


# 凝思 V6.0.80 系统双机多路径部署作业指导书

北京凝思科技有限公司

北京凝思软件股份有限公司

# 目录

**1 概述.....1**
**2 准备.....2**
2.1 安装前准备.....2
2.2 软件版本确认.....2
2.3 存储模式确认.....3
2.4 双机热备需求确认.....3
2.5 风险以及影响范围.....3
**3 凝思多路径软件安装配置.....4**
3.1 多路径软件包的安装.....4
3.2 配置多路径序列号.....4
3.3 多路径软件的配置.....4
3.3.1 wwid 获取.....5
3.3.2 多路径配置模板.....5
3.3.2.1 双控配置模板.....5
3.3.2.2 双活配置模板.....6
3.3.2.3 主备配置模板.....7
3.3.2.4 宏衫存储模板.....9
3.3.3 多路径配置文件概述.....11
3.3.4 启停多路径及运维命令.....12
3.4 逻辑卷的配置方法.....12
**4 验证多路径基本功能.....14**
4.1 映射设备读写文件测试.....14
4.2 拔插光纤测试.....15
**5 凝思双机热备软件的安装配置.....17**
5.1 双机热备软件包的安装.....17
5.1.1 适用凝思安全操作系统 V6.0.80 root 环境.....17
5.1.2 适用凝思安全操作系统 V6.0.80 non-root 环境.....18
5.2 存储的 IO 调度算法配置.....19
5.3 双机热备的序列号配置.....19
5.4 双机热备的环境搭建.....19
5.4.1 双机热备的网络环境.....20
5.4.2 节点名及 hosts 解析.....20
5.4.3 双机节点的时间同步.....20
5.4.4 双机热备的引擎配置（corosync）.....21
5.4.5 双机热备的 stonith 配置.....22
5.4.6 启动 corosync 和 pacemaker.....23
5.4.7 双机热备的资源管理软件配置（pacemaker）.....23

5.4.8 双机热备 pacemaker 配置说明.....25
5.4.9 双机热备的日志轮转.....28
5.4.10 重启服务器.....29
5.4.11 双机软件使用.....29
5.4.11.1 启动/停止凝思双机热备软件.....29
5.4.11.2 查看凝思双机热备系统的运行状态.....29
5.4.11.3 查看凝思双机热备运行日志.....30
5.4.11.4 资源管理.....30
**6 验证双机基本功能.....31**
6.1 双机运行状态.....31
6.2 手动切机.....31
6.3 模拟 lxha 故障切机.....32
6.4 拔心跳线测试.....32
6.5 拔光纤线测试.....32
**7 常见问题解决汇总.....33**
7.1 文件系统使用.....33
7.2 不重启发现新加存储盘的命令.....33
7.3 XFS 文件系统修复的方法.....33

# 1 概述

1. 本文档为凝思 V6.0.80 系统下多路径软件和双机热备软件的安装部署作业指导书。
2. 本文档介绍了安装凝思多路径软件和双机热备软件时需要的准备工作、安装部署过程、参数注解、注意事项等。
3. 本文档可作为公司内部人员安装凝思双机热备软件和凝思多路径软件的评估及参考，指导实施人员正确无误的完成凝思多路径软件和凝思双机热备软件的安装部署。
4. 本文档只适用于凝思 V6.0.80 版本的操作系统，其他系统请选择对应版本的作业指导书进行参考。

# 2 准备

## 2.1 安装前准备

针对凝思安全操作系统不同的版本,选择对应系统版本的双机热备软件和多路径软件进行安装部署。安装部署凝思双机热备软件、多路径软件需要存储厂家、数据库厂家同时在场。

1. 确认存储厂家、数据库厂家到场时间。
2. 确认时间地点及安装环境、确认软件版本。
3. 针对现场环境选择对应的软件包。
4. 建议使用的 iso：Linx-6.0.80-20190130-amd64-DVD-1.iso

## 2.2 软件版本确认

1. 确认系统版本及发布时间,查询方法如下:

```
root@LinxA:~# cat /etc/issue
Linx 6.0.80 2019-01-30 \n \l

root@LinxA:~# uname -a
Linux rocky80-his1 4.9.0-0.bpo.1-linx-security-amd64 #1 SMP Linx 4.9.2-2~bpo8+1linx4 (2018-08-17) x86_64 GNU/Linux
```

2. 多路径、双机软件包确认

多路径安装包：

| 序号 | 多路径包名称 | MD5 值 | 下载 |
|---|---|---|---|
| 1 | multipath-tools-Linx80.tar.gz | d0a108425bcc78ea4a4a438b638db1ca | 点击下载 |
| 官网下载地址:<br>http://www.linx-info.com/download/software/linx-mp/80/multipath-tools-Linx80.tar.gz | | | |

适用于凝思安全操作系统 V6.0.80 non-root 环境：

| 序号 | 双机包名称 | MD5 值 | 下载 |
|---|---|---|---|
| 1 | Linx_LXHA_V1.0.0.20190410+Linx6.0.80_NON-ROOT_amd64.tar.gz | b18931e5b4485150968873cc2cad8a04 | 点击下载 |
| 官网下载地址:<br>http://www.linxinfo.com/download/software/LXHA/Linx_LXHA_V1.0.0.20190410+Linx6.0.80_NON-ROOT_amd64.tar.gz | | | |

适用于凝思安全操作系统 V6.0.80 root 环境：

| 序号 | 双机包名称 | MD5 值 | 下载 |
|---|---|---|---|
| 1 | Linx_LXHA_V1.0.0.20190412+Linx6.0.80_ROOT_amd64.tar.gz | 95d552176ae4cd4f8a1ace8192926960 | 点击下载 |
| 下载地址: | | | |

## 2.3 存储模式确认

针对不同的存储硬件需要进行不同的配置，目前用户使用的存储主要有三种模式，现场存储模式需要和存储厂家工作人员确认：

1. **主备存储模式**，存储路径存在激活状态和就绪状态，当一条路径从激活状态被确定处于故障状态，处于就绪状态的路径将被激活。
2. **双控存储模式**，存储所有路径都处于激活状态，数据流量平均分布在每条路径上，所有路径都参与工作。
3. **双活存储模式**，存在两台双控存储，一台存储处于工作状态，另一台存储的路径也处于工作状态，但是使用优先级低于工作状态存储，当工作状态存储出现故障，备用路径接替 IO 数据工作。

针对存储模式不同，多路径的配置参数也不同，主备模式存储 path_grouping_policy 配置参数为 failover，双控模式存储 path_grouping_policy 配置参数为 multibus，双活存储需要配置参数较多，prio，path_grouping_policy 等参数因产品不同配置参数均不同，需要询问存储厂商。

**注意：如果存储厂商不能确认存储模式时，默认选择双控配置模板进行多路径的配置。**

## 2.4 双机热备需求确认

需要安装双机的两台服务器：

1. 操作系统版本是否均为 Linx 6.0.80 2019-01-30 \n \l，内核版本均为 Linux rocky80-his1 4.9.0-0.bpo.1-linx-security-amd64 #1 SMP Linx 4.9.2-2~bpo8+1linx4 (2018-08-17) x86_64 GNU/Linux。
2. 主网 ip 是否需要更改，心跳 ip 网段的选择以不与现场网段冲突为原则。
3. 机器网口数量不小于 4 个 ，心跳线不少于 2 条。
4. 心跳线 1 两端均为 eth1 网卡，心跳线 2 两端均为 eth3 网卡，确认心跳线为两服务器对应网卡直连。
5. 是否已经安装 HBA 卡，若未安装则需用户联系硬件厂家进行 HBA 卡的安装，HBA 卡到交换机光纤由存储厂家负责连接并确保连接正确。
6. 是否有 ipmi 管理口，是否需要配置 ipmi 资源。
7. 记录用户对于双机热备软件希望测试的功能。

## 2.5 风险以及影响范围

根据此指导书进行凝思 V6.0.80 系统双机热备和多路径软件安装时，请确认操作系统版本为 Linx 6.0.80 2019-01-30 \n \l，及内核版本 4.9.0-0.bpo.1-linx-security-amd64 #1 SMP Linx 4.9.2-2~bpo8+1linx4 (2018-08-17) x86_64 GNU/Linux。

