



Apache Flink

Flink在小米的应用与实践

夏军 · 小米 / 高级工程师

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

Contents

目录 >>

1

小米流计算演进

3

小米典型应用

2

小米相关改造

4

未来展望





Apache Flink

PART 01

小米流计算演进

小米流计算演进



Apache Flink



2014.02



2015.07

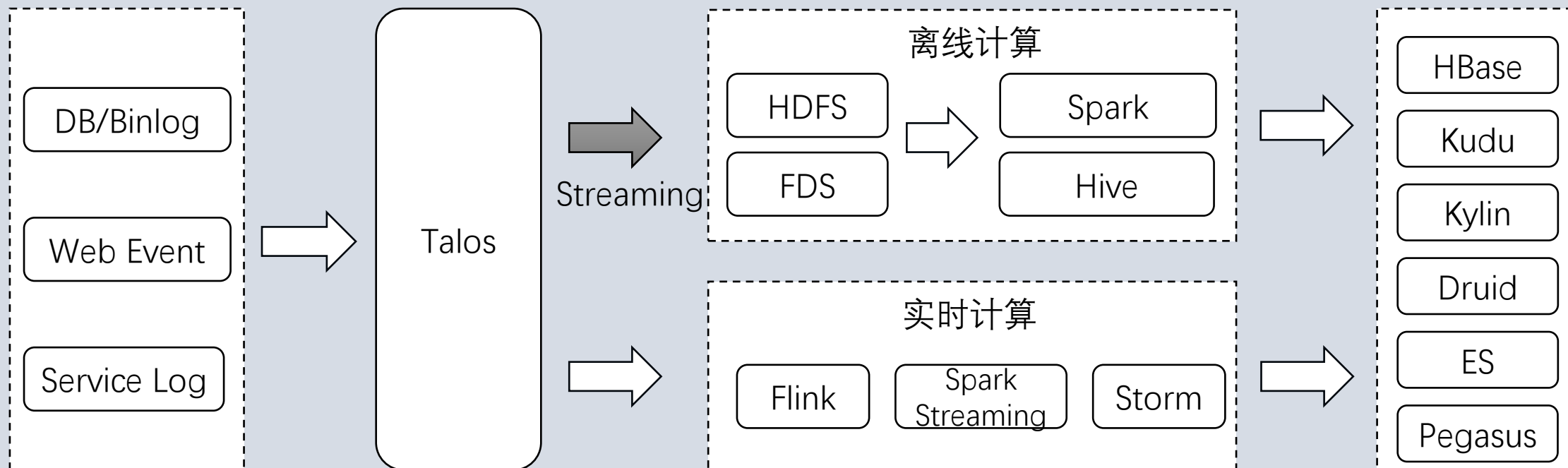


2019.01

基本架构



Apache Flink



集群概况



Apache Flink

7000亿

