Alibaba Cloud | Worldwide Cloud Services Partner

Python on Flink & Flink on Zeppelin

章剑锋(简锋)&孙金城(金竹)

Staff Engineer @Alibaba

04/22/2020

About Me

Alibaba Cloud | Worldwide Cloud Services Partner

Alibaba Group

Staff Engineer
@ Alibaba
Apache Flink

Flink

Apache Flink
PMC

beam

Apache Beam
Committer

ALC Beijing
Member

孙金城（金竹）

但行好事，
莫问前程

2016.10 | 2017.7 | 2019.6 | 2020.1 | 2020.2 | 2020.4 | ?

Apache Flink
Committer

Apache IoTDB
Committer

APACHE IoTDB

ASF
Member

Agenda

① Why PyFlink

② Status of PyFlink

③ Future of PyFlink

④ Flink on Zeppelin

PyFlink 架构、应用案例及未来规划

Alibaba Cloud | Worldwide Cloud Services Partner

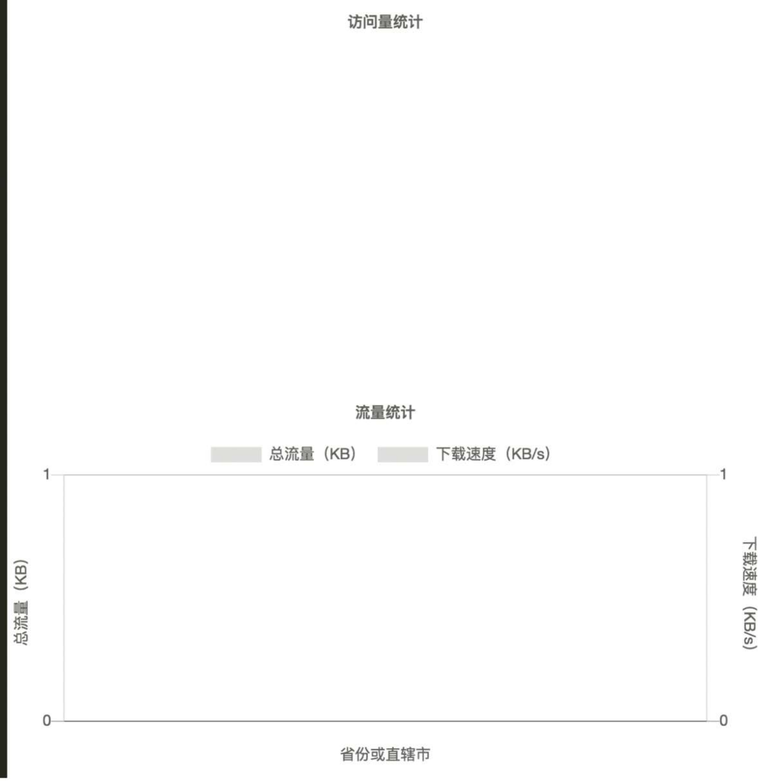
直播85分钟

博客两篇

《必修课！一小时吃透PyFlink》

《PyFlink实现CDN日志实时分析》

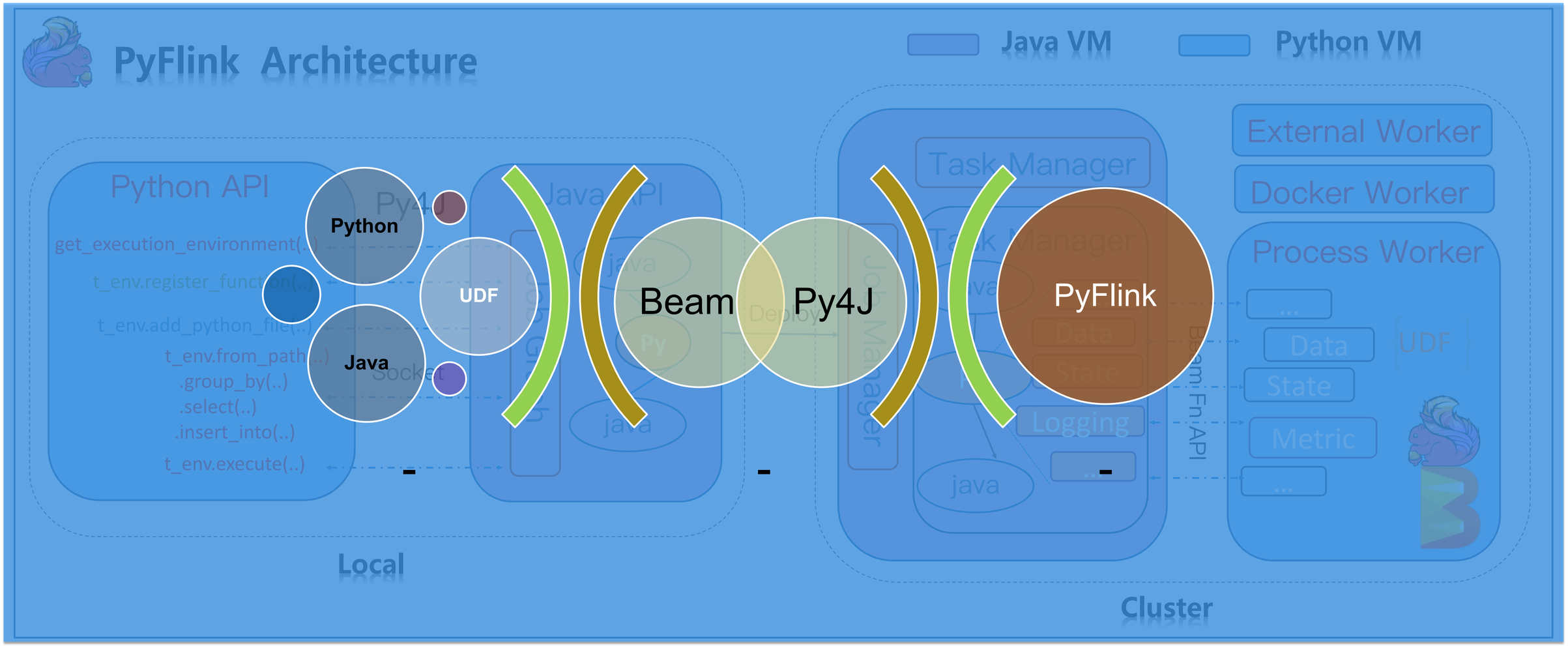
```python
import os

from pyflink.datastream import StreamExecutionEnvironment
from pyflink.table import StreamTableEnvironment, EnvironmentSettings
from enjoyment.cdn.cdn_udf import ip_to_province
from enjoyment.cdn.cdn_connector_ddl import kafka_source_ddl, mysql_sink_ddl

# 创建Table Environment,  并选择使用的Planner
env = StreamExecutionEnvironment.get_execution_environment()
t_env = StreamTableEnvironment.create(
    env,
    environment_settings=EnvironmentSettings.new_instance().use_blink_planner().build())

