海量车牌识别数据集上基于时空划分的旅行时间计算方法

赵卓峰[1,2],丁维龙[1,2],张 帅[1]
(1. 北方工业大学云计算研究中心,北京 100144;2. 大规模流数据集成与分析技术北京市重点实验室,北京 100144)

摘 要: 城市路段旅行时间计算是智能交通领域的一个研究热点.车牌识别数据作为近年来新兴的一种针对城市道路行驶车辆的实时监测数据,具有持续生成且数据量大、时间空间相关等特性.为了利用车牌识别数据集进行高效、准确的旅行时间计算,给出了基于车牌识别数据集的旅行时间计算定义,在此基础上提出一种基于时空划分的流水线式并行计算模型,并给出了该模型基于实时 MapReduce 的实现.通过一组基于海量真实车牌识别数据集的实验表明,本文方法在亿级车牌识别数据集上的旅行时间计算性能方面相对于直接基于 Hadoop 的实现可以提高 3 倍以上,同时具有适合细粒度划分及受路网规模影响小的特点.

关键词: 旅行时间;时空划分;流水线并行;实时 MapReduce;车牌识别数据
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2016)05-1227-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.05.031

A Travel Time Calculation Method Based on Spatio-Temporal Data Partition of Large-Scale License Plate Recognition Data Set

ZHAO Zhuo-feng[1,2], DING Wei-long[1,2], ZHANG Shuai[1]
(1. *Cloud Computing Research Center, North China University of Technology, Beijing* 100144, *China*;
2. *Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, Beijing* 100144, *China*)

Abstract: The calculation of travel time of city roads is an important issue in the domain of the intelligent transportation system research. License plate recognition data is one kind of monitoring data for vehicles running on urban roads, which has some new features, such as high volume, high velocity and spatio-temporal correlation. In order to achieve travel time calculations on massive license plate recognition data collection, we present the formal definition of travel time calculation based on license plate recognition data set, and propose a pipelined parallel computing model based on spatio-temporal data partition. Moreover, the implementation of the computing model is given based on a real-time MapReduce computing system. The corresponding experiments based on real license plate recognition data set show that, the computing performance on million-level data sets of our method can achieve three times increasing compared to traditional travel time calculation methods. Meanwhile our method is more suitable for fine-grained partition and large scale traffic network.

Key words: travel time; spatio-temporal partition; parallel pipeline; real-time mapreduce; large-scale license plate recognition data

1 引言

路段旅行时间作为城市交通出行信息的关键指标,是智能交通系统的重要基础,对其的研究一直是智能交通领域的热点.城市路段旅行时间可以直接用来评判城市道路的运行状况和拥堵水平,有效的旅行时间监测与分析也可以为城市路网规划、城市道路交通管理与控制、公众出行路线选择提供合理依据.

以往,路段旅行时间主要采用基于样本车辆监测数据的测算方式,即采用样本车辆的旅行时间来测算相关道路的旅行时间[1].目前,样本车辆监测数据的采集主要采用基于浮动车的数据采集方式.浮动车一般是指安装了车载 GPS 定位装置并行驶在城市主干道上的公交汽车和出租车,它可以通过车载 GPS 和无线通信接口周期性的采集车辆行驶数据[2].但由于浮动车数据涉及的样本主要为特种车辆,其由于均具有特定的出行行为,因此会

收稿日期:2014-11-24;修回日期:2015-10-19;责任编辑:马兰英
基金项目:北京市自然科学基金(No.4131001,No.4162021);北京市属高等学校创新团队建设项目(No. IDHT20130502);北方工业大学校科研基金

造成数据覆盖面有限的问题,同时 GPS 数据往往与道路路段关系不能直接匹配,数据质量也缺乏保障,因此在计算路段旅行时间上存在一定的不足.

车牌识别技术是近年来新兴的一种城市通行车辆信息采集技术,它通过对部署在城市道路的摄像头所采集的车辆图像信息进行识别来提取车辆的车牌信息,并在此基础上形成包含车辆标识、出现地点、时间等内容的车牌识别数据. 相比浮动车等车辆信息采集技术,基于车牌识别数据的车辆信息采集技术具有工作连续性强、数据与道路关系精确度高、检测样本量大、覆盖车辆范围广等优点[3]. 因此,基于车牌识别数据的旅行时间计算成为当前测算路段旅行时间的一个新途径,对其的研究具有重要的应用价值和学术意义.

