# 课程介绍

课程名称：Storm是什么

课程目标：

通过该课程的学习能够了解离线计算与流式计算的区别、掌握Storm框架的基础知识、了解流式计算的一般架构图。

课程大纲：

1. 离线计算是什么？
2. 流式计算是什么？
3. 流式计算与离线计算的区别？
4. Storm是什么？
5. Storm与Hadoop的区别？
6. Storm的应用场景及行业案例
7. Storm的核心组件（重点掌握）
8. Storm的编程模型（重点掌握）
9. 流式计算的一般架构图(重点掌握)

# 背景介绍

Storm背景介绍

# 课程内容

## 1、离线计算是什么？

离线计算：批量获取数据、批量传输数据、**周期性**批量计算数据、数据展示

代表技术：Sqoop批量导入数据、HDFS批量存储数据、MapReduce批量计算数据、Hive批量计算数据、\*\*\*任务调度

公司数据部门中的人员分工:

1，hivesql

2、调度平台

3、Hadoop集群运维

4、数据清洗（脚本语言）

5、元数据管理

6、数据稽查

7、数据仓库模型架构

## 2、流式计算是什么

流式计算：数据实时产生、数据实时传输、数据实时计算、实时展示

代表技术：Flume实时获取数据、Kafka/**metaq**实时数据存储、**Storm/JStorm**实时数据计算、Redis实时**结果**缓存、持久化存储(mysql)。

一句话总结：将源源不断产生的数据实时收集并实时计算，尽可能快的得到计算结果

## 3、离线计算与实时计算的区别

最大的区别：实时收集、实时计算、实时展示

## 4、Storm是什么？

Flume实时采集，低延迟

Kafka消息队列，低延迟

Storm实时计算，低延迟

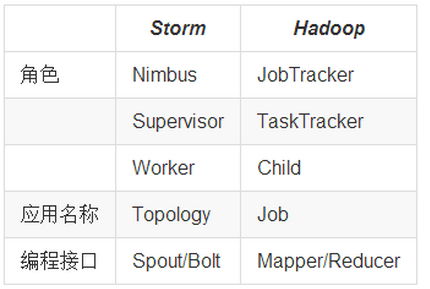
Redis实时存储，低延迟

Storm用来**实时处理数据**，特点：低延迟、高可用、分布式、可扩展、**数据不丢失**。提供简单容易理解的接口，便于开发。

海量数据？数据类型很多，产生数据的终端很多，处理数据能力增强

## 5、Storm与Hadoop的区别

* Storm用于实时计算，Hadoop用于离线计算。
* Storm处理的数据保存在内存中，源源不断；Hadoop处理的数据保存在文件系统中，一批一批。
* Storm的数据通过网络传输进来；Hadoop的数据保存在磁盘中。
* Storm与Hadoop的编程模型相似



Job：任务名称

JobTracker：项目经理（现在是提交到yarn resourcemanager）

TaskTracker：开发组长、产品经理（nodemanager） **yarn可以调度spark，mr等程序，通过appmaster**

Child:负责开发的人员

Mapper/Reduce:开发人员中的两种角色，一种是服务器开发、一种是客户端开发

Topology:任务名称

Nimbus:项目经理

Supervisor:开组长、产品经理

Worker:开人员

Spout/Bolt：开人员中的两种角色，一种是服务器开发、一种是客户端开发

## 6、Storm应用场景及行业案例

Storm用来**实时计算**源源不断产生的数据，如同流水线生产。

### 6.1、运用场景

* **日志分析**

从**海量**日志中分析出特定的**数据**，并将**分析**的结果**存入外部存储器**用来辅佐决策。

* **管道系统**

将一个数据从一个系统传输到另外一个系统，比如将数据库同步到Hadoop

* **消息转化器**

将接受到的消息按照某种格式进行转化，存储到另外一个系统如消息中间件

### 6.2、典型案列

* **一淘-实时分析系统：实时分析用户的属性，并反馈给搜索引擎**

最初，用户属性分析是通过每天在云梯上定时运行的MR job来完成的。为了满足实时性的要求，**希望能够实时分析用户的行为日志**，将最新的用户属性反馈给搜索引擎，能够为用户展现最贴近其当前需求的结果。

**商家还实时监测用户流量（用户浏览行为），来做业务调整，如赠送优惠券，降价等。**

* **携程-网站性能监控：实时分析系统监控携程网的网站性能**

利用HTML5提供的performance标准获得可用的指标，并记录日志。Storm集群实时分析日志和入库。使用DRPC聚合成报表，通过历史数据对比等判断规则，触发预警事件。

