

# Spark计算框架详解

讲师: Yasaka



- kafka(移动的程控交换机获取数据)
- storm ( 实时计算 )
- hbase+zookeeper ( 数据存储 )
- servlet+ehcart(数据展示)
- get 'cell\_monitor\_table','29448-000001\_2015-07-24'













2015 天猫双11全球狂欢节  
2015 TMALL 11.11 GLOBAL SHOPPING FESTIVAL

2015年11月11日 24:00:00

**912.17 亿**

24:00:00 2015 11.11 Global Shopping Festival Total GMV is RMB 912.17 billion.

感谢亲爱的剁手党，感谢路上的快递员，感谢亿万客服，  
感谢每一位商家，感谢加班加点的技术宅，感谢每一张屏幕前的

Thank you to all the "shopaholics", all the delivery personnel, all the customer service professionals,  
all the participating merchants, all the engineers and everyone around the world  
for making this 2015 11.11 Global Shopping Festival a success.





## 天猫双11 | 销售过亿名单 截止 01:00

小米	杰克琼斯
海尔	林氏木业
韩都衣舍	GXG
骆驼	绫致集团
魅族	

2015天猫双11全球狂欢节  
全天交易额

**912.17 亿**

无线交易额

**626.42 亿**

无线占比

**68.67%**

这是消费的力量！  
是新经济的力量！  
是每个人参与的力量！  
更是中国的力量！

**感谢，感恩！**









- 什么是大数据计算框架？
- 公司里为什么需要大数据计算框架？
- 哪些技术属于大数据计算框架？
- 什么是Spark？解决了哪些问题？
- Spark架构是什么？特点是什么？
- 如何搭建Spark？如何运行Spark？
- 剖析API
- 运行样例程序
- 音乐推荐系统展示



