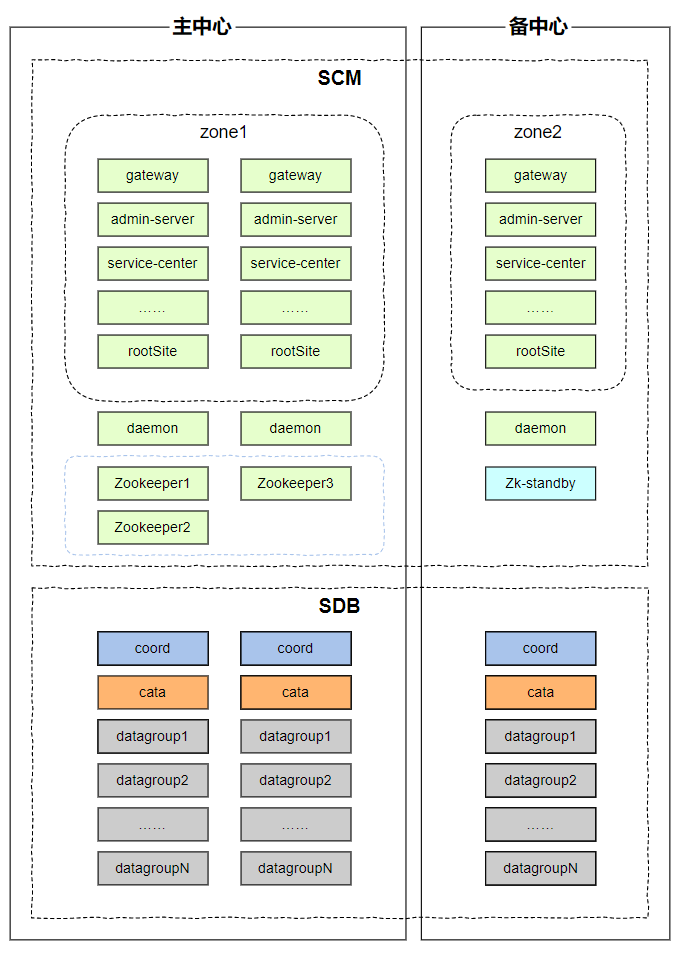
SequoiaCM 灾备部署与故障处理手册

## 同城双中心

### 1.1 部署模型



根据数据中心划分子网，主中心为 SUB1，备中心为 SUB2（本章节以 server1、server2 代指主中心机器、server3 代指备中心机器）

|  |  |
| --- | --- |
| **子网** | **主机** |
| SUB1 | server1、server2 |
| SUB2 | server3 |

#### 部署 SDB

1. 搭建三副本 SDB 集群，两个副本在主中心，一个副本在备中心
2. 设置**备中心**每个数据节点参数 dataerrorop 为 2

db.updateConf({"dataerrorop":2}, {HostName:"server3"})

注：因为同城容灾网络的带宽有限，需要严格控制 SequoiaDB 集群对同城带宽的占用，阻止数据节点在异常终止后进行自动全量同步。

1. 配置所有 SDB 安装机上 sdbcm 节点参数 AutoStart=FALSE 和 EnableWatch=TRUE

vi /opt/sequoiadb/conf/sdbcm.conf

修改内容如下：

……

AutoStart=FALSE

EnableWatch=TRUE

注：此配置是防止在主中心故障恢复后，sdbcm自动拉起主中心的 SequoiaDB 节点，主中心和备中心可能会形成两个独立的可读写 SequoiaDB 集群。主中心 SCM 节点连接到主中心 SequoiaDB 节点进行数据写入，就会出现“脑裂（brain-split）”的情况。

