**Afternoon：**

1. 分库分表概述

**-What’s分库分表**

将存放在一个数据库服务器中的数据，按特定方式拆分，分散存放到多个数据库服务器中。达到分散单台设备负载的效果。

**-垂直分割**

将单个表，拆分成多个表，分散到不同的数据库；

将单个数据库的多个表进行分类，按业务分散到不同数据库。

**-水平分割**

按照表字段的某种规则，把表记录按行切分，分散到多个数据库中。

1. Mycat软件介绍

**-Mycat功能**

基于java，为高并发环境的分布式访问提供解决方案：

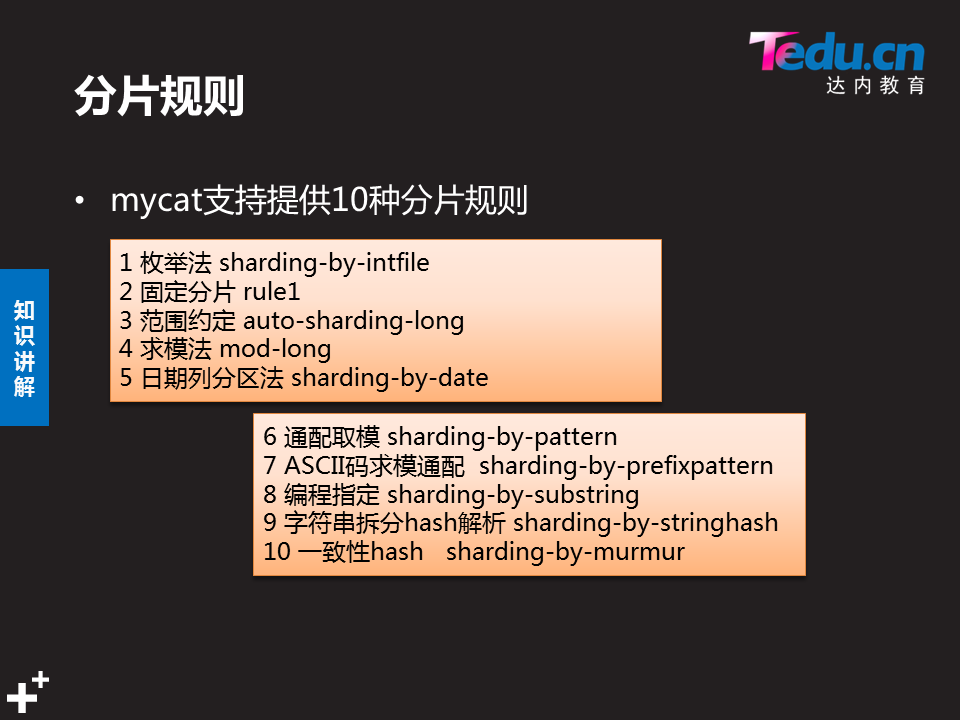
-提供数据读写分离服务

-数据库服务器高可用集群

-提供数据分片服务

-适合数据大规模写入的存储需求，读效率低。

**-分片规则(熟记)**



**-工作过程**

Mycat收到SQL查询 --> 分析涉及的表 --> 查看表是否有分片规则，获取分片列表 --> 将SQL发往分片执行 --> 收集和处理所有数据，返回客户端

3、部署Mycat服务器

**步骤1：装包**

###检查JDK环境###

[root@host56 ~]# rpm -qa | grep -i jdk;

java-1.8.0-openjdk

java-1.8.0-openjdk-headless

###安装Mycat软件包(无需编译)###

[root@host56 ~]# tar -zxf Mycat-server-1.4-...-linux.tar.gz

[root@host56 ~]# mv mycat/ /usr/local/

[root@host56 ~]# ls /usr/local/mycat/

bin //mycat命令

catlet //扩展功能，默认为空

conf //.xml配置文件，.txt算法的调用文件

lib //mycat服务使用的jar包

log //日志。Wrapper.log启动日志，mycat.log错误日志

**步骤2：修改mycat配置文件**

1）重要配置文件说明

**server.xml** #设置连接mycat的帐号信息

**schema.xml** #配置mycat的真实库表

**rule.xml** #定义mycat分片规则及算法，无需修改

1. 查看分片规则与算法

[root@host56 ~]# vim /usr/local/mycat/conf/rule.xml

<tableRule name="sharding-by-intfile">

<rule>

<columns>sharding\_id</columns>

<algorithm>hash-int</algorithm>

</rule>

</tableRule>

#使用此分片规则的表，建表时必须要有sharding\_id字段

#分片规则对应hash-int算法

<function name="hash-int"

class="org.opencloudb.route.function.PartitionByFileMap">

<property name="mapFile">partition-hash-int.txt</property>

</function>

#指定hash-int算法的调用文件xx.txt

[root@host56 ~]# cat /usr/local/mycat/conf/partition-hash-int.txt

10000=0

10010=1

#枚举法：sharding\_id字段值只能取10000/10010，分别把记录存在dn1/dn2

3）配置server.xml

[root@host56 ~]# vim /usr/local/mycat/conf/server.xml

<user name="admin">

<property name="password">123456</property>

<property name="schemas">TESTDB</property>

</user>

#user name客户端连接mycat的用户名，password客户端密码

#schemas客户端连接mycat后，看到的虚拟库名

4）配置schema.xml  
 [root@host56 ~]# vim /usr/local/mycat/conf/schema.xml

<schema name="TESTDB" checkSQL... ...>

#必须与server.xml定义的虚拟库名一致

<table name="travelrecord" dataNode="dn1,dn2"

