QTSDB 设计安案

日期	版本	作者	**Email**	备注
2018/10/25	0.1	刘伟	liuwei3-s@360.cn	创建文档 概要设计

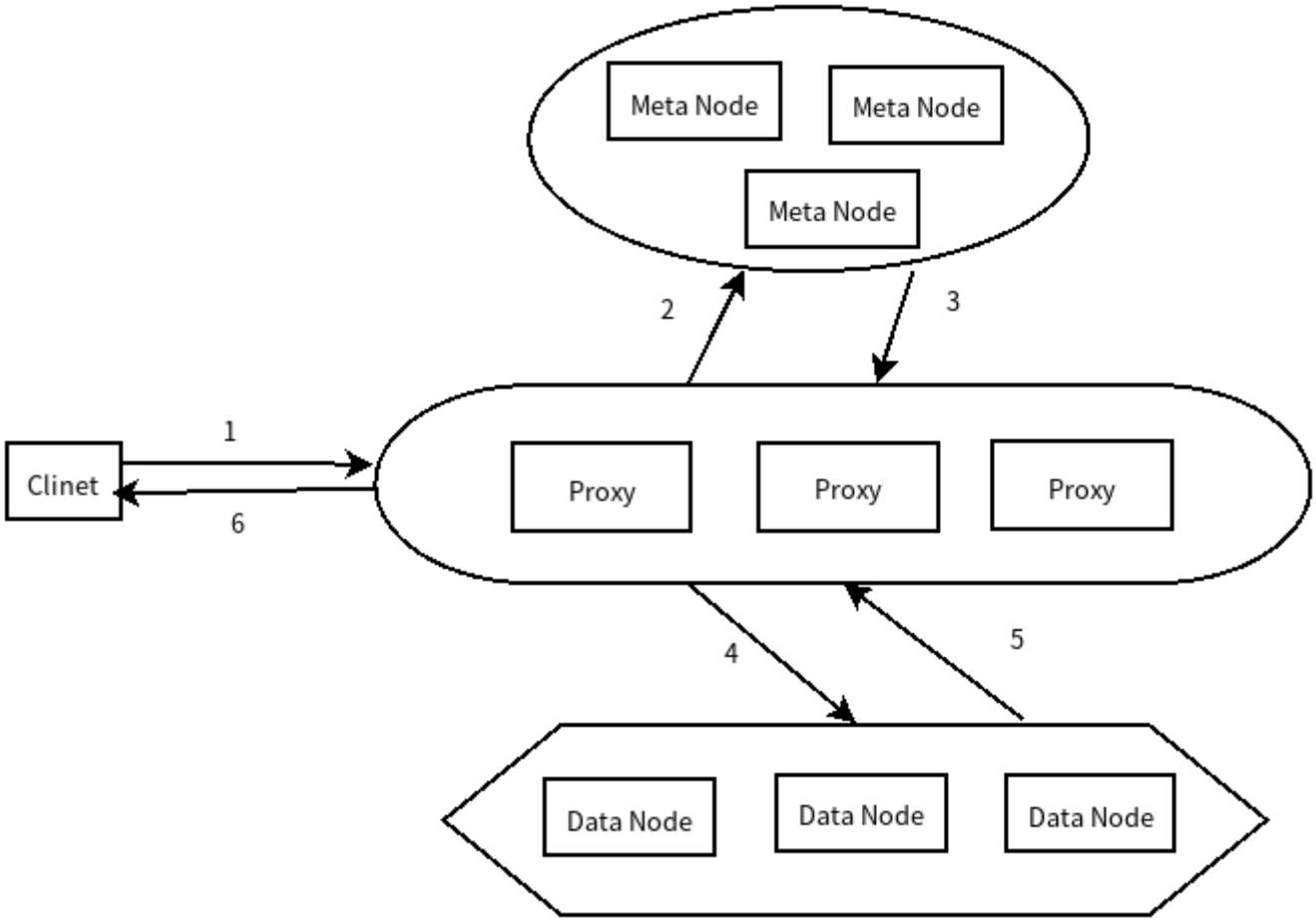
- 分布式时序数据库主要满足的功能点
- 方案简要架构
    - 架构图
    - 集群组件说明
    - 数据分片规则
    - 数据写入流程简述
    - 数据查询流程简述
- 顶层任务拆解

分布式时序数据库主要满足的功能点

1. 节点水平扩展：以满足大数据量的持久存储和吞吐量;扩容无需或仅需少量人工干预;
2. 高可用性：支持多副本，避免单点故障;
3. 查询：分布式多机查询归并的实现和性能优化。

