ArchSummit全球架构师峰会北京站2015

《深度解析云数据库TiDB》

liuqi@pingcap.com
微博: @goroutine

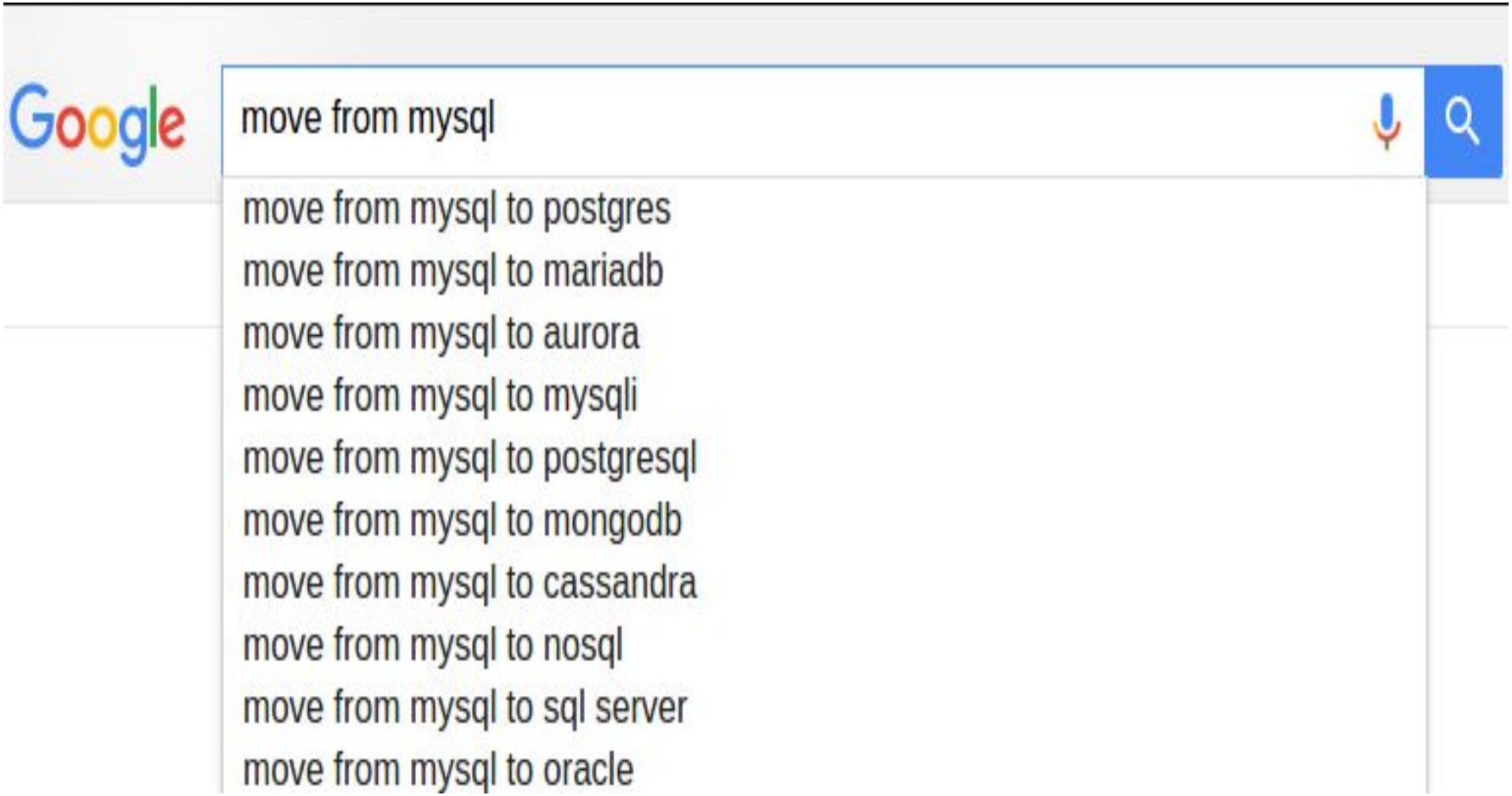
议程

- 关于 MySQL 的一些技术痛点
- 我们需要一个怎样的数据库?
- 技术实现
- 关于多租户的一些讨论
- 一些测试方法

话题从哪里开始 ？

Google 能告诉我们什么

MySQL 君怎么了？


Google
move from mysql
move from mysql to postgres
move from mysql to mariadb
move from mysql to aurora
move from mysql to mysqli
move from mysql to postgresql
move from mysql to mongodb
move from mysql to cassandra
move from mysql to nosql
move from mysql to sql server
move from mysql to oracle