基于车牌识别数据的路段旅行时间计算包括两种情况,即基于实时车牌识别数据的实时旅行时间计算和基于历史车牌识别数据的历史旅行时间计算. 其中,基于实时车牌识别数据的实时旅行时间计算按照一定的周期利用实时接收到车牌识别数据计算城市道路不同路段的实时旅行时间,其结果可用于实时路况信息服务;基于历史车牌识别数据的历史旅行时间计算则主要针对已积累大量历史车牌识别数据而尚未被利用计算旅行时间的情况,来批量处理得到城市道路不同路段历史上的旅行时间数据,其结果可被用来支持对出行规律、道路拥堵特点等的分析. 本文重点解决后一种情况的旅行时间计算问题.

由于车牌识别数据来源于对城市道路行驶车辆的实时监测,其包含车辆标识、监测时间、地理位置等时间、空间以及车辆对象相关的信息,具有典型的时空相关、时序连续、位置可测的特征. 此外,考虑到随着车牌识别摄像头在城市道路大范围部署,车牌识别数据集的规模将大大超过传统采样方法获得数据集的规模. 当前,一个大型城市部署的带车牌识别数据的摄像头可达到 5000 个,假设高峰期每个摄像头车牌识别数据的采集频率可达 1 条/秒,若每天的交通高峰折算率按 0.33 计,则一年将累积车辆识别数据记录数超过 500 亿条,数据存储量超过 2T. 这些数据可构成规模庞大的车牌识别数据集,仅对如此庞大的数据集进行顺序扫描(按 80M/s 的速度计)也需要近 7 个小时. 为此,为满足基于车牌识别数据的路段旅行时间计算需求,迫切需要设计一种针对海量车牌识别数据集上旅行时间计算的高效方法.

本文主要针对海量车牌识别数据集上的路段旅行时间计算的需求,给出了基于车牌识别数据的路段旅行时间形式化定义,并按照该定义分析了海量车牌识别数据集上路段旅行时间计算的关键问题. 在此基础上,利用车牌识别数据的时空相关、对象相关等特征,采取数据划分和任务流水的思路,提出一种基于时空划分的流水线式旅行时间并行计算模型,并给出了一种基于改进的 MapReduce 模型的计算实现和基于真实车牌识别数据集的验证分析.

2 路段旅行时间计算问题

2.1 问题定义

定义 1 受测道路路网. 受测道路路网指由部署在城市道路上监测点及其之间涉及的路段构成的道路结构,可表示为 $R=(N,S)$,其中 N 为道路监测点集合,S 为路段集合.

定义 2 车牌识别数据集. 车牌识别数据集 L 是指受测路网上各监测点捕获的所有车辆信息数据,其中每条车牌识别数据 $l\in L$ 可表示为(v_i,n_k^i),其中 v_i 表示车牌号码(可唯一代表一个车辆),$o(n\times m^2)$ 表示车辆 v_i 经过监测点 n_k. 进一步,$n_k^i=(n_k^i.l,n_k^i.t)$,其中 $n_k^i.l$ 表示车辆经过的监测点 n_k 的地理位置,$n_k^i.t$ 表示车辆经过监测点 n_k 的时间.

定义 3 路段. 路段指连接两个相邻监测点之间的一条道路,路段 s_i 可表示为两个监测点的有序对 $<n_p,n_q>$,n_p 和 n_q 分别表示路段 s_i 的起始点和终止点,$s_i\in S$.

定义 4 单车路段旅行时间. 单车路段旅行时间 $tra_{v_i}^{s_j}$ 指某一车辆 v_i 经过路段 s_j 所花费的行驶时间,其可以通过该车经过路段 s_j 起止监测点的时间差计算得出,即 $tra_{v_i}^{s_j}=n_q^i.t-n_p^i.t$.

定义 5 路段旅行时间. 路段旅行时间 $tra_{\delta_j}^{s_i}$ 指在给定的时间区间 δ_j 内某个路段 s_i 的车辆通行时间,该时间可以通过取时间区间 δ_j 内通过 s_i 的所有车辆的单车路段旅行时间 $tra_{v_i}^{s_j}$ 的中位数获得.

时间区间 δ_j 指用于度量路段旅行时间的一个时间跨度,我们可以将给定的待测时间范围内按照时间周期划分为不同的时间区间 δ_j,所有的时间区间集合为 δ. 在现有的旅行时间研究中,一般选取每 1 小时、每 15 分钟和每 5 分钟三个时间周期进行时间区间划分以对路段旅行时间进行度量[1]. 例如每 15 分钟的时间周期中,将对 0:00 – 0:15、0:15 – 0:20、……、23:45 – 24:00 等一天中的 96 个时间区间进行旅行时间计算.

