

# 处理CPS Mongo复制副本集中损坏的MongoDB的过程

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题](#)

[修复损坏的数据库成员的过程](#)

## 简介

本文档介绍在思科策略套件(CPS)副本集中处理损坏的MongoDataBase(DB)的过程。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- Linux
- CPS
- MongoDB

**注意：**思科建议您必须拥有对CPS CLI的根访问权限。

### 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
- 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

MongoDB是一种源可用的跨平台面向文档的数据库(DB)程序。分类为NoSQL数据库程序。MongoDB广泛用于CPS中，用于管理其不同类型的数据库，即会话、用户配置文件存储库(SPR)、平衡等。

# 问题

当aido\_client在sessionmgr内仍处于活动状态时，执行不正确的数据库碎片整理时，MongoDB将损坏。

这使MongoDB将数据保存在内存中，但无法在数据库路径上本地写入数据。

如果主成员（mongo实例）在受影响的复制副本集上重新启动或会话管理器VM重新启动，这可能导致数据丢失。

为了了解数据库成员看起来如何损坏，您可以登录到其中一个有问题的成员并执行所提供的检查。

步骤1.运行命令show dbs时，未返回DB列表的输出。但是，当您检查数据库中您知道的计数时，它会返回计数。

```
[root@lab-1-pcrfclient01 ~]# mongo --host sessionmgr05:27737
MongoDB shell version v3.6.17
connect to: mongodb://sessionmgr05:27737/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a8f9b0eb-6e78-4bcd-bd63-60a9a9d813d0") }
MongoDB server version: 3.6.17
Server has startup warnings:
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] ** WARNING: Access control is not
enabled for the database.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **           Read and write access to
data and configuration is unrestricted.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.949-0300 I REPL [replexec-0]
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] ** WARNING: This replica set uses arbiters,
but readConcern:majority is enabled
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] **           for this node. This is not a
recommended configuration. Please see
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] **
2022-03-09T00:53:26.949-0300 I REPL [replexec-0]
set01e:PRIMARY>
set01e:PRIMARY> show dbs ## "no dbs reported"
set01e:PRIMARY> use session_cache ## "Switched to a known DB"
switched to db session_cache
set01e:PRIMARY> db.session.count()
223037 ## "DB has the content inside, hence the total record count is shown"
set01e:PRIMARY> use session_cache_2
switched to db session_cache_2
set01e:PRIMARY> db.session.count()
223643
set01e:PRIMARY> use session_cache_3
switched to db session_cache_3
set01e:PRIMARY> db.session.count()
222939
set01e:PRIMARY> use session_cache_4
switched to db session_cache_4
set01e:PRIMARY> db.session.count()
223692
set01e:PRIMARY>
set01e:PRIMARY> exit
bye
```

步骤2.运行diagnostics.sh —get\_shard时，应用程序分片显示数据。这实际上存储在内存中，而不是存储在会话管理器虚拟机(VM)的DBPATH中。

```
[root@lab-1-pcrfclient01 ~]# diagnostics.sh --get_shard
CPS Diagnostics GR Multi-Node Environment
```

```
-----|
-----|
|                                     SHARD STATUS INFORMATION
|      Date : 2022-03-09 11:00:23    |
|-----|
-----|
```

Shard Id	Mongo DB	State	Backup DB	Removed	Session Count
43	sessionmgr01:27717/session_cache	online	false	false	223873
1	sessionmgr01:27717/session_cache_2	online	false	false	222918
2	sessionmgr01:27717/session_cache_3	online	false	false	223720
3	sessionmgr01:27717/session_cache_4	online	false	false	223393
8	sessionmgr05:27737/session_cache	online	false	false	223188
9	sessionmgr05:27737/session_cache_2	online	false	false	223554
10	sessionmgr05:27737/session_cache_3	online	false	false	222920
11	sessionmgr05:27737/session_cache_4	online	false	false	223562
12	sessionmgr07:27747/session_cache	online	false	false	222663
13	sessionmgr07:27747/session_cache_2	online	false	false	222599
14	sessionmgr07:27747/session_cache_3	online	false	false	222475
15	sessionmgr07:27747/session_cache_4	online	false	false	223446
16	sessionmgr09:27757/session_cache	online	false	false	223246
17	sessionmgr09:27757/session_cache_2	online	false	false	223669
18	sessionmgr09:27757/session_cache_3	online	false	false	223711
19	sessionmgr09:27757/session_cache_4	online	false	false	223311
35	sessionmgr13:27717/session_cache	online	true	false	0
36	sessionmgr13:27717/session_cache_2	online	true	false	0
37	sessionmgr13:27717/session_cache_3	online	true	false	0
38	sessionmgr13:27717/session_cache_4	online	true	false	0

```
Rebalance Status: Rebalanced
```

