观澜交通数据处理平台

董振[1)]，禹晓辉[1)]，崔星灿[1)] 宋仁勇[1)] 林立伟[2)]

[1)](山东大学 计算机科学与技术(系)，济南 中国 250101)
[2)](济南观澜数据技术有限公司，济南 中国 250101)

GrandLand Traffic Data Processing Platform

Dong Zhen[1)], Yu Xiaohui [1)], Cui Xingcan [1)], Song Renyong[1)], and Lin Liwei[2)]
[1)](School of Computer Science and Technology, Shandong University, Jinan 250101, China)
[2)](GrandLand Data Technologies Co., Ltd., Jinan 250101, China)

Abstract Recent years have seen an increasing number of traffic surveillance cameras deployed on the main roads and intersections of metropolitan areas. For large and medium-sized cities, these cameras generate an enormous amount of data, including vehicle passage records, photos, etc. Existing solutions based on relational database systems cannot effectively manage and process such large volume of data, nor can they provide support for efficient and scalable support for either analytical tasks or tasks requiring response in real-time. To address these challenges, we have developed the GrandLand Traffic Data Processing Platform (GLPlatform) to provide distributed, scalable processing of traffic surveillance data. Integrating Apache Hadoop and S4, GLPlatform supports both batch analytical workloads as well as real-time processing tasks. GLPlatform has been successfully running in a production environment in a large city in China for almost two years. This paper will present the architecture and implementation of GLPlatform, and discuss the lessons learned from the design and development of this platform.

Key words traffic data; data processing systems; big data

摘 要 近些年，城市中在主要路段和路口设置的交通卡口点及高清摄像头的数目呈增长趋势。对于大中型城市来说，这些摄像头将会产生海量包含通行记录和图像在内的数据。现有的基于传统关系数据库的解决方案已经无法有效地管理如此大规模的数据，也无法为数据的离线分析和实时处理提供具有高效及可伸缩性的保障。为了解决这一系列问题，我们开发了观澜交通数据处理平台（观澜平台）。该平台可以为交通数据提供分布式、具有良好可伸缩性的处理支持。它集成了Apache Hadoop和S4 开源框架，可以同时运行批处理任务和实时处理任务。观澜平台已经在国内某城市的生产环境中成功运行近两年。本文将会给出平台的架构说明以及在设计和开发过程中的一些收获。

关键词 交通数据；数据处理系统；大数据

中图法分类号 TP391

1 引言

近些年，城市中在主要路段和路口设置的交通卡口点及高清摄像头的数目呈增长趋势。当车辆经过卡口时会被拍照，同时与摄像头相连的工控机可运用车辆识别技术实时产生车辆的结构化通行记录。无论是图像还是记录都会被回传至数据中心进行相关处理。此类数据的价值极大，基于它们可以实现很多诸如交通管制、道路规划以及违法犯罪调查等方面的应用。

图 1 卡口摄像头

交通卡口数据涵盖几乎全部的“大数据”特征：数量巨大、产生迅速、异构性强。考虑如下场景：在一个有100万注册车辆的中型城市设有超过1000个的高清卡口点位。如图 1 卡口摄像头所示，通常，每个卡口都会安装多个摄像头（每个摄像头负责一条或多条车道）。假设每条车道平均 30 秒经过一辆车，那么对于一个有 4 车道的监控卡口，每天将会产生大约 12,000 条的通行记录。每条通行记录包含很多数字或文本字段，例如车牌号码、车身颜色、时间戳、卡口点位信息等。除此之外，被抓拍车辆的通行图像（每张约为 500KB）也会被传输到数据中心。因此，整个城市每日会产生超过 10,000,000 条的通行记录和图片，总量约为 10TB（包含冗余）；而一年的数据量将会达到 3.5PB。综上所述，无论从数据的数量、产生速度还是多样性方面来讲，都会给数据处理平台的建设带来的极大的挑战。

除了数据本身带来的一系列难点，如何能够支撑灵活多样的应用及服务也是一个值得考虑的问题。从处理模式的角度讲，这些应用和服务所可被分为两类——用户驱动和数据驱动。在用户驱动处理模式中，数据首先被存储到数据库或文件系统中，然后用户通过发送请求的方式查询或对数据进行处理；相反，在数据驱动处理模式中，处理条件会首先被安置好，随着数据的到来，可能会不断地触发产生结果，出于效率考虑，这一过程通常会在内存中完成。用户驱动模式往往被用来构建那些交互式的查询分析应用，而数据驱动模式大多数情况下被用在那些对实时性要求较高的场景。

