

达梦技术手册

DM8 集群代理使用手册

Service manual of DM8_DCP



前言

概述

本文档主要介绍 DM 集群代理的架构、提供的功能、以及如何搭建和使用 DM 集群代理。

读者对象





本文档主要适用于 DM 数据库的：

- 开发工程师
- 测试工程师
- 技术支持工程师
- 数据库管理员

通用约定

在本文档中可能出现下列标志，它们所代表的含义如下：

表 0.1 标志含义

标志	说明
 警告：	表示可能导致系统损坏、数据丢失或不可预知的结果。
 注意：	表示可能导致性能降低、服务不可用。
 小窍门：	可以帮助您解决某个问题或节省您的时间。
 说明：	表示正文的附加信息，是对正文的强调和补充。

在本文档中可能出现下列格式，它们所代表的含义如下：

表 0.2 格式含义

格式	说明
宋体	表示正文。
Courier new	表示代码或者屏幕显示内容。
粗体	表示命令行中的关键字（命令中保持不变、必须照输的部分）或者正文中强调的内容。标题、警告、注意、小窍门、说明等内容均采用粗体。
<>	语法符号中，表示一个语法对象。
::=	语法符号中，表示定义符，用来定义一个语法对象。定义符左边为语法对象，右边为相应的语法描述。
	语法符号中，表示或者符，限定的语法选项在实际语句中只能出现一个。
{ }	语法符号中，大括号内的语法选项在实际的语句中可以出现 0...N 次 (N 为大于 0 的自然数)，但是大括号本身不能出现在语句中。
[]	语法符号中，中括号内的语法选项在实际的语句中可以出现 0...1 次，但是中括号本身不能出现在语句中。
关键字	关键字在 DM_SQL 语言中具有特殊意义，在 SQL 语法描述中，关键字以大写形式出现。但在实际书写 SQL 语句时，关键字既可以大写也可以小写。

访问相关文档

如果您安装了 DM 数据库，可在安装目录的“\doc”子目录中找到 DM 数据库的各种手册与技术丛书。

您也可以通过访问我们的网站 www.dameng.com 阅读或下载 DM 的各种相关文档。

联系我们

如果您有任何疑问或是想了解达梦数据库的最新动态消息，请联系我们：

网址：www.dameng.com

技术服务电话：400-991-6599

技术服务邮箱：dmtech@dameng.com

目录

1 DM8 集群代理简介	1
2 体系架构	2
3 功能模块管理	3
3.1 集群配置	3
3.2 代理管理	4
3.3 连接池	5
3.4 登录管理	5
3.5 语句请求管理	5
4 接口管理	6
4.1 添加 MPP 站点信息	6
4.2 清除 MPP 站点信息	6
4.3 设置用户名和密码信息	6
4.4 获取 MPP 配置信息	6
4.5 手动释放 DCP 连接缓存	7
5 动态视图管理	8
5.1V\$SESSIONS	8
5.2V\$SYSSTAT	8
5.3V\$DCPINSTS	8
5.4V\$DCP_CONNPOOL	9
6 典型案例	10
6.1 经典情景一：普通 MPP 环境	10
6.2 经典情景二：MPP 主备环境	14
6.3 经典情景三：多个 DCP 同时提供服务	32

1 DM 集群代理简介

DM 集群代理（DM CLUSTER PROXY，简称 DCP）是达梦数据库公司为其大规模并行处理集群 DM MPP 研发的一款具有高效、智能、安全特性的集群代理服务器产品。

DM MPP 是达梦公司研发推出的完全对等无共享式集群，支持多个 DM 节点。DM MPP 系统工作起来就像是一台单独的计算机，对外提供统一数据库服务。用户可连接 DM MPP 系统内任意一个 EP 节点进行数据操作。

DCP 是存在于应用和 DM MPP 之间的一个组件。DCP 并不是一个必要组件。但是，用户使用 DCP 后，可以大幅提高数据库的使用效率和系统的安全性。具体受益如下：

表 1.1 使用 DCP 的益处

益处	描述
提高执行效率	DCP 可根据事务中 SQL 语句所涉及的数据在 DM MPP 中不同 EP 上的分布情况，自动选择连接合适的 EP，相比随机连接 DM MPP 中某个 EP，可减少数据在 EP 间的传输，提高执行效率
提高连接效率	DCP 实现了对 DM MPP 中 EP 节点的连接池，大大减小了物理创建及释放连接的代价
提高系统安全性	用户对 DM MPP 的访问都通过 DCP 进行，不直接访问 DM MPP 中的节点，相当于增加了一层物理隔离，提高了系统的安全性

目前版本的 DCP 在使用上存在以下限制：

- 不支持通讯加密
- 不支持快速装载工具
- 语句提交后，不支持再次获取结果集
- DCP 配置仅支持一套 MPP 环境

2 体系架构

使用 DCP 时，所有数据库应用都和 DCP 代理服务器创建连接，并由 DCP 自动选择实际执行请求的集群节点。DCP 对于应用客户端来说，就相当于 DM 数据库服务器；而对于 DM MPP 各节点来说，又相当于客户端应用。在一个 DM MPP 集群上可以设置多个 DCP，各 DCP 独立运行，互不影响。