步骤3.此输出显示数据库路径内没有应存储实际数据的内容。

```
[SESSION-SET3]
SETNAME=set01e
OPLOG_SIZE=5120
ARBITER=lab-1-arb-sessmgr15:27737
ARBITER_DATA_PATH=/var/data/sessions.1/set01e
PRIMARY-MEMBERS
MEMBER1=lab-1-sessionmgr05:27737
MEMBER2=lab-1-sessionmgr06:27737
SECONDARY-MEMBERS
MEMBER3=lab-2-sessionmgr05:27737
MEMBER4=lab-2-sessionmgr06:27737
DATA_PATH=/var/data/sessions.1/set01e ## "DB DATA Path of set01e replicaset"
[SESSION-SET3-END]
```

安全外壳(SSH)到关联的会话管理器，并导航至mongo配置中提到的DB\_PATH。您可以看到DB\_PATH中的内容为空。

```
[root@lab-1-sessionmgr05 ~]# cd /var/data/sessions.1/set01e
[root@lab-1-sessionmgr05 ~]# ls -lrt
total 0
[root@lab-1-sessionmgr05 ~]#
```

通过这些检查，您可以得出MongoDB已损坏的结论。



```

}
}
}
set01e:PRIMARY> show dbs
admin    0.031GB
config  0.031GB
set01e:PRIMARY> exit

```

步骤4.立即连接到同一副本，并在所有session\_cache\_dbs上执行这些命令。此处简要介绍了session\_cache数据库的示例。

```

# mongo --port 27737
# use session_cache
# db.session.count()           # Use this to check that session counts are still intact
# db.stats(1024*1024*1024)    # Use this to verify that the storage size is proper
# db.repairDatabase()
# exit

```

```

[root@lab-1-sessionmgr05 set01e]# mongo --port 27737
MongoDB shell version v3.6.17
connect to: mongod://127.0.0.1:27737/?gssapiServiceName=mongod
Implicit session: session { "id" : UUID("73794d11-0785-4520-ba82-19f0d2bba338") }
MongoDB server version: 3.6.17
Server has startup warnings:
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] ** WARNING: Access control is not
enabled for the database.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **           Read and write access to
data and configuration is unrestricted.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.949-0300 I REPL  [replexec-0]
2022-03-09T00:53:26.949-0300 I REPL  [replexec-0] ** WARNING: This replica set uses arbiters,
but readConcern:majority is enabled
2022-03-09T00:53:26.949-0300 I REPL  [replexec-0] **           for this node. This is not a
recommended configuration. Please see
2022-03-09T00:53:26.949-0300 I REPL  [replexec-0] **
2022-03-09T00:53:26.949-0300 I REPL  [replexec-0]
set01e:PRIMARY>
set01e:PRIMARY>
set01e:PRIMARY>
set01e:PRIMARY> show dbs
admin    0.031GB
config  0.031GB
set01e:PRIMARY> use session_cache
switched to db session_cache
set01e:PRIMARY>
set01e:PRIMARY> db.stats(1024*1024*1024)
{
  "db" : "session_cache",
  "collections" : 3,
  "views" : 0,
  "objects" : 212467,
  "avgObjSize" : 8175.252062673262,
  "dataSize" : 1.6176805645227432,
  "storageSize" : 2.471107453107834,
  "numExtents" : 22,
  "indexes" : 3,
  "indexSize" : 0.30870679020881653,
  "fileSize" : 0,
  "nsSizeMB" : 16,
  "extentFreeList" : {
    "num" : 0,

