MPP 数据库安装指南

北京人大金仓信息技术股份有限公司

目录

1 前言

1.1概述

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，包括操作的步骤和运行的命令等。

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：

- 安装前准备 – 安装数据库的前期准备，如软件、硬件环境检查
- 安装数据库 – 安装初始化数据库
- 安装后推荐任务 – 安装完成后推荐执行的任务
- 开始使用数据库 – 数据库的基础操作
- 卸载数据库 – 如何卸载数据库
- 附录 – 附录

1.2文档规范

MPP 数据库的文档使用如下规范。

用法	说明	举例
{ }	在命令语法中，花括号指定命令选项分组。不要输入花括号本身。	FROM { ‘filename’ | STDIN }
[ ]	在命令语法中，方括号表示可选参数,不要输入花括号本身。	TRUNCATE [TABLE] name
…	在命令语法中，省略号表示命令、变量或选项的重复，不要输入省略号本身。	DROP TABLE name [,…]
$ system_command	$表示命令提示符,不要输入提示符本身	$ createdb mydatabase
# root_system_command	#表示终端命令提示符	# chown gpadmin -R /datadir
=> gpdb_command	=>表示 SQL 命令提示符	=> SELECT * FROM mytable;
=# su_gpdb_command	=#为 MPP 数据库交互程序提示符 (例如 psql 或 gpssh)。	=# SELECT * FROM pg_database;

2 安装前准备

本章描述了安装 MPP 数据库之前准备工作。包括：

- 硬件环境检查
- 软件环境检查
- 操作系统配置

2.1 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

2.1.1 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
```

或使用 free 命令查看内存和 swap 分区大小。

2.1.2 CPU 架构

CPU 架构只支持 x86_64。
查看 CPU 架构命令：

```
# uname -m
```

2.1.3 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
```

2.2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

2.2.1操作系统要求

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
- Linx 6.0 或者更高
- iSoft Server OS 4.2 或者更高
- Kylin 3.2 或者更高

CentOS 和普华系统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/linx-release
```

Kylin 系统使用以下命令查看当前操作系统版本：

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

操作系统需要满足：

- 所有主机采用同一种操作系统
- 各主机 root 用户的 密码必须一致

2.2.2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
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
Linx 6.0	4.9.0-0.bpo.1-linx-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
```

2.2.3 依赖包

MPP 数据库需要下面的依赖包，

bash
json-c
openssh
openssh-clients
perl
sed
sysstat
tar
vim-minimal
zip
xfsprogs
zlib

以上依赖包均整合在 MPP 数据库安装包中，不需要额外安装。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
```

2.2.4 文件系统推荐

推荐使用 XFS 文件系统。XFS 主要特性包括以下几点：

数据完全性：采用 XFS 文件系统，宕机发生后，首先，由于文件系统开启了日志功能，所以磁盘上的文件不会因意外宕机而损坏。不论目前文件系统上存储的文件与数据有多少，文件系统都可以根据所记录的日志在很短的时间内迅速恢复磁盘文件内容。

传输特性：XFS 文件系统采用优化算法，日志记录对整体文件操作影响非常小。XFS 查询与分配存储空间非常快。XFS 文件系统能连续提供快速的反应时间。

可扩展性：XFS 是一个全 64-bit 的文件系统，它可以支持上百万 T 字节的存储空间。对特大文件及小尺寸文件的支持都表现出众，支持特大数量的目录。 最大可支持的文件大小为 263 = 9 x 1018 = 9 exabytes，最大文件系统尺寸为 18 exabytes。 XFS 使用高的表结构(B+ 树)，保证了文件系统可以快速搜索与快速空间分配。XFS 能够持续提供高速操作，文件系统的性能不受目录中目录及文件数量的限制。

传输带宽：XFS 能以接近裸设备 I/O 的性能存储数据。在单个文件系统的测试中，其吞吐量最高可达 7GB 每秒，对单个文件的读写操作，其吞吐量可达 4GB 每秒。

通过下面命令制作 XFS 文件系统。使用 root 用户：

```
mkfs.xfs devname
```

例如：

```
# mkfs.xfs /dev/sda3 或 # mkfs -t xfs /dev/sda3
```

2.3操作系统配置

2.3.1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 RHEL6.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相关文档。

凝思系统默认没有安装 SELinux 和配置防火墙。

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 相关文档。

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
```

2.3.2 检查网络

生产环境里，MPP 数据库集群里每台主机的网卡至少需要为千兆网卡。
使用下面命令查看网卡速度：

```
# ethtool devname
```

其中 dvename 为网卡名。例如：

```
# ethtool p5p1
    Settings for p5p1:
      Supported ports: [ FIBRE ]
      Supported link modes:   1000baseT/Full
                                  10000baseT/Full
      Supported pause frame use: Symmetric Receive-only
      Supports auto-negotiation: No
      Advertised link modes:  10000baseT/Full
      Advertised pause frame use: No
      Advertised auto-negotiation: No
      Speed: 10000Mb/s
      Duplex: Full
      Port: FIBRE
      PHYAD: 1
      Transceiver: internal
      Auto-negotiation: off
      Supports Wake-on: g
      Wake-on: d
      Current message level: 0x00000000 (0)
```

上面例子中，网卡 p5p1 的网速为 10000Mb/s。

确保集群里所有主机网络均可达，使用下面命令测试网络：

```
# ping host
```

其中 host 为要测试主机 ip 或主机名。

2.3.3 修改主机名和 hosts 文件

2.3.3.1 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。设置完成后重新登录生效。例如：

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

CentOS 6x 、Red Hat Enterprise Linux 6x 和 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
```

2.3.3.2 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 sdw1
192.168.2.115 sdw2
192.168.2.116 sdw3
```