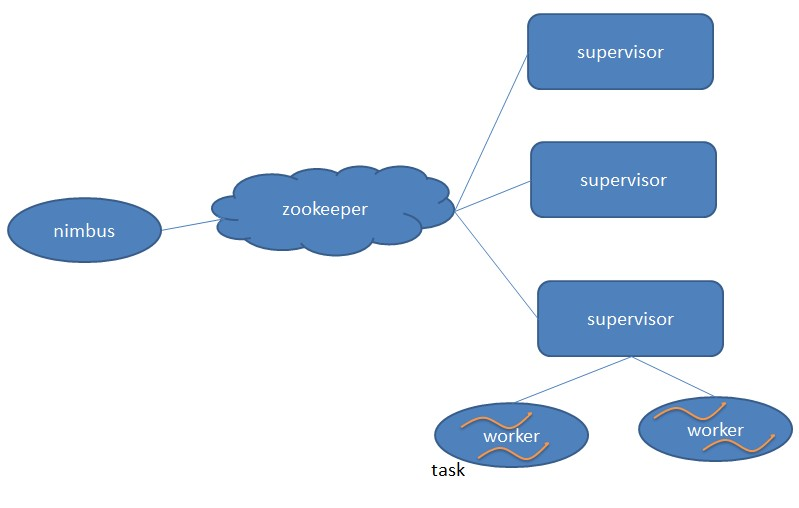
**如实时监控某台服务器cpu要爆满了，赶紧通知去换电脑。**

* **阿里妈妈-用户画像：实时计算用户的兴趣数据**

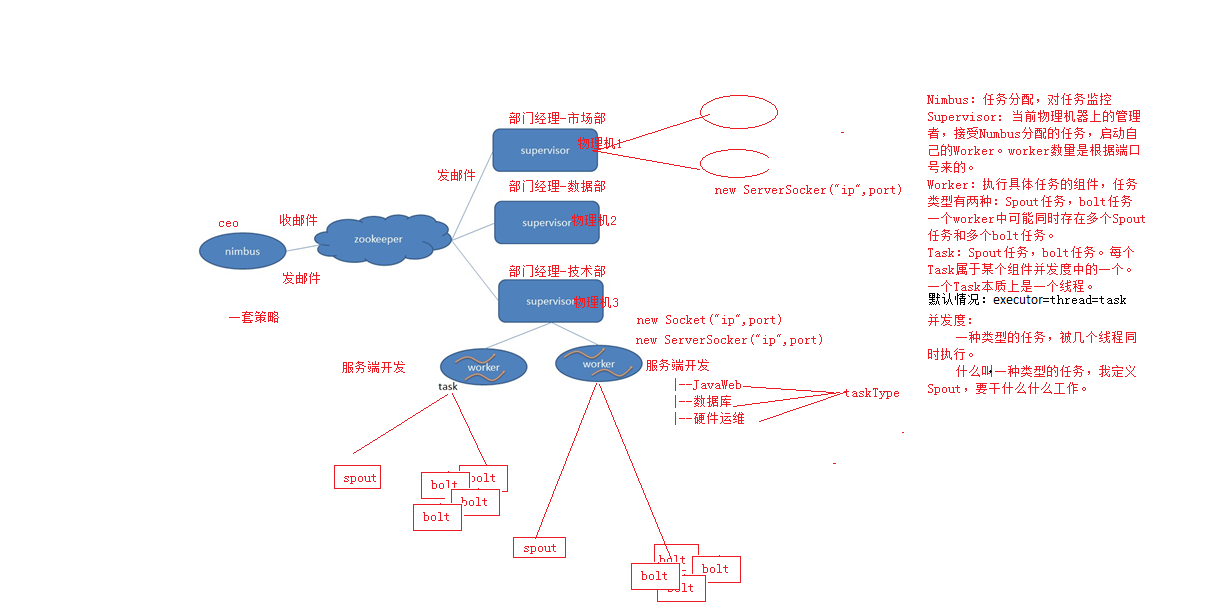
为了更加精准投放广告，阿里妈妈后台计算引擎需要维护每个用户的兴趣点（理想状态是，你对什么感兴趣，就向你投放哪类广告）。用户兴趣主要基于用户的历史行为、用户的实时查询、用户的实时点击、用户的地理信息而得，其中实时查询、实时点击等用户行为都是实时数据。**考虑到系统的实时性**，阿里妈妈使用Storm维护用户兴趣数据，并在此基础上进行受众定向的**广告投放。**

