Dépannage d'une interface graphique APIC lente

Table des matières

Introduction

Ce document décrit la méthodologie générale de dépannage d'une interface utilisateur graphique APIC lente.

Démarrage rapide

Il est fréquent que les problèmes lents de l'interface graphique utilisateur du contrôleur APIC soient le résultat d'un taux élevé de requêtes API provenant d'un script, d'une intégration ou d'une application. Le fichier access.log d'un contrôleur APIC consigne chaque demande d'API traitée. Le fichier access.log d'un APIC peut être rapidement analysé avec le script Access Log Analyzer dans le projet aci-tac-scripts du groupe Github Datacenter.

Informations générales

APIC en tant que serveur Web - NGINX

NGINX est le DME responsable des terminaux API disponibles sur chaque APIC. Si NGINX est en panne, les demandes d'API ne peuvent pas être traitées. Si NGINX est encombré, l'API l'est également. Chaque APIC exécute son propre processus NGINX, il est donc possible qu'un seul APIC puisse avoir des problèmes NGINX si seulement cet APIC est ciblé par des interrogateurs agressifs.

L'interface utilisateur APIC exécute plusieurs requêtes API pour remplir chaque page. De même, toutes les commandes APIC « show » (NXOS Style CLI) sont des wrappers pour les scripts python qui exécutent plusieurs requêtes API, gèrent la réponse, puis la servent à l'utilisateur.

Journaux pertinents

Nom du fichier journal	Emplacement	Dans quel support technique se trouve-t-il ?	Commentaires
access.log	/var/log/dme/log	Carte APIC 3 sur 3	Indépendant de l'ACI, 1 ligne par requête API
error.log	/var/log/dme/log	Carte APIC 3 sur 3	ACI agnostique, affiche les erreurs nginx (limitation incluse)
nginx.bin.log	/var/log/dme/log	Carte APIC 3 sur 3	Spécifique à l'ACI, consigne les transactions DME
nginx.bin.warnplus.log	/var/log/dme/log	Carte APIC 3 sur 3	L'ACI spécifique contient des journaux d'avertissement et de gravité

Méthode

Isoler le déclencheur initial

Quelles sont les conséquences ?

- Quels sont les APIC concernés : un, plusieurs ou tous les APIC ?
- Où la lenteur se manifeste-t-elle, via l'interface utilisateur, les commandes CLI ou les deux ?
- Quelles pages ou commandes de l’interface utilisateur sont lentes ?

Comment la lenteur est-elle ressentie ?

- Cela s'applique-t-il à plusieurs navigateurs pour un seul utilisateur ?

- Plusieurs utilisateurs signalent-ils une lenteur ou un seul sous-ensemble d'utilisateurs ?
- Les utilisateurs concernés partagent-ils un emplacement géographique ou un chemin réseau similaire du navigateur au contrôleur APIC ?

Quand la lenteur a-t-elle été remarquée ?

- Une intégration ACI ou un script ont-ils été ajoutés récemment ?
- Une extension de navigateur a-t-elle été activée récemment ?
- La configuration de l'ACI a-t-elle été modifiée récemment ?

Vérifier l'utilisation et l'intégrité de NGINX

Format d'entrée Access.log

access.log est une fonctionnalité de NGINX et est, par conséquent, agnostique APIC. Chaque ligne représente 1 requête HTTP reçue par le contrôleur APIC. Consultez ce journal pour comprendre l'utilisation NGINX d'un APIC.

Format access.log par défaut sur la version 5.2+ de l'ACI :

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

Cette ligne représente une entrée access.log lorsqu'une moquery -c fvTenant est exécutée :

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

Mappage de l'entrée access.log de l'exemple avec log_format :

Champ log_format	Contenu de l'exemple	Commentaires
$remote_addr	Commutateurs 127.0.0.1	Adresse IP de l'hôte qui a envoyé cette demande
$http_x_real_ip	-	IP du dernier demandeur si des proxys sont utilisés

$remote_user	-	Généralement pas utilisé. Consultez le fichier nginx.bin.log pour savoir quel utilisateur s'est connecté pour effectuer des requêtes
$time_local	07/Avr/2022:20:10:59 +0000	Quand la demande a été traitée
$request	GET /api/class/fvTenant.xml HTTP/1.1	Méthode Http (GET, POST, DELETE) et URI
$status	200	Code d'état de réponse HTTP
$body_bytes_sent	1586	taille de la charge utile de réponse
$http_referer	-	-
$http_user_agent	Python-urllib	Quel type de client a envoyé la demande ?

