

Prüfung des Testdokuments

Einleitung

In diesem Dokument wird die allgemeine Methode zur Fehlerbehebung bei einer langsamen grafischen Benutzeroberfläche des APIC beschrieben.

Schnellstart

Es wird häufig festgestellt, dass langsame APIC-GUI-Probleme das Ergebnis einer hohen Rate von API-Anfragen sind, die von einem Skript, einer Integration oder einer Anwendung stammen. Die Datei access.log eines APIC protokolliert jede verarbeitete API-Anforderung. Die Datei access.log eines APIC kann mit dem Skript [Access Log Analyzer](#) innerhalb des [aci-tac-scripts](#)-Projekts der Github Datacenter-Gruppe schnell analysiert werden.

Hintergrundinformationen

APIC als Webserver - NGINX

NGINX ist die DME, die für die auf den einzelnen APICs verfügbaren API-Endpunkte verantwortlich ist. Wenn NGINX ausgefallen ist, können API-Anfragen nicht verarbeitet werden. Bei Überlastung von NGINX ist die API überlastet. Jeder APIC führt seinen eigenen NGINX-Prozess aus. Daher kann es vorkommen, dass nur ein APIC NGINX-Probleme hat, wenn nur dieser APIC Ziel aggressiver Queries ist.

Die APIC-Benutzeroberfläche führt mehrere API-Anforderungen aus, um jede Seite auszufüllen. Ebenso sind alle APIC-show-Befehle (CLI im NXOS-Stil) Wrapper für Python-Skripts, die mehrere API-Anforderungen ausführen, die Antwort verarbeiten und sie dann an den Benutzer weitergeben.

Relevante Protokolle

Protokolldateiname	Location (Standort)	In welchem Technologiesupport befinden sich die Lösungen?	Kommentare
access.log	/var/log/dme/log	APIC 3 von 3	ACI-unabhängig, 1 Leitung pro API-Anfrage
error.log	/var/log/dme/log	APIC 3 von 3	ACI-unabhängig, zeigt nginx-Fehler

Diese Zeile stellt einen access.log-Eintrag dar, wenn ein mockery -c fvTenant ausgeführt wird:

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

Zuordnung eines Beispieleintrags access.log zu log_format:

Logformat-Feld	Inhalt aus Beispiel	Kommentare
\$remote_addr	127.0.0.1	IP des Hosts, der diese Anforderung gesendet hat
\$http_x_real_ip	-	IP des letzten Anforderers, wenn Proxys verwendet werden
\$remote_user	-	Nicht in der Regel verwendet. Überprüfen Sie nginx.bin.log, um festzustellen, welcher Benutzer sich angemeldet hat, um Anforderungen auszuführen.
\$time_local	07.04.2022:20:10:59 +0000	Wann wurde die Anforderung verarbeitet?
Anfrage in USD	/api/class/fvTenant.xml HTTP/1.1 herunterladen	HTTP-Methode (GET, POST, DELETE) und URI
\$ Status	200	HTTP-Antwortstatuscode
\$body_bytes_sent	1586	Größe der Antwortnutzlast
\$http_referer	-	-
\$http_user_agent	Python-Urllib	Welcher Clienttyp hat die Anforderung gesendet?

Verhalten von Access.log

Anforderungen mit hoher Rate treten über einen langen Zeitraum hinweg auf:

- Kontinuierliche Spitzen von mehr als 40 Anfragen pro Sekunde können eine verlangsamte Benutzeroberfläche verursachen.
- Identifizieren Sie die Hosts, die für die Anfragen verantwortlich sind.
- Reduzieren oder deaktivieren Sie die Quellenangabe, um festzustellen, ob sich dadurch die Reaktionszeit des APIC verbessert.

Konsistente 4xx- oder 5xx-Antworten:

- Wenn gefunden, geben Sie die Fehlermeldung aus nginx.bin.log ein.

Die Datei access.log eines APIC kann mit dem Skript [Access Log Analyzer](#) innerhalb des [aci-tac-scripts](#)-Projekts der Github Datacenter-Gruppe schnell analysiert werden.

NGINX-Ressourcennutzung überprüfen

Die NGINX-CPU- und Speicherauslastung kann mit dem obersten Befehl des APIC überprüft werden:

<#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

Eine hohe NGINX-Ressourcennutzung kann direkt mit einer hohen Rate verarbeiteter Anfragen in Zusammenhang stehen.