指导书对应的双机热备软件支持主备、双活、双控等存储，支持插拔光纤测试。

# 3 凝思多路径软件安装配置

进行双机多路径安装人员必须严格按照本章节进行安装操作。安装过程分以下几步：

1. 安装多路径软件包
2. 配置多路径序列号
3. 配置多路径软件
4. 如果需要参考配置逻辑卷
5. 多路径基本功能的验证

## 3.1 多路径软件包的安装

需要在双机两台节点安装多路径软件。

1. 解压安装包

```
#tar -xf multipath-tools-Linx80.tar.gz
```

2. 进入目录解压 lxmp-80-deb ，执行 install.sh 脚本

```
#cd lxmp-80-deb
#./install.sh
```

3. 检查安装

```
LinxA:~# dpkg -l|grep multipath
ii  multipath-tools                0.6.4-5linx                amd64        maintain
multipath block device access
ii  multipath-tools-boot           0.6.4-5linx                all          Support booting
from multipath devices
```

## 3.2 配置多路径序列号

1. 序列号的申请，多路径安装成功需要填写正确的序列号才能使多路径正常工作，实施人员应根据协调单进行序列号的申请。
2. 序列号的配置文件 multipath_sn.conf 位于/etc/linxsn/目录下，，将序列号添加到 multipath_sn.conf 中“linx_serial=”的后面，以实际申请的序列号为准，格式如下：

   注：凝思多路径软件序列号为 16 进制 0-9 A-F。

```
linx_serial=3F925100930D9AE7
```

## 3.3 多路径软件的配置

针对不同的存储硬件多路径需要不同的配置模板，目前存储主要有三种模式。多路径的配置模板根据存储硬件的不同分为：主备、双控和双活存储配置模板，参考《凝思多路径软件（适用于凝思安全操作系统 V6.0.80）安装配置手册 V1.0(20191011)》。

现场实施人员必须要和存储厂家确认存储模式，并根据存储模式来选择多路径的配置模板，如果存储厂家不能确定存储模式时，默认选择双控模板来进行多路径的配置。

### 3.3.1 wwid 获取

获取 wwid 的命令:

```
LinxA:~ # /lib/udev/scsi_id -g -u /dev/sda
```

### 3.3.2 多路径配置模板

介绍几种常见的存储对应的多路径配置。

#### 3.3.2.1 双控配置模板

普通双控模板:双控存储配置的 path_grouping_policy 参数必须为 multibus。

```
defaults {
        failback immediate
        no_path_retry  30
        rr_min_io   100
        polling_interval    10
        queue_without_daemon   no
            getuid_callout "/lib/udev/scsi_id -g -u /dev/%n"
            hardware_handler   "0"
            user_friendly_names yes
            path_selector "round-robin 0"
            path_grouping_policy  multibus
            path_checker  directio
}

blacklist {
          wwid 36003048002b6ba001ea7da5e0ae3f14e
      }

multipaths {

      multipath {
                wwid 3600b342abdf1052d27f1db4e6d0000db
                alias mpathb
            }
}
```

示例:华为双控存储多路径输出信息显示四条路径可用, 内容如下:

```
root@LinxA :~ # multipath -ll
mpathc (36446a2e100f573590046856000000000) dm-0 HUAWEI,XSG1
size=11T features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=4 status=active
|- 0:0:0:1 sda 8:0 active ready running
```

```
|- 0:0:1:1 sdc 8:32 active ready running
|- 8:0:0:1 sde 8:64 active ready running
`- 8:0:1:1 sdg 8:96 active ready running
mpathb (36446a2e100f5735900469cd300000001) dm-1 HUAWEI,XSG1
size=2.0T features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=4 status=active
|- 0:0:0:2 sdb 8:16 active ready running
|- 0:0:1:2 sdd 8:48 active ready running
|- 8:0:0:2 sdf 8:80 active ready running
`- 8:0:1:2 sdh 8:112 active ready running
```

### 3.3.2.2 双活配置模板

双活存储各厂商差异较大，多路径配置文件和输出信息也不同。

以下为双活存储多路径配置文件模板:

```
defaults {
        failback immediate
        no_path_retry  30
        rr_min_io   100
        polling_interval   10
        queue_without_daemon   no
            getuid_callout "/lib/udev/scsi_id -g -u /dev/%n"
            hardware_handler   "0"
            user_friendly_names yes
            path_selector "round-robin 0"
            path_grouping_policy  multibus
            path_checker  directio
}

blacklist {
          wwid 36003048002b6ba001ea7da5e0ae3f14e
      }

multipaths {

      multipath {
                wwid 3600b342abdf1052d27f1db4e6d0000db
                alias mpathb
            }
}
```

示例:华为双活存储多路径输出信息显示八条路径可用，内容如下：

```
LinxA :~ # multipath -ll
```

```
mpathb (36c81fbe1000b4f540a2de4a500000000) dm-0 HUAWEI,XSG1
size=20T features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
|- 1:0:0:1 sdb 8:16 active ready running
|- 6:0:0:1 sdj 8:144 active ready running
|- 1:0:1:1 sdd 8:48 active ready running
|- 6:0:1:1 sdl 8:176 active ready running
|- 1:0:2:1 sdf 8:80 active ready running
|- 6:0:2:1 sdn 8:208 active ready running
|- 1:0:3:1 sdh 8:112 active ready running
`- 6:0:3:1 sdp 8:240 active ready running
mpatha (36c81fbe1000b4f540a2de64000000001) dm-1 HUAWEI,XSG1
size=8.4T features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
|- 1:0:0:2 sdc 8:32 active ready running
|- 6:0:0:2 sdk 8:160 active ready running
|- 1:0:1:2 sde 8:64 active ready running
|- 6:0:1:2 sdm 8:192 active ready running
|- 1:0:2:2 sdg 8:96 active ready running
|- 6:0:2:2 sdo 8:224 active ready running
|- 1:0:3:2 sdi 8:128 active ready running
`- 6:0:3:2 sdq 65:0 active ready running
```

### 3.3.2.3 主备配置模板

主备存储没有配置 path_grouping_policy 参数为 multibus，系统默认为 failover，以下为主备存储多路径配置文件模板。

```
defaults {
        failback immediate
        no_path_retry  30
        rr_min_io   100
        polling_interval    10
        queue_without_daemon   no
       getuid_callout "/lib/udev/scsi_id -g -u /dev/%n"
       hardware_handler   "0"
```

```
    user_friendly_names yes
    path_selector       "round-robin 0"
    path_grouping_policy  failover
    path_checker  directio
}

blacklist {
        wwid 360030480002b6ba001ea7da5e0ae3f14e
    }
multipaths {
    multipath {
                wwid 3600b342abdf1052d27f1db4e6d0000db
                alias mpathb
            }
}
```

以下为主备存储多路径输出信息

```
LinxA :~ # multipath -ll
mpathc (36006016008c03800a87e5b83a129e411) dm-8 DGC,VRAID
size=100G features='1 queue_if_no_path' hwhandler='1 emc' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| |- 1:0:0:8  sdk  8:160   active ready running
| `- 1:0:1:19 sdbt 68:112  active ready running
`-+- policy='round-robin 0' prio=0 status=enabled
  |- 1:0:1:8  sdbi 67:192  active ready running
  `- 1:0:0:45 sdav 66:240  active ready running
mpathb  (36006016015c0380065828e97a87fe611) dm-71 DGC,VRAID
size=1.0T features='1 queue_if_no_path' hwhandler='1 emc' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| |- 1:0:0:44 sdau 66:224  active ready running
| `- 1:0:0:45 sdav 66:240  active ready running
`-+- policy='round-robin 0' prio=0 status=enabled
  |- 1:0:1:44 sdcs 70:0    active ready running
  `- 1:0:0:20 sdw  65:96   active ready running
```

### 3.3.2.4 宏衫存储模板

因宏杉存储硬件同其他厂商存储不同，配置文件差异较大（也有使用非宏杉模板情况），path_grouping_policy 使用 group_by_prio，并且 prio 设置 alua，path_checker 设置 tur。**经分析宏杉存储目前三种类型存储主备、双控、双活使用的配置文件相同。**

以下为宏杉存储多路径配置文件模板。

```
defaults {
      failback 15
      no_path_retry      30
       polling_interval    10
       checker_timeout 120
       queue_without_daemon   no
       getuid_callout "/lib/udev/scsi_id -g -u /dev/%n"
      hardware_handler   "0"
       flush_on_last_del      yes
      path_selector       "round-robin 0"
      path_grouping_policy group_by_prio
      path_checker       tur
      prio   alua
}

blacklist {
          wwid 36003048002b6ba001ea7da5e0ae3f14e
      }

multipaths {

      multipath {
                wwid 3600b342abdf1052d27f1db4e6d0000db
                alias mpathb
            }
}
```

