

Bulk loading工作梳理

丁国浩

背景

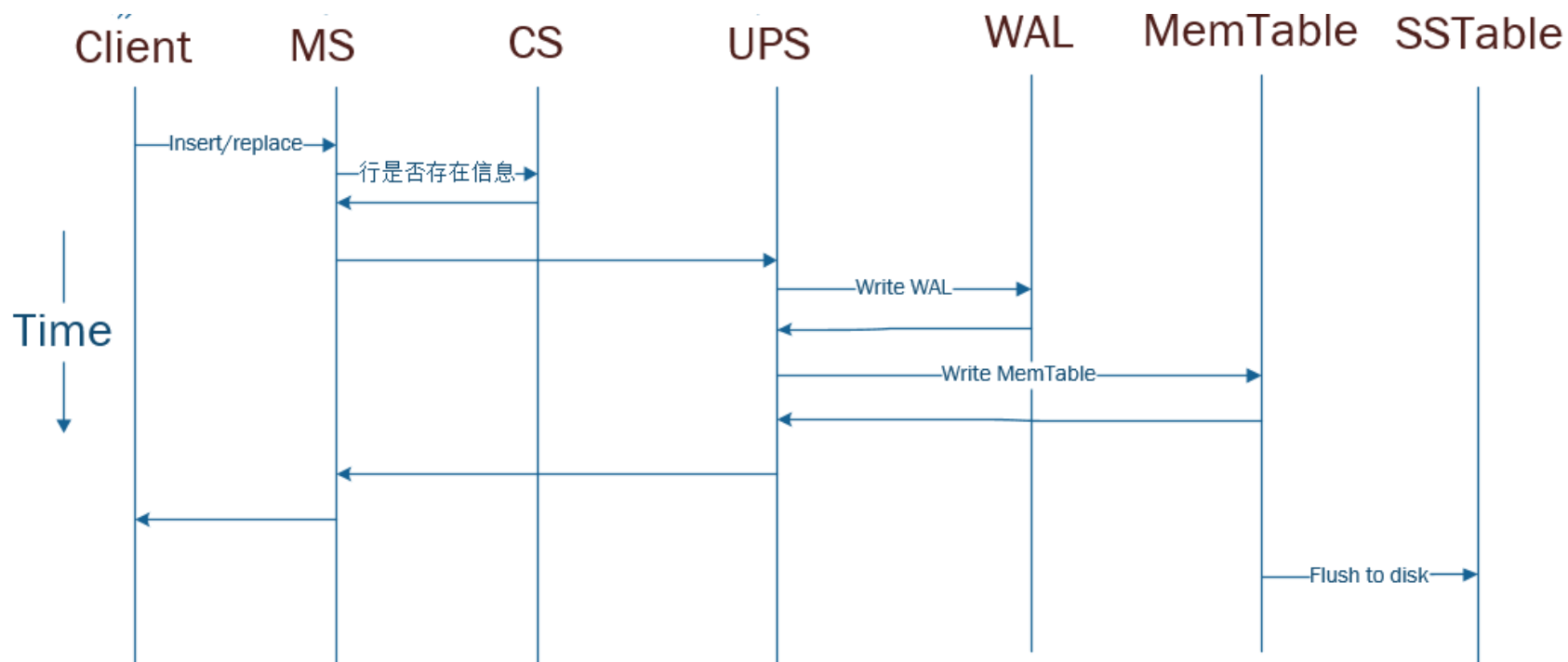
□ 应用场景

- ▣ 新业务上线，需要对空表导入大量初始化数据
- ▣ 系统需要每日导入一些增量更新数据

□ 常见加载方式

- ▣ SQL插入
- ▣ 直接更新内存表(**MemTable**)，即构造事务节点(**UPS**) 内部数据结构(构造**Mutator**)
- ▣ 构造存储节点 (**CS**) 上数据文件存储格式(旁路导入)

写入流程



SQL插入

□ 基本思想

- 将数据文件中的记录构造成**insert/replace**语句插入到数据库中
- 多条记录构造成一条**SQL**语句
 - 如: **Insert into test values(1,1,1),(2,2,2),(3,3,3)**
 - 提高网络传输效率和减少**UPS**事物处理量
- 多线程并发插入
- 分配到多个**MergeServer**

SQL插入使用

□ 参数介绍

-h Rootserver的ip

-t Rootserver 的端口号

--dbname 库名

-t 表名

-f 加载数据文件

[-l 日志文件]

