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，张 帅，丁维龙

（北方工业大学 云计算研究中心 北京 100040）

摘要：城市路段旅行时间计算是智能交通领域研究的热点问题之一，精确的旅行时间计算有助于实现路网调度，减少交通拥塞。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，相对于传统的旅行时间计算方式，本文方法在亿级车牌数据集上的旅行时间计算性能方面可以提高 3 倍以上，同时具有适合细粒度划分、受路网规模影响小及扩展性好的特点。

关键字：旅行时间；时空划分；流水线并行；实时 MapReduce

中图分类号：TP301

A Travel Time Calculation Method Based on Spatio-temporal Data Partition of Massive License Plate Recognition Data Sets

Zhao Zhuo-feng, Zhang Shuai, Ding Wei-long

（Research Center for Cloud Computing, North China University of Technology, Beijing 100041, China）

Abstract: The calculation of travel time of city roads is an important issue in the domain of the intelligent transportation system research. Accurate calculation of travel time can effectively help us to control the urban road system and avoid traffic congestion. 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 method. Meanwhile our method is more suitable for fine-grained partition and large scale traffic network.

Keywords: travel time; spatio-temporal data partition; parallel pipeline processing; real-time mapreduce

Class Number TP301

收稿日期：2014-10-28；修回日期：2014-10-30

资助项目：国家自然科学基金重点项目（61033006），北京市自然科学基金（No. 4133083），北京市属高等学校创新团队建设与教师职业发展计划项目，北方工业大学校科研基金

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，对于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，而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，对缓解城市交通问题也可起到较好的帮助。

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预测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。但由于浮动车数据涉及的采用样本主要为特种车辆，其覆盖面有限，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[2]。

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。随着车牌识别技术的完善以及车牌捕获率与识别率的显著提高，基于车牌识别数据的车辆出行信息采集手段在很多城市被广泛部署应用。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、数据精确度高、检测样本量大、覆盖车辆范围广等优点[2]。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[3]。

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是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。本文所述的基于车牌识别数据的路段旅行时间计算特指后一种情况，即基于历史车牌识别数据的历史旅行时间计算。

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。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，并给出了该模型基于一种实时MapReduce的实现和基于真实数据集的实验分析。

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$\left(v_i, n_k^i\right)$,其中v_i表示车牌号码（可唯一代表一个车辆），n_k^i表示车辆v_i经过监测点n_k。进一步，$n_k^i = \left(n_k^i.l, n_k^i.t\right)$，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$\langle n_p, n_q \rangle$，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

$$tra_{v_i}^{s_j} = n_q^i.t - n_p^i.t \text{ 。}$$

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（针对不同交通情景和需求也可以选择求平均值等其它方法获得）。

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。

此外，路段旅行时间一般要求在特定时间区间内通过该路段的车辆数大于3辆。这种限制能较好的避免实测数据过少情况下单车极值数据对旅行时间计算结果可能带来的偏差影响。

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结果集，即求$TRA = \{TRA^{s_i} \mid 1 \le i \le n, s_i \in S\}$，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$tra_{v_i}^{s_j}$，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$tra_{\delta_j}^{s_i}$，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$TRA^{s_i} = \{tra_{\delta_j}^{s_i} \mid 1 \le j \le m, \delta_j \in \delta\}$，其中$m$为待测时间范围内按照时间周期划分得到的时间区间数。

2.2 问题分析

如上文所述，如何在海量车牌识别数据集（亿级数据记录）基础上计算城市路段旅行时间成为一个亟待解决的关键问题。

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m个时间区间,如图1横坐标δ_1至δ_m，其中的点l_{ix}为车牌识别数据）进行划分，对同一划分中的数据（假设每个划分中数据为n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$O(m \cdot n^2)$。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，m为105120，n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。

