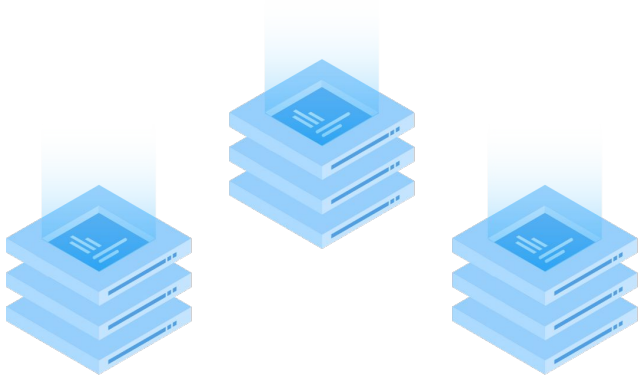
Making HTAP Real with TiFlash

A TiDB Native Columnar Extension

PingCAP TiDB

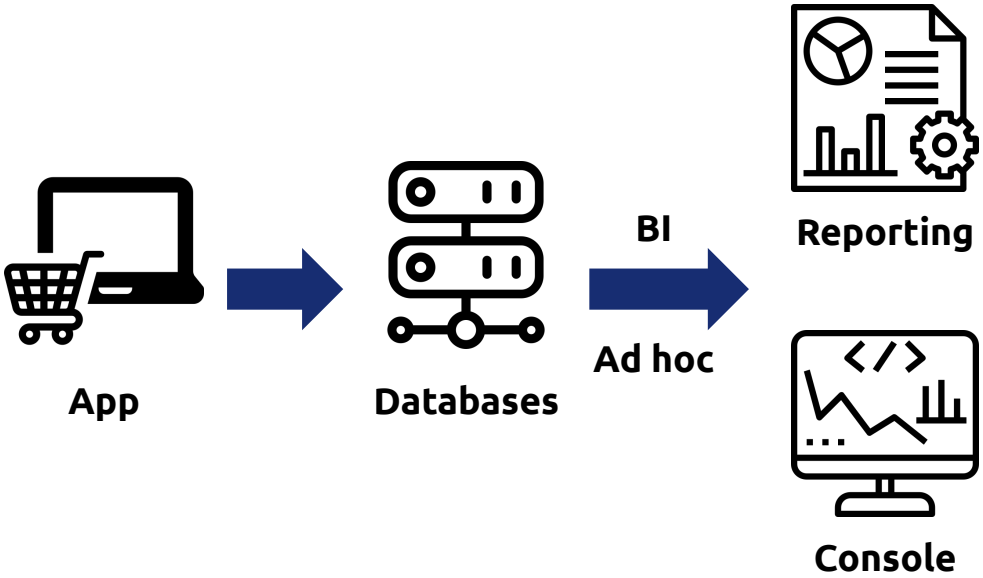


TiDB Community Slack Channel
https://pingcap.com/tidbslack/

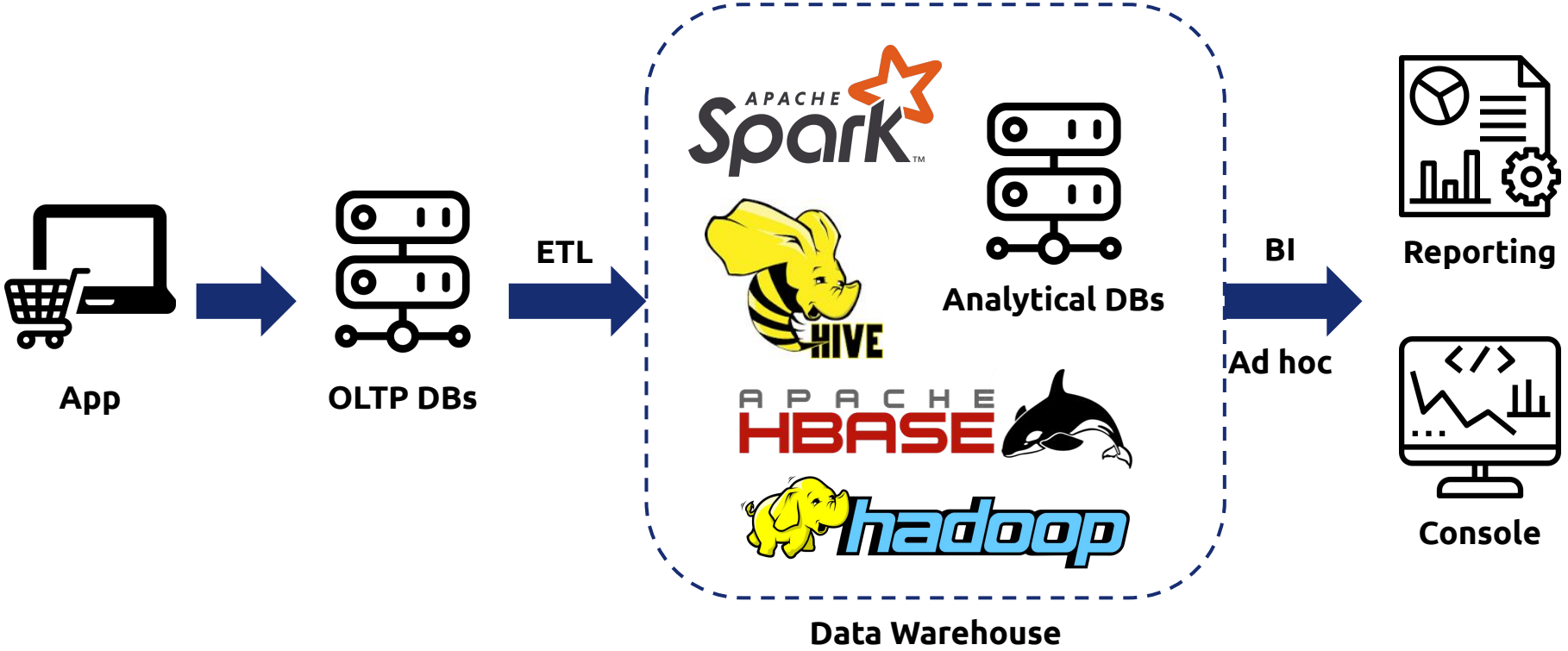
About Me

- Sun Ruoxi 孙若曦
- Database Engineer, Analytical Product Team @ PingCAP
- Was
  - Tech lead, SQL on Hadoop Team @ Transwarp
  - Tech lead, Arch Infra Team @ NVIDIA
- Focused on Big-data / Database / SQL