Auf Kerne prüfen

Ein NGINX-Absturz ist nicht typisch für Probleme mit der langsameren APIC-GUI. Wurden NGINX-Kerne gefunden, fügen Sie sie zur Analyse einem TAC-Serviceticket hinzu. Im [ACI-](#)

[Technischen Support-Handbuch](#) finden Sie die Schritte zur Suche nach Kernen.

Client-Server-Latenz überprüfen

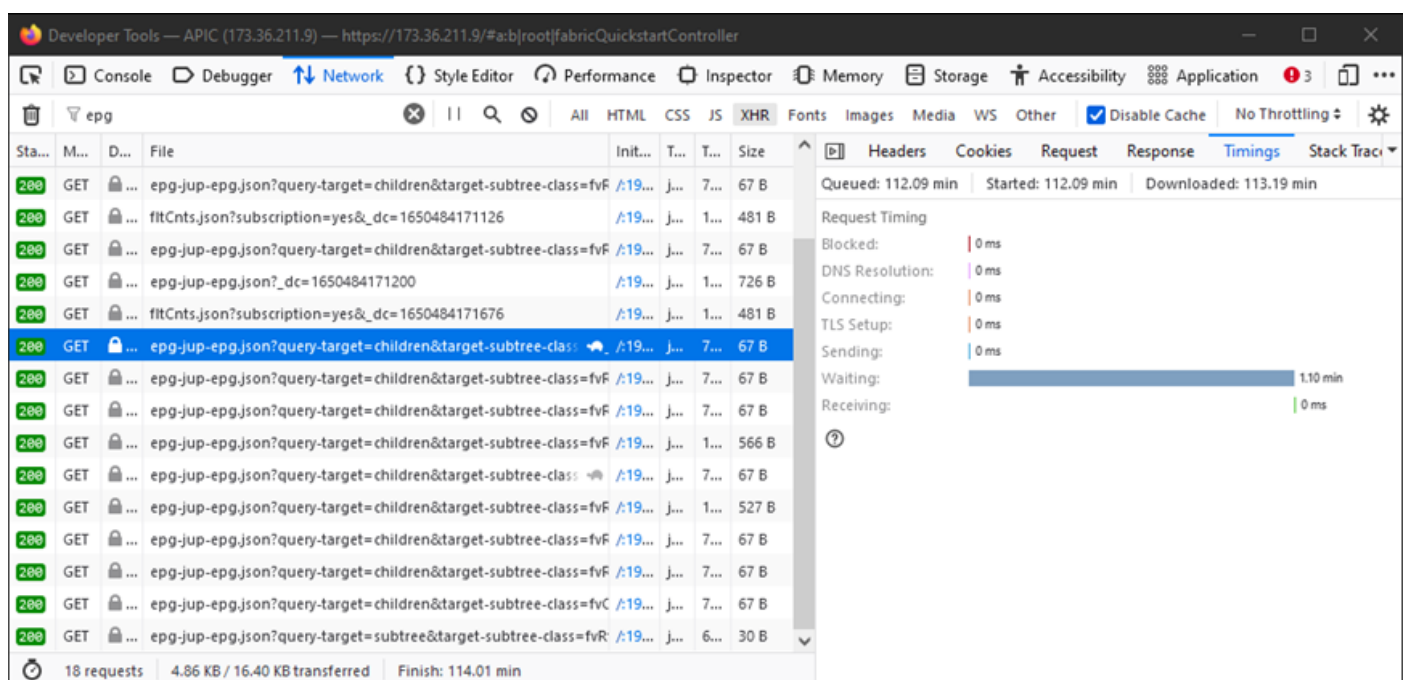
Wenn keine schnellen Anfragen gefunden werden, ein Benutzer jedoch weiterhin eine langsame Benutzeroberfläche aufweist, kann es zu einer Latenz zwischen Client (Browser) und Server (APIC) kommen.

Validieren Sie in diesen Szenarien den Datenpfad vom Browser zum APIC (geografische Entfernung, VPN usw.). Wenn möglich, Bereitstellung und Test des Zugriffs von einem Jump Server in derselben geografischen Region oder demselben Rechenzentrum wie die zu isolierenden APICs. Überprüfen Sie, ob andere Benutzer eine ähnliche Latenz aufweisen.

Browser-Entwicklungstools Registerkarte Netzwerk

Alle Browser können HTTP-Anfragen und -Antworten über ihr Browser Development Toolkit validieren, in der Regel über die Registerkarte Netzwerk.

