TokuDB 引擎实践分享

襄洛
阿里云资深开发工程师

目录

目录

TokuDB 简介

- Tokutek 公司2007 年研发
- Tokutek 公司 2013 年开源
- 教授 + 学生


Michael

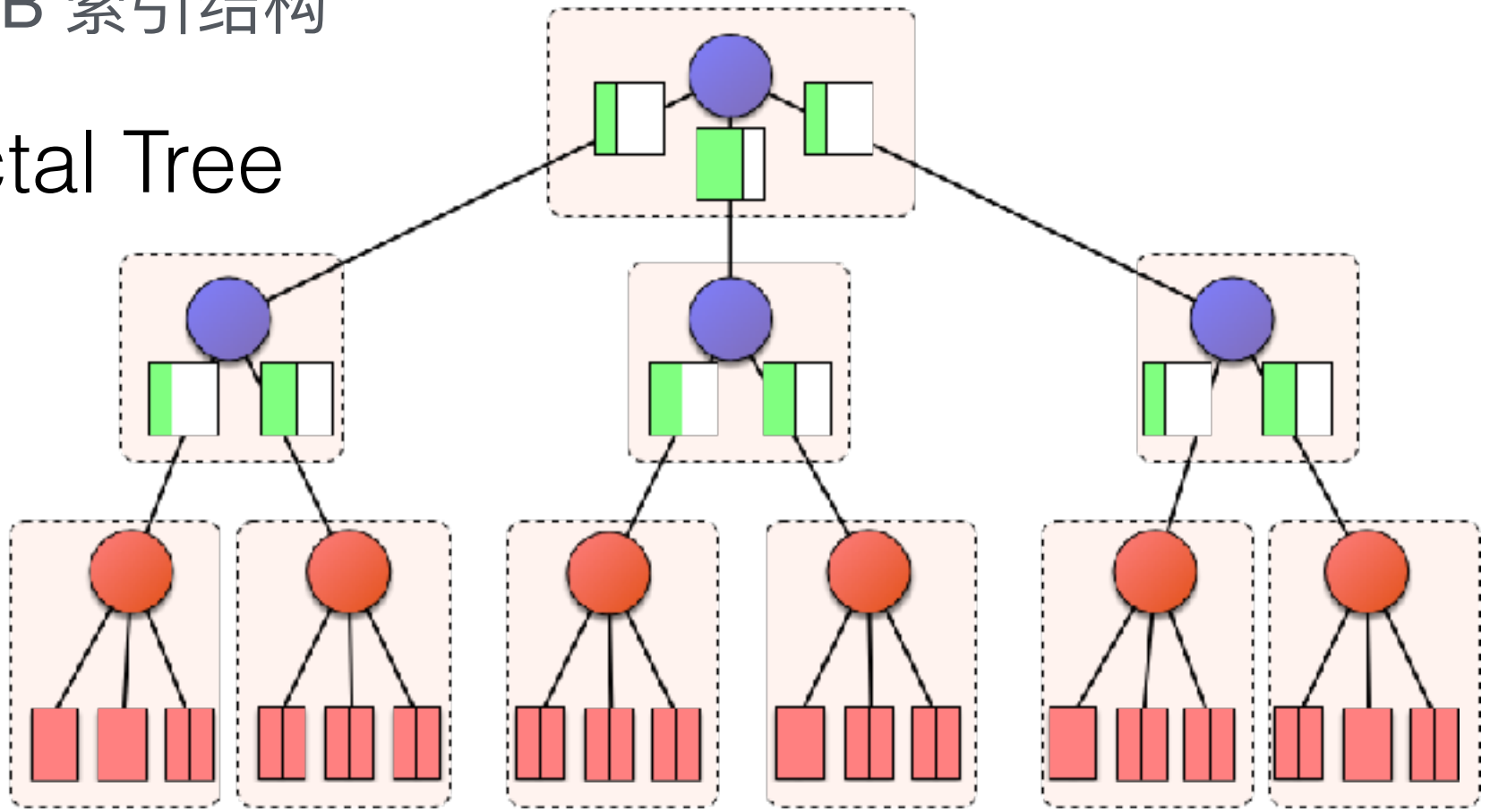

Martin


Bradley

- Percona 2015 年收购 Tokutek

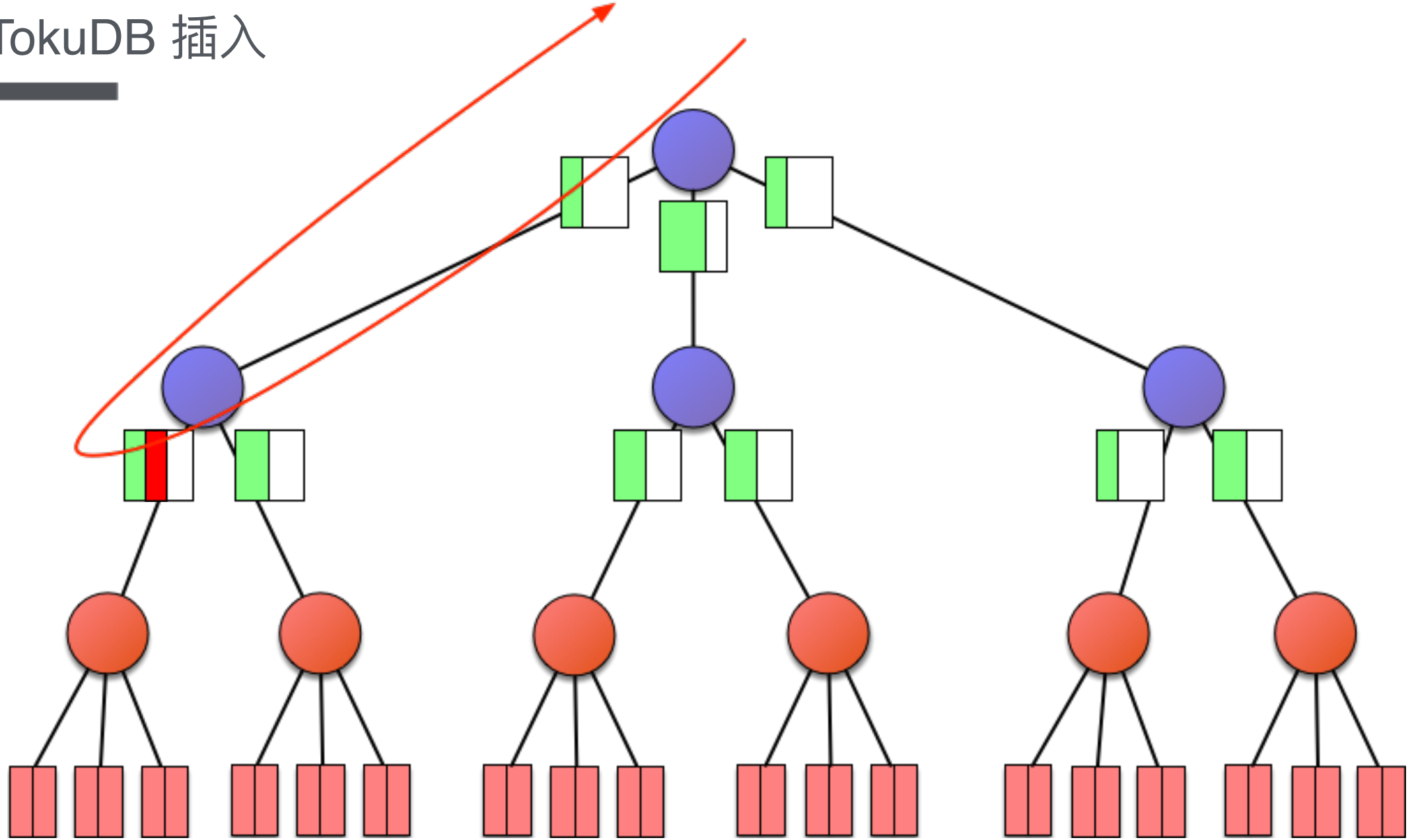
TokuDB 索引结构

Fractal Tree

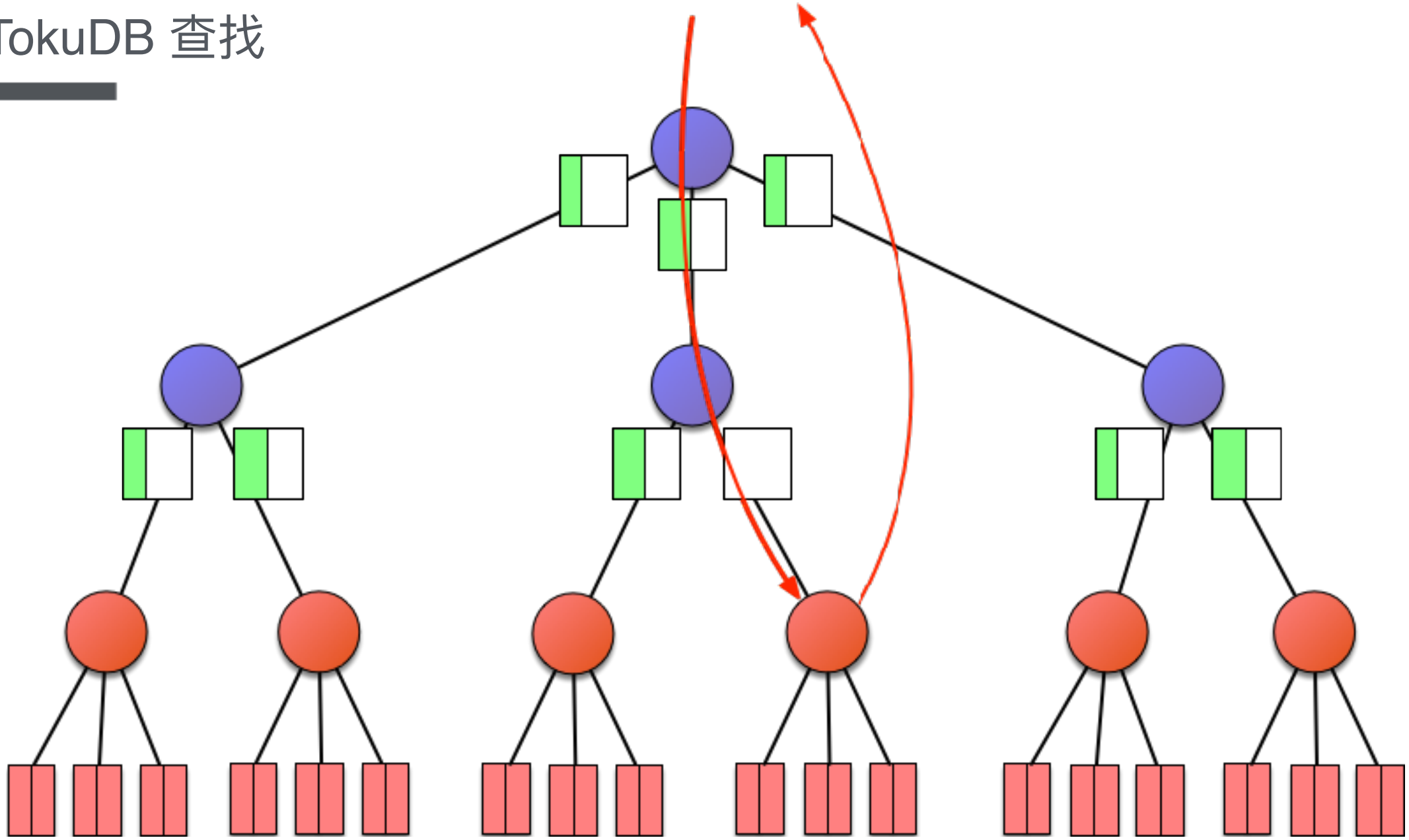


- 每个 node 有 [4-16] 个子节点 (fanout)，node 长度不固定，默认最大4M，落盘压缩
- node 上对应每个子节点有一个 partition
- 非叶节点 partition 是 message buffer，FIFO 队列，加 OMT（key, msn）排序
