

www.vip.com

第三次上海大数据流处理Meetup

2016年7月2日

A large pink circle on the right side of the slide, containing the Vip.com logo and tagline in white text.

唯品会
vip.com
一家专门做特卖的网站

Agenda

12:30 - 13:00 签到/Sign in

13:00 - 13:05 开场白/Opening

13:05 - 13:45 大数据流处理最新社区动态

13:45 - 14:30 Spark Streaming 在唯品会的实践

14:30 - 14:50 茶歇/Tea break

14:50 - 15:30 Redis Cluster在唯品会大数据业务应用

15:30 - 16:15 Apache Beam

16:15 – 17:30 交流时间/Networking time

www.vip.com

大数据流处理最新社区 动态

姜伟华

唯品会
vip.com
一家专门做特卖的网站

2016年流处理社区动态



Storm 1.0



Spark 2.0
(coming)



Kafka 0.10



Flink 1.0



Gearpump
Incubation



Apache
Beam
Incubation



Heron Open
Source

总的观察

流处理领域还没有一个王者出现，诸侯混战

- 目前的巅峰对决是Spark Streaming vs. Beam+Flink
- Storm持续演进，力保基本盘

各个框架的趋同性越来越高

- Storm 1.0和Heron
- Spark 2.X和Beam

Streaming SQL逐渐成为标配

技术深度急剧提高

- End-2-end exactly once成为新的标杆
- 强调批流统一处理
- CEP等DSL的出现

Storm 1.0

于2016.4.12发布，许多新功能！！

性能提高：大部分应用可以有3X的提高。最高：吞吐量16X，延迟降低60%

可编程性：

Distributed Cache

- 支持GB大小的Blob
- 可以用于存放模型、字典等
- 实现在线升级

Native Window Support

- 支持sliding 和 Tumbling Window
- 可以按照时间或者事件数来划分

状态管理

- 支持状态的自动 checkpoint
目前支持内存和 Redis Checkpoint

Storm 1.0: Streaming SQL

很早期，目前还不支持Aggregation, Windowing, Join

```
CREATE EXTERNAL TABLE ORDERS (ID INT PRIMARY KEY,  
    UNIT_PRICE INT, QUANTITY INT) LOCATION  
    'kafka://localhost:2181/brokers?topic=orders' TBLPROPERTIES  
    '...'
```

```
CREATE EXTERNAL TABLE LARGE_ORDERS (ID INT PRIMARY  
    KEY, TOTAL INT) LOCATION  
    'kafka://localhost:2181/brokers?topic=large_orders'  
    TBLPROPERTIE '...'
```

```
INSERT INTO LARGE_ORDERS SELECT ID, UNIT_PRICE *  
    QUANTITY AS TOTAL FROM ORDERS WHERE UNIT_PRICE *  
    QUANTITY > 50
```

Storm 1.0 : 系统管理

扩展性：引入Pacemaker

- 记录worker心跳，避免ZK瓶颈
- 可以支持几千个节点

可靠性：Nimbus HA

- 多个nimbus node可以随时加入或退出，会自动选举

自动反压（Backpressure）

- 根据Task的buffer size确定高低水位，自动让Spout限流

Resource Aware Scheduler

- 根据内存和CPU

长期运行支持

- 动态Log级别
- 分布式Log查询：查询一个topology的所有日志
- 动态Worker profiling

Heron

2016年5月26日正式
宣布open source

特性：

- 兼容Storm
- 更好的性能
- 更好的扩展性
- 更好的可调试性和Profiling
- 反压
- 方便部署和管理

目标：更好的Storm，支持大集群（几千台）

Kafka 0.10

2016年5月24日发布

Kafka Streams

- 一种新的流处理平台
- 嵌入式库，而不是平台
- 两个概念：Stream和Table
- 和Kafka深度整合

机架感知

- 保证replica跨机架，防止整个机架出问题

消息时间戳

- 为每个消息在加入Kafka时自动打上时间戳

Kafka Connect

- 支持配置式的创建Data ETL Pipeline。例如，将数据写入Kafka
- 支持Standalone, YARN, Mesos, Kubernetes
- 0.10支持Kafka Connect的状态/控制 RESTful API

