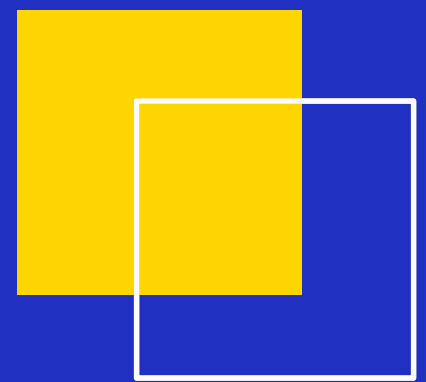


Databend

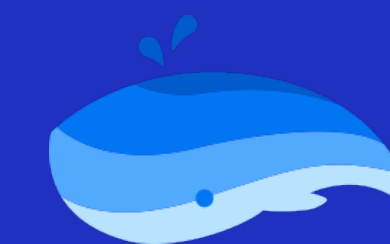
# 基于 Databend 的 TiDB 数据归档实践

主讲人：冯光普

2022.10



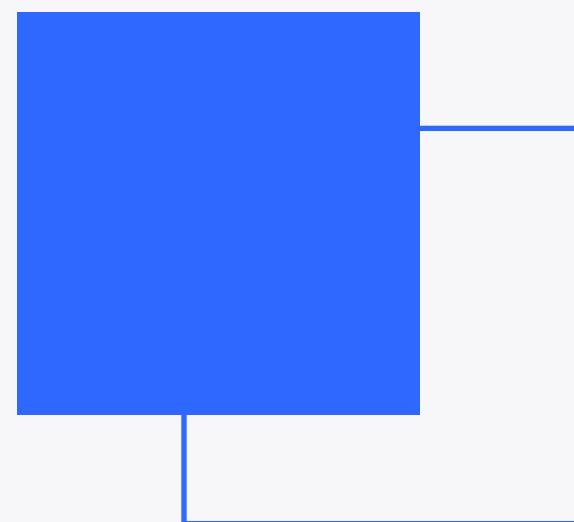
# 冯光普



Databend

- 多点DMALL数据库负责人
  - MySQL / TiDB / Redis / MongoDB
  - 数据库平台
  - DB中间件，双活架构
- 更早，阿里巴巴数据库AlicSQL团队

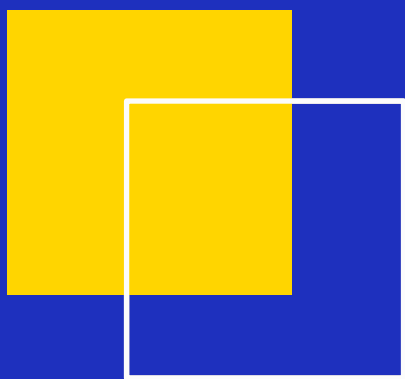




# 目录

CONTENTS

- 为什么选择 Databend 归档 TiDB
- 归档工具、归档流程、实践效果
- 归档实践总结，未来展望



# 为什么选择 Databend 归档 TiDB

# TiDB 数据归档, why?



## TiDB透明水平扩展

- 研发无感：数据增加，架构保持不变

## TiDB基本无限容量

- 循环增强：更多的数据 -> 更多的分析需求 -> 更多的数据

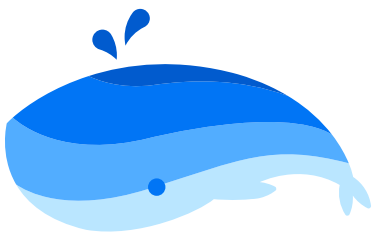
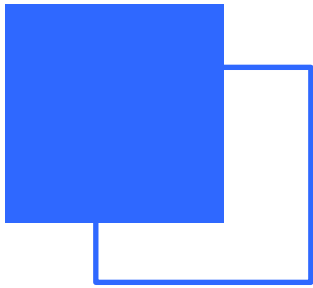
## TiDB扩缩容便捷

