Spark-streaming在京东的项目实践

刘永平 2014.09

JD.京东 COM

www.jd.com

JD.COM 京东

第一部分　项目背景

第二部分　项目实践

第三部分　项目中遇到的问题与解决

第一部分　项目背景

第二部分　项目实践

第三部分　项目中遇到的问题与解决

1、京东云海项目

开放京东商品、商家、客服绩效、品牌、行业五大主题数据,为ISV提供海量数据分布式存储计算平台，提供完整的云端数据仓库解决方案

1) T+1方案
2) Hive
3) 提高用户体验
4) 多、快、好、省

2、大数据处理的三个类型与相关技术

	使用场景	时间跨度	成熟开源软件	Spark
1	复杂的批量数据处理	小时级	MapReduce(Hive)	Spark
2	基于历史数据的交互式查询	分钟级	Impala	Spark sql(shark)
3	基于实时数据流的数据处理	秒级	Storm	Spark-streaming

但存在如下问题：

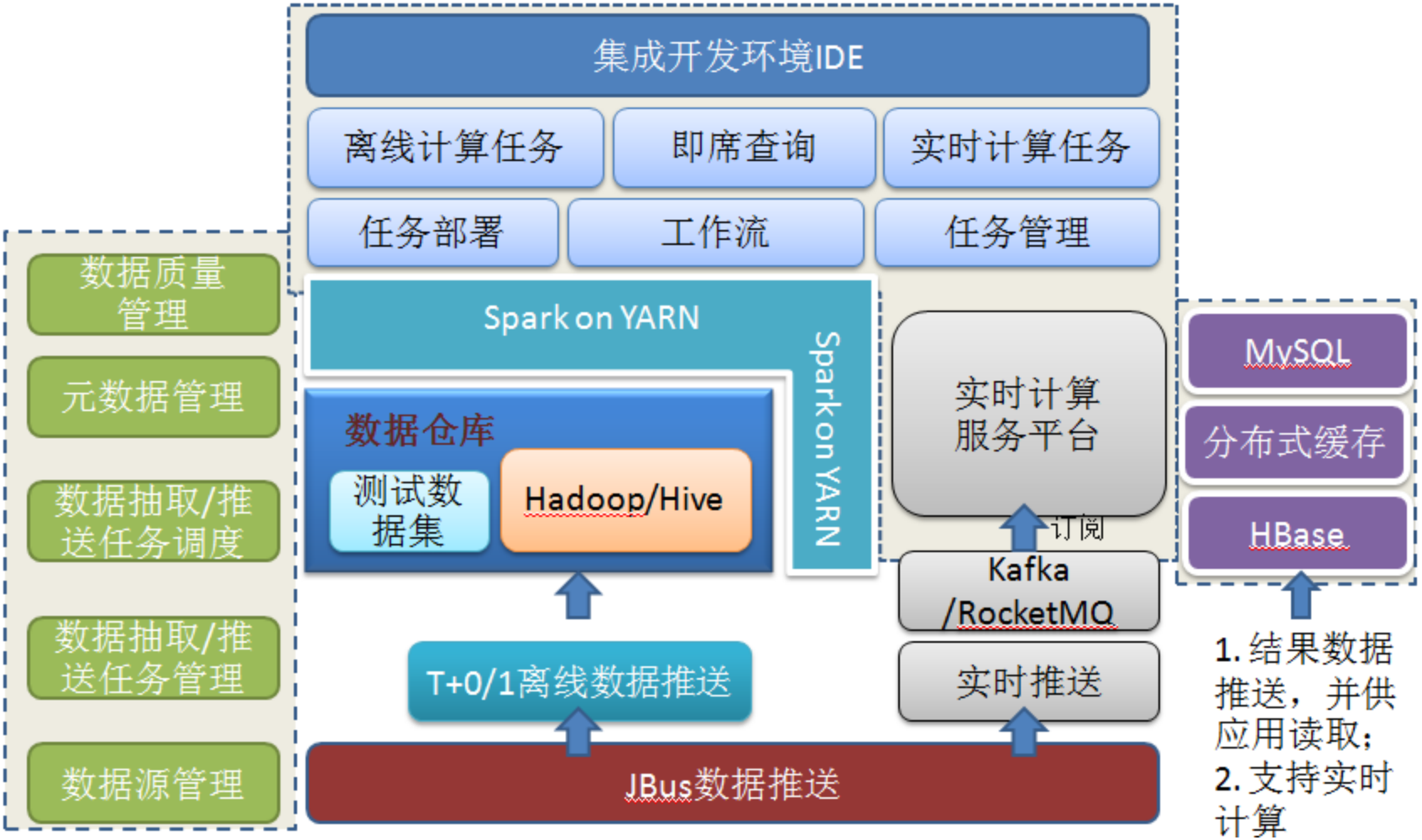
1) 三种情景的输入输出数据无法无缝共享，需要进行格式相互转换。
2) 每一个开源软件都需要一个开发和维护团队，提高了成本。
3) 在同一集群中对各个系统协调资源分配比较困难。