PostgreSQL


ORACLE


MariaDB


APACHE
HBASE

逃离 MySQL ？


mongoDB


amazon
web services™
Announcing
Aurora


Microsoft®
SQL Server®


Cassandra

MySQL 君怎么了？

- 争论从未停止
- 选你所爱

MySQL 君怎么了？

- 不被巨头控制，垄断
- 不被绑架
- 自由

MySQL 君怎么了？

- scale
- schemaless

MySQL 君怎么了？

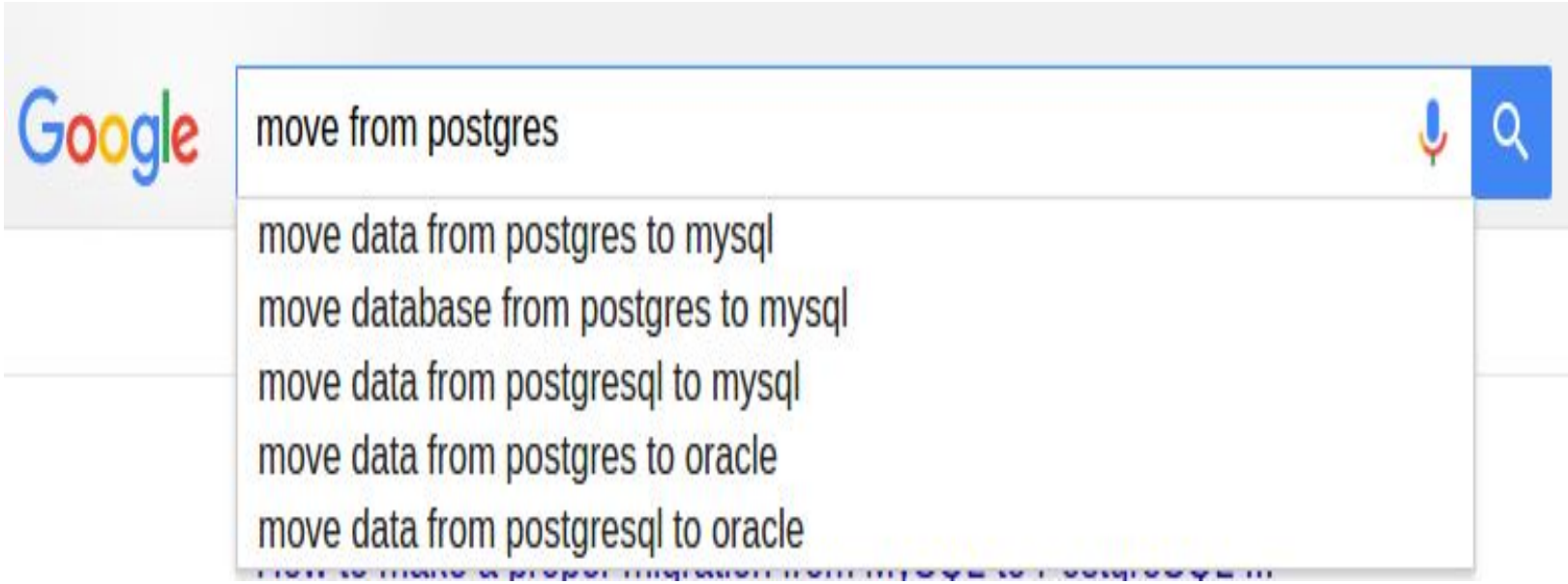
- scale
- fast

MySQL 君怎么了？

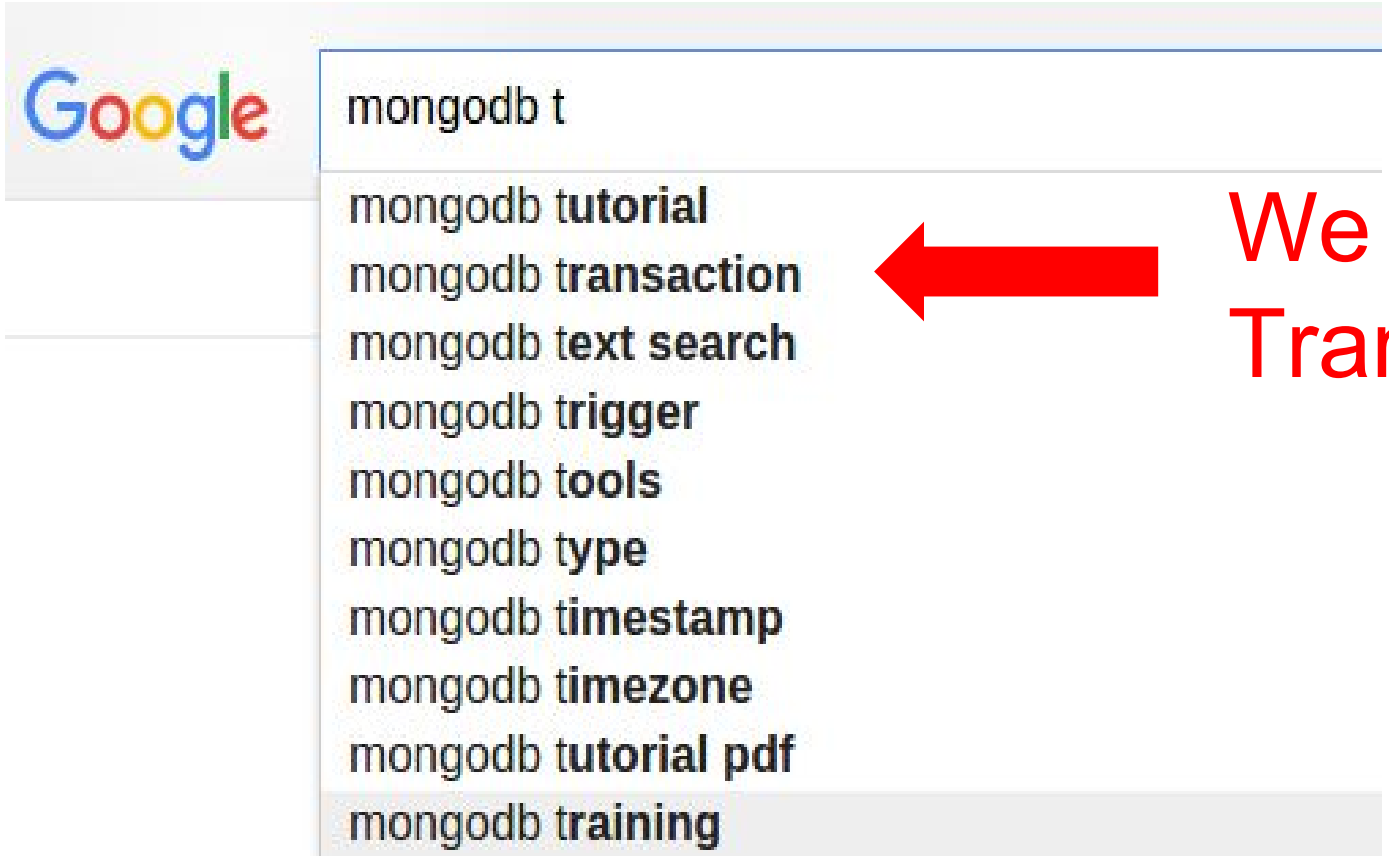
- 数据库前面通常有缓存
- Redis越来越流行
- 其它各种补充
  - HBase
  - Couchbase
  - .....