- DBA友好：加节点即扩容，自动rebalance



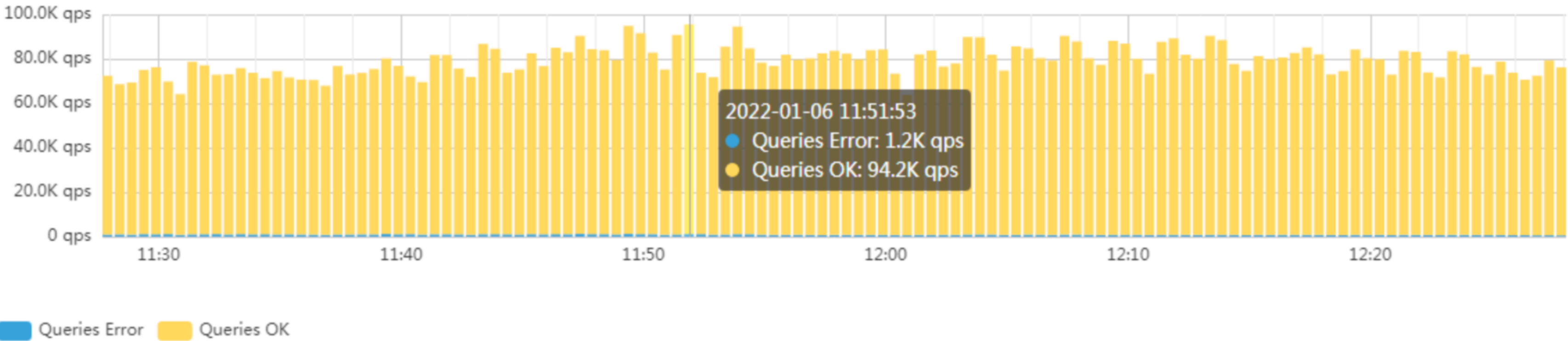
成本



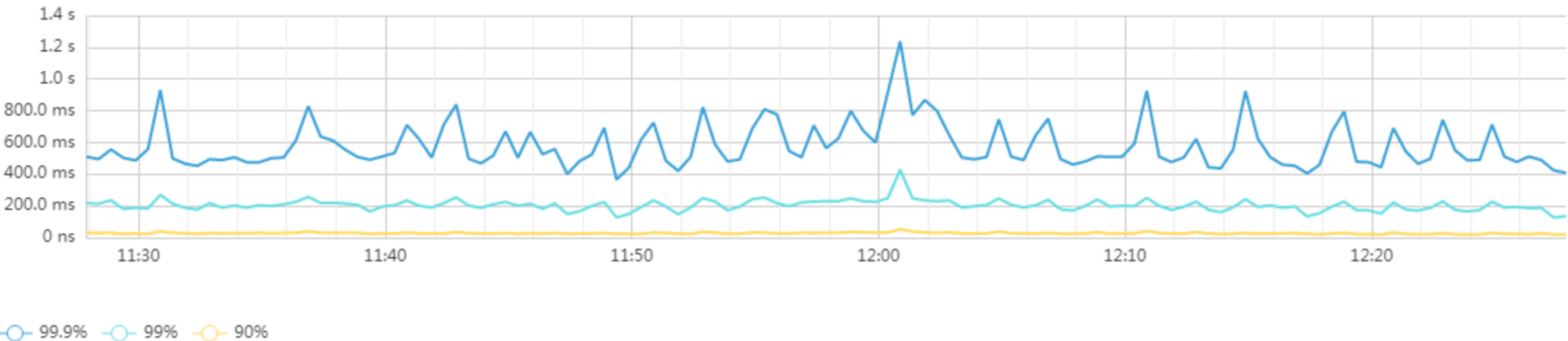


Databend

QPS



延迟



在线实例 >

PD	TiDB
3 / 3	9 / 9
TiKV	TiFlash
21 / 21	3 / 3

监控和告警

[查看监控 >](#)

[查看 57 条告警 >](#)

[运行诊断 >](#)

