

MPP 数据库快速安装指南

北京人大金仓信息技术股份有限公司

目 录

| | |
|--------------------------------|----------|
| 1 前言 | 1 |
| 2 硬件环境检查 | 1 |
| 2.1 内存大小..... | 1 |
| 2.2 CPU 架构..... | 1 |
| 2.3 硬盘容量检查..... | 2 |
| 3 软件环境检查 | 2 |
| 3.1 操作系统..... | 2 |
| 3.2 内核版本..... | 2 |
| 3.3 依赖包..... | 3 |
| 4 关闭 SELinux 和防火墙 | 3 |
| 5 修改主机名和 hosts 文件 | 4 |
| 5.1 修改主机名..... | 5 |
| 5.2 修改 hosts 文件..... | 5 |
| 6 创建用户 | 5 |
| 7 配置 ssh | 6 |

| | |
|---------------------------------|----------|
| 8 挂载硬盘 | 6 |
| 9 配置内核参数 | 6 |
| 9.1 内核参数调整列表 | 7 |
| 9.2 内核参数调整说明 | 7 |
| 10 配置 limit.conf | 8 |
| 11 重启服务器 | 9 |
| 12 获取安装包和 license | 9 |
| 13 安装数据库 | 9 |
| 13.1 安装数据库软件..... | 9 |
| 13.1.1 在 Master 主机上安装集群软件 | 9 |
| 13.1.2 安装到所有主机..... | 10 |
| 13.2 初始化集群 | 11 |
| 13.2.1 创建数据目录..... | 11 |
| 13.2.2 安装 license 文件..... | 12 |
| 13.2.3 创建初始化主机列表文件..... | 13 |
| 13.2.4 创建初始化配置文件..... | 13 |
| 13.2.5 运行初始化工具..... | 14 |

| | |
|------------------------|-----------|
| 13.2.6 设置环境变量..... | 15 |
| 14 数据库安装样例..... | 16 |
| 14.1 安装单机集群..... | 16 |
| 14.2 安装多主机集群..... | 19 |

1 前言

这篇向导主要讲述了如何快速安装、初始化和运行一个 MPP 数据库系统。

本向导假定你已经具备了 Linux/Unix 系统管理，数据库系统管理和 SQL 语言的相关知识和能力。本文档包含如下章节：

- 硬件环境检查
- 软件环境检查
- 关闭 SELinux 和防火墙
- 修改主机名和 hosts 文件
- 创建用户
- 配置 ssh
- 挂载硬盘
- 配置内核参数
- 配置 limit.conf
- 重启服务器
- 获取安装包和 license
- 安装数据库
- 数据库安装样例

2 硬件环境检查

生产环境系统必须满足下面硬件要求：

- 内存大小
- CPU 架构
- 硬盘检查

2.1 内存大小

内存最小值：16 GB

推荐值：128 GB 以上

可使用下面命令查看内存大小：

```
# grep MemTotal /proc/meminfo
```

swap 分区大小推荐为 16 GB。

查看 swap 分区命令：

```
# grep SwapTotal /proc/meminfo
```

也可以使用 free 命令查看内存和 swap 大小。

2.2 CPU 架构

CPU 架构只支持 x86_64。

查看 CPU 架构命令：

```
# uname -m
```

2.3 硬盘容量检查

每台主机最少需要 300MB 来安装 MPP 数据库集群软件。生产环境中，需根据实际数据量大小选择合适的硬盘空间，推荐至少 16GB 以上。

查看硬盘存储命令：

```
# df -h
```

3 软件环境检查

检查下面列出的软件是否满足要求，包括：

- 操作系统
- 内核版本
- 依赖包

3.1 操作系统

操作系统最低支持：

- CentOS 6.0 或者更高
- Red Hat Enterprise Linux (RHEL) 6.0 或者更高
- Linux 6.0 或者更高
- iSoft Server OS 4.2 或者更高
- Kylin 3.2 或者更高

使用下面命令查看当前操作系统版本：

```
# cat /etc/redhat-release
```

凝思系统使用以下命令查看当前操作系统版本：

```
# cat /etc/linux-release
```

麒麟操作系统使用以下命令查看当前操作系统版本：

```
# cat /etc/kylin-release
```

3.2 内核版本

MPP 数据库支持的内核版本参见下表：

| 系统版本 | 内核版本 |
|----------------------------|-------------------------------|
| Red Hat Enterprise Linux 6 | 2.6.32-71.el6.x86_64 or later |

| 系统版本 | 内核版本 |
|----------------------------|---|
| Red Hat Enterprise Linux 7 | 3.10.0-54.0.1.el7.x86_64 or later |
| CentOS 6.0 | 2.6.32-71.el6.x86_64 or later |
| CentOS 7.0 | 3.10.0-54.0.1.el7.x86_64 or later |
| Linux 6.0 | 4.9.0-0.bpo.1.linux-security-amd64 or later |
| iSoft Server OS 4.2 | 3.10.0-957.21.3.el7.1.x86_64 or later |
| Kylin 3.2 | 2.6.32-754.ky3.kb3.pg.x86_64 or later |

查看内核版本命令：

```
# uname -r
```

3.3 依赖包

MPP 数据库需要下面的依赖包，

| |
|-----------------|
| bash |
| ed |
| json-c |
| openssh |
| openssh-clients |
| perl |
| sed |
| sysstat |
| tar |
| vim-minimal |
| zip |
| xfspgrog |
| zlib |

