

# 数据库管理

**NSD DATABASE**

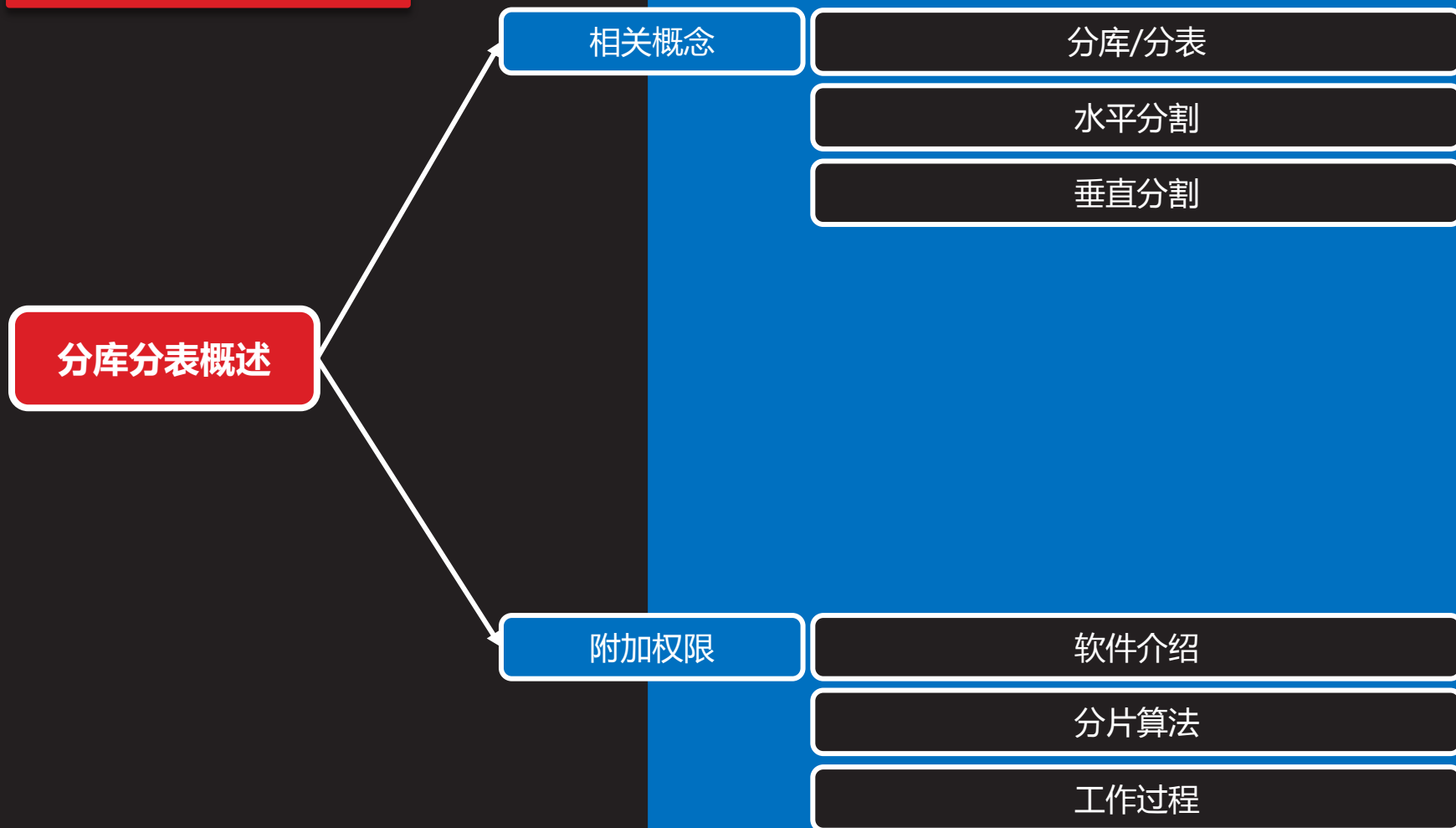
**DAY10**

# 内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	分库分表概述
	10:30 ~ 11:20	
	11:30 ~ 12:00	配置mycat
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



