测试文档TOC

目录

简介

本文档介绍排除APIC GUI体验缓慢故障的一般方法。

快速入门

经常发现，APIC GUI问题缓慢是由于来自脚本、集成或应用的API请求速率较高造成的。APIC的access.log记录每个已处理的API请求。使用Github Datacenter组aci-tac-scripts项目中的访问日志分析器脚本，可以快速分析APIC的access.log。

背景信息

APIC作为Web服务器 — NGINX

NGINX是负责每个APIC上可用的API终端的DME。如果NGINX关闭，则无法处理API请求。如果NGINX拥塞，则API拥塞。每个APIC运行自己的NGINX进程，因此如果任何主动式查询器仅针对某个APIC，则可能只有一个APIC会出现NGINX问题。

APIC UI执行多个API请求以填充每个页面。同样，所有APIC show命令(NXOS Style CLI)都是执行多个API请求、处理响应然后提供给用户的python脚本的包装程序。

相关日志

日志文件名	位置	它位于哪个技术支持部门	备注
access.log	/var/log/dme/log	APIC 3of3	与ACI无关，每个API请求提供1行
error.log	/var/log/dme/log	APIC 3of3	ACI不可知性，显示nginx错误（包括限制）
nginx.bin.log	/var/log/dme/log	APIC 3of3	ACI特定，记录DME事务
nginx.bin.warnplus.log	/var/log/dme/log	APIC 3of3	ACI特定包含警告+严重性的日志

方法

隔离初始触发器

哪些方面受到影响？

- 哪些APIC受到影响；一个、多个或所有APIC?
- 哪里可以看到迟缓；是通过UI、CLI命令还是同时使用两者？
- 哪些特定的UI页面或命令速度较慢？

如何体验低速？

- 单个用户是否可在多个浏览器中看到这种情况？
- 多个用户是否报告速度缓慢？还是仅报告单个/子集用户？
- 受影响用户是否共享从浏览器到APIC的类似地理位置或网络路径？

缓慢的第一次注意到是在什么时候？

- 最近是否添加了ACI集成或脚本？
- 最近是否启用了浏览器扩展？
- 最近是否更改了ACI配置？

检查NGINX使用情况和运行状况

Access.log条目格式

access.log是NGINX的一项功能，因此与APIC无关。每行代表APIC收到的1个HTTP请求。参考此日志以了解APIC的NGINX使用情况。

ACI版本5.2+上的默认access.log格式：

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

此行表示执行moquery -c fvTenant时的access.log条目：

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

示例access.log条目映射到log_format:

log_format字段	示例内容	备注
$remote_addr	127.0.0.1	发送此请求的主机的IP
$http_x_real_ip	-	使用代理时最后一个请求者的IP
$remote_user	-	一般不使用。选中nginx.bin.log以跟踪登录执行请求的用户
$time_local	2022年4月07日：20:10:59 +0000	处理请求时
$request	获取/api/class/fvTenant.xml HTTP/1.1	Http方法(GET、POST、DELETE)和URI
$status	200	HTTP响应状态代码
$body_bytes_sent	1586	响应负载大小
$http_reference	-	-
$http_user_agent	Python-urllib	发送请求的客户端类型