以上依赖包均整合在 MPP 数据库安装包中，无需额外安装。如果依然存在依赖包缺失的情况可以使用 rpm 包方式安装相应的依赖包：

```
# rpm -ivh package_name
```

在凝思系统下也可以使用 deb 包方式安装相应的依赖包：

```
# dpkg -i package_name
```

4 操作系统配置

4.1 关闭 SELinux 和防火墙

集群里所有主机均需要禁用 SELinux。一般情况下，防火墙也要被禁用（例如在 RH

EL6.x、CentOS 6.x 或者 Kylin 3.2 上，使用 iptables 服务）。如果需要安全考虑，可以使用防火墙软件，iptables 和 firewalld 的配置和使用请参照操作系统文档。

凝思系统默认没有安装 SELinux 和配置防火墙，不需要进行关闭操作。

1) 关闭 SELINUX

(1) 查看状态

下面命令检查 SELinux 的状态，以 root 用户执行：

```
# sestatus
SELinuxstatus: disabled
```

(2) 关闭方法

可以编辑 /etc/selinux/config 来禁用 SELinux。以 root 用户修改这个文件中的配置值，并且重启系统：

```
SELINUX=disabled
```

关于更多防火墙信息的，请参见操作系统文档。关于禁用 SELinux 的信息，请参照 SELinux 文档。

2) 关闭防火墙

(1) 对使用 firewalld 的系统（如 CentOS 7、普华），可使用 root 用户运行以下命令检查防火墙的状态：

```
# systemctl status firewalld
```

类似下面这个输出就代表着 firewalld 被禁用：

```
* firewalld.service
Loaded: masked (/dev/null; bad)
Active: inactive (dead)
```

下面命令可禁用 firewalld，以 root 用户登录：

```
# systemctl stop firewalld
# systemctl disable firewalld
```

(2) 对使用 iptables 服务的系统的系统（如 RHEL6.x、CentOS 6.x 和 Kylin 3.2），可使用 root 用户运行以下命令检查 iptables 状态：

```
# /sbin/chkconfig --list iptables
```

这个输出就代表着 iptable 被禁用：

```
iptables 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

下面是禁用 iptables 的一种方法，以 root 用户运行下面的命令，并且重启系统：

```
/sbin/chkconfig iptables off
```

4.2 修改主机名和 hosts 文件

4.2.1 修改主机名

MPP 数据库推荐使用主机名来管理集群内部的主机。

CentOS 7x 、 Red Hat Enterprise Linux 7x、Linux 和普华系统使用 `hostnamectl` 命令设置新主机名，需要 `root` 用户执行命令：

```
# hostnamectl set-hostname new_hostname
```

`new_hostname` 为新主机名。设置完成后重新登陆生效。例如：

```
# hostnamectl set-hostname host2
```

CentOS 6x、Red Hat Enterprise Linux 6x 或者 Kylin 3.2 可直接修改配置参数 `/etc/sysconfig/network`，重启生效。在该文件里找到 `HOSTNAME` 一行，把 `HOSTNAME=` 后面的值改成合适的主机名。例如：

```
HOSTNAME=host1
```

4.2.2 修改 hosts 文件

通常 Master 和 Standby Master 主机都配置外部 IP 和内部 IP，也即内外网分离，外部 IP 不需要添加到 `/etc/hosts` 里。Segment 主机可只配置内部 IP。所有主机的 `/etc/hosts` 文件内容必须保持一致。下面的例子里，集群包含了 4 台主机：

```
192.168.2.113 mdw
192.168.2.114 smdw
192.168.2.115 sdw1
192.168.2.116 sdw2
```

4.3 创建用户

不能以 `root` 身份启动 MPP 数据库。对于生产系统，建议：

- 指定一个系统帐号作为 MPP 数据库安装的属主。
- 总是使用这个帐号启动和管理 MPP 数据库。

在 GNU/Linux 下，可以新建一个用户帐号来运行 MPP 数据库系统，为了方便这里假定是 `gpadmin`，要创建一个新用户，可以 `root` 身份运行如下命令：

```
# useradd -m -r gpadmin
# passwd gpadmin
New password: password
Retype new password: password
```

用户必须有权限去访问安装 MPP 数据库的服务和目录。例如，用户需要能够访问 MPP 数据库的安装目录和数据目录。

集群在部署高可用、配置 ODBC 驱动文件或使用 `gpcgroup` 工具管理 `cgroup` 时，系统用户 `gpadmin` 需要有 `sudo` 免密执行权限。

```
cat >> /etc/sudoers<<EOF
gpadmin ALL=(ALL) NOPASSWD:ALL
```

4.4 配置 ssh

MPP 数据库的管理工具如 `gpinitssystem`、`gpexpand` 等使用 SSH 完成各种管理任务。集群规模比较大时，若每台机器的 SSH 连接数超过允许的最大连接数，可能会出现错误如：`ssh_exchange_identification:Connection closed by remote host`。

此时要修改 SSH 的配置参数 `MaxStartups`，`MaxStartups` 最大允许保持多少个未认证的连接。更新该参数需要修改配置文件 `/etc/ssh/sshd_config`。

若 `MaxStartups` 为一个整数。可直接修改参数值，如：

```
MaxStartups 200
```

若 `MaxStartups` 为 “`start:rate:full`” 这种语法，需要修改为：

```
MaxStartups 10:30:200
```

重启 `sshd` 服务后生效。如：

```
# service sshd restart
```

4.5 挂载硬盘

XFS 文件系统是 MPP 数据库的首选文件系统，在挂载系统前，需要设置如下选项：

