PyFlink 核心技术及案例

孙金城 花名 金竹
阿里巴巴高级技术专家 / Apache Flink PMC

Apache Flink China Meetup 北京
- 2019年09月21日

Apache Flink

About Me

2019
- Apache Flink Python API
- Apache Flink PMC

2017
- Apache Flink Table/SQL API
- Apache Flink Committer

2011
- 云转码/阿里郎/OPLog/云代码/…
- 计算平台/搜索/信息平台/技术平台

https://enjoyment.cool

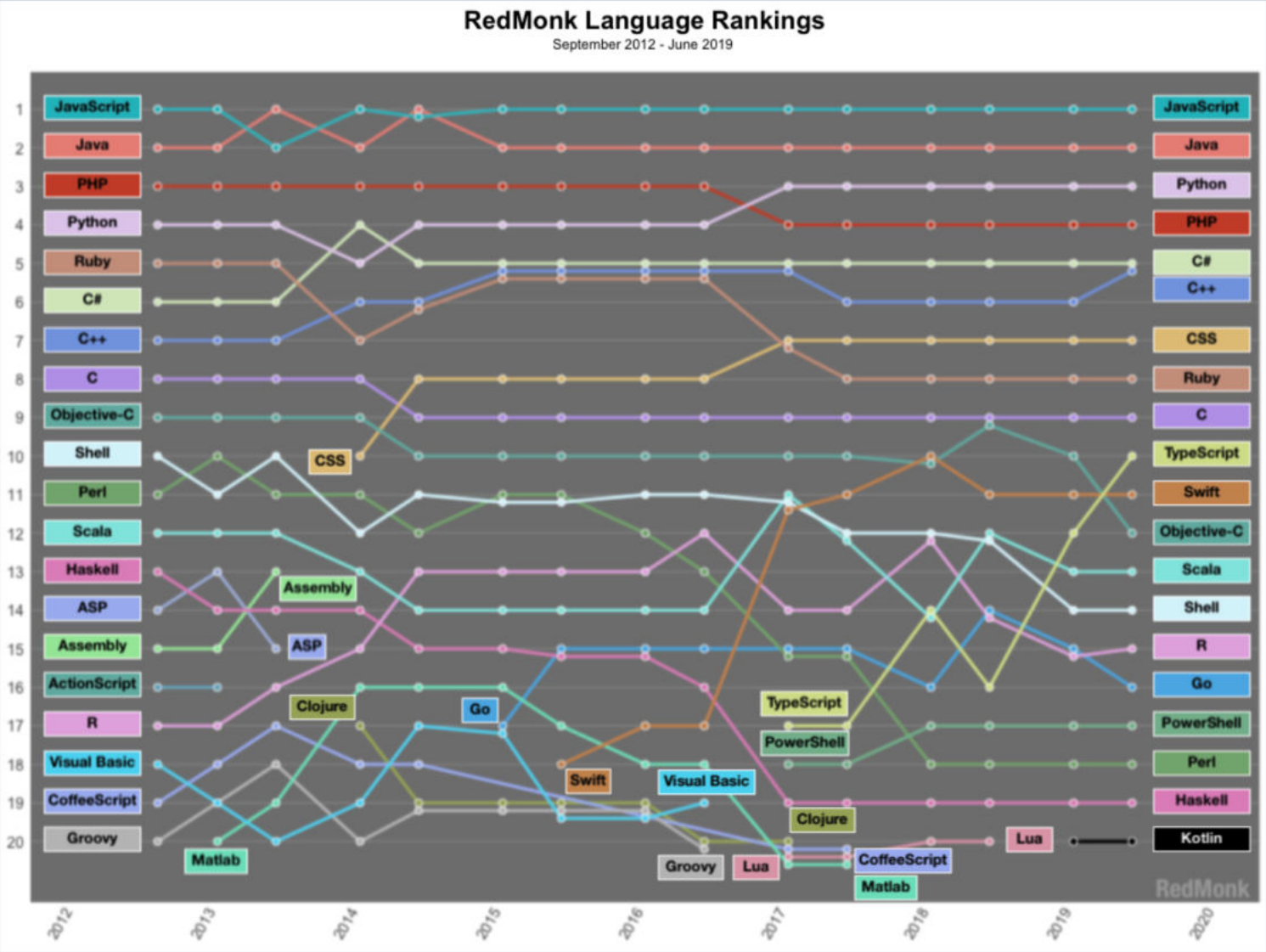
Continuous Queries
Watermark
Fault Tolerance
JOIN
Window