日处理数据

150+

作业规模

7w+

CPU

450TB +

内存

PART 02

小米相关改造



Talos简介

Talos是小米自研的消息队列，提供标准的Topic语义封装，用于实现业务上下游的数据解耦。

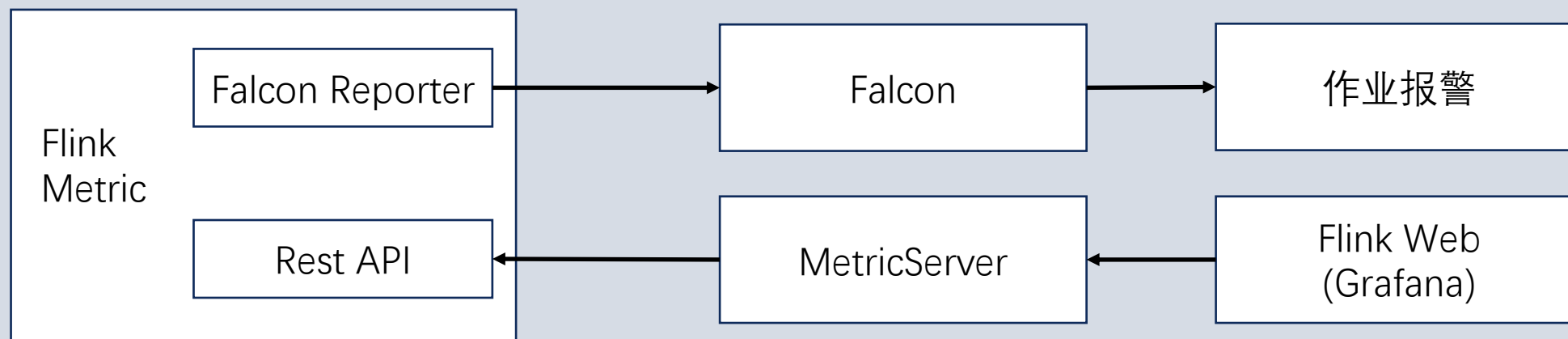
目前已经应用于小米业务的各个场景，也为Streaming计算提供数据源支持。

Flink Talos Source

- FlinkTalosConsumer ,
TalosTableSource
- 自动follow上游Partition变化
- 基于Thrift进行反序列化

Flink Talos Sink

- FlinkTalosProducer ,