我们对于现有交通卡口数据管理系统的调研表明，它们所能支持的摄像头数有限（几百个），并且可伸缩性较差。因此我们设计并实现了观澜交通数据处理平台（观澜平台），该平台主要具有以下特征：

（1）平台对海量交通数据的处理提供了高性能、高吞吐的存储和处理支撑，并且具有良好的可伸缩性（至少可达到 PB 级别）。

（2）平台同时支持基于用户驱动模型和数据驱动模型的应用及服务。现有已部署的应用不仅涵盖交通管理方面，还有部分应用可为违法犯罪案件侦破提供高层次的支持。（详见第 4 部分）

值得一提的是，观澜平台不仅支持交通卡口数据的存储和处理。它在设计之初就被定位为一个通用性较强的平台，因此可被应用于其他具有相似数据特征和业务需求的场景之中。

基于观澜平台，我们已经实现了一个真实的交通卡口系统，该系统已经被部署在济南市并且在生产环境中稳定运行了近两年的时间。我们使用 120 个服务器节点来支撑超过 1000 个卡口（数千个摄像头），每天大约要处理 1200 万条通行记录。绝大多数的节点都是日常用的服务器（两个 Xeon E5620 处理器，16GB 内存，24TB 硬盘）。现在部署和运行在系统之内的应用及服务已经超过了 30 个。

论文接下来的部分组织如下：章节 2 简单回顾已有的交通数据处理平台；章节 3 对我们平台的架构和实现进行说明；在章节 4 将会挑选并介绍几个典型的应用；最后章节 5 将介绍几点收获。

2 相关工作

绝大多数现有的交通管理系统底层都是采用传统的关系型数据库进行数据存储。这样最大的优势在于可以充分利用关系型数据库的多年来发展的积累，同时，已经有无数相关系统的成功案例可供参考。然而，传统关系型数据库伸缩性不强，换句话说，无法通过简单的添置节点来提高系统整体表现[1]。因此，关系数据库的工作效率极大地受限于其宿主机器的性能。由于绝大多数卡口系统都需要对数据进行永久存储，因此，随着数据量的提升，关系数据库的适应性将变得越来越差。

另一方面，现有系统对数据驱动型应用的支撑不足。例如交警业务中常见的布控报警应用，一个典型的做法是选取一台特殊的服务器来对数据进行分析，而随着数据量的增大，服务器的数量也随之增加，每台服务器对应全部数据源中的某一部分。然而，这种做法仅适用于对数据的分析不依赖于跨节点的全局数据的情况。分析节点的独立性，导致了这种方式无法在大数据环境下完成那些较为复杂，需要多节点协作的实时任务处理。