# 分库分表概述



# 相关概念

---

# 分库分表

- 什么是分库分表
  - 通过某种特定条件，将存放在一个数据库(主机)中的数据，分散存放到多个数据库(主机)中。
  - 已达到分散单台设备负载的效果，即分库分表
  - 数据的切分根据其切分规则的类型，分为2中切分模式
  - 垂直分割(纵向) 和 水平分割(横向)



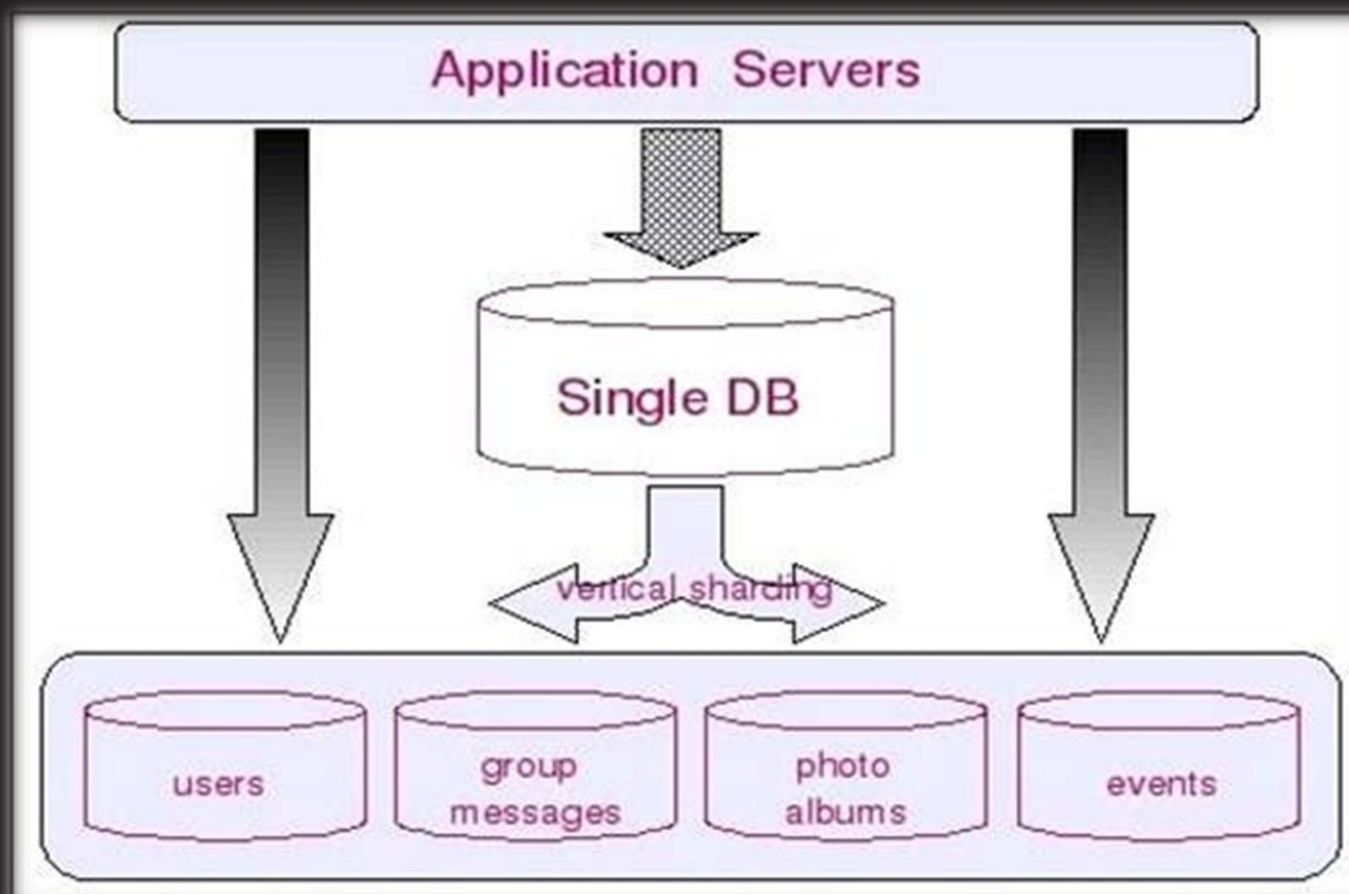
# 垂直分割

- 纵向切分
  - 把单一的表，拆分成多个表，并分散到不同的数据库(主机)上。