TalosTableSink
- 默认Partitioner支持



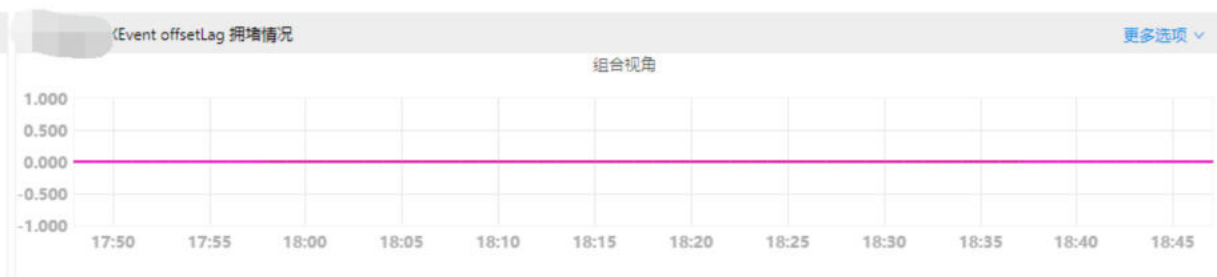
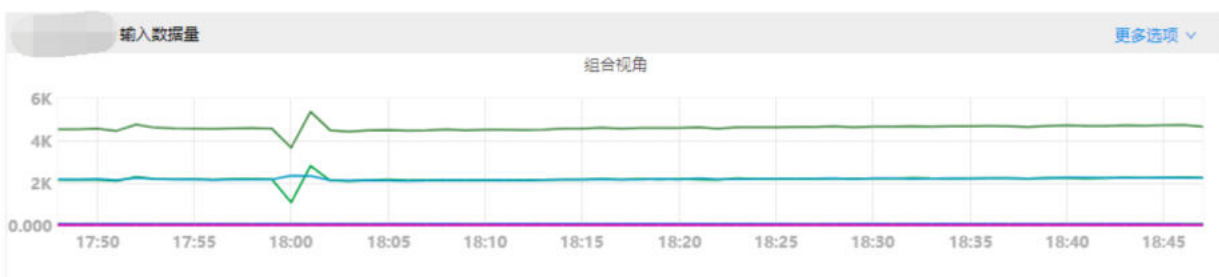
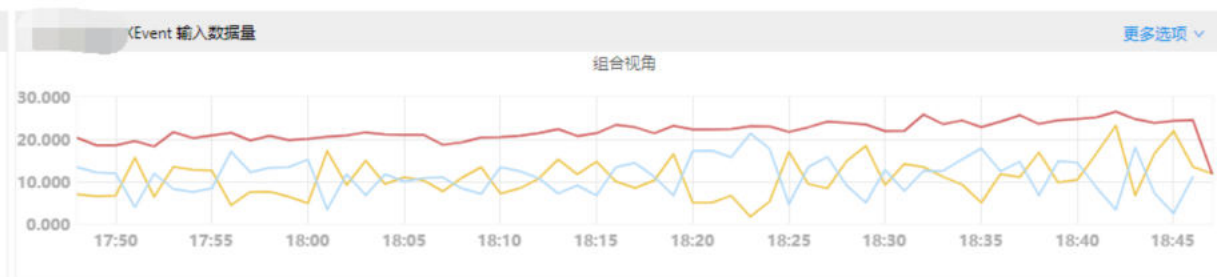
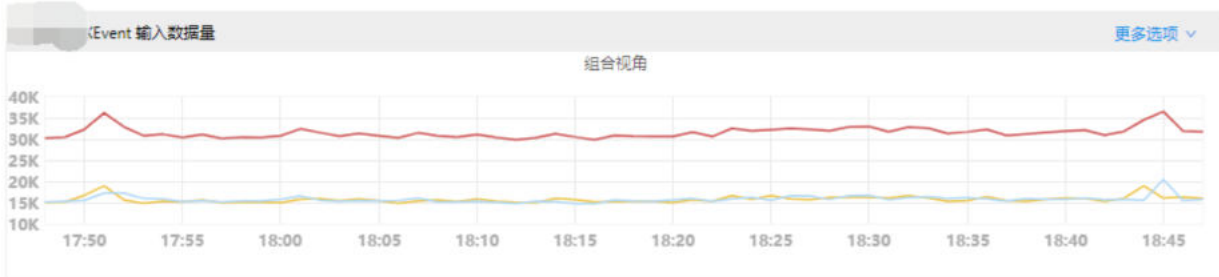
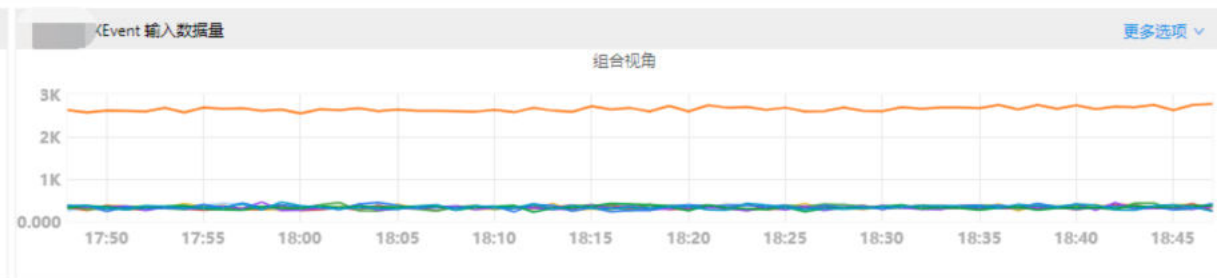
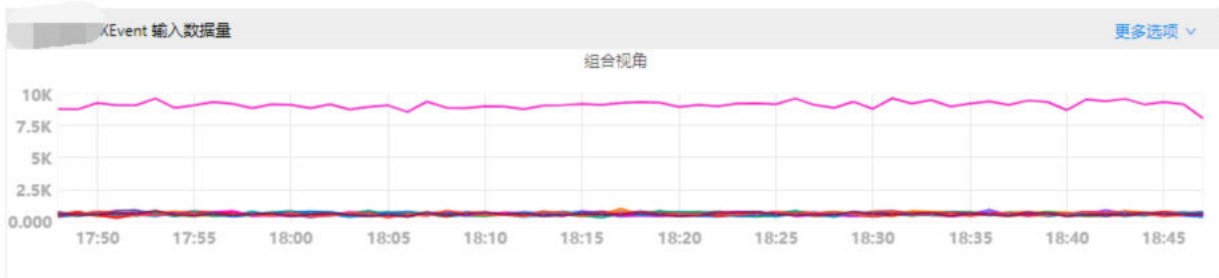
Falcon数据：通过定制化FalconReporter，将数据推动到Falcon实现数据监控与报警，主要包括消费积压、作业异常等需要报警的case。