# 什么是大数据计算框架？

## Hadoop



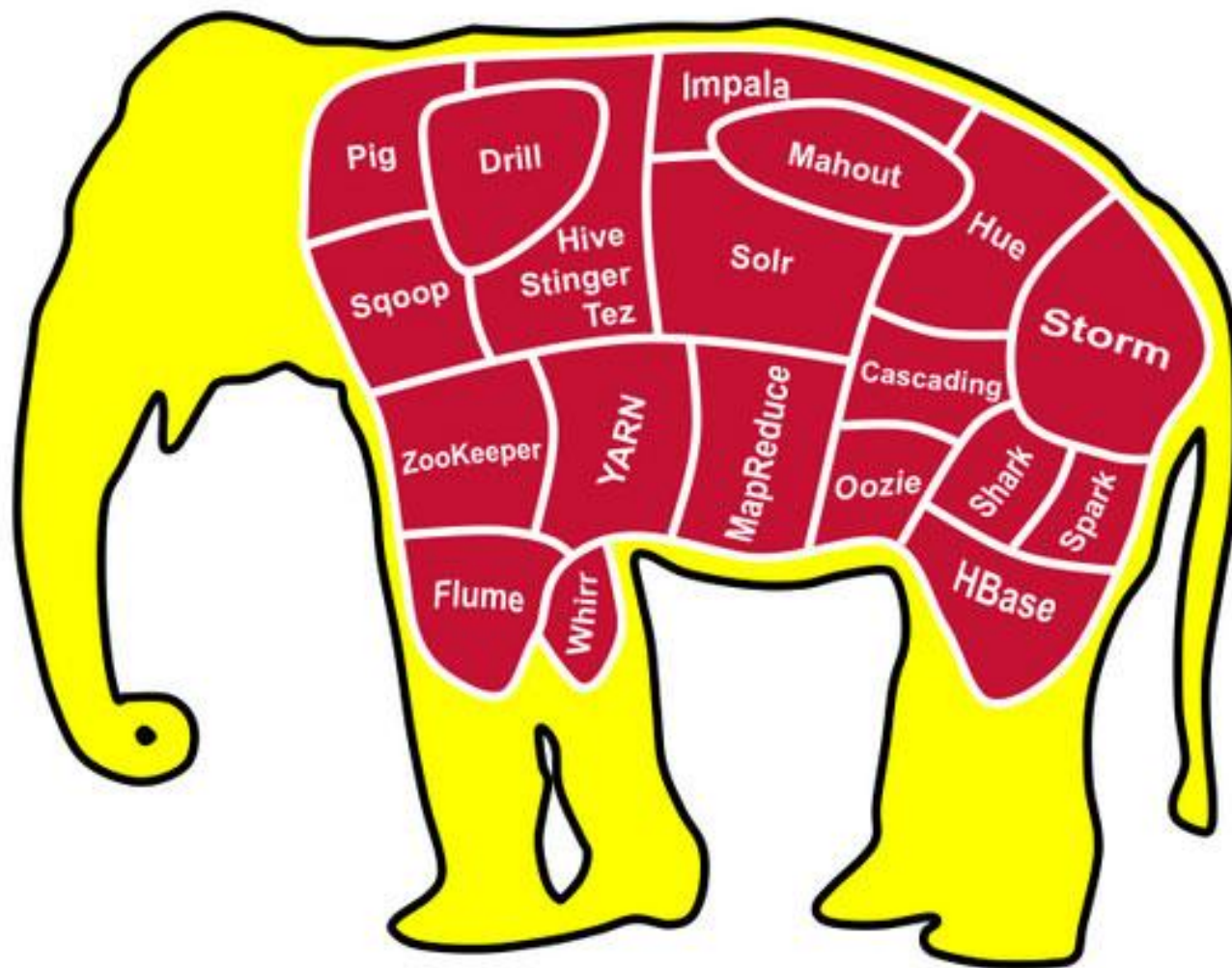
## Storm



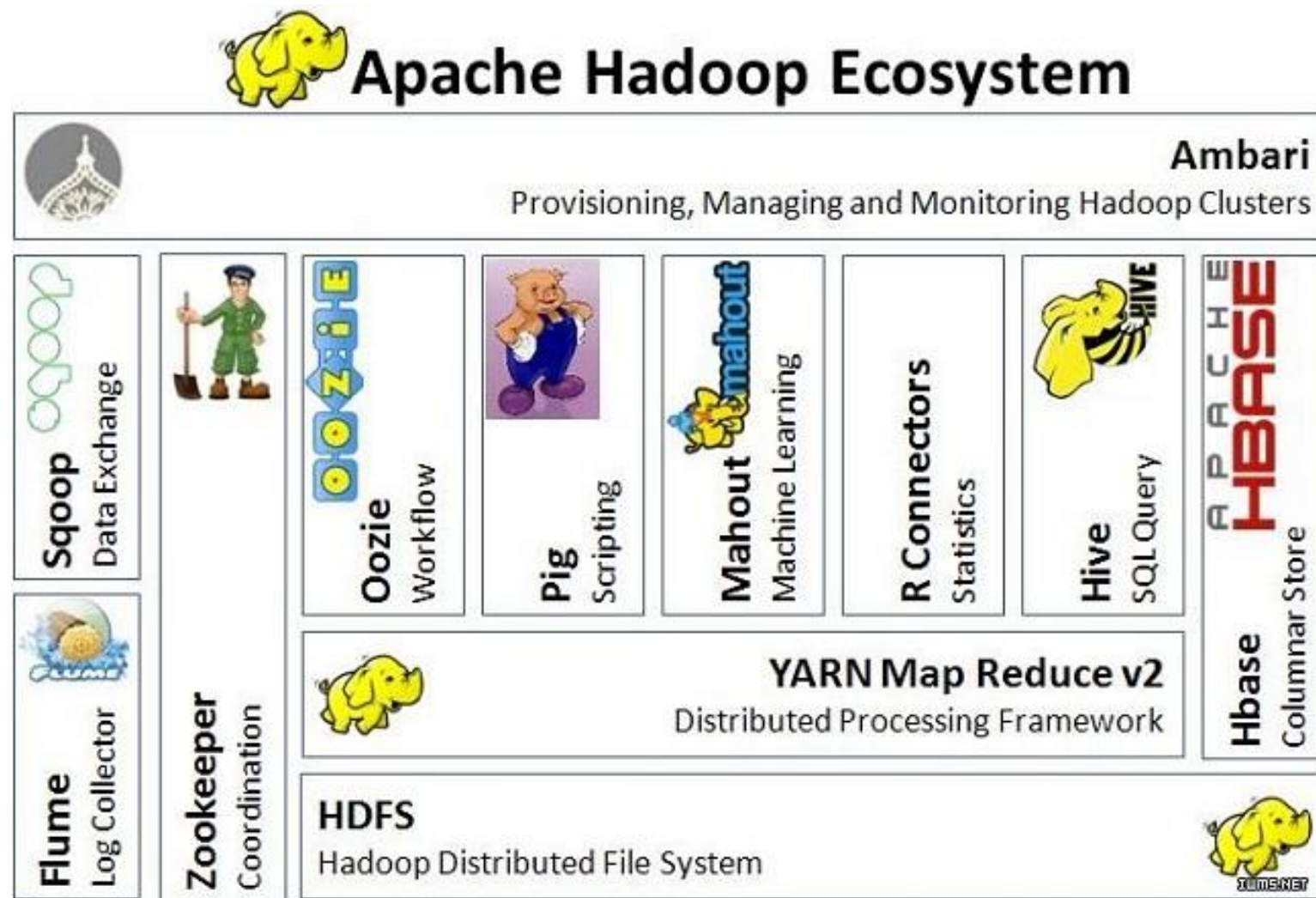


# 什么是大数据计算框架？

Apache Hadoop Ecosystem



# 什么是大数据计算框架？





# 公司里为什么需要大数据计算框架？

- 高并发
- 大数据
- 云计算
- 虚拟化







# 腾讯QQ在线人数统计







**今日推荐**  
11月11天

品牌休闲男装  
低至59元  
降温求保暖



618家跨店满减  
每天300件免单  
任性秒杀抢抢抢



冬季鞋服特卖  
大牌低至11元  
爽购11天  
闪购 >