- 24个存储节点
- 60TB数据

# 选 Databend 归档 TiDB, why?



## 基于对象存储, 成本低

- SSD块设备的1/10
- 按实际用量付费

## 在线查询

- 无须从冷备中恢复
- 可接受的查询性能

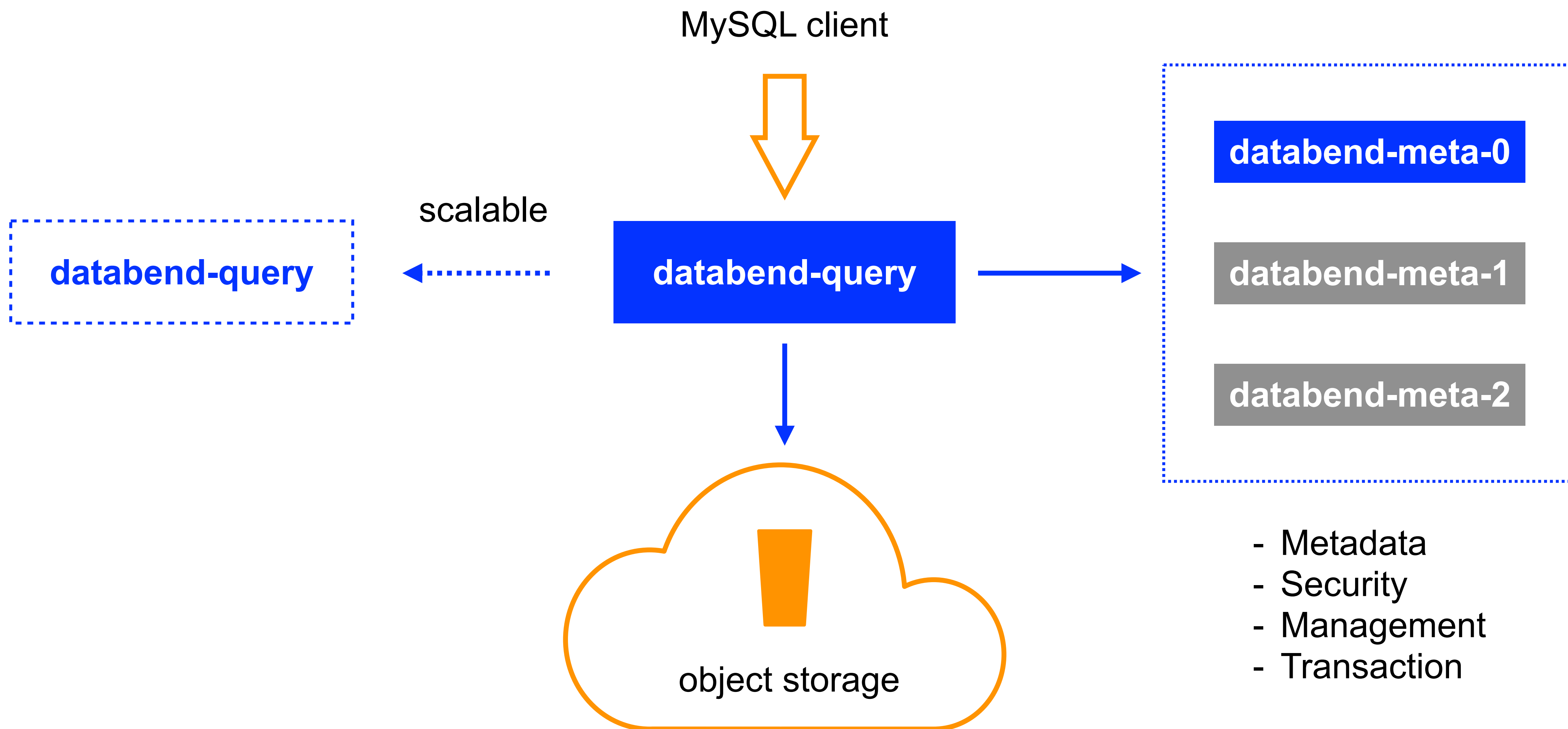
## 支持大单表

- TB级别
- 分批持续归档

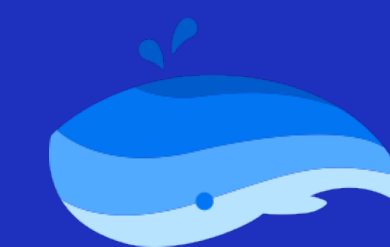
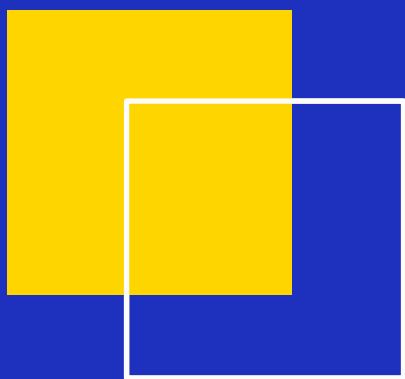
## 兼容TiDB