根据单车旅行时间获得路段旅行时间还可以通过求平均值或加权平均值的方法来计算,但这里考虑到实际车辆出行中的一些特殊因素,如车辆在路段中的临时停留或路边停车以及套牌车等,这些因素下得到的单车旅行时间都会有较大的失真,并会给平均值方法计算路段旅行时间的准确性带来一定的影响. 为此,这里我们选择中位数方法来获得路段旅行时间,以屏蔽不合理的单车旅行时间带来的影响. 此外,这种方法下,路段旅行时间计算一般要求在特定时间区间内通

过该路段的车辆数(即计算得到的单车旅行时间数)大于5辆,以避免由于车辆数过少而造成的中位数代表的旅行时间不精确问题. 在单车旅行时间数小于5条时,将直接将前一时间周期计算得到的路段旅行时间作为当前实际周期的路段旅行时间.

根据上述定义,基于车牌识别数据的路段旅行时间计算问题可以看作是:给定旅行时间计算时间周期和一定时间范围内的历史车牌识别数据集,对受测道路路网 R 中的所有 n 条路段 S 求其在给定时间范围不同历史时间区间上路段旅行时间,即求 $TRA=\{TRA^{s_i}\mid 1\leqslant i\leqslant n, s_i\in S\}$,其中 TRA^{s_i} 表示一个路段在给定时间范围不同历史时间区间上旅行时间结果集. 对于 TRA^{s_i},可以按照定义4计算其在所有不同历史时间区间 δ 上的所有单车旅行时间 $tra^{s_i}_{v_i}$,并进一步按照定义5求得最终不同历史时间区间上路段旅行时间 $tra^{s_i}_{\delta_j}$,从而得到该路段在给定时间范围不同历史时间区间上旅行时间结果集,即 $TRA^{s_i}=\{tra^{s_i}_{\delta_j}\mid 1\leqslant j\leqslant m, \delta_j\in\delta\}$,其中 m 为待测时间范围内按时间周期划分得到的时间区间数.

根据上述定义可以看出,相对于传统基于浮动车采样数据的旅行时间计算方法需要进行路段复合计算以及建立区分拥堵、非拥堵和路口排队等情况的计算模型[4],基于车牌识别数据的路段旅行时间计算模型则相对简单和直观,只是在路段所采集的车牌识别数据较少时可能存在一定的旅行时间计算误差而不得不采取复制前一时间区间旅行时间的计算方式. 实际上,在这种情况下,可以综合浮动车数据进行旅行时间计算,而且此时由于路段车流量不大,不存在前述基于浮动车采样数据的旅行时间复杂计算情况. 此外,基于车牌识别数据的路段旅行时间计算由于使用覆盖范围更广的车牌识别数据,相对与采样性的浮动车数据,其数据规模会更大,因此如何在海量车牌识别数据集(亿级数据记录)基础上计算符合上述定义的路段旅行时间成为一个亟待解决的关键问题.

2.2　问题分析

如图1所示,按照上述路段旅行时间计算定义,在计算时首先需要对车牌识别数据集中的所有车牌识别数据按照时间区间(假设有 j 个时间区间,如图1横坐标 δ_1 至 δ_j,其中的点 l_{ix} 为一条车牌识别数据)进行划分,对同一划分中的数据(假设每个划分中数据为 k 条)两两比对,若两条数据属于同一辆车并且两条数据中涉及的两个监测点(如图纵坐标 n_1 和 n_2)是受测道路路网中的路段,则求出并保存该路段在该历史时间区间的单车旅行时间(如图1中 l_{11}、l_{22} 等为同一车牌的数据,即同一车辆);最后,再对所有路段的不同时间区间上全部单车旅行时间取中位数,得到最终道路历史旅行时间结果集,该算法的复杂度为 $O(j\times k^2)$. 在实际情况中,当需要计算一个大型城市一年的历史数据(车牌识别数据上亿甚至百亿)、计算时间周期为5分钟时,j 为105120,k 最大约为150万左右. 此外,上述计算还受到路网中路段数量的影响,当受测路网中路段数达到1000时,一年的路段旅行时间计算结果集数据条目数也将过亿. 输入、缓存及产生如此大规模数据,都将使得旅行时间计算的执行时间将急剧增加.

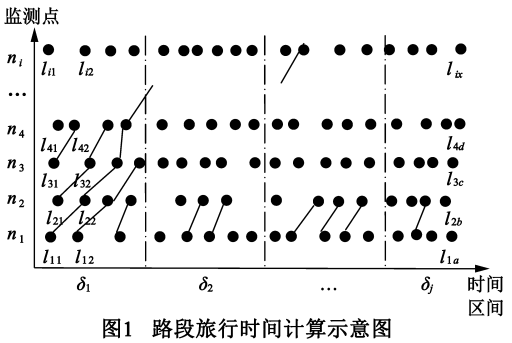

图1　路段旅行时间计算示意图

为提高如上所述规模的车牌识别数据集基础上路段旅行时间计算的性能,需要解决以下两方面关键问题:

(1)如何根据前述路段旅行时间问题的定义并结合具有时空相关等特性的车牌识别数据特征设计旅行时间计算的并行化模式;(2)如何进行有效的车牌识别数据划分,以便于将原始数据和中间结果动态分布到不同节点上处理,并可以尽量避免旅行时间计算过程中跨节点的数据访问.