11.11元还包邮  
充值券等你来拿  
专享剁手价  
团购 >



## 最佳组合



鸭鸭(YAYA)男士加厚外套保暖冬装羽绒服男 A-9  
¥299.00



鸭鸭(YAYA)冬装男士轻羽绒保暖男装羽绒服男  
¥199.00



鸭鸭2015秋冬男短款时尚羽绒马甲背心D-5502  
¥249.00



鸭鸭(yaya)男士短款加厚秋冬韩版羽绒服 A-5  
¥379.00



鸭鸭(YAYA)男款加厚带帽毛领 商务休闲羽绒服  
¥559.00

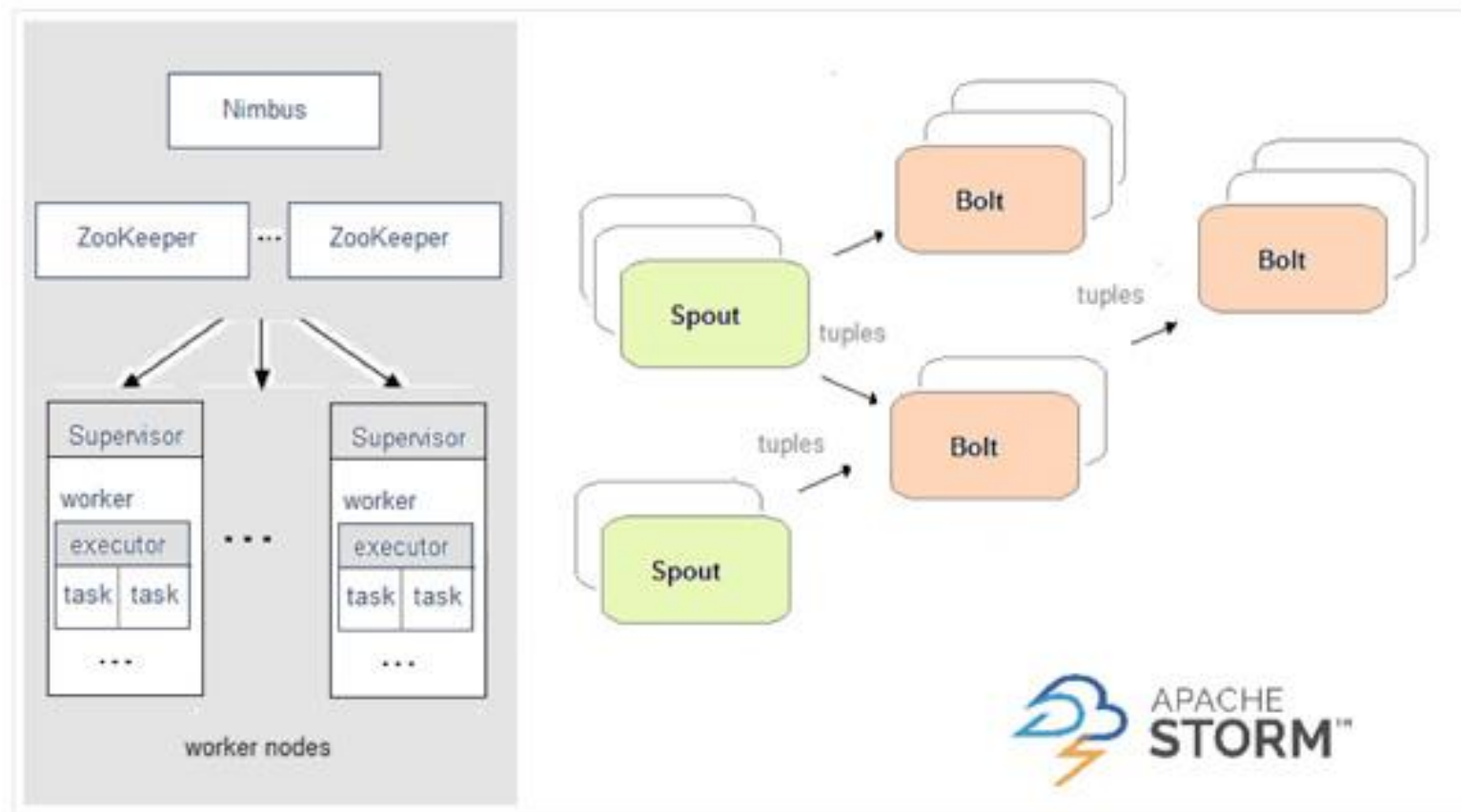


鸭鸭(YAYA)男款中老年款可脱卸帽90%鸭绒  
¥309.00



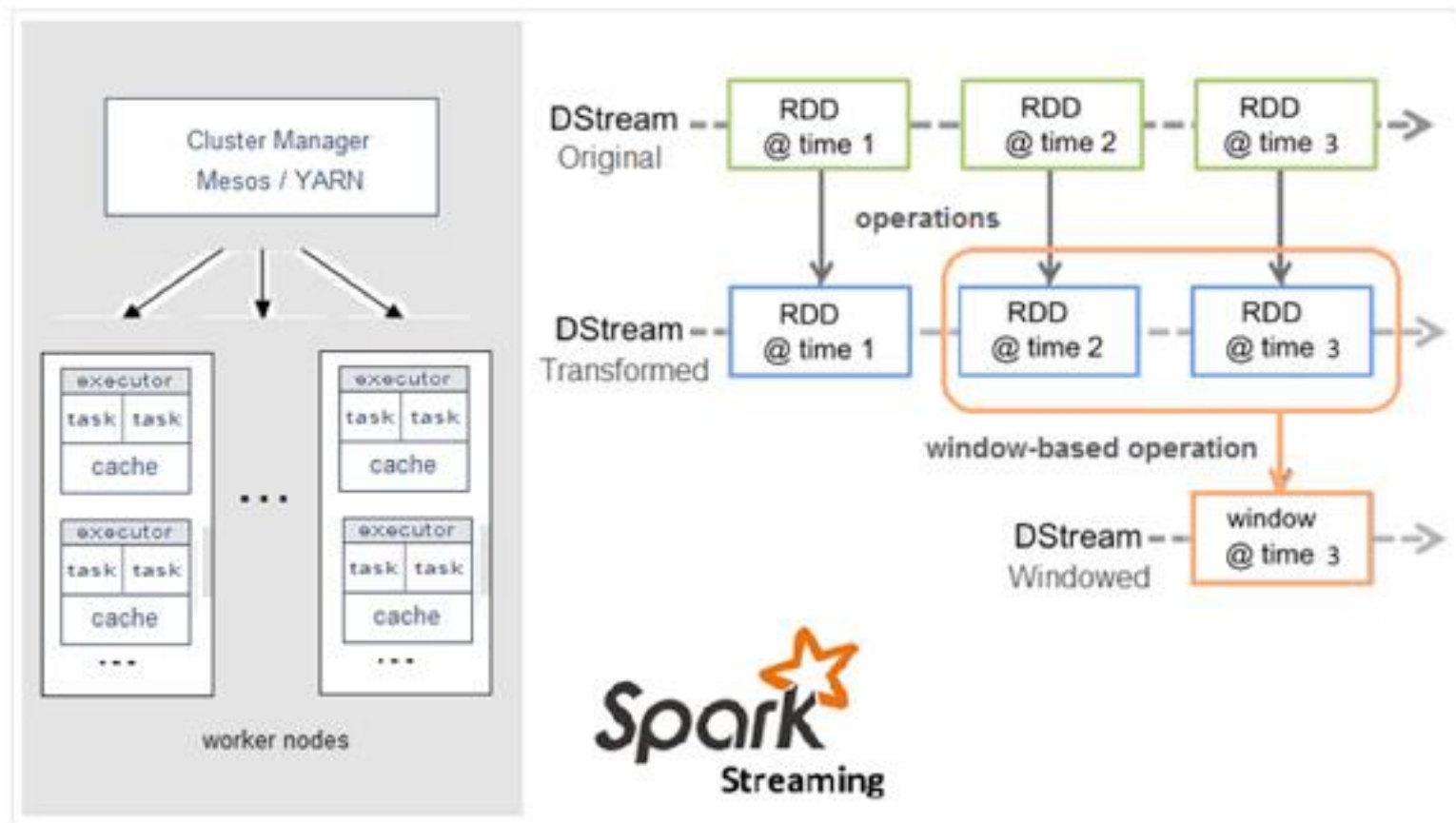
