

金蝶云·苍穹&星瀚 达梦数据库 上线部署说明书

<版本:3.0>



版权声明

本文档属金蝶软件(中国)有限公司内部文档,金蝶软件(中国)有限公司保留本文档所有权利。本文档仅限于金蝶软件(中国)有限公司及云开发平台部内部使用,未经许可,不得以任何方式做项目之外用途或对外披露。版权所有,违者必究。

更新历史

日期	版本	描述	作者
2024年05月24日	<1.0>	初稿,首次发布第1-6章	冷波
2024年11月13日	<2.0>	1) 增加苍穹版本 v7.0.1 ; 2) 增加分库标识 opmc、删除 de 分库 ; 3) 增加参数 OPTIMIZER_OR_NBEXP, 解决 OR 查询问题	冷波
2024年04月09日	<3.0>	1) 回收 mc 库 dba 权限 2) 回收苍穹库 dba 权限	冷波

目录

1	达梦数据库与苍穹适配环境说明.....	5
2	数据库服务部署.....	6
2.1	资源规划.....	6
2.2	安装规划.....	6
2.3	安装准备.....	6
2.3.1	内核参数调整.....	6
2.3.2	limit 调整.....	7
2.3.3	目录创建.....	8
2.3.4	介质挂载.....	8
2.4	安装过程.....	8
2.5	DB 参数调整.....	10
3	应用数据初始化.....	13
3.1	创建苍穹和 MC 库.....	13
3.1.1	分库合并规则.....	13
3.1.2	创建表空间.....	14
3.1.3	创建用户.....	15
3.2	恢复苍穹和 MC 模板库.....	17
3.2.1	恢复 MC 库.....	17
3.2.2	恢复苍穹库.....	18
3.3	MC 初始化.....	23
3.3.1	访问地址设置.....	23
3.3.2	数据库配置.....	23
3.3.3	MQ 组件配置.....	24
3.3.4	Redis 组件配置.....	24
3.3.5	ES 组件配置.....	25
3.3.6	初始化确认.....	26
3.4	MC 更新配置信息.....	26
3.5	创建数据中心.....	26
4	数据库日常运维.....	28
4.1	字符集与排序方式.....	28

4.2	参数修改.....	28
4.3	数据库启停.....	28
4.4	备份与恢复.....	29
4.4.1	物理备份与恢复.....	29
4.4.2	逻辑备份恢复.....	30
4.5	数据字典说明.....	31
4.6	用户与最小权限.....	31
4.7	存储过程文件执行.....	32
4.8	慢查询日志维护.....	33
5	数据库调优.....	35
5.1	查看查询计划.....	35
5.2	统计信息收集.....	35
5.3	会话相关操作.....	37
5.4	查看当前会话执行的 SQL.....	38
5.5	查看存储工程的调试输出.....	38
6	应用参数说明.....	39
6.1	JDBC 驱动程序说明.....	39
6.2	MC 相关参数.....	39

1 达梦数据库与苍穹适配环境说明

苍穹发行版本	金蝶云苍穹 V5.0.1 及以上
星瀚发行版本	金蝶云星瀚 V5.0.1 及以上
苍穹发行日期	2024 年 10 月 28 日，以苍穹 v7.0.1 为准
DB 名称与版本	达梦数据库 DM Database Server 64 V8 (1-2-128-22.06.30-163817-20005-ENT) 及以上
数据库字符集	UTF8
DB 发行日期	2021-11-11
DB 兼容模式	兼容 Oracle 模式 (COMPATIBLE_MODE=2)
OS 与版本	CentOS 7.X 或达梦兼容的操作系统
CPU 架构	X86_64/ARM
JDBC 驱动与版本	DmJdbcDriver-18.jar(8.1.2.84)
DB 厂商	武汉达梦数据库股份有限公司
DB 官网	https://www.dameng.com/
DB 技术文档	https://eco.dameng.com/
备注	<p>安装 DM 数据库必须按照以下安装参数初始化数据库实例</p> <p>PAGE_SIZE=32 数据库块大小 32k</p> <p>EXTENT_SIZE=32 空间分区单元 EXTENT 为 32 个块</p> <p>CASE_SENSITIVE=1 设置为 1 大小写敏感，否则大小写不敏感</p> <p>CHARSET=1 设置 1 字符集 UTF8，为 0 则 GB18030</p> <p>COMPATIBLE_MODE=2 设置 2 兼容 ORACLE 模式</p> <p>LENGTH_IN_CHAR=0 设置 0 以 byte 为单位;默认 byte 为单位</p> <p>PK_WITH_CLUSTER=0 设置 0 创建主键不自动转为聚集主键</p> <p>CHECK_CONS_NAME=0 设置 0 约束不检查重名</p>

2 数据库服务部署

数据库服务生产环境由武汉达梦数据库股份有限公司提供支持，金蝶方不再参与部署过程。以下安装过程仅供测试、开发环境部署达梦数据库使用。

2.1 资源规划

部署项目	部署建议	备注
机器数量	2	主从部署
机器规格	参考金蝶云苍穹标准部署方案，或者由甲方与达梦厂商自行规划。	
存储需求	存储类型：推荐本地 SSD, RAID5 或者 RAID10。 网络：带宽>=2.0Gbps, RTT<=0.5ms 生产环境建议磁盘空间至少 1024G，测试环境建议磁盘空间至少 512G。	

2.2 安装规划

使用用户 dmdba 安装部署，程序文件、数据目录位于本机的高速存储磁盘中。可参考下述过程进行安装预备。

1.创建用户所在的组，命令如下：

```
#groupadd dinstall
```

2.创建用户，命令如下：

```
#useradd -g dinstall -m -d /home/dmdba -s /bin/bash dmdba
```

3.修改用户密码，命令如下：

```
#passwd dmdba
```

2.3 安装准备

2.3.1 内核参数调整

内核参数主要涉及文件描述符、SWAP、连接等参数调整。参数调整过程如下：

1) 修改参数

```
#vi /etc/sysctl.conf
```

添加或者修改如下内容

```
vm.min_free_kbytes = 65536
```

```
vm.swappiness=0
```

```
vm.max_map_count = 2000000 #建议大于 2000000
```

```
fs.file-max=1048576
net.ipv4.tcp_rmem = 4096 262144 16777216
net.ipv4.tcp_wmem = 4096 262144 16777216
net.ipv4.ip_local_port_range = 32768 60999
net.ipv4.tcp_max_syn_backlog=4096
net.core.netdev_max_backlog = 4096
net.core.somaxconn = 4096
```

2) 使参数生效

```
#sysctl -p
```

```
[root@db65 ~]# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
kernel.sysrq = 0
kernel.core_uses_pid = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
vm.swappiness = 0
vm.min_free_kbytes = 65536
net.core.rmem_default = 87383
net.core.rmem_max = 16777216
net.core.wmem_default = 87383
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 262144 16777216
net.ipv4.tcp_wmem = 4096 262144 16777216
net.core.netdev_max_backlog = 2000
```

2.3.2 limit 调整

达梦数据库使用命令启动时，主要依赖于进程的 limit.conf 配置，增加或者修改如下内容，进程数、文件描述符数至少大于如下设置：

1) vi /etc/security/limits.conf

```
dmdba hard nofile 65536
dmdba soft nofile 65536
dmdba hard stack 32768
dmdba soft stack 16384
```

```
dmdba hard nofile 65536
dmdba soft nofile 65536
dmdba hard stack 32768
dmdba soft stack 16384
```

