KubeCon | CloudNativeCon

North America 2023

Building a Scalable and Reliable Change Data Capture for TiKV

Charles Zheng <charlesz@netflix.com>

KubeCon | CloudNativeCon
North America 2023

Outline

- **Why TiCDC**
- **Challenges and design goals**
- **How TiCDC works internally**
  - Overall Workflow
  - TiCDC Cluster Internal
  - Pipeline: Puller, Sorter, Mounter, Sinker
- **Performance improvement**
- **Lessons learned**

Why TiCDC

Two Common Scenarios

01 Incremental data synchronization services for heterogeneous system.

02 Cross-region disaster recovery services based on primary and secondary replication

What TiCDC can do

- **Low-latency incremental data replication for various downstreams**

  TiDB -> TiCDC -> MySQL: Escape Link

  TiDB -> TiCDC -> Kafka: With Canal-JSON, Avro, Open-Protocol.

  TiDB -> TiCDC -> S3: With CSV data format

- **Support database and table filtering**
- **Support most operation through Open API**
- **Support bi-directional replication between DB clusters**

Design Goals and Challenges

Design Goals

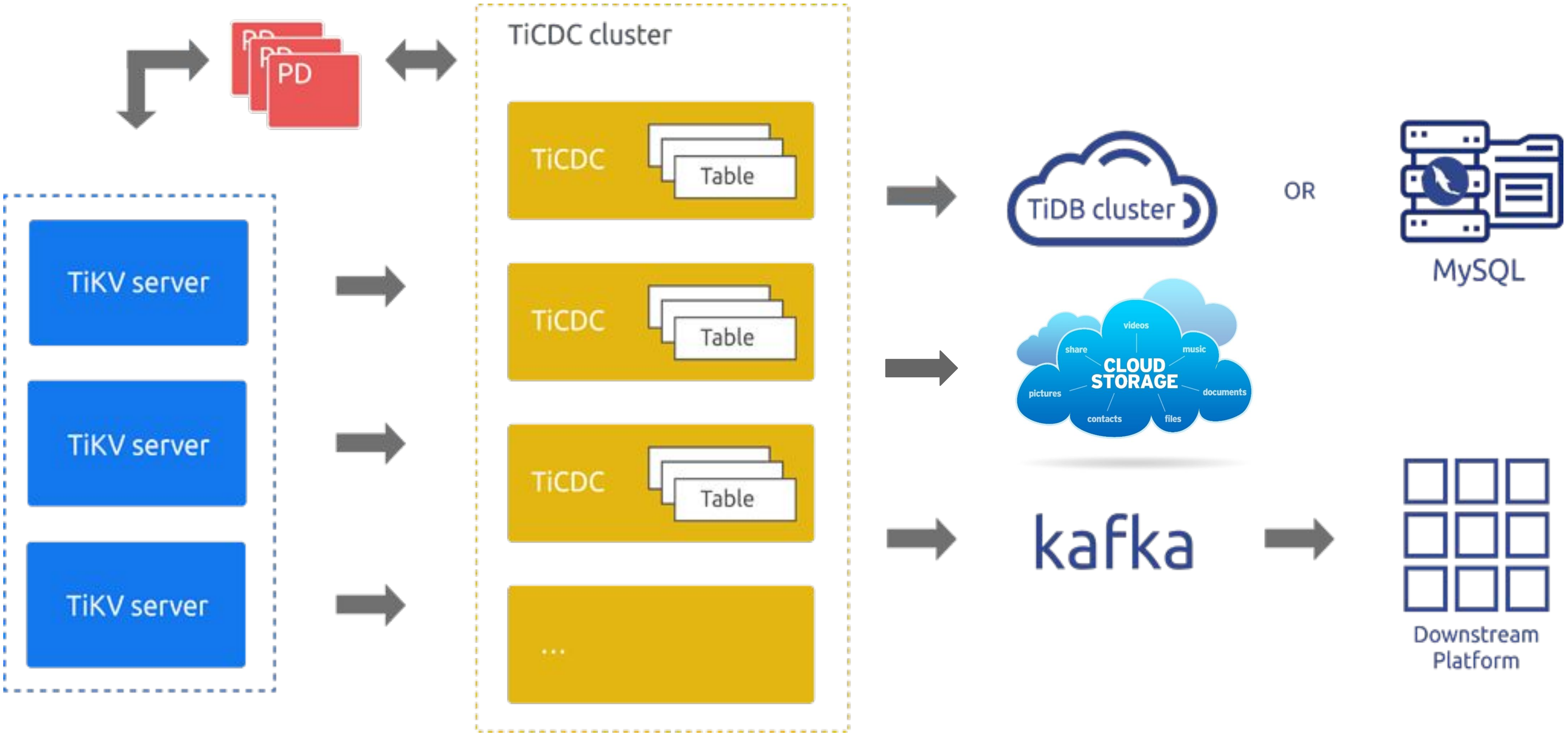
- **High Availability**
  - *Partial nodes crash would not interrupt the data sync*
- **Ensure High Throughput and Low Latency**
  - *Sync large volumes of change events concurrently*
  - *With relatively low latency*
- **Ensure Consistency and Ordering**
  - *Snapshot isolation & Eventual consistency*