针对上述问题,本文利用车牌识别数据的时空相关、对象相关等特征,采取数据划分和任务流水的思路,提出一种基于时空划分的流水线式并行处理模型来解决基于海量车牌识别数据的路段旅行时间高效计算问题.

3　基于时空划分的流水线式旅行时间并行计算模型

3.1　模型描述

根据前文对路段旅行时间计算的分析可以看出,该计算的处理逻辑主要是针对不同时间区间的车牌识别数据,先计算所有路段上的单车旅行时间再通过求单车旅行时间中位数来计算路段旅行时间. 基于此,可以从时间和空间两个角度来对原始车牌识别数据进行划分与组织,而计算过程可以区分为相关车牌识别数据加载、单车旅行时间计算、单车旅行时间中位数计算三个阶段的子任务. 同时,根据旅行时间计算定义可以看出,在具体计算过程中需要进行车辆对象和空间关系(即一条路段所涉及的两个监测点)的判定比对. 而为了减少后续分布式旅行时间计算过程中跨节点的数据传输,为此要求在数据划分时能够减少跨数据划分

分区的计算,特别是顶层的数据划分方法应尽量避免跨分区的计算. 因此,在这里,针对旅行时间计算涉及的时间、空间和对象三个维度,在车牌识别数据划分时应首先从与后续单车旅行时间计算等无关的时间维度进行划分,以确保据此分布到各计算节点的数据不必为了相关计算而进行跨节点的传输.

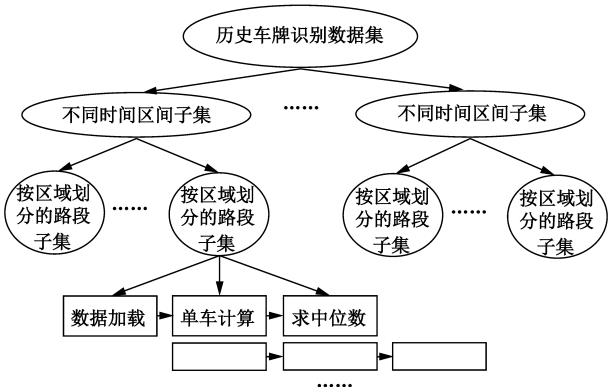

图2 基于时空划分的流水线式并行计算模型

按照上述认识,我们设计了如图 2 所示的基于时空划分的流水线式路段旅行时间并行计算模型. 在该模型中,车牌识别数据首先可以划分为时间维上区分的数据组,每组数据都包含一定数量的不同时间区间的数据子集;进一步某一时间区间的数据子集还可以根据所属路段划分为区域相关的数据子集,区域内数据可以进一步按照车辆数据对象进行组织;在上述两级数据划分基础上,由于相关车牌识别数据加载、单车旅行时间计算、单车旅行时间中位数计算三个子任务的计算结果间不具有直接的因果相关性,即每个子任务仅依赖于特定的中间结果,而不一定必须是其上游任务的输出结果. 因此路段旅行时间的计算可以对此三个子任务进行阶段化划分,从而可以以流水线方式针对不同划分的数据子集来完成路段旅行时间计算. 此外,在按区域划分数据时除了先按路段划分再按车辆对象组织外,还可以选择先按车辆对象划分再按路段组织,在文章后续部分我们将通过实验来验证这两种划分方式下旅行时间计算性能方面的差异并分析相关原因.

上述模型从总体层面看遵循了单控制流多数据流(SPMD,single process and multiple data)的并行模式[5],而在数据并行方面结合车牌识别数据的时空特征可以进一步从时间维和空间维两级划分,在计算任务方面则可以利用旅行时间计算步骤的细分形成流水线式并行.

3.2 基于实时 MapReduce 的模型实现

为实现上述基于时空划分的流水线式路段旅行时间并行处理模型,采用我们之前提出的实时 MapReduce 计算模型[6]思路并在其基础上来实现时空划分的数据并行化及流水线式的计算任务并行化. 实时 MapRedcue 计算模型的核心思想是通过在传统 MapReduce 实现中引入中间结果的缓存机制和 Map 及 Reduce 任务的流水线式处理来提高 MapReduce 模型的处理性能.

在本文中,我们按照上述时空划分的流水线并行计算模型,通过设计适于时空划分处理的分布式 Hash B 树缓存数据结构来优化本地中间结果的高并发读写性能,并进一步通过定义路段旅行时间计算的流水线处理阶段来实现阶段化的流水线式 MapReduce 处理以提高旅行时间计算性能.

