

# 解决CPS中的MongoPrimaryDB分段警报的过程

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题](#)

[解决MongoPrimaryDB分段警报的过程](#)

## 简介

本文档介绍在思科策略套件(CPS)中解决MongoPrimaryDB分段警报的过程。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- Linux
- CPS
- MongoDB

**注意：** Cisco建议您必须拥有对CPS CLI的超级用户权限。

### 使用的组件

本文档中的信息基于以下软件和硬件版本：

- CPS 20.2
- MongoDB v3.6.17
- 统一计算系统(UCS)-B

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

CPS使用MongoDB，其中在Sessionmgr虚拟机(VM)上运行的单一进程构成其基本数据库结构。

当文档移动或删除时，它们会留下孔。MongoDB会尽可能地重新使用这些孔来存储新文档，但随着时间的推移，它发现自己缓慢而稳定地拥有了许多孔，这些孔无法重新使用，因为文档无法容纳这些孔中。这种效果称为分段，在分配内存的所有系统（包括操作系统[OS]）中均常见。

碎片化的效果是浪费空间。由于MongoDB使用内存映射文件，因此磁盘上的任何分段也反映在RAM中的分段中。这会导致在RAM中创建较少的“工作集”，并导致磁盘交换更多。

CPS支持KPI，通过使用Grafana监控MongoDB级别分段，并在MongoDB分段百分比超过指定值时生成简单网络管理协议(SNMP)警报。

此 `/etc/collectd.d/dbMonitorList.cfg` `sessionmgr`虚拟机上的文件包含数据库及其各自的分段阈值百分比值的列表。默认情况下，分段阈值为40 %。可以根据需要更改默认的分段阈值。

可以使用以下命令检查`session_cache`、`sk_cache`、`diameter`和`Subscriber Profile Repository(SPR)`数据库（通过使用主要成员）的分段统计信息：

```
[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 ~]#
```

## 问题

当副本集的主要成员的碎片百分比超过配置的阈值碎片值时，将生成此警报。如果未配置阈值，则当分段百分比超出默认值(40%)时发出警报。

“MongoPrimaryDB分段超出阈值”警报示例：

```
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
```

## 解决MongoPrimaryDB分段警报的过程

为了减少分段百分比，请在生成警报时收缩数据库。一旦数据库缩小（分段百分比减少），将发送清晰的警报。

此过程用于解决提供的示例中的MongoPrimaryDB分段警报。

步骤1.从Cluster Manager或pcrfclient运行此命令以验证复制副本集中主要和辅助成员的状态。

```
#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|
|-----|
|-----|
```

步骤2.从集群管理器或pcrfclient运行此命令以更改sessionmgr01的优先级并使其成为辅助成员。

```
#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|
|-----|
|-----|
```

**注意：**确保sessionmgr01不再为主要(diagnostics.sh —get\_r)，并且存在可用于副本集的主成员。

步骤3.从Sessionmgr01运行此命令以停止AIDO客户端。

```
#monit stop aido_client
```

步骤4.从Sessionmgr01运行此命令以停止各自的Mongo实例（portNum是分片成员的端口号）。

Command syntax:

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

Example:

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

步骤5.为了清理sessionmgr01中的数据库目录，请根据mongo命令的 — dbpath属性从所提及的路径中删除数据目录。从Sessionmgr01运行此命令以检索值（使用分段成员的portNum）。

**注意：**由于与其他sessionmgr dbs关联的端口号和目录不同，请确保您有正确的目录来清理其他sessionmgr dbs。

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/*
```

步骤6.从Sessionmgr01运行此命令以启动相应的Mongo实例。

Command syntax:

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

Example:

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

步骤7.从Sessionmgr01运行此命令以启动AIDO客户端。

```
#monit start aido_client
```

步骤8.从Cluster Manager或pcrfclient运行此命令以重置复制副本集成员的优先级。

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

步骤9.从Cluster Manager或pcrfclient运行此命令以验证复制副本集中主要和辅助成员的状态。

```
#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|
|-----|
|-----|
```

## 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）。