其他改进

- Broker支持协议版本列表
- Consumer poll() 支持最大记录条数

Kafka 0.10: Kafka Streams例子: Word Count

```
KStreamBuilder builder = new KStreamBuilder();  
KStream<String, String> textLines = builder.stream(..., "TextLinesTopic");  
  
KStream<String, Long> wordCounts = textLines  
    .flatMapValues(value -> Arrays.asList(value.split("\\W+")))  
    .map((key, vaue) -> new KeyValue<>(value, value))  
    .countByKey("Counts") .toStream();  
  
wordCounts.to("WordsWithCountsTopic",...);  
  
KafkaStreams streams = new KafkaSteams(builder, config);  
streams.start();
```

Kafka 0.10: Kafka Streams 特点

- 非常小：（9K LOC）
- 嵌入式，嵌入到应用程序中
- At-least Once语义
- 输入输出都是Kafka Topics
- 复用Kafka的功能
 - 数据模型
 - Partitioner
 - Group membership (管理instance的任务指派、生存期、Partition等)
 - Table和其他状态计算复用Kafka的Log Compacted Topics
 - Metrics
- App的位置是由Kafka的consumer offset来确定的
- 使用Kafka 0.10的timestamp功能来进行event-time处理

API稳定性

- 保证1.X系列API通用

新增State存储后端：RocksDB

- 一个嵌入式KV Store DB，可高效存储超过内存大小的数据
- 可持久化到HDFS，S3

Savepoint

- 一种可以手工创建的checkpoint
- 可以用于：版本升级、Flink升级、集群维护、What-if模拟、A/B测试等等应用场景

Complex Event Processing库

增强的监控：任务提交、checkpoint统计、反压

增强的Checkpoint控制：防止checkpoint积压

Kafka 0.9支持

Flink 1.0: Complex Event Processing

```
val env : StreamExecutionEnvironment = ...
env.setStreamTimeCharacteristic(TimeCharacteristic.EventTime)

val input : DataStream[Event] = ...

val partitionedInput = input.keyBy(event => event.getId)

val pattern = Pattern.begin("start")
    .next("middle").where(_.getName == "error")
    .followedBy("end").where(_.getName == "critical")
    .within(Time.seconds(10))

val patternStream = CEP.pattern(partitionedInput, pattern)

val alerts = patternStream.select(createAlert(_))
```

Apache Beam

原来是Google Dataflow SDK
刚发布了0.1.0版本

一种流/批处理统一建模工具

- 很高大上
- 改变人生观、世界观 😊

Runner支持：

1. Google Dataflow
2. Flink
3. Spark
4. Gearpump (开发中)
5. Storm (计划)

Gearpump Incubation

2016年3月进入Apache Incubation

Incubation Champion: Andrew Purtell

目前已经完成:













1. 代码清理
2. 版权保护
3. 包名替换
4. 网站建设
5. LOGO
6. ...