2.3.4 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
EOF
```

(可选) 如果你在同一台机器上用多个管理帐号去运行多个 MPP 数据库实例，可能需要生成一个用户组做为 MPP 数据库安装的属主。本文档使用 gpadmin 作为组名。要添加一个新组，可以以 root 身份运行如下命令：

```
# groupadd gpadmin
# usermod -g gpadmin ka_user1
# usermod -g gpadmin ka_user2
```

2.3.5 配置 ssh

MPP 数据库的管理工具如 gpinitsystem、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
```

sshd 监听的默认端口为 22，若端口号改为非默认值，需要在 source 文件/home/gpadmin/gpdb/mpp_path.sh 里修改下面内容：

```
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。

2.3.6 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=16
m 0 0
```

2.3.7 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。

2.3.7.1 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
```

2.3.7.2 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= 10000 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 gpinitsystem 相关手册。

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

对 Red Hat 6.x 、Centos 6.x 和 Kylin 3.2 ，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 的相关文档。

2.3.9配置删除临时文件

对 Red Hat 7.x 、 Centos 7.x 和普华系统，系统重启后，默认情况下需要不会自动删除/tmp 下的临时文件。若操作系统异常正常关闭，可能会造成数据库生成的临时文件无法自动删除，使数据库启动失败。通过下面方式，配置系统重启后可自动删除/tmp 里临时文件，向/usr/lib/tmpfiles.d/tmp.conf 文件里追加下面内容：

```
r! /tmp/.s.PGSQL.*
r! /tmp/.gpdb_master_ha.pid
r! /tmp/.gpstop.*.lock
```

2.3.10修改 IO 调度策略

系统的 IO 调度策略可以是：CFQ，AS 和 deadline。

对于 MPP 数据库来说，需要设置成 deadline 调度策略。需要执行如下命令设置磁盘的调度策略，并重启系统。

```
# echo schedulername > /sys/block/devname/queue/scheduler
```

例如：

```
# echo deadline > /sys/block/sbd/queue/scheduler
```

在 RHEL 7.x 、 CentOS7.x 和普华系统 的 grub2 中，启动时指定 IO 调度策略需要使用 grubby。需要以 root 用户运行如下 命令：

```
# grubby --update-kernel=ALL --args="elevator=deadline"
```

添加参数后，需要重启系统。

关于 grubby 的更多信息，请参见操作系统手册。

其他系统可以在系统启动的时候设置调度算法，添加参数设置 elevator=deadline 到内核启动的 GRUB 配置文件中： /boot/grub/grub.conf。下面是一个 RHEL 6.x 或者 CentOS 6.x 的内核命令的 grub.conf 配置文件。设置成多行显示是为了可读性，实际上在 grub.conf 中是写在一行上的。

```
kernel /vmlinuz-2.6.18-274.3.1.el5 ro root=LABEL=/
     elevator=deadline crashkernel=128M@16M quiet console=tty1
     console=ttyS1,115200 panic=30 transparent_hugepage=never
```

```
initrd /initrd-2.6.18-274.3.1.el5.img
```

2.3.11修改文件预读大小

每一个磁盘设备都一个 read-ahead（预读）（blockdev）设置值：16384
获取目前的预读值：

```
# /sbin/blockdev --getra devname
```

例如：

```
# /sbin/blockdev --getra /dev/sdb
```

设置目前的预读值：

```
# /sbin/blockdev --setra bytes devname
```

例如：

```
# /sbin/blockdev --setra 16384 /dev/sdb
```

使用 man blockdev 获取更多信息。

2.3.12禁用 RemoveIPC

对 RHEL7.2 、 CenOS7.x 和普华系统，需要禁用 RemoveIPC。默认的 systemd 设置是 RemoveIPC=yes，意味着当非系统用户登录退出时，删除 IPC 连接。这个会引起 MPP 数据库的 gpinitsystem 出错。试着使用如下方式来避免这个问题。

禁用 RemoveIPC 特性。在所有的 MPP 数据库主机上，修改 /etc/systemd/logind.conf 文件， 添加如下参数：

```
RemoveIPC=no
```

当重启 systemd-login 服务，或者重启系统后，修改生效。使用如下命令重启这个服务，需要以 root 用户执行：

```
service systemd-logind restart
```

或者在创建用户时同时将其设置为系统账号。即在使用 useradd 命令创建用户时，需要加上 -r 选项（创建的用户是系统用户）和 -m 选项（创建用户的主目录）

2.4重启服务器

为确保上述参数配置生效，重启各台服务器。

2.5获取安装包和 license

请联系技术支持获取安装包和合法的 license。

3 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
- 同步系统时钟
- 检查系统性能
- 初始化集群
- 单机集群安装

3.1 安装数据库软件

本章介绍如安装 MPP 数据库系统软件。安装数据库软件包括以下主题：

- 安装内容简介
- 在 Master 主机上安装集群软件
- 安装到所有主机

3.1.1 安装内容简介

MPP 数据库集群软件安装成功后，包括下面内容：

- mpp_path.sh - 这个文件包括了 MPP 数据库的环境变量
- bin - 包含了 MPP 数据库的管理工具，也包含 MPP 数据库的客户端和服务器程序
  注意： license 文件需要放到 bin 目录下，并改名为 license.dat 。 licens e.dat 文件需要放在 bin 目录下，以保证系统的正常使用。请联系人大金仓技术支持，获取合法的 license 文件。
- demo - 包含了 MPP 数据库的的实例程序
- docs - 包含了工具的帮助文档
- include - 头文件
- lib - 库文件
- sbin - 支持和内部运行的脚本程序
- share - 共享目录

3.1.2 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，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：

```
     $ /bin/bash KingbaseAnalyticsDB-V003R002C001B0001-CENTOS6-x86_6
4.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
```

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 里修改下面内容：

```
   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。

3.1.3 安装到所有主机

运行 gpseginstall，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。gpseginstall 支持以 gpadmin 用户或 root 用户执行。

