

Verfahren zur Auflösung der MongoPrimaryDB-Fragmentierungswarnung in CPS

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Problem](#)

[Verfahren zum Beheben der Fragmentierungswarnung MongoPrimaryDB](#)

Einleitung

In diesem Dokument wird das Verfahren zur Auflösung der MongoPrimaryDB-Fragmentierungswarnung in Cisco Policy Suite (CPS) beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Linux
- CPS
- MongoDB

Anmerkung: Cisco empfiehlt, dass Sie über Berechtigungen für den Root-Zugriff auf die CPS CLI verfügen müssen.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- CPS 20.2
- MongoDB v3.6.17
- Unified Computing System (UCS)-B

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle verstehen.

Hintergrundinformationen

CPS verwendet MongoDB, wobei mongode Prozesse, die auf SessionManager Virtual Machines (VMs) ausgeführt werden, die grundlegende DataBase-Struktur bilden.

Wenn Dokumente verschoben oder entfernt werden, hinterlassen sie Löcher. MongoDB versucht, diese Löcher für neue Dokumente wiederzuverwenden, wann immer dies möglich ist, aber mit der Zeit hat es viele Löcher, die langsam und stetig wieder verwendet werden können, da Dokumente nicht in sie passen. Dieser Effekt wird als Fragmentierung bezeichnet und ist in allen Systemen üblich, die Speicher zuweisen, d. h. auch in Ihrem Betriebssystem.

Die Fragmentierung führt zu Platzverschwendung. Aufgrund der Tatsache, dass MongoDB Memory-Mapped-Dateien verwendet, jede Fragmentierung auf der Festplatte spiegelt in Fragmentierung im RAM als auch. Dies führt dazu, dass der 'Working Set' weniger in den RAM passt und die Festplatte mehr ausgetauscht wird.

CPS unterstützt KPIs zur Überwachung der Fragmentierung auf MongoDB-Ebene durch die Verwendung von Grafana und erzeugt einen SNMP-Alarm, wenn der MongoDB-Fragmentprozentsatz einen festgelegten Wert überschreitet.

Die Fehlermeldung `/etc/collectd.d/dbMonitorList.cfg` Datei, die auf virtuellen Maschinen von sessionmgr vorhanden ist, enthält die Liste der Datenbanken und die entsprechenden prozentualen Fragmentierungsschwellenwerte. Standardmäßig beträgt der Fragmentierungsschwellenwert 40 %. Der Standardschwellenwert für die Fragmentierung kann nach Bedarf geändert werden.

Die Fragmentierungsstatistik für Datenbanken mit den Parametern `session_cache`, `sk_cache`, `durchmesser` und `Subscriber Profile Repository (SPR)` (durch Verwendung primärer Member) kann mit dem folgenden Befehl überprüft werden:

```
[root@installer ~]# diagnostics.sh --get_frag
CPS Diagnostics HA Multi-Node Environment
-----
```

```
Ping check for qns03 Adding to IGNORED_HOSTS...[FAIL]
```

```
|-----|
|-----|
| Mongo:v3.6.17 DATABASE LEVEL FRAGMENTATION STATUS INFORMATION Date : 2022-09-17 07:19:29 |
| SET TYPE : HA [MEMBER_ROLE : PRIMARY] |
|-----|
|-----|
| setname dbName storageSize(MB) datasize(MB) indexSize(MB) fileSize(MB) derivedFS(MB) frag% |
|-----|
|-----|
| ADMIN:set06 |
| Status via sessionmgr01:27721 |
| set06 diameter 9.56 0.04 0.05 64.00 0 NoFrag |
|-----|
|-----|
| BALANCE:set02 |
| Status via sessionmgr01:27718 |
| set02 balance_mgmt db not found - - - - - |
|-----|
|-----|
| SESSION:set01 |
```

```

| Status via sessionmgr01:27717 |
| set01 session_cache 0.02 0.00 0.02 16.00 0 NoFrag |
|-----|
| SESSION:set01 |
| Status via sessionmgr01:27717 |
| set01 sk_cache 0.02 0.00 0.01 16.00 0 NoFrag |
|-----|
| SPR:set04 |
| Status via sessionmgr01:27720 |
| set04 spr 0.04 0.00 0.13 64.00 0 NoFrag |
|-----|
[root@installer ~]#

```

Problem

Wenn der Fragmentierungsprozentsatz des primären Members für den Replikatsatz den konfigurierten Fragmentierungsschwellenwert überschreitet, wird dieser Alarm generiert. Wenn der Schwellenwert nicht konfiguriert ist, wird der Alarm ausgelöst, wenn der Fragmentierungsprozentsatz den Standardwert (40 %) überschreitet.

Beispiel für eine Warnung zum Überschreiten des Schwellenwerts durch die MongoPrimaryDB-Fragmentierung:

```

id=7100,values={sub_id=7107, event_host=sessionmgr01, status=down, msg=MongoPrimaryDB
fragmentation exceeded the threshold value, CURR_FRAG = 40%, THRESHOLD = 40% at
sessionmgr01:27717 for session_cac

```

Verfahren zum Beheben der Fragmentierungswarnung MongoPrimaryDB

Um den Fragmentierungsprozentsatz zu reduzieren, verkleinern Sie die Datenbank, wenn ein Alarm generiert wird. Sobald die Datenbank verkleinert wird (der Fragmentierungsprozentsatz nimmt ab), wird ein Alarm gesendet, der gelöscht wird.

