	target-ITL-flow レコードと関連付けられている未処理の read コマンドカウントの数。
active_io_write_count	waIO	メタデータ	数	○	target-ITL-flow レコードと関連付けられている未処理の write コマンドカウントの数。
total_read_io_count	rtIO	メトリック	数	○	target-ITL-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_count	wtIO	メトリック	数	○	target-ITL-flow レコードの外部で確認された write コマンドデータの合計。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_seq_read_io_count	rstIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された逐次な read コマンドデータの合計。
total_seq_write_io_count	wrstIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された逐次な write コマンドデータの合計。
total_read_io_time	rtIOt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された read コマンドの完了時間の累計。 この情報を使用して、読み取りIOの平均完了時間を計算できます。
total_write_io_time	wtIOt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された write コマンドの完了時間の累計。 この情報を使用して、 write コマンドの平均完了時間を計算できます。
total_read_io_initiation_time	rtIOint	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された read コマンドの開始時間の累計（時間はIOコマンドとストレージからの最初の応答までの間隔、最初の応答は、 READ コマンドの最初のデータフレーム、または WRITE コマンドの最初の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。 この情報を使用して、読み取りIOの平均開始時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_initiation_time	wtIOint	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された write コマンドの開始時間の累計（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。 この情報を使用して、 write コマンドの平均開始時間を計算できます。
total_read_io_bytes	rtIOb	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された read コマンドデータの合計。
total_write_io_bytes	wtIOb	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された write コマンドデータの合計。
total_read_io_inter_gap_time	rtIOigt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された read コマンドのインターギャップ時間の累計。 この情報を使用して、読み取り IO の平均インターギャップ時間を計算できます。
total_write_io_inter_gap_time	wtIOigt	メトリック	マイクロ秒	×	target-ITL-flow レコードの外部で確認された write コマンドのインターギャップ時間データの累計。 この情報を使用して、 write コマンドの平均インターギャップ時間を計算できます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_time_metric_based_read_io_count	tmrtIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計。
total_time_metric_based_write_io_count	tmwtIOc	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計。
total_time_metric_based_read_io_bytes	tmrtIOb	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した read コマンドデータの合計 (バイト)。
total_time_metric_based_write_io_bytes	tmwtIOb	メトリック	数	×	target-ITL-flow レコードの外部で確認された完了した write コマンドデータの合計 (バイト)。
read_io_rate	rIOr	メトリック	1 秒あたりの IO 数	○	target-ITL-flow レコード上の LUN の外部で確認された read コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_rate	prIOr	メトリック	1 秒あたりの IO 数	×	target-ITL-flow レコード上の外部で確認された read コマンドのピーク レート。
write_io_rate	wIOr	メトリック	1 秒あたりの IO 数	○	target-ITL-flow レコード上の外部で確認された write コマンドのレート。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_rate	pwIOr	メトリック	1 秒あたりの IO 数	×	target-ITL-flow レコード上の外部で確認された write コマンドのピーク レート。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_bandwidth	rIObw	メトリック	1秒あたりのバイト数	○	target-ITL-flow レコードの外部で確認された read コマンド帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_read_io_bandwidth	prIObw	メトリック	1秒あたりのバイト数	×	target-ITL-flow レコードの外部で確認されたピーク read コマンド帯域幅。
write_io_bandwidth	wIObw	メトリック	1秒あたりのバイト数	○	target-ITL-flow レコードの外部で確認された write コマンド帯域幅。 このメトリックは、NPU から 4 秒間隔で収集された平均値です。
peak_write_io_bandwidth	pwIObw	メトリック	1秒あたりのバイト数	×	target-ITL-flow レコードの外部で確認されたピーク write コマンド帯域幅。
read_io_size_min	rIOsMi	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された read コマンドサイズの最小値。
read_io_size_max	rIOsMa	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された read コマンドサイズの最大値。
write_io_size_min	wIOsMi	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された write コマンドサイズの最小値。
write_io_size_max	wIOsMa	メトリック	バイト	○	target-ITL-flow レコードの外部で確認された write コマンドサイズの最大値。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
read_io_completion_time_min	rIOctMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最小値。
read_io_completion_time_max	rIOctMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最大値。
write_io_completion_time_min	wIOctMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最小値。
write_io_completion_time_max	wIOctMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンド完了時間の最小値。
read_io_initiation_time_min	rIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。
read_io_initiation_time_max	rIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の read コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency と呼ばれることもあります。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_initiation_time_min	wIOitMi	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最小の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
write_io_initiation_time_max	wIOitMa	メトリック	マイクロ秒	○	initiator-ITL-flow レコードの外部で確認された最大の write コマンドの開始時間（時間は IO コマンドとストレージからの最初の応答までの間隔、最初の応答は、READ の最初のデータフレーム、または WRITE の txfr_rdy である可能性があります）。開始時間は、 data access latency とも呼ばれることもあります。
read_io_inter_gap_time_min	rIOigtMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_inter_gap_time_max	rIOigtMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された read コマンドインターギャップ時間の最小値。 read_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_inter_gap_time_min	wIOigtMi	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された write コマンドインターギャップ時間の最小値。 write_io_inter_gap_time_min は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
write_io_inter_gap_time_max	wIOigtMa	メトリック	マイクロ秒	○	target-ITL-flow レコードの外部で確認された write コマンドインターギャップ時間の最大値。 write_io_inter_gap_time_max は連続した IO コマンド間の期間で、マイクロ秒の 1/256 で測定されます。
read_io_aborts	rIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された read コマンドの数。
write_io_aborts	wIOa	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、中止された write コマンドの数。
read_io_failures	rIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した read コマンドの数。
write_io_failures	wIOf	メトリック	数	○	initiator-ITL-flow レコードの外部で確認された、失敗した write コマンドの数。
read_io_nvme_lba_out_of_range_count	rIONLbaorct	メトリック	数	×	表示された read コマンド <i>lba out of range</i> エラーの数。
write_io_nvme_lba_out_of_range_count	wIONLbaorct	メトリック	数	×	表示された write コマンド <i>lba out of range</i> エラーの数。
read_io_nvme_ns_not_ready_count	rIONnsnrc	メトリック	数	×	表示された read コマンドの <i>namespace not ready</i> エラーの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_nvme_ns_not_ready_count	wIONnsnrc	メトリック	数	×	表示された write コマンドの <i>namespace not ready</i> エラーの数。
read_io_nvme_reservation_conflict_count	rIONrecct	メトリック	数	×	target-ITN-flow レコードの外部で確認された read コマンドの予約競合の数。
write_io_nvme_reservation_conflict_count	wIONrecct	メトリック	数	×	target-ITN-flow レコードの外部で確認された write コマンドの予約競合の数。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド ホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