Content

目录 >>

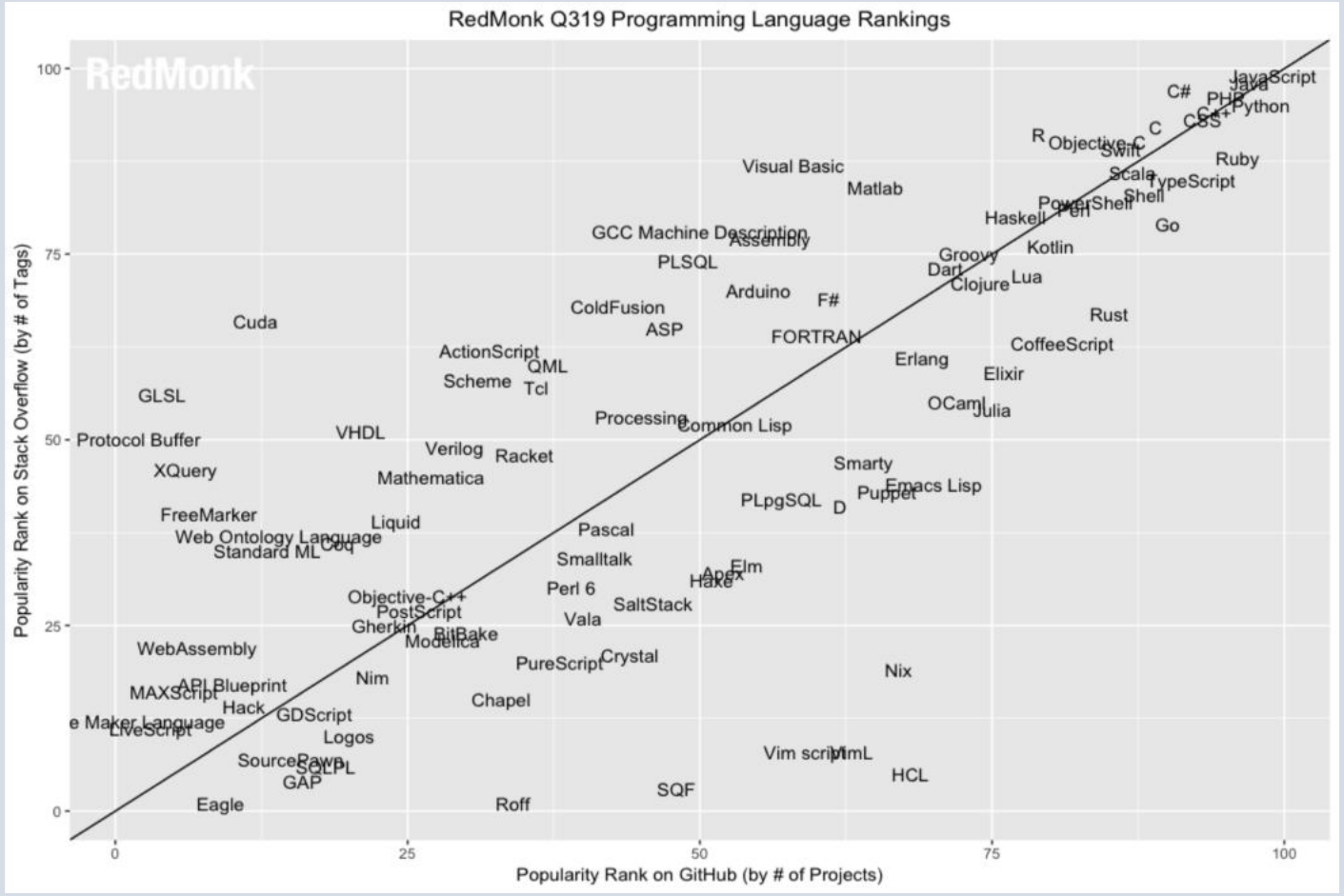
Why-最流行的开发语言

Apache Flink



Why-最流行的开发语言

Apache Flink



- JavaScript
- Java
- **Python**
- PHP
- C++
- C#
- CSS
- Ruby
- C
- TypeScript

统计数据来源于RedMonk 2019.06月

Why-广泛应用于大数据领域

Apache Flink

PyHDFS
2019.6.15

Kafka-python
2019.4.3

PyHive
2018.9.10

PyFlink
Release-1.9

Flink

APACHE
STORM™

Streamparse
2019.4.8

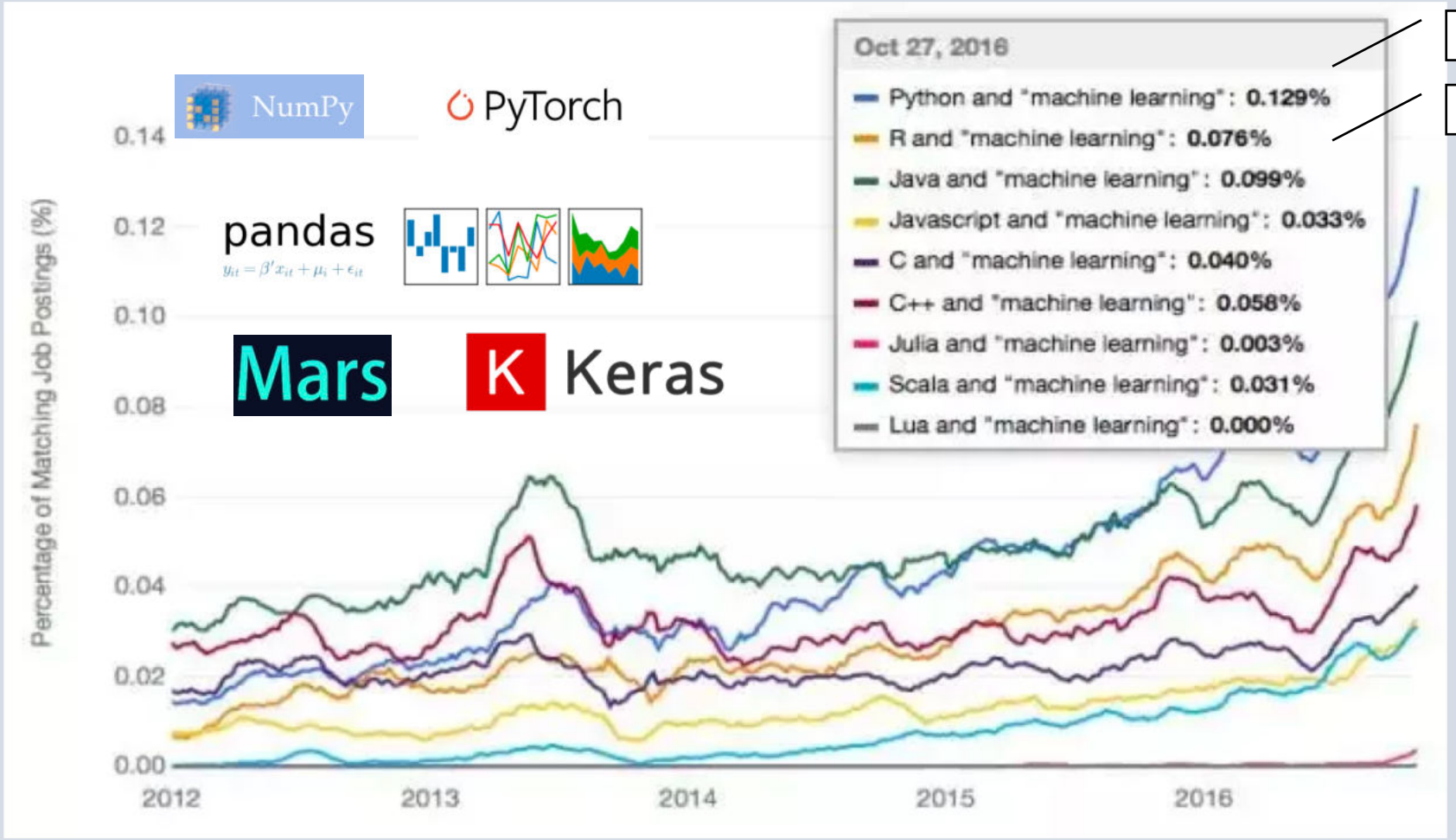
PySpark
2019

PyHbase
2010.6.18

ApexPy
2018.4.6

Why-深受机器学习的青睐

Apache Flink

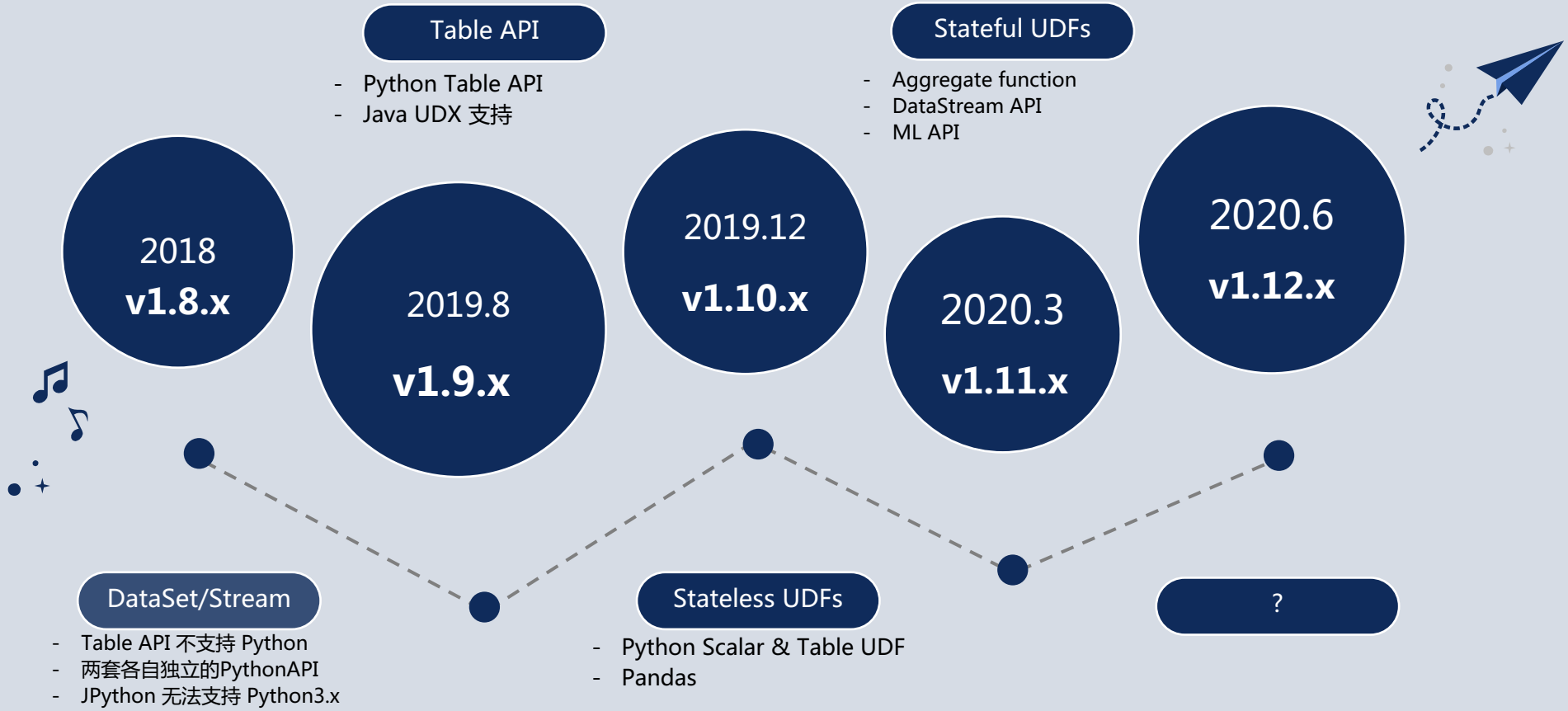


Python 0.129%

R 0.076%

What- PyFlink Roadmap

Apache Flink



What- On Going

Apache Flink

FLIP-58 Python UDFs(1.10)