Dieses Tool kann verwendet werden, um die Zeit zu validieren, die für jede Phase der Browser-Anfragen benötigt wird, wie im Bild gezeigt.



Beispiel für einen Browser, der 1,1 Minuten auf Antwort des APIC wartet

Verbesserungen für bestimmte Benutzeroberflächenseiten

Seite "Policy Group":

Cisco Bug-ID [CSCvx14621](#) - Die APIC-GUI wird langsam in die IPG-Richtlinien auf der Registerkarte "Fabric" geladen.

Schnittstelle auf der Inventarseite:

Cisco Bug-ID [CSCvx90048](#) - Initial Load von "Layer 1 Physical Interface Configuration" (Physische Layer-1-Schnittstellenkonfiguration), Registerkarte "Operational" (Betrieb) ist lang/induziert "Freeze" (Einfrieren).

Allgemeine Empfehlungen für Client > Server-Latenz

Bestimmte Browser wie Firefox ermöglichen standardmäßig mehr Webverbindungen pro Host.

- Überprüfen Sie, ob diese Einstellung für die verwendete Browserversion konfigurierbar ist.
- Dies ist für Seiten mit mehreren Abfragen von größerer Bedeutung, z. B. für die Seite "Policy Group" (Richtliniengruppe).

VPN und die Entfernung zum APIC erhöhen die Gesamtlangsames der Benutzeroberfläche angesichts der Anfragen von Client-Browsern und der Reaktionszeit des APIC. Ein geografisch lokales Jump-Box für die APICs verkürzt den Browser erheblich auf die APIC-Reisezeiten.

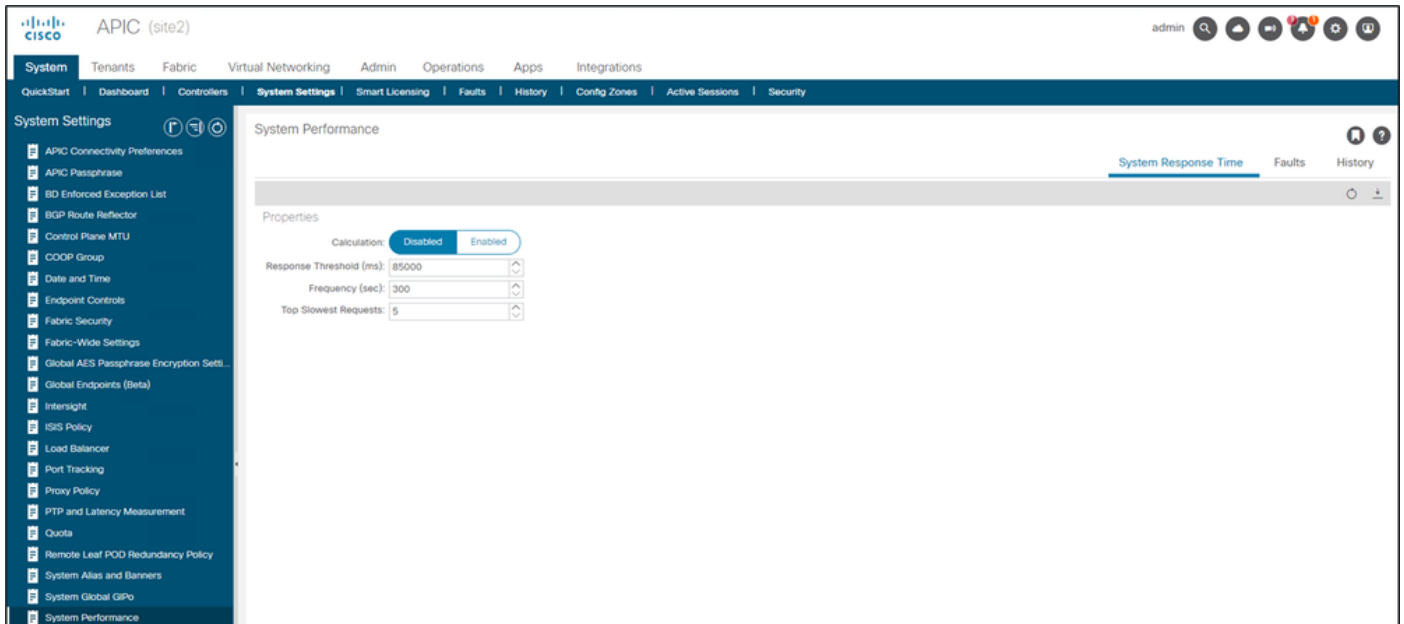
Überprüfen auf Long-Web-Anforderungen

Wenn ein Webserver (NGINX auf dem APIC) eine große Anzahl von Long-Web-Anfragen verarbeitet, kann dies die Leistung anderer parallel empfangener Anfragen beeinträchtigen.

