Risoluzione dei problemi relativi a un'interfaccia grafica APIC lenta

Sommario

Introduzione

Questo documento descrive la metodologia generale per risolvere i problemi relativi a un'interfaccia GUI APIC lenta.

Avvio rapido

Spesso si riscontra che i problemi lenti dell'interfaccia grafica APIC sono il risultato di un'alta percentuale di richieste API originate da uno script, un'integrazione o un'applicazione. Il file access.log di un file APIC registra ogni richiesta API elaborata. Il file access.log di un APIC può essere analizzato rapidamente con lo script Access Log Analyzer all'interno del progetto aci-tac-scripts del gruppo Datacenter di Github.

Premesse

APIC come server Web - NGINX

NGINX è il DME responsabile degli endpoint API disponibili su ciascun APIC. Se NGINX non è attivo, le richieste API non possono essere gestite. Se NGINX è congestionato, l'API è congestionata. Ogni APIC esegue il proprio processo NGINX, quindi è possibile che solo un singolo APIC possa avere problemi con NGINX se solo tale APIC è oggetto di query aggressive.

L'interfaccia utente APIC esegue più richieste API per popolare ogni pagina. Analogamente, tutti i comandi 'show' APIC (NXOS Style CLI) sono wrapper per script Python che eseguono più richieste API, gestiscono la risposta, quindi la servono all'utente.

Registri rilevanti

Nome file di log	Posizione	In quale supporto tecnico	Commenti
file access.log	/var/log/dme/log	APIC 3 di 3	ACI agnostic, fornisce 1 linea per richiesta API
errore.log	/var/log/dme/log	APIC 3 di 3	ACI Agnostic, mostra errori Inginx (limitazione inclusa)
nginx.bin.log	/var/log/dme/log	APIC 3 di 3	specifico di ACI, registra le transazioni DME
nginx.bin.warnplus.log	/var/log/dme/log	APIC 3 di 3	Contiene registri con livello di avviso + gravità

Metodologia

Isolamento trigger iniziale

Quali sono le conseguenze?

- Quali APIC sono interessati; uno, molti o tutti?
- Dove si nota la lentezza: tramite UI, comandi CLI o entrambi?
- Quali pagine o comandi specifici dell'interfaccia utente sono lenti?

Come si sperimenta la lentezza?

- Questa condizione viene rilevata in più browser per un singolo utente?
- Più utenti segnalano la lentezza o solo un singolo utente o un sottoinsieme di utenti?
- Gli utenti interessati condividono una posizione geografica o un percorso di rete simile dal browser all'APIC?

Quando è stata notata per la prima volta la lentezza?

- È stata aggiunta di recente un'integrazione o uno script ACI?
- L'estensione del browser è stata abilitata di recente?
- Si è verificata una modifica recente nella configurazione ACI?

Verifica stato e utilizzo NGINX

Formato voce Access.log

access.log è una funzionalità di NGINX ed è pertanto indipendente da APIC. Ogni riga rappresenta una richiesta HTTP ricevuta da APIC. Fare riferimento a questo registro per informazioni sull'utilizzo di NGINX di un APIC.

Formato predefinito di access.log in ACI versione 5.2+:

```
log_format proxy_ip '$remote_addr ($http_x_real_ip) - $remote_user [$time_local]'
                    '"$request" $status $body_bytes_sent '
                    '"$http_referer" "$http_user_agent"';
```

Questa riga rappresenta una voce access.log quando viene eseguita una moquery -c fvTenant:

```
127.0.0.1 (-) - - [07/Apr/2022:20:10:59 +0000]"GET /api/class/fvTenant.xml HTTP/1.1" 200 15863 "-" "Pyth
```