# 创建Kafka数据源表
t_env.sql_update(kafka_source_ddl)
# 创建MySql结果表
t_env.sql_update(mysql_sink_ddl)

# 注册IP转换地区名称的UDF
t_env.register_function("ip_to_province", ip_to_province)

# 添加依赖的Python文件
t_env.add_python_file(
    os.path.dirname(os.path.abspath(__file__)) + "/enjoyment/cdn/cdn_udf.py")
t_env.add_python_file(os.path.dirname(
    os.path.abspath(__file__)) + "/enjoyment/cdn/cdn_connector_ddl.py")

# 核心的统计逻辑
t_env.from_path("cdn_access_log")\
    .select("uuid, "
            "ip_to_province(client_ip) as province, " # IP 转换为地区名称
            "response_size, request_time")\
    .group_by("province")\
    .select( # 计算访问量
            "province, count(uuid) as access_count, "
            # 计算下载总量
            "sum(response_size) as total_download,  "
            # 计算下载速度
            "sum(response_size) * 1.0 / sum(request_time) as download_speed") \
    .insert_into("cdn_access_statistic")

# 执行作业
t_env.execute("pyflink_parse_cdn_log")
```



Kafka Message: 5934d8ae-d66f-4095-a01c-68a7ea8701a6,36.24.191.206,260,150280,https://www.aliyun.com/product/bastionhost?spm=5176.224200.h2v3icoap.119.1f956ed6muyb0V&aly_as=96NleqHS

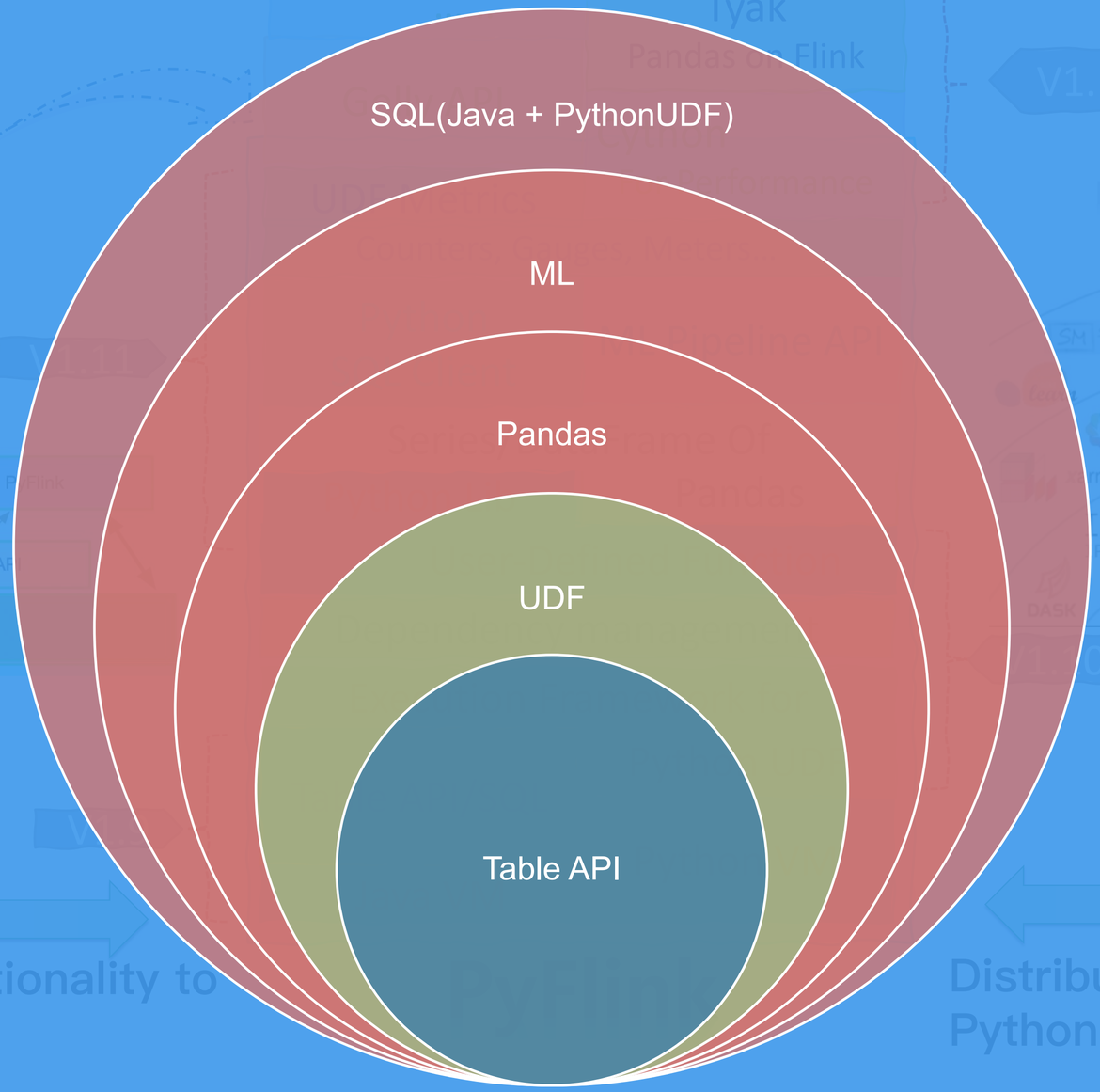
Random | Send | Clear and Truncate Mysql

