

# kingwow 分布式数据库

## 系统安装手册



上海丛云信息科技有限公司

2017 年 8 月

## 目录

1. 简介.....	4
1.1. 原理及架构.....	4
2. 系统规划.....	5
2.1. 软硬件要求.....	5
2.2. 部署模式.....	6
2.3. 安装规划.....	7
2.4. 服务器规划.....	7
2.5. 目录规划.....	8
2.6. 磁盘挂载点规划.....	8
3. 配置安装环境.....	9
3.1. 操作系统配置.....	9
3.1.1. 配置网卡名称 (RHEL6 下) .....	9
3.1.2. 配置内核参数.....	10
3.1.3. 配置时间同步.....	10
3.1.4. 配置资源限制.....	12
3.2. 创建安装用户.....	12
3.3. 配置环境变量.....	13
3.4. 配置服务器间免登录.....	13
3.4.1. 配置过程.....	13
3.4.2. 验证.....	13
4. RPM 包安装.....	13
4.1. 复制安装包.....	14
4.2. 安装依赖库.....	14
3.1. 安装 kingwow 数据库.....	14
3.2. 创建数据目录.....	14
5. 启动与管理系统.....	15
5.1. 数据库服务启动命令.....	15
5.1.1. MG 启动命令.....	15
5.1.2. TG 启动命令.....	15
5.1.3. DG 启动命令.....	16
5.1.4. CG 启动命令.....	16
5.2. 单集群启动示例.....	16

5.3. 多集群启动示例.....	17
5.4. 使用 MySQL 客户端.....	18

# 1. 简介

kingwow 是一个新型的分布式关系型数据库系统，支持多活部署，不仅具有良好的扩展性、高可用性、支持关系模型、支持事务处理、采用标准 SQL 接口，同时不需要用户进行分库分表，没有跨节点操作限制，因此更适用于大规模结构化数据管理和高并发事务处理的应用场景。

## 1.1. 原理及架构

kingwow 是运行于 PC 服务器集群上的分布式数据库系统，通过对传统关系数据库系统功能的拆分，实现了数据的分布式存储和查询的分布式处理，能够支持海量结构化数据的管理。如图 1 所示