2) 验证 limit 设置成功

```
#su - dmdba
```

```
#ulimit -a
```

```
[root@localhost ~]# su - dmdba
上一次登录: 二 10月 13 14:57:10 CST 2020pts/1 上
[dmdba@localhost ~]$ ulimit -a
core file size          (blocks, -c) 0
data seg size          (kbytes, -d) unlimited
scheduling priority     (-e) 0
file size              (blocks, -f) unlimited
pending signals        (-i) 3780
max locked memory      (kbytes, -l) 64
max memory size        (kbytes, -m) unlimited
open files             (-n) 65536
pipe size              (512 bytes, -p) 8
POSIX message queues   (bytes, -q) 819200
real-time priority     (-r) 0
stack size             (kbytes, -s) 16384
cpu time               (seconds, -t) unlimited
max user processes     (-u) 3780
virtual memory         (kbytes, -v) unlimited
file locks             (-x) unlimited
[dmdba@localhost ~]$
```

2.3.3 目录创建

创建数据库软件安装目录并授权:

```
#mkdir /dm8
#chown dmdba:dinstall -R /dm8/
#chmod -R 755 /dm8
```

2.3.4 介质挂载

- 1) 使用 root 用户将 DM 数据库的 iso 安装包保存在任意位置, 例如 /opt 目录下
- 2) 执行如下命令挂载镜像:

```
mount -o loop /opt/dm8_setup_rh7_64_ent_8.1.1.45_20191121.iso /mnt
```

```
[root@localhost /]# mount -o loop /opt/dm8_setup_rh7_64_ent_8.1.1.87_20200601.iso /mnt
mount: /dev/loop0 写保护, 将以只读方式挂载
[root@localhost /]#
```

2.4 安装过程

测试环境(主机资源:32C64G)以命令行模式安装单节点达梦数据库为例(图形化安装参考官方文档), 生产环境或者主从部署请咨询达梦官方提供技术支持服务。以下操作由 dmdba 用户执行:

- 1) 进入安装程序目录

```
$cd /mnt
```

- 2) 执行 **dmunit** 初始化实例:

```
dmunit \
```

```
DB_NAME=dmdb \
```

```
INSTANCE_NAME=dmdb \
```

```
COMPATIBLE_MODE=2
PAGE_SIZE=32 \
EXTENT_SIZE=32 \
CASE_SENSITIVE=1 \
CHARSET=1 \
LENGTH_IN_CHAR =0 \
SYSDBA_PWD="Hn@dameng123" \
PATH=/dm8/data/
```

参数说明一：

- DB_NAME=dmdb #数据库名称
- INSTANCE_NAME=dmdb #实例名称
- COMPATIBLE_MODE=2 #设置 ORACLE 兼容模式
- SYSDBA_PWD="XXXX" #设置 SYSDBA 密码
- PATH=/dm8/data/ #设置数据存放路径

参数说明二：

- PAGE_SIZE=32 #数据库块大小 32k
- EXTENT_SIZE=32 #空间分配单元 EXTENT 为 32 个块
- CASE_SENSITIVE=1 #设置为 1 大小写敏感，否则大小写不敏感
- CHARSET=1 #设置 1 字符集 UTF8，默认 0 为 GB18030
- LENGTH_IN_CHAR =0 #设置 varchar 长度单位,0 以 byte 为单位

注意事项：

- ✓ 参数说明二列举的 5 参数在实例创建后无法更改，在创建实例前需根据客户需求合理规划好；
- ✓ SYSDBA 密码根据客户实际要求进行设置。

3) 重启数据库参数生效

```
systemctl restart DmServiceDMSEVER.service
```

4) 登录数据库并查询版本

根据初始化实例的 sysdba 及密码登录数据库，执行以下 SQL 进行查询：

disql 用户/密码@ip:port

```
SQL>select *,id_code from v$version;;
```

出现正确版本信息则说明安装正确。

```
SQL> select *,id_code from v$version;

LINEID      BANNER                                id_code
-----
1           DM Database Server 64 V8             1-2-94-21.11.11-150650-10038-SEC
2           DB Version: 0x7000c                  1-2-94-21.11.11-150650-10038-SEC

used time: 4.247(ms). Execute id is 7700.
```

2.5 DB 参数调整

以 32C64G 为例，相关参数建议如下：

1. 数据库参数

参数名	值	说明
COMPATIBLE_MODE	2	达梦数据库兼容模式： 0:none, 1:SQL92, 2:Oracle, 3:MS SQL Server, 4:MySQL, 5:DM6, 6:Teradata
PK_WITH_CLUSTER	0	PK_WITH_CLUSTER 默认值为 1，即建表时指定的主键不会自动转化为聚集主键；若设置 1，则主键自动变为聚集主键。
LENGTH_IN_CHAR	0	设置 VARCHAR 单位，0 表是以 byte 为单位，默认设置
CHECK_CONS_NAME	0	创建/重命名约束是否需要检查约束名，0 表示不检查(解决名称冲突问题)，默认值 1
OPTIMIZER_OR_NBEXP	0	OR 表达式的优化方式，0：不优化；1：生成 UNION_FOR_OR 操作符时，优化为无 KEY 比较方式；2：OR 表达式优先考虑整体处理方式；
MAX_SESSIONS	5000	数据库最大并发会话数
CLOB_LIKE_MAX_LEN	20480	CLOB 转 VARCHAR 查询使用 Like 最大长度
MAX_OS_MEMORY	85	数据库服务器能使用的最大内存占操作系统物理内存与虚拟内存总和的百分比，取值范围 40~100
MEMORY_POOL	4096	共享内存池大小，以 M 为单位，推荐物理内存*6%取整
MEMORY_TARGET	12000	共享内存池在扩充到此大小以上后，空闲时收缩回此指定大小，推荐 MEMORY_POOL*2~3 倍，单位 M
BUFFER	40000	系统缓冲区大小，以 M 为单位。推荐值：系统缓冲区大小为可用物理内存的 60%~80%
SESS_POOL_SIZE	128	会话缓冲区大小，单位 KB，取值范围 16~1024*1024。若所申请的内存超过实际能申请的大小，则系统将按 128KB 大小重新申请

参数名	值	说明
SORT_BUF_SIZE	2048	排序机制下，排序缓存区最大值，以 M 为单位；32G 设置 1024M
DICTIONARY_BUF_SIZE	1024	字典缓冲区大小，以 M 为单位
RECYCLE	2048	RECYCLE 缓冲区大小，以 M 为单位。高并发或大量使用 with、临时表、排序等
MAX_PARALLEL_DEGREE	64	单个查询默认的最大并行任务个数,推荐 64
PARALLEL_THRD_NUM	16	并行任务最大线程数，默认值 10
PARALLEL_POLICY	2	设置为 2 表示手动控制并行度；0 表示禁用并行，1 表示自动运行并行
ENABLE_HASH_JOIN	1	启动 HASH JOIN 特性
HJ_BUF_SIZE	500	单个 HASH 连接操作符的数据总缓存大小，以 M 为单位。在 OLTP 环境中，建议采用默认值。在 OLAP 环境下，可以根据参与 HASH JOIN 的数据量调大。
HJ_BUF_GLOBAL_SIZE	5000	HASH 连接操作符的数据总缓存大小 (\geq HJ_BUF_SIZE)，系统级参数，以 M 为单位。
CACHE_POOL_SIZE	2000	SQL 缓冲池大小，以 M 为单位。有效值范围：32 位平台下为 (1~2048)；64 位平台下为 (1~67108864)。单位：MB 推荐 2000
MAX_SESSION_STATEMENT	2000	单个会话上允许同时打开的语句句柄最大数，有效值范围 (64~20480)，推荐值 2000
VM_POOL_SIZE	256	系统执行时虚拟机内存池大小，在执行过程中用到的内存大部分是从这里申请的，它的空间是从操作系统中直接申请的，有效值范围 (32~1024*1024)，推荐值 256

