

外部

**单机**

**达梦数据库安装部署文档**

**武汉达梦数据库股份有限公司**

**2022年 11月 30日**

**武汉达梦数据库股份有限公司**

目录

[1. 单机部署过程 4](#_Toc14334)

[1.1部署规划 4](#_Toc5148)

[1.2 数据库软件安装及环境配置 4](#_Toc26303)

版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **作者/修改人** | **修改内容** |
| V1.1 | 2021/10/18 | 熊锋 | 初版 |
| V1.2 | 2021/10/18 | 林建荣 | 修改 |
| V1.3 | 2021/10/19 | 陈世伟 | 修改 |
| V1.4 | 2021/10/19 | 林佳佳 | 修改 |
| V1.5 | 2021/10/19 | 刘欢 | 修改 |
| V1.6 | 2021/10/22 | 熊锋 | 修改评审内容 |
| V1.7 | 2021/10/25 | 熊锋 | 修改自动优化脚本打开sql日志，全备新增一次性调度 |
| V1.8 | 2021/10/31 | 熊锋 | 修改读写分离服务名dm\_svc.conf配置文件及权限要求 |
| V1.9 | 2021/11/01 | 熊锋 | 修改备份为：压缩级别1，增量备份为累积增量 |
| V2.0 | 2021/11/04 | 熊锋 | 修改系统参数 |
| V2.1 | 2021/11/05 | 熊锋 | 修改监视器端口错误 |
| V2.2 | 2021/11/06 | 熊锋 | 添加端口说明 |
| V2.3 | 2022/03/29 | 方志鹏 | 增加开启对慢sql日志监控 |

**实施信息**

|  |  |
| --- | --- |
| DM实施人 | 王莹娜 15875358932 |
| 检查日期 | 2022.11.30 |

## 单机部署过程

### 1.1部署规划

单机部署规划

IP规划：

|  |  |  |  |
| --- | --- | --- | --- |
| 主机名 | 服务ip | 数据库名 | 实例名 |
|  | 10.224.207.166 | dmdb | dmdb |

端口规划：

|  |  |
| --- | --- |
| 实例名 | 实例端口 |
| dmdb | 5236 |

本地存储LVM磁盘规划：（按实际需求调整磁盘空间）

|  |  |
| --- | --- |
| 磁盘 | LVM磁盘名/用途 |
| /dev/vda(3TG) | /dev/vda (安装盘) |
| /dev/vda(3T) | /dev/vda(数据盘) |

目录规划：（在对应服务器上创建目录owner为dmdba组为dinstall）

|  |  |
| --- | --- |
| 数据库软件安装目录 | /home/dmdba/dmdbms |
| 实例安装目录 | /dmdata/ |
| 归档日志存放目录 | /dmdata/arch |
| 备份文件存放目录 | /dmdata/dmbak/ |

### 1.2 数据库软件安装及环境配置

#### 1.2.1 操作系统配置

修改内核参数

|  |
| --- |
| vi /etc/sysctl.conf |

添加以下内容

|  |
| --- |
| fs.file-max = 6815744  fs.aio-max-nr = 1048576  kernel.shmmni = 4096  kernel.sem = 250 32000 100 128  net.ipv4.ip\_local\_port\_range = 9000 65500  net.core.rmem\_default = 4194304  net.core.rmem\_max = 4194304  net.core.wmem\_default = 262144  net.core.wmem\_max = 1048576  vm.dirty\_ratio = 80  vm.dirty\_expire\_centisecs = 500  vm.dirty\_writeback\_centisecs = 100  kernel.core\_pattern = /dmdata/core.%p  vm.swappiness=10  vm.dirty\_background\_ratio = 0  vm.min\_free\_kbytes = 2097152 |

使参数生效，执行

|  |
| --- |
| sysctl -p |

新建数据库管理用户和组

执行以下命令，新建用户组 dinstall。

|  |
| --- |
| groupadd dinstall -g 2001 |

执行以下命令，新建用户 dmdba。

|  |
| --- |
| useradd -G dinstall -m -d /home/dmdba -s /bin/bash -u 2001 dmdba |

