User Group Analysis with ClickHouse in Bytedance

Niu Zhaojie
zhaojie.niu@bytedance.com

ByteDance 字节跳动

Outline

- Background
- First Experience of ClickHouse
- Problem and Optimization

Outline

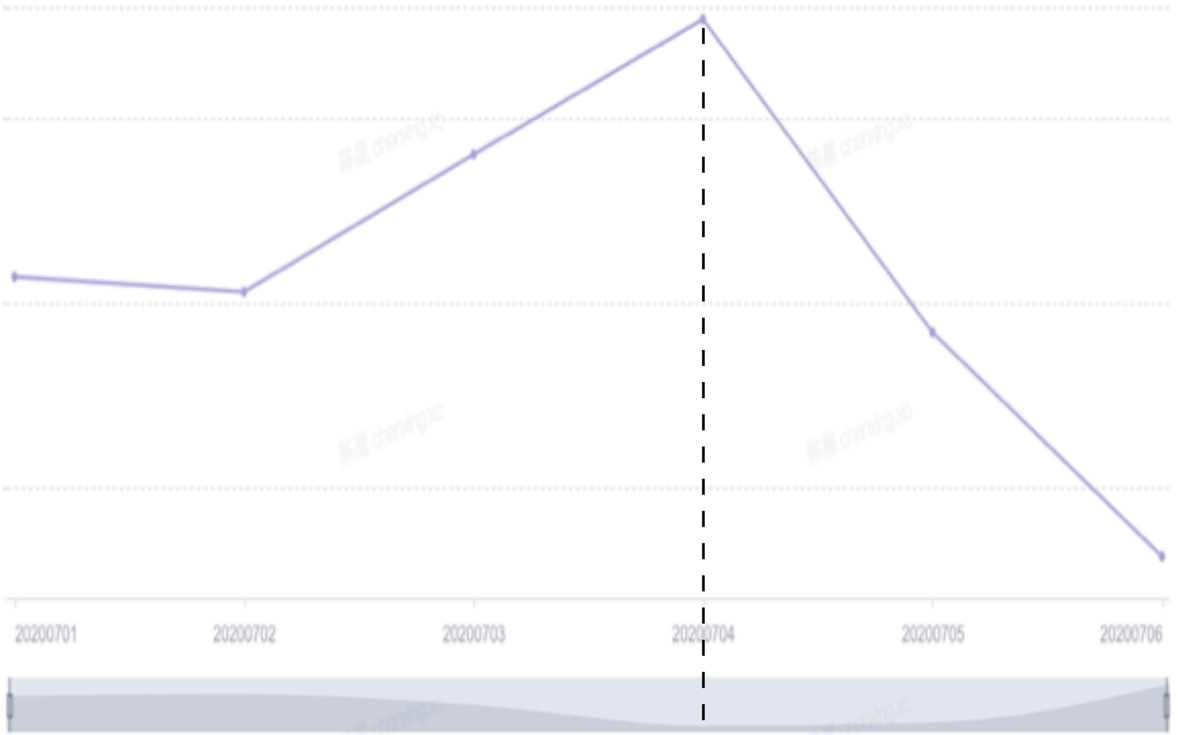
- **Background**
- First Experience of ClickHouse
- Problem and Optimization

Business Background

- Help business to increase Daily Active Users (DAU).
  - DAU = new users + retained users + returning users.
- Evaluate the impact on DAU.