- 叶子节点 partition 是 basementnode，OMT 有序结构

TokuDB 插入



TokuDB 查找



TokuDB 特点

- 更新 (I/U/D) 都是 message，平摊思想(amortized)，异步批量下刷
- ACID 事务支持，MVCC 数据本身多版本(leaf)
- Redo 日志保护，log manager 管理，文件编号递增
- 定期 checkpoint (sharp)
- node 变长，Copy On Write，BTT 维护块的物理位置
- node 大，压缩效率高

- hot schema change，广播消息
- secondary clustering index
- fast insert/update
- read free replication

The Right Read Optimization is Write Optimization!

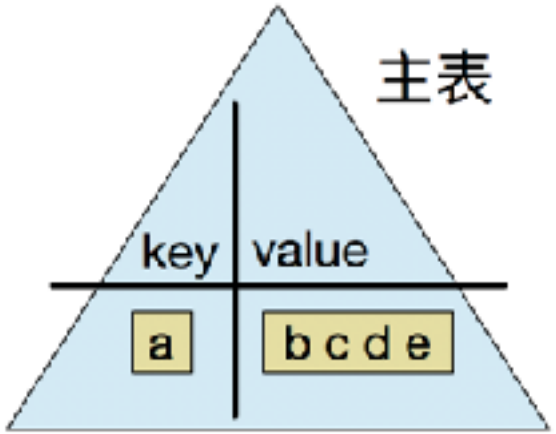
目录

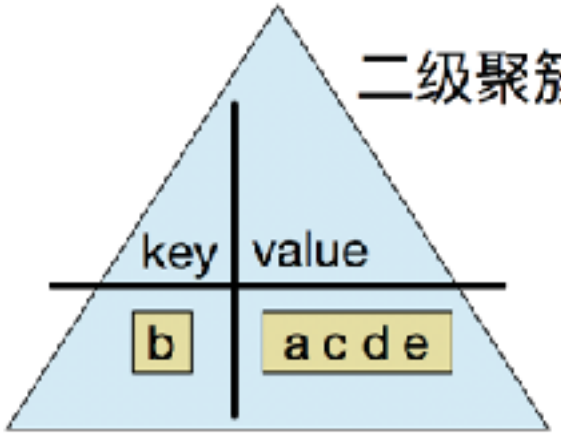
二. 阿里云使用和改进

Secondary Clustering Index

- server 层优化器改进 MRR
- 二级索引回表(hidden join)
- covering index
- secondary clustering index

select c from t where b > 10 and b < 100





- 索引代价 = 维护+存储

Fast Insert

- replace into
  新数据取代老数据
  search + del + insert
  FT_INSERT
- insert ignore
  有冲突保留老数据
  search + insert
  FT_INSERT_NO_OVERWRITE