执行以下命令，修改 dmdba 用户密码。

|  |
| --- |
| passwd dmdba |

Hn@dameng123

修改用户资源限制

执行以下命令，修改 dmdba 用户资源限制。

|  |
| --- |
| vim /etc/security/limits.conf |

文件末尾添加如下内容：

|  |
| --- |
| dmdba soft nice 0  dmdba hard nice 0  dmdba soft as unlimited  dmdba hard as unlimited  dmdba soft fsize unlimited  dmdba hard fsize unlimited  dmdba soft nproc 65536  dmdba hard nproc 65536  dmdba soft nofile 65536  dmdba hard nofile 65536  dmdba soft core unlimited  dmdba hard core unlimited  dmdba soft data unlimited  dmdba hard data unlimited |

需要检查/etc/security/limits.d/目录下是否有其他限制资源的配置文件，如果有也需要修改。

执行以下命令，修改 dmdba 用户环境变量。

|  |
| --- |
| vi /home/dmdba/.bash\_profile |

文件末尾添加如下内容：

|  |
| --- |
| export DM\_HOME=/home/dmdba/dmdbms export PATH=$PATH:$DM\_HOME/bin export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$DM\_HOME/bin |

使环境变量生效

|  |
| --- |
| source /home/dmdba/.bash\_profile |

#### 1.2.2数据库软件安装

1.使用root用户挂载数据库软件的iso文件

|  |
| --- |
| mkdir -p /dmsoft  mount -o loop dm8\_20220729\_FTarm\_kylin10\_64.iso /dmsoft/ |

2.使用dmdba用户安装软件

|  |
| --- |
| su - dmdba |

执行以下命令，切换到 /dmsoft目录下

|  |
| --- |
| cd /dmsoft |

执行 DMInstall.bin 文件开始安装，选择【-i】参数以命令行方式安装。

|  |
| --- |
| ./DMInstall.bin -i |

选择安装程序的语言 c/C 为中文，e/E 为英文。

提示是否安装 key 文件，输入 N 跳过。

选择时区，21 即东 8 区。

选择安装类型，默认典型安装（包含所有内容）。

选择软件安装目录，默认/home/dmdba/dmdbms

确认安装

安装完成提示使用root执行脚本：

|  |
| --- |
| /home/dmdba/dmdbms/script/root/root\_installer.sh |

#### 1.3使用 dminit 工具初始化实例

执行以下命令

|  |
| --- |
| dminit path=/dmdate page\_size=32 extent\_size=32 charset=1 case\_sensitive=0 log\_size=2048 db\_name=dmdb instance\_name=dmdb SYSDBA\_PWD="Hn@dameng123" SYSAUDITOR\_PWD="Hn@dameng123" SYSSSO\_PWD="Hn@dameng123" |

注意:

初始化参数中除了 path 参数必须指定，其它参数都有默认值，如果需求与默认值不同，初始化的时候请指定需要的值。因为部分参数初始化后是无法修改的例如：page\_size（页大小），charset（字符集），case\_sensitive（大小写敏感）等。更多参数./dminit help 查看，是否无法修改的参数可以查询 v$dm\_ini 视图，para\_type=’READ ONLY’ 表示无法修改。

2.key与版本一定要匹配，key也是区分企业版和安全版

#### 1.4 注册服务并启动实例

以系统服务方式启动实例，DM 提供脚本将数据库实例注册为操作系统服务。

root 用户下切换到 /home/dmdba/dmdbms/script/root/：

|  |
| --- |
| cd /home/dmdba/dmdbms/script/root/ |

执行以下命令，执行脚本注册服务：

|  |
| --- |
| ./dm\_service\_installer.sh -t dmserver -p DM -dm\_ini /dmdata/dmdb/dm.ini |

执行以下命令，以服务方式启动实例：

|  |
| --- |
| systemctl start DmServiceDM |

#### 1.5 登录数据库

dmdba用户下执行以下命令：

|  |
| --- |
| disql SYSDBA/'"Hn@dameng123"' |

说明：密码使用单引号+双引号包围起来

#### 1.6 归档配置

生产环境必须开启归档日志，且必须限制归档日志保留量，限制方法：

1. 设置归档空间大小限制即指定 SPACE\_LIMIT 参数（单位是 MB）。
2. 定期删除归档日志（设置定时作业）。

例如开启归档并限制归档空间为 300 G（按实际存储空间给合适的值），如下所示

|  |
| --- |
| alter database mount; alter database add archivelog 'dest=/dmdata/arch ,TYPE=local,FILE\_SIZE=1024,SPACE\_LIMIT=20000'; alter database archivelog; alter database open; |

