



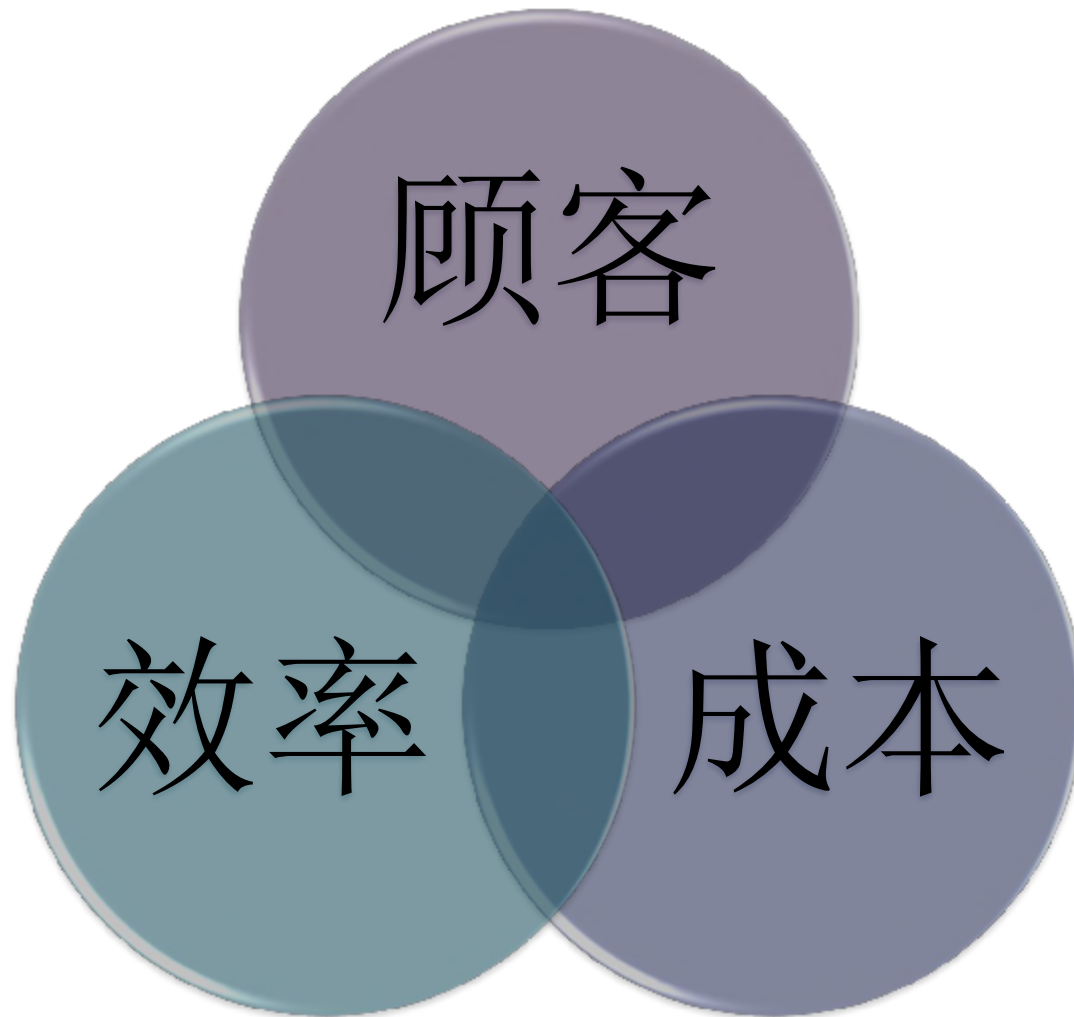
**SDCC 2016**  
中国软件开发者大会  
SOFTWARE DEVELOPER CONFERENCE CHINA

# Spark Streaming 在苏宁物流天眼

## 全流程监控系统中的应用

嘉宾：俞恺

- 什么是苏宁物流天眼
- 为什么选择Spark Streaming
- 天眼应用案例介绍
- 性能优化和经验总结



顾客打电话来，我的包裹到哪里？什么时候能送到？为什么我的包裹昨天下午就到快递点了，现在还不送？

有辆运输车坏了，影响了哪些包裹，这些包裹还能按时送达吗？

哪些仓库周转不过来了？哪些包裹堆在地上已经半天没动了？

哪些岗位工人忙不过来了，哪些岗位工人没任务做了？

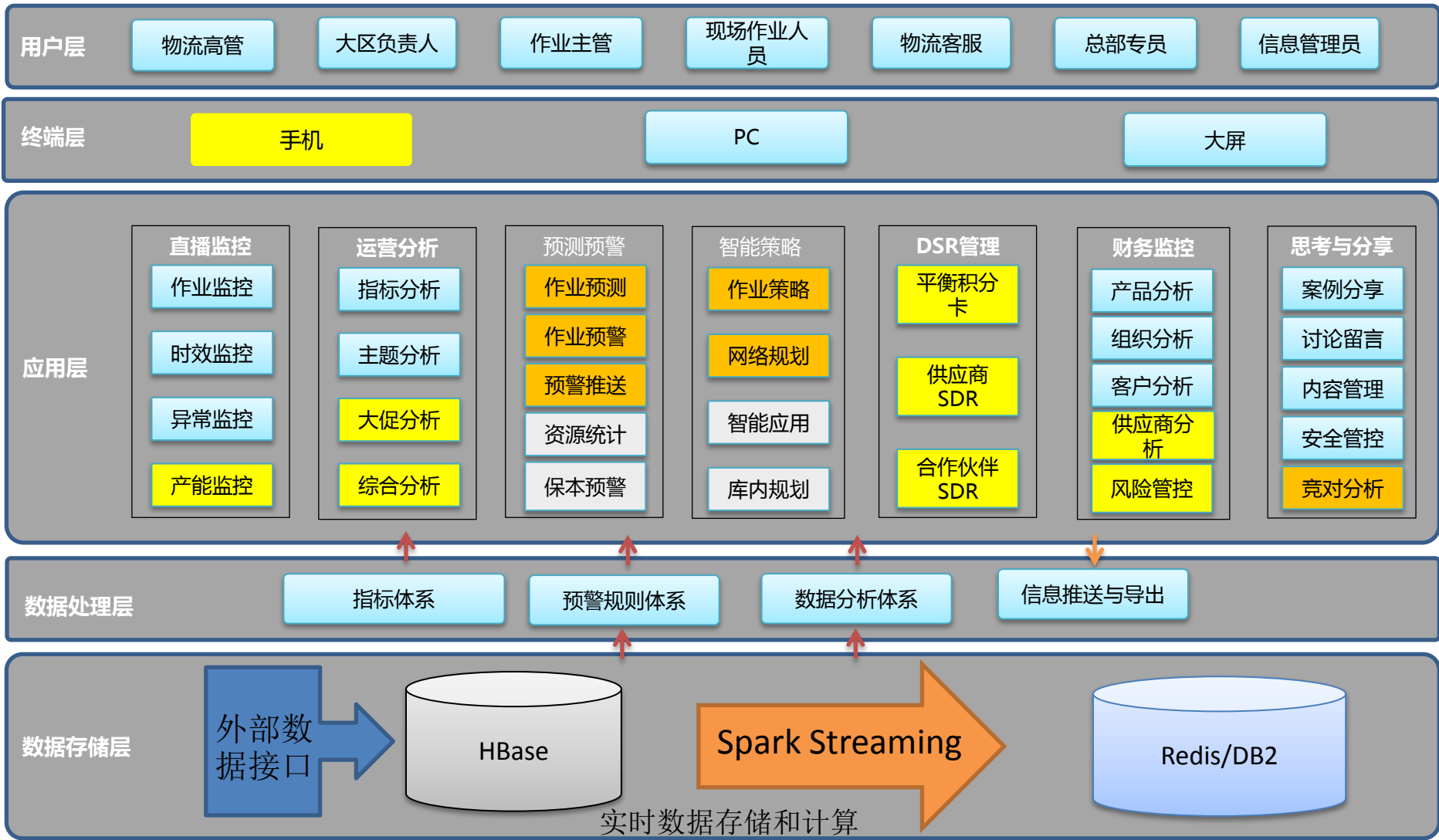
订单全流  
程跟踪

作业异常  
捕获和报  
警

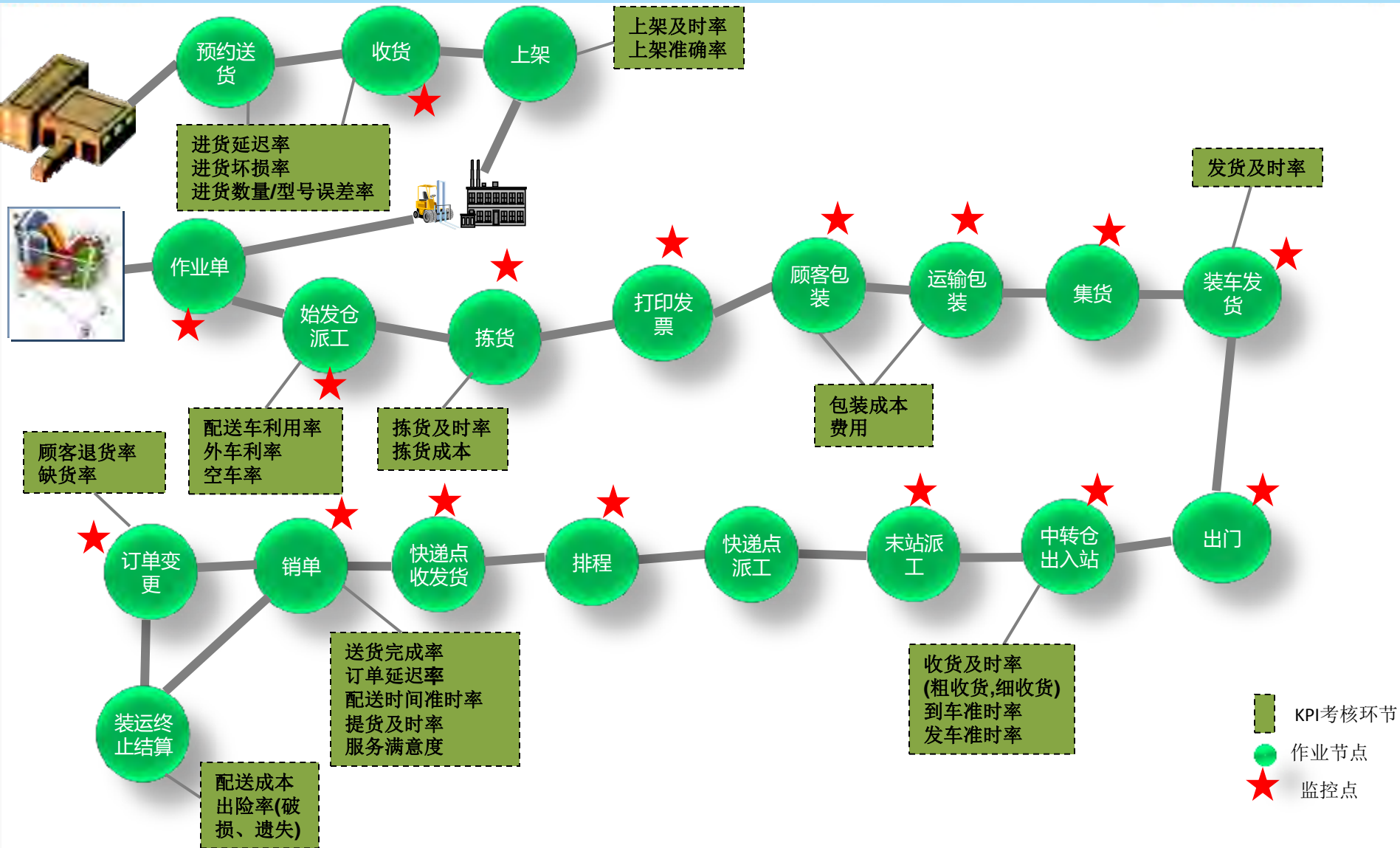
物流各环  
节作业监  
控和预警

资源计划  
编排和动  
态调整

# 天眼应用架构图



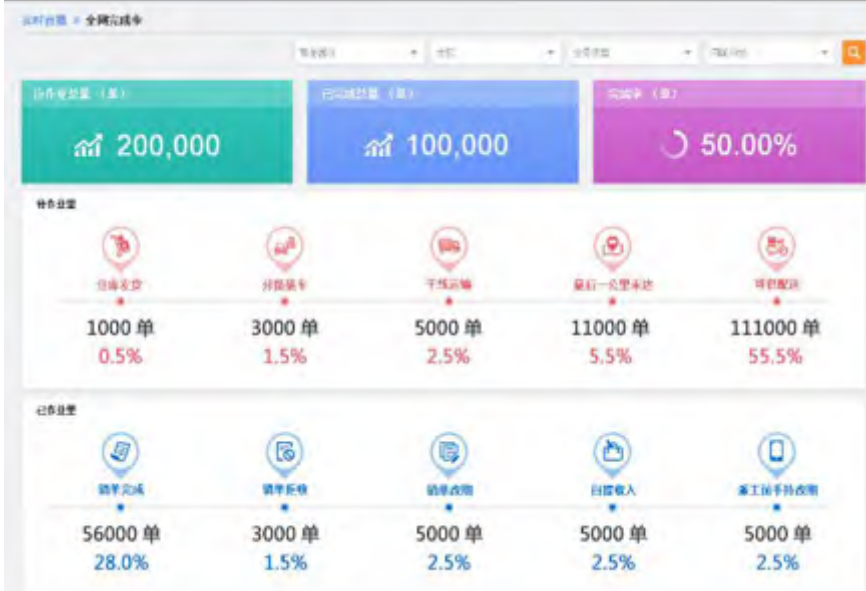
# 物流全流程监控图





出发线路

线路	车次	计划发车时间	装车队	收货数	装车开始时间	装车结束时间	作业状态
南京-合肥	第 A12049	08:30	1200	1	08:35	08:35	装车中
南京-合肥	第 A12048	09:30	1200	2	09:35	09:35	装车中





全国近万个物流网点，几万个作业工位全覆盖监控。

每天可以接收订单状态10亿条。

每秒可处理订单数10万条。

核心监控报表秒级数据延迟。

全网妥  
投率上  
升2.3%

作业异  
常比例  
下降  
5.4%。

客诉率  
下降  
0.1%

- 什么是苏宁物流天眼
- 为什么选择Spark Streaming
- 天眼应用案例介绍
- 性能优化和经验总结

# 天眼技术难点



高可用的分布式计算框架

高扩展性的分布式数据存储

大吞吐量的分布式队列

高并发写入的数据库

# Spark Streaming 技术特点简介 - 1

- 它的核心执行引擎是spark
- Spark Streaming是一个批处理的流式计算框架
- 适合实时数据与历史数据混合处理的场景

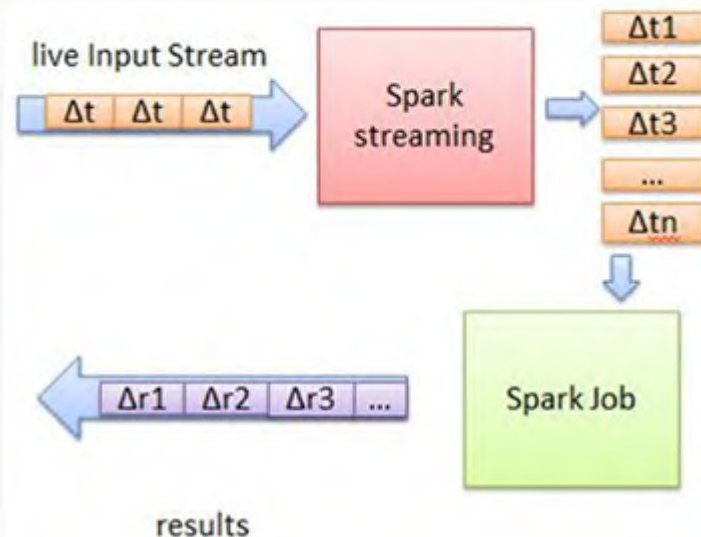


# Spark Streaming 技术特点简介 - 2

Spark Streaming基本原理是将流数据分成小的时间片段（几秒），以类似批处理方式来处理这部分小数据。

处理流程：

- Spark Streaming把实时输入数据流以时间片 $\Delta t$ （如1秒）为单位切分成块
- Spark Streaming会把每块数据作为一个RDD，并使用RDD操作处理每一小块数据
- 每个块都会生成一个Spark Job处理
- 最终结果也返回多块



## Spark Streaming program

```
var t1= new TLogInputDStream(...)
var t2= new TLogInputDStream(...)
```

```
t = t1.join(t2).map(...)
```

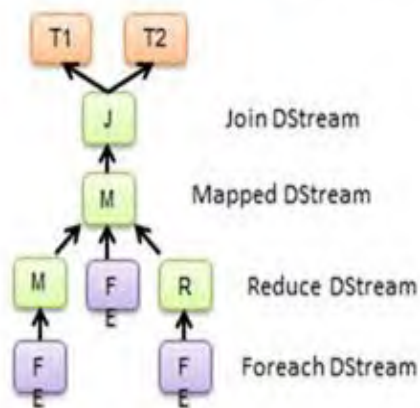
```
t.map(...).foreach(...)
```

```
t.foreach(...)
```

```
t.reduce(...).foreach()
```



## DStream Graph



- 使用Spark Streaming编写的程序与编写Spark程序非常相似
- 在Spark程序中，主要通过操作RDD（Resilient Distributed Datasets弹性分布式数据集）提供的接口，如map、reduce、filter等，实现数据的批处理。
- 而在Spark Streaming中，则通过操作DStream（表示数据流的RDD序列）提供的接口，这些接口和RDD提供的接口类似。

# Spark Streaming 与其它计算框架对比

	Spark Streaming	Storm	分布式数据库
数据延迟	秒级延时	毫秒级延时	分钟级延时
处理方式	小批次处理	单条处理	批次处理
开发便利	Sacla、java、sql等	Clojure,Java等	Sql
扩展性	扩展方便	扩展方便	相对复杂



# Spark Streaming 优势总结

## 实时性

- 物流作业监控对实时性的要求在秒级数据延迟，适合小批次计算，既能满足实时性，又可以批次运算提高效率。

## 可用性

- 分布式计算架构，容错性强，单台物理机宕机不会导致数据丢失或者系统不可用。

## 扩展性

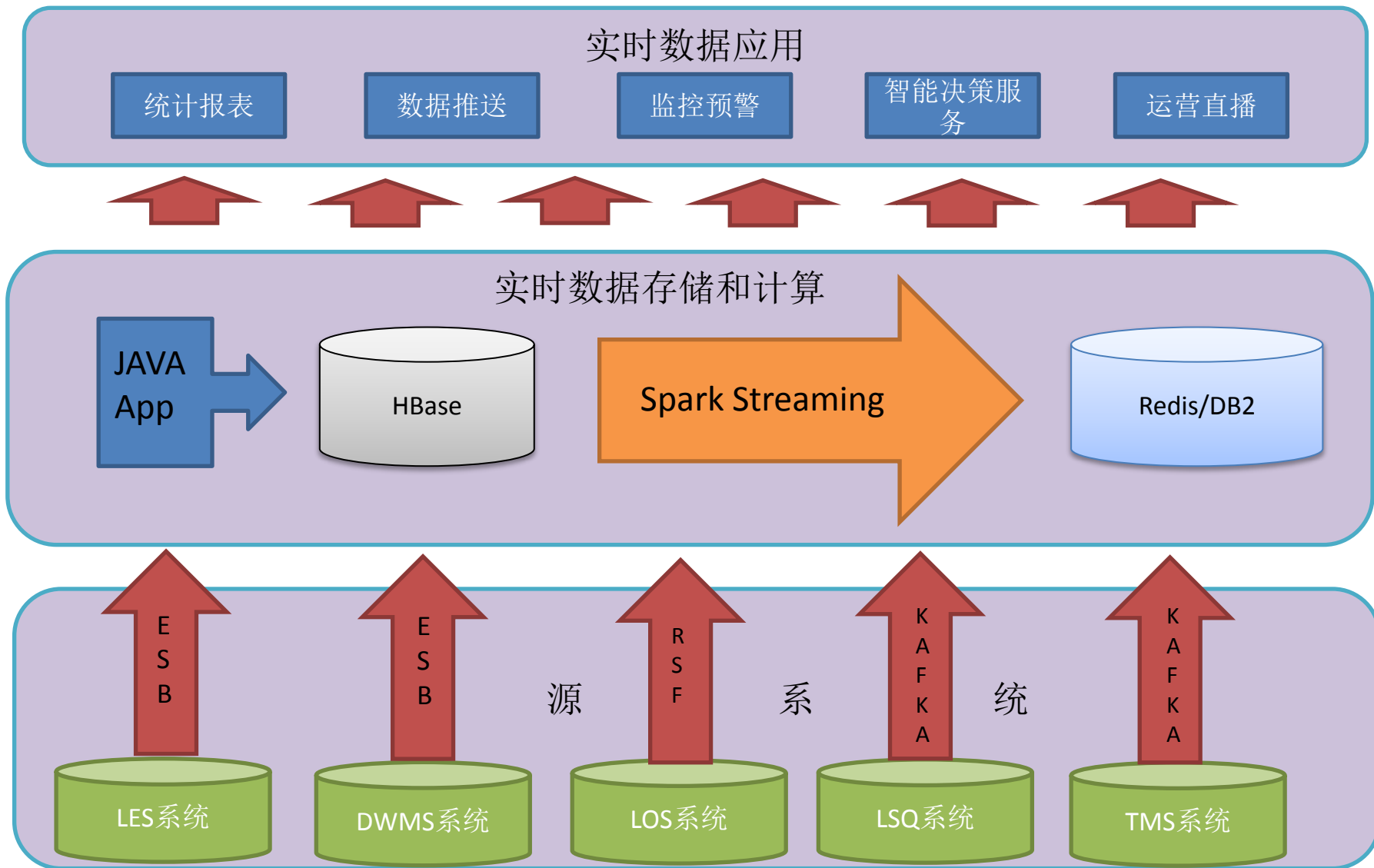
- 可以通过集群添加节点的方式增加计算和存储性能，扩展方便快捷。

## 开发便捷性

- 之前很多的应用逻辑是用sql写的，使用spark可以很方便的把原来sql的逻辑移植过来。



# 天眼的整体系统架构



- 什么是苏宁物流天眼
- 为什么选择Spark Streaming
- 天眼应用案例介绍
- 性能优化和经验总结

# 天眼流式计算的设计原则



# 天眼模块划分

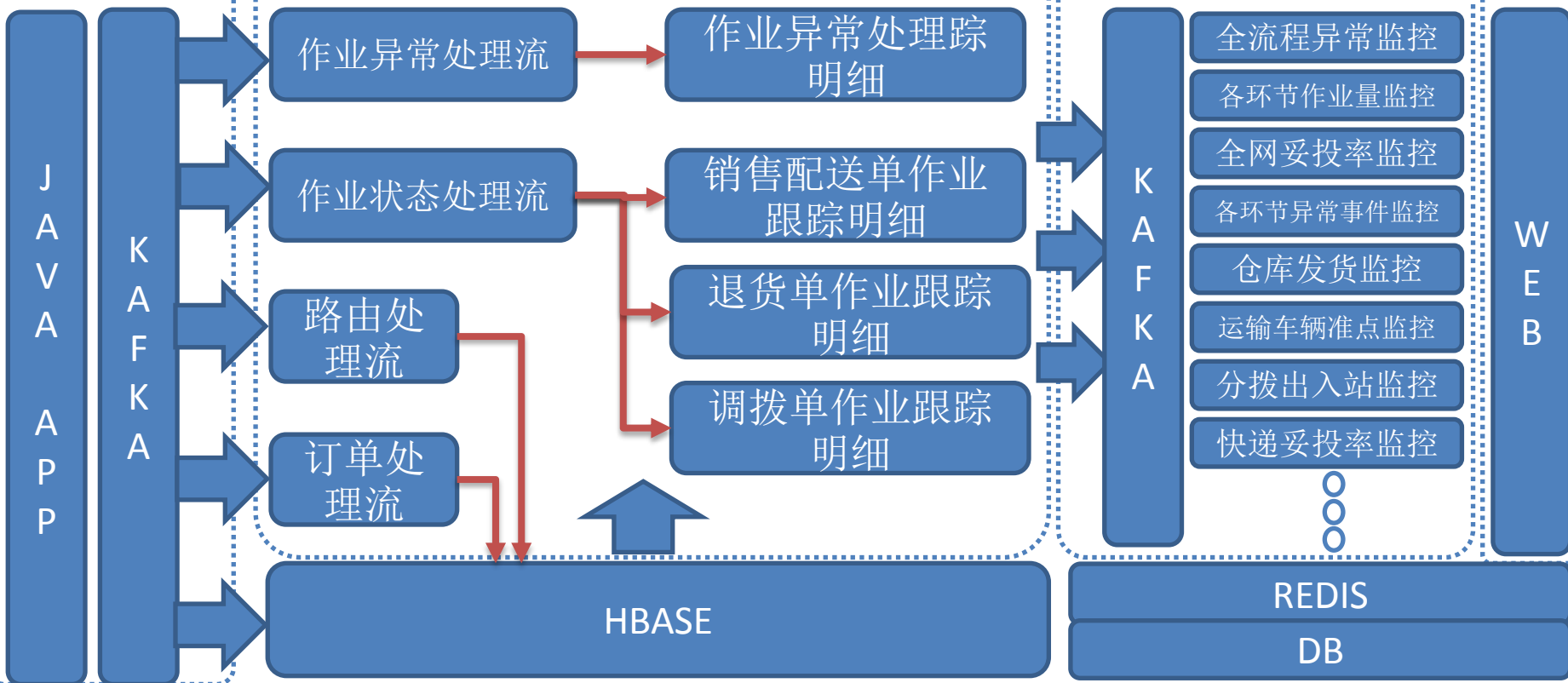
## 数据整合类 & 指标统计类

数据接入

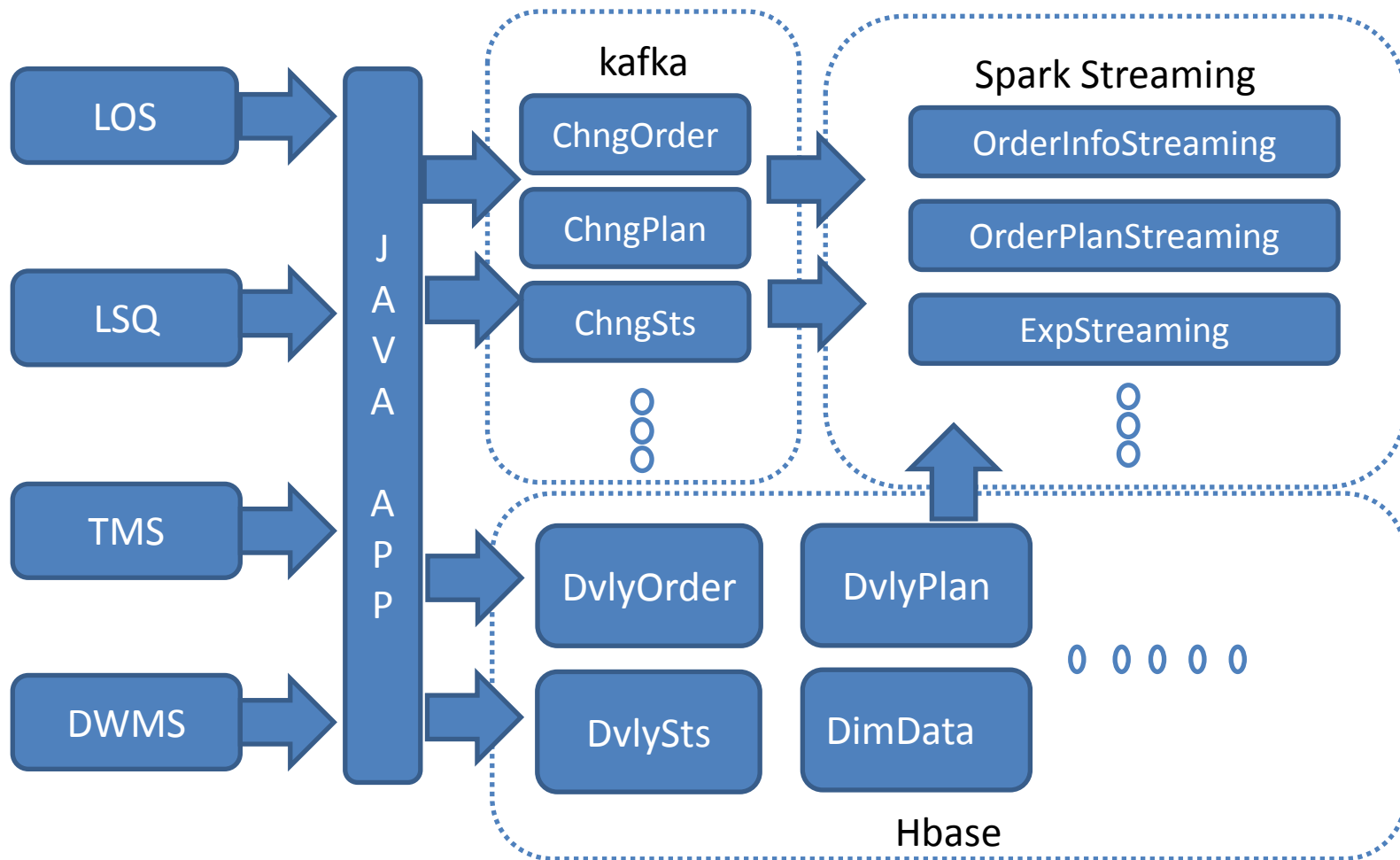
Spark Streaming 数据清理和整合

Spark Streaming 指标统计

展示



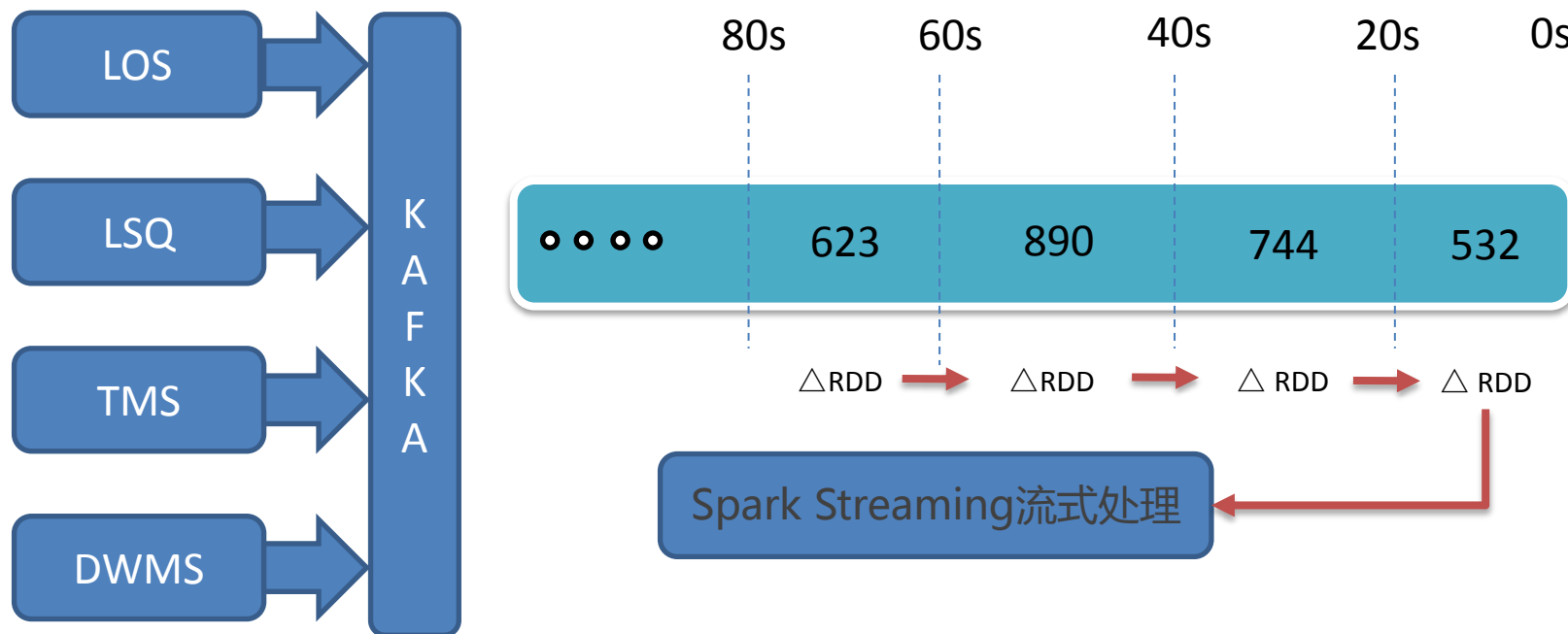
# 数据采集和存储



所有的数据接收下来先进HBASE。  
变化的数据抛送kafka，进入下面的流式计算。

## Spark Streaming整合kafka

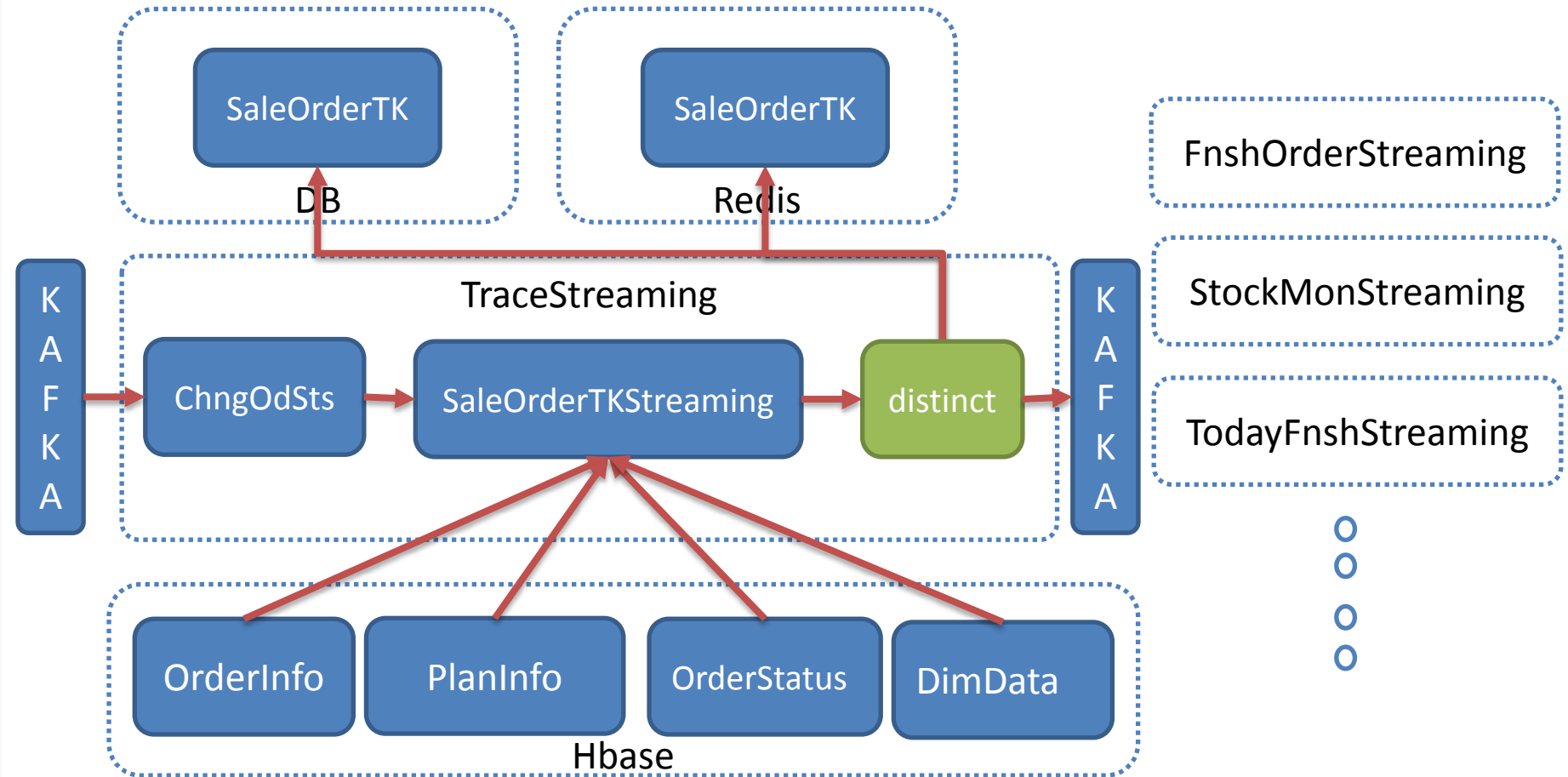
```
val km = new KafkaManager(kafkaParams)
val messages: InputDStream[(String, String)] = km.createDirectStream[String, String,
StringDecoder, StringDecoder](ssc, kafkaParams, topicsSet)
```



Spark Streaming会记录每个RDD对应的offset，但是如果流处理异常停止则需要记录当前的offset，以便下次流启动时从当前位置继续处理。  
通过设置spark.Streaming.kafka.maxRatePerPartition参数，避免batch过大。



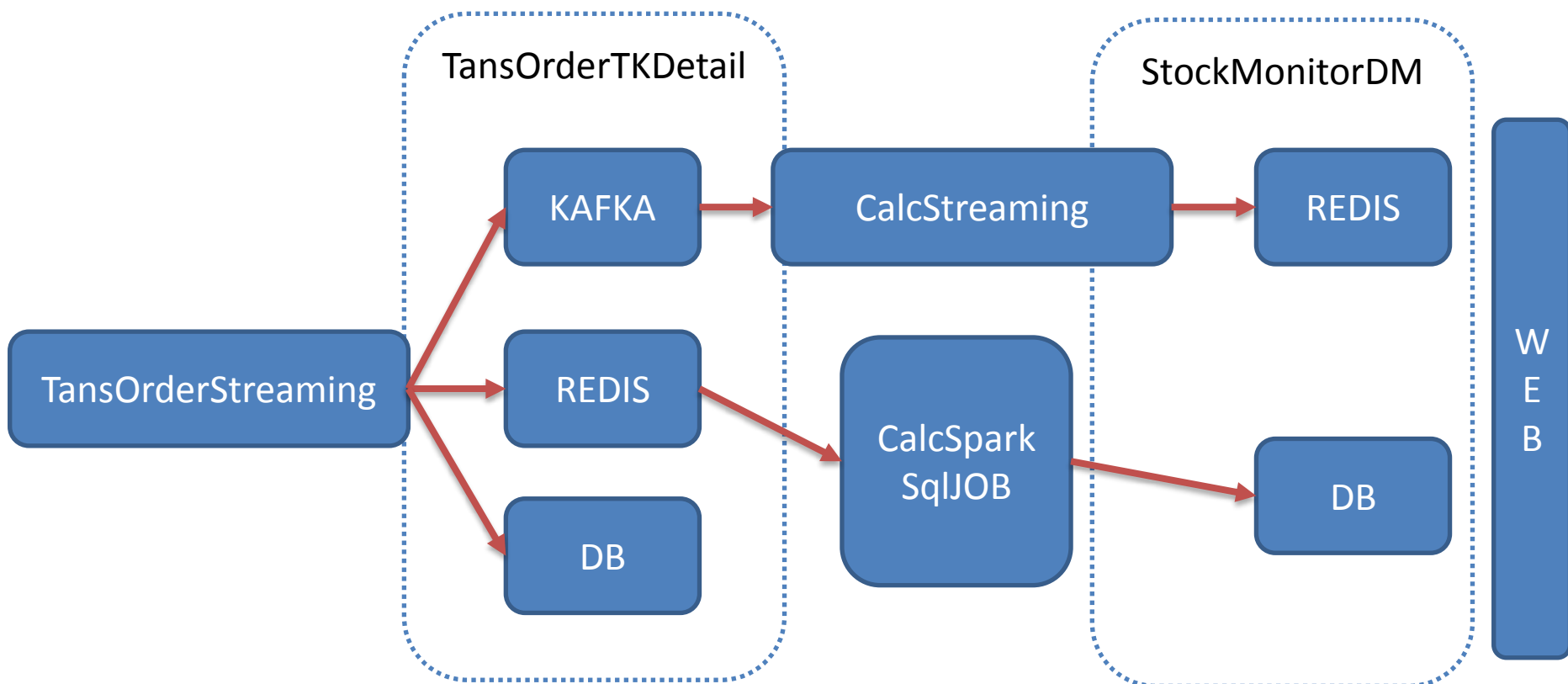
# 正向配送跟踪流计算



数据计算完成后需要做distinct。

Distinct不能完全使用RDD的distinct方法，要考虑自己的业务。

## 仓库作业监控任务流



流式计算需设计初始化功能：

- 1.每天做个全量初始化，然后再做增量累加。
- 2.当流式计算出现异常时，可以通过初始化恢复数据。

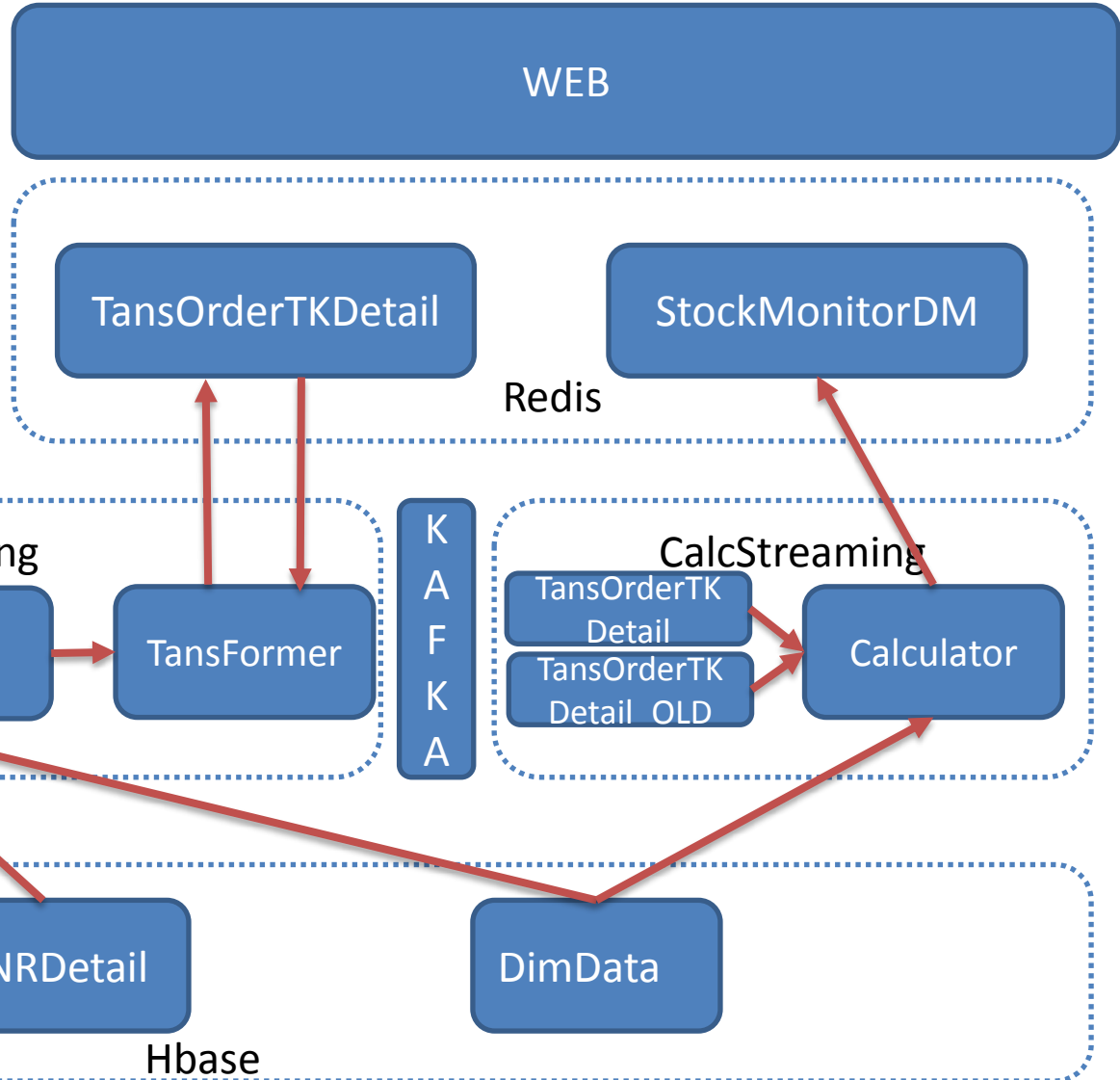
# 仓库作业监控数据流

TansOrderTKDetail\_old

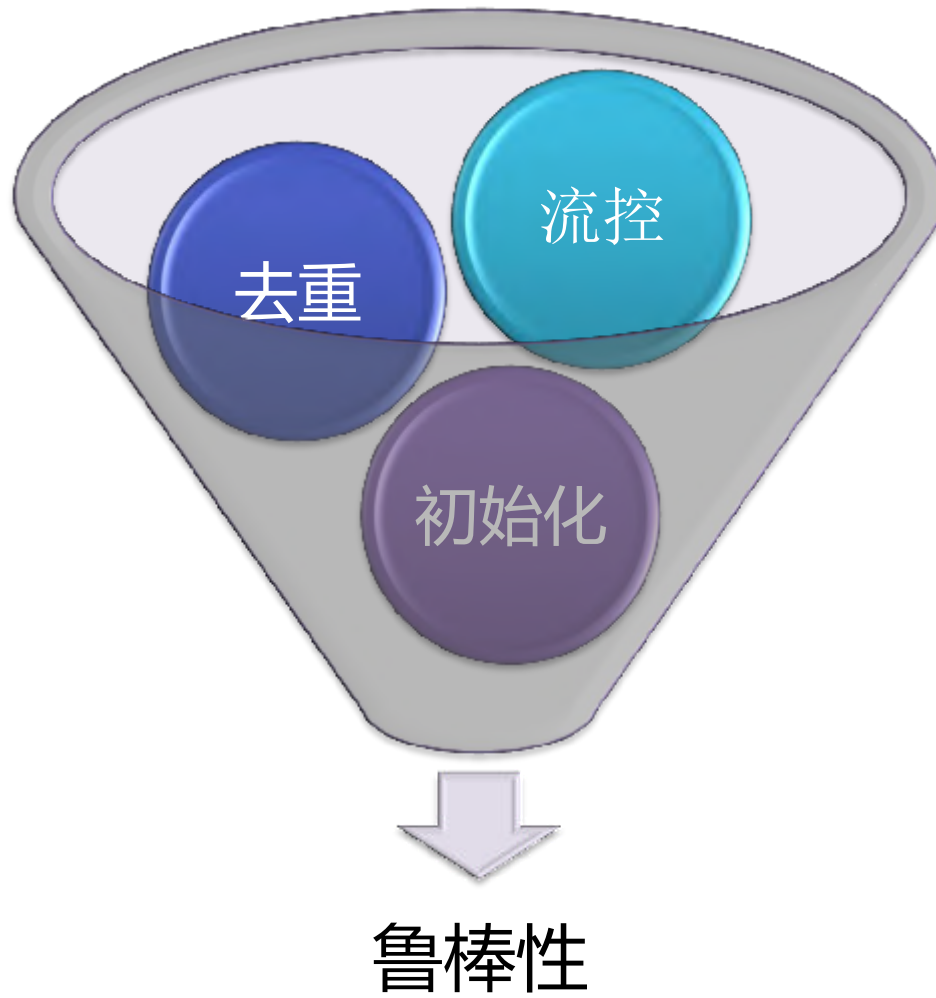
Logid	isTask	IsPick	isPack
00001	1	0	0
00002	1	1	0