Mappa della voce access.log di esempio a log_format:

campo log_format	**Contenuto da esempio**	**Commenti**
$remote_addr	127.0.0.1	IP dell'host che ha inviato la richiesta
$http_x_real_ip	-	IP dell'ultimo richiedente se proxy in uso
$utente_remoto	-	Generalmente non utilizzato. Selezionare nginx.bin.log per tenere traccia dell'utente che ha eseguito l'accesso per eseguire le richieste
$ora_locale	07/Apr/2022:20:10:59 +0000	Quando la richiesta è stata elaborata
$request	SCARICA /api/class/fvTenant.xml HTTP/1.1	Metodo Http (GET, POST, DELETE) e URI
$status	200	Codice di stato risposta HTTP
$body_bytes_sent	1586	dimensioni payload risposta

$http_referer	-	-
$http_user_agent	Python-urllib	Tipo di client che ha inviato la richiesta

Comportamenti di Access.log

Frequenza elevata di picchi di richieste in un lungo periodo di tempo:

- I burst continui di oltre 15 richieste al secondo possono causare rallentamenti dell'interfaccia utente
- Identificare gli host responsabili delle query
- Ridurre o disabilitare l'origine delle query per verificare se questo migliora i tempi di risposta di APIC.

Risposte 4xx o 5xx coerenti:

- Identificare il messaggio di errore da nginx.bin.log

Verifica utilizzo risorse NGINX

Per controllare l'utilizzo della CPU e della memoria NGINX, usare il comando **top** dell'APIC:

<#root>

```
top - 13:19:47 up 29 days, 2:08, 11 users, load average: 12.24, 11.79, 12.72
Tasks: 785 total, 1 running, 383 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.5 us, 2.0 sy, 0.0 ni, 94.2 id, 0.1 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 13141363+total, 50360320 free, 31109680 used, 49943636 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 98279904 avail Mem

PID   USER PR NI VIRT    RES   SHR   S %CPU %MEM TIME+     COMMAND
21495 root 20 0  4393916 3.5g 217624 S
```

2.6

2.8 759:05.78

nginx.bin

L'elevato utilizzo delle risorse NGINX può essere direttamente correlato a un elevato numero di richieste elaborate.

Verifica core

Un arresto anomalo di NGINX non è tipico per i problemi relativi all'interfaccia grafica dell'APIC lento. Tuttavia, se vengono individuati core NGINX, collegarli a un TAC SR per l'analisi. Per la procedura di verifica dei core, consultare la guida ACI Techsupport.

