

# 当NSO消耗高CPU时收集数据

## 目录

---

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[要收集的数据](#)

[其他信息](#)

[相关信息](#)

---

## 简介

本文档介绍当CPU消耗量增加到100-150%时所需的网络服务协调器(NSO)数据收集。

## 先决条件

### 要求

本文档没有任何特定的要求。

### 使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始(默认)配置。如果您的网络处于活动状态,请确保您了解所有命令的潜在影响。

## 背景信息

当从NB处理多个事务时,NSO CPU消耗增加到正常消耗的约100-150%。发生这种情况时,您需要找出导致CPU性能下降的原因。同时,NSO不会正确响应RESTCONF(如果使用)查询。

本文重点介绍在问题期间需要收集的所有重要数据,以便正确解决问题并提出一些补救步骤。


## 要收集的数据

从Linux的角度来看:

- `lscpu`
- 顶部

- free -h
- vmstat
- cat /proc/meminfo
- pstree -c
- ps auxw | 排序

---

 注意：您可以定期捕获这些详细信息（除了“lscpu”），以便了解当请求来自NB时系统的行为。

---

### 从NSO的角度来看：

- ncs — 状态 | grep lock
- 启用进度跟踪：

```
admin@ncs(config)# commit dry-run
```

```
cli {
```

```
  local-node {
```

```
    数据进度{
```

```
      +全部跟踪{
```

```
        +目标{
```

```
          + file progress-all.txt;
```

```
          +格式日志；
```

```
        +}
```

```
      +}
```

```
    }
```

```
  }
```

```
}
```

```
admin@ncs(config)# commit
```

- 每隔“n”秒捕获一次下一信息（它可作为脚本运行）：

```
seq=0
```

```
当ncs —status >& /dev/null ; 执行
```

```
ncs —debug-dump ncs.dd.$(seq++);
```

```
ncs — 状态> ncs.stat.$(seq++);
```

```
睡眠30;#Configured according 到用户
```

```
done
```

接下来是一些补救步骤，也可以执行这些步骤来缓解此问题：

1. 按如下方式限制会话数量（目前，您没有此设置）：

```
<session-limits>
  <session-limit>
    <context>rest</context>
    <max-sessions>100</max-sessions>
  </session-limit>
</session-limits>
```

- b. 启用审核规则，以确定NSO流程是否被某事破坏，如果是，则将其记录在audit.log中：

```
sudo auditctl -a exit, always -F arch=b64 -S kill -k audit_kill
```

要进行故障排除和分析，您需要以前的详细信息以及audit.log、devel.log（最好设置为level=trace）、ncs-java-vm.log和NB日志。

## 其他信息

问：NSO实际上如何处理来自NB应用的RESTCONF请求？

A. 当北向应用发送RESTCONF请求时，根据NSO将其视为唯一事务。这意味着NSO可以锁定整个CDB，并且在当前事务完成之前不允许任何其他事务。如果执行此操作，将保留NSO的事务性质，并确保在出现任何问题时可以执行回滚。

NSO commit-queue可以在每个后续事务请求完成时对其进行处理，并且可以在事务开始/完成时在devel.log中跟踪事务锁定。在使用情况下，如果执行大量查询，这会在NSO中引入大量开销；而事务在提交队列中的时间比预期长。如果RESTCONF请求被分组，吞吐量将增加，因为事务开销会减少。此外，NSO可以在单个交易中同时执行尽可能多的操作。例如，如果事务包含2个设备配置更改，则NSO可以锁定CDB，同时访问和编辑两个设备，然后完成事务。这与每个包含1个设备且两者都更改的2个事务不同；因为NSO可以锁定第一个事务的CDB，编辑第一个设备，完成事务，然后对第二个设备执行相同的步骤。

## 相关信息

- [思科技术支持和下载](#)

## 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）。