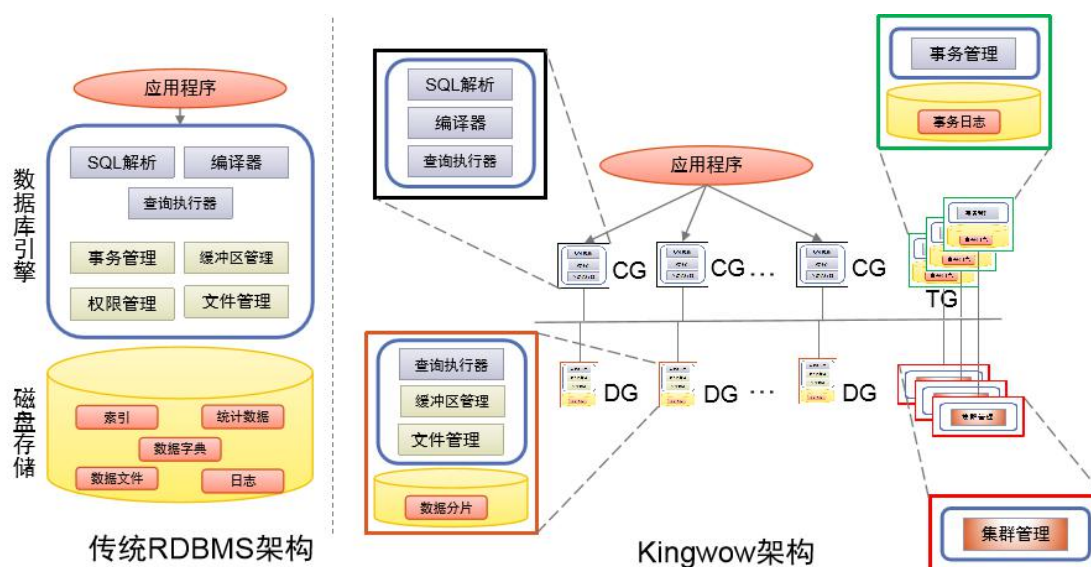


图 1 kingwow 与传统 RDBMS 架构对比

kingwow 集群中存在 4 种角色的服务进程，其中 CG 进程实现了传统数据库的查询解析、编译和执行的功能；DG 进程实现了数据文件的分布式存储与访问功能；TG 进程基于内存引擎实现了高吞吐量的事务处理功能；MG 实现了集群管理功能。集群中通常部署多个 DG 和 CG 节点，通过添加节点数量可以扩展系统的存储和查询处理能力。集群中可以部署多个 TG 节点，采用基于 Quorum 的提交协议，能够保证在多数派节点正常的情况下不丢失数据。集群中同样可以部署多个 MG 节点，通过 Paxos 协议来实现自动选主，保证了集群的高可用。

**ManEng：简称 MG，系统的控制节点，负责集群管理和数据分布。**

- 集群中可以设置多个 MG 节点（一主多备），通过 Paxos 协议实现自动选主，能够保证集群的高可用；
- 通过租约机制（Lease）来管理集群中的 TG 节点，保证同一时刻只有一个 TG 提供事务处理服务；
- 通过心跳机制（Heartbeat）来管理集群中的 CG 和 DG 节点，感知节点的上下线；
- kingwow 中的数据表按主键范围被划分为多个子表（称为 Tablet），每个子表可以有多个副本，分别存储在不同的 DG 节点上。MG 通过 RootTable 结构来记录 Tablet 在 DG 上的分布位置信息，并管理 Tablet 的副本数量、复制、迁移和合并；

**TransEng：简称 TG，系统中的内存事务处理引擎。**

- 与传统数据库系统一样采用 WAL (Write-Ahead Logging) 技术来保证事务的原子性和持久性；
- 集群中可以部署多个 TG 节点（一主多备），由主 MG 通过租约来确定主 TG，主 TG 的日志采用基于 Quorum 协议的方式同步到备 TG，能够保证在超过半数 TG 正常的情况不丢失任何数据；
- 事务处理产生的增量数据被保存在 TG 的内存中，通过合并操作与 DG 中保存的静态数据合并，产生新版本的静态数据；
- 支持跨表跨行的事务，不存在分布式事务，不需要两阶段提交协议；
- 采用多版本并发控制协议（MVCC）实现 Read Committed 事务隔离级别；

#### CalEng：简称 CG，实现了查询处理功能。

- 支持 MySQL 通讯协议，对 SQL 语句进行解析生成执行计划，并管理语句执行计划的运行；
- 缓存 Tablet 的分布位置信息，将 SQL 语句涉及的数据请求同时发送给对应的 DG，并将所有 DG 返回的结果合并；
- CG 本身无状态，宕机不会对应用系统产生影响，应用可以通过其它 CG 获得服务；

#### DatEng：简称 DG，实现了数据的分布式存储和访问。

- 存储和管理 Tablet 副本，提供本地数据查询服务，支持数据缓存；
- 响应 CG 的数据请求时，会从 TG 拉取对应的增量数据，融合后得到数据的最新状态，返还给 CG；
- 在启动数据合并操作后，DG 将会从 TG 拉取所有本地数据对应的增量数据，与本地数据合并后产生新版本的 Tablet；
- 数据合并操作期间，DG 正常提供数据查询服务；

## 2. 系统规划

### 2.1. 软硬件要求

虽然使用一台物理机就可以安装和运行 kingwow 系统，但这样的环境多用于学习和测试。为了充分发挥分布式系统的优势，建议使用 3 台以上的物理机环境。由于集群中不同类型的节点所消耗的主要系统资源不同，在规划 kingwow 数据库集群时，通常将 MG 和 TG 安装在同一台物理机上，将 DG 和 CG 安装在同一台物理机上。