方案简要架构

架构图

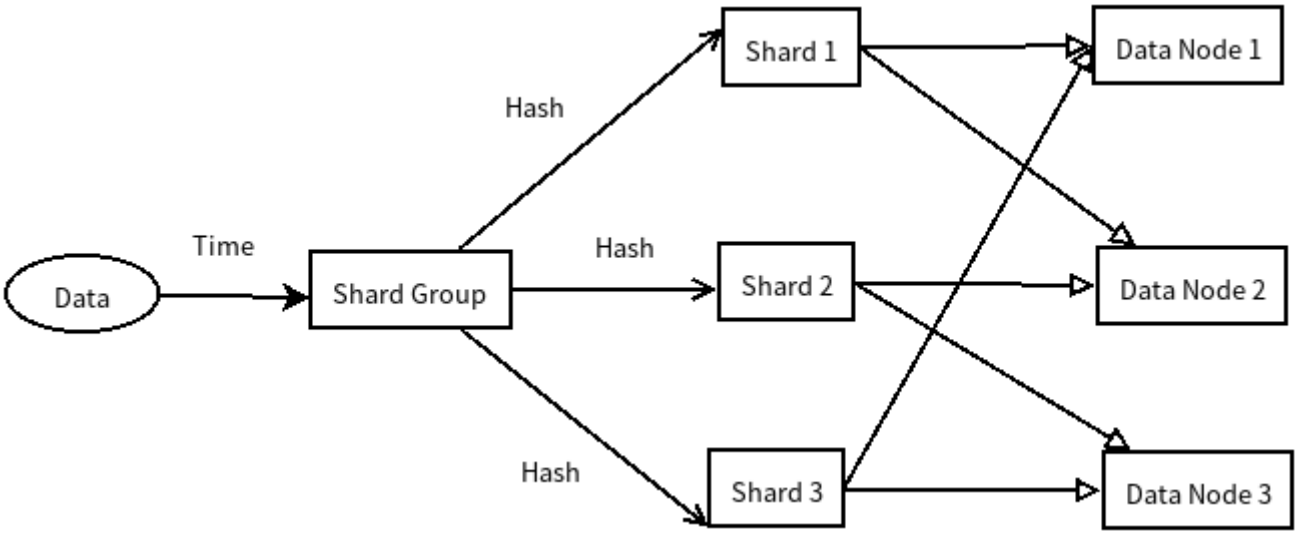


集群组件说明

1. Meta集群： 强一致存储(CP), 存储整个集群数据的元信息，可简单理解成写入数据和查询时的路由表;
2. Proxy: 接收客户端的所有请求, 并将响应返回给客户端，无状态服务，任意水平扩展;
3. Data集群： 存储数据，可理解成是一台台单机版InfluxDB;

数据分片规则

1. 数据以时间来切分不同的shard group来存储，比如时间间隔一小时，那么每1个小时就会产生一个新的shard group;
2. 每个shard group下分为若干个shard, 每个shard对应到不同的Data node机器上，如果有复本，则一个shard对应到多台 Data node上;
3. 写入请求中带有tag set（可简单理解为索引），将此tag set作 hash 然后根据当前shard group下shard个数散列到某一个shard上，写入对应的Data node;
4. 上述shar group的所有信息均作为元信息存储在Meta集群内;
5. 下面是2复本下数据分布情况：



数据写入流程简述

1. Client发送写入请求到任一台Proxy;
2. Proxy上如果没有缓存相应的Meata信息，则从Meta集群获取其Shard Group信息;
3. 根据请求中的tag set作 hash, 散列确定写入的shard;
4. 写数据写入shard对应的Data node;

数据查询流程简述

1. Client发送查询请求到任一台Proxy;
2. Proxy上如果没有缓存相应的Meata信息，则从Meta集群获取其Shard Group信息;
3. Proxy将查询请求转发给Shard Group下所有的Shard所在的Data node;
4. Proxy将3中所有子查询的返回结果作合并，返回给Client;

顶层任务拆解

1. 集群元数据的设计和存储, 要求强一致;
2. Proxy对元数据的管理;
3. Proxy与Data Node的通讯方式，交互协议设计;
4. 数据写入，包括复本间数据同步策略;
5. 数据查询，子查询结果的合并，包括较复杂的聚合操作;
6. Data node节点增加时，shard 规则自适应;
7. 其他在具体开发中遇到的问题…