- Beam 社区讨论: http://1t.click/amEu
- Flink 社区讨论: http://1t.click/amE9
- 设计文档: http://1t.click/amEK
- 主JIRA: FLINK-14013

FLIP-38 Python Table API(1.9)

- 全新的架构 Py4j (Py VM & JVM)
- Flink 社区讨论: http://1t.click/amE9
- 设计文档: http://1t.click/amFh
- 主JIRA: FLINK-12308

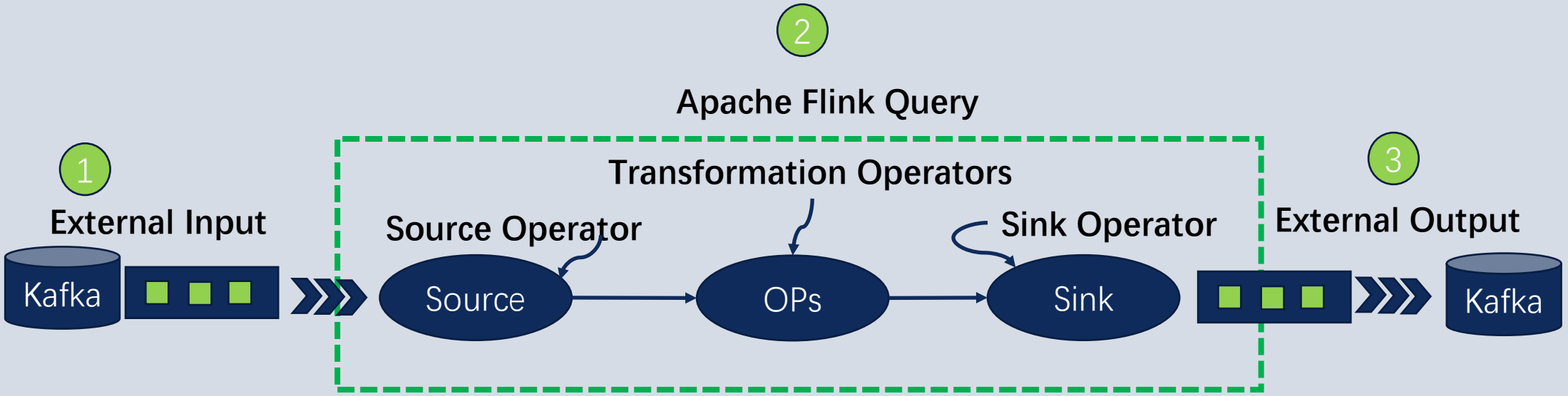
https://enjoyment.cool

Content

目录 >>

典型的Flink Job结构

Apache Flink



案例1 - Word Count

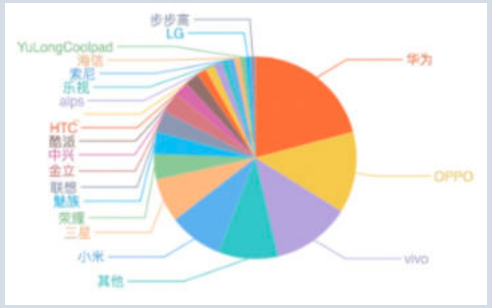
Apache Flink

word
Apache Flink
cool
enjoyment
Apache Flink

Word Count

word	count
Apache Flink	2
cool	1
enjoyment	1

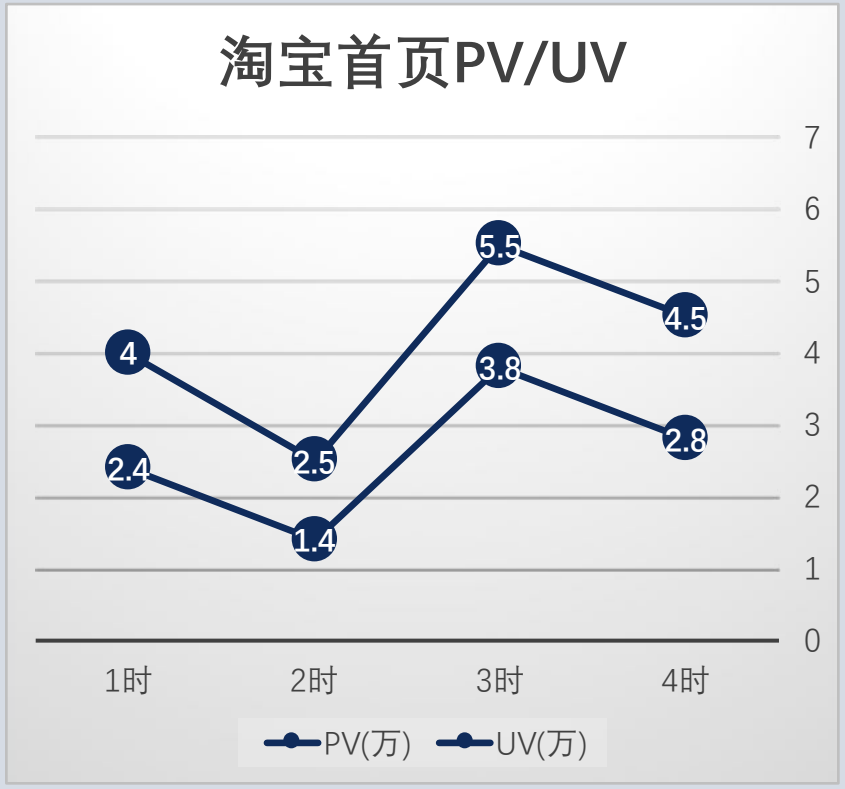
```
t_env.scan('mySource').group_by('word') .select('word, count(1) ').insert_into('mySink')
```



https://github.com/sunjincheng121/enjoyment.code/blob/master/myPyFlink/enjoyment/word_count.py

案例2- 淘宝首页PV/UV

Apache Flink



场景描述及分析

场景：

假设我们的淘宝首页的用户访问记录全部存储到Kafka中，记录包含用户信息，访问时间，访问来源等关键信息。我们统计每小时淘宝首页的PV和UV指标。

分析：

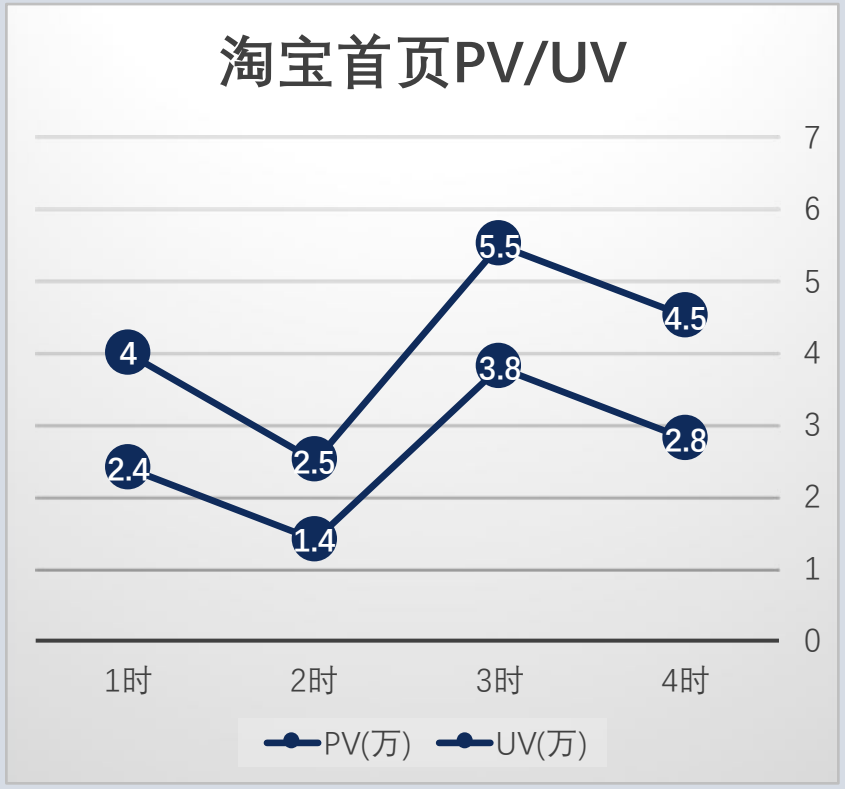
每小时 -> Tumble 窗口

PV -> 简单的Count 聚合

UV -> 按用户去重的Count 聚合

案例2- 淘宝首页PV/UV

Apache Flink



用户行为数据结构

数据存储： Kafka

数据示例 JSON：

```
{
        "user_id": "543462",
        "item_id":"1715",
        "category_id": "1464116",
        "behavior": "pv",
        "ts": "2017-11-26T01:00:00Z”
}
```

