

分布式数据库原理 和架构设计

演讲人：蒲聪

跨界互联
数聚未来

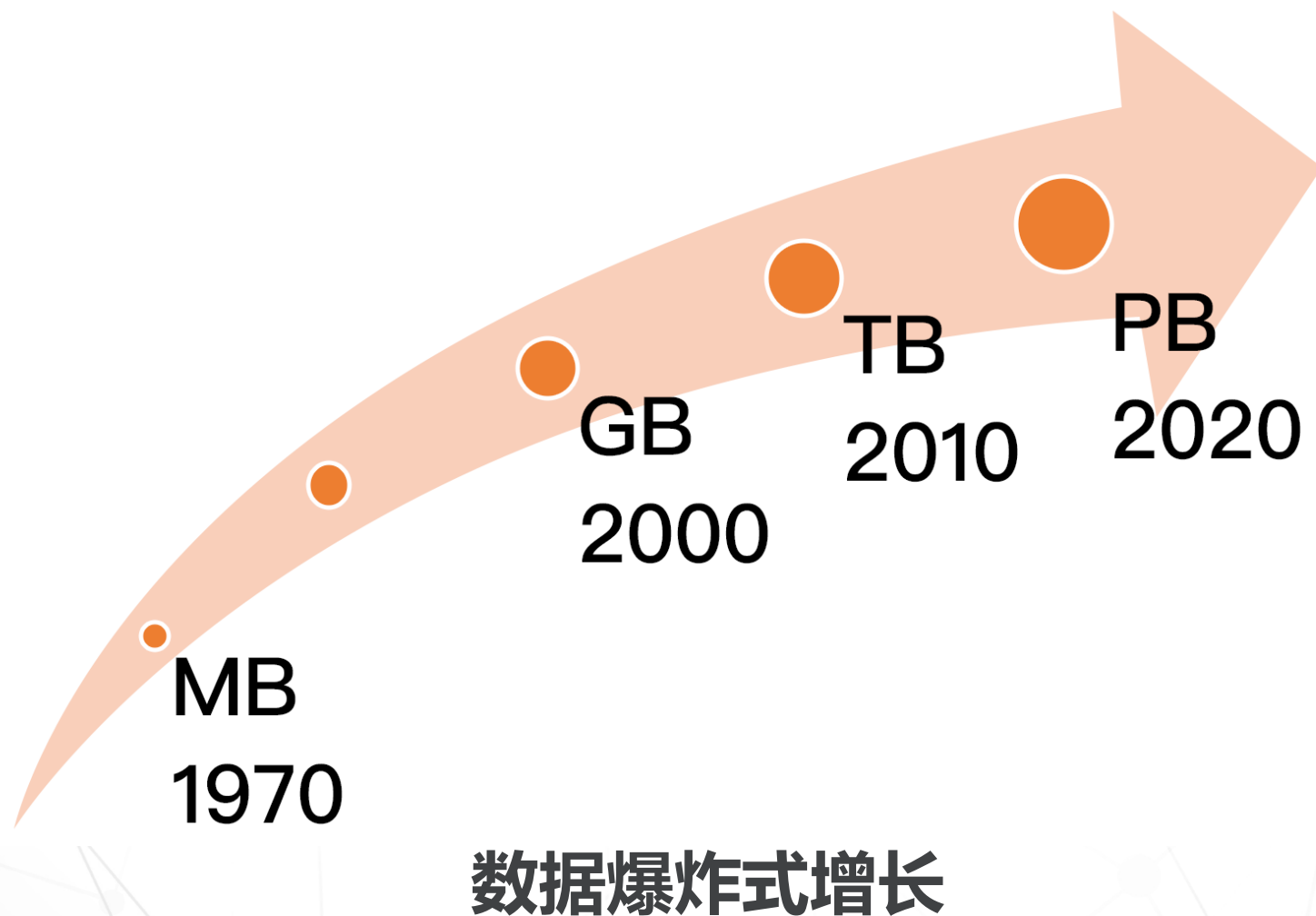
第四届中国数据分析师行业峰会
CHINA DATA ANALYST SUMMIT

