Apache Flink 社区最新动向及 1.9 版本的功能展望

伍翀（云邪）

GTLC
GLOBAL
TECH LEADERSHIP
CONFERENCE

全球技术领导力峰会

Geekbang. 极客邦科技 | TGO 鲲鹏會

500+高端科技领导者与你一起探讨
技术、管理与商业那些事儿

2019年6月14–15日 | 上海圣诺亚皇冠假日酒店

扫码了解更多信息

自我介绍

- 伍翀（云邪，Jark）

- Apache Flink Committer
  - 自 Flink v1.0 开始在社区贡献
  - 专注工作于 Flink Table & SQL 已有3年

- 阿里巴巴 Blink SQL 的开发与优化

目录

- Flink 当前架构与问题
- Flink 未来架构与解决方案
- Flink 1.9 新特性预览
- Flink 社区最新动态总结

背景

- Apache Flink: 流批统一的新一代大数据引擎
- Alibaba Blink: 阿里巴巴基于 Apache Flink 打造的企业级计算引擎
- Blink 开源：2019-01-28 《阿里正式向 Apache Flink 贡献 Blink 源码》
- 合并 Blink 计划：2019-02-13
  《Batch as a Special Case of Streaming and Alibaba's contribution of Blink》

本演讲所涉及的部分内容还在讨论和设计阶段，并不代表最终呈现的样子。

当前架构

Flink 当前架构

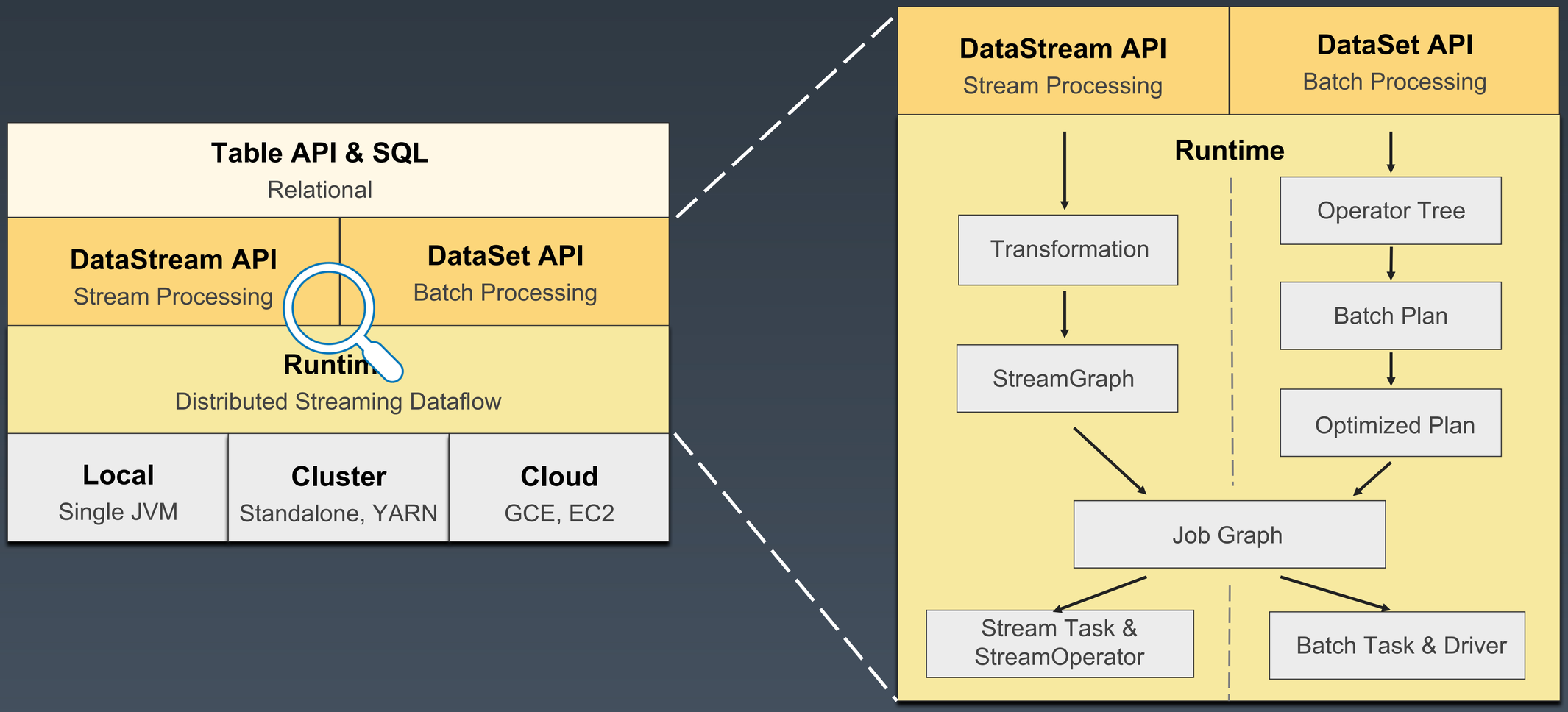


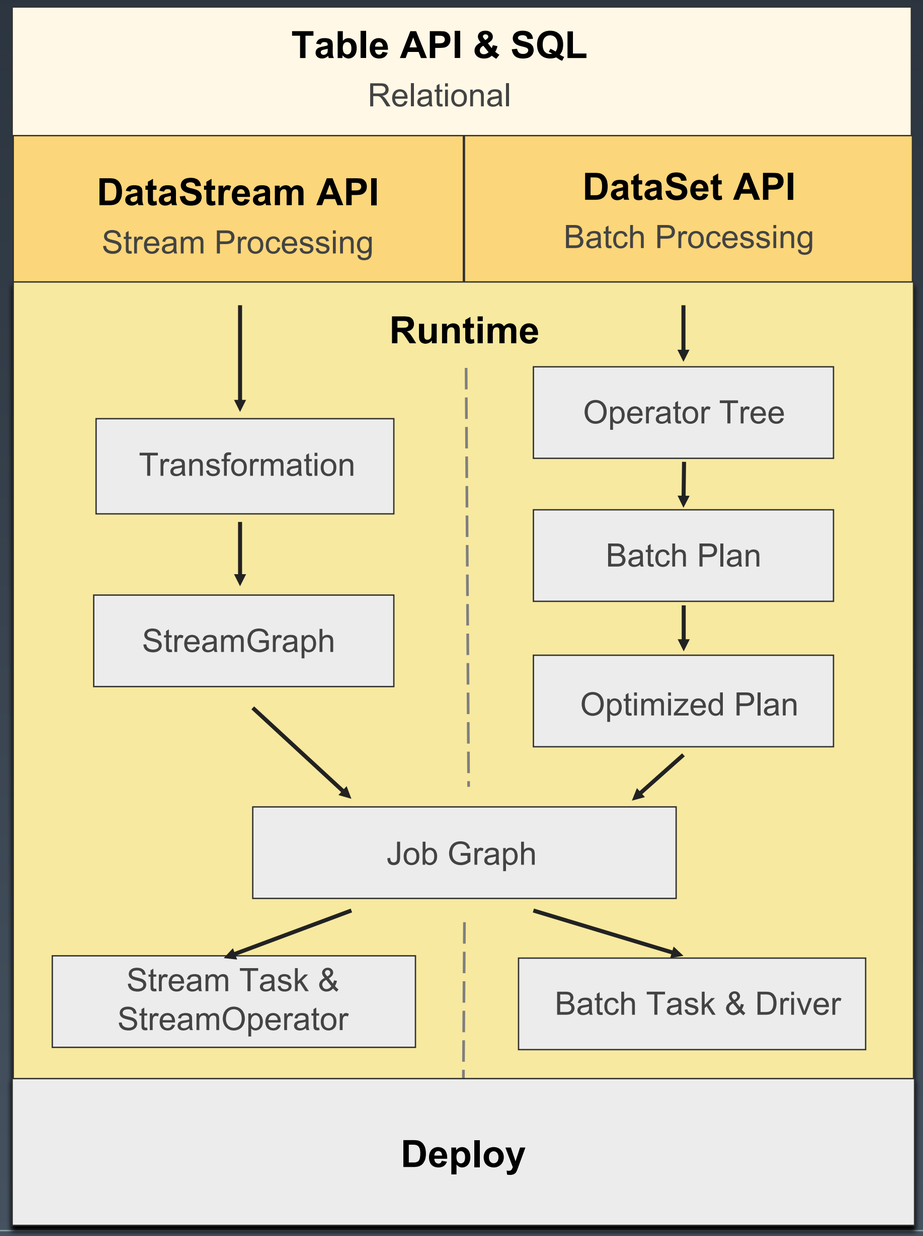
Table API & SQL
Relational
DataStream API
Stream Processing
DataSet API
Batch Processing
Runtime
Distributed Streaming Dataflow
Local
Single JVM
Cluster
Standalone, YARN
Cloud
GCE, EC2
DataStream API
Stream Processing
DataSet API
Batch Processing
Runtime
Transformation
StreamGraph
Operator Tree
Batch Plan
Optimized Plan
Job Graph
Stream Task & StreamOperator
Batch Task & Driver