DCP 具有集群配置、代理管理、登录管理、连接池管理和语句请求管理等功能模块，其体系架构如下图所示：

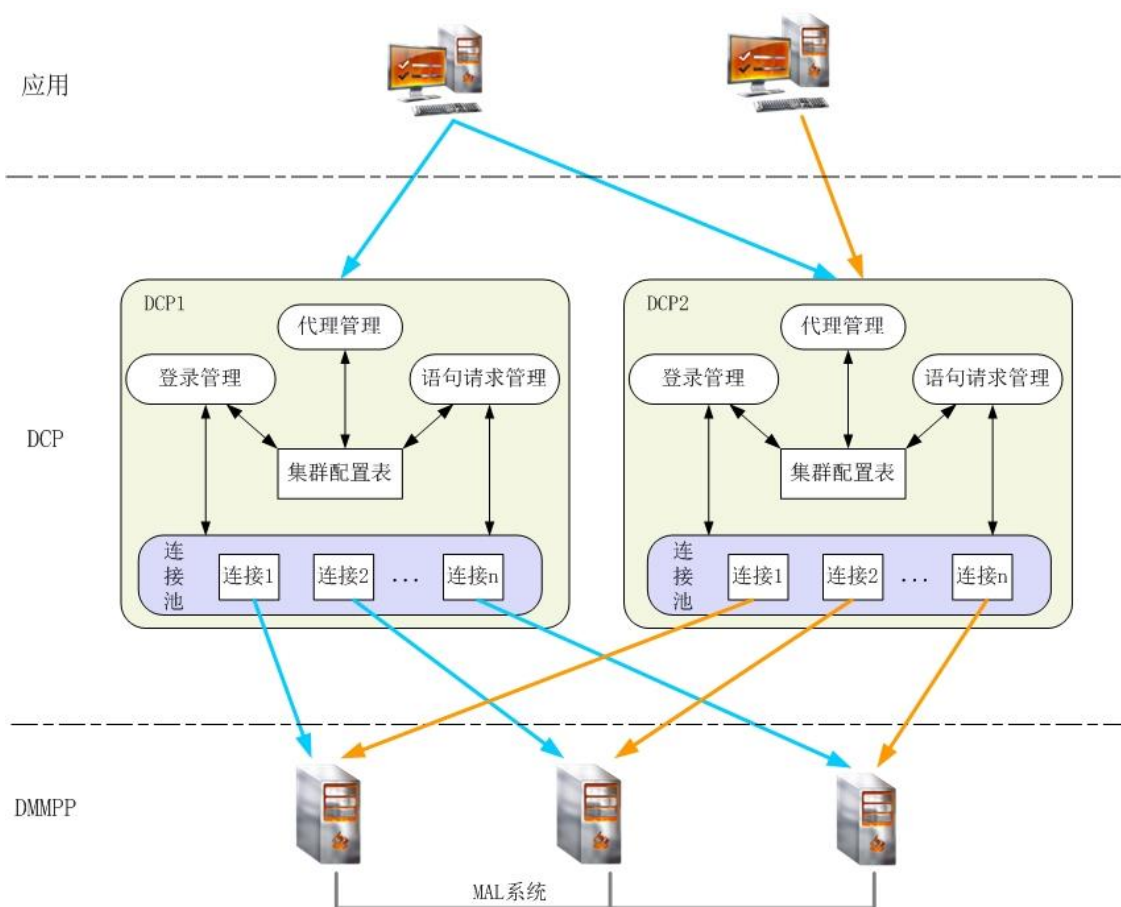


图 2.1 DCP 体系架构

3 功能模块管理

DCP 本身也是一个 DM 数据库服务器,当将其 INI 参数 `ENABLE_DCP_MODE` 置为 1 时,则服务器启动后便作为集群代理而运行。

DCP 提供两套端口,在 INI 参数中,一种是 `PORT_NUM` 用户端口,为普通客户端应用连接使用;另一种是 `DCP_PORT_NUM` 管理端口,客户端通过管理端口连接,可以在 DCP 上进行代理相关管理维护操作,如添加集群节点等。

单个 DCP 的配置过程非常简单,下文通过具体的实例详细介绍。相对于单个 DCP,配置多个 DCP 同时服务,每个 DCP 配置过程和单个 DCP 配置过程完全相同,只需要增加一个步骤,就是在 `dm_svc.conf` 中添加各 DCP 节点的服务名。

下面详细介绍 DCP 各功能模块的作用。

3.1 集群配置

DCP 在 SYS 模式下建立系统表 `SYSDCPINSTS` 和 `SYSDCPLOGIN`,用以记录集群服务器的节点配置信息。

`SYSDCPINSTS` 表登记 DCP 所对应 DM MPP 集群中一个或多个服务器的节点信息,进而通过这些节点信息来获取完整的集群节点配置及映射信息。

普通用户登录不能访问此表,只有通过代理管理连接才能访问。在管理连接中,通过调用接口 `SP_DCP_ADD_INST` 为该表添加相关信息。

表 3.1 SYSDCPINSTS

字段名	类型	说明
NAME	VARCHAR(128)	节点实例名称
IP	VARCHAR(100)	IP 地址
PORT	INT	端口号

`SYSDCPLOGIN` 表登记 DCP 登录 DM MPP 集群节点服务器获取配置信息时所需的登录用户名和口令。口令以密文形式存放,而且不可解密。

该表中只有一行数据,其访问规则与 `SYSDCPINSTS` 类似,也是在管理连接中调用接口 `SP_DCP_SET_LOGIN` 进行设置或添加。

表 3.2 SYSDCPLOGIN

字段名	类型	说明
USERNAME	VARCHAR(128)	登录用户名
PWD	VARCHAR(100)	登录口令（以密文形式存放）

3.2 代理管理

代理管理模块，通过代理管理连接来维护集群配置表的信息。

代理管理连接（PROXY MANAGEMENT CONNECT，缩写为 PMC）：管理连接是专门针对 DCP 的一种特殊连接。客户端通过管理连接在 DCP 上进行代理的相关维护操作，如添加集群节点等。代理管理连接仍然是 TCP 连接，只不过是通过特殊的管理端口建立的。

代理管理连接的建立方法非常简单，只要使用 INI 参数 DCP_PORT_NUM 指定的端口号成功登录 DCP 服务器即可。管理连接使用的登录账号就是 DCP 代理数据库的本地用户账号。

1. 添加节点

DBA 角色的用户，调用接口 SP_DCP_ADD_INST 来添加集群节点。节点信息可以在 SYSDCPINSTS 中查看。

2. 清除节点

DBA 角色的用户，调用接口 SP_DCP_CLEAR_INST 来清除系统表 SYSDCPINSTS 中的指定节点信息。

3. 设置节点登录

DBA 角色的用户，可以调用接口 SP_DCP_SET_LOGIN 来设置系统表 SYSDCPLOGIN 数据。如果表中尚无数据，则插入新行，否则修改原行数据。

4. 获取配置信息

DBA 角色用户，可以调用接口 SP_DCP_REFRESH_CONFIG，通知 DCP 登录 DM MPP 节点实例，获取最新的配置信息。该接口没有参数。该过程执行时，将重构 DCP 中的集群配置信息，并清除连接池中所有缓存连接。

5. 查询配置信息

所有使用管理连接的用户，都可以查询到系统动态视图 V\$DCPINST 的数据，以此获取集群节点配置信息。

3.3 连接池

DCP 实现了对 DM MPP 集群服务器节点的连接池，大大降低了物理创建及释放连接的代价。连接池大小由动态 INI 参数 DCP_CONN_POOL_SIZE 限制，有效值在 1 到 10000 之间。

DCP 连接池中缓存的连接，最初由用户登录时建立。内部维护 LRU 链表进行淘汰。

3.4 登录管理

DCP 接收到应用客户端登录请求时，首先在连接池中查找同名缓存连接。如果找到并匹配口令成功，身份验证通过。如果未找到同名缓存连接，则使用接收到的登录请求消息，随机选择一个 DM MPP 节点进行登录连接，如果登录成功，则身份验证通过，并将此连接加入缓存。只有身份验证通过，DCP 才创建本地会话，确认客户端登录成功。

