

密 级: 对外公开

编号: JCBG20201057



KADB 产品与星辰天合企业级分布式 统一数据平台 edp v4.0 产品

兼容性测试报告

星辰天合(北京)数据科技有限公司

北京人大金仓信息技术股份有限公司

2020年11月26日



目录

1 测试目的.....	1
2 厂商信息.....	1
3 测试时间/地点/人员.....	1
4 测试环境.....	1
4.1 操作系统环境.....	1
4.2 软件环境.....	2
4.3 硬件环境.....	2
4.4 测试环境架构及说明.....	2
5 测试策略及用例.....	3
5.1 测试策略.....	3
5.2 测试列表及测试结果.....	3
5.3 测试用例及测试记录.....	4
6 测试结论.....	8
6 附录（非必填，如不填写请删除）.....	错误！未定义书签。

1 测试目的

对待测产品星辰天合企业级分布式统一数据平台 edp v4.0 与人大金仓分析型数据库产品 KADB 的兼容性、功能等方面加以验证，为金仓认证得授予提供依据，同时为双方更广泛的合作打下基础。

2 厂商信息

2.1 厂商信息

公司名称	星辰天合（北京）数据科技有限公司		
公司地址	北京市海淀区蓝靛厂金源时代购物中心 B 区 2#B 座 806-1		
邮政编码	100081	公司网址	www.xsky.com
联系人	李远军	电子邮件	
电话	13810010394	传真	

2.2 人大金仓公司信息

公司名称	北京市人大金仓信息技术股份有限公司		
公司地址	北京市朝阳区容达路 7 号 E 座 2 层		
邮政编码	100102	公司网址	https://www.kingbase.com.cn/
联系人	张文军	电子邮件	wjzhang@kingbase.com.cn
电话	18811401011	传真	010-5911 1032

