



QCon 全球软件开发大会
INTERNATIONAL SOFTWARE
DEVELOPMENT CONFERENCE

BEIJING 2018

RadonDB

新一代分布式关系型数据库

演讲者 / 张雁飞

SPEAKER

- ▶ TokuDB内核维护者、XeLabs核心成员
- ▶ 淘宝核心系统/阿里云数据库内核组/青云数据库团队
- ▶ 目前在青云从事新一代数据库产品设计与研发工作



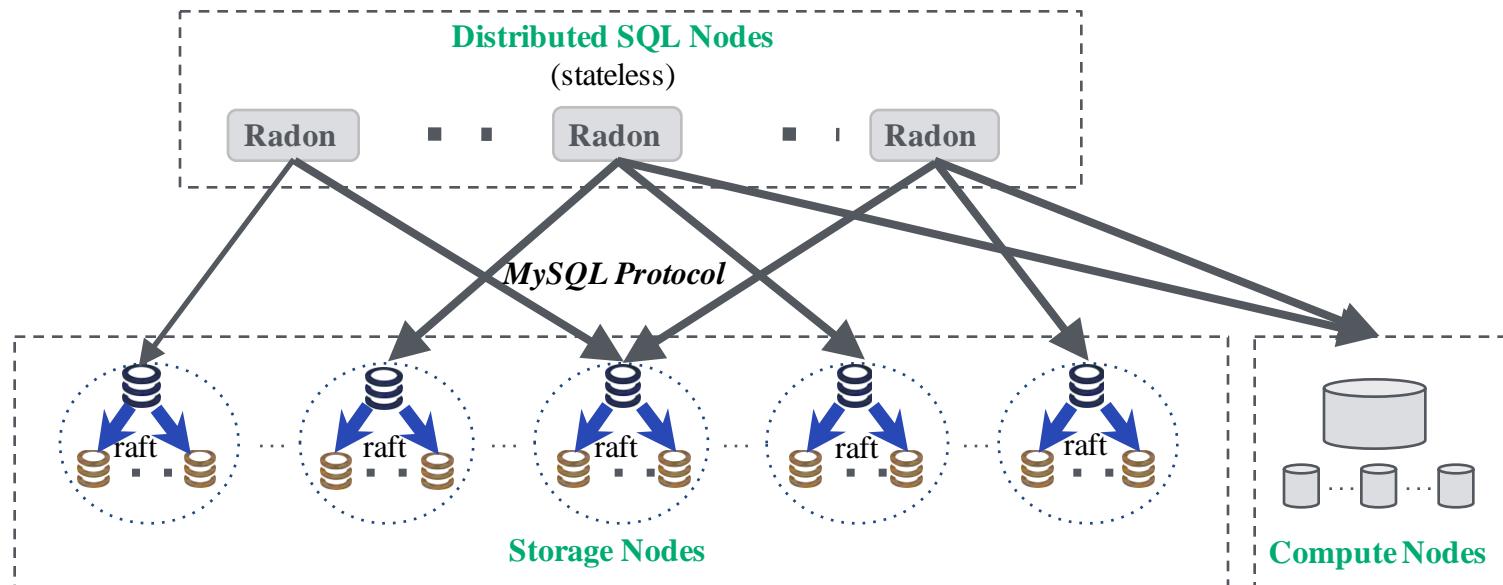
@BohuTANG

RadonDB

- ▶ 可扩展
- ▶ 高可用
- ▶ 强一致
- ▶ 易部署
- ▶ MyNewSQL



Architecture



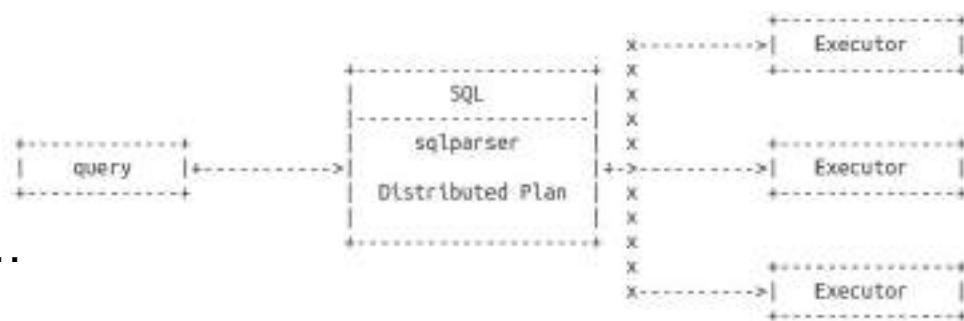
Distributed SQL

- ▶ 生成分布式执行计划

- ▶ 执行器并行执行

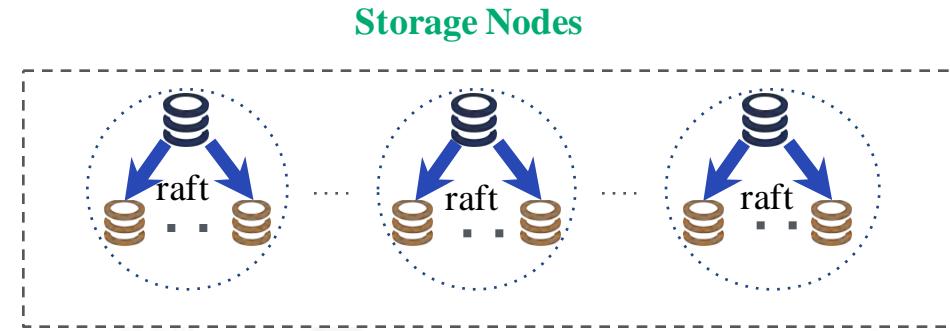
- ▶ orderby/limit/groupby/aggr/join ...

- ▶ 主从模式



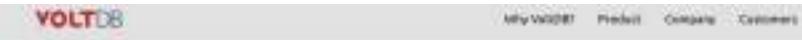
Storage Nodes

- ▶ 存储层由多个 node 组成
- ▶ 每个node 由多副本组成
- ▶ 每个副本为一个 MySQL
- ▶ 不仅存储还有计算能力



副本

- ▶ 为什么不是KV? MySQL!
- ▶ 稳定可靠、多索引写原子保证
- ▶ 计算下推，数据就近计算原则
- ▶ 不仅存储还有计算能力
- ▶ SQL 与 Storage 数据传输最小化
- ▶ MySQL 8.0更加强大...



140, 63 MB

FoundationDB's Lesson: A Fast Key-Value Store Is Not Enough

The late and subsequent closure of FoundationDB is sort of a grand experiment. FoundationDB, conceived as a key-value store, had decided to add flexibility in the form of programming and query-model "Layers" on top of its base structure. First up was SQL, followed by one or more layers on top of core FoundationDB and provided SQL, relations, indexes and queries, a graph interface and possibly other "Layers" would follow.

So how did the SQL system work out? [Running networked TDS-SQL was less than half as fast as MySQL on a single machine](#). This is by their own measurements. They do claim their system scales well as you add more machines, but crucially, they don't give actual numbers for the distributed SQL performance on spherical, just "worstized performance." I'd bet some money that absolute performance was not good. If the network overhead of the loopback interface was bottlenecking the single-machine test, then an actual network with actual latency wouldn't have helped.

<http://www.vldb中心.com/2012/04/01/the-implications-of-a-fast-key-value-store-on-sql/>

Spanner: Becoming a SQL System

David F. Bacon	Nathan Bates	Nico Bruno	Brian F. Cooper	Adam Dickinson
Andrew Fikes	Campbell Fraser	Andrey Gubarev	Milind Joshi	Eugene Kogan
Alexander Lloyd	Sergey Melnik	Rajash Rao	David Shue	Christopher Taylor

Google, Inc.