3.5 语句请求管理

DCP 收到事务开始后的首个语句执行请求时，需要对 SQL 语句进行解析，判断该语句涉及到的所有数据是否分布在同一个 DM MPP 节点中。如果是，则建立对此节点的连接，并将语句对其发送执行。如果不是，则随机选取 DM MPP 中的某一个节点执行。

为了确保数据库的事务特性，事务的后继 SQL 请求，都仍需要在最初选择的 DM MPP 节点执行，直至事务完成。DM MPP 的特性确保了这样的请求必然能够正确执行完成，不过这样执行的效率不一定是最高的。为了确保执行效率的最大化，我们强烈建议用户慎重地构造同一事务中的相关语句。

4 接口管理

4.1 添加 MPP 站点信息

```
SP_DCP_ADD_INST(  
  
    NAME VARCHAR(128),           //节点实例名称  
  
    IP VARCHAR(64),              //实例 IP 地址  
  
    PORT INT                      //实例端口号  
);
```

添加的信息记录在 SYS.SYSDCPINSTS 表中。不要求完整添加 MPP 集群中所有服务器节点，但至少需要有一行有效值，否则 DCP 无法正常工作。

4.2 清除 MPP 站点信息

```
SP_DCP_CLEAR_INST(  
  
    NAME VARCHAR(128)           //节点实例名称  
);
```

4.3 设置用户名和密码信息

```
SP_DCP_SET_LOGIN(  
  
    USERNAME VARCHAR(128),  
  
    PWD VARCHAR(100)  
);
```

添加的信息记录在 SYS.SYSDCPLOGIN 中，密码以密文形式存放。若多次执行此过程，后设置的信息会覆盖之前设置的信息。

4.4 获取 MPP 配置信息

```
SP_DCP_REFRESH_CONFIG();
```

该接口通过 SYS.SYSDCPINSTS 和 SYS.SYSDCPLOGIN 的信息，登录 MPP 站点，获取

MPP 站点配置信息，包括实例名、EPNO、IP 地址、端口号信息。



说明： 在第一次使用 `SP_DCP_ADD_INST` 和 `SP_DCP_SET_LOGIN` 配置信息后，需要执行 `SP_DCP_REFRESH_CONFIG`，之后 DCP 才可以对外提供服务。后续若 DCP 服务器重启，如果 DM MPP 环境已处于运行状态，DCP 服务器会自动执行 `SP_DCP_REFRESH_CONFIG()`，不需要再手动调用。

4.5 手动释放 DCP 连接缓存

```
SP_DCP_CLEAR_CONN(  
  
    NUM INT                                // 清除多少个  
  
)
```

5 动态视图管理

本章介绍与 DCP 相关的动态视图。

5.1 V\$SESSIONS

动态视图 V\$SESSIONS 中的 DCP_FLAG 列显示了会话与 DCP 相关的属性：

表 5.1 V\$SESSIONS 中的增加列

列	数据类型	说明
DCP_FLAG	CHAR	1. 是否是通过 DCP_PORT 登录 DCP 服务器的会话，Y 是，N 否 2. 是否是通过 DCP 代理连接到 MPP 的会话，Y 是，N 否

当在 DCP 服务器上查询 V\$SESSIONS 时，DCP_FLAG 表示是否是通过 DCP_PORT 登录 DCP 服务器的会话。

当在 DM MPP 的 EP 节点上查询 V\$SESSIONS 是，DCP_FLAG 表示是否是通过 DCP 代理连接到 MPP 的 EP 节点的会话。

5.2 V\$SYSSTAT

动态视图 V\$SYSSTAT 中有两个与 DCP 相关的系统统计项，即两行记录：

- `trx exec count`：表示通过 DCP 执行事务的总次数。
- `ep changed count`：表示通过 DCP 执行事务，切换站点的次数。

在 DCP 服务器上查询 V\$SYSSTAT，只有通过 DCP_PORT 连接 DCP 服务器时查询 V\$SYSSTAT 这两行记录才有效。

5.3 V\$DCPINSTS

V\$DCPINSTS 动态视图展示了 DCP 对应 DM MPP 集群中所有的节点信息。该信息是 DCP 根据 SYSDCPINSTS 表中提供的节点信息，再连接 DM MPP 服务器节点请求得到的。管理连接中用户可以查询该表以了解 DM MPP 的配置情况。该视图中的数据也是在管理连接中调用接口 SP_DCP_REFRESH_CONFIG 得到的。

表 5.2 V\$DCPINSTS

列	数据类型	说明
NAME	VARCHAR(128)	节点实例名称
EPNO	INT	节点编号
IP	VARCHAR(100)	IP 地址
PORT	INT	端口号

5.4 V\$DCP_CONNPOOL

动态视图 V\$DCP_CONNPOOL 记录了 DCP 缓冲池的信息。

表 5.3 V\$DCP_CONNPOOL

列	数据类型	说明
ID	BIGINT	会话 ID
USERNAME	VARCHAR(128)	会话用户名
CUR_SCH	VARCHAR(128)	当前模式名
EPNO	INT	当前连接的 MPP 的 EPNO
EXEC_MODE	INT	当前事务执行模式，1：探测；2：强制
FREE_FLAG	CHAR	是否空闲，Y：是；N：否
RELEASE_TS	INT	上一次放回缓冲池的时间戳，仅当 FREE_FLAG=Y 时有效

6 典型案例

使用 DCP 的典型场景包括普通 MPP 环境、MPP 主备环境和多个 DCP 同时提供服务三种情况，下面分别进行介绍如何进行环境搭建。

6.1 经典情景一：普通 MPP 环境

普通 MPP 环境指不包含主备的 MPP 环境。下面为包含 2 个实例节点的 MPP 普通环境配置 DCP 代理，步骤如下：

6.1.1 节点部署规划

表 6.1 MPP 配置节点规划

实例名	机器 IP	服务器端口	MAL 端口	MPP_SEQNO
EP01	192.168.0.1	5236	5269	0
EP02	192.168.0.2	5237	5270	1

表 6.2 DCP 节点规划

实例名	机器 IP	服务器端口	DCP 端口
DCP	192.168.0.3	5238	5239

6.1.2 配置 dm.ini

首先，初始化 EP01，EP02 两个库，初始化时 dm.ini 文件里的参数要求完全一致。然后分别配置两个库中的 dm.ini。

EP01 的 dm.ini 配置如下：

```
INSTANCE_NAME = EP01
PORT_NUM      = 5236
MAL_INI       = 1
MPP_INI       = 1
```