# 哪些技术属于大数据计算框架？

- Storm



# 哪些技术属于大数据计算框架？

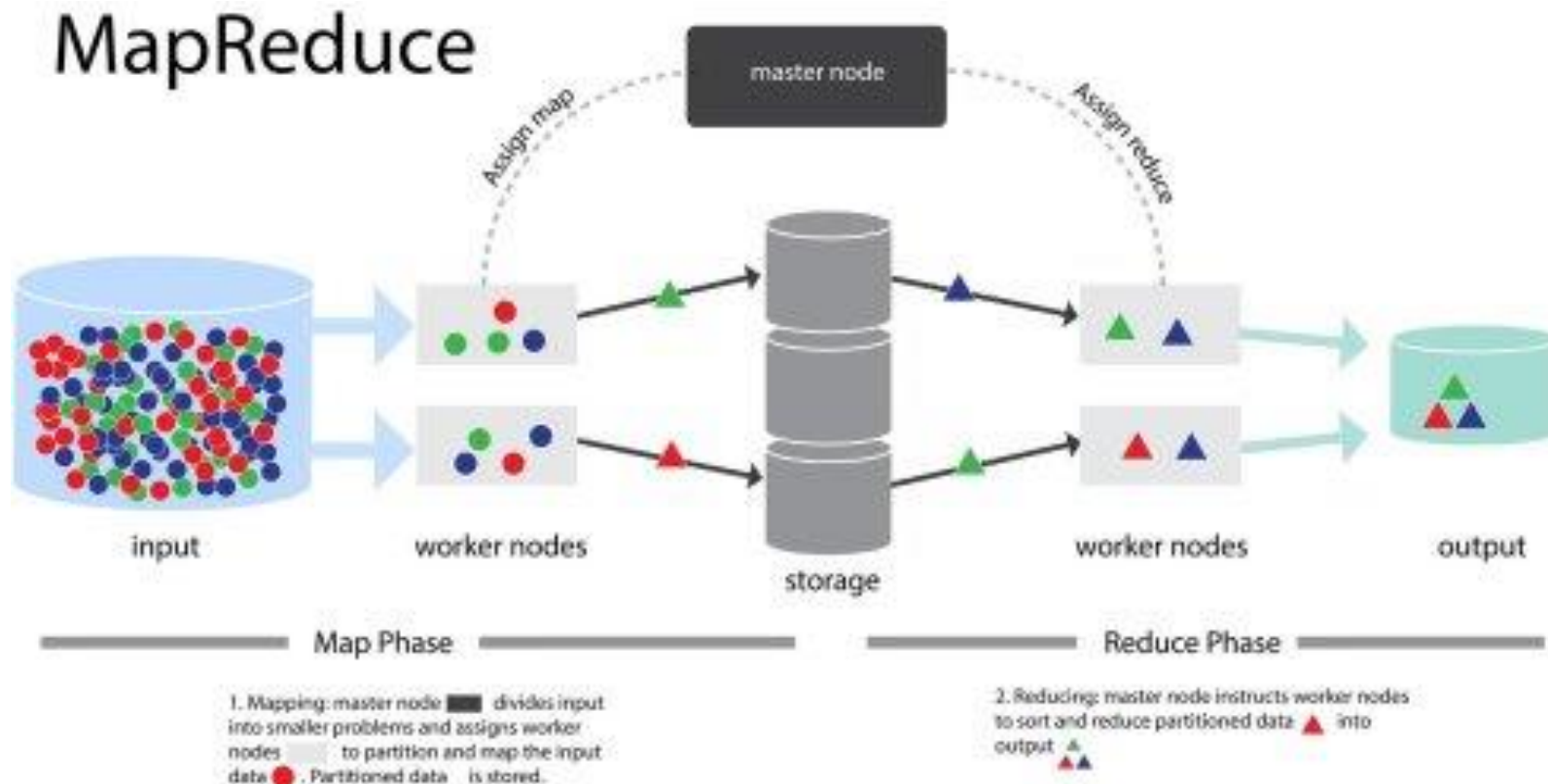
- Spark





# 哪些技术属于大数据计算框架？

- Hadoop→MapReduce+GFS+BigTable



- Spark，是一种通用的大数据计算框架，正如传统大数据技术Hadoop的MapReduce、Hive引擎，以及Storm流式实时计算引擎等
- 2009年，Spark诞生于伯克利大学的AMPLab实验室；2010年，伯克利大学正式开源了Spark项目；2013年，Spark成为了Apache基金会下的项目；2014年，Spark以飞快的速度称为了Apache的顶级项目；2015年~，Spark在国内IT行业变得愈发火爆
- 支持多语言



