Probleemoplossing voor een langzame APIC GUI

Inhoud

Inleiding

Dit document beschrijft de algemene methodologie voor probleemoplossing bij een langzame APIC GUI-ervaring.

Snelle start

Vaak wordt gevonden dat langzame APIC GUI-problemen het resultaat zijn van een hoog aantal API-aanvragen afkomstig van een script, integratie of toepassing. De access.log van een APIC registreert elk verwerkt API verzoek. De access.log van een APIC kan snel geanalyseerd worden met het Access Log Analyzer script binnen het Github Datacenter groep aci-tac-scripts project.

Achtergrondinformatie

APIC als webserver - NGINX

NGINX is de DME die verantwoordelijk is voor de API-endpoints die beschikbaar zijn op elke APIC. Als NGINX niet beschikbaar is, kunnen API-aanvragen niet worden verwerkt. Als NGINX verstopt is, is de API verstopt. Elke APIC heeft zijn eigen NGINX-proces, dus het is mogelijk dat slechts één APIC NGINX-problemen kan hebben als alleen die APIC wordt geviseerd door agressieve queriers.

De APIC UI voert meerdere API-verzoeken uit om elke pagina te vullen. Op dezelfde manier zijn alle APIC 'show' commando's (NXOS Style CLI) wrappers voor python scripts die meerdere API verzoeken uitvoeren,

de reactie behandelen, dan dienen het aan de gebruiker.

Relevante logbestanden

Logbestandsnaam	Location (Locatie)	Welke technische ondersteuning is het in	Opmerkingen
access.log	/var/log/dme/log	APIC 3of3	ACI-agnost, geeft 1 regel per API-aanvraag
error.log	/var/log/dme/log	APIC 3of3	ACI Agnostic, toont nginx fouten (throttling inbegrepen)
nginx.bin.log	/var/log/dme/log	APIC 3of3	ACI-specifiek, logt DME-transacties in
nginx.bin.warnplus.log	/var/log/dme/log	APIC 3of3	ACI-specifiek bevat logbestanden die waarschuwen+ ernst zijn

Methodologie

Eerste trigger isoleren

Wat is hiervan het gevolg?

- Welke APIC's zijn betrokken; één, veel of alle APIC's?
- Waar wordt traagheid gezien; via UI, CLI-opdrachten, of beide?
- Welke specifieke UI-pagina's of -opdrachten zijn traag?

Hoe wordt de traagheid ervaren?

- Is dit te zien over meerdere browsers voor één gebruiker?
- Melden meerdere gebruikers traagheid of slechts een enkele/subset van gebruikers?
- Delen de betrokken gebruikers een vergelijkbare geografische locatie of netwerkpad van browser naar APIC?

Wanneer werd de traagheid voor het eerst opgemerkt?

- Is er onlangs een ACI-integratie of -script toegevoegd?
- Is een browser extensie onlangs ingeschakeld?
- Was er een recente wijziging in de ACI-configuratie?

Gebruik en status van NGINX controleren

Indeling voor vermeldingen in Access.log

access.log is een functie van NGINX en is daarom APIC agnost. Elke regel staat voor 1 HTTP-verzoek dat de APIC heeft ontvangen. Verwijs naar dit logboek om het gebruik van NGINX van een APIC te begrijpen.

