FLINK FORWARD

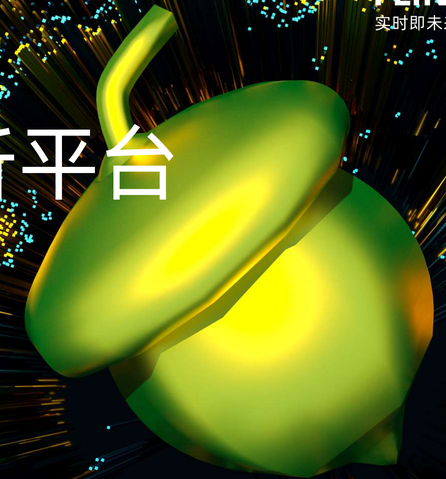
FLINK FORWARD # ASIA
实时即未来 # Real-time Is The Future

# 基于 Apache Flink 的大规模准实时数据分析平台

Apache Flink empowered large-scale near real-time (NRT) data analytics platform

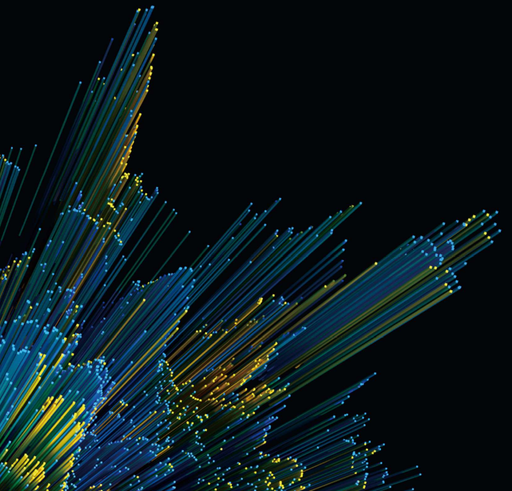

**徐赢 (Ying Xu)**
Streaming Platform, Lyft Inc

**高立 (Li Gao)**
Data Compute Platform, Lyft Inc

# 纲要

Contents



# 01 Lyft 的流数据与场景

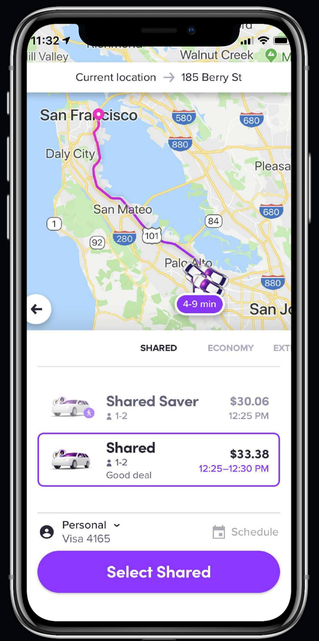
01 Streaming data scenarios at Lyft

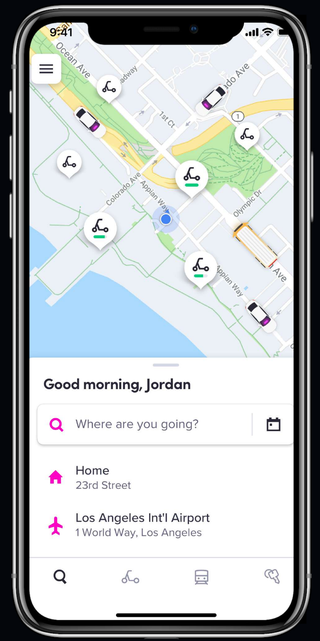
# 关于 Lyft

About Lyft

## 提供世界最好的交通方案来改善人们的生活

Improve people's life with the world's best transportation



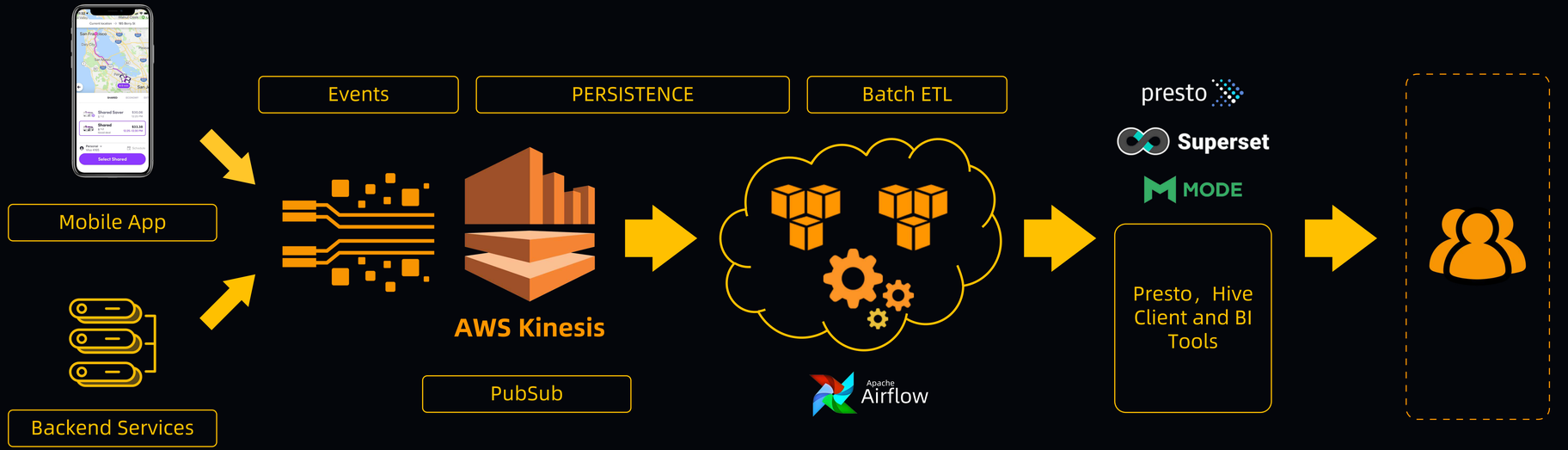
# Lyft 的流数据场景

Streaming data scenarios at Lyft