3、Spark 比较完美的解决了上述的问题

Spark-streaming有更丰富的数据转换处理的API，并且我们也尝试用StreamingSQL做大数据的实时增量计算。

Spark-streaming项目背景

JD.COM 京东



JD.COM 京东

客户期望(效果展示)

限时体验

实时概览（数据更新时间： 2014-09-04 15:51:23）

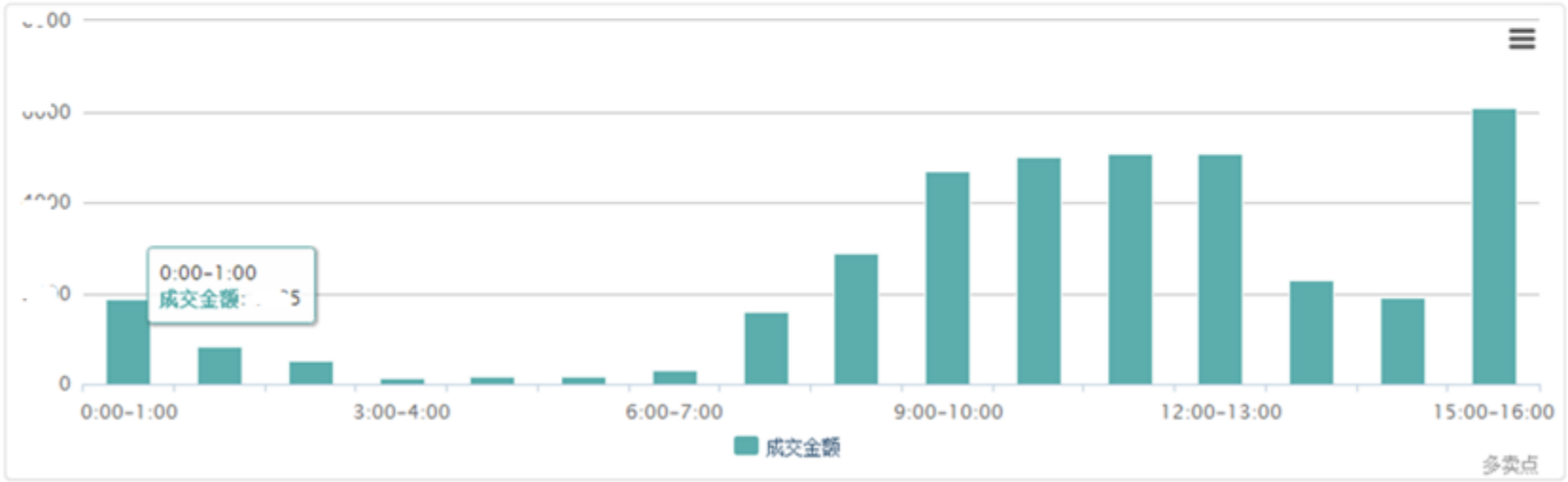
浏览量：7□□4　　下单人数：...　　下单单数：1□　　下单金额：□□□□

访客数：□□□　　成交人数：□□7　　成交单数：□□7　　成交金额：□□□□5

实时趋势

成交金额　成交人数　访客数　浏览量

各时段的数据



第一部分　项目背景

第二部分　项目实践

第三部分　项目中遇到的问题与解决

JD.COM 京东