3 测试时间/地点/人员

- 1、测试地点：
- 2、测试时间：2020 年 11 月 1 日 至 2020 年 11 月 26 日
- 3、测试人员：

姓名	职务/职称	联系电话
靳邻	技术支持工程师	18501140627
金东宜	技术经理	18611103715

4 测试环境

4.1 操作系统环境

操作系统环境信息截图

```
Linux kadb-host3 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

4.2 软件环境

软件环境如下表：

被测软件

软件名称 (与软件著作权名称一致)	产品英文 (与软件著作权名称一致)	版本号	软件用途及简介
金仓分析性数据库系统 V3.0	KingbaseAnalyticsDBV3	V3	数据仓库
星辰天合企业级分布式统一数据平台 edp v4.0	XEDP V4.0	V4	分布式存储

被测软件运行、测试依赖的第三方软件

类型	软件名称	软件版本	软件厂商
OS	CentOS	7.4	红帽
数据库			
中间件			

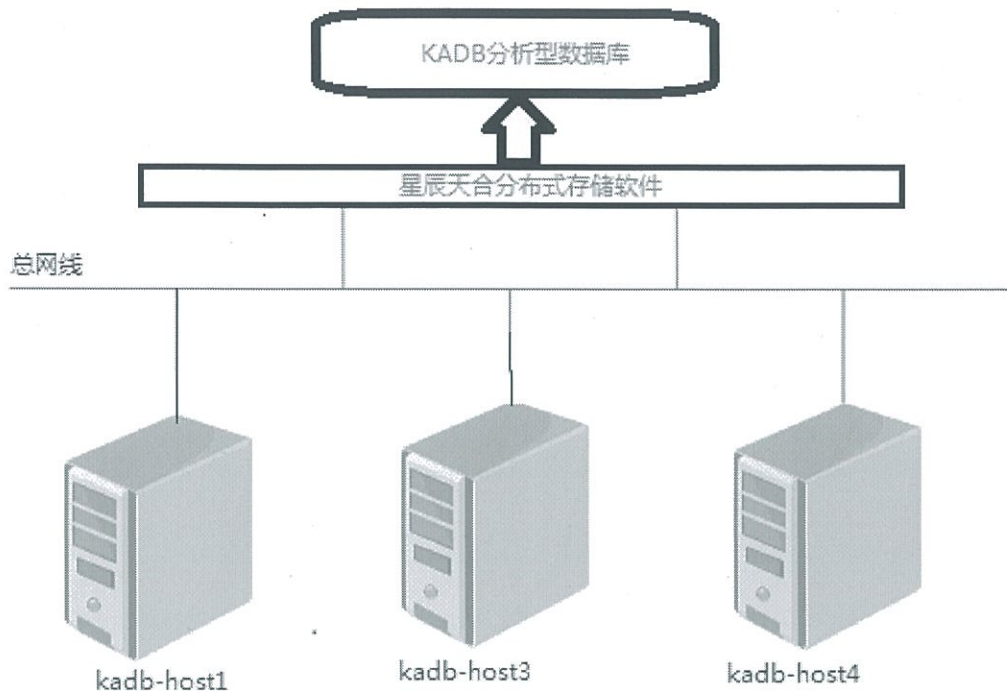
4.3 硬件环境

硬件环境如下表：（需写清适配的 cpu 的型号等信息）

服务器名称	配置类别	型号参数	数量	设备用途及说明
kadb-host1 kadb-host3 kadb-host4	处理器	Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz	3	分布式存储及 KADB (mpp) 数据库安装环境
	内存	16GB		
	硬盘	STAT 盘		
	网卡	千兆网卡		
	处理器			
	内存			
	硬盘			
	网卡			

4.4 测试环境架构及说明

1、（单台）测试环境架构图



2、测试环境说明

本次测试主要目的是测试人大金仓分析型数据库系统是否可以正常运行在星辰天合分布式存储介质上，且无明显缺陷或不兼容问题。采用的是三台物理节点，共虚拟 12 个实例进行测试。

5 测试策略及用例

5.1 测试策略

手动测试

测试项包括：

1. 数据库安装
2. 数据库正常启停
3. 数据库基本功能
4. 数据分布
5. 数据库行列存储
6. 数据库快速卸载

测试工具：

无

5.2 测试列表及测试结果

● 功能测试

测试方案	用例编号	用例名称	结果 (Pass/Fals)
------	------	------	----------------

测试用例	1	数据库安装	PASS
	2	数据库正常停止	PASS
	3	数据库基本功能	PASS
	4	数据分布	PASS
	5	数据库行列存储	PASS
	6	卸载测试	PASS

5.3 测试用例及测试记录

5.3.1 数据库安装

项目	功能测试	分项目	数据库安装
用例编号	1	版本	
测试目的	检查数据库系统是否支持该系统		
预置条件	1. 硬件平台操作系统安装正常; 2. 数据库安装包和授权文件传到服务器上		
测试步骤	1、根据数据库安装文档安装 MPP 数据库 2、安装完成		
执行命令语句	01-MPP安装部署 规范手册.docx		
预期结果	数据库安装完成; 数据库正常启动;		
测试结果	数据库安装成功		
备注			
测试中心人员:	靳邻	测试时间	2020.11.26
测试厂商人员:			

5.3.2 数据库启停

项目	功能测试	分项目	数据库启停
用例编号	2	版本	
测试目的	检查数据库系统是否可以在服务器上进行正常启停		
预置条件	1. 数据库已安装且正常运行;		

测试步骤	1、 正常关闭数据库 gpstop -aM fast 2、 正常启动数据库 apstart -a		
执行命令语句			
预期结果	语句执行结果正确		
测试结果	数据库系统可以在长城服务器正常启停		
备注			
测试中心人员:	靳邻	测试时间	2020. 11. 26
测试厂商人员:			

5.3.3 数据库基本功能

项目	功能测试	分项目	数据库基本功能
用例编号	3	版本	
测试目的	检查数据库系统是否可以在服务器上进行基本操作		
前置条件	1. 数据库已安装且正常运行;		
测试步骤	1、 连接数据库 <code>psql postgres</code> 2、 创建测试表 test <code>CREATE table test(id int,name bigint);</code> 3、 插入一定量的数据 <code>INSERT INTO test values(generate_series(1,1000),5);</code> 4、 更新表中数据 <code>UPDATE test set name=6 where id < 500;</code> 5、 查询表中数据 <code>SELECT * from test limit 10;</code> 6、 删除测试表 <code>DROP TABLE test;</code>		
执行命令语句			
预期结果	SQL 语句执行结果正确		

测试结果	数据库系统可以在长城服务器正常执行 SQL		
备注			
测试中心人员:	靳邻	测试时间	2020.11.26
测试厂商人员:			

5.3.4 数据分布

项目	功能测试	分项目	数据分布
用例编号	4	版本	
测试目的	检查数据库系统能否在服务器上实现数据均匀分布		
前置条件	数据库已安装且正常运行		
测试步骤	1、连接数据库 psql postgres 2、创建测试表 test, 并插入部分数据库 CREATE TABLE test (id int,name bigint); INSERT INTO test VALUES(generate_series(1,1000),5); 3、通过函数查看各个实例中存放的数据量 SELECT gp_segment_id,count(1) FROM test GROUP BY gp_segment_id;		