Flink Web Metric数据展示：通过定制化MetricServer将Flink Metric数据转换为Json，并实现数据基于timestamp的缓存。FlinkWeb通过嵌入Grafana页面，实现Metric数据的定制化展示。主要包括 流量信息，checkpoint信息，jvm信息，gc信息等。

监控与报警



Apache Flink



监控与报警



Apache Flink

Apache Flink Dashboard

Overview Timeline Exceptions Configuration **Metrics**

Source: Custom Source -> Filter -> Map
Parallelism: 2

HA BH Sink: sinkToHistoryService
Parallelism: 2

Subtasks Task Metrics Watermarks Accumulators Checkpoints Back Pressure

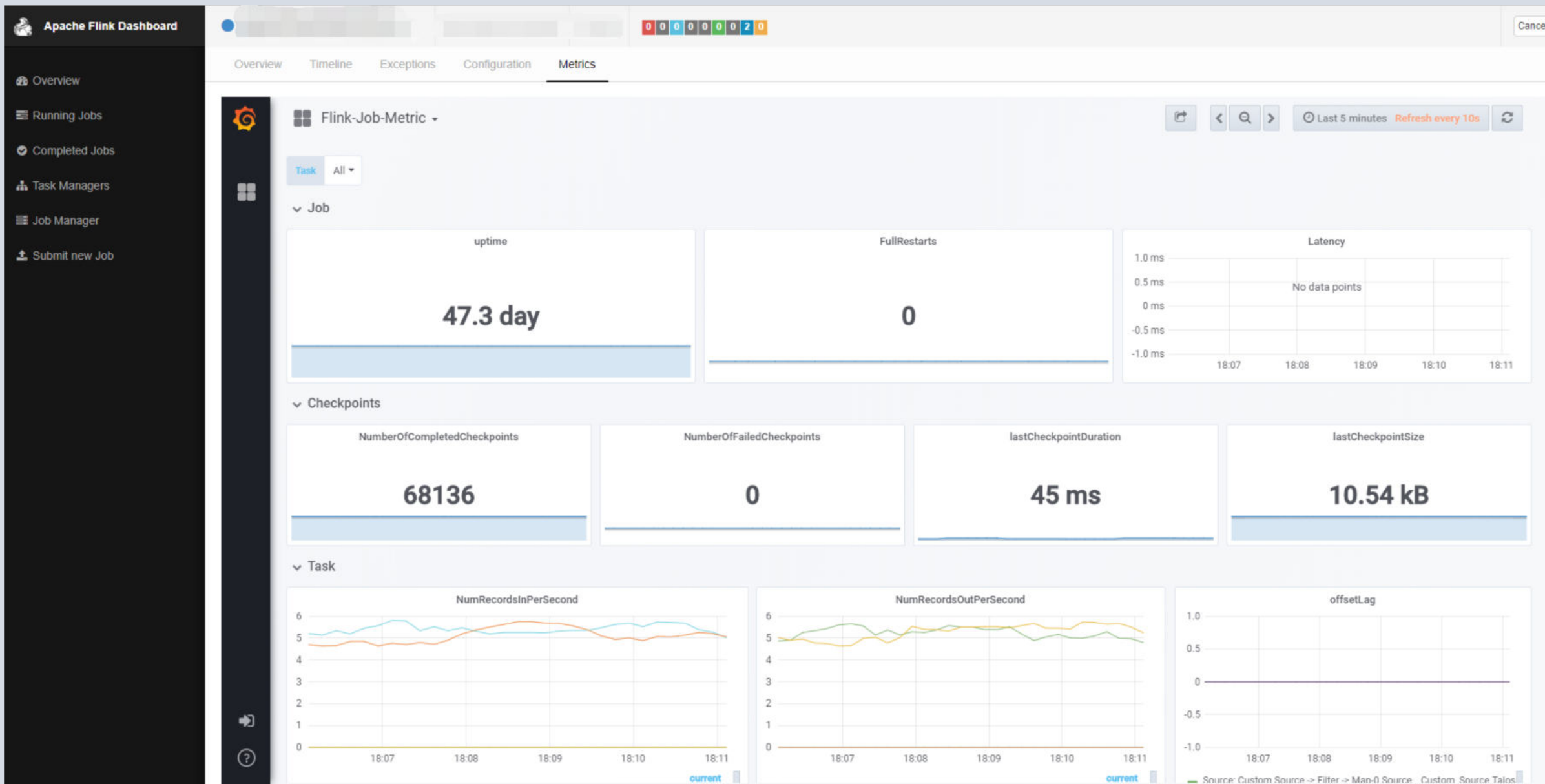
☐ Aggregate task statistics by TaskManager

Start Time	End Time	Duration	Name	Bytes received	Records received	Bytes sent	Records sent	Parallelism	Tasks	Status
				0 B	0	1.30 GB	18,721,593	2	0 0 2 0 0 0 0	RUNNING
				1.31 GB	18,721,593	0 B	0	2	0 0 2 0 0 0 0	RUNNING