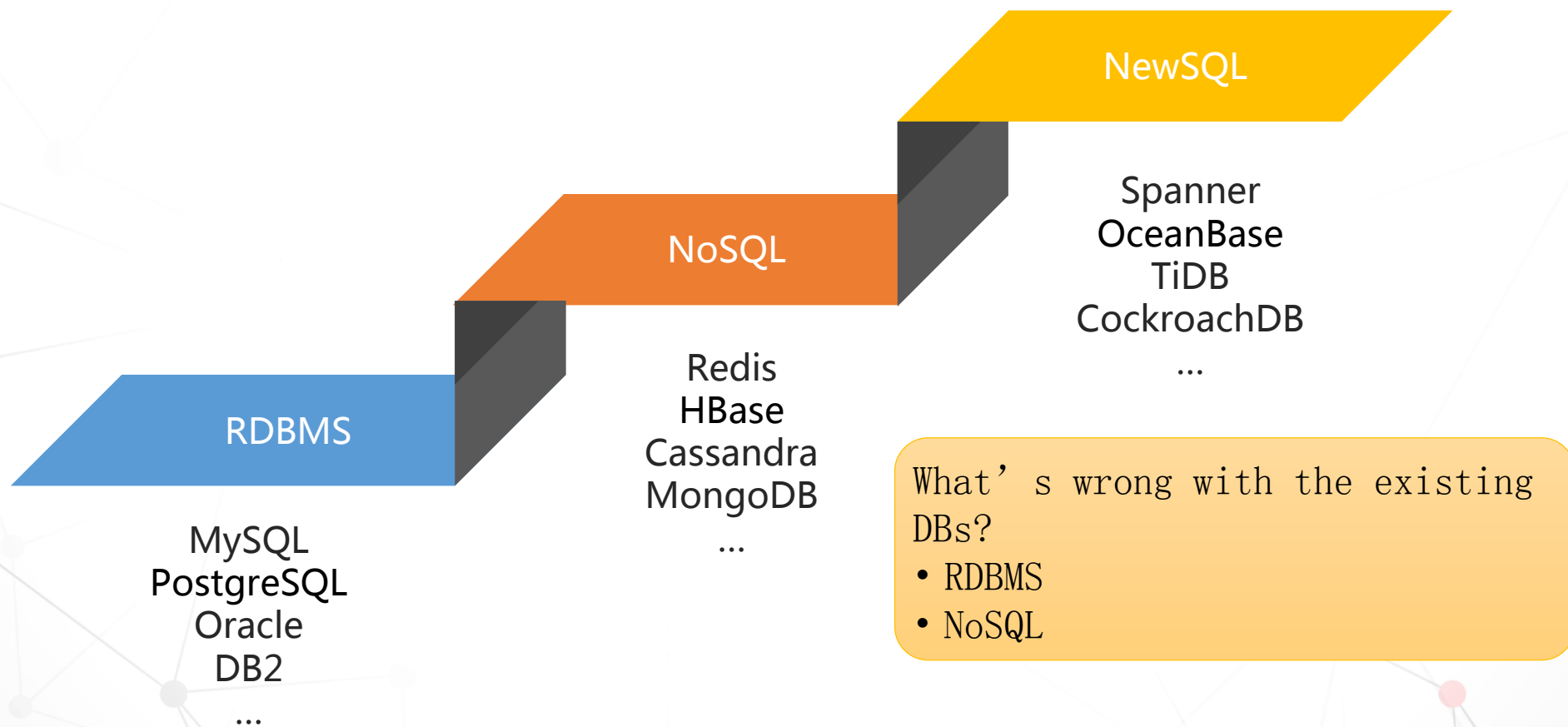
北京 中国大饭店 2017.07

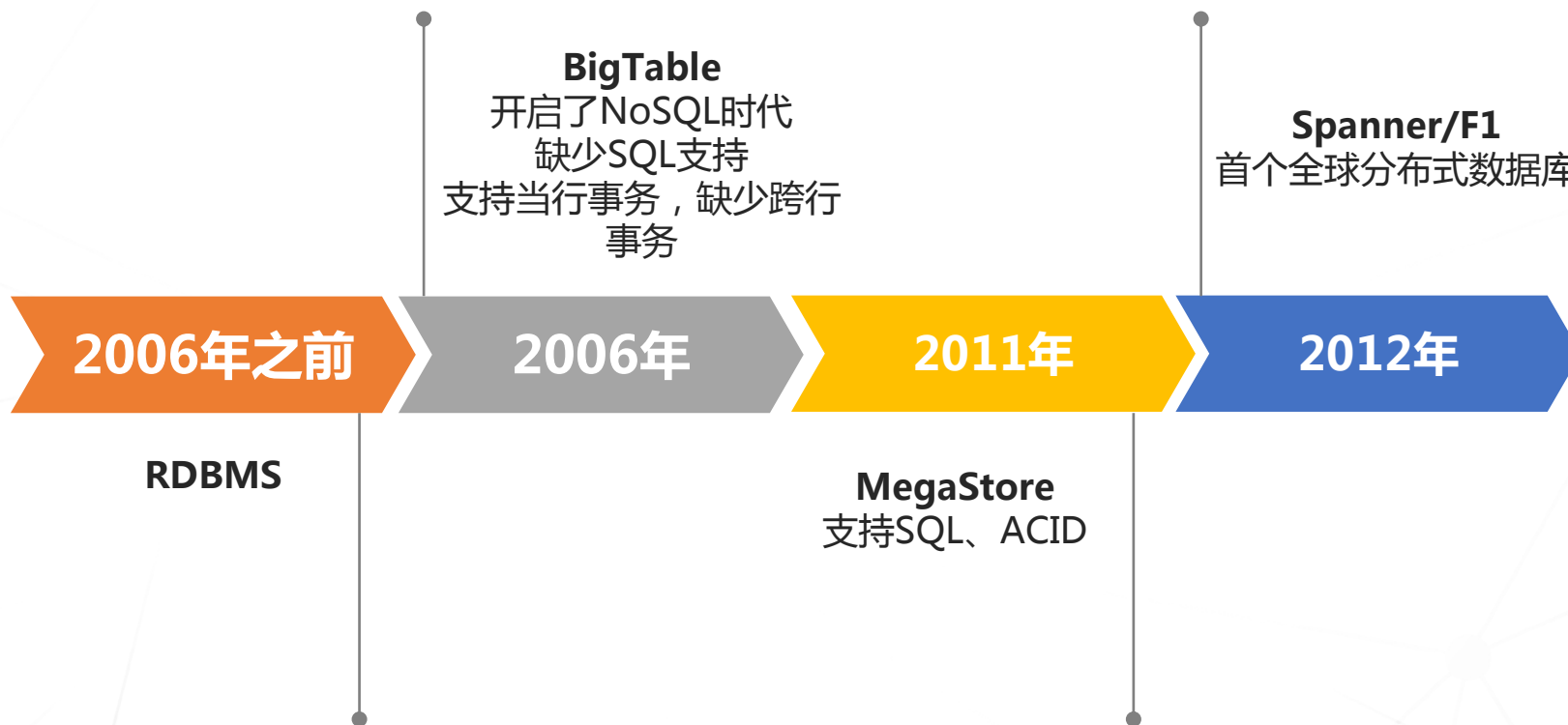


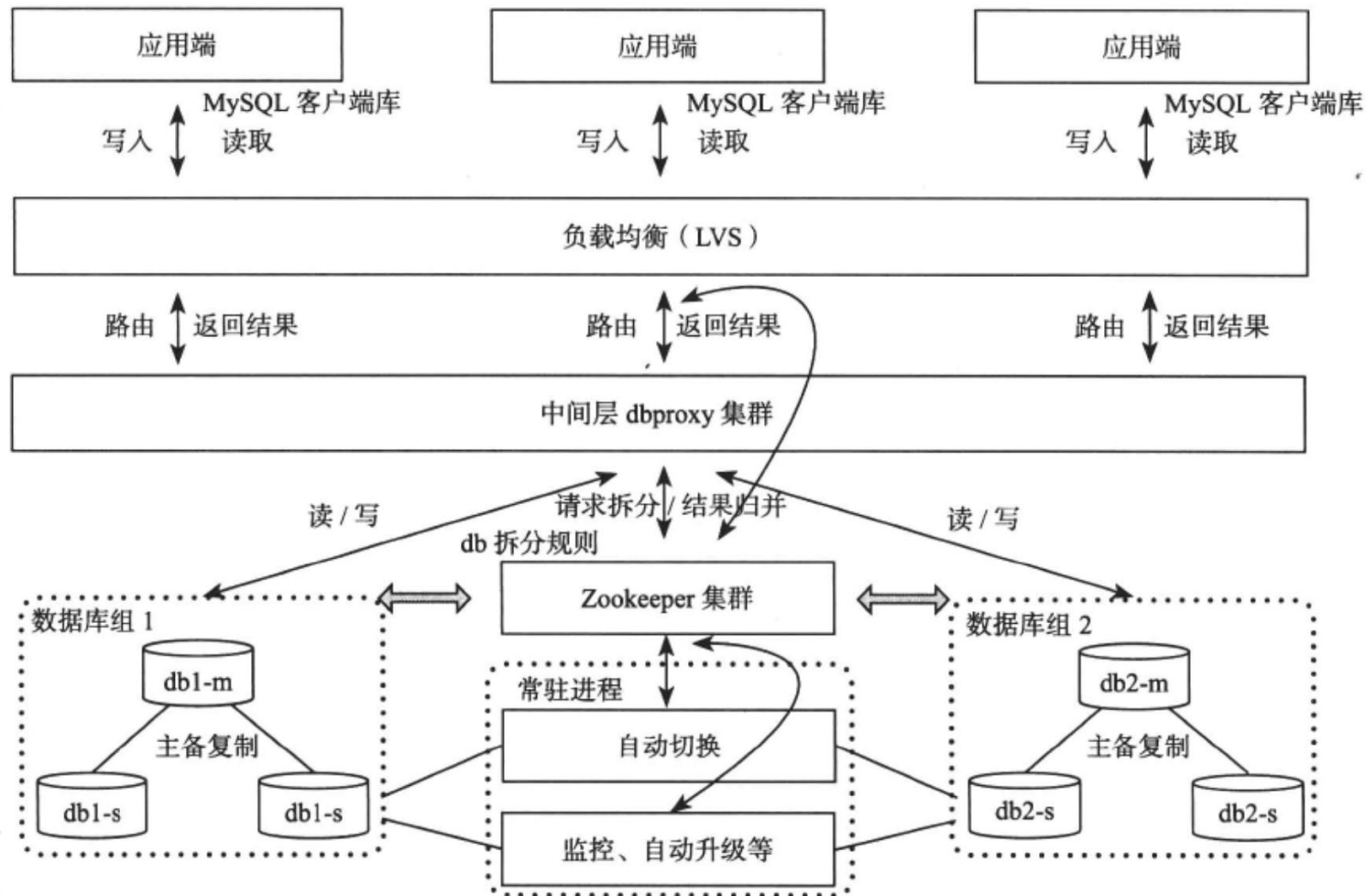
去哪儿资深DBA，拥有6年的MySQL和HBase数据库运维管理经验，曾就职于百度、艺龙，2014年6月加入去哪儿网。工作期间负责MySQL数据库的运维架构和自动化平台设计，从无到有建立去哪儿网HBase运维体系，在MySQL和Hbase数据库上有丰富的架构、调优和故障处理的经验。目前专注于分布式数据库领域的研究和实践工作。

