FLINK FORWARD

The Apache Flink® Conference

Stream Processing | Event Driven | Real Time

San Francisco 1-2, 2019

A big thanks to our Sponsors

Platinum	Google Cloud, Alibaba.com
Gold	theTradeDesk, aws, lyft, DELL EMC
Silver	yelp, COGILITY, eventador.io
Flink Fest	COMCAST
Community	APACHE, womenhack, Data Council, WOMEN WHO CODE, BERLIN BUZZWORDS 2019 JUNE 16-18
Media	EVENTBROWSE

A big thanks to our Program Committee

Tyler Akidau

Stefan Richter

Eric Sammer

Sonali Sharma

Jamie Grier

Fabian Hueske

Dean Wampler

A big thanks to our Speakers

Get involved

Flink Forward App

Rate speakers and sessions with a chance to win a Flybrix drone!

Apache Flink User Survey

Win a trip to one of the next Flink Forward conferences of your choice!

Flink Forward Survey

Help us improve the quality of Flink Forward. We appreciate your feedback!

Community Contribution

Sign up as a content contributor for blog posts or speaking opportunities.

Flink Forward Social Feed - View & Engage!

FLINK FORWARD

INTRODUCING VERVERICA

Kostas Tzoumas

dataArtisans

Founded in 2014 by the original creators of Apache Flink to commercialize the open source project and support the community

dataArtisans + Alibaba Group

= ververica

Why?

- Alibaba has been the largest user of Flink and second largest contributor for years
- Deeply committed to open source and creating technological impact
- Joining forces made a lot of sense for the two teams in order to collaborate even closer and accelerate their contributions to Flink

Flink at Alibaba (few examples)

淘宝网
Taobao.com

Taobao is the largest e-commerce platform globally with more than **600 million monthly active users**. Every time a user logs into the Taobao app they see a different landing page personalized for the user and depending on the latest real-time activity in the platform. Using Flink for real-time machine learning at Taobao has resulted in over **20% increase in purchase conversion rate**. At peak during Singles Day last year, the system processed over **1.7 billion events/sec**.

In the Hangzhou City Brain Project, Flink is used to process in real-time data from a variety of sensors (traffic cameras, map applications, etc), and manage traffic signals in 128 intersections. The City Brain project has **halved traveling times** for ambulances and commuters. Traffic accidents can be detected immediately, and help can reach the accident site within 5 minutes.

What's in a name?

verum ("real" in Latin)

Understanding the truth about the world by getting the real-time view

What is Ververica?

Our #1 goal is to position Apache Flink for the next 10 years of its life

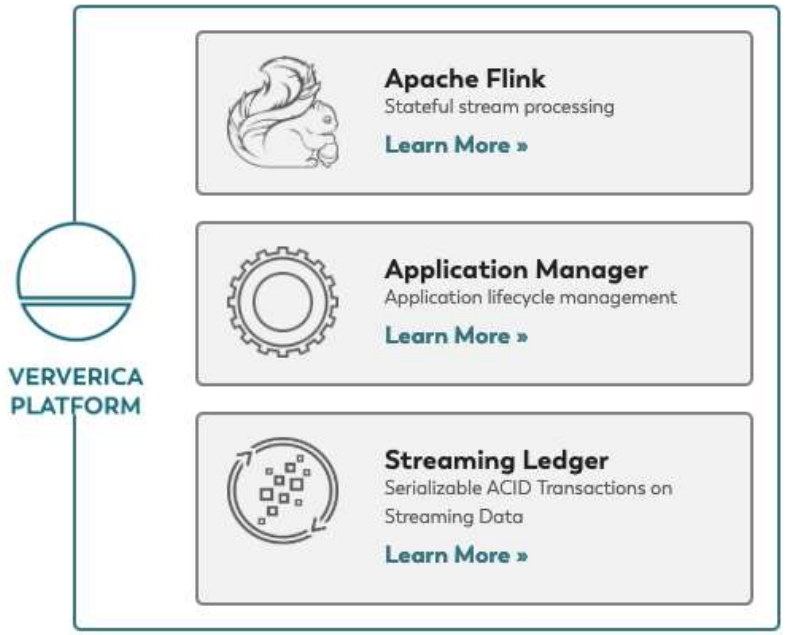
1. Double down on the open source community and improve its health and diversity
2. Contribute a number of innovations to the open source project starting with Alibaba's Blink for batch processing
3. Create an ecosystem and foundation for the commercial success of Flink projects and products across the world

Ververica Commercial Products

Full continuation of our commercial products and services

- Ververica Platform including Apache Flink, Application Manager, and Streaming Ledger
- Apache Flink Training and Consulting Services
- Enterprise Support

A lot of innovation coming here as well leveraging existing work in Alibaba Cloud



Announcing: Ververica Partner Program

We are looking for partners to help us develop the broader Flink ecosystem

- **Ververica Platform Partner**
  Preferred partners of our commercial products around the globe

- **Ververica Services Partner**
  Service provider on Apache Flink certified by Ververica

Sign up here! ververica.com/partner-program

From Stream Processor to Unified Data Processing System

Stephan Ewen
Xiaowei Jiang
Robert Metzger

ververica

Use Cases Presented Today

airbnb
Alibaba Group
ARCTIC WOLF
branch
COGILITY
COMCAST

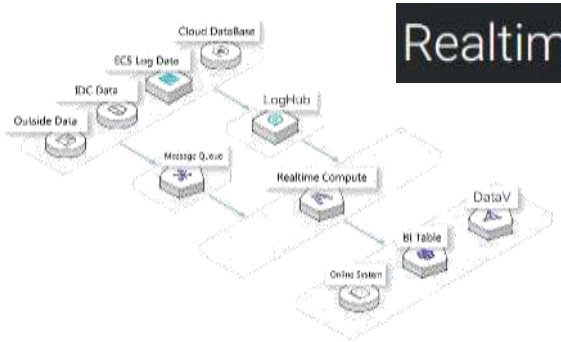
Google
imply
INSTRUCTURE
intuit
JOHN DEERE
KLAVIYO

