

# Informix - Hohe CPU-Auslastung

## Inhalt

[Einführung](#)

[Informationen zu Funktionen](#)

[Fehlerbehebungsmethode](#)

[Datenanalyse](#)

[Häufige Probleme](#)

## Einführung

In diesem Dokument wird beschrieben, wie Aktivitäten von Unified Contact Center Express (UCCX), die lokalen UCCX-Datenbankzugriff erfordern, langsam durchgeführt werden können. Dies bewirkt, dass die AppAdmin-Seiten langsam geladen werden, dass Aktualisierungen an AppAdmin eine lange Anwendungsverzögerung erfordern, dass die Antwort auf eine Wallboard-Abfrage verzögert wird, dass Workforce Manager UCCX-Daten nicht abfragen kann und andere Leistungs- und Stabilitätsprobleme auftreten.

Der Befehl **show process load**, der in der CLI eingegeben wurde, zeigt, dass der **Uccxoninit** eine große CPU-Menge verbraucht. Der **uccxoninit**-Prozess stellt die UCCX Informix-Datenbankinstanz dar, die auf dem UCCX-Server ausgeführt wird.

Unterstützt von Sridhar Chandrasekharan, Ryan LaFountain und Ben Wollak, Cisco TAC Engineers.

## Informationen zu Funktionen

Die Datenbank-Engine, die die UCCX-Anwendung unterstützt, ist IBM Informix. Konfigurations- und Verlaufsdaten, die der AppAdmin-Seite von UCCX hinzugefügt und von der UCCX-Anwendung erstellt werden, werden in der UCCXInformix-Instanz gespeichert.

Die UCCX-Anwendung stellt drei Benutzer bereit, die zum direkten Zugriff auf die UCCX-Datenbank verwendet werden können, um Informationen für die Zwecke von Wallboard-Anwendungen, Qualitätsmanagement, Workforce Management und benutzerdefinierten Verlaufsberichten zu extrahieren.

Die Benutzerinformationen, die Berechtigungen der einzelnen Benutzer und der Zweck der einzelnen Benutzer werden hier beschrieben:

- **uccxhruser**: Dieser Benutzer hat Berechtigungen für viele Konfigurations- und Verlaufstabellen in der UCCX-Datenbank und sollte nur für benutzerdefinierte Verlaufsberichte und Cisco Unified Workforce Management (WFM) verwendet werden. Abfragen und gespeicherte Prozeduren, die von diesem Benutzer ausgeführt werden, können komplexe, lange ausgeführte Abfragen ausführen. Aufgrund des Profils eines typischen Verlaufsberichts- oder WFM-Benutzers sollten diese Abfragen und gespeicherten Prozeduren nicht häufig ausgeführt werden, wie dies bei einer Wallboard-Anwendung der Fall wäre.

Obwohl viele Wallboard-Anwendungen Daten benötigen, die in der Konfiguration und in Verlaufstabellen enthalten sind, auf die der uccxhruser Zugriff hat, wird es technisch nicht unterstützt, diesen Benutzer zu verwenden, um komplexe, häufige Abfragen der UCCX-Datenbank für die Zwecke einer Wallboard-Anwendung auszuführen.

- **Nachwachskräfte** - Der uccxworkforce-Benutzer hat Zugriff auf die Tabellen "Team", "Ressourcen" und "Supervisor" und sollte für das Cisco Unified Quality Management (QM) verwendet werden. Das Workforce Management sollte uccxhruser verwenden, da es Zugriff auf Verlaufstabellen von Daten erfordert, auf die der uccxworkforce-Benutzer nicht zugreifen kann.
- **uccxwallonisch** - Dieser Benutzer verfügt nur über die Berechtigungen für Echtzeit-Datenbanktabellen, die Snapshots von Echtzeitstatistiken enthalten, die aus dem Speicher der UCCX Engine geschrieben wurden. Die auf die Tabellen RTCSQsSummary und RTICDStatistics beschränkten Auswahlberechtigungen bedeuten, dass der Uccxwallboard-Benutzer dazu verwendet werden sollte, die UCCX-Datenbank häufig mit einfachen, nicht komplexen Abfragen abzufragen, die von einer Wallboard-Anwendung stammen sollen.