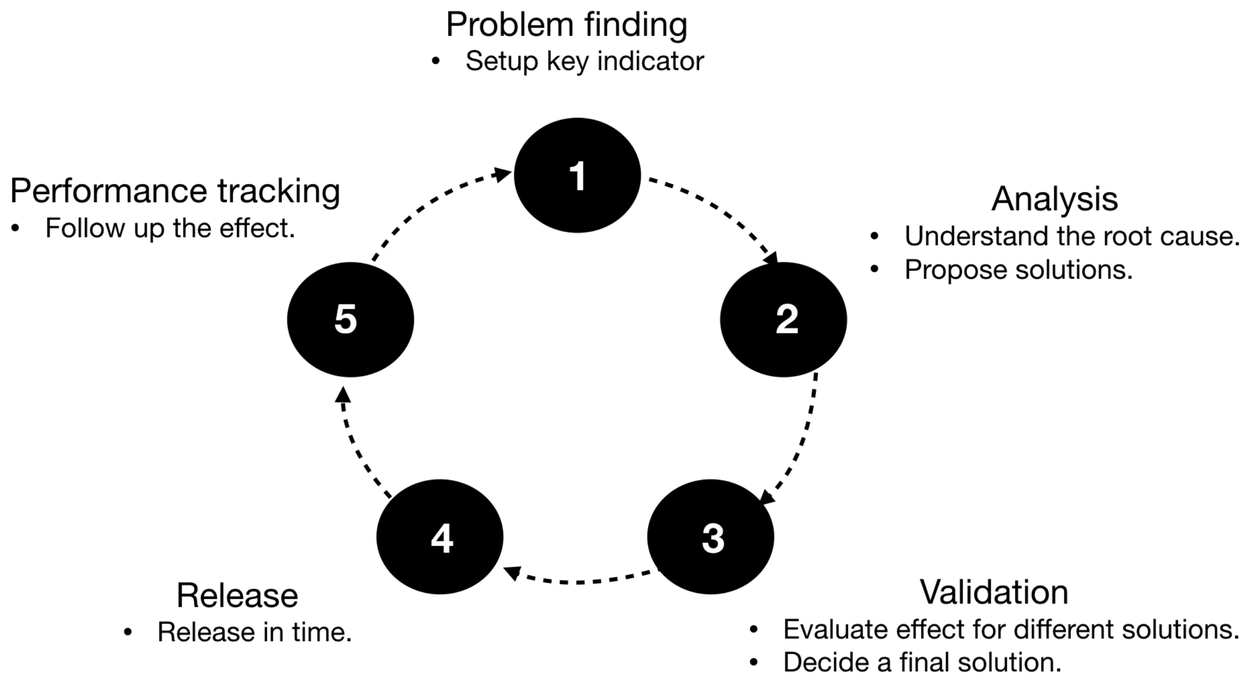
Business Background - Story

- One business find the DAU is reduced after a new release.



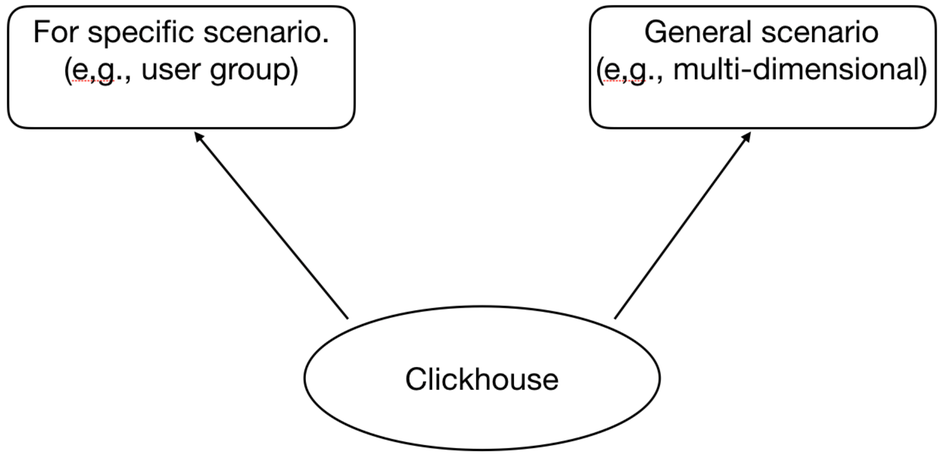
Business Background - Methodology

- The common methodology used to improve business.



Business Background - Platform

- Platform for user group analysis.
  - Fixed query pattern.
  - Indicator calculation is complicated.
  - Total volume is large.
- Platform for multi-dimensional analysis.
  - Complex query pattern.
  - Multiple data sources/models.