安装 kingwow 数据库系统所需的软硬件环境如表 1-1 所示。

表 1-1：kingwow 数据库系统软硬件配置需求

		MG/TG	DG/CG
最低配置	数量	3	3
	系统	Red Hat Enterprise Linux Server release 6.8 (64bit)及以上版本 <sup>注</sup>	
	CPU	8 核*1	8 核*1
	内存	64GB	32GB
	磁盘	SAS * 4	SAS * 4

	网络	千兆口*1	千兆口*1
推荐配置	数量	3	3
	系统	Red Hat Enterprise Linux Server release 6.8 (64bit)及以上版本 <sup>注</sup>	
	CPU	24 核*2	24 核*1
	内存	128GB	64GB
	磁盘	SSD * 8	SSD * 8
	网络	千兆口*2	千兆口*2

注：也可使用具有相同内核的 CentOS 操作系统；

## 2.2. 部署模式

kingwow 分布式数据库系统支持多种部署方式，见表 2-2。用户可以根据需要选择合适的部署方式，并且能够在使用过程中进行灵活的调整。

表 1-2：kingwow 数据库系统的部署方式

部署方式	说明
单机	安装在一台服务器上，只启动 1 个 MG、1 个 TG、1 个 CG 和 1 个 DG 进程。这种方式能够保证 kingwow 正常地运行，但无法体现出分布式系统的优势，通常只用于学习和测试的场景。
单机房 (单 MG/TG)	安装在同一机房的多台服务器上，其中一台服务器上启动 MG 和 TG 进程，其它每台服务器上启动 CG 和 DG 进程。这种方式无法处理 MG/TG 节点宕机的场景，只能用于对可用性要求不高的应用系统。
单机房 (多 MG/TG)	安装在同一机房的多台服务器上，其中多台服务器分别启动主备 MG/TG 节点（通常为 1 主 2 备），其它每台服务器上启动 CG 和 DG 进程。这种方式可以满足大多数应用系统对可靠性和可用性的需求。
多机房	安装在多个机房的多台服务器上，通常每个机房中都部署有 MG、TG、CG 和 DG。这种方式能够满足应用系统对异地容灾的需求，当

	主机房发生故障时，可以自动将备机房切换为主机房，继续对外提供服务。
--	-----------------------------------

### 2.3. 安装规划

在安装 kingwow 数据库之前，数据库管理员首先应当对系统的资源和环境进行规划，包括服务器的使用，IP 地址和端口、安装目录、数据目录和磁盘挂载点等内容。

### 2.4. 服务器规划

数据库管理员可以根据示例表 1-3 对 kingwow 系统所使用的服务器进行规划，包括服务器所属集群，IP 地址、网卡名称、安装用户、安装目录、所运行服务、使用端口等信息。

表 1-3：kingwow 数据库系统服务器使用规划示例

集群#	服务器	kingwow 服务	服务端口	内部端口
1	IP 地址：192.168.1.2 网卡名称：eth0 安装用户：admin 用户密码：Kingwow@123 安装目录：/home/admin/kingwow 应用名称：obhdfs	主 MG	2500	-
		主 TG	2700	2701
	IP 地址：192.168.1.3 网卡名称：eth0 安装用户：admin 用户密码：Kingwow@123 安装目录：/home/admin/kingwow 应用名称：obhdfs	DG	2600	-
		CG	2880	2800
	IP 地址：192.168.1.4 网卡名称：eth0 安装用户：admin 用户密码：Kingwow@123 安装目录：/home/admin/kingwow 应用名称：obhdfs	DG	2600	-
		CG	2880	2800
	IP 地址：192.168.1.5 网卡名称：eth0 安装用户：admin 用户密码：Kingwow@123 安装目录：/home/admin/kingwow 应用名称：obhdfs	DG	2600	-
		CG	2880	2800

注：这一个单机房(单 MG/TG)的部署，使用了 4 台服务器；

## 2.5. 目录规划

数据库管理员需要规划和管理每台服务器上不同服务存储数据的目录位置, 并且需要根据数据规模、使用频率、访问性能等需求将数据目录与相应的磁盘挂载点关联起来。kingwow 系统中的服务需要持久化到磁盘上的内容包括日志和数据, 表 1-4 是一个目录规划的示例。