## Fehlerbehebungsmethode

Geben Sie in UCCX Version 10.0 und höher den Befehl **utils uccx database dbperf start <totalHours> <interval> ein**, um mit der Performance-Ablaufverfolgung in der UCCX-Datenbank zu beginnen. Das **interval**-Argument in diesem Befehl bestimmt die Periodizität der Ablaufverfolgungsaufzählung, und das **totalHours**-Argument bestimmt die Gesamtdauer, die die Ablaufverfolgung ausführt, bevor sie deaktiviert wird. Diese Parameter sind optional. Wenn sie bei Ausführung des Befehls nicht angegeben werden, werden die Standardwerte von 20 Minuten und 10 Stunden verwendet.

Geben Sie z. B. den Befehl **utils uccx database dbperf start 24 30** ein, um die Performance-Ablaufverfolgung in der Datenbank zu aktivieren und alle 30 Minuten Daten über Leistungsstatistiken für 24 Stunden zu erfassen.

In der Befehlsausgabe werden Anweisungen zum Erfassen der vom CLI-Befehl erhaltenen Daten ausgegeben.

```
admin:utils uccx database dbperf stop
Execution of dbperf has been stopped
Command Successful
admin:utils uccx database dbperf start 24 30
The script runs every 30 minutes over a total duration of 24 hours.
Please collect files after 24 hours

Use "file get activelog uccx/cli/dbperf_171013134928.log" to get the file
Use "file view activelog uccx/cli/dbperf_171013134928.log" to view the file
Command Successful
admin:█
```

Nach dem angegebenen **totalHours** wird die Datenerfassung automatisch beendet. Um die Datenerfassung manuell zu beenden, geben Sie den Befehl **utils uccx database dbperf stop** ein.

```
admin:utils uccx database dbperf stop
Execution of dbperf has been stopped
Command Successful
admin:█
```

Wenn die UCCX-Version Version 9.0(2) oder früher ist und die **utils uccx datenbank dbperf** nicht verfügbar ist, wenden Sie sich an das Technical Assistance Center (TAC), um weitere Unterstützung zu erhalten.

Das TAC führt das der Cisco Bug-ID [CSCuc68413](#) beigefügte dbperf.sh-Skript manuell mit Remote Support Account-Zugriff aus.

Wenn Sie festlegen, wann die Skriptausführung entweder manuell oder über den CLI-Befehl, die Periodizität und die Gesamtdauer gestartet werden soll, stellen Sie sicher, dass die vom **Uccxoninit** Prozesse schwanken erheblich oder bleiben in diesen Zeiträumen hoch, um die erforderlichen Informationen für die Ursachenanalyse zu sammeln.

Zusätzlich geben Sie in regelmäßigen Abständen den Befehl **show process load** ein, um zu bestimmen, wann die CPU schwankt, um die vom dbperf-Ablaufverfolgungsskript erfassten Protokolle zu korrelieren.

## Datenanalyse

Die Protokolle, die durch die Ausführung von **onstat -g ses 0** durch das dbperf-Skript gesammelt wurden, zeigen aktive Abfragen an, die in der UCCX-Datenbank ausgegeben werden. Hohe CPU auf dem **uccxoninit**-Prozess ist in der Regel das Ergebnis komplexer Abfragen, deren Ausführung lange Zeit in Anspruch nimmt. Das Ziel besteht darin, die Abfragen zu bestimmen, die die meisten Ressourcen nutzen, den Quellclient für diese Abfragen zu bestimmen, die Abfragen des Clients zur sofortigen Auflösung zu deaktivieren und die langlaufenden Abfragen zur dauerhaften Auflösung zu optimieren.

Suchen Sie in den vom dbperf-Skript gesammelten Protokollen nach Abfragen, die höchstwahrscheinlich hohe CPU-Schwankungen verursachen oder eine anhaltend hohe CPU-Nutzung durch den **uccxoninit**-Prozess verursachen.

Verdächtige Abfragen:

- werden aus Sitzungen ausgegeben, die als **uccxhruser** verbunden sind - Wie bereits beschrieben, **uccxhruser** verfügt über Berechtigungen zur Auswahl von Informationen aus einer großen Anzahl von Konfigurations- und Verlaufstabellen. Auf diese Weise können komplexe, lange ausgeführte Abfragen über mehrere Tabellen hinweg erstellt werden, was sich auf die Leistung der UCCX-Datenbank auswirken kann. Auch wenn dies nicht absolut ist, haben die **Benutzer von uccxwallboard** und **uccxworkforce** derart eingeschränkten Zugriff auf Tabellen in der UCCX-Datenbank. Komplexe Abfragen, die die Leistung dieser Benutzer beeinträchtigen, sind unwahrscheinlich. Darüber hinaus werden Anfragen vom UCCX Historical Reporting Client (HRC) oder Cisco Unified Intelligence Center (CUIC) an die UCCX-Datenbank ausgegeben. Diese Abfragen sind statisch und können nicht geändert werden. Die Abfragen sowie die relevanten Indizes wurden geschrieben, getestet und auf minimale Auswirkungen auf die Leistung abgestimmt.

