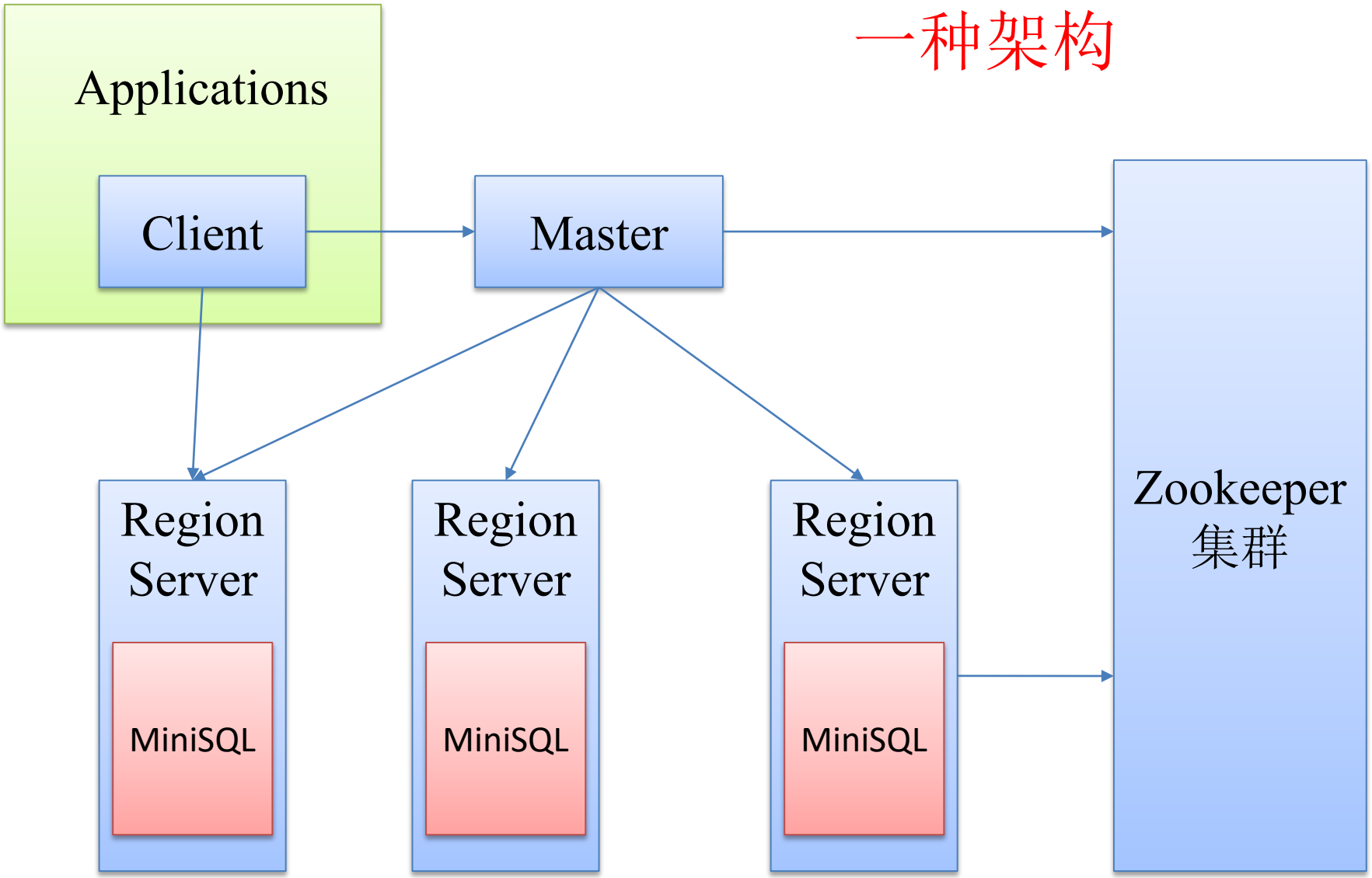
大规模信息系统构建技术导论

分布式MiniSQL

浙江大学计算机学院
鲁伟明

整体架构


一种架构
Applications
Client
Master
Region Server
MiniSQL
Region Server
MiniSQL
Region Server
MiniSQL
Zookeeper 集群

Zookeeper集群

- 集群管理
  - RegionServer管理
    - Master和RegionServer监控Zookeeper中的目录
    - 知道Region集群中有哪些服务器；
    - 当RegionServer崩溃时，通过Zookeeper可以通知Master，Master做出适当的调整（容错容灾）
  - 小数据存储
    - 例如Hbase在，有个meta table，存储了其他表格的信息

Master

- 负责管理和维护表的分区信息（或者分布信息）等元数据信息
- 维护Region服务器列表
- 分配Region（简单起见，Region可以直接对应一个Table，而不需要进行切分，也不需要分裂和合并）
- 实现不同Region服务器之间的负载均衡
- 管理用户对表的增加、删除、修改、查询等操作
- 对发生故障失效的Region服务器上的Region进行迁移

RegionServer

- Region服务器负责存储和维护分配给自己的Region，处理来自客户端的读写请求
- 简单起见，RegionServer利用MiniSQL来管理Region，负责MiniSQL的启动和管理，和Client的通信
- 进一步，可以有缓存机制等

Client

- 客户端并不是直接从Master主服务器上读取数据，而是在获得Region的存储位置信息后，直接从Region服务器上读取数据
- 客户端可以不依赖Master，可以通过Zookeeper来获得Region位置信息（需要设计一套定位机制）或者从Master中获得，大多数客户端甚至从来不和Master通信，这种设计方式使得Master负载很小
- 为减轻Master负担，在客户端可以有缓存，保存Table定位信息

副本维护

- 可以采用主从复制策略，选择其中一个表为主副本，负责副本的复制操作
- 简单起见，可以同时发请求，要求都写完才算完成写操作

负载均衡

- Master检测到一台RegionServer繁忙时，Master会将其中的某些Region重新分配到其他的RegionServer中。
- RegionServer之间需要传输数据，等负载均衡后修改Table定位信息。（何时传输？避免热点更热）

容错容灾

- 当某个RegionServer失效后，Master会将其中的Region重新分配到其他的RegionServer中。

谢 谢！