2.归档日志参数

参数名	值	说明
ARCH_TYPE	LOCAL	Redo 日志归档类型，LOCAL/REMOTE/REALTIME/TIMELY/ASYNC/SYNC/(RAFT/LEARNER)，分别表示本地归档/远程归档/实时归档/即时归档/异步归档/同步归档/RAFT 归档（RAFT 归档仅用于 DPC 环境下）
ARCH_DEST	/path/to/dm_arch/arch	归档目标，本地归档为归档文件存放路径，其他归档方式设置为目标数据库实例名
ARCH_FILE_SIZE	1024	单个 Redo 日志归档文件大小，单位 MB，取值范围 64~2048，仅对本地归档有效，缺省为 1024MB，即 1G
ARCH_SPACE_LIMIT	2048	Redo 日志归档空间限制，当节点上的本地归档文件达到限制值时，系统自动删除最早生成的归档日志文件。取值范围 0 或 1024~2147483647，单位 MB，仅对本地归档有效，缺省为 0，表示无空间限制

3 应用数据初始化

达梦数据库与 Oracle 类似。一个达梦数据主机可以有多个实例，每个实例可以创建多个用户；一个用户所拥有的对象集合称为模式，模式是数据库对象的集合，即一个苍穹业务数据集合。在进行应用数据初始化时，按照以下步骤执行：

3.1 创建苍穹和 MC 库

3.1.1 分库合并规则

在模板库中,存在 43 个库文件,建议合并为 7 个分库进行生产部署,其合并规则如下：

序号	7 个分库	7.0.1 模板库 (43 个)	分库标识	分库标识名称
1	元数据库(meta)	BIZ_TPL_COMM_META	sys.meta	元数据库
2	系统库(sys)	BIZ_TPL_COMM_SYS	sys	系统库
		BIZ_TPL_COMM_WFS	wfs	流程服务库
		BIZ_TPL_COMM_LOG	log	日志库
		BIZ_TPL_COMM_AIDB	aidb	AI 服务云
		BIZ_TPL_COMM_MBR	mbr	会员中台库
		BIZ_TPL_COMM_EIP	eip	集成服务库
		BIZ_TPL_COMM_BARC	sys.barc	条码服务库
		BIZ_TPL_COMM_RPAC	rpac	RPA 服务库
		BIZ_TPL_COMM_IMSC	imsc	实施服务库
			sys.bc	
			apay	
	icc			
3	财务库(fi)	BIZ_TPL_COMM_FI	fi	财务库
		BIZ_TPL_COMM_CAL	cal	存货核算库
		BIZ_TPL_COMM_TMC	tmc	资金库
		BIZ_TPL_COMM_BDAI	bdai	数据智能库
		BIZ_TPL_COMM_EPM	epm	企业绩效库
		BIZ_TPL_COMM_IMC	imc	发票云库
		BIZ_TPL_COMM_TAXC	taxc	税务库
		BIZ_TPL_COMM_FIAS	fias	财务分析服务
		BIZ_TPL_COMM_EBG	ebg	银企服务云库
			epm.epdm	

			ismc	
4	渠道库(drp)	BIZ_TPL_COMM_DRP	drp	渠道库
		BIZ_TPL_COMM_KMALL	kmall	在线商城库
5	供应链库(scm)	BIZ_TPL_COMM_SCM	scm	供应链库
		BIZ_TPL_COMM_QMC	qmc	质量云
		BIZ_TPL_COMM_PHM	phm	制造云库
		BIZ_TPL_COMM_CR	cr	项目库
		BIZ_TPL_COMM_PMC	pmc (生产制造)	项目管理库
			emc	
6	HR 库(hr)	BIZ_TPL_COMM_HR	hr	人力资源库
		BIZ_TPL_COMM_HPDI	hpdi	薪资前端数据集成库
		BIZ_TPL_COMM_HSPP	hspp	工资条库
		BIZ_TPL_COMM_SWC	swc	薪酬福利库
		BIZ_TPL_COMM_HCDM	hcdm	薪酬管理库
		BIZ_TPL_COMM_HRB	hrb	核心人力业务库
		BIZ_TPL_COMM_HMP	hmp	基础中台库
		BIZ_TPL_COMM_HCF	hcf	候选人库
		BIZ_TPL_COMM_SIT	sit	社保个税库
		BIZ_TPL_COMM_TDC	tdc	人才发展库
		BIZ_TPL_COMM_WTC	wtc	工时与假勤库
		BIZ_TPL_COMM_HDTC	hdtc	HR 交付工具库
		BIZ_TPL_COMM_HROS	hros	HR 运营服务云
		BIZ_TPL_COMM_TSC	tsc	人才供应库
		BIZ_TPL_COMM_TSCI	tsci	内部招聘库
		BIZ_TPL_COMM_OPMC	opmc	个人绩效业务库
		7	扩展开发库 (secd)	BIZ_TPL_COMM_SECD
	trans			

3.1.2 创建表空间

创建表空间 COSMIC_DATA、MC_DATA，用于存放 mc 及苍穹数据及索引。



1) 创建数据表空间及索引表空间，单个数据文件 32767m：

```
create tablespace COSMIC_DATA datafile '/path/to/data/cosmic_data01.dbf' size 32767;
create tablespace MC_DATA datafile '/path/to/data/mc_data01.dbf' size 32767;
```

2) 根据客户数据库大小进行表空间扩容，苍穹初始扩容数据表空间为 200GB，后续根据实际需要扩容：

--数据表空间初始化扩容

```
alter tablespace COSMIC_DATA add datafile '/path/to/data/cosmic_data02.dbf' size 32767;
alter tablespace COSMIC_DATA add datafile '/path/to/data/cosmic_data03.dbf' size 32767;
alter tablespace COSMIC_DATA add datafile '/path/to/data/cosmic_data04.dbf' size 32767;
alter tablespace COSMIC_DATA add datafile '/path/to/data/cosmic_data05.dbf' size 32767;
alter tablespace COSMIC_DATA add datafile '/path/to/data/cosmic_data06.dbf' size 32767;
alter tablespace COSMIC_DATA add datafile '/path/to/data/cosmic_data07.dbf' size 32767;
```

3.1.3 创建用户

1 创建 MC 用户

```
--mc用户
create user "cosmic_mc" identified by "密码"
default tablespace "MC_DATA" default index tablespace "MC_DATA";
```

2 创苍穹用户

```
--meta用户
create user "cosmic_meta" identified by "密码" default tablespace "COSMIC_DATA" default index tablespace
"COSMIC_DATA";
--sys用户
create user "cosmic_sys" identified by "密码" default tablespace "COSMIC_DATA" default index tablespace
"COSMIC_DATA";
--fi用户
create user "cosmic_fi" identified by "密码" default tablespace "COSMIC_DATA" default index tablespace
"COSMIC_DATA";
--drp用户
create user "cosmic_drp" identified by "密码" default tablespace "COSMIC_DATA" default index tablespace
"COSMIC_DATA";
--scm用户
create user "cosmic_scm" identified by "密码" default tablespace "COSMIC_DATA" default index tablespace
```



```
"COSMIC_DATA";
--hr用户
create user "cosmic_hr" identified by "密码" default tablespace "COSMIC_DATA" default index tablespace
"COSMIC_DATA";
--secd用户
create user "cosmic_secd" identified by "密码" default tablespace "COSMIC_DATA" default index tablespace
"COSMIC_DATA";
```

3 用户授权

MC 用户授权

```
grant "PUBLIC","RESOURCE","SOI","VTI" to "cosmic_mc";
```

苍穹用户授权

--1. 苍穹的OB用户使用最小权限，如下角色ROL_COSMIC_APP进行授权和业务测试：

--创建角色

```
DROP ROLE ROL_COSMIC_APP;
```

```
CREATE ROLE ROL_COSMIC_APP;
```

--角色授予最小权限

--会话

```
GRANT CREATE SESSION TO ROL_COSMIC_APP;
```

--类型

```
--GRANT CREATE TYPE TO ROL_COSMIC_APP;
```

```
--GRANT ALTER ANY TYPE TO ROL_COSMIC_APP;
```

--数据库对象：序列、存储过程、触发器、包、函数

```
GRANT CREATE SEQUENCE TO ROL_COSMIC_APP;
```

```
GRANT CREATE SYNONYM TO ROL_COSMIC_APP;
```

```
GRANT CREATE PROCEDURE TO ROL_COSMIC_APP;
```

```
GRANT CREATE TRIGGER TO ROL_COSMIC_APP;
```