PyFlink 架构


PyFlink Architecture
Java VM
Python VM
Python API
get_execution_environment(..)
t_env.register_function(..)
t_env.add_python_file(..)
t_env.from_path(..)
.group_by(..)
.select(..)
.insert_into(..)
t_env.execute(..)
Python
Java
Py4J
Socket
UDF
Beam
Py4J
Deploy
Job Manager
Task Manager
PyFlink
Data
State
Logging
Beam Fn API
External Worker
Docker Worker
Process Worker
Data
State
Metric
UDF
Local
Cluster

PyFlink 规划

Alibaba Cloud | Worldwide Cloud Services Partner



Flink 1.11

Flink 1.10

Flink 1.9

Expose Flink functionality to Python users

Distributed functionality of Python through Flink

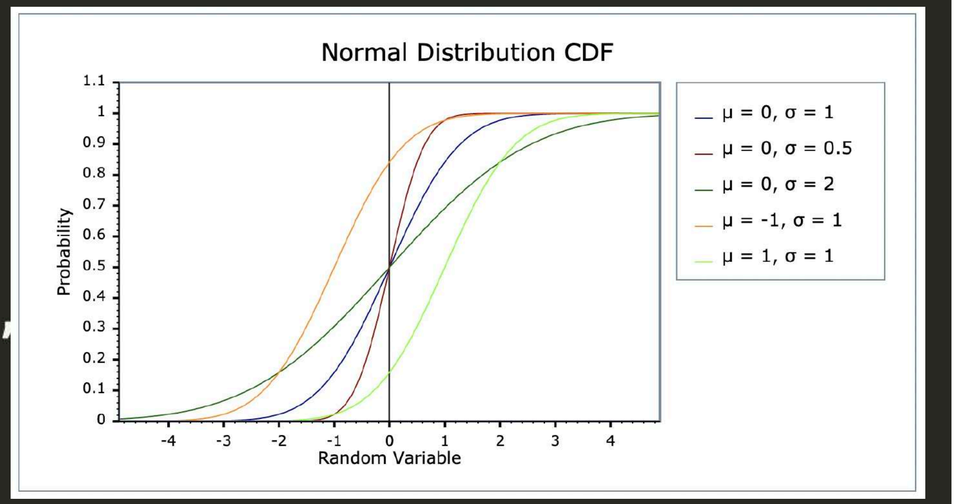
PyFlink支持PandasUDF

Alibaba Cloud | Worldwide Cloud Services Partner

Functionality → Performance → Usability

Pandas UDF

```python
import pandas as pd
from scipy import stats
from pyflink.table import DataTypes
from pyflink.table.udf import udf


@udf(input_types=[DataTypes.DOUBLE()],
     result_type=DataTypes.DOUBLE(),
     udf_type="pandas")
def cdf(v):
    return pd.Series(stats.norm.cdf(v))