一　概述
架构图等

二　运行与监控
提供图表进行直观分析

三　输入(流)
消息队列：mq与kafka　限流 位点存储

四　输出(结果)
hbase 预分区 多线程　计数器（Increment）

JD.COM 京东

一 概述
架构图等

二 运行与监控
提供图表进行直观分析

三 输入(流)
消息队列：mq与kafka 限流 位点存储

四 输出(结果)
hbase 预分区 多线程 计数器（Increment）

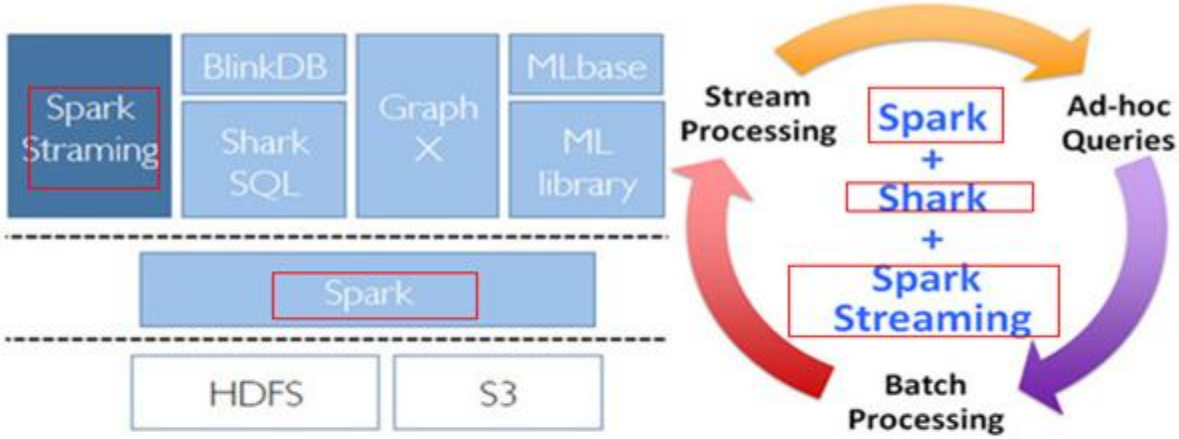


图1 BDAS软件栈

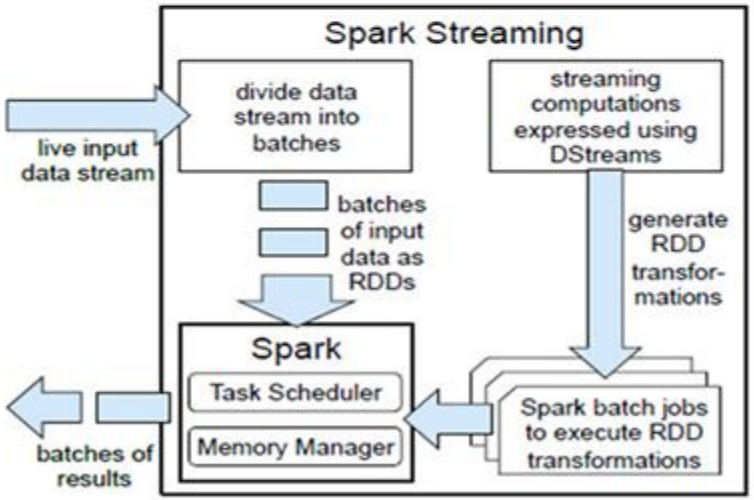
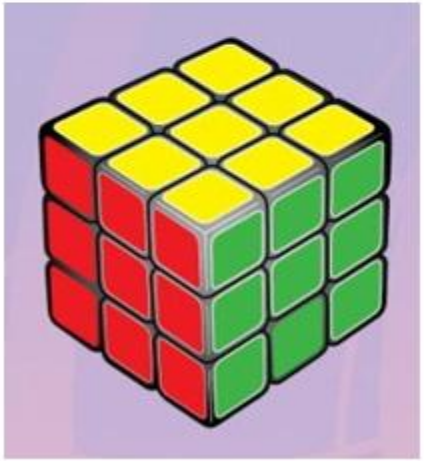
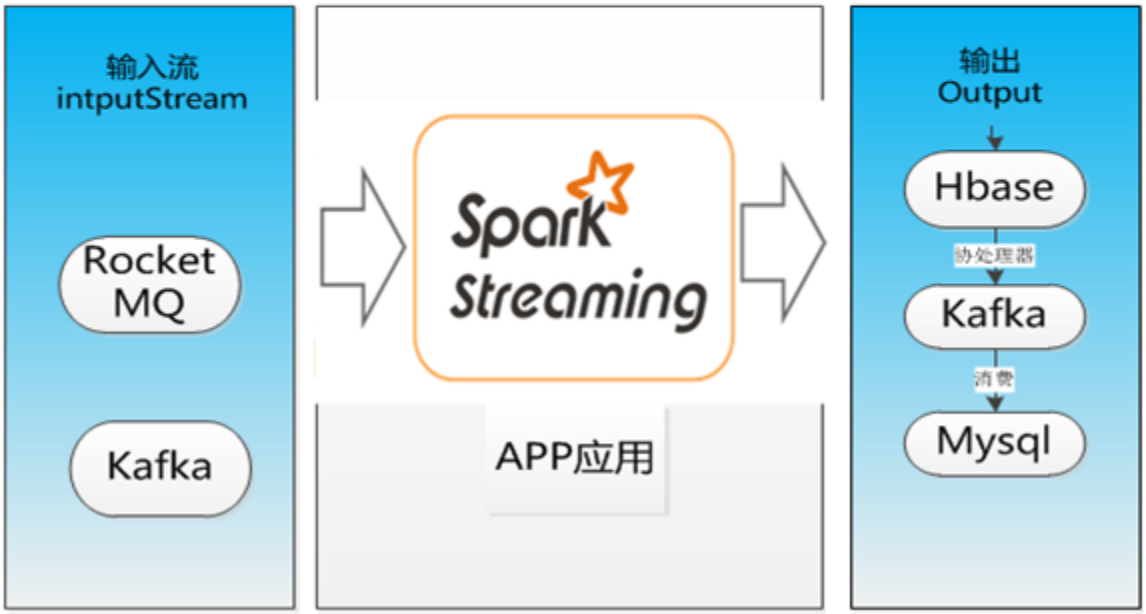