--表

```
GRANT CREATE TABLE TO ROL_COSMIC_APP;
```

```
GRANT SELECT ANY TABLE TO ROL_COSMIC_APP;
```

```
GRANT INSERT ANY TABLE TO ROL_COSMIC_APP;
```

```
GRANT UPDATE ANY TABLE TO ROL_COSMIC_APP;
```

```
GRANT DELETE ANY TABLE TO ROL_COSMIC_APP;
```

--索引

```
GRANT CREATE INDEX TO ROL_COSMIC_APP;
```

--查询数据字典

```
GRANT SELECT ANY DICTIONARY TO ROL_COSMIC_APP;
```

```
GRANT SELECT ANY TABLE TO ROL_COSMIC_APP;
```

```
grant "PUBLIC","RESOURCE","SOI","VTI",dba TO ROL_COSMIC_APP;
```

--2. 将ROL_COSMIC_APP授权给mc以及苍穹用户

```
grant ROL_COSMIC_APP to cosmic_mc;
grant ROL_COSMIC_APP to cosmic_meta;
grant ROL_COSMIC_APP to cosmic_sys;
grant ROL_COSMIC_APP to cosmic_fi;
grant ROL_COSMIC_APP to cosmic_drp;
grant ROL_COSMIC_APP to cosmic_scm;
grant ROL_COSMIC_APP to cosmic_hr;
grant ROL_COSMIC_APP to cosmic_secid;
```

3.2 恢复苍穹和 MC 模板库

3.2.1 恢复 MC 库

1) 使用 dimp 恢复 mc 数据库。(注意 dimp 命令路径为达梦数据库软件安装路径, 例如 \$DM_HOME/bin)

```
dimp cosmic_mc/密码@IP:端口 FILE=BIZ_TPL_COMM_MC.dmp log=imp_BIZ_TPL_COMM_MC.log fromuser=BIZ_TPL_COMM_MC
touser=cosmic_mc REMAP_TABLESPACE=BIZ_TPL_COMM_DATA:COSMIC_DATA TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
root@23db1719be88:~# dimp cosmic_meta/ FILE=...t.dmp log=imp_meta.log fromuser=...Touser=cosmic_meta REMAP_TABLE=BIZ_TPL_COMM_DATA:COSMIC_DATA TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
dimp V8
local code: PG_GB18030 dump file code: PG_GB18030
start importing schema[...].
start importing the NECESSARY GLOBAL objects of the schema...
finish importing the NECESSARY GLOBAL objects of the schema...
----- [2024-05-24 10:08:46]import table:T1 -----
create table T1
import table T1 , has copied with 10001 rows
import index of table:
IDX_T1_ID
import successfully...
import index of table:
IDX_T2_NAME
import successfully...
start importing the GLOBAL objects of the schema...
finish importing the GLOBAL objects of the schema...
import schema[COSMIC_META] finish...
all the import process spent total 0.247 s
terminate import success without warning
```

2) 数据恢复完成后, 回收 mc 库 dba 权限

```
SQL>
SQL> revoke dba from cosmic_mc;
executed successfully
used time: 3.978(ms). Execute id is 268503.
```

3.2.2 恢复苍穹库

1) 以达梦数据用户 dmdba 登录数据库 : su - dmdba

2) 进入备份文件的存放目录(以/var/backup 为例) : cd /var/backup/

3) 恢复 dmp 文件(共 43 个),恢复数据库(恢复成 7 个目标库,参考前面的分库规则) :

--1.meta库

```
dimp cosmic_meta/'密码'@IP:端口 FILE=BIZ_TPL_COMM_META.dmp log=imp_BIZ_TPL_COMM_META.log
fromuser=BIZ_TPL_COMM_META touser=cosmic_meta REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

--2.sys库

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_SYS.dmp log=imp_BIZ_TPL_COMM_SYS.log
fromuser=BIZ_TPL_COMM_SYS touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_WFS.dmp log=imp_BIZ_TPL_COMM_WFS.log
fromuser=BIZ_TPL_COMM_WFS touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_LOG.dmp log=imp_BIZ_TPL_COMM_LOG.log
fromuser=BIZ_TPL_COMM_LOG touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_AIDB.dmp log=imp_BIZ_TPL_COMM_AIDB.log
fromuser=BIZ_TPL_COMM_AIDB touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_MBR.dmp log=imp_BIZ_TPL_COMM_MBR.log
fromuser=BIZ_TPL_COMM_MBR touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_EIP.dmp log=imp_BIZ_TPL_COMM_EIP.log
fromuser=BIZ_TPL_COMM_EIP touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_BARC.dmp log=imp_BIZ_TPL_COMM_BARC.log
fromuser=BIZ_TPL_COMM_BARC touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```



金蝶云·苍穹&星瀚达梦数据库上线部署说明书

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_RPAC.dmp log=imp_BIZ_TPL_COMM_RPAC.log
fromuser=BIZ_TPL_COMM_RPAC touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_sys/'密码'@IP:端口 FILE=BIZ_TPL_COMM_IMSC.dmp log=imp_BIZ_TPL_COMM_IMSC.log
fromuser=BIZ_TPL_COMM_IMSC touser=cosmic_sys REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

--3.fi财务库

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_FI.dmp log=imp_BIZ_TPL_COMM_FI.log fromuser=BIZ_TPL_COMM_FI
touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_CAL.dmp log=imp_BIZ_TPL_COMM_CAL.log
fromuser=BIZ_TPL_COMM_CAL touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_TMC.dmp log=imp_BIZ_TPL_COMM_TMC.log
fromuser=BIZ_TPL_COMM_TMC touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_BDAI.dmp log=imp_BIZ_TPL_COMM_BDAI.log
fromuser=BIZ_TPL_COMM_BDAI touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_EPM.dmp log=imp_BIZ_TPL_COMM_EPM.log
fromuser=BIZ_TPL_COMM_EPM touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_IMC.dmp log=imp_BIZ_TPL_COMM_IMC.log
fromuser=BIZ_TPL_COMM_IMC touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_TAXC.dmp log=imp_BIZ_TPL_COMM_TAXC.log
fromuser=BIZ_TPL_COMM_TAXC touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_FIAS.dmp log=imp_BIZ_TPL_COMM_FIAS.log
fromuser=BIZ_TPL_COMM_FIAS touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```



金蝶云·苍穹&星瀚达梦数据库上线部署说明书

```
dimp cosmic_fi/'密码'@IP:端口 FILE=BIZ_TPL_COMM_EBG.dmp log=imp_BIZ_TPL_COMM_RBG.log
fromuser=BIZ_TPL_COMM_EBG touser=cosmic_fi REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

--4.drp渠道库
dimp cosmic_drp/'密码'@IP:端口 FILE=BIZ_TPL_COMM_DRP.dmp log=imp_BIZ_TPL_COMM_DRP.log
fromuser=BIZ_TPL_COMM_DRP touser=cosmic_drp REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

dimp cosmic_drp/'密码'@IP:端口 FILE=BIZ_TPL_COMM_KMALL.dmp log=imp_BIZ_TPL_COMM_KMALL.log
fromuser=BIZ_TPL_COMM_KMALL touser=cosmic_drp REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