lyft
NETFLIX
splunk>
theTradeDesk
Uber
yelp

Tencent 腾讯

Apache Flink and Public Clouds

Alibaba Cloud



Realtime Compute

Google Cloud

beam → cloud

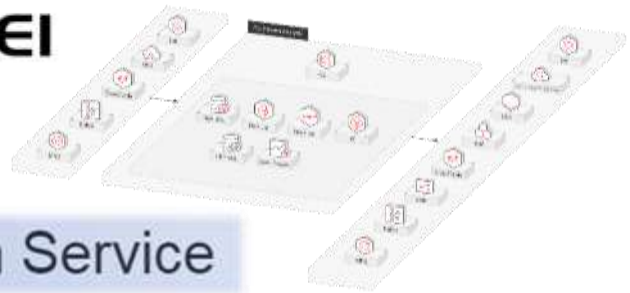
beam → on-prem

amazon web services

Amazon Kinesis Data Analytics

Query and analyze streaming data

HUAWEI



Cloud Stream Service

Data Processing Applications

Stream Processing

Batch Processing	Continuous Processing	Data Pipelines	Streaming Analytics	Event-driven Applications	Transactional Processing

more lag time ⟷ more real time

Stream Processing

Batch Processing	Continuous Processing	Data Pipelines	Streaming Analytics	Event-driven Applications	Transactional Processing

more lag time ⟷ more real time

Batch Processing	&	Continuous Streaming
Analytics	&	Applications

Stream Processing

Flink community's focus over the last releases

Recent Features

Batch Processing | Continuous Processing | Data Pipelines | Streaming Analytics | Event-driven Applications | Transactional Processing

more lag time ⟷ more real time

```
SELECT
    o.time AS time,
    o.price * r.rate AS price
FROM
    Orders AS o,
    LATERAL TABLE (Rates(o.time)) AS r
WHERE r.crcy = o.crcy
```

Orders

time	price	crcy
10:15	2	EUR
10:30	1	USD
10:32	50	YEN
10:52	3	EUR
11:04	5	USD

⋈

Rates

time	crcy	rate
09:00	USD	102
09:00	EUR	114
09:00	YEN	1
10:45	EUR	116
11:00	USD	105

=

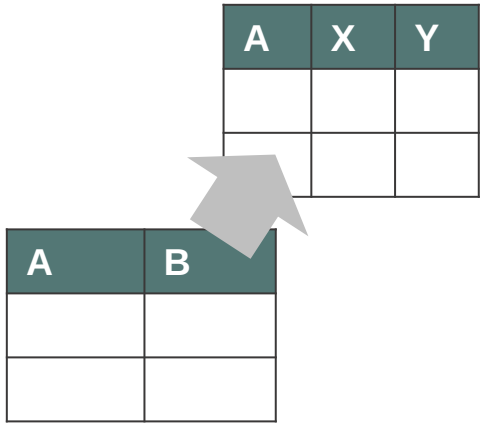
Result

time	price
10:15	228
10:30	102
10:32	50
10:52	348
11:04	525

Time-versioned Joins

```
SELECT *
FROM TaxiRides
MATCH_RECOGNIZE (
   PARTITION BY driverId
   ORDER BY rideTime
   MEASURES
       S.rideId as sRideId
   AFTER MATCH SKIP PAST LAST ROW
   PATTERN (S M{2,} E)
   DEFINE
       S AS S.isStart = true,
       M AS M.rideId <> S.rideId,
       E AS E.isStart = false
           AND E.rideId = S.rideId)
```

MATCH_RECOGNIZE



Schema Upgrades

Stream Processing

"Steam Processing takes on ACID"
by Seth Wiesman

11am, Nikko I

Stream Processing

Batch Performance
Batch Fault Tolerance

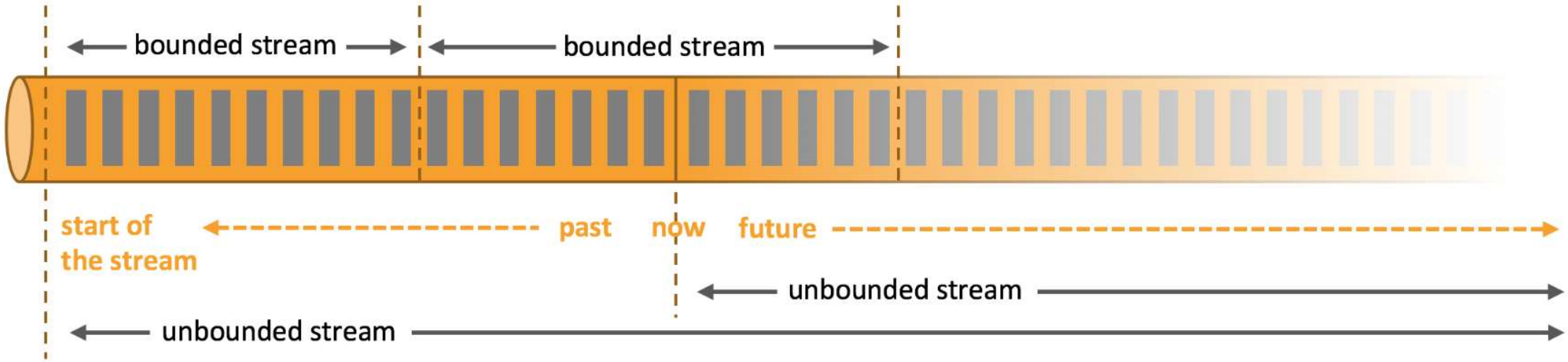
Machine Learning
Graphs

SQL and SQL Ecosystem / tools

Interactive Queries
Dashboards

The Relationship between Batch and Streaming

Everything Streams



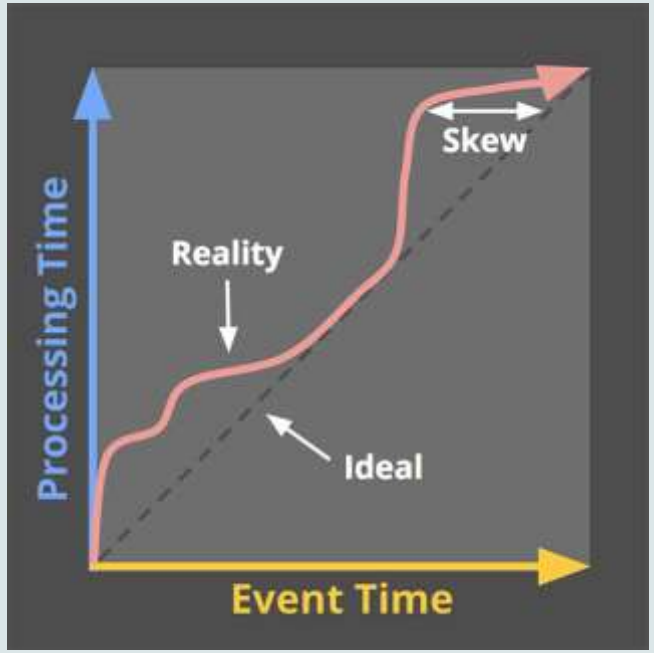
That is about 60% of the truth…

The remaining 40% of the truth

Continuous Streaming	Batch Processing
Data is incomplete	Data is as complete as it gets within the job
Latency SLAs	No Low Latency SLAs
Completeness and Latency is a tradeoff	