```
rw,nodev,noatime,nobarrier,inode64,allocsize=16m
```

参照 `mount` 命令手册，获得这些参数的详细信息。

这些参数可以在 `/etc/fstab` 文件中设置，例如：

```
/dev/data /data xfs nodev,noatime,nobarrier,inode64,allocsize=16m 0 0
```

4.6 配置内核参数

MPP 数据库要正常运行，需要设置操作系统的内核参数。通常来说，如下的参数需要被设置：

共享内存 - MPP 数据库实例需要操作系统中的共享内存大小设置合理。通常来说，很多操作系统的共享内存的默认设置太低，不足以满足 MPP 数据库。关于 MPP 数据库的 `shared_buffers` 参数请参见 MPP 数据库的 SQL 参考手册。

网络优化 - MPP 数据库存储了大量的数据，需要对网络的参数进行调优，优化 MPP 数据库主机之间的网络互连。

内存分配 - MPP 数据库处理排序、聚集等操作时会使用较多内存，要减小内存过度使用带来的风险。

core 参数 - MPP 数据库运行过程中异常终止或崩溃，操作系统会将进程当时的内存状态记录下来，保存到 `core` 文件里，方便问题分析。

4.6.1 内核参数调整列表

使用 root 用户在配置文件 /etc/sysctl.conf 文件里追加下面参数：

```
kernel.shmmax = 500000000
kernel.shmmni = 4096
kernel.shmall = 4000000000
kernel.sem = 250 512000 100 2048
kernel.sysrq = 1
kernel.core_uses_pid = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.msgmni = 2048
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.conf.all.arp_filter = 1
net.ipv4.ip_local_port_range = 10000 65535
net.core.netdev_max_backlog = 10000
net.core.rmem_max = 2097152
net.core.wmem_max = 2097152
vm.overcommit_memory = 2
```

重启或执行下面命令生效：

```
# sysctl -p
```

4.6.2 内核参数调整说明

需要调整的内核参数分为下面几类：

1) 共享内存

kernel.shmmax: 单个共享内存段的最大尺寸

kernel.shmmni: 共享内存段的最大数量

kernel.shmall: 共享内存页数的最大值

kernel.sem: 信号量参数

kernel.sysrq: 可中断的系统挂起，系统因为某种原因已经停止对大部分正常服务的响应，但是系统仍然可以响应键盘的按键中断请求

kernel.msgmnb: 每个消息队列的大小（单位：字节）

kernel.msgmax: 从一个进程发送到另一个进程的消息的最大长度

kernel.msgmni: 消息队列标识的最大数目

2) 网络优化

net.ipv4.tcp_syncookies: 开启 SYN Cookies。当出现 SYN 等待队列溢出时，启

用 cookies 来处理，可防范少量 SYN 攻击

net.ipv4.conf.default.accept_source_route: 禁用所有 IP 源路由

net.ipv4.tcp_tw_recycle: 开启 TCP 连接中 time_wait sockets 的快速回收

net.ipv4.tcp_max_syn_backlog: 表示 SYN 队列的长度，可以容纳更多等待连接的网络连接数

net.ipv4.conf.all.arp_filter: arp 检查，通过了反向路由检查的包才会发出去

net.ipv4.ip_local_port_range: 对外连接端口范围

net.core.netdev_max_backlog: 每个网络接口接收数据包的速率比内核处理这些包的速率快时，允许送到队列的数据包的最大数目

net.core.rmem_max: 为 TCP socket 预留用于接收缓冲的内存最大值（单位：字节）

net.core.wmem_max: 为 TCP socket 预留用于发送缓冲的内存最大值（单位：字节）

3) 内存分配

vm.overcommit_memory: 是 2 时减小内存过度使用的风险，需要同时设置 vm.overcommit_ratio 值。对于 vm.overcommit_ratio 设置的计算，参见 MPP 数据库 SQL 手册中的 gp_vmem_protect_limit

4) core 参数

kernel.core_uses_pid: core 文件的文件名是否添加 pid 作为扩展

为了 MPP 数据库不和其它应用程序产生冲突，MPP 数据库的端口号不要设置在操作系统参数 net.ipv4.ip_local_port_range 之间。例如，如果 net.ipv4.ip_local_port_range= 100 00 65535，那么应该设置 MPP 数据库的端口如下值：

```
PORT_BASE = 6000
MIRROR_PORT_BASE = 7000
REPLICATION_PORT_BASE = 8000
MIRROR_REPLICATION_PORT_BASE = 9000
```

关于更多关于 MPP 数据库的端口信息，参见 gpinitssystem。

4.7 配置 limit.conf

用户限制 - 操作系统的用户资源限制控制一个用户可以启动多少个进程。MPP 数据库要求更高的设置值，允许优化启动更多的进程，打开更多的文件。默认的设置可能导致 MPP 数据库的查询无法执行，报告文件描述符不够等错误。在 /etc/security/limits.conf 中设置如下参数：