| 秒级别 | 分钟级别 | | 不高于5分钟 |
|---|---|---|---|
| **流事件处理** | **实时自适应定价** | **欺诈和异常检测** | **机器学习特征工程** | **准实时数据交互式查询** |
| Streaming events enrichment (seconds) | Real-time adaptive pricing (minute) | Fraud and anomaly detection (minute) | ML feature engineering (minute) | Near real-time interactive query (< 5 minutes) |

# Lyft数据分析平台架构

Lyft's data analytics platform architecture



# 既往平台问题

Issues of the legacy platform

导入数据无法满足准实时查询的要求

Persisted data cannot be ready for query in near real-time

基于 KCL 的流式数据导入性能不足

Streaming persistence using KCL with limited performance

导入数据存在太多小文件导致下游操作性能不足

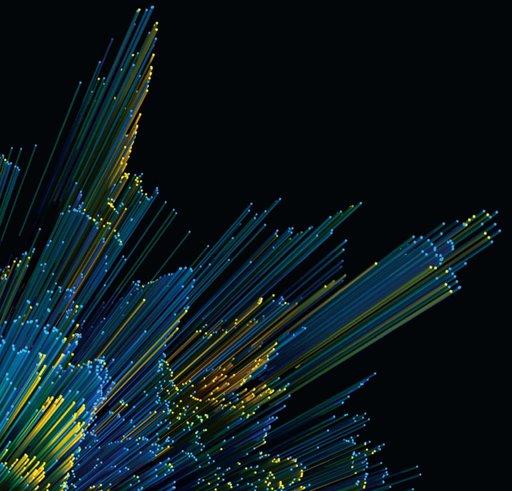
Presence of too many small files limits performance of S3 operations