执行命令语句			
预期结果	数据均匀分布在各个实例节点		
测试结果	数据库可以在服务器上实现数据均匀分布		
备注			
测试中心人员:	靳邻	测试时间	2020.11.26
测试厂商人员:			

5.3.5 行列存储

项目	功能测试	分项目	行列存储
用例编号	5	版本	
测试目的	检查数据库系统是否能在服务器上实现行、列存储模式		
前置条件	数据库已安装且正常运行		
测试步骤	1、创建行存表 heap_table, 并查看表结构		

	<pre>CREATE TABLE heap_table(id int); \d+ heap_table 2、创建列存表 ao_table, 并查看表结构 CREATE TABLE ao_table (id int) WITH (APPENDONLY=true, ORIENTATION=COLUMN, COMPRESSTYPE=ZLIB, COMPRESSLEVEL=7); \d+ ao_heap</pre>		
执行命令语句			
预期结果	SQL 被正确执行		
测试结果	<u>数据库支持在长城服务器上</u> 进行行列存储模式的创建		
备注			
测试中心人员:	靳邻	测试时间	2020.11.26
测试厂商人员:			

5.3.6 数据库卸载

项目	功能测试	分项目	数据库卸载
用例编号	6	版本	
测试目的	检查数据库系统是否能在服务器上实现快速卸载		
预置条件	数据库已安装且正常运行		
测试步骤	1、执行数据库卸载语句		
执行命令语句	gpdeletesystem /home/mppadmin/dbdata/master/mppseg-1		-d
预期结果	数据库被卸载		
测试结果	<u>数据库系统可以在服务器上正常卸载</u>		
备注			
测试中心人员:	靳邻	测试时间	2020.11.26
测试厂商人员:			

6 测试结论

本次执行测试用例 6 条，测试通过 6 条，没有发现明显缺陷，测试基本达到预期效果，基于测试给出的结果认为星辰天合企业级分布式统一数据平台产品在兼容性、功能方面与人大金仓数据库完全兼容。

MPP 安装部署规范手册

一、MPP 安装过程概述

1. 选择 MPP 安装方式

MPP 数据库安装提供两种安装方式，一种为单机、一种为多机。

- 单机安装

安装 MPP 在一台服务器上，master、segment 使用同一 IP 不同端口运行，适合在于适配、介绍、开发以及功能测试等工作中使用，无法确保高可用。

- 多机安装

安装 MPP 在多台服务器上，master、segment 及其备用节点使用不同 IP 不同端口运行，适合在部署、上线、POC 以及性能测试等工作中使用。

2. 选择 MPP 依赖项检测

MPP 依赖项检测是检测安装操作系统(linux, 暂不支持 windows 操作系统搭建, 但可使用 windows 命令行连接搭建好的数据库集群)的环境是否满足数据库运行。

- Linux

libbz2.so.1
librt.so.1
libssl.so.10
libcrypto.so.10
libz.so.1
libdl.so.2
libcurl.so.4
libpthread.so.0
libm.so.6
libc.so.6
libgssapi_krb5.so.2
libkrb5.so.3
libcom_err.so.2
libk5crypto.so.3

libidn.so.11
libldap-2.4.so.2
libssl3.so
libsmime3.so
libnss3.so
libnssutil3.so
libplds4.so
libplc4.so
libnspr4.so
libssh2.so.1
libkrb5support.so.0
libkeyutils.so.1
libresolv.so.2
liblber-2.4.so.2
libsasl2.so.2
libselinux.so.1
libcrypt.so.1
libfreebl3.so
libyaml-0.so.2
libevent-1.4.so.2
libapr-1.so.
libnsl.so.1
libuuid.so.1

装客户端，需要安装依赖项才能正常安装和运行数据库。

3. 软硬件要求

下表所示为 MPP 集群支持的最小配置，建议咨询技术人员确定最终的硬件配置。

项目	配置	检查语句
操作系统	CentOS6.5 以上	cat /proc/version cat /etc/redhat-release
文件系统	建议使用 XFS 文件系统	which xfs_quota

CPU	Pentium Pro compatible(64 位)	cat /proc/cpuinfo
内存	每个服务器最小 2G	free -g
磁盘	150M 用于 MPP 的安装 300M 用于 segment 储存的元数据 数据磁盘空间的用量不要超过 70%	fdisk -lu; df -lh
网络	Gigabit 以太网络以上（千兆网络）	ethtool
软件和工具	bash shell, GNU tars, GNU zip, GNU sed	

4. 数据库 license

数据库 license 是 MPP 产商提供的授权文件，其中会对数据库有效日期、发布类型、最大并发连接数、最大 segment 数、磁盘容量等信息进行设置。

如果在 license 中信息与操作系统中信息不匹配，会导致数据库无法启动。

5. 操作系统配置

操作系统配置具体如下（以下为 CentOS 操作系统下的操作规范，如有其它版本，请按照对应的命令操作）：

- Linux（在所有机器上用 root 用户执行）

1. 修改/etc/hosts（通过主机名访问所有服务器）