Dies gilt insbesondere für Systeme mit verteilten Datenbanken, wie APICs. Eine einzelne API-Anforderung kann zusätzliche Anforderungen und Suchvorgänge erfordern, die an andere Knoten in der Fabric gesendet werden. Dies kann zu erwartungsgemäß längeren Antwortzeiten führen. Ein plötzlicher Anstieg dieser Long-Web-Anfragen innerhalb eines kleinen Zeitrahmens kann die Anzahl der erforderlichen Ressourcen erhöhen und zu unerwartet längeren Reaktionszeiten führen. Außerdem können dann empfangene Anfragen eine Zeitüberschreitung (90 Sekunden) verursachen, was aus Benutzersicht zu unerwartetem Systemverhalten führt.

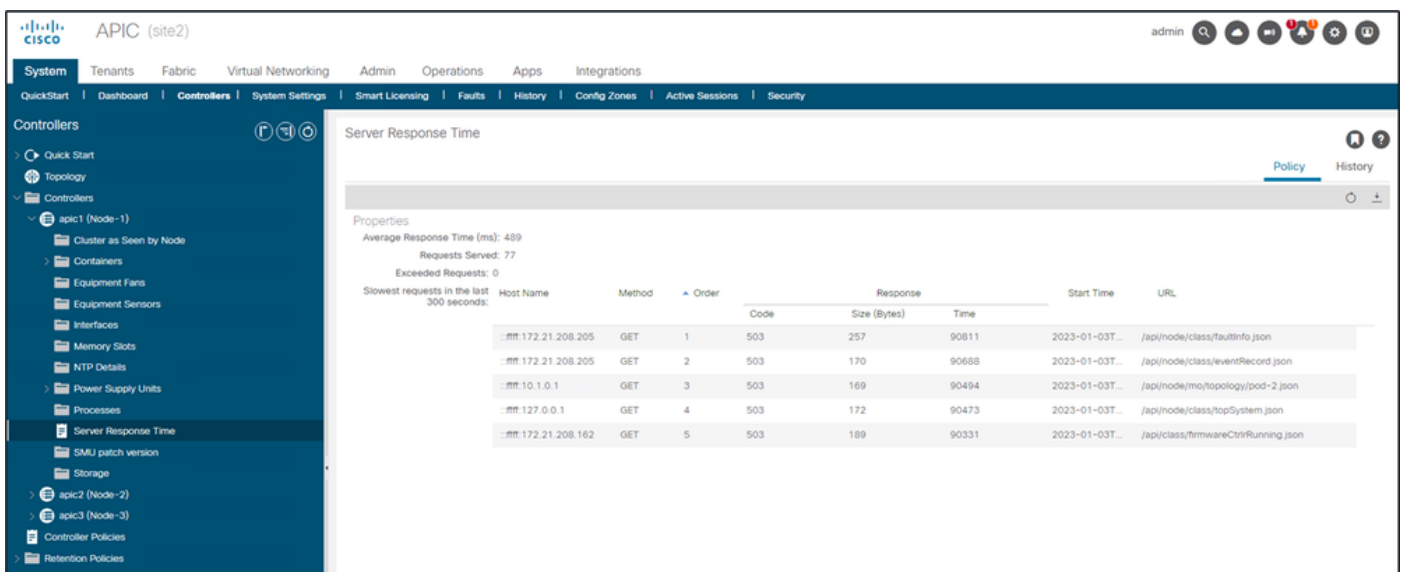
Systemreaktionszeit - Berechnung der Serverreaktionszeit aktivieren

In Version 4.2(1)+ kann ein Benutzer die "Berechnung der Systemleistung" aktivieren, mit der API-Anforderungen verfolgt und hervorgehoben werden, deren Bearbeitung viel Zeit in Anspruch genommen hat.



Die Berechnung kann über System - Systemeinstellungen - Systemleistung aktiviert werden.