监控与报警



Apache Flink



监控与报警



Apache Flink

Apache Flink Dashboard

Overview Timeline Exceptions Configuration Metrics

Flink-Job-Metric

2.0 18:27 18:28 18:29 18:30 18:31

Status.JVM.GarbageCollector.PS_MarkSweep.Count 3

200 ms 18:27 18:28 18:29 18:30 18:31

Status.JVM.GarbageCollector.PS_MarkSweep.Time 314 n

7548.5 18:27 18:28 18:29 18:30 18:31

Status.JVM.GarbageCollector.PS_Scavenge.Count 7551.0

1.17617 min 18:27 18:28 18:29 18:30 18:31

Status.JVM.GarbageCollector.PS_Scavenge.Time 1.17656

TaskManager

TaskManager-JVM-CPUload

0.300% 18:27 18:28 18:29 18:30 18:31

container_e12_1560936712321_14520_01_000003-Status.JVM.CPU Load 0.2576

container_e12_1560936712321_14520_01_000002-Status.JVM.CPU Load 0.2577

TaskManager-JVM-Heap

1.5 GB 1.0 GB 500 MB 0 B 18:27 18:28 18:29 18:30 18:31

container_e12_1560936712321_14520_01_000003-Status.JVM.Memory.Heap.Used

container_e12_1560936712321_14520_01_000002-Status.JVM.Memory.Heap.Used

TaskManagers-JVM-DirectMemoryUsed

150.715 MB 150.710 MB 150.705 MB 150.700 MB 150.695 MB 150.690 MB 150.685 MB 18:27 18:28 18:29 18:30 18:31

container_e12_1560936712321_14520_01_000003-Status.JVM.Memory.Direct.Memo

container_e12_1560936712321_14520_01_000002-Status.JVM.Memory.Direct.Memo

TaskManagers-JVM-GC-MarkSweep-Count

4.0 3.5 3.0 2.5 2.0 18:27 18:28 18:29 18:30 18:31

container_e12_1560936712321_14520_01_000003-Status.JVM

container_e12_1560936712321_14520_01_000002-Status.JVM

TaskManagers-JVM-GC-MarkSweep-Time

265 ms 260 ms 255 ms 250 ms 245 ms 240 ms 18:27 18:28 18:29 18:30 18:31

container_e12_1560936712321_14520_01_000003-Status.JVM

container_e12_1560936712321_14520_01_000002-Status.JVM

TaskManagers-JVM-GC-Scavenge-Count

44640 44620 44600 44580 44560 44540 18:27 18:28 18:29 18:30 18:31

container_e12_1560936712321_14520_01_000003-Status.JVM

container_e12_1560936712321_14520_01_000002-Status.JVM

TaskManagers-JVM-GC-Scavenge-Time

6.83 min 6.75 min 6.67 min 6.58 min 18:27 18:28 18:29 18:30 18:31

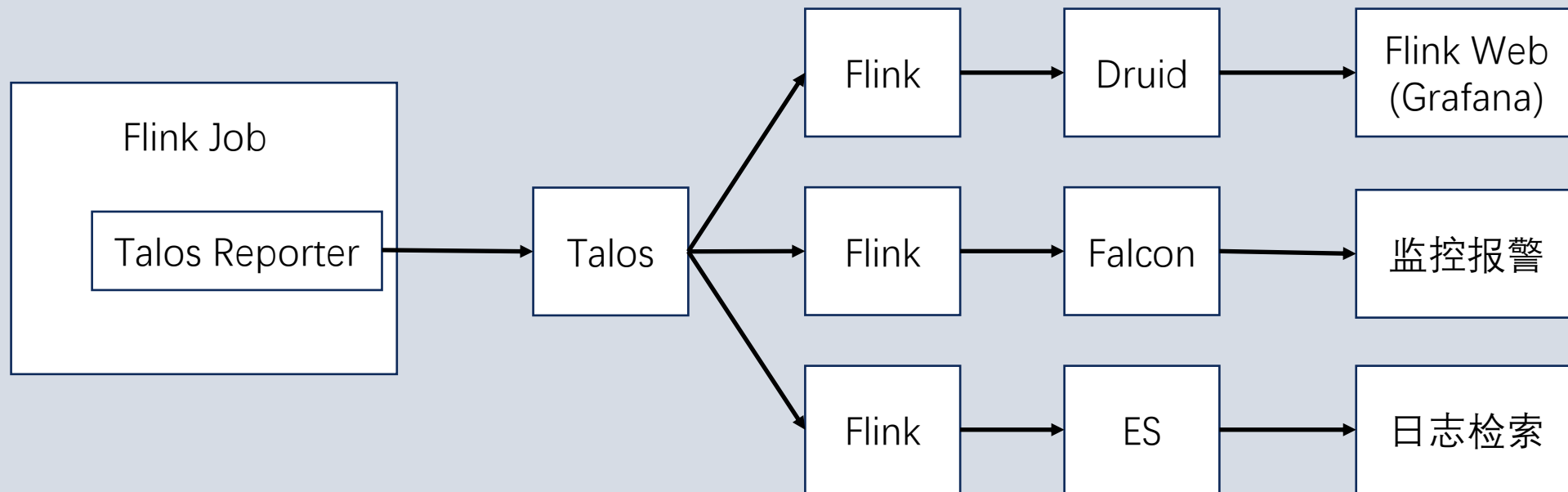
container_e12_1560936712321_14520_01_000003-Status.JVM

container_e12_1560936712321_14520_01_000002-Status.JVM

Flink数据体系（重构中）



Apache Flink



数据收集：所有数据通过Reporter收集到消息队列Talos

Metric展示：Talos + Flink + Druid + Grafana的方式实现Metric数据展示；

监控报警：Talos + Flink + Falcon的方式实现监控报警；

日志检索：Talos + Flink + ES的方式实现日志检索；

上述三种方案均会和Flink Web进行整合；

作业管理



Apache Flink

作业名	集群	创建时间	状态	操作
[REDACTED]	[REDACTED]	2019-09-06 15:59:00	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-07-29 11:30:25	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-07-29 10:35:05	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-04-23 17:55:25	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-09-19 11:04:30	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-09-19 15:32:22	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-08-01 10:38:00	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-07-29 10:35:15	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-09-19 19:09:36	● STARTED	停止 修改 删除 克隆
[REDACTED]	[REDACTED]	2019-07-31 11:13:14	● STARTED	停止 修改 删除 克隆