Het standaardformaat voor access.log op ACI versie 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"';
```

Deze lijn vertegenwoordigt een access.log ingang wanneer moquery -c fvTenant wordt uitgevoerd:

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

Kaart van voorbeeld access.log-vermelding naar log_format:

log_format veld	**Inhoud van voorbeeld**	**Opmerkingen**
$remote_addr	127.0.0.1	IP van host die dit verzoek heeft verzonden
$http_x_real_ip	-	IP van de laatste aanvrager indien proxyâ€™s in gebruik zijn
$remote_user	-	Niet algemeen gebruikt. Controleer nginx.bin.log welke gebruiker ingelogd is om verzoeken uit te voeren
$tijd_lokaal	07/apr/2022:20:10:59 +000	Toen het verzoek werd verwerkt
$request	ONTVANG /api/class/fvTenant.xml HTTP/1.1	HTTP-methode (GET, POST, DELETE) en URI
$status	200	HTTP-responsstatuscode
$body_bytes_verzonden	1586	grootte van antwoordlading
$http_referer	-	-

$http_user_agent	Python-urllib	Welk type client heeft de aanvraag verzonden?

Gedrag van Access.log

Het hoge tarief verzoek barst over een grote periode:

- Doorlopende uitbarstingen van 15+ verzoeken per seconde kunnen vertraging van gebruikersinterface veroorzaken
- Identificeer welke host(s) verantwoordelijk zijn voor de vragen
- Verminder of blokkeer de bron van de vragen om te zien of dit de APIC-responstijd verbetert.

Consistente 4xx- of 5xx-reacties:

- Indien gevonden, identificeer de foutmelding van nginx.bin.log

NGINX-resourcegebruik controleren

NGINX CPU en geheugengebruik kunnen worden gecontroleerd met de **bovenste** opdracht van de 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

Hoog NGINX-brongebruik kan direct correleren met een hoog percentage verwerkte aanvragen.

Op kernen controleren

Een NGINX crash is niet typisch voor Slow APIC GUI problemen. Als echter NGINX-kernen worden gevonden, kunt u deze voor analyse aan een TAC SR bevestigen. Raadpleeg de ACI Techsupport handleiding voor stappen om te controleren of er kernen zijn.

Latentie van client naar server controleren

Als de snelle verzoeken niet worden gevonden maar een gebruiker blijft tonen UI traagheid, kan het probleem Cliënt (browser) aan Server (APIC) latentie zijn.

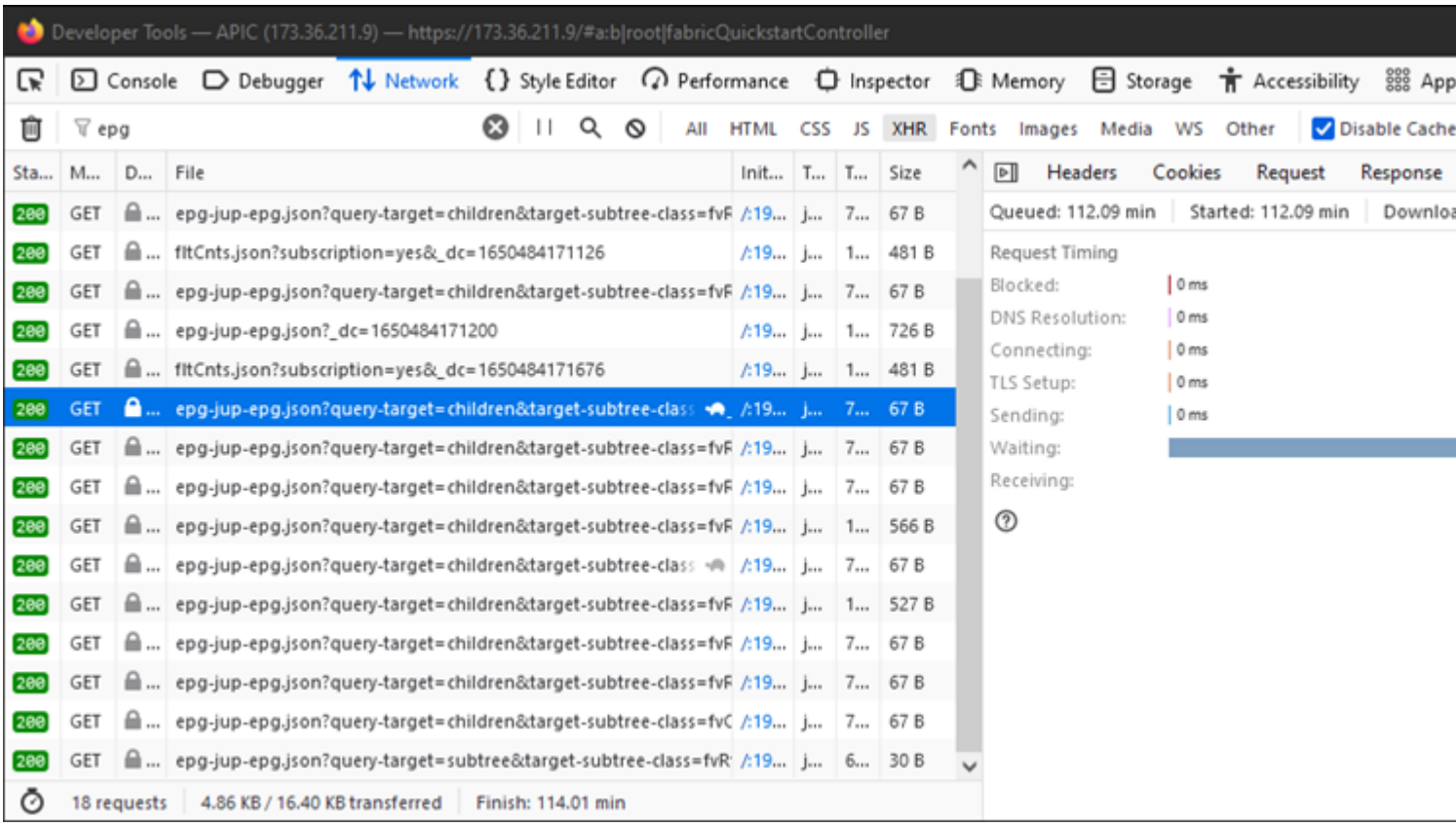
In deze scenario's, valideer het gegevenspad van browser aan APIC (Geografische afstand, VPN, enz.). Indien mogelijk, implementeer en test toegang vanaf een sprongserver in dezelfde geografische regio of

datacenter als de APIC's om te isoleren. Valideren als andere gebruikers een vergelijkbare hoeveelheid latentie vertonen.

Tabblad Browser Development Tools voor netwerk

Alle browsers hebben de mogelijkheid om HTTP-verzoeken en antwoorden te valideren via de **Browser Development** toolkit, meestal binnen een tabblad **Network**.