- Spark包含了大数据领域常见的各种计算框架：比如Spark Core用于离线计算，Spark SQL用于交互式查询，Spark Streaming用于实时流式计算，Spark MLlib用于机器学习，Spark GraphX用于图计算。
- Spark主要用于大数据的计算，而Hadoop以后主要用于大数据的存储（比如HDFS、Hive、HBase等），以及资源调度（Yarn）。
- Spark+Hadoop的组合，是未来大数据领域最热门的组合，也是最有前景的组合！



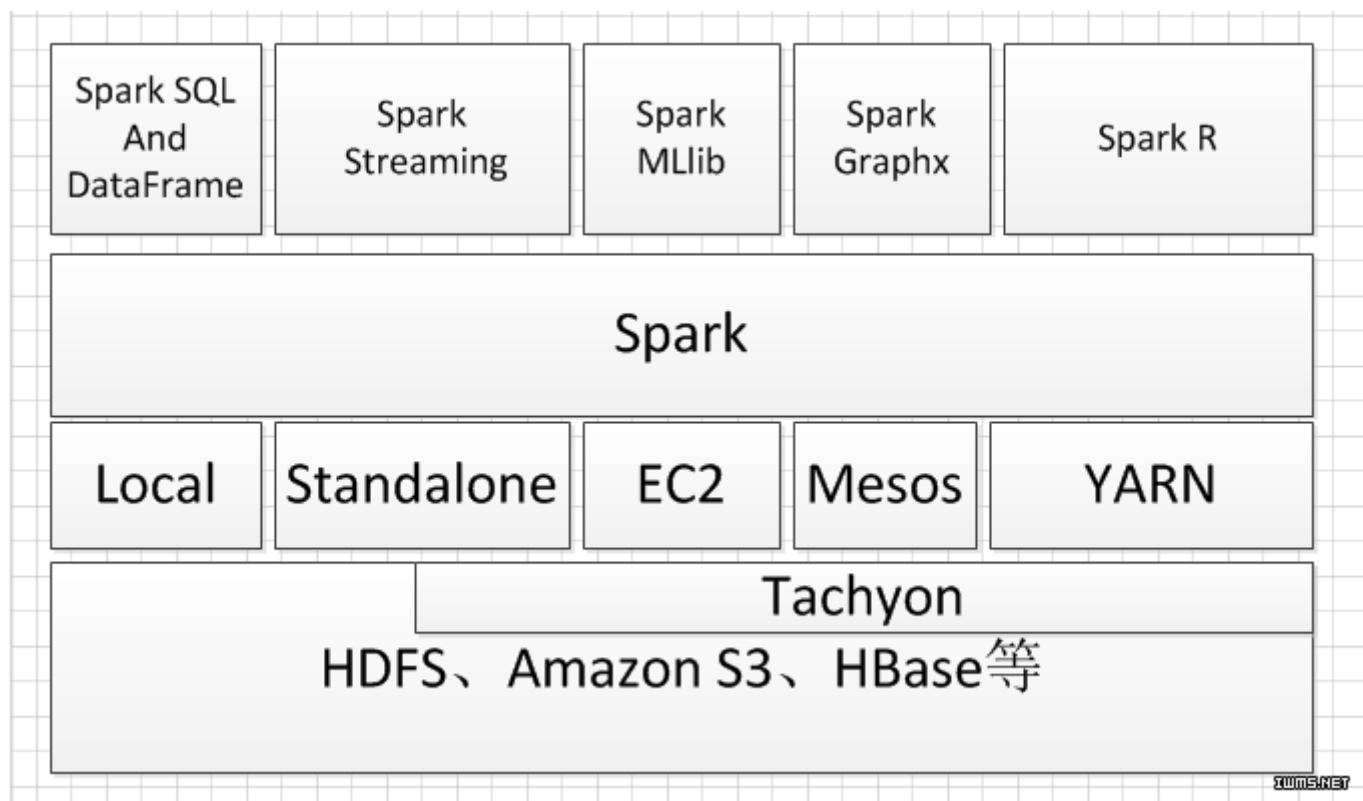


# 解决了哪些问题？

- 互联网广告实时流量统计
- 互联网数据质量实时监控
- 交通超速频发路段监控
- 交通基于GPS的实时路况分析
- 移动互联语音实时墙
- 运营商网络流量流向实时分析
- 中国移动小区基站预警
- 建设智慧城市
- 推荐系统构建