Access.log行为

在较长时间内高速突发请求：

- 每秒40多个请求的持续突发会导致用户界面缓慢
- 确定负责查询的主机
- 减少或禁用查询源，以查看这是否能缩短APIC响应时间。

一致的4xx或5xx响应：

- 如果找到，请识别来自nginx.bin.log的错误消息

使用Github Datacenter组aci-tac-scripts项目中的访问日志分析器脚本，可以快速分析APIC的access.log。

检查NGINX资源使用情况

可以使用APIC中的top命令检查NGINX CPU和内存使用情况：

<#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

高NGINX资源使用率可以直接与高处理率请求相关联。

检查内核

NGINX崩溃不常用于慢速APIC GUI问题。但是，如果找到NGINX核心，请将其附加到TAC SR进行分析。有关检查核心的步骤，请参阅ACI技术支持指南。

检查客户端到服务器的延迟

如果找不到快速请求，但用户继续表现出UI缓慢，则问题可能是客户端（浏览器）到服务器(APIC)延迟。

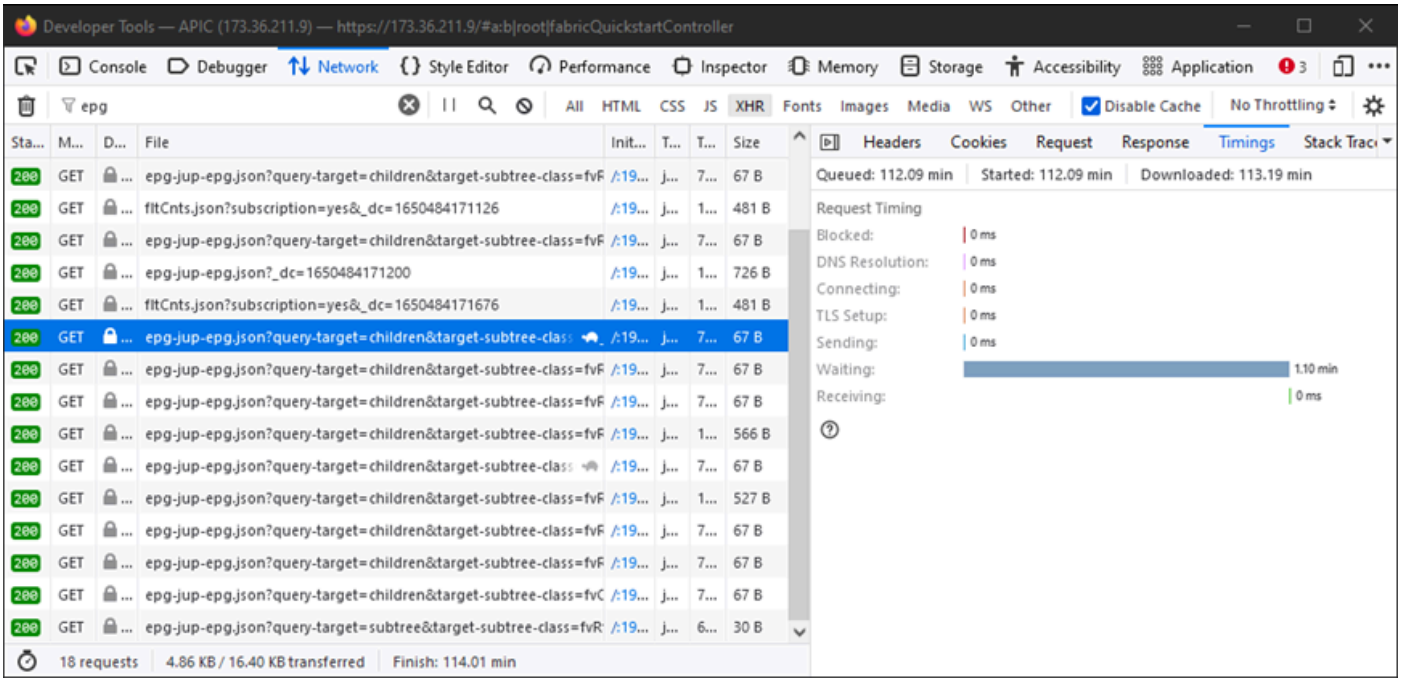
在这些情况下，验证从浏览器到APIC的数据路径（地理距离、VPN等）。 如果可能，请部署并测试从与APIC位于同一地理区域或数据中心的跳转服务器进行的访问，以隔离。验证其他用户是否显

示类似的延迟量。

浏览器开发工具网络选项卡

所有浏览器都能够通过其Browser Development（浏览器开发）工具包(通常位于Network（网络）选项卡中)验证HTTP请求和响应。

此工具可用于验证来自浏览器的请求的每个阶段所需的时间，如图所示。



浏览器等待1.1分钟APIC响应的示例

特定UI页面的增强功能

“策略组”页：

思科漏洞ID CSCvx14621 - APIC GUI在“交换矩阵”(Fabric)选项卡中的IPG策略上缓慢加载。

“资产”页面下的界面：

Cisco Bug ID CSCvx90048 — “Layer 1 Physical Interface Configuration”（第1层物理接口配置）的“Operational”（操作）选项卡的初始负载较长/导致“Freeze”（冻结）。

