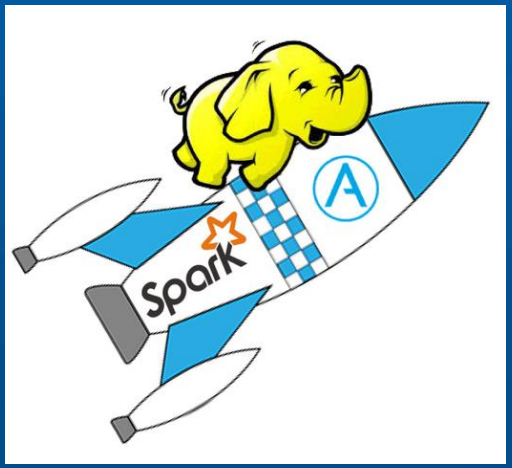
Spark Streaming 实时流计算　20:30开始

高级互联网大数据架构师：Yasaka



QQ群： **172599077 , 156927834**

北京尚学堂大数据极限班课程官网地址：
http://www.bjsxt.com/html/cloud/

好消息！！！ 大数据登陆上海滩！！！

(北京)大数据线下班将于5月6日开班！火热报名中！！！

(上海)大数据线下班将于6月21日开班！火热报名中！！！

-- 老师面授课程！传统式教室教学已开班多期！学习完美就业！

(北京)大数据周末班将于5月7日再次开班！火热报名中！！！

线上班Hadoop阶段推出后，第二阶段Spark线上班5月22日正式推出！！！

--附送随堂讲课视频

贾老师：1786418286

何老师：1926106490

詹老师：2805048645

讨论技术可以加入以下QQ群： 172599077，156927834

第一阶段linux+搜索+hadoop体系

Linux基础 → shell编程 → 高并发架构 → lucene,solr搜索 → hadoop体系 →

HDFS → mapreduce → hbase → zookeeper → hive →

flume → sqoop → 项目实战一

第二阶段机器学习

R语言 → mahout → 项目实战二