使用 gpseginstall 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。

以 gpadmin 用户安装到所有主机的步骤如下：

执行 gpssh-exkeys 交换所有提供的主机的密钥

```
   # source /home/gpadmin/gpdb/mpp_path.sh
   # gpssh-exkeys -f hostfile_gpssh_allhosts
```

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，hostfile_gpssh_allhosts 文件里包含了所有交换公钥的主机列表。

gpseginstall 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_seginstall
      sdw1
      sdw2
      sdw3
```

执行 gpseginstall 安装软件：

```
   # gpseginstall
```

也可以选择以 root 用户执行 gpseginstall，当使用 root 用户运行时该工具还会在目标主机上自动进行以下操作：

- 创建操作系统用户，默认为 gpadmin，可通过-u 参数指定
- 设置其密码，默认为 changeme，可以通过-p 参数设置
- 设置安装目录的属主
- 在所有提供的主机间交换 SSH 公钥

注意，并不是所有系统的 root 用户可以成功执行 gpseginstall 完成安装功能，如海康 os。

以 root 用户执行 gpseginstall 安装软件：

```
# source /home/gpadmin/gpdb/mpp_path.sh
# gpseginstall -f hostlist_seginstall -u gpadmin -p changeme
```

其中 -u 指定操作系统用户，此选项仅在以 root 身份运行 gpseginstall 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

3.1.4 确认安装

安装完成后，需要按照如下步骤，来确认 MPP 数据库是否正确安装。

1、以 gpadmin 登录 Master 主机：

```
$ su - gpadmin
```

2、执行如下命令，设置登录用户的 MPP 数据库环境变量

```
$ source /home/gpadmin/gpdb/mpp_path.sh
```

3、使用 gpssh 工具确认登录 MPP 数据库的所有主机是否需要密码输入，并确认 MPP 数据库是否在所有主机都安装了。

```
$ gpssh -f hostfile_gpssh_allhosts -e ls -l $GPHOME
```

hostfile_gpssh_allhosts 文件包括了集群的所有主机，如：

```
$ cat hostfile_gpssh_allhosts
    mdw
    sdw1
    sdw2
    sdw3
```

如果成功安装了 MPP 数据库，那么在登录集群中任何机器时都不需要输入密码。在所有机器上都应该能够看到相同的安装目录并且都是 gpadmin 是这些安装目录的属主。

如果提示了密码，需要运行如下命令重新进行 SSH 公钥的安装。

```
$ 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。

3.2 同步系统时钟

需要使用 NTP（网络时间协议）来同步 MPP 数据库中所有主机的时钟，参见 www.ntp.org 获取更多关于 NTP 的信息。

所有 Segment 主机的系统时间应该以 Master 主机作为时间源，并且以 Master St

andby 作为第二个时间源。在 Master 和 Master Standby 上，将他们的时间源指向一个新的时间服务器。

在 Master 主机，使用 root 登录，并修改配置文件 /etc/ntp.conf。设置 server 参数为数据中心的时间服务器，例如：（如果 10.6.220.20 是数据中心的 NTP 服务器）

server 10.6.220.20

在每个 Segment 主机上，以 root 用户登录系统，并编辑 /etc/ntp.conf 文件。设置首选 server 参数的值为 Master 主机，候选 server 的参数值为 Master Standby 主机。例如：

```
server mdw prefer
server smdw
```

其中 mdw 为 Master 主机名，smdw 为 Standby 主机名。

在 Master Standby 主机上，以 root 用户登录系统，并编辑 /etc/ntp.conf 文件，设置首选 server 的参数指向 Mater 主机，并且候选 server 指向数据中心的 NTP 时间服务器，如下：

```
server mdw prefer
server 10.6.220.20
```

在 Master 主机上，使用 NTP 命令同步所有主机的时钟，例如：

```
# gpssh -f hostfile_gpssh_allhosts -v -e 'ntpd'
```

hostfile_gpssh_allhosts 里包含集群里的所有主机。

3.3检查系统性能

MPP 数据库提供工具 gpcheckperf 可以检测操作系统性能。该工具可以在 $GPHOME/bin 下找到。在初始化 MPP 数据库之前，应该先检测系统性能。包括：

- 网络性能 (gpnetbench*)
- 磁盘 IO (dd)
- 内存带宽 (stream)

执行 gpcheckperf 之前需要所有主机可免密登录可以使用工具 gpssh-exkeys 来交换公钥。gpcheckperf 工具会调用 gpssh 和 gpscp ，所以上述工具都要在环境变量 $PATH 内。

3.3.1检测网络性能

为了检测网络性能，执行 gpcheckperf 可有以下选项：并行 (-r N)，串行 (-r n) 和矩阵 (-r M)。gpcheckperf 在不同主机间 传输 5 秒的数据流。默认情况，以并行方式测试不同主机间的网络性能，测试完成后，输出网络最大、最小、和平均值，单位是 MB/s。

MPP 数据库大多数主机都配有多个网卡。测试网络性能时要包含每个网卡。例如，每个主机配有两个网卡：

主机网络配置示例

主机	网卡 1	网卡 2
Segment 1	sdw1-1	sdw1-2
Segment 2	sdw2-1	sdw2-2
Segment 3	sdw3-1	sdw3-2

检测主机文件内容

hostfile_gpchecknet_ic1	hostfile_gpchecknet_ic2
sdw1-1	sdw1-2
sdw2-1	sdw2-2
sdw3-1	sdw3-2

每个主机文件执行一次 gpcheckperf 。例如：

```
$ gpcheckperf -f hostfile_gpchecknet_ic1 -r N -d /tmp > subnet1.out
$ gpcheckperf -f hostfile_gpchecknet_ic2 -r N -d /tmp > subnet2.out
```