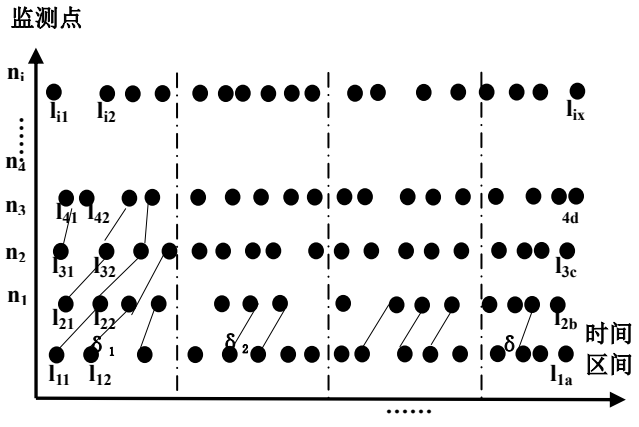

图1 路段旅行时间计算示意图

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

(1) 高效的缓存数据结构。旅行时间计算要求读入海量的历史车牌识别数据值，同时在计算过程中会产生大量中间结果，并对中间结果频繁的进行读取和写入。为此，要求能够建立高效的缓存数据结构以支持高速、连续数据处理中的高并发读写需求。

(2) 可并行化的计算模型。应对海量数据处理的一个自然途径就是利用分布式环境并通过计算的并行化来提升处理性能。因此，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，成为提升路段旅行时间计算性能的关键。

(3) 可灵活伸缩的计算架构。由于不同城市历史数据数据量、受测路网规模等都不同，要求旅行时间计算能够根据不同的负载和运行支撑环境规模进行适应性的灵活伸缩。同时，这也对上面提到的缓存数据结构提出了可划分性的要求，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。

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

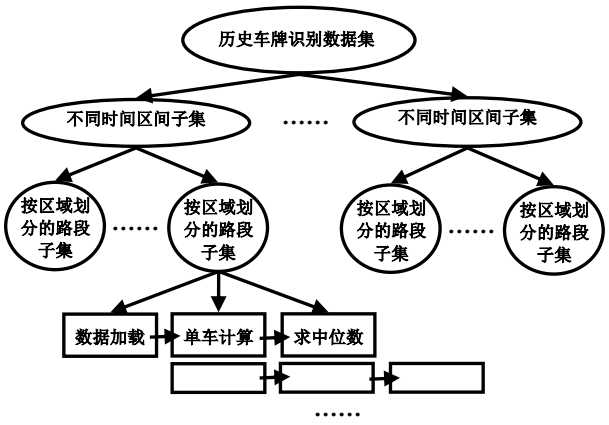

图2 基于时空划分的流水线式并行计算模型

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。

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n组时间维上区分的数据，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；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。

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[4]，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逻辑的任务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，基于我们之前完成的实时MapReduce工作[5][6]来支持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

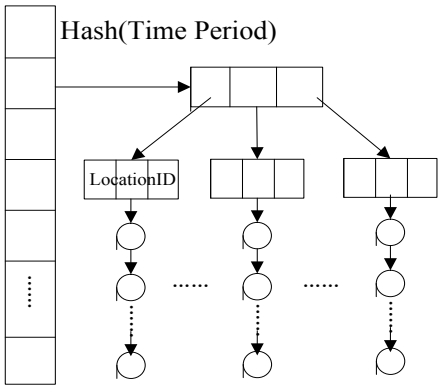

图3 车牌识别数据缓存结构

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 树来进行缓存。如图 3 所示，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。

基于上述数据缓存结构，由于对 Hash 表的任意划分也是对 Hash B 树的划分，所以该结构具有可划分性，可实现原始车牌识别数据的并行划分处理。同时，为保证并行处理节点增删时车牌识别数据的划分可灵活调整，我们按照分布式哈希算法[7]的方式来对不同时间区间的车牌识别数据进行划分。

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。

（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最终结果的示例如<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360秒。

如图4所示，在每次Map和Reduce处理过程中，我们利用两个独立的线程池来减少初始化开销，同时采用异步的数据传递方式来避免Map和Reduce阶段间的同步。其中，流水线处理模块由数据缓冲区、工作线程池和阶段控制器三部分组成，而阶段之间则通过阶段间控制器来进行调度。具体的阶段化流水线调度机制可以参考我们之前的工作[5][6]。

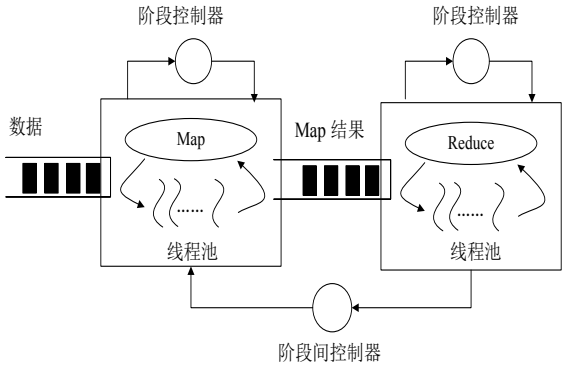

图4 Map和Reduce阶段流水线

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机上搭建的集群环境，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：该实验主要用来考察在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小时三个时间周期的路段旅行时间计算性能情况。同时，还选取直接基于Hadoop实现的旅行时间计算方法(LMR)[8]作为比较对象，与本文基于时空划分的流水线式并行计算方法（CMR）进行性能比较。