具体的模型实现机制如下:

(1)基于分布式 Hash B 树的数据缓存

对于旅行时间计算中涉及的海量历史车牌识别数据,我们采用 Hash B 树来进行内存中的缓存. 在该结构中,首先为支持时间区间划分采用时间区间作为 Key,相同时间区间的车牌识别数据在 Hash 表的同一项中用 B 树组织;其次,监测点作为空间划分基础并用来组织最终的车牌识别数据,每个监测点的车牌识别数据在 B 树的叶节点用链表按照时间顺序进行组织.

对于计算过程中产生的大量单车旅行时间计算结果,我们同样采用类似的结构来进行缓存,主要区别就是将上述 Hash B 树中叶节点由监测点识别数据替换为特定车辆在不同路段的旅行时间链表.

此外,在缓存机制方面,进一步还可利用 B 树在平均搜索性能和平衡性方面的特征,使用多路搜索方式提升了 B 树的并发查询性能,并且利用读写开销估算和缓冲区信息改造外存 SSTable 文件读写策略和内外存替换机制. 由于篇幅所限,更多的计算中间结果缓存机制细节可以参见我们之前在文献[7]中给出的介绍.

(2)阶段化流水线处理

针对旅行时间计算处理逻辑中涉及的相关车牌识别数据加载、单车旅行时间计算、单车旅行时间中位数查找三个子任务,可采用两次 MapReduce 迭代以及 Map 和 Reduce 阶段的流水线控制机制来实现阶段化流水线处理. 其中,第一次 MapReduce 处理中的 Map 函数完成车牌识别数据的划分读入和如前所述的数据结构组织,得到形如 < Key:时间区间 + 监测点,Value:车牌号 + 时间 > 的键值对,Reduce 函数根据路段信息对中间结果按照车牌号进行重组织得到形为 < Key:时间区间 + 路段(监测点 1,监测点 2),Value:车牌号及时间点 1 和时间点 2 > 的键值对;第二次 MapReduce 处理中的 Map 函数仅计算单车路段旅行时间而不做数据变换,Reduce 函数进行所有单车旅行时间中位数查找,最后得到键值对 < Key:时间区间 + 路段,Value:旅行时间值 > . 一个路段旅行时间计算结果示例形如 < 201310020830 + LD0014 [JCD06,JCD07],360 > ,该示例表示在 2013 年 10 月 2 号

8 点 30 分到 8 点 45 分的时间区间,0014 号路段(即监测点 JCD06 到监测点 JCD07 的路段)的旅行时间为 360s. 通过上述描述可以看到,基于区域路段的划分方式下旅行时间计算通过两次循环对监测点 Map 列表中的 key 值进行两两比对,判断是否为路段基础表中数据. 然后通过对于路段两个上下游监测点对应的过车记录列表进行对比,找出相同车辆及其通过的时间,并将得到的属性值整合成 < Key:时间区间 + 监测点,Value:车牌号 + 时间 1 + 时间 2 > 的键值对形式,其算法时间复杂度为 $O(n \times m^2)$,其中 m 为全路网中路段监测点数目,n 为一个时间区间下通过一个监测点的车辆数.

4　实验与分析

4.1　实验设置

实验环境采用的是在五台服务器上搭建的集群环境,并在其上部署我们基于 Hadoop 扩展了中间结果缓存和流水线处理机制后的路段旅行时间计算实现. 其中,Master 节点配置为 4 核 CPU、4G 内存,master 节点同时也被当作计算节点;另外四台 Slave 节点配置为 2 核 CPU、4G 内存,作为计算节点. 此外,每台服务器的有效容量为 80G,集群总存储容量为 400G. 实验中采用的数据为北京市一千多个带识别功能的摄像头 2012 - 10 - 17 到 2013 - 01 - 04 这 80 天采集到的真实车牌识别数据,总数据量近 5 亿条.

为了从性能对比、关键参数影响和扩展性三方面对本文提出的旅行时间计算方法进行验证分析,以及考察旅行时间计算中路段数目(代表受测路网规模)、车牌识别数据记录数和 Hadoop 集群节点数三个参数对旅行时间计算的不同影响,我们设计了如下的一组实验:

实验 1　考察在 5 个计算节点、路段数目固定为 210 的情况下,分别测试从 5000 万到 4 亿等不同数量的车牌识别数据集下 5 分钟、15 分钟和 1 小时三个时间周期路段旅行时间计算性能情况. 并选取直接基于 Hadoop 实现的旅行时间计算方法(LMR)[8] 作为比较对象,与本文方法(CMR)进行性能比较.