#### 1.7 定制备份策略

根据应用需求，定制备份策略如下所示

| 备份类型 | 备份周期 | 备份时间 |
| --- | --- | --- |
| 全量备份 | 每周 | 每周六 23 点 |
| 增量备份 | 每天 | 除周六外每天 23 点 |
| 删除备份 | 每天 | 每天 23 点 30 |

使用dmdba用户创建备份目录

|  |
| --- |
| mkdir -p /dmdata/dmbak |

登录主数据库

|  |
| --- |
| disql SYSDBA/'"Hn@dameng123"' |

创建作业系统表

|  |
| --- |
| SP\_INIT\_JOB\_SYS(1); |

全量备份（每周六 23 点全备）：其中有1分钟后的一次性全备调度，执行完成后检查备份是否成功。

|  |
| --- |
| call SP\_CREATE\_JOB('bakfull',1,0,'',0,0,'',0,'');  call SP\_JOB\_CONFIG\_START('bakfull');  call SP\_ADD\_JOB\_STEP('bakfull', 'bak01', 6, '00000000/dmdata/dmbak', 0, 0, 0, 0, NULL, 0);  call SP\_ADD\_JOB\_SCHEDULE('bakfull', 'std1', 1, 2, 1, 64, 0, '23:00:00', NULL, '2022-11-30 11:44:00', NULL, '');  call SP\_ADD\_JOB\_SCHEDULE('bakfull', 'once1', 1, 0, 0, 0, 0, NULL, NULL, sysdate+1/1440, NULL, '');  call SP\_JOB\_CONFIG\_COMMIT('bakfull'); |

增量备份（每周除周六外每天 23 点增量备份）：

|  |
| --- |
| call SP\_CREATE\_JOB('bakincr',1,0,'',0,0,'',0,'');  call SP\_JOB\_CONFIG\_START('bakincr');  call SP\_ADD\_JOB\_STEP('bakincr', 'bak2', 6, '40000000/dmdata/dmbak|/dmdata/dmbak', 0, 0, 0, 0, NULL, 0);  call SP\_ADD\_JOB\_SCHEDULE('bakincr', 'std2', 1, 2, 1, 63, 0, '23:00:00', NULL, '2022-11-02 11:12:00', NULL, '');  call SP\_JOB\_CONFIG\_COMMIT('bakincr'); |

备份定期删除（每天 23：30 删除 14 天前备份）：

|  |
| --- |
| call SP\_CREATE\_JOB('delbak',1,0,'',0,0,'',0,'');  call SP\_JOB\_CONFIG\_START('delbak');  call SP\_ADD\_JOB\_STEP('delbak','bak1',0, 'SF\_BAKSET\_BACKUP\_DIR\_ADD(''DISK'',''/dmdata/dmbak'');  call sp\_db\_bakset\_remove\_batch(''DISK'',now()-14);', 1, 2, 0, 0, NULL, 0);  call SP\_ADD\_JOB\_SCHEDULE('delbak', 'del01', 1, 1, 1, 0, 0, '23:30:00', NULL, '2022-11-30 11:44:00', NULL, '');  call SP\_JOB\_CONFIG\_COMMIT('delbak'); |