别人家的够好吗？

PG 君怎么了？


Google
move from postgres
move data from postgres to mysql
move database from postgres to mysql
move data from postgresql to mysql
move data from postgres to oracle
move data from postgresql to oracle

MongoDB 君怎么了？



We Need Transaction!

先小结下：

离开 MySQL 的原因:

- 水平扩容/缩容
- 分布式事务
- 容错
  - 半同步转异步，数据一致性问题

大家是怎么解决这些问题的？

水平伸缩

- NoSQL 在这方面有很好的积累
  - HBase
  - Cassandra
  - MongoDB

分布式事务

- 典型做法
- 两阶段提交（2PC）
  - Google spanner (2PL + 2PC)

隔离级别

- SI
  - 可重复读
  - Write skew
    - update set a = a + 1
- SSI
- External consistency

容错

- 多副本(raft 协议复制，N / 2 + 1)
- 去中心化的事务冲突检测

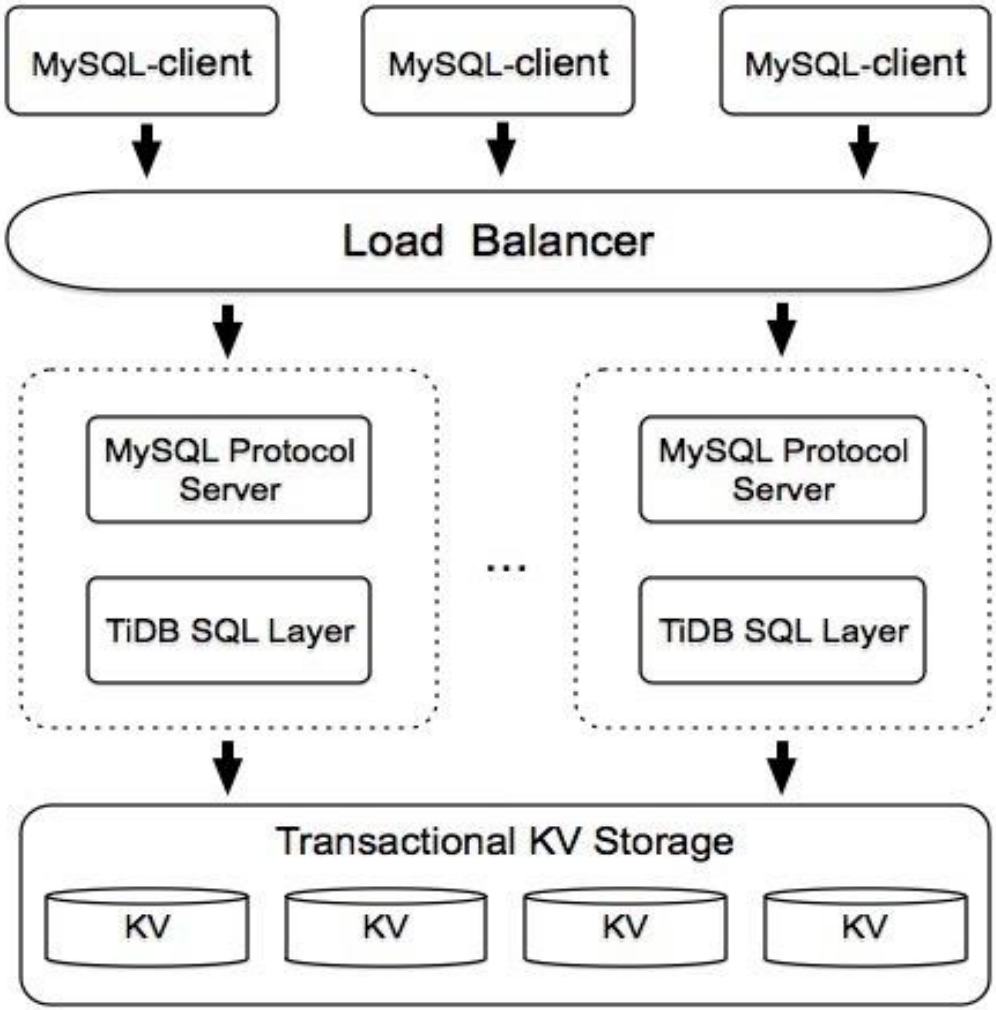
TiDB 如何解决这些问题？

受 Google Spanner 和 Google F1 启发

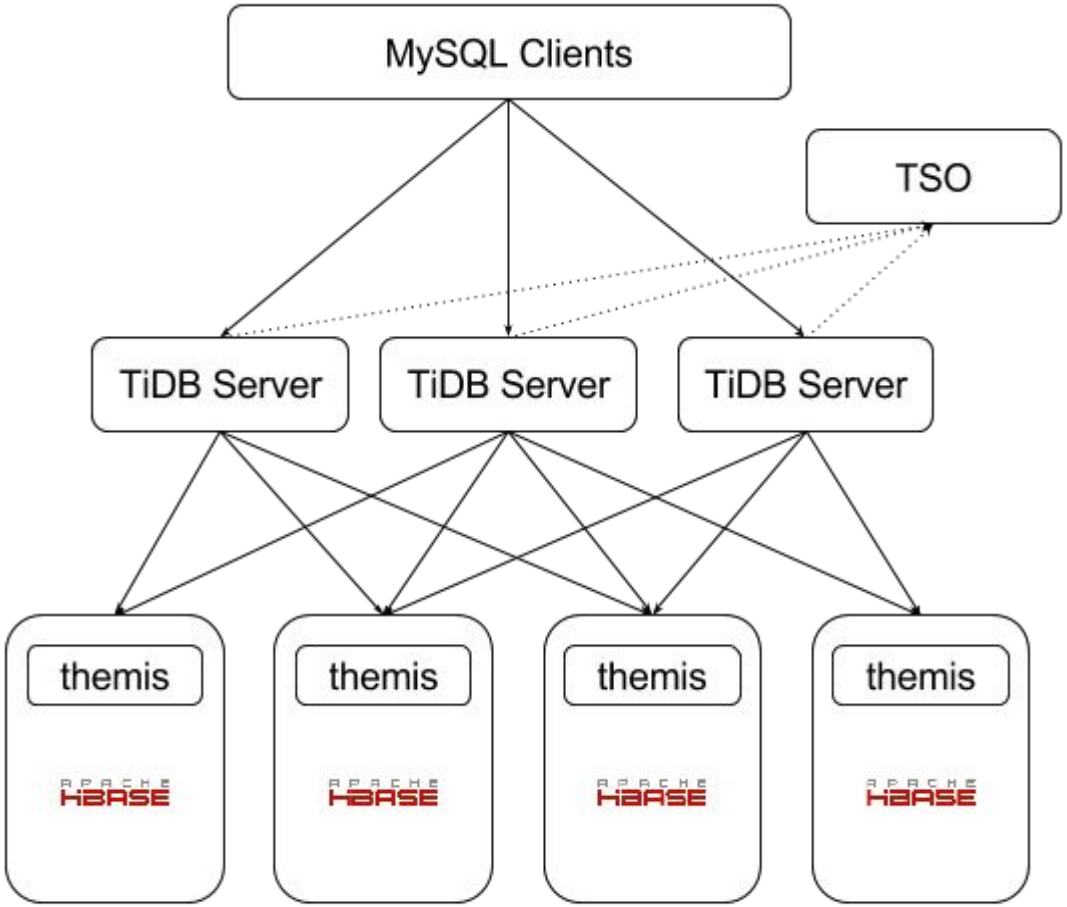
重点支持:

- 水平扩容/缩容
- 分布式事务
- 隔离级别
- 异步 schema 变更
- 容错

TiDB 架构图


MySQL-client
MySQL-client
MySQL-client
Load Balancer
MySQL Protocol Server
TiDB SQL Layer
...
MySQL Protocol Server
TiDB SQL Layer
Transactional KV Storage
KV
KV
KV
KV