1. 初始化容灾环境。集群信息初始化操作用于将主节点分配至主中心。（执行该操作前，用户需转移或删除集群部署信息文件 datacenter\_init.info，该文件位于 SequoiaDB 安装目录）
   1. 根据数据中心划分子网，主中心为 SUB1，备中心为 SUB2
   2. 选择 SUB1 的任意一台主机，切换至目录 /opt/sequoiadb/tools/dr\_ha

cd /opt/sequoiadb/tools/dr\_ha

* 1. 修改配置文件 cluster\_opr.js

vim cluster\_opr.js

修改内容如下：

if ( typeof(SEQPATH) != "string" || SEQPATH.length == 0 )

{ SEQPATH = "/opt/sequoiadb/" ; }

if ( typeof(SDBUSERNAME) != "string" ) { SDBUSERNAME = "sdbadmin" ; }

if ( typeof(SDBPASSWD) != "string" ) { SDBPASSWD = "sdbadmin" ; }

if ( typeof(SUB1HOSTS) == "undefined" ) { SUB1HOSTS = [ "server1", "server2" ] ; }

if ( typeof(SUB2HOSTS) == "undefined" ) { SUB2HOSTS = [ "server3" ] ; }

if ( typeof(COORDADDR) == "undefined" ) { COORDADDR = [ "server1:11810" ] }

if ( typeof(CURSUB) == "undefined" ) { CURSUB = 1 ; }

if ( typeof(CUROPR) == "undefined" ) { CUROPR = "split" ; }

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = true ; }

if ( typeof(NEEDBROADCASTINITINFO) == "undefined" ) { NEEDBROADCASTINITINFO = false }

* 1. 集群信息初始化

sh init.sh

* 1. 选择 SUB2 的主机，切换至目录 /opt/sequoiadb/tools/dr\_ha

cd /opt/sequoiadb/tools/dr\_ha

* 1. 修改配置文件 cluster\_opr.js

vim cluster\_opr.js

修改内容如下：

if ( typeof(SEQPATH) != "string" || SEQPATH.length == 0 )

{ SEQPATH = "/opt/sequoiadb/" ; }

if ( typeof(SDBUSERNAME) != "string" ) { SDBUSERNAME = "sdbadmin" ; }

if ( typeof(SDBPASSWD) != "string" ) { SDBPASSWD = "sdbadmin" ; }

if ( typeof(SUB1HOSTS) == "undefined" ) { SUB1HOSTS = [ "server1", "server2" ] ; }

if ( typeof(SUB2HOSTS) == "undefined" ) { SUB2HOSTS = [ "server3" ] ; }

if ( typeof(COORDADDR) == "undefined" ) { COORDADDR = [ "server1:11810" ] }

if ( typeof(CURSUB) == "undefined" ) { CURSUB = 2 ; }

if ( typeof(CUROPR) == "undefined" ) { CUROPR = "split" ; }

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = false ; }

if ( typeof(NEEDBROADCASTINITINFO) == "undefined" ) { NEEDBROADCASTINITINFO = true }

* 1. 集群信息初始化

sh init.sh

#### 部署 SCM

SCM 集群部署于两个中心。配置 zone1、zone2 来分别描述主中心、备中心

1. SCM 部署的每个服务或站点，均有节点实例落于主中心、备中心
2. SCM 每台部署机上均部署了守护进程，在部分节点出现故障时可以及时拉起
3. 搭建 3 节点 Zookeeper 集群作为 SCM 集群依赖服务，均落于主中心
4. 灾备中心[部署独立模式 Zookeeper](#_冷备_Zookeeper_部署) 作为冷备（不启动，只有主中心故障时才启动）

### 容灾处理

#### 主中心故障

需要人工干预，剔除故障节点。

1. 当主中心发生故障，SequoiaDB 集群失去半数以上的节点，无法为 SCM 提供存储服务。
2. Zookeeper 集群节点均故障，无法支撑 SCM 分布式锁、服务节点选举能力，SCM 集群不可用。

**故障处理**

1. 停止备中心所有 SCM 服务节点（先停网关）
2. 应用系统隔离主中心，只连备中心网关
3. 备中心与主中心进行网络隔离，如下以 iptables 的方法作为示例（在备中心的所有机器上执行）

iptables -A INPUT -s server1主机IP -j DROP

iptables -A OUTPUT -s server1主机IP -j DROP  
iptables -A INPUT -s server3主机IP -j DROP  
iptables -A OUTPUT -s server3主机IP -j DROP