```
* soft nfile 65536
* hard nfile 65536
* soft nproc 131072
* hard nproc 131072
```

对 Red Hat 6.x、Centos 6.x 和 kylin，在 /etc/security/limits.d/90-nproc.conf 中的参数设置会覆盖 limits.conf 中的设置，因此需要确保 90-nproc.conf 中这些参数的设置是正确的。Linux 中的 pam_limits 模块会根据 limits.conf 和 90-nproc.conf 中的参数设置，来

确定用户的限制。关于更多 PAM 和用户限制的信息，请参见 PAM 和 pam_limits 的文档。

5 重启服务器

为确保上述参数配置生效，重启各台服务器。

6 获取安装包和 license

请联系技术支持获取安装包和合法的 license。

7 安装数据库

本节讲述如何安装和初始化 MPP 数据库集群。内容包括：

- 安装数据库软件
 - ◆ 在 Master 主机上安装集群软件
 - ◆ 安装到所有主机
- 初始化集群
 - ◆ 创建数据目录
 - ◆ 安装 license 文件或加密狗
 - ◆ 创建初始化主机列表文件
 - ◆ 创建初始化配置文件
 - ◆ 运行初始化工具
 - ◆ 设置环境变量

7.1 安装数据库软件

本章介绍如安装 MPP 数据库系统软件。包含以下主题：

- 在 Master 主机上安装集群软件
- 安装到所有主机

安装数据库软件包括两部分，先在 Master 主机上安装集群软件，然后通过 Master 主机安装到所有主机。

7.1.1 在 Master 主机上安装集群软件

MPP 数据库集群软件要先安装到 Master 主机上，然后由 Master 主机安装到其它主机。以管理员用户 gpadmin 安装到 Master 主机的步骤如下：

- 1、复制 MPP 数据库的安装包到 Master 主机上。

2、运行 MPP 数据库的安装包。

```
# /bin/bash KingbaseAnalyticsDB-V003R002C001B0001-CENTOS6-x86_64.run
```

3、在提示输入目标安装目录时，输入一个新的绝对安装目录，并确保当前用户有写权限。例如：

```
Verifying archive integrity... All good.
Uncompressing KingbaseAnalyticsDB Database
.....
.....
.....
.....
.....
.....
Please input target installation directory:
/home/gpadmin/gpdb
target directory '/home/gpadmin/gpdb' does not exists, create it
(y/N)?
Y
```

4、需要在其他的机器上进行系统配置，并安装 MPP 数据库，请参照 7.1.2 安装到所有主机。

若 sshd 监听的端口号为非默认值 22，需要在 source 文件/home/gpadmin/gpdb/mpp_path.sh 里更改下面内容：

```
export SSH_PORT=55555
```

其中等号右边的值为 sshd 监听的端口号，根据实际填写。/home/gpadmin/gpdb 为集群软件安装目录。

7.1.2 安装到所有主机

运行 gpsegininstall，可以从当前主机拷贝 MPP 数据库的二进制文件到需要安装的主机中。

如果 gpsegininstall 命令执行失败，则需要先交换各主机的密钥。手动交换主机间公钥命令为：

```
# gpssh-exkeys -f hostfile_gpssh_allhosts
```

其中，hostfile_gpssh_allhosts 文件里包含了所有交换公钥的主机列表。另外，hostfile_gpssh_allhosts 文件中的所有 hostname 必须在/etc/hosts 文件中存在，并且主机名对应的 IP 地址不能是 127.0.0.1。

gpsegininstall 需要一个主机列表文件，文件内包括要把数据库软件安装到哪些主机，如：

```
# cat hostlist_segininstall
smdw
sdw1
sdw2
```

执行 gpsegininstall 安装软件：

```
# source /home/gpadmin/gpdb/greenlum-path.sh
```

```
# gpsegininstall -f hostlist_segininstall -u gpadmin -p changeme
```

注意：如果操作系统没有提供 source 命令，例如部分 Kylin 系统，则可以使用点命令执行该 sh 文件，最终执行的命令为 `./home/gpadmin/gpdb/greenlum-path.sh`，在其他使用 source 命令的地方也同样需要使用点命令代替。

其中 `-u` 指定操作系统用户，此选项仅在以 root 身份运行 gpsegininstall 时可用；`-p` 指定该用户密码。可把 changeme 改成实际的密码。建议最佳的安全策略：

- 不要在生成环境中使用默认密码
- 在安装时候之后要立即修改密码

使用以上命令初始化 MPP 数据库后，系统会包含一个预先定义的超级用户 gpadmin，也是操作系统用户。这个用户拥有和管理 MPP 数据库系统的权限。

7.2 初始化集群

MPP 数据库是分布式的，所以初始化一个 MPP 数据库管理系统（DBMS）包括初始化多个独立的 Kingbase 数据库实例（称为 Segment 实例）。

系统所有主机上的每个数据库实例（Master 和所有 Segment）都必须以这种方式初始化，这样它们才能作为一个统一的 DBMS 协同工作。MPP 数据库的工具 gpinitssystem 负责初始化 Master 和所有 Segment 实例，并以正确的顺序启动它们。

在 MPP 数据库系统初始化并启动后，则可以连接到 Master 实例上，开始使用数据库。

初始化一个 MPP 数据库分为如下几个大步骤：

- 1、确保已经完成前面章节的准备任务，创建数据目录，详见 创建数据目录
- 2、安装 license 文件或加密狗。
- 3、创建主机列表文件，包含所有 Segment 主机信息。详见 创建初始化主机列表文件
- 4、创建初始化数据库配置文件。详见 创建初始化配置文件
- 5、在 Master 主机上运行 MPP 数据库初始化工具。详见 运行初始化工具
- 6、初始化完成后设置相应环境变量，详见 设置环境变量

7.2.1 创建数据目录

每一个 MPP 数据库的 Master 或者 Segment 实例都需要在磁盘上划分存储区域，被称为是数据目录，集群实例可以在这个文件路径下存储数据。Master 实例需要一个数据目录来存储自己的数据，每一个 Segment 实例也需要一个数据目录存储位置来存储自己的数据，对应的他们的镜像实例也需要一个存储位置。

7.2.1.1 在 Master 主机上创建数据目录

MPP 数据库的 Master 实例需要一个数据存储区域，来存储自己的系统表数据，以及其他系统元数据信息。

Master 的数据目录可以和 Segment 主机上不同，Master 实例不存储用户数据，只是

存储系统表和系统元信息，因此 Master 实例的数据存储设计的不用太大。

创建一个本地文件目录，用于 Master 实例存储数据。属主是 gpadmin。例如，以 gpadmin 用户登录，并执行如下命令：