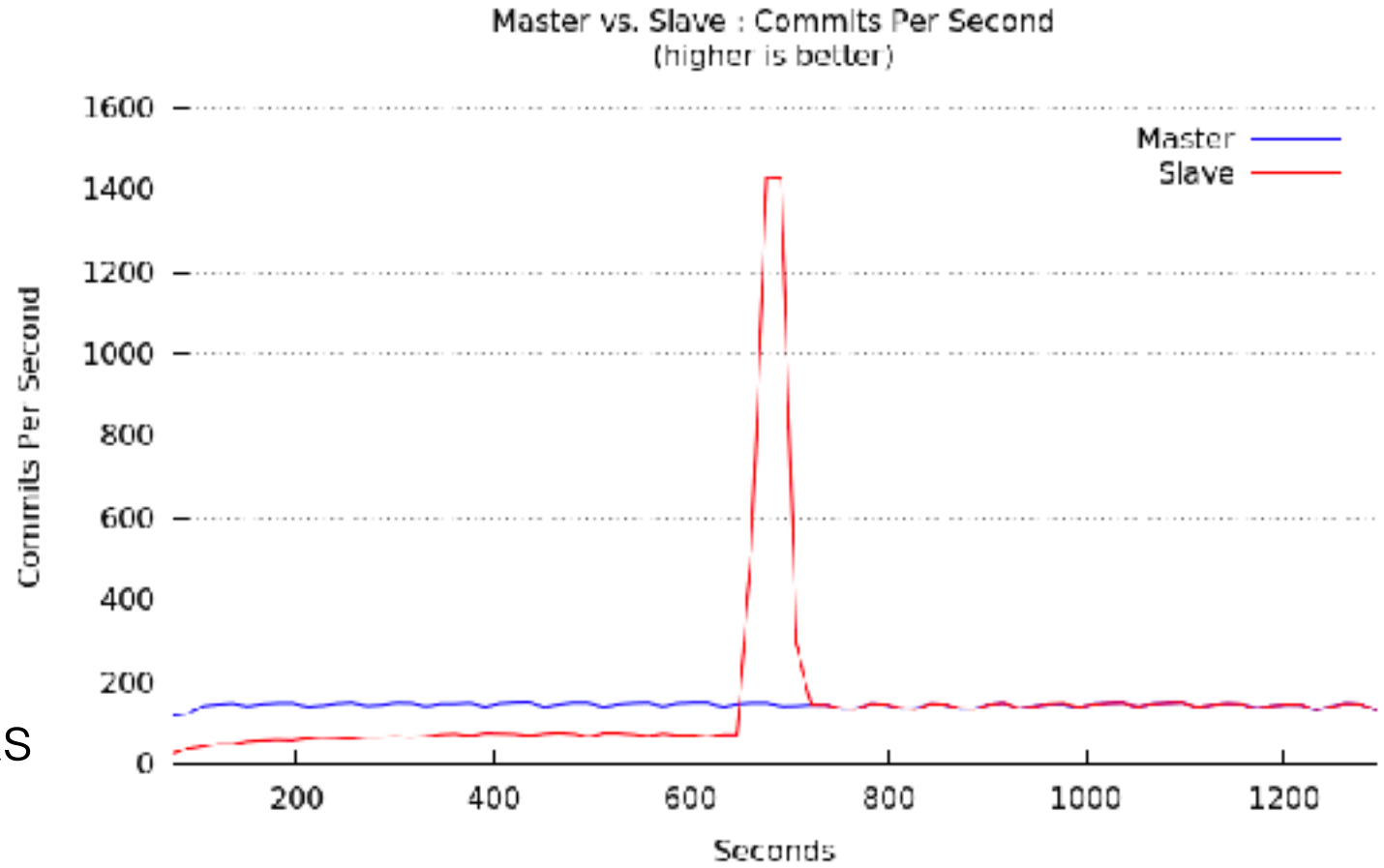
优化开启/关闭对比

	TPS	CPU%
REPLACE INTO OFF	3438.21	900
REPLACE INTO ON	6590.31	240
INSERT IGNORE OFF	6165.36	1000
INSERT IGNORE ON	6702.45	240

- 二级索引字段必须和 PK 一致
- binlog 用 statement 格式，或者关闭 binlog
- 表上不能有 trigger
- tokudb_pk_insert_mode=[0|1|2]