数据 ETL 大多是高延迟多日多步架构

Most ETL were scheduled on a daily basis and have multi-day latency

平台对于嵌套数据提供支持不足

Legacy platform offers limited support for nested data

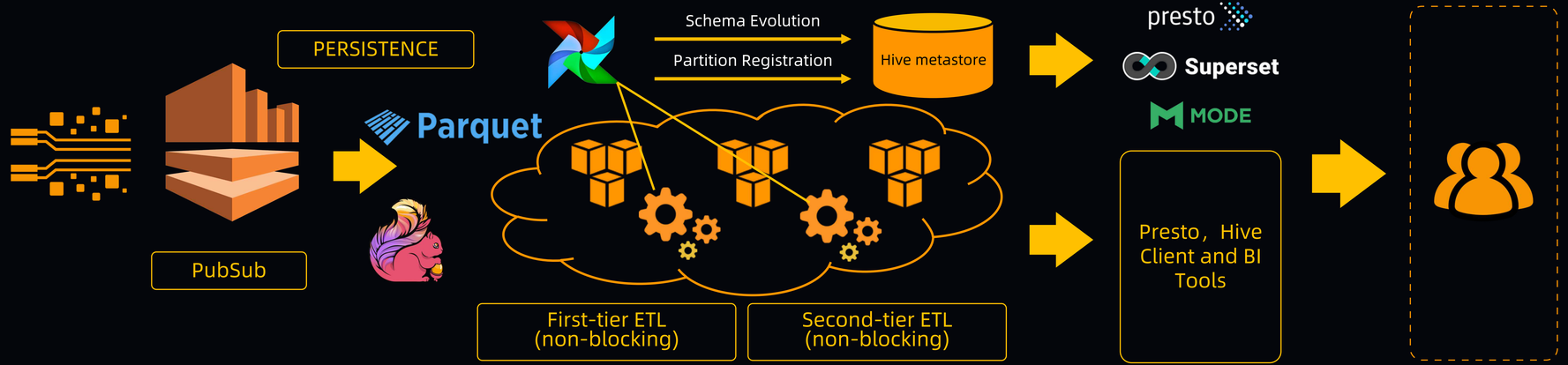


# 02 准实时数据分析平台架构

02 Architecture of near real-time data ingest and analytics platform

# 准实时平台架构

Near real-time data analytics platform architecture



# 平台设计

Platform design

高速有效的流数据接入：Flink

Highly efficient streaming persistence: Flink (StreamingFileSink)

Parquet 格式的数据支持交互式查询

Data persisted in Parquet format supporting interactive query

基于已有 AWS 云端存储 (无需特殊存储形式)

Built on top of existing AWS S3 storage (no specialized format)

多级ETL进程以确保更好性能和数据质量

Multi-stage hourly ETL for enhanced performance and data quality

兼顾性能容错及可演进性

Performance, fault tolerance and evolvability built into the design

# 平台特征及应用

Platform characteristics and use cases

**每天处理千亿级事件**

Hundreds of billions of events per day


**自发的交互式查询**
Ad-hoc interactive queries

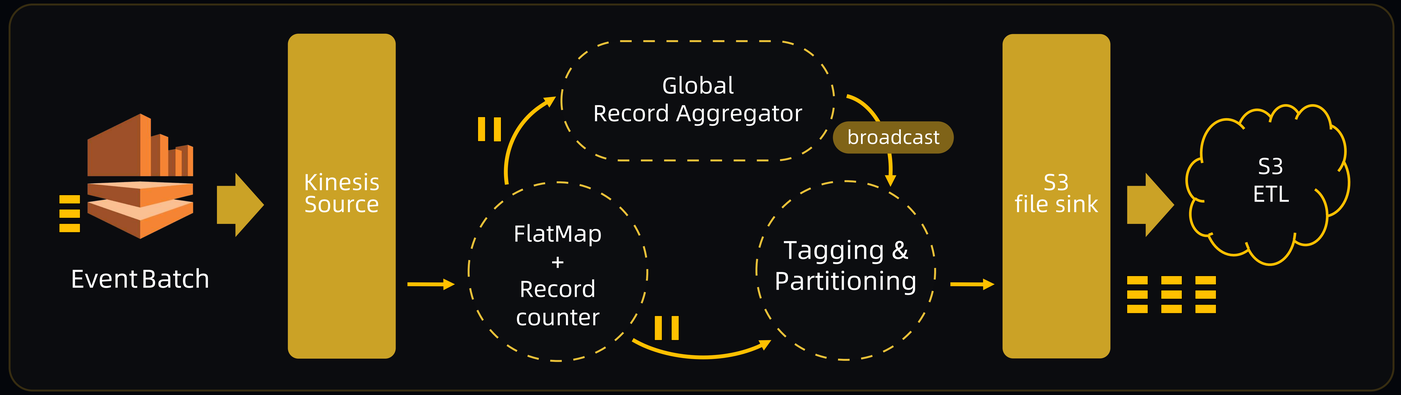
**实时机器学习模型正确性预警**
Real-time alerting on ML model accuracy