```
$ mkdir -p /home/gpadmin/data/master
```

修改当前目录的属主为 gpadmin，例如：

```
$ chown gpadmin /home/gpadmin/data/master
```

使用 gpssh 在 Master Standby 主机上创建数据目录，例如：

```
$ source /home/gpadmin/gpdb/mpp_path.sh
$ gpssh -h smdw -e 'mkdir -p /home/gpadmin/data/master'
$ gpssh -h smdw -e 'chown gpadmin /home/gpadmin/data/master'
```

其中，smdw 为 Master Standby 主机名。

7.2.1.2 在 Segment 主机创建数据目录

MPP 数据库的 Segment 实例需要数据存储区域来存储数据，Segment Mirror 实例也需要数据目录。

创建主机列表文件 hostfile_gpssh_segonly。这个文件应该包含所有 Segment 主机的一个机器名，例如如果有两个主机服务器：

```
sdw1
sdw2
```

使用上面的配置文件，用 gpssh 来创建主 Segment 和 Segment Mirror 的数据目录。例如：

```
$ source /home/gpadmin/gpdb/mpp_path.sh
$ gpssh -f hostfile_gpssh_segonly -e 'mkdir -p /home/gpadmin/data/primary'
$ gpssh -f hostfile_gpssh_segonly -e 'mkdir -p /home/gpadmin/data/mirror'
$ gpssh -f hostfile_gpssh_segonly -e 'chown gpadmin /home/gpadmin/data/primary'
$ gpssh -f hostfile_gpssh_segonly -e 'chown gpadmin /home/gpadmin/data/mirror'
```

7.2.2 安装 license 文件

MPP 数据库的 license 文件名必须为 license.dat。license.dat 需要放在 <\$GPHOME>/bin/ 下面，每台主机都需要安装 license.dat。另外，当 gpsegininstall 工具在目标主机安装数据库时，可以将 master 主机的 license.dat 拷贝到目标主机。

7.2.3 创建初始化主机列表文件

初始化工具 `gpinitssystem` 需要一个主机列表文件，包含每个 Segment 的主机地址。工具根据主机文件中指定的每个主机列出的主机地址数，乘以配置文件 `gpinitssystem_config` 中数据目录的个数，计算出每个主机上 Segment 实例的个数。

这个文件只包含 Segment 主机地址（不包含 Master 和 Master Standby）。如果 Segment 主机有多个网卡，文件应该列出其每个网卡，一个网卡一行。

以 `gpadmin` 登录：

```
$ su - gpadmin
```

创建文件 `hostfile_gpssh_segonly`。在文件中添加 Segment 主机接口名，每个一行，不要有多余空行和空格。例如，有 4 台主机，每台 2 个网卡：

```
sdw1-1
sdw1-2
sdw2-1
sdw2-2
sdw3-1
sdw3-2
sdw4-1
sdw4-2
```

保存并关闭文件。

注意：如果不确定使用的主机名和主机接口名，可以查看文件 `/etc/hosts`。

7.2.4 创建初始化配置文件

初始化的配置文件告诉初始化工具何配置 MPP 数据库系统。配置文件的样例位于 `$GPHOME/docs/cli_help/gpconfigs/gpinitssystem_config`。

1. 以 `gpadmin` 登录

```
$ su - gpadmin
```

2. 拷贝一份 `gpinitssystem_config` 文件

```
$ cp $GPHOME/docs/cli_help/gpconfigs/gpinitssystem_config \
/home/gpadmin/gpconfigs/gpinitssystem_config
```

注意：要保证目录 `/home/gpadmin/gpconfigs` 已存在。

3. 打开并编辑文件

一个 MPP 数据库系统必须包含 Master 实例和至少 1 个 Segment 实例。

`DATA_DIRECTORY` 参数决定每个主机上可以创建多少个 Segment。如果 Segment 主机有多个网卡，并且在主机文件中列出了这些网口，Segment 会在多个网卡上均匀分布。

下面是 `gpinitssystem_config` 文件中必选参数的样例（为了可读性，将 `DATA_DIRECTORY` 写成了多行，最好是写在一行）：

```
ARRAY_NAME="Greenplum DW"
SEG_PREFIX=gpseg
PORT_BASE=40000
```

```
declare -a DATA_DIRECTORY=(/home/gpadmin/data/primary /home/gpadmin/
data/primary /home/gpadmin/data/primary /home/gpadmin/data/primary)
MASTER_HOSTNAME=mdw
MASTER_DIRECTORY=/home/gpadmin/data/master
MASTER_PORT=5432
CHECK_POINT_SEGMENTS=8
ENCODING=UNICODE
```

4. (可选) 配置 Segment Mirror

如果要部署 Segment Mirror, 打开注释并根据你的环境设置镜像参数。下面是 gpinitssystem_config 文件中可选镜像 Mirror 参数的样例(为了可读性, 将 DATA_DIRECTORY 写成了多行, 最好是写在一行):

