

中国移动运营分析实时监控平台

1. 项目背景

中国移动公司旗下拥有很多的子机构,基本可以按照省份划分. 而各省份旗下的充值机构也非常的多.

目前要想获取整个平台的充值情况,需要先以省为单元,进行省份旗下的机构统计,然后由下往上一层一层的统计汇总,过程太过繁琐,且统计周期太长. 且充值过程中会涉及到中国移动信息系统内部各个子系统之间的接口调用, 接口故障监控也成为了重点监控的内容之一.

为此建设一个能够实时监控全国的充值情况的平台, 掌控全网的实时充值, 各接口调用情况意义重大.

2. 技术选型

2.1. 难点分析

- 移动公司旗下子充值机构众多, 充值数据量大.
- 数据实时性要求高

2.2. 可用技术选型

- 实时流式计算框架 Storm
- 实时流式计算框架 Spark Streaming
- 实时流式计算框架 Flink

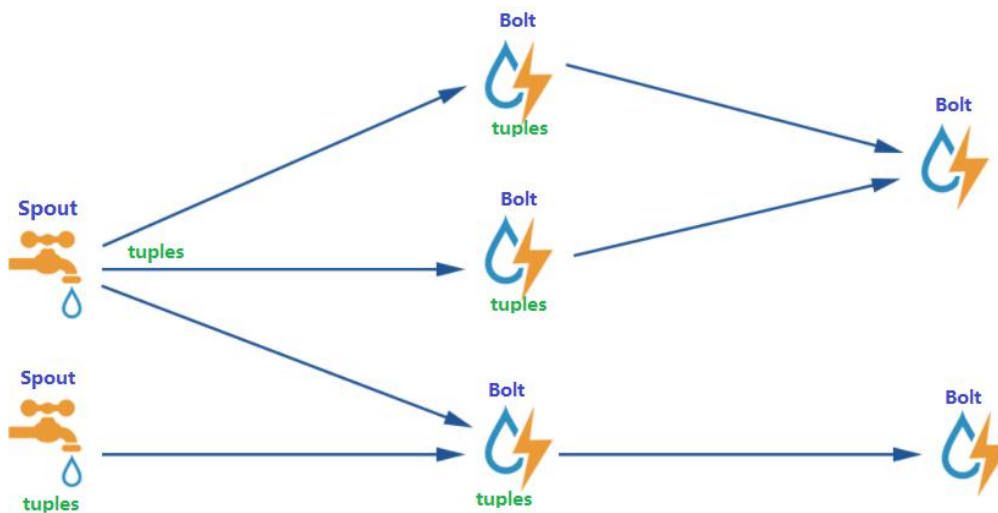
2.3. 对比分析

Storm、Spark streaming、Flink 都是开源的分布式系统, 具有低延迟、可扩展和容错性诸多优点, 允许你在运行数据流代码时, 将任务分配到一系列具有容错能力的计算机上并行运行, 都提供了简单的 API 来简化底层实现的复杂程度。

Apache Storm

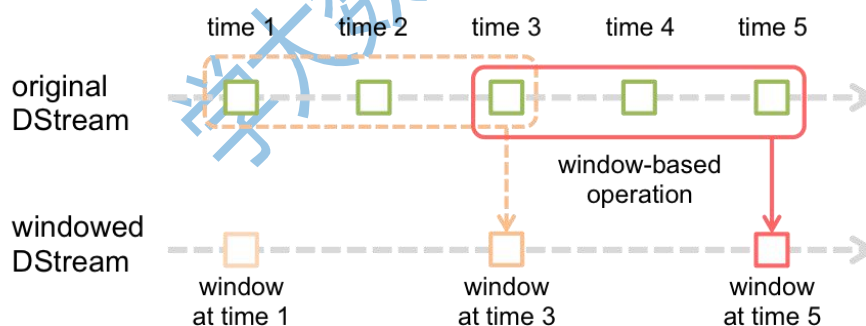
在 Storm 中, 先要设计一个用于实时计算的图状结构, 我们称之为拓扑 (topology)。这个拓扑将会被提交给集群, 由集群中的主控节点 (master node) 分发代码, 将任务分配给工作节点 (worker node) 执行。一个拓扑中包括 spout 和 bolt 两种角色, 其中 spout 发送消息,

负责将数据流以 **tuple** 元组的形式发送出去；而 **bolt** 则负责转换这些数据流，在 **bolt** 中可以完成计算、过滤等操作，**bolt** 自身也可以随机将数据发送给其他 **bolt**。由 **spout** 发射出的 **tuple** 是不可变数组，对应着固定的键值对。