Challenges

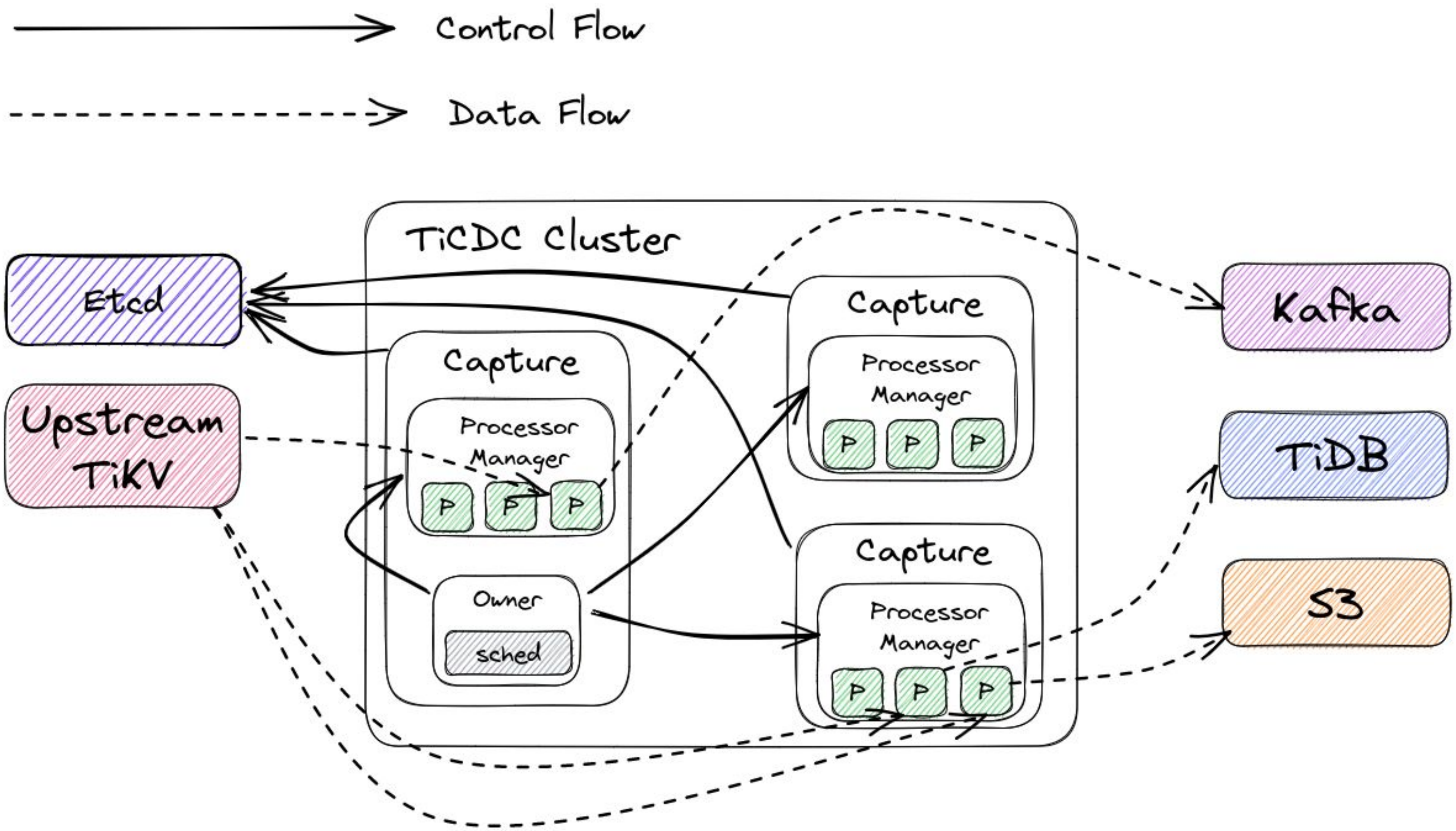
- Capture the change data instantaneously
- Aware of and catch up with the schema evolution
- Tradeoff between ordering and high throughput
- Tradeoff between consistency and low latency
- Fetch data spread across multiple nodes
- Minimize operational complexity

