DAMS

中国数据资产管理峰会

CHINA DATA ASSET MANAGEMENT SUMMIT

TiDB: A Hybrid OLTP & OLAP Database

Shen Li @ PingCAP

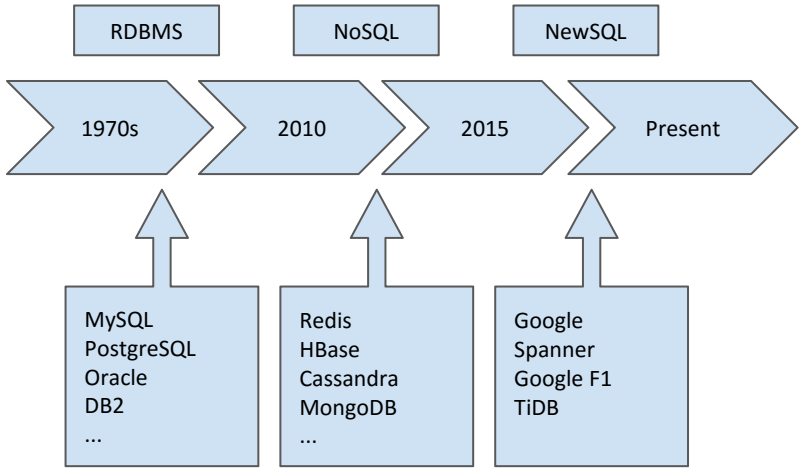
About me

- Shen Li (申砾)
- Tech Lead of TiDB, VP of Engineering
- Netease / 360 / PingCAP
- Infrastructure software engineer

WHY DO WE NEED A NEW DATABASE?

Brief History

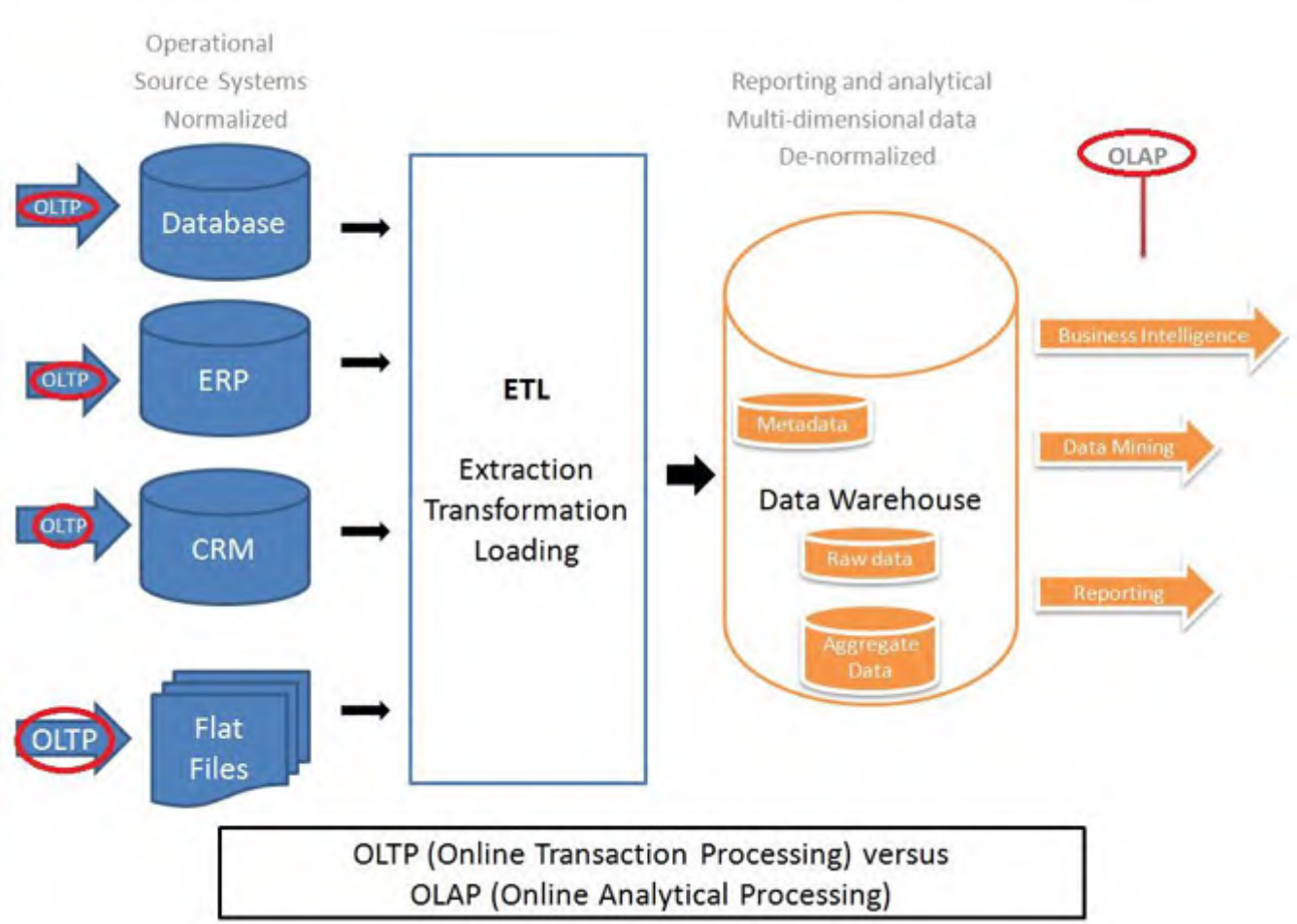
- Standalone RDBMS
- NoSQL
- Middleware & Proxy
- NewSQL



NewSQL database

- Horizontal Scalability
- ACID Transaction
- High Availability
- Auto-Failover
- SQL

OLTP & OLAP



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

Why use two separate systems

- Huge data size
- Complex query logic
- Latency VS Throughput
- Point query VS Full range scan
- Transaction & Isolation level

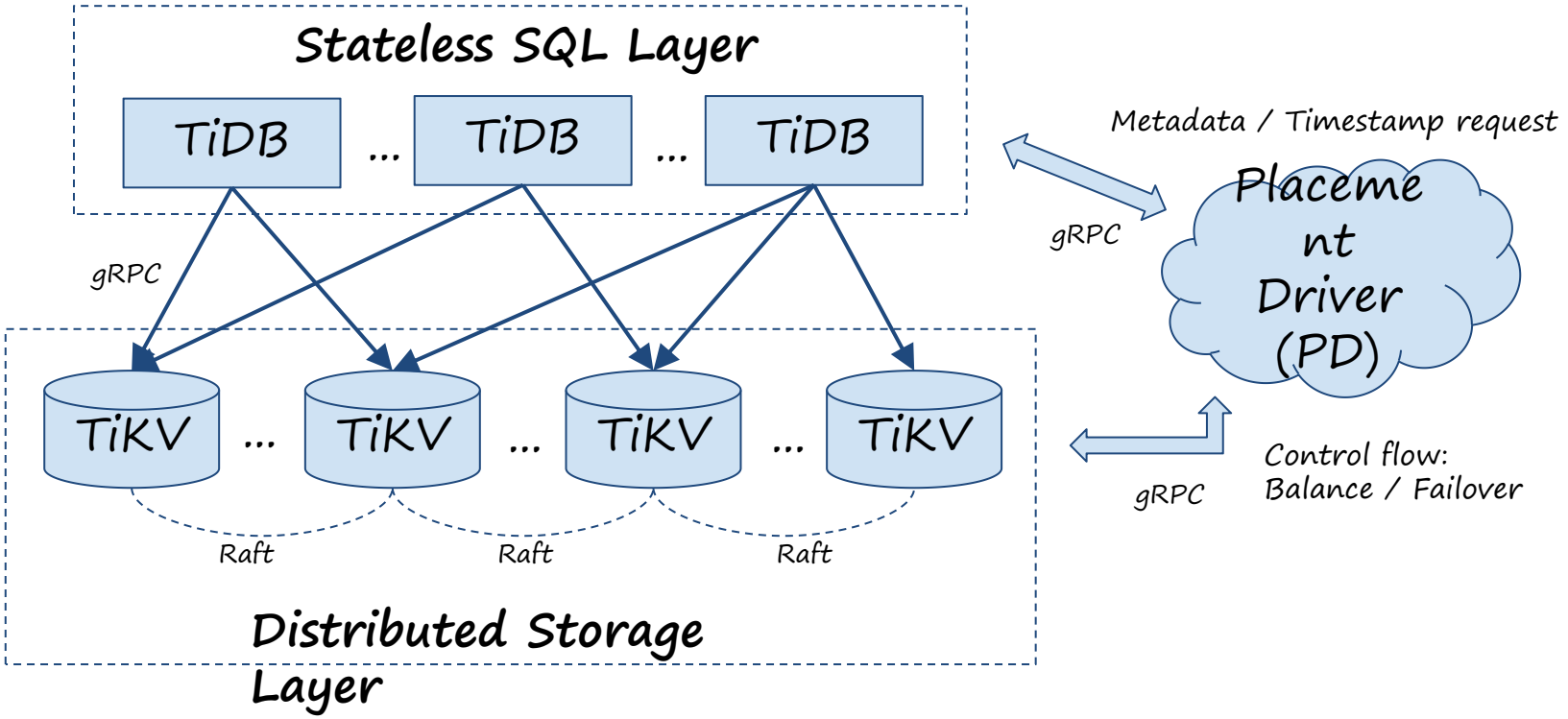
HOW DO WE BUILD A NEWSQL DATABASE?