rule="auto-sharding-long" />

<table ... ... />

#name定义表名，dataNode定义数据节点(必须dn开头)，rule定义分片规则

<dataNode name="dn1" dataHost="mysql54" database="db1" />

<dataNode name="dn2" dataHost="mysql55" database="db2" />

#name指定数据节点，dataHost数据库服务器标识，database**真实数据库名**

<dataHost name="mysql54" maxCon="1000" ... ...>

<heartbeat>select user()</heartbeat>

<writeHost host="hostM1" url="192.168.4.54:3306" user="gjq"

password="123456">

</writeHost>

#指定数据库服务器IP:端口，授权用户、密码

</dataHost>

<dataHost name="mysql55" maxCon="1000" ... ...>

... ...

</dataHost>

**步骤3：修改mysql服务器配置**

mysql> grant all on \*.\* to gjq@"%" identified by "123456";

[root@mysqlxx ~]# vim /etc/my.cnf

lower\_case\_table\_names=1

[root@mysqlxx ~]# systemctl restart mysqld

#mysql54、mysql55服务器授权用户，配置成**表名忽略大小写**

**步骤4：启动mycat服务**

[root@host56 ~]# /usr/local/mycat/bin/mycat start

[root@host56 ~]# netstat -anptul | grep 8066

#如果启动失败，查看wrapper.log日志

**步骤5：客户端测试**

[root@client50 ~]# mysql -uadmin -p123456 -P8066 -h'192.168.4.56'

mysql> use TESTDB;

mysql> create table employee(

-> id int primary key auto\_increment,

-> name char(10),

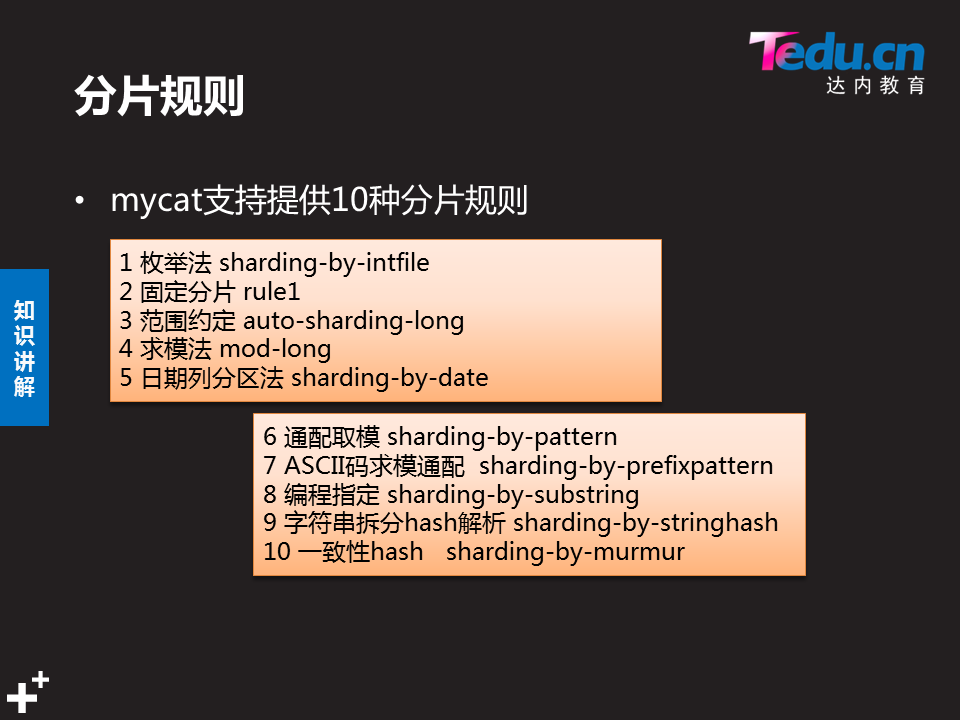
-> pay float(7,2),

-> sharding\_id int );

mysql> insert into employee(id,name,pay,sharding\_id)

-> values(1,"bob",9000,10000);

1. 【拓展】分片规则详解



|  |  |
| --- | --- |
| 枚举法 | 算法：hash-int  通过partition-hash-int.txt指定user\_id的**枚举值与数据节点的关系** |
| 固定分片 | 算法：func1  通过partitionCount指定**分片个数**列表，partitionLength**分片范围**列表  Sum(count[i]\*length[i])=1024，按照user\_id所在范围进行分片 |
| 范围约定 | 算法：rang-log  通过autopartition-long.txt指定**分片数据节点对应**user\_id**范围** |
| 求模法 | 算法：mod-long  通过count指定数据**节点数**，将user\_id与count求模控制分片 |
| 日期列分区法 |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |