

Master Server 模块详细设计

3180106071 刘轩铭

1. ZooKeeper Server 模块设计

本模块主要对 ZooKeeper 集群进行管理，需要完成的功能如下所示：

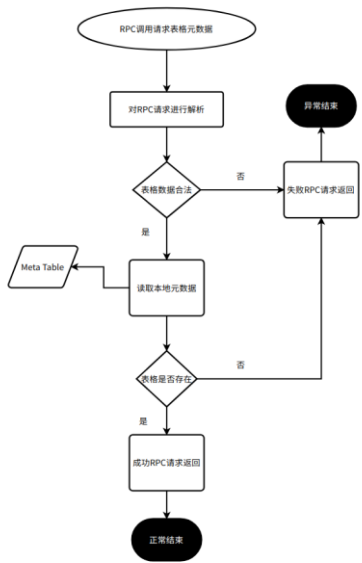
- 安装 ZooKeeper 服务端
- 根据服务器的注册信息获取该服务器的地址和主机名
- 以直接确定或选举等策略选取集群中的一定量机器作为 Region Server，另外一定量机器为对应的副本机。
- 监控集群的生命状态，设置回调函数处理服务机发现、服务机失效等事件。

2. Master Server Manager 模块设计

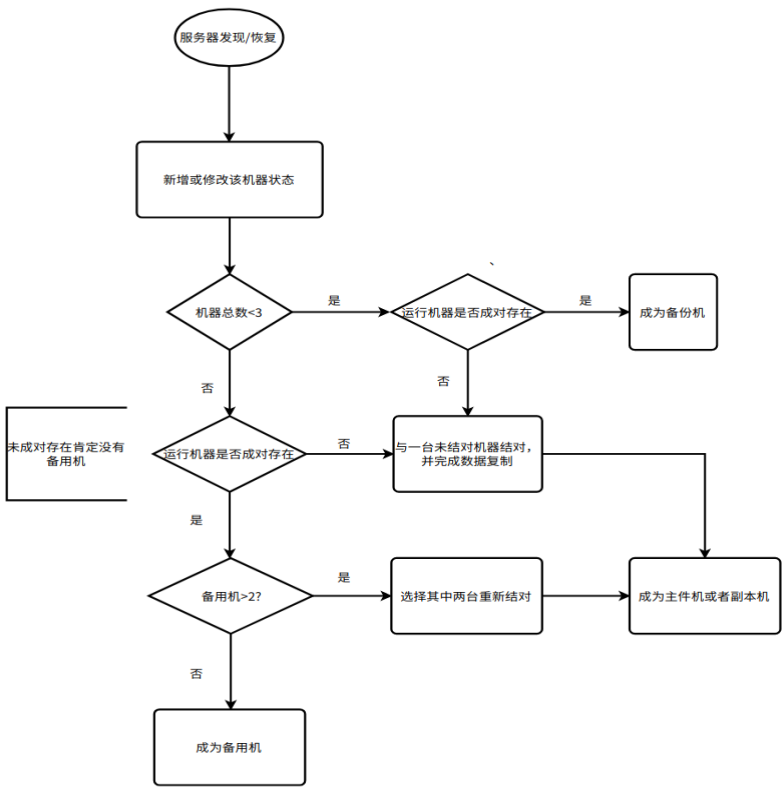
本模块是 Master 服务机上对各 Region Server 进行元数据（Meta Data）进行管理的模块。需要完成的功能如下所示：