- 1 分布式数据库的诞生
- 2 面对大数据-过去和现在
- 3 开源分布式数据库TiDB
- 4 适用场景和未来计划









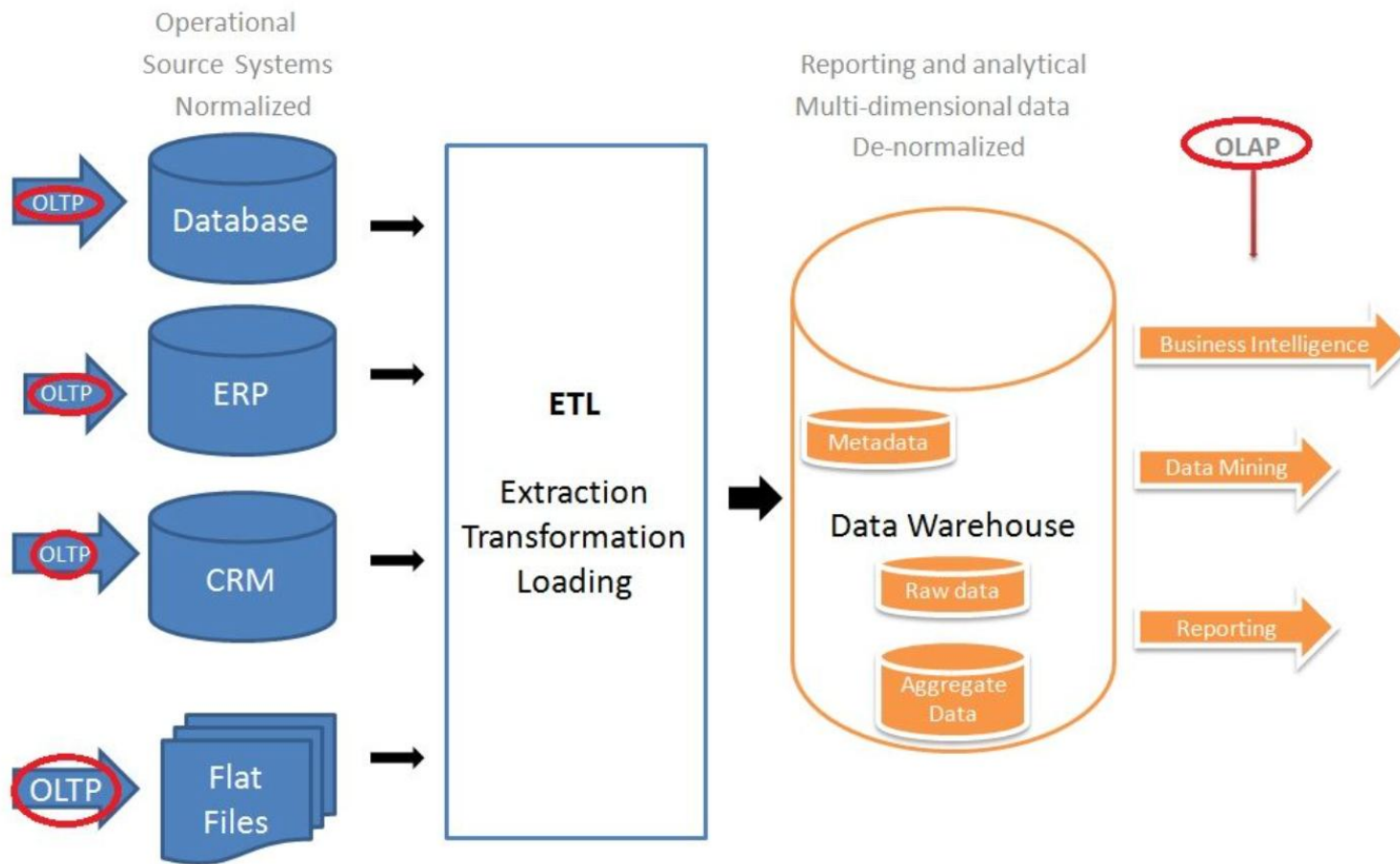
• 历史的努力

- 百度的DDBS
- youtube的Vitess
- 淘宝的Cobar、TDDL
- 基于Cobar的MyCat
- 360的Atlas
- 京东的JProxy

What's wrong with the Proxy?

- 执行计划
- 分布式事务，跨节点join
- 主从延时
- 扩容复杂



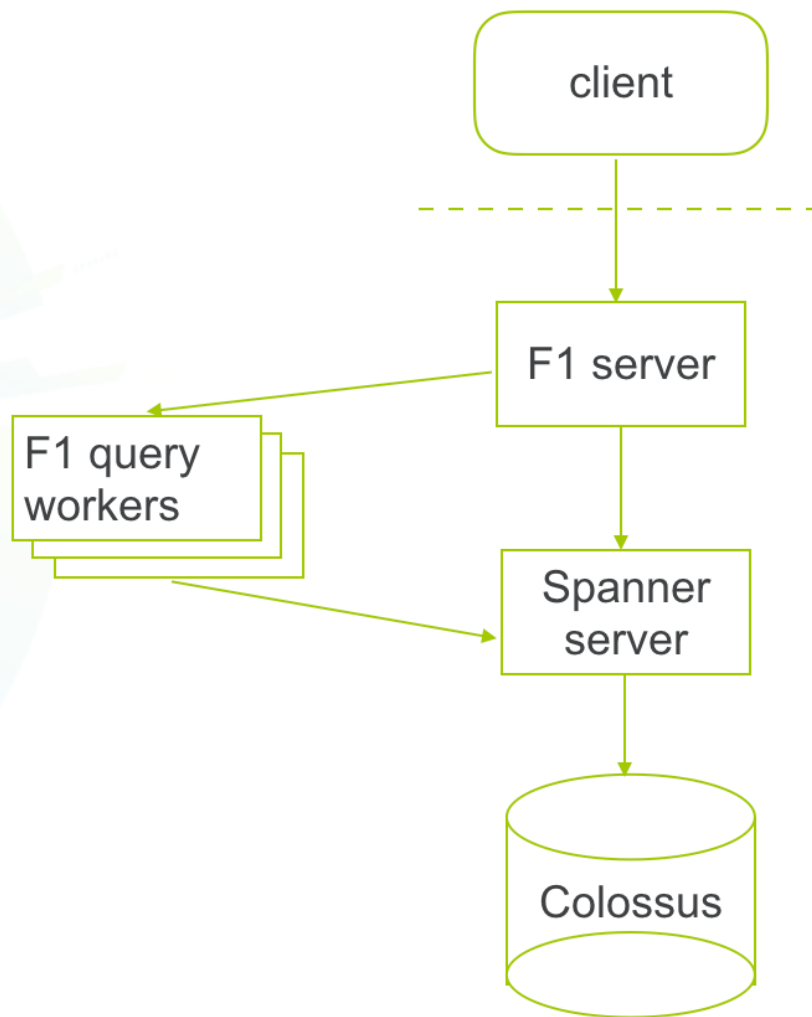


OLTP (Online Transaction Processing)
OLAP (Offline Analytical Processing)



OLTP+OLAP=HTAP (Hybrid Transactional/Analytical Processing)







• What is TiDB

- Inspired by the design of Google F1 and Google Spanner
- Compatible with MySQL, at most cases
- Scale is easy
- 7/24 availability, even in case datacenter outages
- Design Goals: 100% of the OLTP scene and 80% of the OLAP scene
- Open source