```

```

"totalSize" : 0
},
"dataFileVersion" : {
"major" : 4,
"minor" : 22
},
"fsUsedSize" : 38.36811065673828,
"fsTotalSize" : 47.044921875,
"ok" : 1,
"operationTime" : Timestamp(1647321405, 102),
"$clusterTime" : {
"clusterTime" : Timestamp(1647321405, 103),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
set01e:PRIMARY> db.repairDatabase()
{
"ok" : 1,
"operationTime" : Timestamp(1647321444, 84),
"$clusterTime" : {
"clusterTime" : Timestamp(1647321444, 84),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
set01e:PRIMARY> show dbs
admin            0.031GB
config           0.031GB
session_cache    2.499GB

```

**注意：**对session\_cache数据库的其余部分重复步骤4。

**步骤5.**确保当您连接同一个mongo实例时，show dbs现在列出所有数据库。

```

mongo --port 27737
set01e:PRIMARY> show dbs
admin 0.031GB
config 0.031GB
session_cache 2.499GB
session_cache_2 2.499GB
session_cache_3 2.499GB
session_cache_4 2.499GB

```

**步骤6.**确保数据库路径现在包含会话管理器内的所有本地数据。您可以检查复制副本集的各自数据路径。在本例中为/var/data/sessions.1/set01e。

```

[root@lab-1-sessionmgr05 set01~]# cd /var/data/sessions.1/set01e
[root@lab-1-sessionmgr05 set01e]# ls
admin session_cache session_cache_2.1 session_cache_2.7 session_cache_3.1 session_cache_3.7
session_cache_4.1 session_cache_4.7 session_cache.8
admin.0 session_cache.0 session_cache_2.2 session_cache_2.8 session_cache_3.2 session_cache_3.8
session_cache_4.2 session_cache_4.8 session_cache.ns
admin.ns session_cache.1 session_cache_2.3 session_cache_2.ns session_cache_3.3

```

```
session_cache_3.ns session_cache_4.3 session_cache_4.ns _tmp
config session_cache.2 session_cache_2.4 session_cache.3 session_cache_3.4 session_cache.4
session_cache_4.4 session_cache.5
config.0 session_cache_2 session_cache_2.5 session_cache_3 session_cache_3.5 session_cache_4
session_cache_4.5 session_cache.6
config.ns session_cache_2.0 session_cache_2.6 session_cache_3.0 session_cache_3.6
session_cache_4.0 session_cache_4.6 session_cache.7
```

步骤7.通过SSH连接到同一站点辅助成员，并执行数据路径与主成员的本地同步。

```
ssh to lab-1-sessionmgr06 (Secondary member)
Ensure to stop aido_client
```

```
# monit stop aido_client
```

```
Ensure to stop mongo processes
```

```
# /etc/init.d/sessionmgr-27737 stop # Wait for 10 seconds and start the service
back on
```

确保数据路径/var/data/sessions.1/set01e为空，如果为空，则使用rm -rf /var/data/sessions.1/set01e/\*删除这些路径，然后启动mongo进程。

```
# /etc/init.d/sessionmgr-27737 start
```

```
[root@lab-1-sessionmgr06 ~]# monit stop aido_client
[root@lab-1-sessionmgr06 ~]# monit status aido_client
Monit 5.26.0 uptime: 52d 20h 59m
```

```
Process 'aido_client'
status Not monitored
monitoring status Not monitored
monitoring mode active
on reboot start
data collected Wed, 23 Mar 2022 08:08:46
```

```
[root@lab-1-sessionmgr06 ~]#
[root@lab-1-sessionmgr06 ~]# /etc/init.d/sessionmgr-27737 stop
stop sessionmgr-27737 (via systemctl): [ OK ]
[root@lab-1-sessionmgr06 ~]# rm -rf /var/data/sessions.1/set01e/*
[root@lab-1-sessionmgr06 ~]# cd /var/data/sessions.1/set01e/
[root@lab-1-sessionmgr06 set01e]# ls
[root@lab-1-sessionmgr06 set01e]#
[root@lab-1-sessionmgr06 set01e]# /etc/init.d/sessionmgr-27737 start
Starting sessionmgr-27737 (via systemctl): [ OK ]
```