以下为宏杉主备存储多路径输出信息，两条路径状态为 active，另外六条状态为 enabled。

```
mpathc (3600b342c7dc9e39de88ed94a0d0000df) dm-2 MacroSAN,LU
size=5.6T features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 7:0:0:2 sdd  8:48   active ready running
| `- 8:0:0:2 sdl  8:176  active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 7:0:1:2 sdh  8:112  active ready running
  |- 7:0:2:2 sdt  65:48  active ready running
  |- 7:0:3:2 sdx  65:112 active ready running
 |- 8:0:1:2 sdp  8:240  active ready running
  |- 8:0:2:2 sdab 65:176 active ready running
  `- 8:0:3:2 sdaf 65:240 active ready running
mpathb (3600b342351597cdd50b5db26bd0000df) dm-1 MacroSAN,LU
size=5.6T features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 7:0:3:1 sdw  65:96  active ready running
| `- 8:0:3:1 sdae 65:224 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
 |- 7:0:0:1 sdc  8:32   active ready running
  |- 7:0:1:1 sdg  8:96   active ready running
  |- 7:0:2:1 sds  65:32  active ready running
  |- 8:0:0:1 sdk  8:160  active ready running
  |- 8:0:1:1 sdo  8:224  active ready running
  `- 8:0:2:1 sdaa 65:160 active ready running
```

以下为宏杉双控存储多路径输出信息，宏杉双控存储也有使用非宏杉配置文件模板的情况。

```
mpathc (3600b342c5c84119dd609d69a3d0000dc) dm-1 MacroSAN,LU
size=8.4T features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 1:0:0:1 sdc 8:32  active ready running
  |- 6:0:0:1 sdg 8:96  active ready running
  |- 1:0:1:1 sde 8:64  active ready running
  `- 6:0:1:1 sdi 8:128 active ready running
mpathb (3600b34216e5b7cada118d1872d0000dc) dm-0 MacroSAN,LU
```

```
size=2.2T features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
 |- 1:0:0:0 sdb 8:16  active ready running
 |- 6:0:0:0 sdf 8:80  active ready running
 |- 1:0:1:0 sdd 8:48  active ready running
 `- 6:0:1:0 sdh 8:112 active ready running
```

### 3.3.3 多路径配置文件概述

1. 多路径软件配置文件属于文本文件, 其内容按行排列, 注释以符号#开头。其内容格式如下：

```
<section> {
 <attribute> <value>
 ...
 <subsection> {
     <attribute> <value>
      ...
  }
}
```

section 称为部分, attribute 是该部分的属性, value 是属性值。subsection 是部分的子部分, 子部分中同样包含属性和属性值。

多路件软件配置文件包含五个部分:

- defaults, 默认配置。
- backlist, 不被视为多路径的指定设备列表。
- blacklist_exceptions, 多路径设备列表, 即使将这些设备放置在 blacklist 部分。
- multipaths, 独立多路径设备的特定属性设置。
- devices, 独立存储控制器的设置。

说明：multipaths 部分的参数值会覆盖在 defaults 和 devices 部分设置的参数值，devices 部分的参数值会覆盖在 defaults 部分设置的参数值。如果使用的不是多路径软件默认支持的存储阵列, 则需要为这些存储阵列创建 devices 部分。

2. 配置设备黑名单

多路径配置文件中的 blacklist 部分指定在系统配置多路径设备时不能使用的存储设备。黑名单中的设备将无法分组到多路径设备中。

- 根据 WWID 将设备列入黑名单

  可根据全球标识符将单独的设备列入黑名单, 即在配置文件的 blacklist 部分加入 wwid 条目。

  例 1. 将 wwid 为 26358900f02796769 的设备列入黑名

  ```
  blacklist {
       wwid 26358900f02796769
  }
  ```

- 根据设备名称将设备列入黑名单

  可根据设备名称将设备列入黑名单, 即在配置文件 blacklist 部分指定 devnode 条目以便不会将它们分组到多路径设备中。

例 2. 将所有的 SCSI 设备列入黑名单

```
blacklist {
        devnode "^sd[a-z]"
}
```

不推荐这种做法,因为除非通过 udev 规则做静态映射,否则无法保证在系统重启后每个设备的名称保持原样。例如:重启后,设备的名称可能从/dev/sda 为/dev/sdb。

### 3.3.4 启停多路径及运维命令

1. linx8.0 系统启动、停止和查看状态命令:

```
启动多路径服务使用  systemctl start multipathd.service
停止多路径服务使用  systemctl stop multipathd.service
查看路径服务状态使用 systemctl status multipathd.service
```

2. 查看多路径状态

```
# multipath -ll
mpathc (23366303330383335) dm-2 SCST_FIO,HEARTBEAT
size=953M features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=2 status=active
|- 0:0:1:2 sdk 8:160 active ready running
`- 1:0:1:2 sdc 8:32 active ready running
mpathb (26462383331613865) dm-1 SCST_FIO,DBBAK
size=19G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=2 status=enabled
|- 0:0:1:1 sdh 8:112 active ready running
`- 1:0:1:1 sdb 8:16 active ready running
```

3. 删除现有没有使用的路径

```
LinxA :~ # multipath -F
```

4. 查看多路径映射出来的设备

```
LinxA :~ # ls /dev/mapper
```

## 3.4 逻辑卷的配置方法

1. 创建新的物理卷

使用 pvcreate 命令,针对多路径的别名创建物理卷;使用 pvdisplay 或 pvscan 命令,检查创建的物理卷是否正确。

```
LinxA :~ # pvcreate /dev/mapper/mpathb
Physical volume "/dev/mapper/mpathb"successfully created
LinxA :~ # pvcreate /dev/mapper/mpathc
Physical volume "/dev/mapper/mpathc"successfully created
```

2. 创建卷组

使用 vgcreate 命令,创建卷组;使用 vgdisplay 命令,检查创建的卷组是否正确。如下,使用目前存在的 mpathb、mpathc 物理卷分别创建名叫 vg01 和 vg02 的卷组。

```
LinxA :~ # vgcreate vg01  /dev/mapper/mpathb
Volume group "vg01"successfully created
LinxA :~ # vgcreate vg02 /dev/mapper/mpathc
Volume group "vg02"successfully created
```

3. 创建逻辑卷

使用命令 lvcreate 命令,可以创建逻辑卷;使用 lvdisplay 命令,检查创建的逻辑卷是否正确。如下,在卷组 vg01 和 vg02 上创建一个名为 lv01 和 lv02 的逻辑卷。

```
LinxA :~ # lvcreate --name lv01 -l +100%FREE vg01
Logical volume "lv01"created
LinxA :~ # lvcreate --name lv02 -l +100%FREE vg02
Logical volume "lv02"created
```

4. 手动挂载逻辑卷测试

a. 在两台主机上创建目录/dbdata、/dbbak 用来挂载刚创建的逻辑 lv01、lv02。

```
LinxA :~ # mkdir /dbdata
LinxA :~ # mkdir /dbbak
```

b. 格式化逻辑卷

```
LinxA :~ # mkfs.ext3 /dev/vg01/lv01
LinxA :~ # mkfs.ext3 /dev/vg02/lv02
```

或

```
LinxA :~ # mkfs.xfs /dev/vg01/lv01
LinxA :~ # mkfs.xfs /dev/vg02/lv02
```

c. 挂载逻辑卷到指定目录（**注意存储盘只能在一台机器上挂载，不能两台同时挂载使用**）

```
LinxA :~ # mount /dev/vg01/lv01 /dbdata
LinxA :~ # mount /dev/vg02/lv02 /dbbak
```

d. 使用 df 或者 mount 命令查看挂载信息

5. 由于多路径软件是在 lvm 后面启动的,如果不在 rc.local 中重新扫描逻辑卷,就会影响逻辑卷的正常识别和使用。修改/etc/rc.local 文件末尾添加扫描逻辑卷的命令:

```
#/sbin/vgscan
#/sbin/vgchange -a y
```

6. 如果对逻辑卷创建了分区还要在/etc//rc.local 文件添加扫描逻辑卷分区的命令:

```
LinxA :~ # kpartx -a /dev/mapper/卷组名  1-逻辑卷名*
LinxA :~ # kpartx -a /dev/mapper/卷组名  2-逻辑卷名*
```

示例:

```
LinxA :~ # kpartx -a /dev/mapper/vg01-lv01*
LinxA :~ # kpartx -a /dev/mapper/vg02-lv02*
```

7. 手动挂载测试无异常之后, 需要将手动挂载的逻辑卷设备全部卸载, 之后在双机的配置过程中将其作为存储资源由双机自动加载。

# 4 验证多路径基本功能

## 4.1 映射设备读写文件测试

附:模拟数据读写的脚本:

1. 模拟数据读脚本 1:test-read.sh:

```
#! /bin/bash
while true;do
      echo "$1 test read"
      dd if=/$1/test-700 of=/dev/null
done
```