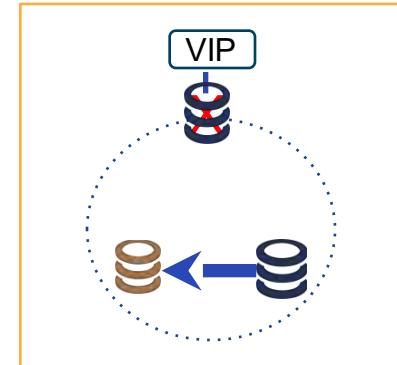
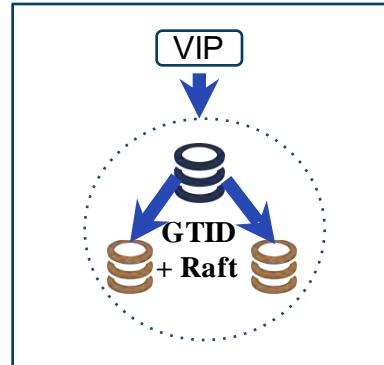
ABSTRACT

Spanner is a globally-distributed data management system that

like papers, we focus on the "database system" aspects of Spanner; in particular how query execution has evolved and forced the use of

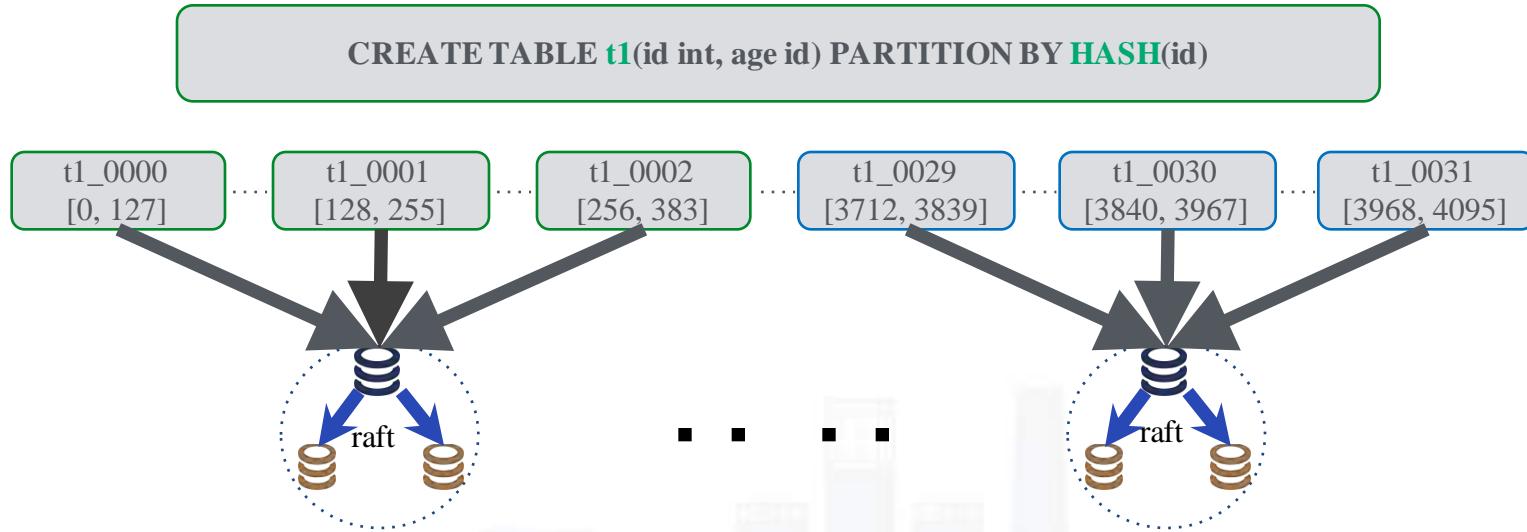
高可用

- GTID 作为 Raft Log Index
- Raft 协议选主、Log 并行复制
- 主副本故障秒级切换即可服务
- 强 Semi-Sync 确保事务不丢失
- 单副本故障可快速流式重建
- 无中心化，可跨机房部署