EP02 的 dm.ini 配置如下：

```
INSTANCE_NAME = EP02

PORT_NUM      = 5237

MAL_INI       = 1

MPP_INI       = 1
```

6.1.3 配置 dmmal.ini

两个实例的 dmmal.ini 配置一样，直接拷贝即可。

```
[mal_inst1]

mal_inst_name = EP01

mal_host      = 192.168.0.1

mal_port      = 5269

mal_inst_host = 192.168.0.1

mal_inst_port = 5236


[mal_inst2]

mal_inst_name = EP02

mal_host      = 192.168.0.2

mal_port      = 5270

mal_inst_host = 192.168.0.2

mal_inst_port = 5237
```

6.1.4 配置 dmmppctl

dmmppctl 是一个二进制文件，用户不能直接配置。用户需要首先配置 dmmpp.ini，然后利用 dmctlcvt 工具进行转换得到生成 dmmppctl 文件。两个实例使用相同的 ctl 文件，只需拷贝一份。

表 6.3 dmmpp.ini 配置项

配置项	配置含义
MPP_2PC_FLAG	可选项，是否采用两阶段方式提交事务。

	取值 0, 1。0：不采用，1：采用。默认 0，一般使用默认即可。
[SERVICE_NAME]	标识每个实例的选项名
MPP_SEQ_NO	实例在 mpp 系统内的序号，取值范围为 0~1023
MPP_INST_NAME	节点实例名

dmmp.ini 配置如下：

```
[service_name1]

mpp_seq_no    = 0

mpp_inst_name = EP01


[service_name2]

mpp_seq_no    = 1

mpp_inst_name = EP02
```

利用工具转换成 dmmpctl。示例如下：

```
./dmctlcvt t2c /dm/data/DAMENG/dmmp.ini /dm/data/DAMENG/dmmpctl
```

以上步骤完成之后，只需分别启动两个实例节点，如果能够登录成功，说明配置成功了。

6.1.5 配置 DCP 服务器的 dm.ini

DCP 服务器的配置可以在初始化库的时候指定 DCP_MODE 和 DCP_PORT_NUM 参数，如：

```
./dmninit path=/dm/data/ instance_name=DCP DCP_MODE=1 DCP_PORT_NUM=5239
```

或者，也可以在初始化库完成之后修改 dm.ini 文件，效果是一样的。如：

```
ENABLE_DCP_MODE          = 1

DCP_PORT_NUM              = 5239
```

6.1.6 启动 DM MPP 和 DCP

192.168.0.1 上：

```
./dmserver /dm/data/DAMENG/dm.ini
```

192.168.0.2 上：

```
./dmserver /dm/data/DAMENG/dm.ini
```


192.168.0.3 上:

```
./dmserver /dm/data/DAMENG/dm.ini
```

6.1.7 添加 MPP 节点信息

使用 DIsql 或 Manager 工具登录 192.168.0.3: 5239 (DCP 管理端口), 执行:

```
SP_DCP_ADD_INST('MPP_EP01', '192.168.0.1', 5236);
SP_DCP_ADD_INST('MPP_EP02', '192.168.0.2', 5237);
SP_DCP_SET_LOGIN('SYSDBA', 'SYSDBA');
```

此时查询 SYS.SYSDCPINSTS 和 SYS.SYSDCPLOGIN, 可查询到对应记录;

```
SQL> select * from SYS.SYSDCPINSTS;
```

行号	NAME	IP	PORT
1	MPP_EP01	192.168.0.1	5236
2	MPP_EP02	192.168.0.2	5237

```
SQL> select * from SYS.SYSDCPLOGIN;
```

行号	USERNAME
1	SYSDBA

6.1.8 获取配置信息

获取配置信息, 先执行如下语句:

```
SP_DCP_REFRESH_CONFIG();
```

查询 V\$DCPINSTS, 可查询到 MPP 的配置信息。

```
SQL> select * from V$DCPINSTS;
```

行号	NAME	EPNO	IP	PORT
1	EP01	0	192.168.0.1	5236
2	EP02	1	192.168.0.2	5237

至此，环境配置搭建完毕。应用可直接通过 DCP 的 PORT_NUM 端口进行 DM MPP 的相关操作。

6.2 经典情景二：MPP 主备环境

6.2.1 节点部署规划

下面以一个具体的实例，说明基于主备的 MPP 系统的配置过程。配置 MPP 节点数为 2，主库实例名为 EP01、EP02，对应的备库实例名分别为 EP11、EP22。

准备 4 台机器 A、B、C、D。A 和 B 用来交叉部署主库、备库；C 用来部署监视器；D 用来搭建 DCP。其中 A 和 B 是两台刀片服务器，配置两块网卡，一块网卡接入内部网络交换模块，一块网卡接入到外部交换机，C 和 D 是两台普通 PC 接入内部网络。

机器事先都安装了 DM，安装路径为 '/dm'，执行程序保存在 '/dm/bin' 目录中，数据存放路径为 '/dm/data/EP01' 和 '/dm/data/EP02'。

表 6.4 配置环境说明

机器名	IP 地址	初始状态	操作系统
A	192.168.1.131	主库 EP01	Linux
	192.168.0.141	备库 EP22	rh6-2.6.32-220.el6.x86_64
B	192.168.1.132	主库 EP02 备库 EP11	Linux
	192.168.0.142		rh6-2.6.32-220.el6.x86_64
C	192.168.0.144	监视器	Linux rh6-2.6.32-220.el6.x86_64
D	192.168.0.207	DCP	Linux rh6-2.6.32-220.el6.x86_64

主库：

表 6.5 基于主备的配置节点规划—主库

实例名	机器 IP	端口	MAL 端口	MPP_SEQNO
EP01	192.168.0.141	5236	5337	0
EP02	192.168.0.142	5236	5337	1

备库：

表 6.6 基于主备的配置节点规划—备库

实例名	对应主库	机器 IP	端口	MAL 端口
EP11	EP01	192.168.0.142	5237	5338
EP22	EP02	192.168.0.141	5237	5338

守护进程：

守护进程 1 配置在机器 A 上，两个守护组分别守护实例 EP01 和 EP22。守护进程 2 配置在机器 B 上，两个守护组分别守护实例 EP02 和 EP11。

表 6.7 基于主备的配置节点规划—守护进程

组名	UDP 端口	实例名	所在机器
GROUPA	5243	EP01	192.168.0.141
		EP11	192.168.0.142
GROUPB	5244	EP02	192.168.0.141
		EP22	192.168.0.142