[--badfile 未加载成功的记录存放文件]

[--progress] 显示导入进度

[--rowbyrow] 每批导入数据中只有一条数据

[---concurrency] 导入线程并发数，默认等于MS数

□ 使用举例

```
ob_import_v2 -h 182.119.80.57 -p 2500 --dbname test -t t1 -l 1.log -  
-badfile 1.bad -f 1.del
```

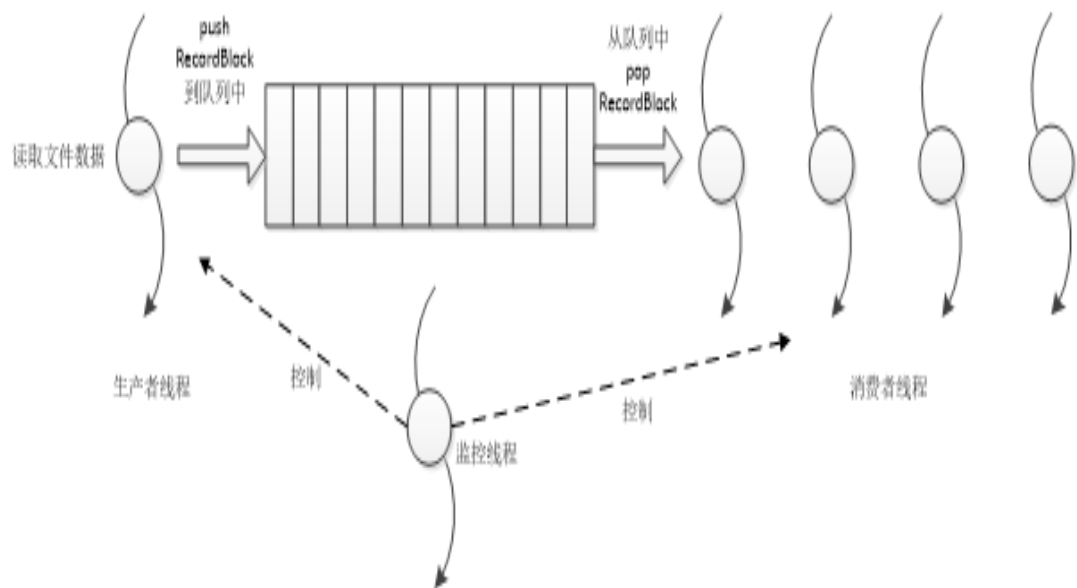
SQL插入执行流程

- 1)命令行参数解析
- 2)表结构组装
 - ▣ 获取表schema
- 3)文件数据读取
- 4)生产者-消费者多线程模型
- 5)数据格式化检查和格式转换
- 6)SQL拼接和执行
- 7)数据库状态监控
 - ▣ UPS切换
 - ▣ 是否发生合并

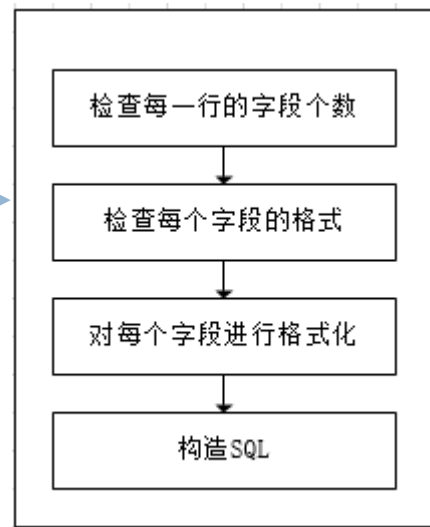
生产者-消费者多线程模型

生产者：读取文件数据

消费者：拼接sql写入数据库



导入数据



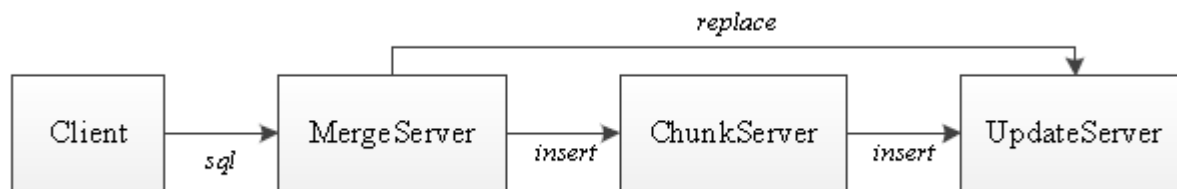
SQL插入优缺点

□ 优点

- ▣ 简单、易理解、易操作

□ 缺点

- ▣ 频繁网络传输限制数据加载的性能和效率
 - **MS:sql**解析、逻辑计划和物理计划生成等操作
 - **UPS:**执行读写事物、存储更新增量、记录日志等操作