Raft+MySQL = Raft 选主+GTID 并行复制+强 Semi-Sync 数据强一致、切换零丢失

数据分布



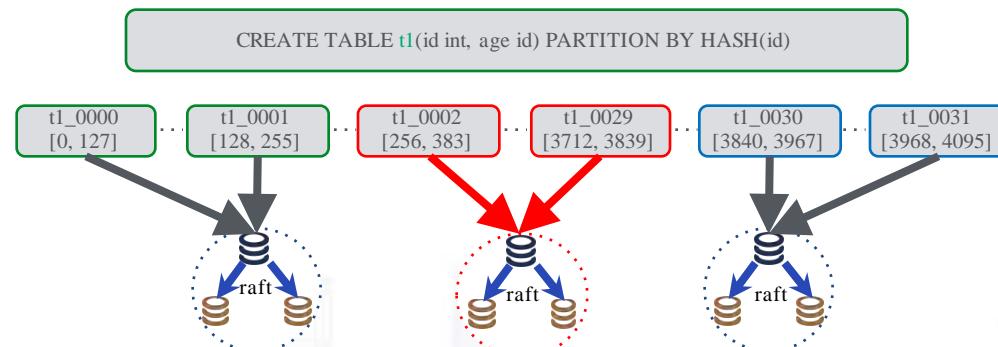
► 整张表共 4096 slots

► 每个小表 128 slots

► 小表均匀分散在 node 节点

扩容

- ▶ 小表可动态漂移
- ▶ 先全量后增量
- ▶ 较大/热度高者优先
- ▶ 资源分配最优化



分布式事务

- ▶ 事务管理
- ▶ 事务可靠性
- ▶ Snapshot Isolation 隔离级别



SI隔离级别

► 未提交不可见

► 部分提交不可见

```
client1> select * from t1 where id>0;  
client2> update t1 set a=1 where id>0;
```

```
client1 got 1:  
case1. time .....>  
    | -> client2-update | -> client1-select
```

```
client1 got 0:  
case1. time .....>  
    | -> client1-select | -> client2-update  
  
case2. time .....>  
    | -> client1-select  
        | -> client2-update  
  
case3. time .....>  
    | -> client1-select  
        | -> client2-update
```

SI检测

► xelabs/go-jepsen

► 1个更新线程，16个扫表线程

► 100多亿次操作和检测

► 随机 kill 存储节点主副本

```
Thread1:  
update jepsen_si set score=0;
```

```
ThreadN:  
select score from jepsen_si;  
for cur := row.next() {  
    if pre != cur {  
        errors++  
    }  
}
```

time	thds	w-ops	r-ops	error(s)	total-ops
[3599s]	[r:16,u:1]	88000	3310000	0	11799980000
[3600s]	[r:16,u:1]	78000	3130000	0	11803180000

Radon - Binlog

- ▶ Statement + GTID格式
- ▶ 可被订阅用于数据同步(计算节点)
- ▶ show binlog events [GTID] [limit]
- ▶ 实时流式获取

OLTP + OLAP

- ▶ 独立计算节点(Compute Node)
- ▶ 数据通过 Radon Binlog 进行快速同步
- ▶ SQL 层自动路由复杂查询到计算节点
- ▶ 优点: 高并发事务与复杂查询资源隔离
- ▶ 缺点: 存储 2 份, 目前使用压缩解决

审计日志

- ▶ 支持多种审计模式
- ▶ 可定位慢查询等

```
type event struct {
    ▶ Start      time.Time      "json:\"start\""           // Time the query was start.
    ▶ End        time.Time      "json:\"end\""            // Time the query was end.
    ▶ Cost       time.Duration "json:\"cost\""           // Cost.
    ▶ User        string        "json:\"user\""           // User.
    ▶ UserHost   string        "json:\"user_host\""       // User and host combination.
    ▶ ThreadID   uint32        "json:\"thread_id\""       // Thread id.
    ▶ CommandType string        "json:\"command_type\"" // Type of command.
    ▶ Argument   string        "json:\"argument\""        // Full query.
    ▶ QueryRows  uint64        "json:\"query_rows\""      // Query rows.
}
```