Comportements Access.log

Rafales de demandes à haut débit sur une longue période :

- Des rafales continues de plus de 15 requêtes par seconde peuvent entraîner une lenteur de l'interface utilisateur
- Identifier le ou les hôtes responsables des requêtes
- Réduisez ou désactivez la source des requêtes pour voir si cela améliore le temps de réponse APIC.

Réponses 4xx ou 5xx cohérentes :

- S'il est trouvé, identifiez le message d'erreur de nginx.bin.log

Vérifier l'utilisation des ressources NGINX

L'utilisation du processeur NGINX et de la mémoire peut être vérifiée avec la commande top de l'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

Une utilisation élevée des ressources NGINX peut être directement corrélée à un taux élevé de demandes traitées.

Vérification des coeurs

Une panne NGINX n'est pas typique pour les problèmes de GUI Slow APIC. Toutefois, si des coeurs NGINX sont trouvés, les joindre à un TAC SR pour analyse. Reportez-vous au guide ACI Techsupport pour connaître les étapes de vérification des coeurs.

Vérifier la latence client-serveur

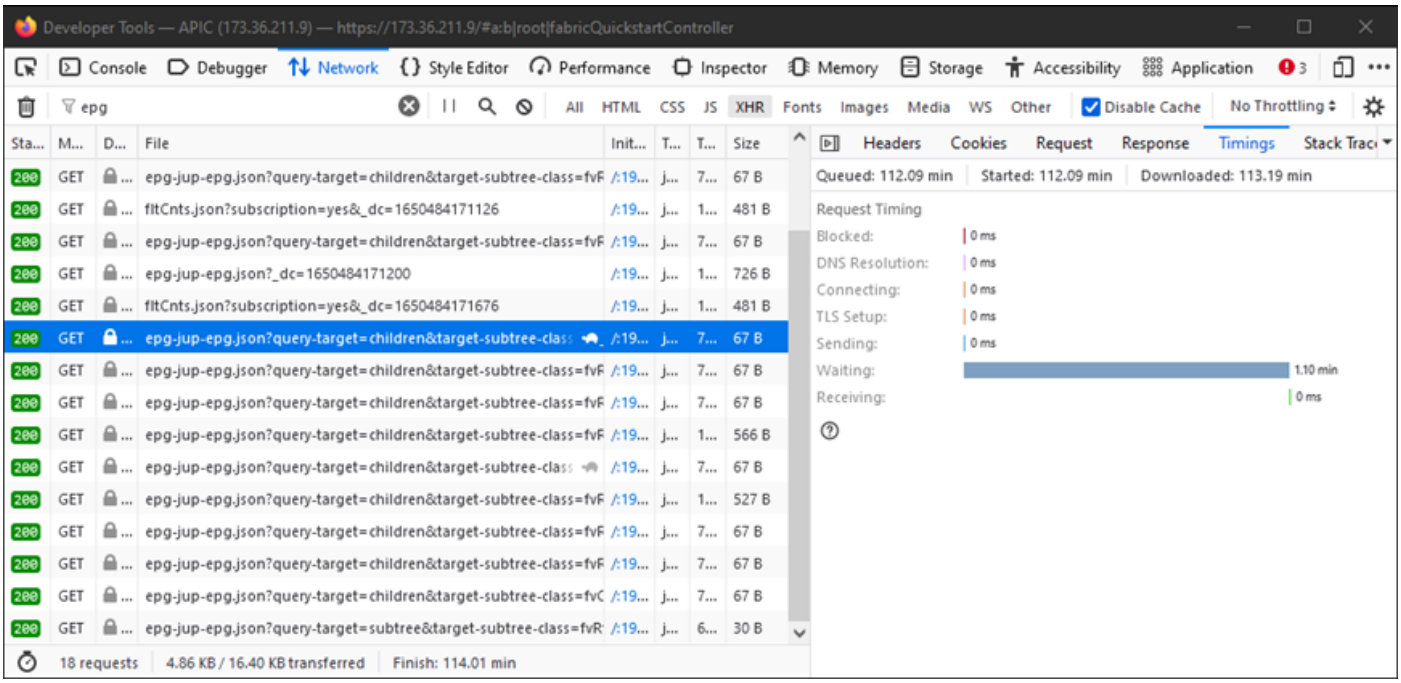
Si aucune requête rapide n'est trouvée mais qu'un utilisateur continue à présenter une lenteur de l'interface utilisateur, le problème peut être la latence entre le client (navigateur) et le serveur (APIC).