How CDC Works

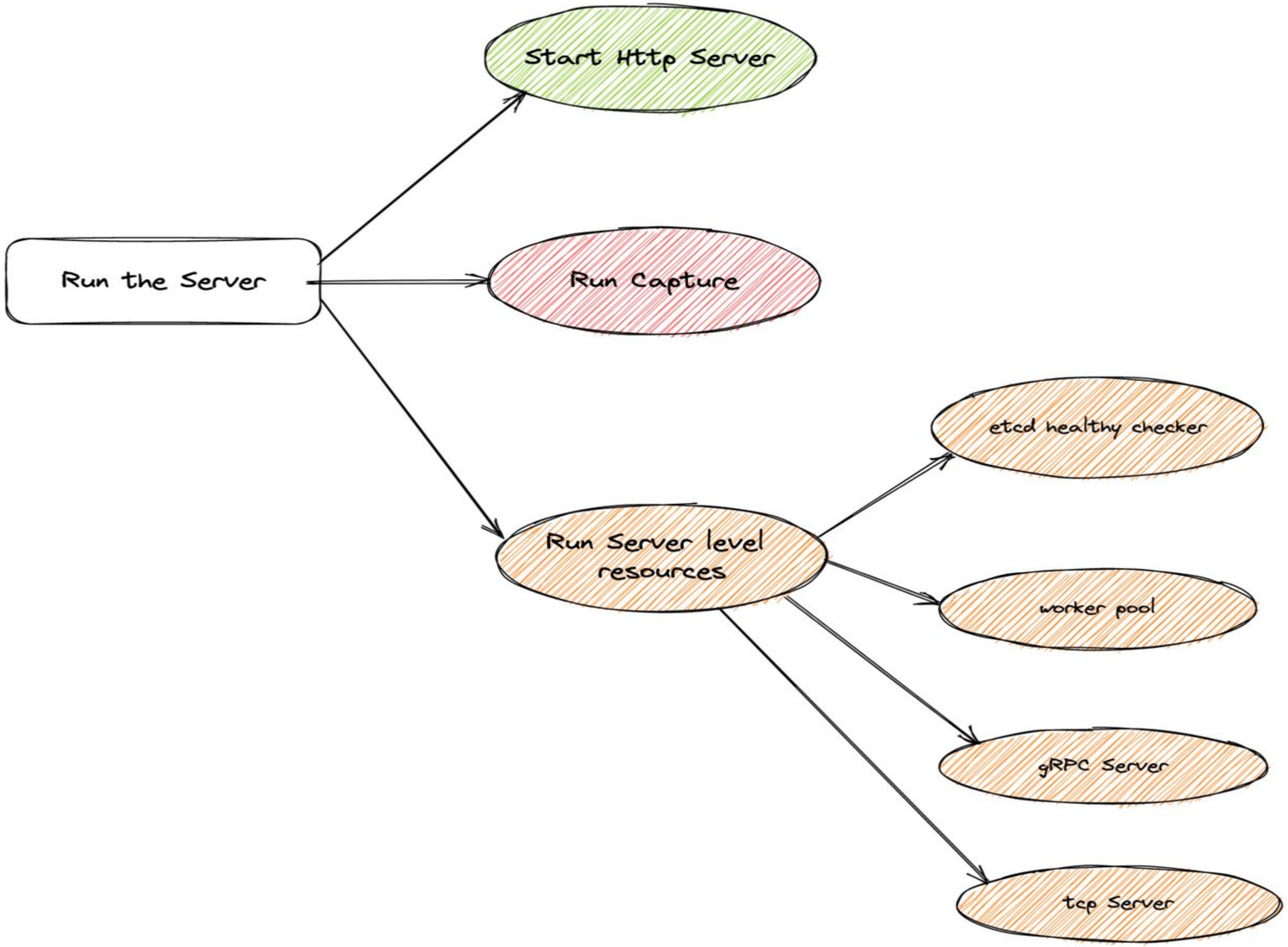
Overall Architecture



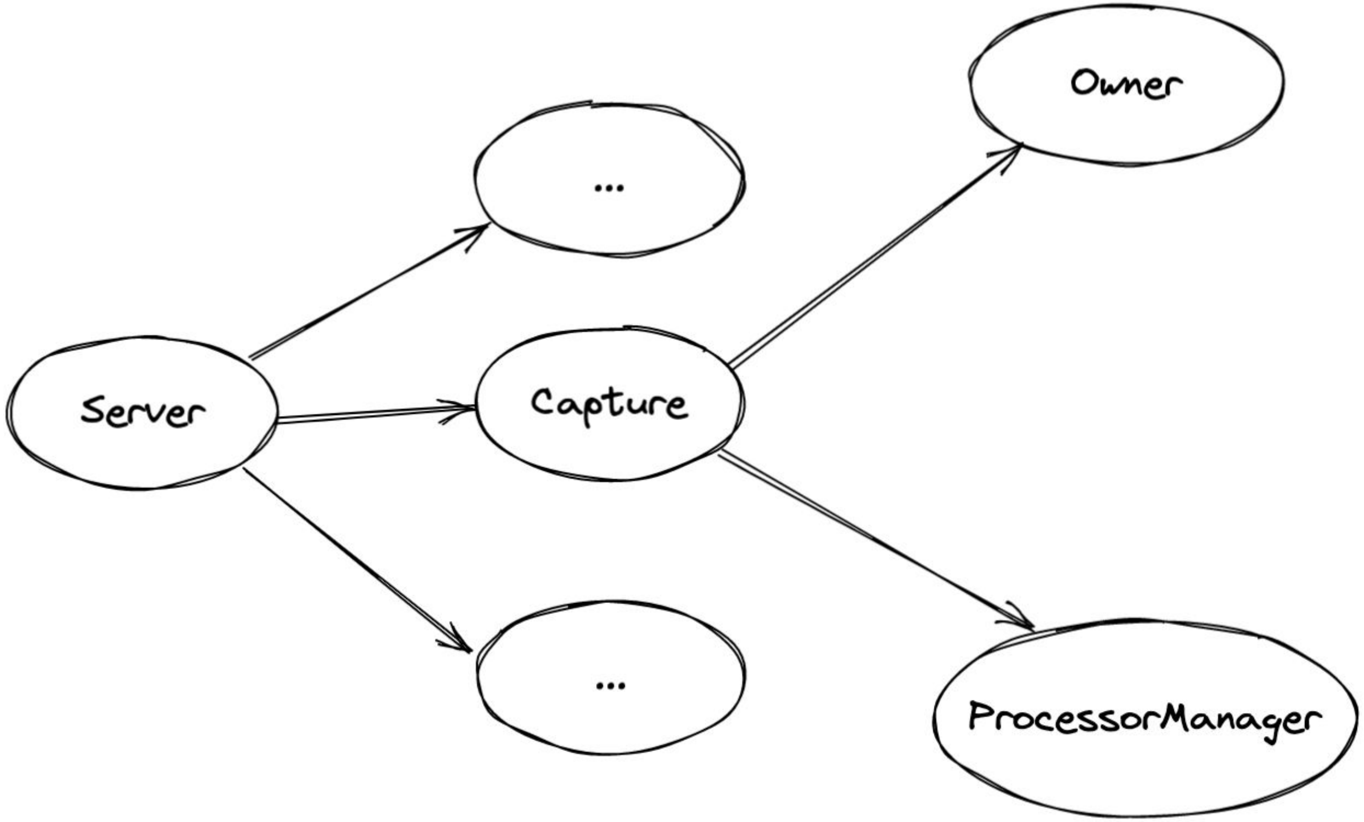
TiCDC Cluster Internal



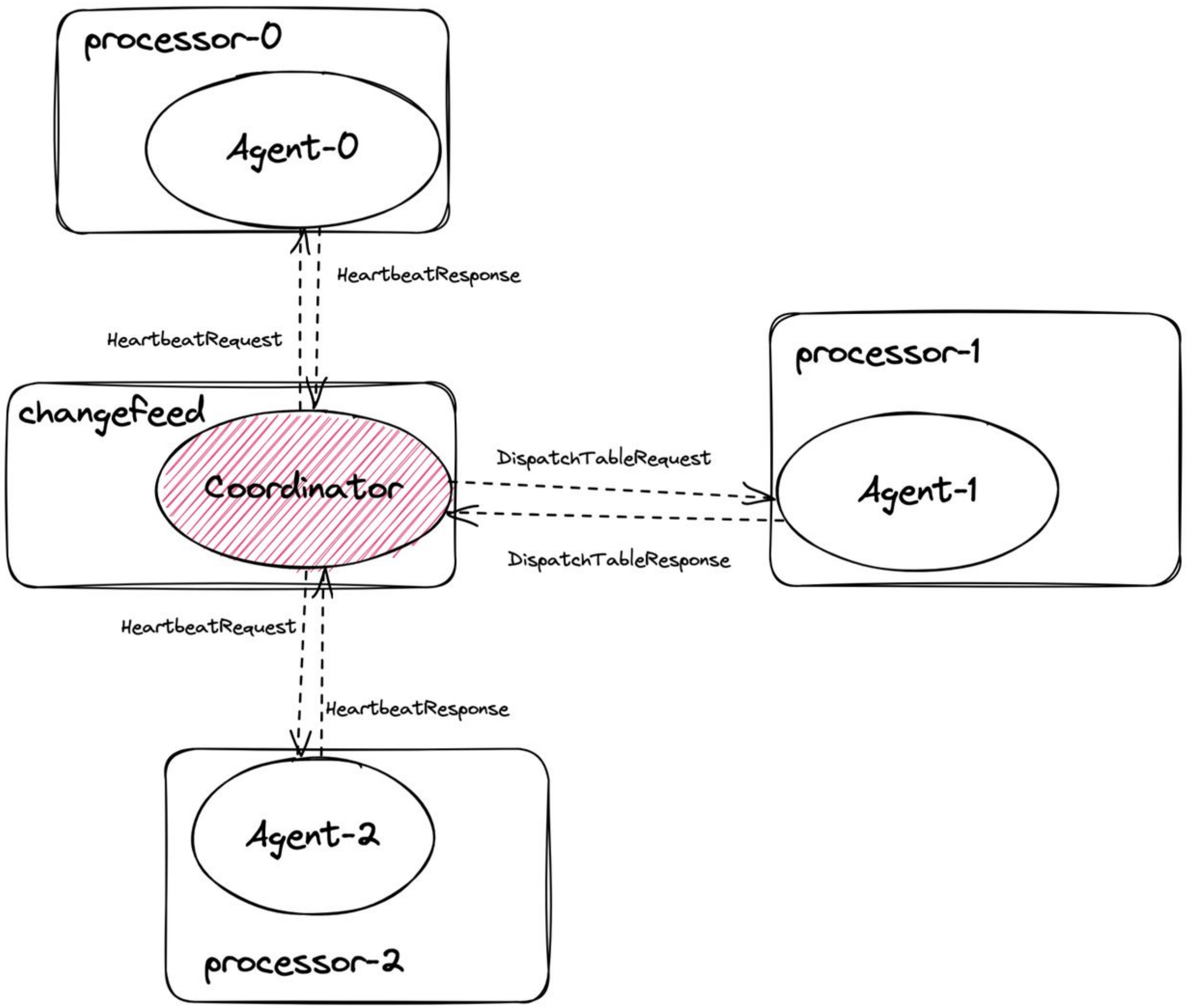
Inside a Capture Server



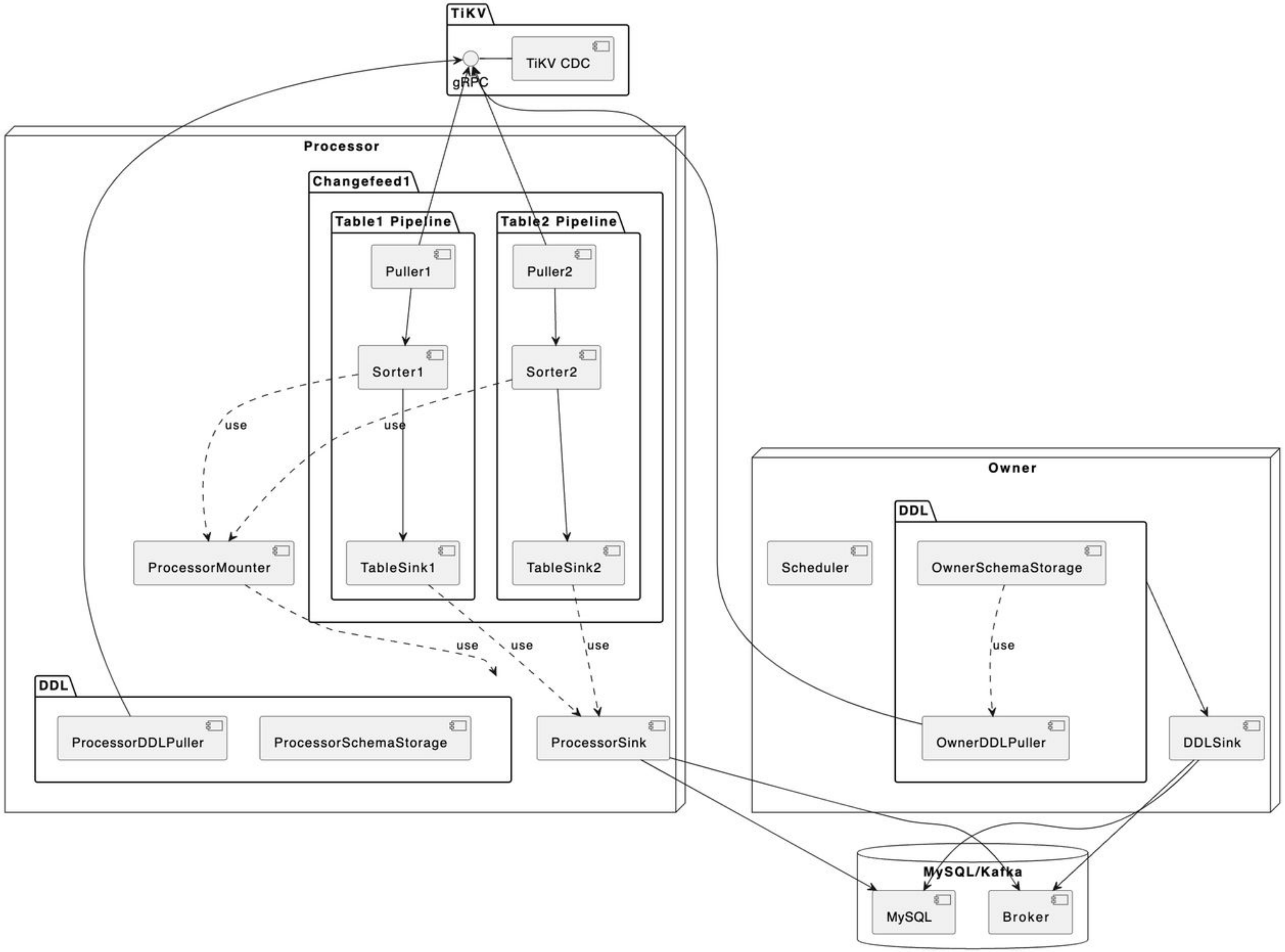
Campaign



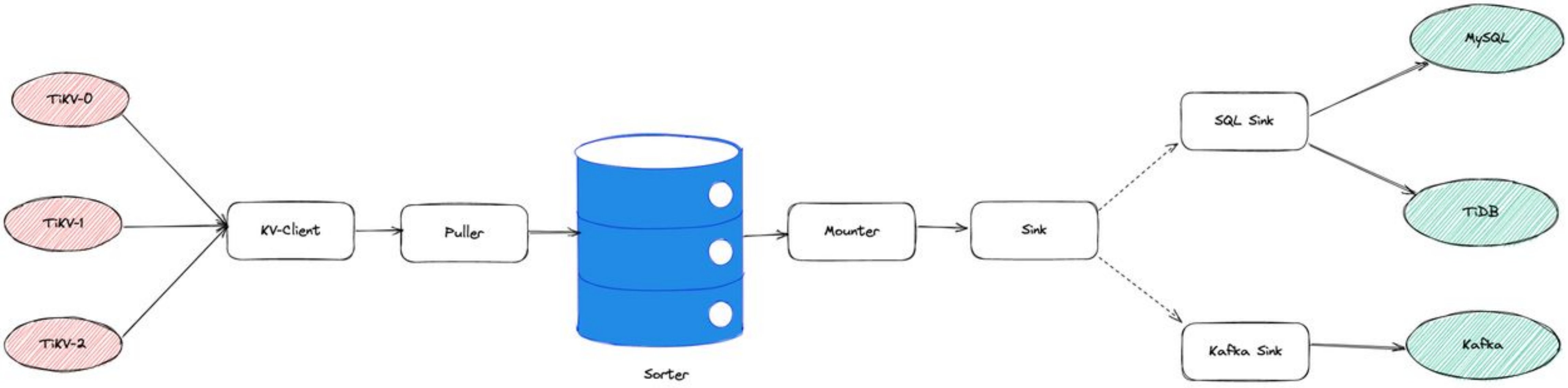
ChangeFeed Scheduling