イニシエーター IO フロー ビュー インスタンス (scsi_initiator_io および nvme_initiator_io)

表 33: イニシエーター IO フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
initiator_id	sid	キー	テキスト	いいえ	スイッチのポートの背後で展開されているイニシエータで IO トランザクションが実行されている Initiator Fibre Channel の ID。
target_id	did	キー	テキスト	いいえ	メトリックが最後にクリアされてからの、IO のあるスイッチポートの外部の Initiator Fibre Channel の ID。
lun	lun	キー	数	×	IO が実行されるイニシエータと関連付けられている論理ユニット番号 (LUN)。
connection_id	ci	キー	数	×	メトリックが最後にクリアされた以降の IO を備えたスイッチポートの外部の NVMe 接続 ID。
namespace_id	ni	キー	数	×	名前空間 ID は、名前空間の NVMe コントローラの一意の識別子であり、1 ~ 255 の値に設定できます。これは、SCSI の論理ユニット番号 (LUN) に似ています。
exchange_id	oxid	キー	数	×	IO トランザクションと関連付けられている、発信者によって割り当てられた交換 ID。
extended_exchange_id	exXID	メタデータ	数	×	IO トランザクションと関連付けられている、応答者によって割り当てられた拡張交換 ID。
io_lba	iolba	メタデータ	数	×	IO が実行されている論理ブロックアドレス (LBA)。
io_size	iosize	メタデータ	数	×	IO のサイズ。つまり、IO 関連のデータのバイトの数。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
io_start_time	iost	メトリック	数	○	IO 開始時のタイムスタンプ。
sampling_start_time	samStm	メトリック	UNIX 時間	×	サンプリング時間間隔の開始時間。
sampling_end_time	samEtm	メトリック	UNIX 時間	×	サンプリング時間間隔の終了時間。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチ ポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