What is TiDB

- Scalability as the first class feature
- SQL is necessary
- Compatible with MySQL, in most cases
- OLTP + OLAP = HTAP (Hybrid Transactional/Analytical Processing)
- 24/7 availability, even in case of datacenter outages
- Open source, of course

TiDB
A Distributed SQL Database

Architecture

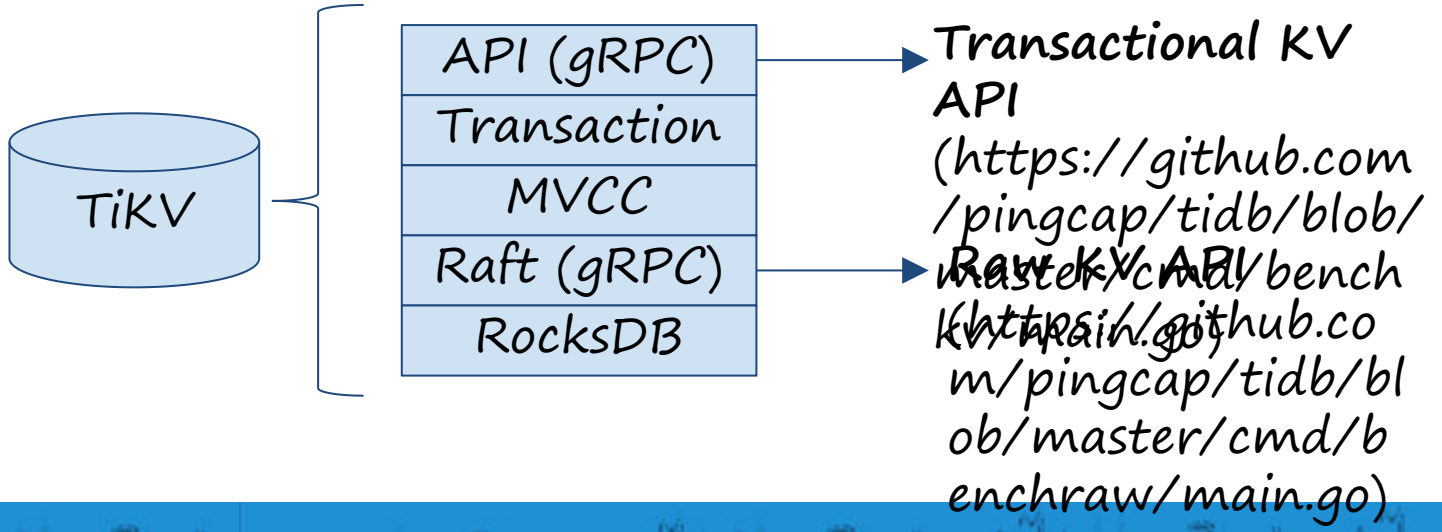


Data distribution

- Hash Based Partition
  - Redis
  - Scale well
  - Bad for scan
- Range Based Partition
  - HBase
  - Good for SQL workload
  - Range size should be small enough and large enough

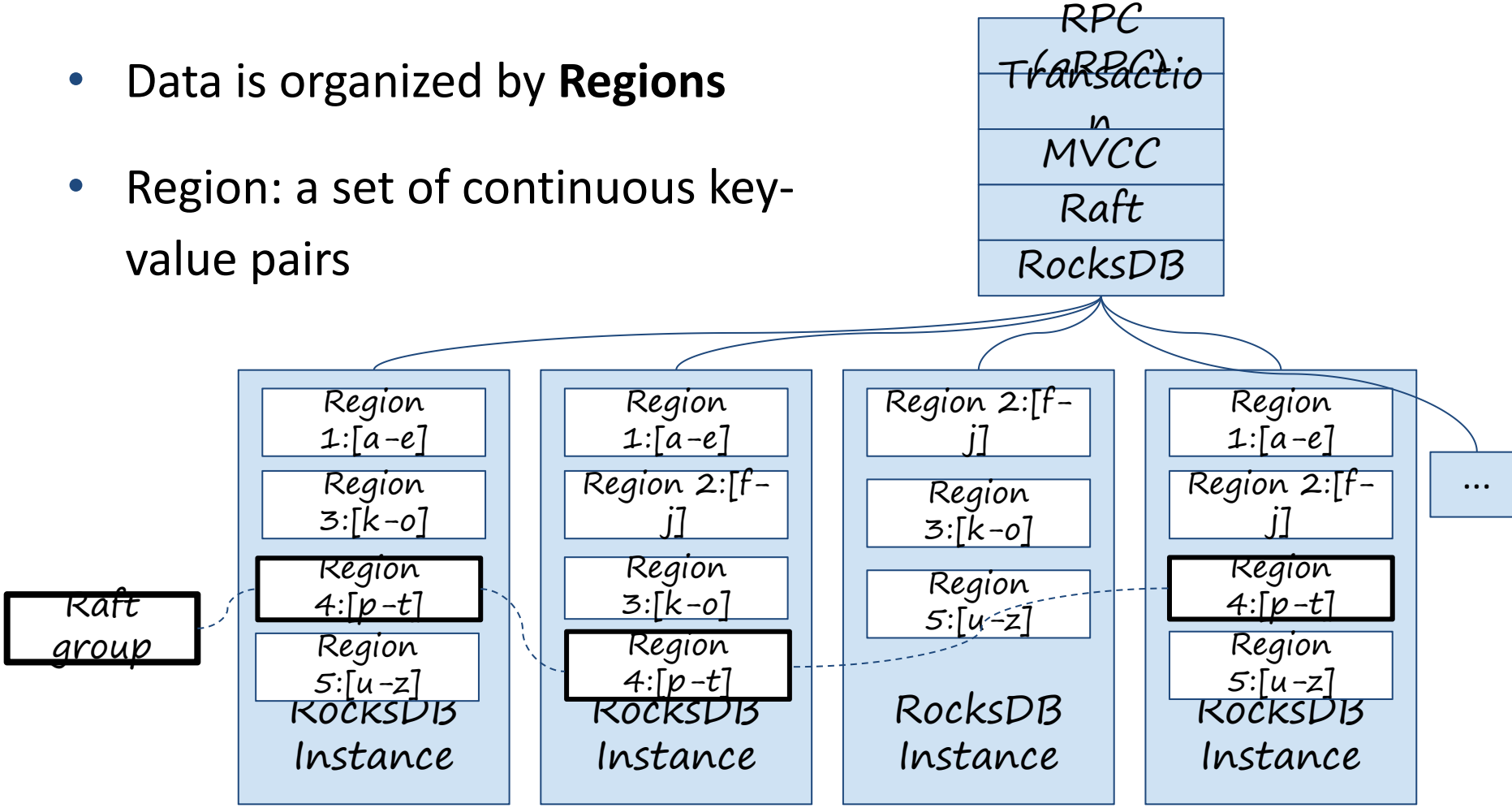
Storage stack 1/2

- TiKV is the underlying storage layer
- Physically, data is stored in RocksDB
- We build a Raft layer on top of RocksDB
  - What is Raft?
- Written in Rust!