```
cat >> /etc/hosts <<EOF
IPADDRESS1 HOSTNAME1
.....
IPADDRESSn HOSTNAMEn
EOF
```

2. 修改主机名

```
#centos6 中修改/etc/sysconfig/network 文件
HOSTNAME=HOSTNAME1
```

```
source /etc/sysconfig/network
```

```
hostname $HOSTNAME
```

```
#centos7 中
```

```
hostnamectl --static set-hostname HOSTNAME1
```

3. 创建 mppadmin 用户，假定密码为 123456：（mppadmin 为标准化安装中管理 MPP 集群的超级用户）

```
useradd mppadmin -m -p $(openssl passwd -1 123456)
```

4. 修改 linux 系统参数

```
#centos6 中
```

```
service iptables stop; chkconfig iptables off; setenforce 0;
```

```
#centos7 中
```

```
systemctl status firewalld
```

```
systemctl stop firewalld
```

```
systemctl disable firewalld
```

```
sed -i 's/^GSS/#&/g' /etc/ssh/sshd_config; service sshd reload;
```

```
sysctl -p - >> /etc/sysctl.conf <<EOF
```

```
# MPP configurations
```

```
kernel.sysrq=1
```

```
kernel.core_pattern=core
```

```
kernel.core_uses_pid=1
```

```
kernel.msgmnb=65536
```

```
kernel.msgmax=65536
```

```
kernel.msgmni=2048
```

```
kernel.shmmax = 68719476736
```

```
kernel.shmmni = 4096
```

```
kernel.shmall = 4294967296
```

```
kernel.sem=25600 3200000 10000 14200
```

```
net.ipv4.tcp_syncookies=1
```

```
net.ipv4.ip_forward=0
```

```
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=1025 65535
net.core.netdev_max_backlog=10000
net.core.rmem_max=33554432
net.core.wmem_max=33554432
vm.overcommit_memory=2
vm.overcommit_ratio=95
net.netfilter.nf_conntrack_max = 655360
EOF
```

```
#=====
```

#如果加载 net.netfilter.nf_conntrack_max 参数时报错, 查看系统是否加载了 ip_conntrack 模块 (未报错则跳过此步骤):

```
lsmod | grep conntrack
```

#如果这是空的, 用以下命令加载:

```
modprobe ip_conntrack
```

#再加载 net.netfilter.nf_conntrack_max 参数

#将 ip_conntrack 模块儿加载添加到开机启动项中

```
cat >>/etc/rc.d/rc.local<<EOF
```

```
modprobe ip_conntrack
```

```
EOF
```

```
#=====
```

#centos6 中

```
cat >> /etc/security/limits.d/90-nproc.conf <<EOF
```

```
#MPP configurations
```

```
* soft nofile 65536
```

```
* hard nofile 65536
```

```
* soft nproc 131072
```

```
* hard nproc 131072
```

```
EOF
```

#centos7 中

```
cat >>/etc/security/limits.d/20-nproc.conf <<EOF
```

```
#MPP configurations
```

```
* soft nofile 65536
```

```

* hard nfile 65536
* soft nproc 131072
* hard nproc 131072
EOF

cat >> /usr/lib/tmpfiles.d/tmp.conf <<EOF
r! /tmp/.s.POSTGRES.*
r! /tmp/*_master_ha.pid
r! /tmp/.gpstop.*.lock
EOF

#注意如果系统为 CentOS7.2, 需要修改下面参数
echo "RemoveIPC=no" >> /etc/systemd/logind.conf
service systemd-logind restart

```

5. 修改磁盘列队方式以及文件预读大小, 假设磁盘名为/dev/sda