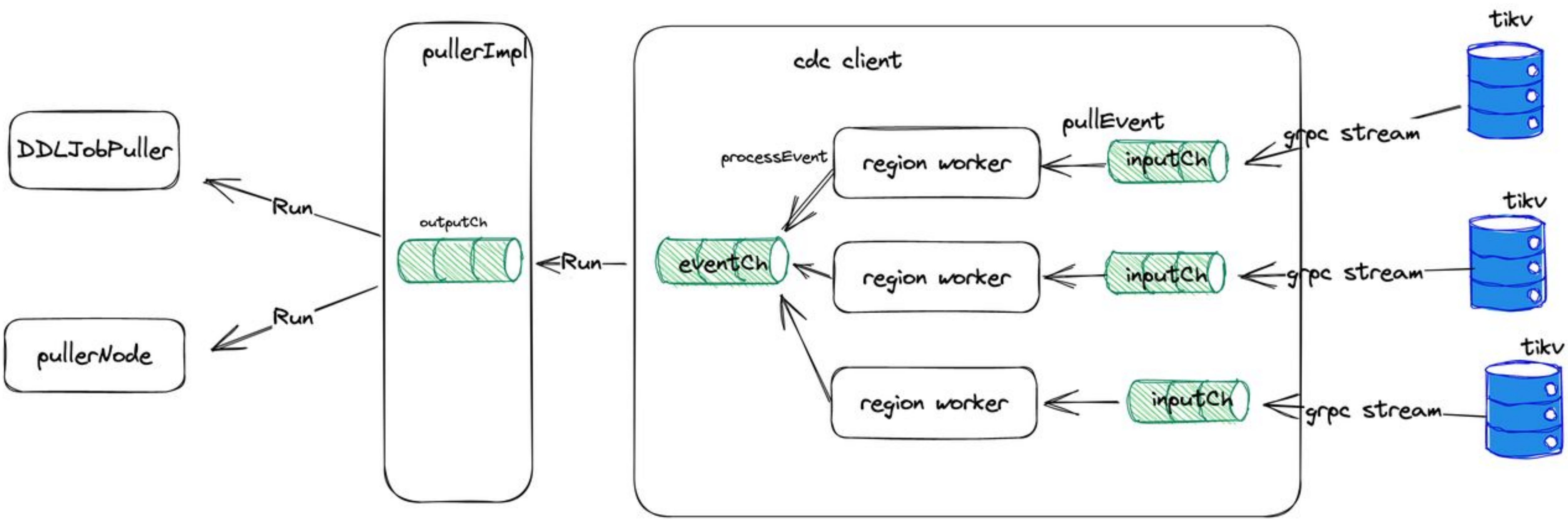
Topology


TiKV
TiKV CDC
gRPC
Processor
Changefeed1
Table1 Pipeline
Table2 Pipeline
Puller1
Puller2
Sorter1
Sorter2
use
use
ProcessorMounter
TableSink1
TableSink2
use
use
use
DDL
ProcessorDDLPuller
ProcessorSchemaStorage
ProcessorSink
Owner
DDL
Scheduler
OwnerSchemaStorage
use
OwnerDDLPuller
DDLSink
MySQL/Kafka
MySQL
Broker