3.3.2检测硬盘 I/O 和内存带宽

gpcheckperf 使用 (-r ds) 选项检测硬盘和内存性能。检测硬盘使用的是 dd 命令。检测内存使用的是 stream 工具。检测结果单位为 MB/s。

1、先以 gpadmin 用户登录 Master 主机。

2、source mpp_path.sh，例如：

```
$ source /home/gpadmin/gpdb/mpp_path.sh
```

3、创建主机文件 hostfile_gpssh_segonly，每个 Segment 主机占一行，不要包括 Master 主机。例如：

```
sdw1
sdw2
sdw3
```

4、执行 gpcheckperf 命令。-d 选项指定文件系统某个目录(必须有写权限)。例如：

```
$ gpcheckperf -f hostfile_gpssh_segonly -r ds -d /home/gpadmin
```

测试可能会执行一段时间，因为可能会在主机间拷贝很大的文件。测试完成后，可以看到硬盘写、硬盘读、和内存带宽的相应数据。

3.4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 gpinitsystem 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；

2、安装 license 文件；

3、创建主机列表文件，包含所有 Segment 主机信息；

4、创建初始化数据库配置文件；

5、在 Master 主机上运行 MPP 数据库初始化工具；

6、初始化完成后设置相应的环境变量。

3.4.1 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 实例需要一个数据存储来存储自己的数据，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。

3.4.1.1 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 root 用户登录（此例需要在/根目录下创建文件夹，所以使用了 root 用户），并执行如下命令：

```
# mkdir -p /data/Master
```

修改当前目录的属主为 gpadmin，例如：

```
# chown 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 /data/Master'
# gpssh -h smdw -e 'chown gpadmin /data/Master'
```

其中 smdw 为 Master Standby 主机名。

3.4.1.2 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 实例也则需要数据目录。例如：

以 root 用户登录 Master 主机（此例同上节例子在/根目录下创建数据目录，所以使用了 root 用户）：

```
# su
```

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，例如如果有三个主机服务器：

```
sdw1
sdw2
sdw3
```

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 /data/primary'
# gpssh -f hostfile_gpssh_segonly -e 'mkdir -p /data/mirror'
# gpssh -f hostfile_gpssh_segonly -e 'chown gpadmin /data/primary'
# gpssh -f hostfile_gpssh_segonly -e 'chown gpadmin /data/mirror'
```

3.4.2 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。每台主机必须有 license.dat。

3.4.3 创建初始化主机列表文件

初始化工具 gpinitsystem 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 gpinitsystem_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_gpinitsystem 。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 。

3.4.4 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 /home/gpadmin/gpdb/docs/cli_help/gpconfigs/gpinitsystem_config 。

1、以 gpadmin 登录。

$ su - gpadmin

2、拷贝一份 gpinitsystem_config 文件

```
$ cp /home/gpadmin/gpdb/docs/cli_help/gpconfigs/gpinitsystem_config   /home/gpadmin/gpconfigs/gpinitsystem_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 gpinitsystem_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="MPP DW"
SEG_PREFIX=gpseg
PORT_BASE=40000
declare -a DATA_DIRECTORY=(/data/primary /data/primary
        /data/primary /data/primary /data/primary /data/primary)
MASTER_HOSTNAME=mdw
MASTER_DIRECTORY=/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 gpinitsystem_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=(/data/mirror /data/mirror /data/mirror /data/mirror /data/mirror /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。

3.4.5运行初始化工具

3.4.5.1 执行初始化

gpinitsystem 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（ gpinitsystem_config ）和主机文件（hostfile_gpssh_segonly ）。例如：

```
$ cd ~
$ gpinitsystem -c gpconfigs/gpinitsystem_config -h gpconfigs/hostfile_gpssh_segonly
```

如果要建立全冗余系统（带有备用 Master），请包含 -s 选项。例如：

```
$ gpinitsystem -c gpconfigs/gpinitsystem_config -h gpconfigs/hostfile_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. gpinitsystem 指定数据库编码为 zh_CN.utf8，例如：

```
gpinitsystem -c gpconfigs/gpinitsystem_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 gpinitsystem 会检查配置信息，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，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，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。

3.4.5.2 使用撤销脚本

如果 gpinitsystem 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_gpinitsystem_<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 gpinitsystem 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：

```
$ sh backout_gpinitsystem_gpadmin_20171031_121053
```

3.4.6 设置环境变量

初始化完成后，必须在 Master（和 Master Standby）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=/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 Master Standby，将环境变量文件拷贝到 Master Standby 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。

3.5 单机集群安装

单机安装集群是多机集群安装的一个特例，步骤和集群安装类似。

3.5.1 单机安装步骤

1、安装前期准备，包括硬件、软件环境是否满足数据库安装和初始化；

2、安装集群软件，初始化数据库；

3、数据库初始化完成后，执行推荐任务。

3.5.2 与多主机集群不同点

1、单机集群安装不再需要把软件安装到其它主机，省略 gpseginstall 步骤；
2、单机集群安装的初始化主机列表文件里只包括一行，也即本机的主机名或 IP；
3、单击安装若添加 Master Standby，Standby 的端口和数据目录不能和 Master 相同。

4 安装后推荐任务

本章介绍数据库安装完成后的推荐任务。包括：

- 设置数据库日志级别
- 允许客户端连接
- locale 本地化设置

4.1 设置数据库日志级别

MPP 数据库日志可记录数据库内部多种操作，包括数据库的启动和停止，执行出错的 SQL 语句，数据库的连接和退出等。日志里也包括所有执行的 SQL 语句，根据不同的日志级别，日志内容也会相应增加减或减少。

Master 和 Segment 实例维护各自的日志文件。数据库的日志可从 <data_dir>/pg_log/gpdb-yyyy-mm-dd_hhmmss.csv 里查找。