The remaining 40% of the truth

Continuous Streaming

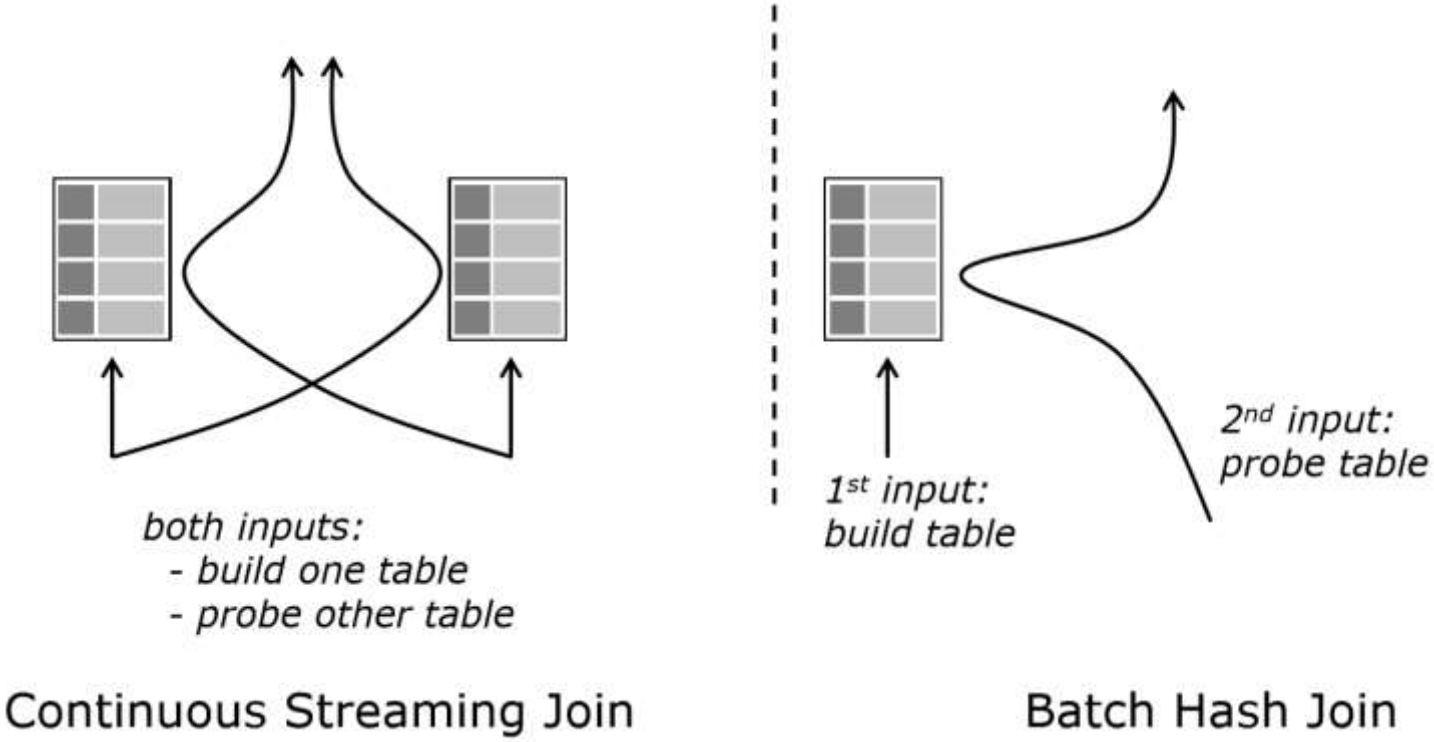


Batch Processing

Data is as complete as it gets within the job

No Low Latency SLAs

Streaming versus Batch Join



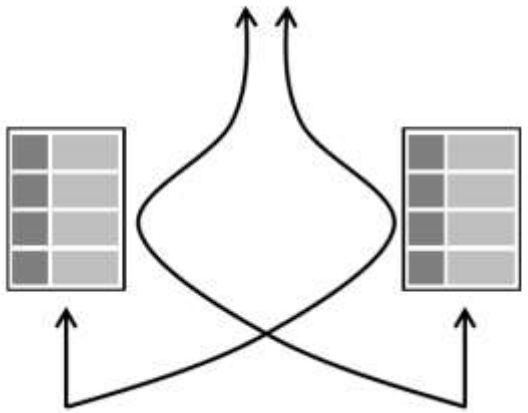
Streaming versus Batch Join

2x RocksDB
LSM-Trees

push-based
operators

low-latency

minimize
in-flight data

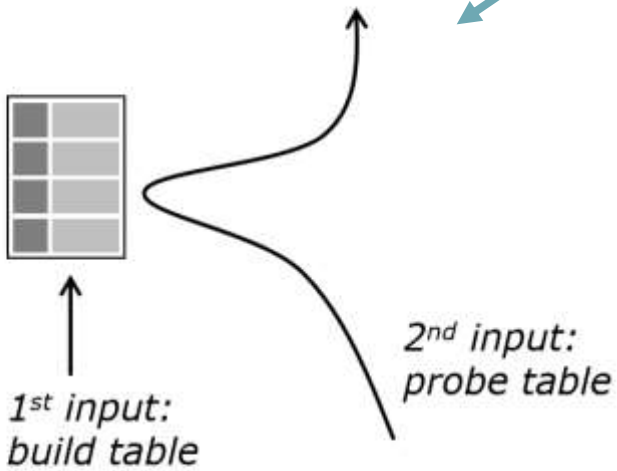


both inputs:
- build one table
- probe other table

Continuous Streaming Join

DataStream API

1x Hybrid Hash Join



Batch Hash Join

DataSet API

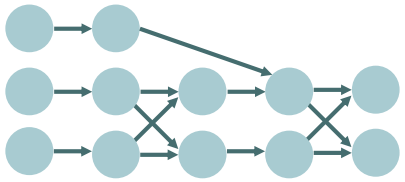
pull-based
operators

flexible data
flow control

high latency

no checkpoints

Exploiting the Batch Special Case



Planner/Optimizer

activates additional optimizer choices

```
if (bounded && non-incremental)
```

Core operators, cover all cases

Continuous Operators

Streaming Scheduler Rules

Additional Bounded Operators

Additional Scheduling Strategies

See also: "Towards Flink 2.0: Rethinking the stack and APIs to unify Batch & Stream" by Aljoscha Krettek, 2pm, Nikko II/III