即将发布进入Apache Incubation后的第一个版本
一个主要工作是支持Apache Beam

Spark Streaming 2.0: Structured Streaming



Spark 2.x Streaming.pdf

												
	Flume	NiFi	Gearpump	Apex	Kafka Streams	Spark Streaming	Storm	Storm + Trident	Samza	Flink	Ignite Streaming	Beam (*GC DataFlow)
Current version	1.6.0	0.6.1	incubating	3.3.0	0.10.0.0	1.6.1	1.0.0	1.0.0	0.10.0	1.0.2	1.5.0	incubating
Category	DC/SEP	DC/SEP	SEP	DC/ESP	ESP	ESP	ESP/CEP	ESP/CEP	ESP	ESP/CEP	ESP/CEP	SDK
Event size	single	single	single	single	single	micro-batch	single	mini-batch	single	single	single	single
Available since (incubator since)	June 2012 (June 2011)	July 2015 (Nov 2014)	(Mar 2016)	Apr 2016 (Aug 2015)	May 2016 (July 2011)	Feb 2014 (2013)	Sep 2014 (Sep 2013)	Sep 2014 (Sep 2013)	Jan 2014 (July 2013)	Dec 2014 (Mar 2014)	Sep 2015 (Oct 2014)	(Feb 2016)
Contributors	26	78	19	53	183	891	215	215	54	184	65	82
Main backers	Apple Cloudera	Hortonworks	Intel Lightbend	Data Torrent	Confluent	AMPLab Databricks	Backtype Twitter	Backtype Twitter	LinkedIn	dataArtisans	GridGain	Google
Delivery guarantees	at least once	at least once	exactly once at least once (with non-fault-tolerant sources)	exactly once	at least once	exactly once at least once (with non-fault-tolerant sources)	at least once	exactly once	at least once	exactly once	at least once	exactly once*
State management	transactional updates	local and distributed snapshots	checkpoints	checkpoints	local and distributed snapshots	checkpoints	record acknowledgements	record acknowledgements	local snapshots distributed snapshots (fault-tolerant)	distributed snapshots	checkpoints	transactional updates*
Fault tolerance	yes (with file channel only)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes*
Out-of-order processing	no	no	yes	no	yes	no	yes	yes	yes (but not within a single partition)	yes	yes	yes*
Event prioritization	no	yes	programmable	programmable	programmable	programmable	programmable	programmable	yes	programmable	programmable	programmable
Windowing	no	no	time-based	time-based	time-based	time-based	time-based	time-based	time-based	time-based	time-based	time-based
Back-pressure	no	yes	yes	yes	N/A	yes	yes	yes	yes	yes	yes	yes*
Primary abstraction	Event	FlowFile	Message	Tuple	KafkaStream	DStream	Tuple	TridentTuple	Message	DataStream	IgniteDataStreamer	PCollection
Data flow	agent	flow (process group)	streaming application	streaming application	process topology	application	topology	topology	job	streaming dataflow	job	pipeline
Latency	low	configurable	very low	very low	very low	medium	very low	medium	low	low (configurable)	very low	low*
Resource management	native	native	YARN	YARN	Any process manager (e.g. YARN, Mesos, Chef, Puppet, Salt, Kubernetes, ...)	YARN Mesos	YARN Mesos	YARN Mesos	YARN	YARN	YARN Mesos	integrated*
Auto-scaling	no	no	no	yes	yes	yes	no	no	no	no	no	yes*
In-flight modifications	no	yes	yes	yes	yes	no	yes (for resources)	yes (for resources)	no	no	no	no
API	declarative	compositional	declarative	declarative	declarative	declarative	compositional	compositional	compositional	declarative	declarative	declarative
Primarily written in	Java	Java	Scala	Java	Java	Scala	Clojure Scala Java	Java	Scala	Java	Java	Java
API languages	text files Java	REST (GUI)	Scala Java	Java	Java	Scala Java Python	Clojure Python Ruby Yahoo! Spotify Groupon	Java Python Scala	Java	Java Scala Python	Java .NET C++	Java*
Notable users	Meebo Sharethrough SimpleGeo	N/A	Intel Levi's Honeywell	Capital One GE Predix PubMatic	N/A	Kelkoo Localytics AsialInfo Opentable Faimdata Guavus	The Weather Channel Alibaba Baidu Yelp WebMD	Klout GumGum CrowdFlower	LinkedIn Netflix Intuit Uber	King Otto Group	GridGain	N/A

<https://databaseline.wordpress.com/2016/03/12/an-overview-of-apache-streaming-technologies/>



我们在招聘！

请联系： weihua.jiang@vipshop.com

www.vip.com