Dans ces scénarios, validez le chemin des données du navigateur vers le contrôleur APIC (distance géographique, VPN, etc.). Si possible, déployez et testez l'accès à partir d'un serveur de saut situé dans la même région géographique ou le même data center que les APIC à isoler. Vérifier si d'autres utilisateurs présentent une latence similaire.

Onglet Réseau des outils de développement de navigateur

Tous les navigateurs peuvent valider les requêtes et les réponses HTTP via leur boîte à outils Browser Development, généralement dans un onglet Network.

Cet outil peut être utilisé pour valider le temps nécessaire à chaque étape des requêtes provenant d'un navigateur, comme le montre l'image.



Exemple de navigateur en attente de réponse du contrôleur APIC pendant 1,1 minute

Améliorations pour des pages d'interface utilisateur spécifiques

Page Groupe de stratégies :

ID de bogue Cisco CSCvx14621 - L'interface graphique du contrôleur APIC se charge lentement sur les stratégies IPG dans l'onglet Fabric.

Interface sous la page Inventaire :

ID de bogue Cisco CSCvx90048 - Le chargement initial de l'onglet opérationnel « Configuration de l'interface physique de couche 1 » est long/provoque un « gel ».

Recommandations générales pour Client > Latence du serveur

Certains navigateurs, tels que Firefox, permettent par défaut davantage de connexions Web par hôte.

- Vérifiez si ce paramètre est configurable sur la version du navigateur utilisée
- Cela est plus important pour les pages à requêtes multiples, telles que la page Groupe de stratégies

Le VPN et la distance au contrôleur APIC augmentent la lenteur globale de l'interface utilisateur en fonction des demandes du navigateur client et du temps de réponse du contrôleur APIC. Une zone de saut géographiquement locale aux APIC réduit considérablement les temps de déplacement du navigateur vers l'APIC.

Vérifier les requêtes Long-Web

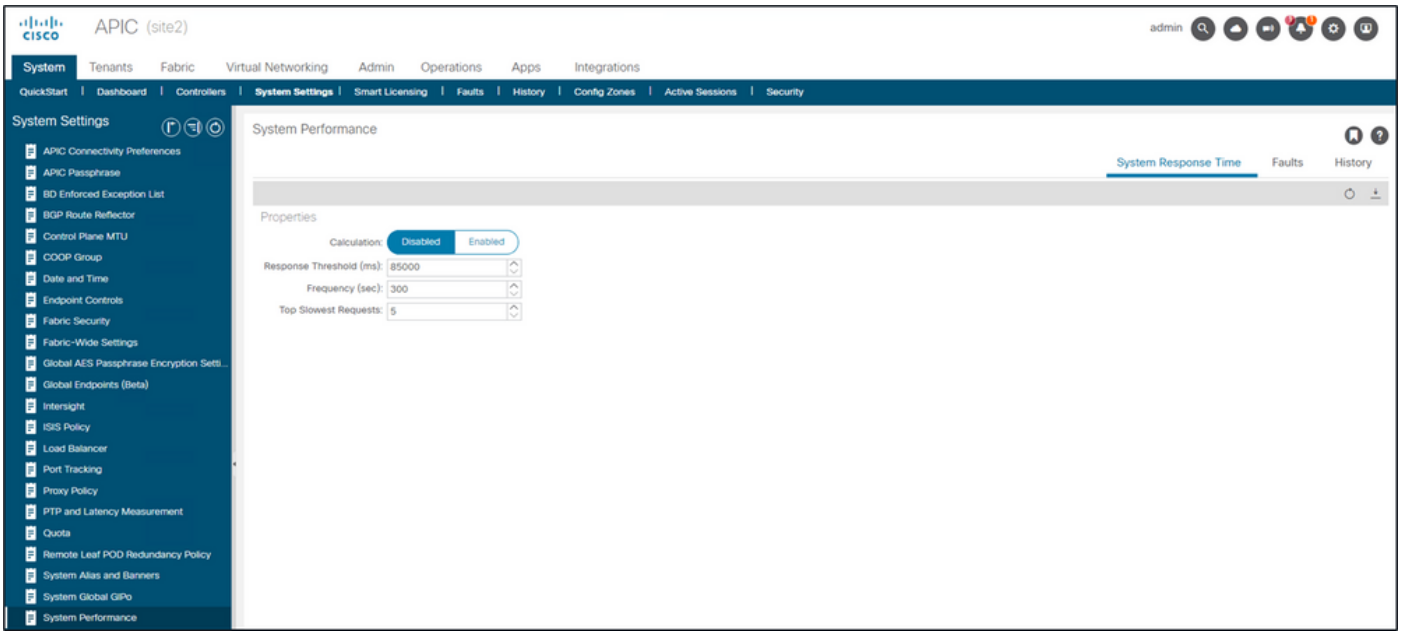
Si un serveur Web (NGINX sur APIC) gère un volume élevé de requêtes Long-Web, cela peut