Data Platform - What You Think It Is



Data Platform - What It Really Is



Why

VS

Fundamental Conflicts

- Different access patterns
  - OLTP
    - Short / point access to small number of records
    - Row-based format
  - OLAP
    - Large / batch process of subset of columns
    - Column-based format
- Workload interference
  - OLAP queries can easily occupy large amount of system resources
  - OLTP latency / concurrency will be dramatically down

A Popular Solution

- Use different types of databases
  - OLTP specialized database for transactional data
  - Hadoop / analytical database for historical data

- Offload transactional data via ETL process into Hadoop / analytical database
  - Periodically, usually per day

Good enough, really?

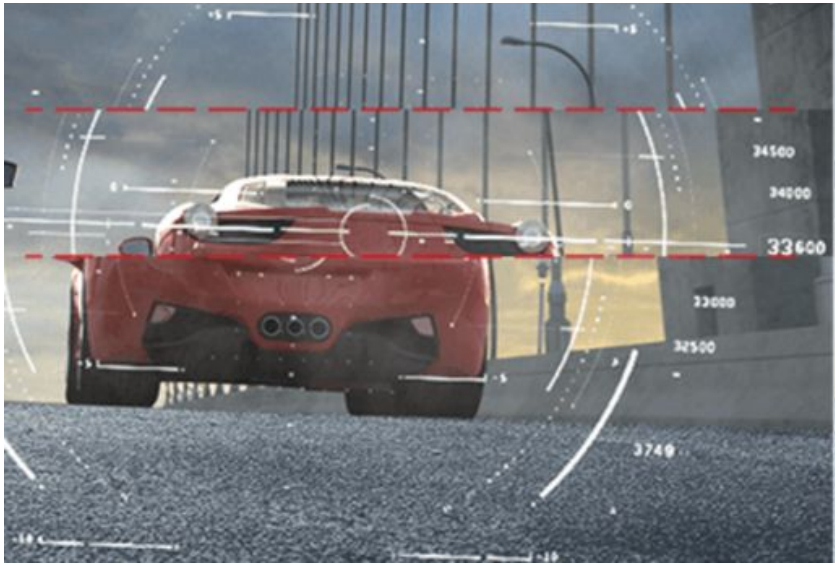
Complexity

OR

Freshness

OR

Consistency



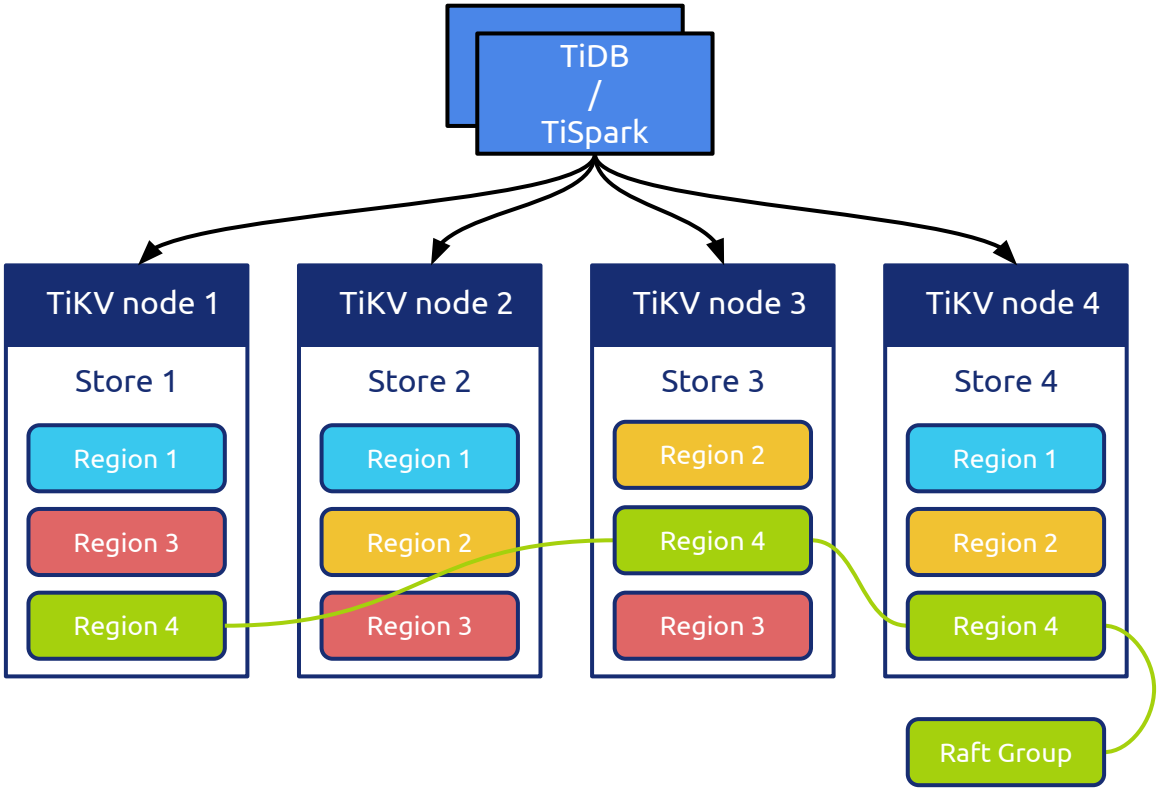
OR

TiFlash

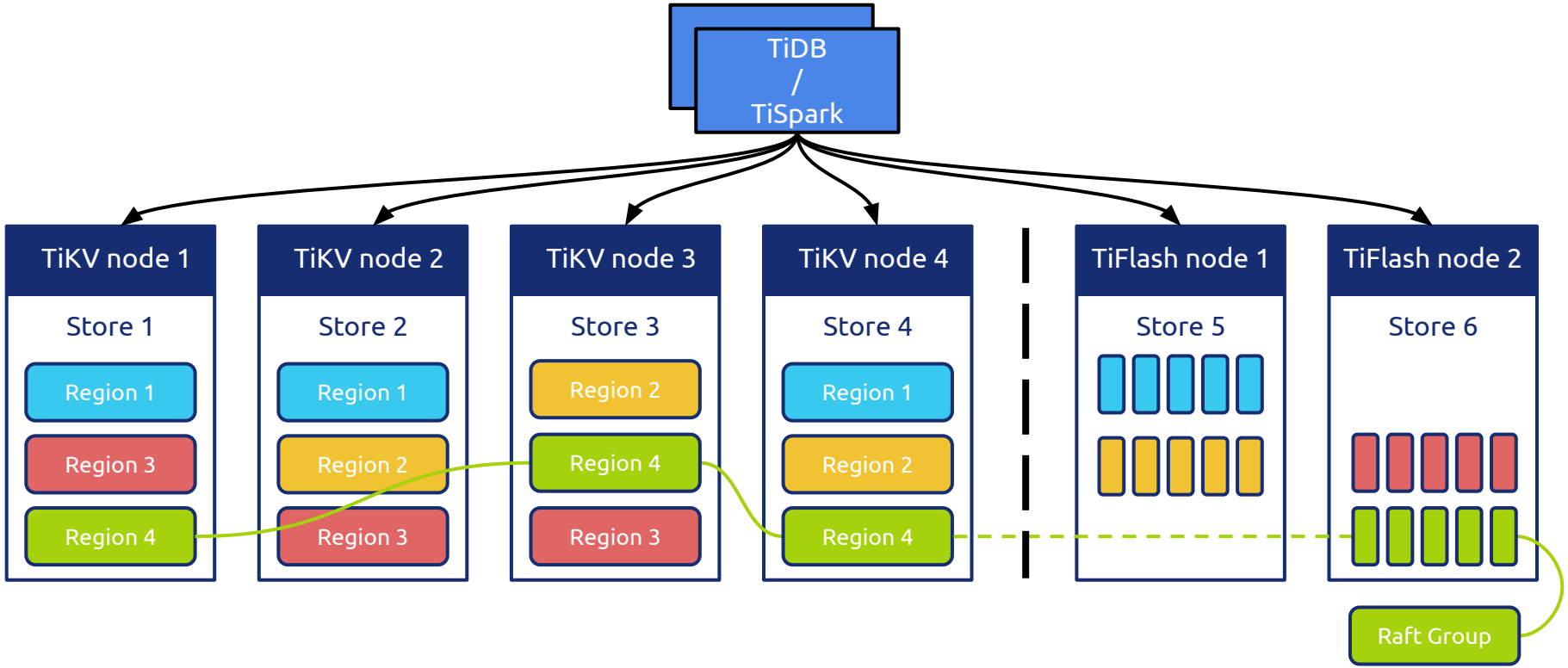
What Is TiFlash?

- An extended analytical engine for TiDB
  - Columnar storage and vectorized processing
  - Partially based on ClickHouse with tons of modifications
  - Enterprise offering
- Data sync via extended Raft consensus algorithm
  - Strong consistency
  - Trivial overhead
- Strict workload isolation to eliminate the impact on OLTP
- Tight integration with TiDB

TiDB Architecture



TiDB with TiFlash Architecture



Columnstore VS Rowstore

- Columnstore
  - Suitable for analytical workload
  - Efficient CPU utilization using vectorized processing
  - High compression rate
  - Bad small random read / write
- Rowstore
  - Researched and optimized for OLTP scenario for decades
  - Cumbersome for analytical workload

Columnstore VS Rowstore

Rowstore

id	name	age
0962	Jane	30
7658	John	45
3589	Jim	20
5523	Susan	52

SELECT AVG(age) FROM emp;