```



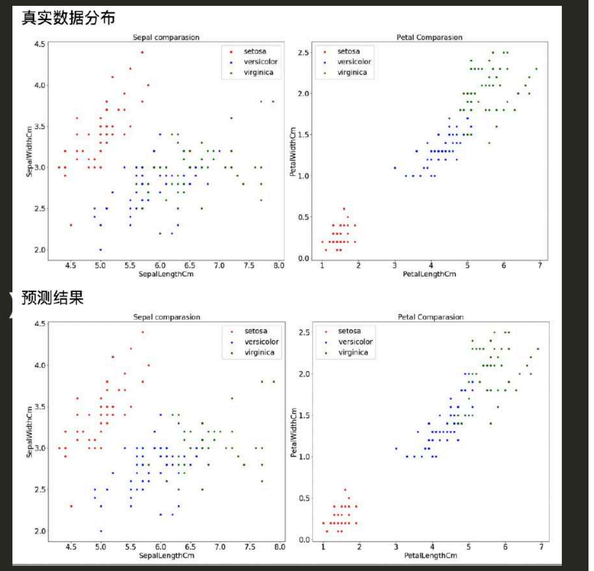
PyFlink支持ML

Alibaba Cloud | Worldwide Cloud Services Partner

Functionality → Performance → Usability

ML Pipeline API

```python
# transformer
va = VectorAssembler()\
    .set_selected_cols(["sepal_length",
        "sepal_width", "petal_length", "petal_width"])\
    .set_output_col("features")

# estimator
kmeans = KMeans() \
    .set_vector_col("features") \
    .set_k(3) \
    .set_reserved_cols(["sepal_length",
        "sepal_width", "petal_length", "petal_width", "category"])
    .set_prediction_col("prediction_result")

# pipeline
pipeline = Pipeline().append_stage(va).append_stage(kmeans)
pipeline \
    .fit(t_env, sourceTable) \
    .transform(t_env, sourceTable) \
    .insert_into('kmeansResults')

t_env.execute('KmeansTest')
```



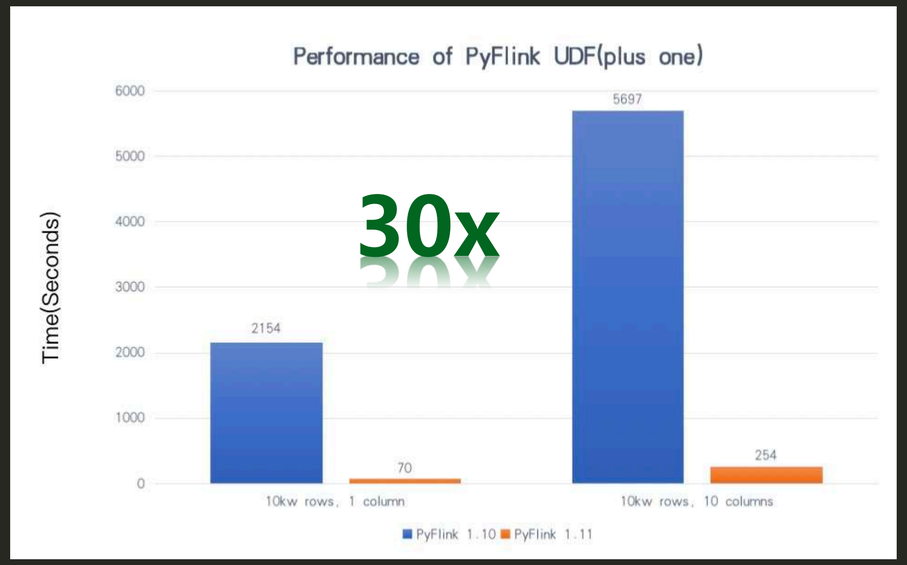
PyFlink性能优化

Alibaba Cloud | Worldwide Cloud Services Partner

Functionality → **Performance** → Usability

High Performance

1.Code gen Python UDF
2. Add support for Cython
3. Optimizing Beam WindowValuedCoder
4. Replacing Flink row with Python list
5. …



PyFlink支持Java SQL + Python UDF

Alibaba Cloud | Worldwide Cloud Services Partner

Functionality → Performance → Usability

PyUDF in SQL Client

```
Syntax:

CREATE
[TEMPORARY | TEMPORARY SYSTEM] FUNCTION
[IF NOT EXISTS]
[catalog_name.db_name.]function_name AS identifier
LANGUAGE PYTHON;

Example:
CREATE TEMPORARY SYSTEM FUNCTION func1 AS 'pymodule.udf.func1' LANGUAGE PYTHON;
```

```
Welcome! Enter 'HELP;' to list all available commands. 'QUIT;' to exit.

Flink SQL> CREATE TEMPORARY SYSTEM FUNCTION func1 AS 'pymodule.udf.func1' LANGUAGE PYTHON;
> SELECT func1(f0) FROM tab1;
```

起点的选择？环境 vs 功能

起点的选择？环境

Alibaba Cloud | Worldwide Cloud Services Partner

pip install apache-flink

FFSF — -bash — 80×24

```
jincheng:FFSF jincheng.sunjc$
```

起点的选择？环境

Alibaba Cloud | Worldwide Cloud Services Partner

在本地尝试以pre-job模式部署作业时，发现会提示如下报错，导致任务提交失败

RuntimeError: Python versions prior to 3.5 are not supported for PyFlink [sys.version_info(major=2, minor=7, micro=16, releaselevel='final', serial=0)].

[libprotobuf FATAL google/protobuf/... r.cc:1370] CHECK failed: GeneratedDatabase()->Add(encoded_file_descriptor, si...

libc++abi.dylib: terminating wi... ion:

CHECK failed: GeneratedDa...

java.lang.NoClassDefF... s.