客户端>服务器延迟的一般建议

默认情况下，某些浏览器（如Firefox）允许每台主机拥有更多网络连接。

- 检查此设置是否可以在使用的浏览器版本上配置
- 这对于多查询页面(例如“策略组”(Policy Group)页面)更为重要

VPN和与APIC的距离会增加整体UI延迟，因为客户端浏览器请求和APIC响应传输时间会更长。APIC在地理上本地的跳转框显着减少了浏览器到APIC的传输时间。

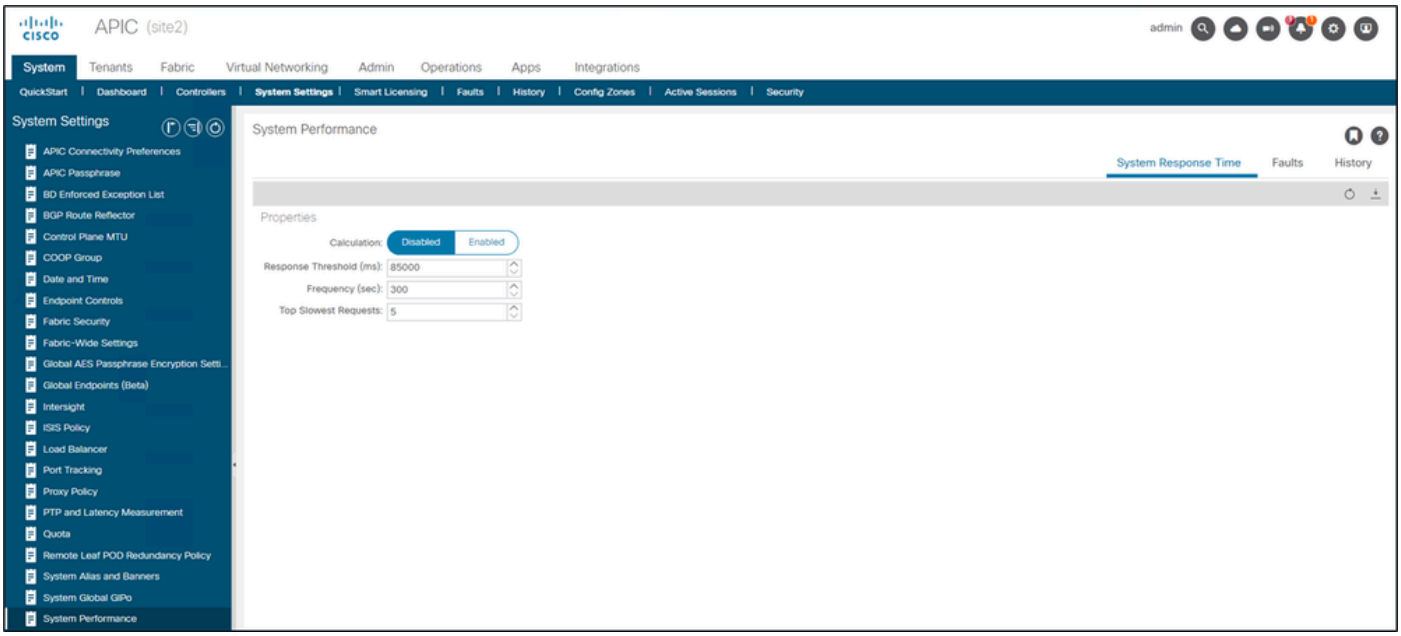
检查长期Web请求

如果Web服务器（APIC上的NGINX）处理大量长Web请求，这可能会影响并行收到的其他请求的性能。

对于具有分布式数据库的系统（如APIC）尤其如此。单个API请求可能需要向交换矩阵中的其他节点发送额外的请求和查找，这会导致预期的响应时间更长。这些长Web请求在较短的时间帧内突发，可能会增加所需的资源量，导致意外的较长响应时间。此外，收到的请求会超时（90秒），导致从用户角度而言出现意外的系统行为。