Columnstore

id
0962
7658
3589
5523

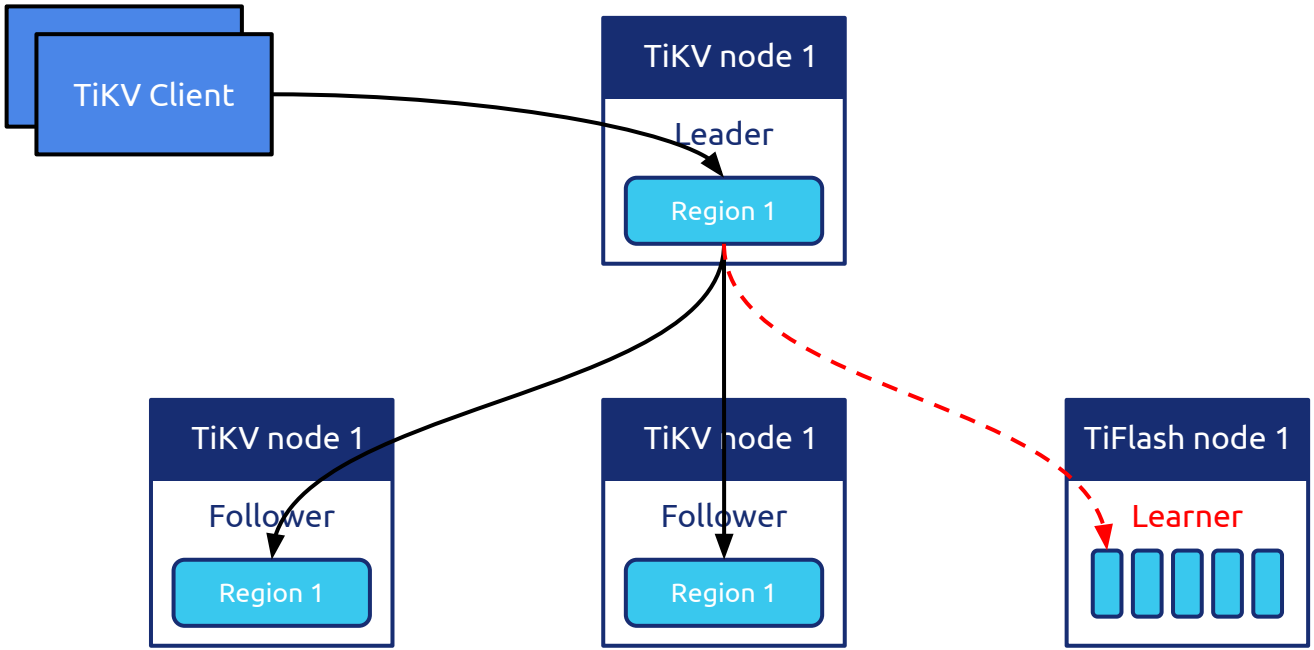
name
Jane
John
Jim
Susan

age
30
45
20
52

Low-cost Data Replication

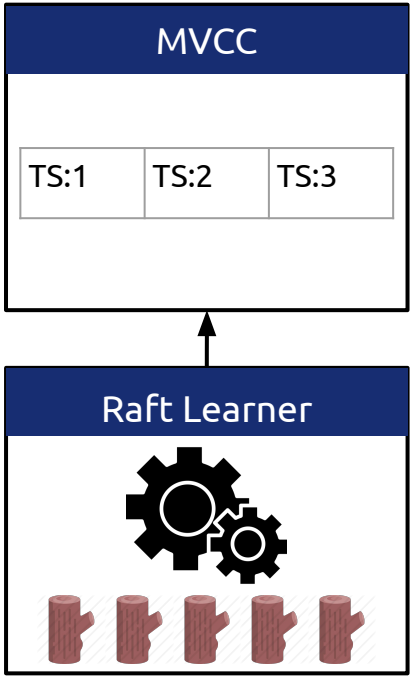
- Data is replicated to TiFlash via Raft Learner
  - Extended Raft consensus algorithm
  - Out of leader election
  - Async replication
  - Almost zero overhead to OLTP workload