实验 2　对两种数据二次划分方式进行对比,(1)CMR 方法中最终采取的先按路段划分再按车辆对象组织方式(记为 RegionList 方式),(2)先按车辆对象划分再按路段组织方式(记为 VehicleList 方式). 实验在 5 个计算节点、路段数目不变的情况下,通过输入不同数量的车牌识别数据,考察两种划分方式下路段旅行时间计算性能差异.

实验 3　考察受测路网中的路段数目对路段旅行时间计算方法性能的影响. 选取 20 天的车牌识别数据(约 1 亿条)作为原始计算数据集,在 5 个计算节点下,对受测路网中的路段数目从 10 到 210 进行调整,并分别测试 5 分钟、15 分钟和 1 小时三个时间周期下的计算性能情况.

4.2　实验结果分析

(1)计算性能对比分析

通过实验 1 得到如图 3 所示结果. 从图中可看出,随着参与计算的车牌识别数据集数据量的增加,两种计算方法的计算时间均呈线性增加. 但 CMR 方法在计算效率上比 LMR 方法有较高的提升,并且 CMR 方法受时间周期差异的影响比 LMR 方法小很多,5 分钟、15 分钟和 1 小时三个不同时间周期下计算时间的差异均在 100 秒以内.

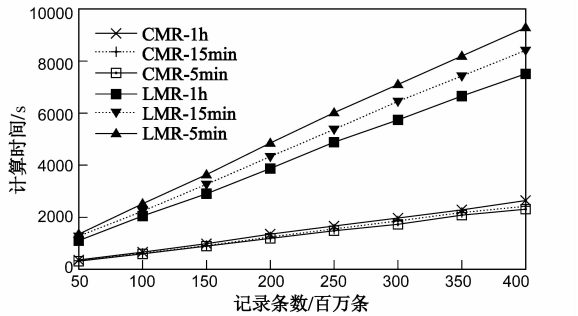

图3　车牌识别数据量对旅行时间计算性能的影响

此外,从图中还可以看到,LMR 方法在计算时间周期越短(即时间段划分粒度越细)的情况下,计算时间越长,5 分钟周期下的计算时间最长,而 CMR 方法恰好相反,在计算时间周期变短的情况下,计算时间反而会略微减少,5 分钟周期下的计算时间最短. 究其原因,主要因为当计算时间周期较小时,需要计算旅行时间的时间区间数量会大幅增加,使得 Hadoop 运行态中的 Map 和 Reduce 任务大增并带来较大的任务执行调度代价,传统 LMR 方法由于未根据车牌识别数据进行划分优化,执行中需要大量的 Map 和 Reduce 任务间的同步等待,从而使得小时间周期下的计算时间变长. 而 CMR 方法通过时空划分和流水线执行避免了无必要 Map 和 Reduce 任务间由于数据依赖的同步等待,同时优化了执行效率,这样单个 Map 和 Reduce 任务一次处理的数据量(受时间区间大小影响)成为影响计算时间的主要因素,因此使得短时间周期下的计算时间反而变短. 由此可见,CMR 方法更能适应细粒度时间周期的旅行时间计算.

(2)不同数据划分方式的对比分析

通过图 4 所示的实验 2 结果可以看出,两种不同的数据划分方法下的旅行计算时间均随数据量增加线性增长,但是随着数据量增加,在计算效率上先按路段划分再按车辆对象组织(RegionList)的划分方式比先按车辆对象划分再按路段组织(VehicleList)的划分方式表现出更好的性能. 通过分析可以看出,在单车旅行时间求解步骤中的第一种划分方式的计算复杂度主要依赖

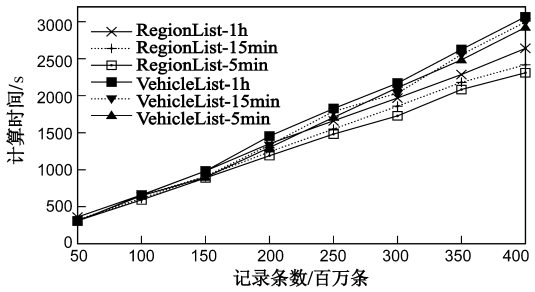

图4 不同数据划分方式对旅行时间计算的影响

全路网中监测点数目和一个时间区间下通过一个监测点的车辆数,而第二种划分方式的计算复杂度则依赖于一个时间区间下一辆车经过检测点数目和全路网中车辆数. 由于实际情况中,全路网中车辆数的量级明显大于其它参数的量级,因此先按路段划分再按车辆对象组织的区域划分方式能更好的适应路网中存在大规模车辆的情况.