ターゲット IO フロー ビュー インスタンス (scsi_target_io および nvme_target_io)

表 34: ターゲット IO フロー ビュー インスタンスのフロー メトリック

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
port	port	キー	テキスト	いいえ	SAN Analytics 機能が有効になっているスイッチのポート。
vsan	vsan	キー	数	×	メトリックが最後にクリアされてから、IO を使用してスイッチポートに構成された VSAN。
app_id	app_id	キー	数	×	スイッチのポートの背後でホストされているアプリケーションのアプリケーション ID。
target_id	did	キー	テキスト	いいえ	前回のメトリックのクリア以降の IO を持つスイッチ ポートの外部の Target Fibre Channel ID。
initiator_id	sid	キー	テキスト	いいえ	スイッチのポートの背後で展開されているイニシエータで IO トランザクションが実行されている Initiator Fibre Channel の ID。
lun	lun	キー	数	×	IO が実行されるターゲットと関連付けられている論理ユニット番号 (LUN)。

フローメトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
connection_id	ci	キー	数	×	メトリックが最後にクリアされた以降の IO を備えたスイッチポートの外部の NVMe 接続 ID。
namespace_id	ni	キー	数	×	名前空間 ID は、名前空間の NVMe コントローラの一意識別子であり、1 ~ 255 の値に設定できます。これは、SCSI の論理ユニット番号 (LUN) に似ています。
exchange_id	oxid	キー	数	×	IO トランザクションと関連付けられている、発信者によって割り当てられた交換 ID。
extended_exchange_id	exXID	メタデータ	数	×	IO トランザクションと関連付けられている、応答者によって割り当てられた拡張交換 ID。
io_lba	iolba	メタデータ	数	×	IO が実行されている論理ブロックアドレス (LBA)。
io_size	iosize	メタデータ	数	×	IO のサイズ。つまり、IO 関連のデータのバイトの数。
io_start_time	iost	メトリック	数	○	IO 開始時のタイムスタンプ。
total_busy_period	totBsy	メトリック	数	×	ビューインスタンスが現用系だった合計時間。
total_write_io_first_burst_count	totWrFirBu	メトリック	数	×	スイッチポートの外部で確認された write コマンドの最初のバーストの累計。
total_write_io_array_delay_time	totWrArrDel	メトリック	数	×	スイッチポートの外部で確認された write コマンド配列遅延の累計。
total_write_io_host_delay_time	totWrHosDel	メトリック	数	×	スイッチポートの外部で確認された write コマンドホスト遅延の累計。

フロー メトリック		タイプ	単位	ソート可能	説明
正式名称	回線のテキストラベル				
total_write_io_sequences_count	totWrSeq	メトリック	数	×	スイッチポートの外部で確認された write コマンドシーケンスの累計。
write_io_host_delay_time_min	wrHosDelMn	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最小値。
write_io_host_delay_time_max	wrHosDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンドホスト遅延の最大値。
write_io_array_delay_time_max	wrArrDelMx	メトリック	数	×	スイッチポートの外部で観測した write コマンド配列遅延の最大値。
multisequence_exchange_write_io_sequences_min	wrIoSeqMn	メトリック	数	×	スイッチポートの外部で観測した最小 write コマンドマルチシーケンス交換シーケンス。
multisequence_exchange_write_io_sequences_max	wrIoSeqMx	メトリック	数	×	スイッチポートの外部で観測した最大 write コマンドマルチシーケンス交換シーケンス。

インターフェイス カウンタ

次の表に、サポートされているインターフェイスカウンタのリストに関する情報を示します。

表 35: インターフェイス カウンタ

カウンタ名	説明
CTS SPI Mismatch	セキュリティ関連 ID が一致しない FCSP-ESP フレーム。
Diag Generated Frames	内部パケット ジェネレーターによって生成されたテスト フレーム。
FC2 InFrames	クラス 2 フレームを受信した回数。
FC2 OutFrames	クラス 2 フレームを送信された回数。
FC2 InOctets	クラス 2 入力オクテットの数。