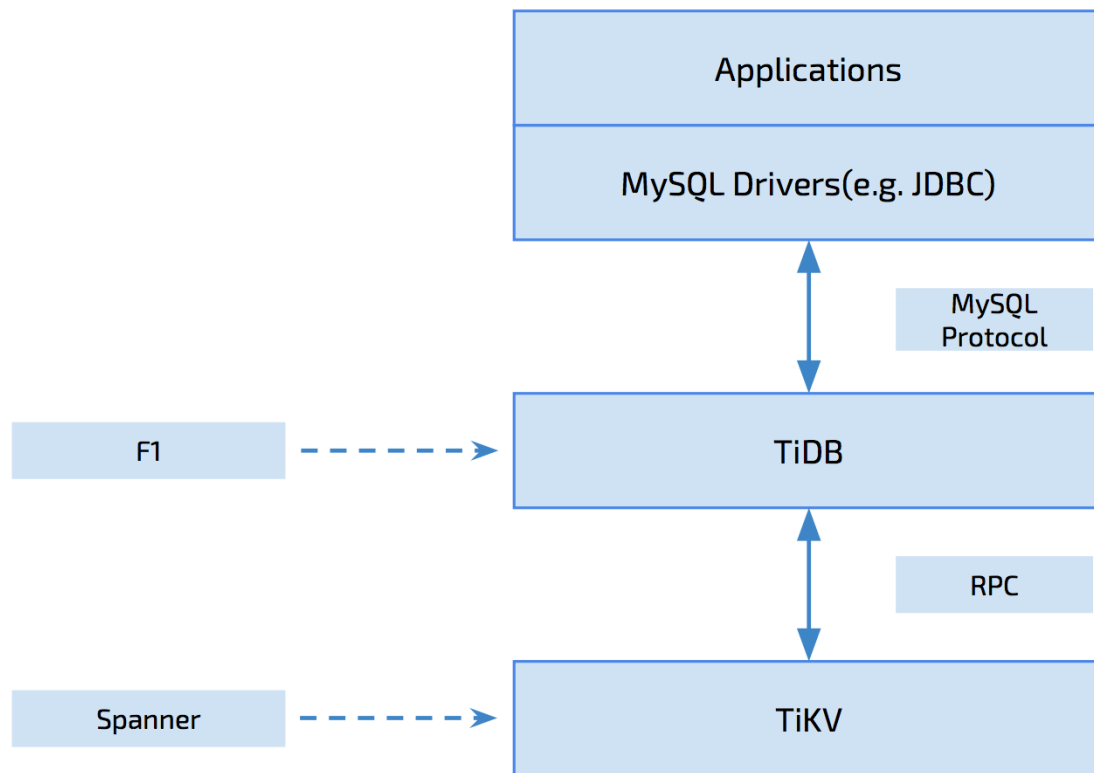
核心特性

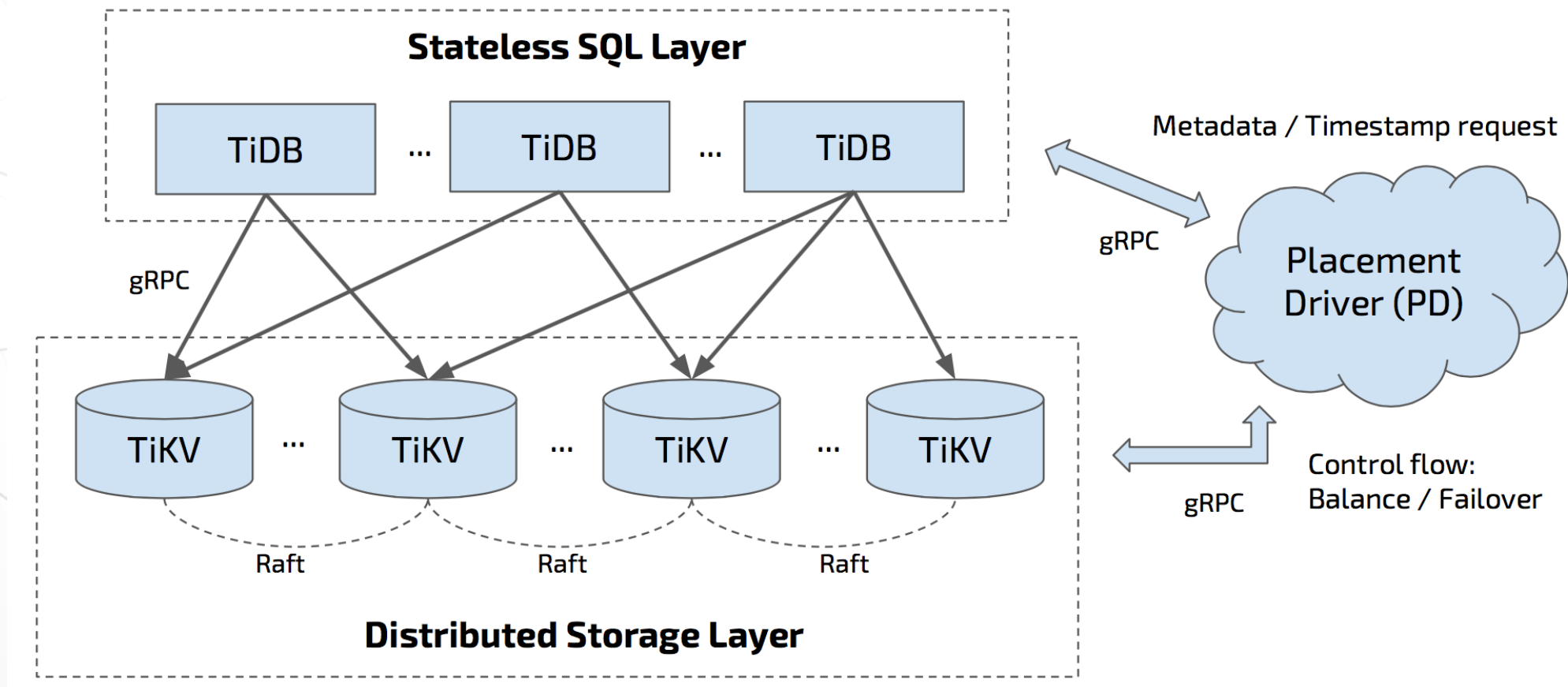
水平线性扩展

故障自恢复高可用

SQL支持+ACID事务+OLAP

跨数据中心数据强一致





TiDB Server

负责接收SQL请求，处理SQL相关逻辑，通过PD找到所需数据的TiKV地址
无状态、本身不存储数据，只负责计算，可以无限水平扩展

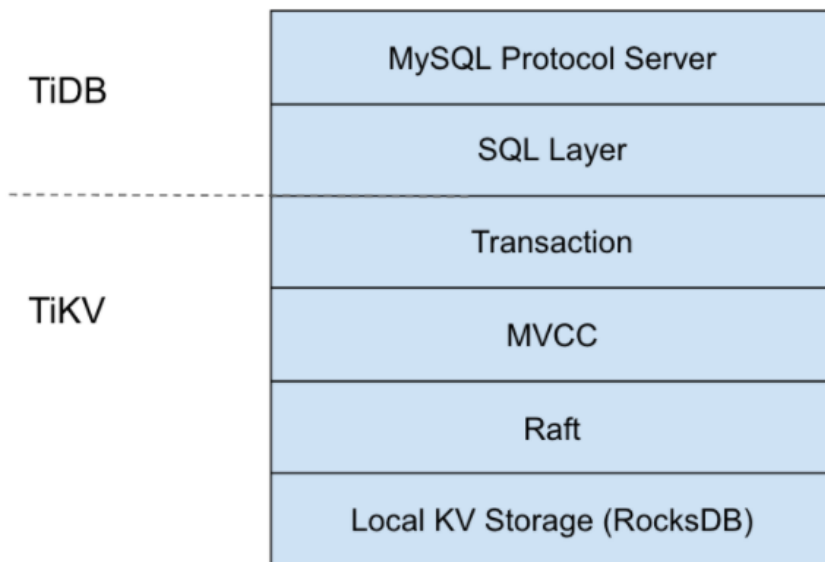
PD Server

存储集群元信息（某个Key存储在哪个TiKV节点）
对TiKV集群进行调度和负载均衡（数据迁移、Raft group leader迁移等）
分配全局唯一且递增的事务ID

TiKV Server

负责存储数据，存储数据基本单位是Region，每个TiKV节点负责多个Region
TiKV使用Raft协议做复制，保证数据一致性和容灾
数据在多个TiKV之间的负载均衡由PD调度，以Region为单位调度