# Spark架构是什么？

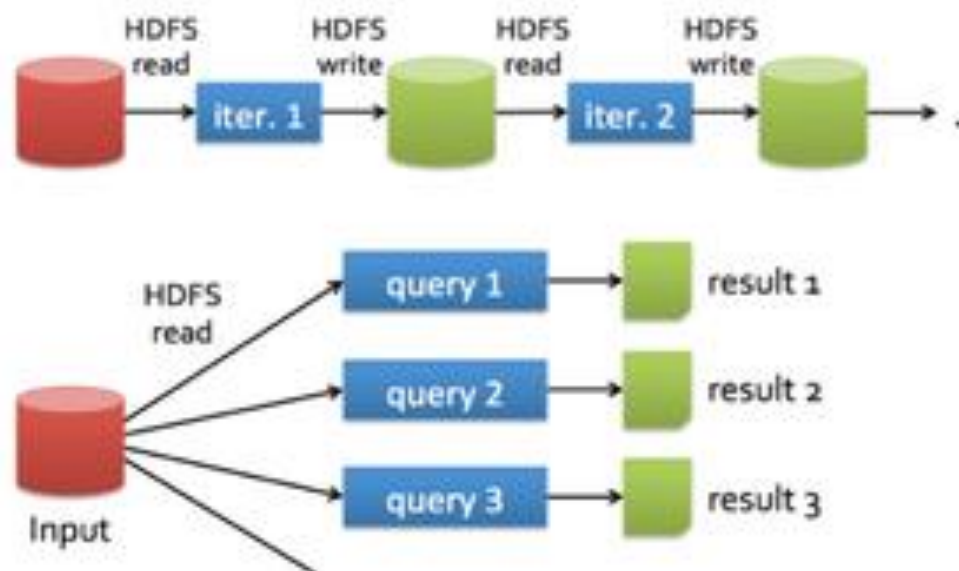


- 速度快
- 容易上手开发
- 超强的通用性
- 集成hadoop
- 很高的活跃度

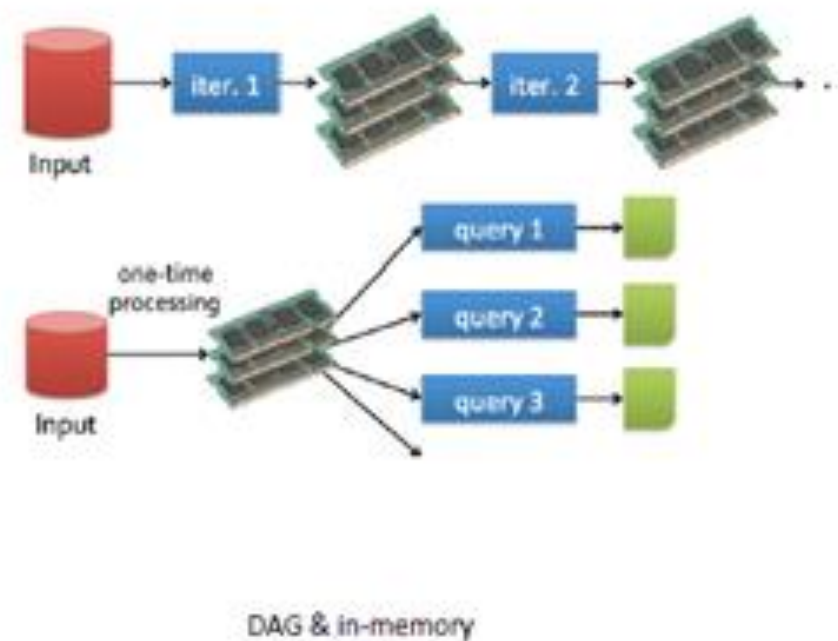


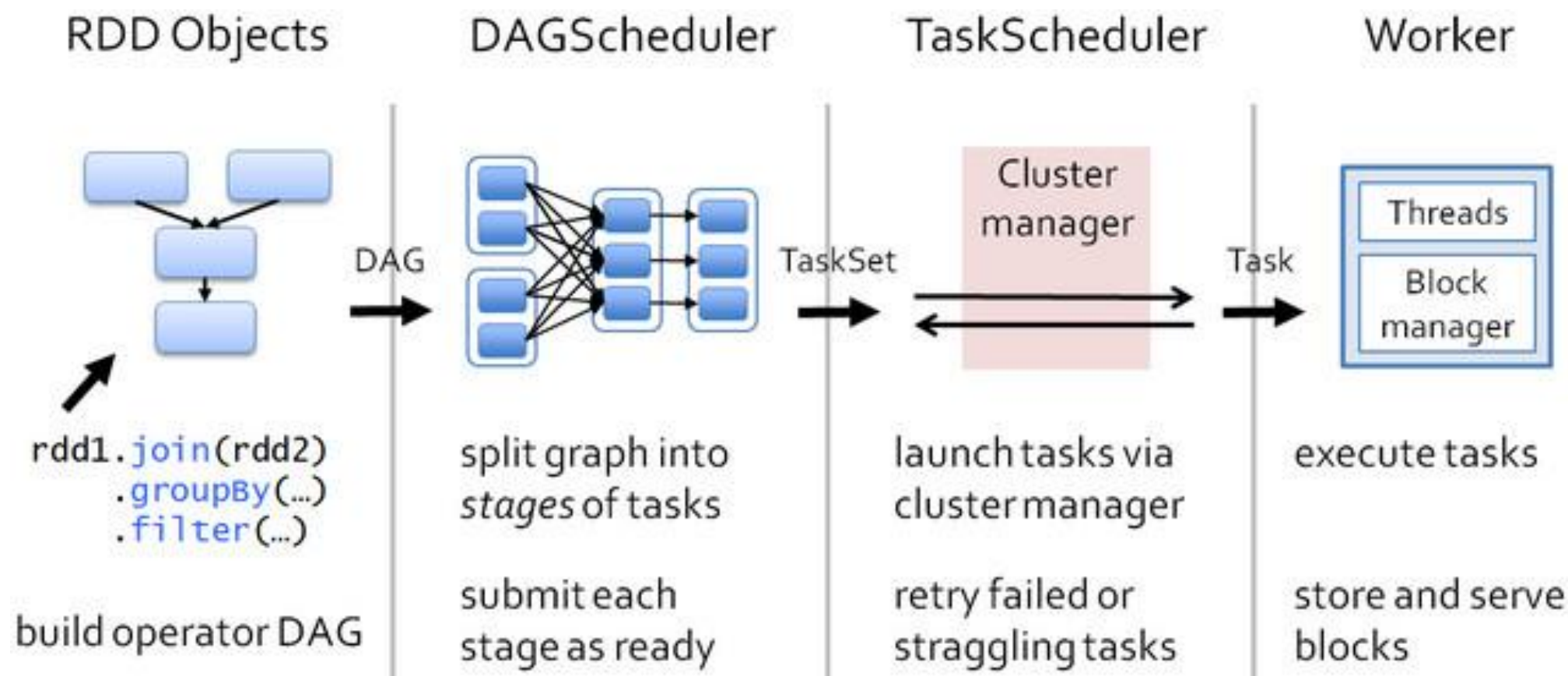


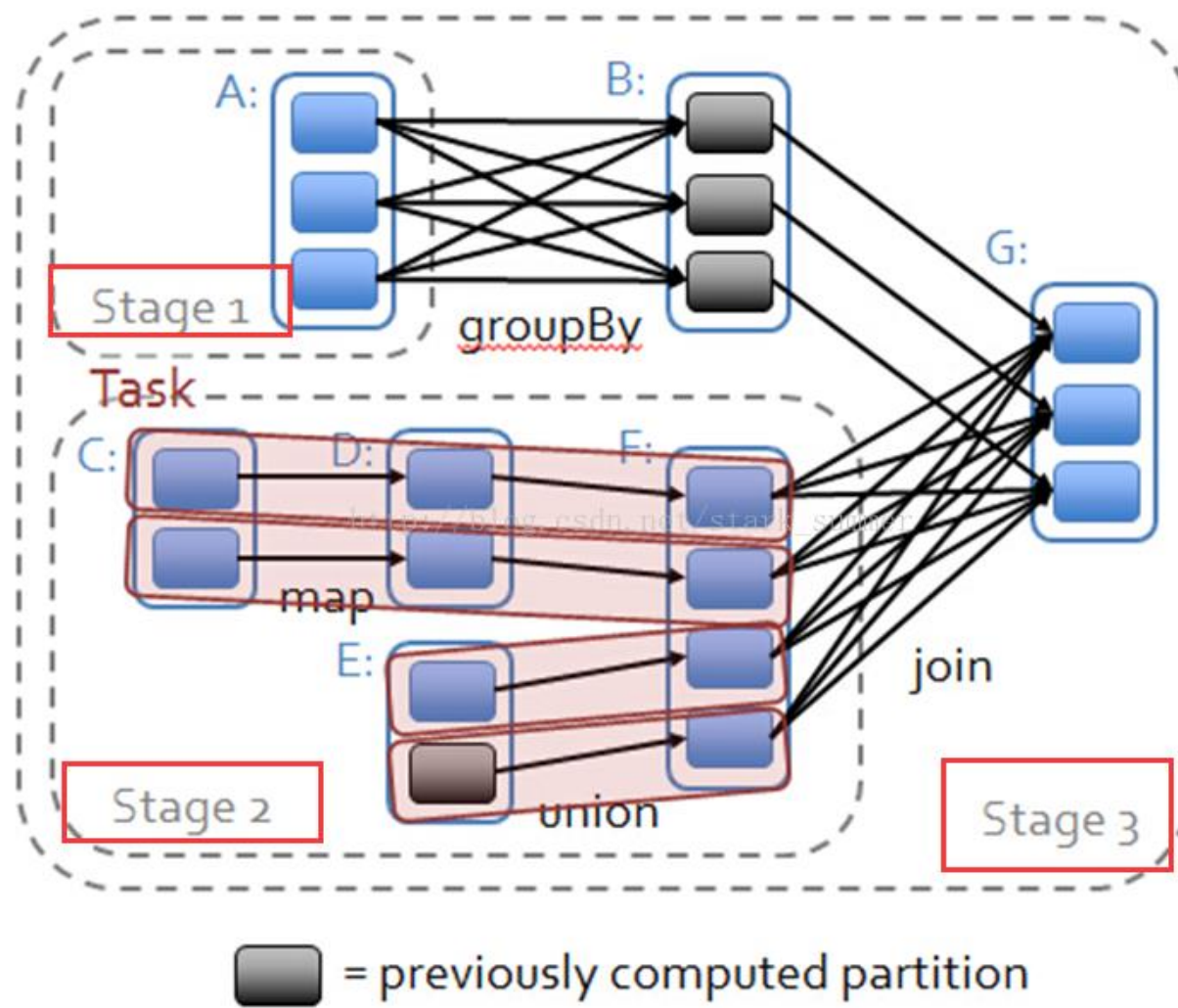
## Hadoop Data Sharing



## Spark Data Sharing









- 单机模式
- 独立集群→原生集群模式
- 验证：访问<http://localhost:8080>，运行Pi example
- YARN集群
  - Cluster
  - Client
- 验证：访问<http://localhost:8088>，运行Pi example



- local单机模式：
- `./bin/spark-submit --class org.apache.spark.examples.SparkPi --master local[1] ./lib/spark-examples-1.3.1-hadoop2.4.0.jar 100`
- standalone集群模式：
- `./bin/spark-submit --class org.apache.spark.examples.SparkPi --master spark://spark001:7077 --executor-memory 1G --total-executor-cores 1 ./lib/spark-examples-1.3.1-hadoop2.4.0.jar 100`
- Yarn集群模式：
- `./bin/spark-submit --class org.apache.spark.examples.SparkPi --master yarn-cluster --executor-memory 1G --num-executors 1 ./lib/spark-examples-1.3.1-hadoop2.4.0.jar 100`