表 1-4 : kingwow 数据库服务目录使用规划示例

数据库服务	类型	存储位置
MG	数据	/home/admin/kingwow/data/mg
	日志	/home/admin/kingwow/data/log/mg_commitlog
TG	数据	/home/admin/kingwow/data/tg_data/raid0/store0 /home/admin/kingwow/data/tg_data/raid0/store1 /home/admin/kingwow/data/tg_data/raid1/store0 /home/admin/kingwow/data/tg_data/raid1/store1 ... .. /home/admin/kingwow/data/tg_data/raidn/store0 /home/admin/kingwow/data/tg_data/raidn/store1
	日志	/home/admin/kingwow/data/tg_commitlog
DG	数据	/home/admin/kingwow/data/1 /home/admin/kingwow/data/2 ... .. /home/admin/kingwow/data/n

## 2.6. 磁盘挂载点规划

kingwow 系统中的 TG 服务和 DG 服务分别需要存储大规模的增量数据和静态数据, 建议不与操作系统和其它服务共享磁盘, 而使用单独的磁盘来存储这些服务的数据。数据库管理员需要对每台服务器上的磁盘挂载点进行规划, 并且能够与目录规划相对应。

例如, 在一台运行了 DG 的服务器上, 规划 8 块磁盘用于专门存储 DG 的数据, 数据库管理员可以在 kingwow 的安装目录下创建 DG 的数据目录, 如下 :

- /home/admin/kingwow/data/1
- /home/admin/kingwow/data/2
- ... ..

- /home/admin/kingwow/data/8

然后将 8 块磁盘的挂载点设为 :

- /data/1



- /data/2
- ... ..
- /data/8

最后在挂载点目录与数据目录之间建立链接：

- `ln -s /data/1 /home/admin/kingwow/data/1`
- `ln -s /data/2 /home/admin/kingwow/data/2`
- ... ..
- `ln -s /data/3 /home/admin/kingwow/data/3`

### 3. 配置安装环境

为了能够正常运行 kingwow 数据库系统，数据库管理员需要对安装环境进行必要的配置，包括设置网卡、内核参数、时间同步、环境变量等。

#### 3.1. 操作系统配置

##### 3.1.1. 配置网卡名称（RHEL6 下）

在启动 kingwow 系统服务时，需要在启动命令中指定所使用的网卡名称。为了便于记忆和管理，建议在所有的服务器上使用相同的网卡名称，例如“eth0”。如果在不同服务器上使用了不同的网卡名称，在启动服务时需要特别注意指定网卡名称的参数是否正确。

设置服务器网卡名称为“eth0”的操作步骤如下：

1. 以 root 用户登录 kingwow 服务器；
2. 编辑修改“/etc/udev/rules.d/70-persistent-net.rules”文件的内容；

```
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x1533 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="7*", ATTR{address}=="38:d5:47:02:59:79", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x8086:0x1533 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="7*", ATTR{address}=="38:d5:47:02:59:7a", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
```

3. 编辑修改“/etc/sysconfig/network-scripts/ifcfg-eth0”文件的内容；

```
DEVICE="eth0"
BOOTPROTO=static
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE=Ethernet
NETMASK=255.255.255.0
IPADDR=192.168.1.7
```

注：IPADDR 需要设置为实际使用的 IP 地址

4. 通过执行 reboot 命令重新启动服务器；

5. 通过执行 `ifconfig` 命令，确认网卡名称设置正确；

```
[root@0Bse7 admin]# ifconfig
eth0      Link encap:Ethernet  HWaddr 38:D5:47:02:59:79
          inet addr:192.168.1.7  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::3ad5:47ff:fe02:5979/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:194326504 errors:0 dropped:0 overruns:0 frame:0
          TX packets:194306775 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:41414712520 (38.5 GiB)  TX bytes:50879904730 (47.3 GiB)
          Memory:fa200000-fa27ffff
```