存在的问题？

从架构设计、代码质量、开发的角度来看

- 不同的 DAG 表示形式和翻译路径
- 不同的算子实现：StreamOperator, Driver
- 不同的 Task 执行：StreamTask，BatchTask
- DataSet 有自己的小型优化器与 SQL 优化器打架
- 基于 DataSet 开发的语义很难和标准 SQL 保持一致
- 算子在流批之间无法共享
- 两套完全独立的 connector 集合
- 潜在问题：两条独立的技术栈 -> 需要更多的人力

  -> 功能开发变慢、性能提升变难，bug变多



未来架构

批是流的一个特例，我们是否可以...?

DataStream API
Stream Processing

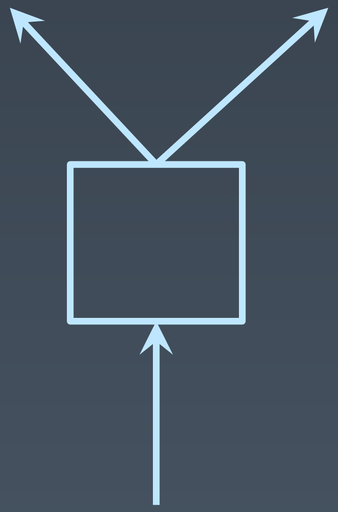
DataSet API
Batch Processing

完成！

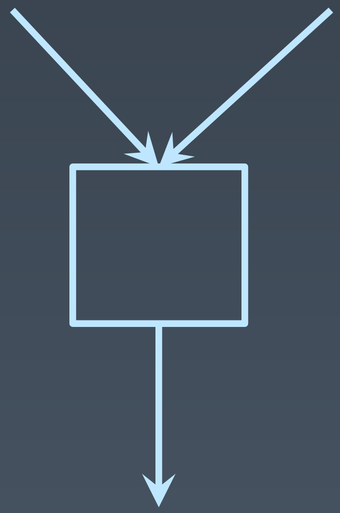
Runtime 改动

- JobGraph 需要加强，携带上有界性等信息
- FLINK-11875：基于 push 模型的可选边的 Operator

Batch: pull-based (Driver)



Stream: push-based (StreamOperator)