Technical Decision

- Existing solution (commercial, open–source).
- The requirements changes quickly and are diverse (PMs, Users).
- Low cost, highly flexible.

Open source + Self development

Using ClickHouse

- High available.
- Easy extension.
- High scalability.
- Interactive response.

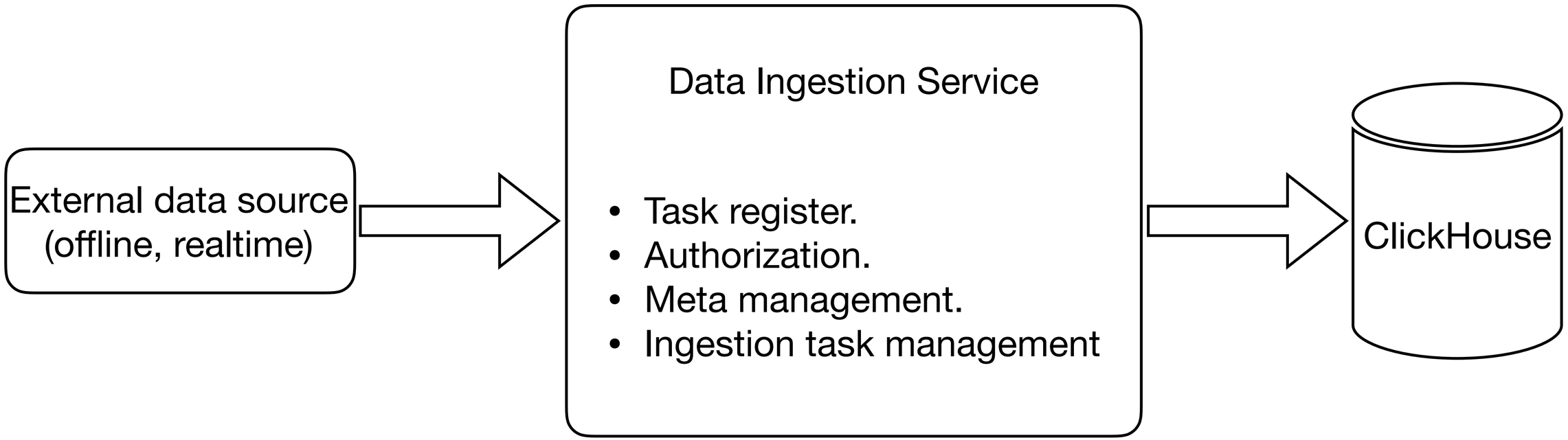
- ReplicatedMergeTree.
- Engine is easy to customize. (SQL & C++)
- Multi server & shared nothing design.
- High performance.

Outline

- Background
- **First Experience of ClickHouse**
- Problem and Optimization

Using ClickHouse at Early Stage - Ingestion

- Simplify the data ingestion for end users.



Using ClickHouse at Early Stage - SQL Enhancement

- SQL-based indicator calculation.
- UDAF enhancement.
- SQL grammar enhancement.
- Data visualization tools.

Using ClickHouse at Early Stage - Experience

- Feasibility – validated successfully in real applications.
- Interactive user experience.
- Scale well.
- Fast iteration.
- Availability satisfies requirements in most case.

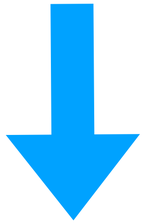
Scale further -> More users -> New challenges.