1. 登录备中心机器，使用 SDB 的分裂脚本，将备中心上的一副本脱离原集群，成为具备读写功能的独立集群，以恢复 SequoiaDB 服务
   1. 在 server3 机器上修改 cluster\_opr.js 中的 ACTIVE 参数为 true

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = true ; }

* 1. 执行分裂

sh split.sh

1. 启动备中心server3 上的冷备 Zookeeper

/opt/sequoiacm/zk-standby/zookeeper-3.4.12/bin/zkServer.sh start zoo.cfg

检查启动状态，显示Mode: standalone 表示启动成功

/opt/sequoiacm/zk-standby/zookeeper-3.4.12/bin/zkServer.sh status

1. 修改 SCM 的站点集合信息，只保留备中心 SDB 协调节点的地址
   1. 编辑scmupdate.json文件，主站点 meta、data 只保留备中心 SDB 协调节点的地址，将文件以及相关脚本上传至灾备中心主机 server3 上

{

"sdbpath": "/opt/sequoiadb",

"coordhost": "server3",

"coordport": "11810",

"rootsite": {

"meta.url": "server3:11810",

"data.url": "server3:11810"

}

……

}

* 1. 修改主站点的 meta、data 地址

python scmupdate.py --updatemeta

1. 修改备中心上 SCM 节点配置和日志配置（需要在备中心所有 SCM 的部署机上执行）
   1. 编辑scmupdate.json文件，Zookeeper 替换为冷备 Zookeeper，仅保留备中心 SDB 的协调节点

{

"scmpath": "/opt/sequoiacm",

"config": {

"sdburls": "server3:11810",

"zkurls": "server3:3981" // 冷备 zookeeper 的连接地址

}

……

}

* 1. 修改 SCM 节点配置

python scmupdate.py --updateconf

1. 启动备中心上的所有 SCM 服务节点（先启动 service-center，最后启动网关）

**故障恢复**

1. 停止备中心上的所有 SCM 服务节点（先停网关）
2. 停止备中心 server3 上的冷备 Zookeeper

/opt/sequoiacm/zk-standby/zookeeper-3.4.12/bin/zkServer.sh stop zoo.cfg

1. 主中心执行分裂
   1. 登录至 server1，修改 cluster\_opr.js 中的 ACTIVE 参数为 false

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = false ; }

* 1. 执行分裂

sh /opt/sequoaidb/tools/dr\_ha/split.sh

检查主中心集群状态：连接主中心集群，执行“写”操作的命令，如创建集合（创建失败，报错信息：The cluster is read only）

1. 主中心和备中心执行合并
   1. 在备中心 server3 上先执行合并

sh /opt/sequoaidb/tools/dr\_ha/merge.sh

* 1. 在主中心 server1 上执行合并

sh /opt/sequoaidb/tools/dr\_ha/merge.sh

1. 确认 SequoiaDB 集群同步是否同步完成

# 连接协调节点

db = new Sdb("server3", 11810)

# 查看非正常状态节点，如果记录为 0，表示节点状态正常，且同步完成

db.snapshot(SDB\_SNAP\_HEALTH, {Status:{"$ne":"Normal"}}, {"NoedeNmae":null, "Status":null})

1. 参照上文 部署 SDB 步骤4，再次执行初始化，恢复 SDB 集群最初状态（将主节点重新分布到主中心）
   1. 主中心（SUB1）设置 ACTIVE=true
   2. 备中心（SUB2）设置 ACTIVE=false
2. 启动主中心上的 Zookeeper 集群
3. 修改 SCM 的站点集合信息，添加主中心 SDB 协调节点的地址
   1. 在 server3 上编辑scmupdate.json文件，主站点 meta、data 添加主中心 SDB 协调节点的地址

