中国矿业大学计算机学院
2020 级本科生课程实验报告

课程名称：	大数据存储与管理课程设计
报告时间：	2022 年 11 月
学生姓名：	杨再润
学号：	06203203
专业班级：	大数据 2020-1 班

3. 大数据存储案例设计

3.1 实验目标

使用深圳市公开的高速公路 ETC 数据集进行大数据存储的设计，并可视化展示。

3.2 实验要求

- 生成流式数据
- 设计 HBase 或 Mycat 存储数据
- 对数据可视化展示并实现定时刷新
- 对高速数据进行缓存
- 实现交互式查询
- 设计预警算法

3.3 系统架构设计

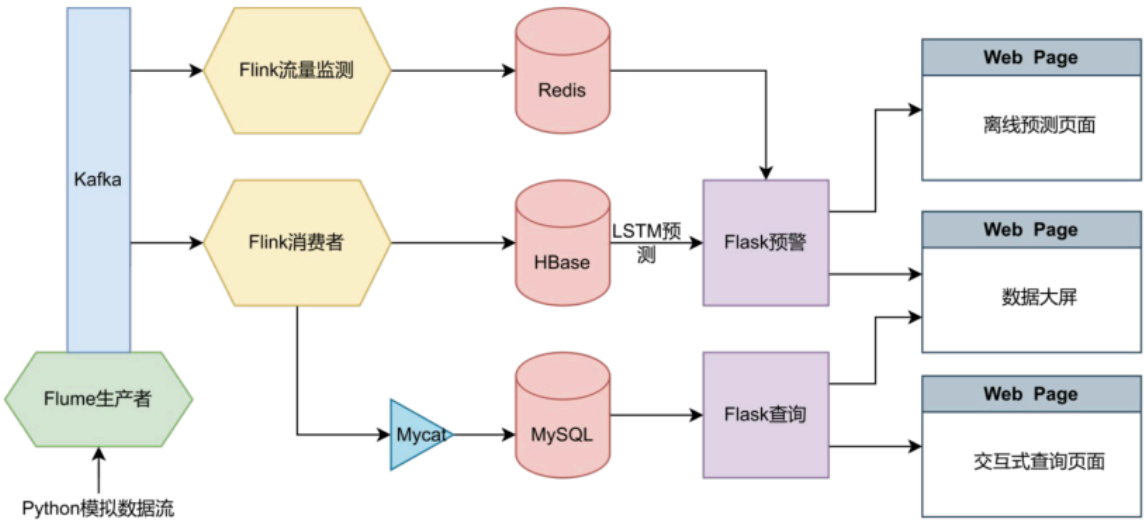


图 3.3.1 系统架构图

整个系统的架构如上图，包含数据流生产、消费、存储、后端与前端 5 个层

面。其中数据存储用到了 MySQL、HBase 和 Redis。数据由脚本文件模拟生成，通过 Kafka 使用 Flink 分发到数据库中，网页通过 API 请求 Flask 查询需要的数据并展示。

系统主要有四个功能，分别是数据大屏、交互式查询、实时车流量报警与离线车流量预测。数据大屏通过多种图表展示数据，交互式查询实现了自定义条件对数据进行展示和导出，流量报警是通过 Flink 进行的实时报警，流量预测则使用神经网络来预测未来的车流量。

3.4 个人工作

个人工作如下图所示，总的来说是实现在线车流量报警和离线车流量预测这两个功能，包含设计 Flink 程序统计实时车流量、设计 Redis 缓存、设计 HBase 存储、LSTM 神经网络预测车流量、后端预警相关的 API 接口以及一个离线预测车流量的网页。下面将对这些工作进行详细介绍。

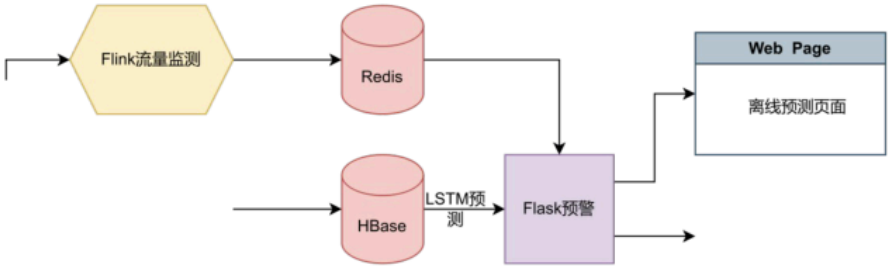


图 3.3.2 个人工作架构图

（1） Flink 统计实时车流量并报警

首先在华为云上以 Standalone 模式搭建 Flink 集群，随后设计 Flink 程序，实现从 kafka 读取消息，用 Flink 统计车流量，最后将结果存入 redis，下面是详细说明。