Outline

- Background
- First Experience of ClickHouse
- **Problem and Optimization**

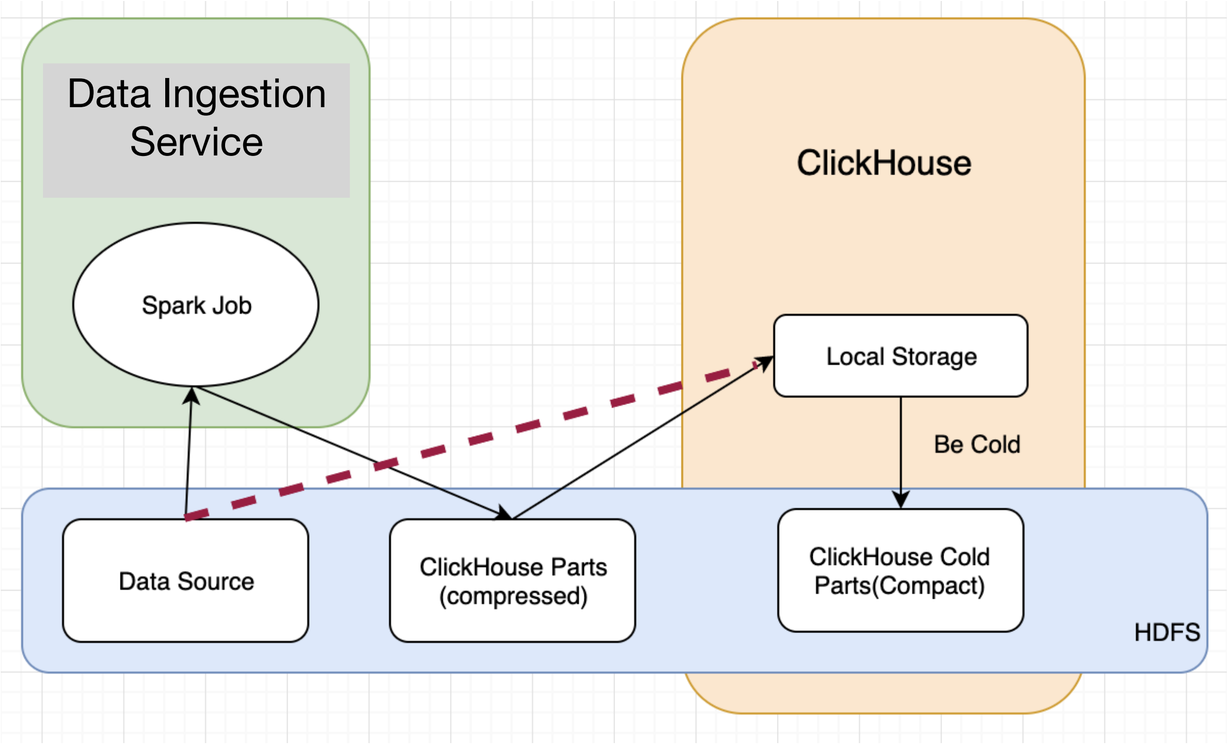
Data Related Issue - Massive Data

- Heavy data ingestion task impacts other services.
- Limitation of local storage.



- Construct data outside ClickHouse for high load business.
- Local + shared (hot/cold tired storage).

