spark流计算项目 - 基于Kafka、SparkStreaming、Hbase、Highcharts等的多维度销售额流计算项目

1. 项目内容

使用**Kafka->Spark->HBase->前端**的架构，实现**按天、地区（省份）进行销售额的实时计算（5s延迟）**，并进行**可视化**展示。通过实时计算，为不同地区提供销售分析，没达到预期的则进行销售策略调整。

本项目为简易版本的流计算项目，业务功能简单。但是依旧要考虑系统健壮性，下文对架构进行了简单分析和设计。

2. 架构设计

2.1 架构分析

- 后台SparkStreaming重启、维护时不影响前台展示；后台重启后，前台展示的数据准确
- 前台Tomcat重启，前端不更新但依旧有显式，不影响实时计算

即，**前后台应互不影响，独立、解耦，提高架构健壮性。**

2.2 架构设计

总架构：kafka-生产数据->spark-消费数据，实时计算，写库->hbase<-读库-servlet<--jsp、Highcharts可视化

为实现上文架构分析中提到的效果，使用**HBase作为持久化层**，实现前后台独立，互不影响。选用Hbase是因为Hbase稳定性强、实时查询效果好。

各项规划：

- 数据源：Kafka
- 前台作业：HighCharts、Servlet、Jsp读库，可视化
- 后台作业：SparkStreaming实时消费、实时计算、写库

架构图：略

框架

- jdk1.8.0_40
- scala-2.11.12
- hadoop-2.7.5
- spark-2.4.4-bin-hadoop2.7

- kafka_2.11-2.2.0
- hbase-1.3.6
- zookeeper-3.4.9
- highcharts4.x

其它

关于hbase的知识以及本项目中关于hbase的程序，可以查看本人的一篇博客文章HBase从介绍到Java客户端开发

在jar依赖上遇到了没有解决的问题，加之pom.xml文件一修改，就要fail在downloading maven-metadata-...上。所以最后决定spark和kafka依赖pom.xml；hadoop和hbase则是自己添加依赖到lib目录下，依赖来源是$HBASE_HOME/lib/，除了明显无关的（带test、yarn、mapreduce字样），其它的一起拖放到了lib下