2. 模拟数据读脚本 2:run-read.sh:

```
#!/bin/bash
disks="dbdata dbbak" # 此处列表为存储盘挂载目录名
for i in $disks
do
      ./test-read.sh $i &
done
```

3. 模拟数据写脚本 1:test-write.sh

```
#! /bin/bash
while true;do
      echo "$1 test write"
      dd if=/dev/zero of=/$1/test-700 bs=4k count=100000 oflag=direct
done
```

4. 模拟数据写脚本 2:run-write.sh

```
#!/bin/bash
disks="dbdata dbbak"
for i in $disks
do
      ./test-write.sh $i &
done
```

5. 脚本使用

测试过程中将存储挂载到对应的目录, 并根据现场情况修改 disks 列表中的挂载目录名称, 运行如下命令开始模拟数据读写, 在存储盘有数据读写的情况下进行拔光纤测试:

```
# ./run-read.sh
# ./run-write.sh
```

如何停止脚本:

```
# killall test-read.sh
# killall test-write.sh
```

**注意：读写脚本具有破坏性，对于刚搭建完可以进行测试，有数据上线使用的机器需经过组长以及经理确认后在进行测试。**

## 4.2 拔插光纤测试

- 在存储盘有读写的情况下,插拔部分光纤,不影响数据读写。
- 在存储盘有读写的情况下,插拔全部光纤,数据读写阻塞,待光纤恢复连接,读写恢复。

A. 执行命令 multipath -ll 查看当前路径状态,输出如下信息,表明路径状态正常。

```
# multipath -ll
mpathe (360060e801013b430058b3b1300000003) dm-3 HITACHI,DF600F
[size=3.1T][features=1 queue_if_no_path][hwhandler=0][rw]
\_ round-robin 0 [prio=1][active]
\_ 6:0:0:3 sdm 8:192 [active][ready]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:1:3 sdi 8:128 [active][ready]
\_ 6:0:1:3 sdq 65:0 [active][ready]
```

B. 加载测试用逻辑分区

```
LinxA :~ # mount /dev/vg01/lv01 /dbbak
```

C. 在逻辑卷分区执行 run-read.sh 和 run-write.sh 脚本,然后新打开一个终端,通过命令 iostat 1 来查看数据 IO 写入的状态,可看到数据在 sde、sdm 这两条路径上进行 IO 操作,如下所示。

```
avg-cpu: %user %nice %system %iowait %steal %idle
         0.03  0.00   2.63   5.26    0.00  92.39
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
dm-3   405.00   0.00   415344.00    0      415344
dm-5  49792.00  0.00   398336.00    0      398336
sdm   352.00    0.00   359424.00    0      359424
sdi    0.00      0.00   0.00           0       0
sde   55.00     0.00   56320.00     0      56320
sdq   0.00      0.00   0.00           0       0
```

D. 拔掉一条线路来模拟路径出现问题,路径将发生改变,显示如下路径信息,可以看到路径 sde 已经被删除:

```
# multipath -ll
mpathe (360060e801013b430058b3b1300000003) dm-3 HITACHI,DF600F
[size=3.1T][features=1 queue_if_no_path][hwhandler=0][rw]
\_ round-robin 0 [prio=1][active]
\_ 6:0:0:3 sdm 8:192 [active][ready]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:1:3 sdi 8:128 [active][ready]
\_ 6:0:1:3 sdq 65:0 [active][ready]
```

E. iostat 输出信息如下,sde 路径同样也被删除。

```
avg-cpu: %user %nice %system %iowait %steal %idle
         0.06  0.00  5.92     10.43   0.00  84.22
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
dm-3    396.00   0.00  401808.00     0      401808
dm-5   49280.00  0.00  394240.00     0      94240
sdi       0.00     0.00    0.00           0       0
```

```
sdq      0.00     0.00    0.00          0       0
sdm    394.00     0.00  401408.00     0     401408
```

F. 从 iostat 和 multipath -ll 的输出信息即可判断多路径工作是否正常。

# 5 凝思双机热备软件的安装配置

多路径功能测试正常之后，根据现场要求选择安装双机热备软件，部署时请严格按照以下步骤进行：

1. 双机热备软件包的安装。
2. 两台机器分别安装双机软件。
3. 双机热备软件需要正确的序列号才能运行，根据协调单申请双机热备序列号并按照要求进行配置。
4. 双机热备环境的搭建，包括网络环境、节点名、hosts 解析、时间同步、引擎配置、stonith 配置、资源管理软件配置、日志轮转配置等。
5. 以下具体介绍双机热备软件安装配置的步骤。

## 5.1 双机热备软件包的安装

### 5.1.1 适用凝思安全操作系统 V6.0.80 root 环境

1. 适用于凝思安全操作系统 V6.0.80 root 环境软件包：

```
Linx_LXHA_V1.0.0.20190412+Linx6.0.80_ROOT_amd64.tar.gz
```

2. 解压安装包，进入解压后的目录

```
# tar  -vxf  Linx_LXHA_V1.0.0.20190412+Linx6.0.80_ROOT_amd64.tar.gz
```

3. 检查系统中是否已经安装 HA80-others.tar.gz 凝思双机热备软件包。如果已经安装，进入 lxha 目录，运行 uninstall.sh 脚本，卸载已安装的凝思双机热备软件。如果没有安装，跳过此步骤。

```
# ./uninstall.sh
```

4. 运行 install.sh 脚本，安装凝思双机热备软件。执行脚本过程，如询问选择（y/n）时，均选择 y。

```
# ./install.sh
```

5. 使用 dpkg 命令检查所安装软件包版本是否正确。

```
root@LinxA:~# dpkg -l |grep -E "cluster-glue|corosync|crmsh|libqb|lxha-stonith|pacemaker|
resource-agents"
ii  cluster-glue                          1.0.12-5                  amd64
LXHA_V1.0.0.20190412 for Linx OS V6.0.80 root
ii  corosync                              2.4.2-3                   amd64
LXHA_V1.0.0.20190412 for Linx OS V6.0.80 root
pi  corosync-dev                           1.4.6-1.1                   amd64        Standards-based
cluster framework (generic devel files)
ii  crmsh                                 2.3.2-4                   amd64
LXHA_V1.0.0.20190412 for Linx OS V6.0.80 root
ii  libqb                                 1.0.1-1                   amd64
LXHA_V1.0.0.20190412 for Linx OS V6.0.80 root
ii  lxha-stonith                           2.0-2                     amd64
```

```
LXHA_V1.0.0.20190412 for Linx OS V6.0.80 root
ii  pacemaker                          1.1.16-1                amd64
LXHA_V1.0.0.20190412 for Linx OS V6.0.80 root
ii  resource-agents                    4.0.0                   amd64
LXHA_V1.0.0.20190412 for Linx OS V6.0.80 root
```

### 5.1.2 适用凝思安全操作系统 V6.0.80 non-root 环境

1. 适用于凝思安全操作系统 V6.0.80 non-root 环境软件包 ：

```
Linx_LXHA_V1.0.0.20190410+Linx6.0.80_NON-ROOT_amd64.tar.gz
```

2. 在 non-root 环境下运行的双机热备软件默认在 sysadmin 用户下运行。**如果系统中没有 sysadmin 用户，需要安装 non-root 升级包，进行无 root 化升级。**

3. 解压安装包，进入解压后的目录

```
# tar -zvxf Linx_LXHA_V1.0.0.20190410+Linx6.0.80_NON-ROOT_amd64.tar.gz
```

4. 安装 kernel 补丁，进入 kernel_patch 目录，运行 install.sh 脚本

```
#./install.sh
```

5. 安装完成后，重启主机。以 root 用户登录（如果不能以 root 用户登录，则以 sysadmin 用户登录），如果弹出的序列号验证窗口，输入凝思操作系统序列号，完成验证；进入系统，打开终端，在 root 用户下，执行 reboot，再次重启主机。

6. 开机后登录系统，查看/proc/sys/kernel/sysrq 文件权限，应该为 664。查看/etc/group 文件，确认 sysadmin 用户在 root 组中。

7. 安装双机热备软件

- 卸载 HA80-others.tar.gz 双机热备软件。进入 lxha 目录中，如果系统已经安装了 HA80-others.tar.gz 双机热备软件包，运行 uninstall.sh 脚本，卸载已安装的双机热备软件，如果没有安装，跳过此步骤。

```
#./uninstall
```

- 运行 install.sh 脚本，安装双机热备软件。执行脚本过程，如询问选择(y/n)时，均选择 y。

```
#./install
```

8. 安装 systemd 补丁，进入 systemd_patch 目录中，运行 install.sh 脚本，安装 systemd 补丁。

```
# ./install.sh
```