Stream Processing, Analytics, and Applications

How we showed the API stack in the past…

High-level Analytics API

Stream SQL / Tables *(dynamic tables)*

```
SELECT room, TUMBLE_END(rowtime, INTERVAL '1' HOUR), AVG(temp)
FROM sensors
GROUP BY TUMBLE(rowtime, INTERVAL '1' HOUR), room
```

Stream- & Batch Data Processing

DataStream API *(streams, windows)*

```
val stats = stream
    .keyBy("sensor")
    .timeWindow(Time.seconds(5))
    .sum((a, b) -> a.add(b))
```

Stateful Event-Driven Applications

Process Function *(events, state, time)*

```
def processElement(event: MyEvent, ctx: Context, out: Collector[Result]) = {
  // work with event and state
  (event, state.value) match { … }

  out.collect(…) // emit events
  state.update(…) // modify state

  // schedule a timer callback
  ctx.timerService.registerEventTimeTimer(event.timestamp + 500)
}
```

Applications
(physical)

Types are Java / Scala classes

Transformation Functions

Executes as described

Explicit control over State

Explicit control over Time

DataStream API

Analytics
(declarative)

Logical Schema for Tables

Declarative Language (SQL, Table DSL)

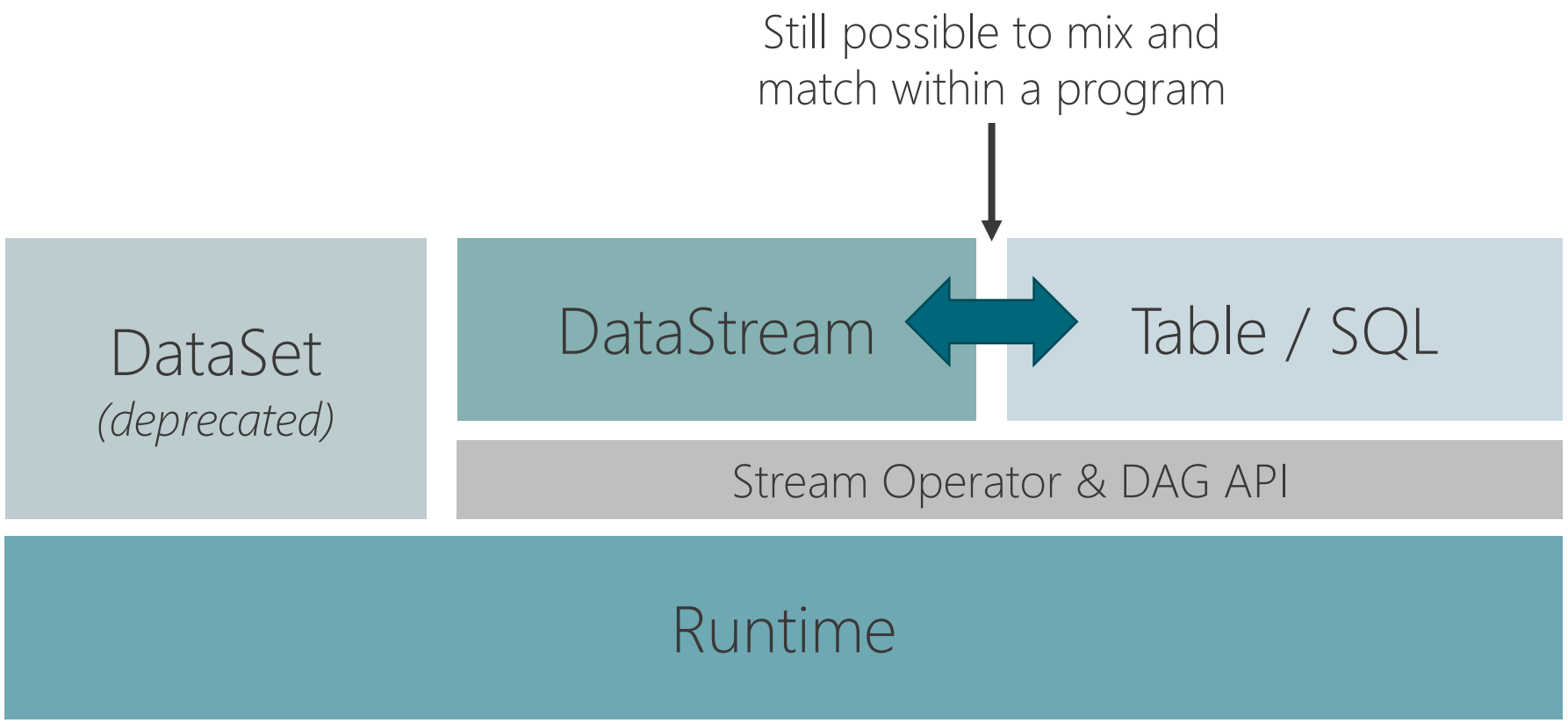
Automatic Optimization

State implicit in operations

SLAs define when to trigger

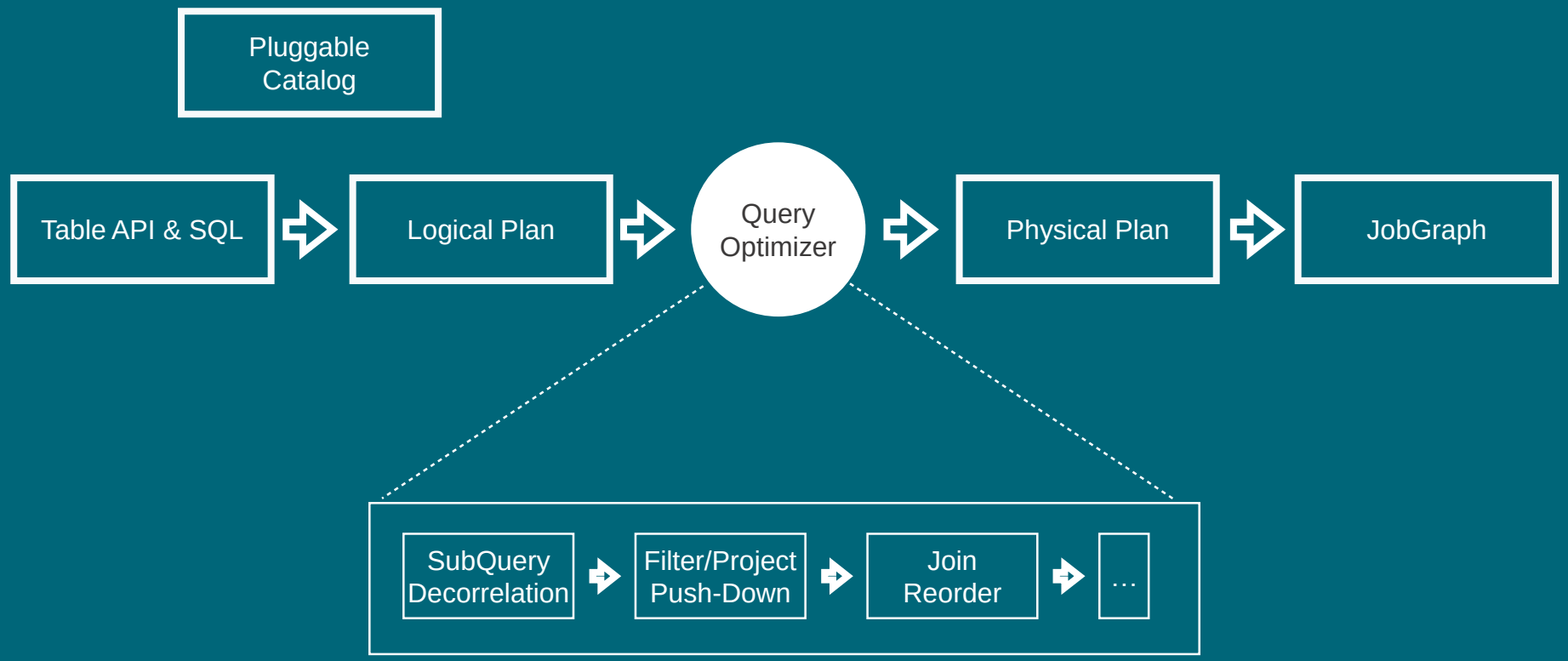
Table API

Rethinking the Flink Stack



SQL, Notebooks, and Machine Learning

Adding a new Table API / SQL Query Processor (Blink)



Functional Improvements

- ANSI Syntax
- Data Type
- Join
- Sub-Query
- Over Window
- Window Operator
- Advanced Aggregates
- TPC-H/TPC-DS

Performance Improvements

Expression Optimizations

Record Format
Operate binary data
JVM intrinsics
Hot method codegen

Performant Operators

Operator codegen
HashAgg/Local-global Agg
Improved HashJoin
Semi/Anti join
Vectorization

Resource Optimizations

Stats based estimation
Dynamic memory allocation

Query Execution

Cost Based

Join order
Join type
Agg strategy
……

Advanced Rules

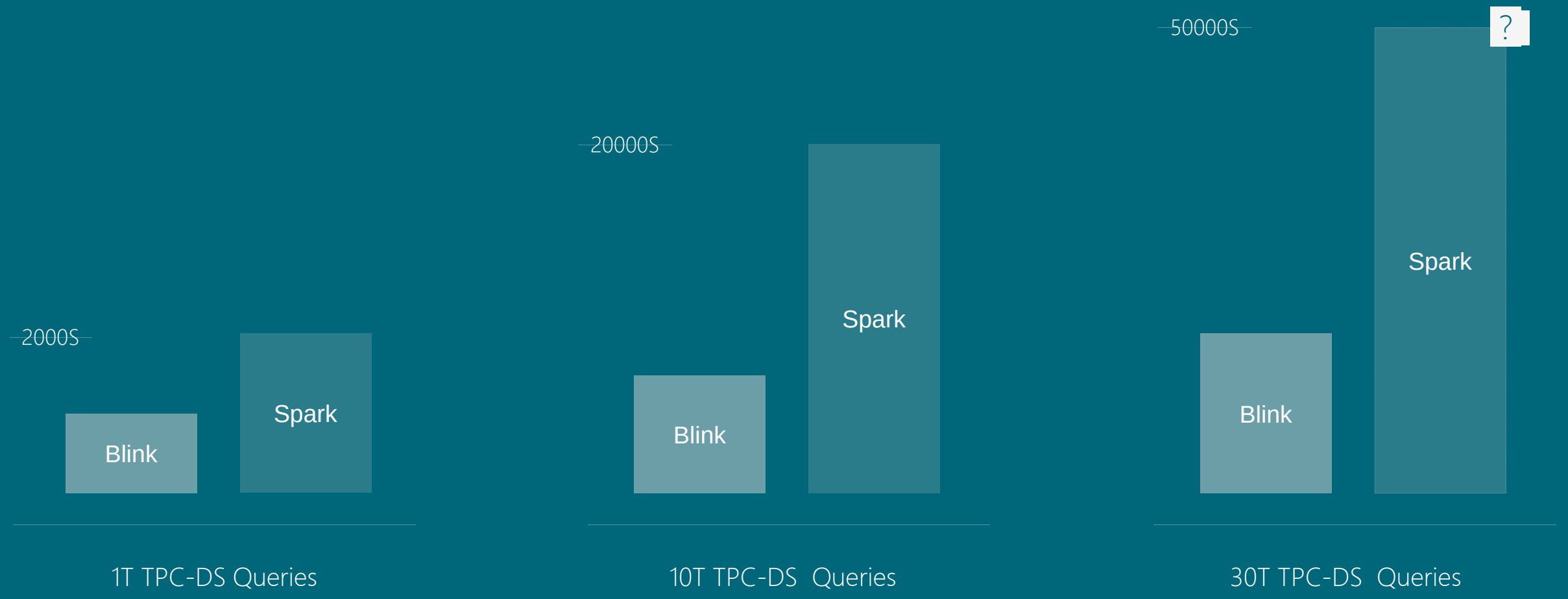
Subplan reuse
Join condition expansion
Shuffle removal
Distinct Agg rewrite
….