3 系统架构和实现

我们的目标是构建一个可以部署在主流配置

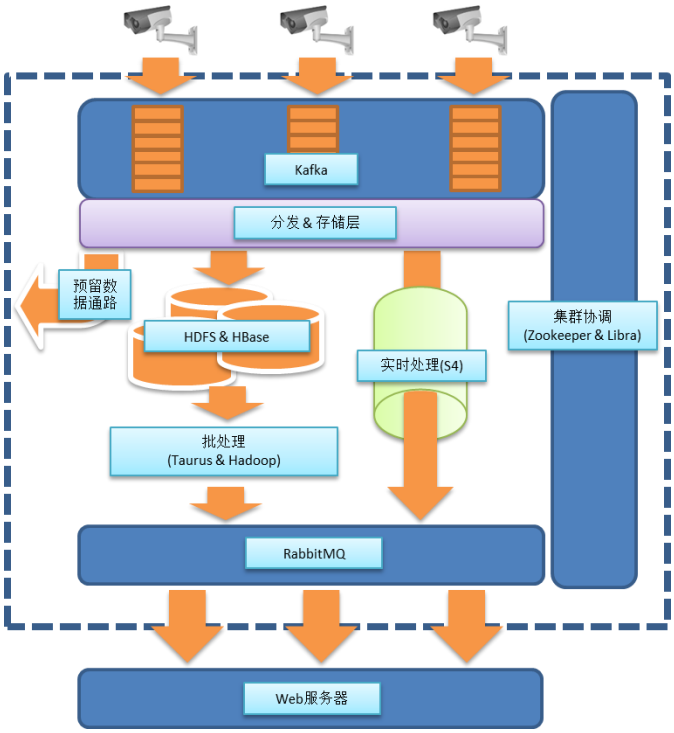

图 2 观澜平台系统架构

服务器集群上的数据处理平台，具有良好的伸缩性，能够支撑包含交通管理、道路规划、违法取证等多方面的服务及应用。为达到这一系列目的，平台设计主要考虑如下方面：

（1）出于平台将会持续从多种来源（集成新建和多个旧有系统）进行数据收集的考虑，需要实现一个统一的、可伸缩性强的数据接入模块。通过这个模块，不同数据来源的车辆通行记录连同其对应的通行图像可以被方便地进行采集。

（2）如上所述，系统需要对海量数据进行长期存储，因此有必要选择一个分布式的数据库系统（如 HBase[2]）为底层数据管理提供支撑。此外，为了完成高效的数据查询处理，需要考虑针对交通数据的索引策略。

（3）虽然流行的分布式处理框架（如 MapReduce[3]）很擅长解决大数据集上的批处理问题，但它无法被简单地拿来构建对实时性要求很高的数据驱动模型。因此除了 MapReduce 我们还需要引入另外的实时处理框架，并且保证二者之间可以良好的协调工作。

（4）最后我们还需要考虑分布式系统中的一些通用问题，如整个系统的高可用性、可伸缩性和负载均衡等。

图 2 观澜平台系统架构展示了我们系统的一个整体架构。自上而下，结构化的通行记录及其对应的通行图像通过一个定义良好的接口被发送到 Kafka[4]消息队列中。

随后，分发和存储层负责从 Kafka 消息队列中取出消息。同时，它能够以最少一次的语义创建多个独立的数据流，即保证每条通行数据在每个数据流中至少被推送一次。每一条数据流之中都可以涵盖全部通行数据，这个设计为后续扩展带来的极大的方便，在有新的数据获取需求时，可以很容易地增加推送数据流。现阶段有两个独立的数据流，分别为后继的批处理模型和实时处理模型提供输入。

批处理模块。结构化的通行记录与其对应图像分离之后被存储到 HBase 之中，而图像被归档并存储在 HDFS[5]文件系统中。我们为每一条通行记录增加了一个字段，用以标明其对应图像的存储地址。要注意的是，为了适应分发存储层所提供的最少一次语义，这里的存储过程需要被设计成幂等的，即对同一条通行记录进行多次存储调用得到的最终结果应是相同的。为了更好地处理相关请求和提供服务，我们设计并实现了一个名为 Taurus 的框架，该框架在功能上有些类似于 YARN[6]，能够对业务请求进行分发，对包含 MapReduce 任务在内的实际业务程序进行调用，并可完成对整个数据处理过程的监控。为了实现交通数据的高效模糊搜索，我们设计并实现了一个简易的查询语言，其处理引擎被绑定在 HBase 节点之上。基于批处理模块，开发人员可以方便的实现诸如交通流量统计、异常行为车辆分析、路径预测等基于用户驱动模型的应用及服务。

实时处理模块。我们的实时处理模块是基于开源的分布式流处理系统 S4[7]实现。该模块可以通过一些预先定义的模式，持续主动地产生并推送结果。由于交管领域相关业务的特性决定了系统需要具备较高的可用性和故障恢复特性，因此我们为 S4 添加了故障恢复策略[8]，使得当运行 S4 处理单元的某一节点宕机无法继续提供服务时，会有其他节点对业务进程和中间结果进行无损失的接管，保证了系统的高可用性。开发人员可以在此模块之上构建复杂多样的基于数据驱动模型的应用，如：实时套牌车辆分析、实时伴随车辆分析等。

下一模块主要由另一个分布式消息队列——RabbitMQ[9]组成，该模块将批处理和实时处理的数据流重新统一起来。Web 服务器的用户请求或是系统产生的结果都通过该模块的队列进行传递。此