Pipeline

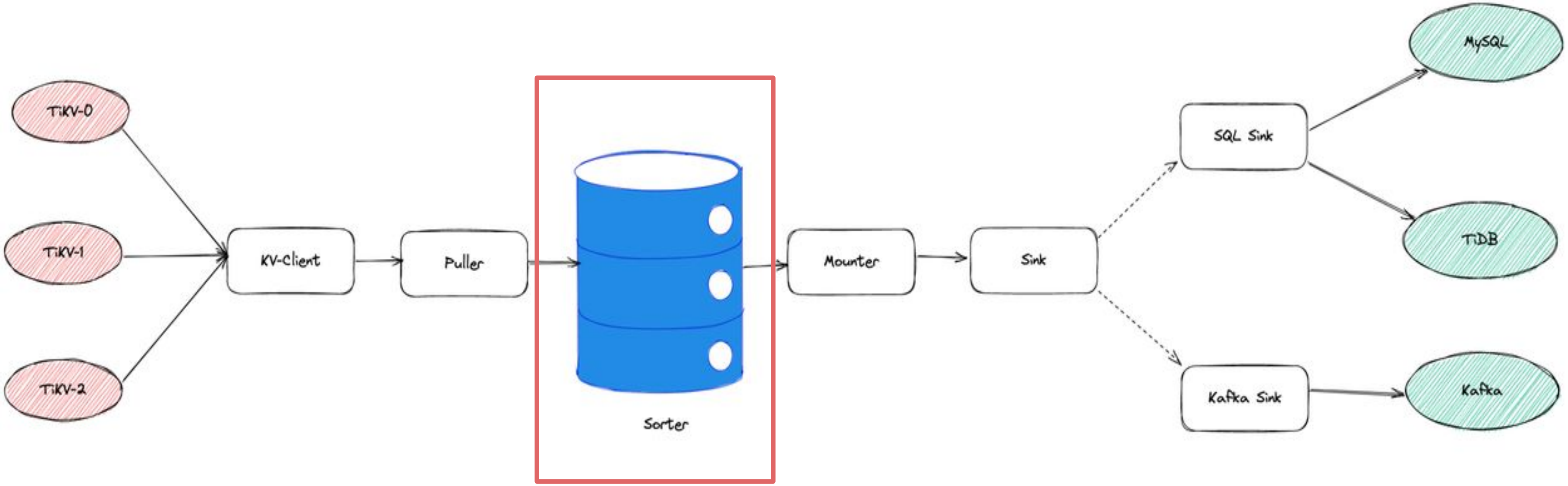

TiKV-0
TiKV-1
TiKV-2
KV-Client
Puller
Sorter
Mounter
Sink
SQL Sink
Kafka Sink
MySQL
TiDB
Kafka

How Puller Work

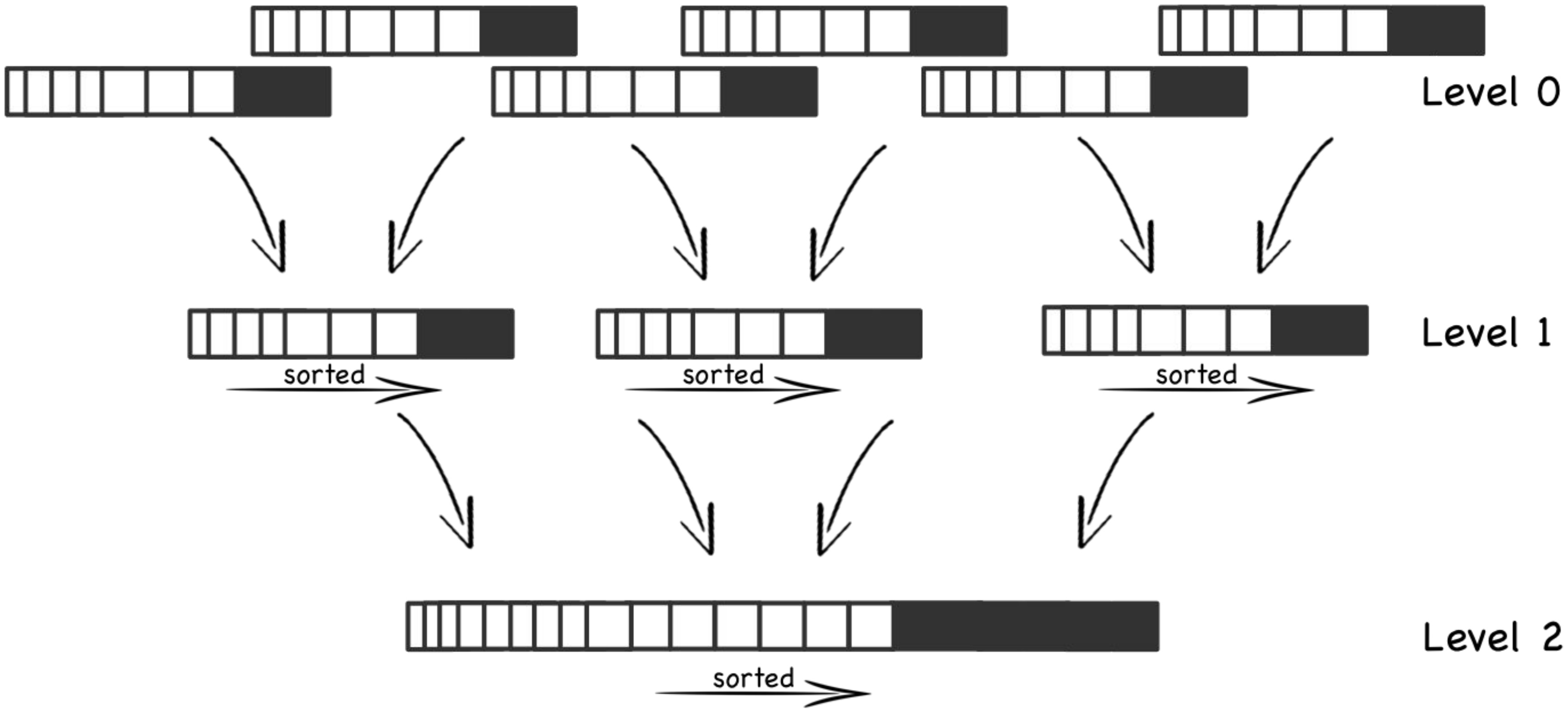


Why Sorter

- **Buffer:** Smooth out the peaks and valleys of upstream data flow
- **Sort:** Incoming events may not be in chronological order