通过流式作业管理平台实现了Flink Streaming作业的提交/停止/删除/克隆等基本操作，同时支持作业运行历史，Flink Web连接等集成。



创建作业

Please select

▼

* 作业名:

1~80个字符，只能包含数字、字母、"_"和"-", 不能以"_"或"-开头

▼

* Kerberos账户:

* Yarn队列:

* Driver Memory:

示例: 512M,1G

* Driver Cores:

* Number Executors:

* Per Executor Memory:

示例: 512M,1G

* Per Executor Cores:

其他框架参数:

```
spark.yarn.driver.memoryOverhead=512
spark.yarn.executor.directMemoryOverhead=512
```


PART 03

小米典型应用



Hive数据转储

基于Message Event进行Hive partition选择，彻底解决数据延迟带来的Partition错乱问题

基于BucketingSink 重写Buckter实现

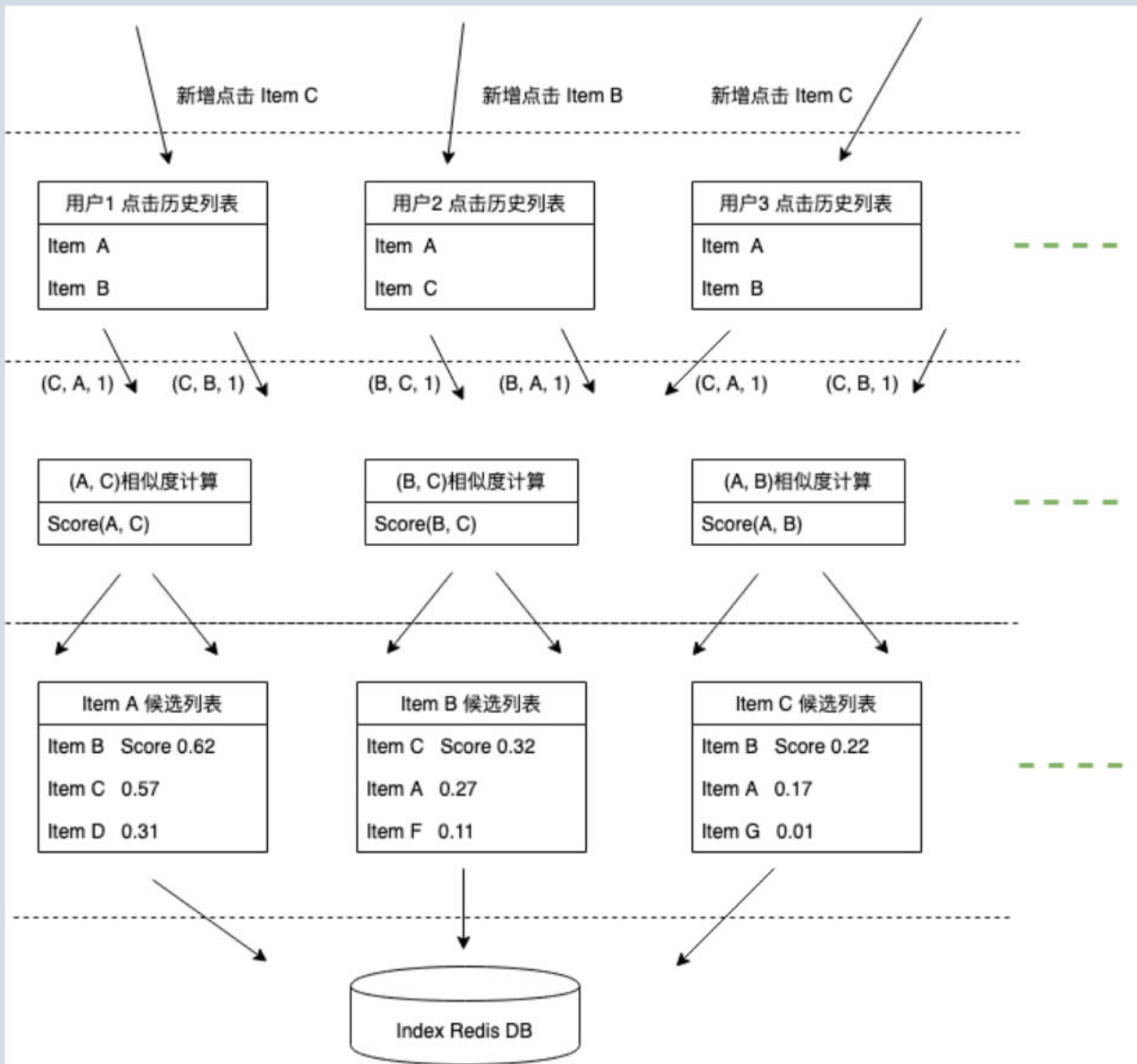
```
val fileSink = new BucketingSink[Tuple2[BytesWritable, BytesWritable]](defineOutputHDFSFile)
fileSink.setBucketer(new MiddleDataDateTimeBucketer[Tuple2[BytesWritable, BytesWritable]])
fileSink.setWriter(new SequenceFileWriter[BytesWritable, BytesWritable]())
fileSink.setBatchSize(1024 * 1024 * 400) // 每个part超过 400MB 就截断
fileSink.setBatchRolloverInterval(20 * 60 * 1000); // 超过 20min 也截断

outputStream.startNewChain().addSink(fileSink).name("rollingHdfsSink")
```

```
4.3 G
4.5 G
3.8 G
2.3 G
551.9 M
794.3 M
1.5 G
2.3 G
2.8 G
2.9 G
2.9 G
$ zjyhdfs -du -h
19/08/22 00:21:32 INFO security.UserGroupInformation: Can't login from keytab, try to login from ticket cache
165.1 M
400.0 M
360.9 M
233.4 M
166.6 M
400.0 M
360.4 M
232.9 M
date=20190821/hour=20
date=20190821/hour=21
date=20190821/hour=22
date=20190821/hour=23
date=20190821/hour=3
date=20190821/hour=4
date=20190821/hour=5
date=20190821/hour=6
date=20190821/hour=7
date=20190821/hour=8
date=20190821/hour=9
date=20190821/hour=23
date=20190821/hour=23/part-0-0
date=20190821/hour=23/part-0-1
date=20190821/hour=23/part-0-2
date=20190821/hour=23/part-0-3
date=20190821/hour=23/part-1-0
date=20190821/hour=23/part-1-1
date=20190821/hour=23/part-1-2
date=20190821/hour=23/part-1-3
```