**实时 dashboards 监测供需市场健康状况**
Real-time dashboards for marketplace health monitoring

# 基于 Flink 的准实时数据导入

Flink empowered near real-time data ingestion

**Flink 有向图**
Flink DAG




**Flink Kinesis 链接器 ( Lyft 团队贡献的水印功能)**
Flink kinesis source connector (watermark support Contributed by Lyft)

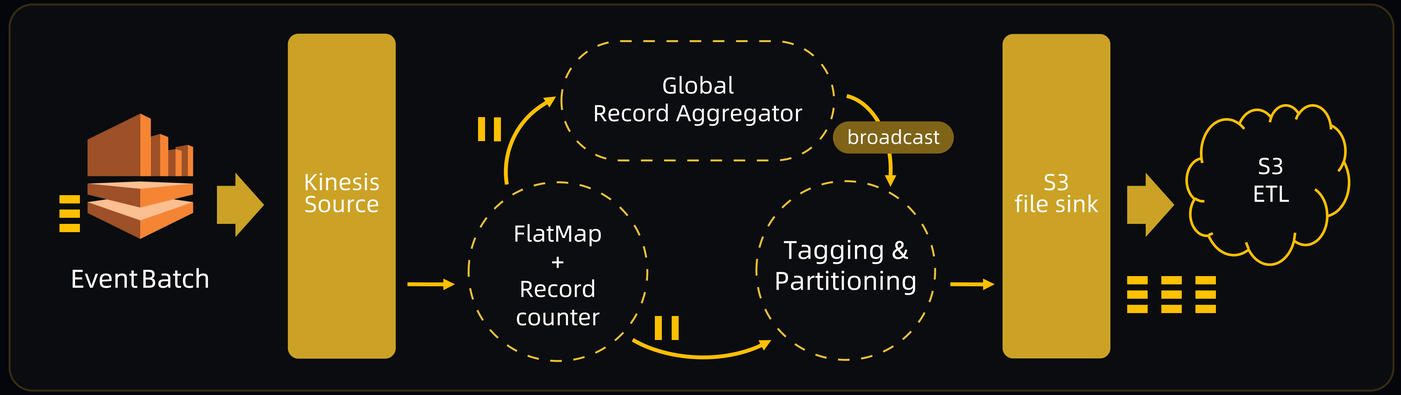
**基于 StreamingFileSink 将批量数据存为 Parquet 格式**
StreamingFileSink unlocks writing bulk-encoded data in Parquet

**每三分钟做一次 Checkpoint 操作**
Checkpoint interval: 3 minutes

# 基于 Flink 的准实时数据导入

Flink empowered near real-time data ingestion

**太多数目的小文件**
Too many small files




**Subtask 记录本地事件权重**
Subtask records local event counts and broadcast periodically

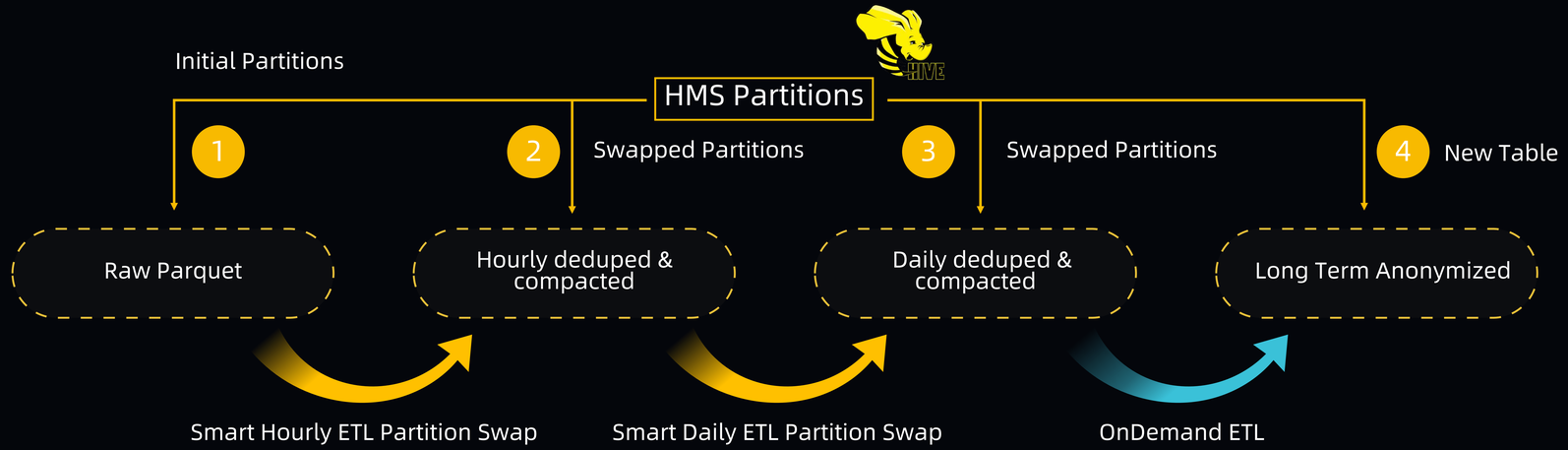
**全局记录聚合器 – 计算全局事件权重并广播**
Global record aggregator - computing global event weights and broadcast