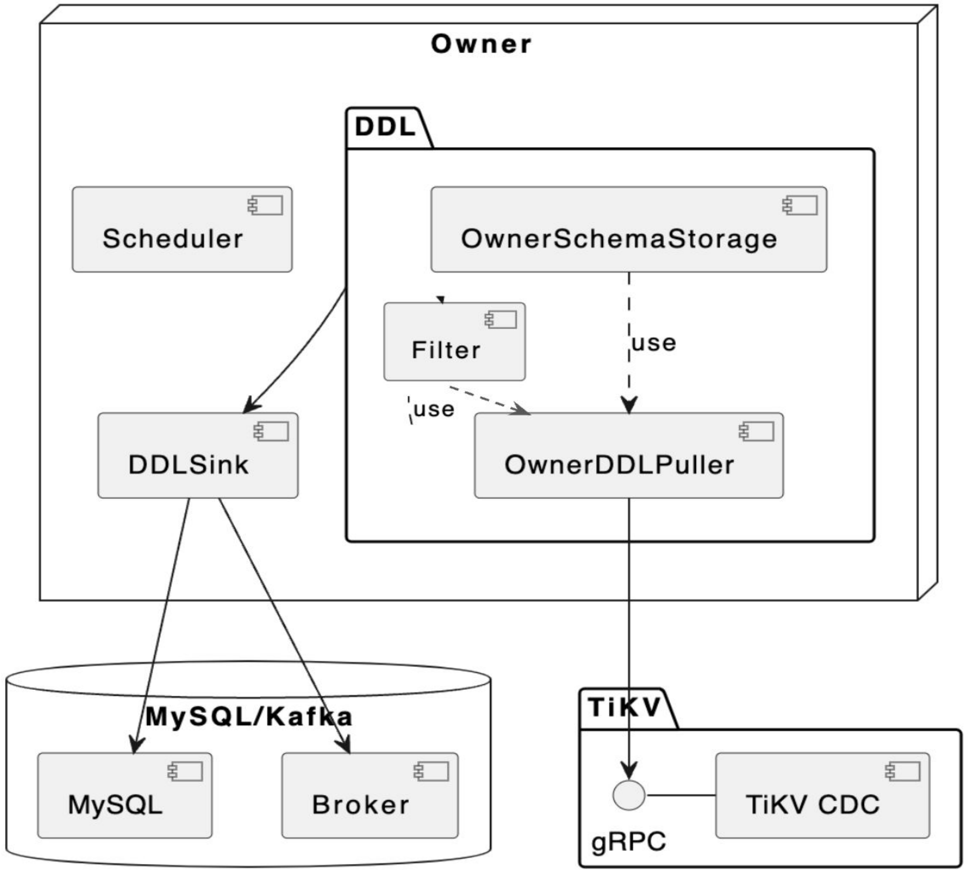


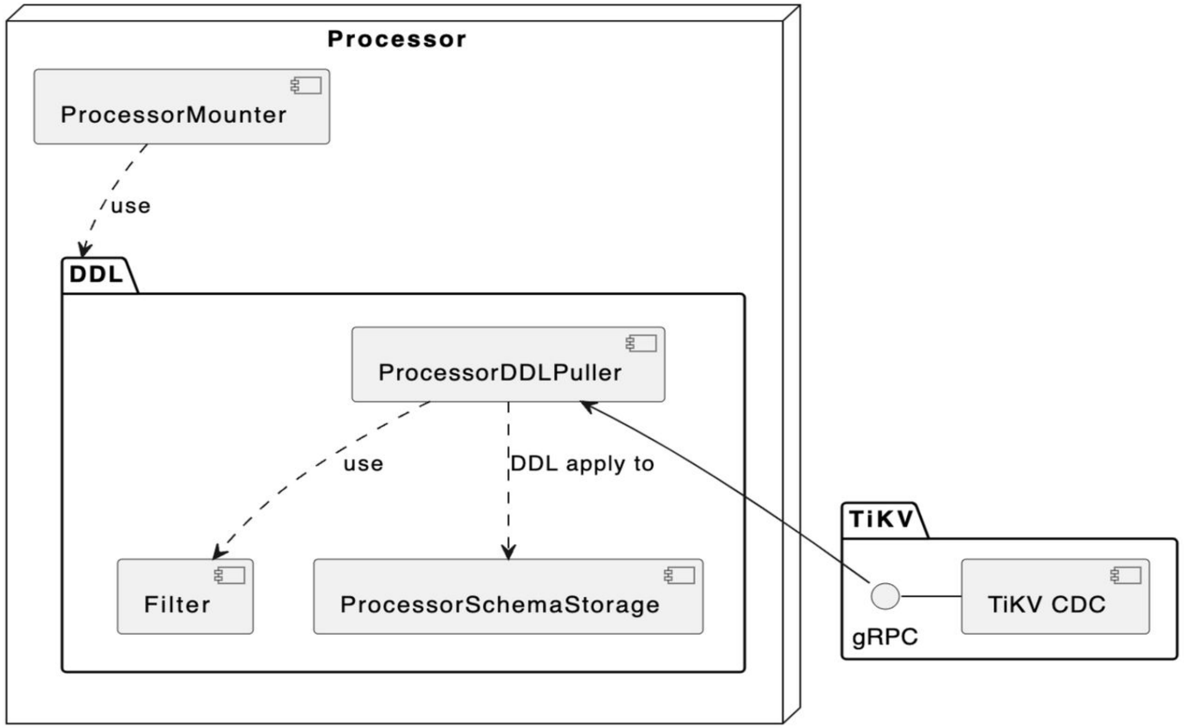
How Sorter Work



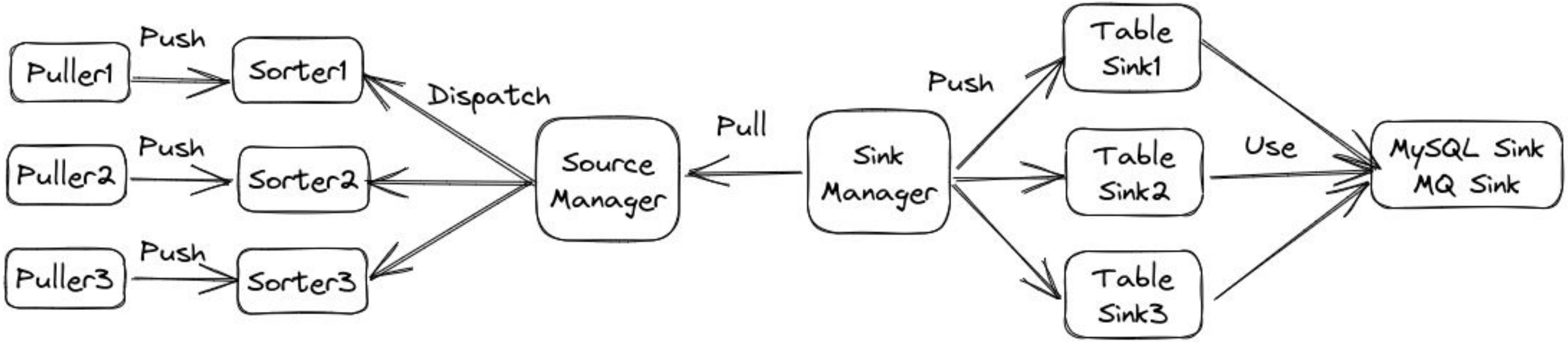
Compaction continues creating fewer, larger and larger files