- MySQL协议
- 数据类型兼容

# Databend 归档 TiDB

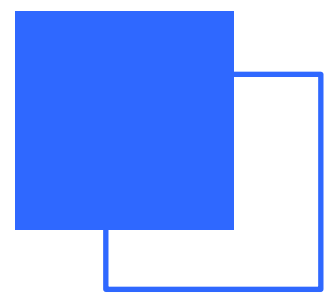




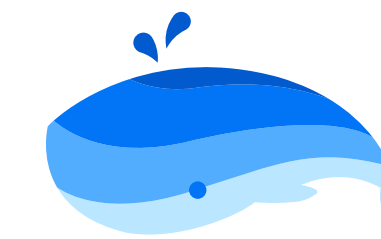


Databend

# 归档工具、归档流程、实践效果



# 理想的归档工具



Databend

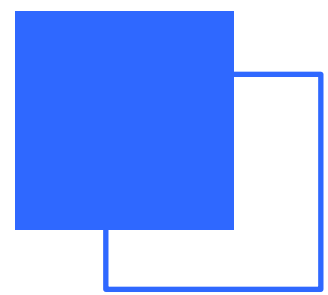
## pt-archiver的问题

- bulk模式下（load file），遇到JSON字段特殊字符，处理有异常
- 串行方式执行SELECT-INSERT-DELETE，难以发挥目标端性能
- 批量数据量太多，可能导致TiDB故障，或TiCDC延迟
- 批量数据量太少，Databend中会产生大量snapshot，归档越来越慢

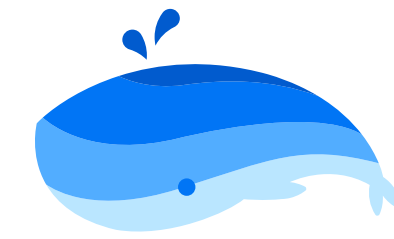
保护源端，可自适应调整执行速率

源端读取  
目标端写入  
解耦

发挥目的端性能



# 归档工具（安全+高效）



Databend

TiDB

Databend

1

CREATE table...

table transform(type, index)

2

SELECT... limit N

SELECT... limit N

SELECT... limit N

flow control

batch ~ 10MB

memory limit

3

channel

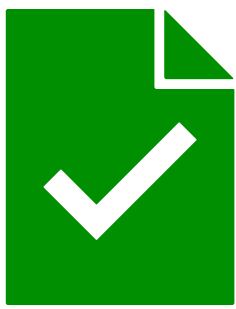
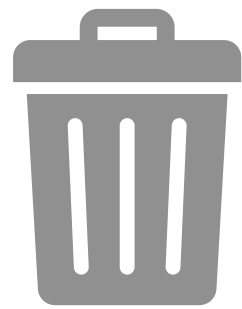
INSERT into ...

async

4

DELETE from ...