  - 一个数据库由多个表构成，每个表对应不同的业务，可以按照业务对表进行分类，将其分布到不同的数据库(主机)上，实现专库专用，让不同的库(主机)分担不同的业务。



## 垂直分割(续1)

- 垂直分割



# 水平分割

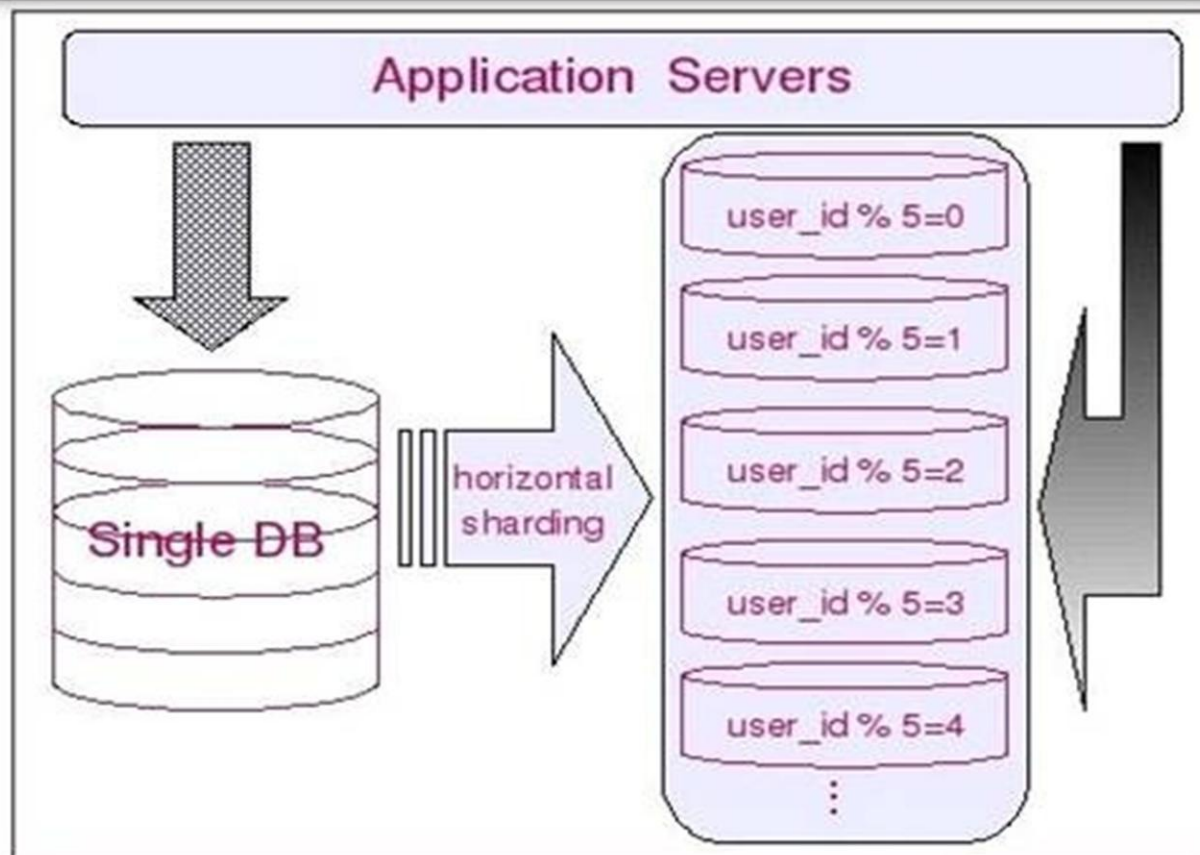
- 横向切分
  - 按照表中某个字段的某种规则，把向表中写入的记录分散到多个库(主机)中。
  - 简单来说，就是按照数据行切分，将表中的某些行存储到指定的数据库(主机)中。