PipelineOptionsFactory

at org.apache.flink.p... onFunn

java:173)

at org.apache.flink.tabl... perator

Apache Flink 扫雷系列 – PyFlink如何解决多JAR包依赖问题

Posted on 2020-03-31 | Edited on 2020-04-01 | In Apache Flink 扫雷系列 | Comments: | Views: 107

Python 版本问题

macos 10.15 (Catalina) 引起的

这些问题都可以解决！

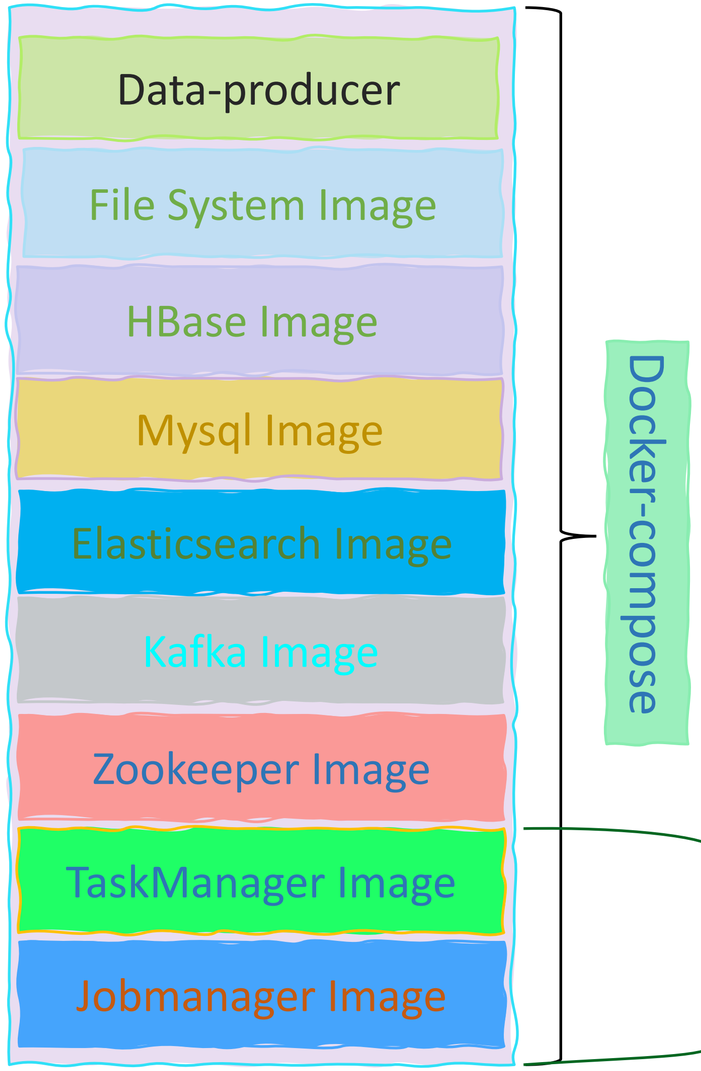
但没必要作为学者的起点！

多Connector依赖问题

Apache Flink

起点的选择？功能

Alibaba Cloud | Worldwide Cloud Services Partner



docker-compose.yml

```yaml
version: '2.1'
services:
  jobmanager:
    image: pyflink/playgrounds:1.10.0
    volumes:
      - ./examples:/opt/examples
    hostname: "jobmanager"
    expose: <1 item>
    ports: <1 item>
    command: jobmanager
    environment: <1 item>
  taskmanager: <7 keys>
  zookeeper:
    image: wurstmeister/zookeeper:3.4.6
    ports: <1 item>
  kafka:
    image: wurstmeister/kafka:2.12-2.2.1
    ports: <1 item>
    depends_on: <1 item>
    environment: <3 keys>
    volumes: <1 item>
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch-oss:6.3.1
    environment: <4 items>
    ports: <2 items>
    ulimits: <2 keys>
  data-producer:
    image: fhueske/flink-sql-training:1-FLINK-1.9-scala_2.11
    command: "java -classpath /opt/data/data-producer.jar com.ververica.sql_training.data_producer
    depends_on: <3 items>
    environment: <4 keys>
  db:
    image: mysql
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    ports:
      - 3306:3306
    environment: <2 keys>
    volumes: <1 item>
  adminer:
    image: adminer
    restart: always
    ports:
      - 8080:8080
```

① ② ③ ④ ⑤

FileSystem， HBase, Hive ...

Docker-compose.yml

Tags

This repository contains 2 tag(s).