affecter les performances des autres requêtes reçues en parallèle.

Ceci est particulièrement vrai pour les systèmes qui ont des bases de données distribuées, comme les APIC. Une seule requête API peut nécessiter des requêtes et des recherches supplémentaires envoyées à d'autres noeuds du fabric, ce qui peut entraîner des temps de réponse plus longs. Une rafale de ces requêtes Long-Web dans un laps de temps réduit peut augmenter la quantité de ressources requises et entraîner des temps de réponse plus longs que prévu. En outre, les demandes reçues peuvent alors expirer (90 secondes), ce qui entraîne un comportement inattendu du système du point de vue de l'utilisateur.

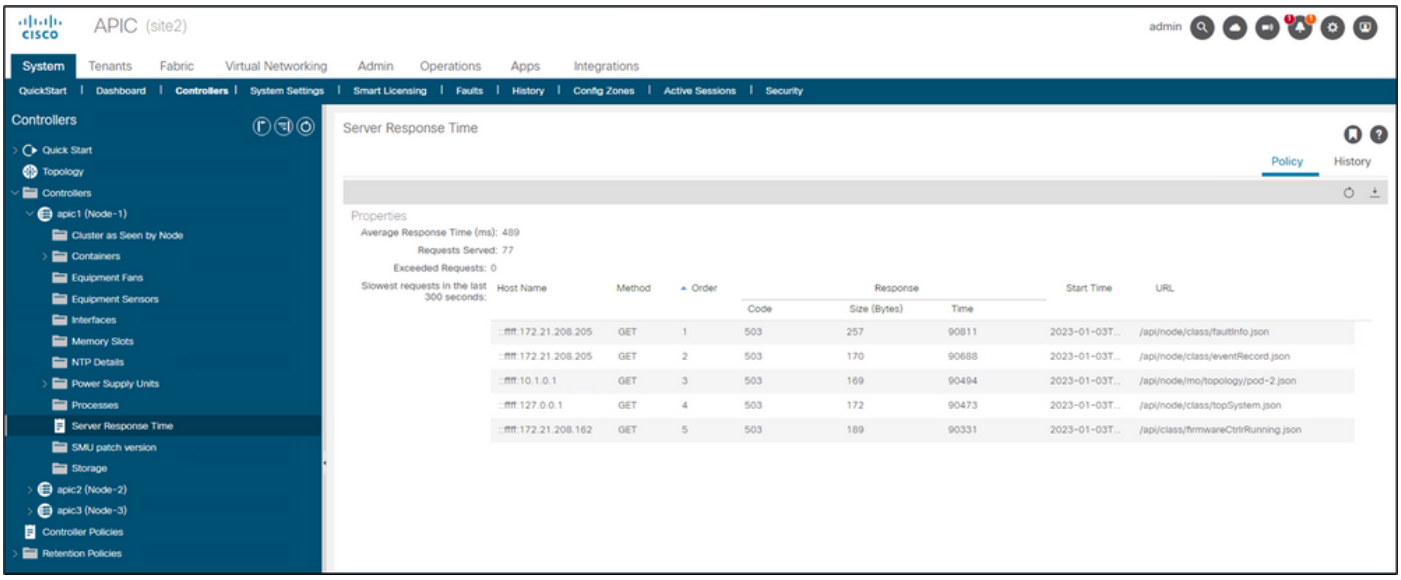
Temps de réponse du système - Activer le calcul pour le temps de réponse du serveur