TansOrderTKDetail

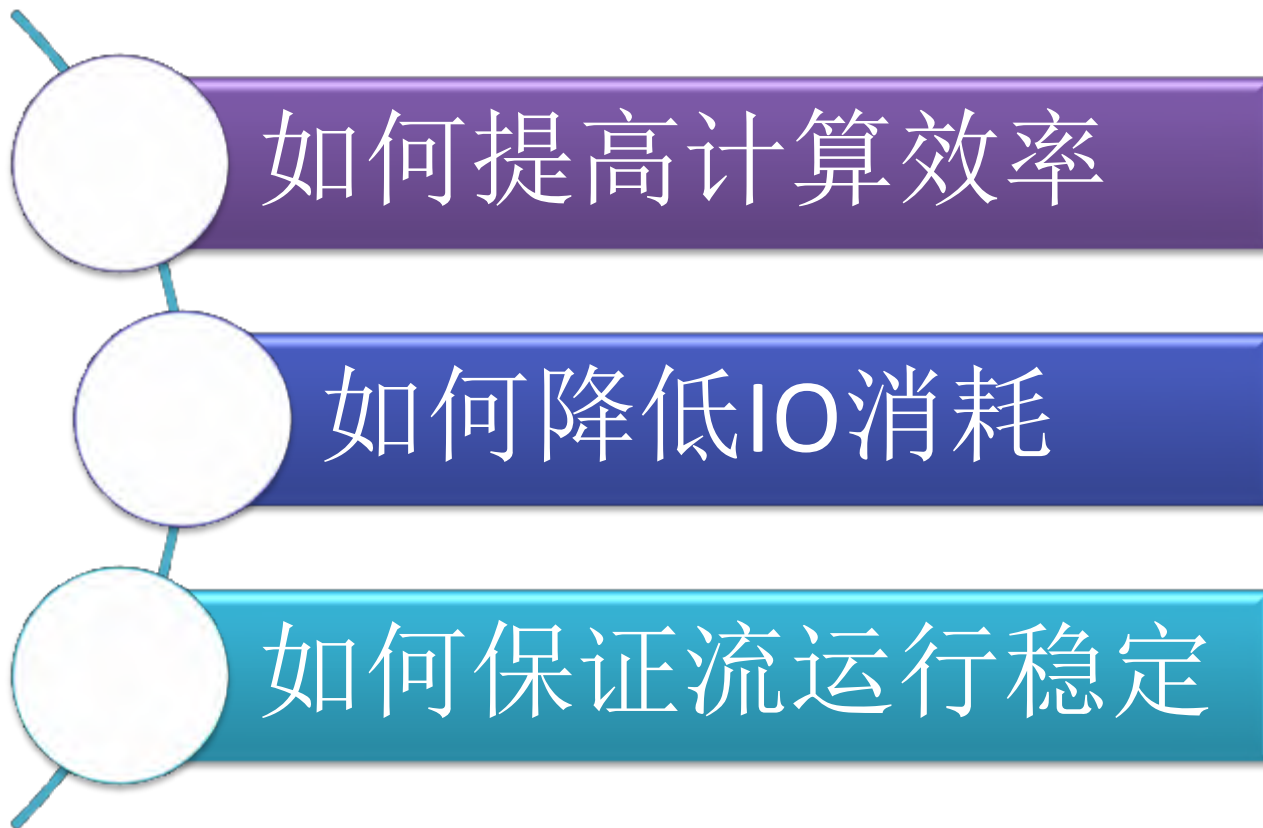
Logid	isTask	IsPick	isPack
00001	1	1	0
00002	1	1	1



# 流式计算系统设计的注意点



- 什么是苏宁物流天眼
- 为什么选择Spark Streaming
- 天眼应用案例介绍
- 性能优化和经验总结



# Spark Streaming 动态调整分片大小

## 分布式运行是spark的核心

- 提高计算性能最简单的方法就是加大并发。

## 分片与数据的匹配

- 分片不是越多越好，并发数要与数据量匹配
- 根据不同任务类型选择CPU和内存的比例。计算复杂加大CPU，数据量大增加内存。

## 动态设置分片大小

- `spark.default.parallelism`参数用于设置分片个数。
- 计算每个RDD的数据量，并根据数据量调整分片个数。

# Spark Streaming 避免数据倾斜

## 为什么会出现数据倾斜

- 数据倾斜主要出现在shuffle的阶段，数据在各个分片分步不均匀。

## 如何定位数据倾斜

- 从执行监控中找到执行时间长的stage，出现每个worker执行的时间差异很大，基本就是发生了数据倾斜，定位代码，join、reduce、groupby等算子容易出现数据倾斜。

## 解决数据倾斜的常用方法

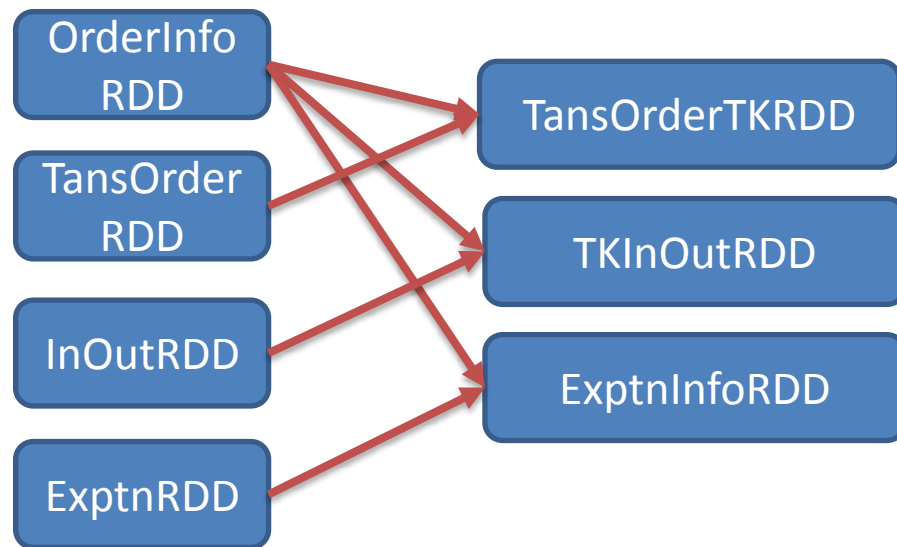
- 过滤空值
- 使用Map join，把小表广播到每个节点
- 业务上优化、数据拆分



# Spark Streaming 性能优化 – 持久化

```
val odInfoDf = sqlContext.read.parquet(TablesInfo.SORPath +  
"t_od_info_hbase") //cache并且注册临时表  
odInfoDf.cache()  
odInfoDf.registerTempTable("T_OD_INFO")
```

- 迭代计算，一个RDD被多个RDD依赖
- 缓存可以让RDD的动作效率提高10倍
- Persist()和cache()方法，StorageLevel参数
- 缓存具有容错性



## Spark Streaming 性能优化 – 持久化

- Spark Streaming 使用缓存的注意事项：
  - 当缓存过期，调用RDD.unpersist()方法，及时清除缓存，否则会出现内存不足。
  - 维度表缓存。需要设置定时更新的机制，或者设置定期失效后重新加载。
  - sqlContext.sparkContext.broadcast()方法，把维度表广播到每个节点，提高计算效率

# Spark Streaming 性能优化 – HBase

## 表设计优化

- Hbase表设计和使用时，尽量避免scan()，必要时采用中间表，将scan转化为get()操作。

## 优化批次提交

- 操作HBASE表时，尽量使用批次操作，不要单条操作。合理设置批次提交的数量，批次过小会导致频繁与HBASE交互，批次过大会导致HBASE的GC时间过长，影响性能。

## 数据均匀分布

- HBASE表使用前，需要做预分片，避免热点分区，分片Key要具备随机分布特点。

# Spark Streaming 流的监控与稳定性

## 流元数据管理

- 记录每个流的基本信息，储存在数据库中

## 正常运行监控和报警

- 在流提交时将app\_id记录在数据库，建立一个守护进程，通过yarn的接口，定期查询运行健康状况，发现异常短信报警。

## 异常失败自动重启

- 系统监控发现Streaming在运行过程中如果出现异常中断，根据流元数据信息自动重运行任务

## 运行堆积监控和报警

- 调用spark metrics查询流运行信息，监控waitingBatch，发现对接短信报警。
- 监控每个批次的运行时长，超时杀掉重启。

## 分享总结

Spark Streaming是分布式小批量流式计算引擎，具备实时性、高可用、高扩展性等特点。

天眼系统整合Spark Streaming，HBASE，Kafka，Redis等技术实现了秒级数据延迟的物流全程监控。

天眼系统中流式计算系统的两个分类和四个设计原则。

Spark Streaming性能优化和稳定性保证经验。



**SDCC 2016**  
**中国软件开发大会**  
SOFTWARE DEVELOPER CONFERENCE CHINA

**谢谢！**

苏宁云商IT总部技术总监 俞恺  
[121503021@qq.com](mailto:121503021@qq.com)