图2 Spark Streaming构架图



官网应用构架

项目应用构架



JD.COM 京东

一　概述
架构图等

二　运行与监控
提供图表进行直观分析

三　输入(流)
消息队列：mq与kafka　限流 位点存储

四　输出(结果)
hbase 预分区 多线程　计数器（Increment）

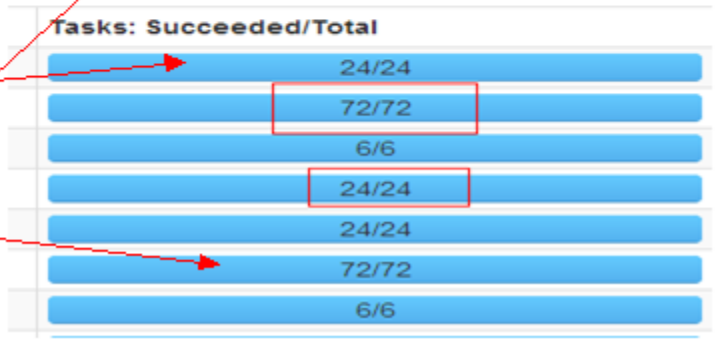
JD.COM 京东

1、启动命令（spark on yarn）

SPARK_JAR=./assembly/target/scala-2.10/spark-assembly-0.9.0-incubating-hadoop2.3.0.jar ./bin/spark-class org.apache.spark.deploy.yarn.Client **--jar /home/hadoop/running/kafkawithdiff.jar --class com.ode.realtime.app.AppKafkaTest --args 10010** --num-workers 6 --master-memory 3g --worker-memory 2g --worker-cores 4 **--name 10010_limitspeedworking**

2、报表级的参数配置

```
maps | parallelism | memfraction | currentjobs | ttl
-----+-------------+-------------+-------------+-----
  72 |          24 |      0.4000 |           3 | 3600
```



```
System.setProperty("spark.shuffle.consolidateFiles", "***");
System.setProperty("spark.streaming.blockInterval", "***");
System.setProperty("spark.serializer",
"org.apache.spark.serializer.KryoSerializer");
System.setProperty("spark.default.parallelism", "***");
System.setProperty("spark.storage.memoryFraction", "***");
System.setProperty("spark.streaming.concurrentJobs", "***");
System.setProperty("spark.cleaner.ttl",, "***");
System.setProperty("spark.shuffle.compress",, "***");
System.setProperty("spark.shuffle,spill.compress",, "***");
```

在创建JavaStreamingContext之前 进行属性配置

```
JavaStreamingContext ssc = new JavaStreamingContext(runmode,
appName,new Duration(3000), System.getenv("SPARK_HOME"),
JavaStreamingContext.jarOfClass(App.class));
```

Spark-streaming项目实践-运行与监控

JD.COM 京东

此图为剪裁的效果

- Cluster
  - About
  - Nodes
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Compl
66	0	16	50