第三阶段storm流式计算

kafka → storm → redis → 项目实战三

第四阶段spark内存计算

scala编程 → spark core → spark sql → spark streaming → spark mllib →

spark graphx → 项目实战四 → python机器学习 → spark python编程 → 项目实战五 →

项目实战六

第五阶段云计算平台

docker → kvm → openstack云计算 → 就业指导

教学多重保障：

1，贯彻实战教育理念

2，每节随堂笔记，有图有代码

3，提供服务器配置，搭建步骤说明

4，有问题老师一对一辅导

5，同学们良好的学习氛围

6，尚学堂科技有限公司是一个实体公司，已做教育多年，有着良好的口碑

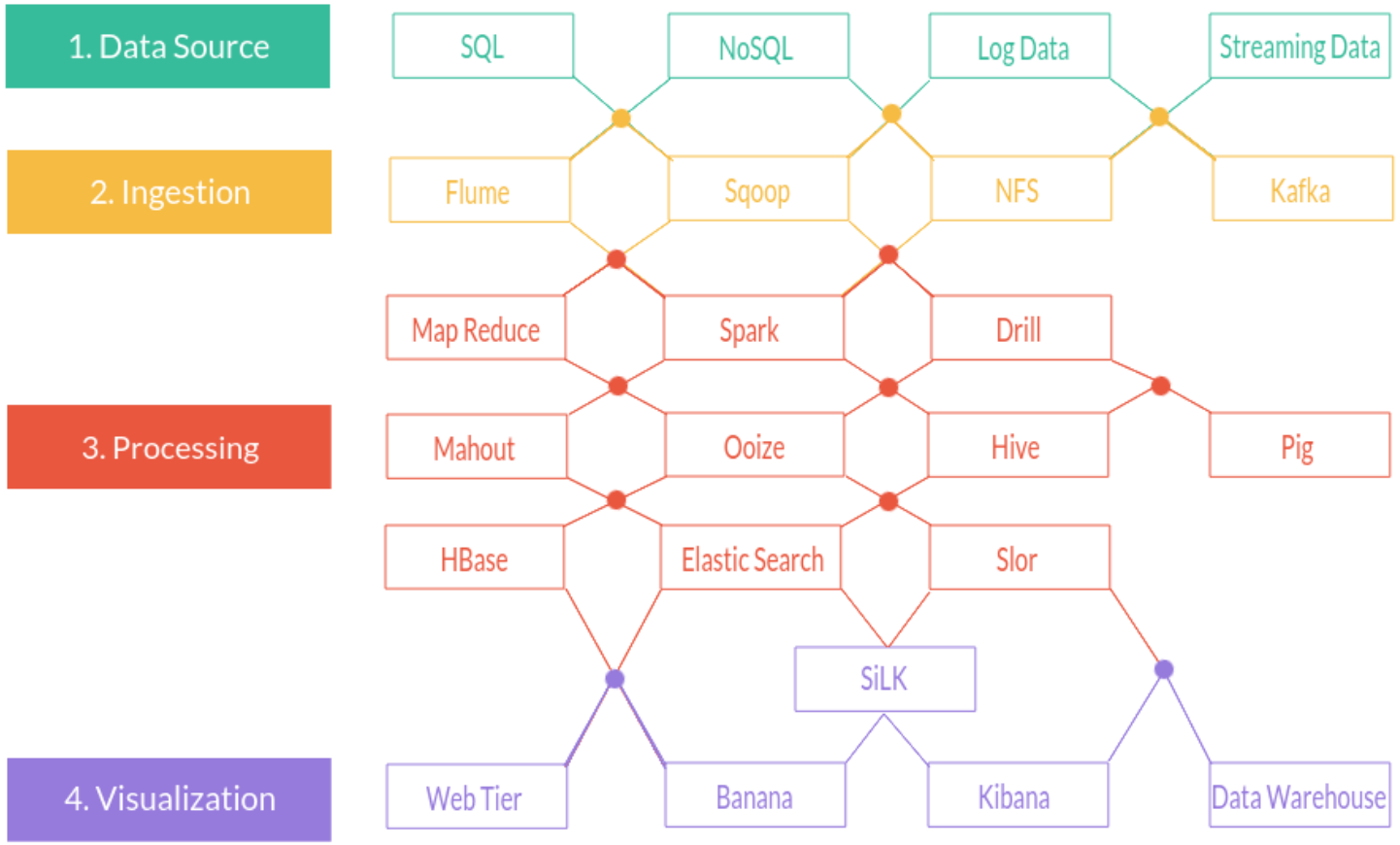
7，尚学堂大数据班老师均有丰富的授课经验，线下线上课都有

官网：

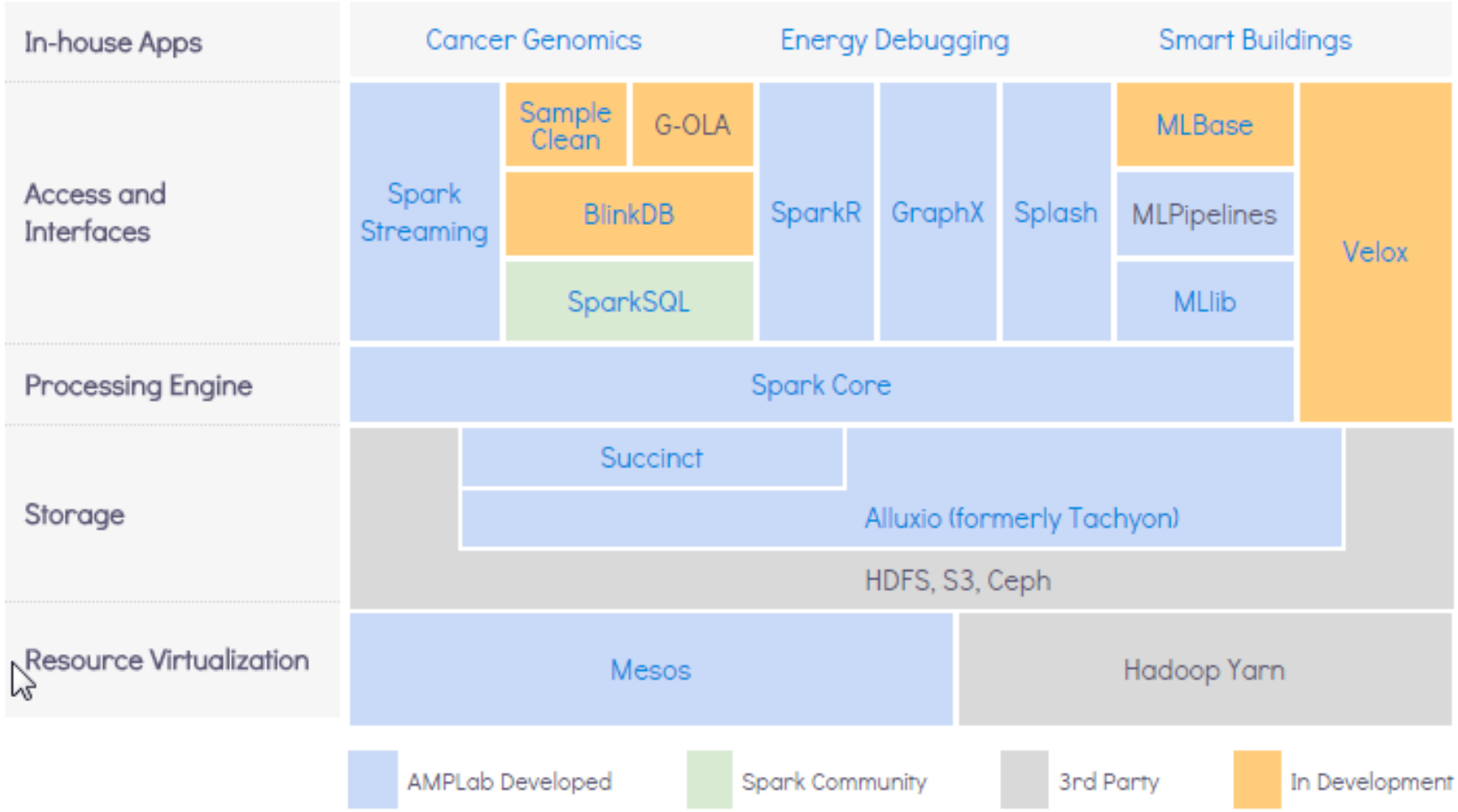
http://www.bjsxt.com/html/cloud/

- 分享主题内容
  - 1.Spark在大数据生态的位置
  - 2.Spark生态系统
  - 2.实时流计算解决的问题
  - 3.大数据生态里实时流计算都有哪些
  - 4.着重对比Spark Streaming 与 Storm
  - 5.Spark Streaming架构和原理
  - 6.Spark Streaming的使用详解
  - 7.Spark Streaming实战
  - 8.Spark Streaming为何如此令人着迷

- Is Apache Spark going to replace Hadoop?



- http://spark.apache.org/



- QQ在线


（图为一分钟前中国区域QQ同时在线分布）
2 : 15 : 00
QQ 同时在线人数
161,585,894
历史最高在线：259,478,960
昨天 02:15
今天 02:14

- 天猫双十一大屏幕


全球交易国家/地区排行
Top Countries/Regions by GMV
进口国家/地区排行
Top Countries/Regions Selling to China
出口国家/地区排行
Top Countries/Regions Buying from China
1 日本 JP
2 美国 US
3 韩国 KR
4 澳大利亚 AU
5 德国 DE
1 香港 HK
2 美国 US
3 中国台湾 TW
4 澳门 MO
5 澳大利亚 AU


菜鸟快递网络
Cainiao Express Delivery Network
467,764,069
120,576,789
617
17.675
913
731
548
366
183