**如你刚到某个地方，实时监测你的位置，来给你推荐周围的商圈。**

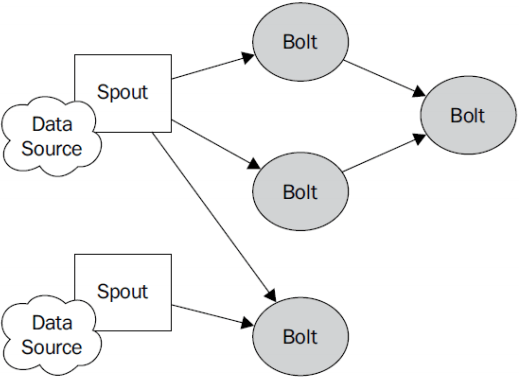
## 7、Storm核心组件（重要）

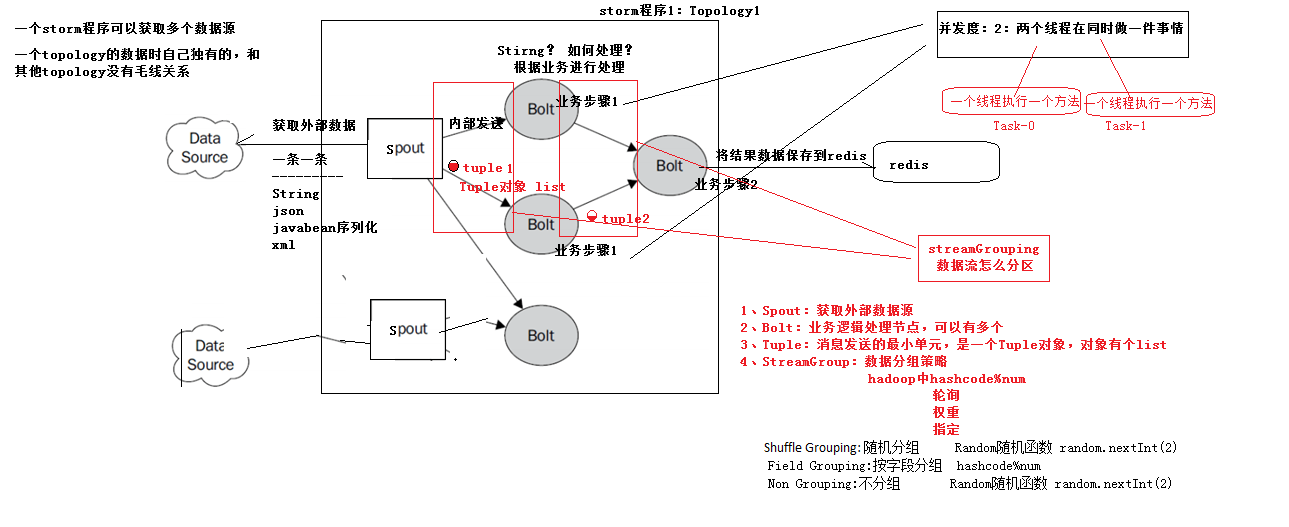


* Nimbus：负责资源分配和任务调度。
* Supervisor：负责接受nimbus分配的任务，启动和停止属于自己管理的worker进程。**---通过配置文件设置当前supervisor上启动多少个worker。**
* Worker：运行具体处理组件逻辑的**进程，就是一个jvm**。Worker运行的任务类型只有两种，**一种是Spout任务，一种是Bolt任务**。
* Task：**一个spout和bolt中可能有多个线程，一个task就是其中的一个线程，r（executor是执行器，用于执行Task）。**worker中每一个spout/bolt的线程称为一个task. **在storm0.8之后**，task不再与物理线程对应，不同spout/bolt的task可能会共享一个物理线程，该线程称为executor，一个 executor 是由 worker 进程生成的一个线程。**在 一个executor 中可能会有一个或者多个 task**，这些 task 都是为同一个组件（spout 或者 bolt）服务的。在拓扑的整个生命周期中每个组件的 task 数量都是保持不变的，不过每个组件的 executor 数量却是有可能会随着时间变化。**在默认情况下 task 的数量是和 executor 的数量一样的，也就是说，默认情况下 Storm 会在每个线程上运行一个 task。**