Read Free Replication

- 消除备库的读操作，加快 apply 速度
- row 格式
- 备库只读

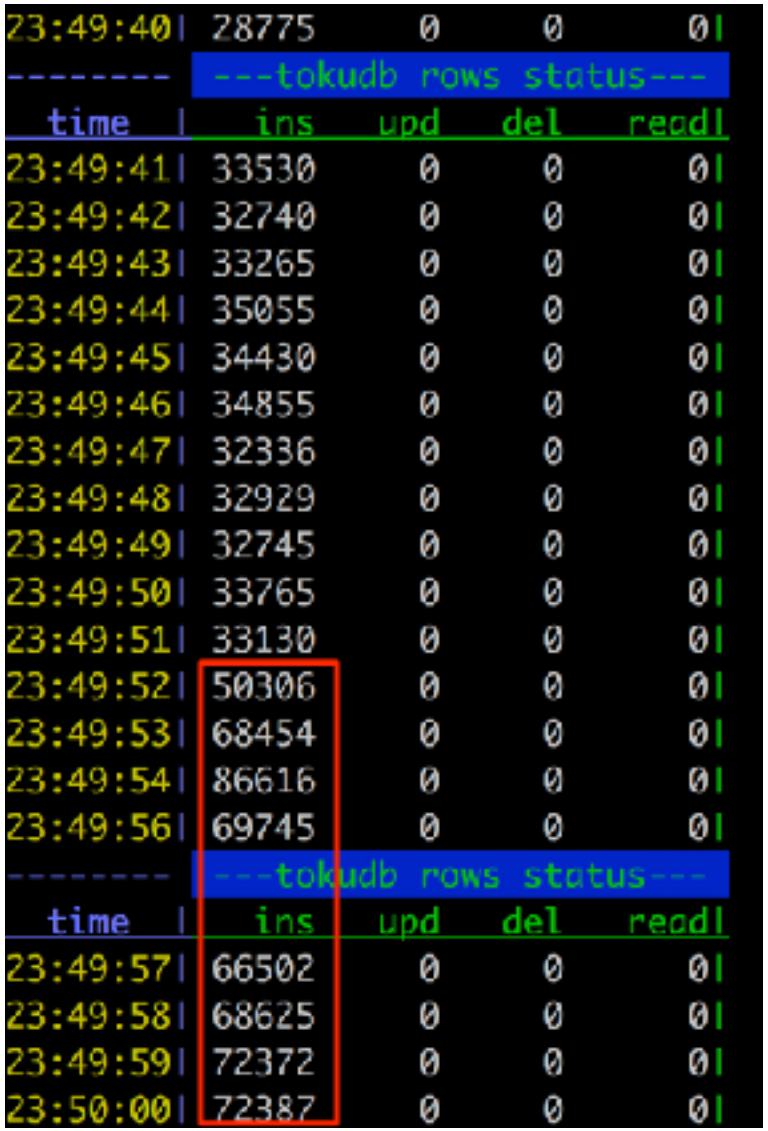


- tokudb_rpl_lookup_rows
- tokudb_rpl_unique_checks

- 特性需要 server 层支持，分区表不支持，已经 fix 提交官方

简单事务并行复制

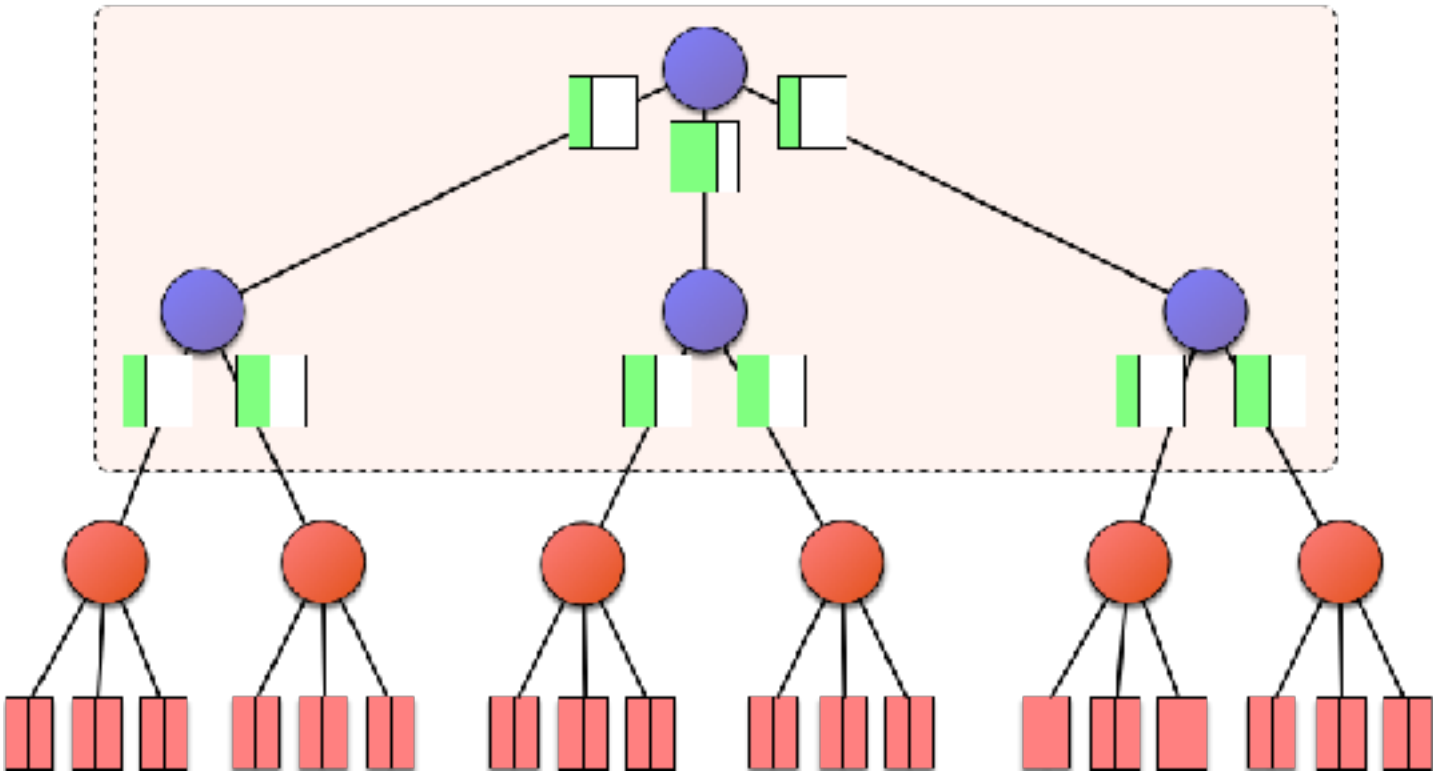
- 官方并行复制库级别
- AliSQL 支持表级别
  slave_pr_mode=table
- 事务级别（CPU+Mem）
  slave_pr_mode=TRX
- 简单事务级别（场景依赖）
  slave_pr_mode=STRX