外，它还可以提供反馈机制以保证所有放入队列的消息都会被得到充分处理。

平台最后一个模块被称为集群协调模块。作为整个集群的管理者，它由 ZooKeeper 和一系列的协调工具组成，可以为整个集群的良好协作及高可用性提供保障。

4 应用案例

观澜平台集成了 Hadoop 的批处理特性以及 S4 的流处理特性，可以为以用户驱动模型为基础的历史数据分析应用和以数据驱动模型为基础的实时数据处理应用提供良好的支撑。前者主要涵盖交互式的多字段模糊查询及各类对交通数据的离线分析；后者主要包含实时伴随车辆分析、实时套牌车辆分析等一系列对响应时间要求较高的应用。下文将会给出基于两种模型的典型应用案例。

图 3 套牌车辆案例

4.1 套牌车辆分析

不同车辆的号牌种类及号牌号码不应相同。但在实际中，同样的车牌可能会被非法复制多个，分别安装在不同的车辆上。在这些车牌之中，只有一个才是真的，其他都算作套牌（见图 3 套牌车辆案例）。套牌行为往往被车主用作逃避某些费用的手段；个别情况下也会被不法分子利用，以逃避警方的追踪。后种情况下将会给社会公共安全带来一定威胁。为此我们设计并实现了套牌车辆分析这一应用，通过对道路行驶车辆的路径进行实时比对，寻找那些同一时间段出现多条可疑行驶路径的车辆，进而锁定套牌行为并实时推送分析结果。这是一个典型的基于数据驱动模型的应用，传统关系型数据库对此类应用的支持将会十分困难。

4.2 路径预测

如果可以根据车辆历史行驶记录和当前位置预测车辆的下一位置，那么将会给道路通行状况预测和路径推荐带来极大方便。基于现实情况考虑，我们对马尔可夫模型进行改进，通过联系个体与全局的行为概率矩阵，以及引入时间因子，提出并实现了一个全新的车辆下一位置预测模型（详见引文 11）。其在历史数据集上进行学习的过程需要依据时间窗口进行持续的分布式聚类，观澜平台所提供的批处理组件可以帮助方便完成这一任务。通过构建核心的 MapReduce 作业和进行一些基本的输入输出参数配置，系统即可高效自动的完成学习和路径预测。

4.3 伴随车辆分析

伴随车辆是指在一段时间内结伴而行的车辆，一个比较典型的场景是车队。某些违法犯罪行为，如协同犯罪或盗抢机动车辆，嫌疑人在抵达或离开案发现场时也会表现出伴随行为。考虑伴随车辆特征，我们设计并实现了一个基于频繁模式挖掘的实时伴随车辆分析算法，这同样是一个建立在数据驱动模型之上的应用。出于业务需求考虑，算法产生结果的延迟需要尽可能低，以 S4 为基础的实时处理模块可以用来很好的解决这一问题。

4.4 应用实施效果

基于观澜平台，我们实现了多种用户驱动模型和数据驱动模型的应用，它们已被部署在生产环境并稳定运行近两年时间，为相关业务部门及时提供合理的信息化决策支持，创造了巨大的经济、社会效益。

5 收获

在平台设计和实现过程中，我们提出过很多不同的方案，经过理论分析或实践验证，最终保留了现有方案，下面给出几点从中学习的收获。

5.1 Kafka队列对于接入和分发的优势

Kafka 是一个分布式发布/订阅消息系统，我们的平台使用它来完成数据的接入和分发。

Kafka 支持高效的数据持久化，它充分利用操作系统缓存和顺序读写磁盘的特性，以批处理的形式处理磁盘 I/O，大大增加了系统吞吐量。在我们的系统中，交通数据量（包括文本记录和图像）非常庞大，并且后续处理模块基本是顺序获取数据，恰恰可以利用 Kafka 高效顺序 I/O 的特性。借助 Kafka 队列，一方面可以实现海量交通数据的快速持久化，另一方面又为处理模块提供了高效获取数据的方式，从而出色地完成数据接入工作。