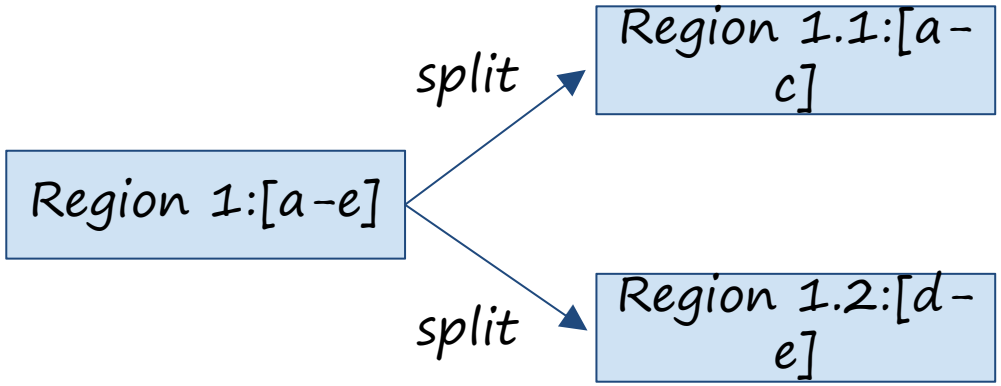
Storage stack 2/2

- Data is organized by **Regions**
- Region: a set of continuous key-value pairs

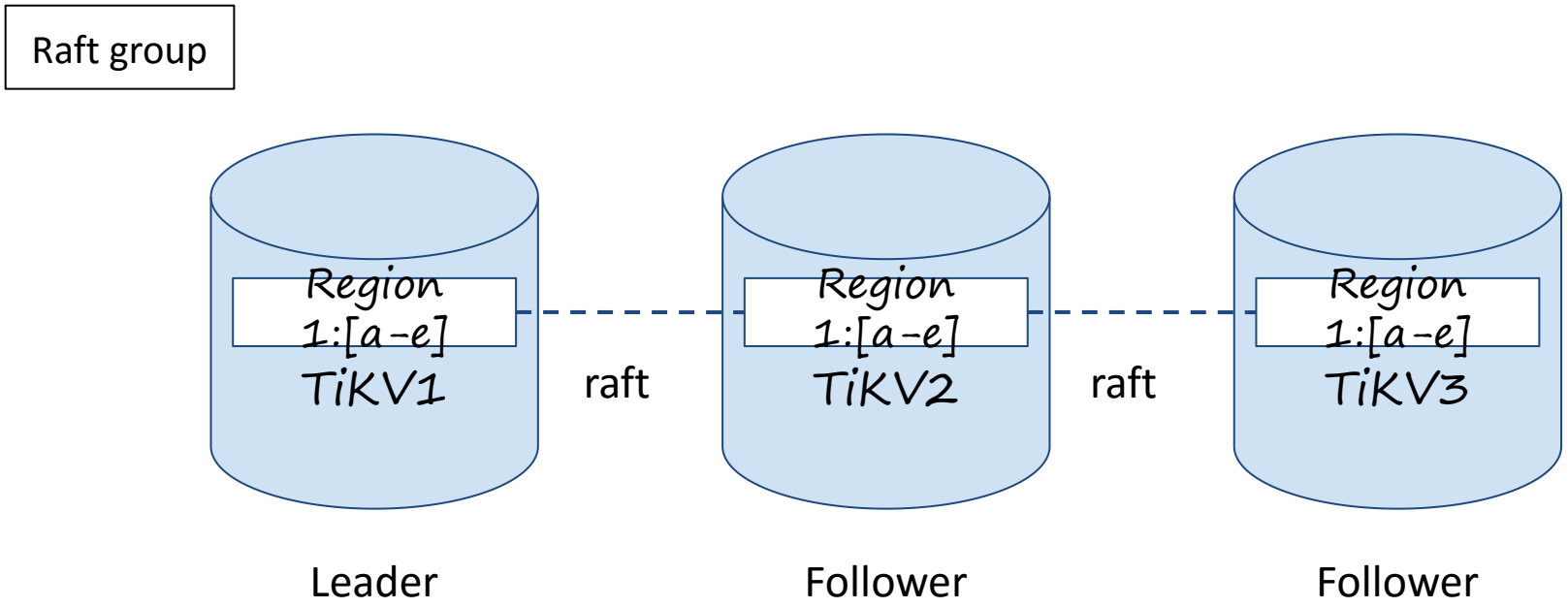


Dynamic Multi-Raft

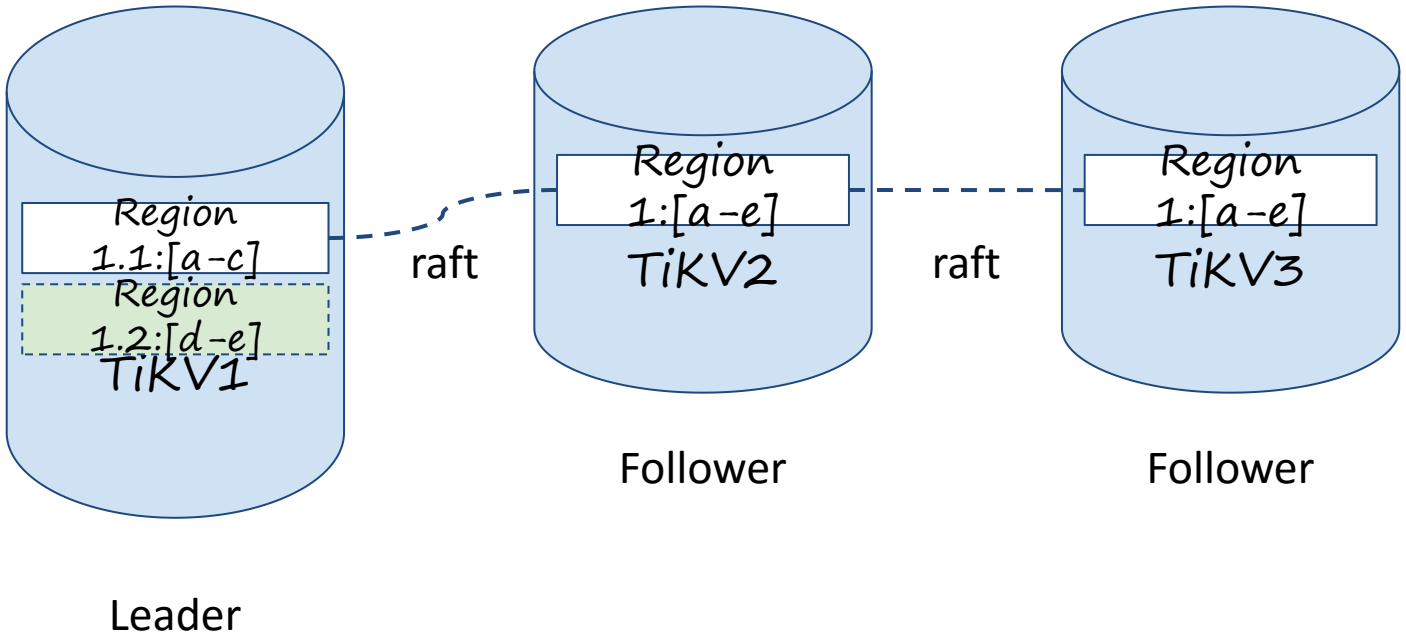
- What's Dynamic Multi-Raft?
  - Dynamic split / merge
- Safe split / merge