参数 log_statement 控制哪些 SQL 语句被记录。有效值是 none (off)、ddl、mod 和 all（所有语句）。ddl 记录所有数据定义语句，例如 CREATE、ALTER 和 DROP 语句。mod 记录所有 ddl 语句，外加数据修改语句例如 INSERT, UPDATE、DELETE、TRUNCATE, 和 COPY FROM。 如果 PREPARE、EXECUTE 和 EXPLAIN ANALYZE 包含合适类型的命令，它们也会被记录。对于使用扩展查询协议的客户端，当收到一个执行消息时会产生日志并且会包括绑定参数的值（任何内嵌的单引号会被双写）。

log_statement 默认值为 all，应用开发阶段，默认值可增加日志量，方便应用查找错误。若线上业务量较大，log_statement = all 会使日志量较大，占用过多的存储资源。因此，若业务量大的应用系统里，推荐参数值为 ddl 或 mod。

使用下面命令设置参数：

```
# gpconfig -c log_statement -v value
```

其中 value 为对应参数值，例如设为 ddl：

```
# gpconfig -c log_statement -v ddl
```

使用下面命令生效：

```
# gpstop -u
```

查看参数值方法如下：

```
# gpconfig -s log_statement
```

4.2 允许客户端连接

当 MPP 数据库第一次初始化后，它仅允许 gpadmin （或其他运行 gpinitsystem 的系统用户）从本地访问。如果你想要其他用户或从其他机器上访问 MPP 数据库，必须给它们访问权限。详细内容请参见章节设置用户认证 。

4.3locale 本地化设置

本章描述 MPP 数据库的可用的本地化特性。MPP 数据库支持：

- 使用操作系统的 locale 特性指定排序规则、数字格式等
- 服务器支持不同字符集，有多字节字符集来支持存储不同语言文本，提供客户端和服务器的字符转换

4.3.1关于 locale 的支持

locale 设置会影响字符、排序和数字格式等。MPP 数据库使用标准 ISO C 和 POSIX。数据初始化时会指定 locale。初始化工具 gpinitsystem 会默认使用操作系统的 locale。若不想用默认的 locale，可以在初始化时使用 -n 选项指定，例如：

```
$ gpinitsystem -c gpconfigs/gpinitsystem_config -n sv_SE
```

上述例子中把 locale 设为 sv_SE。当然也有其他的选项如 en_US 和 fr_CA 等。大多数系统中，可使用命令 locale -a 列出所有的 locale。

偶尔也会有多个混合规则设置 locale，例如排序规则使用 English，而消息为 Spanish。为了支持上述情形，locale 又分多个子类：

- LC_COLLATE - 比较和排序规则
- LC_CTYPE - 语言符合及其分类
- LC_MESSAGES - 信息
- LC_MONETARY - 货币单位
- LC_NUMERIC - 数字
- LC_TIME - 时间格式

如果操作系统的 locale 不是你想要的，可以指定特殊的 locale 为 C 或 POSIX。

若上述 locale 的值已经确定，数据库初始化完成后就不能再更改。LC_COLLATE 和 LC_CTYPE 会影响排序规则。若初始化数据库时未指定 locale，则默认使用操作系统 locale。

需要注意的是默认 locale 取决于服务器而不是客户端的 locale。因此启动数据库之前，需要小心配置好 MPP 数据库每个主机的 locale。

4.3.2locale 的行为

locale 的设置会影响下列的 SQL 特性：

- ORDER BY 子句的排序规则
- LIKE 子句使用索引
- upper，lower 和 initcap 函数
- to_char 函数族

使用 locale 而不是 C 或 POSIX 的缺点是影响性能。会减低字符的处理性能并且影响 LIKE 子句使用索引。因此除非真正需要才会用 locale。

4.3.3故障排除

如果 locale 工作和预期不同，先检查操作系统是否配置正确。可使用 locale -a 来检测系统安装的 locale。

查看 MPP 数据库数据库是否使用 locale。 LC_COLLATE 和 LC_CTYPE 初始化完成后就不能再修改。LC_MESSAGE 和 LC_MONETARY 为 MASTER 主机和 Segment 主机环境变量决定，也可以修改不同实例的 postgresql.conf 文件修改这两个参数。可使用 SHOW 命令检查 Master 主机的 locale 设置。注意，所有的 Master 和 Segment 实例应该使用相同的 locale。

4.3.4 字符集支持

多字符集支持允许 MPP 数据库存储不同字符集的文本，如单字节字符 ISO 8859，多字节字符 EUC，UTF-8 等。所有支持的字符级可透明的在客户端使用，但有一些服务器端不支持。默认的字符集是在初始化数据库时指定的。在创建新数据库时可以重新指定字符集，因此可以有多个数据库，每个数据库的字符集各不相同。

4.3.5 设置字符集

gpinitsystem 初始化数据库时根据初始化配置文件中的 ENCODING 参数设置数据库默认字符集。默认字符集是 UNICODE 或 UTF-8。

也可以不使用默认字符集新创建一个数据库，例如：

```
   => CREATE DATABASE korean WITH ENCODING 'EUC_KR';
```

尽管可以在创建数据库指定字符集。选择一个和 locale 不同的字符集是不明智的。

4.3.6 服务器客户端字符集转换

MPP 数据库支持客户端和服务器之间的自动字符集转换。转换信息存在 Master 实例 pg_conversion 的系统表。也可以使用 CREATE CONVERSION 创建新转换规则。

客户端/服务器字符集转换

服务器字符集	可用的客户端字符集（可以相互转换）
BIG5	不支持作为服务器编码
EUC_CN	EUC_CN，MULE_INTERNAL，UTF8
EUC_JP	EUC_JP，MULE_INTERNAL，SJIS，UTF8
EUC_KR	EUC_KR，MULE_INTERNAL，UTF8
EUC_TW	EUC_TW，BIG5，MULE_INTERNAL，UTF8
GB18030	不支持作为服务器编码
GBK	不支持作为服务器编码
ISO_8859_5	ISO_8859_5，KOI8，MULE_INTERNAL，UTF8，WIN866，WIN1251
ISO_8859_6	ISO_8859_6，UTF8

