

Collecter des données lorsque NSO consomme beaucoup de CPU

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Données à collecter](#)

[Additional Information](#)

[Informations connexes](#)

Introduction

Ce document décrit la collecte de données de Network Services Orchestrator (NSO) nécessaire lorsque la consommation du CPU augmente à 100-150 %.

Conditions préalables

Exigences

Aucune exigence spécifique n'est associée à ce document.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

Lorsque plusieurs transactions sont traitées à partir du NB, la consommation de CPU du NSO augmente à environ 100-150 % de la consommation normale. Lorsque cela se produit, vous devez trouver la cause qui dégrade les performances du processeur. Et, en même temps, le NSO ne répond pas correctement aux requêtes RESTCONF (si elles sont utilisées).


Cet article met en évidence toutes les données importantes qui doivent être collectées pendant le problème afin que le problème puisse être correctement résolu et suggère également quelques

étapes de correction.

Données à collecter

Du point de vue Linux :

- lscpu
- peigné
- free -h
- vmstat
- cat /proc/meminfo
- pstree -c
- ps auxw | tri

 Remarque : vous pouvez capturer ces détails (à l'exception de « lscpu ») à intervalles réguliers afin de comprendre le comportement du système lorsque les requêtes proviennent du NB.

Du point de vue des ONS :

- ncs —status | verrou en grep
- Activez la trace de progression :

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

```
cli {
```

```
    local-node {
```

```
        progression des données {
```

```
            + tracer tout {
```

```
                + destination {
```

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

```
                    + journal de format ;
```

```
                +}
```

```
            +}
```

```
        }
```

```
    }
```

```
}
```

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

- Capturez les informations suivantes toutes les n secondes (elles peuvent être exécutées sous forme de script) :

```
seq=0
while ncs --status >& /dev/null ; do
ncs --debug-dump ncs.dd.$(seq++);
ncs --status > ncs.stat.$(seq++);
sommeil 30 ; #Configured according à l'utilisateur
done
```

Vous trouverez ci-dessous quelques mesures correctives qui peuvent également être prises pour atténuer le problème :

1. Limitez le nombre de sessions comme suit (actuellement, vous n'avez pas cet ensemble) :

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

- b. Activez la règle d'audit pour vérifier si le processus NSO a été tué par quelque chose et, si tel est le cas, enregistrez-le dans audit.log :

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

Pour dépanner et analyser, vous avez besoin des détails précédents ainsi que des journaux audit.log, devel.log (de préférence défini à level=trace), ncs-java-vm.log et NB.

Additional Information

Q. Comment NSO gère-t-il réellement les requêtes RESTCONF d'une application NB ?

R. Lorsqu'une application en direction du nord envoie une requête RESTCONF, elle est traitée comme une transaction unique basée sur NSO. Cela signifie que NSO peut verrouiller l'intégralité de la CDB et n'autoriser aucune autre transaction tant que la transaction en cours n'est pas terminée. Si cela est fait, la nature transactionnelle de NSO est préservée et garantit qu'une restauration peut être effectuée en cas de problèmes.

La file d'attente de validation NSO peut traiter chaque demande de transaction suivante à mesure qu'elle se termine, et vous pouvez suivre le verrou de transaction dans devel.log à mesure qu'ils

commencent/se terminent. Dans les cas d'utilisation où un grand nombre de requêtes sont effectuées, cela introduit une surcharge importante dans NSO ; et les transactions sont dans la file d'attente de validation plus longtemps que prévu. Si les requêtes RESTCONF étaient regroupées, le débit augmenterait, car la surcharge de transaction serait réduite. En outre, NSO serait en mesure de faire tout ce qu'il peut en même temps, à l'intérieur d'une transaction unique. Par exemple, si une transaction contient 2 modifications de configuration de périphérique, NSO peut verrouiller la CDB, joindre et modifier les deux périphériques en même temps, puis terminer la transaction. Cela est différent de 2 transactions qui contiennent chacune 1 périphérique et les deux sont modifiées. Comme NSO peut verrouiller la CDB pour la première transaction, modifier le premier périphérique, terminer la transaction, puis effectuer les mêmes étapes pour le second périphérique.

Informations connexes

- [Assistance technique et téléchargements Cisco](#)

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.