案例2- 淘宝首页PV/UV

Apache Flink

创建Kafka数据源表

Kafka链接信息 → 数据格式定义 → Flink 表Schema定义

```
st_env
.connect(
  Kafka()
 .version("0.11")
.topic("user_behavior")
.start_from_earliest()
.property("zookeeper.connect",
          "localhost:2181")
.property("bootstrap.servers",
          "localhost:9092")
)
```

```
.with_format(
   Json().json_schema(
"{type: 'object',"
"  properties: {"
"user_id: {type: 'string'},"
"item_id: {type: 'string'},"
"category_id: {type: 'string'},"
"behavior: {type: 'string'},"
"ts: {"
"type: 'string',"
"format: 'date-time'}}}"
```

```
.with_schema(Schema()
.field("user_id", STRING())
.field("item_id", STRING())
.field("category_id", STRING())
.field("behavior", STRING())
.field("rowtime", TIMESTAMP())
.rowtime(Rowtime()
  .timestamps_from_field("ts")
  .watermarks_periodic_bounded(1000))
) )
```

) Watermark 原理： http://1t.click/apzf

案例2- 淘宝首页PV/UV

Apache Flink

创建JDBC结果表

假设统计结果存储到关系数据中，比如Derby中，根据统计逻辑可以知道结果表包含 **startTime,endTime,pv,uv**

链接属性

```
custom_connector = CustomConnectorDescriptor('jdbc', 1, False)
    .property("connector.driver", "org.apache.derby.jdbc.ClientDriver")
    .property("connector.url", "jdbc:derby://localhost:1527/firstdb")
    .property("connector.table", "pv_uv_table")
    .property("connector.write.flush.max-rows", "1")
```

表的结构定义

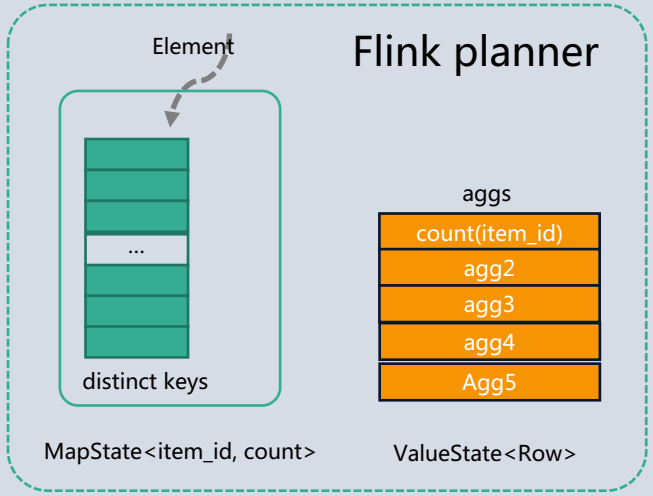
```
st_env.connect(custom_connector)
    .with_schema(
    Schema()
        .field("startTime", DataTypes.TIMESTAMP())
        .field("endTime", DataTypes.TIMESTAMP())
        .field("pv", DataTypes.BIGINT())
        .field("uv", DataTypes.BIGINT())
).register_table_sink("sink")
```

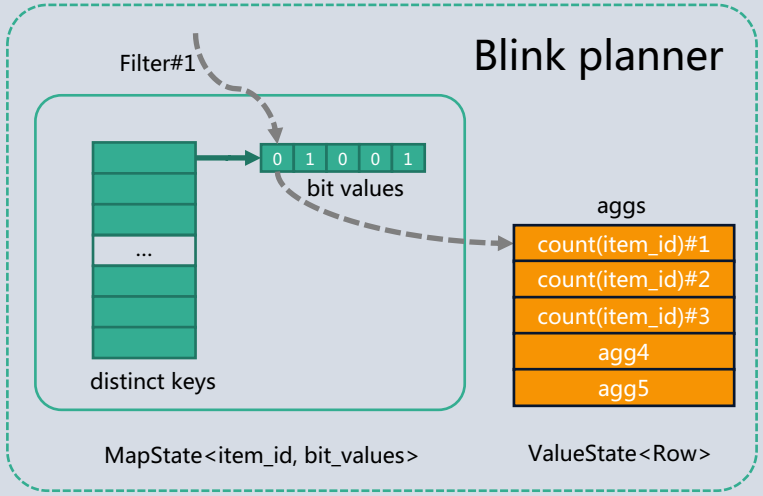
案例2- 淘宝首页PV/UV

Apache Flink

编写PV/UV统计逻辑

```
st_env.scan("source")
   .window(Tumble.over("1.hours").on("rowtime").alias("w"))
   .group_by("w")
   .select("w.start as startTime, w.end as endTime, COUNT(1) as pv, user_id.count.distinct as uv")
   .insert_into("sink")
```