--5.scm供应链库
dimp cosmic_scm/'密码'@IP:端口 FILE=BIZ_TPL_COMM_SCM.dmp log=imp_BIZ_TPL_COMM_SCM.log
fromuser=BIZ_TPL_COMM_SCM touser=cosmic_scm REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

dimp cosmic_scm/'密码'@IP:端口 FILE=BIZ_TPL_COMM_QMC.dmp log=imp_BIZ_TPL_COMM_QMC.log
fromuser=BIZ_TPL_COMM_QMC touser=cosmic_scm REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

dimp cosmic_scm/'密码'@IP:端口 FILE=BIZ_TPL_COMM_PHM.dmp log=imp_BIZ_TPL_COMM_PHM.log
fromuser=BIZ_TPL_COMM_PHM touser=cosmic_scm REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

dimp cosmic_scm/'密码'@IP:端口 FILE=BIZ_TPL_COMM_CR.dmp log=imp_BIZ_TPL_COMM_CR.log fromuser=BIZ_TPL_COMM_CR
touser=cosmic_scm REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

dimp cosmic_scm/'密码'@IP:端口 FILE=BIZ_TPL_COMM_PMC.dmp log=imp_BIZ_TPL_COMM_PMC.log
fromuser=BIZ_TPL_COMM_PMC touser=cosmic_scm REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

--6.hr库
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HR.dmp log=imp_BIZ_TPL_COMM_HR.log fromuser=BIZ_TPL_COMM_HR
touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HPDI.dmp log=imp_BIZ_TPL_COMM_HPDI.log
fromuser=BIZ_TPL_COMM_HPDI touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HSPP.dmp log=imp_BIZ_TPL_COMM_HSPP.log
fromuser=BIZ_TPL_COMM_HSPP touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
```



TABLE_EXISTS_ACTION=REPLACE PARALLEL=4

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_SWC.dmp log=imp_BIZ_TPL_COMM_SWC.log
fromuser=BIZ_TPL_COMM_SWC touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HCDM.dmp log=imp_BIZ_TPL_COMM_HCMD.log
fromuser=BIZ_TPL_COMM_HCDM touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HRB.dmp log=imp_BIZ_TPL_COMM_HRB.log
fromuser=BIZ_TPL_COMM_HRB touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HMP.dmp log=imp_BIZ_TPL_COMM_HMP.log
fromuser=BIZ_TPL_COMM_HMP touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HCF.dmp log=imp_BIZ_TPL_COMM_HCF.log
fromuser=BIZ_TPL_COMM_HCF touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_SIT.dmp log=imp_BIZ_TPL_COMM_SIT.log
fromuser=BIZ_TPL_COMM_SIT touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_TDC.dmp log=imp_BIZ_TPL_COMM_TDC.log
fromuser=BIZ_TPL_COMM_TDC touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_WTC.dmp log=imp_BIZ_TPL_COMM_WTC.log
fromuser=BIZ_TPL_COMM_WTC touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HDTC.dmp log=imp_BIZ_TPL_COMM_HDTC.log
fromuser=BIZ_TPL_COMM_HDTC touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_HROS.dmp log=imp_BIZ_TPL_COMM_HROS.log
fromuser=BIZ_TPL_COMM_HROS touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA: COSMIC_DATA
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_TSC.dmp log=imp_BIZ_TPL_COMM_TSC.log  
fromuser=BIZ_TPL_COMM_TSC touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA:COSMIC_DATA  
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_TSCI.dmp log=imp_BIZ_TPL_COMM_TSCI.log  
fromuser=BIZ_TPL_COMM_TSCI touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA:COSMIC_DATA  
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

```
dimp cosmic_hr/'密码'@IP:端口 FILE=BIZ_TPL_COMM_OPMC.dmp log=imp_BIZ_TPL_COMM_OPMC.log  
fromuser=BIZ_TPL_COMM_OPMC touser=cosmic_hr REMAP_TABLESPACE=BIZ_TPL_COMM_DATA:COSMIC_DATA  
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

--7.secd库

```
dimp cosmic_secd/'密码'@IP:端口 FILE=BIZ_TPL_COMM_SECD.dmp log=imp_BIZ_TPL_COMM_SECD.log  
fromuser=BIZ_TPL_COMM_SECD touser=cosmic_secd REMAP_TABLESPACE=BIZ_TPL_COMM_DATA:COSMIC_DATA  
TABLE_EXISTS_ACTION=REPLACE PARALLEL=4
```

4) 回收苍穹库 dba 权限

```
SQL> revoke dba from cosmic_meta ;  
revoke dba from cosmic_fi ;  
revoke dba from cosmic_sys ;  
revoke dba from cosmic_drp ;  
revoke dba from cosmic_hr ;  
revoke dba from cosmic_drp ;  
revoke dba from cosmic_scm ;executed successfully  
used time: 3.656(ms). Execute id is 268535.  
SQL> executed successfully  
used time: 1.314(ms). Execute id is 268536.  
SQL> executed successfully  
used time: 1.210(ms). Execute id is 268537.  
SQL> executed successfully  
used time: 1.562(ms). Execute id is 268538.  
SQL> executed successfully  
used time: 1.231(ms). Execute id is 268539.  
SQL> revoke dba from cosmic_drp ;
```


3.3.3 MQ 组件配置

- 1、MQ 类型支持 rabbitmq/rocketmq，默认为 rabbitmq；
- 2、填写现场可用的 MQ 组件连接信息，包括 ip，端口，用户名，密码，vhost 等；
- 3、务必保证 vhost 已通过 mq 命令或 mq 管理控制台创建。

MQ组件配置

类型* rabbitmq

地址*

用户*

密码* 请输入MQ密码

vhost* mcl

请确保已在MQ服务中创建vhost，再进行填写。

3.3.4 Redis 组件配置

- 1、Redis 支持单机、集群、哨兵三种模式，现场按需选择；
- 2、填写完整的服务地址，若服务开启认证则需填写密码。

Redis组件配置

类型 * 单机 ▼

地址 *

密码 *

Redis如未设置密码认证, 无需填写密码

[上一步](#) [下一步](#)

3.3.5 ES 组件配置

填写 MC 自身所需的日志相关服务地址，以及 monitor 登录用户和密码。

ES组件配置

地址 * 127.0.0.1:9200

用户 *

密码 *

ES日志地址 * 127.0.0.1:9300

Kafka日志地址 * 127.0.0.1:9300

[上一步](#) [下一步](#)

3.3.6 初始化确认

列出前面步骤已填写的配置信息供再次确认，确定后将写入 MC 库，同时发布到配置中心（Zookeeper）。

初始化确认

服务访问

服务地址: [redacted] 3:8080/baseline_mc

数据库	消息队列(MQ)
类型: PostgreSQL	类型: rabbitmq
地址: [redacted]	地址: [redacted]
实例: [redacted]	用户: [redacted]
用户: [redacted]	密码:
密码: [redacted]	vhost: mc [redacted]

日志服务(ES、Kafka)	缓存服务(Redis)
地址: 127.0.0.1:9200	模式: redis
用户: [redacted]	地址: [redacted]
密码:	密码:
ES日志: 127.0.0.1:9300	
Kafka日志: 127.0.0.1:9300	

[上一步](#) [确认](#)

3.4 MC 更新配置信息

- 1、MC 自身配置已在 **3.1MC 初始化**中完成；
- 2、创建集群及租户与标准部署流程一致，无特殊注意事项。

3.5 创建数据中心

创建数据库中心过程：

登录 MC：数据库管理->新增数据库服务。达梦数据库支持集群模式与非集群模式，生产环境建议使用一主一备集群模式。

- 1) 单机部署

数据库管理 ×

名称*
合并报表5.0.011达梦升级库_MET,CN

类型*
DaMeng(达梦数据库) ▼

地址*
172.17.54.74

端口*
5236

用户名*
test_upgrade_dm_meta

密码*

启用集群模式

退出 测试连接 保存

2) 集群模式，需要输入一主一备的 IP 端口。

数据库管理 ×

名称*
172.17.32.65-DMCLOUD-SYS1 CN

类型*
DaMeng(达梦数据库) ▼

用户名*
PT_BIZBL_BIG_SYS

密码*

启用集群模式

地址 新增 | 删除

<input type="checkbox"/>	IP	端口
<input type="checkbox"/>	172.17.32.65	5,236
<input type="checkbox"/>	172.17.32.66	5,236