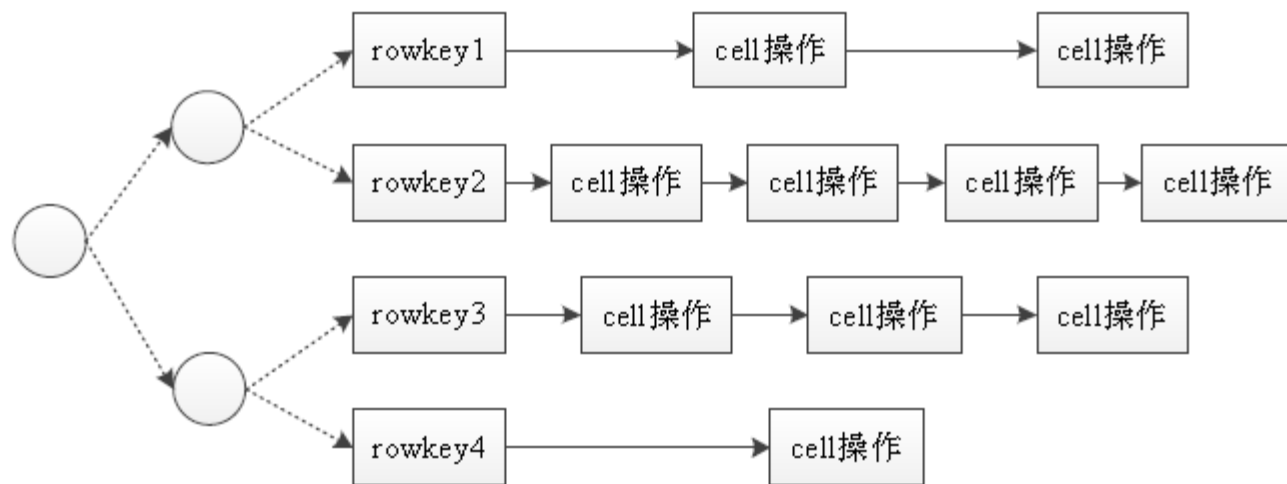


直接更新内存表

□ 基本思想

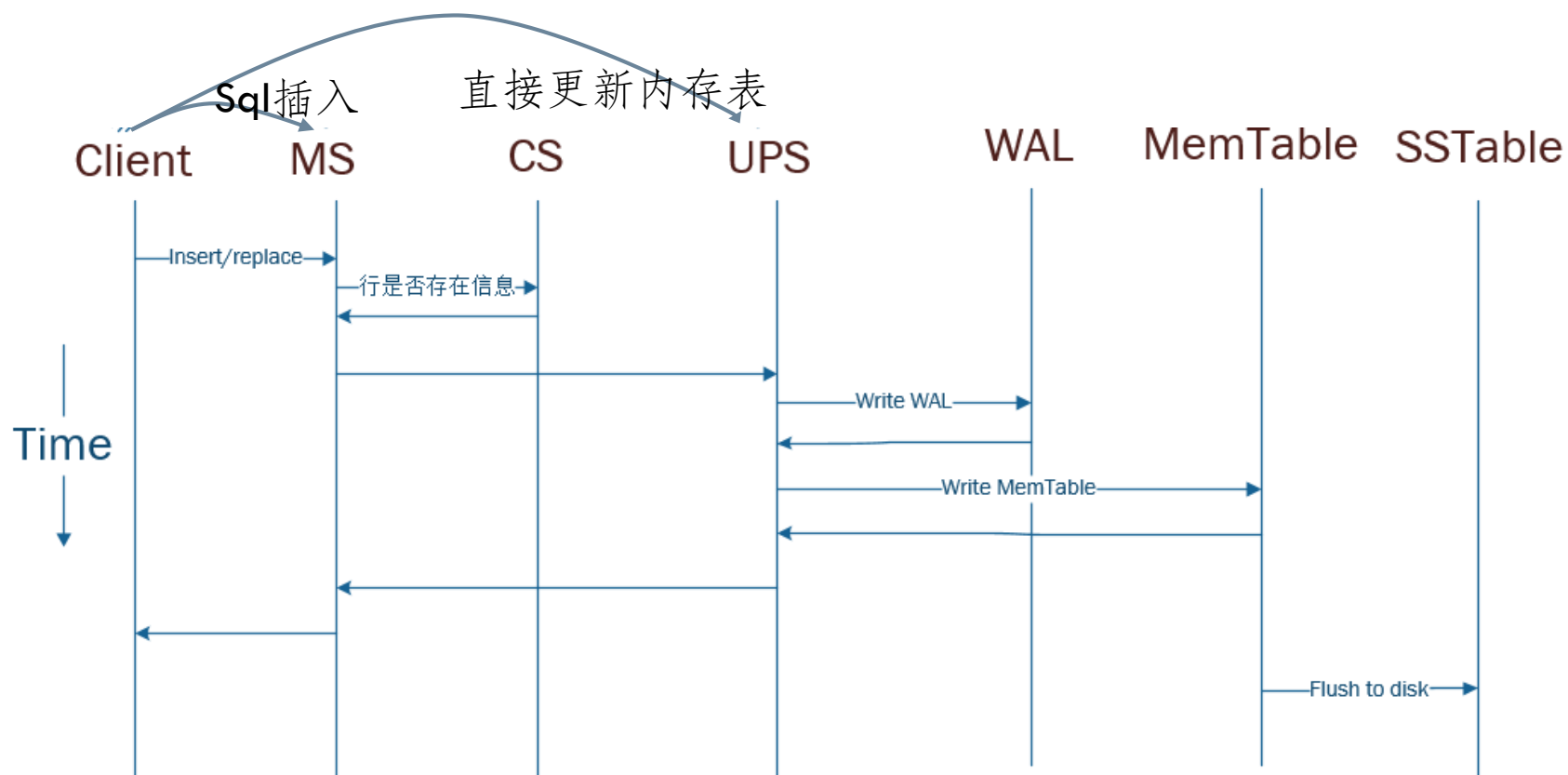
- ▣ 数据加载程序根据事物节点(**UPS**)的内存表数据节点的数据结构，将每条插入的记录直接构造一个数据节点，然后将该节点直接发到**UpdateServer**上更新内存表
- ▣ 多个加载任务并发执行，每个加载任务处理一批数据