User Metrics for dr.who

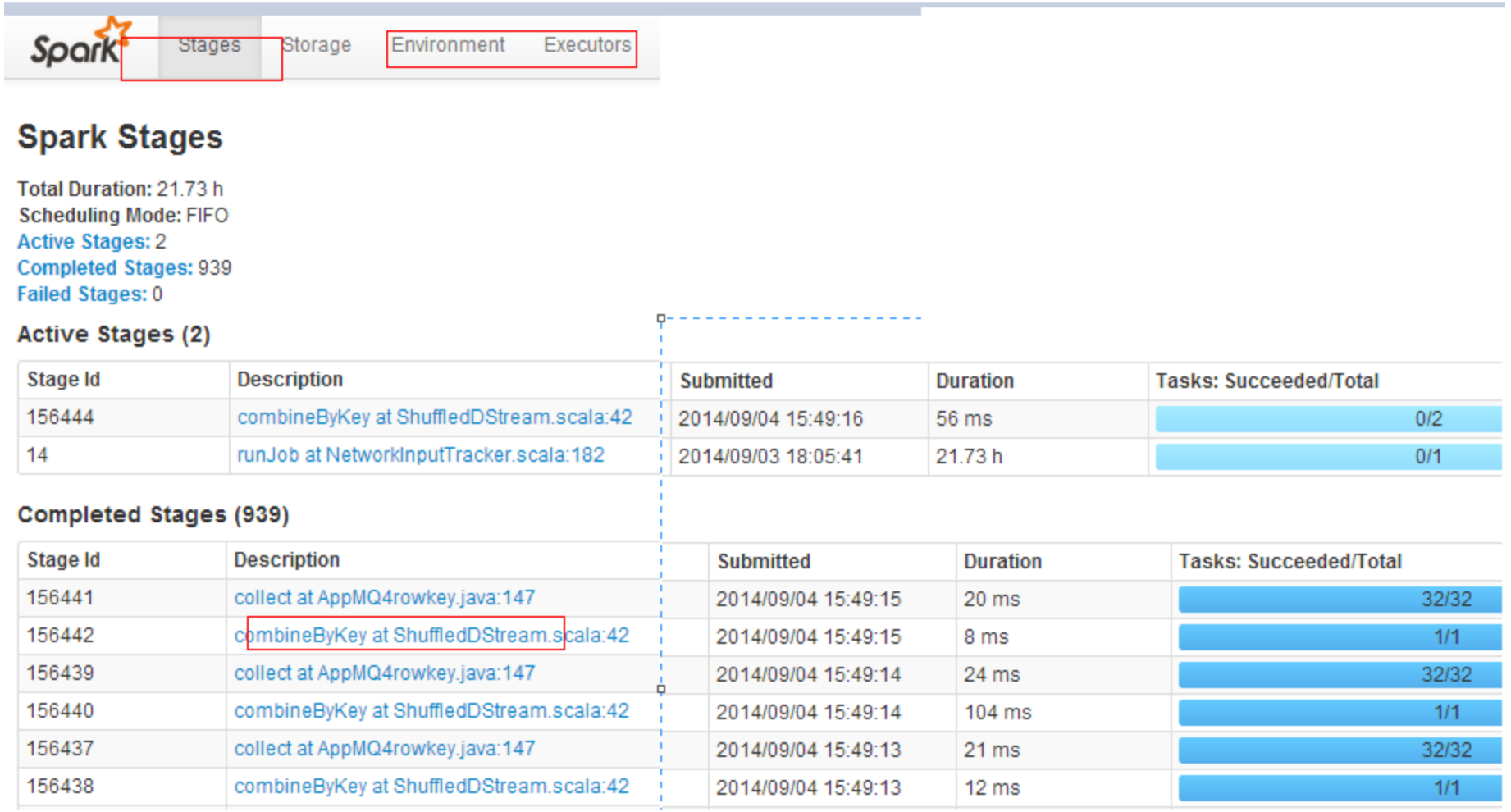
Apps Submitted	Apps Pending	Apps Running
0	0	16

Show 20 entries

ID	User	Name	Progress	Tracking UI
application 140[illegible] 574 0066	hadoop	com[illegible]ucc		ApplicationMaster
application 140[illegible] 74 0065	hadoop	com[illegible]otal		ApplicationMaster
application 140[illegible] 4 0064	hadoop	com[illegible]tal		ApplicationMaster
application 1409[illegible] 4 0063	hadoop	com[illegible]succ		ApplicationMaster
application 1409[illegible] 4 0062	hadoop	com[illegible]cc		ApplicationMaster
application 1409[illegible] 74 0061	hadoop	rowl[illegible]up		ApplicationMaster
application 1409[illegible] 74 0060	hadoop	2_[illegible]up		ApplicationMaster
application 1409[illegible] 74 0059	hadoop	com[illegible]al		ApplicationMaster
application 1409[illegible] 074 0057	hadoop	com[illegible]al		ApplicationMaster
application 1409[illegible] 3674 0053	hadoop	SharkServer_AL[illegible]		ApplicationMaster