# 水平分割(续1)

- 水平分割



# 软件介绍

---

# 软件介绍

- 基于Java的分布式数据库系统中间层，为高并发下的分布式提供解决方案
  - 支持JDBC形式连接
  - 支持MySQL、Oracle、Sqlserver、Mongodb等
  - 提供数据读写分离服务
  - 可以实现数据库服务器的高可用
  - 提供数据分片服务
  - 基于阿里巴巴Cobar进行研发的开源软件
  - 适合数据大量写入数据的存储需求



# 分片算法

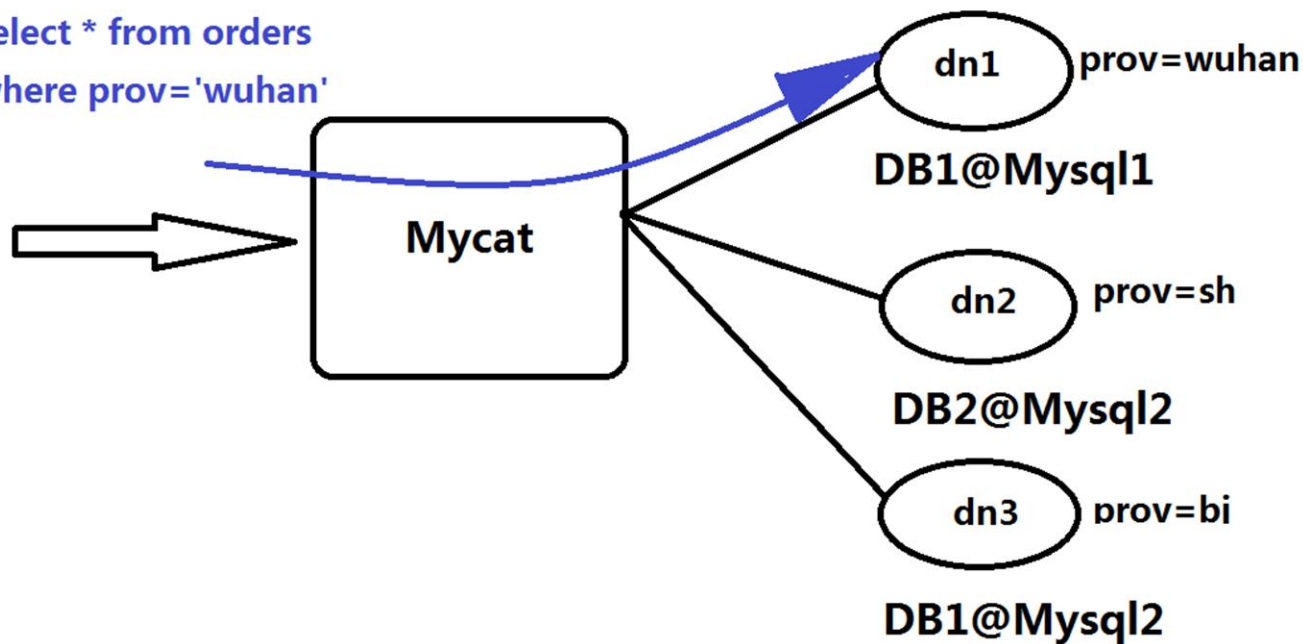
- mycat服务提供10种分片算法。
  - 1枚举法 sharding-by-intfile
  - 2固定分片hash算法 rule1
  - 3范围约定 auto-sharding-long
  - 4求模法 mod-log
  - 5日期列分区法 sharding-by-date
  - 6通配取模 sharding-by-pattern
  - 7ASCII码求模通配 sharding-by-prefixpattern
  - 8编程指定 sharding-by-substring
  - 9字符串拆分hash解析 sharding-by-stringhash
  - 10一致性hash sharding-by-murmur