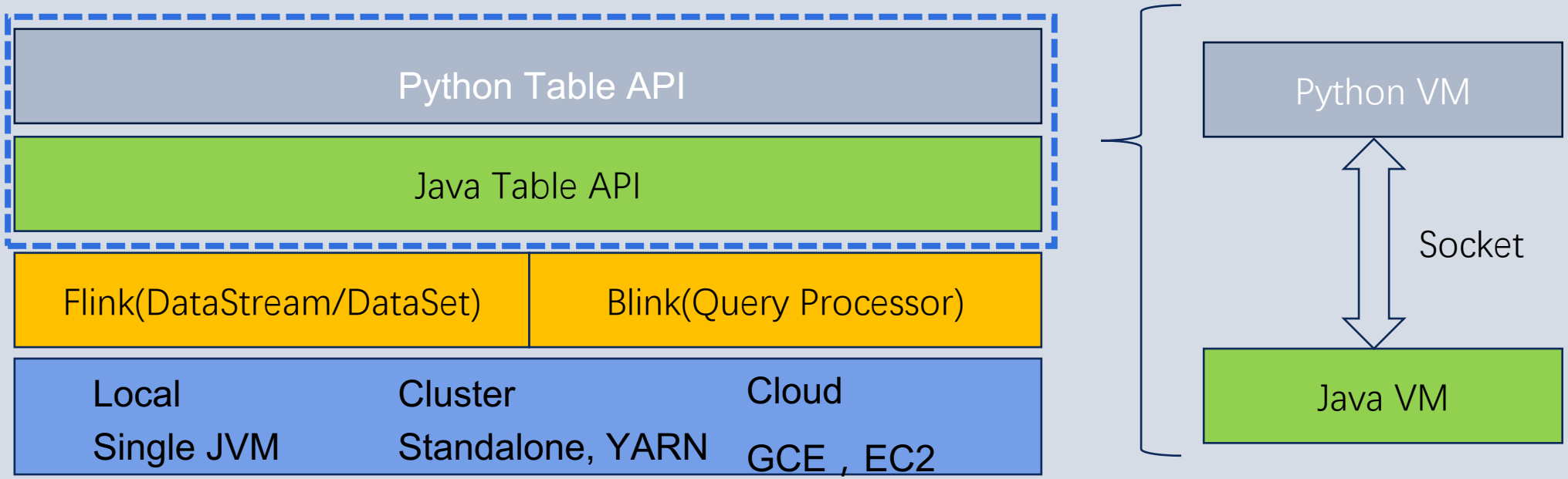
Content

目录 >>

Python Table API 性能怎么样？

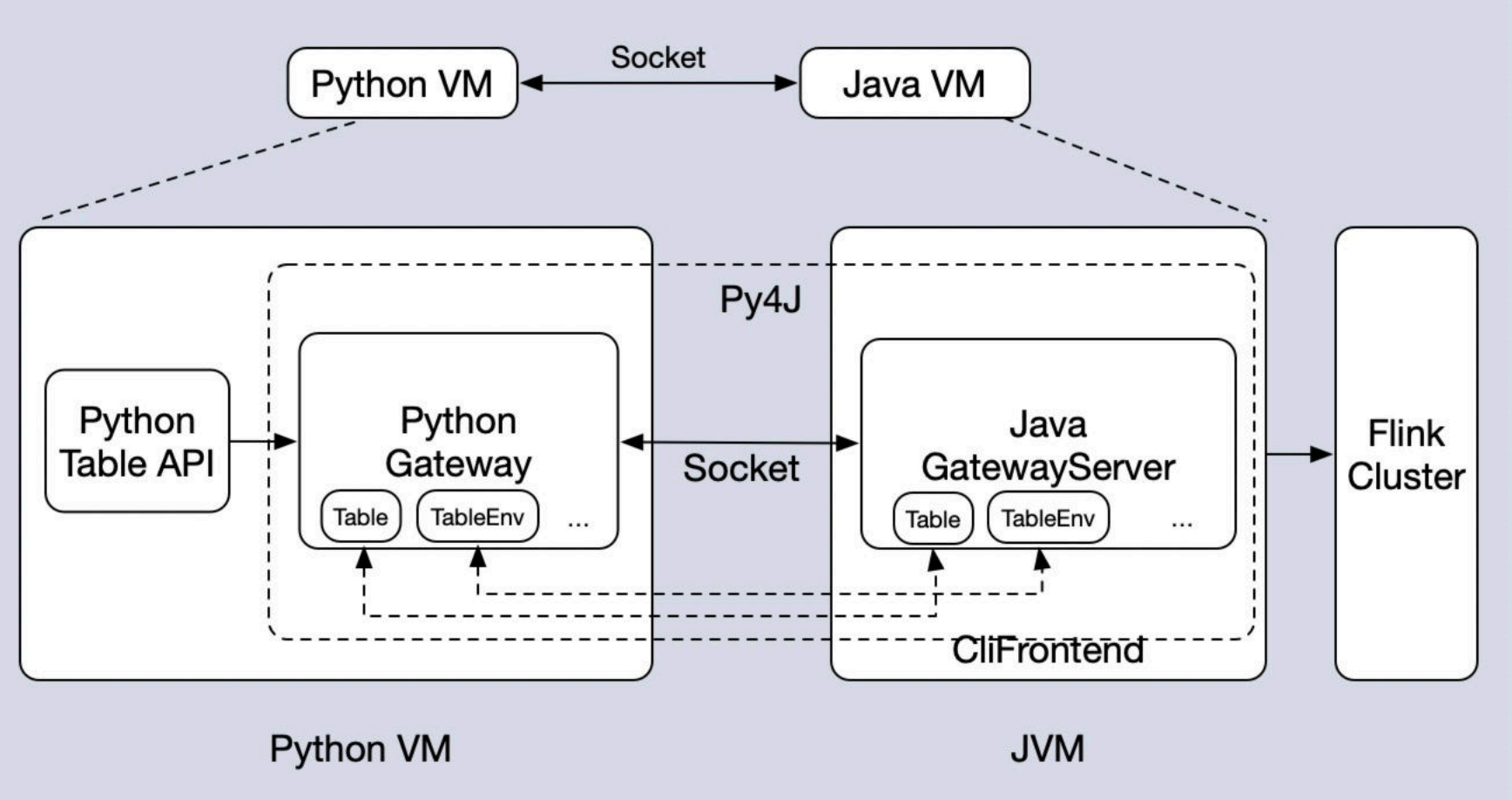
Apache Flink 1.9 新架构

Apache Flink

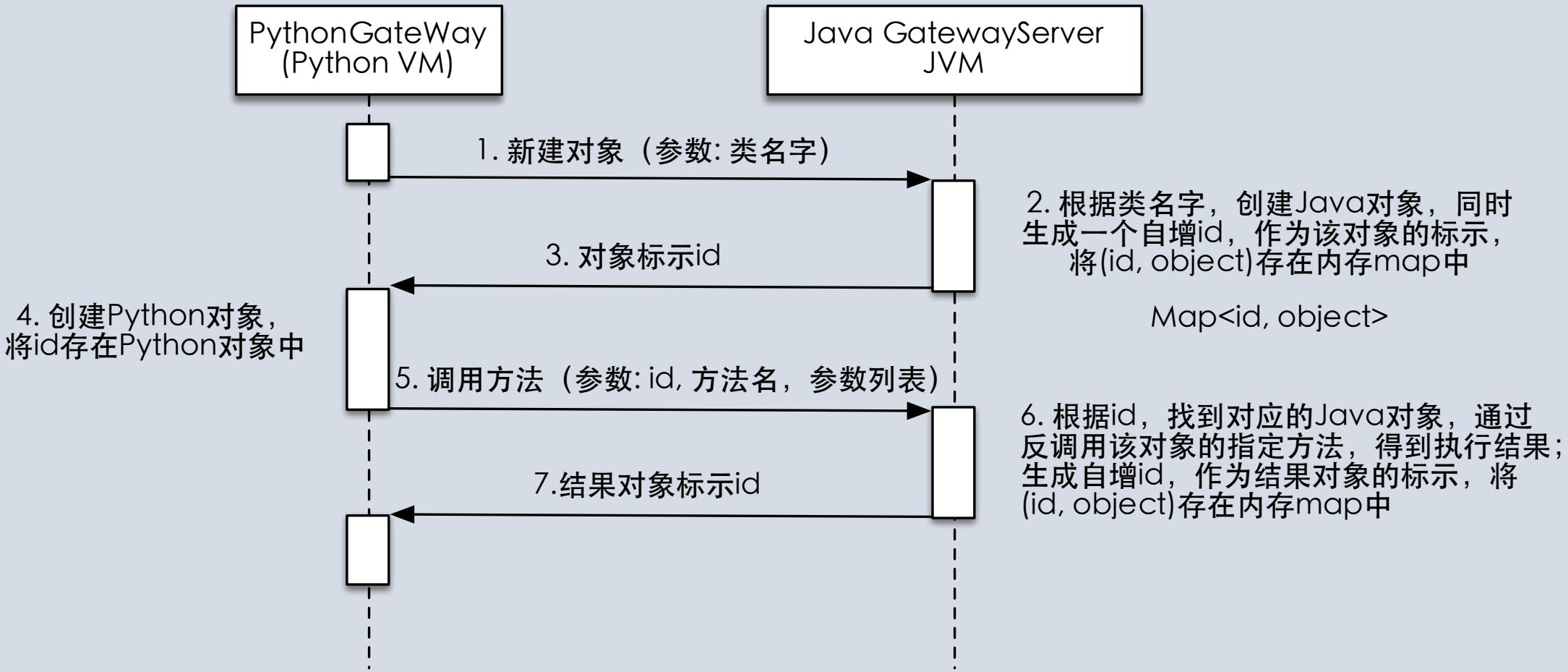


Apache Flink 1.9 新架构

Apache Flink



Apache Flink 1.9 新架构



Python Table API 性能怎么样？

Python Table API性能

.equals(Java Table API性能)?