### 3.1.2. 配置内核参数

配置操作系统内核参数的操作步骤如下：

1. 以 root 用户登录 kingwow 服务器；
2. 编辑 “/etc/sysctl.conf” 文件，在末尾添加以下内容：

```
# for kingwow
net.core.somaxconn = 2048
net.core.netdev_max_backlog = 10000
net.core.rmem_default = 16777216
net.core.wmem_default = 16777216
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.ip_local_port_range = 3500 65535
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.tcp_syncookies = 0
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_max_syn_backlog = 16384
net.ipv4.tcp_fin_timeout = 15
net.ipv4.tcp_max_syn_backlog = 16384
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
vm.swappiness = 0
```

3. 执行以下命令，使修改生效；

```
sysctl -p /etc/sysctl.conf
```

### 3.1.3. 配置时间同步

kingwow 数据库系统通常运行于 PC 服务器集群之上，对于集群中服务器间的时钟误差有较为严格的要求，可以通过 NTP 时钟同步来保证所有的服务器时间一致。在系统运行期间，如果发生服务器重启的情况，需要关注 NTP 时钟同步服务是否正常。

使用 NTP 时钟同步需要关闭所有服务器的防火墙(或者开放 NTP 服务相应端口)，关闭防火墙的命令为：

```
service iptables stop
```

## ● 开启 NTP 服务

如果没有现成可用的 NTP 服务器，可以在 kingwow 数据库集群中任选一台服务器作为 NTP 服务器，在其上开启 NTP Server 服务，操作步骤如下：

1. 以 root 用户登录 NTP Server 服务器；
2. 编辑修改 “/etc/ntp.conf” 文件内容；

添加一行 server 127.127.1.0

```
driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
#restrict default kod nomodify notrap nopeer noquery
restrict default kod nomodify

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1
restrict 192.168.1.0 mask 255.255.255.0

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 210.72.145.44
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

3. 执行以下命令，重启 NTP Server 服务；

```
service ntpd restart
```

4. 执行以下命令，查看 NTP Server 服务是否启动；

```
service ntpd status
```

5. 执行以下命令，设置 NTP 服务开机自动启动；

```
chkconfig ntpd on
```

## ● 开启 NTP 客户端

在启动 NTP Server 服务之后，需要等待 5 分钟，才能在其它服务器上开启 NTP Client 服务，否则会提示 “no server suitable for synchronization found” 错误。假设 NTP Server 服务器的 IP 地址为 192.168.1.4，在所有其它服务器上开启 NTP Client 服务的操作步骤如下：

1. 以 root 用户登录服务器；
2. 执行以下命令，停止 NTP 服务；

```
service ntpd stop
```

3. 执行以下命令，同步 NTP Server 的时间；

```
ntpdate 192.168.1.4
```

4. 编辑修改 “/etc/ntp.conf” 文件内容；

添加一行 `server 192.168.1.4`

5. 重启 NTP 服务

```
service ntpd start
```

### 3.1.4. 配置资源限制

Linux 操作系统默认所使用堆栈和文件数的限制值较小，为了正常运行 kingwow 数据库服务，需要放大对资源的限制，操作步骤如下：

1. 以 root 用户登录服务器；
2. 编辑修改配置文件 “/etc/security/limits.conf”，添加如下信息：

```
admin soft stack 204800
admin hard stack 204800
admin soft nofile 65535
admin hard nofile 65535
```

其中 *admin* 为 kingwow 安装用户名

## 3.2. 创建安装用户

kingwow 默认安装用户为 admin,在所有 kingwow 服务器中创建安装用户的操作步骤如下：

1. 以 root 用户登录服务器；
2. 执行如下命令，创建安装用户 admin；

```
useradd admin
```
3. 执行如下命令，为用户 admin 设置密码；

```
passwd admin
```
4. 为用户 admin 赋予 sudo 权限（非必须）；
  - a) 执行以下命令，添加 “/etc/sudoers” 文件的写权限

```
chmod u+w /etc/sudoers
```
  - b) 编辑 “/etc/sudoers” 文件中 “root ALL=(ALL) ALL” 后添加如下黑体语句：

```
root ALL=(ALL) ALL
admin ALL=(ALL) ALL
```
  - c) 执行以下命令，删除 “/etc/sudoers” 文件的写权限；