Low-cost Data Replication

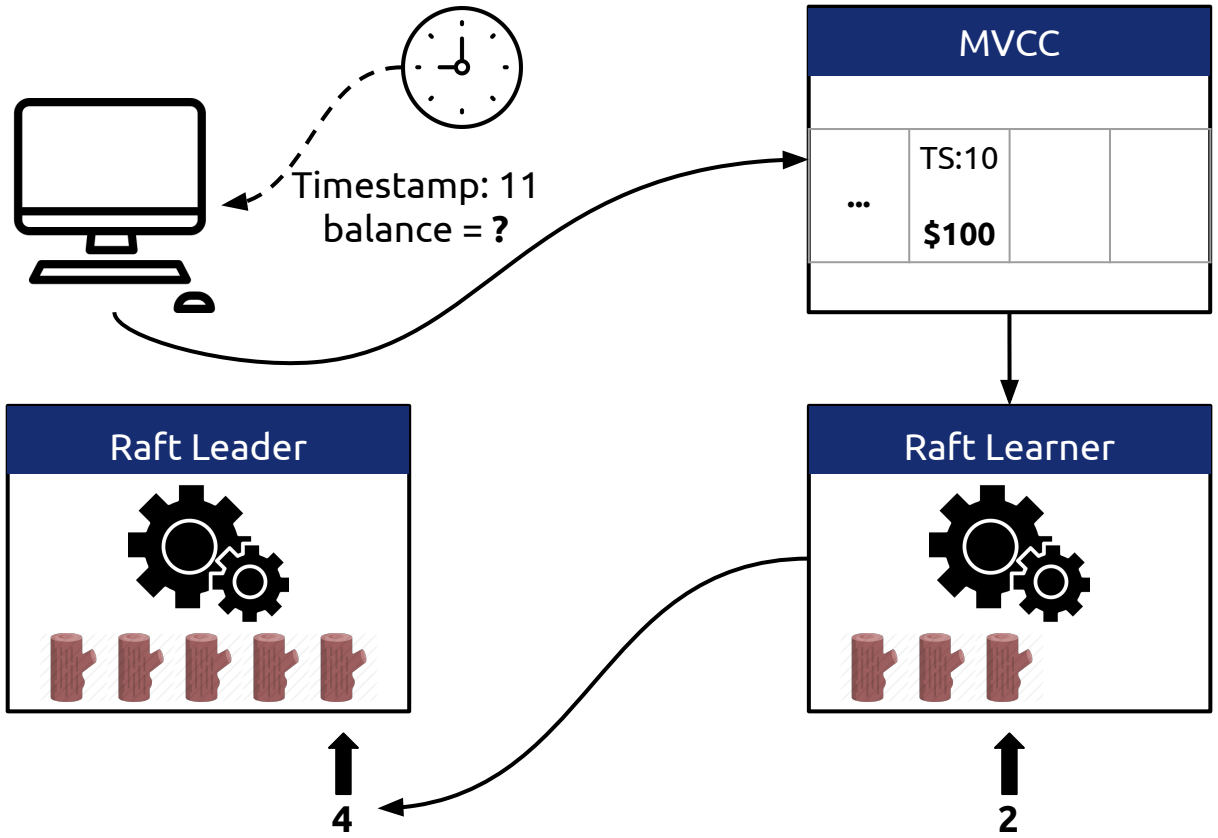


Strong Consistency

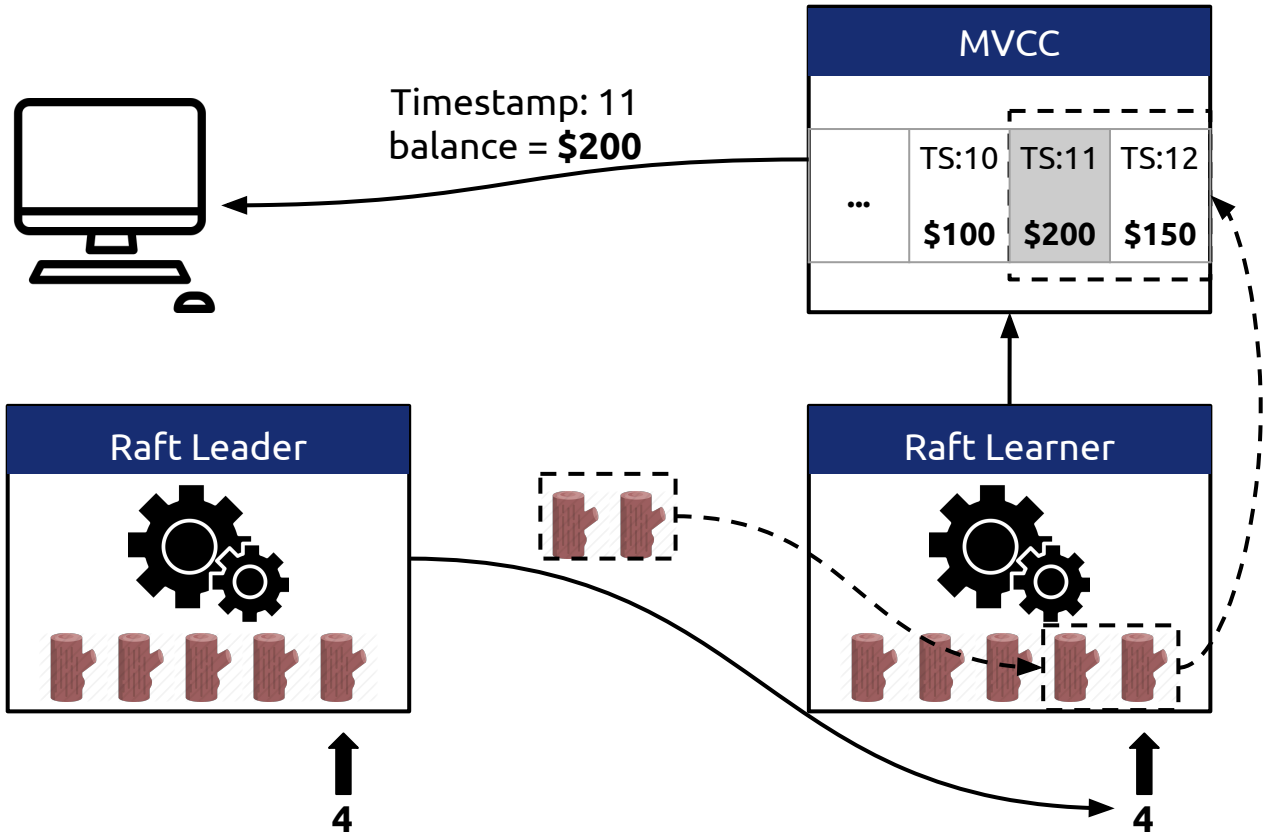
- Logically the same view as in rowstore
  - Same data
  - Same isolation level (SI)
- TiFlash keeps casual consistency via async replication
  - 99.99…% in-sync
  - 0.00…1% out-of-sync