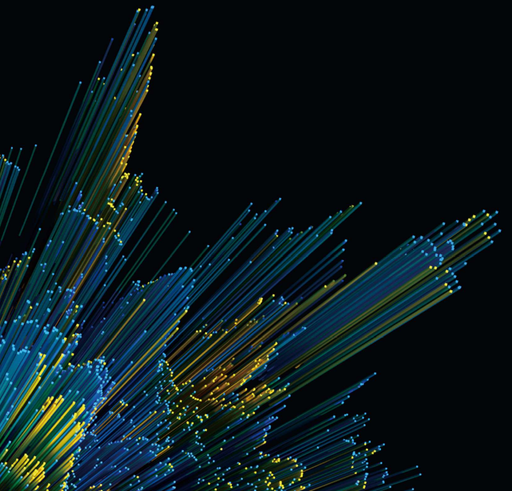
**标记操作 – 连接广播状态并将事件分配给 sink**
Tagging function - connect broadcast state and assign event to sink

(# of subtasks) = event_weight * sink_parallelism

# ETL多级压缩和去重

ETL Multi-tier compaction deduplication





# 03 平台性能及容错深入分析

03 Deep dive on performance and fault tolerance

# 事件时间驱动的分区感测 - Flink

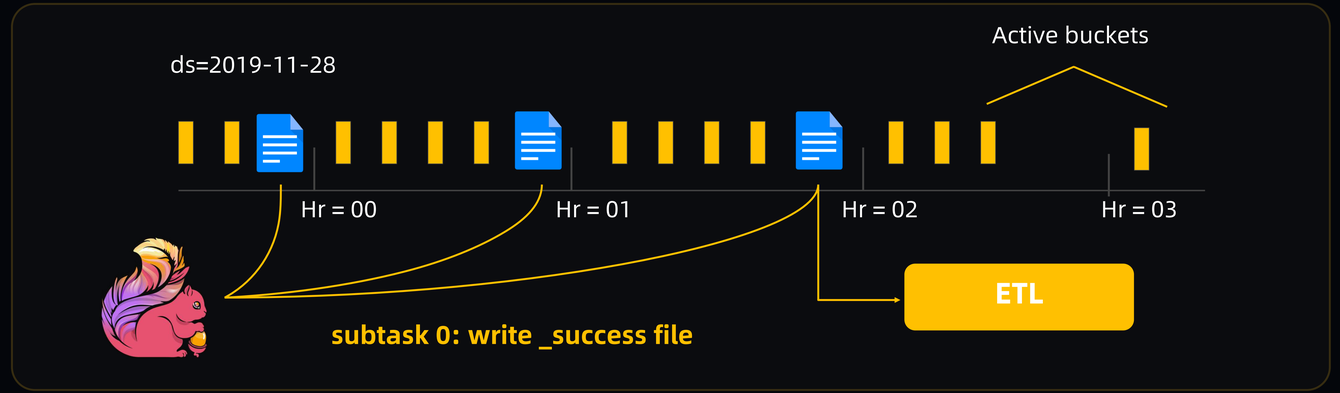
Event-time driven partition sensing - Flink