实验2：该实验用来考察受测路网中的路段数目对本文提出的路段旅行时间计算方法性能的影响。实验中，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路段旅行时间计算的性能情况。

实验3：该实验用来考察本文提出的路段旅行时间计算方法的扩展性。实验中选用20天的车牌识别数据（约1亿条）作为原始计算数据集，并在路段数目固定为210不变的情况下从1到5增加集群节点的数目，以考察1小时、15分钟和5分钟三个周期下本文所述路段旅行时间计算方法在节点数增加时计算性能的变化情况。

4.2 实验结果分析

（1）计算性能对比分析

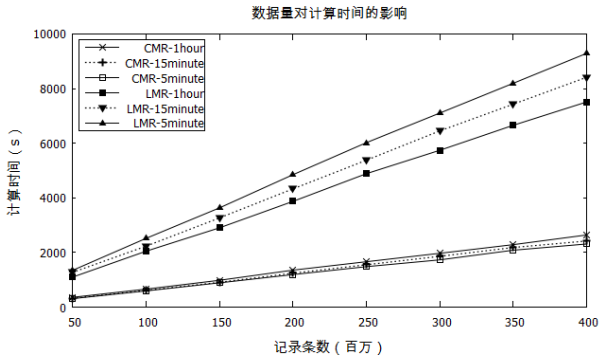

图5 车牌识别数据量对旅行时间计算性能的影响

通过实验1得到如图5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。

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）关键参数影响对比分析

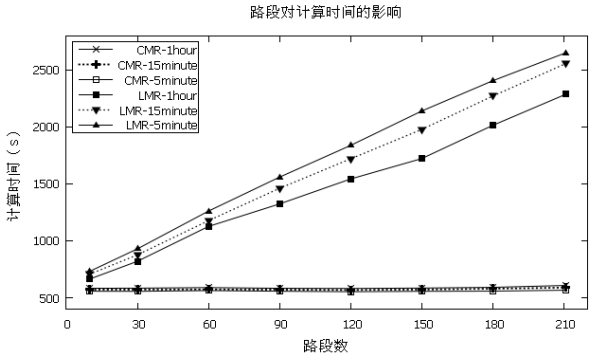

图6 路段数对旅行时间计算的影响

由实验2得到如图6所示的结果。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。

（3）扩展性分析

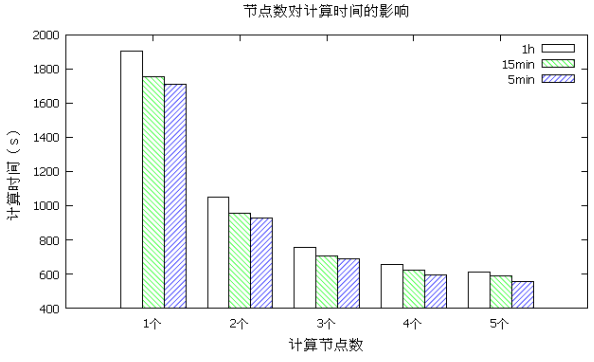

图7 计算节点数对旅行时间计算的影响

通过实验3得到如图7所示结果。从实验结果可看出，

随着计算节点数的增加，本文CMR计算方法的计算时间会逐步降低，并且可以看出细粒度时间周期的计算时间更少。这表明CMR计算方法的实现并未影响原Hadoop架构的扩展性。

通过上述实验分析可以看出，本文基于时空划分的流水线式并行计算方法能很好的解决海量车牌识别数据集上的路段旅行时间计算问题，并且计算性能受路段数目变化的影响较小。相对于直接基于Hadoop的计算方法，本文方法也表现出了更高的性能和效果。

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[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。我们之前的工作[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。