ISO_8859_7	ISO_8859_7，UTF8
ISO_8859_8	ISO_8859_8，UTF8
JOHAB	JOHAB，UTF8
KOI8	KOI8，ISO_8859_5，MULE_INTERNAL，UTF8，WIN866，WIN1251
LATIN1	LATIN1，MULE_INTERNAL，UTF8
LATIN2	LATIN2，MULE_INTERNAL，UTF8，WIN1250
LATIN3	LATIN3，MULE_INTERNAL，UTF8
LATIN4	LATIN4，MULE_INTERNAL，UTF8
LATIN5	LATIN5，UTF8
LATIN6	LATIN6，UTF8
LATIN7	LATIN7，UTF8
LATIN8	LATIN8，UTF8
LATIN9	LATIN9，UTF8
LATIN10	LATIN10，UTF8
MULE_INTERNAL	MULE_INTERNAL，BIG5，EUC_CN，EUC_JP，EUC_KR，EUC_TW，ISO_8859_5，KOI8，LATIN1 to LATIN4，SJIS，WIN866，WIN1250，WIN1251
SJIS	不支持作为服务器编码
SQL_ASCII	不支持作为服务器编码
UHC	不支持作为服务器编码
UTF8	所有可以支持的编码
WIN866	WIN866
ISO_8859_5	KOI8，MULE_INTERNAL，UTF8，WIN1251
WIN874	WIN874，UTF8
WIN1250	WIN1250，LATIN2，MULE_INTERNAL，UTF8
WIN1251	WIN1251，ISO_8859_5，KOI8，MULE_INTERNAL，UTF8，WIN866
WIN1252	WIN1252，UTF8
WIN1253	WIN1253，UTF8
WIN1254	WIN1254，UTF8
WIN1255	WIN1255，UTF8
WIN1256	WIN1256，UTF8
WIN1257	WIN1257，UTF8
WIN1258	WIN1258，UTF8

使用自动数据字符集转换，可以告诉服务器客户端字符集，有多种方式指定：

在 psql 里使用 encoding 命令，允许在线修改客户端编码。使用 SET client_encoding TO 命令：

```
=> SET CLIENT_ENCODING TO 'value';
```

查询当前客户端编码：

```
=> SHOW client_encoding;
```

重置客户端编码：

```
=> RESET client_encoding;
```

使用环境变量 KICLIENTENCODING。客户端连接数据库时会自动选择环境变量指定的编码。上述方法可以覆盖使用环境变量指定的客户端编码。

配置文件中指定 client_encoding。若在 Master 实例的 postgresql.conf 文件指定 client_encoding，新的连接客户端编码则为 client_encoding 指定。

5 开始使用数据库

这一章提供了一部分 MPP 数据库基础操作的简介。具体使用方法，请参考《MPP 数据库管理员指南》和《MPP 数据库参考指南》

5.1 启动和停止集群

MPP 数据库安装目录包含了集群停止、启动并查看当前状态的命令工具。 在 MPP 数据库安装目录下的 bin 文件夹中有这些命令，也就是，/home/gpadmin/gpdb/bin 文件夹。当你的环境变量已经 source 过 /home/gpadmin/gpdb/mpp_path.sh， 便可以在你的当前路径下使用命令。

使用运行 MPP 数据库的用户执行这些命令， 例如 gpadmin。 可以使用 --help 选项获取在线帮助。

5.1.1 检查集群实例状态

运行 gpstate 命令：

```
$ gpstate
```

命令输出 Master 和 Segment 实例状态。 如果 MPP 数据库系统没有运行， 命令会输出错误信息。

5.1.2 停止集群

运行 gpstop 命令：

```
$ gpstop
```

输出将要被停止的 Master 以及 Segment 进程参数。

当提示停止 MPP 数据库实例时，输入 y 。

5.1.3 启动集群

运行 gpstart 命令：

```
$ gpstart
```

输出将要被停止的 Master 和 Segment 进程参数。

当提示启动 MPP 数据库实例时，输入 y 。

5.1.4 更多信息

如需要学习如何管理 MPP 数据库。 请查阅 《MPP 数据库工具参考手册》来获取更多信息。

5.2 实例，数据库，模板数据库

MPP 数据库集群被叫做实例或集群。一个集群中可能有多个实例。

一个实例可以管理多个数据库。一个数据库包括应用所需的所有数据或对象。一个客户端同时只能连接一个数据库，一个查询不可操作多个数据库。

新初始化的集群默认有三个数据库：

- template1 数据库是创建新数据库的模板数据库。如果想要一个对象在所有数据库中都存在，可以把该对象创建在 template1 数据库中。
- postgres 和 template0 数据库供内部使用，不应更改或删除。如果更改了 template1 数据库，可以使用 template0 数据库当做模板，这样新数据库中就不会有 template1 中已更改的对象。

5.3 使用 psql 连接数据库

psql 是连接数据库的交互性命令行工具。默认情况下， psql 尝试连接本机（localhost）上端口为 5432 的 postgres 数据库，默认连接的用户名和数据库是当前操作系统登录用户。例如，如果当前用户是 gpadmin，psql 默认用 gpadmin 用户连接 gpadmin 数据库。默认是没有 gpadmin 数据库的，所以使用 psql 时必须指定要连接的数据库。

默认数据库名和其他连接操作可以在命令行指定，或使用环境变量指定。必要的连接参数是主机地址，Master 端口号，用户名和数据库名。

5.3.1 使用默认连接参数连接数据库

```
$ psql postgres
```

5.3.2 指定连接参数

```
$ psql -h localhost -p5432 -Ugpadmin postgres
```

5.3.3 使用环境变量设置连接参数

```
$ export PGPORT=5432
$ export PGHOST=localhost
$ export PGDATABASE=postgres
$ psql
```