{

"sdbpath": "/opt/sequoiadb",

"coordhost": "server3",

"coordport": "11810",

"rootsite": {

"meta.url": "server1:11810,server2:11810,server3:11810",

"data.url": "server1:11810,server2:11810,server3:11810"

}

……

}

* 1. 修改主站点的 meta、data 地址

python scmupdate.py --updatemeta

1. 修改备中心上 SCM 节点配置和节点日志配置，添加主中心 SDB 协调节点的地址（需要在备中心所有 SCM 的部署机上执行）
   1. 编辑scmupdate.json文件，Zookeeper 替换为Zookeeper 集群，添加主中心 SDB 的协调节点

{

"scmpath": "/opt/sequoiacm",

"config": {

"sdburls": " server1:11810,server2:11810,server3:11810",

"zkurls": "server1:2981,server2:2981,server3:2981" // zookeeper 集群的连接地址

}

……

}

* 1. 修改 SCM 节点配置

python scmupdate.py --updateconf

1. 解除备中心对主中心的隔离（在备中心的所有机器上执行）

iptables -D INPUT -s server1主机IP -j DROP

iptables -D OUTPUT -s server1主机IP -j DROP  
iptables -D INPUT -s server3主机IP -j DROP  
iptables -D OUTPUT -s server3主机IP -j DROP

1. 启动主备中心的所有 SCM 节点（先启动 service-center，最后启动网关）
2. 应用解除对主中心的隔离

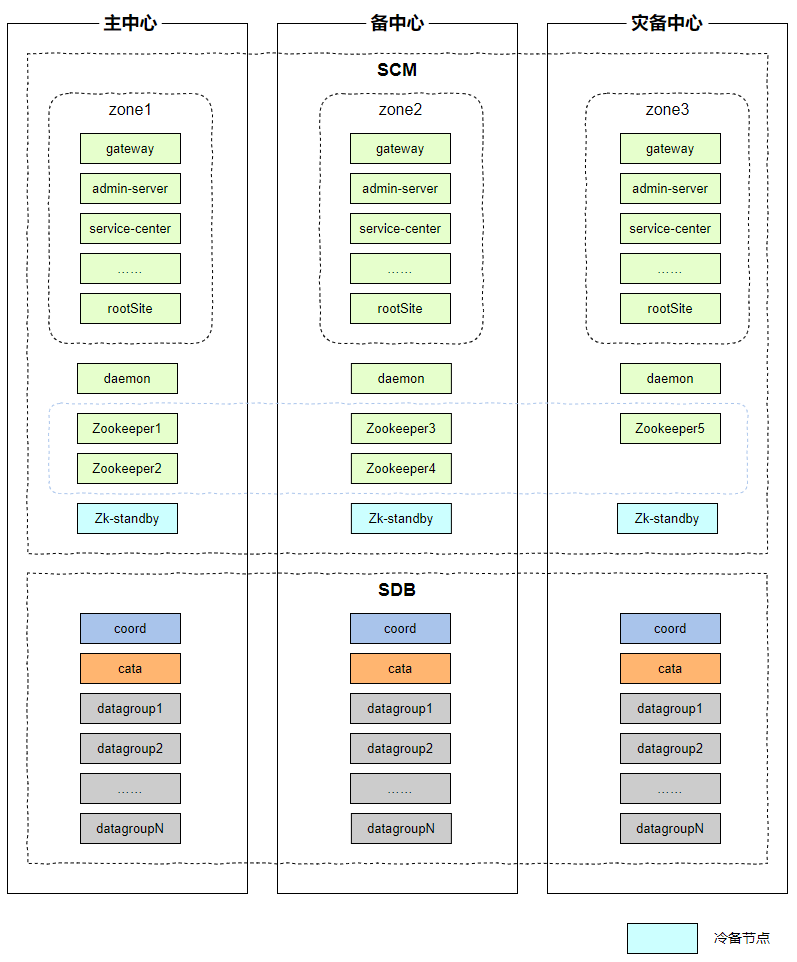
#### 备中心故障

无需采取特别的应对措施，只需要及时修复故障节点，并恢复故障节点数据即可。原因如下：

* SDB 每个数据组都有两副本在主中心，存活节点数大于数据组总节点数的 1/2，因此每个数据组仍然能够为 SCM 应用层提供数据读写服务。
* Zookeeper 集群节点均落于主中心，可正常对外提供服务，故 SCM 集群仍然可用。