- Read operation guarantees strong consistency
  - Learner Read
  - MVCC



Learner Read + MVCC



Learner Read + MVCC



TiFlash is beyond columnar storage

Scalability

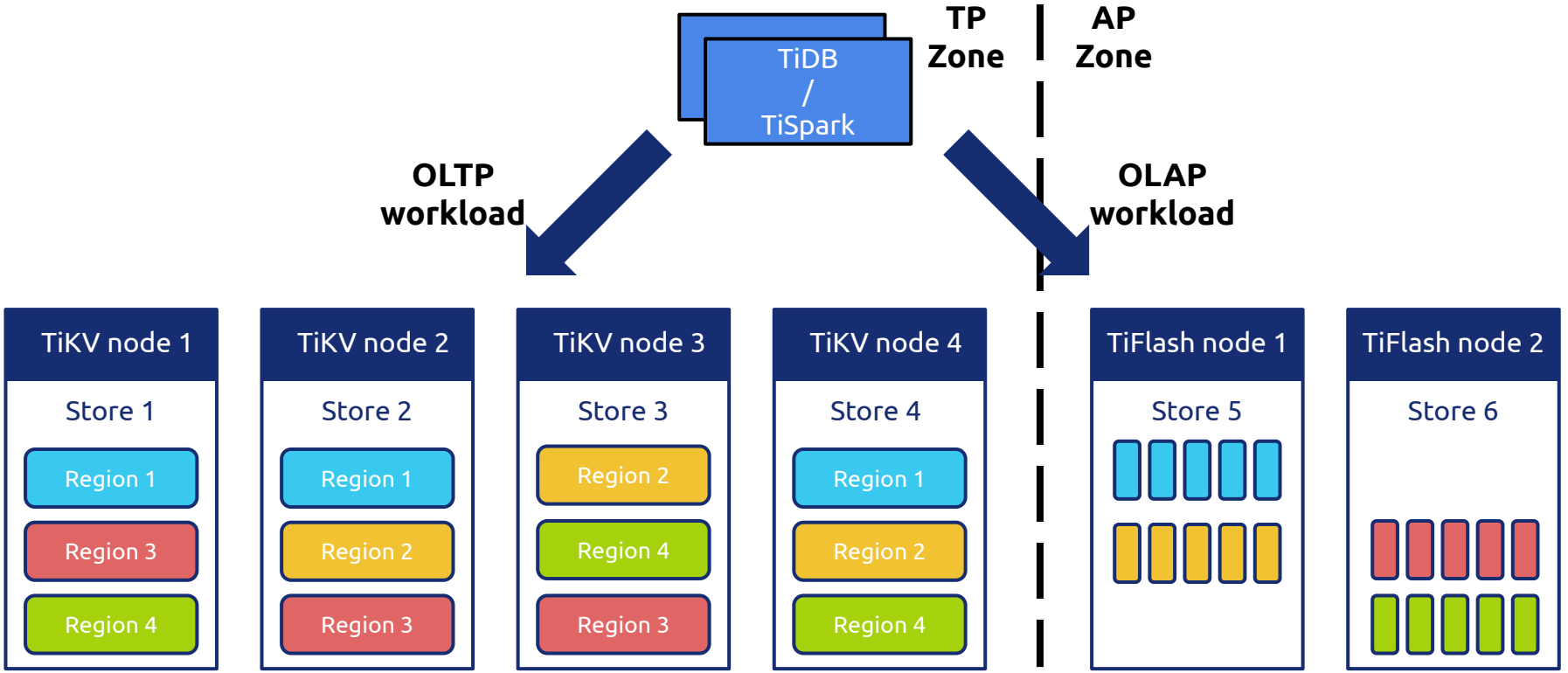
- TiDB relies on Multi-Raft for scalibility
  - One command to add / remove node
  - Scaling is fully automatic
  - Smooth and painless data rebalance