9. 安装 systemd 补丁，进入 systemd_patch 目录中，运行 install.sh 脚本，安装 systemd 补丁。

```
# ./install.sh
```

10. 使用 dpkg 命令检查所安装软件包版本是否正确。

```
root@rLinxA:~# dpkg -l |grep -E "cluster-glue|corosync|crmsh|libqb|lxha-stonith|pacemaker|
resource-agents"
ii  cluster-glue                       1.0.12-5                amd64
LXHA_V1.0.0.20190410 for Linx OS V6.0.80 non-root
ii  corosync                           2.4.2-3                 amd64
LXHA_V1.0.0.20190410 for Linx OS V6.0.80 non-root
ii  corosync-dev                       1.4.6-1.1               amd64        Standards-based
```

```
cluster framework (generic devel files)
ii  crmsh                                  2.3.2-4                    amd64
LXHA_V1.0.0.20190410 for Linx OS V6.0.80 non-root
ii  libqb                                  1.0.1-1                    amd64
LXHA_V1.0.0.20190410 for Linx OS V6.0.80 non-root
ii  lxha-stonith                           2.0-2                      amd64
LXHA_V1.0.0.20190410 for Linx OS V6.0.80 non-root
ii  pacemaker                              1.1.16-1                   amd64
LXHA_V1.0.0.20190410 for Linx OS V6.0.80 non-root
ii  resource-agents                        4.0.0                      amd64
LXHA_V1.0.0.20190410 for Linx OS V6.0.80 non-root
```

## 5.2 存储的 IO 调度算法配置

对于数据库服务器，deadline 算法是推荐的，经过工程验证，由于插拔 HBA 卡的光纤线，会引起阵列磁盘盘符的变化，应该将该算法放到 grub 文件中：

1. 修改 /etc/default/grub 文件，在 GRUB_CMDLINE_LINUX 行参数中，增加 elevator=deadline 参数，如：

```
GRUB_CMDLINE_LINUX="linx_serial=32339CEDD2D34779   security=linx
rootflags=data=writeback elevator=deadline"
```

2. 运行 update-grub 命令，自动修改 /boot/grub/grub.cfg 文件，使其生效

## 5.3 双机热备的序列号配置

1. 序列号的申请,多路径安装成功需要填写正确的序列号才能使多路径正常工作,实施人员需根据协调单进行双机热备序列号的申请。
2. 创建/etc/linxsn/HA_sn.conf 文件，并修改内容。

```
# mkdir -p /etc/linxsn
# touch /etc/linxsn/HA_sn.conf
```

例如：文件内容修改为 linx_serial=3F925100930D9AE7（等号后为凝思软件序列号，以实际申请的序列号为准），在另一台机器上重复前面的操作。

注：凝思双机热备软件的序列号为 16 进制 0-9，A-F。

```
linx_serial=3F925100930D9AE7
```

## 5.4 双机热备的环境搭建

6.0.80 系统 root 和 non-root 双机环境搭建步骤基本相同，可以参照如下配置。

### 5.4.1 双机热备的网络环境

在下面的配置过程中，假设运行双机热备集群的两台服务器(High Availability)以 下简称节点机)的主机名分别为 LinxA 和 LinxB。双机系统的网络配置主要是配置两个节点机上网卡的 IP 地址，其配置规则如下：

- 网卡 eth0 为局域网中使用的网址(根据实际情况进行配置)；
- 网卡 eth1，网卡 eth2 是双机软件检测对方服务状态的心跳网卡，要求心跳线两侧的网卡配置在同一网段即可；
- 心跳线直连时，心跳网卡的 broadcast 和 gateway 建议不配置；
- 网卡 eth0、网卡 eth1 和网卡 eth2 要配置在不同的网段。

配置示例:
网卡 eth0:
LinxA:172.16.0.100
LinxB:172.16.0.200
网卡 eth1:
LinxA:10.10.10.1
LinxB:10.10.10.2
网卡 eth2:
LinxA:10.10.20.1
LinxB:10.10.20.2
172.16.0.10 为双机对外提供服务的虚拟 IP 地址，在双机配置资源时使用 。

配置完成后执行以下命令使 IP 地址生效，或者重启服务器。

```
# /etc/init.d/networking restart
```

### 5.4.2 节点名及 hosts 解析

根据实际情况，配置双机热备两节点的主机名，主机名通过/etc/hostname 文件来修改；编辑/etc/hosts 将两台节点的 IP 和主机名进行映射。

配置示例：

```
127.0.0.1    localhost
172.16.0.100  LinxA
172.16.0.200  LinxB
```

### 5.4.3 双机节点的时间同步

双机热备引擎(corosync)会根据双机配置对双机系统的心跳信号进行检测，由此实现对双机运行状态的监测，提供集群基础结构(通信和成员资格)服务。所以双机热备集群中的两个节点时间须一致。项目实施中通过配置 ntp 服务来进行时间同步，以下介绍 ntp 服务配置：

1. 修改 LinxA 机器/etc/ntp.conf 文件，将 LinxA 机器配置为对时服务器，需要有如下内容：

```
server  127.127.1.0  iburst       #本地
tos cohort 1
```

```
fudge  127.127.1.0  stratum 3
```

2. 修改 LinxB 机器/etc/ntp.conf 文件，与 LinxA 机器对时，需要有如下内容：

```
server  172.16.0.100  iburst       #LinxA ip 地址
tos cohort 1
fudge  127.127.1.0  stratum 10
```

3. 重启 ntp 服务并设置开机启动链接：

```
/etc/init.d/ntp  restart
systemctl  enable ntp
```

### 5.4.4 双机热备的引擎配置（corosync）

1. 生成 corosync 认证文件

在两台节点机中的任意一台上运行 corosync-keygen 命令生成认证文件 authkey。生成的认证文件保存在/etc/corosync/authkey 中, 然后将 authkey 拷贝到另一台服务器的/etc/corosync 目录下。(High Availability) 注:有时生成认证文件比较缓慢, 可以从键盘上输入任意字符。)

```
LinxA # cd /etc/corosync
LinxA # corosync-keygen
```

2. 配置/etc/corosync/corosync.conf 文件

修改配置模板中的 bindnetaddr 参数, 该参数应该配置心跳 IP 的网络地址。例如, 两节点的一条心跳线两端的 IP 分别是:LinxA:10.10.10.1 、LinxB:10.10.10.2, 子网掩码均是 255.255.255.0 那么两个 IP 地址的网络地址均为 10.10.10.0。

修改该参数为“bindnetaddr: 10.10.10.0”, 另一条心跳线配置类推, 修改为“bindnetaddr:10.10.20.0 ”。

参考 corosync.conf 文件的 man 手册配置模板中的其他参数保持不变, 编辑完成后将/etc/corosync/corosync.conf 文件, 拷贝到另一台服务器的/etc/corosync 目录下。

配置示例：

```
# Please read the corosync.conf.5 manual page
# 80 corosync.conf  V0.1

totem {
    version: 2
    secauth: on
    rrp_mode:passive
    token: 5000
    token_retransmits_before_loss_const: 10

    interface {
        ringnumber: 0
        bindnetaddr: 10.10.10.0
        mcastaddr: 226.94.3.1
        mcastport: 5400
    }
```

```
    interface {
        ringnumber: 1
        bindnetaddr: 10.10.20.0
        mcastaddr: 226.94.4.1
        mcastport: 5405
    }
}

logging {
    fileline: off
    to_stderr: no
    to_logfile: yes
    to_syslog: no
    logfile: /var/log/corosync.log
    debug: off
    timestamp: on
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}

quorum {
    provider: corosync_votequorum
    expected_votes: 2
    two_node: 1
}
```

## 5.4.5 双机热备的 stonith 配置

配置的 stonith 隔离设备是 lxha, 相关配置需要根据 5.3.1 节双机系统的网络配置情况进行。

1. 编辑/etc/corosync/lxha_clientlist.conf 文件, 需要在该文件中增加两行, 分别包括主机和备机的主机名、外网 IP、心跳 1 的 IP、心跳 2 的 IP, 每行的四个参数之间以空格分隔, 行尾不能有空格, 行尾直接回车; 如果只有一条心跳线, 则其中的参数心跳 1 的 IP 和心跳 2 的 IP 可以重复, 每行必须要有 3 个 IP。例如:

```
LinxA 172.16.0.100 10.10.10.1 10.10.20.1
LinxB 172.16.0.200 10.10.10.2 10.10.20.2
```