Rich Stats

NDV
NULL count
Avg length
Max length
Min
Max

Query Optimizer

Batch SQL Benchmark



Flink SQL in Production at Alibaba

10K

10K

1.7B

Sub-
Second

100TB

Table API

Common Implementation

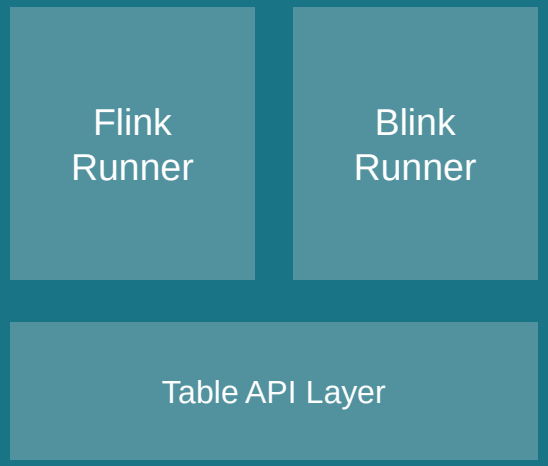
Modular/ Composable Applications

Dynamic Query Logic

Interactive Programming

Ease of Use

Blink SQL Merge Plan

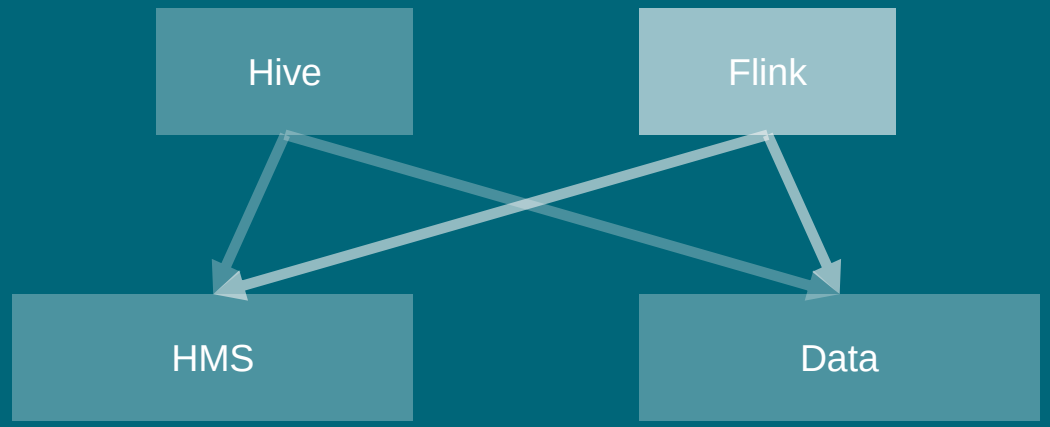


Hive Integration

Flink

+

Hive



For More ?
Integrate Flink with Hive Ecosystem
Xuefu Zhang & Bowen Li, Alibaba 12:20pm - 1:00pm Carmel

Zeppelin support for Flink

localhost:8086/#/notebook/2DXQMS9M8

```
}).assignAscendingTimestamps(_._1)

stenv.registerOrReplaceDataStream("log", data, 'time, 'url, 'rowtime.rowtime)
```

```
warning: there was one deprecation warning; re-run with -deprecation for details
import org.apache.flink.streaming.api.functions.source.SourceFunction
import org.apache.flink.table.api.TableEnvironment
import org.apache.flink.streaming.api.TimeCharacteristic
import org.apache.flink.streaming.api.checkpoint.ListCheckpointed
import java.util.Collections
import scala.collection.JavaConversions._
data: org.apache.flink.streaming.api.scala.DataStream[(Long, String)] = org.apache.flink.streaming.api.scala.DataStream@3b5093f3
```

Total Clicks

FLINK JOB RUNNING 0%

```
%flink.ssql(type=single, refreshInterval=1000, template=<h1>{}</h>, enableSavePoint=true,
    runWithSavePoint=true)

select count(1) from log
```

251

Started a few seconds ago.

Clicks Per Page

ABORT

```
%flink.ssql(refreshInterval=2000, enableSavePoint=true, runWithSavePoint=false)

select
    url,
    count(1) as pv
from log
    group by url
```

Took 1 min 17 sec. Last updated by anonymous at December 18 2018, 11:44:31 PM. (outdated)

Clicks Per Page Every 5 seconds

ABORT

```
%flink.ssql(refreshInterval=1000, enableSavePoint=true, runWithSavePoint=false)

select
    TUMBLE_START(rowtime, INTERVAL '5' SECOND) as start_time
```

Proposal for Machine Learning

For More ?

When Table meets AI: Build Flink AI Ecosystem on Table API
Shaoxuan Wang, Alibaba
4:30pm - 5:10pm Nikko II & II

High performance ML library based on Flink
Xu Yang, Alibaba
2:50pm - 3:10pm Carmel

Unified Interface

ML algorithms

Common Utilities

Regression
Linear regression
Lasso regression
Ridge regression
Generalized linear regression
Survival regression
Isotonic regression

Classifier
Binomial logistic regression
Multinomial logistic regression
Multilayer perceptron classifier
Linear Support Vector Machine
Naive Bayes
Random Forest
GBDT
Decision Tree

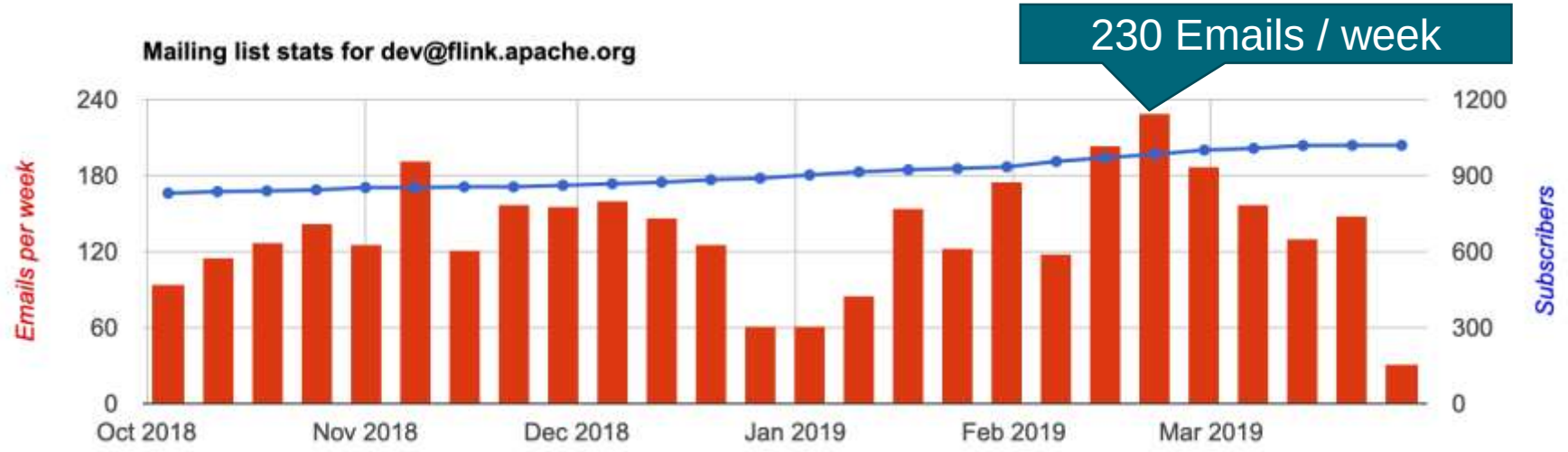
Clustering
K-means
Latent Dirichlet allocation (LDA)
Bisecting k-means
Gaussian Mixture Model (GMM)