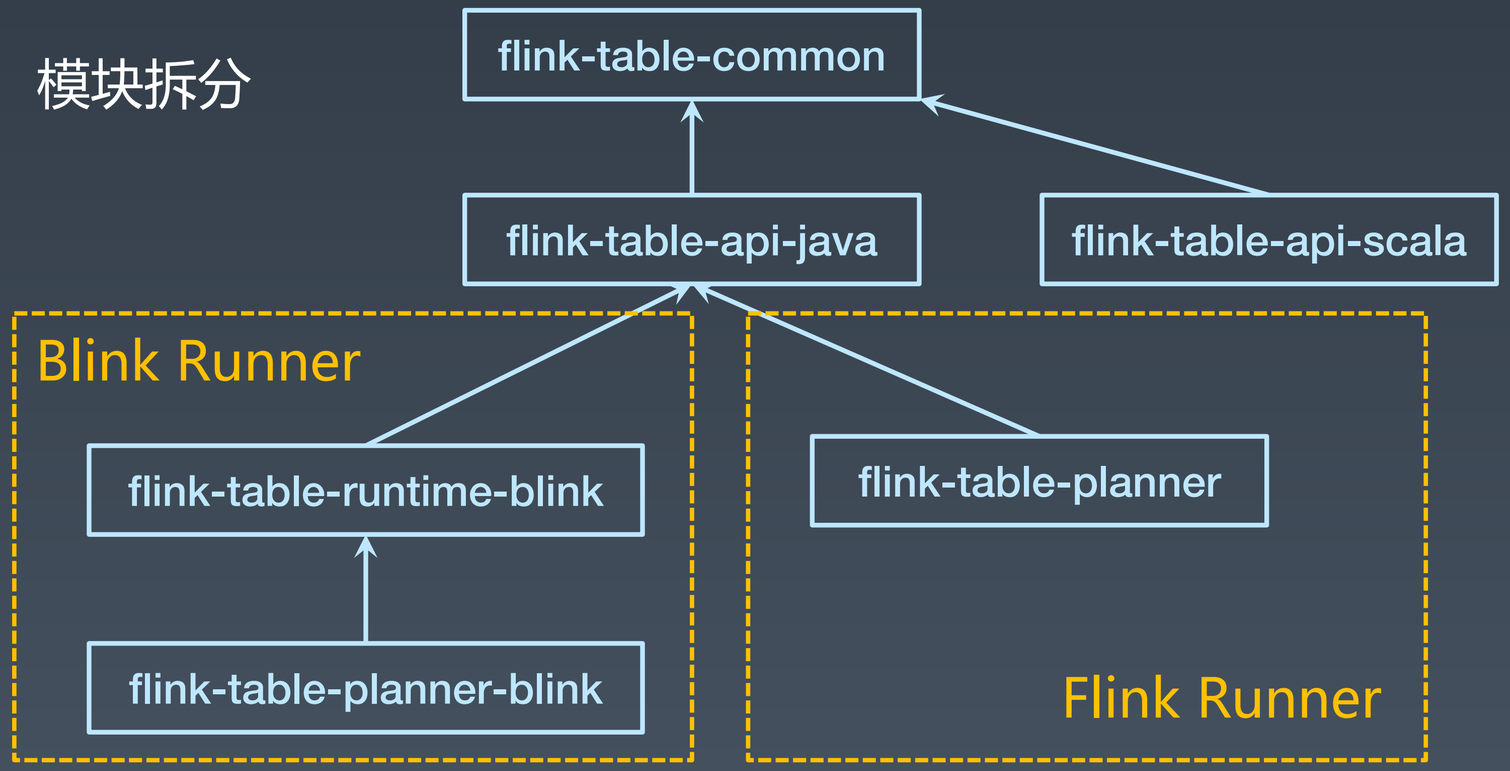


Runtime 改动

- JobGraph 需要加强，节点携带上有界性等信息
- FLINK-11875：基于 push 模型的可选边的 Operator
- FLINK-10429：插件化调度机制
- FLINK-10288：高效的批处理作业恢复
- FLINK-10653：插件化 Shuffle Service