天猫双11 天猫成交额 变化图
（单位：元）
最终成交额
912亿
00:01 10亿
00:03 30亿
00:05 47亿
00:12 100亿
00:20 146亿
00:30 191亿
00:40 215亿
01:00 247亿
01:30 317亿
07:45 417亿
09:52 500亿
11:49 571亿
12:49 600亿
13:00 605亿 无线占比 71.22%
14:00 633亿 无线占比 70.88%
15:00 658亿 无线占比 70.52%
16:00 683亿 无线占比 70.09%
16:39 700亿 无线占比 69%
17:22 717亿 无线占比 69%
18:20 737亿 无线占比 69%
21:00 800亿 无线占比 68%
22:32 850亿 无线占比 68%
23:47 900亿 无线占比 68%
24:00 912亿

- 春运实时迁徙


北京8小时迁徙图- 腊月三十 11点前
Baidu 地图
01.30 腊月三十(除
11 点
北京
迁入热省
迁出热省
1 河北 → 北京 402‰
2 四川 → 北京 201‰
3 河南 → 北京 64‰
4 天津 → 北京 49‰
5 山东 → 北京 37‰
6 广东 → 北京 35‰
7 江苏 → 北京 29‰
8 安徽 → 北京 23‰
9 山西 → 北京 22‰
TAKEFOTO CN


全国8小时迁徙图- 腊月廿八 15点前

- Storm / Trident /Spark Streaming / Samza / Flink

Streaming Model	Native	Micro-batching	Micro-batching	Native	Native
API	Compositional		Declarative	Compositional	Declarative
Guarantees	At-least-once	Exactly-once	Exactly-once	At-least-once	Exactly-once
Fault Tolerance	Record ACKs		RDD based Checkpointing	Log-based	Checkpointing
State Management	Not build-in	Dedicated Operators	Dedicated DStream	Stateful Operators	Stateful Operators
Latency	Very Low	Medium	Medium	Low	Low
Throughput	Low	Medium	High	High	High
Maturity	High		High	Medium	Low