步骤8.验证数据现在已本地复制到/var/data/sessions.1/set01e中。

```
[root@lab-1-sessionmgr06 ~]# cd /var/data/sessions.1/set01e/
[root@lab-1-sessionmgr06 set01e]# ls
admin.0 local.1 local.3 local.7 mongod.lock session_cache_2.3 session_cache_2.7
session_cache_3.1 session_cache_3.5 session_cache_3.ns
admin.ns local.10 local.4 local.8 session_cache_2.0 session_cache_2.4 session_cache_2.8
session_cache_3.2 session_cache_3.6 storage.bson
diagnostic.data local.11 local.5 local.9 session_cache_2.1 session_cache_2.5 session_cache_2.ns
session_cache_3.3 session_cache_3.7 _tmp
local.0 local.2 local.6 local.ns session_cache_2.2 session_cache_2.6 session_cache_3.0
session_cache_3.4 session_cache_3.8
[root@lab-1-sessionmgr06 set01e]#
```

**注意：**对地理站点辅助成员重复步骤7.和步骤8.。在实验室中，成员为lab-2-sessionmgr05和lab-2-sessionmgr06。

步骤9.恢复所有辅助数据库（本地和地理站点）后，在主成员上重新启动mongo服务。

```
[root@lab-1-sessionmgr05 ~]# /etc/init.d/sessionmgr-27737 stop
stop sessionmgr-27737 (via systemctl): [ OK ]
```