Table API & SQL 改动

- FLINK-11439：INSERT INTO flink_sql SELECT * FROM blink_sql



Flink 新架构

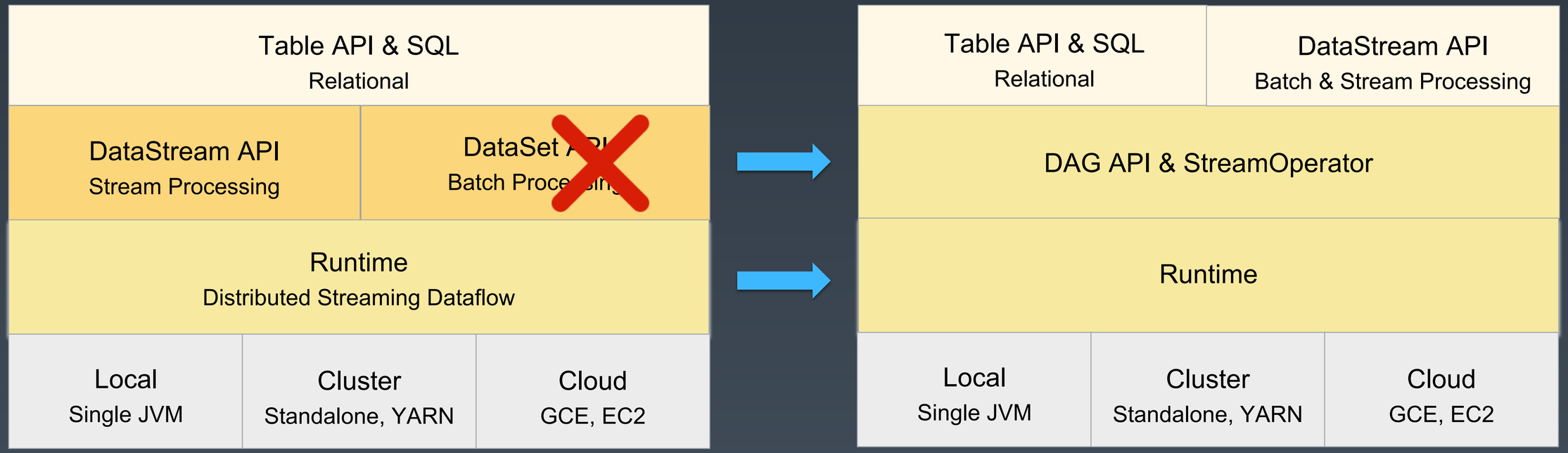


Table API & SQL
Relational
DataStream API
Stream Processing
DataSet API
Batch Processing
Runtime
Distributed Streaming Dataflow
Local
Single JVM
Cluster
Standalone, YARN
Cloud
GCE, EC2
Table API & SQL
Relational
DataStream API
Batch & Stream Processing
DAG API & StreamOperator
Runtime
Local
Single JVM
Cluster
Standalone, YARN
Cloud
GCE, EC2