- Spark vs. Storm

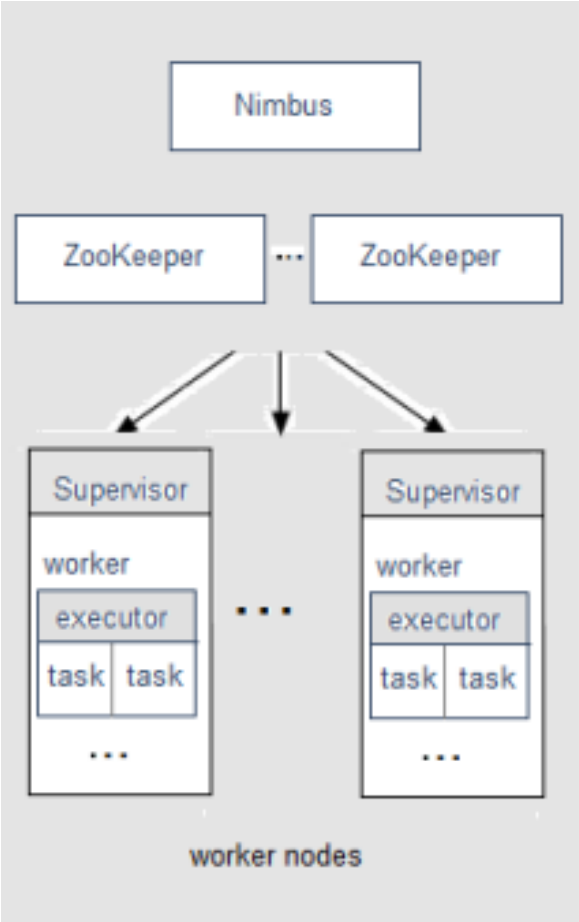
Hadoop compatibility

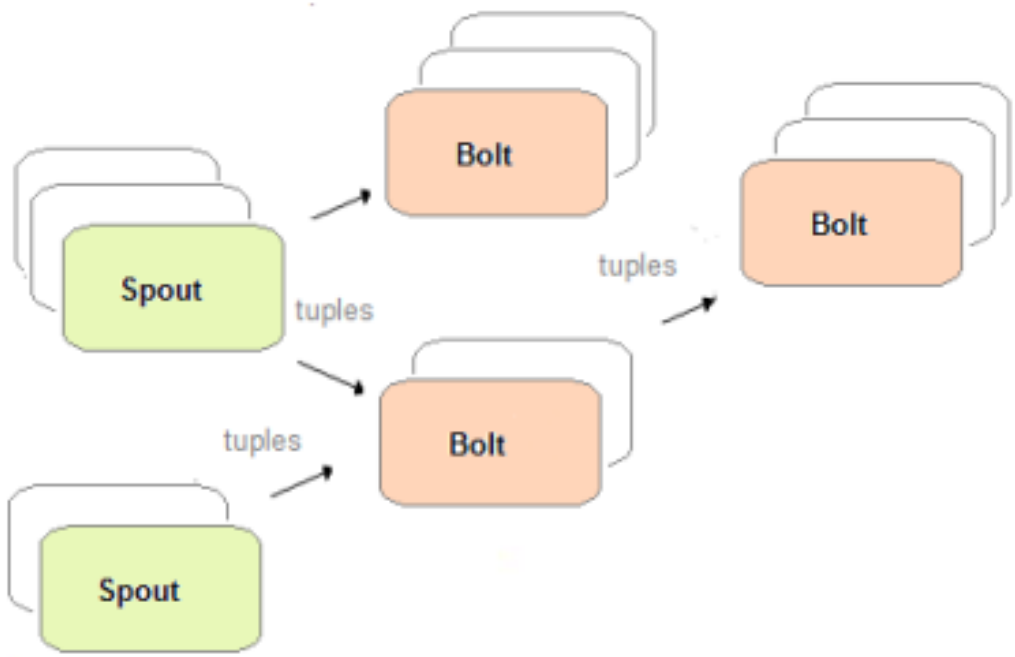
	Spark	Storm
Data sources	HDFS, Hbase, Cassandra	HDFS, Hbase, Kafka
Ressource Manager	YARN, Mesos	Mesos
Latency	Few seconds	< 1 second
Fault tolerance (every record processed)	Exactly once	At least once
Reliability	Improved reliability (Spark + YARN)	Guarantees no data loss (Storm + Kafka)