退出 测试连接 保存

4 数据库日常运维

4.1 字符集与排序方式

达梦数据库安装时建议字符集为 UTF8，默认字符集为 GB18030。

4.2 参数修改

1) 查询参数方式

```
SELECT * FROM v$dm_ini WHERE para_name='BUFFER';
```

2) 修改参数的方式

sp_set_para_value(参数类型,'参数名',参数值) :

其中参数类型

--1: 内存和初始化参数同时修改，立即生效

--2: 初始化参数修改，重启生效

例如，修改参数BUFFER=400000，重启生效

```
SP_SET_PARA_VALUE(1, 'BUFFER', 400000);
```

4.3 数据库启停

以 dmdba 用户操作。

1) 数据库关闭方式

--方法一：使用shutdown immediate关闭

disql 超级用户/'密码'@IP:端口

登录后执行如下SQL

```
SHUTDOWN IMMEDIATE;
```

```
exit;
```

即关闭数据库后，退出disql客户端。

```
SQL> shutdown immediate;
executed successfully
used time: 0.246(ms). Execute id is 0.
SQL> Server is stopping...
listener closed and all sessions disconnected
adjust undo retention & wakeup purge thread...full check point starting...
generate force checkpoint, rlog free space[529583184], used space[7279616]
checkpoint begin, used_space[7279616], free_space[529583184]...
checkpoint end, 0 pages flushed, used_space[1824], free_space[536861696].
full check point end.
shutdown audit subsystem...OK
shutdown schedule subsystem...OK
shutdown timer successfully.
pre-shutdown MAL subsystem...OK
shutdown worker threads subsystem...OK
shutdown local parallel threads pool successfully.
shutdown replication subsystem...OK
shutdown sequence cache subsystem...OK
wait for mtsk link worker to exit...OK
shutdown mpp session subsystem...OK
wait for rapply is all over...OK
rapply worker threads exit successfully.
pre ending task & worker threads...OK
shutdown oblink subsystem...OK
shutdown session subsystem...OK
shutdown rollback segments purging subsystem...OK
shutdown rps subsystem...OK
shutdown transaction subsystem...OK
shutdown locking subsystem...OK
shutdown dbms lock subsystem...OK
ending tsk and worker threads...OK
ckpt2 exec immediately begin.
shutdown archive subsystem...OK
shutdown redo log subsystem...OK
shutdown MAL subsystem...OK
shutdown message compress subsystem successfully.
shutdown task subsystem...OK
shutdown trace subsystem...OK
shutdown svr_log subsystem...OK
shutdown plan cache subsystem...OK
shutdown file subsystem...OK
shutdown database dictionary subsystem...OK
shutdown mac cache subsystem...OK
shutdown dynamic login cache subsystem...OK
shutdown ifun/bifun/stun/afun cache subsystem...OK
shutdown crypt subsystem...OK
shutdown pipe subsystem...OK
shutdown compress component...OK
shutdown slave redo subsystem...OK
shutdown kernel buffer subsystem...OK
shutdown SQL capture subsystem...OK
shutdown control file system...OK
shutdown dtype subsystem...OK
shutdown huge buffer and memory pools...OK
close lsnr socket
DM Database Server shutdown successfully.
```

--方法二: systemctl关闭达梦服务

```
systemctl stop DmServiceDMSERVER.service
```

2) 数据库启动方式

Systemctl启动达梦数据库服务:

```
systemctl stop DmServiceDMSERVER.service
```

4.4 备份与恢复

4.4.1 物理备份与恢复

1) dmrman 备份

以 dmdba 用户执行 dmrman 登录

```
RMAN>BACKUP DATABASE '/path/to/data/dmdb/dm.ini'
```

```
FULL BACKUPSET '/path/to/dbbackup/dmbak/dm_db_full_bak01';
```

```
[dmdba@db65 dmbak]$ dmrman
dmrman V8
RMAN> BACKUP DATABASE '/var/data/dmtest/dmtest/dm.ini' FULL BACKUPSET '/var/dbbackup/dmbak/dm_db_full_bak01';
BACKUP DATABASE '/var/data/dmtest/dmtest/dm.ini' FULL BACKUPSET '/var/dbbackup/dmbak/dm_db_full_bak01';
file dm.key not found, use default license!
Database mode = 0, oguid = 0
Normal of FAST
Normal of DEFAULT
Normal of RECYCLE
Normal of KEEP
Normal of ROLL
EP[0]'s cur_lsn[3442445], file_lsn[3442445]
Processing backupset /var/dbbackup/dmbak/dm_db_full_bak01
[Percent:100.00%][Speed:0.00M/s][Cost:00:00:00][Remaining:00:00:00]
backup successfully!
time used: 00:00:01.496
```

2) dmrman 恢复

```
RMAN>RESTORE DATABASE '/path/to/data/dmdb/dm.ini'
FROM BACKUPSET '/path/to/dbbackup/dmbak/dm_db_full_bak01';
```

```
[dmdba@db65 dmbak]$ dmrman
dmrman V8
RMAN> RESTORE DATABASE '/var/data/dmtest/dmtest/dm.ini' FROM BACKUPSET '/var/dbbackup/dmbak/dm_db_full_bak01';
RESTORE DATABASE '/var/data/dmtest/dmtest/dm.ini' FROM BACKUPSET '/var/dbbackup/dmbak/dm_db_full_bak01';
file dm.key not found, use default license!
Normal of FAST
Normal of DEFAULT
Normal of RECYCLE
Normal of KEEP
Normal of ROLL
[Percent:100.00%][Speed:0.00M/s][Cost:00:00:02][Remaining:00:00:00]
restore successfully.
time used: 00:00:02.657
RMAN>
```

3) Recover 恢复数据库到一致性状态

```
RECOVER DATABASE '/path/to/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini'
FROM BACKUPSET '/home/dm_bak/db_full_bak_for_recover_backupset';
```

4) 更新 DB_MAGIC 完成恢复

```
RECOVER DATABASE '/path/to/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' UPDATE DB_MAGIC;
```

4.4.2 逻辑备份恢复

1) 逻辑备份 dexp

--1.备份整个数据库

```
dexp 用户/'密码'@IP:端口 file=full_bak.dmp log=expfull.log full=y parallel=4
```

--2.备份指定 SCHEMA

```
dexp 用户/'密码'@IP:端口 owner=test file=test.dmp log=expfull.log parallel=4
```

--3.备份单张表对象

```
dexp 用户/'密码'@IP:端口 tables=t,t_null file=test_tbs.dmp log=expfull.log parallel=4
```

2)逻辑恢复 dimp

--1.全库恢复

```
dimp 用户/'密码'@IP:端口 file=full_bak.dmp log=imp_full.log full=y parallel=4
--2.恢复某个 user/schema
dimp 用户/'密码'@IP:端口 owner=test1 file=full_bak.dmp log=imp_test.log parallel=4
--3.从 user1 恢复到 user2

dimp 用户/'密码'@IP:端口 fromuser=test touser=test1 file=full_bak.dmp log=imp_test_test1.log parallel=4
--4.恢复特定几张表
dimp 用户/'密码'@IP:端口 file=full_bak.dmp log=imp_tables.log tables=t,t_null parallel=4
```

4.5 数据字典说明

```
--当前用户的全部表
SELECT * FROM USER_TABLES;
--当前用户的全部索引
SELECT * FROM USER_INDEXES;
--表的列定义
SELECT * FROM USER_TAB_COLS;
--查看索引定义的列
select * from user_ind_columns;
```