How Mounter Works





How Sinker Works



Performance Improvement

TiCDC < 6.5

TiDB -> TiCDC -> TiDB

1 TiCDC node 16c/64g

Single table throughput	Big table (1200 bytes/row) 80K write QPS (insert only) throughput: 120MB MB/s
Max single table size	Up to 30~40T
Max upstream cluster data size	No limit

TiCDC < 6.5

TiDB -> TiCDC -> Kafka

1 TiCDC node 16c/64g

Single table throughput	Big table (1200 bytes/row) 35K write QPS (only insert) throughput: 52.8 MB/s
Max single table size	Up to 30~40T
Max upstream cluster data size	No limit

Throughput by Components

	sysbench	ccb	jitu	Average
puller	158k	112k	97k	128k
sorter	522k	214k	131k	250k
mounter	262k	120k	78k	161k
sink	TBD	TBD	22k (kafka sink)	76k

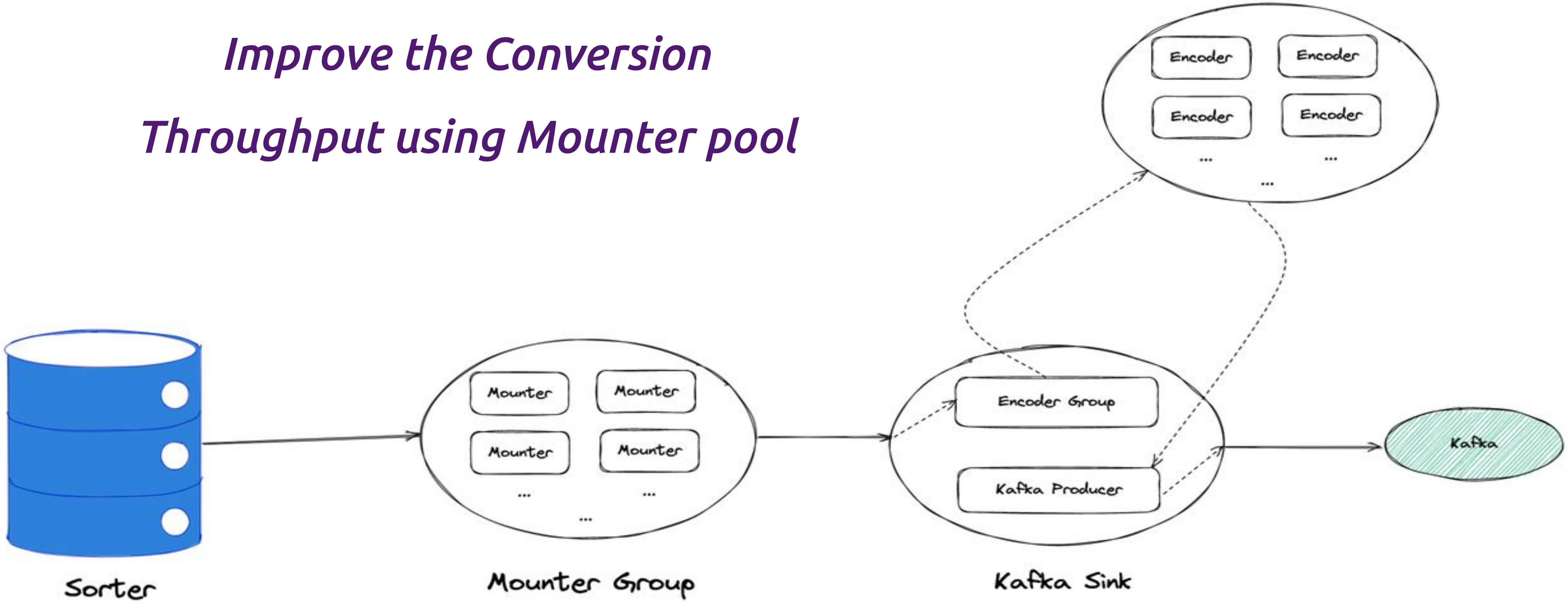
Sink is the bottleneck in most cases

How to Improve Puller

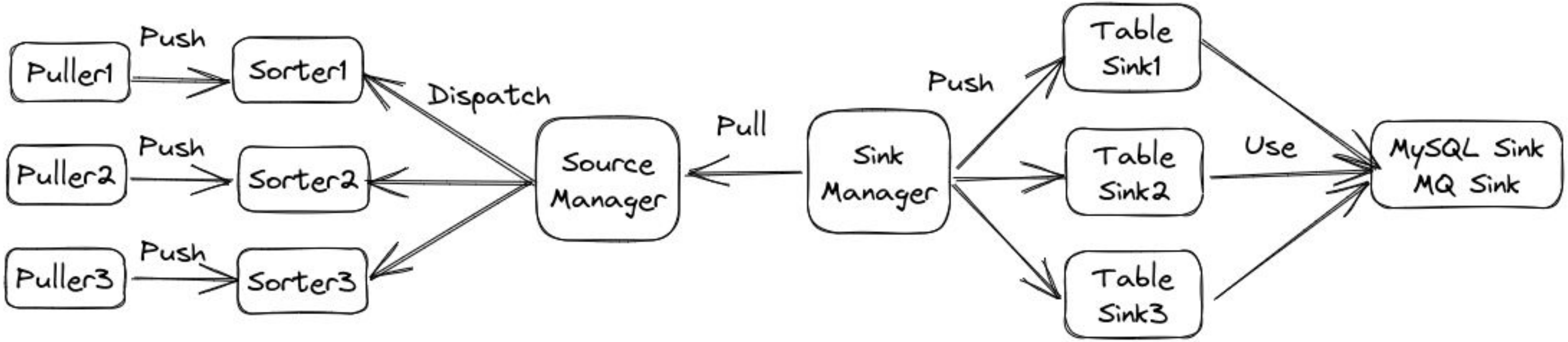
- **Process the *ResolvedTS* in batch**
- **Use read/write lock instead of the exclusive lock**
- **Optimize how the frontier inspect the region split/merge**
- **Remove unnecessary memory allocation**