2. 编辑/etc/corosync/lxha_iplist.conf 文件, 向该文件中添加主机和备机的所有对外的 IP 地址和心跳的 IP 地址;每行一个 IP 地址行尾不能有空格, 每行输入完 1 个 IP 地址后直接回车;否则, 在 lxha 方式触发双机的 stonith 机制时, 会引起不能正常切机的问题。例如:

```
172.16.0.100
10.10.10.1
10.10.20.1
172.16.0.200
```

```
10.10.10.2
10.10.20.2
```

### 5.4.6 启动 corosync 和 pacemaker

在双机两台节点启动 corosync 和 pacemaker

1. 启动 corosync

```
# /etc/init.d/corosync start
Starting Corosync Cluster Engine (High Availability)corosync): [ OK ]
```

2. 启动 pacemaker

```
# /etc/init.d/pacemaker start
Starting Pacemaker Cluster Manager[ OK ]
```

3. 查看心跳状态

```
# corosync-cfgtool -s
Printing ring status.
Local node ID 16843010 #不同双机热备集群 ID 不同
RING ID 0
        id = 10.10.10.1
        status = ring 0 active with no faults
RING ID 1
        id = 10.10.20.1
        status = ring 1 active with no faults
```

- **注意事项**:corosync 要在 pacemaker 之前启动,关闭要先关 pacemaker 后关 corosync;也可以用 service 或 systemctl 命令来起/停双机服务。

### 5.4.7 双机热备的资源管理软件配置（pacemaker）

1. 首先启动双机热备服务,在/tmp 目录下创建一个 pacemaker 的临时配置文件 pacemaker.conf。

```
# touch pacemaker.conf
```

2. 编辑 pacemaker.conf 文件,根据具体情况添加配置项,例如下面的配置模板。

```
配置模板:
primitive dmserver lsb:dmserverd \
        op monitor interval=60s timeout=60s \
        op start timeout=60s \
        op stop timeout=300s \
        meta migration-threshold=3 failure-timeout=300s
primitive rsc-fss ocf:heartbeat:Filesystem \
        op monitor interval=60s timeout=60s OCF_CHECK_LEVEL=20 \
        op start timeout=60s \
        op stop timeout=60s \
        params device="/dev/mpatha" directory="/dbdata" fstype=xfs \
        meta migration-threshold=3 failure-timeout=300s
primitive rsc-vip-public ocf:heartbeat:IPaddr \
        op monitor interval=60s timeout=60s \
```

```
    op start timeout=60s \
    op stop timeout=60s \
    params ip=172.16.0.10 cidr_netmask=255.255.255.0 nic=eth0 broadcast=172.10.0.255 \
    meta migration-threshold=3 failure-timeout=300s
    primitive st-lxha stonith:external/lxha \
    params hostlist="LinxA LinxB"
group rsc-group rsc-vip-public rsc-fss dmserver
clone fencing st-lxha
location rsc-loc1 rsc-group 100: LinxA
location rsc-loc2 rsc-group 100: LinxB
property cib-bootstrap-options: \
    start-failure-is-fatal=false \
    cluster-recheck-interval=600s \
    stonith-enabled=true \
    stonith-action=reboot \
    no-quorum-policy=ignore
```

3. pacemaker.conf 文件编辑完成后，执行下面的命令，导入 pacemaker 配置。

```
LinxA :~ # crm configure load replace /tmp/pacemaker.conf
```

4. 双机资源管理器的配置在两个节点自动同步，所以在两台节点机中任意一台上完成配置即可；配置完成后 crm_mon 可以看到双机热备软件的运行状态。

```
LinxA :~ #crm_mon
Stack: corosync
Current DC: LinxA (version 1.1.16-94ff4df) - partition with quorum
Last updated: Mon Nov  4 15:52:16 2019
Last change: Mon Nov  4 15:52:16 2019  by root via crm_attribute on LinxA

2 nodes configured
10 resources configured

Online: [ LinxA LinxB ]

Active resources:

 Resource Group: rsc-group
    rsc-vip-public (ocf::heartbeat:IPaddr):    Started LinxA
    rsc-fso     (ocf::heartbeat:Filesystem):    Started LinxA
    rsc-fss     (ocf::heartbeat:Filesystem):    Started LinxA
    dmserver (lsb:dmserverd):  Started LinxA
 Clone Set: fencing [st-lxha]
    Started: [ LinxA LinxB ]
```

5. 重启双机热备的两台节点。重启成功后，可以执行如下命令查看和修改 pacemaker 配置。

```
LinxA :~ #crm configure edit
```

## 5.4.8 双机热备 pacemaker 配置说明

本节将基于前面的 pacemaker 配置模板，对 pacemaker 配置进行说明，着重说明配置过程中需要修改的配置项，不需要修改的配置内容请保持与模板一致。本节中，红色字体标记的字段是配置中需要修改的参数。

1. 资源组配置说明

```
group rsc-group rsc-vip-public rsc-fss dmserver
```

“**group**”表示配置一个资源组，资源组 ID 为“rsc-group”，可以自定义，但要避免出现相同的资源 ID。“rsc-vip-public rsc-fss dmserver”表示资源组中包含了 rsc-vip-public、rsc-fss 和 dmserver 三个原始资源。资源组中的原始资源需要根据实际的配置需求进行添加或删除。

资源组中的原始资源，在启动时会按出现的顺序依次启动，在停止时会按相反的顺序停止。如上面的配置，启动时会先启动 rsc-vip-public 资源，然后启动 rsc-fss 资源，最后 启动 dmserver 资源。停止时会先停 dmserver 资源，然后停止 rsc-fss 资源，最后停止 rsc-vip-public 资源。因此，如果资源组中的原始资源的运行存在依赖关系时，要合理安排资源组中各个原始资源的启动顺序。一般是先启动虚拟 IP 资源(rsc-vip-public)，然后启动文件系统资源(rsc-fss)，然后启动数据库资源(dmserver)。资源组中的原始资源总是在同一个节点上运行。

2. 虚拟 IP 资源说明

```
primitive rsc-vip-public ocf:heartbeat:IPaddr \
    op monitor interval=60s timeout=60s \
    op start timeout=60s \
    op stop timeout=60s \
    params ip=172.16.0.10 cidr_netmask=255.255.255.0 nic=eth0
broadcast=172.16.0.255 \
    meta migration-threshold=3 failure-timeout=300s
```

虚拟 IP 也叫浮动 IP。“primitive”表示配置一个原始资源。“rsc-vip-public”是该原始资源的 ID，可以自定义，但要避免出现相同的资源 ID。

虚拟 IP 资源所用的资源代理脚本是/usr/lib/ocf/resource.d/heartbeat/目录下的 IPaddr。在该脚本中可以查看可配置的“params”参数详情。IPaddr 常用的“params”配置项如下。

- **ip**：虚拟 IP 地址，根据客户需求进行配置，是必须配置参数。在同一个双机热备系统中不允许配置两个相同的虚拟 IP。
- **cidr_netmask**：子网掩码，非必须配置参数。
- **nic**：在那个网卡设备上虚拟 IP，不能指定类似“eth0:1”这样的设备，非必须配置参数。
- **broadcast**：广播地址，非必须配置参数。

如果双机热备集群中需要配置多个虚拟 IP，可在临时配置文件 pacemaker.conf 中增加多个上述配置段落。注意需要指定不同的资源 ID，ip 参数不能相同。

3. 文件系统资源说明

```
primitive rsc-fss ocf:heartbeat:Filesystem \
    op monitor interval=60s timeout=60s OCF_CHECK_LEVEL=20 \
    op start timeout=60s \
    op stop timeout=60s \
```

```
    params device="/dev/mpatha" directory="/dbdata" fstype=xfs \
    meta migration-threshold=3 failure-timeout=300s
```

“**primitive**”表示配置一个原始资源。“rsc-fss”是该原始资源的 ID, 可以自定义, 但要避免出现相同的资源 ID。

文件系统资源所用的资源代理脚本是/usr/lib/ocf/resource.d/heartbeat/目录下的 Filesystem。在该脚本中可以查看可配置的“params”参数详情。Filesystem 常用的“params”配置项如下。

- **device**: 文件系统的挂载设备, 是必须配置参数。配置的设备一般是多路径设备, 可以通过“multipath -ll”命令查看。可以用 device=“-U<UUID>”的形式, 直接指定挂载设备的 UUID。
- **directory**: 文件系统的挂载点, 是必须配置参数。
- **fstype**: 指定文件类型, 是必须配置参数。优先配置为“xfs”类型。

如果双机热备集群中需要配置多个文件系统资源, 可在临时配置文件中增加多个上述配段落。注意需要指定不同的资源 ID, 以及不同的 device 和 directory。