Others
Collaborative filtering
FP-Growth
PrefixSpan

The Apache Flink Community

A growing Apache Flink Community

… not only Flink's codebase that is growing massively …



https://flink.apache.org/zh/poweredby.html

Flink

- Apache Flink 是什么?
- 应用场景
- Flink 用户
- 常见问题
- 下载
- 教程
- 文档
- 获取帮助
- Flink 博客
- 社区 & 项目信息
- 开发计划
- 如何参与贡献
- Flink on GitHub
- English
- @ApacheFlink
- Plan Visualizer

Flink 用户

Apache Flink 为全球许多公司和企业的关键业务提供支持。在这个页面上，我们展示了一些著名的 Flink 用户，他们在生产中运行着有意思的用例，并提供了展示更详细信息的链接。

在项目的 wiki 页面中有一个 谁在使用 Flink 的页面，展示了更多的 Flink 用户。请注意，该列表*并不全面*。我们只添加明确要求列出的用户。

如果你希望加入此页面，请通过 Flink 用户邮件列表告诉我们。

全球最大的零售商阿里巴巴（Alibaba）使用 Flink 的分支版本 Blink 来优化实时搜索排名。

阅读更多有关 Flink 在阿里巴巴扮演角色的信息

Amazon Kinesis Data Analytics 是一种用于流处理完全托管的云服务，它部分地使用 Apache Flink 来增加其 Java 应用程序功能。

BetterCloud 是一个多 SaaS 管理平台，它使用 Flink 从 SaaS 应用程序活动中获取近乎实时的智能。

请参阅 BetterCloud 在 Flink Forward SF 2017 上的分享

Bouygues Telecom 正在运行由 Flink 提供支持的 30 个生产应用程序，每天处理 100 亿个原始事件。

请参阅 Bouygues Telecom 在 Flink Forward 2016 上的分享

财富 500 强金融服务公司 Capital One 使用 Flink 进行实时活动监控和报警。

了解 Capital One 的欺诈检测用例

康卡斯特（Comcast）是一家全球媒体和技术公司，它使用 Flink 来实现机器学习模型和近实时事件流处理。

了解 Flink 在康卡斯特的应用

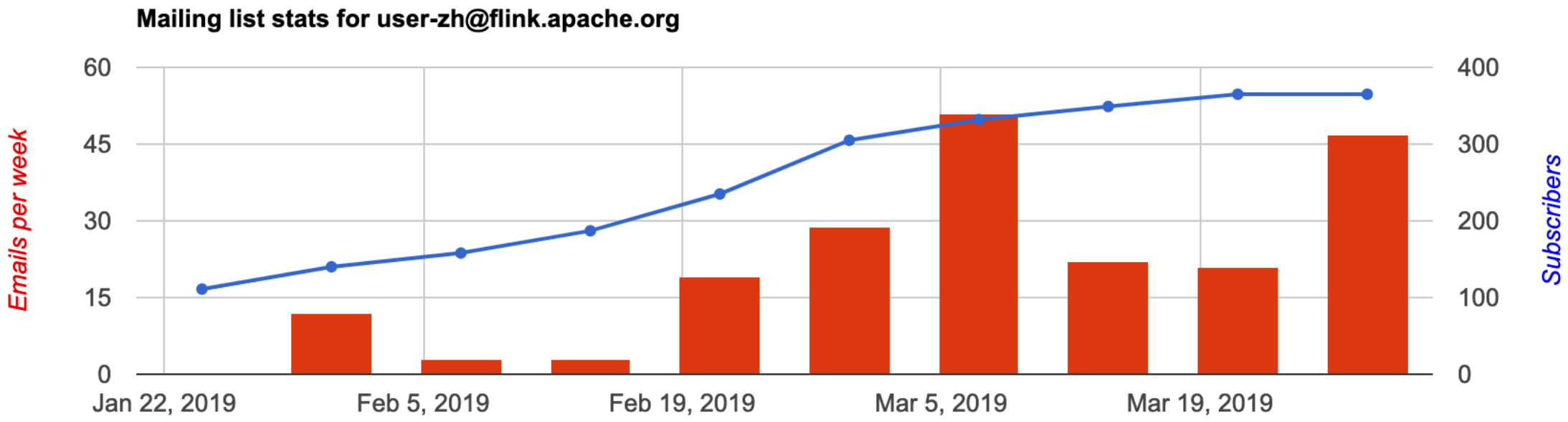
Criteo 是开放互联网的广告平台，使用 Flink 进行实时收入监控和近实时事件处理。