# 工作过程

- 工作过程

`select * from orders  
where prov='wuhan'`



# 工作过程(续1)

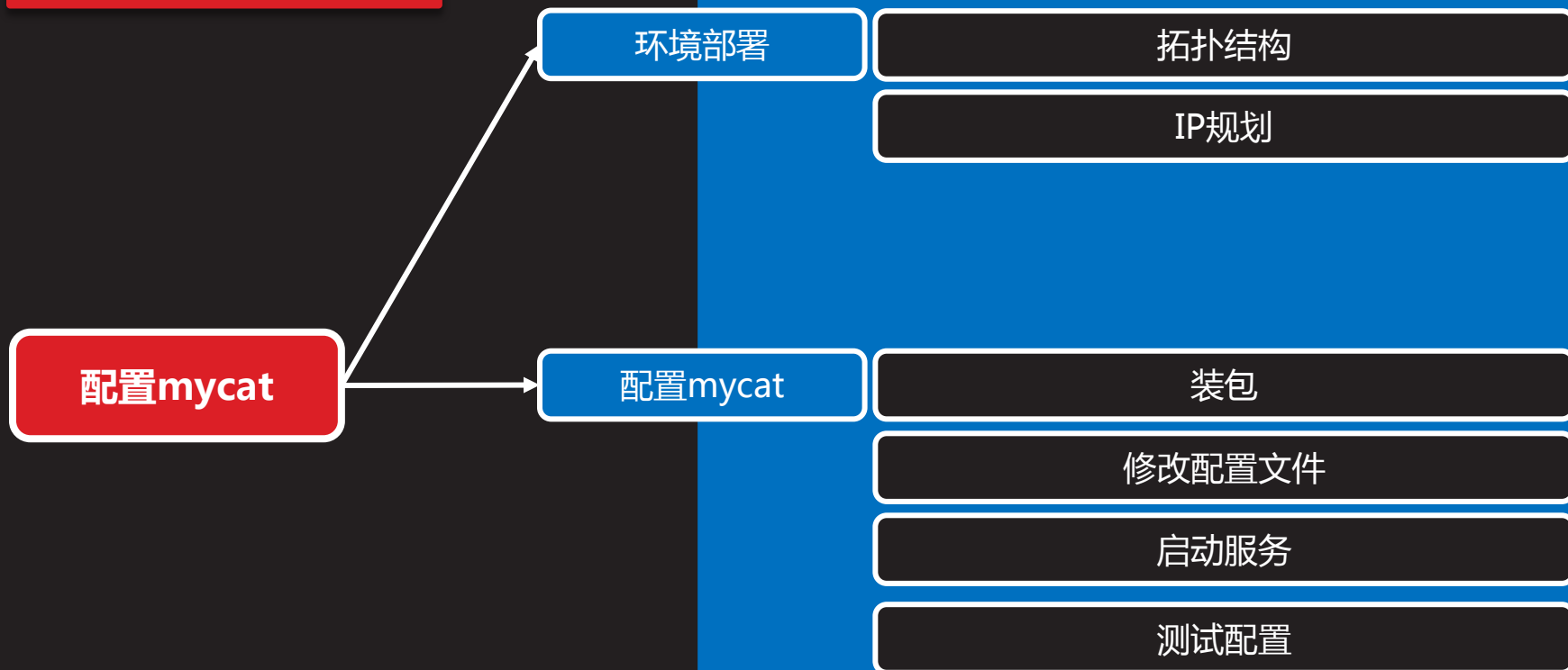
- 当Mycat收到一个SQL时
  - 会先解析这个SQL查找涉及到的表，然后看此表的定义
  - 如果有分片规则，则获取到SQL里分片字段的值，并匹配分片函数，得到该QL对应的分片列表
  - 然后将SQL发往这些分片去执行，最后收集和处理所有分片返回的结果数据，并输出到客户端