(3)关键参数影响对比分析

由实验 3 我们得到如图 5 所示的实验结果. 从实验结果可看出,随着受测路网中路段数目的增加,本文 CMR 计算方法的计算时间基本平滑,而 LMR 方法则在路段数增大时表现出计算时间线性增长的趋势. 这表明 CMR 计算方法的计算性能基本不受路段数目的影响,当我们增加受测路网规模(即增加路段数)时,并不影响旅行时间计算的计算性能. 其中的主要原因在于,路段数在旅行时间计算中主要影响是会增大计算中间结果的规模,CMR 方法由于采用基于分布式 Hash B 树缓存结构优化了中间结果的处理,因此其受路段数变化的影响较小.

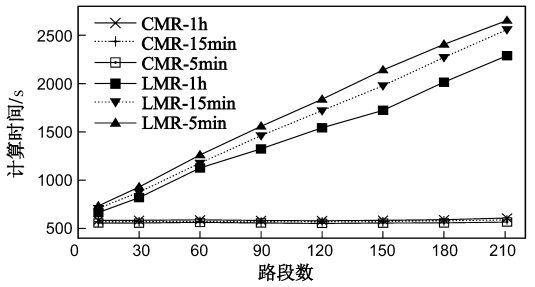

图5 路段数对旅行时间计算的影响

5 相关工作

旅行时间计算一直是智能交通研究中的一个热点问题. 文献[3]利用十余个道路交叉口的车牌识别数据,应用假设检验的方法研究城市快速路及主干道的旅行时间. 然而上述工作使用的采样数据仅为 1 小时内数据,数据量较小,在考虑城市所有路段旅行时间计算时,这种分布估计方法很难适用. 文献[8]通过 Hadoop 实现了基于海量车牌识别数据的城市道路旅行时间实测计算,支持自定义路段集下不同时间区间的道路旅行时间计算. 但该工作中并未根据车牌识别数据的时空特性进行专门设计,仅直接运用 Hadoop 给出了旅行时间计算的朴素实现. 文献[9]将 MapReduce 分布式计算框架应用于道路交通流量统计计算中,证明了利用 Hadoop 技术进行交通数据存储和处理是合理、可行、高效的,但该工作仅适用于简单的数据统计计算,对于旅行时间计算这种需考虑到数据时空特性的计算还需对现有 MapReduce 模型进行相应修改才能提高计算效率.

近年来,很多研究者根据不同应用需求对 MapReduce 模型及其开源实现 Hadoop 进行改进,以适应不同类型大数据的处理需求. 文献[10]设计并实现了 G - Hadoop,在此平台上可以应用 MapReduce 编程框架在多个集群上进行并行计算. 与本文思路相似,文献[11]按照重用 MapReduce 处理过程中中间结果的思路设计了一种 Hadoop 扩展实现的 HaLoop,HaLoop 提供了在 MapReduce 基础上表达循环式处理的编程模型以及可保证多次循环中任务可以调度到同一节点的执行调度器,基于这种设计 MapReduce 处理过程中的中间结果可以被缓存与重用. 同样,文献[12]在集中式处理基础上,设计了一个支持可扩展的并行分布式流处理系统,也能响应流式数据上的连续的查询请求. 为了提升 MapReduce 对迭代执行类的程序的支持能力,文献[13]设计了一种扩展的 MapReduce 框架,与 HaLoop 工作的不同之处在于该框架还支持对 Map 任务的异步执行同时允许每次迭代不必充分创建 Map/Reduce 任务. Spark[14] 是近两年兴起的一类新的分布式计算框架,其采用基于内存的方式来提升迭代式 MapReduce 作业的计算效率,在中间结果处理的设计思路上与我们的工作有相似之处,但是本文工作的主要特点是在计算中结合交通数据特征融入具有时空特性的数据模型进行优化,更适于实现基于时空划分的旅行时间流水线式计算.

综上,现有的相关工作大都集中在通用的计算平台,在时空对象数据划分及执行优化方面缺乏从交通数据及其计算特征方面的针对性设计,因此不能直接用来解决具有周期性批数据处理特点的旅行时间计算问题. 相对于上述工作,本文则是利用车牌识别数据的时空特性并参考上述相关工作,重点通过数据的时空划分和计算任务的流水线设计实现对旅行时间计算模型进行优化,以达到提高旅行时间计算性能的目的.

6 结束语

海量车牌识别数据集上的旅行时间计算是当前智能交通应用建设中一个新的探索. 本文针对该问题,定义了基于车牌识别数据的旅行时间计算概念,提出一

种基于时空划分的流水线式并行计算模型,并给出了该模型基于实时 MapReduce 的实现. 通过一组基于海量真实车牌识别数据集的实验表明,相对于传统的旅行时间计算方式,本文方法表现出了较高的性能,同时具有适合细粒度划分及受路网规模影响小等特点. 下一步的工作包括在百亿记录级模拟数据上的实验测试以及时空并行计算模型在其它交通应用中的适用性验证与分析.