Sobald "Calculation" aktiviert ist, kann ein Benutzer unter Controllers zu bestimmten APICs navigieren, um die langsamsten API-Anfragen innerhalb der letzten 300 Sekunden anzuzeigen.



System - Controller - Controller-Ordner - APIC x - Server-Reaktionszeit

APIC API-Nutzungsüberlegungen

Allgemeine Zeiger, um sicherzustellen, dass ein Skript Nginx nicht beschädigt

- Jeder APIC führt seine eigene NGINX-DME aus.
 - Nur das NGINX von APIC 1 verarbeitet Anfragen an APIC 1. Das NGINX von APIC 2 und 3 verarbeitet diese Anfragen nicht.
- Im Allgemeinen schränkt mehr als 40 API-Anfragen pro Sekunde über einen langen Zeitraum NGINX ein.
 - Reduzieren Sie ggf. die Aggressivität der Anfragen.

- Wenn der Host für Anfragen nicht geändert werden kann, sollten Sie [NGINX-Ratenlimits](#) auf dem APIC in Betracht ziehen.

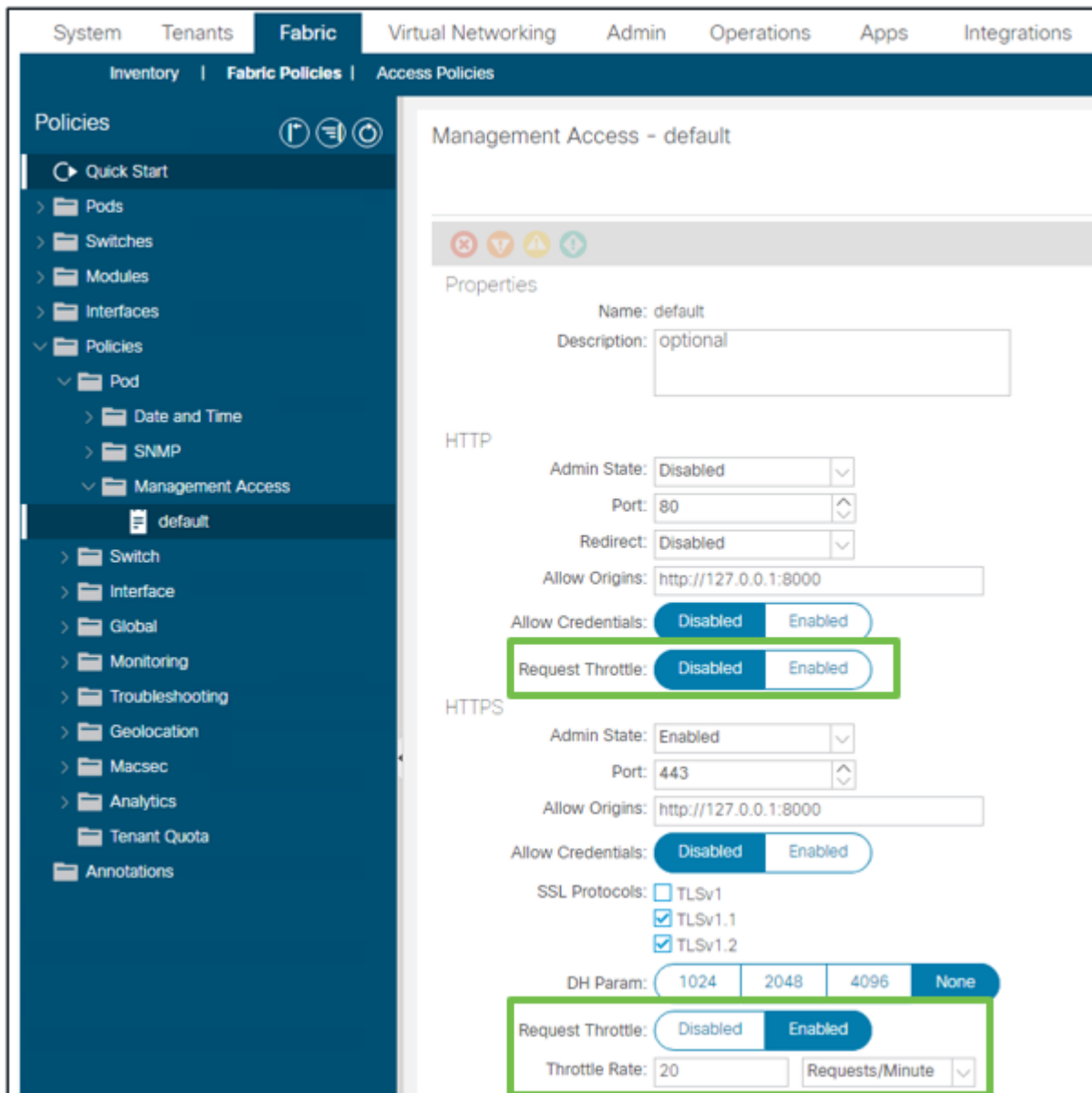
Ineffizienzen bei Adressskripten

- Melden Sie sich nicht vor jeder API-Anforderung an bzw. ab.
 - Das Standard-Timeout für eine Anmeldesitzung beträgt 10 Minuten. Dieselbe Sitzung kann für mehrere Anfragen verwendet und zur Verlängerung der Gültigkeitsdauer aktualisiert werden.
 - Siehe [Cisco APIC REST API Configuration Guide - Accessing the REST API - Authenticating and Maintaining an API Session](#).
- Wenn das Skript viele DNs abfragt, die ein übergeordnetes Objekt gemeinsam nutzen, statt die Abfragen in eine einzelne logische übergeordnete Abfrage mit [Abfragefiltern](#) zu reduzieren.
 - Siehe [Cisco APIC REST API-Konfigurationsleitfaden - Erstellen von REST API-Abfragen - Anwenden von Abfragebereichsfiltern](#).
- Wenn Sie Aktualisierungen für ein Objekt oder eine Objektklasse benötigen, [sollten Sie WebSocket-Abonnements](#) anstelle von schnellen API-Anforderungen [in Betracht ziehen](#).

NGINX-Anforderungsdrosselung

