各位评委老师大家好，我是花满溪作品的成员梁焯炫，为了满足在大量用户使用的情况下，提供精准的推荐服务，本古诗学习系统的推荐模块，采用大数据的架构进行搭建。

现在展示的这个图，就是推荐模块使用的基于大数据的架构图，在这里，我们使用了Flume 、Hadoop、Spark、Redis等组件，使用了Word2Vec、SVD、Xgboost+LR这些算法，我们的推荐模块主要是为了实现三个功能，相似古诗推荐，猜你喜欢推荐，。。和每日推荐，而我们所使用的大数据技术和算法都是为了实现这些功能所设计，整个架构的设计是围绕“如何更好地存储和利用用户行为数据”而设计的

整个架构的设计可以分为离线层和在线层。这里我们可以看到，web端产生的用户行为数据会被Flume所收集，分别存入HDFS和进入Spark-Streaming，与Flume连接的这两个分支就对应了本系统架构中的离线层和在线层

--------------------------------------------------

(ˉ▽￣～) 切~~

--------------------------------------------------

本系统中Flume的配置如图，在本系统中Flume使用的Source为Http Source，使用了2个Sink分别为HDFS Sink和Avro Sink，分别用于对接HDFS和Spark-Streaming，HDFS Sink会把数据存入HDFS，让数据进入离线层，Avro Sink会把数据传给Spark-Streaming，让数据进入在线层

-------------------------------------------------------

(ˉ▽￣～) 切~~

接下来说一下，相似古诗功能的实现，上面这个图所展示的，就是这个功能实现的整体流程，这里我为评委们介绍一下这个功能详细的实现流程。

相似古诗推荐的列表，是根据某一首古诗，计算与当前古诗文本相似度最高的10首古诗所得到的，由于计算量大，我们会先根据古诗的诗人、朝代、标签先筛选出一部分古诗，使用Spark将所有古诗的标题和正文作为语料库，对古诗进行分词，使用分词后的数据训练Word2Vec模型，用Word2Vec模型预测出每个词对应的词向量，并计算每个词的TFIDF值，对每首古诗所包含的词向量进行加权求平均得到每首古诗对应的古诗向量，并使用余弦相似度计算与当前古诗文本相似度最高的10首古诗，进行相似古诗的推荐。

--------------------------------------------------

(ˉ▽￣～) 切~~

接下来说一下，每日推荐功能的实现。通过Flume进入离线层的数据会被存储在HDFS中，系统后台会在每天零点调用已经封装好的Spark程序读取HDFS里面格式化好的用户行为数据，对数据进行清洗和处理，使用处理好的数据训练召回模型SVD和排序模型Xgboost+LR模型，使用SVD模型对数据进行粗排序，得到每个用户对应的古诗推荐候选集，使用Xgboot+LR模型对每个用户对应的古诗推荐候选集进行精排序，得到每个用户每日推荐的古诗列表

--------------------------------------------------

(ˉ▽￣～) 切~~

接下来说一下，猜你喜欢功能的实现，用户在使用本系统时发生的行为会以Json的格式，通过Flume传输给Spark-Streaming，Spark-Streaming会提取这个Json数据里面的内容，并存入Redis，

当用户请求猜你喜欢的服务时，后台会实时的从Redis里面召回对应的古诗标签，并根据古诗标签从Mongodb里召回待排序的候选古诗数据，从Mysql数据库中查询数据，对这些古诗信息进行特征拼接等操作，得到符合Xgboost+LR模型的输入格式，使用Xgboost+LR模型对其进行排序，并返回猜你喜欢的古诗推荐列表。