5.3.4 在 psql 中输入命令

当成功连接到数据库中，psql 显示提示符， 数据库名后加 =#。
例如：

```
$ psql postgres
psql (8.3devel.V003R002C001B0030.65c1c1c)
Type "help" for help.

postgres=#
```

如果使用非超级用户连接，提示符是：

```
postgres=>
```

SQL 语句可能比较长，可以在多行输入语句。分多行输入时，psql 提示符变成数据库名加 -# ，或数据库名加 -> ，数据库以分号作为一条 SQL 语句的结束符。 psql 把用户输入存到一个查询缓冲区中，直到遇到分号，然后开始执行该语句。

- psql 快捷命令， 以分隔符 \ 头
- 输入 e 使用外部编辑器 (默认是 vi)
- 输入 p 显示查询缓冲区内容
- 输入 g 代替分号标识一条语句的结束
- 输入 r 重置查询缓冲区，丢弃已输入的查询语句
- 输入 l 列出所有数据库
- 输入 d 列出表，视图和 sequence
- 输入 q 退出 psql
- 输入 h 查看 SQL 语句帮助
- 输入 ? 查看 psql 帮助

输入 sql 语句后也可以使用 h 命令，所有已输入的语句将会被忽略。

5.4 设置用户认证

集群初始化后，用户认证功能开始生效，Master 实例只接受 gpadmin 用户的连接。在 Master 主机，允许其他用户或其它主机连接数据库，必须配置集群关于认证的配置文件。Segments 节点自动配置为只接受 Master 的连接。

集群支持多种认证方法，包括密码，LDAP，Kerberos (GSSAPI)， Radius， 客户端认证，PAM， 和 SSPI。这个例子中，我们使用 MD5 加密的密码认证。密码被加密并保存在数据库中。查看 MPP 数据库管理员手册获取更多认证方法。

在 $MASTER_DATA_DIRECTORY/pg_hba.conf 文件中配置认证信息。以 # 开头的行是注释，主要描述配置文件的语法和配置选项。

非注释行是认证一个连接请求入口。第一个匹配连接请求的行决定认证方法，所以配置文件中每行的先后顺序需要考虑清楚。每行认证信息取决于连接请求的类型，要连接的数据库和用户名。

连接类型可以是 local(本地 socket 连接)， host (非加密的 TCP/ IP 连接)， 或者 hostssl (使用 SSL 加密的 TCP/IP 连接)。 如果连接类型是 host 或 hostssl，新添加的一行配置包括一个 CIDR 掩码，掩码主要用来决定哪一个网段的主机可以连接数据库。

下面例子中，设置认证信息，使 users 组内的任何一个成员都可以输入密码后访问 tutorial 数据库，其它主机连接请求需要 SSL 认证。

1、gpadmin 用户登录，编辑配置文件$MASTER_DATA_DIRECTORY/pg_hba.conf：

```
$ vi $MASTER_DATA_DIRECTORY/pg_hba.conf
```

2、在该文件最后，添加下面内容：

```
local       tutorial    +users                          md5
local       tutorial    +users      127. 0. 0. 1/28     md5
hostssl     tutoriala   +users      samenet             md5
```

3、查看 《MPP 数据库管理员手册》中完整的语法描述和每一行数据的有效值， 或者阅读 pg_hba.conf 文件中的注释。

4、保存更改，使用下面命令重新加载配置文件 pg_hba.conf 和 postgresql.conf：

```
$ gpstop -u
```

6 卸载数据库

MPP 数据库的工具 gpdeletesystem 工具可卸载数据库， gpdeletesystem 完成下面两个任务：

- 停止所有 kingbase 进程（Segment 实例和 Master 实例）
- 删除所有的数据目录

在运行 gpdeletesystem 之前：

- 将所有备份文件移出 Master 数据目录和 Segment 数据目录
- 确保 MPP 数据库在运行
- 如果用户当前位于 Segment 数据目录中，请将目录更改为另一个位置。从 Segment 数据目录中运行时，该工具会失败，并显示错误。

注意，该工具不会卸载 MPP 数据库软件。

例如：

```
# gpdeletesystem -d /data/Master/gpseg-1
```

其中 /data/Master/gpseg-1 为 Master 实例数据目录，必须和环境变量 MASTER_DATA_DIRECTORY 值相同。

7 附录

7.1数据库端口管理

数据库端口号在初始化配置文件里指定后，Master 和 Segment 实例的端口号已经写入到系统表 gp_segment_configuration 里。 数据库启动过程中就是通过读取该系统表来得知所有的实例的端口号。

修改端口号需要手动更新系统表 gp_segment_configuration，包括以下步骤：

1、先停掉数据库集群：

```
# gpstop
```

2、以维护模式启动 Master 实例：

```
# gpstart -m
```

3、使用 psql 以 utility 模式进入数据库，其中 5432 为当前端口号，5433 为新端口号：

```
# GPOPTIONS='-c gp_session_role=utility' psql postgres -p 5432 -U gpadmin
  psql (8.3devel.V003R002C001B0030.65c1c1c)
  Type "help" for help.

  postgres=# set allow_system_table_mods to 'DML';
  SET
  postgres=# update gp_segment_configuration set port = 5433 where dbid = 1;
  UPDATE 1
```