```

/sbin/blockdev --getra /dev/sda; /sbin/blockdev --setra 16384 /dev/sda
echo deadline > /sys/block/sda/queue/scheduler

```

6. 数据库初始化配置选项

数据库初始化过程中需要对一些基本参数进行设置, 确定 database 的基本数据, 在初始化前需要先检查端口或目录是否以及被使用。

参数	配置值
Master Port(master 节点端口)	5432 (默认, 可以修改)
Segment Port(segment 节点初始端口)	40000 (默认, 可以修改)
Database Administrartor Name (数据库用户名)	mppadmin (根据实际情况可以自行修改)
Database Administrartor Possword (数据库密码)	Local 登录时无密码 (根据实际情况可以自行修改)
Default database (数据库名称)	postgres (根据实际情况可以自行修改)
Master Data path (master 数据目录路径)	/home/mppadmin/dbdata/master (根据实际情况可以自行修改)
Segment Data path (segment 数据目录路径)	/home/mppadmin/dbdata/primary (根据实际情况可以自行修改)
Encoding (字符集)	UNICODE

二、MPP 预安装工作

1. linux

1) 软硬件环境准备

- 推荐硬件环境:

CPU: 主流 64 位 CPU。

内存: 16GB 以上。

硬盘: 1TB 以上空闲空间。

- 软件依赖:

- Linux 系统

系统	版本
CentOS	6.5 64bit
XFS 相关包 (可选)	

- HA

HA	版本

- 中间件

中间件版本	版本

2) 安装环境准备

a) 操作系统用户要求

安装用户对于安装路径需要有“读”、“写”、“执行”的权限。linux 中，需

要以非 root 用户执行安装程序，且该用户需要有标准的 home 目录。

注意：建议安装数据库使用新建的 mppadmin 用户，密码设置为 123456，可修改为比较复杂的密码，但每台服务器主机上密码需要相同。

b) 安装客户端（Linux 系统，使用 mppadmin 用户在 master 主机上执行）

上传 MPP 介质及 license 文件到搭建集群的所有服务器/home/mppadmin/目录下。

MPP 客户端安装过程中需要输入全路径，在所有服务器安装客户端，命令如下：

```
sh /home/mppadmin/ MPP-V001R001C001B0009-CENTOS7-x86_64.run
/home/mppadmin/MPP
```

也可以自定义安装路径，为方便管理，建议安装路径为/home/mppadmin/MPP。

所有服务器上，将 license 拷贝到/home/mppadmin/MPP/bin 目录下：

```
cp /home/mppadmin/license.dat /home/mppadmin/MPP/bin
```

注意：在正式的生产环境中，master 和 master standby 的 license 都会根据网卡的 mac 地址生产，需要将 license 放到相对应的主机的 MPP 目录中，且 license 的名字必须为 licence.dat。

在 greenplum_path.sh 里设置 SSH_PORT 端口号（默认端口为 22），如端口号为默认端口则跳过此步骤：

```
cat >>/home/mppadmin/MPP/greenplum_path.sh <<EOF
export SSH_PORT=更改后的端口号
EOF
```

为方便每次登陆后直接使用 MPP 命令，可使用如下命令：

```
cat >> ~/.bashrc <<EOF
source /home/mppadmin/MPP/greenplum_path.sh
EOF
source ~/.bashrc
```

由于集群的原因，需要将所有服务器交换密钥，能够互相之间免密访问，命令如下：

```
#创建集群 host 配置文件，也就是你安装客户端的所有主机名
```

```
cat >> /home/mppadmin/hostfile <<EOF
host1
host2
.....
hostn
EOF
#交换公钥
gpssh-exkeys -f /home/mppadmin/hostfile
#按照提示输入密码
```

c) 数据目录

创建数据目录及备份目录

```
#创建数据目录/dbdata, 备份目录/dbbackup
gpssh -f /home/mppadmin/hostfile -e 'mkdir -p /home/mppadmin/dbdata'
gpssh -f /home/mppadmin/hostfile -e 'mkdir -p /home/mppadmin/dbbackup'
#创建次一级 master 实例目录和 segment 实例目录
gpssh -f /home/mppadmin/hostfile -e 'mkdir -p /home/mppadmin/dbdata/master'
gpssh -f /home/mppadmin/hostfile -e 'mkdir -p /home/mppadmin/dbdata/primary'
```

3) 系统环境测试

使用脚本检测集群节点间网络和各节点 I/O 是否正常。

```
#假设数据所在目录为/home/mppadmin/dbdata
#检查网络
gpcheckperf -f /home/mppadmin/hostfile -r N -D -d /home/mppadmin/dbdata
#检查磁盘, 注意如果不带-S 参数, 默认生成服务器内存大小的数据
gpcheckperf -f /home/mppadmin/hostfile -d /home/mppadmin/dbdata -r d -D -v
```

三、MPP 数据库初始化

1. 数据库初始化

数据库初始化主要功能是创建实例, 并生成相应 master 和 segment 的数据目录。使用之前的 hostfile 中的主机创建集群, 确定某一个节点为 master, 其它节

点为 segment。首先创建配置文件，命令如下：