目标：最大化赋能DBA、研发



研发提归档工单

业务审批

DBA审批

归档任务

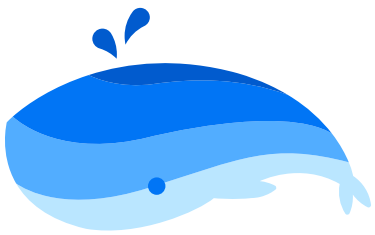
自动授权

查询归档数据



- 全自动流程
- 24小时安全运行
- 研发自助捞归档数据

# 从TiDB 到 Databend



Databend

## 类型转换

- Databend数据类型更少

## 索引处理

- Databend表无主键、二级索引

## NULL

- Databend表字段默认NOT NULL

## 函数

- 不完全与MySQL对应

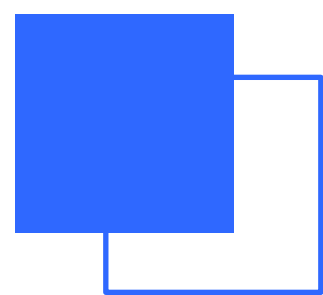
### 类型映射

TINYINT	BIGINT
SMALLINT	
MEDIUMINT	
INT	
BIGINT	

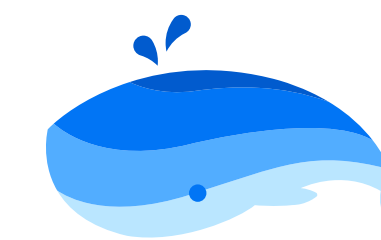
YEAR	STRING
TIME	
DATE	DATE
DATETIME	DATETIME
TIMESTAMP	

FLOAT	STRING
DOUBLE [PRECISION], REAL	
DECIMAL(M,D), NUMERIC(M,D)	
BIT(M)	
CHAR(M)	
BINARY(M)	
VARCHAR(M), VARBINARY(M)	
ENUM('value1','value2',...)	
SET('value1','value2',...)	
BLOB, TEXT	
TINYBLOB, TINYTEXT	
MEDIUMBLOB, MEDIUMTEXT	
LOBLOB, LONGTEXT	
JSON	