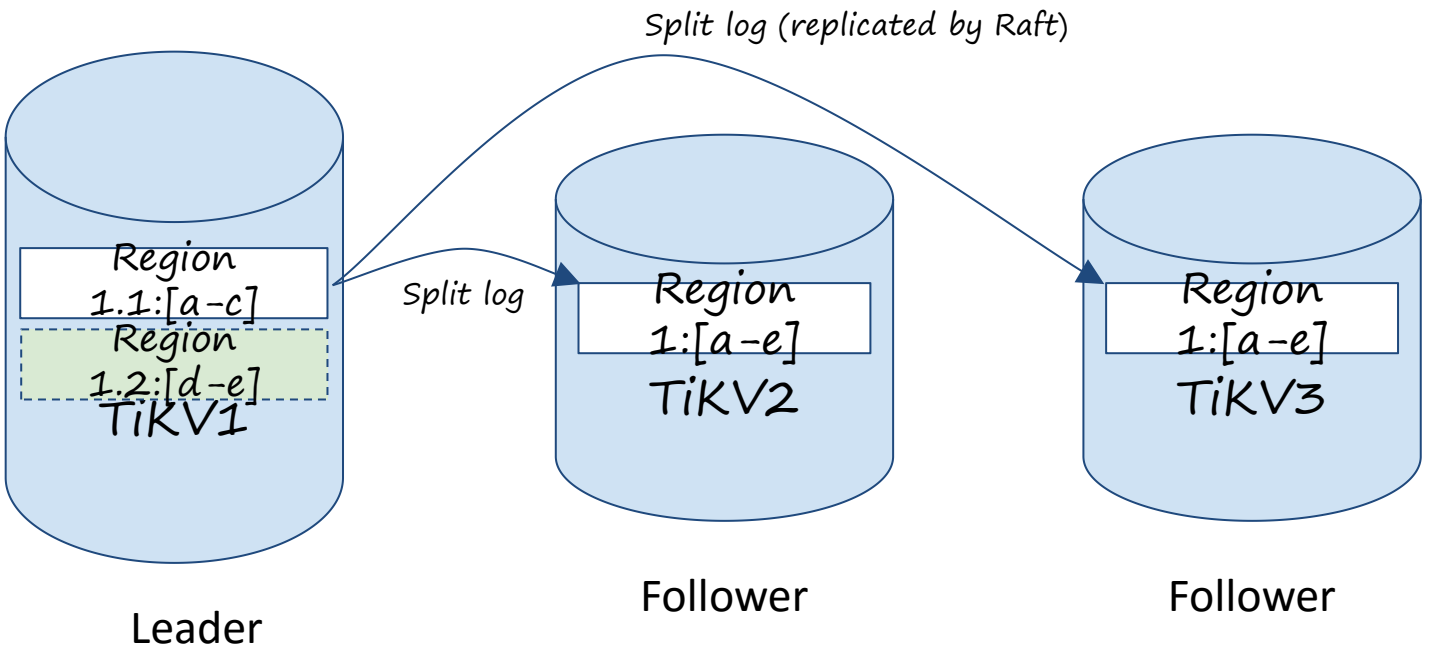
Safe Split: 1/4



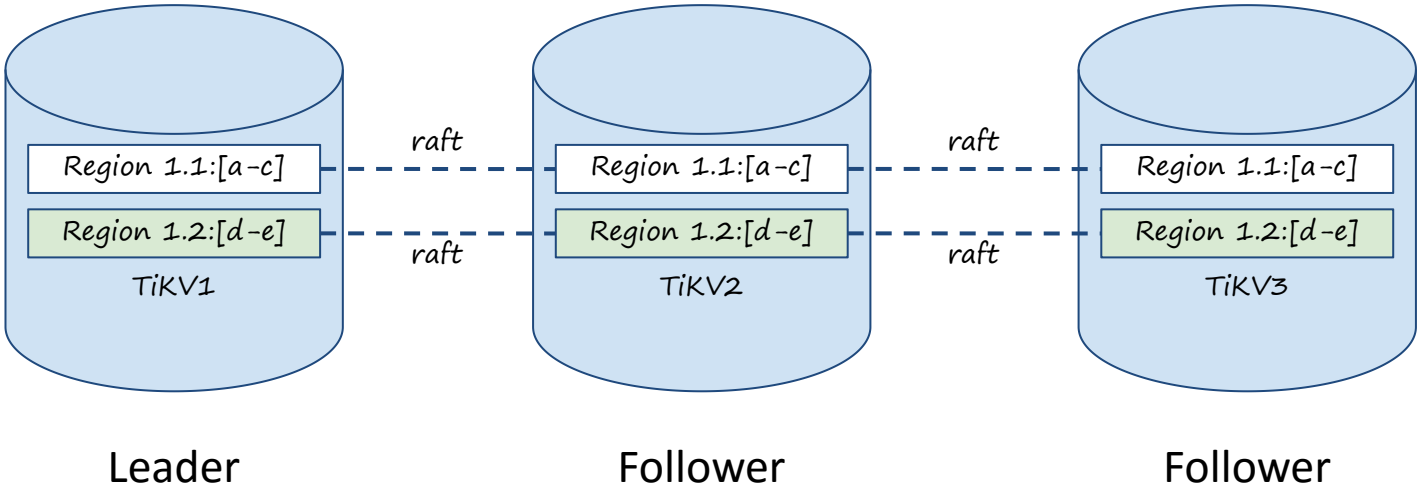
Safe Split: 2/4



Safe Split: 3/4

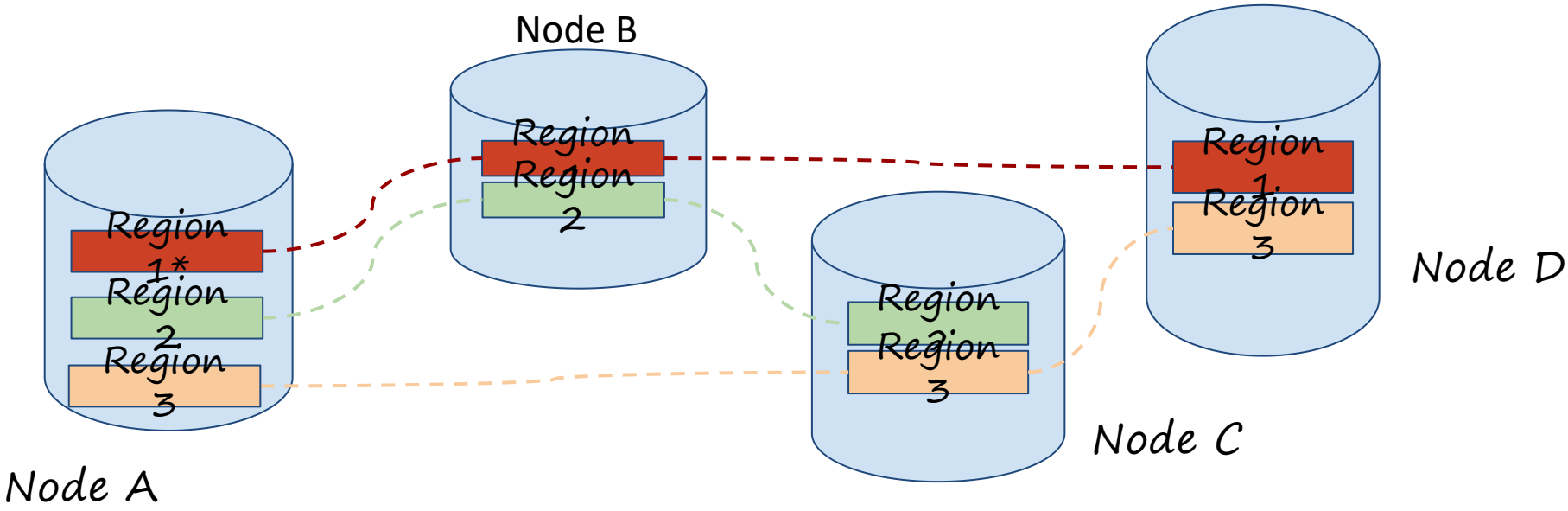