- 高度分层
- SQL层和KV存储层分离
- 使用Raft来保证副本一致性和可扩展性
- 没有分布式文件系统



- RockDB

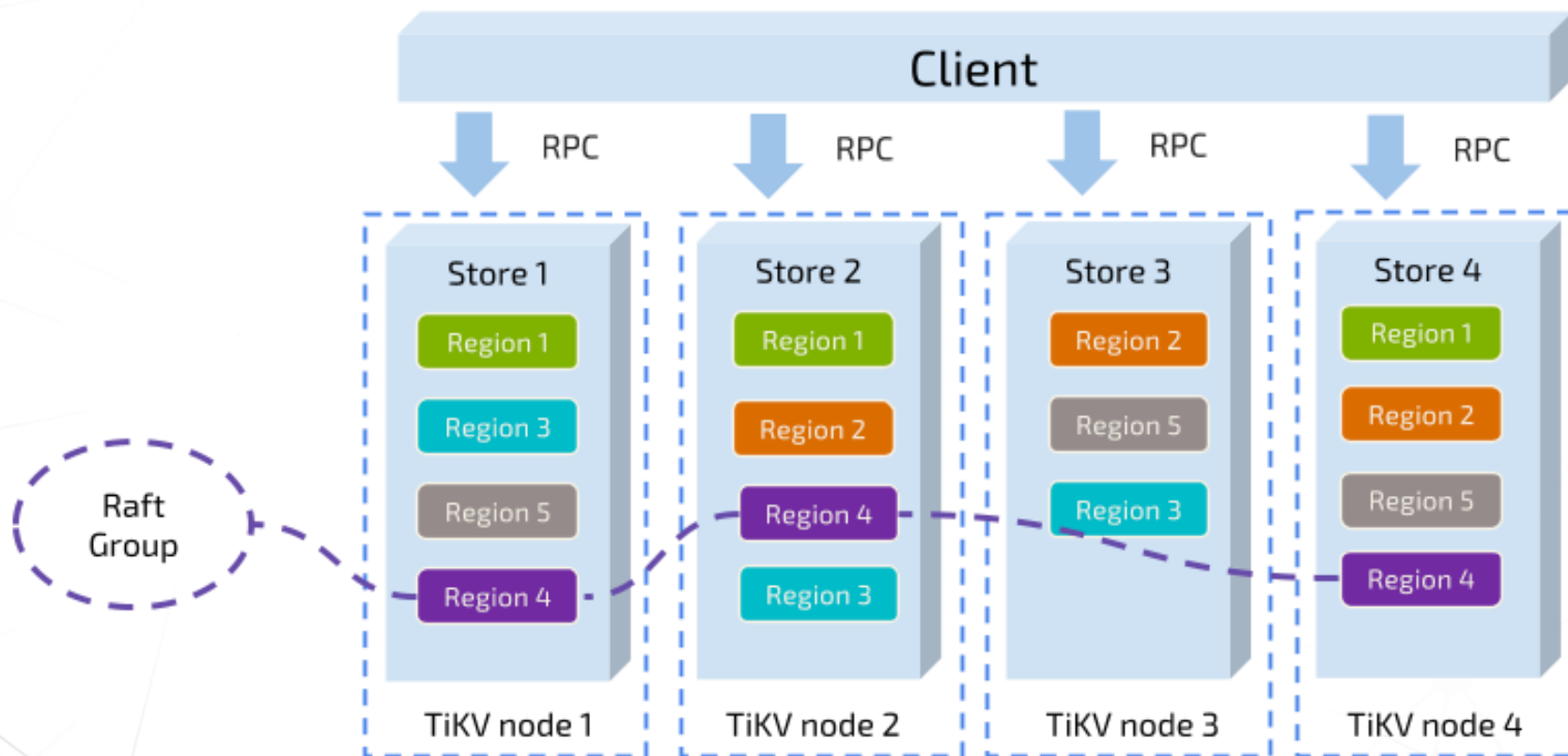
- fast and stable
- 不能容忍磁盘故障
- 扩展依赖磁盘容量

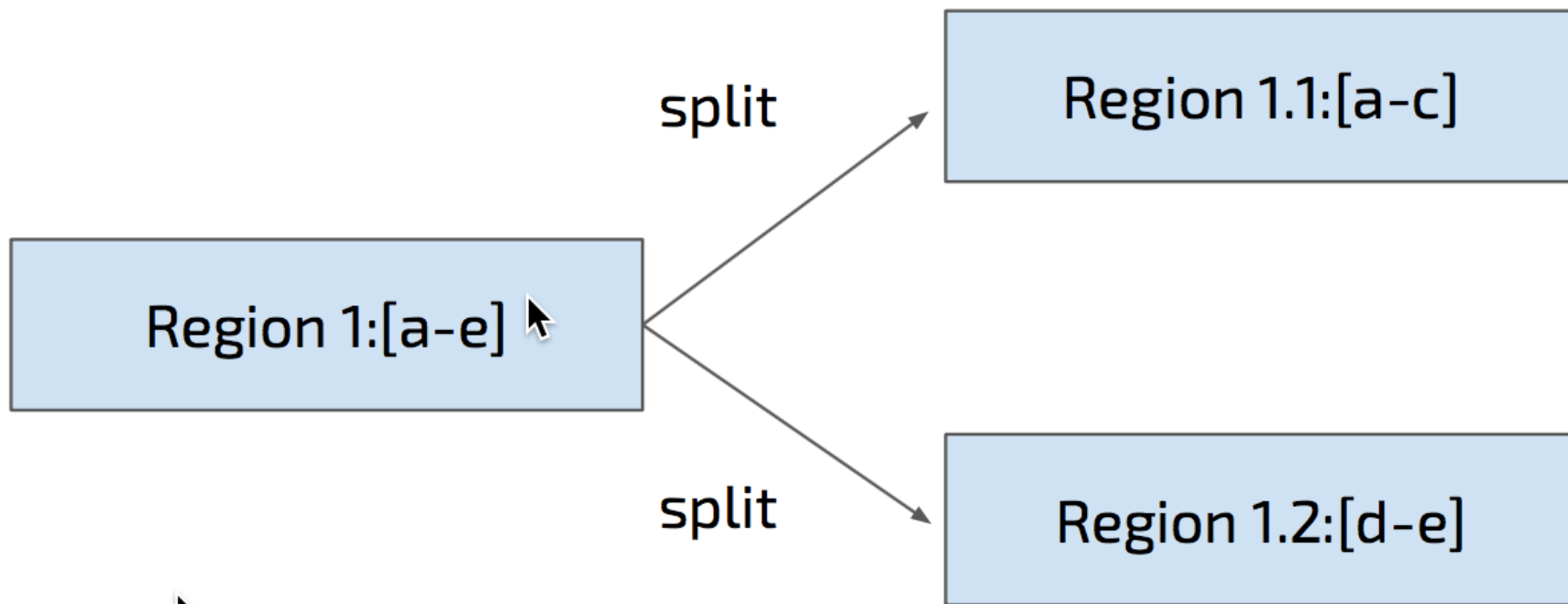
- Raft

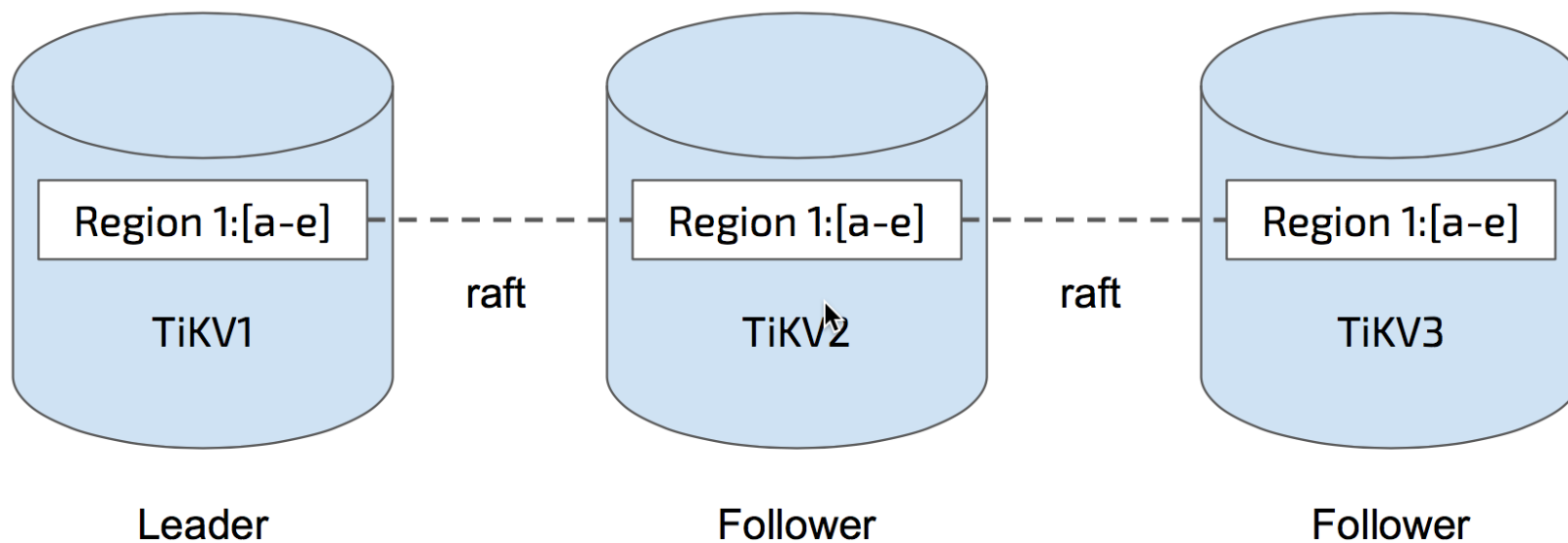