Apache Storm

Spark

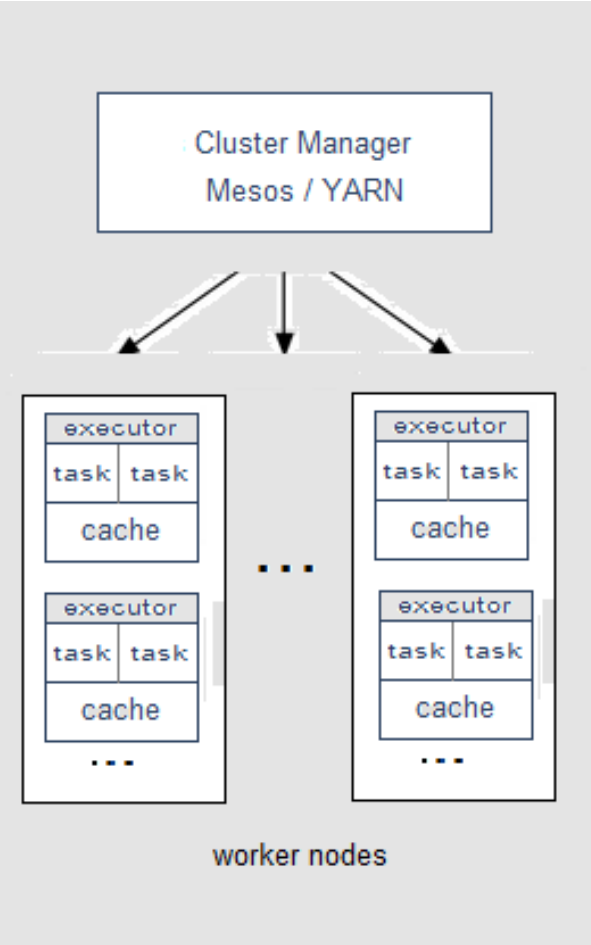
- Storm架构

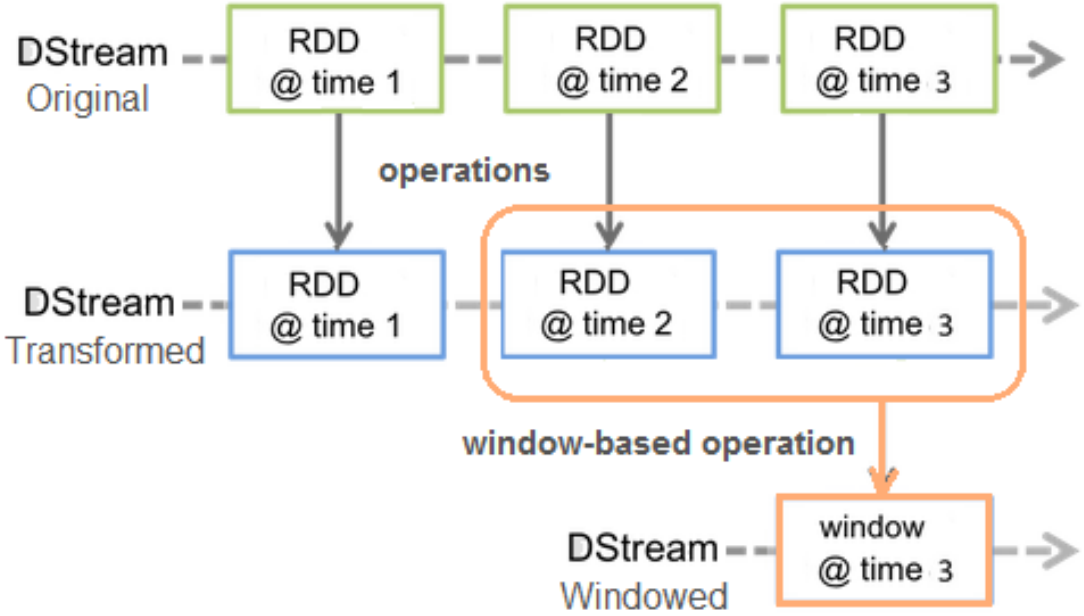