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]改进MapReduce，设计了Hadoop Online Prototype(HOP)实现基于流水化和缓存技术的数据流处理系统，也能响应连续的查询请求。为了提升MapReduce对迭代执行类的程序的支持能力，[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]是近期对于迭代式MapReduce优化的一个开源计算框架，其采用基于内存的思路来提升迭代式MapReduce的计算效率，其在设计思路上与我们的实时MapReduce工作非常相似，但是本文工作的特点是在迭代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，特别适于实现基于时空划分的并行化流水线式计算。

综上，现有的相关工作大都集中在通用的计算平台，在领域相关数据划分及执行优化方面并未进行特别设计，因此不能直接用来解决本文的旅行时间计算问题。相对于上述工作，本文则是基于车牌识别数据时空特性的利用并参考上述围绕MadpReduce模型及实现的相关工作，重点通过数据的时空划分和计算任务的流水线设计对旅行时间计算模型进行优化，以提高旅行时间计算性能。

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扩展性等特点。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]. 北京交通大学,2008.
Zhu Ai. Research on Link Travel Time Prediction Method Based on Data Collected by Floating Car[D]. Beijing Jiaotong University,2008.(in Chinese)

[2] 姜桂艳,常安德,牛世峰. 基于车牌识别数据的交通拥堵识别方法. 哈尔滨工业大学学报, 2011,43(4):131-135.
Jang Jiayan,Chang Ande,Niu Shifeng. Traffic Congestion Identification Method Based on License Plate Recognition Data[J]. Journal of Harbin Institute of Technology, 2011,43(4):131-135.(in Chinese)

[3] 柴华骏,李瑞敏,郭敏. 基于车牌识别数据的城市道路旅行时间分布规律及估计方法研究[J]. 交通运输系统工程与信息,2012,12(6):41-47.
Cai Huajun, Li Ruimin, Guo Min. Travel Time Distribution and Estimation of Urban Traffic Using Vehicle Identification Data[J]. Transportation Systems Engineering and Information,2012,12(6):41-47(in Chinese)

[4] 朱定局. 并行时空模型[M]. 科学出版社, 2009 年 10 月.
Zhu Dingjun. Parallel space-time model[M]. Science Press, 2009.(in Chinese)

[5] 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)

[6] 亓开元, 韩燕波, 赵卓峰, 房俊. 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. MapReduce Intermediate Result Cache for Concurrent Data Stream Processing[J]. Journal of Computer Research and Development, 2013, 50(1):111-121.(in Chinese)

[7] G. DeCandia, D. Hastorun, et al. Dynamo: Amazon's highly available key-value store[C]. In Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP'07), Stevenson, Washington, USA, 2007, pp. 205-220.

[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[C]. 第十一届全国 Web 信息系统及其应用学术会议, 南开大学, 天津, 中国,2014.
Zhang Shuai,Zhao Zhuofeng,Ding Weilong. Urban Road Trip Time Measured Calculation Based on MapReduce[C]. The 11th Web Information System and Application Conference, at NANKAI UNIVERSITY, TianJin, China, 2014(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, 2013,5:82-86.
Liao Fei,Huang Sheng,Gong Dejun. Distributed Storage and Data Mining Analysis of Urban Road Traffic Based on Hadoop[J]. Highways & Automotive Applications, 2013,5:82-86(in Chinese)

[10] Lizhe Wang, Jie Tao, Rajiv Ranjan, and 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 and et al. The HaLoop Approach to Large-scale Iterative Data Analysis[J]. VLDB Journal, 2012, 21(2): 169-190.

[12] Tyson Condie,Neil Conway,Peter Alvaro, and et al. MapReduce Online[C]. Technical Report, University of California, Berkeley, 2009, available at http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-136.html.

[13] Yanfeng Zhang, Qinxin Gao, Lixin Gao, Cuirong Wang. 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. University of California, Berkeley, Technical Report No. UCB/EECS-2014-12, February, 2014.

作者简介

赵卓峰 男。1977 年 5 月出生于山东莱芜。2005 年获得中国科学院计算技术研究所工学博士学位。现为北方工业大学副研究员。主要研究方向为云计算、流数据处理、服务计算、智能交通。E-mail：edzhao@ncut.edu.cn。

张 帅 男。1989 年 12 月出生于山东聊城。2012 年获得山东科技大学理学学士学位，现为北方工业大学硕士研究生，研究方向为数据挖掘，智能交通。E-mail：zhsh1222@126.com。

丁维龙 男。1983 年出生于山东泰安。2013 年获得中国科学院计算技术研究所工学博士学位。现为北方工业大学助理研究员。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。