Safe Split: 4/4



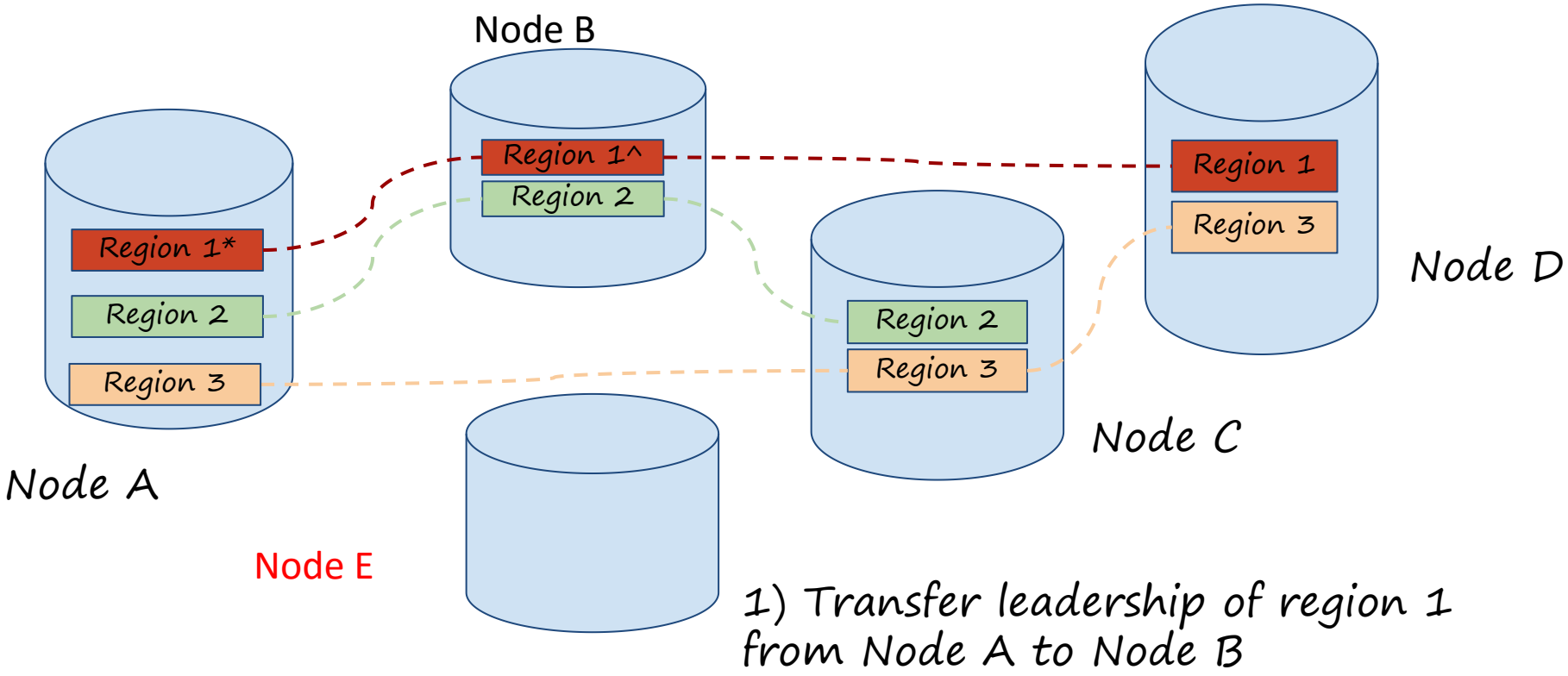
Scale-out (initial state)

- Node A is running out of space

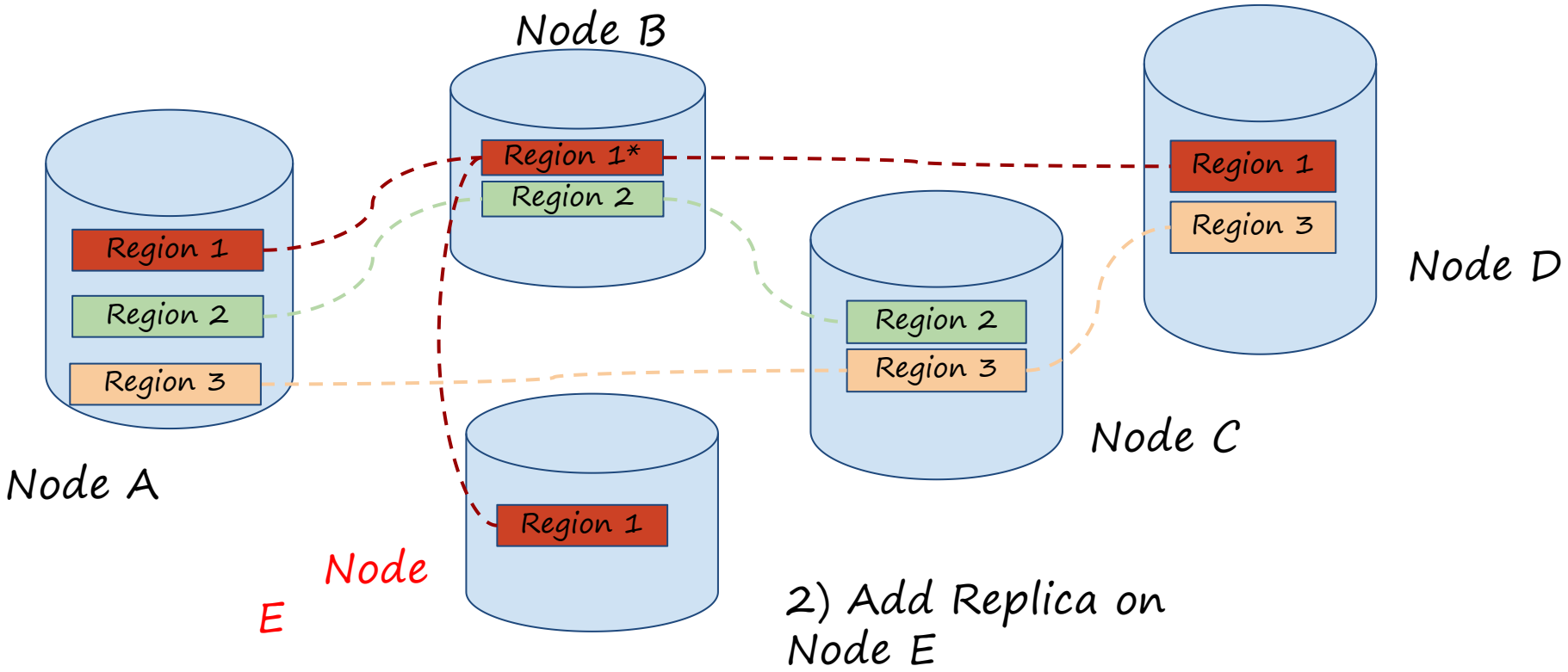


Scale-out (add new node)