- TiFlash fully inherits these abilities

Isolation

- Perfect resource isolation to prevent workload interference
- Dedicated nodes for TiFlash
- Nodes are clustered into “zone”s
  - TP Zone
    - TiKV nodes, for OLTP workload
  - AP Zone
    - TiFlash nodes, for OLAP workload

Isolation


TiDB
/
TiSpark
TP
Zone
AP
Zone
OLTP
workload
OLAP
workload
TiKV node 1
Store 1
Region 1
Region 3
Region 4
TiKV node 2
Store 2
Region 1
Region 2
Region 3
TiKV node 3
Store 3
Region 2
Region 4
Region 3
TiKV node 4
Store 4
Region 1
Region 2
Region 4
TiFlash node 1
Store 5
TiFlash node 2
Store 6

Integration

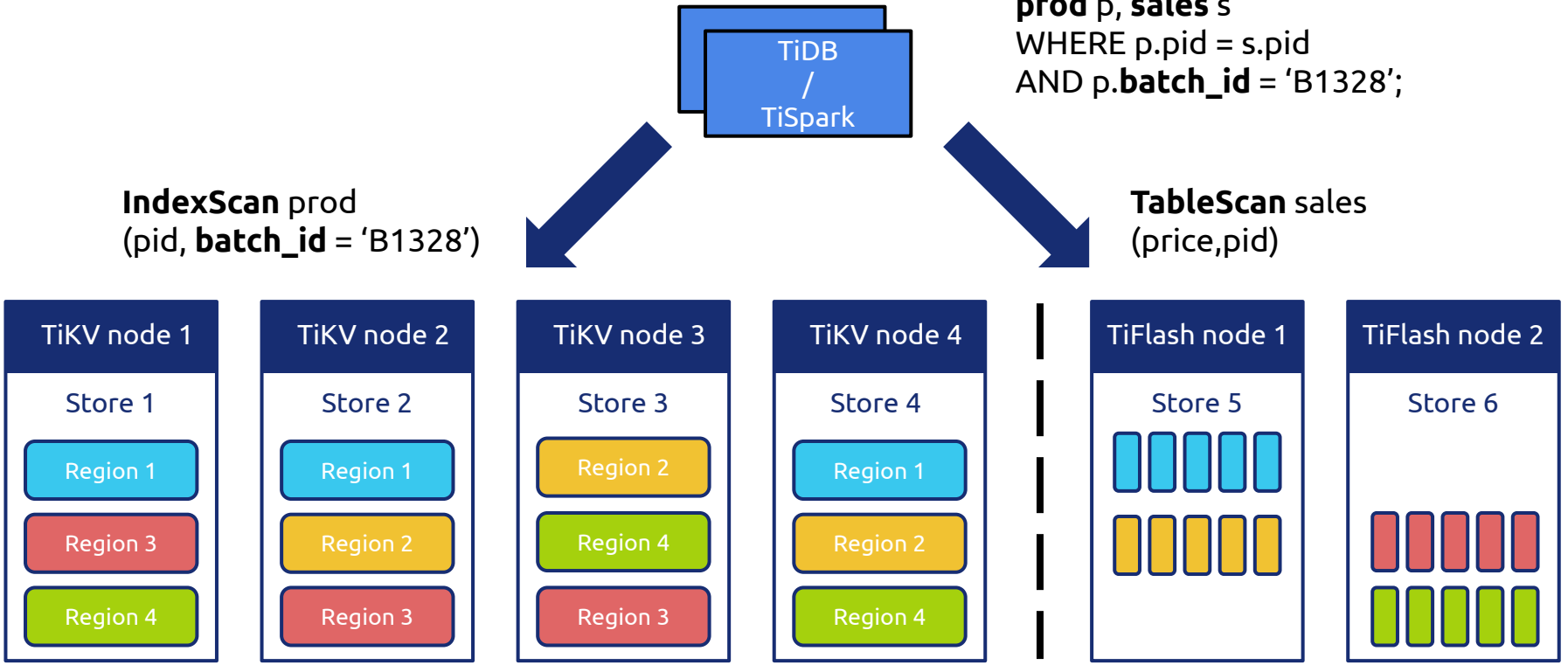
- TiDB / TiSpark might choose to read from either side
  - Based on cost
  - Columnstore is treated as a special kind of index

- Upon TiFlash replica failure, read TiKV replica transparently

- Join data from both sides in a single query

Integration

SELECT AVG(s.price) FROM
prod p, **sales** s
WHERE p.pid = s.pid
AND p.**batch_id** = 'B1328';