TiDB 可插拔存储引擎



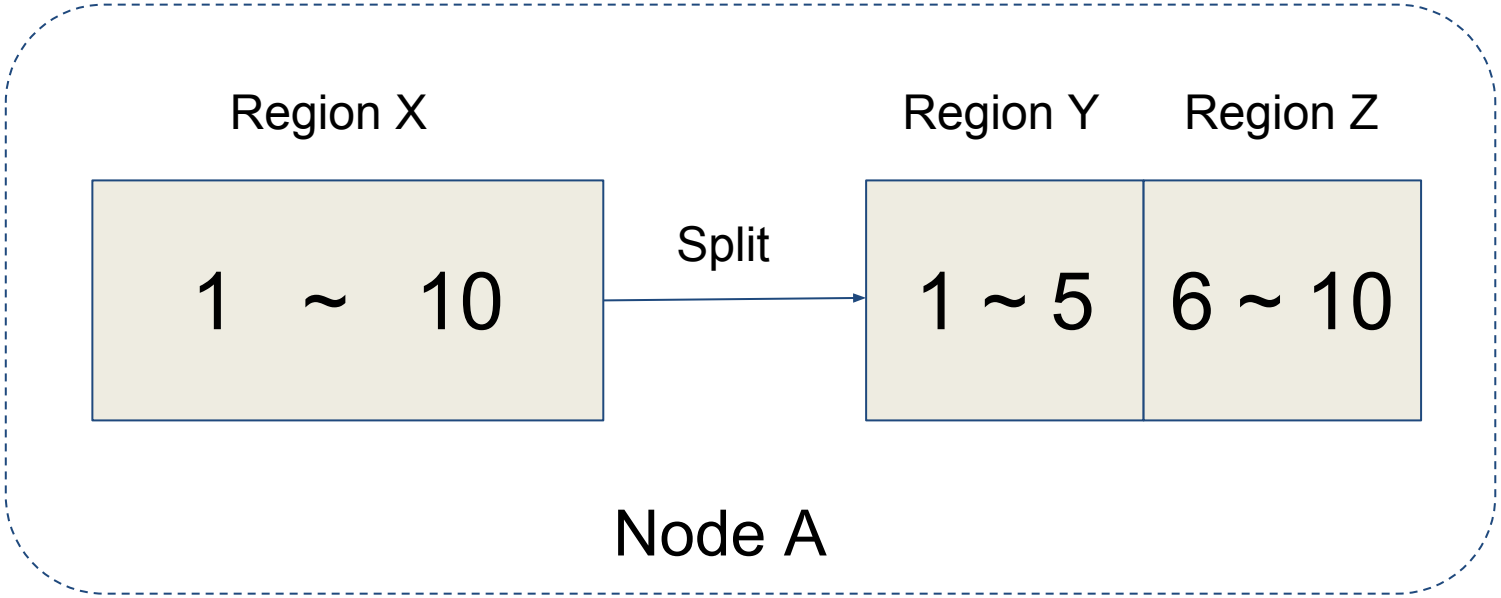
TiDB 如何解决 Scale 问题？

数据组织

```
KV {
        Node [ 1 … N ] {
                region [ 1 … X ]
        }
 }
```