Deze tool kan worden gebruikt om de hoeveelheid tijd te valideren die nodig is voor elke fase van browsergebaseerde verzoeken zoals getoond in de afbeelding.



Voorbeeld van de browser wachten 1,1 minuten voor de APIC om te reageren

Verbeteringen voor specifieke UI-pagina's

Pagina Beleidsgroep:

Cisco bug-id CSCvx14621 - APIC GUI laadt langzaam op IPG-beleid in het tabblad Fabric.

Interface onder Inventarispagina:

Cisco bug-id CSCvx90048 - Eerste lading van "Layer 1 Physical Interface Configuration" Operationeel tabblad is lang/induceert 'bevriezen'.

Algemene aanbevelingen voor client > serverlatentie

Bepaalde browsers, zoals Firefox, staan voor meer webverbindingen per host standaard toe.

- Controleer of deze instelling configureerbaar is voor de gebruikte browserversie
- Dit is meer van belang voor multiquery-pagina's, zoals de pagina Policy Group

VPN en afstand tot APIC verhogen algemene UI traagheid gegeven client browser verzoeken en APIC respons reistijd. Een springbox geografisch lokaal aan de APIC's reduceert de browser aanzienlijk tot APIC reistijden.

Controleer op lange webaanvragen

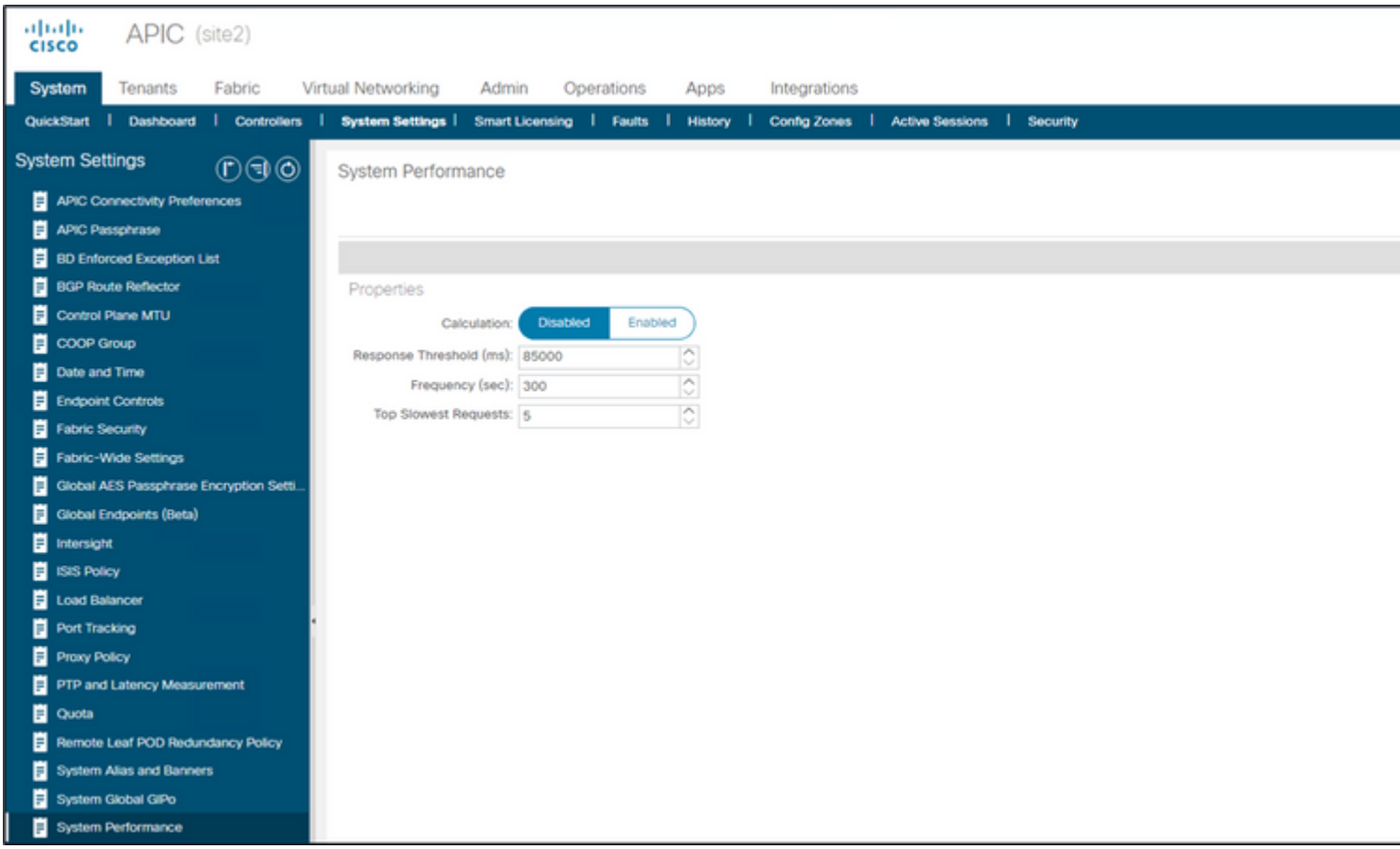
Als een Webserver (NGINX op APIC) een hoog volume van Long-Web Verzoeken behandelt, kan dit de prestaties van andere verzoeken beïnvloeden die parallel worden ontvangen.