了解 Criteo 的 Flink 用例

Drivetribe是由前"Top Gear"主持人创建的数字社区，它使用 Flink 作为指标和内容推荐。

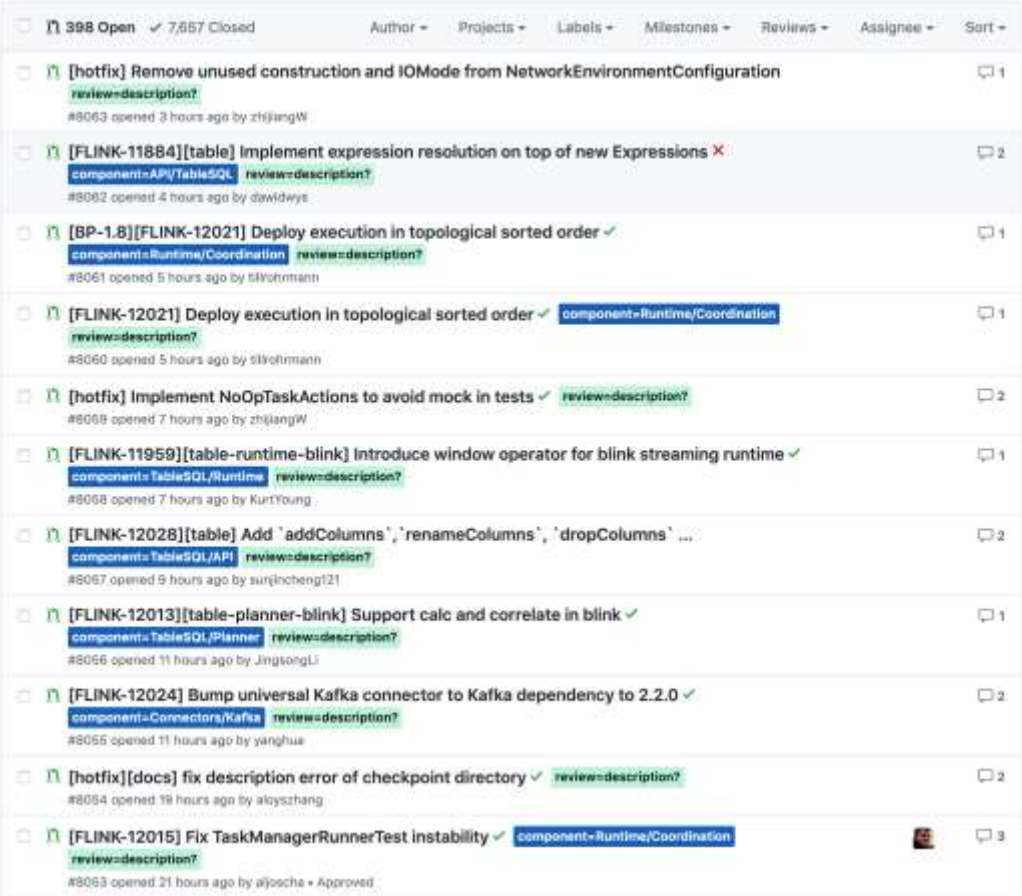
了解 Flink 在 Drivetribe stack 的应用

Launch of a new Chinese language user support mailing list

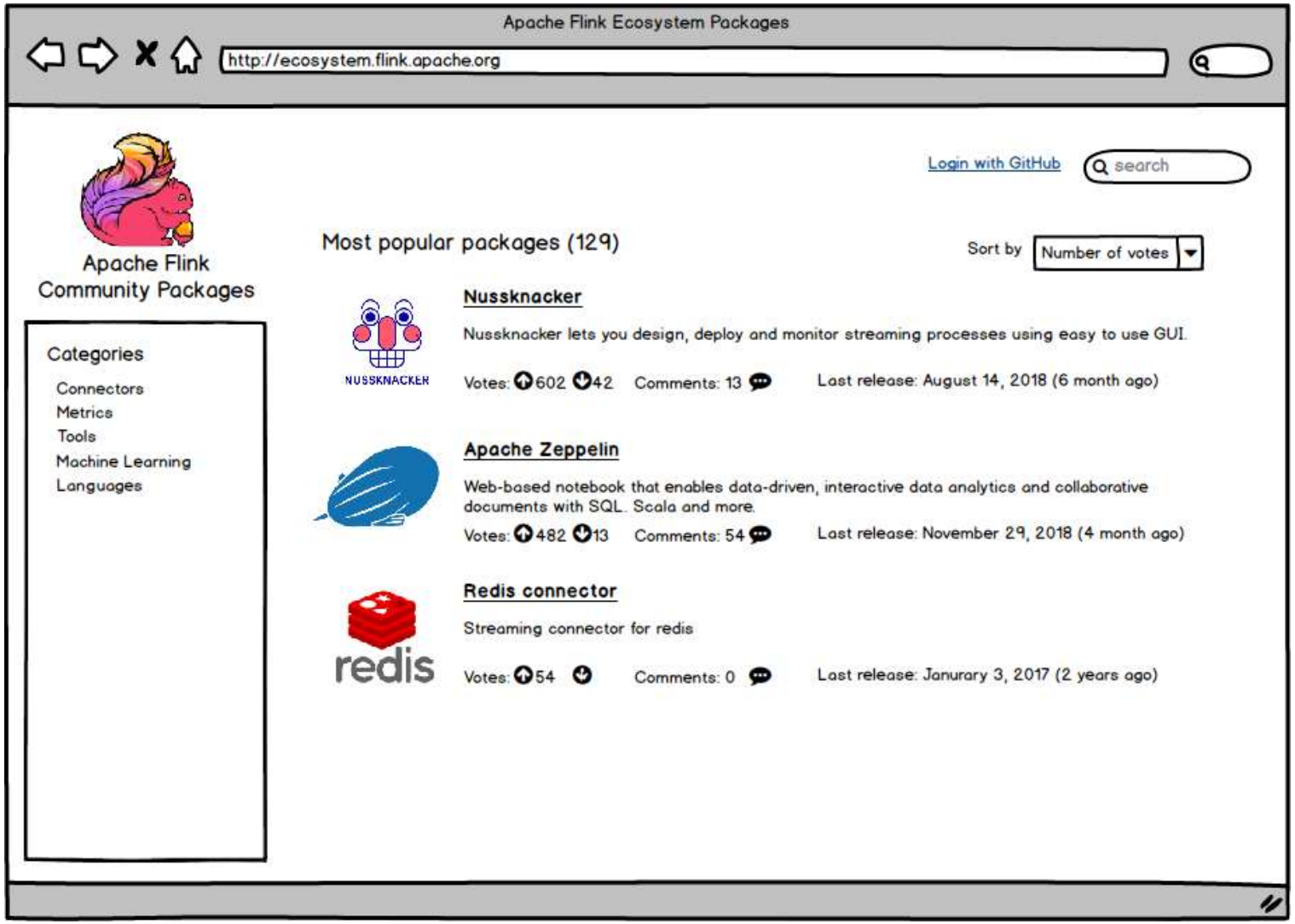


Growing the Contributors Community

- Cleanup & reorganization of the Jira components
- Flinkbot: Improve pull request reviews and labeling
- Discussions about improving the contribution workflow
- PMC mentoring new committer candidates
- Flink Community Packages website



Flink Community Packages



Closing

Apache Flink continues to evolve with the Stream Processing space.

Seamlessly integrate *analytics*, *machine learning*, *applications* and *very fast batch processing* on top of stream processing

The Apache Flink community is more active than ever

Thank you!