实时 Item CF索引



更新频率：Spark Streaming 天级更新 + 小时级更新 => Flink 秒级更新

收益：图文浏览时长提升15s，实时度提升50%

峰值QPS：7w/s

State：27GB

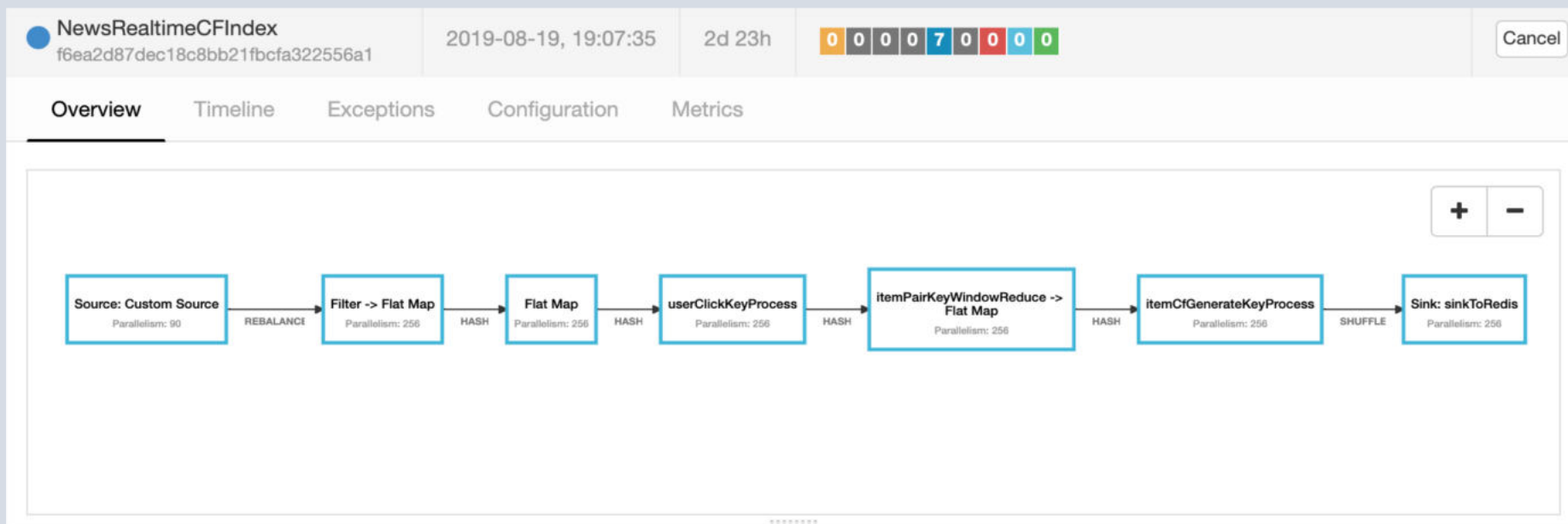
实时 Item CF索引



Apache Flink

```
val itemCfStream = datasource.filter(defineEntryProcessor.filterLogLine(_)).setParallelism(env.getParallelism)
    .flatMap(defineEntryProcessor.extractUserClickEntry(_))
    .keyBy(_._.itemId)
    .flatMap(new HbaseFlatMapFunction)
    .keyBy(_._.userId)
    .process(new UserRecentClickFunction()).name("userClickKeyProcess")
    .keyBy(defineEntryProcessor.generateSortedPairHashKey(_))
    .timeWindow(Time.seconds(5))
    .reduce((a, b) => defineEntryProcessor.reduceToSortedItemPairFunction(a, b)).uid("itemPairKeyWindowReduce")