**S3 分区格式**
S3 partition scheme

**s3://rawevents/$randomPrefix/$eventname/ds=yyyy-MM-dd/hr=HH**

**基于事件时间驱动的分区**
Event-time driven partitioning

**基于 Success 文件的分区感测**
Success file driven partition sensing



# 事件时间驱动的分区感测 - Flink

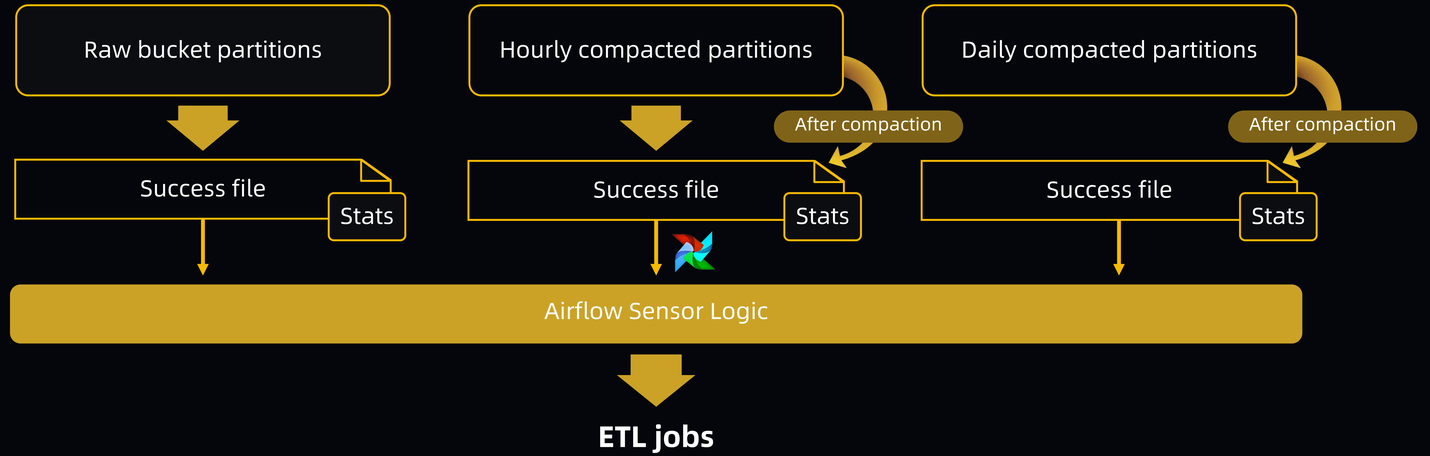
Event-time driven partition sensing - Flink

**构建 Bucket 水印**
Constructing bucket watermark

**Bucket 水印作为存储状态**
Bucket watermark stored as job state

# 事件时间驱动的分区感测 - ETL

Event-time driven partition sensing - ETL



# Schema 演进的挑战

Schema evolution challenges

## 不同引擎的数据类型

Hive v.s. Parquet v.s.
Protobuf v.s. Presto Types

## 嵌套结构的演变

Nested struct Type changes v.s.
Column changes

## 数据类型演变对去重逻辑的影响

Dedupe logic on changed
column types

# S3 深入分析

S3 deep dive

## S3 熵数前缀

S3 entropy
and atomic swap

## 分区标记文件

Partition marker manifest to guide
ETL decisions

## 输入数据的统计直方图

Input data size histogram manifest
to guide ETL decisions

# Parquet 优化方案

Parquet Optimization

**文件数据值大小范围统计信息**

Cardinality Stats & Min/Max Stats

**文件系统统计信息**

File Stats (Success File stats)

**基于主键数据值的排序**

Clustering

**二级索引的生成**

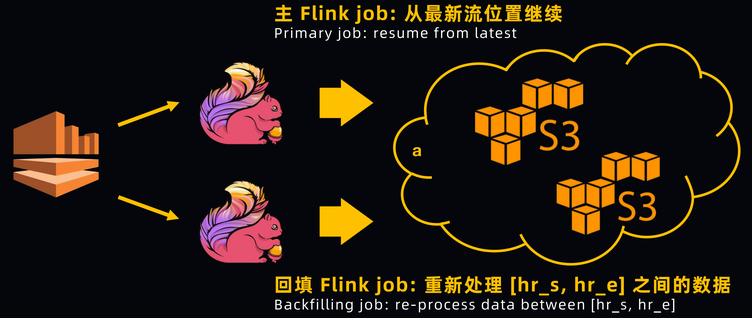
Secondary Indexes

# 基于数据回填的平台容错机制 - Flink

Data backfilling: coping with failures and outages - Flink

**Flink 可以从短暂系统失败中恢复，但是长时间的系统瘫痪呢？**

Flink can recover from short-term system failures. What about long outages ?