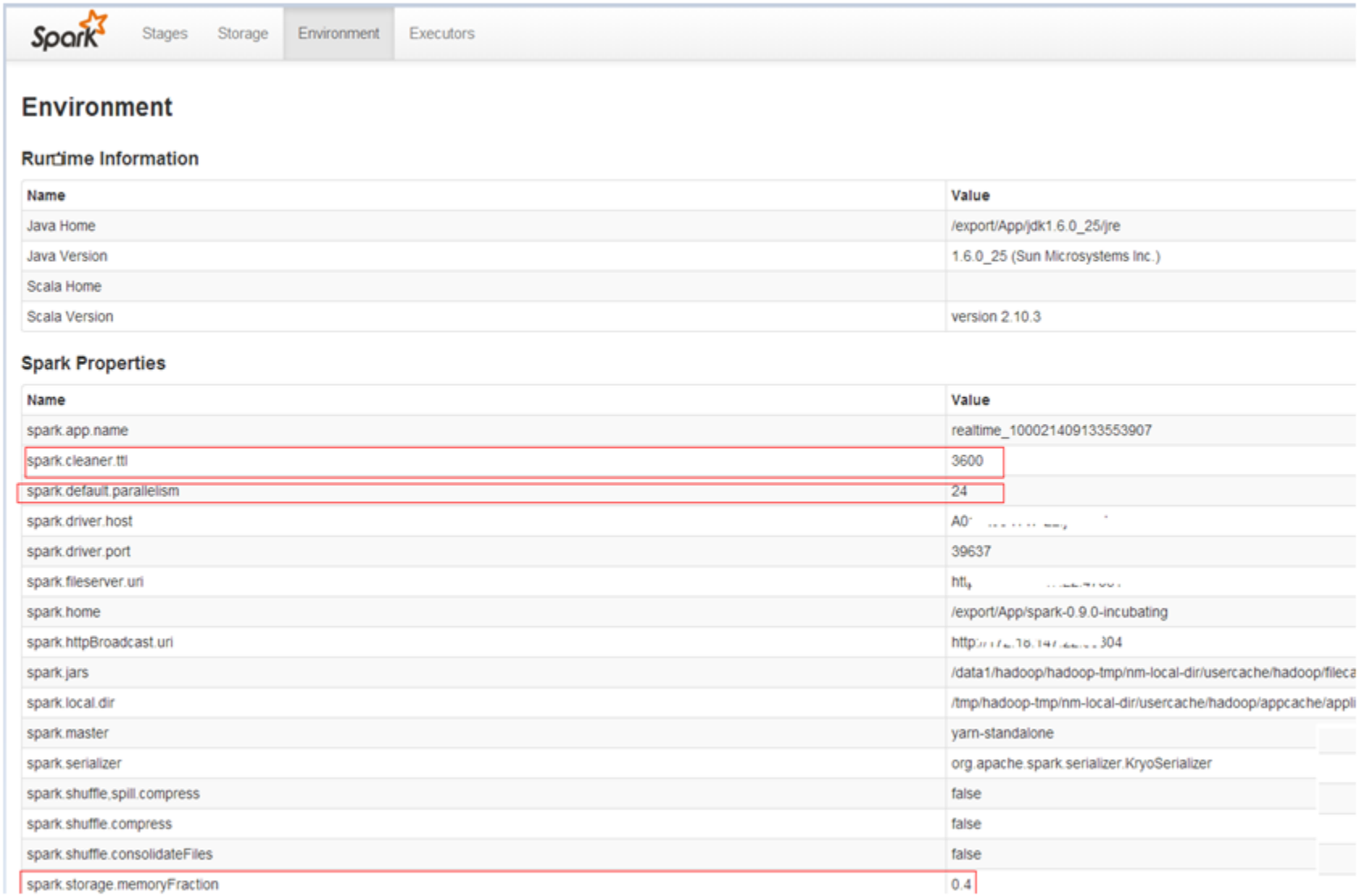
Spark-streaming项目实践-运行与监控

JD.COM 京东



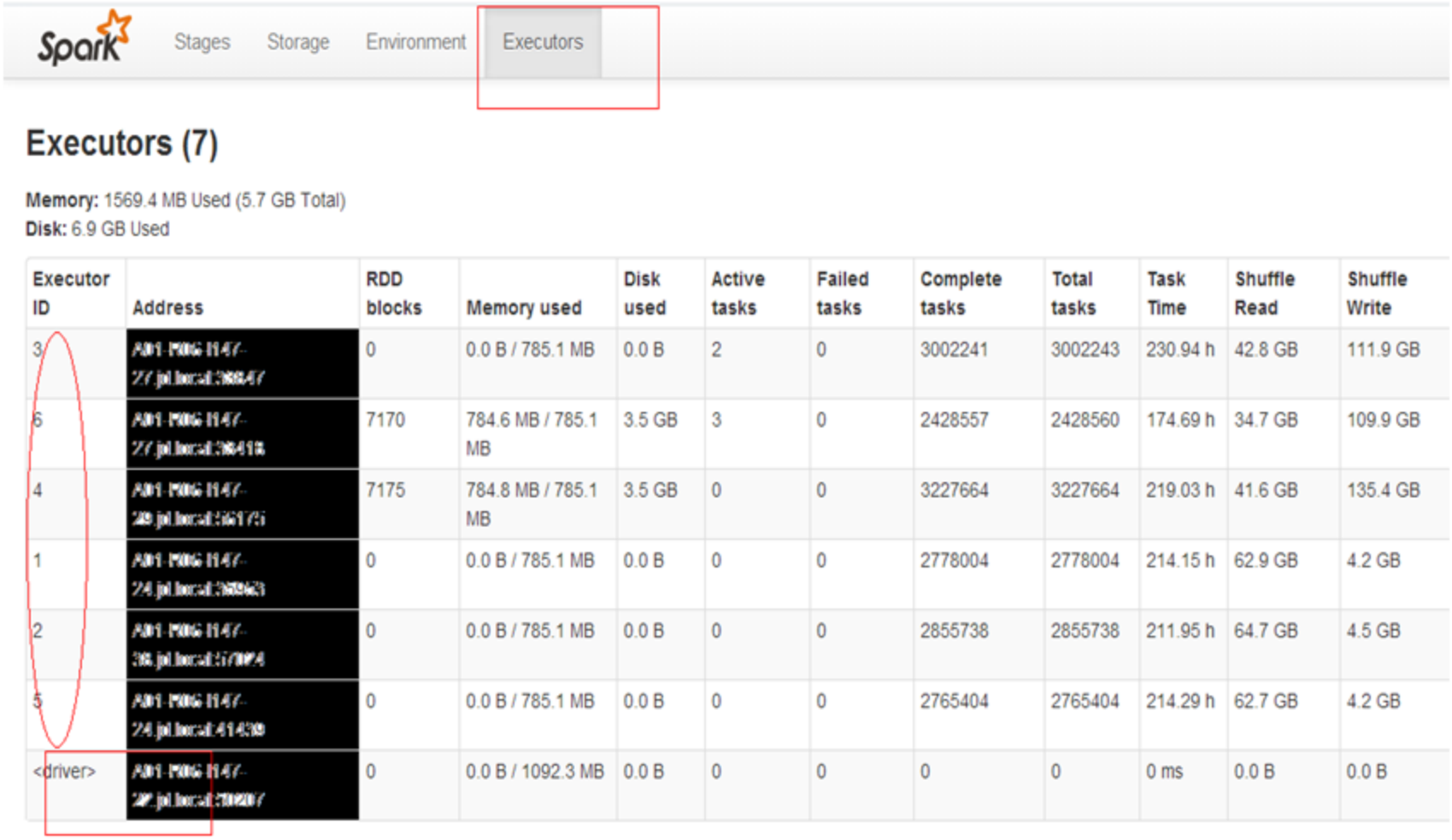
Spark-streaming项目实践-运行与监控

JD.COM 京东



Spark-streaming项目实践-运行与监控

JD.COM 京东



Executors (7)

Memory: 1569.4 MB Used (5.7 GB Total)
Disk: 6.9 GB Used

Executor ID	Address	RDD blocks	Memory used	Disk used	Active tasks	Failed tasks	Complete tasks	Total tasks	Task Time	Shuffle Read	Shuffle Write
3	[redacted]	0	0.0 B / 785.1 MB	0.0 B	2	0	3002241	3002243	230.94 h	42.8 GB	111.9 GB
6	[redacted]	7170	784.6 MB / 785.1 MB	3.5 GB	3	0	2428557	2428560	174.69 h	34.7 GB	109.9 GB
4	[redacted]	7175	784.8 MB / 785.1 MB	3.5 GB	0	0	3227664	3227664	219.03 h	41.6 GB	135.4 GB
1	[redacted]	0	0.0 B / 785.1 MB	0.0 B	0	0	2778004	2778004	214.15 h	62.9 GB	4.2 GB
2	[redacted]	0	0.0 B / 785.1 MB	0.0 B	0	0	2855738	2855738	211.95 h	64.7 GB	4.5 GB
5	[redacted]	0	0.0 B / 785.1 MB	0.0 B	0	0	2765404	2765404	214.29 h	62.7 GB	4.2 GB
<driver>	[redacted]	0	0.0 B / 1092.3 MB	0.0 B	0	0	0	0	0 ms	0.0 B	0.0 B