Cachetable 优化

- put msg 有下推操作，最多2层
- evictor 线程做 partial evict
- 可能遇到 cache miss

- 将下推路径上的节点尽可能 hold 在内存



Transportable Tablespace

导出步骤

1.flush table t for export;
2.copy tokudb#*.cfg 以及里面的文件到固定目录
3.unlock tables;

导入步骤

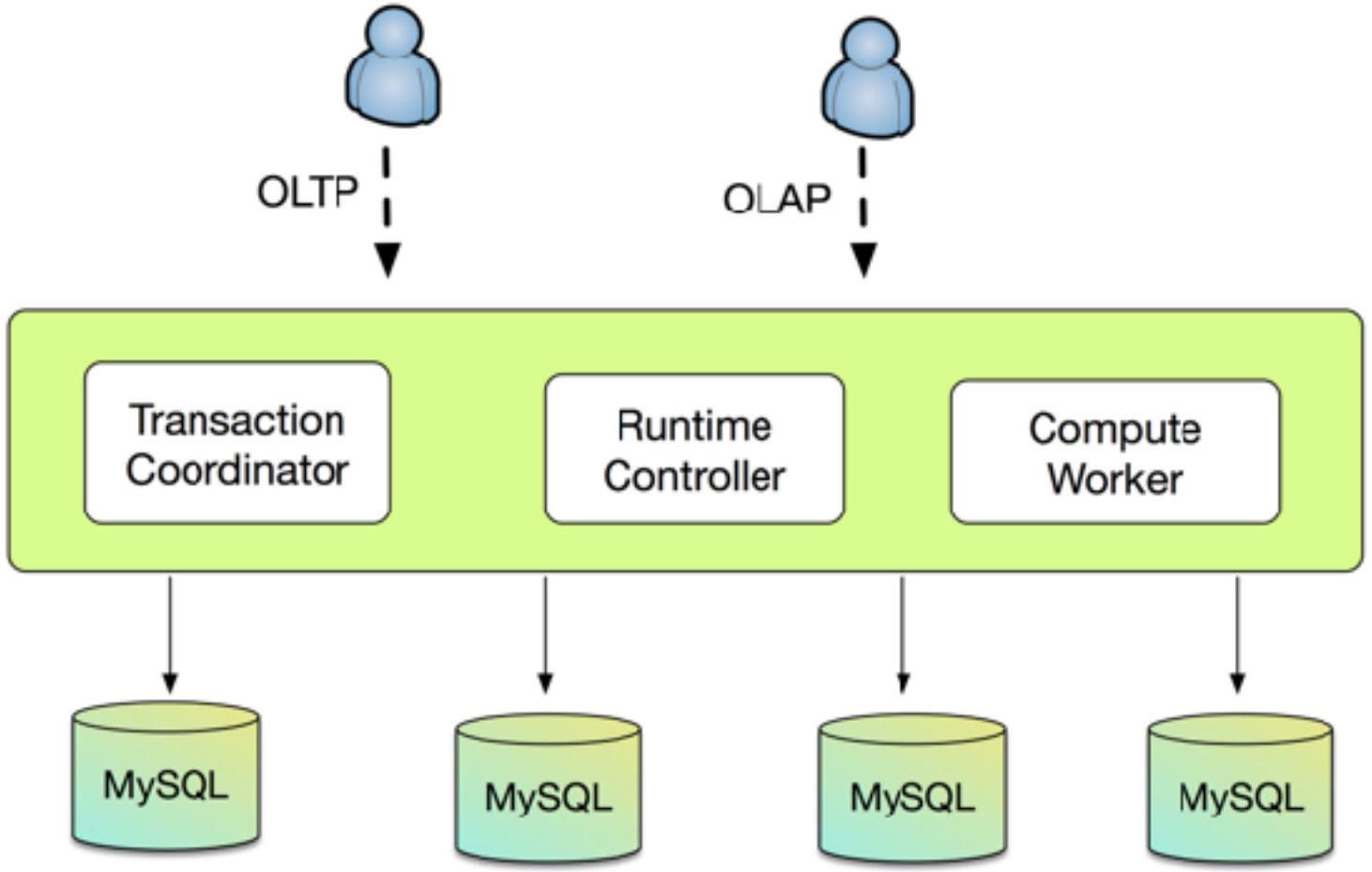
1.create table 与 t 完全一样;
2.copy 目录所有文件到数据目录
3.alter table t import tablespace;
4.flush table t;