Dieses Verfahren dient dazu, die MongoPrimaryDB-Fragmentierungswarnung in dem bereitgestellten Beispiel zu beheben.

Schritt 1: Führen Sie diesen Befehl im Cluster-Manager oder im pcrfclient aus, um den Status der primären und sekundären Mitglieder im Replikatsatz zu überprüfen.

```

#diagnostics.sh --get_r
|-----|
|SESSION:set01a|
|Status via sessionmgr01:27717 sessionmgr02:27717 |
|Member-1-27717 : 192.168.29.14-ARBITER-pcrfclient01- ON-LINE--0| -----|
|Member-2-27717 : 192.168.29.35-PRIMARY-sessionmgr01- ON-LINE--3| -----|
|Member-3-27717 : 192.168.29.36-SECONDARY-sessionmgr02- ON-LINE--2| 1 sec|
|-----|

```

Schritt 2: Führen Sie diesen Befehl im Cluster Manager oder im pcrfclient aus, um die Priorität von

sessionmgr01 zu ändern und daraus ein sekundäres Mitglied zu machen.

```
#sh set_priority.sh --db session --replSet set01a --asc
```

Expected output in #diagnostics.sh --get_r

```
|-----|
|-----|
|SESSION:set01a|
|Status via sessionmgr02:27717 sessionmgr01:27717 |
|Member-1-27717 : 192.168.29.14-ARBITER-pcrfclient01- ON-LINE--0| -----|
|Member-2-27717 : 192.168.29.35-PRIMARY-sessionmgr02- ON-LINE--3| -----|
|Member-3-27717 : 192.168.29.36-SECONDARY-sessionmgr01- ON-LINE--2| 1 sec|
|-----|
|-----|
```

Anmerkung: Stellen Sie sicher, dass sessionmgr01 nicht mehr primär ist (diagnostics.sh —get_r) und ein primäres Mitglied für den Replikatsatz verfügbar ist.

Schritt 3: Führen Sie diesen Befehl aus Sessionmgr01 aus, um den AIDO-Client zu beenden.

```
#monit stop aido_client
```

Schritt 4: Führen Sie diesen Befehl aus Sessionmgr01 aus, um die entsprechende Mongo-Instanz zu beenden (portNum ist die Portnummer des fragmentierten Elements).

Command syntax:

```
#/etc/init.d/sessionmgr-<portNum> stop
```

Example:

```
#/etc/init.d/sessionmgr-27717 stop
```

Schritt 5: Um das Datenbankverzeichnis in sessionmgr01 zu bereinigen, entfernen Sie das Datenverzeichnis aus dem angegebenen Pfad mit dem —dbpath-Attribut des mongo-Befehls. Führen Sie diesen Befehl in SessionMgr01 aus, um den Wert abzurufen (verwenden Sie die portNum des fragmentierten Elements).

Anmerkung: Da sich die Portnummern und Verzeichnisse anderer Sitzungsmgr-Datenbanken unterscheiden, stellen Sie sicher, dass Sie über die richtigen Verzeichnisse verfügen, um andere Sitzungsmgr-Datenbanken zu bereinigen.

Command syntax:

```
#grep -w DBPATH= /etc/init.d/sessionmgr-<portNum>
```

Example:

```
#grep -w DBPATH= /etc/init.d/sessionmgr-27717
```

Sample Output: DBPATH=/var/data/sessions.1/a

Copy the DBPATH from output.

Command syntax:

```
#rm -rf <DBPATH>/*
```

Example:

```
#rm -rf /var/data/sessions.1/a/*
```

Schritt 6: Führen Sie diesen Befehl aus Sessionmgr01 aus, um die entsprechende Mongo-Instanz zu starten.

Command syntax:

```
#/etc/init.d/sessionmgr-<portNum> start
```

Example:

```
#/etc/init.d/sessionmgr-27717 start
```

Schritt 7: Führen Sie diesen Befehl aus Sessionmgr01 aus, um den AIDO-Client zu starten.

```
#monit start aido_client
```

Schritt 8: Führen Sie diesen Befehl im Cluster-Manager oder im pcrfclient aus, um die Prioritäten der Replikatgruppenmitglieder zurückzusetzen.

```
#sh set_priority.sh --db session --replSet set01a
```

Schritt 9: Führen Sie diesen Befehl im Cluster-Manager oder im pcrfclient aus, um den Status der primären und sekundären Mitglieder im Replikatsatz zu überprüfen.

```
#diagnostics.sh --get_r
```

```
|-----|
|-----|
|SESSION:set01a|
|Status via sessionmgr01:27717 sessionmgr02:27717 |
|Member-1-27717 : 192.168.29.14-ARBITER-pcrfclient01- ON-LINE--0| -----|
|Member-2-27717 : 192.168.29.35-PRIMARY-sessionmgr01- ON-LINE--3| -----|
|Member-3-27717 : 192.168.29.36-SECONDARY-sessionmgr02- ON-LINE--2| 1 sec|
|-----|
|-----|
```

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.