

# Region Server 模块详细设计

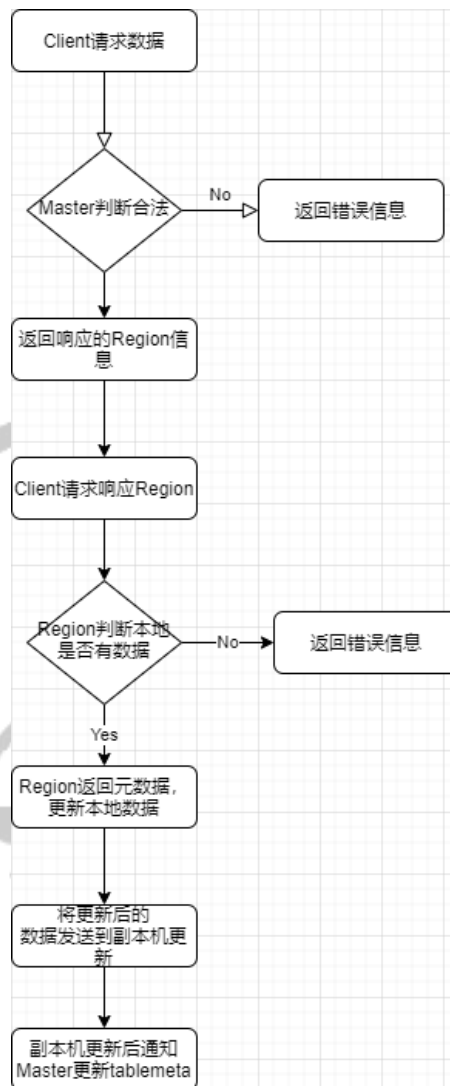
3180106074 王绍兴

## 1. Region Server Manager 任务概述

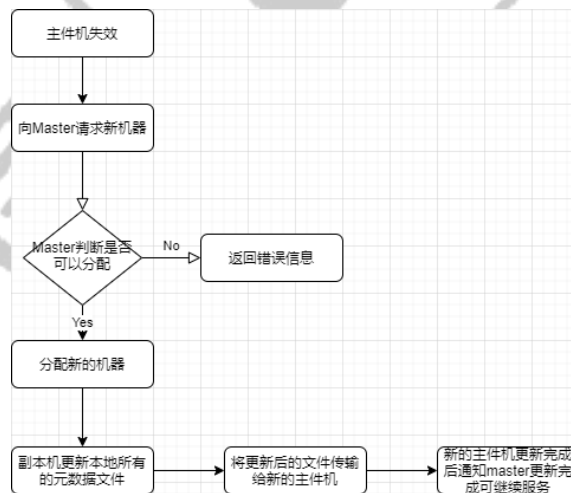
本模块是 Zookeeper 架构中 Region 服务机响应 Master 指令并与各 Client 进行元数据传输 (Meta Data)，管理 Region 本地元数据的模块。需要完成的功能如下所示：

- 管理和维护本地 Meta Table，管理与本机结对的副本机状态，其中对于副本机需要管理其机上的数据中 Locatedname 保持与主机一致，主机需要在与 Master，Client 的数据交换中实时更新与其结对的副本机上的数据。
- 管理和维护 Region 发生的变化，如副件机失效，主件机应从 Master 请求新的副本机，并将其中的数据拷贝过去，如主件机失效，副本机应将本地数据更新后请求 Master 分配新的主件机供其配对。
- 对于 Client 的请求，进行响应，返回请求的元数据，并对 Client 发送的请求进行储存和备份
- Region Server 模块工作流程图如下

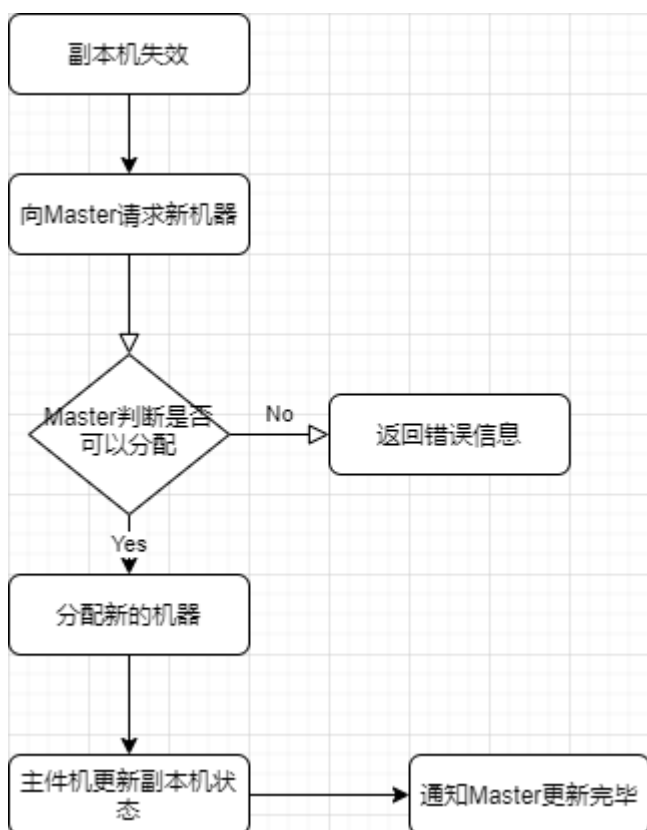
Client 请求流程：



主件机失效流程:



副本机失效流程:



为了解决传输过程中和储存中数据结构不一致的情况，采用 UtilCopy 封装类完成数据的复制。

## 2. 功能说明

### 2.1 数据存储

- 将 Client 发送的数据实时更新到 Region 本地
- 存储 Master 发送的本机状态和对偶机状态并实时更新

### 2.2 数据传输

- 对于 Server 端所有操作后通知 Master 更新元数据 Meta，通知副本机实时更新数据
- 对于 Client 端，将 Client 的请求查询到相应数据后返回给 Client

### 2.3 备份管理

- 对于实时更新的数据需要对两台机器进行（包括但不限于插入删除创建）

### 2.4 容错容灾

- 副本机主件机时刻保持数据更新相同，同时在主件机失效时副本机需要

更新自己的状态以保持跟新的主件机一致。

### 3. 重要函数

- QueryTableResponse(QueryTableRequest);
- NotifyStateChange(NotifyState);
- NotifyTableChange(NotifyTableChangeRequest);
- exelTablecopy();

### 4. 相关工具类

#### 4.1 FileServer

- 将输入的文件名列表查询后返回
- 更新元数据

#### 4.2 UtilCopy

- 将元数据复制给传输的数据
- 将传输的数据赋值给元数据

### 5. 开发心得

本次开发我主要负责的是协助组长完成 Server 架构的搭建以及相关工具类的开发。

Region 做为逻辑上的中间类，既要与 Master 协同要与 Client 沟通，所以开发时我跟组长跟与开发 Client 的组员都需要很深入的交流，了解他们的整体逻辑和需要调用的接口，在这个过程中沟通能力得到了增长，虽然在开发的过程中会有一些意见的不同，但开发完成后对于整个项目的熟悉程度让我有种豁然开朗的感觉。同时也让我更了解了文档的重要性，这次项目虽然事务繁杂但从没在开发过程中有过迷茫的感觉，这得益于我们一开始制定的详尽的文档。

技术方面，了解了 Zookeeper 的开发，了解了 Thrift 的使用开发方法，同时在与老师的沟通中让我对分布式以及数据库都有了新的理解，这对以后的开发学习生活都大有帮助。