4、同时修改对应实例数据目录的配置文件 postgres.conf 里最后一行追加 port 选项，如：

```
port = 5433
```

5、关闭 Master 实例：

```
# gpstop
```

6、重新启动集群：

```
# gpstart
```

7.2 问题定位

本节为 MPP 数据库常见问题及解决方法。

7.2.1 安装问题

7.2.1.1 初始化数据库前问题

Q：kingbase 执行文件依赖问题

A：MPP 数据库 安装完成后，使用 ldd $GPHOME/bin/postgres,如果有库找不到的，执行 export LD_LIBRARY_PATH=$GPHOME/lib:$LD_LIBRARY_PATH。如果还有问题，那么请在系统中安装该依赖库。

Q：初始化集群过程中出现找不到 Master 目录

A：初始化前需要创建 Master 和 Segment 对应的数据目录（所有机器上）

Q：交换公钥失败

A：没有关闭防火墙，家目录的权限（应为 700）。

Q：无法使用命令,command not found
A：请 source $GPHOME/mpp_path.sh，确认执行命令都在 Linux 的 PATH 路径下。

7.2.1.2 初始化数据库问题

Q：初始化集群过程中不成功。
A：查看 Master 和 Segment 对应目录中是否有文件或者文件夹，请删除它们。使用命令 netstat －anp | grep port 查看配置文件中对应 Master 或者 Segment 的端口是否被占用。

Q：为什么初始化过程中频繁需要输入密码。
A：未交换公钥，请使用 gpssh-exkeys 命令交换集群间 host 公钥。

Q：安装 mirror 时出现超长等待的情况，并返回错误。
A：初始化 mirror 时先对数据库做 checkpoint。

Q：为什么在安装时出现这个错误 No /home/gpadmin/gpdb/lib on Segment instance h2 Script Exiting!。
A：原因是没有在 h2 上安装 MPP 数据库 客户端，请注意，初始化集群前，请确认所有机器上都安装了 MPP 数据库 客户端，并且路径都相同，建议使用 gpseginstall 安装。

Q：为什么在初始化集群时出现类似 semctl 参数错误的问题。
A：这个问题会在 RHEL7 以上版本中出现，原因是当用户退出 os 后，会 remove 掉所有的 IPC objects，RHEL7 以上版本中 RemoveIPC 默认为 yes，也就是用户推出后会移除掉用户的 shared memory Segments 和 semaphores，而 MPP 数据库 使用了 shared memory Segments 所以会报错，修改方法为：

```
echo “RemoveIPC=no” >> /etc/systemd/logind.conf
systemctl restart system-logind.service
```

Q：安装 mirror 过程中出现了 no such file or directory 样子的错误。
A：由于在写 /etc/hosts 时使用 scp 到其它的节点上，被动将 localhost 信息修改成了同一个 hostname，所以当执行 scp 命令时可能找不到文件。

Q：初始化集群过程中出现了类似 ERROR:root:code for hash md5 was not found
A：这个问题是由于数据库自带的 python 版本与操作系统给出的 lib 库不兼容导致的，使用系统自带的 python 可解决问题，暨将 gpdb/mpp_path.sh 文件中的

```
export PATH=$GPHOME/bin:$PATH
```

修改为

```
export PATH=$PATH:$GPHOME/bin
```

然后重新设置环境变量。

Q：初始化集群时报错，failed to complete obtain psql count Master。

A：可能是由于 license 过期，或者无 license.dat 文件。

7.2.2 访问问题

Q：如何控制外部访问。

A：修改 $MASTER_DATA_DIRECTORY/pg_hba.conf 文件。方式为

TYPE DATABASE USER CIDR-ADDRESS METHOD

例如

```
host kingbase gpadmin 192.168.0.1/32 md5
```

Q：连接用户过多错误解决方式。

A：如果您购买的是控制了连接数的 License，那请联系商务。如果您购买的是控制 CPU 或者磁盘空间的 License，请修改 $MASTER_DATA_DIRECTORY/postgresql.conf 文件中的 max_connections 来增大连接数。

Q：忘记了密码怎么办。

A：首先请修改 $MASTER_DATA_DIRECTORY/pg_hba.conf 中的相关对应参数，将用户的认证方式设置为 trust，然后重新加载配置文件（gpstop -u）。重新登录，修改好密码再将用户的认证方式修改为 md5 或其它认证方式，最后重新加载配置文件。

Q：只修改了 pg_hba.conf 文件后，GPDB 无法停止。

A：host all gpadmin ::1/128 trust 这一行是不允许被删除的。

7.2.3 字符集问题

Q：为什么我查看数据库中的中文是乱码。

A：由于你插入到数据库中存储的字符集为 UTF8，而客户端显示的也是 UTF8 形式，如果需要显示中文可以将 client_enconding 设置为 gb18030。

7.2.4 其它

Q：服务器日志如何获取？

A：服务器日志分为 Master 日志和 Segment 日志，分别存放在 Master 目录和 Segment 目录下的 pg_log 目录中，使用 gpdb-timestamp.csv 的形式记录，在服务器未重启的情况下，每天一个日志。

Q： Segment 节点过多时，会导致 python 执行脚本出错: stderr='ssh_exchange_identification: read: Connection reset by peer'。

A： 原因是 ssh 连接过多，导致错误，可以通过增大 /etc/ssh/sshd_config 中的 MaxStartups 30:90:300（默认为 10:30:100），重启服务 sshd 生效。

Q： 启动时出现如下错误：DTM initialization: failure during startup recovery, retry fa

iled, check Segment status。

A： 可能出现的情况如下：

1. 检查所有服务器的防火墙是否关闭；
2. 检查 Master 上的 shared_buffers 是否设置过大。

Q： 启动时出现如下错误：FAILED host ‘xxx’ datadir ‘xxxxx’ with reason: ‘PG_CTL failed.’。

A： 请检查数据目录文件夹的权限是否被修改。

Q：带分区的表使用 explain SQL 查看执行计划时，发现很慢。

A：这个问题是由于在做执行计划时，会对分区进行大小查询，如果统计信息中有就跳过了，因此会很慢，如果做过 analyze 就不会慢了。