- 单机失效，数据不丢失不出错
- 分布式协议，与Paxos等价，但更易理解

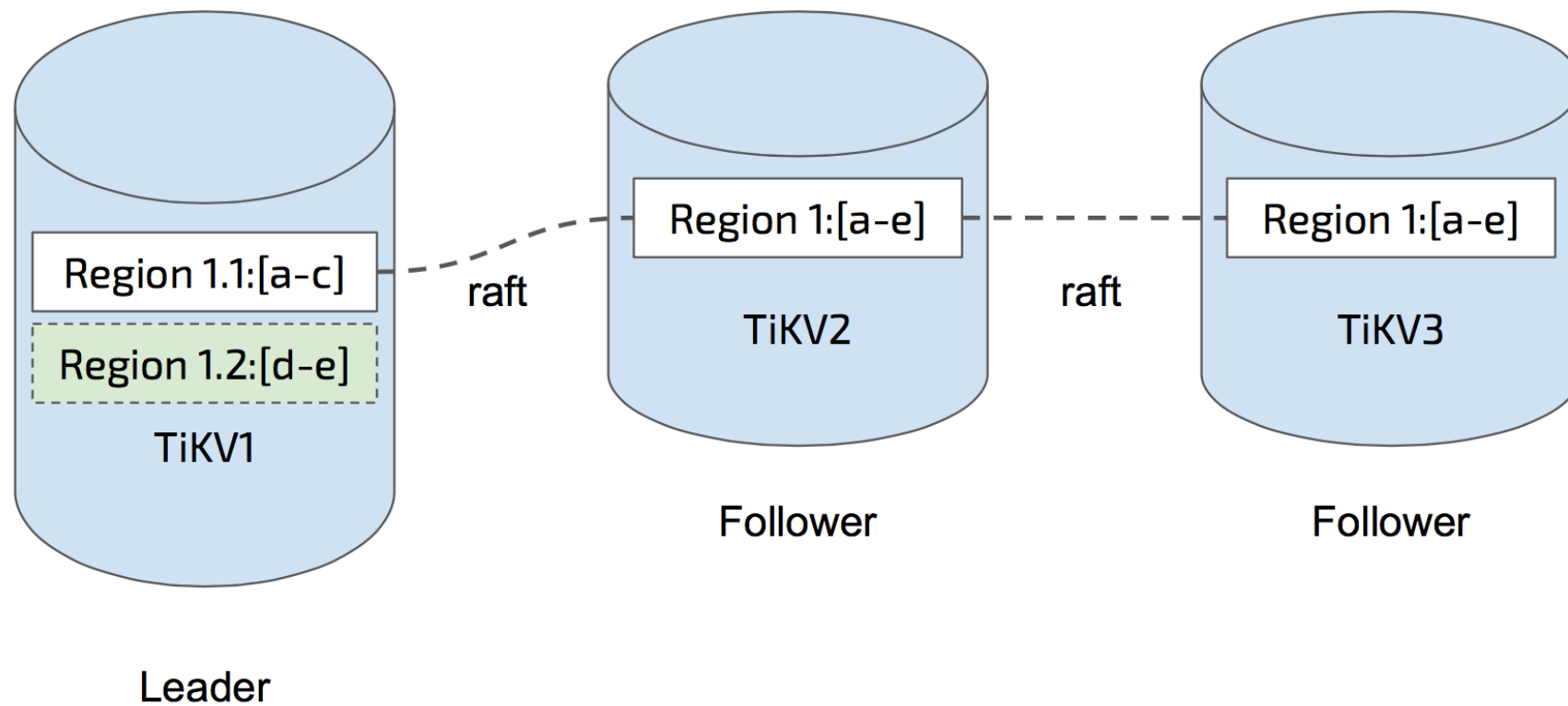
- Region

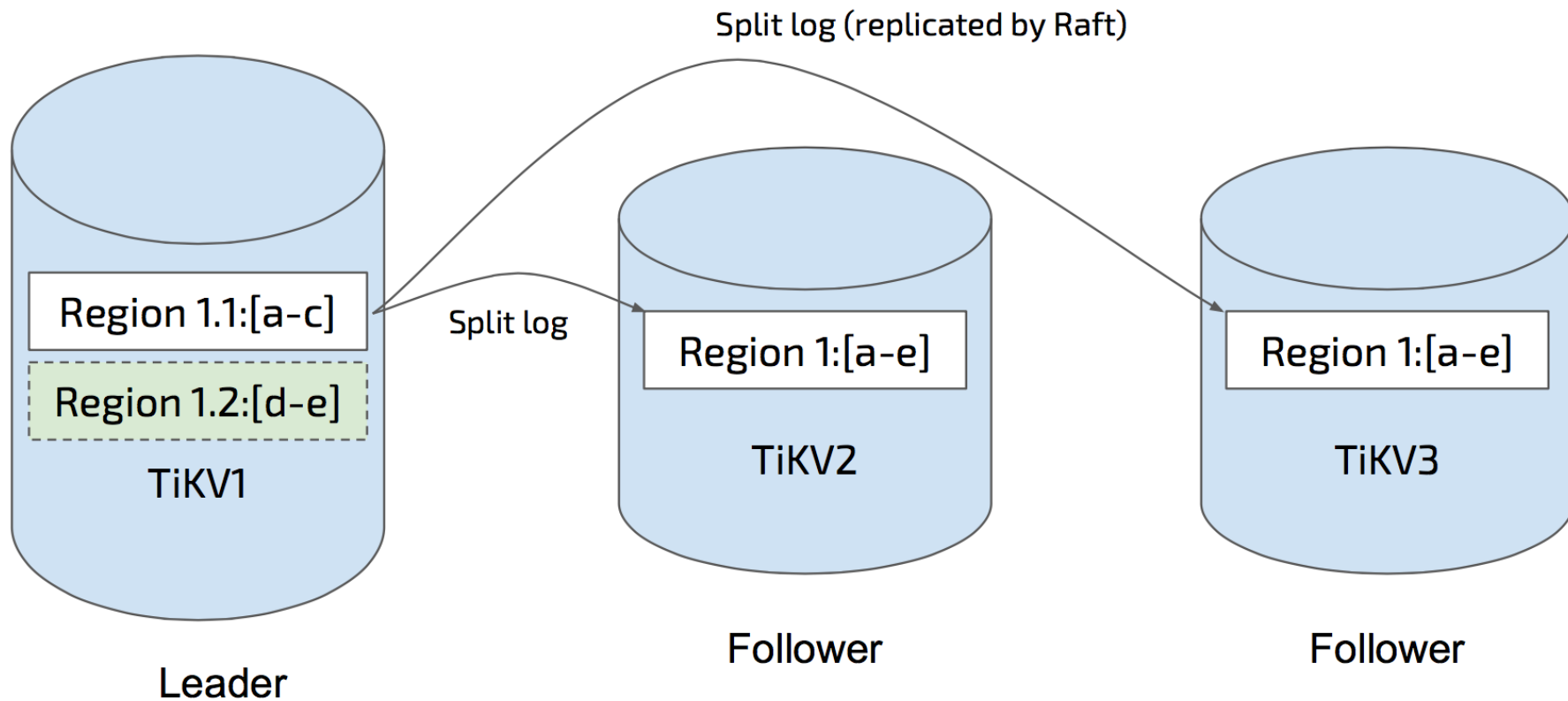
- Hash partitioning or Range partitioning
- 以Region为单位，数据分散在各个节点，并实现负载均衡
- 以Region为单位Raft group，读写都通过Leader进行