从 Kafka Source 实时数据，只需要通过调用 flink-connector-kafka 并做好配置即可。

为了让 Flink 能够处理历史流数据（重播的历史数据，最开始打算这么做，但后来改成了使用脚本实时生成数据），通过重写 Watermark 定义事件时间，使得 Flink 的计算时间是数据传入的历史时间（2020 年）而不是当下，这样的好处是，一方面 Flink 的处理逻辑就会贴近实际情况，另一方面则能避免数据传输延迟对计算结果的影响。

在 Flink 中设定时长为 1 秒的滑动窗口，统计窗口内各出站口的车流量通过 aggerate 方法实现，这是 Flink 的有状态的计算方法，这样每次计算都只关注窗口内的数据，可以减少开销。

将计算结果经过滤后 Sink 到 Redis 中，利用 Redis 的 Hash Table 对数据高效缓存，等待 API 请求使用。

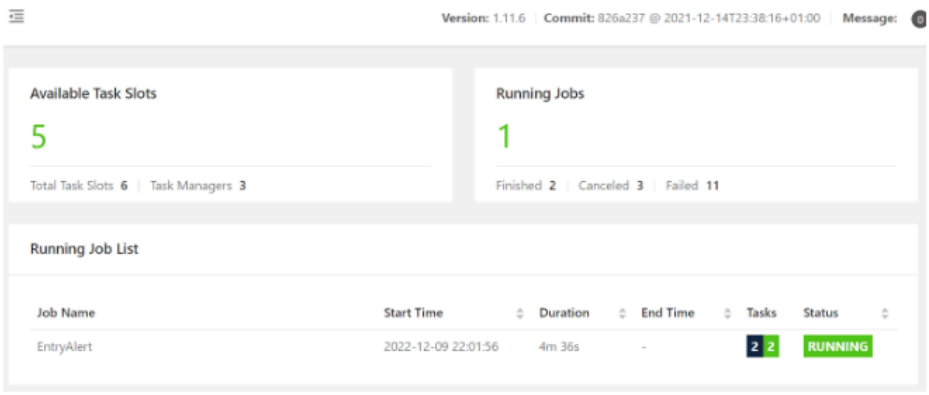


图 3.3.3 Flink 正常运行

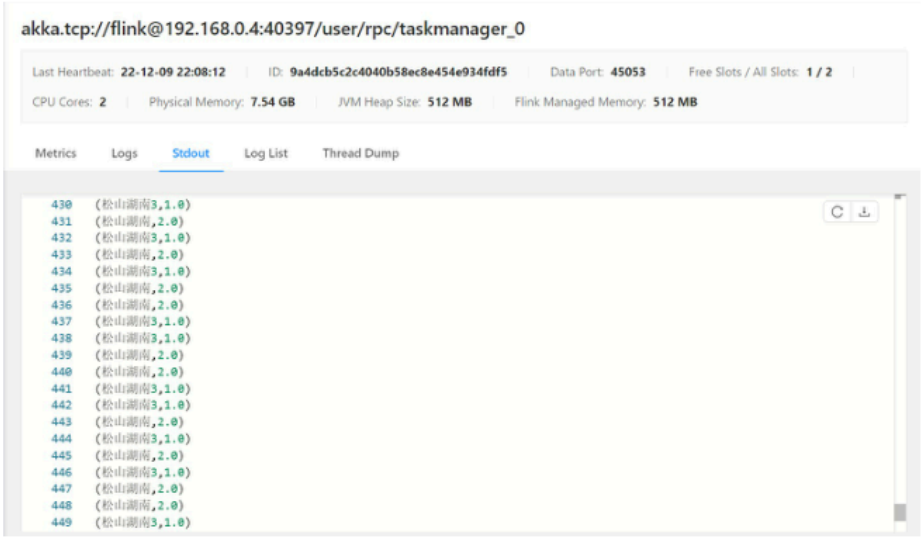


图 3.3.4 报警信息正确显示

如上图，kafka 控制台显示实时计算的结果数据，相同的数据也会存入 redis 一份，只不过是以键值对的形式存入的。

（2） Redis 缓存报警信息

Redis 使用比较简单，部署好 Redis 后建立一个哈希表即可按键值对读写。这一部分主要说明为什么选择 Redis 作为缓存。

首先就是 Redis 极高的性能。报警信息的缓存考虑过 HBase 与 Kafka，Hbase 虽然也自带内存缓存并以“写快”著称，但其读写性能仍无法超越纯内存存储的 Redis，而且 Redis 使用起来（编程）比 HBase 简单一些，使得我个人偏向 Redis。考虑 Kafka 是想把报警信息作为一种消息由 Kafka 统一分发，但在开发过程中有两点理由使我拒绝了 Kafka。其一是这么做会增大系统不同模块的耦合程度，其二是从建立 Kafka 连接到接收到实时消息，需要数秒之久，如果用于开发 API 是无法接受的。