以select \* from Orders where prov=?语句为例，查到prov=wuhan，按照分片函数，wuhan返回dn1，于是SQL就发给了MySQL1，去取DB1上的查询结果，并返回给用户。

如果上述SQL改为elect \* from Orders where prov in ('wuhan','beijing')，那么，SQL就会发给ySQL1与MySQL2去执行，然后结果集合并后输出给用户。但通常业务中我们的SQL会有Order By 以及Limit翻页语法，此时就涉及到结果集在Mycat端的二次处理。



# 配置mycat



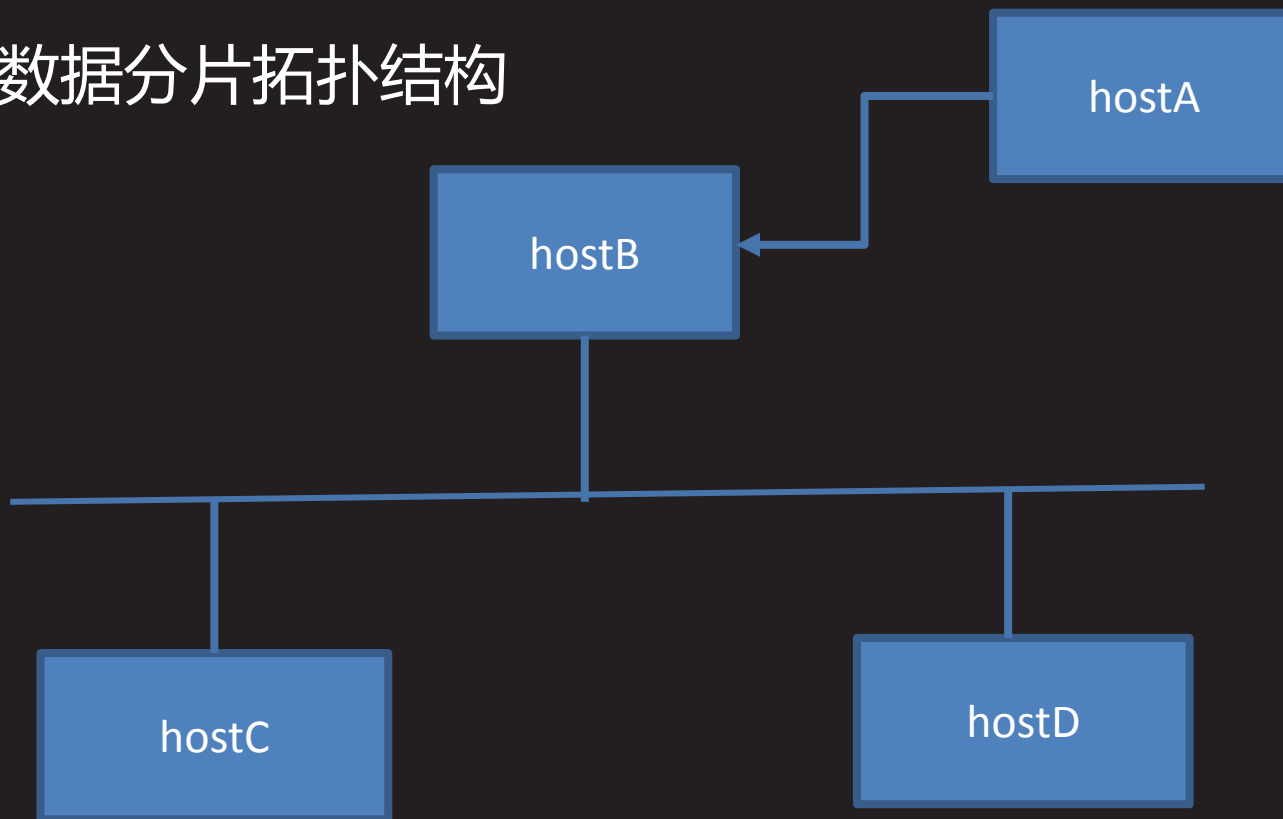
# 环境部署

---



# 拓扑结构

- 数据分片拓扑结构



# IP规划

- 具体配置如下：

拓扑名称	主机名	角色	数据库名	IP地址
hostA	client	客户端	无	192.168.4.254/24
hostB	Mycat	Mycat服务器	无	192.168.4.56/24
hostC	c1	数据库服务器	db1	192.168.4.55/24
hostD	c2	数据库服务器	db2	192.168.4.54/24



# 配置mycat

---

# 装包

- 安装JDK

```
[root@localhost ~]# rpm -qa | grep -i jdk #安装系统自带的即可
java-1.8.0-openjdk-1.8.0.65-3.b17.el7.x86_64
java-1.8.0-openjdk-headless-1.8.0.65-3.b17.el7.x86_64
```

- 安装mycat服务软件包

```
[root@localhost ~]# tar -zxf Mycat-server-1.4-beta-20150604171601-linux.tar.gz #免安装，解压后即可使用
[root@localhost ~]# mv mycat/ /usr/local/
```

```
[root@localhost ~]# ls /usr/local/mycat/
bin catlet conf lib logs version.txt
```



# 修改配置文件

- 目录结构说明
  - bin mycat命令 如 启动 停止 等
  - catlet 扩展功能
  - conf 配置文件
  - lib mycat使用的jar包 mycat是java开发的
  - log mycat启动日志和运行日志
  - wrapper.log mycat服务启动日志 , 启动有问题可以  
看这个日志的内容
  - mycat.log 记录 sql脚本执行后的具体报错内容



# 修改配置文件(续1)

- 重要配置文件说明
  - server.xml 设置连接mycat服务的账号、密码等