Apache Spark

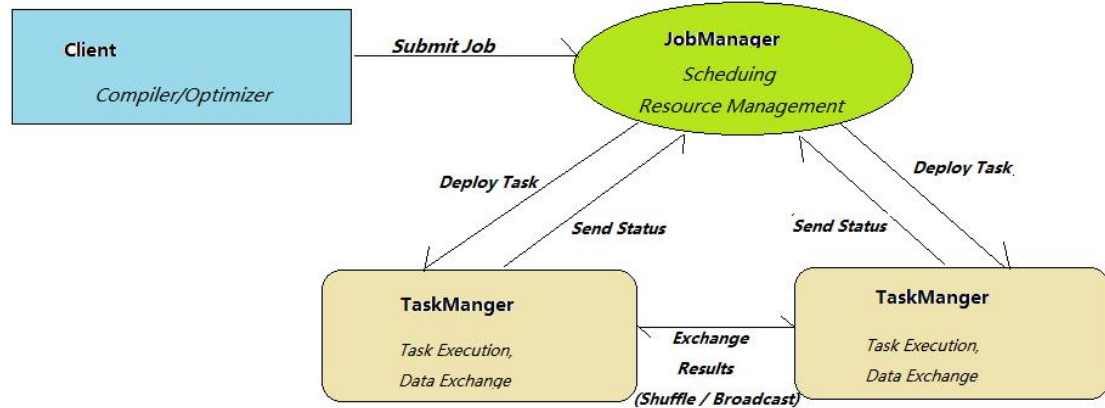
Spark Streaming 是核心 Spark API 的一个扩展，它并不会像 Storm 那样一次一个地处理数据流，而是在处理前按时间间隔预先将其切分为一段一段的批处理作业。Spark 针对持续性数据流的抽象称为 DStream (DiscretizedStream)，一个 DStream 是一个微批处理 (micro-batching) 的 RDD (弹性分布式数据集)；而 RDD 则是一种分布式数据集，能够以两种方式并行运作，分别是任意函数和滑动窗口数据的转换。



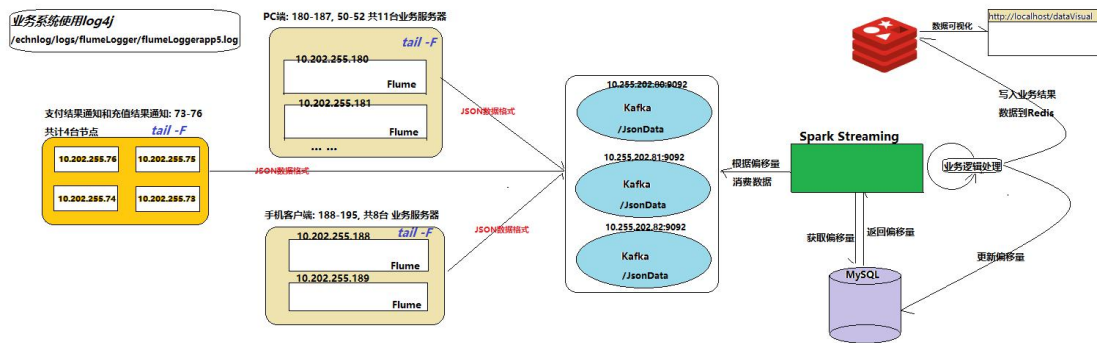
Apache Flink

Flink 是一个针对流数据和批数据的分布式处理引擎。它主要是由 Java 代码实现。对 Flink 而言，其所要处理的主要场景就是流数据，批数据只是流数据的一个极限特例而已。再换句话说，Flink 会把所有任务当成流来处理，这也是其最大的特点。Flink 可以支持本地的快速迭代，以及一些环形的迭代任务。并且 Flink 可以定制化内存管理。在这点上，如果要对比 Flink 和 Spark 的话，Flink 并没有将内存完全交给应用层。这也是为什么 Spark 相对于 Flink，

更容易出现 OOM 的原因（out of memory）。就框架本身与应用场景来说，Flink 更相似与 Storm。



3. 项目架构



73 到 76 主机，每台主机上面部署了 6 个 server, 2 个负责支付结果通知，2 个负责发起充值请求，2 个负责充值结果通知.目前退费没有监控.

4. 集群规模--Flume

- 业务节点分为 PC 端和手机端:
PC 端一共有 11 台, flume 会监控每台节点生成的日志文件, 一共有 11 个 Flume 节点
手机端一共有 8 台, flume 会监控每台节点生成的日志文件, 一共有 8 个 Flume 节点.
 - 支付结果通知和充值结果通知:
该业务有 4 台节点, flume 会监控每台节点生成的日志文件, 一共有 4 个 Flume 节点.
- 机器配置?

5. 项目数据量

- 数据量每天大概 2000 到 3000 万笔的下单量, 每条数据大概在 0.5KB 左右, 下单量数据大概在 15GB 左右.

- 最后充值成功的大概 500 到 1000 万,平时充值成功的大概五六百万笔.
- 月初和月末量比较大

重点:

扫码留下手机号可获取海量大数据实战相关资料!