カウンタ名	説明
FC2 OutOctets	クラス 2 出力オクテットの数。
FC2 Discards	タイムアウト、中止、オフラインなどの理由で、出力でクラス 2 フレームがドロップされた回数。
FC2 PRJT Frames	ポートが受信したクラス 2 フレームを拒否した数。
FC3 InFrames	クラス 3 フレームを受信した回数。
FC3 OutFrames	クラス 3 フレームを送信された回数。
FC3 InOctets	クラス 3 入力オクテットの数。
FC3 OutOctets	クラス 3 出力オクテットの数。
FC3 Discards	タイムアウト、中止、オフラインなどの理由で、出力でクラス 3 フレームがドロップされた回数。
FC InFrames	クラス F フレームを受信した回数。
FCF OutFrames	クラス F フレームを送信された回数。
FC InOctets	クラス F 入力オクテットの数。
FCF OutOctets	クラス F 出力オクテットの数。
FCF Discards	タイムアウト、中止、オフラインなどの理由で、出力でクラス F フレームがドロップされた回数。
Link Failures	Offline Sequence (OLS) エラーまたは Not Operational Sequence (NOS) エラーを受信したために、ファイバチャネルリンクがダウンした回数。
Sync Loss	Rx でファイバチャネルポートの同期が失われた回数。
Sig Loss	ファイバチャネルポートでレーザー信号が失われた回数。
Invalid Tx Words	ポートが無効な Tx ワードを受信した回数。
Invalid CRCs	ポートが内部 Cyclic Redundancy Check (CRC; 巡回冗長検査) エラーがあるフレームを受信した回数。
Delimiter Errors	デリミタ (フレームの先頭 (SoF) またはフレームの最後 (EoF)) のエラーがあるフレームを受信した回数。
Link Reset Ins	ポートがアクティブなときに、ファイバチャネルポートがリンクリセット (LR) プリミティブシーケンスを受信した回数。

カウンタ名	説明
Link Reset Outs	ポートがアクティブなときに、ファイバチャネルポート d d ええ LR プリミティブシーケンスを送信した回数。
OLS Ins	ファイバチャネルポートが OLS プリミティブシーケンスを受信した回数。
OLS Outs	ファイバチャネルポートが OLS プリミティブシーケンスを送信した回数。
Runt Frames In	CRC または FCS エラーに関係なく、ファイバチャネルポートが受信したフレームのうち、許容される最小フレーム長より短いフレームの数。
Jabber Frames In	ファイバチャネルポートが受信した、最大フレーム長より長く、CRC または FCS エラーもあるフレームの数。
TxWait	TxWait カウンタは、ポートの送信待機時間をカウントする集約時間カウンタです。送信待機とは、ポートに利用可能な送信クレジットがなく (tx b2b=0)、フレームが送信待ちになっている状態です。カウンタは 2.5 マイクロ秒単位で増加します。カウント値を秒単位で計算するには、TxWait 数に 2.5 を掛けて、1,000,000 で割ります。
Frames Too Long	設定されている最大ファイバチャネルフレームサイズより長いフレームを受信した回数。
Frames Too Short	設定されている最小ファイバチャネルフレームサイズより短いフレームを受信した回数。
FLRR In	ポートがアクティブなときに、ファイバチャネルポートがリンクリセット応答 (LRR) プリミティブシーケンスを受信した回数。
FLRR Out	ポートがアクティブなときに、ファイバチャネルポートがリンクリセット応答 (LRR) プリミティブシーケンスを送信された回数。
NOS In	ポートで NOS が受信された回数。
NOS Out	ポートで NOS が送信された回数。
EOF Frames	無効なフレームの最後 (EoF) フレームを受信した回数。
Unknown Class Frames	不明なクラス フレームを受信した回数。
Tx BBcredit Transition to Zero	インターフェイスの Tx BB_credit がゼロのまま、送信できなかった回数。

カウンタ名	説明
Rx BBCredit Transition to Zero	インターフェイスの Tx BB_credit がゼロのまま、送信できなかった回数。
ELS Frames Discard	廃棄された Extended Link Service (ELS) フレームの数。
Framing Error Frames	フレーミング エラーのあるフレームを受信した回数。
In Multicast Pkts	マルチキャスト フレームを受信した回数。
In Broadcast Pkts	ブロードキャスト フレームを受信した回数。
Out Multicast Pkts	マルチキャスト フレームが送信された回数。
Out Broadcast Pkts	ブロードキャスト フレームが送信された回数。
HC InOctets	大容量入力オクテットの数。
HC InUcast Pkts	ユニキャスト フレームを受信した回数。
HC InMulticast Pkts	マルチキャスト フレームを受信した回数。
HC InBroadcast Pkts	ブロードキャスト パケットを受信した回数。
HC OutOctets	大容量出力オクテットの数。
HC OutUCast Pkts	ユニキャスト パケットが送信された回数。
HC OutMulticast Pkts	マルチキャスト パケットが送信された回数。
HC OutBroadcast Pkts	ブロードキャスト パケットが送信された回数。
InOctets	入力オクテットの数。
In UCast Pkts	ユニキャスト フレームを受信した回数。
In Discards	廃棄を受信した回数。
In Errors	受信したエラーの数。
Out Octets	出力オクテットの数。
Out Ucast Pkts	ユニキャスト パケットが送信された回数。
Out Discards	廃棄が送信された回数。
FC Out Errors	ファイバ チャンネル エラーが送信された回数。
LIP F8 In	Loop Initiation Protocol (LIP) F8 プリミティブを受信した回数。
LIP F8 Out	LIP F8 プリミティブが送信された回数。

カウンタ名	説明
Non Lip F8 In	非 LIP F8 プリミティブを受信した回数。
Non Lip F8 Out	非 LIP F8 プリミティブが送信された回数。
IfIn Frames	入力フレームの数。
IfIn Octets	入力フレームの数 (バイト数)。
IfIn Discards	入力フレームがドロップされた回数。
IfOut Frames	出力フレームの数。
IfOut Octets	出力フレームの数 (バイト数)。
IfOut Discards	出力フレームがドロップされた回数。
IfIn Errors	入力エラーの数。
IfOut Errors	出力エラーの数。
Zone Drops	ポートグループのデバイスに対するゾーン分割が設定されていないためにドロップされたフレームの数。
FIB Drops	ポートグループのフォワーディングルックアップミスが原因でドロップされたフレームの数。
Xbar Drops	ポートグループのファブリックスイッチング (クロスバー) エラーが原因でドロップされたフレームの数。
そのほかの削除	ポートグループのその他のエラーが原因でドロップされたフレームの数。
BB_SCs クレジット再送信アクション	ポートが失われたフレームを検出し、追加のクレジット (R_RDY) を再送信することによってピアクレジットアカウントを修正した回数。
BB_SCrTx クレジット増分アクション	ポートが R_RDY の損失を検出し、TXB2B クレジット利用可能ステータスをインクリメントすることによってローカルクレジットアカウントを修正した回数。
VL0 の Tx B2B クレジットがゼロに遷移	インターフェイスがゼロ Tx BB_credits で残り、仮想リンク 0 で送信できなかった回数。
VL1 の Tx B2B クレジットがゼロに遷移	インターフェイスがゼロ Tx BB_credits に残り、仮想リンク 1 で送信できなかった回数。
VL2 の Tx B2B クレジットがゼロに遷移	インターフェイスがゼロ Tx BB_credits で残り、仮想リンク 2 で送信できなかった回数。