APACHE STORM™

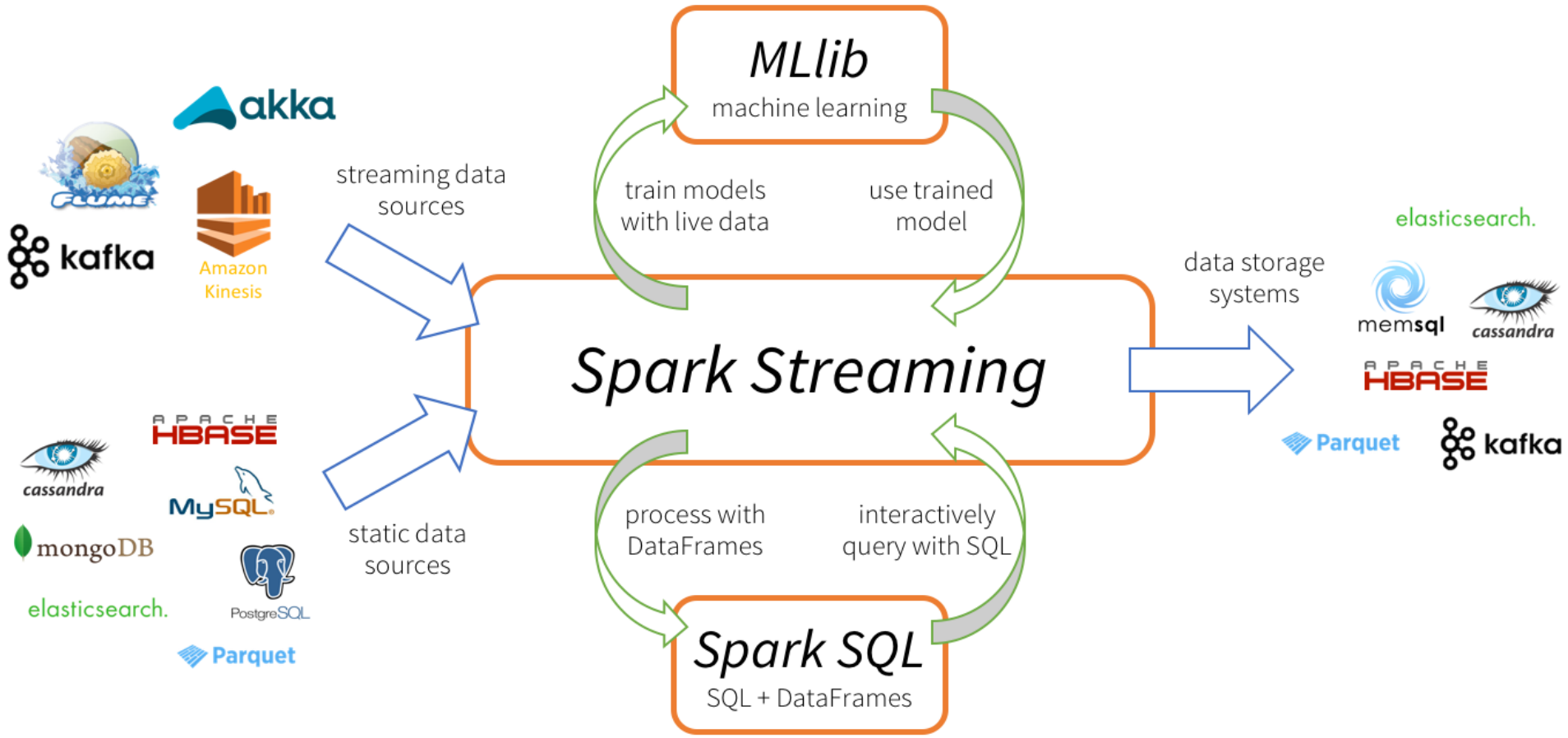
- Spark Streaming架构 http://spark.apache.org/streaming/