That’s TRUE！

Python Table API 特点

Apache Flink

低延迟

快速容错

支持
大数据量

高性能

流批统一

正确
处理乱序

Python Table API 功能

Apache Flink


Query Config

Java UDX


JOIN


Retraction


Window
AGG


Query
Optimization

Python Table API 功能

Apache Flink

Query Config

Query Config

- IdleState TTL
- NULL check
- Timezone
- etc..

```
query_config
.set_local_timezone("Asia/Shanghai")
```

```
query_config
.set_max_generated_code_length(32000)
```

Python Table API 功能

Apache Flink

Java UDX

Java UDX

- UDF
- UDTF
- UDAF

```
t_env.register_java_function(
"len", "org.apache.flink.udf.UDFLength")
...
.select("word, len(word), count(1) as count")
```

完整示例代码: http://1t.click/an2t

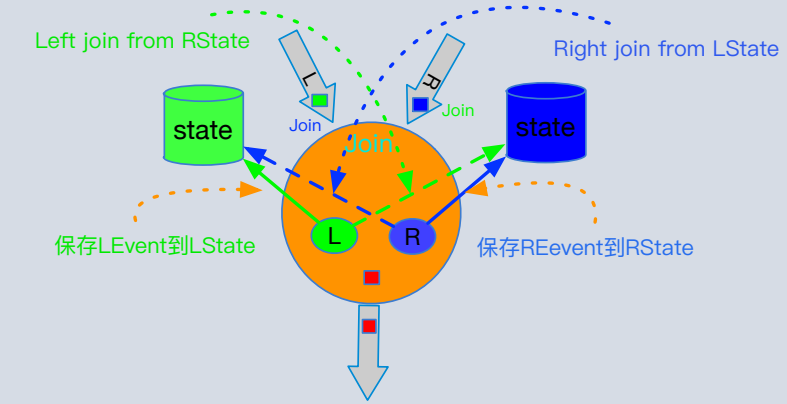
Python Table API 功能

Apache Flink

JOIN

JOIN

- JOIN
- LEFT_JOIN
- RIGHT_JOIN
- FULL_JOIN




```
t1.join(t2, "a = d")
t1.left_outer_join(t2, "a = d")
```

JOIN原理介绍: http://1t.click/an6J

Python Table API 功能

Apache Flink

Window AGG

Window

- Group Window – TUMBLE/SLIDE/SESSION
- Over Window

```
t.window([WIN]).group_by(" w, … ")

Tumble.over("2.rows").on("a").alias("w"))
Slide.over("2.seconds").every("1.seconds").on("a").alias("w")
Session.with_gap("1.seconds").on("a").alias("w")

t.over_window([OVER_WIN])
Over.partition_by("c").order_by("a").preceding("2.rows").following
("current_row").alias("w"))
```

Window原理介绍: http://1t.click/an7y

Python Table API 功能

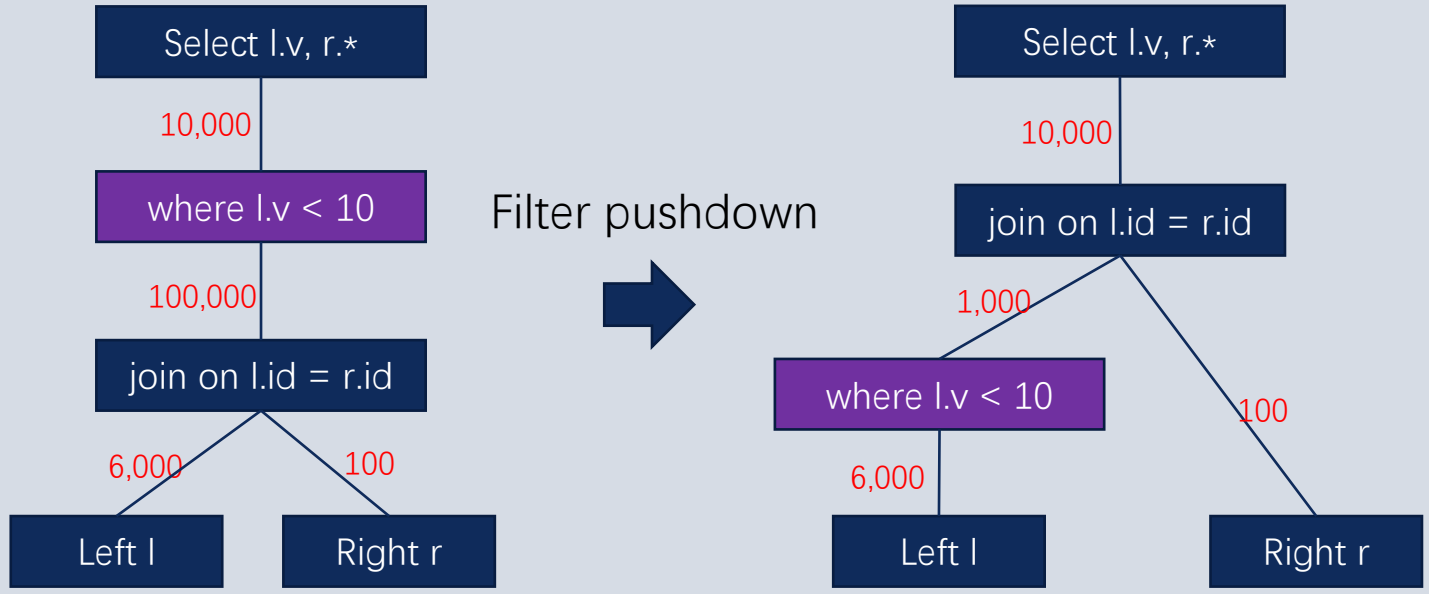
Apache Flink

Query Optimization

Optimization

基于Calcite 优化模型,有上百种优化规则，比如：Filter Pushdown，

Calc merge， local-global ...



Python Table API 功能

Apache Flink

Retraction

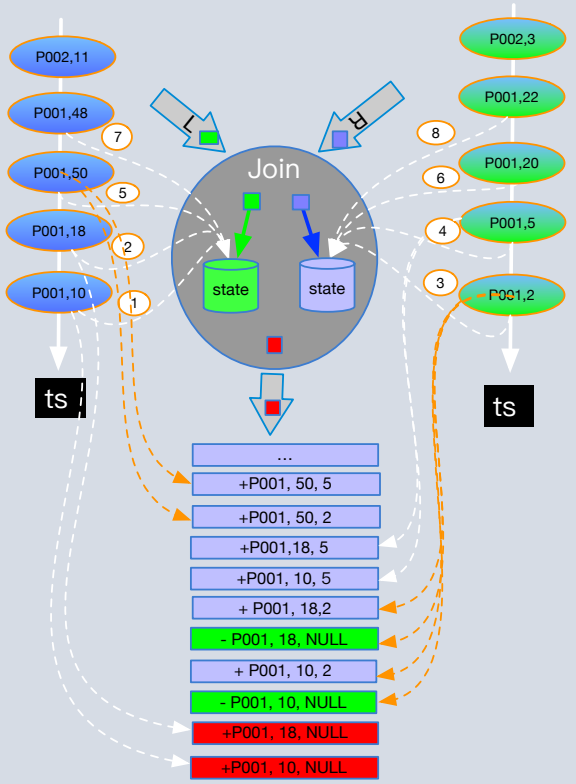
Retraction

- Early Emit – 保障低延时
- Retraction – 保障数据正确性

(flag，Data)

(-，Data)