Flink SQL 1.9 新特性预览

Flink SQL 1.9 新特性预览

BinaryRow

DDL

维表 Join

TopN

大量性能优化

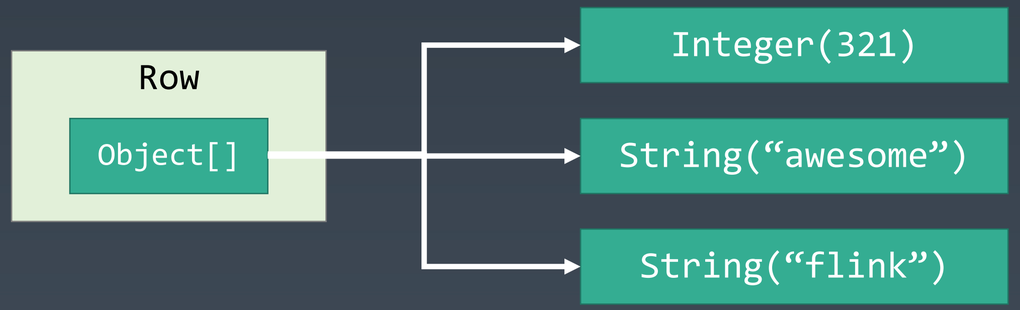
完整高效的批处理支持

Hive 集成

改进的基础数据结构—— BinaryRow

旧数据结构: Row

- Java 对象的空间开销高
- 主类型的装箱和拆箱开销
- 昂贵的 hashCode() 和 (反)序列化

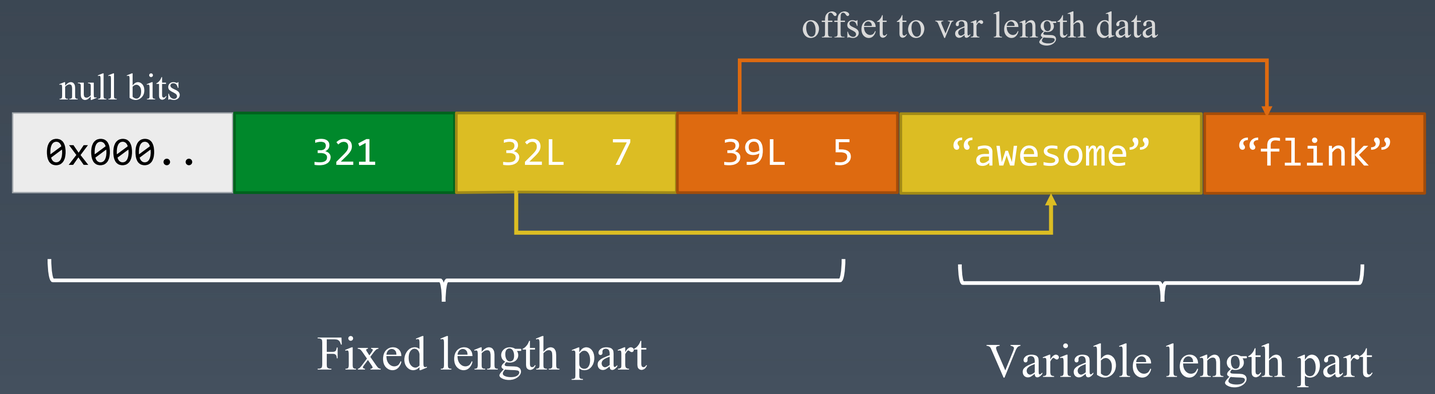


改进的基础数据结构—— BinaryRow

新数据结构: BinaryRow

- 避免了很多反序列化开销
- 与内存管理紧密结合
- CPU 缓存友好

不仅在批处理中表现出色，
在流处理中也收获了一倍的提升



Unified SQL DDL （preview）

```
CREATE TABLE kafka_orders (
  order_id VARCHAR,
  product VARCHAR,
  amount BIGINT,
  order_ts TIMESTAMP,
  proctime AS PROCTIME(),
  WATERMARK FOR order_ts AS BOUNDED WITH DELAY '10' SECOND
) WITH (
  connector='kafka',
  kafka.topic='orders',
  kafka.zookeeper.connect='localhost:2181',
  kafka.bootstrap.servers='localhost:9092',
  kafka.group.id='testGroup',
  kafka.startup-offset='earliest',
  kafka.end-offset='none',
  ...
);

SELECT product, TUMBLE_START(order_ts, INTERVAL '1' MINUTE), COUNT(*)
FROM kafka_orders
GROUP BY product, TUMBLE(order_ts, INTERVAL '1' MINUTE);
```

定义了schema

定义了一个计算列

定义了watermark

定义了表的属性，包括存储类型，
连接信息，读取的范围，有界性

流处理

Unified SQL DDL （preview）

```
CREATE TABLE kafka_orders (
  order_id VARCHAR,
  product VARCHAR,
  amount BIGINT,
  order_ts TIMESTAMP,
  proctime AS PROCTIME(),
  WATERMARK FOR order_ts AS BOUNDED WITH DELAY '10' SECOND
) WITH (
  connector='kafka',
  kafka.topic='orders',
  kafka.zookeeper.connect='localhost:2181',
  kafka.bootstrap.servers='localhost:9092',
  kafka.group.id='testGroup',
  kafka.startup-offset='earliest',
  kafka.end-offset='2019-05-28 00:00:00',
  ...
);
```