使用建议

- 减少唯一索引使用，TokuDB 性能杀手
- hot schema change
- 慎用 optimize
- 适当调整 tokudb_fs_reserve_percent
- 注意文件句柄使用
- 用批量插入
- set optimizer_switch = ‘mrr=off';
- set optimizer_switch = 'index_condition_pushdown=off';

tokudb_cache_size (bytes)
tokudb_commit_sync (ON/OFF)
tokudb_fsync_log_period (ms)

PetaData 分布式数据库



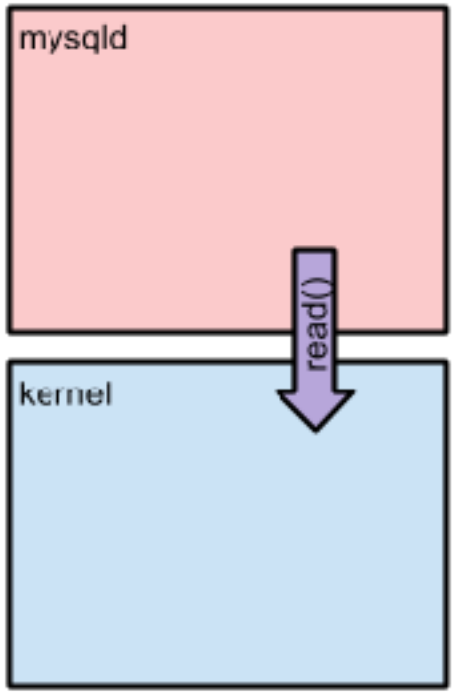
https://www.aliyun.com/product/petadata

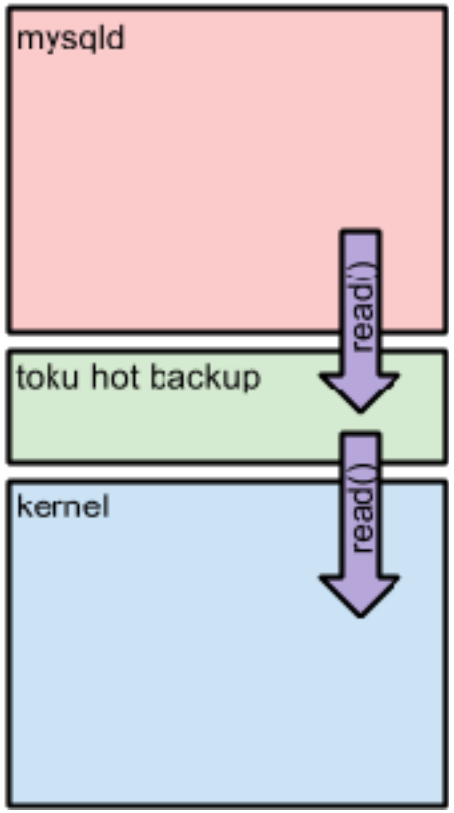
目录

三. 备份功能支持

TokuDB 官方备份方案

- 物理热备
- 插件方式集成在 mysqld (shim)
- 对引擎完全透明的
- 需要写本地文件系统



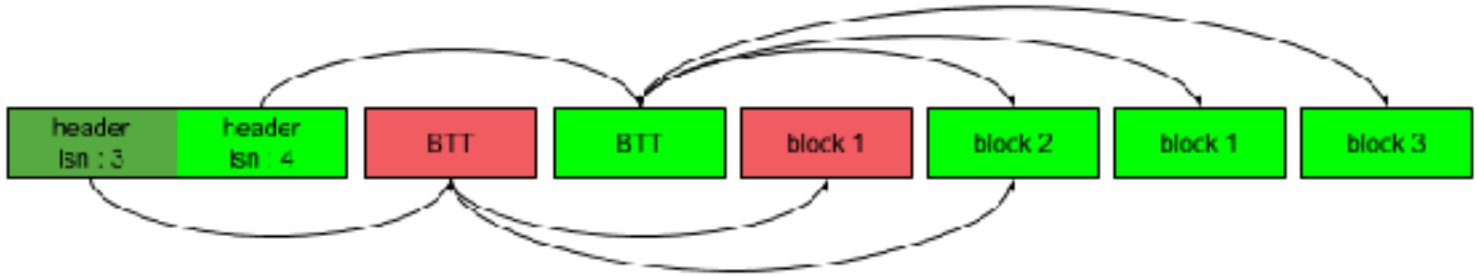


tokudb_backup_allowed_prefix
tokudb_backup_dir
tokudb_backup_exclude
tokudb_backup_last_error
tokudb_backup_last_error_string
tokudb_backup_throttle

```
set tokudb_backup_dir='/path/to/backupdir';
```

阿里云备份方案

- AliSQLBackup(XtraBackup)
- 物理热备
- 利用 checkpoint 机制，持有 checkpoint锁