另一方面，Kafka 采用消息发布/订阅模型，它提供一套基于生产者/消费者模型的 API，非常适合于多模块对数据的复用；同时，它利用批量消息传输机制，以及减少网络发送时数据在内存中的拷贝等方法提高网络传输效率。在观澜平台中，Kafka 充当了消息缓冲队列，前端数据首先被接入到 Kafka 进行持久化，然后批处理模块和实时处理模块再分别从 Kafka 获取数据进行处理。这样做的好处是提供了很好的数据通路扩展性，可预留新的数据处理流程。

5.2 批处理和实时处理分离的好处

在观澜平台中，两个处理模块（即批处理模块与实时处理模块）相互独立、分离，它们分别从 Kafka 队列获取数据，各自进行后续的数据处理。

两个处理模块并行化的方式使它们的性能互不影响。一方面实时处理模块不会因批处理模块的入库操作而降低实时性；另一方面批处理模块也不会因实时处理模块的事件触发型处理模式而损失吞吐量。

处理模块的分离使得平台架构更加简单清晰，极大地降低了设计、开发和运维的复杂性。两个处理模块分别对应于两种不同类型的业务场景，二者的分离使得功能界限更加明确，有利于开发团队进行内部分工，同时也使得系统的可维护性得到提高。当其中某一模块出现故障不会使另一者受到影响，从而使故障检测过程得到了简化。

5.3 实时处理催生新的业务模型

实时处理模块催生了新的业务模型，即数据驱动。传统业务模型大多采用用户驱动模型，即数据首先被存储到数据库中，然后用户根据特定条件进行查询、分析。而在数据驱动模型中，数据不经磁盘 I/O，直接在内存中按照用户预定义的模式被处理，业务结果被推送给用户。

数据驱动模型的出现，丰富了交通管理领域的业务应用。以往的业务模式都是对交通数据的“事后分析”，时效性相对较低，有时并不能满足应用需求。而“数据主动模型”使得“事前预警”成为现实，随着交通数据的产生和流入，一些分析或预测数据便立即产生，可以为诸如信号调度等一些交通措施的制定提供合理的依据。

致谢 本文相关工作受到国家自然科学基金（61272092）、山东省自然科学基金（ZR2012FZ004）、山东省科技发展计划（2014GGE27178）、973 计划（2015CB352500）、山东大学自主创新基金（2012ZD012）、泰山学者计划、加拿大自然科学基金支持。

参 考 文 献

[1] D. Agrawal, A. El Abbadi, S. Das, and A. J. Elmore, “Database scalability, elasticity, and autonomy in the cloud,” in Database Systems for Advanced Applications. Springer,2011, pp. 2–15.

[2] HBase - Apache HBase Home. [Online]. Available: http://hbase.apache.org/

[3] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.

[4] J. Kreps, N. Narkhede, and J. Rao, “Kafka: A distributed messaging system for log processing,” in Proceedings of the NetDB, 2011.

[5] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE,2010, pp. 1–10.

[6] Apache Hadoop 2.5.0 – YARN [Online], Available: http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

[7] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, “S4: Distributed stream computing platform,” in Data Mining Workshops (ICDMW), 2010 IEEE International Conference on. IEEE, 2010, pp. 170–177.

[8] L. Lin, X. Yu, and N. Koudas, “Pollux: towards scalable distributed real-time search on microblogs,” in EDBT, 2013,pp. 335–346.

[9] Rabbitmq - messaging that just works. [Online]. Available: http://www.rabbitmq.com/

[10] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, “Zookeeper :wait-free coordination for internet-scale systems,” in Proceedings of the 2010 SENIX conference on USENIX annual technical conference, vol. 8, 2010, pp. 11–11.

[11] M. Chen, Y. Liu, and X. Yu, “NLPMM: a next location predictor with Markov modeling,” in PAKDD, 2014.

董 振 男，1991 年生，硕士研究生，主要研究方向为分布式数据管理。

禹晓辉 男，1977 年生，教授，博士生导师，主要研究方向为数据库理论、数据挖掘、大数据管理与分析。

崔星灿 男，1989 年生，博士研究生，主要研究方向为分布式流数据处理、数据质量管理。

宋仁勇 男，1991 年生，硕士研究生，主要研究方向为分布式系统。

林立伟 男，1987 年生，硕士研究生，主要研究方向为分布式流数据处理。