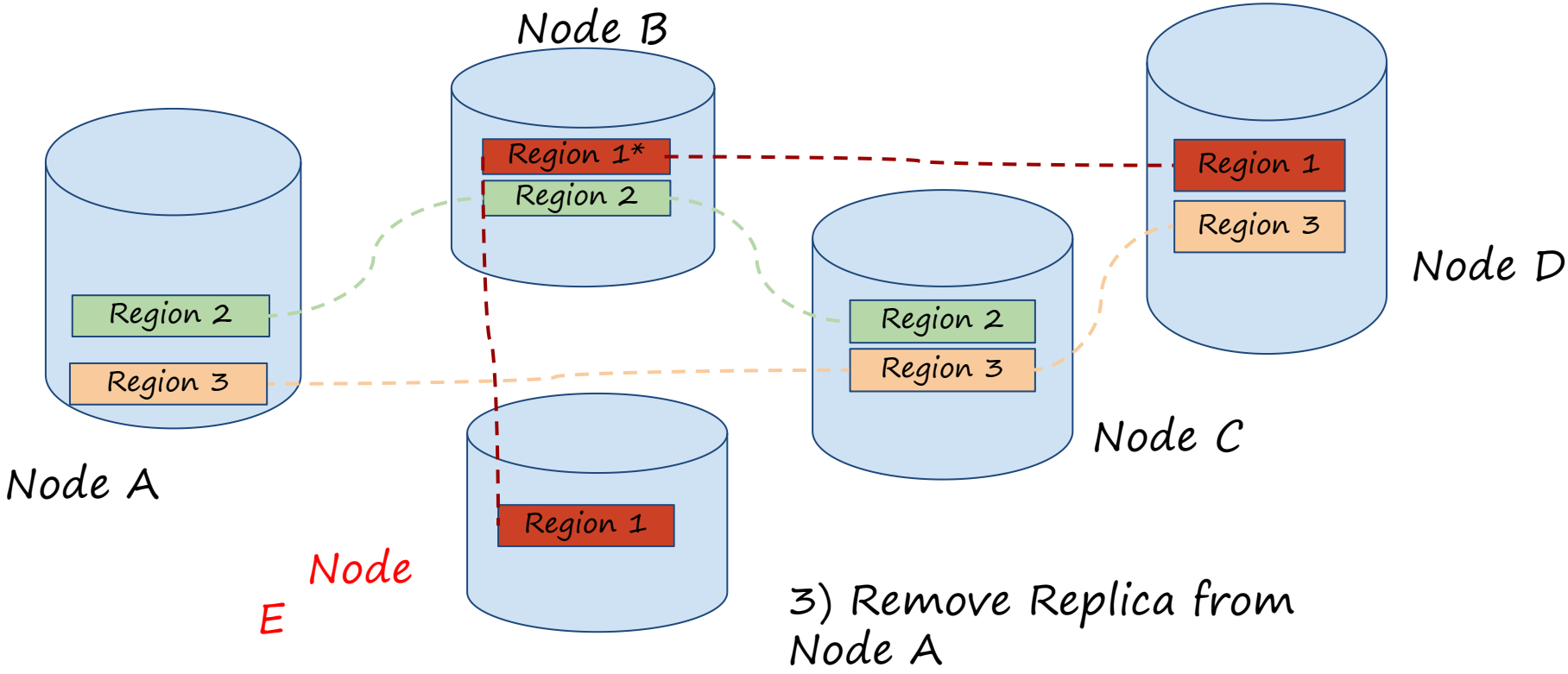
- Add a new node E



Scale-out (balance)



Scale-out (balance)



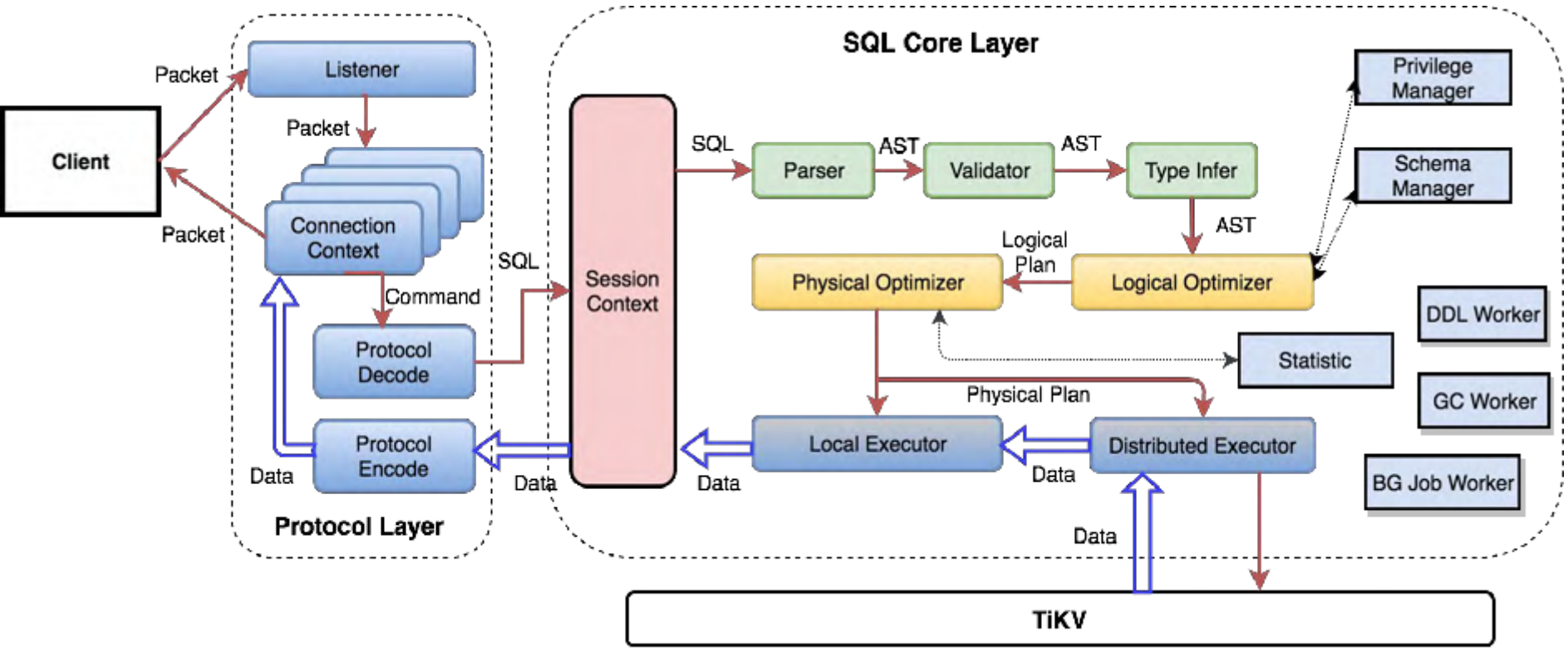
ACID Transaction

- Based on Google Percolator
- ‘Almost’ decentralized 2-phase commit
  - Timestamp Allocator
- Optimistic transaction model
- Default isolation level: Repeatable Read
- External consistency: Snapshot Isolation + Lock
  - SELECT … FOR UPDATE

Distributed SQL

- Full-featured SQL layer
- Predicate pushdown
- Cost-based optimizer
- Parallel Operators
- Multiple Join Operators
- Hash/Streaming Operators