DCP 节点：

表 6.8 DCP 节点规划

实例名	机器 IP	服务器端口	DCP 端口
DCP	192.168.0.207	5238	5239

6.2.2 数据准备

配置数据守护之前，必须先同步主备库数据，确保两者保持完全一致；主数据库可以是新初始化的数据，也可以是正在生产、使用中的数据。DM 提供了两种方式初始化同步主备库数据，数据文件拷贝以及备份还原方式。

6.2.2.1 数据文件拷贝

如果搭建数据守护之前，数据库系统已经上线运行了，那么我们可以通过拷贝数据文件的方式实现主备数据库的同步。具体步骤包括：

1. 正常关闭数据库。

2. 严格按照数据文件在主库上的分布，拷贝数据文件到备库的对应目录。
3. 如果数据文件统一存放在一个目录下，则直接拷贝整个目录即可。
4. 如果目标目录不一样，可以借助 dmctlcvt 工具来修改控制文件，再修改 dm.ini 中路径。



注意：拷贝文件方式必须保证拷贝时数据库已正常关闭。

6.2.2.2 备份还原方式

用户也可以通过脱机备份、脱机还原的方式同步主备库数据，更详细的说明可以参考备份恢复相关文档。具体步骤包括：

1. 正常关闭数据库。
2. 进行脱机备份。

```
./dmbackup TYPE=FULL INI_PATH=/dm/data/EP01/DAMENG/dm.ini NAME=BACKUP_FILE
./dmbackup TYPE=FULL INI_PATH=/dm/data/EP02/DAMENG/dm.ini NAME=BACKUP_FILE
```

3. 交叉拷贝备份文件到备库。
4. 执行脱机数据库恢。

```
./dmrestore INI_PATH=/dm/data/EP01/DAMENG/dm.ini
FILE=/dm/data/DB_DAMENG_xx.bak
./dmrestore INI_PATH=/dm/data/EP02/DAMENG/dm.ini
FILE=/dm/data/DB_DAMENG_xx.bak
```

6.2.2.3 检查数据一致性

切换到主库 A 的执行码目录 /dm/bin，以 mount 方式启动数据库 EP01、EP22。

```
./dmserver /dm/data/EP01/DAMENG/dm.ini mount
./dmserver /dm/data/EP02/DAMENG/dm.ini mount
```

切换到备库 B 的执行码目录 /dm/bin，以 mount 方式启动数据库 EP02、EP11。

```
./dmserver /dm/data/EP02/DAMENG/dm.ini mount
```

```
./dmserver /dm/data/EP01/DAMENG/dm.ini mount
```

启动命令行工具 DISql，登录主数据库 EP01、EP02 查询 LSN 值。

```
SQL>select file_lsn, cur_lsn from v$rlog;
```

行号	FILE_LSN	CUR_LSN
1	10250	10250

启动命令行工具 disql，登录备数据库 EP11、EP22 查询 LSN 值。

```
SQL>select file_lsn, cur_lsn from v$rlog;
```

行号	FILE_LSN	CUR_LSN
1	10250	10250

我们可以发现主备数据库的 FILE_LSN 和 CUR_LSN 值相同，可以确保主备数据库的数据完全一致。

确保永久魔数是否一致。

```
SQL>select permanent_magic;
```

行号	PERMANENT_MAGIC
1	1385101984

正常关闭主备数据库，继续配置数据守护。

6.2.2.4 配置 dmmal.ini

配置 MAL 系统，主备数据库的 dmmal.ini 配置完全一致，MAL_HOST 使用内部网络 IP，MAL_PORT 与 dm.ini 中 PORT_NUM 使用不同的端口值，MAL_INST_PORT 要和相应实例中 dm.ini 中的 PORT_NUM 一致，配置如下：

```
MAL_CHECK_INTERVAL      = 5

MAL_CONN_FAIL_INTERVAL  = 5

[MAL_INST1]

MAL_INST_NAME           = EP01

MAL_HOST                 = 192.168.0.141
```

```
MAL_PORT                = 5337

MAL_INST_HOST            = 192.168.1.131

MAL_INST_PORT            = 5236


[MAL_INST2]

MAL_INST_NAME            = EP02

MAL_HOST                 = 192.168.0.142

MAL_PORT                 = 5337

MAL_INST_HOST            = 192.168.1.132

MAL_INST_PORT            = 5236


[MAL_INST3]

MAL_INST_NAME            = EP11

MAL_HOST                 = 192.168.0.142

MAL_PORT                 = 5338

MAL_INST_HOST            = 192.168.1.131

MAL_INST_PORT            = 5237


[MAL_INST4]

MAL_INST_NAME            = EP22

MAL_HOST                 = 192.168.0.141

MAL_PORT                 = 5338

MAL_INST_HOST            = 192.168.1.131

MAL_INST_PORT            = 5237
```

将此文件拷贝到每个实例的库目录下/dm/data/EP0x/DAMENG/下面。

6.2.2.5 配置 dmmpp.ctl

dmmpp.ctl 是二进制文件,由 dmmpp.ini 文本通过 dmctlcvt 工具转换而来,两节点的 dmmpp.ini 配置如下:

```
[service_name1]
```

```

mpp_seq_no      = 0

mpp_inst_name = EP01

[service_name2]

mpp_seq_no      = 1

mpp_inst_name = EP02

```

转换命令如下：

```
./dmctlcvrt t2c /dm/data/EP01/DAMENG/dmmpc.ini /dm/data/EP01/DAMENG/dmmpc.ctl
```

然后，将 dmmpc.ctl 拷贝到 EP02 目录下。

6.2.2.6 初始化 DCP 代理

在 DCP 所在的机器 D 上初始化代理：

```
./dmninit path=/dm/data/DCP instance_name=DCP DCP_MODE=1 DCP_PORT_NUM=5237
```

6.2.3 配置主数据库 EP01

6.2.3.1 配置 dm.ini

在 A 机器上配置主数据库实例名为 EP01，dm.ini 参数修改如下：

```

INSTANCE_NAME      = EP01

PORT_NUM           = 5236

DW_UDP_PORT        = 5243

INST_UDP_PORT      = 5253

MAIL_INI           = 1

ARCH_INI           = 1

MPP_INI            = 1

```

6.2.3.2 配置 dmarch.ini

修改 dmarch.ini，配置本地归档和 MPP 归档，MPP 归档的 ARCH_DEST 为备数据库

实例名。操作如下：

```
[ARCHIVE_MPP1]

ARCH_TYPE          = MARCH

ARCH_DEST          = EP11


[ARCHIVE_LOCAL1]

ARCH_TYPE          = LOCAL

ARCH_DEST          = /dm/data/EP01/DAMENG

ARCH_FILE_SIZE     = 128    #单位 Mb

ARCH_SPACE_LIMIT   = 0      #单位 Mb, 0 表示无限制, 范围 1024~4294967294M
```