1.10.0	an hour ago
1.11-snapshot	7 days ago

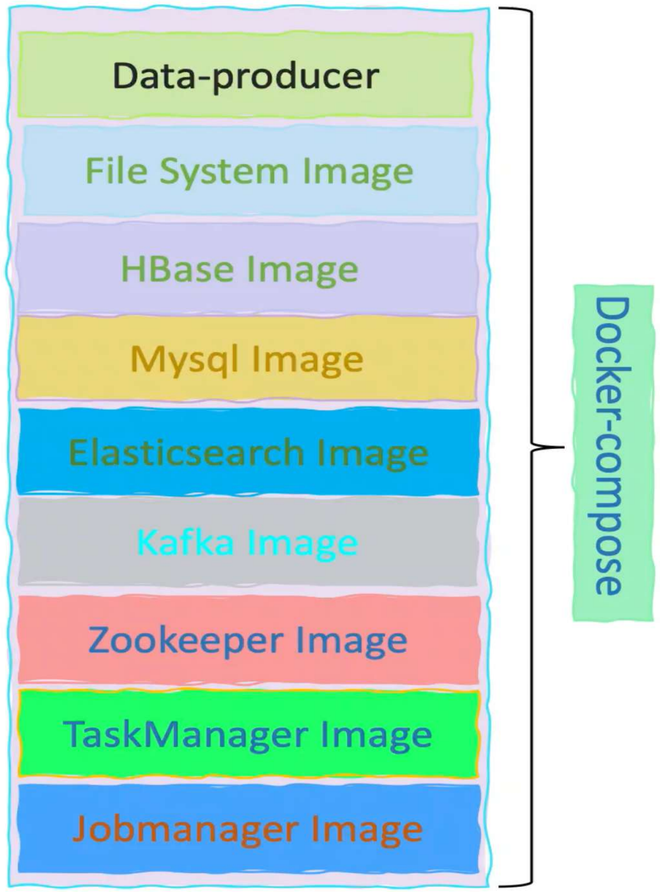
See all

DockerHub

起点的选择？功能

Starting point of PyFlink? Start with Functionality

Alibaba Cloud | Worldwide Cloud Services Partner



1. Install Docker : https://www.docker.com/
2. Config the image of PyFlink in Docker-compose.yml
3. Download and start-up: docker-compose up -d
4. Config volumes: ./examples:/opt/examples
5. Write and Submit PyFlink Job …
6. Check the Result
   and multiple modifications and attempts
7. Read PyFlink API Doc for creative development

Apache Flink

起点的选择？功能

Alibaba Cloud | Worldwide Cloud Services Partner

```
Table(object)
  __init__(self, j_table)
  __str__(self)
  _j_table
  add_columns(self, fields)
  add_or_replace_columns(self, fields)
  alias(self, fields)
  distinct(self)
  drop_columns(self, fields)
  fetch(self, fetch)
  filter(self, predicate)
  full_outer_join(self, right, join_predicate)
  get_schema(self)
  group_by(self, fields)
  insert_into(self, table_path)
  intersect(self, right)
  intersect_all(self, right)
  join(self, right, join_predicate=None)
  join_lateral(self, table_function_call, join_predicate=None)
  left_outer_join(self, right, join_predicate=None)
  left_outer_join_lateral(self, table_function_call, join_predicate=None)
  minus(self, right)
  minus_all(self, right)
  offset(self, offset)
  order_by(self, fields)
  over_window(self, *over_windows)
  print_schema(self)
  rename_columns(self, fields)
  right_outer_join(self, right, join_predicate)
  select(self, fields)
  union(self, right)
  union_all(self, right)
  where(self, predicate)
  window(self, window)
```

- select
- as
- filter
- where
- group_by
- distinct
- join(inner, left, right, full)
- join_lateral(inner, left)
- minus
- minus_all
- union
- union_all
- window(window)
- add_or_replace_columns(..)
- drop_columns(..)
- add_columns(..)
- rename_columns(..)
- ...

- OverWindow
- SessionWindow
- TumblingWindow
- SlidingWindow

Powerful than SQL

博客

https://enjoyment.cool/

Stateful computations over streams

Unified programming model

Python & Java & Scala

github.com/pyflink/playgrounds

r jump to... /

pyflink / playgrounds

推荐阅读

005Three Min Series - How to create UDF in PyFlink 1.10

004Three Min Series - How to using PyFlink Shell

Deep dive how to support Python UDF in Apache Flink 1.10

003Three Min Series - Three Min Series - How PyFlink does ETL

002Three Min Series - Run the Example of WordCount in PyFlink 1.9

001Three Min Series - Setting up the dev environment for Pyflink 1.9

Apache Flink 说道系列 - PyFlink 作业的多种部署模式

Apache Flink 说道系列 - PyFlink 1.11 功能规划讨论, 有你的声音吗?

Apache Flink 说道系列 - 如何在PyFlink 1.10中自定义Python UDF

Apache Flink 说道系列- Python API 中如何使用 Java 自定义 Connector

Apache Flink 说道系列- Python API 中如何使用 Kafka

Apache Flink 说道系列- 如何在IDE中运行使用Java UDFs 的Python 作业

Apache Flink 视频系列 - 2019.08.06直播(德国柏林)

Apache Flink 说道系列- Python Table API 开发环境搭建

Apache Flink 说道系列- Python API 时代的产物

推荐阅读

Apache Flink 漫谈系列 - 序

Apache Flink 漫谈系列 - 概述

Apache Flink 漫谈系列 - Watermark

Apache Flink 漫谈系列 - State

Apache Flink 漫谈系列 - Fault Tolerance

Apache Flink 漫谈系列 - 流表对偶(duality)性

Apache Flink 漫谈系列 - 持续查询(Continuous Queries)

Apache Flink 漫谈系列 - SQL概览

Apache Flink 漫谈系列 - JOIN 算子

Apache Flink 漫谈系列 - JOIN LATERAL

Apache Flink 漫谈系列- Temporal Table JOIN

Apache Flink 漫谈系列 - Time Interval(Time-windowed) JOIN

Apache Flink 漫谈系列 - Table API 概述

Apache Flink 漫谈系列 - DataStream Connectors之Kafka

Agenda

① Why PyFlink

② Status of PyFlink

③ Future of PyFlink

④ Flink on Zeppelin

Who is Jeff Zhang

AUTODESK.