## 两地三中心

### 2.1 部署模型



根据数据中心划分子网，主中心为 SUB1，备中心、灾备中心为 SUB2（本章节以 server1、server2 代指主中心机器、server3 代指备中心机器）

|  |  |
| --- | --- |
| **子网** | **主机** |
| SUB1 | server1 |
| SUB2 | server2、server3 |

#### 部署 SDB

1. 搭建三副本 SDB 集群，主中心、备中心，灾备中心分别部署一副本
2. 设置备中心、灾备中心每个数据节点参数 dataerrorop=2

db.updateConf({"dataerrorop":2}, {HostName:"server2"})

db.updateConf({"dataerrorop":2}, {HostName:"server3"})

注：因为同城容灾网络的带宽有限，需要严格控制 SequoiaDB 集群对同城带宽的占用，阻止数据节点在异常终止后进行自动全量同步。

1. 配置所有 SDB 安装机上 sdbcm 节点参数 AutoStart=FALSE 和 EnableWatch=TRUE

vi /opt/sequoiadb/conf/sdbcm.conf

修改内容如下：

……

AutoStart=FALSE

EnableWatch=TRUE

注：此配置时防止多中心发生故障，故障中心（例如主中心、灾备中心）恢复后，sdbcm自动拉起故障中心的 SequoiaDB 节点，正常中心（备中心）和故障中心（主中心、灾备中心）会形成两个独立的可读写 SequoiaDB 集群。此时如果故障中心 SCM 节点连接到故障中心 SequoiaDB 节点进行数据写入，就会出现“脑裂（brain-split）”的情况。

1. 初始化容灾环境。集群信息初始化操作用于将主节点分配至主中心。（执行该操作前，用户需转移或删除集群部署信息文件 datacenter\_init.info，该文件位于 SequoiaDB 安装目录）
   1. 根据数据中心划分子网，主中心为 SUB1，备中心、灾备中心为 SUB2
   2. 选择 SUB1 的任意一台主机，切换至目录 /opt/sequoiadb/tools/dr\_ha

cd /opt/sequoiadb/tools/dr\_ha

* 1. 修改配置文件 cluster\_opr.js

vim cluster\_opr.js

修改内容如下：

if ( typeof(SEQPATH) != "string" || SEQPATH.length == 0 )

{ SEQPATH = "/opt/sequoiadb/" ; }

if ( typeof(SDBUSERNAME) != "string" ) { SDBUSERNAME = "sdbadmin" ; }

if ( typeof(SDBPASSWD) != "string" ) { SDBPASSWD = "sdbadmin" ; }

if ( typeof(SUB1HOSTS) == "undefined" ) { SUB1HOSTS = [ "server1" ] ; }

if ( typeof(SUB2HOSTS) == "undefined" ) { SUB2HOSTS = [ "server2" , "server3" ] ; }

if ( typeof(COORDADDR) == "undefined" ) { COORDADDR = [ "server1:11810" ] }

if ( typeof(CURSUB) == "undefined" ) { CURSUB = 1 ; }

if ( typeof(CUROPR) == "undefined" ) { CUROPR = "split" ; }

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = true ; }

if ( typeof(NEEDBROADCASTINITINFO) == "undefined" ) { NEEDBROADCASTINITINFO = false }

* 1. 集群信息初始化

sh init.sh

* 1. 选择 SUB2 的主机，切换至目录 /opt/sequoiadb/tools/dr\_ha

cd /opt/sequoiadb/tools/dr\_ha

* 1. 修改配置文件 cluster\_opr.js

vim cluster\_opr.js

修改内容如下：

if ( typeof(SEQPATH) != "string" || SEQPATH.length == 0 )