4.6 用户与最小权限

1) 创建角色

创建最小权限角色ROL_COSMIC_APP便于用户授权。

--苍穹的OB用户使用最小权限，如下角色ROL_COSMIC_APP进行授权和业务测试：

--创建角色

```
DROP ROLE ROL_COSMIC_APP;
```

```
CREATE ROLE ROL_COSMIC_APP;
```

--角色授予最小权限

--角色授予最小权限

--1 会话

```
GRANT CREATE SESSION TO ROL_COSMIC_APP;
```

--2 类型

```
--GRANT CREATE TYPE TO ROL_COSMIC_APP;
```

```
--GRANT ALTER ANY TYPE TO ROL_COSMIC_APP;
```

--3 数据库对象：序列、存储过程、触发器、包、函数

```
GRANT CREATE SEQUENCE TO ROL_COSMIC_APP;
```

```
GRANT CREATE SYNONYM TO ROL_COSMIC_APP;
```

```
GRANT CREATE PROCEDURE TO ROL_COSMIC_APP;
```

```
GRANT CREATE TRIGGER TO ROL_COSMIC_APP;
```

```
--4 表
GRANT CREATE TABLE TO ROL_COSMIC_APP;
GRANT SELECT ANY TABLE TO ROL_COSMIC_APP;
GRANT INSERT ANY TABLE TO ROL_COSMIC_APP;
GRANT UPDATE ANY TABLE TO ROL_COSMIC_APP;
GRANT DELETE ANY TABLE TO ROL_COSMIC_APP;

--5 索引
GRANT CREATE INDEX TO ROL_COSMIC_APP;

--5 SELECT ANY DICTIONARY
GRANT SELECT ANY DICTIONARY TO ROL_COSMIC_APP;
GRANT SELECT ANY TABLE TO ROL_COSMIC_APP;

--6.通用权限
grant "PUBLIC","RESOURCE","SOI","VTI",dba TO ROL_COSMIC_APP;
```

2) 创建用户

```
CREATE USER 用户 IDENTIFIED BY "密码" default tablespace BIZ_TPL_COMM_data index tablespace COSMIC_DATA;
GRANT ROL_COSMIC_APP TO 用户;
```

3) 修改用户密码

登录管理员数据库用户修改用户密码：

```
alter user 用户名 identified by "密码";
```

```
SQL> alter user test1 identified by "ABC@123456";
executed successfully
used time: 3.995(ms). Execute id is 4200.
```

4.7 存储过程文件执行

1) 执行 sql 文件

使用如下方式执行 SQL：

```
disql 用户名/密码@服务器 IP:端口 'a.sql'
```

```
SQL> exit
[dmdba@db65 ~]$ disql test/'"ABC@123456"'@172.17.32.65:5237 'a.sql'
Server[172.17.32.65:5237]:mode is normal, state is open
login used time : 0.577(ms)
disql V8
SQL> insert into t values(1,'This is Insert test!');
affect rows 1
used time: 0.805(ms). Execute id is 4800.
```

2) 存储过程调试

如果存储过程文件执行中，如果执行类型定义相关 SQL 中断，则建议手工执行。错误

描述如下：

```
SQL> call pkg_a;

call pkg_a;

[-2207]:Error in line: 1

Member access [pkg_a] unresolved.

used time: 0.472(ms). Execute id is 0.

SQL> call pkg_a;
[-2207]:Error in line: 1
Member access [pkg_a] unresolved.
used time: 0.215(ms). Execute id is 0.
SQL> 2 3 4
```

使用如下方式执行类型定义的 SQL：

```
SQL> CREATE OR REPLACE TYPE UDT_OBJ_IDS FORCE IS object( FID VARCHAR(32000));

2 /

executed successfully

used time: 12.810(ms). Execute id is 1700.

SQL> CREATE OR REPLACE TYPE UDT_SPLIT_TBL is table of UDT_OBJ_IDS;

2 /

executed successfully

used time: 9.850(ms). Execute id is 1701.

SQL>
```

```
SQL> CREATE OR REPLACE TYPE UDT_OBJ_IDS FORCE IS object( FID VARCHAR(32000));
2 /
executed successfully
used time: 12.810(ms). Execute id is 1700.
SQL> CREATE OR REPLACE TYPE UDT_SPLIT_TBL is table of UDT_OBJ_IDS;
2 /
executed successfully
used time: 9.850(ms). Execute id is 1701.
SQL> SQL> CREATE OR REPLACE TYPE UDT_OBJ_IDS FORCE IS object( FID VARCHAR(32000));
2 /
executed successfully
used time: 12.810(ms). Execute id is 1700.
SQL> CREATE OR REPLACE TYPE UDT_SPLIT_TBL is table of UDT_OBJ_IDS;
2 /
```

4.8 慢查询日志维护

1) 开启慢查询日志功能

执行以下 SQL 修改初始化参数，开启日志记录功能：

```
SP_SET_PARA_VALUE(1, 'SVR_LOG', 1);
```

2) 设置慢查询阈值

修改配置文件/dmdata/DAMENG/sqllog.ini，慢查询超过 1 秒记录日志

```
MIN_EXEC_TIME =1500
```

```
root@23db1719be88:/opt/dmdbms/data/DAMENG# cat sqllog.ini
BUF_TOTAL_SIZE      = 10240          #SQLs Log Buffer Total Size(K)(1024~1024000)
BUF_SIZE            = 1024          #SQLs Log Buffer Size(K)(50~409600)
BUF_KEEP_CNT        = 6            #SQLs Log buffer kepted count(1~100)

[SLOG_ALL]
FILE_PATH           = /opt/dmdbms/log/slog
PART_STOR           = 0
SWITCH_MODE         = 2
SWITCH_LIMIT        = 128
ASYNC_FLUSH         = 1
FILE_NUM            = 5
ITEMS               = 0
SQL_TRACE_MASK      = 1
MIN_EXEC_TIME       = 1000
USER_MODE           = 0
USERS               =
```

3) 慢查询功能激活

修改慢查询阈值后，需要关闭日志记录功能后再启用才能激活慢查询功能：

```
SP_REFRESH_SVR_LOG_CONFIG();
```

4) 查询慢查询文件

慢查询日志位于 sqllog.ini 配置文件 FILE_PATH 指定路径 (/opt/dmdbms/log/slog)：

```
root@23db1719be88:/opt/dmdbms/log/slog# tail -f dmsql_DMSEVER_20240307_143036.log
2024-03-07 14:35:29.987 (EP[0] sess:0x7f33dc011360 thrd:283 user:NULL trxid:0 stmt:NULL appname:) STARTUP
2024-03-07 14:35:29.987 (EP[0] sess:0x7f33dc011360 thrd:283 user:SYSDBA trxid:0 stmt:NULL appname:disql) LOGIN
2024-03-07 14:35:29.988 (EP[0] sess:0x7f33dc011360 thrd:283 user:SYSDBA trxid:0 stmt:NULL appname:disql) ALLOC STMT
2024-03-07 14:35:42.004 (EP[0] sess:0x7f33dc011360 thrd:283 user:SYSDBA trxid:0 stmt:0x7f33dc034b58 appname:disql ip::ffff:127.0.0.1) [ORA]: begin
dbms_lock.sleep(5);
end;
2024-03-07 14:35:42.004 (EP[0] sess:0x7f33dc011360 thrd:283 user:SYSDBA trxid:46456 stmt:NULL appname:disql) TRX: START
2024-03-07 14:35:47.045 (EP[0] sess:0x7f33dc011360 thrd:283 user:SYSDBA trxid:46456 stmt:0x7f33dc034b58 appname:disql ip::ffff:127.0.0.1) [CAL] begin
dbms_lock.sleep(5);
end; EXECUTE: 5041(ms).
```