其次是 Redis“定时过期”的特点。这本是 Redis 因为内存存储带来的妥协，但在这里反而算是一种优点。因为报警信息时效性很强，时间久远的报警信息几乎没有价值。

```
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> hgetall  my-hash-key
1) "\xe6\x9d\xbe\xe5\xb1\xb1\xe6\xb9\x96\xe5\x8d\x97"
2) "1.0"
3) "\xe6\x9d\xbe\xe5\xb1\xb1\xe6\xb9\x96\xe5\x8d\x971"
4) "1.0"
5) "\xe6\x9d\xbe\xe5\xb1\xb1\xe6\xb9\x96\xe5\x8d\x972"
6) "1.0"
7) "\xe6\x9d\xbe\xe5\xb1\xb1\xe6\xb9\x96\xe5\x8d\x973"
8) "1.0"
127.0.0.1:6379>
```

图 3.3.5 查看 Redis 缓存的报警信息

上图是报警信息在 Redis 中的记录，以出站口名称-车流量的键值对形式存储。最后，传统的做法是让 Redis 做 MySQL 的缓存，但由于本系统主要处理流数据，这么做缓存命中率太低，不仅起不到缓存的效果反而会拖累查询性能，因此 Redis 并没有做 MySQL 查询的缓存，而是由相关同学设计增量查询的方法来缓解。

（19）设计 HBase

如下图所示，HBase 集群包含 3 个 Region Server，分别位于 3 个 HDFS datanode 节点上。

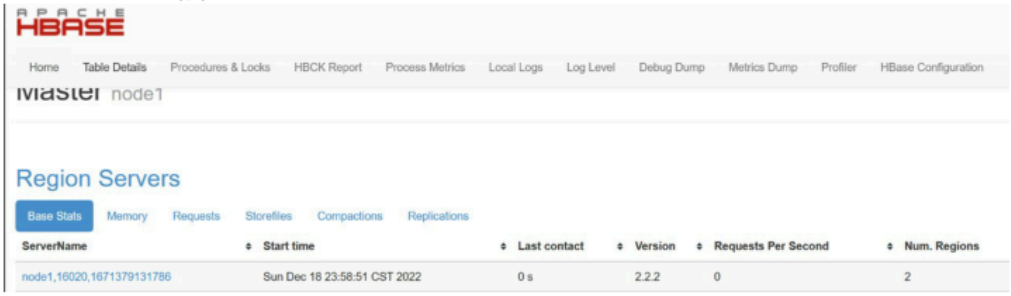


图 3.3.6 HBase 集群

HBase 中存有两张表，ETC_raw 存放实时模拟的流数据，该表主要用于查询获取图表数据，ETC_static 存放根据真实数据集制作的出站口每分钟车流量数据，主要用于与 LSTM 预测的车流量进行对比展示，在系统使用过程中，该表只读不写。

设计 ETC_raw 表。依据散列原则，希望不同 ETC 入口的记录要尽可能随机分布，因此可以将收费站入口编号传入哈希函数，得到的哈希值对 3 取模，余数添加到 rowkey 首部；其余各字段同原始数据，设计结果如下表：

