

region部分设计报告

姓名：朱雨珂

学号：3190103312

个人分工

主要写了region除了副本备份之外的部分，包括与客户端、主服务器端的连接、命令传输、minisql连接等。

主要功能

连接客户端Client

- 通过socket与客户端连接
- 接收客户端sql查询信息
- 将sql处理结果返回给客户端

连接主服务器Master

- 通过socket与主服务器连接
- 在启动时，将自身所拥有的表信息、索引信息发给主服务器
- 在用户更改数据表之后（create 和 drop），将更新消息返回给主服务器
- 接受主服务器的容错容灾指令，进行容错容灾处理
- 重启后，接受主服务器的恢复指令，进行恢复处理

连接minisql


















- 连接minisql，处理客户端传来的sql语句，并返回处理结果给客户端

响应容错容灾命令

这部分和组内同学商量共同设计：

- 实现对master容错容灾命令的响应，从备份节点迁移文件
- 实现对master恢复命令的响应，删除从节点本地及在备份节点相应的文件

文件结构

- ▼  miniSQL
 - >  BUFFERMANAGER
 - >  CATALOGMANAGER
 - >  FILEMANAGER
 - >  INDEXMANAGER
 - >  lexer
 - >  RECORDMANAGER
 -  API
 -  Interpreter
 - ▼  RegionController
 - ▼  SocketController
 - >  ClientSocketController.java
 - >  FileController.java
 -  MasterSocketController
 -  RegionController
 -  zkServiceController
 -  RegionServer
-

RegionServer开启RegionController——

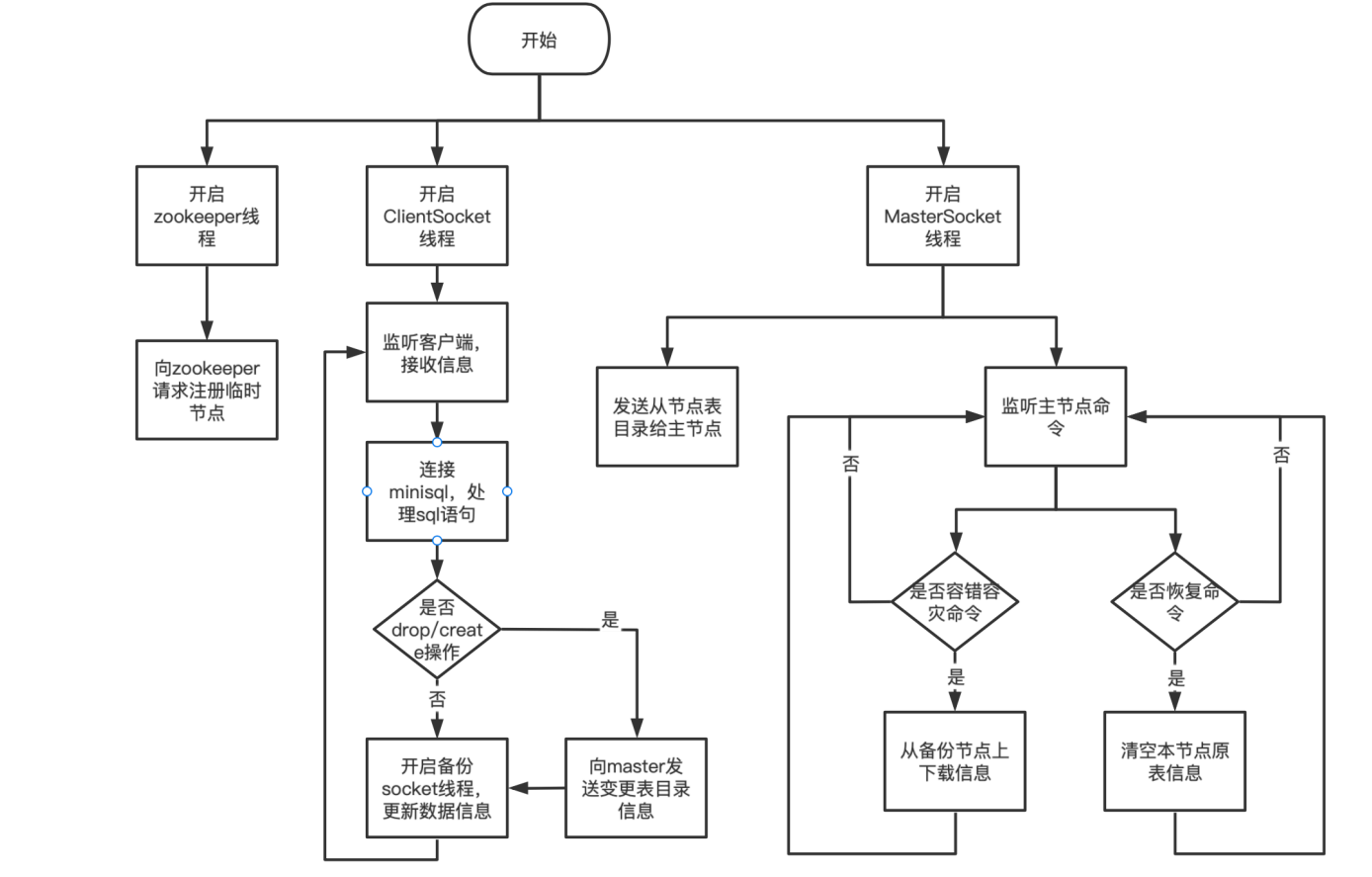
- miniSQL是包装好的miniSQL，通过Interpreter进行连接
- SocketController包处理所有Socket线程连接，收发信息