- Durchführen intensiver Abfragen von Verlaufstabellen: Abfragen, bei denen die UCCX-Datenbank mehrere Verknüpfungen über Tabellen hinweg ausführt, große Datenmengen auswählt oder nicht indizierte Felder ausführt, können Leistungseinbußen für die UCCX-Datenbank verursachen.

Ein Beispiel mit einer komplexen Abfrage, die eine als **uccxhruser** ausgeführte HR-Tabelle umfasst, ist hier dargestellt:

```
session #RSAM total used dynamic
id user tty pid hostname threads memory memory explain
435050 uccxhrus WBBOX 836 10.16.5. 1 90112 80712 off

.....

Current SQL statement :
SELECT x.resourceName, t.eventType, x.datetime, x.extension FROM ( SELECT
t1.resourceID, t1.resourceName, t1.extension, MAX(t2.eventDateTime) AS
datetime FROM Resource AS t1, AgentStateDetail AS t2 WHERE t2.agentID
= t1.resourceID AND t1.assignedTeamID = 21 and t1.active GROUP BY
t1.resourceID, t1.resourceName, t1.extension ) AS x, AgentStateDetail AS
t WHERE t.agentID = x.resourceID AND t.eventDateTime = x.datetime
ORDER BY x.resourceName
```

Das obige Beispiel zeigt eine komplexe Abfrage, die von **uccxhruser** aus der **WBBOX**-Host-**WBBOX** eingegeben wird und die Leistungseinbußen für die UCCX-Datenbank verursachen kann, wenn sie häufig eingegeben wurde oder regelmäßig eingegeben wurde, bevor die vorherige Abfrage Ergebnisse zurückgegeben hatte.

Die UCCX-Datenbankleistung ist zwar selten, kann jedoch auch die CPU-Auslastung des **Uccxoninit** Prozess schwankt oder bleibt hoch), aufgrund des integrierten Löschmoduls. Der Löschmodul dient zum Löschen von Daten aus der Konfiguration und den Verlaufstabellen in der UCCX-Datenbank, um die Größe der Datenbank beizubehalten. Die Löschung kann basierend auf der Größe der Datenbank oder dem ältesten Datensatz in der Datenbank geplant werden.

Beim Ausführen des Löschmoduls werden die Daten mit einer Abfrage entfernt. Der Vorgang wird basierend auf der Anzahl der zu entfernenden Datensätze nicht iterativ durchgeführt. Das bedeutet, wenn die Löschung eine große Datenmenge erkennt, die entfernt werden muss, wird eine einzige Abfrage ausgegeben, um diese Daten zu entfernen.

Die Änderung des Zurücksetzungszeitplans oder der Parameter auf der UCCXAppAdmin-Seite, um die Löschung großer Datenmengen zu planen, kann dazu führen, dass diese einzelne Abfrage bei der nächsten planmäßigen Löschung eine beträchtliche Zeitspanne in Anspruch nimmt. Daher wird die CPU-Auslastung der Datenbankinstanz erhöht.

In der Ausgabe des dbperf-Skripts wird die Löschemfrage angezeigt. Dabei sollte es sich um die einzige Abfrage handeln, die vom Benutzer **uccxuser** eingegeben wurde, der die gespeicherte **sp\_purge**-Prozedur aufruft.

```
session #RSAM total used dynamic
id user tty pid hostname threads memory memory explain
5628 uccxuser - -1 CC-EXPR- 1 544768 523408 off
```

```
Current SQL statement in procedure db_cra:sp_purge
proc-counter 0x0x4ccf9260 opcode SQL
```

```

delete from contactroutingdetail
where (exists
(select 1
 from contactcalldetail as ccdr
 where (and (and (and (and (and (= contactroutingdetail.sessionid,
ccdr.sessionid), (= contactroutingdetail.nodeid, ccdr.nodeid)),
(= contactroutingdetail.sessionseqnum, ccdr.sessionseqnum)),
(= contactroutingdetail.profileid, ccdr.profileid)), (>= ccdr.enddatetime,
p_purgefrom)), (< ccdr.enddatetime, p_purgeto)))));