TiDB SQL Layer overview

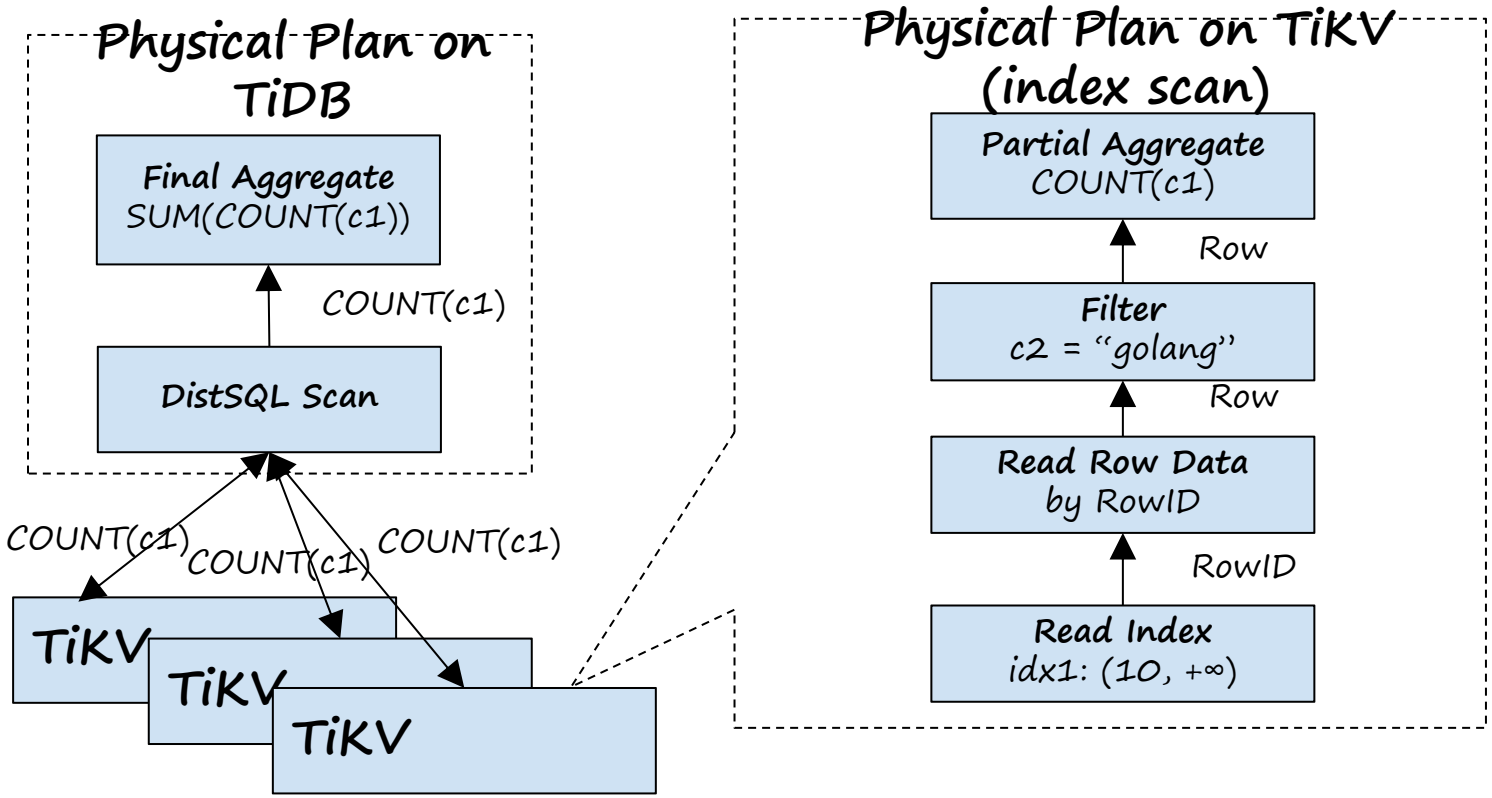


What happens behind a query

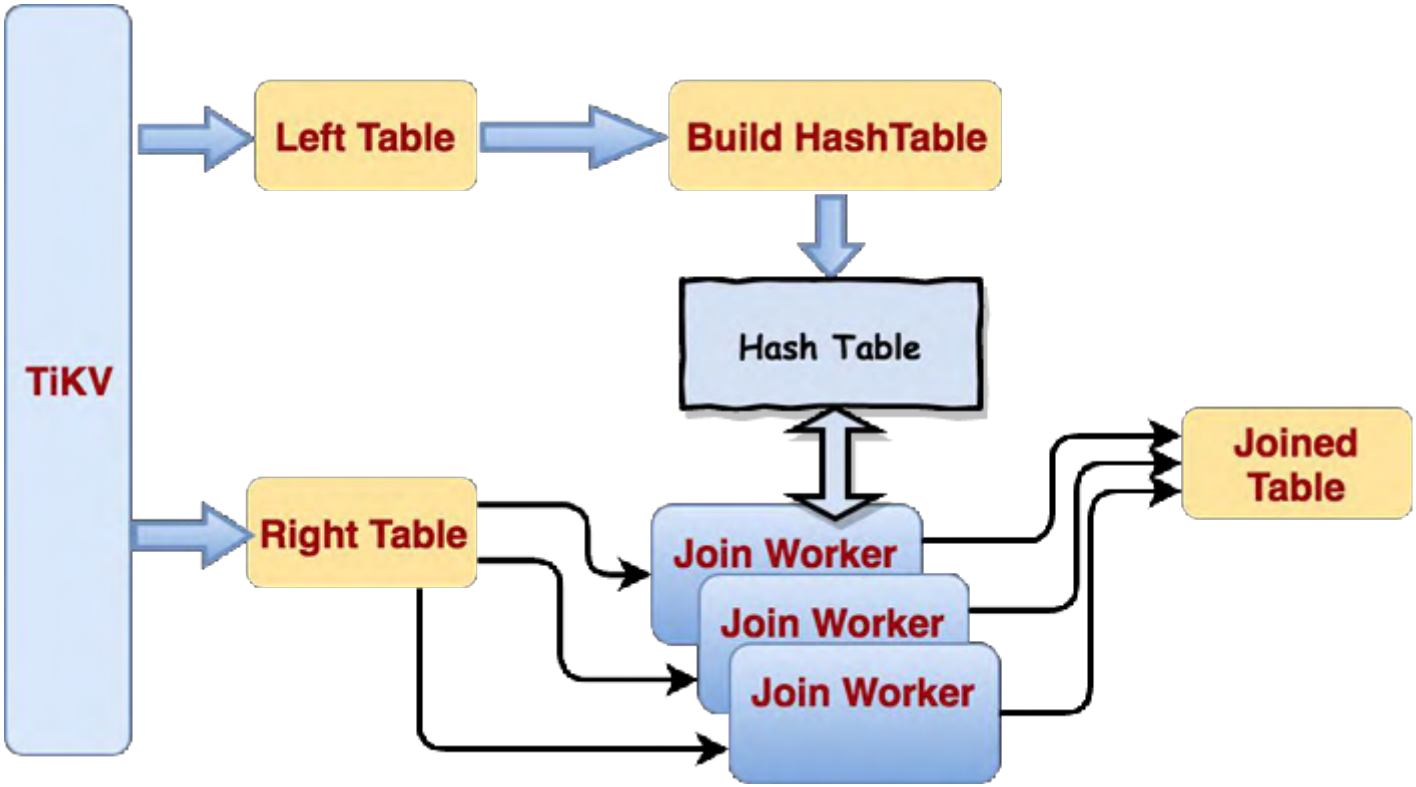
```
CREATE TABLE t (c1 INT, c2 TEXT,
KEY idx_c1(c1));

SELECT COUNT(c1) FROM t WHERE
c1 > 10 AND c2 = 'golang';
```