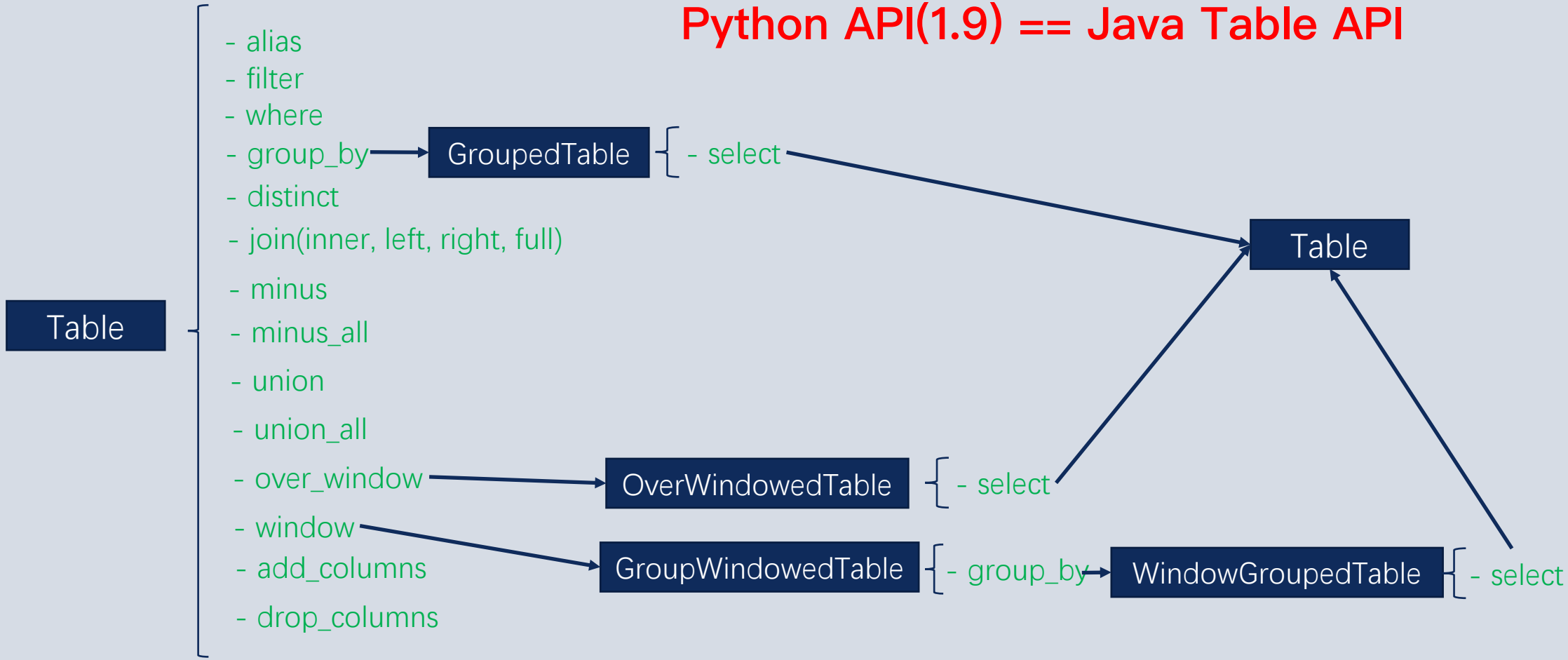
(+，Data)



《Flink SQL 关键技术及实现原理》 视频: http://1t.click/aphj

Python Table API 功能

Apache Flink



Content

目录 >>

Notebook - Zeppelin

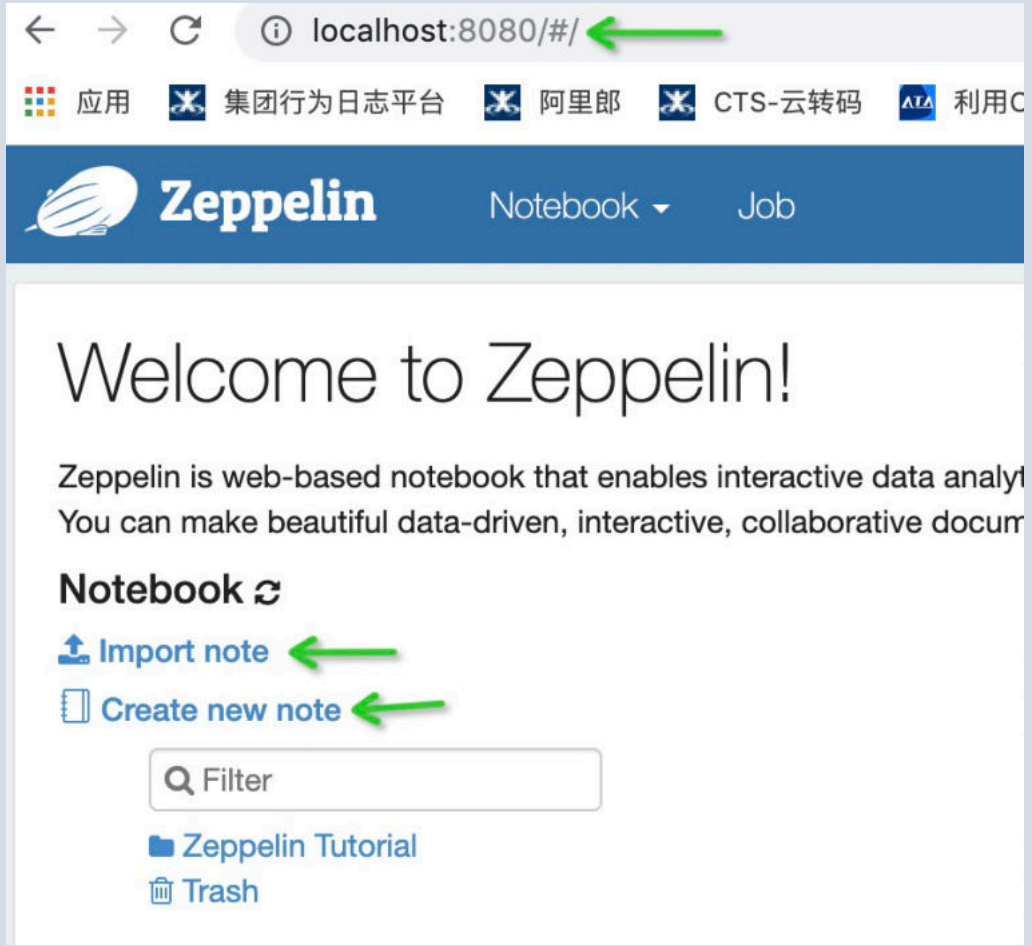
Apache Flink

下载源码 → 编译构建 → 启动web

- `git clone https://github.com/apache/zeppelin.git`
- `mvn clean package -DskipTests -pl zeppelin-web,zeppelin-server,flink -am`
- `bin/zeppelin-daemon.sh start`