Massive Data - Optimization

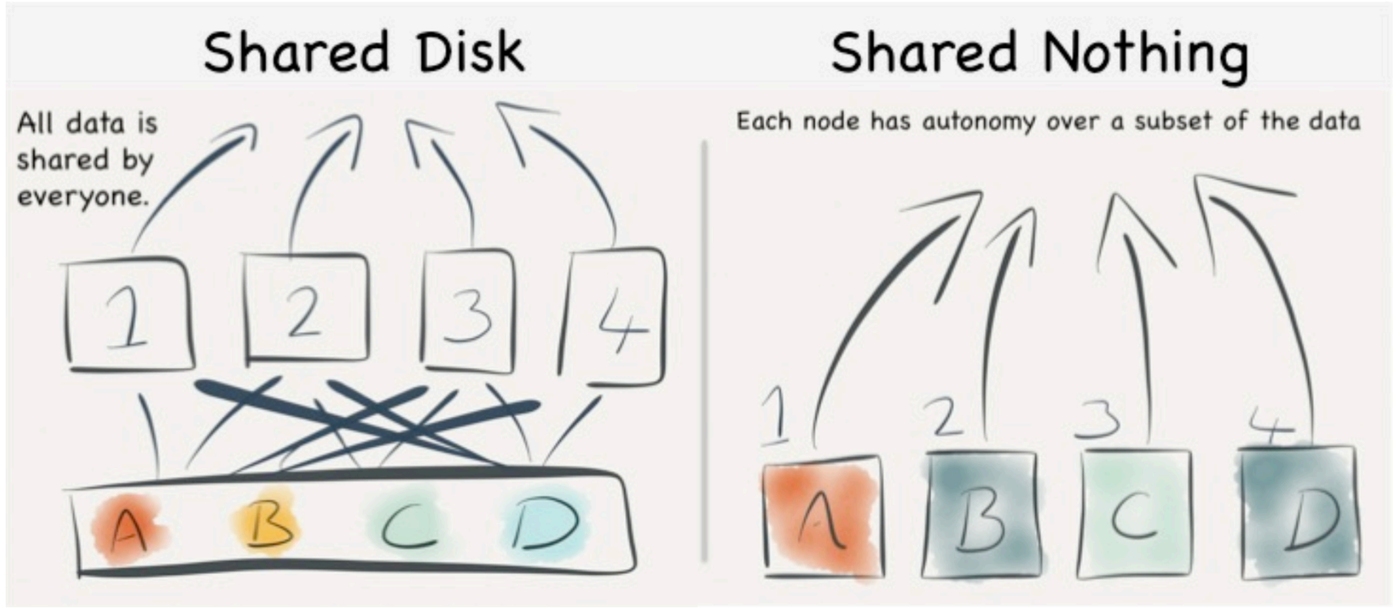


- Scale on-demand
  - Compute resource/IO.
  - Storage resource.

Is the elasticity good enough?

Massive Data - Shared Storage vs Shared Nothing

Exploration on shared storage architecture.



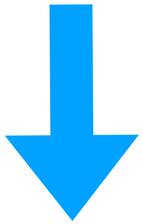
Shared storage + local cache.

- Benefit
  - Better elasticity.
  - Cloud friendly.
- Limitation
  - Extra dependency.
  - Carefully network design.

Figure is from http://www.benstopford.com/2009/11/24/understanding-the-shared-nothing-architecture/

Data Related Issue - Dynamic Schema

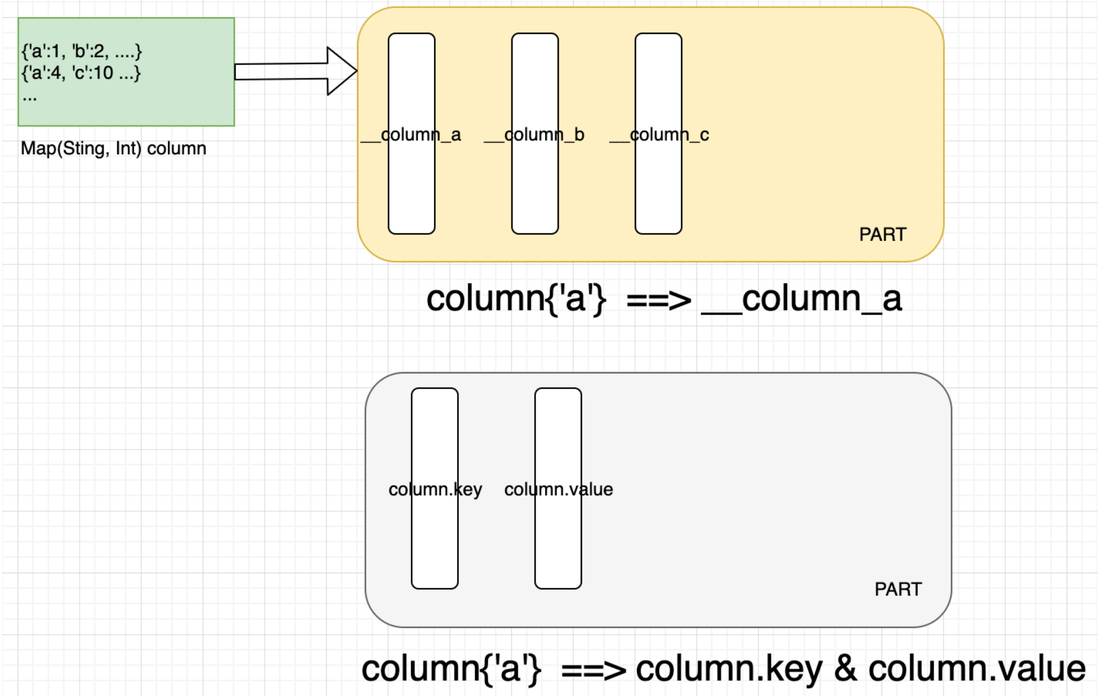
- Data model:
  - Dynamic schema (JSON format).
  - Flexibility vs Performance.