カウンタ名	説明
VL 3 の Tx B2B クレジットがゼロに移行	インターフェイスがゼロ Tx BB_credits で残り、仮想リンク 3 で送信できなかった回数。
Rx B2B credit transitions to zero for VL 0	インターフェイスで仮想リンク 0 の残りの Rx BB_credits がゼロだった回数。
Rx B2B credit transitions to zero for VL 1	インターフェイスで仮想リンク 1 の残りの Rx BB_credits がゼロだった回数。
Rx B2B credit transitions to zero for VL 2	仮想リンク 2 の残りの Rx BB_credits がゼロだった回数。
Rx B2B credit transitions to zero for VL 3	仮想リンク 3 の残りの Rx BB_credits がゼロだった回数。
送信クレジットが不足しているため、TxWait 2.5us	インターフェイスが 2.5 マイクロ秒の間、Tx クレジットがゼロだった回数。
TxWait 2.5us due to lack of transmit credits for VL 0	仮想リンク 0 で 2.5 マイクロ秒の間、インターフェイスがゼロ Tx クレジットであった回数。
TxWait 2.5us due to lack of transmit credits for VL 1	インターフェイスが仮想リンク 1 で 2.5 マイクロ秒間ゼロ Tx クレジットであった回数。
TxWait 2.5us due to lack of transmit credits for VL 2	インターフェイスが仮想リンク 2 で 2.5 マイクロ秒間ゼロ Tx クレジットであった回数。
TxWait 2.5us due to lack of transmit credits for VL 3	インターフェイスが仮想リンク 3 で 2.5 マイクロ秒間、Tx クレジットがゼロだった回数。
Timeout Discards	輻輳ドロップタイムアウトまたは no-credit-drop タイムアウトが原因でスイッチでドロップされたフレームは、タイムアウト廃棄として考慮されます。タイムアウト破棄の増分は、送信方向の輻輳を示します。
TxWait VL0	VL 0 の送信クレジットがないため、TxWait 2.5us。
TxWait VL1	VL 1 の送信クレジットがないため、TxWait 2.5us。
TxWait VL2	VL 2 の送信クレジットがないため、TxWait 2.5us。
TxWait VL3	VL 3 の送信クレジットがないため、TxWait 2.5us。
Rx BBZ VL0	Rx B2B クレジットは VL 0 でゼロに遷移します。
Rx BBZ VL1	Rx B2B クレジットは VL 1 のゼロに遷移します。
Rx BBZ VL2	VL 2 の Rx B2B クレジットはゼロに遷移します。

カウンタ名	説明
Rx BBZ VL3	VL 3 の Rx B2B クレジットはゼロに遷移します。
送信BBZ VL0	VL 0 の Tx B2B クレジットはゼロに遷移します。
Tx BBZ VL1	VL 1 の Tx B2B クレジットはゼロに遷移します。
Tx BBZ VL2	VL 2 の Tx B2B クレジットはゼロに遷移します。
Tx BBZ VL3	VL 3 の Tx B2B クレジットはゼロに遷移します。

SAN テレメトリ ストリーミング Proto ファイル

このセクションでは、コンパクト GPB で使用される *.proto* ファイルについて説明します。

次に、*telemetry_bis.proto* ファイルの内容を示します。

```

/* -----
 * telemetry_bis.proto - Telemetry protobuf definitions
 *
 * August 2016
 *
 * Copyright (c) 2016 by Cisco Systems, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 * -----
 */

syntax = "proto3";

option go_package = "telemetry_bis";

option cc_enable_arenas = true;

/*
 * Common message used as a header to both compact and self-describing
 * telemetry messages.
 */

message Telemetry {
  oneof node_id {
    string node_id_str = 1;
    // bytes node_id_uuid = 2;           // not produced
  }
  oneof subscription {
    string subscription_id_str = 3;
  }
}

```

```

    // uint32  subscription_id = 4;           // not produced
  }
  // string  sensor_path = 5;               // not produced
  string  encoding_path = 6;
  // string  model_version = 7;           // not produced
  uint64  collection_id = 8;
  uint64  collection_start_time = 9;
  uint64  msg_timestamp = 10;
  repeated TelemetryField data_gpbkv = 11;
  TelemetryGPBTable data_gpb = 12;
  uint64  collection_end_time = 13;
  // uint64  heartbeat_sequence_number = 14; // not produced
}

/*
 * Messages used to export content in GPB K/V form.
 *
 * The set of messages in this .proto are sufficient to decode all
 * telemetry messages.
 */

message TelemetryField {
  uint64      timestamp = 1;
  string      name = 2;
  oneof value_by_type {
    bytes      bytes_value = 4;
    string     string_value = 5;
    bool       bool_value = 6;
    uint32     uint32_value = 7;
    uint64     uint64_value = 8;
    sint32     sint32_value = 9;
    sint64     sint64_value = 10;
    double     double_value = 11;
    float      float_value = 12;
  }
  repeated TelemetryField fields = 15;
}

/*
 * Messages used to export content in compact GPB form
 *
 * Per encoding-path .proto files are required to decode keys/content
 * pairs below.
 */

message TelemetryGPBTable {
  repeated TelemetryRowGPB row = 1;
}

message TelemetryRowGPB {
  uint64 timestamp = 1;
  bytes  keys = 10;
  bytes  content = 11;
}