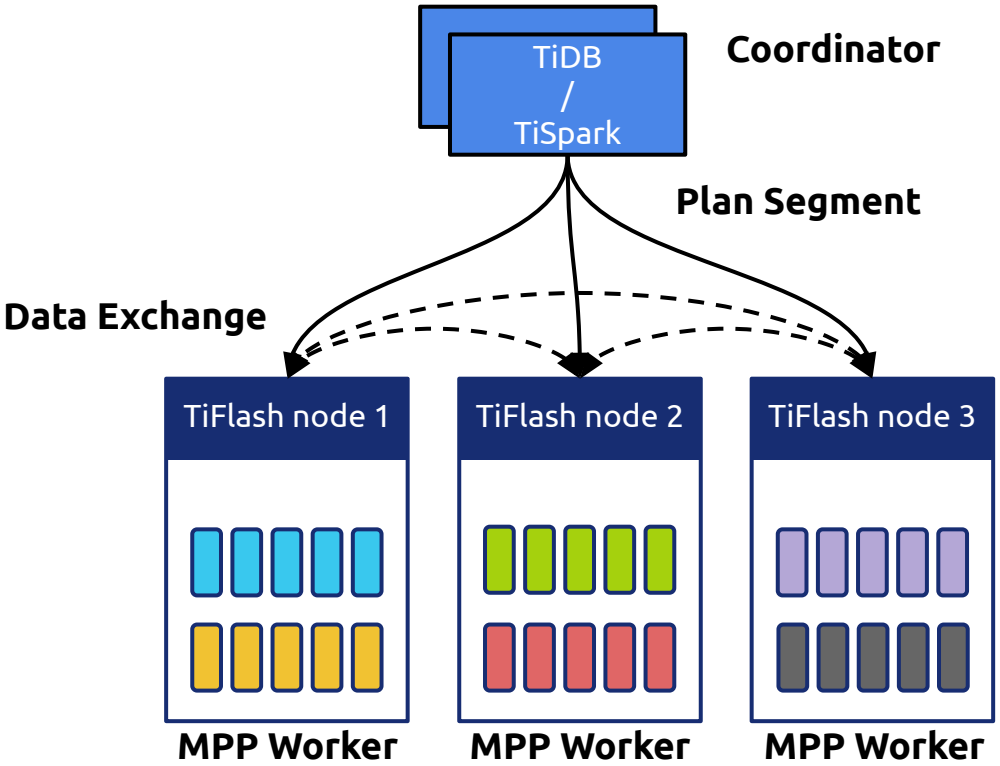


MPP Support

- TiFlash nodes form a MPP cluster by themselves

- Full computation support will
  - Further speed up TiDB by pushing down more computations
  - Speed up TiSpark by avoiding writing disk during shuffle

MPP Support

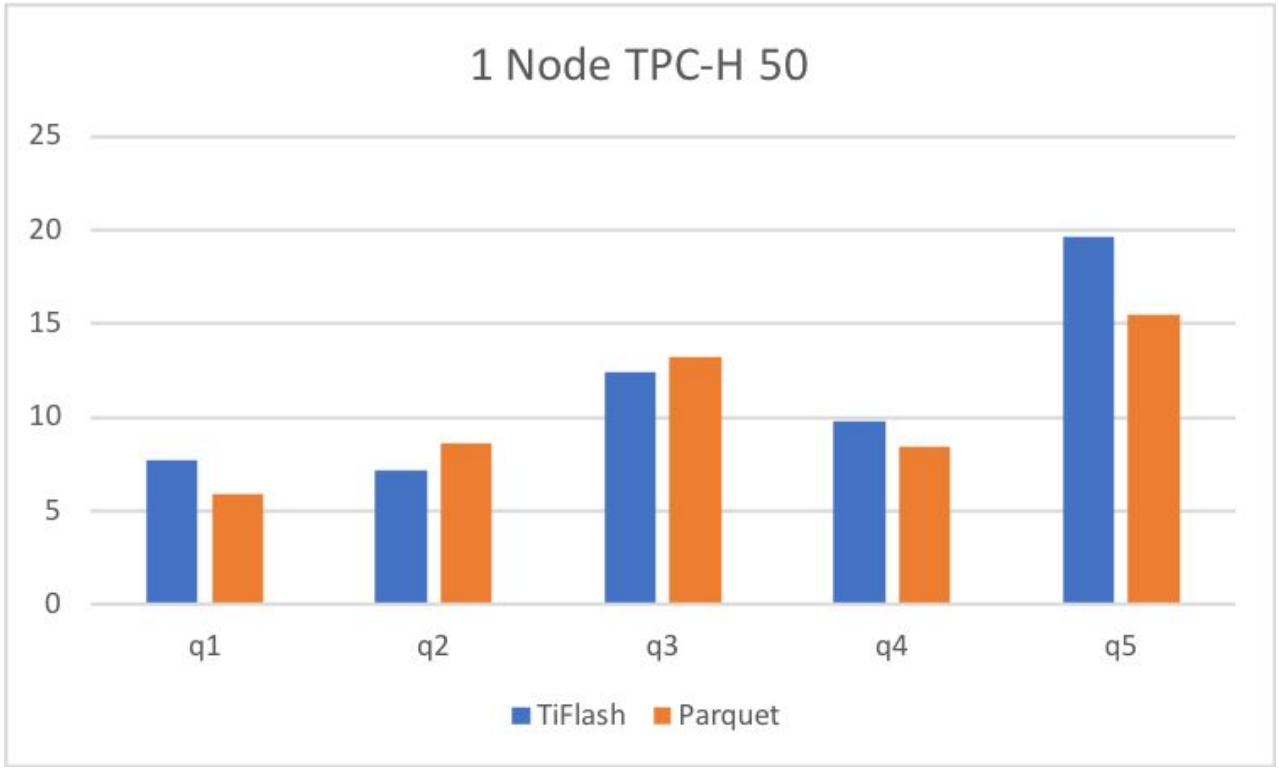

TiDB
/
TiSpark
Coordinator
Plan Segment
Data Exchange
TiFlash node 1
TiFlash node 2
TiFlash node 3
MPP Worker
MPP Worker
MPP Worker

Performance

- Comparable performance against Parquet format
  - Underlying storage format supports Multi-Raft + MVCC

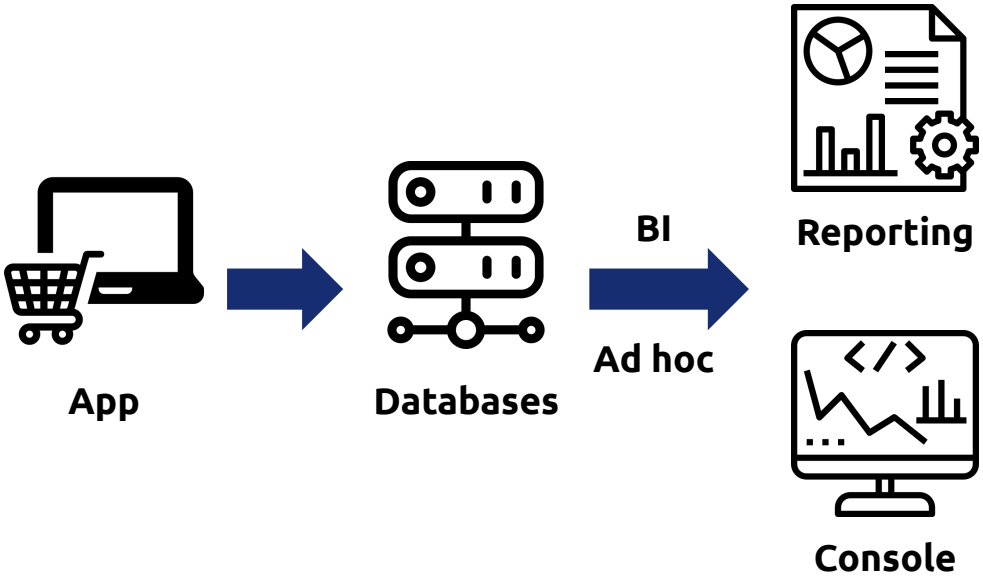
- Benchmark against Apache Spark 2.3 on Parquet
  - Pre-POC version of TiFlash + Spark