Dit geldt in het bijzonder voor systemen die databases hebben gedistribueerd, zoals APIC's. Een enkele API-

aanvraag kan extra verzoeken en zoekopdrachten vereisen die naar andere knooppunten in de stof worden verzonden en die kunnen resulteren in vermoedelijk langere reactietijden. Een uitbarsting van deze Long-Web Verzoeken binnen een klein tijdsbestek kan de hoeveelheid benodigde middelen en leiden tot onverwacht langere reactietijden. Bovendien kunnen ontvangen verzoeken dan uittijd (90 seconden) wat resulteert in onverwacht systeemgedrag vanuit een gebruikersperspectief.

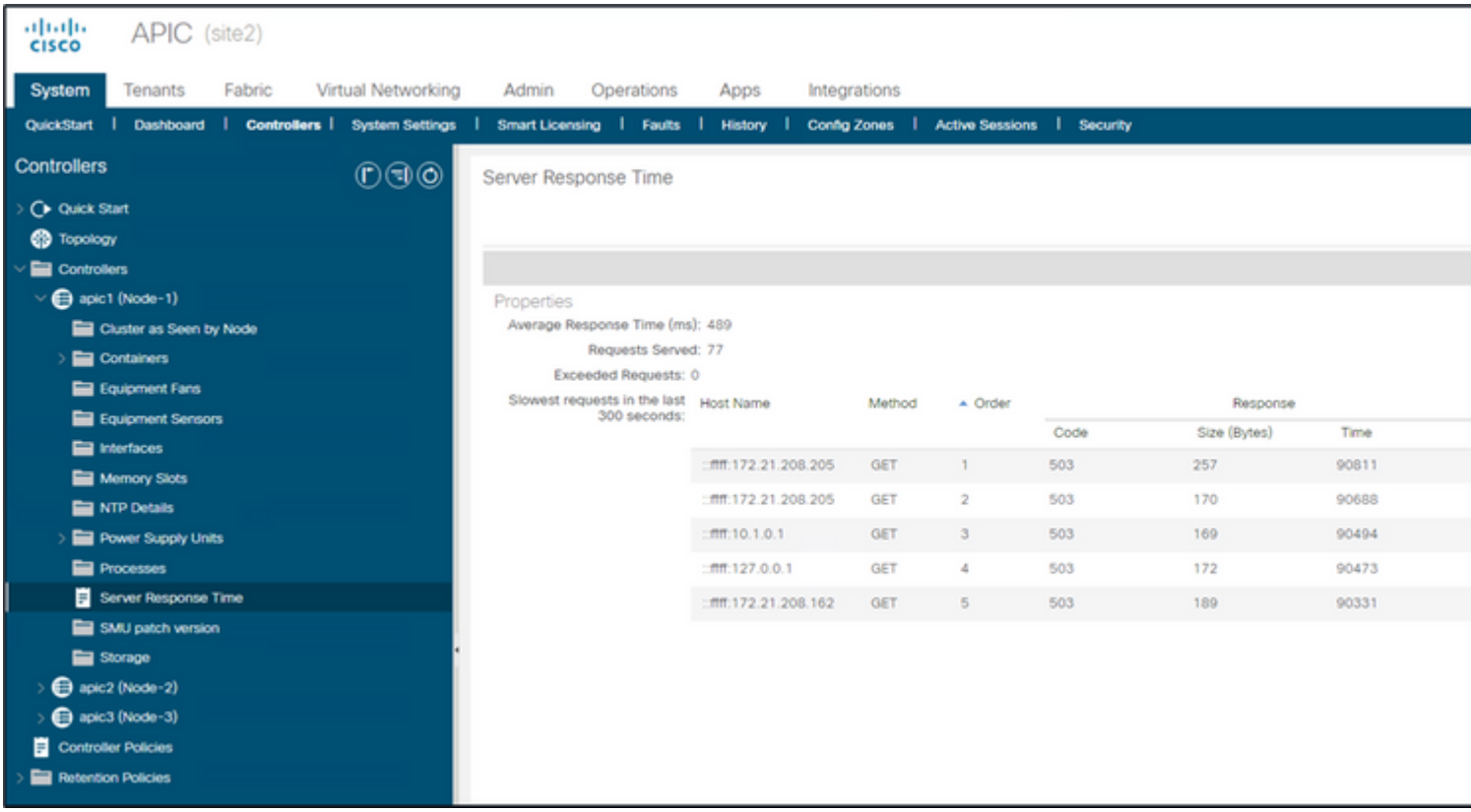
Systeemresponstijd - Berekening voor responstijd van server inschakelen

In 4.2(1)+ kan een gebruiker "System Performance Calfication" inschakelen, dat API-verzoeken bijhoudt en markeert die lang hebben geduurd.



Berekening kan worden ingeschakeld vanuit Systeem - Systeeminstellingen - Systeemprestaties

Als "Berekening" is ingeschakeld, kan een gebruiker naar specifieke APIC's onder Controllers navigeren om de langzaamste API-aanvragen in de afgelopen 300 seconden te bekijken.



Systeem - controllers - map voor controllers - APIC x - Server Response Time

Overwegingen bij gebruik van APIC API

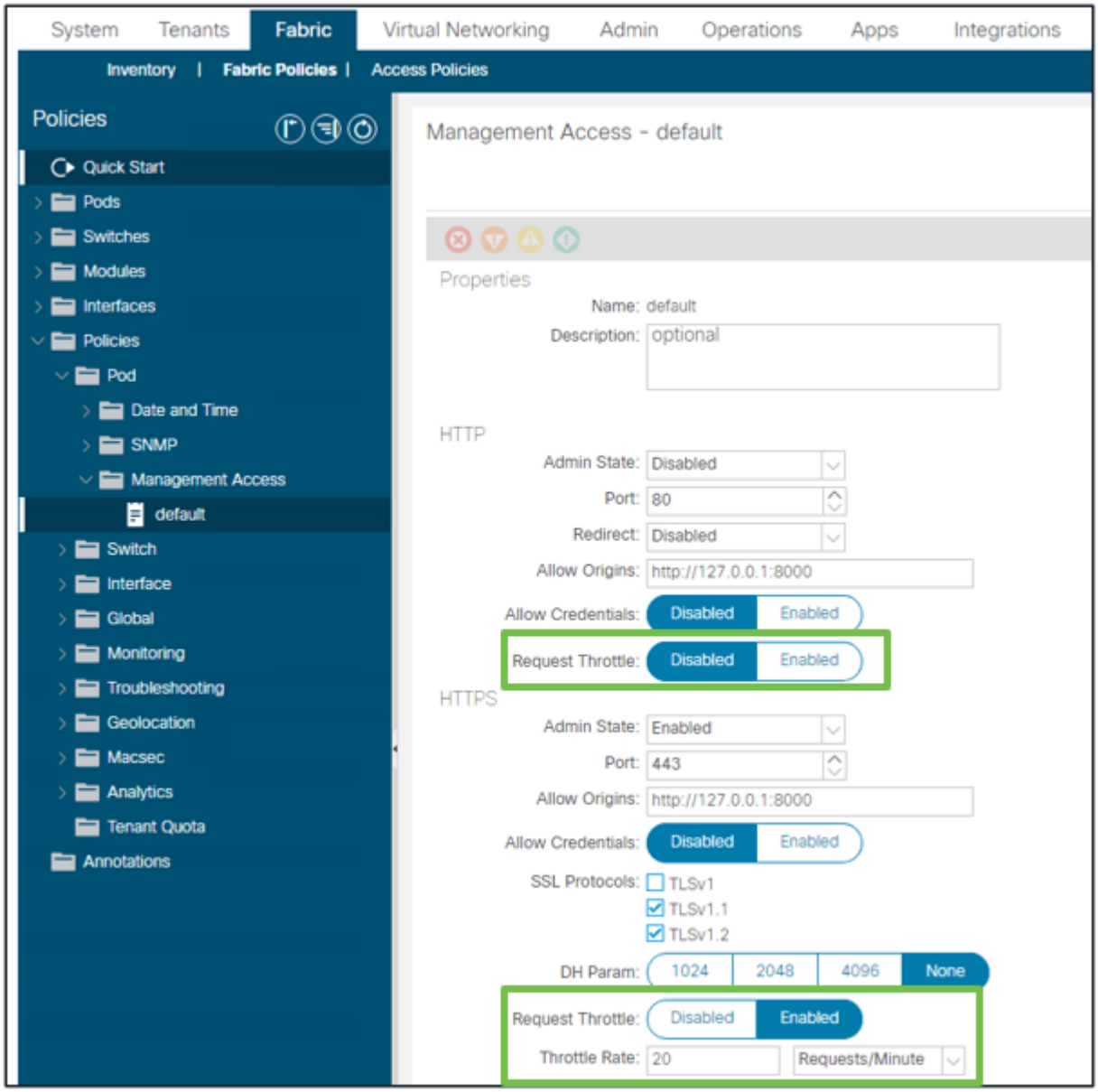
- In het algemeen, 15+ API verzoeken per seconde over een lange periode van tijd debiliteert NGINX.