{ SEQPATH = "/opt/sequoiadb/" ; }

if ( typeof(SDBUSERNAME) != "string" ) { SDBUSERNAME = "sdbadmin" ; }

if ( typeof(SDBPASSWD) != "string" ) { SDBPASSWD = "sdbadmin" ; }

if ( typeof(SUB1HOSTS) == "undefined" ) { SUB1HOSTS = [ "server1" ] ; }

if ( typeof(SUB2HOSTS) == "undefined" ) { SUB2HOSTS = [ "server2" , "server3" ] ; }

if ( typeof(COORDADDR) == "undefined" ) { COORDADDR = [ "server1:11810" ] }

if ( typeof(CURSUB) == "undefined" ) { CURSUB = 2 ; }

if ( typeof(CUROPR) == "undefined" ) { CUROPR = "split" ; }

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = false ; }

if ( typeof(NEEDBROADCASTINITINFO) == "undefined" ) { NEEDBROADCASTINITINFO = true }

* 1. 集群信息初始化

sh init.sh

#### 部署 SCM

SCM 集群部署于三个中心。配置 zone1、zone2、zone3 来分别描述主中心、备中心、灾备中心

1. SCM 部署的每个服务或站点，均有一个节点实例落于主中心、备中心、灾备中心
2. SCM 每台部署机上均部署了守护进程，在部分节点出现故障时可以及时拉起
3. 搭建 5 节点 Zookeeper 集群作为 SCM 集群依赖服务。在 SCM 集群描述文件中，Zookeeper 的配置顺序为灾备中心、主中心、备中心，可以保证 Zookeeper 主节点落在主中心上，示例如下

#zookeeper 节点部署数量建议为奇数

[zookeeper]

HostName, ServerPort, ClientPort

server3, 2988:3988, 2981

server1, 2988:3988, 2981

server1, 2999:3999, 3981

server2, 2988:3988, 2981

server2, 2999:3999, 3981

1. 在每个中心上[部署独立模式 Zookeeper](#_冷备_Zookeeper_部署) 作为冷备

### 2.2 容灾处理

#### 单个中心故障

任何一个中心出现故障时，不需要人工干预，平台仍能正常对外提供服务。

1. SDB 每个数据组都有两个副本存活，存活数大于数据组节点总数的 1/2，因此每个数据组仍然能够为 SCM 应用层提供读写服务。
2. Zookeeper 集群节点存活数 >= 3，可正常对外提供服务，SCM 集群仍然可用。

#### 双中心故障

当其中两个中心整体发生故障，整个集群环境将会失去超过 1/2 的节点。这个时候就需要人工干预，把存活的中心分裂成单独的集群提供读写服务。本文以**主中心**和**灾备中心**故障为例。

**故障处理**

1. 停止备中心所有 SCM 服务节点（先停网关）
2. 应用系统隔离主中心、灾备中心，只连备中心网关
3. 备中心与主中心进行网络隔离，如下以 iptables 的方法作为示例（在备中心的所有机器上执行）

iptables -A INPUT -s server1主机IP -j DROP

iptables -A OUTPUT -s server1主机IP -j DROP  
iptables -A INPUT -s server3主机IP -j DROP  
iptables -A OUTPUT -s server3主机IP -j DROP

1. 重新划分子网，将正常机房（备中心）划分为 SUB1，故障机房（主中心、灾备中心）划分为 SUB2

|  |  |
| --- | --- |
| **子网** | **主机** |
| SUB1 | server2 |
| SUB2 | server1、server3 |