5 数据库调优

5.1 查看查询计划

1) 查询分析器

使用 explain 查看执行计划，例如：

```
SQL> explain select * from t where id>1;
1  #NSET2: [1, 1, 86]
2  #PRJT2: [1, 1, 86]; exp_num(3), is_atom(FALSE)
3  #SLCT2: [1, 1, 86]; t.id > var1
4  #CSCN2: [1, 2, 86]; INDEX33555513(t)

used time: 1.306(ms). Execute id is 0.
SQL>
```

2) autotrace 查看详细执行计划(新版本才有该功能)

--1.执行 set autotrace on;

--2.执行 SQL 语句

--3.autotrace 执行计划

```
1  #NSET2: [8266, 182620, 647]
2  #LOCAL COLLECT: [8266, 182620, 647]: op_id(3) n_grp_by (0) n_cols(0) n_keys(0) for_sync(TRUE)
3  #PRJT2: [8266, 182620, 647]: exp_num(28), is_atom(FALSE)
4  #FRJT2: [8266, 182620, 647]: exp_num(28), is_atom(FALSE)
5  #LOCAL GATHER: [8266, 182620, 647]: op_id(1) n_grp_by (0) n_cols(28) n_keys(1) invoke_flag(0) top_flag(1) KEY(COL_14)
6  #SORT3: [8266, 182620, 647]: key_num(1), is_distinct(FALSE), is_adaptive(0), MEM_USED(10950835KB), DISK_USED(0KB)
7  #SLCT2: [4510, 18262031, 647]: (A.FORGID = var2 AND A.FBIZDATA.RECORDID <= var3 AND A.FBSED >= var4 AND A.FBSED < var5)
8  #CSCN2: [4510, 18262031, 647]: INDEX33579333(T_HSAS_NONRECURBIZDATA)

Statistics
-----
0          data pages changed
0          undo pages changed
2462536    logical reads
0          physical reads
0          redo size
8418      bytes sent to client
1751      bytes received from client
1          roundtrips to/from client
0          sorts (memory)
0          sorts (disk)
20        rows processed
0          io wait time(ms)
1510      exec time(ms)

已用时间: 00:00:01.510. 执行号:56301.
```

5.2 统计信息收集

统计信息对查询计划的选择至关重要，因此做如下详细说明。

1) 收集用户的统计信息

```
DBMS_STATS.GATHER_SCHEMA_STATS (
    ownname          VARCHAR,
    estimate_percent DOUBLE,
    block_sample     BOOLEAN,
    method_opt       VARCHAR,
    degree           NUMBER,
```

```
granularity    VARCHAR,
cascade        BOOLEAN
);
```

其中的参数依次为:

- ownname 收集统计信息用户名
- estimate_percent 采样率, 范围为[0.0001,1], NULL 或者 1 表示全量收集
- block_sample 是否采用块级采样
- method_opt 列统计信息选项
 - a) FOR ALL COLUMNS [size_clause]: 收集所有列统计信息
 - b) FOR COLUMNS [column_clause] [size_clause]: 收集指定列统计信息
 size_clause : SIZE { integer | AUTO } 用于指定直方图信息
 1. integer: 直方图的 bucket 数量, 范围为[1,2048]
 2. AUTO: 由数据库决定是否生成直方图
- degree 统计信息收集并行度, 提升并行度可以提升统计信息收集的效率
- granularity 分区统计粒度
 - a) ALL: 收集表和分区
 - b) GLOBAL: 只收集表
 - c) GLOBAL AND PARTITION: 收集表和分区
 - d) PARTITION: 只收集分区
 - e) AUTO: 由分区的类型自动决定
- cascade 是否收集索引统计信息

举例: 收集 SYSDBA 统计信息, 采样率为 50%, 对该模式下所有的列做统计信息, 大字段的列除外。如下所示:
 DBMS_STATS.GATHER_SCHEMA_STATS('SYSDBA',50,TRUE,'FOR ALL COLUMNS SIZE AUTO');

2)收集表统计信息

```
DBMS_STATS.GATHER_TABLE_STATS (
  ownname          VARCHAR,
  tabname          VARCHAR,
  partname         VARCHAR,
  estimate_percent NUMBER,
  block_sample     BOOLEAN,
  method_opt       VARCHAR,
  degree           NUMBER,
  granularity      VARCHAR,
  cascade          BOOLEAN
);
```

其中的参数依次为:

- ownname 统计信息收集的用户名
 - tabname 统计信息收集表名
 - partname 统计信息收集分区名
- 其余参数设置参考官方文档说明。

举例: 收集 TEST 用户 T 表, 采样频率 0.2%, 不收集直方图信息, 级联收集索引统计信息, 统计信息收集并行度 2。如下所示:

```
DBMS_STATS.GATHER_TABLE_STATS(
  OWNNAME=>'TEST',
```

```
TABNAME=>'T',  
ESTIMATE_PERCENT=>0.2,  
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1',  
DEGREE=>2,  
CASCADE=>TRUE,  
NO_INVALIDATE=>FALSE);
```

3)收集索引统计信息

```
DBMS_STATS.GATHER_TABLE_STATS (  
    OWNNAME          VARCHAR(128),  
    INDNAME          VARCHAR(128),  
    PARTNAME         VARCHAR(128),  
    ESTIMATE_PERCENT DOUBLE,  
    DEGREE           INT,  
    GRANULARITY      VARCHAR,  
    NO_INVALIDATE    BOOLEAN,  
    FORCE             BOOLEAN  
);
```

其中的参数依次为:

- OWNNAME 统计信息索引所属用户名
- INDNAME 统计信息索引名
- PARTNAME 统计信息索引分区名

其余参数参考达梦官方文档。

举例：收集 TEST 用户下非分区索引 IDX_T_ID 统计信息，采样率 0.2%，并行度 4。如下所示：

```
DBMS_STATS.GATHER_INDEX_STATS(  
    OWNNAME=>'TEST',  
    INDNAME =>'IDX_T_ID',  
    ESTIMATE_PERCENT=>0.2,  
    DEGREE=>2,  
    NO_INVALIDATE=>FALSE);
```

5.3 会话相关操作

1) 获取会话 ID

使用如下 SQL 查看会话:

```
select sess_id from v$sessions WHERE user_name='PRD_SCM';
```

2) 使用如下 SQL 杀死会话

```
SP_CLOSE_SESSION(2410147992);
```

5.4 查看当前会话执行的 SQL

使用如下 SQL 进行探查执行时长超过 5 秒的 SQL:

```
SELECT *
FROM
( SELECT SESS_ID,SQL_TEXT,DATEDIFF(SS, LAST_RECV_TIME, SYSDATE) Y_EXETIME, SF_GET_SESSION_SQL(SESS_ID)
FULLSQL,CLNT_IP FROM V$SESSIONS WHERE STATE='ACTIVE') WHERE Y_EXETIME>=5;
```

LINEID	SESS_ID	SQL_TEXT	Y_EXETIME	FULLSQL	CLNT_IP
1	47302488167632	begin dbms_lock.sleep(10); end;	4	begin	
		dbms_lock.sleep(10);			
		end; ::ffff:172.17.32.65			

5.5 查看存储工程的调试输出

调试存储过程使用 DBMS_OUTPUT.PUT_LINE() 进行调试输出, 在 disql 中使用 set serveroutput on 开启日志打印调试功能:

```
SQL> set serveroutput on;
SQL> call p_test;
Output verify!
DMSQL executed successfully
used time: 0.159(ms). Execute id is 1817.
```

6 应用参数说明

6.1 JDBC 驱动程序说明

驱动版本: DmJdbcDriver-18.jar(8.1.2.84)

驱动类: dm.jdbc.driver.DmDriver

URL: jdbc:dm://服务 IP:端口/db

6.2 MC 相关参数

无。