```
#创建初始化集群系统配置文件，
cat >> /home/mppadmin/gpinitssystem_config <<EOF
ARRAY_NAME="MPP"
#segment 节点名 prefix
SEG_PREFIX=mppseg
#segment 节点基础端口，每多一个 segment 会自动增加 1，默认为 40000
PORT_BASE=40000
#用于申明 segment 目录地址以及共多少个 segment，其中每台主机有多少个 segment 就
对应多少个文件夹，下面为两个文件夹，暨每台主机 2 个 segment
declare -a DATA_DIRECTORY=(/home/mppadmin/dbdata/primary
/home/mppadmin/dbdata/primary)
#master 主机地址
MASTER_HOSTNAME=host1
#master 目录
MASTER_DIRECTORY=/home/mppadmin/dbdata/master
#master 端口，默认为 5432
MASTER_PORT=5432
TRUSTED_SHELL=ssh
CHECK_POINT_SEGMENTS=8
ENCODING=UNICODE
#选择需要初始化 segment 的主机名，列出到 hostfile 目录
MACHINE_LIST_FILE=/home/mppadmin/hostfile
EOF
注：如果更改了默认端口，则需要将配置文件中的 TRUSTED_SHELL=ssh 一行注释掉，
即改为# TRUSTED_SHELL=ssh
#初始化集群
gpinitssystem -c /home/mppadmin/gpinitssystem_config -a
```

注意：在初始化完成后，请将 MASTER_DATA_DIRECTORY 目录写入.bashrc 中。如：

```
cat >> ~/.bashrc << EOF
export MASTER_DATA_DIRECTORY=/home/mppadmin/dbdata/master/mppseg-1
EOF
source ~/.bashrc
```

2. 数据库高可用模式

1) master standby

由于 MPP 为单主控节点模式，所以在主控节点失效后，无法继续对外提供接口，因此需要在主控节点主机之外的主机上创建一个 standby 节点。

创建方法如下：

```
#使用 gpssh 在 standby 所使用的节点上创建数据目录，假定 master 目录为
/home/mppadmin/dbdata/master, standby 节点为 host2:

gpssh -h host2 -e "mkdir -p /home/mppadmin/dbdata/master"

#分别在主备管理节点创建 master 备份目录（用于重添加之前备份原 master 目录）

gpssh -h host2 -e "mkdir /home/mppadmin/master_backup"

mkdir /home/mppadmin/master_backup

#启动虚拟 IP 地址

sudo ifconfig eth0:0 xxx.xxx.xxx.xxx netmask 255.255.255.0 up
```

手动配置 HA 配置文件——cluster_ha.ini，配置参数如下（master 与 master standby 均需配置，cluster_ha.ini 的模板默认路径为 /home/mppadmin/MPP/bin/cluster_ha.ini）：

```
©master 节点

RUNLEVEL=0 #脚本运行等级，0: ha+autostate、1: ha、2: autostate

ENV_FILENAME=greenplum_path.sh #数据库管理工具环境

INSTALL_DIRECTORY=/home/mppadmin/MPP #数据库安装目录

CURRENT_HOSTNAME=host1 #本机主机名

CURRENT_PORT=5432 #本机 master 实例端口

CURRENT_DATA_DIRECTORY=/home/mppadmin/dbdata/master/mppseg-1 #本机 master
实例目录

PEER_PORT=5432 #探测主机 master 实例端口

PEER_HOSTNAME=host2 #探测主机主机名

PEER_DATA_DIRECTORY=/home/mppadmin/dbdata/master/mppseg-1 #探测主机 master
实例目录

DBNAME=postgres #连接数据库名

CURRENT_MASTER_BACKUP_DIRECTORY=/home/mppadmin/master_backup#本机
master 目录备份路径
```

PEER_MASTER_BACKUP_DIRECTORY=/home/mppadmin/master_backup #备机 master
目录备份路径

TRUST_SERVER=192.168.2.1 #受信赖的第三方 IP 地址 (防脑裂)

LINUX_USER=mppadmin #linux 系统登录数据库的用户名

ENABLE_VIP=0 #是否启用 VIP, 0: 启用、1: 停用

VIP_CARD_CURRENT=eth0 #本机虚 IP 所在网卡

VIP_CARD_PEER=eth0 #探测主机虚 IP 所在网卡

VIP_CARD_LABEL=1 #虚拟网卡号

VIP_ADDR=10.0.0.251 #虚 IP 地址

NETMASK=255.255.255.0 #子网掩码

CURRENT_GPSTOP_FILE="/tmp/.\$ {GPSTOP}.\$ {CURRENT_PORT}.lock" #本机数据库

锁文件名称

PEER_GPSTOP_FILE="/tmp/.\$ {GPSTOP}.\$ {PEER_PORT}.lock" #备机数据库锁文件名称

称

CHECK_MASTER_RUNNING_STATE_TIMES=5 #状态监测次数

CHECK_MASTER_RUNNING_STATE_INTERVAL=1 #状态监测间隔时间

PING_COUNT=5 #ping 主机次数

PING_INTERVAL=0.5 #ping 主机时间间隔

PING_DEADLINE=5 #连接时长

LOG_DIRECTORY=/tmp #log 目录路径

CRONTAB_INTERVAL=1 #crontab 的执行时间间隔 (单位为分钟)

LOG_MIN_MESSAGE=1 #日志级别

STANDBY_NUMBER=1 #

PID_FILE=/tmp/gpdb_cluster_ha.pid #临时 pid

CONNECT_TIMEOUT=10 #连接超时时间

RECYCLE_TIMES=15 #探测次数

INTERVAL=1 #探测间隔时间

SSH="ssh -T -o StrictHostKeyChecking=no"

©master standby 节点

RUNLEVEL=0 #脚本运行等级, 0: ha+autostate、1: ha、2: autostate

ENV_FILENAME=greenplum_path.sh #数据库管理工具环境

INSTALL_DIRECTORY=/home/mppadmin/MPP #数据库安装目录

CURRENT_HOSTNAME=host2 #本机主机名

CURRENT_PORT=5432 #本机 master 实例端口