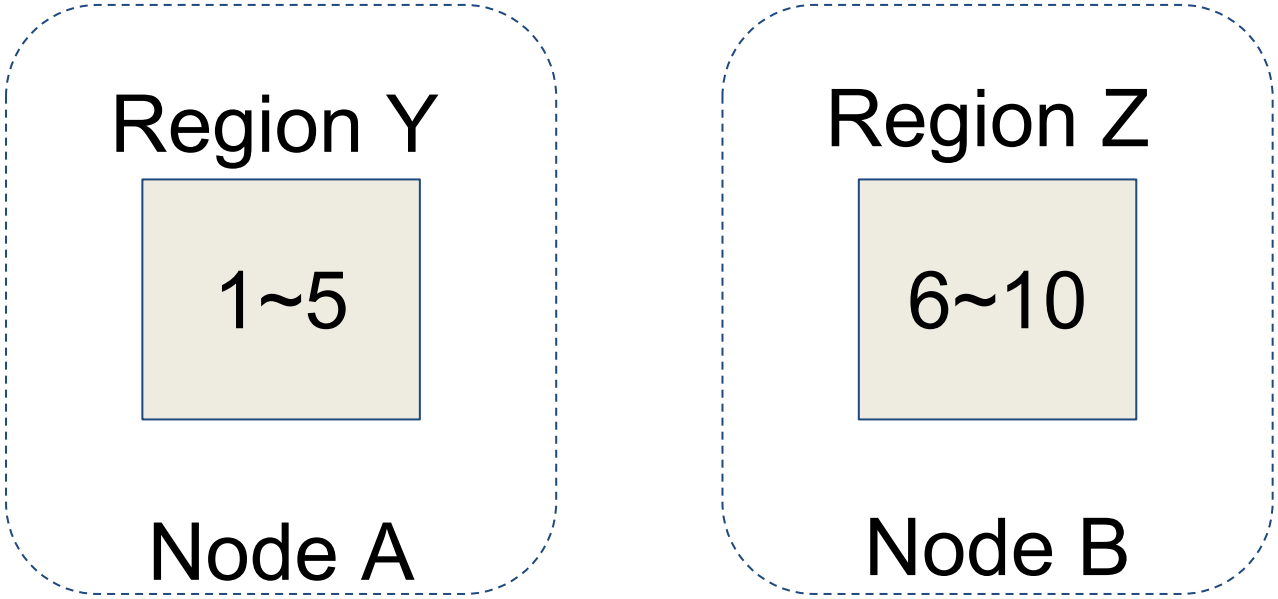
TiDB 如何解决 Scale 问题？

内部分裂

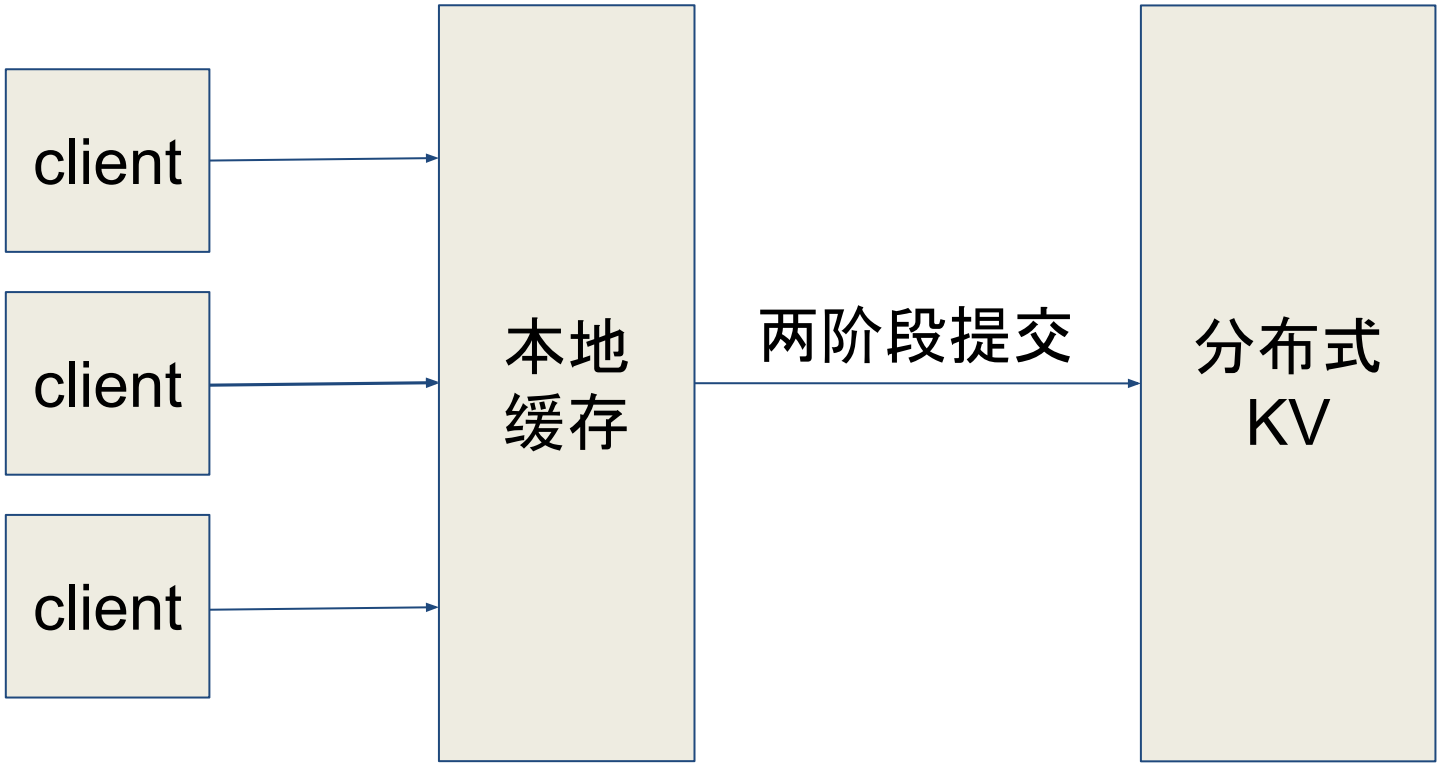


TiDB 如何解决 Scale 问题？

迁移



TiDB 如何解决分布式事务？



TiDB 分布式事务的优化

- 1PC（单个 region 内的事务）
  - 比如更新一行
- Group commit

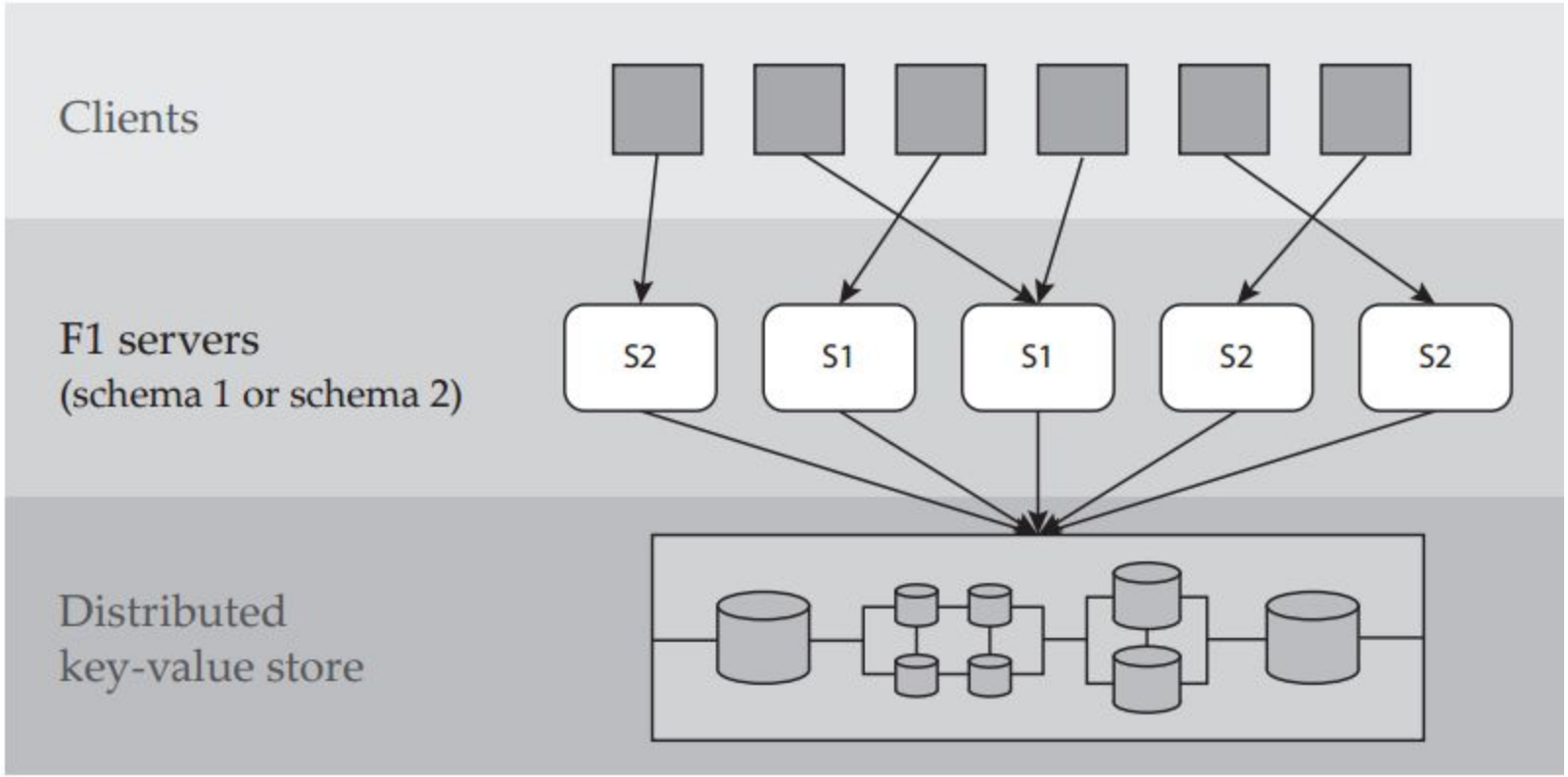
TiDB 如何解决隔离级别？

- SI + 乐观锁