Spark-streaming项目实践-运行与监控

JD.COM 京东

1、大致说明：左上（延迟） 右上 （流入数） 左下 （写入速度） 右下（存储数）

2、延迟有两个高峰，分别有两个异常引起的，即突然大量流入数据与写入速度太慢。

JD.COM 京东

一　概述
架构图等

二　运行与监控
提供图表进行直观分析

三　输入(流)
消息队列：mq与kafka 限流 位点存储

四　输出(结果)
hbase 预分区 多线程 计数器（Increment） 10000

消息队列

1、RocketMQ

2、kafka

扩展：

1）限流

2）位点(消费点)服务端存储(MQ)

1、 MQ

```
JavaDStream<byte[]> messagesmq= RocketMQUtils.createStream(
        ssc
        , mqServer , groupName,consumerid , topic
        ,tag , startStr , stopStr
        , StorageLevel.MEMORY_AND_DISK_2()
        , new DelayAndLimitCallback(consumerDelay, rateLimit)
 );
```

2、KAFKA

```
JavaPairDStream<String, String> messages = KafkaUtils.createStream(ssc ,
topicMap , kafkaParams);
```

JD.COM 京东

一　概述
架构图等

二　运行与监控
提供图表进行直观分析

三　输入(流)
消息队列：mq与kafka　限流 位点存储

四　输出(结果)
hbase 预分区 多线程　计数器（Increment）10000

Hbase Kafka mysql

1、对写入的hbase表进行预分区，写入速度比未预分区提高速度达5-10倍

2、多线程写入

3、hbase 协处理器(触发器),发送至 kafka，最终被消费并存储在 mysql中

但实时计算过程中，有些场景则没有reduce阶段，**而只有map**，这时如果想**多线程或批量**操作，以提高性能，则使得Spark-streaming提供的API中的**mappartition**.