1. 登录备中心机器，使用 SDB 的分裂脚本，将备中心上的一副本脱离原集群，成为具备读写功能的独立集群，以恢复 SequoiaDB 服务
   1. 在 server2 机器上修改 cluster\_opr.js 配置文件，修改内容如下

if ( typeof(SUB1HOSTS) == "undefined" ) { SUB1HOSTS = [ "server2" ] ; }

if ( typeof(SUB2HOSTS) == "undefined" ) { SUB2HOSTS = [ "server1", "server3" ] ; }

if ( typeof(COORDADDR) == "undefined" ) { COORDADDR = [ "server2:11810" ] }

if ( typeof(CURSUB) == "undefined" ) { CURSUB = 1 ; }

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = true ; }

* 1. 执行分裂

sh split.sh

1. 启动备中心server2 上的冷备 Zookeeper

/opt/sequoiacm/zk-standby/zookeeper-3.4.12/bin/zkServer.sh start zoo.cfg

检查启动状态，显示Mode: standalone 表示启动成功

/opt/sequoiacm/zk-standby/zookeeper-3.4.12/bin/zkServer.sh status

1. 修改 SCM 的站点集合信息，只保留备中心 SDB 协调节点的地址
   1. 编辑scmupdate.json文件，主站点 meta、data 只保留备中心 SDB 协调节点的地址，将文件以及相关脚本上传至灾备中心主机 server3 上

{

"sdbpath": "/opt/sequoiadb",

"coordhost": "server3",

"coordport": "11810",

"rootsite": {

"meta.url": "server2:11810",

"data.url": "server2:11810"

}

……

}

* 1. 修改主站点的 meta、data 地址

python scmupdate.py --updatemeta

1. 修改备中心上 SCM 节点配置和日志配置（需要在备中心所有 SCM 的部署机上执行）
   1. 编辑scmupdate.json文件，Zookeeper 替换为冷备 Zookeeper，仅保留备中心 SDB 的协调节点

{

"scmpath": "/opt/sequoiacm",

"config": {

"sdburls": "server2:11810",

"zkurls": "server2:3981" // 冷备 zookeeper 的连接地址

}

……

}

* 1. 修改 SCM 节点配置

python scmupdate.py --updateconf

1. 启动备中心上的所有 SCM 服务节点（先启动 service-center，最后启动网关）

**故障恢复**

1. 停止备中心上的所有 SCM 服务节点（先停网关）
2. 停止备中心 server3 上的冷备 Zookeeper

/opt/sequoiacm/zk-standby/zookeeper-3.4.12/bin/zkServer.sh stop zoo.cfg

1. 开启灾备中心（除主中心之外的故障中心）全量同步，避免节点重启失败

逐一修改所有灾备中心节点的配置文件 sdb.conf

vim /opt/sequoiadb/conf/local/<端口号>/sdb.conf

修改内容如下：

……

dataerrorop=1

……

1. 主中心执行分裂
   1. 在 server1 机器上修改 cluster\_opr.js 配置文件，修改内容如下

if ( typeof(SUB1HOSTS) == "undefined" ) { SUB1HOSTS = [ "server2" ] ; }

if ( typeof(SUB2HOSTS) == "undefined" ) { SUB2HOSTS = [ "server1", "server3" ] ; }

if ( typeof(COORDADDR) == "undefined" ) { COORDADDR = [ "server1:11810" ] }

if ( typeof(CURSUB) == "undefined" ) { CURSUB = 2 ; }

if ( typeof(ACTIVE) == "undefined" ) { ACTIVE = false ; }

* 1. 执行分裂

sh /opt/sequoaidb/tools/dr\_ha/split.sh

检查主中心集群状态：连接主中心集群，执行“写”操作的命令，如创建集合（创建失败，报错信息：The cluster is read only）

1. 主中心和备中心执行合并
   1. 在备中心 server2 上先执行合并

sh /opt/sequoaidb/tools/dr\_ha/merge.sh

* 1. 在主中心 server1 上执行合并

sh /opt/sequoaidb/tools/dr\_ha/merge.sh

1. 确认 SequoiaDB 集群同步是否同步完成

# 连接协调节点

db = new Sdb("server2", 11810)

# 查看非正常状态节点，如果记录为 0，表示节点状态正常，且同步完成

db.snapshot(SDB\_SNAP\_HEALTH, {Status:{"$ne":"Normal"}}, {"NoedeNmae":null, "Status":null})