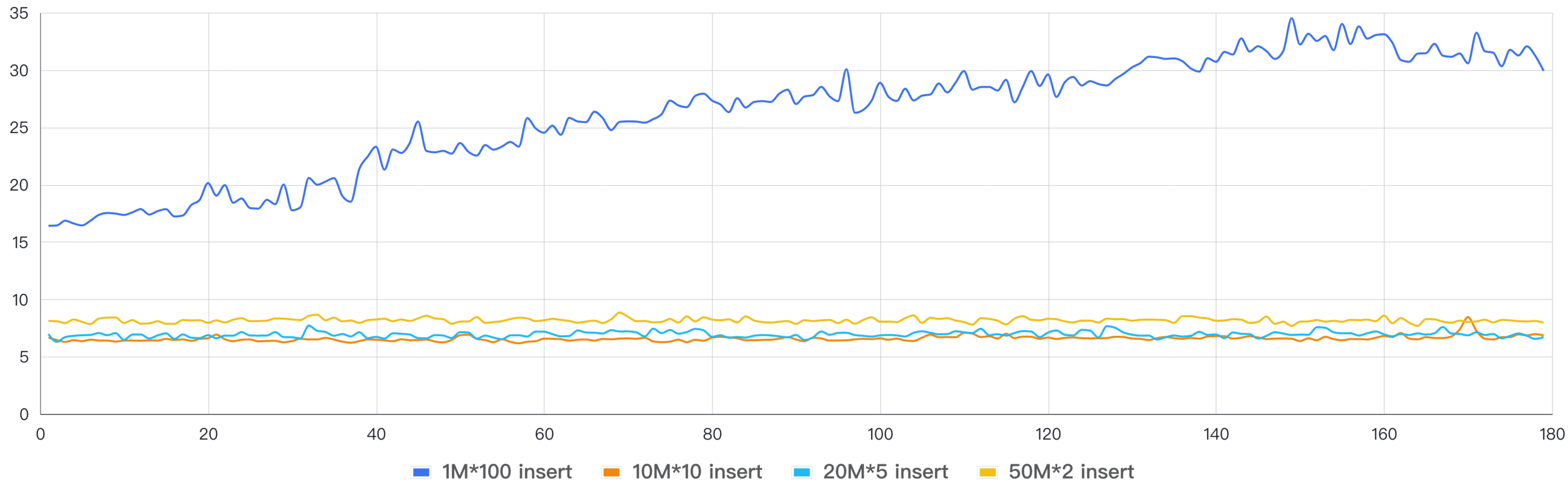


# 实践效果 – 性能



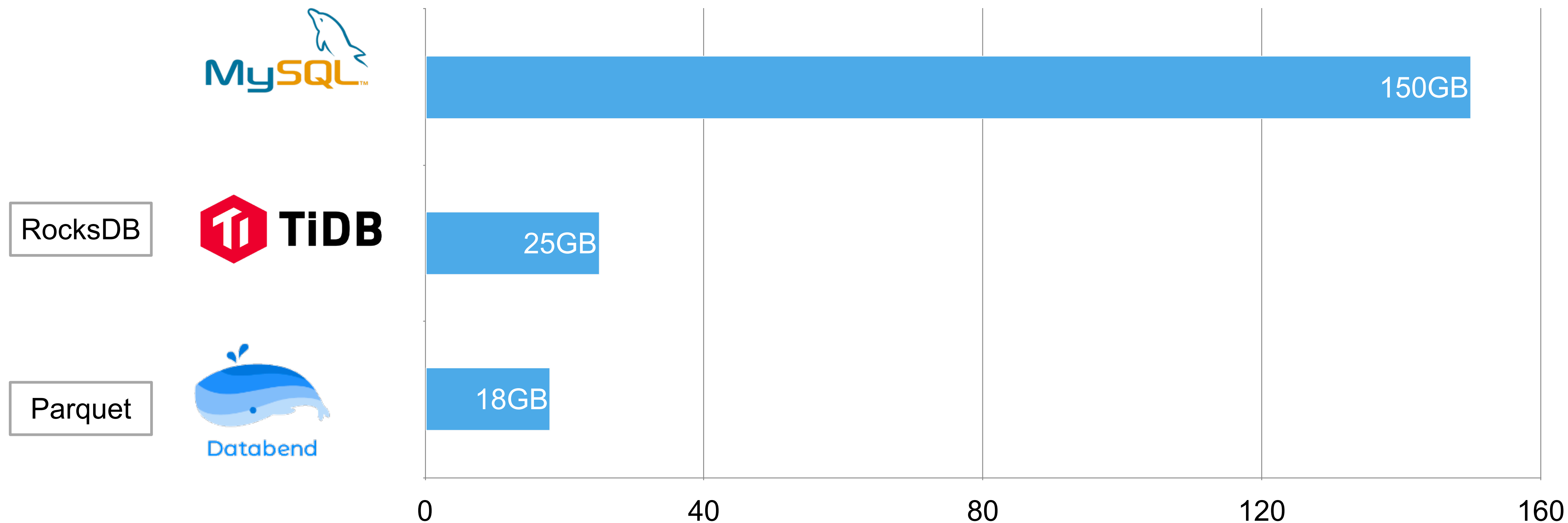
Databend

Databend insert 性能测试 (单个并发, 每写入100MB耗时-秒)