  - schema.xml 配置mycat使用的真实数据库和表
  - rule.xml 定义mycat分片规则



## 修改配置文件(续2)

- 修改配置文件/usr/local/mycat/conf/server.xml

```
<user name="test"> #连接mycat服务时使用的用户名 test
    <property name="password">test</property>
#使用test用户连接mycat用户时使用的密码
    <property
name="schemas">TESTDB</property> #连接上mycat服务后，可
以看到的库名多个时，使用逗号分隔（是逻辑上的库名）
    </user>
```

```
<user name="user">
    <property name="password">user</property>
    <property
name="schemas">TESTDB</property>
    <property name="readOnly">true</property>
#定义只读权限，使用定义的用户连接mycat服务后只有读记录的
权限
    </user>
```



# 修改配置文件(续3)

- 修改配置文件/usr/local/mycat/conf/schema.xml
  - 定义分片信息

```
<schema name="TESTDB" checkSQLSchema="false" sqlMaxLimit="100">
  #逻辑库名 要与server.xml定义的一样

  <table name="travelrecord" dataNode="dn1,dn2" rule="auto-sharding-long" />
  #定义分片的表

  <table name="company" primaryKey="ID" type="global" dataNode="dn1,dn2" />
  #定义分片的表

  <table name="goods" primaryKey="ID" type="global" dataNode="dn1,dn2" />
  #定义分片的表

  <table name="hotnews" primaryKey="ID" dataNode="dn1,dn2" rule="mod-long" />
  #定义分片的表

  <table name="employee" primaryKey="ID" dataNode="dn1,dn2" rule="sharding-by-intfile" />
  #定义分片的表

  <table name="customer" primaryKey="ID" dataNode="dn1,dn2" rule="sharding-by-intfile" />
  #定义分片的表
</schema>
```