Ein Benutzer, der in Version 4.2(1)+ verfügbar ist, kann die Anforderungsdrosselung für HTTP und HTTPS unabhängig aktivieren.

 Hinweis: Ab ACI-Version 6.1(2) wurde die unterstützte maximale Rate für diese Funktion auf 40 Anfragen pro Sekunde (R/s) oder 2.400 Anfragen pro Minute (R/m) von 10.000 R/m reduziert.



Fabric - Fabric-Richtlinien - Richtlinienordner - Verwaltungszugriffsordner - Standard

Wenn aktiviert:

- NGINX wird neu gestartet, um Konfigurationsdateiänderungen anzuwenden.
 - Eine neue Zone, `httpsClientTagZone`, wird in die `nginx`-Konfiguration geschrieben.
- Die Drosselungsrate kann in Anfragen pro Minute (r/m) oder Anfragen pro Sekunde (r/s) festgelegt werden.
- Die Anfragedrosselung basiert auf der [Implementierung](#) der [Ratenbegrenzung in NGINX](#).
 - API-Anfragen für den `/api/URI` verwenden die benutzerdefinierte Drosselungsrate + Burst= (Drosselungsrate x 2) + nodelay
 - Es gibt eine nicht konfigurierbare Drossel (Zone `aaaApiHttps`) für `/api/aaaLogin` und `/api/aaaRefresh`, die Durchsatzbegrenzungen bei `2r/s + Burst=4 + nodelay` hat.
 - Request Throttle wird pro Client-IP-Adresse nachverfolgt.
 - API-Anfragen von der APIC-Self-IP (UI + CLI) umgehen die Drosselung.
 - Jede Client-IP-Adresse, die die benutzerdefinierte Drosselungsrate + den Burst-Schwellenwert überschreitet, erhält eine Antwort des APIC von 503.

- Diese 503s können in den Zugriffsprotokollen korreliert werden.
- error.log enthält Einträge, die angeben, wann die Drosselung aktiviert wurde (Zone httpsClientTagZone) und für welche Client-Hosts

```
<#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"
```

In der Regel dient Request Throttle nur dazu, den Server (APIC) vor DDoS-ähnlichen Symptomen zu schützen, die durch abfrageaggressive Clients hervorgerufen werden. Verstehen und isolieren Sie den anforderungsaggressiven Client für endgültige Lösungen in der App-/Skriptlogik.