6.2.3.3 配置 dmmwatch.ini

修改 dmmwatch.ini 配置守护进程，可以选用 TCP 或 UDP 配置方式。分别说明如下：

1. UDP 配置方式

MDW_BRO_ADDR 为广播地址，MDW_UDP_PORT 配置项与 dm.ini 的 DW_UDP_PORT 配置项保持一致，INST_UDP_PORT 配置项与 dm.ini 的 INST_UDP_PORT 配置项保持一致。

```
[GROUP_A]

MDW_BRO_ADDR       = 192.168.0.255

MDW_UDP_PORT       = 5243

MDW_INST_ERROR_TIME = 5

MDW_FAR_ERROR_TIME = 5

INST_OGUID         = 1000

INST_NAME          = EP01

INST_UDP_PORT      = 5253

INST_STARTUP_CMD   = /dm/bin/dmserver /dm/data/EP01/DAMENG/dm.ini

INST_STARTUP_MODE   = 1

INST_AUTO_RESTART   = 0


[GROUP_B]
```



```

MDW_BRO_ADDR          = 192.168.0.255
MDW_UDP_PORT          = 5244
MDW_INST_ERROR_TIME    = 5
MDW_FAR_ERROR_TIME     = 5
INST_OGUID             = 2000
INST_NAME              = EP22
INST_UDP_PORT          = 5254
INST_STARTUP_CMD        = /dm/bin/dmserver /dm/data/EP02/DAMENG/dm.ini
INST_STARTUP_MODE       = 1
INST_AUTO_RESTART       = 0

```

2. TCP 配置方式

MDW_TCP_PORT 指定 dmmppwatch 的 tcp 端口，dmmppwatch 根据 MDW_MAL_INI 和 MDW_ARCH_INI 指定的 dmmal.ini 和 dmarch.ini 获取远程 dmmppwatch 的 IP 地址，MDW_ARCH_TYPE 指定归档类型为 MPP 归档。

由于实例和守护进程之间还是以 UDP 发送，因此 TCP 方式下还需要设置 2 个 UDP 端口，无需设置广播地址。

```

[GROUP_A]

MDW_TCP_PORT          = 5263
MDW_MAL_INI           = /dm/data/EP01/DAMENG/dmmal.ini
MDW_ARCH_INI          = /dm/data/EP01/DAMENG/dmarch.ini
MDW_ARCH_TYPE         = MARCH
MDW_UDP_PORT          = 5243
MDW_INST_ERROR_TIME    = 5
MDW_FAR_ERROR_TIME     = 0
INST_OGUID             = 1000
INST_NAME              = EP01
INST_UDP_PORT          = 5253
INST_STARTUP_CMD        = /dm/bin/dmserver /dm/data/EP01/DAMENG/dm.ini
INST_STARTUP_MODE       = 1          #0: 服务方式启动; 1: 控制台方式启动

```

```

INST_AUTO_RESTART      = 0

[GROUP_B]

MDW_TCP_PORT           = 5264

MDW_MAL_INI            = /dm/data/EP02/DAMENG/dmmal.ini

MDW_ARCH_INI           = /dm/data/EP02/DAMENG/dmarch.ini

MDW_ARCH_TYPE          = MARCH

MDW_UDP_PORT           = 5244

MDW_INST_ERROR_TIME    = 5

MDW_FAR_ERROR_TIME     = 5

INST_OGUID              = 2000

INST_NAME              = EP22

INST_UDP_PORT          = 5254

INST_STARTUP_CMD        = /dm/bin/dmserver /dm/data/EP02/DAMENG/dm.ini

INST_STARTUP_MODE       = 1                #0: 服务方式启动; 1: 控制台方式启动

INST_AUTO_RESTART      = 0

```

6.2.4 配置主数据库 EP02

6.2.4.1 配置 dm.ini

在 B 机器上配置主数据库实例名为 EP02，dm.ini 参数修改如下：

```

INSTANCE_NAME          = EP02

PORT_NUM               = 5236

DW_UDP_PORT            = 5244

INST_UDP_PORT          = 5254

MAL_INI                = 1

ARCH_INI               = 1

MPP_INI               = 1

```

6.2.4.2 配置 dmarch.ini

修改 dmarch.ini，配置本地归档和 MPP 归档，MPP 归档的 ARCH_DEST 为备数据库实例名。操作如下：

```
[ARCHIVE_MPP1]

ARCH_TYPE          = MARCH

ARCH_DEST          = EP22


[ARCHIVE_LOCAL1]

ARCH_TYPE          = LOCAL

ARCH_DEST          = /dm/data/EP02/DAMENG

ARCH_FILE_SIZE     = 128    #单位 Mb

ARCH_SPACE_LIMIT   = 0      #单位 Mb，0 表示无限制，范围 1024~4294967294M
```

6.2.4.3 配置 dmmwatch.ini

修改 dmmwatch.ini 配置守护进程，可以选用 TCP 或 UDP 配置方式，分别说明如下：

1. UDP 配置方式

MDW_BRO_ADDR 可根据计算得到，MDW_UDP_PORT 配置项与 dm.ini 的 DW_UDP_PORT 配置项保持一致，INST_UDP_PORT 配置项与 dm.ini 的 INST_UDP_PORT 配置项保持一致。

```
[GROUP_A]

MDW_BRO_ADDR       = 192.168.0.255

MDW_UDP_PORT       = 5243

MDW_INST_ERROR_TIME = 5

MDW_FAR_ERROR_TIME = 5

INST_OGUID         = 1000

INST_NAME          = EP11

INST_UDP_PORT      = 5253

INST_STARTUP_CMD   = /dm/bin/dmserver /dm/data/EP01/DAMENG/dm.ini

INST_STARTUP_MODE   = 1
```

```

INST_AUTO_RESTART      = 0

[GROUP_B]

MDW_BRO_ADDR           = 192.168.0.255

MDW_UDP_PORT           = 5244

MDW_INST_ERROR_TIME    = 5

MDW_FAR_ERROR_TIME     = 5

INST_OGUID             = 2000

INST_NAME              = EP02

INST_UDP_PORT          = 5254

INST_STARTUP_CMD       = /dm/bin/dmserver /dm/data/EP02/DAMENG/dm.ini

INST_STARTUP_MODE      = 1

INST_AUTO_RESTART      = 0

```

2. TCP 配置方式

MDW_TCP_PORT 指定 dmmppwatch 的 tcp 端口, dmmppwatch 根据 MDW_MAL_INI 和 MDW_ARCH_INI 指定的 dmmal.ini 和 dmarch.ini 获取远程 dmmppwatch 的 IP 地址, MDW_ARCH_TYPE 指定归档类型为 MPP 归档。