- Map type
  - Easy extension.
  - Specific query mode to achieve good performance.

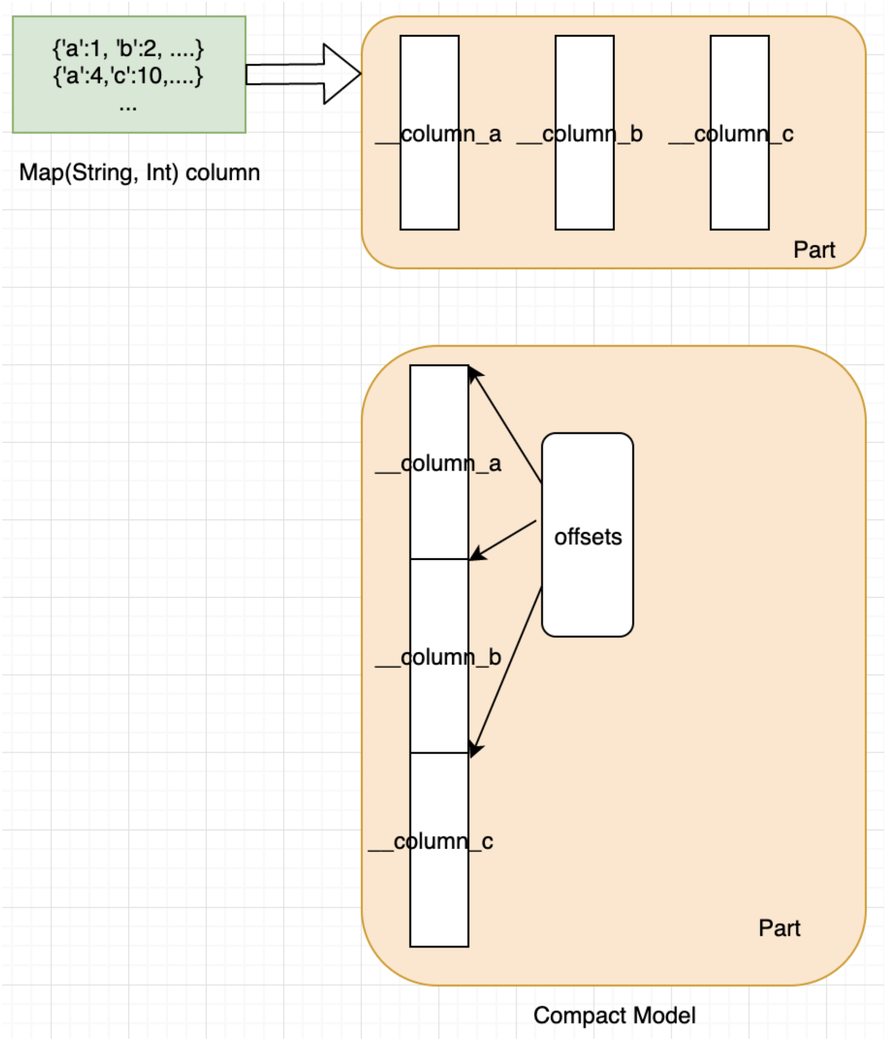
Dynamic Schema - Solution

- Extend json to columns by keys.