批处理

```
SELECT product, TUMBLE_START(order_ts, INTERVAL '1' MINUTE), COUNT(*)
FROM kafka_orders
GROUP BY product, TUMBLE(order_ts, INTERVAL '1' MINUTE);
```

维表 JOIN （preview）

```
CREATE TABLE mysql_products (
  product_id VARCHAR,
  product_name VARCHAR,
  price DECIMAL,
  PRIMARY KEY (productId)
) WITH (
  connector = 'mysql'
  …
);


SELECT o.*, p.*
FROM kafka_orders AS o
JOIN mysql_products FOR SYSTEM_TIME AS OF o.proctime AS p
ON o.product_id = p.product_id
```

TopN

```
SELECT *
FROM (
  SELECT *,
    ROW_NUMBER() OVER
      (PARTITION BY category
       ORDER BY sales DESC) AS rownum
  FROM shop_sales)
WHERE rownum <= 3
```

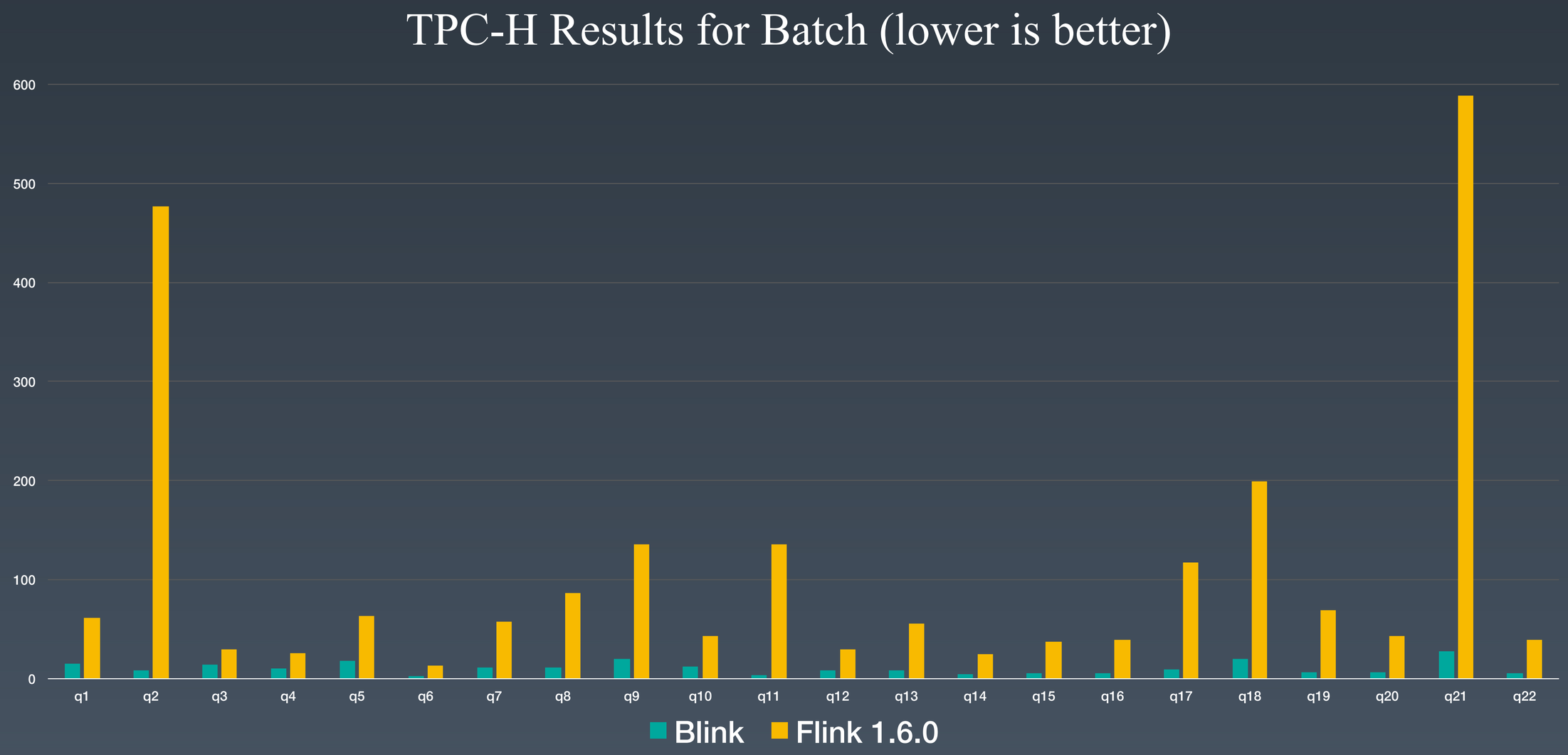
result			
category	**shopId**	**sales**	**rownum**
book	shop-43	89	1
book	shop-46	56	2
book	shop-58	43	3
fruit	shop-12	78	1
fruit	shop-44	67	2
fruit	shop-32	57	3
…	…	…	…