    .flatMap(defineEntryProcessor().transformItemPairToItemSequence(_))
    .keyBy(_._.itemA)
    .process(new ItemCFGenerateFunction(defineEntryProcessor())).uid("itemCfGenerateKeyProcess")

//sink to index Redis
itemCfStream.shuffle.addSink(new IndexSinkDatabaseFunction(defineBusinessType())).name("sinkToRedis")
```



实时Impression拼接



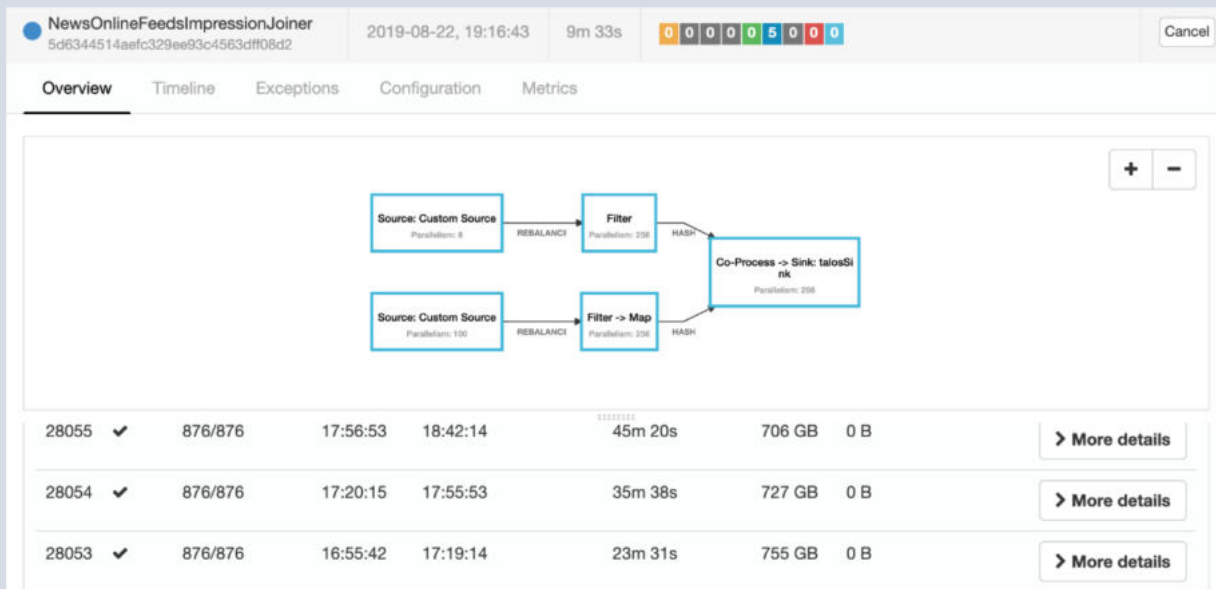
Apache Flink



State大小：800GB+

结果数据：17TB/天

基于connect，CoProcessFunction和Timer实现



PART 04

未来规划

未来规划



Apache Flink

- Flink 1.9 + Flink SQL
- 平台化建设
- 资源优化 + 动态调度
- 更多的社区参与



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

Apache Flink 社区微信公众号「Ververica」



Meetup动态 / Release 发布信息 / Flink 应用实践



THANKS

Apache Flink China Meetup

▪ BEIJING