- External Consistency
  - 全局的时钟服务
  - 低延迟
    - 相对Hybrid Logical Clocks的实现, 见论文
    - TrueTime in (Google Spanner)

TiDB 如何解决schema变更？

参考 Google F1 架构



图片来源：http://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/41376.pdf

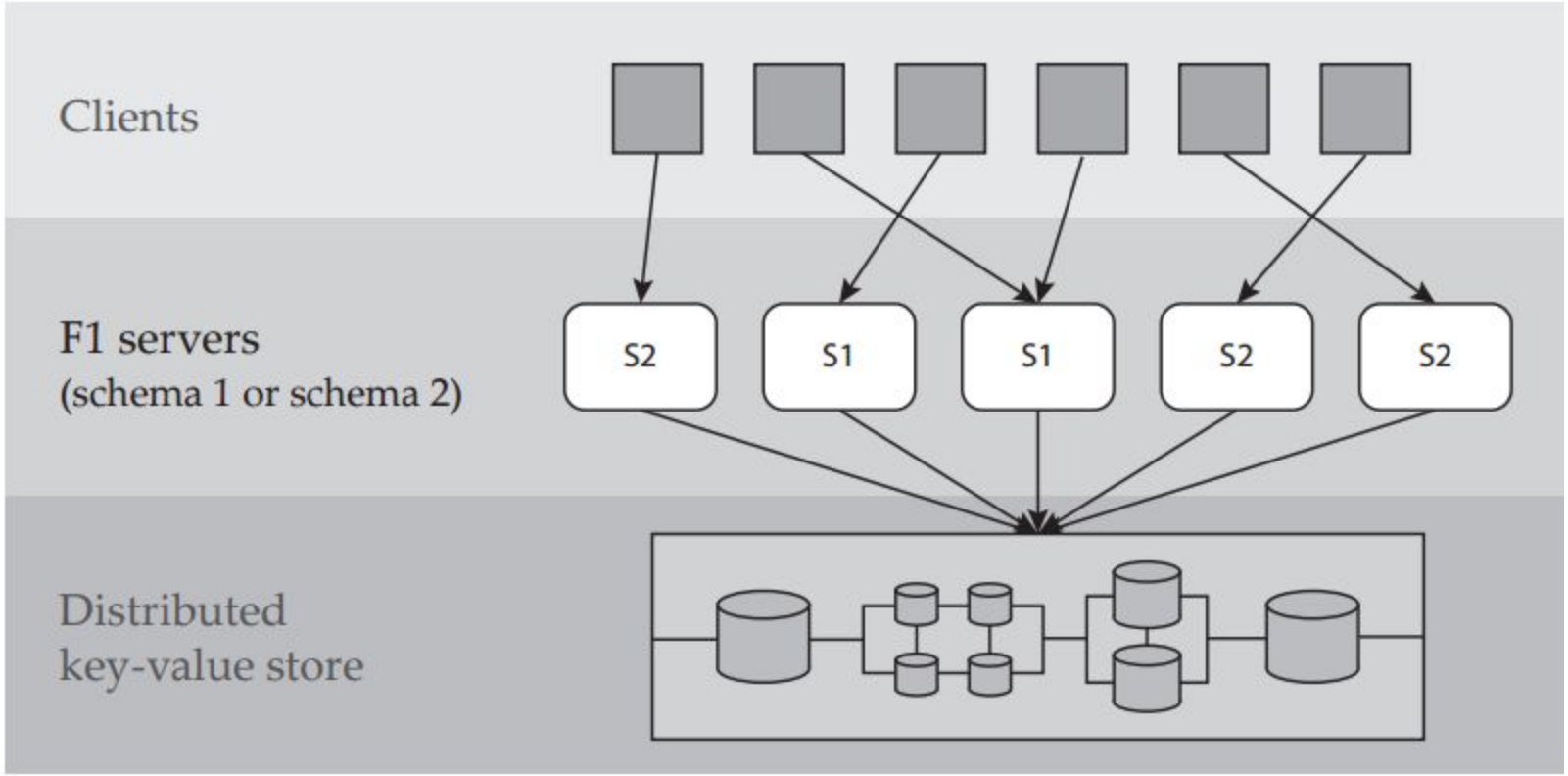
TiDB 如何解决schema变更？

假设需要：

- 从 schema S1 变成 schema S2(在表R上添加索引 I).
- 两个不同的机器M1 and M2,
  - 机器执行操作的顺序:
  - Server M2：使用 schema S2, 插入一行 r 到表 R. 由于 S2 包含索引 I, server M2 添加索引对应的 key–value.
  - Server M1, 使用 schema S1, deletes r. 然而 S1并不包含索引, 所以 M1 删除行 r 的时候没有删除索引
  - 索引对应的 kv 就残留下来了，此时如果有依赖索引的SQL就出错了