内存表MemTable



B+树

直接更新内存表流程



直接更新内存表优缺点

□ 优点

▣ 速度快

- 减少网络传输
- 不经过sql语句词法语法解析、逻辑计划和物理计划生成等操作

□ 缺点

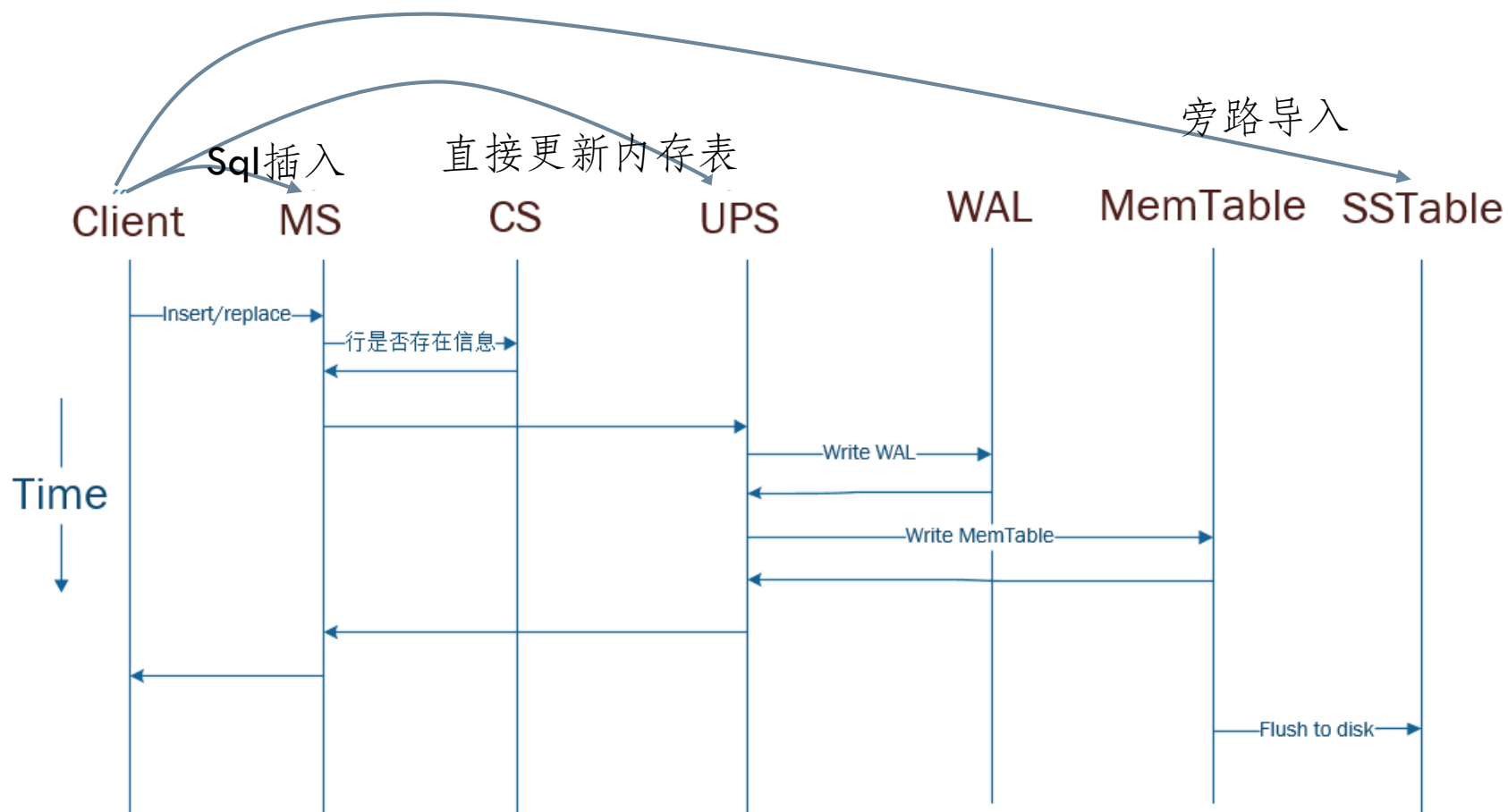
- ▣ 对事物节点（UPS）造成比较大负载

旁路导入

- 基本思想

- ▣ 将数据文件转换成存储节点上的物理存储格式
- ▣ 分别将生成的数据文件分发到不同存储节点上
(**ChunkServer**)

旁路导入

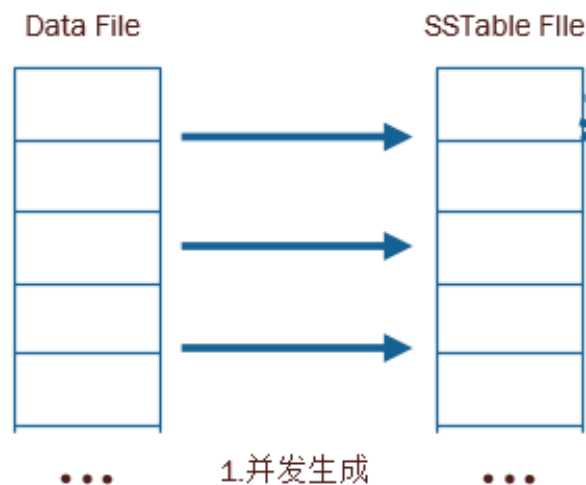