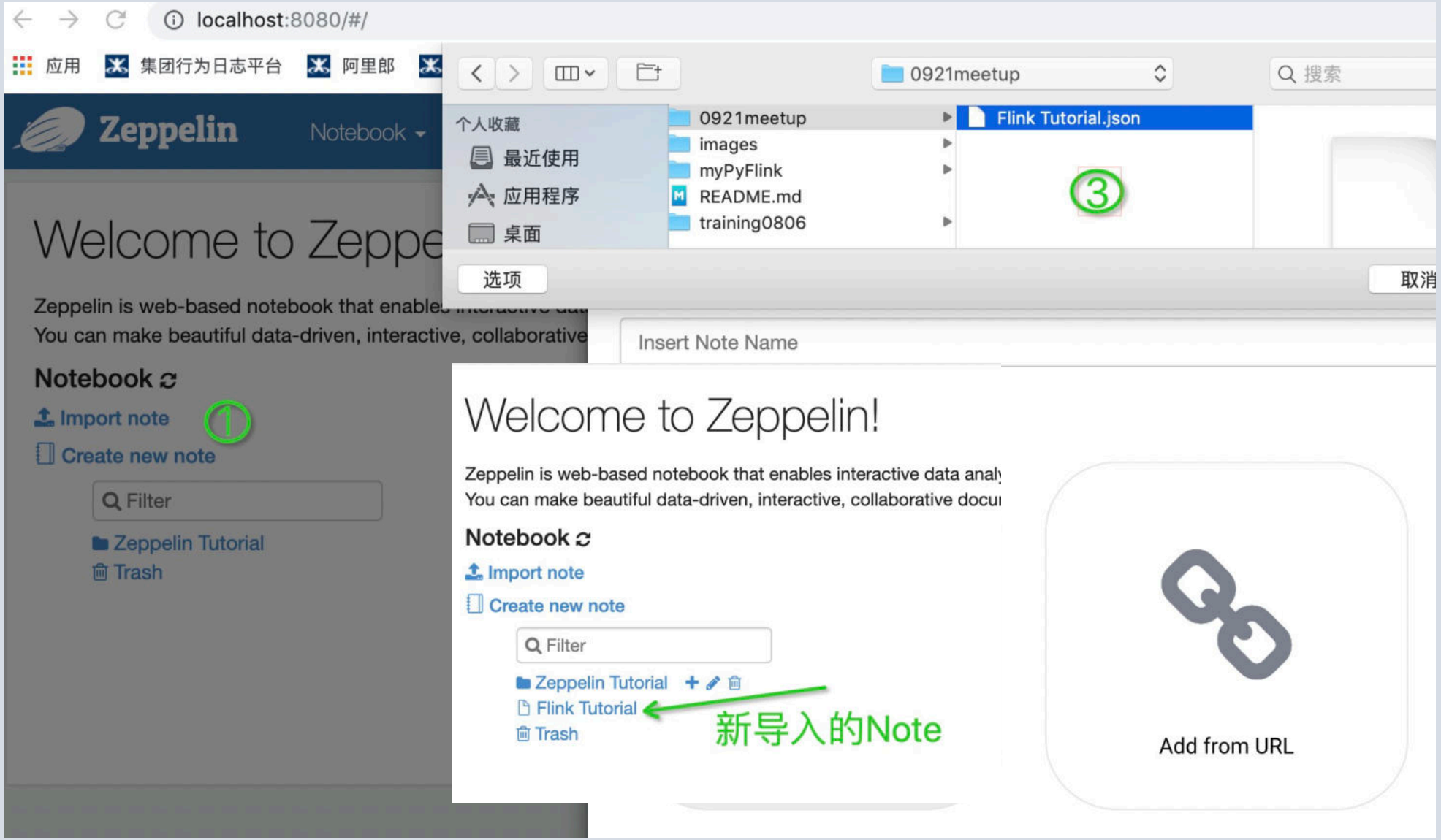
http://localhost:8080

Flink Tutorial.json: http://1t.click/apPk



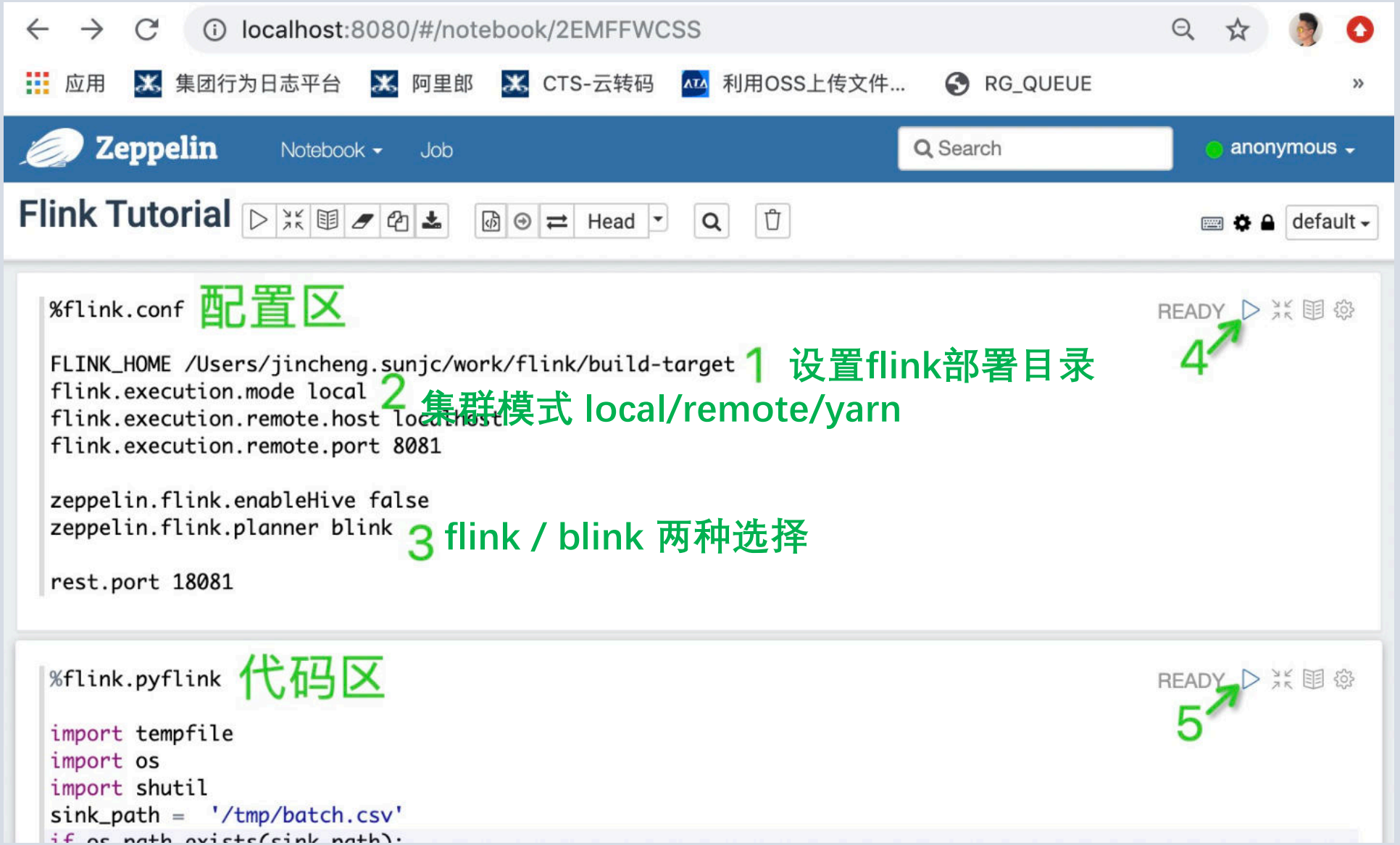
Notebook - Zeppelin

Apache Flink



Notebook - Zeppelin

Apache Flink



Notebook - Zeppelin

Apache Flink

localhost:8080/#/notebook/2EMFFWCSS

应用 集团行为日志平台 阿里郎 CTS-云转码 利用OSS上传文件... RG_QUEUE

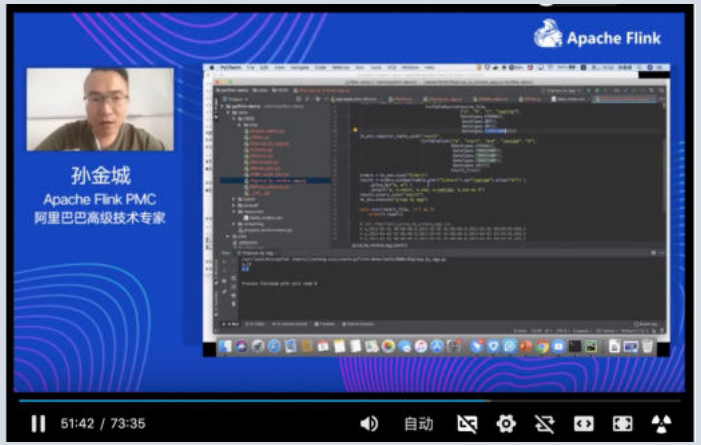
```
%flink.pyflink

import tempfile
import os
import shutil
sink_path =  '/tmp/batch.csv'
if os.path.exists(sink_path):
  if os.path.isfile(sink_path):
        os.remove(sink_path)
  else:
        shutil.rmtree(sink_path)
b_env.set_parallelism(1)
t = bt_env.from_elements([(1, 'hi', 'hello'), (2, 'hi', 'hello')], ['a', 'b', 'c'])
bt_env.connect(FileSystem().path(sink_path)) \
                .field("c", DataTypes.STRING())) \
    .register_table_sink("batch_sink")

t.select("a + 1, b, c").insert_into("batch_sink")

bt_env.execute("batch_job")
```

FINISHED

5

查看结果

```
jincheng:enjoyment.code jincheng.sunjc$ cat /tmp/batch.csv
2,hi,hello
3,hi,hello
jincheng:enjoyment.code jincheng.sunjc$
```



直播视频

学习资料

Apache Flink

Ververica，微信公众号，由 Apache Flink Community China 运营管理，旨在联合国内的 Flink 大 V，向国内宣传和普及 Flink 相关的技术。

我的个人博客，分享Apache Flink 相关的技术，涉及到核心概念剖析，算子语义和实现原理介绍以及用户案例和部分视频资源。

https://enjoyment.cool

FLINK FORWARD

Organized by 阿里云

Created by ververica

Flink Forward Asia

全球最大的 Apache Flink 官方会议

预计 **2000+** 参会人员， 2019年11月28-30日 @北京国家会议中心

国内外一线厂商悉数参与

阿里巴巴、腾讯、字节跳动、intel、 DellEMC 、Uber、美团点评、Ververica ...

大会官网，查看更多

THANKS

Apache Flink China Meetup

· BEIJING