Backup & restore

- ▶ xelabs/go-mydumper
- ▶ 批量并行流式导出
- ▶ 批量并行导入

性能

sysbench: 16表, 512线程, 随机写, 5000万条数据

	Transaction Per Second(TPS)	Response Time(avg)	规格
RadonDB (1SQL节点, 4存储节点)	26,589	20ms	4 存储节点(16C64G超高性能主机) sync_binlog=1 innodb_flush_log_at_trx_commit=1
单机 MySQL (QingCloud RDB)	9,346	73ms	RDB(16C64G超高性能主机) sync_binlog=1 innodb_flush_log_at_trx_commit=1

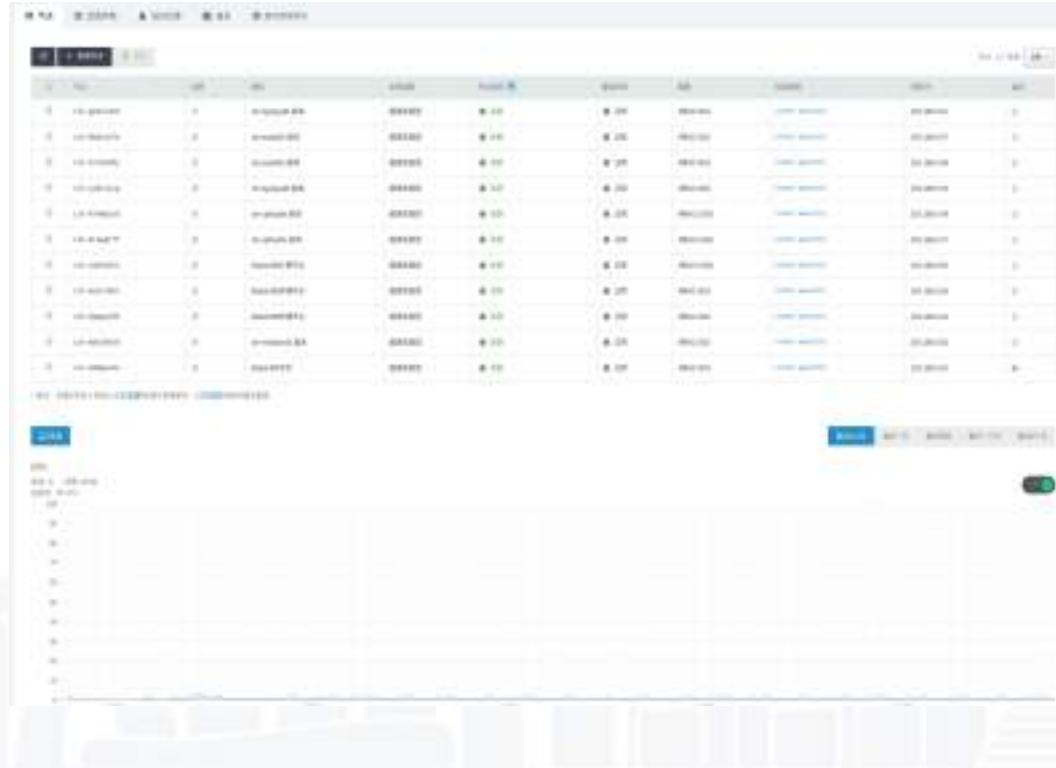
部署

- ▶ 云化部署
- ▶ 简单快捷



资源监控

- ▶ CPU
- ▶ 内存
- ▶ 硬盘IOPS/使用率...



监控

- ▶ 全链路监控
 - ▶ mysql> show processlist;
 - ▶ mysql> show txnz;
 - ▶ mysql> show queryz;

卷之三

40 (1999) 1009-1010

展望

- MyNewSQL 刚刚开始
- Hybrid Row/Column Data Storage
- RadonDB 即将开源



关注QCon微信公众号，
获得更多干货！

Thanks!



INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

极客邦
Geekbang > InfoQ