添加自动收集统计信息的任务（每周六1点收集全库统计信息）

|  |
| --- |
| call SP\_CREATE\_JOB('statistics',1,0,'',0,0,'',0,'');  call SP\_JOB\_CONFIG\_START('statistics');  call SP\_ADD\_JOB\_STEP('statistics', 'statistics1', 0, 'begin  for rs in (select ''sf\_set\_SESSION\_para\_value(''''HAGR\_HASH\_SIZE'''',(select cast(  case when max(table\_rowcount(owner,table\_name))<=(select max\_value from v$dm\_ini  where para\_Name=''''HAGR\_HASH\_SIZE'''') and max(table\_rowcount(owner,table\_name))>=(  select min\_value from v$dm\_ini where para\_Name=''''HAGR\_HASH\_SIZE'''') then  max(table\_rowcount(owner,table\_name)) when max(table\_rowcount(owner,table\_name))<(  select min\_value from v$dm\_ini where para\_Name=''''HAGR\_HASH\_SIZE'''') then  (select min\_value from v$dm\_ini where para\_Name=''''HAGR\_HASH\_SIZE'''') else  (select max\_value from v$dm\_ini where para\_Name=''''HAGR\_HASH\_SIZE'''') end as bigint)  from dba\_tables where owner=''''''||NAME||''''''));''  sql1,''DBMS\_STATS.GATHER\_SCHEMA\_STATS(''''''||NAME||'''''',100,TRUE,''''FOR ALL COLUMNS SIZE AUTO'''');''  sql2  from SYS.SYSOBJECTS where TYPE$=''SCH'' ) loop  execute immediate rs.sql1;  execute immediate rs.sql2;  end loop;  end;', 0, 0, 0, 0, NULL, 0);  call SP\_ADD\_JOB\_SCHEDULE('statistics', 'statistics1', 1, 2, 1, 64, 0, '01:00:00', NULL, '2022-11-30 11:44:00', NULL, '');  call SP\_JOB\_CONFIG\_COMMIT('statistics'); |

注意：上线前，必须提前48小时，完成数据迁移和统计信息收集。

检查作业

SELECT \* FROM sysjob.sysjobs;

删除作业

**call** SP\_DROP\_JOB('jobname');

如有数据迁移，需要收集一下统计信息

select 'DBMS\_STATS.GATHER\_SCHEMA\_STATS('''||username||''',100,TRUE,''FOR ALL COLUMNS SIZE AUTO'');' from all\_users;

查看各表的记录

select t.table\_name,t.num\_rows from user\_tables t

#### 1.8数据库参数优化

登录数据库执行以下脚本，脚本会更新，清获取最新脚本



如果服务器上需要创建多个实例，请调整每个实例占用物理内存的百分比，再执行脚本。

注意，执行完成需要重启数据库生效。

参数检查

|  |
| --- |
| **select**  PARA\_NAME,  PARA\_VALUE  **from**  v$dm\_ini  **where**  para\_name **in**('FAST\_POOL\_PAGES', 'ENABLE\_FREQROOTS', 'BUFFER', 'RECYCLE', 'MAX\_BUFFER', 'MAX\_SESSION', 'MAX\_SESSION\_STATEMENT', 'USE\_PLN\_POOL', 'OLAP\_FLAG', 'OPTIMIZER\_MODE', 'VIEW\_PULLUP\_FLAG', 'MAX\_OS\_MEMORY', 'BUFFER\_POOLS', 'RECYCLE\_POOLS', 'HJ\_BUF\_GLOBAL\_SIZE', 'HJ\_BUF\_SIZE', 'DICT\_BUF\_SIZE', 'TEMP\_SIZE', 'VM\_POOL\_SIZE', 'SESS\_POOL\_SIZE', 'CACHE\_POOL\_SIZE', 'MEMORY\_TARGET', 'VM\_POOL\_TARGET', 'SESS\_POOL\_TARGET', 'RT\_HEAP\_TARGET', 'ADAPTIVE\_NPLN\_FLAG', 'PARALLEL\_PURGE\_FLAG', 'PARALLEL\_POLICY', 'UNDO\_EXTENT\_NUM', 'ENABLE\_INJECT\_HINT'); |

检查字符集、大小写敏感、页大小，VARCHAR长度是否以字符为单位

|  |
| --- |
| SELECT ID\_CODE,PERMANENT\_MAGIC,SF\_GET\_UNICODE\_FLAG() AS CHARSET,SF\_GET\_CASE\_SENSITIVE\_FLAG() AS CASE\_SENSITIVE,SF\_GET\_PAGE\_SIZE() AS PAGE\_SIZE ,SF\_GET\_LENGTH\_IN\_CHAR() as LENGTH\_IN\_CHAR; |

ssl登录失败

sp\_set\_para\_value(2,'ENABLE\_ENCRYPT',0);