4. 数据库资源说明

```
primitive dmserver lsb:dmserverd \
    op monitor interval=60s timeout=60s \
    op start timeout=60s \
    op stop timeout=300s \
    meta migration-threshold=3 failure-timeout=300s
```

这里以达梦数据库为例子。“primitive”表示配置一个原始资源。“dmserver”是该数据库资源的 ID, 可以自定义, 但要避免出现相同的资源 ID。

数据库资源所用的资源代理脚本是 dmserverd, 在/etc/init.d/目录下 , 具体咨询数据库厂家。采用“lsb”(Linux Standard Base)类型资源代理脚本的原始资源, 没有“params”配置参数。

5. lxha-stonith 隔离设备资源说明

```
primitive st-lxha stonith:external/lxha \
params hostlist="LinxA LinxB"

clone fencing st-lxha
```

“**primitive**”表示配置一个原始资源。“st-lxha”是该原始资源的 ID, 可以自定义, 但要避免出现相同的资源 ID。

该隔离设备所用的资源代理脚本是/usr/lib/stonith/plugins/external 目录下的 lxha。在该脚本中可以查看可配置的“params”参数详情。lxha 脚本只需要配置 hostlist 参数。

- **hostlist**: 隔离设备可以控制的主机列表, 是必须配置项。引号中是双机热备两台节点的主机名, 中间以空格隔开。需要注意要按照 5.3.1 节, 正确配置/etc/hosts 和/etc/hostname 文件。

需要将 lxha-stonith 原始资源配置为克隆资源, 让该原始资源在两台节点上均运行。“clone”表示配置克隆资源, “fencing”是克隆资源 ID, 可以自定义, 但要避免出现相同的资源 ID, “st-lxha”表示将前面定义的 st-lxha 原始资源进行克隆。

6. 位置约束配置说明

```
location rsc-loc1 rsc-group 100: LinxA
location rsc-loc2 rsc-group 100: LinxB
```

**location** ”表示配置位置约束，“rsc-loc1”表示位置约束规则的 ID，可以自定义，但要避免出现相同的 ID，“rsc-group”表示受约束的对象是资源组 rsc-group，“100: LinxA”表示资源组 rsc-group 在节点 LinxA 上运行的优先级分数是 100。分数越高，受约束的资源在该节点上运行的优先级越高。

需要配置的是节点名称和优先级分数。节点名称根据双机热备两节点的主机名进行配置。优先级分数根据需要进行配置。例如上述配置，资源组 rsc-group 在节点 LinxA 和 LinxB 上运行的优先级分数是一样的，资源组不会“偏爱”在哪个节点上运行。当修改为“location rsc-loc1 rsc-group 1000: LinxA”时，资源组 rsc-group 将优先在节点 LinxA 上运行。

7. ipmi 隔离设备资源说明

```
primitive ipmiA stonith:external/ipmi \
    op monitor interval=60s timeout=60s \
    params hostname=LinxA ipaddr=172.16.0.102 userid=admin passwd=admin interface=lanplus
primitive ipmiB stonith:external/ipmi \
    op monitor interval=60s timeout=60s \
    params hostname=LinxB ipaddr=172.16.0.103 userid=admin passwd=admin interface=lanplus

location rsc-loc-ipmiA ipmiA -inf: LinxA
location rsc-loc-ipmiB ipmiB -inf: LinxB
```

隔离设备除了前面介绍的 lxha-stonith 外，常用到 ipmi 隔离设备。不可同时配置这两种隔离设备资源。当配置 ipmi 隔离设备资源时，需要配置两个 ipmi 原始资源，分别控制两个节点。

“primitive”表示配置一个原始资源。“ipmiA”是该原始资源的 ID，可以自定义，但要避免出现相同的资源 ID。该隔离设备所用的资源代理脚本是/usr/lib/stonith/plugins/external 目录下的 ipmi。在该脚本中可以查看可配置的“params”参数详情。常用的“params”配置项如下。

- **hostname**：该隔离资源所控制的主机名称，是必须配置参数。
- **ipaddr**：“hostname”参数所指定主机的 ipmi 管理网口的 IP 地址，是必须配置参数。
- **userid**：登录该主机 ipmi 设备的用户名，是必须配置参数。
- **passwd**：登录该主机 ipmi 设备的密码，是必须配置参数。

配置的两行 location 位置约束规则的含义是：控制节点 LinxA 的隔离设备资源 ipmiA 不允许在节点 LinxA 上运行，只能在 LinxB 上运行；控制节点 LinxB 的隔离设备资源 ipmiB 不允许在节点 LinxB 上运行，只能在 LinxA 上运行。

如果要配置 ipmi 的隔离设备，将上述配置段落加入到 pacemaker.conf 文件中。删掉 lxha-stonith 隔离设备的相关配置。

8. pingd 资源说明

```
primitive rsc-pingd ocf:pacemaker:ping \
    op monitor interval=60s timeout=60s \
    op start timeout=60s \
    op stop timeout=60s \
    params host_list="172.16.0.254 " multiplier=100 dampen=5s
name=pingd \
    meta failure-timeout=300s
```

```
clone rsc-pingd-clone rsc-pingd

location rsc-pingd-rule rsc-group \
rule -inf: pingd eq 0
```

有的客户有配置 pingd 资源的需求。“primitive”表示配置一个原始资源。“rsc-pingd”是该原始资源的 ID, 可以自定义, 但要避免出现相同的资源 ID。

**pingd** 资源所用的资源代理脚本是/usr/lib/ocf/resource.d/pacemaker 目录下的 ping。在该脚本中可以查看可配置的“params”参数详情。常用的“params”配置项如下。

- **host_list**: 需要 ping 的节点列表, 如果需要 ping 多个节点, 用空格隔开, 是必须配置参数。

上述配置的含义是, 两个节点均运行 pingd 资源, 会定时去 ping 地址“172.16.0.254”的第三节点(一般为网关机), 如果 ping 不通(该双机节点网络可能存在异常), 则“name”参数 pingd 得分为 0。当 pingd 等于零时, 在该节点运行资源组 rsc-group 的优先级分数为负无穷大, 即不允许在该节点上运行资源组 rsc-group。

如果要配置 ipmi 的隔离设备, 将上述配置段落加入到 pacemaker.conf 文件中。不建议配置 pingd 资源。

## 5.4.9 双机热备的日志轮转

1. 编辑双机热备日志轮转配置文件/etc/logrotate.d/corosync, 在该文件中配置的日志路径为“/var/log/corosync.log”, 修改 corosync.log 的权限为 666, 该日志路径需要与/etc/corosync/corosync.conf 文件中配置的双机日志路径一致;

```
# /etc/logrotate.d/corosync   v 0.1


/var/log/corosync.log {
   missingok
   compress
   delaycompress
   monthly
   rotate 12
   notifempty
   sharedscripts
   nomail
     create 0666 root root
}
```

2. 强制轮转日志, 测试配置是否生效。

```
# logrotate -f /etc/logrotate.d/corosync
```

3. 查看双机热备日志文件 corosync.log 权限, 确认日志文件的所属用户和组与双机热备软件的用户和组相同, 且仅双机热备软件的用户和组有读/写权限。例如双机热备软件在

root 用户和组下运行, 查看 corosync.log 文件权限为如下情况。

```
# ls -l /var/log/corosync.log
-rw-rw---- 1 root root 131091 9 月 28 16:41 corosync.log
```

双机热备软件在 non-root 用户和组下运行, 查看 corosync.log 文件权限为如下情况。

```
# ls -l /var/log/corosync.log
-rw-r--r-- 1 sysadmin sysadmin 1356605 9 月 30 11:27 corosync.log
```

## 5.4.10 重启服务器

重启服务器,让上述配置生效。 双机热备相关服务已配置为开机自启动。

## 5.4.11 双机软件使用

### 5.4.11.1 启动/停止凝思双机热备软件

1. 启动

```
# /etc/init.d/corosync start
# /etc/init.d/pacemaker  start
```

2. 停止

```
# /etc/init.d/pacemaker  stop
# /etc/init.d/corosync  stop
```

3. 可以用 service 和 systemctl 命令来起/停凝思双机热备软件

```
# service  corosync(pacemaker)  start(stop)
# systemctl  start(stop)  corosync.service(pacemaker.service)
```