```

次に、*fabric_telemetry.proto* ファイルの内容を示します。



(注) 出力の *exceed_count* カウンタは、将来の Cisco MDS NX-OS リリースでサポートされる予定です。


```
/* -----  
* fabric_telemetry.proto - Fabric Telemetry protobuf definitions  
*  
* July 2018  
*  
* Copyright (c) 2018 by Cisco Systems, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
*   http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
* -----  
*/  
  
syntax = "proto3";  
  
option go_package = "fabric_telemetry";  
  
option cc_enable_arenas = true;  
  
message ControlInformation {  
    string version = 1;  
    uint32 chunk_sequence = 2;  
    uint32 total_chunks_count = 3;  
}  
  
message FlowRecordsTable {  
    ControlInformation control_info = 1;  
    repeated FlowRecordRow row = 2;  
}  
  
message FlowRecordRow {  
    string port = 1;  
    uint32 app_id = 2;  
    uint32 vsan = 3;  
    string target_id = 4;  
    string initiator_id = 5;  
    string lun = 6;  
    string exchange_id = 7;  
    uint32 scsi_target_count = 8;  
    uint32 scsi_initiator_count = 9;  
    uint32 io_app_count = 10;  
    uint32 logical_port_count = 11;  
    uint32 scsi_target_app_count = 12;  
    uint32 scsi_initiator_app_count = 13;  
    uint32 active_io_read_count = 14;  
    uint32 active_io_write_count = 15;  
    uint32 scsi_target_tl_flow_count = 16;  
    uint32 scsi_target_it_flow_count = 17;  
    uint32 scsi_initiator_it_flow_count = 18;  
    uint32 scsi_target_itl_flow_count = 19;  
    uint32 scsi_initiator_itl_flow_count = 20;  
    uint32 scsi_target_lun_count = 21;  
    uint32 scsi_target_entity_it_flow_count = 22;  
    uint32 scsi_initiator_entity_it_flow_count = 23;  
    uint32 scsi_target_entity_itl_flow_count = 24;
```

```
uint32 scsi_initiator_entity_itl_flow_count = 25;
uint64 sampling_start_time = 26;
uint64 sampling_end_time = 27;
string extended_exchange_id = 28;
string io_lba = 29;
uint32 io_size = 30;
uint64 total_read_io_count = 31;
uint64 total_write_io_count = 32;
uint64 total_seq_read_io_count = 33;
uint64 total_seq_write_io_count = 34;
uint64 total_read_io_time = 35;
uint64 total_write_io_time = 36;
uint64 total_read_io_initiation_time = 37;
uint64 total_write_io_initiation_time = 38;
uint64 total_read_io_bytes = 39;
uint64 total_write_io_bytes = 40;
uint64 total_read_io_inter_gap_time = 41;
uint64 total_write_io_inter_gap_time = 42;
uint64 total_time_metric_based_read_io_count = 43;
uint64 total_time_metric_based_write_io_count = 44;
uint64 total_time_metric_based_read_io_bytes = 45;
uint64 total_time_metric_based_write_io_bytes = 46;
uint64 io_start_time = 47;
uint32 read_io_rate = 48;
uint32 peak_read_io_rate = 49;
uint32 write_io_rate = 50;
uint32 peak_write_io_rate = 51;
uint32 read_io_bandwidth = 52;
uint32 peak_read_io_bandwidth = 53;
uint32 write_io_bandwidth = 54;
uint32 peak_write_io_bandwidth = 55;
uint32 read_io_size_min = 56;
uint32 read_io_size_max = 57;
uint32 write_io_size_min = 58;
uint32 write_io_size_max = 59;
uint32 read_io_completion_time_min = 60;
uint32 read_io_completion_time_max = 61;
uint32 write_io_completion_time_min = 62;
uint32 write_io_completion_time_max = 63;
uint32 read_io_initiation_time_min = 64;
uint32 read_io_initiation_time_max = 65;