等待10秒，确认主交换机已成功。

```
[root@lab-1-sessionmgr06 ~]# mongo --port 27737
MongoDB shell version v3.6.17
connect to: mongodb://127.0.0.1:27737/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("ba8e49fa-ad0f-4ac6-8ef8-b4da0a88fe33") }
MongoDB server version: 3.6.17
Server has startup warnings:
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten]
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten] ** WARNING: Access control is not enabled
for the database.
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten] ** Read and write access to data and
configuration is unrestricted.
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten] **
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten]
set01e:PRIMARY>
set01e:PRIMARY>
set01e:PRIMARY> show dbs
admin 0.031GB
config 0.031GB
local 5.029GB
session_cache 2.499GB
session_cache_2 2.499GB
session_cache_3 2.499GB
session_cache_4 2.499GB
set01e:PRIMARY> show dbs
admin 0.031GB
config 0.031GB
local 5.029GB
session_cache 2.499GB
session_cache_2 2.499GB
session_cache_3 2.499GB
session_cache_4 2.499GB
set01e:PRIMARY> rs.status()
{
  "set" : "set01e",
  "date" : ISODate("2022-03-15T06:13:19.991Z"),
  "myState" : 1,
  "term" : NumberLong(36),
  "syncingTo" : "",
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1647324799, 335),
      "t" : NumberLong(36)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1647324799, 335),
      "t" : NumberLong(36)
    }
  }
}
```



```
},
"appliedOpTime" : {
"ts" : Timestamp(1647324799, 338),
"t" : NumberLong(36)
},
"durableOpTime" : {
"ts" : Timestamp(0, 0),
"t" : NumberLong(-1)
}
},
"members" : [
{
"_id" : 0,
"name" : "lab-2-sessionmgr06:27737",
"health" : 1,
"state" : 2,
"stateStr" : "SECONDARY",
"uptime" : 486,
"optime" : {
"ts" : Timestamp(1647324799, 94),
"t" : NumberLong(36)
},
"optimeDurable" : {
"ts" : Timestamp(0, 0),
"t" : NumberLong(-1)
},
"optimeDate" : ISODate("2022-03-15T06:13:19Z"),
"optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
"lastHeartbeat" : ISODate("2022-03-15T06:13:19.267Z"),
"lastHeartbeatRecv" : ISODate("2022-03-15T06:13:18.270Z"),
"pingMs" : NumberLong(0),
"lastHeartbeatMessage" : "",
"syncingTo" : "lab-1-sessionmgr06:27737",
"syncSourceHost" : "lab-1-sessionmgr06:27737",
"syncSourceId" : 4,
"infoMessage" : "",
"configVersion" : 8
},
{
"_id" : 1,
"name" : "lab-1-sessionmgr05:27737",
"health" : 1,
"state" : 2,
"stateStr" : "SECONDARY",
"uptime" : 885,
"optime" : {
"ts" : Timestamp(1647324799, 96),
"t" : NumberLong(36)
},
"optimeDurable" : {
"ts" : Timestamp(0, 0),
"t" : NumberLong(-1)
},
"optimeDate" : ISODate("2022-03-15T06:13:19Z"),
"optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
"lastHeartbeat" : ISODate("2022-03-15T06:13:19.270Z"),
"lastHeartbeatRecv" : ISODate("2022-03-15T06:13:18.270Z"),
"pingMs" : NumberLong(0),
"lastHeartbeatMessage" : "",
"syncingTo" : "lab-1-sessionmgr06:27737",
"syncSourceHost" : "lab-1-sessionmgr06:27737",
"syncSourceId" : 4,
"infoMessage" : "",
"configVersion" : 8
}
```

```
},
{
  "_id" : 2,
  "name" : "lab-1-arb-sessmgr15:27737",
  "health" : 1,
  "state" : 7,
  "stateStr" : "ARBITER",
  "uptime" : 1130,
  "lastHeartbeat" : ISODate("2022-03-15T06:13:19.240Z"),
  "lastHeartbeatRecv" : ISODate("2022-03-15T06:13:18.856Z"),
  "pingMs" : NumberLong(0),
  "lastHeartbeatMessage" : "",
  "syncingTo" : "",
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "infoMessage" : "",
  "configVersion" : 8
},
{
  "_id" : 3,
  "name" : "lab-1-sessionmgr05:27737",
  "health" : 0,
  "state" : 8,
  "stateStr" : "(not reachable/healthy)",
  "uptime" : 0,
  "optime" : {
    "ts" : Timestamp(0, 0),
    "t" : NumberLong(-1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(0, 0),
    "t" : NumberLong(-1)
  },
  "optimeDate" : ISODate("1970-01-01T00:00:00Z"),
  "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
  "lastHeartbeat" : ISODate("2022-03-15T06:13:19.299Z"),
  "lastHeartbeatRecv" : ISODate("2022-03-15T06:11:58.086Z"),
  "pingMs" : NumberLong(0),
  "lastHeartbeatMessage" : "Connection refused",
  "syncingTo" : "",
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "infoMessage" : "",
  "configVersion" : -1
},
{
  "_id" : 4,
  "name" : "lab-1-sessionmgr06:27737",
  "health" : 1,
  "state" : 1,
  "stateStr" : "PRIMARY",
  "uptime" : 1130,
  "optime" : {
    "ts" : Timestamp(1647324799, 338),
    "t" : NumberLong(36)
  },
  "optimeDate" : ISODate("2022-03-15T06:13:19Z"),
  "syncingTo" : "",
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "infoMessage" : "",
  "electionTime" : Timestamp(1647324719, 72),
  "electionDate" : ISODate("2022-03-15T06:11:59Z"),
  "configVersion" : 8,
}
```

```
"self" : true,
"lastHeartbeatMessage" : ""
}
],
"ok" : 1,
"operationTime" : Timestamp(1647324799, 338),
"$clusterTime" : {
"clusterTime" : Timestamp(1647324799, 338),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
```

步骤10.在实验1-sessionmgr05上重新启动mongo服务，该服务之前是主成员。

```
[root@lab-1-sessionmgr05 ~]# /etc/init.d/sessionmgr-27737 start
Starting sessionmgr-27737 (via systemctl): [ OK ]
```

步骤11.在步骤2中停止的set01e复制副本集的所有复制副本成员上启动aido\_client。