```
MIRROR_PORT_BASE=50000
REPLICATION_PORT_BASE=41000
MIRROR_REPLICATION_PORT_BASE=51000
declare -a MIRROR_DATA_DIRECTORY=(/home/gpadmin/data/mirror /home/gp
admin/data/mirror /home/gpadmin/data/mirror /home/gpadmin/data/mirror)
```

注意: 可以先只初始化 MPP 数据库系统的主 Segment, 稍后再运行 gpaddmirrors 工具部署镜像。

5. 保存并关闭文件

7.2.5 运行初始化工具

7.2.5.1 执行初始化

gpinitssystem 工具将使用配置文件中定义的值创建 MPP 数据库系统。执行以下命令, 引用配置文件 (gpinitssystem_config) 和主机文件 (hostfile_gpssh_segonly)。例如:

```
$ cd ~
$ gpinitssystem -c gpconfigs/gpinitssystem_config -h gpconfigs/host
file_gpssh_segonly
```

如果要建立全冗余系统(带有备用 Master), 请包含 -s。例如(可以去掉, 将如下命令写在一行):

```
$ gpinitssystem -c gpconfigs/gpinitssystem_config -h gpconfigs/host
file_gpssh_segonly -s smdw
```

其中, smdw 为 Master Standby 的主机名。

注意:

凝思系统默认只存在 zh_CN.utf8 编码, 不存在 en_US.UTF8 编码。因此初始化集群的时候需要做以下两件事之一, 否则集群初始化会失败:

1. gpinitssystem 指定数据库编码为 zh_CN.utf8, 例如 gpinitssystem -c gpconfigs/gpinitssystem_config -h gpconfigs/hostfile_gpssh_segonly -n zh_CN.utf8;

2. 在/etc/locale.gen 文件中添加 en_US.UTF-8 UTF-8 编码, 然后用 sudo 权限执行 locale-gen 命令添加 en_US 编码。

工具检查配置信息，确保能连接到每个主机，并能访问其上的数据目录。所有这些预检查都完成，工具将提示你确认。例如：

```
=> Continue with Kingbase creation? Yy/Nn
```

输入 y，开始初始化。

工具开始安装并初始化 Master 实例和每个 Segment 实例。每个 Segment 实例是并行安装的。根据 Segment 数量，这可能需要一段时间。

安装完成后，工具将启动 MPP 数据库系统。你将看到：

```
=> Kingbase Database instance successfully created.
```

工具建立任何一个实例时遇到错误，整个处理过程都会失败，并留下一个部分创建了的系统。查看错误信息和日志，以确认失败的原因以及在哪个步骤失败。日志文件在 /home/gpadmin/gpAdminLogs。

若某些步骤发生失败，你有可能需要清除数据并重新运行工具。例如，一些 Segment 实例创建成功了，而另一些失败了，你需要停止那些 kingbase 进程并删除已经创建了的数据目录。如有必要，会创建一个撤销脚本辅助清理。

注意：

如果 gpinitssystem 工具执行失败，如果它留下部分创建了的系统，它会创建如下撤销脚本：

```
/home/gpadmin/gpAdminLogs/backout_gpinitssystem_<user>_<timestamp>
```

你可以使用这个脚本清理部分创建了的系统。撤销脚本会删除任何工具创建的目录，kingbase 进程，以及日志文件。解决了导致 gpinitssystem 执行失败的问题，并撤销完成后，就可以再次初始化系统了。

下面例子显示如何运行撤销脚本：

```
$ /bin/bash backout_gpinitssystem_gpadmin_20171031_121053
```

7.2.6 设置环境变量

必须在 Master（和备用 Master）上配置环境变量。在 GPHOME 路径下有一个 mpp_path.sh 文件，包含了需要设置的环境变量。你可以在 gpadmin 用户的启动脚本（如 .bashrc）中引用这个文件。

MPP 数据库管理工具要求必须设置 MASTER_DATA_DIRECTORY 变量。它指向 Master 实例的数据目录。

设置 MPP 数据库环境变量步骤如下。

1、确保以 gpadmin 登录：

```
$ su - gpadmin
```

2、编辑 profile 文件（如 .bashrc），例如：

```
$ vi ~/.bashrc
```

3、向文件添加几行，引用 mpp_path.sh 文件并设置 MASTER_DATA_DIRECTORY 环境变量。例如：

```
source /home/gpadmin/gpdb/mpp_path.sh
export MASTER_DATA_DIRECTORY=/home/gpadmin/data/master/gpseg-1
```

4、（可选）为了方便，你可能还需要设置一些关于客户端会话的环境变量，如 PGPORT, PGUSER 和 PGDATABASE 等。例如：

```
export PGPORT=5432
export PGUSER=gpadmin
export PGDATABASE=default_login_database_name
```

5、保存并关闭文件。

6、编辑完 profile 文件，引用它使其生效。例如：