**s3://rawevents/xxx/event1/ds=/hr=**

**s3://backfilledevents/xxx/event1/ds=/hr=**

**主 Flink job 和回填 Flink job 并行运行**

Primary and backfilling Flink jobs running in parallel

**基于事件时间驱动的分区：回填进程相对于流处理有幂等性**

Event-time driven partitioning: backfilling process idempotent with stream processing

# 基于数据回填的平台容错机制 - ETL

Data backfilling: coping with failures and outages - ETL

## Airflow 幂等调度系统

Apache Airflow: Idempotent ETL scheduler

## 原子压缩和HMS操作

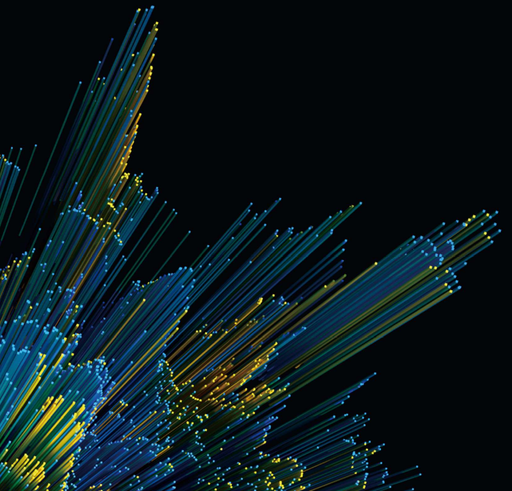
Atomic compaction and HMS operations

## 分区自检自修复体系

Automated metrics to detect partition gaps and data gaps to trigger backfills

## Schema 整合

Schema stitching to hide the complexity of the data backfills and ETL operations



# 04 总结与未来展望

04 Summary and future directions

# 体验与经验教训

Experience and lessons learned

**Flink 准实时注入 Parquet 数据使交互式查询成体验为可能**
Flink persisting Parquet data in near real-time unlocks interactive query experiences

**Flink 的重启和部署可能影响延时 SLO**
Flink full restart or job deployment could affect SLO

**将 Flink 停滞 subtask 对延迟的影响最小化**
Minimizing the impact of stalled Flink subtask on latency

**ETL 分区感应降低成本和延迟**
ETL Partition Sensing and Compaction

**S3 文件布局对性能的提升**
S3 file layout is critical to consistency and performance

**Schema 兼容性至关重要**
Backward compatible schema change is critical to data quality

# 未来展望

Future directions

**Flink 在 Kubernetes 环境下运行**
Flink job operating in k8s environment

**ETL 智能压缩及事件驱动 ETL**
ETL smart compaction and event-driven ETL

**通用的流数据导入框架**
Generalized streaming persistence framework

**存储管理的改进以及查询优化**
Storage management improvement and query optimization

# 团队工作

Team Work

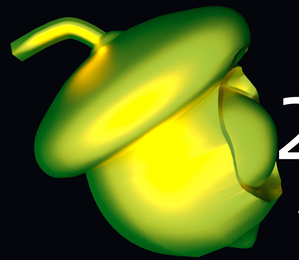
成员 (字母顺序)
Members (Alphabetic order)

Jason Carey，Beto Dealmeida，Li Gao，Mark Grover，Kailash HD，Yash Kumaraswamy，Dev Tagare，Sherin Thomas，Chris Williams and Ying Xu

致谢
Acknowledgement

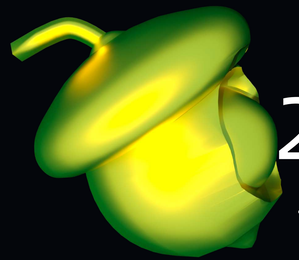
Jamie Grier，Thomas Weise，Bill Graham，James Taylor，Arup Malakar and Lakshmi Rao

# Flink Forward Asia 2019



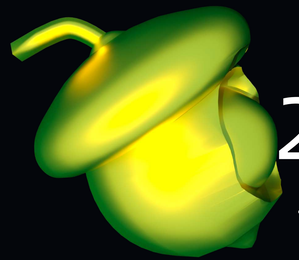
Thanks!

# Flink Forward Asia 2019



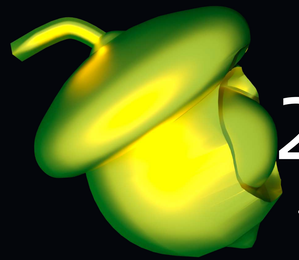
Thanks!

# Flink Forward Asia 2019



Thanks!

# Flink Forward Asia 2019



Thanks!