- Total keys are controllable.
- Small files issues.

Dynamic Schema - Enhancement



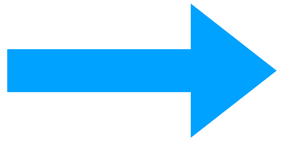
- Quota for map key.
- Compact map storage format.

High Availability Related Issue

- Data partitions keep increasing.
- Business extension.
- Nodes Failures become more.
- Recovery time become longer.
- ReplicatedMergeTree (ZK) becomes bottleneck.
- Operation becomes more complicated.

Failure Recovery

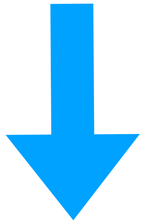
- Restarting
  - Software bug.
  - Hardware issue.
  - OOM.
  - System upgrade.



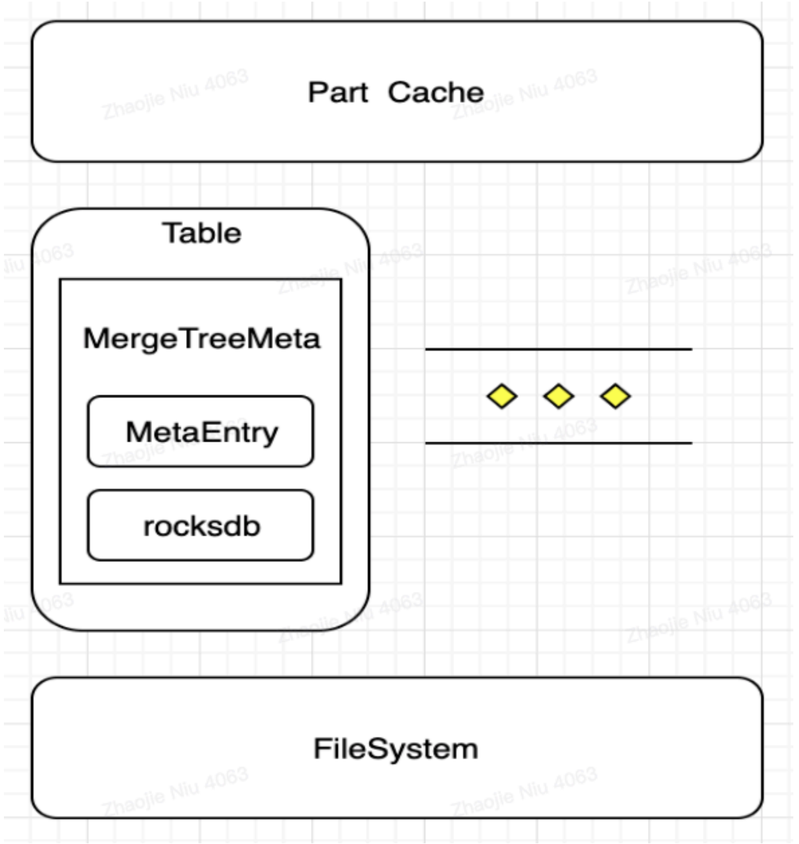
Metadata need long time to load
(~ x hours or even more
In Bytedance)

Failure Recovery - Optimization

- Persistent metadata (part info).
- Cache hot meta in memory.



- Result
  - No obvious performance loss.
  - Can support more parts in single node.
  - Restarting: hours –> minutes.



High Availability Other Issue

- Zookeeper high load: HAMergeTree (Optimized version of ReplicatedMergeTree)
- Operation platform.

More detailes: https://live.bytedance.com/8889/4218416

Performance Related Issues

- Many reasons can cause performance issue.
- Common approach for performance optimization.
  - Adjustment from application side.
  - Customized for specific application.
  - Materialized view.

More detailes: https://live.bytedance.com/8889/4218416

Future Work

- Shared storage architecture.
- Containerization.
- Resource usage and isolation.
- Service stability.
- Data lake support.

We Are Hiring

- Location: CN, SG
- My email: zhaojie.niu@bytedance.com

THANKS.