### 5.4.11.2 查看凝思双机热备系统的运行状态

1. 使用 crm_mon 命令查看，在命令行中执行 crm_mon 即可打印出双机热备系统的运行状态；
2. 使用 ps axu 命令查看系统中的进程。如果双机热备系统运行正常，可以在 ps axu 命令打印出的列表中找到以下进程：

```
/sbin/lxhasrv
/usr/sbin/corosync
/usr/sbin/pacemakerd
/usr/lib/x86_64-linux-gnu/pacemaker/cib
/usr/lib/x86_64-linux-gnu/pacemaker/stonithd
/usr/lib/x86_64-linux-gnu/pacemaker/lrmd
/usr/lib/x86_64-linux-gnu/pacemaker/attrd
/usr/lib/x86_64-linux-gnu/pacemaker/pengine
/usr/lib/x86_64-linux-gnu/pacemaker/crmd
```

3. 查看网络端口。以 root 方式启动系统时，运行 netstat -anp|less 时，能看到如下类似信息：

```
udp        0      0 10.10.10.1:5399     0.0.0.0:*                           1055/corosync
```

```
udp        0      0 226.94.3.1:5400         0.0.0.0:*                           1055/corosync
udp        0      0 10.10.10.15400          0.0.0.0:*                           1055/corosync
udp        0      0 10.10.20.1:5404         0.0.0.0:*                           1055/corosync
udp        0      0 226.94.4.1:5405         0.0.0.0:*                           1055/corosync
udp        0      0 10.10.20.1:5405         0.0.0.0:*                           1055/corosync
```

### 5.4.11.3 查看凝思双机热备运行日志

凝思双机热备的日志记录到/var/log/corosync.log 文件中。

如果/etc/corosync/corosync.conf 配置文件中，*to_syslog: yes*，则双机热备日志还会记录到/var/log/syslog 文件中。

### 5.4.11.4 资源管理

1. 显示资源配置。

```
# crm configure show
```

2. 修改和编辑资源配置。

```
# crm configure edit
```

3. 启动或停止某个资源，比如 rcs-group 资源组。

```
# crm
crm(live)#resource
crm(live)resource#stop rcs-group
crm(live)resource#start rcs-group
```

4. 手动切机和上线。

```
# crm node standby
# crm node online
```

# 6 验证双机基本功能

## 6.1 双机运行状态

1. 通过 crm_mon 命令查看双机状态，以下示例两节点在线状态，,各资源处于 started 状态。

```
root@LinxA:# crm_mon
Stack: corosync
Current DC: LinxA (version 1.1.16-94ff4df) - partition with quorum
Last updated: Mon Nov  4 15:52:16 2019
Last change: Mon Nov  4 15:52:16 2019  by root via crm_attribute on LinxA

2 nodes configured
10 resources configured

Online: [ LinxA LinxB ]

Active resources:

 Resource Group: rsc-group
     rsc-vip-public (ocf::heartbeat:IPaddr):    Started LinxA
     rsc-fso     (ocf::heartbeat:Filesystem):    Started LinxA
     rsc-fss     (ocf::heartbeat:Filesystem):    Started LinxA
     dmserver (lsb:dmserverd): Started LinxA
 Clone Set: fencing [st-lxha]
     Started: [ LinxA LinxB ]
```

2. 通过命令 corosync-cfgtool -s 命令查看心跳状态，两条心跳线显示 ring * active with no faults

```
# corosync-cfgtool -s
Printing ring status.
Local node ID 16843010 #不同双机热备集群 ID 不同
RING ID 0
        id = 10.10.10.1
        status = ring 0 active with no faults
RING ID 1
        id = 10.10.20.1
         status = ring 1 active with no faults
```

## 6.2 手动切机

在两台服务上通过 crm node standby 命令是本集节点离线,查看资源是否可以正常切换到另一台节点上。例如:

```
root@LinxA:# crm node standby
```

```
root@LinxA:# crm_mon
Stack: corosync
Current DC: LinxA (version 1.1.16-94ff4df) - partition with quorum
Last updated: Mon Nov  4 15:54:09 2019
Last change: Mon Nov  4 15:54:02 2019 by root via crm_attribute on LinxA

2 nodes configured
7 resources configured

Node LinxA: standby
Online: [ LinxB ]

Active resources:

 Resource Group: rsc-group
    rsc-vip-public    (ocf::heartbeat:IPaddr):       Started  LinxB
    rsc-fso    (ocf::heartbeat:Filesystem):    Started  LinxB
    rsc-fsa    (ocf::heartbeat:Filesystem):    Started  LinxB
    dmserver (lsb:dmserverd):  LinxB
 Clone Set: fencing [st-lxha]
    Started: [ LinxB ]
```

## 6.3 模拟 lxha 故障切机

1. 检查 crm configure edit 配置　stonith-enabled＝"true"和 stonith-action＝"reboot"。
2. 检查 stonith 服务端口是否开启 netstat -na|grep 11123，11123 端口显示 LISTEN。
3. 手动模拟 lxha 的切机

```
在 LinxA 上执行,LinxB 会重启： /sbin/lxhacli LinxB 的主机名 11123 0 reset
在 LinxB 上执行, LinxA 会重启： /sbin/lxhacli LinxA 的主机名 11123 0 reset
```

若切机失败则检查 hosts 解析、心跳配置、lxha_iplist.conf、lxha_clientlist.conf 等配置。

## 6.4 拔心跳线测试

1. 双机状态运行正常时，查看资源运行节点，拔掉两条心跳线，资源运行节点保持不变，备用节点重启则为功能正常。
2. 若切机失败则检查心跳配置、节点名、hosts 解析、资源配置等。

## 6.5 拔光纤线测试

1. 双机运行正常时，拔一台机器的光纤线全部拔出，观察双机状态是否可以正常切换到备机。
2. 若切机失败则检查双机存储资源配置、lxha_iplist.conf、lxha_clientlist.conf 等配置。

# 7 常见问题解决汇总

## 7.1 文件系统使用

双机默认使用 xfs 文件系统，如果现场用户有要求则跟用户说明使用 xfs 文件系统的好处：

根据国网及其他单位在线双机热备的维护经验，当双机存储资源出现文件系统错误时，ext3、ext4 等文件系统的修复时间根据磁盘容量几个小时到十几个小时不等，且文件系统修复会中断业务。

而 xfs 文件系统在修复时今需要几分钟到十几分钟，业务中断时间远远小于使用 ext3 或者 ext4 文件系统。

所以双机热备使用的存储文件系统格式建议使用 xfs 文件系统。

## 7.2 不重启发现新加存储盘的命令

```
#echo  1 > /sys/class/fc_host/hostX/issue_lip
```

**注意**：此处 X 代表数字，使用 tab 补全可以看到，或者进入/sys/class/fc_host 目录 ls 查看。

## 7.3 XFS 文件系统修复的方法

1. 手动挂载或日志中有文件系统错误记录：

```
# mount /dev/mapper/mpathb   /data
mount: /dev/mapper/mpathb is write-protected, mounting read-only
mount: wrong fs type, bad option, bad superblock on /dev/sdb1, missing codepage or helper program, or other error
In some cases useful info is found in syslog - try
dmesg | tail or so.
```

2. 修复 xfs 文件系统，以下为修复成功

```
# xfs_repair /dev/mapper/mpathb
Phase 1 - find and verify superblock...
Phase 2 - using internal log
        - zero log...
        - found root inode chunk
Phase 3 - for each AG...
        - scan and clear agi unlinked lists...
```

```
        - process known inodes and perform inode discovery...
        - agno = 0
        - agno = 1
        - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
        - setting up duplicate extent list...
        - check for inodes claiming duplicate blocks...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
Phase 5 - rebuild AG headers and trees...
        - reset superblock...
Phase 6 - check inode connectivity...
        - resetting contents of realtime bitmap and summary inodes
        - traversing filesystem ...
        - traversal finished ...
        - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
Maximum metadata LSN (1:39499) is ahead of log (1:2).
Format log to cycle 4.
done
```

3. 如果提示修复失败，可以尝试使用 xfs_repair-L，但可能会造成数据丢失。
4. 执行 xfs_repair -L

```
# xfs_repair -L /dev/sdb1
Phase 1 - find and verify superblock...

Phase 2 - using internal log
        - zero log...
        - found root inode chunk
Phase 3 - for each AG...
        - scan and clear agi unlinked lists...
        - process known inodes and perform inode discovery...
```

```
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
- process newly discovered inodes…
Phase 4 - check for duplicate blocks...
        - setting up duplicate extent list...
        - check for inodes claiming duplicate blocks...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
Phase 5 - rebuild AG headers and trees...
        - reset superblock...
Phase 6 - check inode connectivity...
        - resetting contents of realtime bitmap and summary inodes
        - traversing filesystem ...
        - traversal finished ...
        - moving disconnected inodes to lost+found ...
Maximum metadata LSN (1:39499) is ahead of log (1:2).
Format log to cycle 4.
done
```