## 8、Storm编程模型（重要）





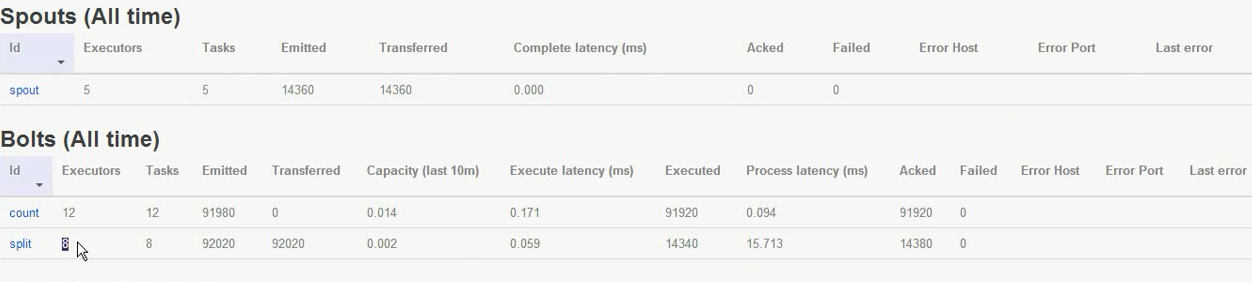
* Topology：Storm中运行的一个实时应用程序的名称。（拓扑）
* Spout：在一个topology中获取**源数据流**的组件。通常情况下spout会从外部数据源中读取数据，然后转换为topology内部的源数据。

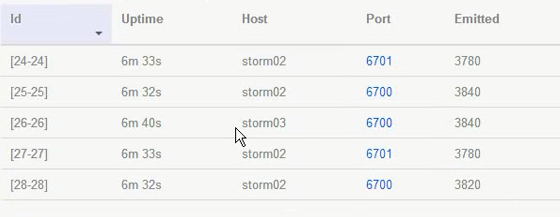
Bolt：接受数据然后执行处理的组件,用户可以在其中执行自己想要的操作。

* Tuple：一次消息传递的基本单元，理解为一组消息就是一个Tuple。
* Stream：表示数据的流向。

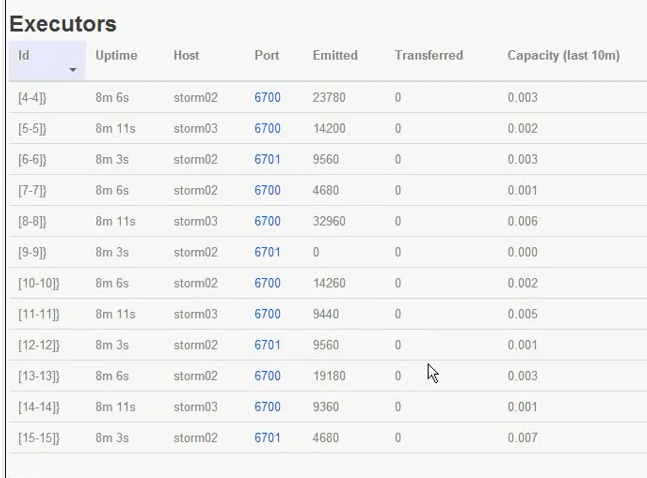
**注：一个spout或blot可以分布运行在多个Supervisor的worker上：**

**如一个spout有5个task要执行，集群会根据负载均衡，在storm02上启动两个worker，每个worker运行两个task，storm03启动一个worker，启动一个exector去执行一个task，这样spout的五个task就分布运行好了。**





**，同样blot也是如此：**



**注：Worker与topology**

**一个worker只属于一个topology,每个worker中运行的task只能属于这个topology，topology被kill掉，worker也随之杀掉。** 反之，一个topology包含多个worker，其实就是这个topology运行在多个worker上。