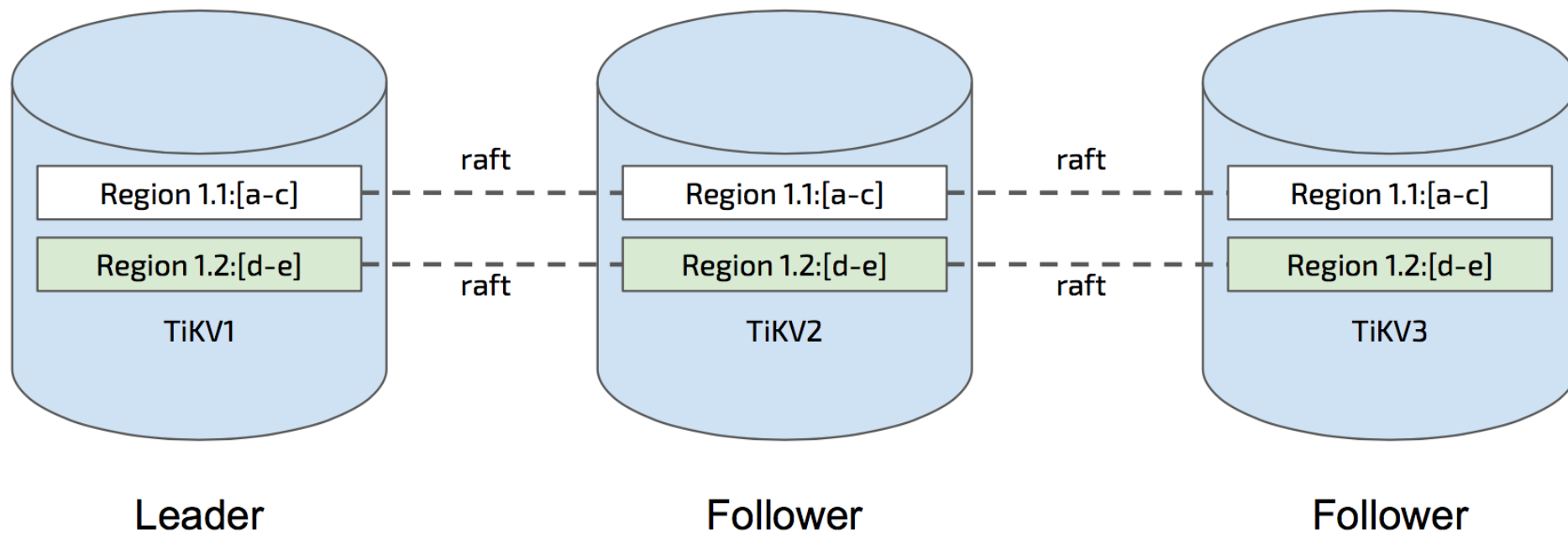


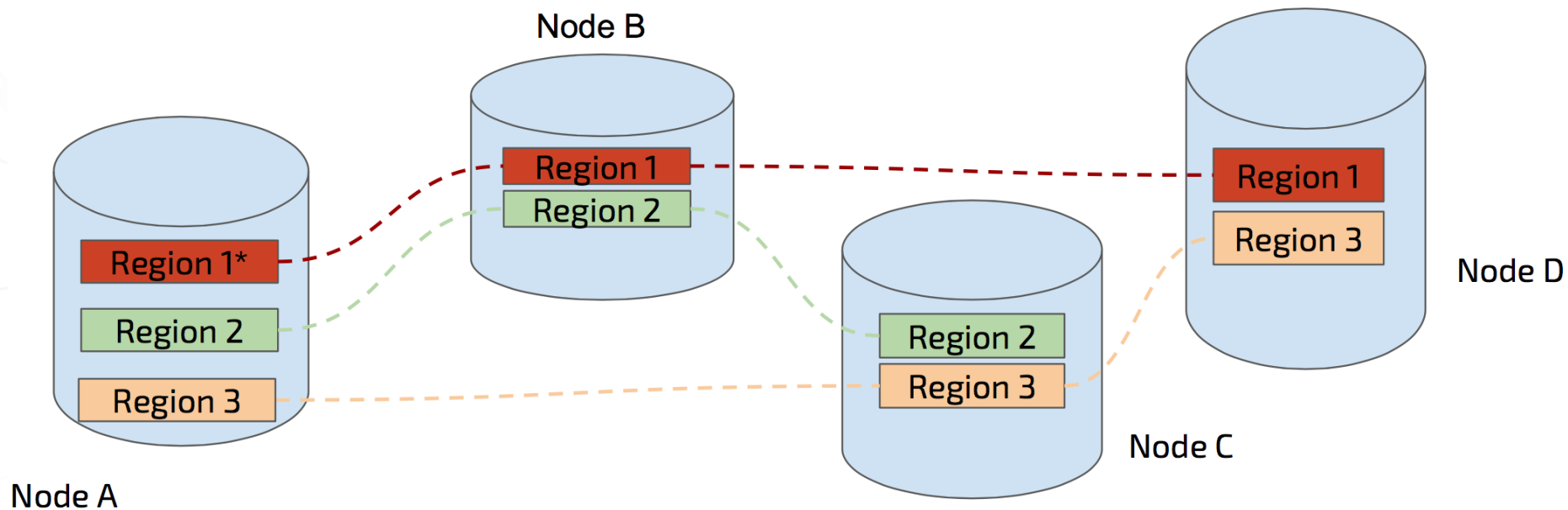


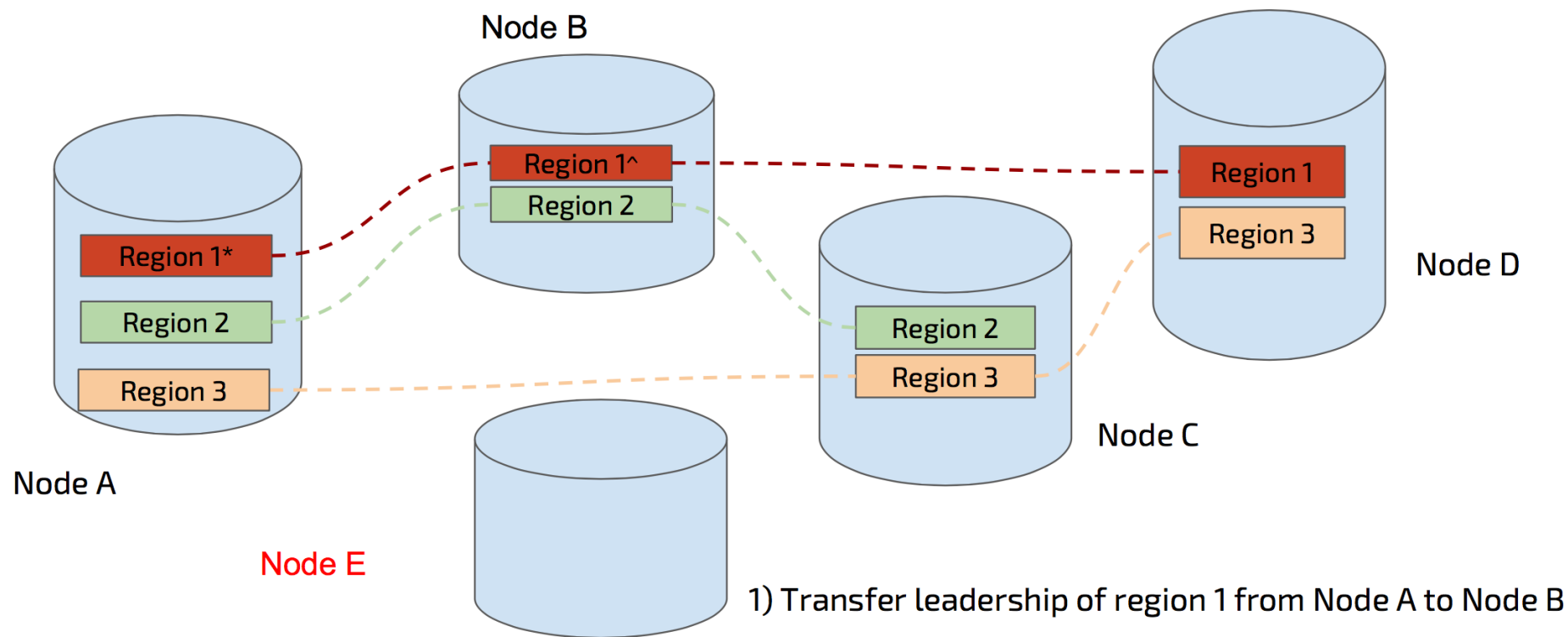


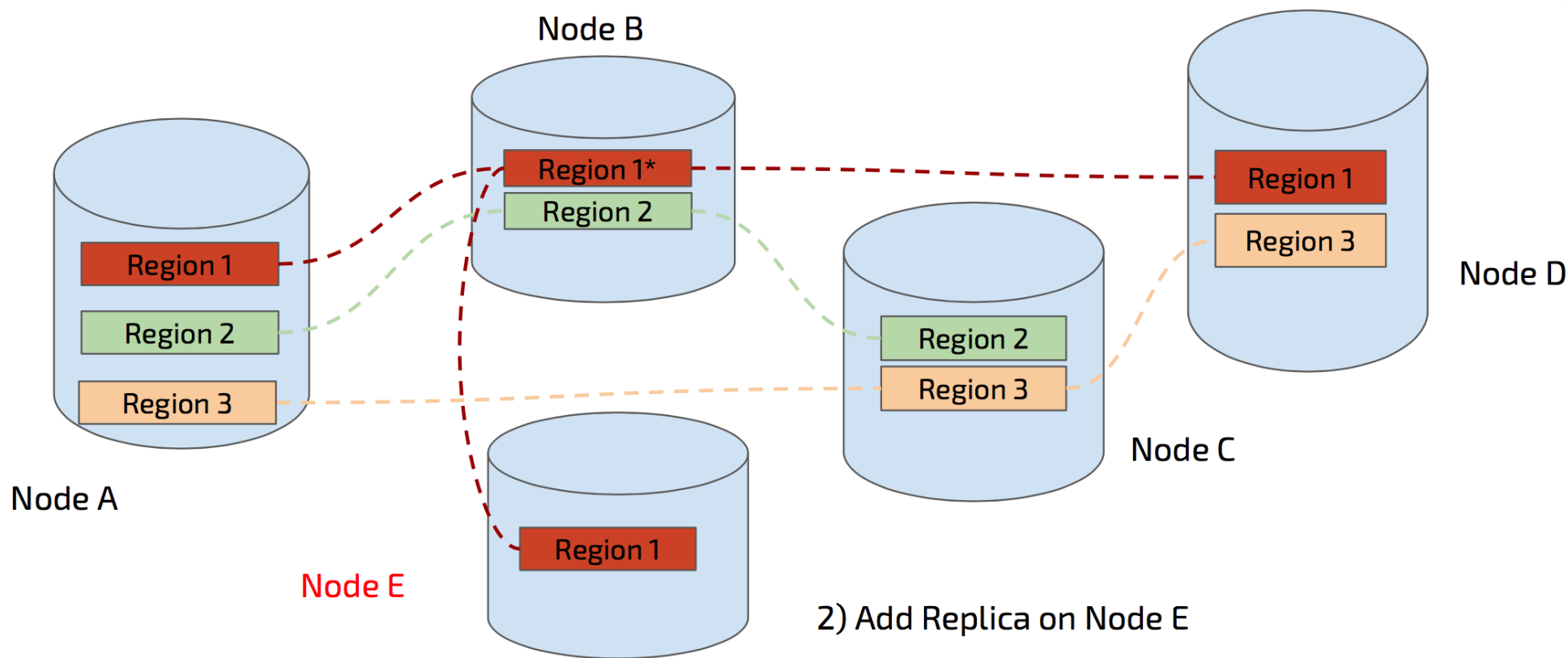


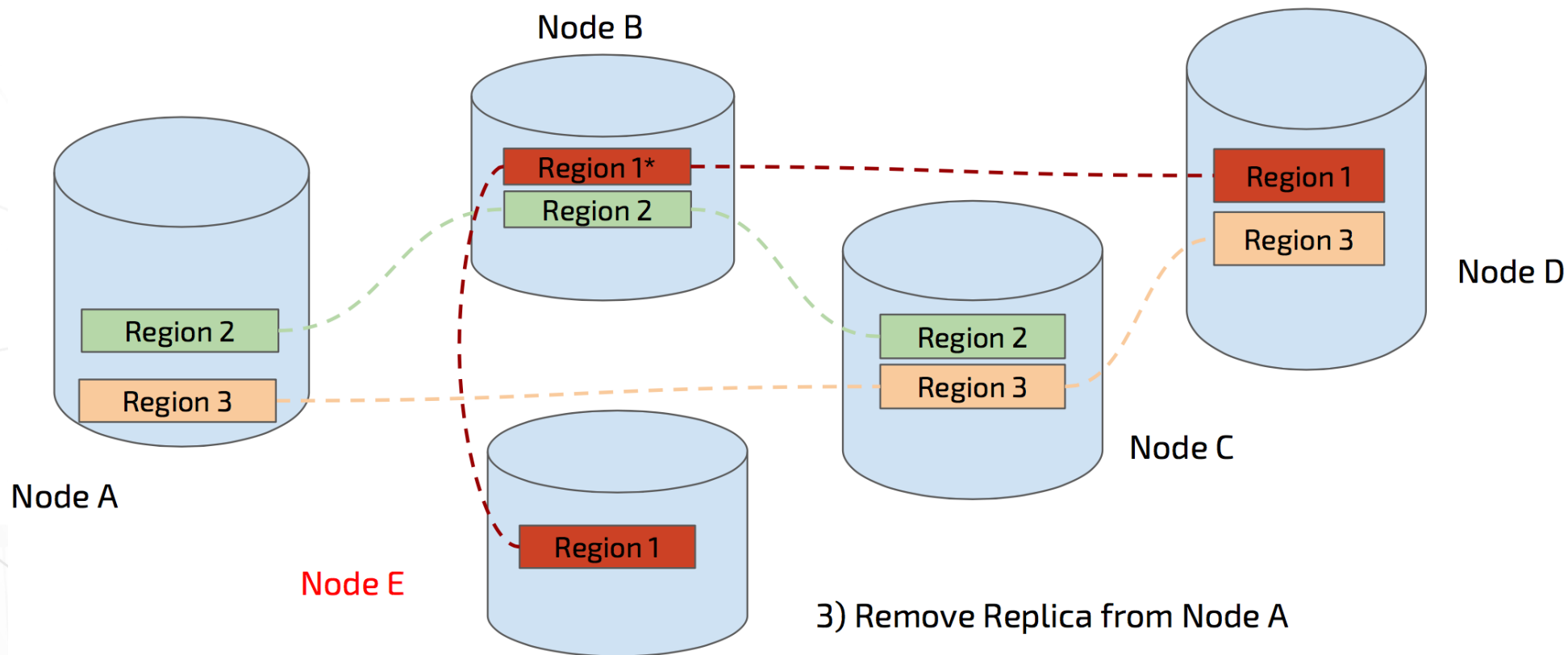












- Data is tagged with version
 - Key_version:Value
- Lock-free snapshot read
- Reading Data from History Version

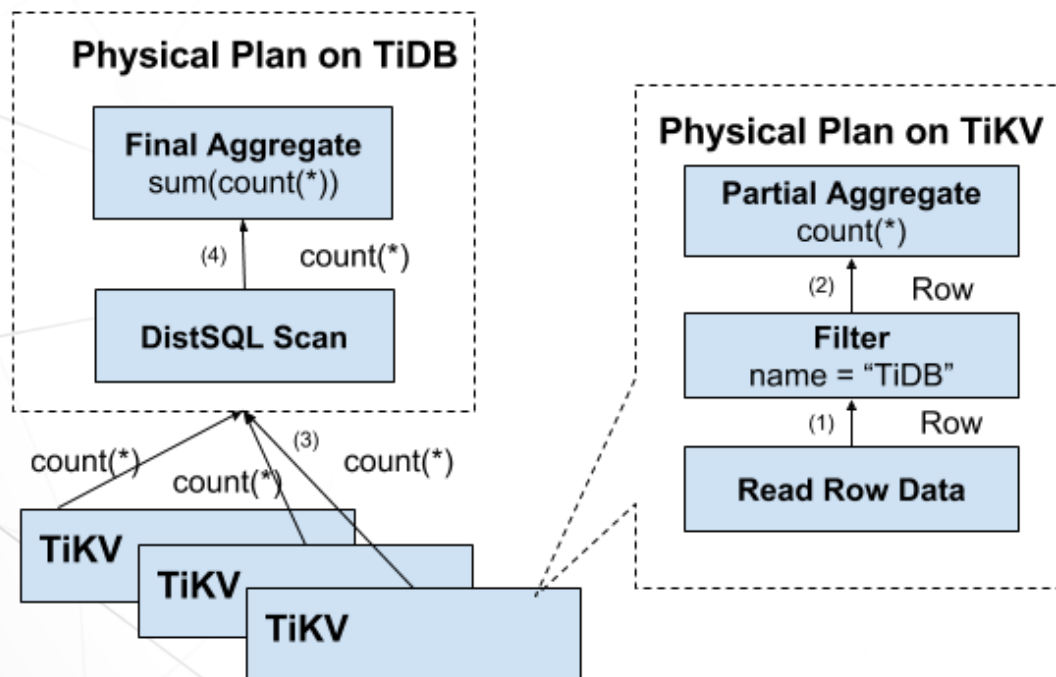
```
Key1 -> Value  
Key2 -> Value  
...  
KeyN-> Value
```

```
Key1-Version2 -> Value  
Key1-Version1 -> Value  
...  
Key2-Version3 -> Value  
Key2-Version2 -> Value  
Key2-Version1 -> Value  
...  
KeyN-Version2 -> Value  
KeyN-Version1 -> Value  
...
```

• 事务

- 基于Percolator模型
- 乐观锁，适合写冲突少业务
- default isolation level:Repeatable Read