```
CURRENT_DATA_DIRECTORY=/home/mppadmin/dbdata/master/mppseg-1 #本机 master
实例目录
PEER_PORT=5432 #探测主机 master 实例端口
PEER_HOSTNAME=host1 #探测主机主机名
PEER_DATA_DIRECTORY=/home/mppadmin/dbdata/master/mppseg-1 #探测主机 master
实例目录
DBNAME=postgres #连接数据库名
CURRENT_MASTER_BACKUP_DIRECTORY=/home/mppadmin/master_backup#本机
master 目录备份路径
PEER_MASTER_BACKUP_DIRECTORY=/home/mppadmin/master_backup #备机 master
目录备份路径
TRUST_SERVER=192.168.2.1 #受信赖的第三方 IP 地址（防脑裂）
LINUX_USER=mppadmin #linux 系统登录数据库的用户名
ENABLE_VIP=0 #是否启用 VIP，0：启用、1：停用
VIP_CARD_CURRENT=eth0 #本机虚 IP 所在网卡
VIP_CARD_PEER=eth0 #探测主机虚 IP 所在网卡
VIP_CARD_LABEL=1 #虚拟网卡号
VIP_ADDR=10.0.0.251 #虚 IP 地址
NETMASK=255.255.255.0 #子网掩码
CURRENT_GPSTOP_FILE="/tmp/.$${GPSTOP}.$${CURRENT_PORT}.lock" #本机数据库
锁文件名称
PEER_GPSTOP_FILE="/tmp/.$${GPSTOP}.$${PEER_PORT}.lock" #备机数据库锁文件名
称
CHECK_MASTER_RUNNING_STATE_TIMES=5 #状态监测次数
CHECK_MASTER_RUNNING_STATE_INTERVAL=1 #状态监测间隔时间
PING_COUNT=5 #ping 主机次数
PING_INTERVAL=0.5 #ping 主机时间间隔
PING_DEADLINE=5 #连接时长
LOG_DIRECTORY=/tmp #log 目录路径
CRONTAB_INTERVAL=1 #crontab 的执行时间间隔（单位为分钟）
LOG_MIN_MESSAGE=1 #日志级别
STANDBY_NUMBER=1 #
PID_FILE=/tmp/.gpdb_cluster_ha.pid #临时 pid
CONNECT_TIMEOUT=10 #连接超时时间
```

```
RECYCLE_TIMES=15 #探测次数
INTERVAL=1 #探测间隔时间
SSH="ssh -T -o StrictHostKeyChecking=no"
```

添加 master standby

```
#利用 gpinitstandby 命令添加 master 的镜像
gpinitstandby -s host2

#可以通过 MPP 数据库系统视图 pg_stat_replication 检查 master 镜像的进程状态。
psql postgres -c 'SELECT pid, state FROM pg_stat_replication;'

通过设置 crontab 定时任务来对 master 和 master standby 进行定时检测（主备 master 均需要设置定时任务）：

crontab -e

*/1 * * * * bash /home/gpadmin/MPP/bin/cluster_ha >> /tmp/cluster_ha.log 2>&1
```

2) segment mirror

在/home/mppadmin/dbdata 文件下创建目录 mirror（所有节点），命令如下：

```
gpssh -f /home/mppadmin/hostfile -e "mkdir -p /home/mppadmin/dbdata/mirror"
#环境变量不变，用 gpaddmirror 命令在 primary segment 的服务器上操作
gpaddmirrors -p 1000
#输入 mirror 目录
/home/mppadmin/dbdata/mirror
```

3. 数据库启停方式

在 MASTER_DATA_DIRECTORY 已写入.bashrc 中时，在 mppadmin 用户下使用如下命令启停：

```
#启动集群
gpstart -a

#停止集群
gpstop -a

#重启集群
gpstop -ra -M fast
```

4. 卸载

- 删除集群

保持集群启动

使用 `gpdeletesystem` 删除集群。

```
gpdeletesystem -d /home/mppadmin/dbdata/master/mppseg-1/
```

■ 删除程序

确认集群停止，然后在所有节点上删除相应的 MPP 目录即可。