Rowkey	Column Family：Car		Column Family: IN		Column Family: OUT	
hash+收费站出口名称	CP（车牌号）	CX（车型）	SFZRKMC（入口名称）	RKSJ	SFZCKMC（出口名称）	CKSJ
7 广东水朗 D 站	藏	一型车（客）	广东南丫匝道站	2022/12/17	广东水朗 D 站	2022/12/19

ETC_static 表中是静态数据，存有每分钟各出站口的车流量情况。行键为 HH:MM 格式的时间，表示特定的某一分钟，列族为三个出站口名称，每个列族仅有一个属性--车流量 count 。设计结果如下表。

Rowkey 时间窗口	Column Family：LTZXZ count	Column Family：SLDZ count	Column Family：SSHN count
07:04	22	23	24

（20）设计神经网络预测车流量

车流量预测使用的是处理时序数据的长短期记忆网络,长短期记忆网络（Long-Short Term Memory），是一种考虑了时间依赖关系的递归神经网络(RNN)。围绕 LSTM 为核心，设计了一个 5 层的神经网络，其结构如下图。

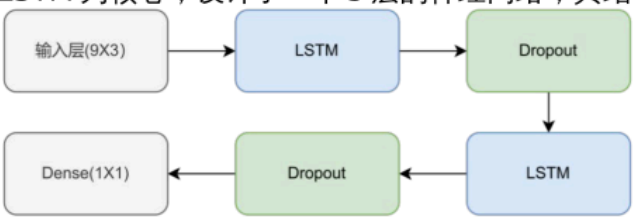


图 3.3.7 神经网络结构图

设计的网络结构包含 2 个 LSTM 层，每个 LSTM 层之后是一个 Dropout 层用于实现部分遗忘，最后通过一个 Dense 层输出结果。网络整体输入为 9×3 的向量，即三个出站口最近 9 分钟的车流量；网络整体输出单个数值结果，即预测的目标出站口下一分钟的车流量。

使用深圳市公开的数据集，对数据进行预处理，按出站口统计每分钟的车流量，得到 637 条数据，数据格式如下表，其中 85%的数据用于训练，15%的

数据用于测试。

时间窗口（分钟）	罗田主线站	水朗 D 站	松山湖南站
17:05	28	3	6

在 epochs=20，batch_size=32 的条件下，得到较优的模型，它在测试集上的结果如下图，准确率可达 92.77%；在华为云 ECS 上使用 CPU 可在 1 秒内完成推理。

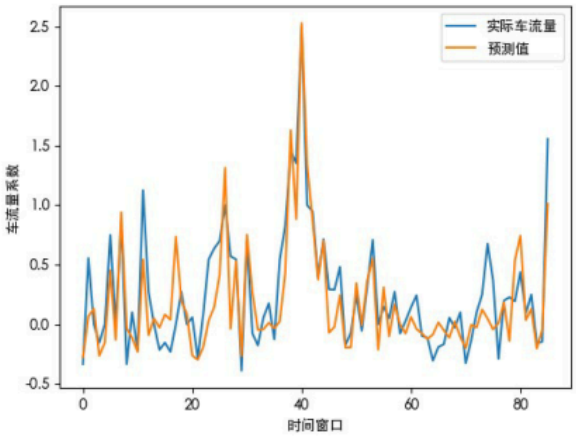


图 3.3.8 神经网络在测试集的表现

（21）Flask 后端接口

使用 Flask 模块，编写后端 API，向前端网页传递信息，接口具体信息见下表。

接口	方法	传入	传出
node2:/alarm	GET		从 Redis 读取的报警信息
node2:/right-top	GET		最近 9 分钟的报警计数信息
node2:/origin	GET	开始时间、结束时间、跨度	指定的某 9 分钟内 3 个出站口车流量信息
node2:/bgpredict	GET	开始时间、结束时间、跨度	根据查询条件，预测下一分钟的车流量

这些接口在 app.py 中实现，每个功能又调用了 util 或 core 文件夹内的对应代码。例如，利用 Flask 框架把 Redis 中的预警信息响应给前端，结果如下图：