Verifica latenza da client a server

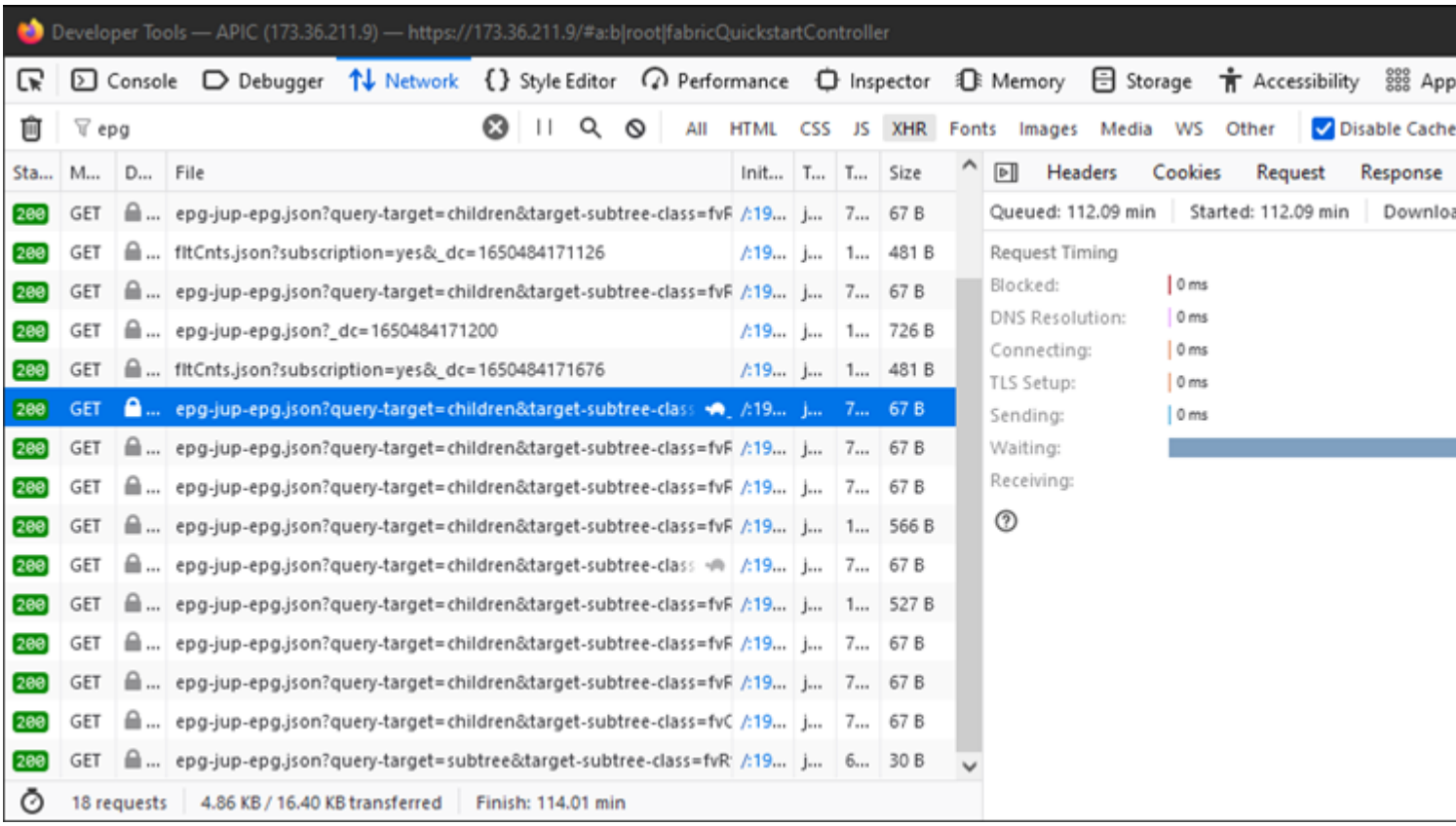
Se non vengono trovate richieste rapide ma un utente continua a mostrare lentezza dell'interfaccia utente, il problema può essere la latenza da client (browser) a server (APIC).

In questi scenari, convalidare il percorso dati dal browser all'APIC (distanza geografica, VPN, ecc.). Se possibile, distribuire e verificare l'accesso da un server di collegamento situato nella stessa area geografica o nello stesso centro dati degli APIC da isolare. Convalida se altri utenti mostrano una latenza simile.

Scheda Rete degli strumenti di sviluppo del browser

Tutti i browser sono in grado di convalidare le richieste e le risposte HTTP tramite il relativo toolkit di **sviluppo dei browser**, in genere all'interno di una scheda **Rete**.

Questo strumento può essere utilizzato per convalidare la quantità di tempo necessaria per ogni fase delle richieste originate dal browser, come mostrato nell'immagine.



Esempio di browser in attesa di una risposta da parte dell'APIC di 1,1 minuti

Miglioramenti per pagine specifiche dell'interfaccia utente

Pagina Gruppo di criteri:

L'ID bug Cisco CSCvx14621 - l'interfaccia grafica APIC viene caricata lentamente sui criteri IPG nella scheda Fabric.

Interfaccia nella pagina Inventory:

Cisco ID bug CSCvx90048 - Il carico iniziale della scheda operativa "Layer 1 Physical Interface Configuration" è lungo e induce al 'blocco'.

Suggerimenti generali per Client > Latenza server

Alcuni browser, come Firefox, consentono per impostazione predefinita più connessioni Web per host.

- Verificare se l'impostazione è configurabile nella versione del browser in uso
- Questo aspetto è più importante per le pagine con più query, ad esempio la pagina Gruppo di criteri

La velocità VPN e la distanza da APIC aumentano la lentezza complessiva dell'interfaccia utente in base alle richieste del browser client e al tempo di spostamento della risposta APIC. Un jump box geograficamente locale rispetto agli APIC riduce in modo significativo i tempi di spostamento del browser rispetto agli APIC.