由于实例和守护进程之间还是以 UDP 发送, 因此 TCP 方式下还需要设置 2 个 UDP 端口, 无需设置广播地址。

```

[GROUP_A]

MDW_TCP_PORT           = 5263

MDW_MAL_INI            = /dm/data/EP01/DAMENG/dmmal.ini

MDW_ARCH_INI          = /dm/data/EP01/DAMENG/dmarch.ini

MDW_ARCH_TYPE          = MARCH

MDW_UDP_PORT           = 5243

MDW_INST_ERROR_TIME    = 5

MDW_FAR_ERROR_TIME     = 0

INST_OGUID             = 1000

INST_NAME              = EP11

```

```

INST_UDP_PORT          = 5253

INST_STARTUP_CMD        = /dm/bin/dmserver /dm/data/EP01/DAMENG/dm.ini

INST_STARTUP_MODE       = 1           #0: 服务方式启动; 1: 控制台方式启动

INST_AUTO_RESTART       = 0

[GROUP_B]

MDW_TCP_PORT           = 5264

MDW_MAL_INI            = /dm/data/EP02/DAMENG/dmmal.ini

MDW_ARCH_INI           = /dm/data/EP02/DAMENG/dmarch.ini

MDW_ARCH_TYPE          = MARCH

MDW_UDP_PORT           = 5244

MDW_INST_ERROR_TIME    = 5

MDW_FAR_ERROR_TIME     = 5

INST_OGUID             = 2000

INST_NAME              = EP02

INST_UDP_PORT          = 5254

INST_STARTUP_CMD        = /dm/bin/dmserver /dm/data/EP02/DAMENG/dm.ini

INST_STARTUP_MODE       = 1           #0: 服务方式启动; 1: 控制台方式启动

INST_AUTO_RESTART       = 0

```

6.2.5 配置备数据库 EP11

dmmppwatch 可以管理同一台机器上部署的不同组的实例，EP02 和 EP11 都在机器 B 上，启动一个 dmmppwatch 即可，EP22 不需要再配置 dmmwatch.ini。

6.2.5.1 配置 dm.ini

在 B 机器上配置备数据库实例名为 EP11，dm.ini 参数修改如下：

```

INSTANCE_NAME          = EP11

PORT_NUM               = 5237

DW_UDP_PORT            = 5243

```

```

INST_UDP_PORT      = 5253

MAIL_INI           = 1

ARCH_INI           = 1

MPP_INI            = 0

```

6.2.5.2 配置 dmarch.ini

修改 dmarch.ini，配置本地归档和 MPP 归档，MPP 归档的 ARCH_DEST 为备数据库实例名。

```

[ARCHIVE_MPP1]

ARCH_TYPE          = MARCH

ARCH_DEST           = EP01


[ARCHIVE_LOCAL1]

ARCH_TYPE          = LOCAL

ARCH_DEST           = /dm/data/EP01/DAMENG

ARCH_FILE_SIZE     = 128  #单位 Mb

ARCH_SPACE_LIMIT    = 0      #单位 Mb, 0 表示无限制, 范围 1024~4294967294M

```

6.2.6 配置备数据库 EP22

dmmppwatch 可以管理同一台机器上部署的不同组的实例，EP01 和 EP22 都在机器 A 上，启动一个 dmmppwatch 即可，EP22 不需要再配置 dmmwatch.ini。

6.2.6.1 配置 dm.ini

在 A 机器上配置主数据库实例名为 EP22，dm.ini 参数修改如下：

```

INSTANCE_NAME      = EP22

PORT_NUM           = 5237

DW_UDP_PORT        = 5244

INST_UDP_PORT      = 5254

```

```

MAL_INI          = 1
ARCH_INI         = 1
MPP_INI          = 0

```

6.2.6.2 配置 dmarch.ini

修改 dmarch.ini，配置本地归档和 MPP 归档，MPP 归档的 ARCH_DEST 为备数据库实例名。

```

[ARCHIVE_MPP1]

ARCH_TYPE        = MARCH

ARCH_DEST        = EP02


[ARCHIVE_LOCAL1]

ARCH_TYPE        = LOCAL

ARCH_DEST        = /dm/data/EP02/DAMENG

ARCH_FILE_SIZE   = 128    #单位 Mb

ARCH_SPACE_LIMIT = 0      #单位 Mb，0 表示无限制，范围 1024~4294967294M

```

6.2.7 启动数据库

A 机器：以 mount 方式启动所有实例

```

./dmserver /dm/data/EP01/DAMENG/dm.ini mount
./dmserver /dm/data/EP02/DAMENG/dm.ini mount

```

B 机器：以 mount 方式启动所有实例

```

./dmserver /dm/data/EP02/DAMENG/dm.ini mount
./dmserver /dm/data/EP01/DAMENG/dm.ini mount

```



注意：

一定要以 **mount** 方式启动数据库，否则系统启动时会重构回滚表空间，生成 REDO 日志；并且，启动后应用可能连接到数据库进行操作，破坏主备数据库的一致性。数据守护配置结束后，守护进程会自动 **OPEN** 数据库。

6.2.7.1 检查数据一致性

启动命令行工具 DISql，登录各实例查询 LSN 值。

```
SQL>select file_lsn, cur_lsn from v$rlog;
```

需要确保 EP01 和 EP11 的 LSN 相等、EP02 和 EP22 的 LSN 相等，如果不一样，需要重新准备数据。

还需要确保永久魔数是否一致。

```
SQL>select permanent_magic;
```

6.2.7.2 设置 OGUID

启动命令行工具 DISql，登录数据库设置 OGUID 值：

```
SQL>sp_set_oguid(1000);
```

根据配置，设置 GROUP_A 的 2 个实例 EP01 和 EP11 的 OGUID 为 1000，GROUP_B 的 2 个实例 EP02 和 EP22 的 OGUID 为 2000。



注意：

系统通过 OGUID 值确定一个守护进程组，由用户保证 OGUID 值的唯一性，并确保数据守护系统中，数据库、守护进程和监视器配置相同的 OGUID 值。

6.2.7.3 修改数据库模式

启动命令行工具 disql，登录数据库修改数据库模式：

对 EP01 和 EP02 设置为 primary：

```
SQL>alter database primary;
```

对 EP11 和 EP22 设置为 standby：

```
SQL>alter database standby;
```

6.2.8 配置监视器

编辑 dmmmon.ini 配置监视器，监视器允许配置为 TCP 或 UDP 配置方式，可以在除主备库之外的任意一台机器上配置监视器，并且可以同时配置多个监视器。UDP 通信方式下，