（22）设计离线预测页面

离线预测页面主要功能是根据设置的预测条件，展示选定的 9 分钟内三个出站口的车流量信息，并且把预测的下一分钟的车流量信息在不同的图表里展示。设计的结果如下图。

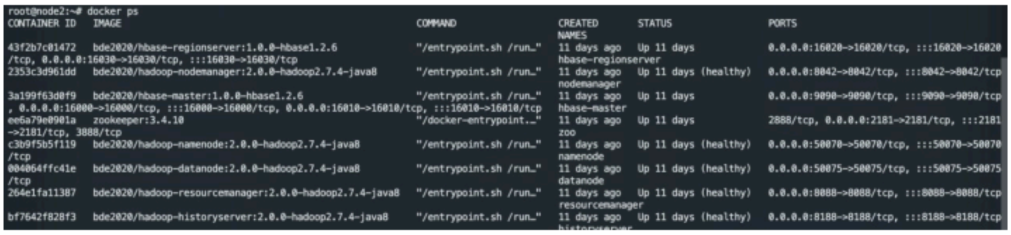
图 3.3.10 离线预测页面

此外，还有一个表格展示 2020-12-22 当日的车流量数据，用于与预测的车流量做对比，还有一个文本框展示治理建议以及时钟天气等装饰组件。

3.5 问题与讨论

（1） HDFS 无法部署于 ECS

在华为云 ECS 上部署了 3 节点的 HDFS 后， Hadoop 和 HDFS 的 Web UI 都能够访问，HDFS 信息也能正确显示，在集群内通过指令也能操作 HDFS，但在集群外却不能正常使用，之后仔细排查发现原因是 HDFS 集群各节点使用的是内网地址。但可能因为 ECS 本身就是虚拟化的云服务器，使用公网地址无法正常搭建 HDFS（Zookeeper 也有相同的问题，但 zookeeper 有相应的虚拟化配置可以解决）。最后通过 Docker 在一台机器上虚拟出 Hadoop 和 HBase 集群，效果如下图，做好容器的端口映射后，可正常对外提供服务。



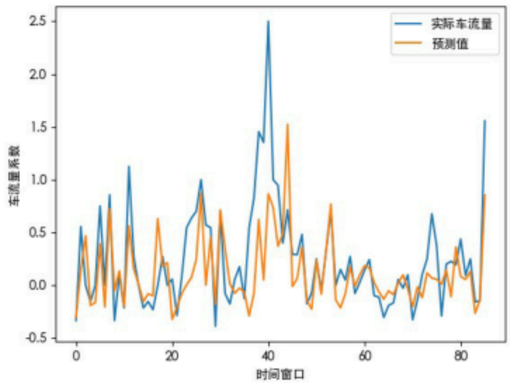
```
root@node2:~# docker ps
CONTAINER ID   IMAGE                                                    COMMAND                  CREATED        STATUS                  PORTS
                                                                                                 NAMES
43f2b7c01472   bde2020/hbase-regionserver:1.0.0-hbase1.2.6              "/entrypoint.sh /run…"   11 days ago    Up 11 days              0.0.0.0:16020->16020/tcp, :::16020->16020
/tcp, 0.0.0.0:16030->16030/tcp, :::16030->16030/tcp                                              hbase-regionserver
2353c3d961dd   bde2020/hadoop-nodemanager:2.0.0-hadoop2.7.4-java8       "/entrypoint.sh /run…"   11 days ago    Up 11 days (healthy)    0.0.0.0:8042->8042/tcp, :::8042->8042/tcp
                                                                                                 nodemanager
3a199f63d0f9   bde2020/hbase-master:1.0.0-hbase1.2.6                    "/entrypoint.sh /run…"   11 days ago    Up 11 days              0.0.0.0:9090->9090/tcp, :::9090->9090/tcp
, 0.0.0.0:16000->16000/tcp, :::16000->16000/tcp, 0.0.0.0:16010->16010/tcp, :::16010->16010/tcp   hbase-master
ee6a79e0901a   zookeeper:3.4.10                                         "/docker-entrypoint.…"   11 days ago    Up 11 days              2888/tcp, 0.0.0.0:2181->2181/tcp, :::2181
->2181/tcp, 3888/tcp                                                                             zoo
c3b9f5b5f119   bde2020/hadoop-namenode:2.0.0-hadoop2.7.4-java8          "/entrypoint.sh /run…"   11 days ago    Up 11 days (healthy)    0.0.0.0:50070->50070/tcp, :::50070->50070
/tcp                                                                                             namenode
004064ffc41e   bde2020/hadoop-datanode:2.0.0-hadoop2.7.4-java8          "/entrypoint.sh /run…"   11 days ago    Up 11 days (healthy)    0.0.0.0:50075->50075/tcp, :::50075->50075
/tcp                                                                                             datanode
264e1fa11387   bde2020/hadoop-resourcemanager:2.0.0-hadoop2.7.4-java8   "/entrypoint.sh /run…"   11 days ago    Up 11 days (healthy)    0.0.0.0:8088->8088/tcp, :::8088->8088/tcp
                                                                                                 resourcemanager
bf7642f828f3   bde2020/hadoop-historyserver:2.0.0-hadoop2.7.4-java8     "/entrypoint.sh /run…"   11 days ago    Up 11 days (healthy)    0.0.0.0:8188->8188/tcp, :::8188->8188/tcp
```

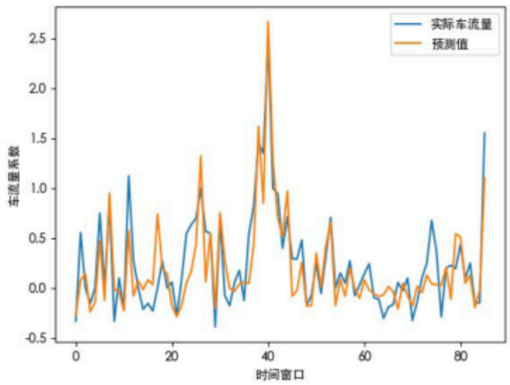
图 3.4.1 docker 集群信息

（23）LSTM 的数据处理

使用神经网络需要花费大量时间在数据的处理上。将原始数据按每分钟统计 3 个站点的车流量，统计出的结果发现数量少了一些，检查数据后发现原始数据中会有某一分钟内都没有记录的情况，这种“伪缺失”数据的情况不会对查询造成影响，但会对神经网络有影响，最后完善统计方式，对于没有记录的某一分钟，其车流量计为 0。

本系统设计的神经网络输入的是 9*3 的向量，即 3 个出站口 9 分钟的车流量，但 3 个出站口的车流量并不在同一范围内，数据尺度有一定差别，如果不进行处理，结果如下图左。





对车流量数据进行正则化处理，选取第一条数据作为基准，进行正则化处理，结果如上右图所示，精度提升明显。

（24）教训

本次实验过程中使用 4 台华为云服务器，算是半个公开环境了，最开始为了开发方便，并没有特别配置防火墙与安全组，直到被黑客入侵，四台差点废掉两台，机器可以再买，但数据无价，如果在实际生产环境中数据库被篡改、删除、脱库等，后果不堪设想。后面通过配置安全组白名单访问，虽然给开发带

来了一些麻烦，但安全始终是第一位的。也庆幸这次安全问题并没有带来很大的损失，这点损失买个教训，让我在以后的开发中始终保持安全意识与数据备份的习惯，如 ssh 设置复杂密码、不用 root 用户登陆、数据库端口不暴露、WEB UI 白名单访问等。

（25）不足与反思

虽然本次设计满足了最开始的若干设计目标，但仍有可改进的地方。除了交互式查询可视化、预测更长时间范围的车流量等碍于工作量的功能实现外，更关键的是整个系统显得“幼稚”--缺少业务理解。比如数据大屏里面的图表完全是看着原始数据想出来的、车流量预测和车流量报警是两个不相关的功能、系统没能给出实质性的建议等，这样技术驱动、缺少真实业务背景的系统，离真正的应用系统还有距离，而且在开发应用系统时，对业务的理解往往比对技术的运用更重要。

（26）总结与感受

本次课程设计用到了很多这学期学的知识，不仅限于大数据存储课程学到的，还包括大数据架构、数据挖掘等，但之前的很多课程设计都还是实验性质的，是在本次课程设计才真正把这些课程的知识用到一起。本次课程设计除了运用已经学到的知识，还促使我学了不少新的知识，比如 Redis 缓存、LSTM 网络的设计等。
如果不是闫老师在课程上的付出，我们很难有这样将理论付诸于实践的机会，感谢闫老师！