Empfehlungen

Diese Empfehlungen sollen dazu beitragen, die Auslastung und die betriebliche Belastung des APIC zu reduzieren, insbesondere in Szenarien, in denen keine einzelne Quelle für ein hohes Volumen an API-Anrufen verantwortlich ist. Durch die Implementierung dieser Best Practices können Sie unnötige Verarbeitungsvorgänge, Protokollierungen und Ereignisgenerierungen in Ihrer Fabric minimieren und so die Systemstabilität und -leistung verbessern. Diese Vorschläge sind besonders in Umgebungen relevant, in denen aggregiertes Verhalten und keine isolierten Vorfälle zur APIC-Belastung beitragen.

ACL-Protokollierung deaktivieren

Stellen Sie sicher, dass die ACL-Protokollierung im Normalbetrieb ausgeschaltet ist. Aktivieren Sie diese Option nur in geplanten Wartungsfenstern für die Fehlerbehebung oder das Debuggen. Die fortlaufende Protokollierung kann zu übertriebenen Informationsmeldungen führen, insbesondere wenn es bei mehreren Switches zu einem Verfall des Datenverkehrs mit hohem Volumen kommt,

was die Arbeitslast des APIC erhöht.

Weitere Informationen finden Sie im Cisco APIC Security Konfigurationsleitfaden (Link zum 5.2.x-Leitfaden):

<https://www.cisco.com/c/en/us/td/docs/dcn/aci/apic/5x/security-configuration/cisco-apic-security-configuration-guide-release-52x/security-policies-52x.html>

Syslog-Konvertierung auf kritische Ereignisse beschränken

Konfigurieren Sie das System so, dass nur Syslog-Meldungen mit dem Schweregrad ALERT in eventRecords konvertiert werden. Vermeiden Sie die Konvertierung der Informationsebene (einschließlich ACL.logging), um zu verhindern, dass der APIC durch Geräuschereignisse überlastet wird:

1. Navigieren Sie zu Fabric → Fabric-Richtlinien → Richtlinien → Überwachung → Gemeinsame Richtlinie → Syslog-Nachrichtenrichtlinien → Standard.
2. Passen Sie den Einrichtungsfilter an, um den Syslog-Schweregrad auf Alert festzulegen.

Squelch-Ereigniscodes (nicht erforderlich)

Unterdrücken Sie Ereigniscodes (squelch), die nicht für Ihre Überwachungsanforderungen relevant sind, um Rauschen zu reduzieren.

Verwenden Sie den folgenden Befehl für eine beliebige APIC-CLI, um den Ereigniscode E4204939 zu löschen:

```
bash
icurl -k -sX POST -d '<fabricInst><monCommonPol><eventSevAsnP code="E4204939" sev="squelched"/></monCom
```

So überprüfen Sie:

```
bash
icurl -k -sX GET 'https://localhost/api/node/class/eventSevAsnP.xml' | xmllint --format -
```

Alternativ können Sie dies über die Benutzeroberfläche überprüfen:

Fabric > Fabric Policies > Policies > Monitoring > Common Policy > Event Severity Assignment Policy

Optimierung der ND-Abonnementaktualisierungen

Bei Fabrics, die von ND-Versionen vor 3.2.2m oder 4.1.1g verwaltet werden, sollten Sie auf eine dieser Versionen oder höher aktualisieren, um die Abonnementaktualisierungsintervalle zu optimieren. Ältere Versionen werden alle 45 Sekunden je MO aktualisiert, was skalierbar zu mehr als 300.000 APIC-Anfragen pro Tag führen kann. Aktualisierte Versionen erhöhen die Zeitüberschreitung für Abonnements auf 3.600 Sekunden (1 Stunde), wodurch die Anzahl der Aktualisierungen auf ca. 5.000 pro Tag reduziert wird.

Intersight-bezogene Abfragen überwachen

Intersight-fähige Fabrics generieren vom DC-Connector periodisch Topsystem-Abfragen (alle 15 Sekunden), was die APIC-Last erhöht.

Ab Version 6.1.2 wurde diese Abfrage optimiert, um den Overhead zu reduzieren.

Aufbewahrungsrichtlinien für Datensätze anpassen

Legen Sie die Aufbewahrungsrichtlinie für eventRecord, errorRecord und healthRecord auf 1.000 fest, um eine übermäßige Anhäufung von Datensätzen zu verhindern. Dies ist besonders nützlich, wenn Sie diese Datensätze regelmäßig für bestimmte betriebliche Aktivitäten extrahieren. Vergleichen Sie stets die Auswirkungen einer präziseren Überwachung mit Ihren Betriebs- und Fehlerbehebungsanforderungen.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.