Spark-streaming项目实践-输出(结果)

JD.COM 京东

```
messages.mapPartitions(new PairFlatMapFunction<Iterator<byte[]>, String, Double>() {
        @Override
  public Iterable<Tuple2<String, Double>> call(Iterator<byte[]> iterator) throws Exception {
    List<Tuple2<String, Double>> list = new ArrayList<Tuple2<String, Double>>();
            todo();
    list.add(new Tuple2<String, Double>("received", Double.valueOf(data.size())));
    return list;
  } }).filter(new Function<Tuple2<String, Double>, Boolean>() {
      @Override
      public Boolean call(Tuple2<String, Double> stringDoubleTuple2) {
        if (stringDoubleTuple2 == null) {
          return false;
         }
        return true;
      }
    }).reduceByKey(new Function2<Double, Double, Double>() {  @Override
      public Double call(Double i1, Double i2) {
        return i1 + i2;
      }
    });
```

```
javaPairDStream.foreachRDD(new Function2<JavaPairRDD<String,
    List<Double>>, Time, Void>() {
        @Override
        public Void call(JavaPairRDD<String, List<Double>>
    stringIntegerJavaPairRDD, Time time) {
List<Tuple2<String, List<Double>>> list = stringIntegerJavaPairRDD.collect();
for (Tuple2 t : list) {
    todo();
 }
    return null;
        }
    });
```

此处为print方法

JD.COM 京东

第一部分　项目背景

第二部分　项目实践

第三部分　项目中遇到的问题与解决

JD.COM 京东

1、block 还未进行计算就被ttl的设置而删除,导致not found错误

System.setProperty(“spark.cleaner.ttl”， “3600”); 单位:秒

以上指自动清理一小时以前的RDD，如果你的计算延迟了一小时，就会报以下的错，并导致应用失败

解决办法：1) 提高性能，以减少延迟的时间，使延迟一直在安全范围内。

2) 对输入流进行**限速，如每秒只允许多少条**，这样，在生产者进行大量数据发送时，延迟还在可控范围之内

```
14/09/01 09:15:38 WARN TaskSetManager: Lost TID 15131 (task 452.0:0)
14/09/01 09:15:38 WARN TaskSetManager: Loss was due to java.lang.Exception
java.lang.Exception: Could not compute split, block input-0-1409534137000 not found
        at org.apache.spark.rdd.BlockRDD.compute(BlockRDD.scala:45)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:232)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:34)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:232)
        at org.apache.spark.rdd.FilteredRDD.compute(FilteredRDD.scala:33)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:232)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:34)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:232)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:161)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:102)
        at org.apache.spark.scheduler.Task.run(Task.scala:53)
        at org.apache.spark.executor.Executor$TaskRunner$$anonfun$run$1.apply$mcV$sp(Executor.scala:213)
        at org.apache.spark.deploy.SparkHadoopUtil.runAsUser(SparkHadoopUtil.scala:49)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:178)
        at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
        at java.lang.Thread.run(Thread.java:662)
```

2、hbase写入速度过慢

此原因一般是hbase进行了相应的后台操作，最多是**Minor compact**

解决办法：1） 预分区表

2）多线程处理（使用队列或mappartition)

目前，我们每条数据写入速度，平均是0.2ms/条。

附：在我们的应用中，除了使用Spark-streaming的.reduceByKey的机制，还使用了*hbase的计数器*（Increment）。

1、**同一批（Duration)数据**的汇总用spark-streaming的reduceByKey。

2、**不同批次**的数据，则使用hbase的计数器进行汇总。

JD.COM 京东

恶性循环

流入量增大 → 流出量增大 → 小压缩 → 写入变慢 → 延迟增加

3、INFO AMRMClientImpl: Waiting for application to be successfully unregistered.

一般是由于配置文件不正确引起的，比如线上环境与测试环境打包时混淆了。

JD.COM 京东

APP运行日志

1) 运行结束后可导出日志，但运行中的不能导出

yarn logs -applicationId application_14##########_0### > log###

2) 运行时日志查看

1 界面 不能看到worker的输出

ID	Logs
application_1407837983011_0287	logs

stderr : Total file length is 964059366 bytes.
stdout : Total file length is 53 bytes.

2 后台 可看到所有的输出，日志所在路径见hadoop的配置中的log4j

Executor ID	Address	RDD blocks
3	[redacted]	0

```
[root@A... container_1407837983011_0287_01_000002]# ll
total 1868688
-rw-rw-r-- 1 hadoop hadoop 1913529019 Sep  3 09:20 stderr
-rw-rw-r-- 1 hadoop hadoop          0 Sep  2 19:20 stdout
[root@A01 ... container_1407837983011_0287_01_000002]# pwd
/export/Logs/hadoop/yarn/application_1407837983011_0287/container_1407837983011_0287_01_000002
[root@A... container_1407837983011_0287_01_000002]#
```

谢谢

We are hiring!

ode@jd.com

JD.COM 京东