Verifica richieste Web lunghe

Se un server Web (NGINX su APIC) gestisce un volume elevato di richieste Web lunghe, ciò può influire sulle prestazioni di altre richieste ricevute in parallelo.
Ciò è particolarmente vero per i sistemi che dispongono di database distribuiti, ad esempio APIC. Una singo

Tempo di risposta del sistema - Abilita calcolo per il tempo di risposta del server

In 4.2(1)+, un utente può abilitare il "Calcolo delle prestazioni del sistema" che tiene traccia ed evidenzia le

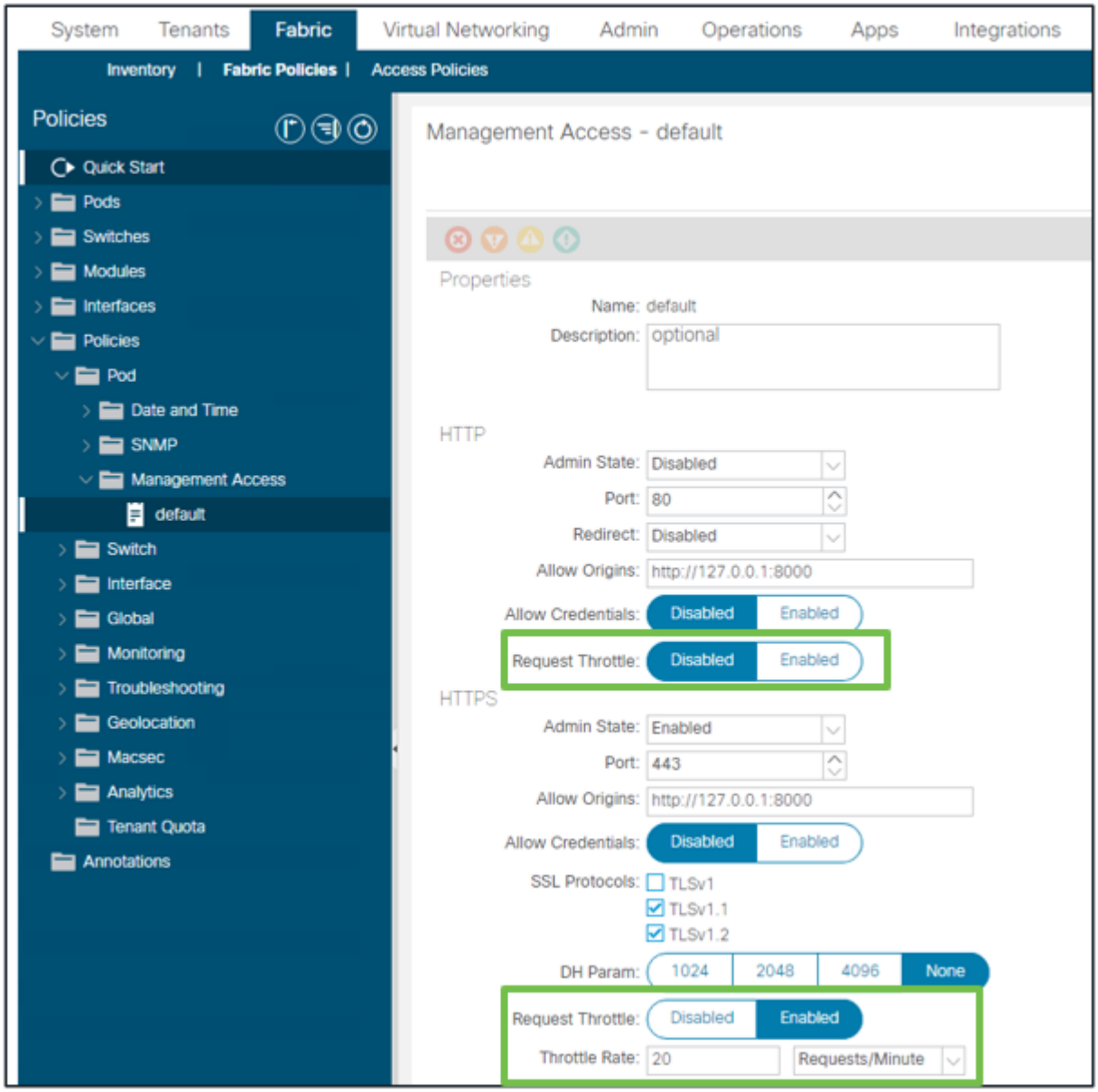
- In generale, più di 15 richieste API al secondo in un lungo periodo di tempo debilita NGINX.
    - Se individuato, ridurre l'aggressività delle richieste.
    - Se l'host Richieste non può essere modificato, prendere in considerazione i limiti di velocità NGINX sull'APIC.