- ClientSocketController处理和client之间的socket连接、数据传输
- FileController处理和副本备份节点之间的socket连接、数据传输
- MasterSocketController处理和master之间的socket连接、数据传输
- ZookeeperController管理Zookeeper操作

类图如下：



运行流程



数据传输

通过socket分别连接master、client、副本节点进行数据连接传输，通信协议如下：

用途	协议格式	返回格式	含义
客户端与主服务器通信	CC{tableName}	MC{tableName} {ip}	客户端读入新建表或索引的命令，客户端向主服务器请求ip地址，主服务器返回ip地址。 第二个字符C代表此时向Master服务器询问ip地址的是一个create命令。 Master服务器返回时tablename与ip之间用空格隔开。
	CN{tableName}	MN{tableName} {ip}	客户端读入非新建命令，客户端向主服务器请求ip地址，主服务器返回ip地址。 第二个字符N代表此时向Master服务器询问ip地址的不是一个create命令。 Master服务器返回时tablename与ip之间用空格隔开。
客户端与从服务器通信	Cl{sqlCommand}	Rl{information}	获得ip后客户端向从服务器发送原sql语句，从服务器将处理结果返回客户端
从服务器与主服务器通信	M1	R1{table_list}	从服务器上线，向主服务器发送该region的表目录，目录中table_name用空格间隔
	M2	R2{table_name} {operation}	从服务器如果create、drop table，会向主服务器传输消息（operation为add，delete）
	M3@ip	R3	主服务器向从服务器发送容错容灾指令，从服务器获取ip文件夹下的所有文件
	M4	R4	主服务器向从服务器发送恢复指令

minisql指令处理

在接收到客户端的sql指令之后，region会将其传给minisql进行指令处理：

```
System.out.println("client_info>>> " + sql); //客户端信息sql
sql = sql + "\n";
InputStream is = new ByteArrayInputStream(sql.getBytes());
BufferedReader buf = new BufferedReader(new InputStreamReader(is)); //转化成
BufferedReader格式
String result = Interpreter.Parsing(buf); //调用minisql接口，返回处理结果result
```

result数据格式为 {operation}{object}{object_name} ——

如果操作为 create 或 drop，即对表目录、索引目录进行了更改，则要传递更改信息给master，并调用副本节点连接线程，将数据更新到副本节点上；

如果是 insert 或 delete 指令，则是对于表文件、索引文件内容进行了修改，不用向master反馈，但同样需要调用副本节点连接线程，将数据更新到副本节点上。

最后将结果原样返回给客户端。

容错容灾设计

当从节点region1突然下线，此时master主节点会检测到region1的下线，将其从在线从节点表中删除，然后挑选一个从节点region2来转移region1的数据，并发送 容错容灾指令 给region2——格式为 M3@ip%name%name。

region2接收到 容错容灾指令 后，会通过分解命令中的 `ip`、`name`，去备份节点寻找相应的表文件、索引文件、目录文件，并将文件备份到region2。

region1重新上线后，会向主节点发送原有的表目录，主节点master收到消息，比对得知region1是重新上线的节点后，会向region1发送 恢复指令，格式为 `M4`。region1收到 恢复指令 后，会删除本地的所有表文件、索引文件、目录文件，同时调用备份节点连接线程，删除备份节点上自己文件夹下的所有文件。