2017

Storm 在卷皮实施经营概览中的应用

聂 聪

共享内容 简介

实时统计分析系统如何对电商平台实时发生经营情况进行统计和分析，是目前电商平台的一大热点。卷皮实时经营概览就是这样一个实时统计分析系统，它主要通过storm技术实时统计了发生的经营情况并展现。

本分享主要介绍卷皮实时经营概览系统架构多维度组合条件统计的实现方法、以及在实现过程中遇到的困难和应对方法。

目录

PART 1
业务需求

PART 2
设计

PART 3
系统架构

PAR 4
Storm拓扑

PART 5
配置参数

PART 6
结果的存储

PART 7
获取并汇总数据

PART 8
监控

PAR 9
遇到的问题和应对

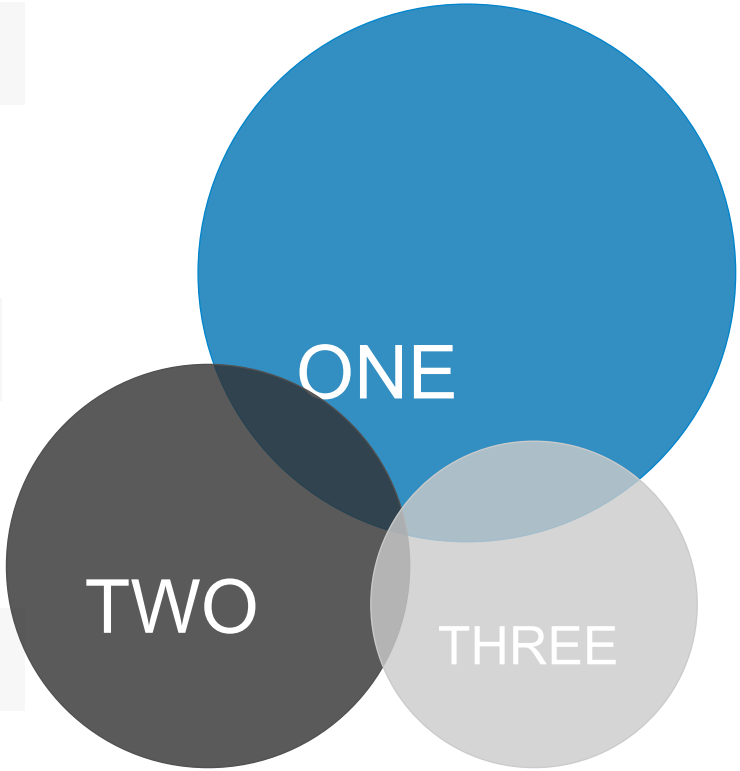
PART 10
总结

PART 11
缺点

业务需求

PART 1

1. 统计经营情况
2. 可支持多维度钻取
3. 实时性要求



设计
PART 2

纬 度

日期、
小时、
终端、
站点、
订单类型、
商品类型、
商品来源、

指 标

下单量、
销售总额、
支付订单量、
销售量、
支付人数、
新客销售总额、
新客支付订单量、
新客销售量、
新客数、
取消订单量、
用户取消订单量

设计
PART 2

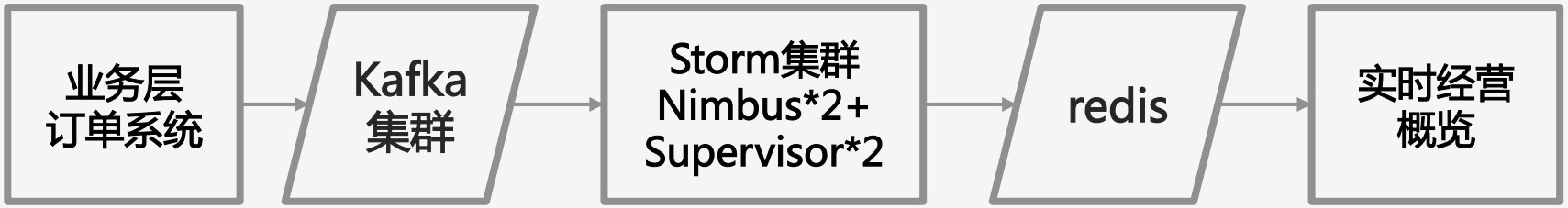
上帝视角
查询条件：日期

订单明细视角
查询条件：日期维度 小时维度 终端维度 站点维度 订单类型维度 商品类型维度 商品来源维度

订单视角
查询条件：日期维度 小时维度 终端维度 站点维度 订单类型维度，订单明细视角

系统架构

PART 3

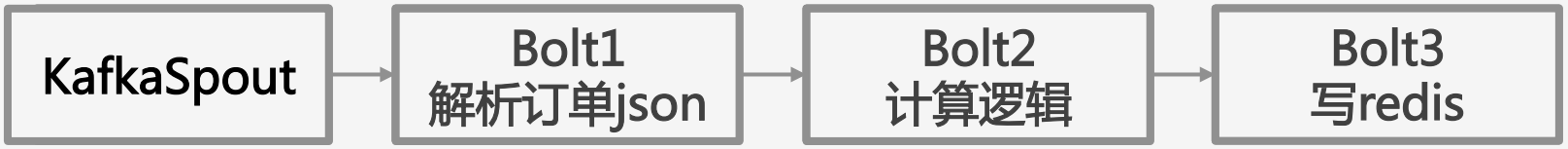


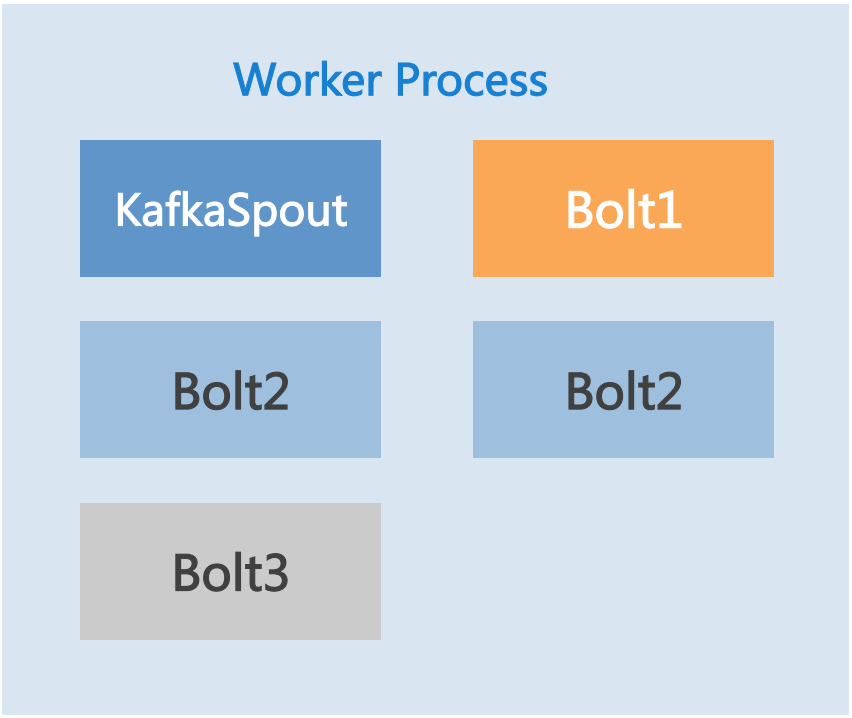
架构目标：轻量

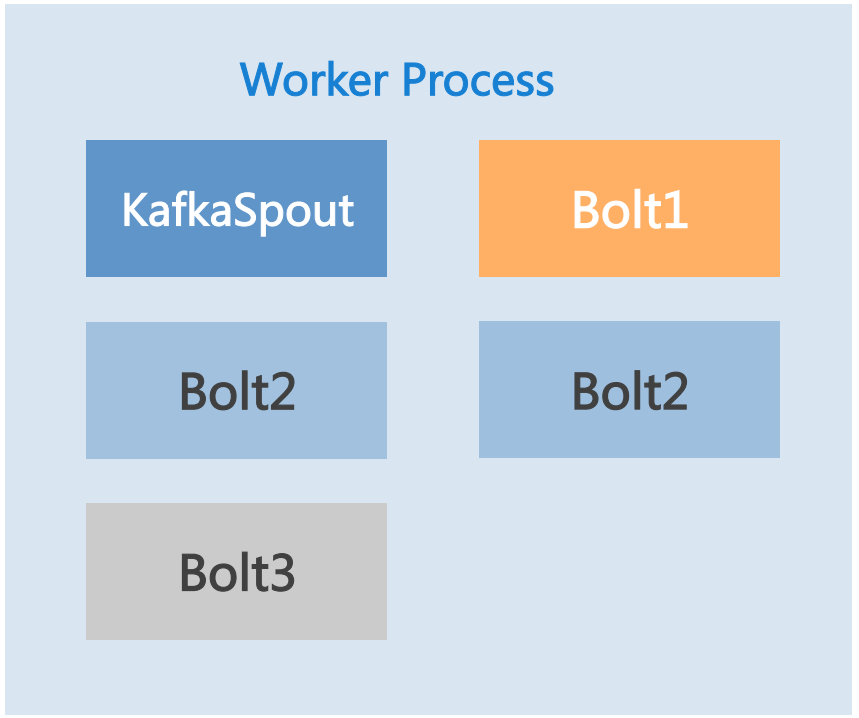
- 采用storm做为实时计算的技术框架
- 采用保证从Spout发出的每个数据都被完全处理。没有使用trident
- 数据持久化问题

Storm 拓扑

PART 4







配置参数

PART 5

topology.config.NumAckers=10

topology.config.MaxSpoutPending=200

topology.config.NumWorkers=2

topology.spout.OrderPaid.KafkaSpout.thread=2

topology.bolt.OrderPaid.ParseOrderPaidBolt.thread=2

topology.bolt.OrderPaid.AnalyseOrderBolt.thread=4

topology.bolt.OrderPaid.SaveMessage2Redis.thread=2

结果的存储

PART 6

key命名。每一个间隔表示一个纬度

jingyinggailan:view=3:date=2016-10-10:hour=10:terminal_id=1:site_id=2:soi_type=1:goods_type_id=1:si_source_type=1

redis中的值是一个hash表。hash表的每一个field对应一个指标

- 值得注意的是：在redis中累加的时候需要用inrc命令

获取并汇总数据

PART 7

如何从redis中取得值并汇总

- 首先确定当前值应该从什么视角来取

- 在redis中 用keys命令 通配符来取 注意key中维度的次序不可以颠倒。

举例：

keys jingyinggailan:view=3*:date=2016-10-10*:hour=10*:site_id=1

监控

PART 8

通过Storm提供的thrift接口来获取Toplogy的监控数据

遇到的问题和应对

PART 9

1 怎么恢复数据

A 什么时候写offset

在每日零点备份当时的offset，记录位置zookeeper。

B 什么情况下需要恢复数据

1、spout数据源异常

2、storm作业异常

C 数据恢复步骤

1、停止作业

2、清除对应日期的redis的key

3、恢复offset

4、重启storm作业

遇到的问题和应对

PART 9

2 return处理

A **什么时候return**

如果在执行过程中遇到的问题是通过尝试无法正确执行的，需要return

B **return的结果是什么**

return的数据被丢失

C **对return进行日志记录**

总结

PART 10

本案例的特点

- 一个轻量级的实时olap体系。
- 适合少量纬度的钻取。
- 秒级汇总经营数据

缺点

PART 11

1 代码相对较复杂，扩展不方便

- 我们实现最细纬度的统计需要手动书写很多代码。代码耦合比较高。
- 如果增加纬度和指标那么需要修改很多细节，并测试。
- 如果可以改造成通过配置的刚发就能增加统计的纬度和指标将大大的提升可用性。并对代码质量的管理较方便

缺点

PART 11

2 并没有实现对组合指标的汇总

- 我们现在实现的是对最细纬度的汇总，然后在展示报表的时候根据钻取条件在再做粗粒度的汇总。
- 如果可以实现将各种组合条件下的纬度预先汇总起来，那么在报表展现的时候不需要实现汇总逻辑。当然带来的问题是预先将各种组合条件下的纬度预先汇总起来面临两个问题：数据量进一步加大、如果纬度数量增加那么组合条件的处理将变得很复杂。

缺点

PART 11

3 不支持distinct count计算

- 假设一个用户在10点和11点都访问过平台。那么按照最细粒度的汇总会分别记录当前用户在10点和11点到访1次。如果做简单的汇总那么用户在当日算到访两次。这样的计算是不合理的。我们希望在粗粒度的汇总的时候能对用户id去重。不过这个需求比较难实现。

缺点

PART 11

4 不支持时间窗的统计

- 站在每个时间点（hour）统计近3个小时的经营情况是一个时间窗的销量统计问题。目前实现这个需求没有一个很方便的方法实现。

Thanks

For your watching

Q/A