```
rm -rf /home/mppadmin/MPP
```

四、安装后任务

1. 调整配置参数

安装完数据库之后，需要针对业务应用系统的需求来进行参数调整。具体的参数配置方法详见 MPP 管理员手册。

配置参数的文件在所有节点（master 以及 segment）的目录中，名字为 `postgresql.conf`。常用配置修改选项见表 4-1。

表 4-1 master（主控节点）常用配置项

配置参数	建议配置数值
<code>shared_buffers</code>	<code>max_connections * 16kB</code>
<code>max_connections</code>	250（具体根据应用来定）
<code>log_statement</code>	<code>mod</code> （包含了 <code>INSERT</code> , <code>UPDATE</code> , <code>DELETE</code> , <code>TRUNCATE</code> , and <code>COPY FROM</code> . <code>PREPARE</code> , <code>EXECUTE</code> 等语句以及 <code>CREATE</code> , <code>ALTER</code> , <code>DROP</code> 等 <code>ddl</code> 语句）
<code>statement_mem</code>	125MB（根据执行计划设置大小）
<code>gp_vmem_protect_limit</code>	物理内存/本机 segment 数/SQL 复杂度 需根据情况调整 $((\text{SWAP} + (\text{RAM} * \text{vm.overcommit_ratio} / 100)) / \text{max_primary_segments_per_server_with_failure})$
<code>log_duration</code>	<code>on</code> （记录 SQL 执行时间）
<code>log_filename</code>	<code>'MPP_log_%a.csv'</code>
<code>log_rotation_age</code>	1440
<code>log_rotation_size</code>	0 该值设为 0 则表示禁用以日志大小来判定创建新的日志文件

log_truncate_on_rotation	On
log_autostats	off (分析数据库时产生的日志, 不用开启)
gp_workfile_limit_files_per_query	100000 (每个 segment 工作时可使用的溢出文件最大数)
gp_workfile_limit_per_query	0 (每个 segment 工作时溢出文件可占用的最大磁盘空间大小, 0 为无限制)
maintenance_work_mem	1GB (创建索引更快)
wal_buffers	512MB
gp_default_storage_options	'appendonly=true,blocksize=8192,compressstype=zlib,compresslevel=5,checksum=true,orientation=row'

表 4-2 segment (计算节点) 常用配置项

配置参数	建议配置数值
shared_buffers	max_connections * 16kB
max_connections	250 (具体根据应用来定)
log_statement	mod (包含了 INSERT, UPDATE, DELETE, TRUNCATE, and COPY FROM. PREPARE, EXECUTE 等语句以及 CREATE, ALTER, DROP 等 ddl 语句)
statement_mem	125MB (根据执行计划设置大小)
gp_vmem_protect_limit	物理内存/本机 segment 数/SQL 复杂度 需根据情况调整 $((\text{SWAP} + (\text{RAM} * \text{vm. overcommit_ratio} / 100)) / \text{max_primary_segments_per_server_with_failure})$
log_duration	on (记录 SQL 执行时间)
log_filename	'MPP_log_%a.csv'
log_rotation_age	1440
log_rotation_size	0 该值设为 0 则表示禁用以日志大小来判定创建新的日志文件
log_truncate_on_rotation	On
log_autostats	off (分析数据库时产生的日志, 不用开启)
gp_workfile_limit_files_per_query	100000 (每个 segment 工作时可使用的溢出文件最大数)
gp_workfile_limit_per_query	0 (每个 segment 工作时溢出文件可占用的最大磁盘空间大小, 0 为无限制)
maintenance_work_mem	1GB (创建索引更快)
wal_buffers	512MB
gp_default_storage_options	'appendonly=true,blocksize=8192,compressstype=zlib,compresslevel=5,checksum=true,orientation=row'

- 注意：针对不同的应用场景，可以调整配置参数，可以带来更好，更稳定的性能。

2. 用户访问

MPP 使用与 kingbase 相同的方式限制方式，配置文件为 `pg_hba.conf`，在 MPP 中只需配置 master 节点和 master standby 的访问节点配置文件即可。例如设置所有用户在使用 host 登录时都需要输入密码：

```
cat >> /home/mppadmin/dbdata/master/mppseg-1/pg_hba.conf <<EOF
host      all          all          0.0.0.0/0    md5
EOF
#重载配置文件
gpstop -u
#登录方式，在 master 上执行
psql postgres
#进入后可以创建表及插入数据
create table t1 (i int, j int);
insert into t1 values (1,1),(2,2),(3,3);
```