参考文献

[1] 朱爱华. 基于浮动车数据的路段旅行时间预测研究[D]. 北京:北京交通大学,2008.
Zhu Aihua. Research on link travel time prediction method based on data collected by floating car[D]. Beijing Jiaotong University,2008. (in Chinese)

[2] 廖律超,等. 一种支持轨迹大数据潜在语义相关性挖掘的谱聚类方法[J]. 电子学报,2015,43(5):956 - 964.
Liao Lvchao,et al. A spectral clustering method for big trajectory data mining with latent semantic correlation[J]. Acta Electronica Sinica,2015,43(5):956 - 964. (in Chinese)

[3] 姜桂艳,等. 基于车牌识别数据的交通拥堵识别方法[J]. 哈尔滨工业大学学报,2011,43(4):131 - 135.
Jang Jiayan, et al. Traffic congestion identification method based on license plate recognition data[J]. Journal of Harbin Institute of Technology,2011,43(4):131 - 135. (in Chinese)

[4] 刘好德. 基于浮动车数据的城市交通状态判别与行程时间计算[R]. 同济大学,博士后出站报告,2010.
Liu Haode. Estimation of urban traffic status and prediction of travel time based on floating car data[R]. Research Report for Post-Doctor,Tongji University,2010. (in Chinese)

[5] 朱定局. 并行时空模型[M]. 北京:科学出版社,2009.
Zhu Dingjun. Parallel Space-Time Model[M]. Beijing,Science Press,2009. (in Chinese)

[6] 亓开元,赵卓峰,房俊,马强. 针对高速数据流的大规模数据实时处理方法[J]. 计算机学报,2012,35(3):477 - 490.
Qi Kaiyuan, Zhao Zhuofeng, Fang Jun. Real-time processing for high speed data stream over large scale data[J]. Chinese Journal of Computers,2012,35(3):477 - 490. (in Chinese)

[7] 亓开元,韩燕波,赵卓峰,等. 支持高并发数据流处理的 MapReduce 中间结果缓存. 计算机研究与发展,2013,50(1):111 - 121.
Qi Kaiyuan,Han Yanbo,Zhao Zhuofeng,et al. MapReduce intermediate result cache for concurrent data stream processing[J]. Journal of Computer Research and Development,2013,50(1):111 - 121. (in Chinese)

[8] 张帅,赵卓峰,丁维龙. 基于 MapReduce 的城市道路旅行时间实测计算[J]. 计算机与数字工程,2014,42(9):1542 - 1546.
Zhang Shuai, Zhao Zhuofeng, Ding Weilong. Urban road trip time measured calculation based on mapReduce[J]. Computer & Digital Engineering, 2014, 42(9): 1542 - 1546. (in Chinese)

[9] 廖飞,黄晟,龚德俊. 基于 Hadoop 的城市道路交通流量数据分布式存储与挖掘分析研究[J]. 公路与汽运,2013,27(5):82 - 86.
Liao Fei, Huang Sheng, Gong Dejun. Distributed storage and data mining analysis of urban road traffic based on hadoop[J]. Highways & Automotive Applications, 2013, 27(5):82 - 86. (in Chinese)

[10] Lizhe Wang,et al. G-Hadoop:MapReduce across distributed data centers for data-intensive computing[J]. Future Generation Computer Systems,2013,29(3):739 - 750.

[11] Yingyi Bu, Bill Howe, Magdalena Balazinska, et al. The haLoop approach to large-scale iterative data analysis[J]. VLDB Journal,2012,21(2):169 - 190.

[12] 张鹏,刘庆云,谭建龙,等. 流水行云:支持可扩展的并行分布式流处理系统[J]. 电子学报,2015,43(4):639 - 646.
Zhang Peng,Liu Qingyun,Tan Jianlong,et al. SPSPS:A scalable parallel-distributed stream processing system[J]. Acta Electronica Sinica,2015,43(4):639 - 646. (in Chinese)

[13] Yanfeng Zhang, et al. iMapReduce: A distributed computing framework for iterative computation[J]. Journal of Grid Computing,2012,10:47 - 68.

[14] Matei Zaharia. An Architecture for fast and general data processing on large clusters[R]. University of California,Berkeley,Technical Report No. UCB/EECS-2014-12,2014.

作者简介

赵卓峰　男,1977 年 5 月出生,山东济南人. 博士,现为北方工业大学云计算研究中心副研究员、副主任. 研究方向为云计算、流数据处理、服务计算、智能交通.
E-mail:edzhao@ncut.edu.cn

丁维龙　男,1983 年 3 月出生,山东泰安人. 博士,现为北方工业大学云计算研究中心助理研究员. 主要研究方向为流数据处理、流计算及分布式系统.
E-mail:dingweilong@ncut.edu.cn