Spark Streaming

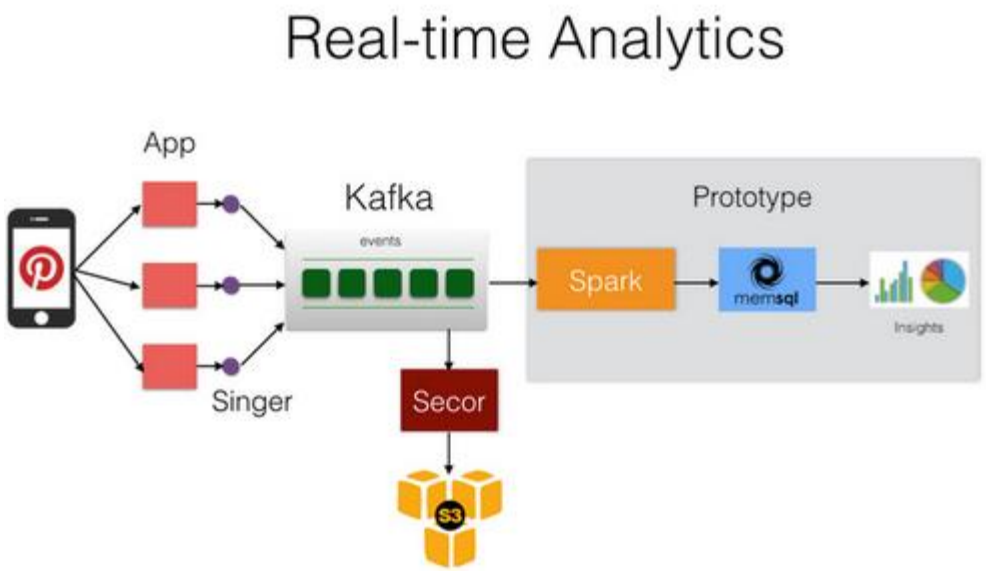
- ^_^

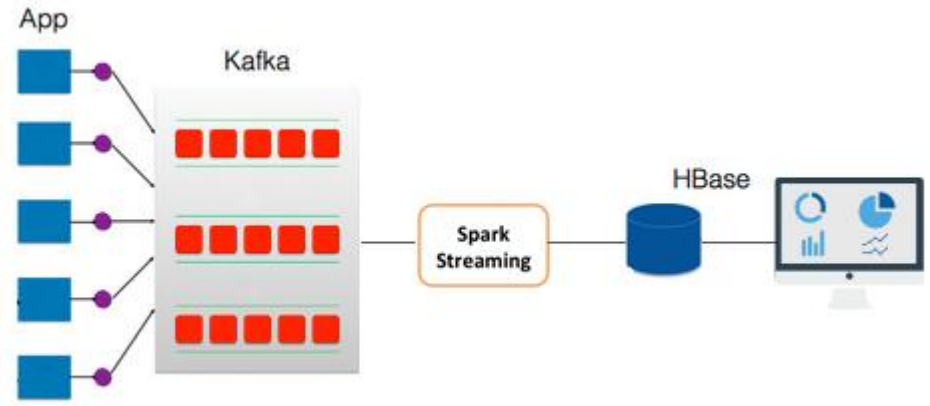

MLlib
machine learning
train models with live data
use trained model
Spark Streaming
Spark SQL
SQL + DataFrames
process with DataFrames
interactively query with SQL
streaming data sources
static data sources
data storage systems
akka
kafka
Amazon Kinesis
cassandra
HBASE
MySQL
mongoDB
elasticsearch.
PostgreSQL
Parquet
memsql
kafka

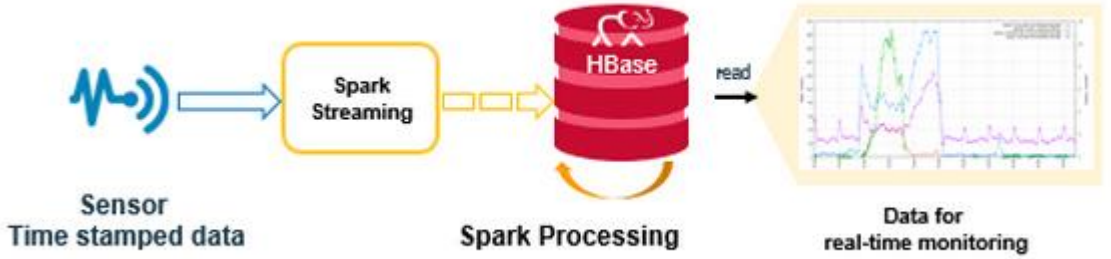
- http://spark.apache.org/docs/latest/streaming-programming-guide.html

```scala
import org.apache.spark._
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext._ // not necessary since Spark 1.3

// Create a local StreamingContext with two working thread and batch interval of 1 second.
// The master requires 2 cores to prevent from a starvation scenario.

val conf = new SparkConf().setMaster("local[2]").setAppName("NetworkWordCount")
val ssc = new StreamingContext(conf, Seconds(1))
```

- 实战案例
  - Kafka + Spark Streaming + HBase







- Beyond your imagination

Combine machine learning with streaming

Learn models offline, apply them online

```
// Learn model offline
val model = KMeans.train(dataset, ...)

// Apply model online on stream
kafkaStream.map { event =>
    model.predict(event.feature)
}
```

- Come here Quick ! Do not hesitate !

- Come here Quick ! Do not hesitate !