旁路导入方案设计

- 转换
 - ▣ 将数据文件转换为存储节点上数据存储格式
(先排序, 大多数情况数据文件已经排好序)
- 加载 (**Loading**)
 - ▣ 将转换后的文件加载到存储节点上
- 汇报
 - ▣ 存储节点向总控节点 (**RS**) 汇报加载的数据信息

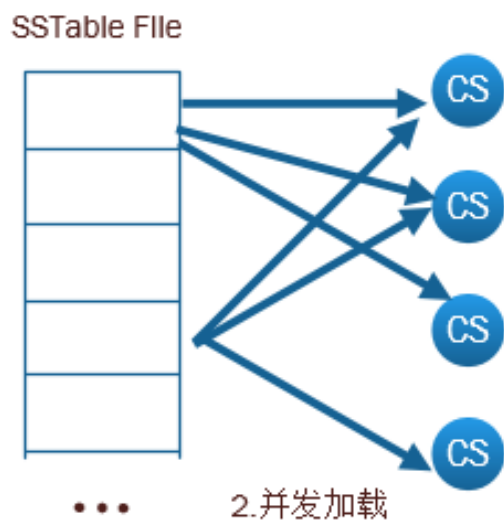
Step 1 转换

- 分块
 - ▣ 每个分块对应生成一个**SSTable**文件
 - ▣ 按照记录数（估算一条记录所占磁盘空间）
- 多线程并发处理
 - ▣ 块-->**SSTable File**