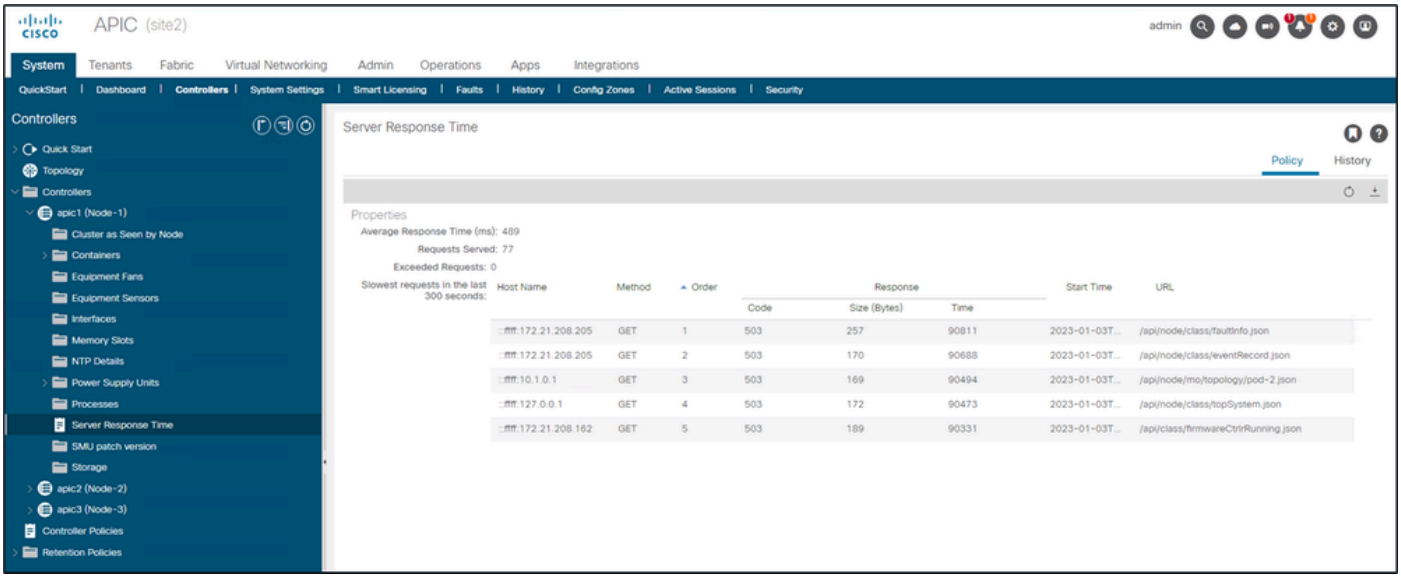
系统响应时间 — 启用服务器响应时间的计算

在4.2(1)+中，用户可以启用“系统性能计算”，跟踪并突出显示需要长时间处理的API请求。



可从“系统 — 系统设置 — 系统性能”中启用计算

启用“计算”后，用户可导航到“控制器”下的特定APIC，查看最后300秒内最慢的API请求。



APIC API使用注意事项

确保脚本不会损害Nginx的一般指针

- 每个APIC运行自己的NGINX DME。
  - 只有APIC 1的NGINX处理发往APIC 1的请求。APIC 2和3的NGINX不处理这些请求。
- 通常，在很长一段时间内每秒钟有40个以上的API请求会使NGINX衰弱。
  - 如果找到，则减少请求的攻击性。
  - 如果无法修改请求主机，请考虑对APIC执行NGINX速率限制。