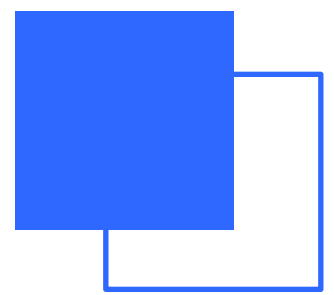


单个并发归档速率: **16MB/s** (**16K rows/s**, 1KB/row) , ~ 1.3TB/day

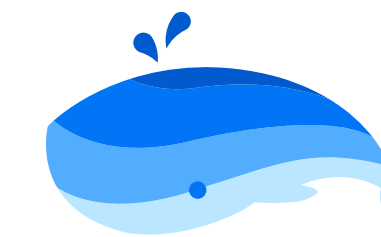
# 实践效果 – 压缩



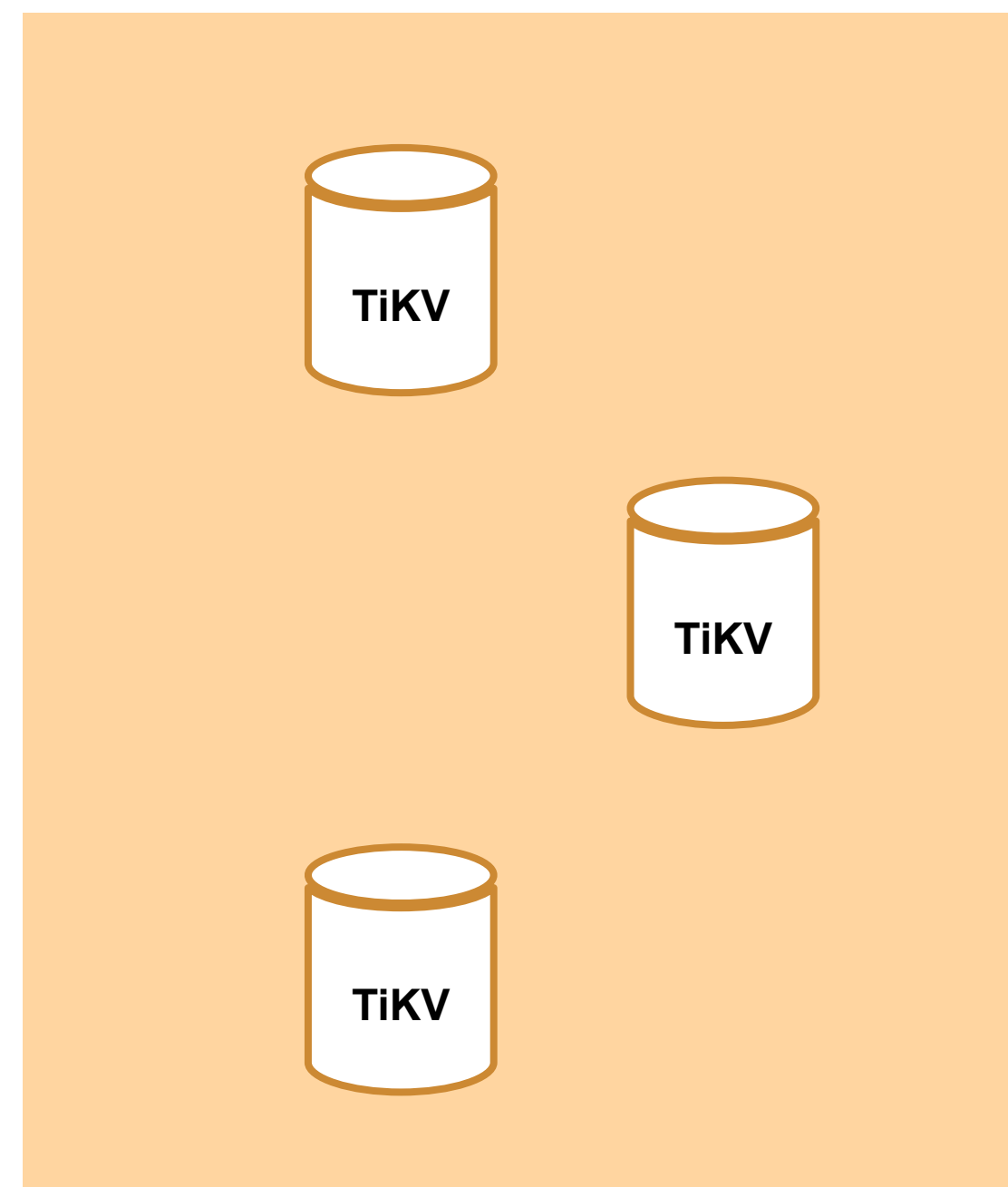
数据仅供参考，结论：TiDB 和 Databend 均有较好的数据压缩效果



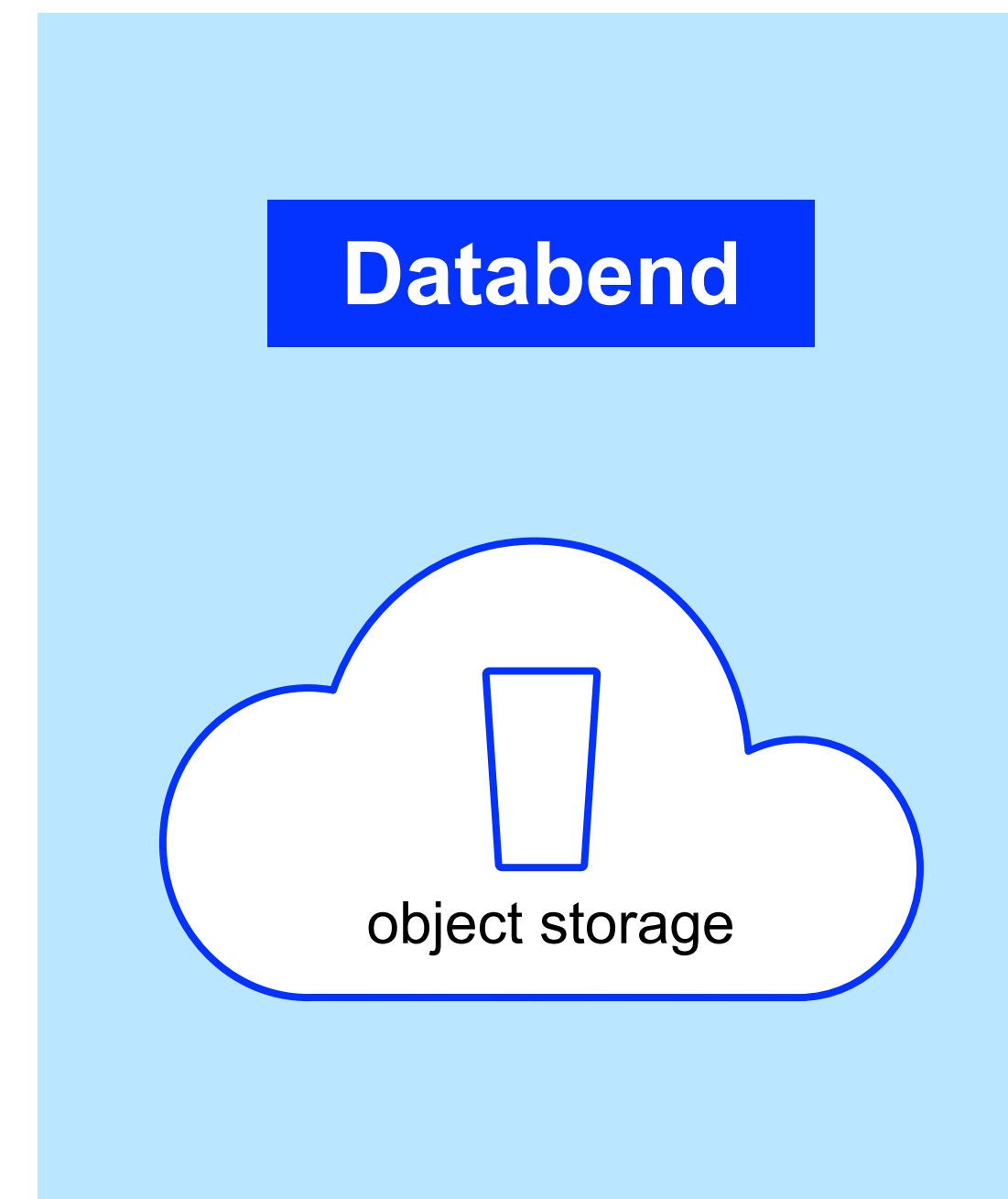
# 实践效果 – 降本98%



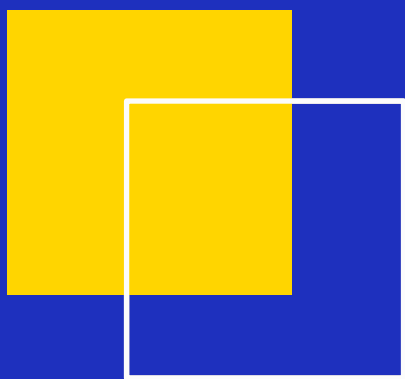
Databend



- 1 3副本 => 1份数据
- 2 不预留，按实际付费
- 3 单价仅 ~ 1/10