Dans la version 4.2(1)+, un utilisateur peut activer le « Calcul des performances du système » qui suit et met en surbrillance les demandes d'API dont le traitement a pris du temps.



Le calcul peut être activé à partir de Système - Paramètres système - Performances système

Une fois le « calcul » activé, un utilisateur peut naviguer jusqu'à des APIC spécifiques sous Contrôleurs pour afficher les requêtes API les plus lentes au cours des 300 dernières secondes.



Système - Contrôleurs - Dossier Contrôleurs - APIC x - Temps de réponse du serveur

Utilisation de l'API APIC

Pointeurs généraux pour s'assurer qu'un script n'endommage pas Nginx

- Chaque APIC exécute son propre DME NGINX.
  - Seul le NGINX du contrôleur APIC 1 traite les demandes adressées au contrôleur APIC 1. Le NGINX des APIC 2 et 3 ne traite pas ces demandes.
- En général, plus de 15 requêtes API par seconde sur une longue période de temps affaiblissent NGINX.
  - S'il est détecté, réduisez l'agressivité des demandes.
  - Si l'hôte Requests ne peut pas être modifié, considérez NGINX Rate Limits sur l'APIC.

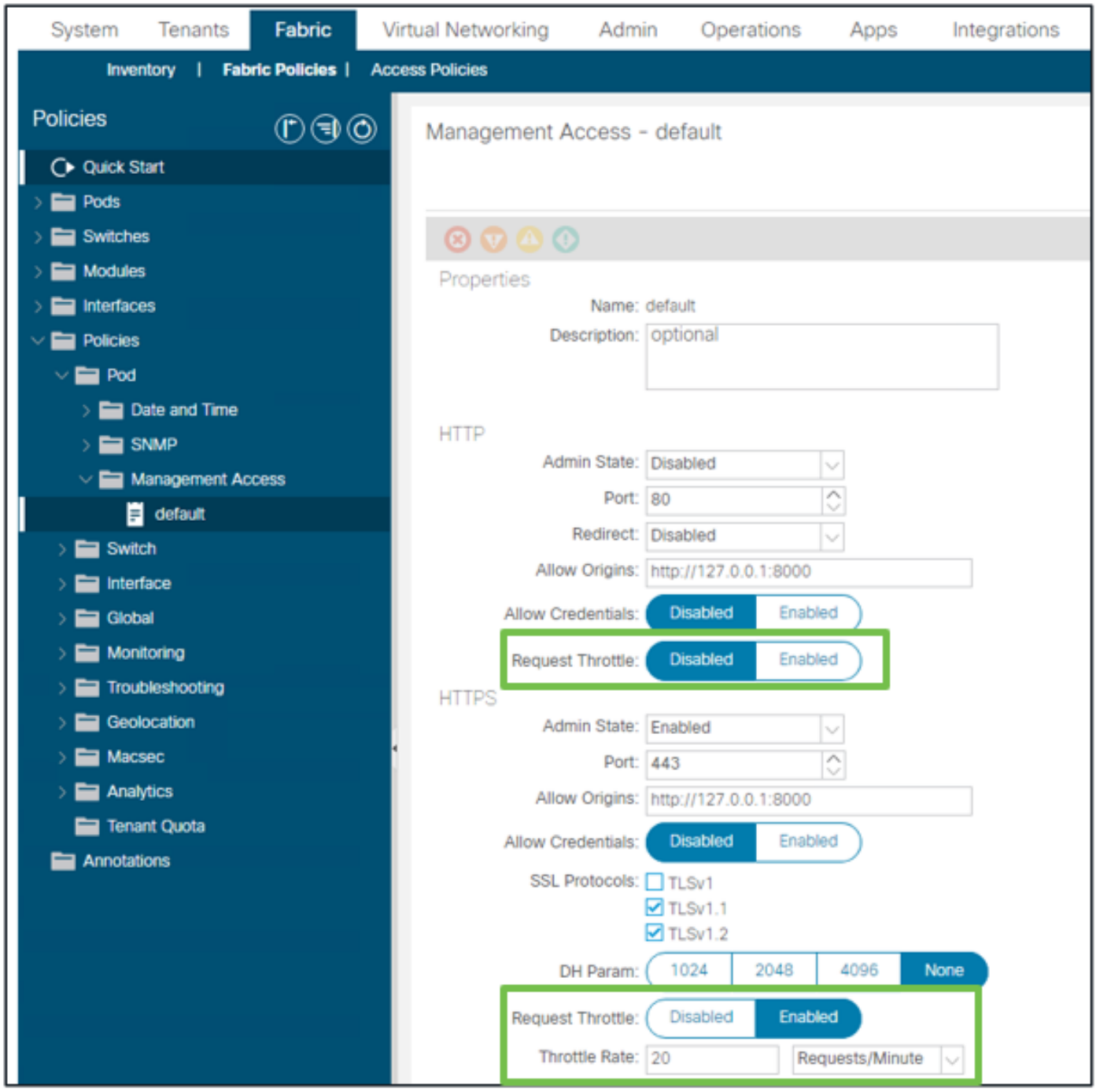
Résolution des inefficacités des scripts