ebay

Pivotal™

HORTONWORKS®

Alibaba.com™

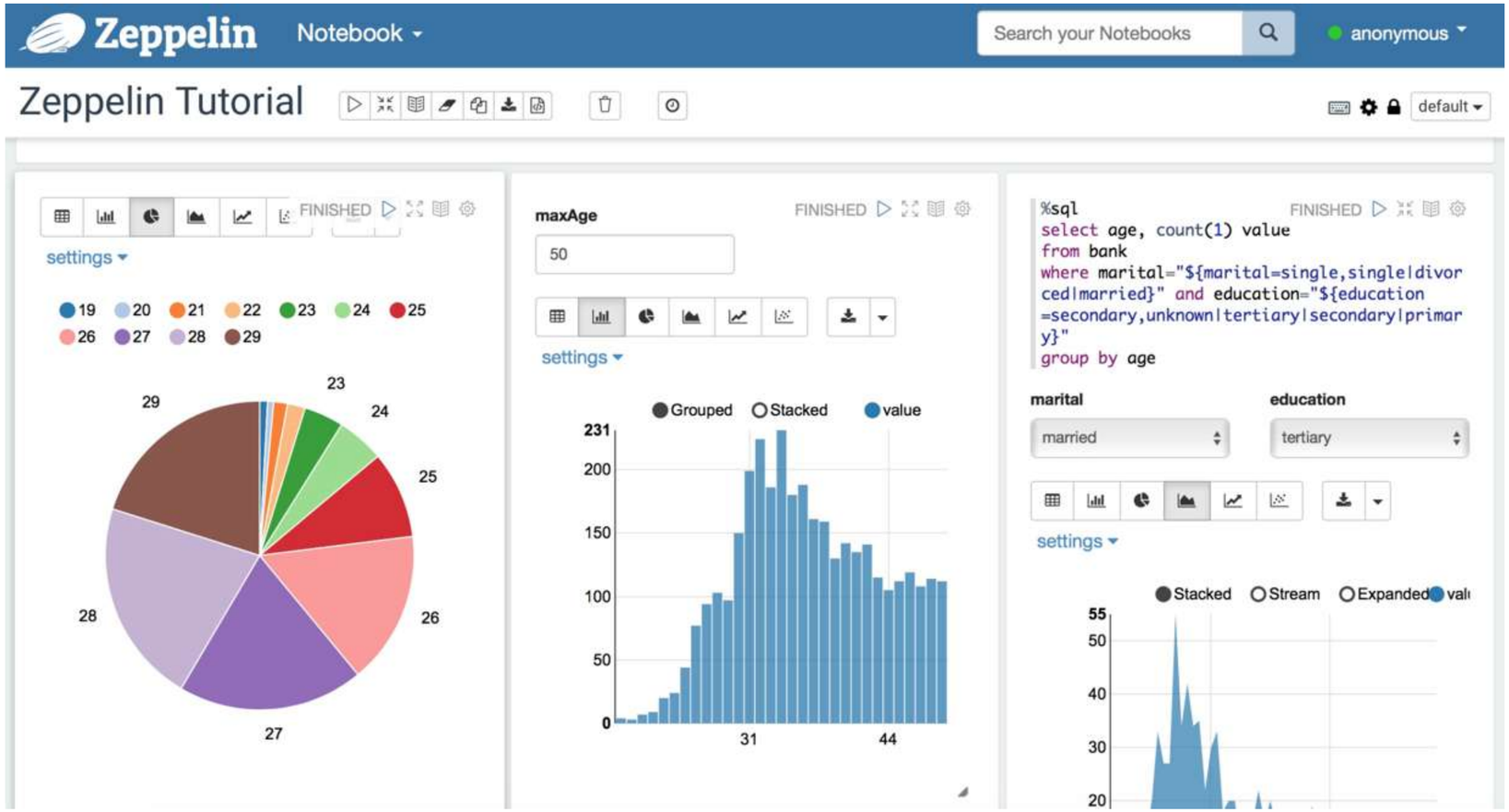
Data

APACHE

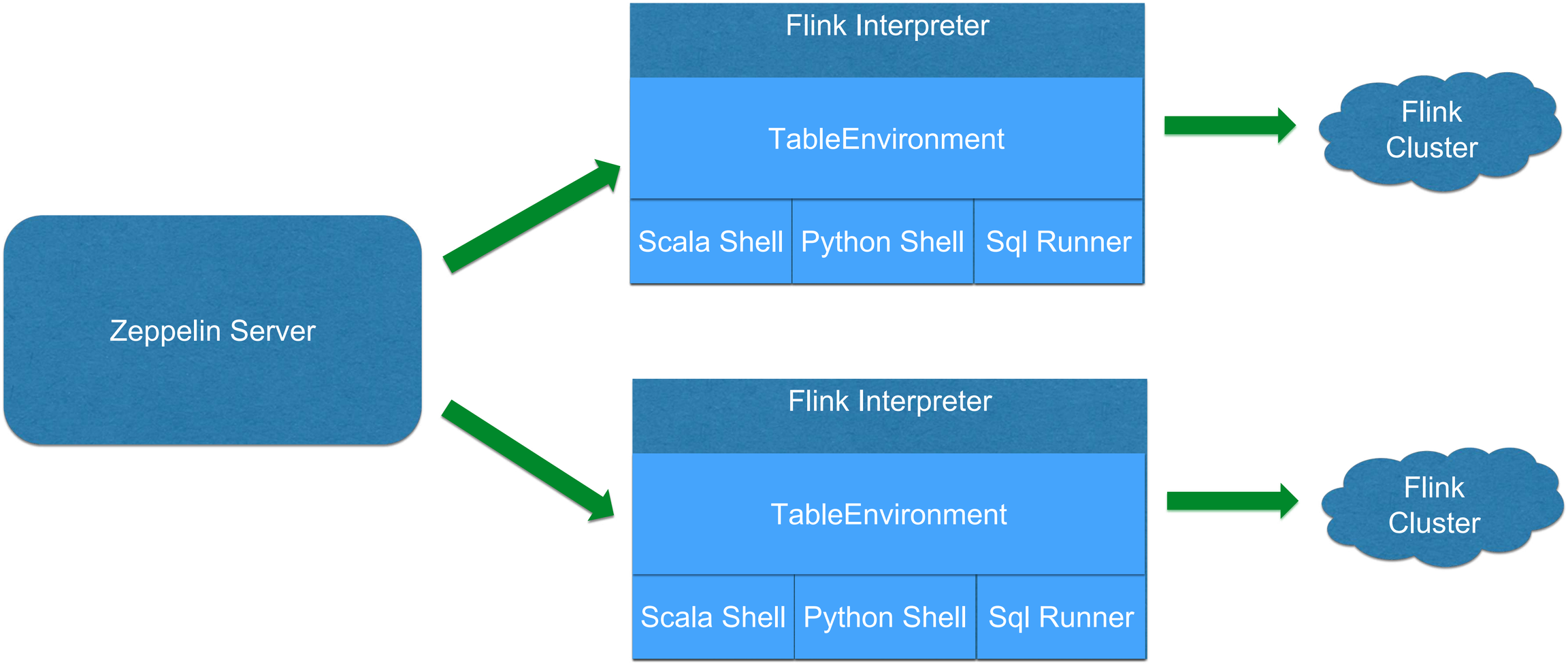
What is Apache Zeppelin

Alibaba Cloud | Worldwide Cloud Services Partner

Web-based notebook that enables data-driven, interactive data analytics and collaborative documents with SQL, Scala, Python and more.



Flink on Zeppelin (Architecture)



Flink on Zeppelin (Features)

Alibaba Cloud | Worldwide Cloud Services Partner

Multiple Language Support
Scala/Python/Sql

Hive Integration

Dependency Management

SQL Result Visualization/Preview

Job Level Configuration

Interactive job control