数据从 TiKV 归档到 Databend 后，存储成本： $(1/3) * (60%) * (1/10) = 2\%$



# 归档实践总结，未来展望

## 归档场景下 Databend 优势

- 降本显著：基于对象存储，冷数据存储成本降低98%
- 云中立：支持AWS、Azure、GCP、阿里云、腾讯云、华为云、青云、火山引擎
- 研发友好：MySQL协议兼容、可在线查询、统计分析性能好
- 运维无忧：无限空间、高可靠、免维护、迁移便捷

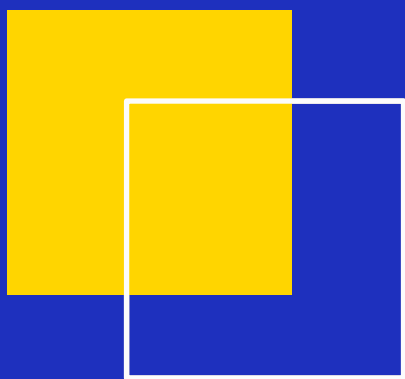


# 对 Databend 未来展望



## 与MySQL 生态，更好地连接

- 兼容性：SQL语法、数据类型、内置function
- 工具：dump / restore 等ETL、甚至准实时DTS/DRC
- 社区：TP + AP、融合、共建



# THANKS!