```
chmod u-w /etc/sudoers
```

### 3.3. 配置环境变量

为了正常运行 kingwow 数据库系统，需要对安装用户 admin 的环境变量进行配置，操作步骤如下：

1. 以 admin 用户登录 kingwow 服务器；
2. 编辑修改 “/home/admin/.bashrc” 文件，添加如下语句：  

```
export  
LD_LIBRARY_PATH=~:/kingwow/lib:/usr/local/lib64:/usr/:/usr/lib/:/usr/local/lib:/lib  
export PATH=~:/kingwow/bin:$PATH  
ulimit -c unlimited
```
3. 执行以下命令，让环境配置生效；  

```
source ~/.bashrc
```

### 3.4. 配置服务器间免登录

在所有的 kingwow 服务器中选择一台作为本机，配置该服务器到所有服务器的免登录（包括本机到本机）。配置免登录后，该服务器在连接其他服务器时无需输入密码。

#### 3.4.1. 配置过程

假设本机的 IP 地址为 “192.168.1.4”，配置免登录的操作步骤如下：

1. 以 admin 用户登录服务器；
2. 执行以下命令，并按 “Enter” 键，直至生成公钥；  

```
ssh-keygen -t rsa -P ""
```
3. 执行以下命令，并根据提示输入登录密码，配置免登录；  

```
ssh-copy-id admin@192.168.1.2  
ssh-copy-id admin@192.168.1.3  
ssh-copy-id admin@192.168.1.4  
ssh-copy-id admin@192.168.1.5
```

#### 3.4.2. 验证

配置免登录完成后，在本机中输入 “ssh admin@192.168.1.X” ；

- 如果无需输入密码，则表示配置免登录成功；
- 如果仍需要输入密码，则请重新配置免登录或关闭 SELINUX 后重试；

## 4. RPM 包安装

## 4.1. 复制安装包

将 kingwow 数据库的安装文件复制到“~/install”目录下；

## 4.2. 安装依赖库

1. 以 admin 用户登录服务器；
2. 执行以下命令，进入安装目录；

```
cd ~/install
```

3. 执行以下命令，安装依赖库；

```
sudo yum install mysql (如已有 mysql 客户端则不必安装)
sudo yum install snappy-1.1.2-1.x86_64.rpm
sudo yum install lzo-2.06-0.x86_64.rpm
```

## 3.1. 安装 kingwow 数据库

执行一下命令，安装 kingwow 数据库：

```
sudo rpm -ivh kingwow-1.0.44.x86_64.rpm --prefix=/home/admin/kingwow
```

注：如果不加 prefix 参数，默认安装在/home/admin/kingwow 目录下；

## 3.2. 创建数据目录

本样例安装挂载使用了 4 块数据盘到/data/1..4，数据目录：~/kingwow/data 最终结构如下

data

```
├── 1 -> /data/1/admindata/dg_data
├── 2 -> /data/2/admindata/dg_data
├── 3 -> /data/3/admindata/dg_data
├── 4 -> /data/4/admindata/dg_data
├── mg
├── mg_commitlog -> /data/1/admindata/commitlog/mg_commitlog
├── tg_commitlog -> /data/1/admindata/commitlog/tg_commitlog
├── tg_data
│   ├── raid0
│   │   ├── store0 -> /data/1/admindata/tg_data/1
│   │   └── store1 -> /data/2/admindata/tg_data/2
│   └── raid1
│       ├── store0 -> /data/3/admindata/tg_data/3
│       └── store1 -> /data/4/admindata/tg_data/4
```

1. 以 admin 用户登录服务器，并进入安装目录~/kingwow；
2. 执行以下命令，创建和配置所需数据目录：