    - Als dit wordt gevonden, vermindert u de agressiviteit van de aanvragen.
    - Als de Verzoeken-host niet kan worden aangepast, overweeg dan NGINX Rate Limits op de APIC.

Inefficiënties in adresscripts

- Log niet in/log uit voor elke API-aanvraag.
    - De standaard timeout voor een login sessie is 10 minuten. Deze sessie kan gebruikt worden voor meerdere aanvragen en kan worden ververst om de geldigheidstijd te verlengen.
    - Zie Cisco APIC REST API Configuration Guide - Access the REST API - Verifiëren en onderhouden van een API-sessie.
- Als uw script veel DN's die een parent delen opvraagt, in plaats van de query samen te vouwen in één logische parent query met Query Filters.
    - Zie Cisco APIC REST API Configuration Guide - Composite REST API Queries - Appliance Query Scoping Filters.
- Als u updates van een object of objectklasse nodig hebt, overweeg dan websocket-abonnementen in plaats van snelle API-aanvragen.

NGINX-verzoekgaspedaal

Verkrijgbaar in 4.2(1)+, kan een gebruiker een aanvraagthrottle inschakelen tegen HTTP en HTTPS onafhankelijk.



Fabric - Fabric Policies - Policy Folder - Management Access Folder - standaard

Als deze optie is ingeschakeld:

- NGINX wordt opnieuw gestart om wijzigingen in configuratiebestanden toe te passen
  - Een nieuwe zone, **httpsClientTagZone**, is geschreven naar nginx config