Multiple thread job submission

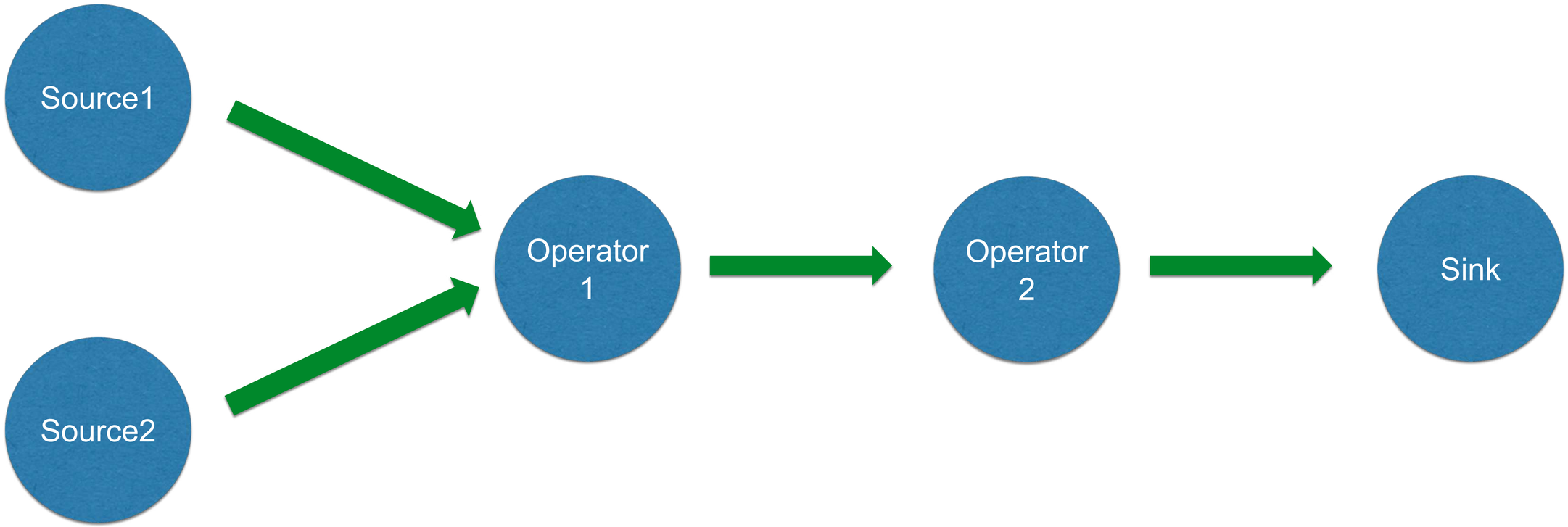
Machine Learning Support

3 essential elements of Flink job

Step 1. Define Source/Sink

Step 2. Build DAG

Step 3. Business Logic



Demo

Materials

- http://zeppelin.apache.org/

- Flink on Zeppelin tutorials
  1. Get started https://link.medium.com/oppqD6dIg5
  2. Batch https://link.medium.com/3qumbwRIg5
  3. Streaming https://link.medium.com/RBHa2lTIg5
  4. Advanced usage https://link.medium.com/CAekyoXIg5

- 钉钉群

【外部】Flink on Zeppelin...
3人

扫一扫群二维码，立刻加入该群。

【外部】Flink on Zeppelin...
786人

扫一扫群二维码，立刻加入该群。

Alibaba Cloud | Worldwide Cloud Services Partner

WWW.ALIBABA CLOUD.COM