```

## Häufige Probleme

Basierend auf den Erfahrungen des Cisco TAC und der Cisco Development Engineering sind dies die am häufigsten auftretenden Probleme, die eine hohe CPU-Auslastung im **uccxoninit**-Prozess verursachen:

- Ein Client im Unternehmen stellt eine Verbindung als **uccxhruser her** und führt häufig komplexe Abfragen an den Wallboard-Tabellen (RTICDStatistics und RTCSQsSummary) zusammen mit den historischen Tabellen aus, um eine Wallboard- oder benutzerdefinierte Reporting-Lösung bereitzustellen. Verwenden Sie für die Verwendung des Wandboards nur den Benutzer **uccxwallboard** und beschränken Sie die Abfragen auf die Echtzeittabellen. Die Möglichkeit, die Verlaufs- oder Konfigurationstabellen von einer Wandtafel oder mit einer ähnlichen Frequenz wie eine Wandtafel abzurufen, wird nicht unterstützt.
- Ein Client versucht, benutzerdefinierte Verlaufsberichte auf dem aktiven Primärknoten statt auf dem sekundären Knoten auszuführen. Führen Sie nur benutzerdefinierte oder standardmäßige gespeicherte Prozeduren aus, die Verlaufsberichte auf dem Standby-Knoten erstellen. CUIC und HRC führen standardmäßig Abfragen auf dem Standby-Knoten aus. Beim Entwickeln eines benutzerdefinierten Verlaufsberichts hat der Entwickler jedoch die Wahl, auf welchem Knoten diese Abfragen ausgeführt oder diese gespeicherten Prozeduren ausgeführt werden sollen.
- Cisco Workforce Management (WFM) stellt in der Tabelle ContactRoutingDetail eine komplexe Abfrage aus, um zu versuchen, das Feld "Startzeit" zu filtern. In dieser Tabelle wird in diesem Feld standardmäßig kein Index erstellt, sodass die Leistung dieser Abfrage gering ist. WFM gibt diese Abfrage regelmäßig aus, um Daten von UCCX mit WFM zu synchronisieren. Dieses Problem wird in der Cisco Bug-ID [CSCtz23710](#) erfasst und in WFM Version 9.0(1)SR4 behoben. Kunden, die dieses Problem haben, sollten ein Upgrade auf eine WFM-Version durchführen, die eine Behebung für die Cisco Bug-ID [CSCtz23710](#) enthält.
- Die Löschschwellenwerte werden so geändert, dass bei der nächsten planmäßigen Löschung versucht wird, eine große Datenmenge zu entfernen. Anstatt die Purge-Parameter in einem einzelnen Update erheblich zu ändern, werden die Änderungen des Löschrplans iterativ vorgenommen, wobei einige Tage zwischen den Konfigurationsänderungen für das Zurücksetzen liegen. Auf diese Weise können beim Löschvorgang kleinere Datensätze in jedem Durchgang entfernt werden, was die Leistung des Löschvorgangs verbessert.
- Die DialingList-Tabelle ist extrem groß. In der Wählliste werden alle Kontakte gespeichert, die an ausgehende Kampagnen hochgeladen wurden. In UCCX-Versionen 8.0 und 8.5 werden nach dem Hochladen von Millionen Datensätzen in Outbound-Kampagnen Leistungsprobleme auftreten, und die Tabelle wird abgefragt (dies führt zu einer hohen CPU im uccxoninit-Prozess und langsamen Laden der AppAdmin-Seite). Um Leistungsprobleme zu beheben, öffnen Sie ein TAC-Ticket für die Installation eines Cron-Job-Skripts, das die DialingList-

Tabelle bereinigt. In UCCX Version 9.0 wurde dieser Tabelle ein Index hinzugefügt, um effektivere Abfragen von AppAdmin zu ermöglichen, um die Leistung zu verbessern. Diese Änderung löste das Problem in allen bis auf die extremsten Fälle aus. In UCCX-Version 10.0 wurde die Wählliste in zwei Tabellen aufgeteilt: eine für aktive Kontakte und eine andere für historische Kontakte, die eine umfassende Lösung für dieses Problem bietet.