当一个新的子表产生时,主服务器通过一个加载命令将其分配给一个空间足够的子表服务器。创建新表、表合并以及较大子表的分裂都会产生一个或多个新子表。对于前面两种,主服务器会自动检测到,因为这两个操作是由主服务器发起的,而较大子表的分裂是由子服务发起并完成的,所以主服务器并不能自动检测到,因此在分割完成之后子服务器需要向主服务发出一个通知。由于系统设计之初就要求能达到良好的扩展性,所以主服务器必须对子表服务器的状态进行监控,以便及时检测到服务器的加入或撤销。Bigtable 中主服务器对子表服务器的监控是通过 Chubby 来完成的,子表服务器在初始化时都会从 Chubby 中得到一个独占锁。通过这种方式所有的子表服务器基本信息被保存在 Chubby 中一个称为服务器目录 (server directory) 的特殊目录之中。主服务器通过检测这个目录就可以随时获取最新的子表服务器信息,包括目前活跃的子表服务器,以及每个子表服务器上现已分配的子表。对于每个具体的子表服务器,主服务器会定期向其询问独占锁的状态。如果子表服务器的锁丢失或没有回应,则此时可能有两种情况,要么是 Chubby 出现了问题(虽然这种概率很小,但的确存在,Google 自己也做过相关测试),要么是子表服务器自身出现了问题。对此主服务器首先自己尝试获取这个独占锁,如果失败说明 Chubby 服务出现问题,需等待 Chubby 服务的恢复。如果成功则说明 Chubby 服务良好而子表服务器本身出现了问题。这种情况下主服务器会中止这个子表服务器并将其上的子表全部移至其他子表服务器。当在状态监测时发现某个子表服务器上负载过重时,主服务器会自动对其进行负载均衡操作。

基于系统出现故障是一种常态的设计理念(Google 几乎所有的产品都是基于这个设计理念),每个主服务器被设定了一个会话时间的限制。当某个主服务器到时退出后,管理系统就会指定一个新的主服务器,这个主服务器的启动需要经历以下 4 个步骤。

- (1) 从 Chubby 中获取一个独占锁,确保同一时间只有一个主服务器。

- (2) 扫描服务器目录,发现目前活跃的子表服务器。

- (3) 与所有的活跃子表服务器取得联系以便了解所有子表的分配情况。

- (4) 通过扫描元数据表(metadata table),发现未分配的子表并将其分配到合适的子表服务器。如果元数据表未分配,则首先需要将根子表(root tablet)加入未分配的子表中。由于根子表保存了其他所有元数据子表的信息,确保了扫描能够发现所有未分配的子表。

在成功完成以上 4 个步骤后主服务器就可以正常运行了。