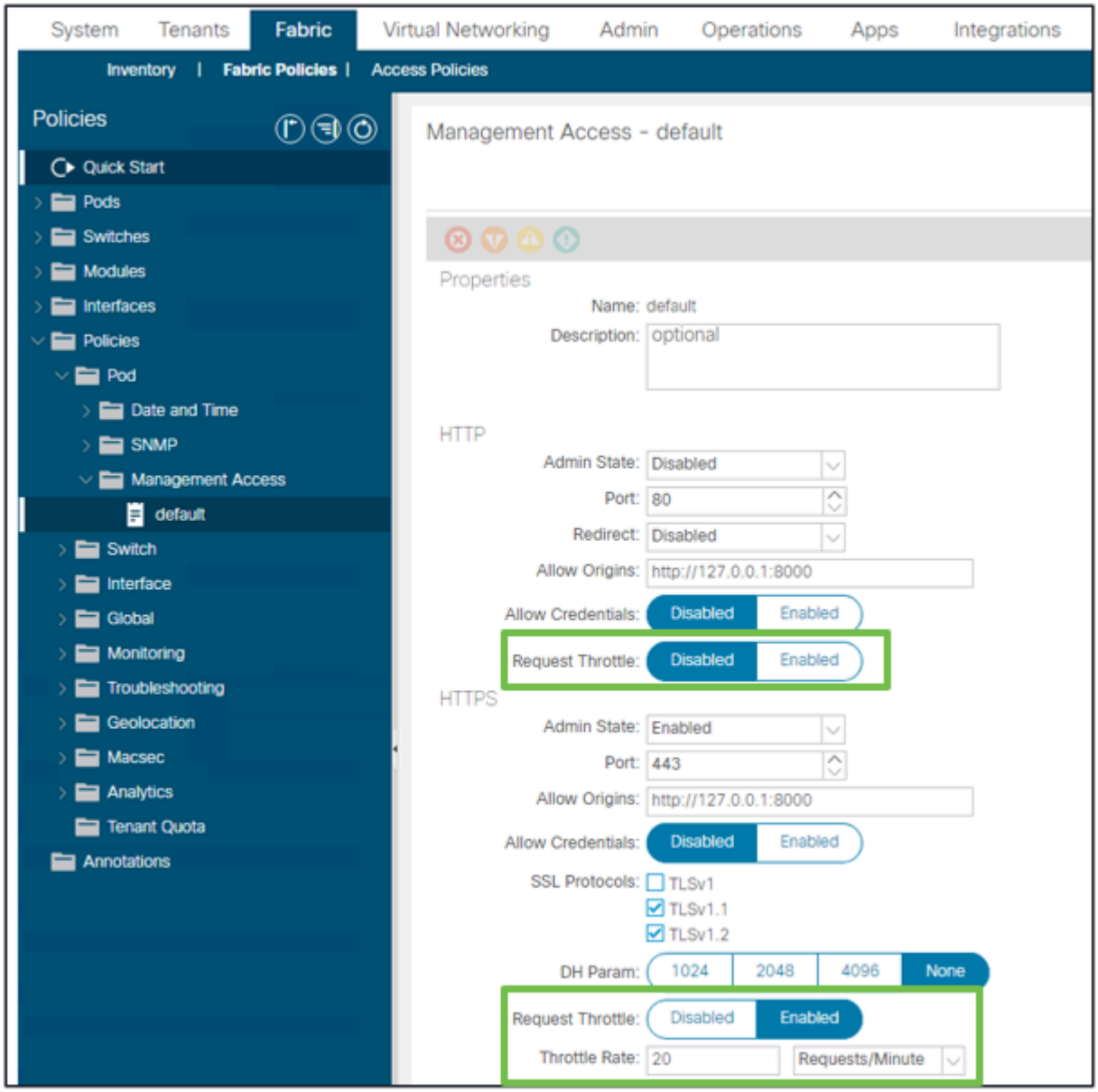
解决脚本效率低下问题

- 请勿在每个API请求之前登录/注销。
  - 一个登录会话的默认超时为10分钟。同一会话可用于多个请求，并可刷新以延长有效时间。
  - 请参阅思科APIC REST API配置指南 — 访问REST API — 身份验证和维护API会话。
- 如果您的脚本查询共享父级的多个DN，而不是使用查询过滤器将查询折叠为单个逻辑父级查询。
  - 请参阅Cisco APIC REST API配置指南 — 合成REST API查询 — 应用查询范围过滤器。
- 如果需要更新对象或对象类，请考虑websocket订阅，而不是快速API请求。