Inefficienze degli script di indirizzo

- Non eseguire l'accesso/la disconnessione prima di ogni richiesta API.
    - Il timeout predefinito per una sessione di accesso è 10 minuti. La stessa sessione può essere utilizzata per più richieste e può essere aggiornata per estendere il periodo di validità.
    - Vedere Guida alla configurazione dell'API REST Cisco APIC - Accesso all'API REST - Autenticazione e gestione di una sessione API.
- Se lo script esegue query su molti DN che condividono un padre, anziché comprimere le query in una singola query padre logica con Filtri query.
    - Vedere Guida alla configurazione dell'API REST Cisco APIC - Composizione delle query dell'API REST - Applicazione dei filtri di ambito delle query.
- Se è necessario aggiornare un oggetto o una classe di oggetti, prendere in considerazione le sottoscrizioni websocket anziché le richieste API rapide.

Limitazione richieste NGINX

Disponibile nella versione 4.2(1)+, un utente può abilitare la limitazione delle richieste su HTTP e HTTPS in modo indipendente.



Fabric - Criteri fabric - Cartella criteri - Cartella accesso gestione - predefinito

Quando abilitato:

- NGINX viene riavviato per applicare le modifiche al file di configurazione
  - Una nuova zona, **httpsClientTagZone**, viene scritta nella configurazione di Inginx
- La velocità può essere impostata in **Richieste al minuto** (r/m) o **Richieste al secondo** (r/s).
- La limitazione delle richieste si basa sull'implementazione del limite di velocità inclusa in NGINX
  - Le richieste API sull'interfaccia **/api/**URI utilizzano la velocità definita dall'utente + burst= (velocità x 2) + nodelay
    - Esiste una limitazione non configurabile (zona **aaaApiHttps**) per **/api/aaaLogin** e **/api/aaaRefresh** che limita la velocità a 2r/s + burst=4 + nodelay
  - La velocità delle richieste viene registrata in base all'indirizzo IP del client
  - Le richieste API originate dall'indirizzo IP automatico APIC (UI + CLI) ignorano la limitazione
  - Qualsiasi indirizzo IP del client che supera la velocità definita dall'utente + soglia di burst riceve una risposta 503 dall'APIC
  - Questi 503 possono essere correlati nei log degli accessi
  - error.log includerà voci che indicano quando è stata attivata la limitazione (zona **httpsClientTagZone**) e per quali host client

<#root>

```
apic#
```

less /var/log/dme/log/error.log

```
...
2023/04/17 20:19:14 [error] ...
```

limiting requests

```
, excess: 40.292 by zone "
```

httpsClientTagZone

```
", client: h.o.s.t, ... request: "GET /api/class/...", host: "a.p.i.c"
2023/04/17 20:19:14 [error] ...
```

limiting requests

```
, excess: 40.292 by zone "
```

httpsClientTagZone

```
", client: h.o.s.t, ... request: "GET /api/node/...", host: "a.p.i.c"
```

Come regola generale, Request Throttle serve solo a proteggere il server (APIC) da sintomi simili a quelli di DDOS indotti da client che aggrediscono le query. Comprendere e isolare il client di richiesta per le soluzioni finali nella logica dell'app/script.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).