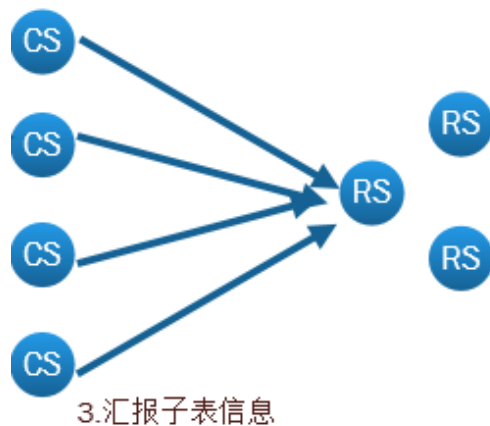
Step 2 loading

- 预分配
 - ▣ 计算每个CS上分配子表数 ($\text{总子表数} * 3 / \text{CS数}$)
- 多线程并发拷贝
- CS并行加载SSTable文件

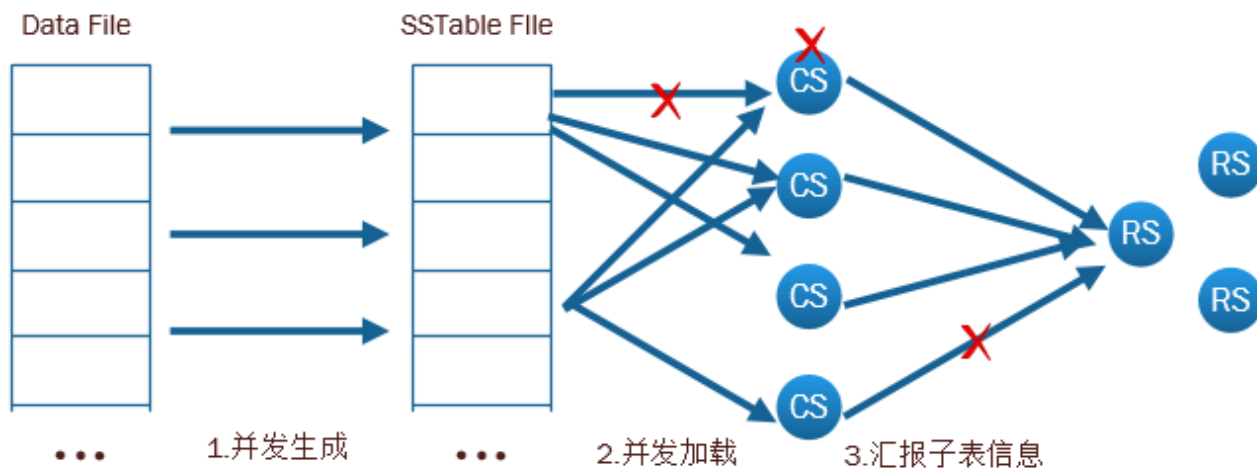


Step 3 汇报

- CS加载完本地SSTable文件后向RS汇报子表信息
- RS将汇报的子表信息更新到RootTable
 - ▣ RootTable是一张存储子表数据分布的有序表格



故障处理



故障处理

□ CS故障

- ▣ 重新选择另一台CS加载

□ 加载进程故障

- ▣ 日志：加载过程中记录成功加载的SSTable信息，失败的SSTable信息写入bad文件，可以重新加载

□ 汇报失败

- ▣ 网络故障：超时重试
- ▣ CS宕机：上线后重新汇报

旁路导入优缺点

□ 优点

- ▣ 速度快
- ▣ 消除对集群插入压力

□ 缺点

- ▣ 插入操作不产生任何日志信息，系统出现故障，数据库无法通过回放日志来修复数据
- ▣ 不对数据进行主键检查，不能进行增量加载

□ 适用场景

- ▣ 建表初始化时全量数据的加载
- ▣ 每天的全量加载

Thank you!