How to Improve Mounter

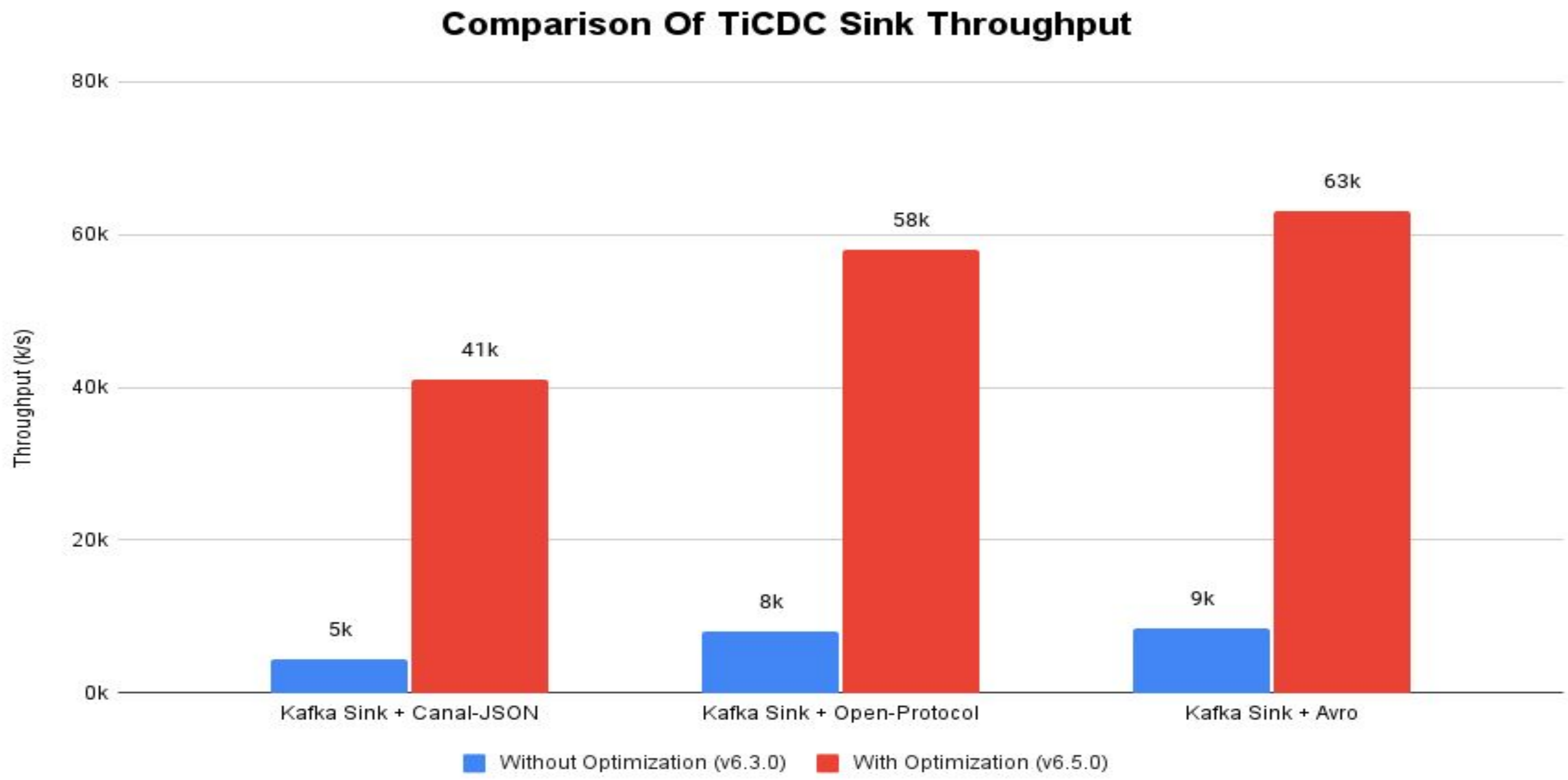
Improve the Conversion Throughput using Mounter pool



How to Improve Sinker



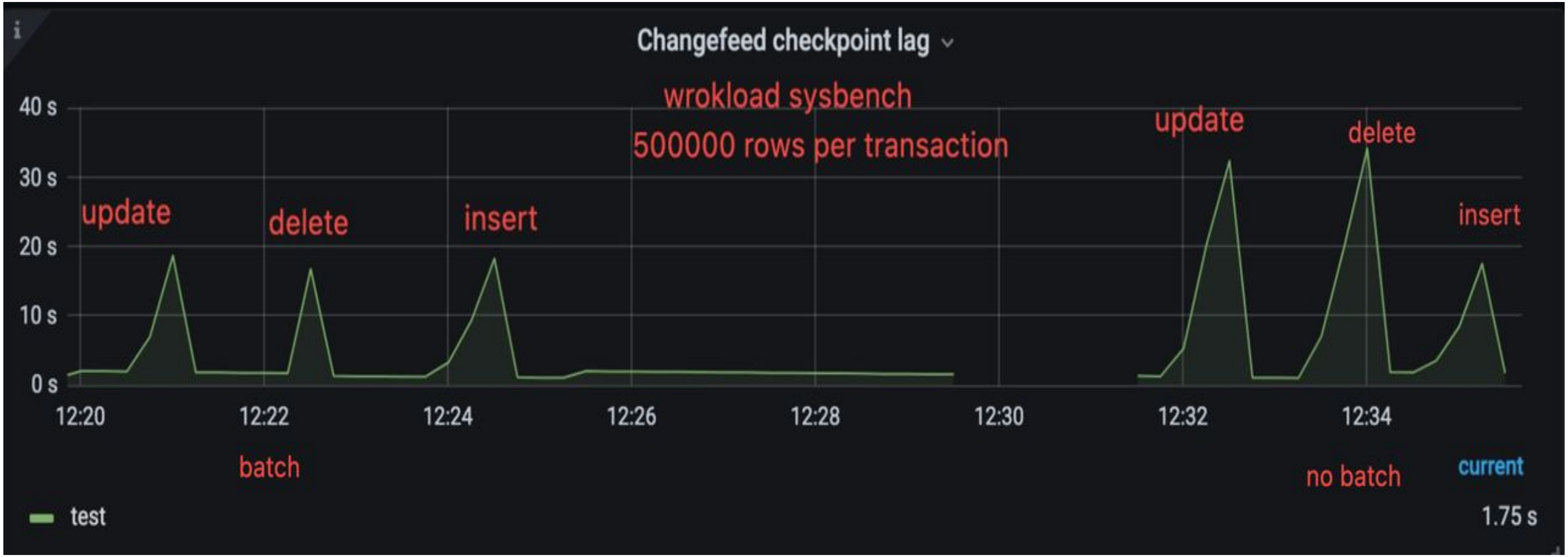
Performance Improvement (to Kafka)



Performance Improvement (to MySQL)

KubeCon | CloudNativeCon North America 2023

Sync DML events in batch



Lessons Learned

Lessons Learned

- Design the system architecture align with upstream Database
- Clear boundary between subcomponents
- Choose the push model and pull model carefully
- Implement the old value feature from day 1
- Implementing an efficient sorter as well as keeping the scale in mind

THANKS !

TiCDC: https://github.com/pingcap/tiflow

Any Questions?
find me at
Charles Zheng <charlesz@netflix.com>

Please scan the QR Code above to leave feedback on this session