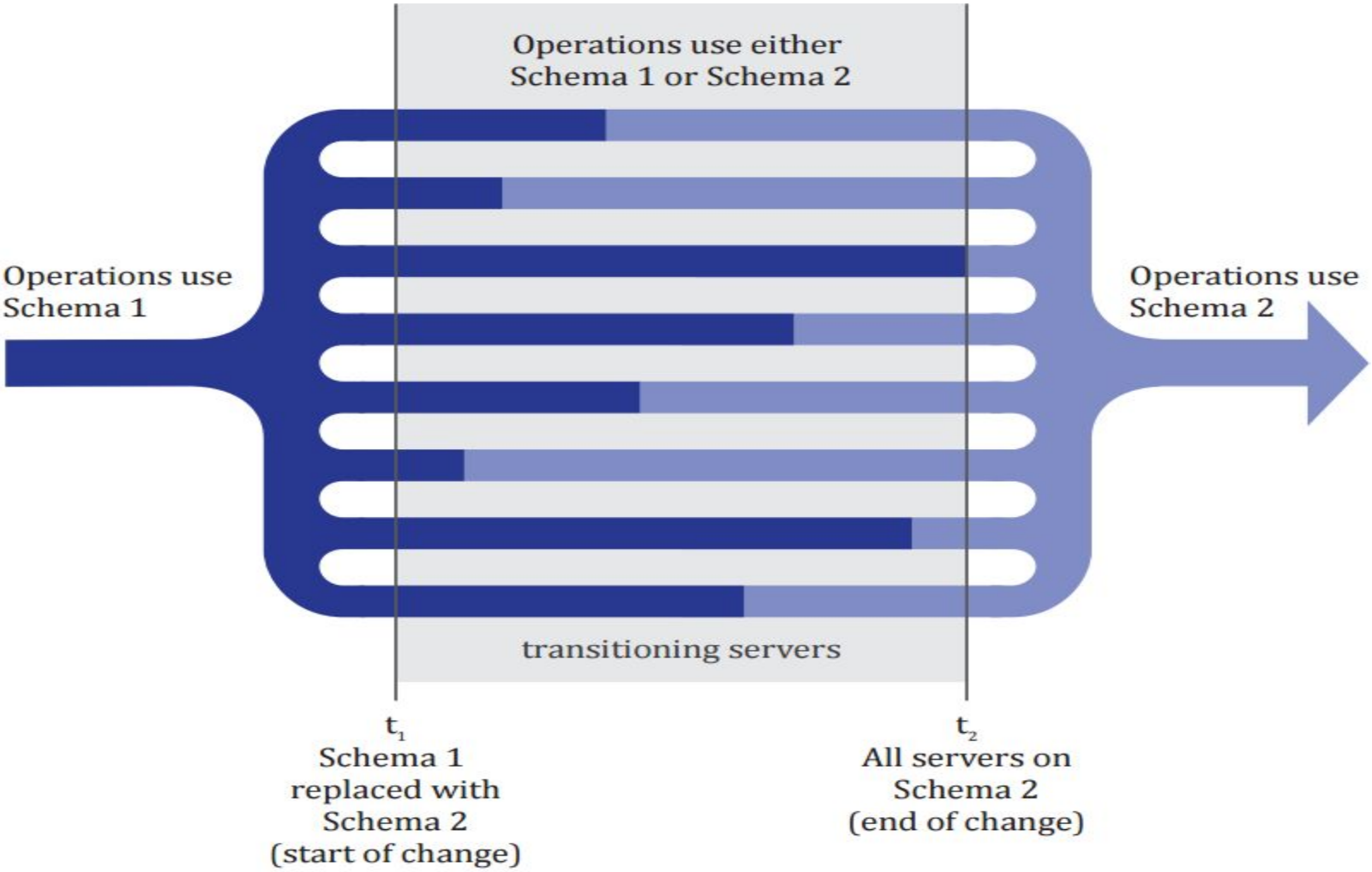
TiDB 如何解决schema变更？

状态机



图片来源：http://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/41376.pdf

TiDB 如何解决schema变更？



图片来源：http://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/41376.pdf

TiDB 如何解决schema变更？

添加索引的状态迁移过程

delete only

→ write only

→ back fill

→ public

多租户问题？

- 做到哪一层？
  - SQL ？
  - KV ？
- 怎么隔离 ？
  - 隔离到什么程度？
    - 租户之间资源隔离
    - 单个租户内部的服务隔离
- 公平调度?

测试问题？

- 有些bug只有在特定的执行顺序才会复现
- 换个思路看待:
  - 程序 = 状态机 + 输入
- 如何保证出现一次后每次都能出现
  - 确定性
  - 不确定性
    - 随机数
    - 线程的执行顺序
    - 模拟所有的输入

更多技术细节

更多细节请参考 TiDB 文档和代码，以及相关的参考论文

<<从零开始写分布式数据库>>

Thanks!