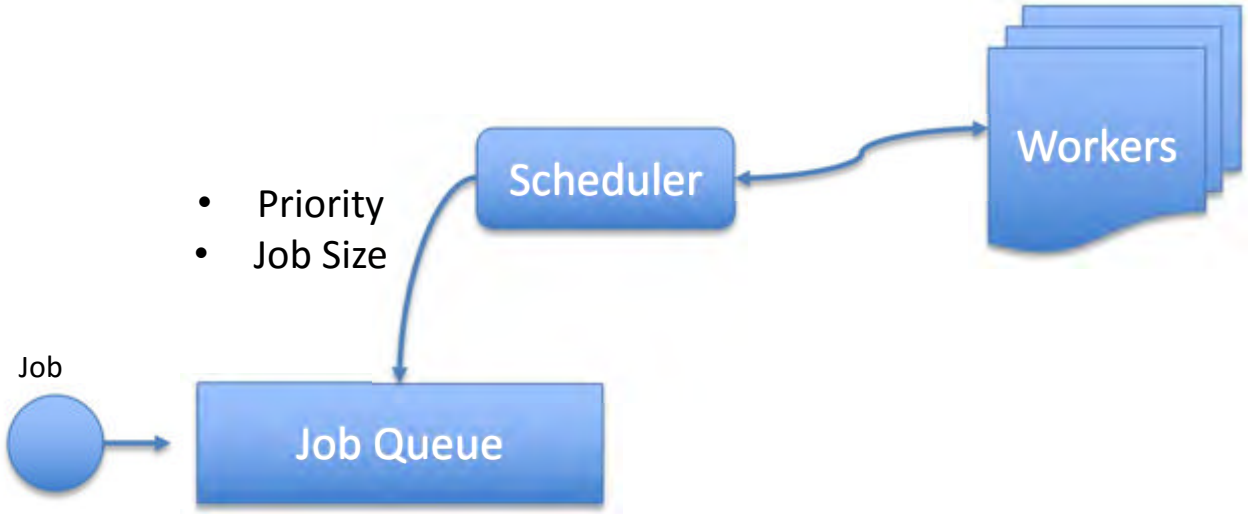
Query Plan


Physical Plan on TiDB
Final Aggregate
SUM(COUNT(c1))
COUNT(c1)
DistSQL Scan
COUNT(c1)
COUNT(c1)
COUNT(c1)
TiKV
TiKV
TiKV
Physical Plan on TiKV (index scan)
Partial Aggregate
COUNT(c1)
Row
Filter
c2 = "golang"
Row
Read Row Data
by RowID
RowID
Read Index
idx1: (10, +∞)

Distributed Hash Join



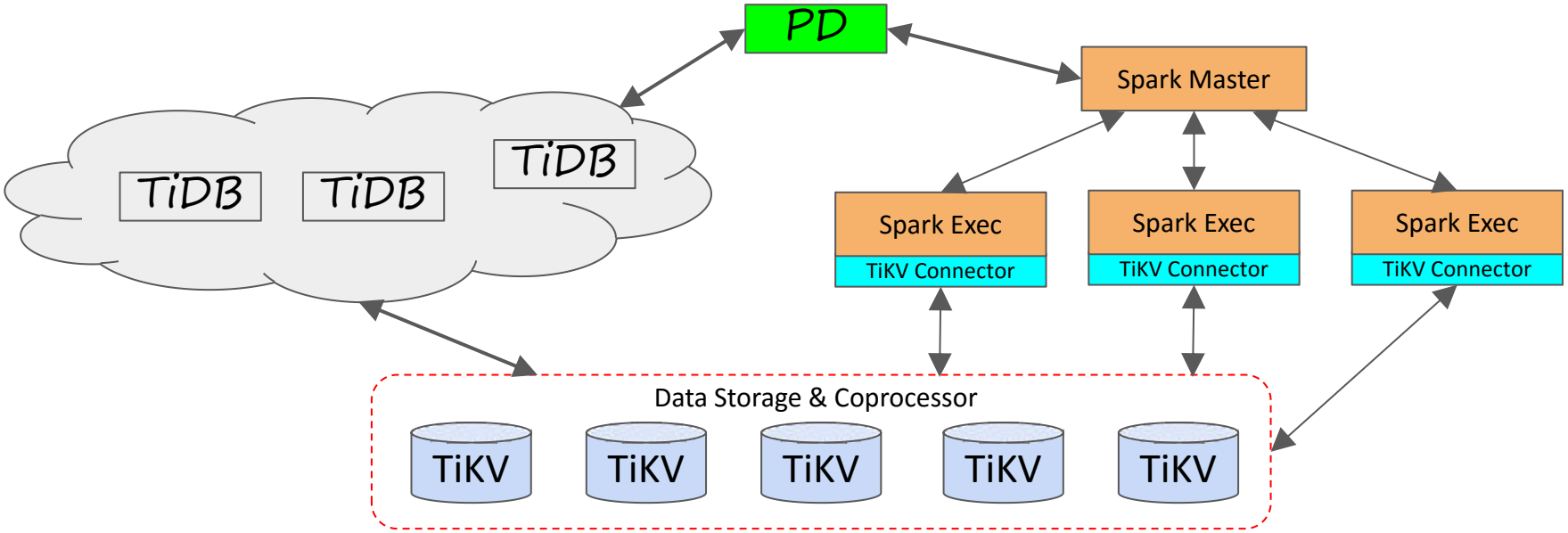
Job Scheduling



SQL IS NOT ENOUGH

TiSpark 1/3

- TiDB + SparkSQL = TiSpark



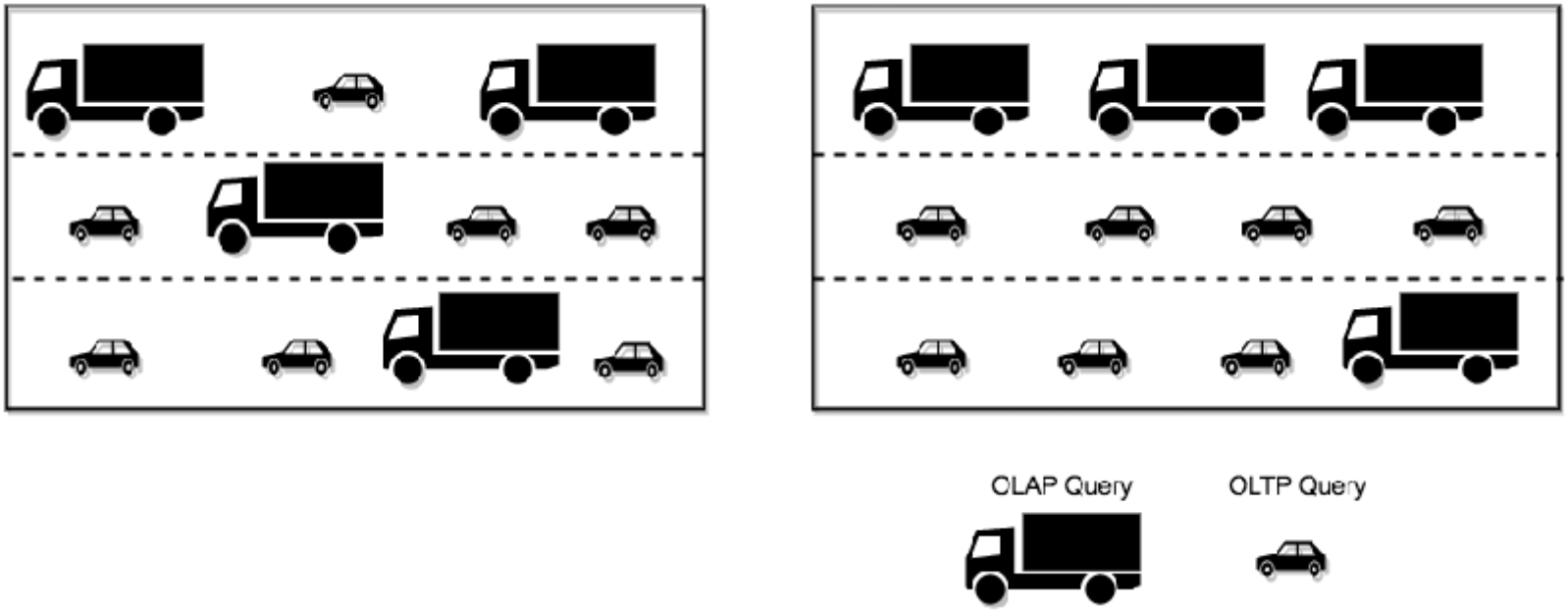
TiSpark 2/3

- TiKV Connector is better than JDBC connector
- Index support
- Complex Calculation Pushdown
- CBO
  - Pick up right Access Path
  - Join Reorder
- Priority & Isolation Level

TiSpark 3/3

- Analytical / Transactional support all on one platform
  - No need for ETL
  - Real-time query with Spark
  - Possibility for get rid of Hadoop
- Embrace Spark echo-system
  - Support of complex transformation and analytics with Scala / Python and R
  - Machine Learning Libraries
  - Spark Streaming

Hybrid Transactional/Analytical Processing



DAMS

中国数据资产管理峰会

CHINA DATA ASSET MANAGEMENT SUMMIT

THANK YOU !