监视器需要配置在内部网络。配置文件中的配置项的值要与守护进程配置文件 `dmmwatch.ini` 对应配置项保持一致。

1. UDP 配置方式

```
MON_BRO_ADDR = 192.168.0.255

MON_LOG_PATH = /dm/data/log

MON_LOG_INTERVAL = 60    #单位 s

MON_LOG_FILE_SIZE = 64    #单位 Mb

MON_LOG_SPACE_LIMIT = 0    #单位 Mb, 0 表示无空间限制

[GROUP_A]

MON_UDP_PORT = 5243

MON_INST_OGUID = 1000

[GROUP_B]

MON_UDP_PORT = 5244

MON_INST_OGUID = 2000
```

2. TCP 配置方式

`MON_MDW_IP` 对应每个守护进程所在机器 IP，也即实例所在机器 IP，该配置项可以有多个，不允许有重复值。

```
MON_LOG_PATH = /dm/data/log

MON_LOG_INTERVAL = 60    #单位 s

MON_LOG_FILE_SIZE = 64    #单位 Mb

MON_LOG_SPACE_LIMIT = 0    #单位 Mb, 0 表示无空间限制

[GROUP_A]

MON_INST_OGUID = 1000

MON_MDW_TCP_PORT = 5263    #TCP 连接端口

MON_MDW_IP = 192.168.0.141 #主库所在机器 IP

MON_MDW_IP = 192.168.0.142 #实时备库所在机器 IP
```

```
[GROUP_B]

MON_INST_OGUID           = 2000

MON_MDW_TCP_PORT         = 5264           #TCP 连接端口

MON_MDW_IP               = 192.168.0.141  #主库所在机器 IP

MON_MDW_IP               = 192.168.0.142  #实时备库所在机器 IP
```

6.2.9 启动守护进程

启动主库 A 上的守护进程。

```
./dmppwatch /dm/data/EP01/DAMENG/dmwatch.ini
```

启动备库 B 上的守护进程。

```
./dmppwatch /dm/data/EP02/DAMENG/dmwatch.ini
```

守护进程启动后，进入 STARTUP 状态，此时主备数据库都处于 MOUNT 状态。守护进程开始广播自身和其监控数据库的状态信息，结合自身信息和远程守护进程的广播信息，守护进程先将备数据库切换为 OPEN 状态，再将主数据库切换为 OPEN 状态，然后修改守护进程为 OPEN 状态。

6.2.10 启动监视器

启动主库 C 上的监视器。

```
./dmppmon /dm/data/dmmon.ini
```

至此 DM MPP 主备守护系统搭建完毕，可以提供完善的数据保护和高可用性特性。

6.2.11 启动 DCP 代理服务器

```
./dmserver /dm/data/DCP/DAMENG/dm.ini
```

6.2.12 添加主备 MPP 节点信息

使用 DIsql 或 Manager 工具登录 192.168.0.207: 5239 (DCP 管理端口)，执行：

```
SP_DCP_ADD_INST('MPP_EP01', '192.168.0.141', 5236);
```

```

SP_DCP_ADD_INST('MPP_EP02', '192.168.0.142', 5236);

SP_DCP_ADD_INST('MPP_EP11', '192.168.0.142', 5237);

SP_DCP_ADD_INST('MPP_EP22', '192.168.0.141', 5237);

SP_DCP_SET_LOGIN('SYSDBA', 'SYSDBA');

```

此处添加 MPP 节点信息时，需要将备库也添加进去。查询 V\$DCPINSTS 得到的是主库的配置信息。

当 MPP 主备环境切换主备库后，默认情况下，DCP 调度线程每过 256 秒会重新获取 MPP 配置，此时查询 V\$DCPINSTS 得到的才是最新主库的配置信息。

如果切换后，DCP 上有事务在进行，DCP 内部会收到错误信息，从而立马重新获取配置信息。或者用户可以手动执行 SP_DCP_REFRESH_CONFIG() 刷新配置信息。

此时查询 SYS.SYSDCPINSTS 和 SYS.SYSDCPLOGIN，可查询到对应记录。

```
SQL> select * from SYS.SYSDCPINSTS;
```

行号	NAME	IP	PORT
1	EP01	192.168.0.141	5236
2	EP02	192.168.0.142	5236
3	EP11	192.168.0.142	5237
4	EP22	192.168.0.141	5237

```
SQL> select * from SYS.SYSDCPLOGIN;
```

行号	USERNAME
1	SYSDBA

6.2.13 获取配置信息

获取配置信息，先执行如下语句：

```
SP_DCP_REFRESH_CONFIG();
```

查询 V\$DCPINSTS，可查询到 MPP 主备环境中主库的配置信息。

```
SQL> select * from V$DCPINSTS;
```

行号	NAME	EPNO	IP	PORT
----	------	------	----	------

```
-----  
1          EP01 0          192.168.0.141 5236  
2          EP02 1          192.168.0.142 5236
```

至此，环境配置搭建完毕。应用可直接通过 DCP 的 PORT_NUM 端口进行 DM MPP 的相关操作。

6.3 经典情景三：多个 DCP 同时提供服务

在实际生产环境中，一个 DCP 代理可能会因为一些不可抗拒因素而出现故障，或者达到自身瓶颈。所以通过搭建多个 DCP 代理（在 dm_svc.conf 中配置服务名）同时对外服务，可以有效提升 DCP 性能和提高系统稳定性。

DCP 与 DCP 之间是独立的，每一个 DCP 的搭建方式同经典场景一、经典场景二中完全相同，只需要增加一个步骤，在 dm_svc.conf 中添加各 DCP 节点的服务名。

假设有搭建好两个 DCP 代理，IP 分别为 192.168.0.207 和 192.168.0.208，端口号（DCP 代理的 PORT_NUM）均为 5236。编辑 dm_svc.conf（DM 安装时生成的一个配置文件 dm_svc.conf，在 Windows 操作平台下此文件位于 %SystemRoot%\system32 目录，在 Linux 平台下此文件位于 /etc 目录），添加服务名，语句如下：

```
dcp=(192.168.0.207:5236,192.168.0.208:5236)
```

至此，环境配置搭建完毕。客户端工具可以直接使用 DCP 服务名连接。

咨询热线：400-991-6599

技术支持：dmtech@dameng.com

官网网址：www.dameng.com



武汉达梦数据库有限公司

Wuhan Dameng Database Co.,Ltd.

地址：武汉市东湖新技术开发区高新大道999号未来科技大厦C3栋16—19层

16th-19th Floor, Future Tech Building C3, No.999 Gaoxin Road, Donghu New Tech Development Zone,Wuhan,Hubei Province,China

电话：(+86) 027-87588000 传真：(+86) 027-87588810