兼容模式

sp\_set\_para\_value(2,'COMPATIBLE\_MODE',0);

是否兼容其他数据库模式。0：不兼容，1：兼容 SQL92 标准，2：部分兼容 ORACLE，3：部分兼容 MS SQL SERVER，4：部分兼容 MYSQL，5：兼容 DM6，6：部分兼容 TERADATA

#### 1.9 配置sql日志

修改sqllog.ini到实例路径下

vi /dmdata/dmdb/sqllog.ini（记录1.5s以上sql）

|  |
| --- |
| BUF\_TOTAL\_SIZE = 10240  BUF\_SIZE = 1024  BUF\_KEEP\_CNT = 6  [SLOG\_ALL]  FILE\_PATH = ../log  PART\_STOR = 1  SWITCH\_MODE = 2  SWITCH\_LIMIT = 512  ASYNC\_FLUSH = 1  FILE\_NUM = 5  ITEMS = 0  SQL\_TRACE\_MASK = 2:3:25:28  MIN\_EXEC\_TIME = 1500  USER\_MODE = 0  USERS = |

开启慢日志

|  |
| --- |
| SP\_SET\_PARA\_VALUE(1,'SVR\_LOG',1); |

如果对 sqllog.ini 进行了修改，可通过调用以下函数即时生效，无需重启数据库

|  |
| --- |
| SP\_REFRESH\_SVR\_LOG\_CONFIG(); |

#### 1.10迁移信息统计查询

如有数据迁移，需要收集一下统计信息，收集指定用户下所有表所有列的统计信息：

|  |
| --- |
| select 'DBMS\_STATS.GATHER\_SCHEMA\_STATS('''||username||''',100,TRUE,''FOR ALL COLUMNS SIZE AUTO'');' from all\_users; |

迁移信息统计

|  |
| --- |
| --根据指定用户统计用户下的各对象类型和数目  select object\_type,count(\*) from all\_objects where owner='SYSDBA' group by object\_type;  --统计指定用户下所有的对象，并记录到新的记录表中  create table dm\_objects(obj\_owner varchar(100),obj\_name varchar(100),obj\_type varchar(50));  insert into dm\_objects select owner,object\_name,object\_type from all\_objects where owner='SYSDBA';  --统计所有表及每个表的数据量到表数据记录表  create table dm\_tables(tab\_owner varchar(100),tab\_name varchar(100),tab\_count int);  declare  begin  for rec in (select owner,object\_name from all\_objects where owner='SYSDBA'--用户名  and object\_type='TABLE') loop  execute immediate 'insert into dm\_tables select '''|| rec.owner ||''','''||  rec.object\_name ||''',count(\*) from '|| rec.owner || '.' || rec.object\_name;  end loop;  end;  select \* from dm\_tables;  --统计所有视图及视图的数据量到表数据记录表  create table dm\_view(tab\_owner varchar(100),tab\_name varchar(100),tab\_count int);  declare  begin  for rec in (select owner,object\_name from all\_objects where owner='SYSDBA'--用户名  and object\_type='VIEW') loop  execute immediate 'insert into dm\_view select '''|| rec.owner ||''','''||  rec.object\_name ||''',count(\*) from '|| rec.owner || '.' || rec.object\_name;  end loop;  end;  select \* from dm\_view;  --统计用户下的全部存储过程  select owner,object\_name from all\_objects where owner='SYSDBA'--用户名  and object\_type='PROCEDURE'  --统计索引  CREATE TABLE SYSDBA.XPDI AS  select  TABLE\_OWNER TAO,  TABLE\_NAME TAB ,  INDEX\_NAME IND ,  LISTAGG('"'||COLUMN\_NAME||'"', ',')WITHIN GROUP(  ORDER BY  COLUMN\_POSITION) CCF  from  dba\_ind\_columns  where  TABLE\_OWNER NOT LIKE '%SYS%'  GROUP BY  TABLE\_OWNER,  TABLE\_NAME ,  INDEX\_NAME    --序列  select owner,object\_name,OBJECT\_TYPE from all\_objects where OBJECT\_TYPE='SEQUENCE' |