- De Throttle kan worden ingesteld in **Verzoeken per minuut** (r/m) of **Verzoeken per seconde** (r/s).
- Request Throttle is afhankelijk van de [Rate Limit Implementatie inbegrepen in NGINX]
  - API-aanvragen tegen de **/api/**URI maken gebruik van de door de gebruiker gedefinieerde Throttle Rate + burst= (Throttle Rate x 2) + nodelag
    - Er is een niet-configureerbare gasklep (zone **aaaApiHttps**) voor **/api/aaaLogin** en **/api/aaaRefresh** die snelheidslimieten bij 2r/s + burst=4 + nodelag
  - De aanvraagbeperking wordt per client-ip-adres bijgehouden
  - API-verzoeken afkomstig van de APIC self-ip (UI + CLI) omzeilen de throttle
  - Elke client-IP-adres dat de door de gebruiker gedefinieerde throttle rate + burst-drempel overschrijdt, ontvangt een 503-antwoord van de APIC
  - Deze 503s kunnen worden gecorreleerd binnen de toegangslogboeken
  - error.log bevat vermeldingen die aangeven wanneer throttling is geactiveerd (zone **httpsClientTagZone**) en waartegen 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 de regel dient request throttle alleen om de server (APIC) te beschermen tegen DDOS-achtige symptomen die worden geïnduceerd door query-aggresive Clients. Begrijp en isoleer de vraag-agressieve Klant voor definitieve oplossingen in de app/scriptlogica.

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.