NGINX请求限制

在4.2(1)+版本中，用户可以独立启用针对HTTP和HTTPS的请求限制。

注意：从ACI 6.1(2)版开始，此功能支持的最大速率从10,000次/秒降至40个请求/秒(r/s)或2400个请求/分钟(r/m)。



交换矩阵 — 交换矩阵策略 — 策略文件夹 — 管理访问文件夹 — 默认

启用时：

- 重新启动NGINX以应用配置文件更改
  - 新区域httpsClientTagZone将写入nginx配置
- 限制速率可以设置为每分钟请求数(r/m)或每秒请求数(r/s)。
- 请求限制依赖于NGINX中包含的速率限制实施
  - 针对/api/URI的API请求使用用户定义的限制速率+ burst=（限制速率x 2）+ nodelay
    - /api/aaaLogin和/api/aaaRefresh有一个不可配置的限制(区域aaaApiHttps)，速率限制为2r/s + burst=4 + nodelay
  - 基于每个客户端IP地址跟踪请求限制
  - 来自APIC self-ip(UI + CLI)的API请求绕过限制
  - 任何超过用户定义的限制速率+突发阈值的客户端IP地址都会收到来自APIC的503响应
  - 这些503可在访问日志中关联
  - error.log包含指示何时激活限制（区域httpsClientTagZone）以及针对哪些客户端主机的条目

```
<#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"
```

一般来说，Request Throttle仅用于保护服务器(APIC)免受查询激进型客户端引起的类似DDOS的症状。在应用/脚本逻辑中了解并隔离请求激进型客户端，以获得最终解决方案。

建议

这些建议旨在帮助减少APIC的负载和运营压力，尤其是在没有单一来源负责大量API调用的情况下。通过实施这些最佳实践，您可以最大限度地减少交换矩阵中不必要的处理、日志记录和事件生成，从而提高系统稳定性和性能。这些建议在聚合行为而非孤立事件导致APIC紧张的环境中尤其适用。

禁用ACL日志记录

确保ACL日志记录已在正常操作期间关闭。仅在计划维护时段启用该功能，以便进行故障排除或调试。连续日志记录可能会生成过多信息性消息，尤其是当多台交换机发生大量流量丢弃时，会增加APIC工作负载。

有关详细信息，请参阅思科APIC安全配置指南（5.2.x指南的链接）：
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

将系统日志转换限制为严重事件

配置系统，以便仅将严重性为ALERT的系统日志消息转换为eventRecords。避免转换信息级别（包括ACL.logging），以防止噪声事件淹没APIC:

1.导航到Fabric → Fabric Policies → Policies → Monitoring → Common Policy → Syslog Message Policies → Default。
2.调整设备过滤器以将系统日志严重性设置为警报。

静噪Non-Essential事件代码

抑制（抑制）与监控无关的事件代码需要降低噪音。
要静音事件代码E4204939，请在任何APIC CLI上使用此命令：

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

要验证：

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

或者，通过UI检查：
Fabric > Fabric Policies > Policies > Monitoring > Common Policy > Event Severity Assignment Policy

优化ND订用刷新

对于由3.2.2m或4.1.1g之前的ND版本管理的交换矩阵，请升级到其中一个版本或更高版本以优化订用刷新间隔。早期版本每MO每45秒刷新一次，这样在规模上，每天可产生300,000多个APIC请求。更新的版本将订用超时增加到3600秒（1小时），将刷新减少到每天大约5,000次。

监视与监视相关的查询

支持Intersight的交换矩阵从DC连接器生成定期的topsystem查询（每15秒），从而增加APIC负载。在版本6.1.2及更高版本中，此查询已优化以降低开销。

调整记录的保留策略

将eventRecord、faultRecord和healthRecord的保留策略设置为1,000，以防止记录过度累积。当您为任何特定操作活动定期提取这些记录时，此功能尤其有用。请务必根据您的操作和故障排除要求评估降低监控精细度的影响。

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。