```
$ source ~/.bashrc
```

7、如果你有备用 Master，将环境变量文件拷贝到备用 Master 上。例如：

```
$ cd ~
$ scp .bashrc standby_hostname:`pwd`
```

注意：这个 .bashrc 文件不应有任何输出。如果想要在用户登录时向其显示信息，请使用 .profile 文件。

8 数据库安装示例

本章举例安装数据库集群。前期硬件和软件检查部分在本章中不再举例说明。

8.1 安装单机集群

本样例中会初始化一台主机的集群，包括 Master、Master Standby、2 个 Segment 和两个 Segment Mirror。

具体步骤如下，其中 1 至 6 步需要使用 root 或者使用 sudo 权限执行，其他步骤使用 gpadmin 用户执行：

1、关闭 SELinux 和防火墙：

```
# systemctl stop firewalld
# systemctl disable firewalld
```

2、修改主机名和 hosts 文件。

修改主机名：

```
# hostnamectl set-hostname mdw
```

修改 hosts 文件，在 /etc/hosts 里添加新主机 h112 的项，例如：

```
# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.lo
caldomain4
::1 localhost localhost.localdomain localhost6 localhost6.loc
aldomain6
192.168.2.112 mdw
```

3、创建用户 gpadmin

```
# useradd -m gpadmin
# passwd gpadmin
New password: password
```

```
Retype new password: password
```

4、配置内核参数

在配置文件 `/etc/sysctl.conf` 文件里追加下面参数:

```
kernel.shmmax = 500000000
kernel.shmmni = 4096
kernel.shmall = 4000000000
kernel.sem = 250 512000 100 2048
kernel.sysrq = 1
kernel.core_uses_pid = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.msgmni = 2048
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.conf.all.arp_filter = 1
net.ipv4.ip_local_port_range = 10000 65535
net.core.netdev_max_backlog = 10000
net.core.rmem_max = 2097152
net.core.wmem_max = 2097152
vm.overcommit_memory = 2
```

执行下面命令生效:

```
# sysctl -p
```

5、配置 `limit.conf`

在 `/etc/security/limits.conf` 中设置如下参数:

```
* soft nofile 65536
* hard nofile 65536
* soft nproc 131072
* hard nproc 131072
```

6、重启机器

上述步骤执行完成后, 重启机器:

```
# reboot
```

7、获取安装包和合法的 license, 如:

```
# ls /home/gpadmin/
KingbaseAnalyticsDB-V003R002C001B0001-CENTOS6-x86_64.run  licens
e.dat
```

8、安装数据库软件

使用 `gpadmin` 用户解压安装包:

```
$ sh KingbaseAnalyticsDB-V003R002C001B0001-CENTOS6-x86_64.run
Verifying archive integrity... All good.
Uncompressing KingbaseAnalyticsDB Database
.....
```

```
.....  
.....  
.....  
.....  
Please input target installation directory:  
/home/gpadmin/gpdb  
target directory '/home/gpadmin/gpdb' does not exists, create it  
(y/N)?  
y
```

9、安装 license 文件

```
$ cp license.dat /home/gpadmin/gpdb/bin
```

10、创建数据目录

创建 Master 和 Segment 数据目录:

```
$ mkdir -p /home/gpadmin/data/master  
$ mkdir -p /home/gpadmin/data/primary  
$ mkdir -p /home/gpadmin/data/mirror
```

11、创建初始化主机列表文件

主机列表文件内容如下:

```
$ cat hostlist_singlenode  
mdw
```

12、创建初始化配置文件

```
$ cat gpinitssystem_singlenode  
ARRAY_NAME="GPDB SINGLENODE"  
SEG_PREFIX=gpseg  
PORT_BASE=40000  
declare -a DATA_DIRECTORY=(/home/gpadmin/data/primary /home/gpad  
min/data/primary)  
MASTER_HOSTNAME=mdw  
MASTER_DIRECTORY=/home/gpadmin/data/master  
MASTER_PORT=5432  
TRUSTED_SHELL=ssh  
CHECK_POINT_SEGMENTS=8  
ENCODING=UNICODE  
MIRROR_PORT_BASE=50000  
REPLICATION_PORT_BASE=41000  
MIRROR_REPLICATION_PORT_BASE=51000  
declare -a MIRROR_DATA_DIRECTORY=(/home/gpadmin/data/mirror /hom  
e/gpadmin/data/mirror)  
MACHINE_LIST_FILE=/home/gpadmin/hostlist_singlenode
```

13、设置免密登陆:

```
$ source /home/gpadmin/gpdb/mpp_path.sh  
$ gpssh-exkeys -h mdw
```

14、初始化集群

```
$ gpinitssystem -c gpinitssystem_singlenode
```

初始化成功后会输出:

```
Database successfully started
```

15、设置环境变量

在 `/home/gpadmin/.bashrc` 里添加:

```
export MASTER_DATA_DIRECTORY=/home/gpadmin/data/master/gpseg-1
source /home/gpadmin/gpdb/mpp_path.sh
```

16、添加 Master Standby

单机集群的 Master Standby 和 Master 实例在同一主机, Standby 的数据目录及端口必须和 Master 不同。

```
gpinitstandby -s mdw -P 5433 -S /home/gpadmin/data/standby
```

添加成功后, 输出里包括:

```
Successfully created standby master on mdw
```

17、安装和初始化单机集群完成

8.2 安装多主机集群

本样例多主机集群包括 mdw、smdw、sdw1、sdw2 四台主机。其中 mdw 为 Master 节点, smdw 为 Master Standby 节点, sdw1 和 sdw2 作为 Segment 节点, 每个 Segment 节点 里有两个 Segment 实例和两个 Segment Mirror 实例, sdw1 和 sdw2 为互备模式。

安装多主机集群步骤如下, 其中 1 至 6 步需要使用 root 或者使用 sudo 权限执行, 其他步骤使用 gpadmin 用户执行:

1、关闭所有主机的 SELinux 和防火墙, 例如:

```
# systemctl stop firewalld
# systemctl disable firewalld
```

2、修改主机名和 hosts 文件

修改主机名:

```
# hostnamectl set-hostname $hostname
```

其中 \$hostname 为每个主机的主机名。

修改 hosts 文件, 在 `/etc/hosts` 里添加所有主机的主机名, 例如:

```
# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost
4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost
6.localdomain6
192.168.2.113 mdw
192.168.2.114 smdw
192.168.2.115 sdw1
192.168.2.116 sdw2
```

所有主机的 `/etc/hosts` 文件必须保持相同。

3、配置内核参数，在集群内所有主机配置文件 `/etc/sysctl.conf` 文件里追加下面参数：

```
kernel.shmmax = 500000000
kernel.shmmni = 4096
kernel.shmall = 4000000000
kernel.sem = 250 512000 100 2048
kernel.sysrq = 1
kernel.core_uses_pid = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.msgmni = 2048
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.conf.all.arp_filter = 1
net.ipv4.ip_local_port_range = 10000 65535
net.core.netdev_max_backlog = 10000
net.core.rmem_max = 2097152
net.core.wmem_max = 2097152
vm.overcommit_memory = 2
```

执行下面命令生效：

```
# sysctl -p
```

4、配置 `limits.conf`，在集群内所有主机的 `/etc/security/limits.conf` 中设置如下参数：

```
* soft nfile 65536
* hard nfile 65536
* soft nproc 131072
* hard nproc 131072
```

5、Master 主机创建用户 `gpadmin`

```
# useradd gpadmin
# passwd gpadmin
New password: password
Retype new password: password
```

6、重启机器

上述步骤执行完成后，重启机器：

```
# reboot
```

机器重启完成后，下面的步骤只需要在 Master 主机(mdw) 上即可完成集群安装和初始化。

7、获取安装包和 license

获取安装包和合法的 license，如：

```
# ls /home/gpadmin/
KingbaseAnalyticsDB-V003R002C001B0001-CENTOS6-x86_64.run license.dat
```

8、安装数据库软件

在 Master 主机使用 gpadmin 用户解压安装包:

```
$ sh KingbaseAnalyticsDB-V003R002C001B0001-CENTOS6-x86_64.run
Verifying archive integrity... All good.
Uncompressing KingbaseAnalyticsDB Database
.....
.....
.....
.....
.....
.....
.....
.....
Please input target installation directory:
/home/gpadmin/gpdb
target directory '/home/gpadmin/gpdb' does not exists, create it
(y/N)?
y
```

9、安装 license 文件

```
$ cp license.dat /home/gpadmin/gpdb/bin
```

10、免密登陆

在 Master 主机交换公钥:

```
# source /home/gpadmin/gpdb/mpp_path.sh
# gpssh-exkeys -h mdw -h smdw -h sdw1 -h sdw2
```

11、安装到所有主机

先生成安装主机列表文件, 包括 smdw、sdw1、sdw2

```
# cat /home/gpadmin/hostlist_seginstall
smdw
sdw1
sdw2
```

执行 gpseinstall 命令, 安装集群软件到所有主机:

```
# gpseinstall -f /home/gpadmin/hostlist_seginstall -u gpadmin -p
changeme
```

12、创建数据目录

在 Master 主机, 创建 Master 和 Segment 数据目录:

```
$ gpssh -h mdw -h smdw -h sdw1 -h sdw2
=> mkdir -p /home/gpadmin/data/master
=> mkdir -p /home/gpadmin/data/primary
=> mkdir -p /home/gpadmin/data/mirror
=> exit
```

13、创建初始化主机列表文件

在 Master 主机, 创建初始化主机列表文件, 内容如下:

```
$ cat hostlist_gpinitssystem
sdw1
sdw2
```

14、创建初始化配置文件

```
$ cat gpinitssystem_config
ARRAY_NAME="GPDB DW"
SEG_PREFIX=gpseg
PORT_BASE=40000
declare -a DATA_DIRECTORY=(/home/gpadmin/data/primary /home/gpadmin/data/primary)
MASTER_HOSTNAME=mdw
MASTER_DIRECTORY=/home/gpadmin/data/master
MASTER_PORT=5432
TRUSTED_SHELL=ssh
CHECK_POINT_SEGMENTS=8
ENCODING=UNICODE
MIRROR_PORT_BASE=50000
REPLICATION_PORT_BASE=41000
MIRROR_REPLICATION_PORT_BASE=51000
declare -a MIRROR_DATA_DIRECTORY=(/home/gpadmin/data/mirror /home/gpadmin/data/mirror)
MACHINE_LIST_FILE=/home/gpadmin/hostlist_gpinitssystem
```

15、初始化集群

```
$ gpinitssystem -c gpinitssystem_config
```

初始化成功后会输出:

```
Database successfully started
```

16、设置环境变量

在 /home/gpadmin/.bashrc 里添加:

```
export MASTER_DATA_DIRECTORY=/home/gpadmin/data/master/gpseg-1
source /home/gpadmin/gpdb/mpp_path.sh
```

17、添加 Master Standby

```
gpinitstandby -s smdw
```

添加成功后, 输出里包括:

```
Successfully created standby master on smdw
```

18、安装和初始化多机集群完成