- 通过 mysqld 恢复应用 redo
- 工具链支持

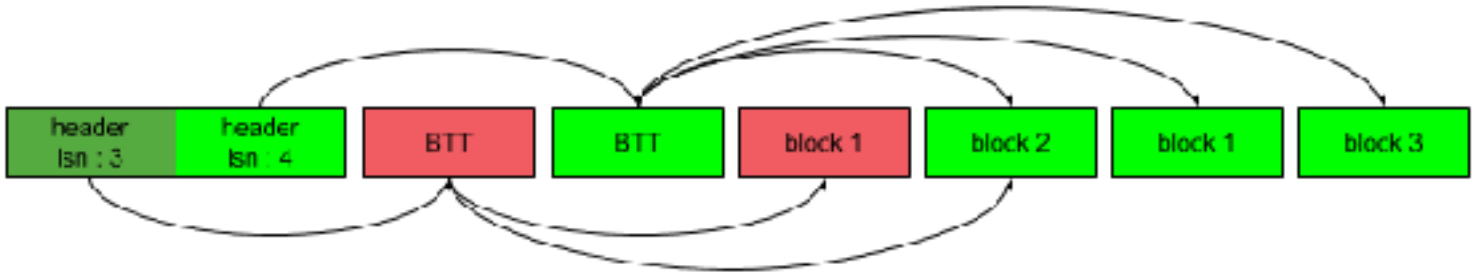


1. SET TOKUDB_CHECKPOINT_LOCK=ON;
2. FLUSH TABLES WITH READ LOCK;
3. 记录 binlog 位点，拷贝 TokuDB redo
4. UNLOCK TABLES;
5. 拷贝 TokuDB FT 数据文件
6. SET TOKUDB_CHECKPOINT_LOCK=OFF;

- Redo log 堆积（空间）
- Crash recover（可用性），加了并行 recover

阿里云备份方案 V2

- 类似 InnoDB 备份
- 工具内嵌 TokuDB
- 备份时持续拷贝 redo
- 恢复时应用 redo
- Checkpoint 加锁时间大大缩短
- 需要 mysqld 支持
- 并行备份



1. SET TOKUDB_CHECKPOINT_LOCK=ON;
2. 开启 TokuDB redo log 拷贝线程
3. 拷贝 FT header and BTT
4. SET TOKUDB_CHECKPOINT_LOCK=OFF;
5. 拷贝 TokuDB FT 数据文件
6. FLUSH TABLES WITH READ LOCK;
7. 停止 redo log 拷贝线程, 记录binlog位点
8. UNLOCK TABLES

关注我们 ^_^

1. AliSQL 开源（Code）
   https://github.com/alibaba/AliSQL
   https://github.com/alibaba/AliSQLBackup

2. 内核月报（技术干货）
   http://mysql.taobao.org/monthly/

3. @阿里丁奇

Thanks

Q&A

为了无法计算的价值 | 阿里云