uint32 write_io_initiation_time_min = 66;
uint32 write_io_initiation_time_max = 67;
uint32 read_io_inter_gap_time_min = 68;
uint32 read_io_inter_gap_time_max = 69;
uint32 write_io_inter_gap_time_min = 70;
uint32 write_io_inter_gap_time_max = 71;
uint32 peak_active_io_read_count = 72;
uint32 peak_active_io_write_count = 73;
uint32 read_io_aborts = 74;
uint32 write_io_aborts = 75;
uint32 read_io_failures = 76;
uint32 write_io_failures = 77;
uint32 read_io_timeouts = 78;
uint32 write_io_timeouts = 79;
uint32 read_io_scsi_check_condition_count = 80;
uint32 write_io_scsi_check_condition_count = 81;
uint32 read_io_scsi_busy_count = 82;
uint32 write_io_scsi_busy_count = 83;
uint32 read_io_scsi_reservation_conflict_count = 84;
uint32 write_io_scsi_reservation_conflict_count = 85;
uint32 read_io_scsi_queue_full_count = 86;
uint32 write_io_scsi_queue_full_count = 87;
uint32 read_io_rate_exceed_count = 88;
```

```
uint32 write_io_rate_exceed_count = 89;
uint32 read_io_bandwidth_exceed_count = 90;
uint32 write_io_bandwidth_exceed_count = 91;
uint32 read_io_size_min_exceed_count = 92;
uint32 read_io_size_max_exceed_count = 93;
uint32 write_io_size_min_exceed_count = 94;
uint32 write_io_size_max_exceed_count = 95;
uint32 read_io_initiation_time_min_exceed_count = 96;
uint32 read_io_initiation_time_max_exceed_count = 97;
uint32 write_io_initiation_time_min_exceed_count = 98;
uint32 write_io_initiation_time_max_exceed_count = 99;
uint32 read_io_completion_time_min_exceed_count = 100;
uint32 read_io_completion_time_max_exceed_count = 101;
uint32 write_io_completion_time_min_exceed_count = 102;
uint32 write_io_completion_time_max_exceed_count = 103;
uint32 read_io_inter_gap_time_min_exceed_count = 104;
uint32 read_io_inter_gap_time_max_exceed_count = 105;
uint32 write_io_inter_gap_time_min_exceed_count = 106;
uint32 write_io_inter_gap_time_max_exceed_count = 107;
uint32 read_io_abort_exceed_count = 108;
uint32 write_io_abort_exceed_count = 109;
uint32 read_io_failure_exceed_count = 110;
uint32 write_io_failure_exceed_count = 111;
uint64 total_abts_count = 112;
uint32 namespace_id = 113;
string connection_id = 114;
uint32 nvme_target_count = 115;
uint32 nvme_initiator_count = 116;
uint32 nvme_target_app_count = 117;
uint32 nvme_initiator_app_count = 118;
uint32 nvme_target_tn_flow_count = 119;
uint32 nvme_target_it_flow_count = 120;
uint32 nvme_initiator_it_flow_count = 121;
uint32 nvme_target_itn_flow_count = 122;
uint32 nvme_initiator_itn_flow_count = 123;
uint32 nvme_target_namespace_count = 124;
uint32 nvme_target_entity_it_flow_count = 125;
uint32 nvme_initiator_entity_it_flow_count = 126;
uint32 nvme_target_entity_itn_flow_count = 127;
uint32 nvme_initiator_entity_itn_flow_count = 128;
uint32 read_io_nvme_lba_out_of_range_count = 129;
uint32 write_io_nvme_lba_out_of_range_count = 130;
uint32 read_io_nvme_ns_not_ready_count = 131;
uint32 write_io_nvme_ns_not_ready_count = 132;
uint32 read_io_nvme_reservation_conflict_count = 133;
uint32 write_io_nvme_reservation_conflict_count = 134;
uint32 read_io_nvme_capacity_exceeded_count = 135;
uint32 write_io_nvme_capacity_exceeded_count = 136;
}
```


翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。