Performance

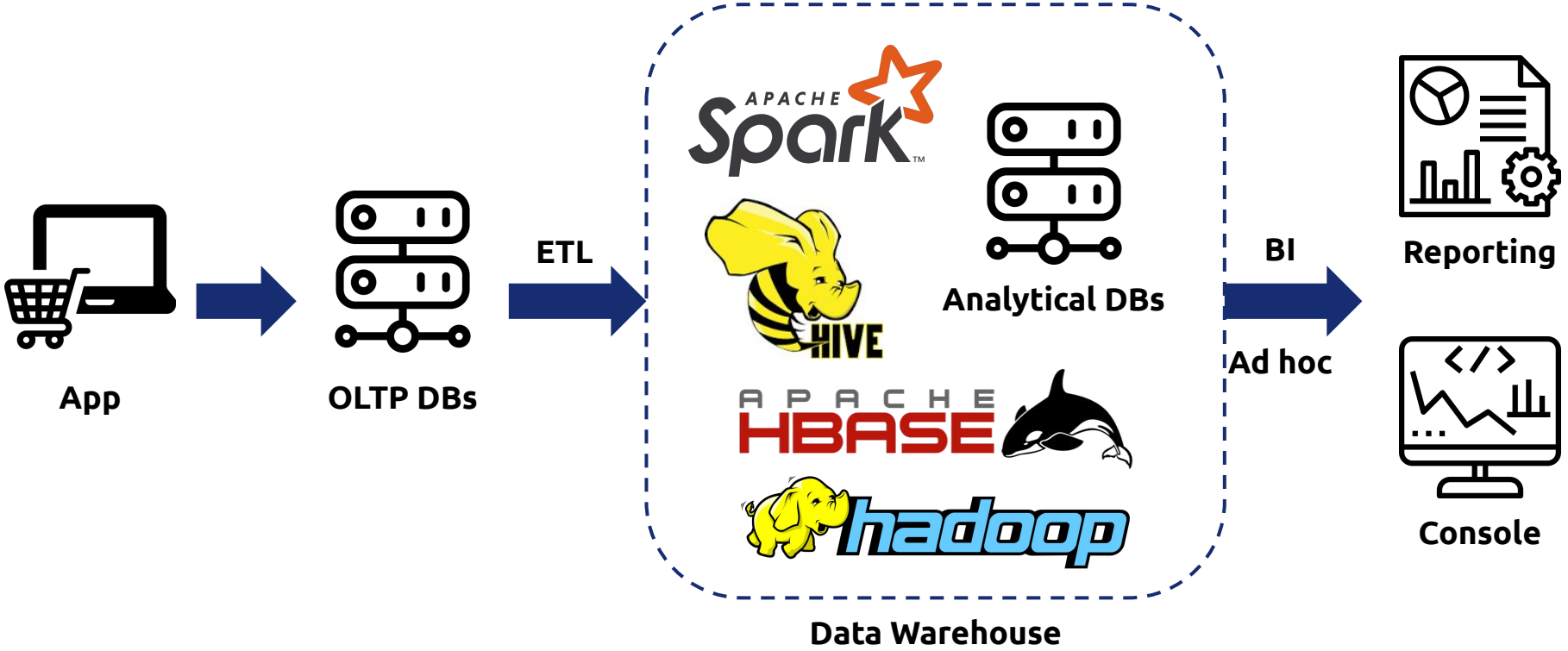


TiDB Data Platform

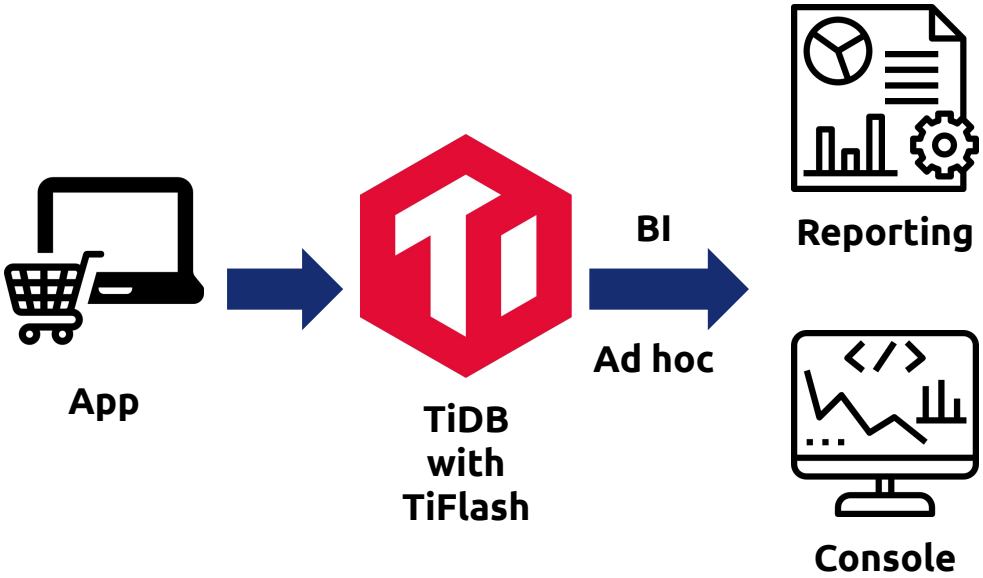
Data Platform - What You Think It Is



Data Platform - What It Really Is



Data Platform - What It Could Be



“What happened yesterday?”

VS

“What’s going on right now?”

Roadmap

- Beta / User POC in May, 2019
  - With columnar engine and isolation ready
    - Access only via Spark
- GA by the end of 2019
  - Unified coprocessor layer
    - Ready for both TiDB / TiSpark
    - Cost based access path selection
  - Possibly MPP layer done

PingCAP

Thank you

TiDB Community Slack Channel
https://pingcap.com/tidbslack/

contact us: pingcap.com