- Ne vous connectez pas et ne vous déconnectez pas avant chaque demande d'API.
  - Le délai d'attente par défaut pour une session est de 10 minutes. Cette même session peut être utilisée pour plusieurs demandes et peut être actualisée pour prolonger la durée de validité.
  - Reportez-vous au Guide de configuration de l'API REST de Cisco APIC - Accès à l'API REST - Authentification et maintenance d'une session API.
- Si votre script interroge de nombreux DN qui partagent un parent, au lieu de réduire les requêtes en une seule requête parent logique avec des filtres de requête.
  - Voir Guide de configuration de l'API REST du contrôleur APIC Cisco - Composer des requêtes d'API REST - Application de filtres de portée de requête.
- Si vous avez besoin de mises à jour d'un objet ou d'une classe d'objets, envisagez des abonnements websocket plutôt que des requêtes API rapides.

NGINX Request Throttle

Disponible dans la version 4.2(1)+, un utilisateur peut activer la limitation de requête

indépendamment des protocoles HTTP et HTTPS.



Fabric - Stratégies de fabric - Dossier Stratégies - Dossier Accès à la gestion - par défaut

Lorsque cette option est activée :

- NGINX est redémarré pour appliquer les modifications du fichier de configuration
  - Une nouvelle zone, httpsClientTagZone, est écrite dans la configuration nginx
- Le taux d'étranglement peut être défini dans Demandes par minute (r/m) ou Demandes par seconde (r/s).
- Request Throttle repose sur l'implémentation de la limite de débit incluse dans NGINX
  - Les requêtes API sur l'/api/URI utilisent le taux d'étranglement défini par l'utilisateur + burst= (taux d'étranglement x 2) + nodelay
    - Il y a un étranglement non configurable (zone aaaApiHttps) pour /api/aaaLogin et

/api/aaaRefresh qui limite le débit à 2r/s + burst=4 + nodelay

- Le contrôle des demandes est effectué par adresse IP de client.
- Les requêtes API provenant de l'interface APIC self-ip (UI + CLI) contournent la limitation
- Toute adresse IP de client qui dépasse le débit d'étranglement défini par l'utilisateur + seuil de salve reçoit une réponse 503 du contrôleur APIC
- Ces 503 peuvent être corrélés dans les journaux d'accès
- error.log aura des entrées indiquant quand la limitation a été activée (zone httpsClientTagZone) et par rapport à laquelle les hôtes du 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"
```

En règle générale, Request Throttle sert uniquement à protéger le serveur (APIC) contre les symptômes de type DDOS induits par les clients agressifs vis-à-vis des requêtes. Comprendre et isoler le client exigeant des solutions finales dans la logique app/script.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.