- 支持分组 TopN
- 针对细分场景，3种不同实现，优化器自动选择

大量流处理性能优化

- MiniBatch
- Local 聚合
- Distinct Agg 自动热点打散
- Distinct State 共享
- 细分场景，特定算子实现
- 100+ 优化规则

完整批处理能力支持和性能提升



Hive 集成

- 统一的 Catalog 接口
- 提供基于内存和可持久化的 Catalog 实现
- 提供 Hive Catalog，支持与 Hive 的互操作
- 支持在 Flink 中运行 Hive UDF

Flink 社区最新动态

Flink 社区最新动态

- 计划于 7 月份发布 Flink 1.9
- SQL:
  - FLIP-32: 重构 Table 模块，使其同时支持多个 Runner
  - FLINK-11439: Merge Blink 分支的大部分 SQL 功能 （进度90%）
  - FLIP-29: 增强 Table API 功能
- Runtime:
  - FLINK-11875：基于 push 模型的可选边的 Operator
  - FLINK-10429：插件化调度机制
  - FLINK-10288：高效的批处理作业恢复
  - FLINK-10653：插件化 Shuffle Service
- 生态:
  - FLIP-30: 插件化 Catalog，支持 Hive Meta Store
  - FLIP-38: Python Table API
  - FLIP-39: 基于 Table API 实现一套全新的 ML Pipeline

总结

总结

- Flink 1.9 将是具有里程碑意义的一个版本
- Flink 有史以来改动最大的一个版本，所有模块都在迎接变化
- 改造之后，Flink 将具备比较完善流批一体的技术架构
- 用户将有一个较好的流批统一的体验
- 希望能有更多人加入到社区一起努力

极客时间 | 企业服务

这是一封正经“福利”贴

想做团队的领跑者 需要迈过这些“槛”

成长型企业，易忽视人才体系化培养 企业转型加快，团队能力又跟不上	VS	从基础到进阶，超100+一线实战 技术专家带你系统化学习成长
团队成员技能水平不一， 难以一“敌”百人需求	VS	解决从小白到资深技术人所遇到 80%的问题
寻求外部培训，奈何价更高且 集中式学习	VS	多样、灵活的学习方式，包括 音频、图文 和视频
学习效果难以统计，产生不良循环	VS	获取员工学习报告，查看学习 进度，形成闭环

课程顾问「橘子」

回复「QCon」
免费获取
学习解决方案

 # 极客时间企业账号 # 解决技术人成长路上的学习问题

THANKS!

QCon 10th