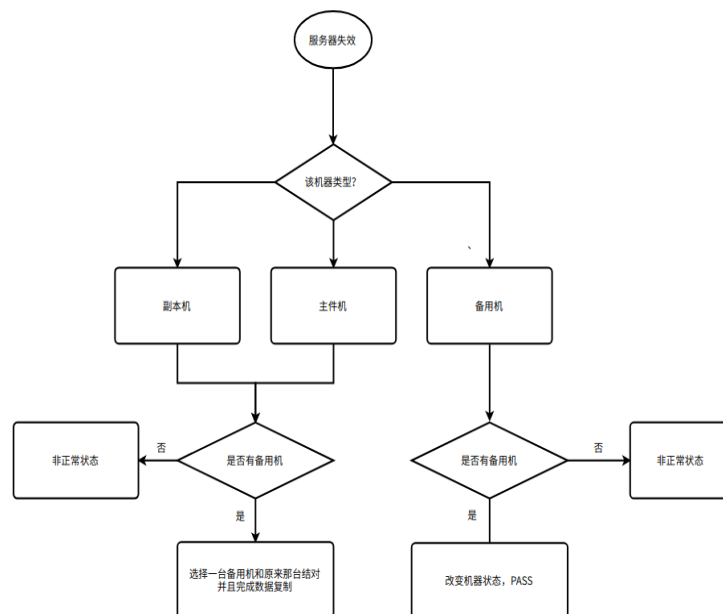
- 管理和维护 Meta Table，其中管理和存储每张表格的名字、版本号、存在的 Region Server 及其地址。
- 管理和维护各个 Region Server 的角色和状态，明确哪些是主本机，哪些是副本机。
- 指示主从复制等事件的发生。
- 接收 Client 发送的请求，根据需要的表格返回其所在 Region Server 的位置等信息。

以下为该功能的流程图。

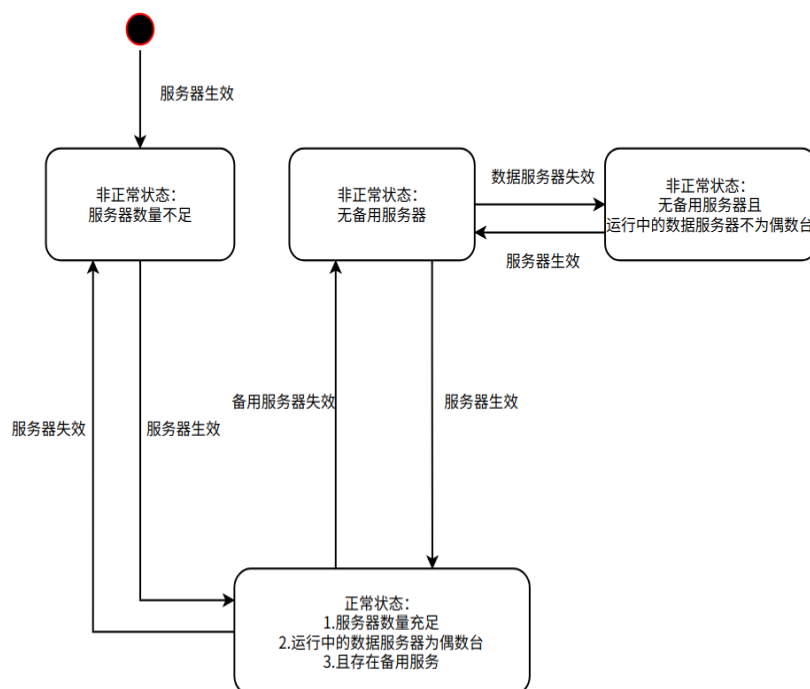


- 制定服务器发现和失效的处理流程，其流程分别如下所示：





为了完成服务，还需要对集群的状态进行管理，规定集群在一定额状态下可以完成服务，其他时候则拒绝服务，该状态机如下所示：



3. 开发心得

本次开发我负责的主要是整个系统的架构，以及 Master Server 部分的开发。

作为组长，除了需要开发以外，我还需要协调各个同学之间的工作，并且统筹开发的进度等，所以整体压力还是较大的。不过，圆满完成任务之后，还是感觉收获较多的。

首先是技术方面，通过本次开发，我学会了一些之前没有接触的技术场景的开发方法，比如 Curator 联接 ZooKeeper，以及 Thrift 的开发方法等。这些在之后的工作学习中都是对自己大有裨益的。

其次是在压力调整和时间轴制定上。由于我们组采取了较为合理的时间轴，同时也进行了一周一会的管理制度，所以还是在规定时间内完成了开发任务。同时，我也学会了如何协调同学的工作，并且更好的对团队创造价值。

答辩中，老师的点评也让我明白了系统作业还存在着一定的改进空间，比如系统的整体架构上。所以，之后有需要还可以进行进一步的完善和总结。