```
sh RebuildDir.sh -m performance -u 4 -c 4 -d obhdfs
```

参数含义：

-m: 创建数据目录的模式：[test|performance] (performance 使用/data/1..n 作为数据挂载点，test 使用~/data 作为数据挂载点，默认 test)

-u: MG/TG 服务所使用的挂载磁盘数目(默认 4)

-c: DG/CG 服务所使用的挂载磁盘数目(默认 4)

-d: 应用名(默认为 obtest)

注：脚本随安装文件提供，sh RebuildDir.sh -h 查看更多参数选择；

## 5. 启动与管理系統

在每台服务器上正确安装 kingwow 软件并且创建数据目录之后，就可以启动数据库服务。

### 5.1. 数据库服务启动命令

进入 kingwow 安装目录：cd ~/kingwow

#### 5.1.1. MG 启动命令

主 MG 启动命令格式：

```
bin/manEng -r 192.168.1.2:2500 -R 192.168.1.2:2500 -i eth0 -U 3 -u 3 -C 1
```

备 MG 启动命令格式：

```
bin/manEng -r 192.168.1.3:2500 -R 192.168.1.2:2500 -i eth0 -C 1
```

参数含义：

-r：所启动 MG 服务的 IP 地址和端口；

-R：系统主 MG 服务的 IP 地址和端口；

-i：绑定的网卡名；

-C：系统所属逻辑集群 ID，必须为数字；

-U：指定系统中 MG 服务的数量；

-u：指定系统中 TG 服务的数量；

#### 5.1.2. TG 启动命令

命令格式：

```
bin/transEng -r 192.168.1.2:2500 -p 2700 -m 2701 -i eth0 -C 1
```

**参数含义：**

- r：系统主 MG 服务的 IP 地址和端口；
- p：所启动 TG 服务的端口号；
- m：所启动 TG 服务的内部端口号；
- i：绑定的网卡名；
- C：系统中所属逻辑集群 ID，必须为数字；

### 5.1.3. DG 启动命令

命令格式：

```
bin/datEng -r 192.168.1.2:2500 -p 2600 -n obhdfs -i eth0 -C 1
```

**参数含义：**

- r：系统主 MG 服务的 IP 地址和端口；
- p：所启动 DG 服务的端口号；
- n：指定应用名，应当与建 DG 数据目录时所设定的应用名一致；
- i：绑定的网卡名；
- C：系统中所属逻辑集群 ID，必须为数字；

### 5.1.4. CG 启动命令

CG 启动命令格式：

```
bin/calEng -r 192.168.1.2:2500 -p 2800 -z 2880 -i eth0 -C 1
```

**参数含义：**

- r：系统主 MG 服务的 IP 地址和端口；
- p：所启动 DG 服务的端口号；
- z：所启动 CG 的 MySQL 协议服务端口号；
- i：绑定的网卡名；
- C：系统中所属逻辑集群 ID，必须为数字；

## 5.2. 单集群启动示例

以一个逻辑集群组成的 kingwow 数据库系统为例，集群号为 1，使用三台服务器，在



每台服务器上启动 MG、TG、CG 和 DG 服务，其中主 MG 所在的服务器上启动 CG 被设置为。指定主 MG 为 192.168.1.4:2500。

以 admin 用户登录所有服务器，在安装目录/home/admin/kingwow 下执行启动命令。

#### Server 1（主 MG 所在服务器）

```
bin/manEng -r 192.168.1.4:2500 -R 192.168.1.4:2500 -i eth0 -U 3 -u 3 -C 1
```

```
bin/transEng -r 192.168.1.4:2500 -p 2700 -m 2701 -i eth0 -C 1
```

```
bin/datEng -r 192.168.1.4:2500 -p 2600 -n obhdfs -i eth0 -C 1
```

```
bin/calEng -r 192.168.1.4:2500 -p 2800 -z 2880 -i eth0 -C 1
```

#### Server 2

```
bin/manEng -r 192.168.1.3:2500 -R 192.168.1.4:2500 -i eth0 -C 1
```

```
bin/transEng -r 192.168.1.4:2500 -p 2700 -m 2701 -i eth0 -C 1
```

```
bin/datEng -r 192.168.1.4:2500 -p 2600 -n obhdfs -i eth0 -C 1
```

```
bin/calEng -r 192.168.1.4:2500 -p 2800 -z 2880 -i eth0 -C 1
```

#### Server 3

```
bin/manEng -r 192.168.1.7:2500 -R 192.168.1.4:2500 -i eth0 -C 1
```

```
bin/transEng -r 192.168.1.4:2500 -p 2700 -m 2701 -i eth0 -C 1
```

```
bin/datEng -r 192.168.1.4:2500 -p 2600 -n obhdfs -i eth0 -C 1
```

```
bin/calEng -r 192.168.1.4:2500 -p 2800 -z 2880 -i eth0 -C 1
```

如果有其它服务器加入到集群中，可以在这些服务器上启动 DG 和 CG 服务，命令如下：

```
bin/datEng -r 192.168.1.4:2500 -p 2600 -n obhdfs -i eth0 -C 1
```

```
bin/calEng -r 192.168.1.4:2500 -p 2800 -z 2880 -i eth0 -C 1
```

注：MG 和 TG 已经达到设定数目；

在所有服务器上执行完毕命令后，在确认所有服务进程都已启动，在任意一台服务的安装目录下执行以下命令进行数据库系统的初始化：

```
bin/mg_admin -r 192.168.1.4 -p 2500 boot_strap
```

## 5.3. 多集群启动示例

以三个逻辑集群组成的 kingwow 数据库系统为例，集群号分别为 1、2、3，可以分别部署在三个机房中。每个逻辑集群使用两台服务器，启动 MG、TG、CG 和 DG 共五个服务，其中 MG/TG/ 部署于一台服务器，CG/DG 部署于另一台服务器，指定主 MG 为 182.119.80.62:8000。

以 admin 用户登录所有服务器，在安装目录/home/admin/kingwow 下执行启动命令。

#### 集群 1：

##### Server1

```
bin/manEng -r 182.119.80.62:2500 -R 182.119.80.62:2500 -i bond0 -U 3 -u 3 -C 1
```

```
bin/transEng -r 182.119.80.62: 2500 -p 8001 -m -i bond0 -C 1
```

##### Server2

```
bin/datEng -r 182.119.80.62: 2500 -p 8005 -n obhdfs -i bond0 -C 1
```

```
bin/calEng -r 182.119.80.62: 2500 -p 2800 -z 2880 -i bond0 -C 1
```

#### 集群 2：

##### Server1

```
bin/manEng -r 182.119.80.63: 2500 -R 182.119.80.62:2500 -i bond0 -C 2
```

```
bin/transEng -r 182.119.80.62: 2500 -p 2700 -m 2701 -i bond0 -C 2
```

##### Server2

```
bin/datEng -r 182.119.80.62: 2500 -p 2600 -n obhdfs -i bond0 -C 2
```

```
bin/calEng -r 182.119.80.62: 2500 -p 2800 -z 2880 -i bond0 -C 2
```

#### 集群 3：

##### Server1

```
bin/manEng -r 182.119.80.64:2500 -R 182.119.80.62:2500 -i bond0 -C 3
```

```
bin/transEng -r 182.119.80.62:2500 -p 2700 -m 2701 -i bond0 -C 3
```

##### Server2

```
bin/datEng -r 182.119.80.62:2500 -p 2600 -n obhdfs -i bond0 -C 3
```

```
bin/calEng -r 182.119.80.62:2500 -p 2800 -z 2880 -i bond0 -C 3
```

在所有服务器上执行完毕命令后，在确认所有服务进程都已启动，在任意一台服务的安装目录下执行以下命令进行数据库系统的初始化：

```
bin/mg_admin -r 182.119.80.62 -p 2500 boot_strap
```

## 5.4. 使用 MySQL 客户端

在数据库系统初始化完成后，可以执行以下命令连接 kingwow 数据库：

```
mysql -h192.168.1.7 -P2880 -uadmin -padmin
```

**参数含义：**

- h：CG 所在服务器的 IP 地址；
- P：启动 CG 服务命令-z 参数所指定的 MySQL 协议服务端口；
- u：kingwow 数据库用户名，admin 为预置超级用户；
- p：kingwow 数据库用户的密码，admin 用户的缺省密码为 admin