# 修改配置文件(续4)

- 修改配置文件/usr/local/mycat/conf/schema.xml
  - 定义分片信息

```
<dataNode name="dn1" dataHost="c1" database="db1" />
#定义分片使用的库, 所在的物理主机, 真正存储数据的db1库在物理主机c1上

<dataNode name="dn2" dataHost="c2" database="db2" />
#定义分片使用的库, 所在的物理主机, 真正存储数据的db2库在物理主机c2上

#指定c1名称主机对应的ip地址
<dataHost name="c1" maxCon="1000" minCon="10" balance="0"
    writeType="0" dbType="mysql" dbDriver="native" >
    <heartbeat>select user()</heartbeat>
    <writeHost host="hostM1" url="192.168.4.55:3306" user="admin"
        password="123456"> #访问数据时 mycat服务连接数据库服务器时使用的用户名和密码
    </writeHost>
</dataHost>

#指定c2名称主机对应的ip地址
<dataHost name="c2" maxCon="1000" minCon="10" balance="0"
    writeType="0" dbType="mysql" dbDriver="native" >
    <heartbeat>select user()</heartbeat>
    <writeHost host="hostM2" url="192.168.4.54:3306" user="admin"
        password="123456"> #访问数据时 mycat服务连接数据库服务器时使用的用户名和密码
    </writeHost>
</dataHost>
```



# 修改配置文件(续5)

- 修改数据库服务器配置文件
  - 添加对应设置后重启mysqld服务
  - 添加授权用户

```
Vim /etc/my.cnf
[mysqld]
.....
.....
lower_case_table_names = 1 #表名区分字母大小写

[root@localhost ~]# systemctl restart mysqld

//mycat 连接数据库服务器用户
MySQL> grant all on *.* to admin@"%" identified by "123456";
```



# 启动服务

- 启动服务
  - 指定java路径、添加PATH路径、启动服务

```
[root@localhost ~]# sed -n '4,5p' /usr/local/mycat/conf/wrapper.conf
# Java Application
wrapper.java.command=java
[root@localhost ~]#
[root@localhost ~]# which java
/usr/bin/java

[root@localhost ~]# echo "export PATH=/usr/local/mycat/bin" >> /etc/profile
[root@localhost ~]# source /etc/profile

[root@localhost ~]# mycat --help
Usage: /usr/local/mycat/bin/mycat { console | start | stop | restart | status | dump }
[root@localhost ~]#

[root@localhost ~]# mycat start
Starting Mycat-server...
[root@localhost ~]#
[root@localhost ~]# netstat -utnlp | grep :8066
tcp6      0      0 :::8066          :::*              LISTEN      4524/java
[root@localhost ~]#
```



# 测试配置

- 在客户端连接mycat服务器
  - mysql -hmycat服务器ip地址 -P端口 -u用户 -p密码

```
[root@room9pc17 ~]# mysql -h192.168.4.56 -P8066 -utest -ptest
MySQL [(none)]> show databases;
```

```
+-----+
| DATABASE |
```

```
+-----+
| TESTDB |
```

```
+-----+
1 row in set (0.00 sec)
```

```
root@room9pc17 ~]# mysql -h192.168.4.56 -P8066 -utest -ptest
MySQL [(none)]> show databases;
```

```
DATABASE |
```

```
TESTDB |
```

```
1 row in set (0.00 sec)
```

```
MySQL [(none)]> use TESTDB;
```

```
MySQL [TESTDB]> show tables;
```

```
Tables in TESTDB |
```

```
company
customer
employee
goods
hotnews
travelrecord
```

```
6 rows in set (0.00 sec)
```



# 案例1：搭建mycat 分片服务器

具体要求如下：

- 1
- 2
- 3
- 4