**一个topology要求的worker数量如果不被满足，集群在任务分配时，根据现有的worker先运行topology。**

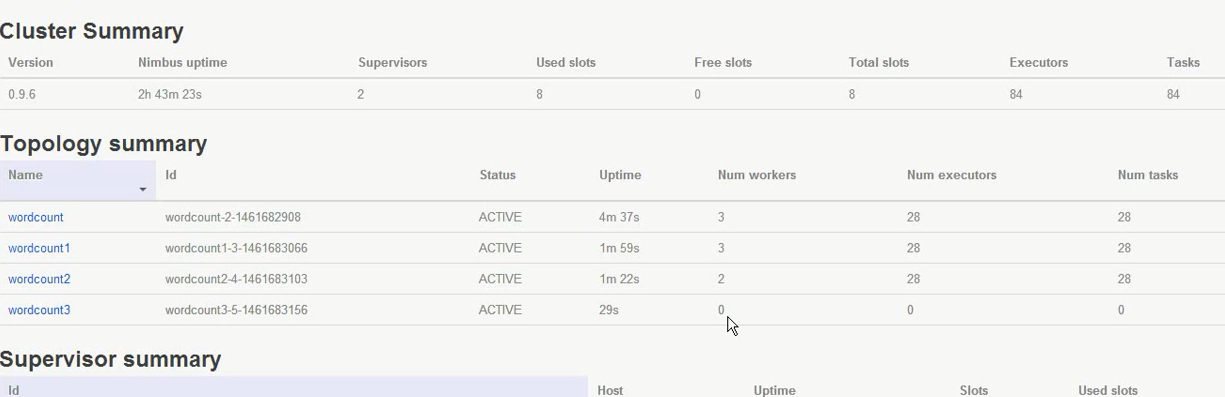
**如果当前集群中worker数量为0，那么最新提交的topology将只会被标识active，不会运行，只有当集群有了空闲资源之后，才会被运行。**

**集群中有两台Supervisor，每台只配置四个端口号，所以做多启动8个worker，**

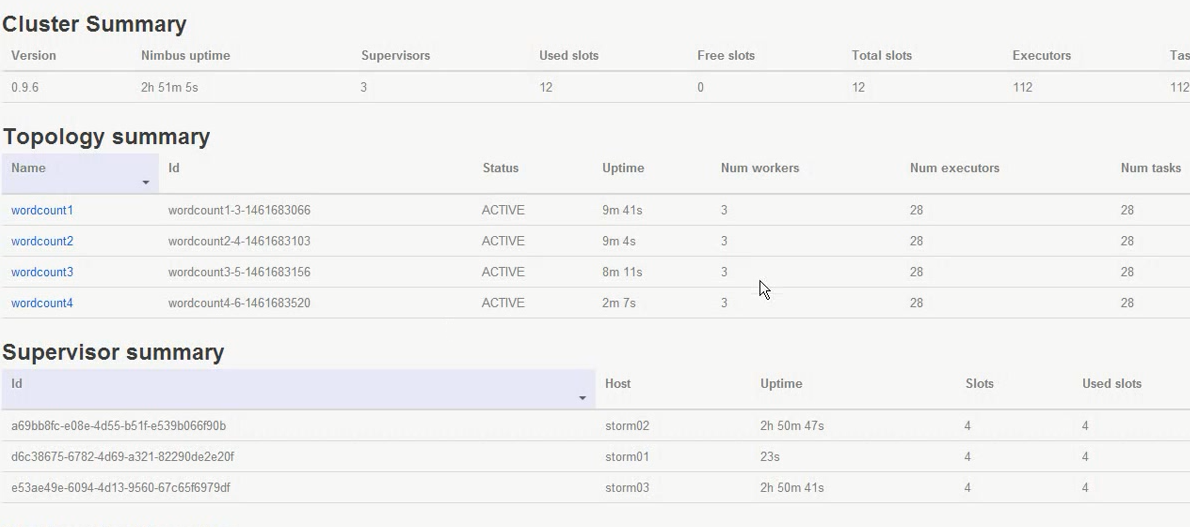
**当一个**topology**启动后它会占用一定的worker，如wordcount1占用了三个，那么只剩下5个worker可以用，所以你再启动两个同样的wordcount，worker就会不够，但是Wordcount3还是显示active，但是Wordcount3不会去运行因为没有分配worker，等集群资源空下下来，worker够了，它才会去执行。**

**也可以去扩展集群，增加Supervisor，如在storm01上再启动一个Supervisor：storm Supervisor，这时集群就多了四个worker，程序就可以继续运行了，先提交的topology就会被先分配好worker。也可以让已经提交的拓扑deactive，把资源空间让出来。**





**在storm01上再增加一个supervisor：**



## 9、流式计算一般架构图（重要）



* 其中flume用来获取数据。
* Kafka用来临时保存数据。
* Strom用来计算数据。
* Redis是个内存数据库，用来保存数据。