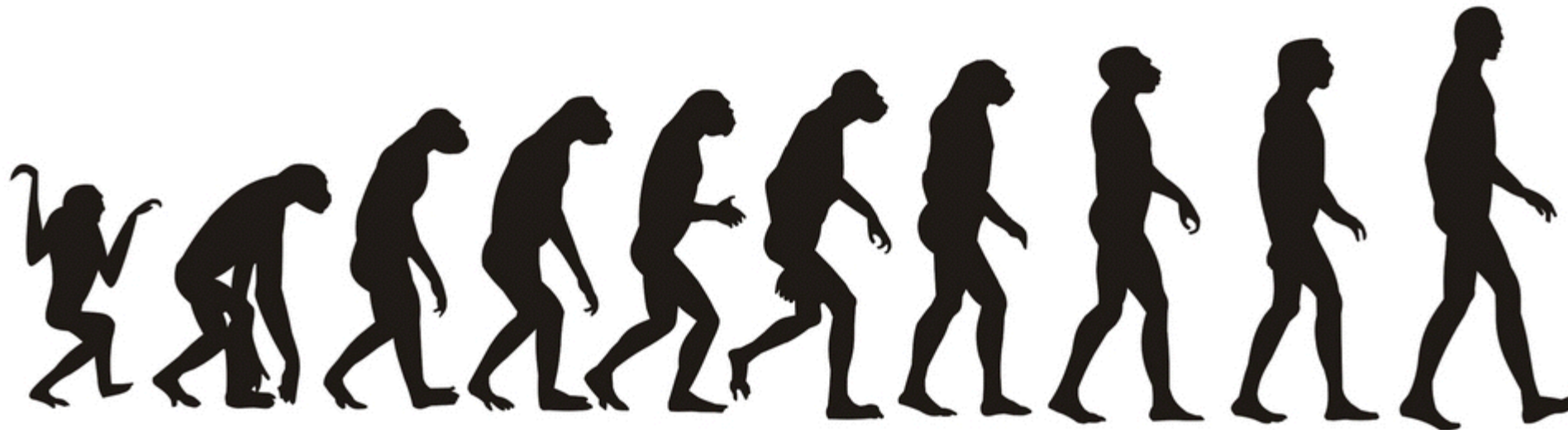
- SparkContext / RDD

<b>Transformations</b>	$\text{map}(f : T \Rightarrow U) : \text{RDD}[T] \Rightarrow \text{RDD}[U]$ $\text{filter}(f : T \Rightarrow \text{Bool}) : \text{RDD}[T] \Rightarrow \text{RDD}[T]$ $\text{flatMap}(f : T \Rightarrow \text{Seq}[U]) : \text{RDD}[T] \Rightarrow \text{RDD}[U]$ $\text{sample}(\text{fraction} : \text{Float}) : \text{RDD}[T] \Rightarrow \text{RDD}[T]$ (Deterministic sampling) $\text{groupByKey}() : \text{RDD}[(K, V)] \Rightarrow \text{RDD}[(K, \text{Seq}[V])]$ $\text{reduceByKey}(f : (V, V) \Rightarrow V) : \text{RDD}[(K, V)] \Rightarrow \text{RDD}[(K, V)]$ $\text{union}() : (\text{RDD}[T], \text{RDD}[T]) \Rightarrow \text{RDD}[T]$ $\text{join}() : (\text{RDD}[(K, V)], \text{RDD}[(K, W)]) \Rightarrow \text{RDD}[(K, (V, W))]$ $\text{cogroup}() : (\text{RDD}[(K, V)], \text{RDD}[(K, W)]) \Rightarrow \text{RDD}[(K, (\text{Seq}[V], \text{Seq}[W]))]$ $\text{crossProduct}() : (\text{RDD}[T], \text{RDD}[U]) \Rightarrow \text{RDD}[(T, U)]$ $\text{mapValues}(f : V \Rightarrow W) : \text{RDD}[(K, V)] \Rightarrow \text{RDD}[(K, W)]$ (Preserves partitioning) $\text{sort}(c : \text{Comparator}[K]) : \text{RDD}[(K, V)] \Rightarrow \text{RDD}[(K, V)]$ $\text{partitionBy}(p : \text{Partitioner}[K]) : \text{RDD}[(K, V)] \Rightarrow \text{RDD}[(K, V)]$
<b>Actions</b>	$\text{count}() : \text{RDD}[T] \Rightarrow \text{Long}$ $\text{collect}() : \text{RDD}[T] \Rightarrow \text{Seq}[T]$ $\text{reduce}(f : (T, T) \Rightarrow T) : \text{RDD}[T] \Rightarrow T$ $\text{lookup}(k : K) : \text{RDD}[(K, V)] \Rightarrow \text{Seq}[V]$ (On hash/range partitioned RDDs) $\text{save}(\text{path} : \text{String}) : \text{Outputs RDD to a storage system, e.g., HDFS}$





- spark-shell 操作 wordcount



```
Rating(2093760,2814,0.030518022934870874)
Rating(2093760,1001819,0.029811221324290038)
Rating(2093760,1300642,0.029380838727060564)
Rating(2093760,4605,0.028917679455914086)
Rating(2093760,1007614,0.028859793426105194)
```

```
Some((2814,50 Cent))
Some((4605,Snoop Dogg))
Some((1007614,Jay-Z))
Some((1001819,2Pac))
Some((1300642,The Game))
```

10771838	2093760	1180	1
10771839	2093760	1255340	3
10771840	2093760	378	1
10771841	2093760	813	2
10771842	2093760	942	7



- 大家如果通过本堂课的讲解，能够较为全面地对Spark有一个感性得认识，就能意识到，Spark在大数据领域中，是未来的一个趋势和方向！
- Spark目前正在变得越来越火爆，招聘的企业正在越来越多，而且目前国内spark人才可以说是稀缺！！！在目前，以及未来，完全供不应求！因此这种趋势，以及这种现状，就决定了，对于我们个人来说，目前进行spark的学习以及研究，完全是未来一个获取快速升值的机会！！！！





- 使用最新版本
- 从零起步
- 涵盖Spark所有功能
- 一线互联网项目实战
- 结合源码对Spark内核进行深度剖析
- 全程配图详解
- 讲解Spark性能调优



- Java / J2EE开发工程师
- Hadoop开发工程师
- Spark入门级别的，或者只有一定基础的
- 在校或者刚毕业的学生



# 大数据，三要！！！！

- 要钱
- 要脸
- 要命