1. 参照上文 部署 SDB 步骤4，再次执行初始化，恢复 SDB 集群最初状态（将主节点重新分布到主中心）
   1. 主中心（SUB1）设置 ACTIVE=true
   2. 备中心（SUB2）设置 ACTIVE=false
2. 启动主中心上的 Zookeeper 集群（依次启动灾备中心、主中心、备中心）
3. 修改 SCM 的站点集合信息，添加主中心、灾备中心 SDB 协调节点的地址
   1. 在 server2 上编辑scmupdate.json文件，主站点 meta、data 添加主中心、灾备中心 SDB 协调节点的地址

{

"sdbpath": "/opt/sequoiadb",

"coordhost": "server2",

"coordport": "11810",

"rootsite": {

"meta.url": "server1:11810,server2:11810,server3:11810",

"data.url": "server1:11810,server2:11810,server3:11810"

}

……

}

* 1. 修改主站点的 meta、data 地址

python scmupdate.py --updatemeta

1. 修改备中心上 SCM 节点配置和节点日志配置，添加主中心、灾备中心 SDB 协调节点的地址（需要在备中心所有 SCM 的部署机上执行）
   1. 编辑scmupdate.json文件，Zookeeper 替换为Zookeeper 集群，添加主中心、灾备中心 SDB 的协调节点

{

"scmpath": "/opt/sequoiacm",

"config": {

"sdburls": " server1:11810,server2:11810,server3:11810",

"zkurls": "server1:2981,server2:2981,server3:2981" // zookeeper集群的连接地址

}

……

}

* 1. 修改 SCM 节点配置

python scmupdate.py --updateconf

1. 解除备中心对主中心、灾备中心的隔离

iptables -D INPUT -s server1主机IP -j DROP

iptables -D OUTPUT -s server1主机IP -j DROP  
iptables -D INPUT -s server3主机IP -j DROP  
iptables -D OUTPUT -s server3主机IP -j DROP

1. 启动所有中心的所有 SCM 节点（先启动 service-center，最后启动网关）
2. 应用解除对主中心、灾备中心的隔离（在备中心的所有机器上执行）

## 附录

### 冷备 Zookeeper 部署

以 Zookeeper 3.4.12 版本为例介绍冷备 Zookeeper 部署流程

1. 在 SequoiaCM 安装目录下创建 zookeeper-standby 目录

mkdir /opt/sequoiacm/zookeeper-stanby

1. 解压 Zookeeper 压缩包至步骤 1 创建的目录

tar -zxvf zookeeper-3.4.12.tar.gz -C /opt/sequoiacm/zookeeper-standby/ --strip-

components 1

1. 在安装目录 /opt/sequoiacm/zookeeper-standby/ 下创建数据目录

mkdir -p /opt/sequoiacm/zookeeper-standy/data/1

1. 在数据目录下创建 myid 文件，并配置服务编号为 1

echo 1 > /opt/sequoiacm/zookeeper-standy/data/1/myid

1. 从 SCM 部署的 Zookeeper 集群中拷贝一份 zoo.cfg 配置文件

cp /opt/sequoiacm/zookeeper-3.4.12/conf/zoo.cfg /opt/sequoiacm/zookeeper-standy

/conf/zoo.cfg

1. 修改 zoo.cfg
   1. 修改 dataDir 为步骤 4 创建的 data 目录

dataDir=/opt/sequoiacm/zookeeper-standby/data

* 1. 修改 clientPort

clientPort=3981

* 1. 修改实例列表（删除已有的 server.x 实例列表）

server.1=（冷备 zk 部署机器IP）:2999:3999

注：以上端口号可灵活配置，确保主机端口不被占用即可。

1. 使用 SequoiaCM 安装包下的 /sequoiacm-deploy/bindata/zk\_shell/3.4.12.sh 替换 bin 目录下的zkServer.sh，以支持守护进程工具监控冷备 zookeeper 的能力