- External consistency:Snapshot Isolation+Lock (select ... for update)

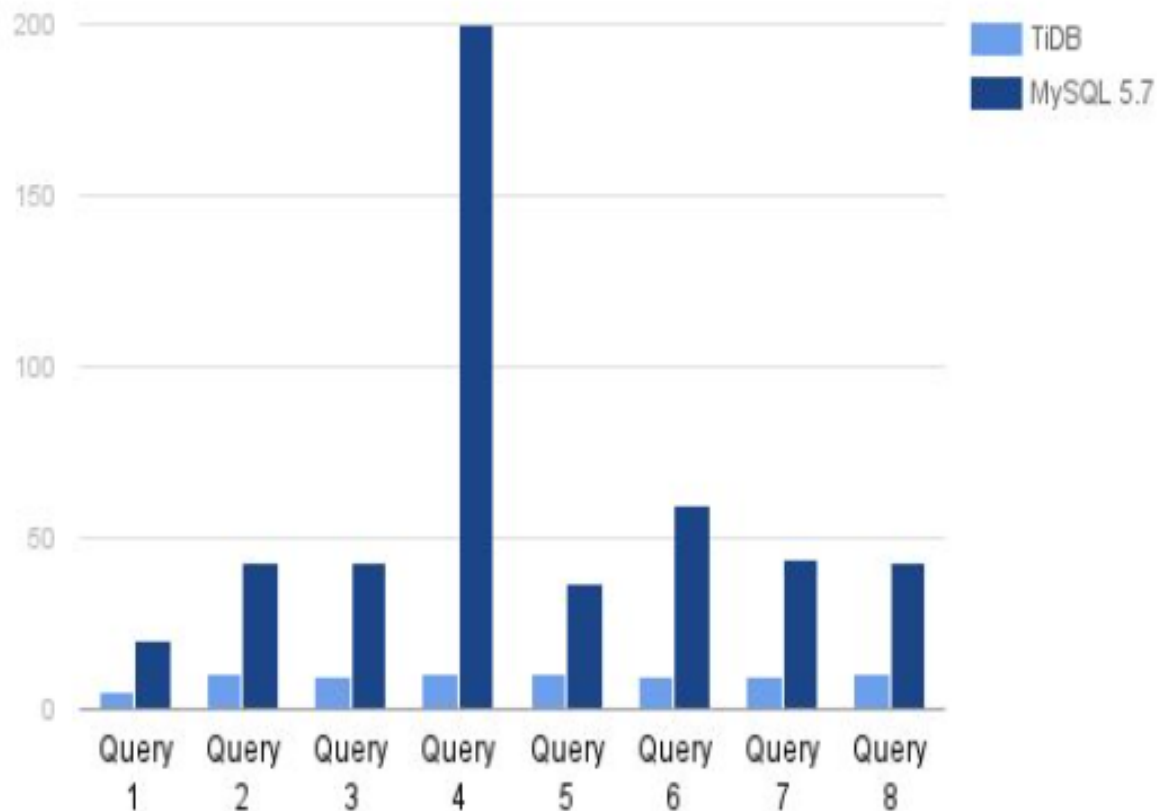
select count(*) from user where name="TiDB"



- 1、计算尽量靠近存储节点，避免RPC
- 2、Filter下推到存储节点，返回有效行
- 3、聚合函数下推存储节点进预聚合
- 4、TiDB Server将count值进行sum

OLAP	MySQL	TiDB
并行扫表/索引	不支持	支持
分布式聚合	不支持	支持
并行Join	不支持	支持
Spark SQL*	不支持	支持

Query elapse (the lower the better)



TiDB Elapse	MySQL Elapse
5.07699437s	19.93s
10.524703077s	43.23s
10.077812714s	43.33s
10.285957629s	>20 mins
10.462306097s	36.81s
9.968078965s	1 min 0.27 sec
9.998030375s	44.05s
10.866549284s	43.18s

- 大数据量，单机存不下
- 离线库
- OLAP
- 访问模式没有明显热点
- 二级索引、事务，强一致，跨机房容灾

- 正在进行功能性能测试RC3版本
- 8月中旬上非核心业务
- 跟进正式1.0的发布（预计8月下旬发布）
- 核心业务
- 私有容器云



CDA 数据分析师
www.cda.cn

THANKS

跨界互联 数聚未来

第四届中国数据分析师行业峰会
CHINA DATA ANALYST SUMMIT