

# 基于HBase+ ElasticSearch的海量交通数据实时存取方案设计

董长青,任女尔,张庆余,田玉靖

北京卡达克数据技术中心软件业务本部,天津 300300

## 摘要

交通数据具有数据海量、存储和交互速率快等特征,因此其数据的采集、存储及检索成为了车辆远程监控平台中的关键问题。采用LVS集群技术进行数据采集负载均衡,队列缓存处理I/O时延,HBase进行分布式数据存储;针对Hadoop实时在线数据处理不足的问题,整合ElasticSearch并构建了分层索引。通过关键技术的设计和实现,车辆监控由400辆扩展到上万辆,PB级数据在线查询速度提升了10~20倍,验证了方案的高效性。

## 关键词

Hadoop/HBase ;ElasticSearch ;Linux虚拟服务器 ;海量数据 ;实时

中图分类号 :TP311

文献标识码 :A

doi: 10.11959/j.issn.2096-0271.2017010

## *Design scheme of massive traffic data real-time access based on HBase and ElasticSearch*

**DONG Changqing, REN Nver, ZHANG Qingyu, TIAN Yujing**

Software Business Department, Beijing CATARC Data & Technology Center, Tianjin 300300, China

## *Abstract*

Traffic data has the characteristics of massive and real-time, and its massive data acquisition, storage and retrieval has become a key issue in the vehicle remote monitoring platform. According to the study of these problems, the cluster technology of LVS was used to solve the data acquisition load balance, the queue cache model was used to solve I/O delay, and HBase distributed data storage scheme was used to solve the massive data storage. HBase integration ElasticSearch, which was aimed to solve the real-time online data processing problems of Hadoop, was designed to build a hierarchical index. Through the design and implementation of the key technologies, the number of vehicle monitoring had been promoted from 400 to 1 million, online query speed increased about 10 to 20 times based on PB level data. The results verified the efficiency of the scheme.

## *Key words*

Hadoop/HBase, ElasticSearch, Linux virtual server, massive data, real-time

## 1 引言

在车联网飞速发展的时代,智能交通在发展中的一些问题也逐步暴露出来。如在数据采集中,常用的通信技术只能同时监控几百辆车;在数据结构单一、海量的数据存取和分析时,传统关系型数据库应用能力大幅度降低等。如果车辆每2 s上传一条约120 byte的全球定位系统(global positioning system, GPS)数据,统计600辆车的数,则月统计数据为87 GB,一年将约有1 TB的数据需要进行存储。在统计过程中,高峰期车辆数量远不止这些。在很多企业的实际环境中,基于传统技术的监控平台进行系统优化处理后,其数据处理能力也只能达到几百辆车的收集功能,一旦进入峰值,系统几乎无法正常运行。此外,车联网数据不仅包括GPS数据,还包括车辆控制器局域网络(controller area network, CAN)数据、图像、视频等数据,在采集和检索技术上往往与其实际生产需求相差较远。

大数据技术的快速发展为这一情况带来了良好的发展前景和机遇,尤其是当前被广泛探索应用的Hadoop、Spark等技术,针对TB甚至PB级的数据,其丰富的组件在数据存储和挖掘、分析方面提出了完善的处理方案。本文通过分析当前的分布式软件技术,结合HBase数据库进行设计改进,设计了一套针对交通数据采集、存储和检索的方案。

## 2 技术介绍

在车联网远程监控平台中,数据的采集、存储及检索问题是传统的关系型数据

平台无法高效处理的。

### 2.1 Linux虚拟服务器技术

在数据采集方面,交通数据到监控平台之间一般采用3G/4G等技术,通过socket进行TCP/IP数据传输。但是每个端口负载能力有限,最高能够支持400辆车同时上传数据,高于400辆车同时传输时,数据时延甚至丢失的现象较为明显。本研究使用Linux虚拟服务器(Linux virtual server, LVS)技术进行负载均衡,从而提供给用户访问一台超高性能服务器的效果。LVS技术支持通过配置多个高速局域网服务器来进行任务分配,其对外的接口则为固定的IP地址和端口,从而提供任务分发转移的负载调度功能。此外,LVS技术支持热处理,在正常工作的情况下增加或者删除节点。LVS的负载调度器(director)可以设置3种工作模式:一是地址转换,负载调度器通过算法将内网地址进行映射,外网数据分组通过映射的地址进行分发;二是通过IP隧道,负载调度器进行调度请求,通过IP隧道客户端数据分组进行封装发给服务器,服务器直接响应客户端;三是直接路由,适用于集群内的服务器都在一个网段,数据分组直接由负载调度器发给实际服务器,该种方式速度快,且开销较少。LVS调度算法主要有等比例轮转、加权轮循、目标地址散列调度、源地址散列调度等静态调度算法以及最少连接、加权最少连接、永不排队、动态目标地址散列、带复制的动态目标地址散列等动态调度算法。IP虚拟服务器(IP virtual server, IPVS)技术是一种工作在网络模型第四层的高效交换机,可以针对不同的网络选择不同的调度算法。

在小型的分布式负载中,唯一从软件技术上能达到硬件F5量级的方案就是采用LVS技术。参考文献[1]使用LVS技术解

决负载均衡网站的负载均衡配置问题,参考文献[2]描述了LVS技术的高可用负载均衡方案,但其后续处理案例也主要针对超文本传输协议(hypertext transfer protocol, HTTP)。本文基于LVS的IP虚拟分发技术进行socket通信集群负载。

## 2.2 HBase存储技术

数据采集接收伴随着海量数据的存储问题,性能较好的传统关系型数据库一般采用Oracle、DB2等。但在实际应用中,交通流数据结构单一,结构内部却比较灵活,当有些字段在数据存储中只为部分数据设计时,其存储空间将造成极大的浪费,再加上数据量巨大,交通流数据显然不适合传统的存储方式。随着分布式数据库的发展,HBase作为列式存储数据库,为交通流的数据存储提供了高效的写入性能和灵活的存储方式。在其存储结构上,HBase按照字典顺序进行排序,行键(RowKey)直接作为其中的一级索引,为其良好的读写性能提供了基础。

## 2.3 HBase检索

HBase内置了两张存储表(ROOT

和META)进行区域分布及区域详细情况的存储。在普通的数据检索过程中,首先从ZooKeeper上定位到META表的位置,然后从中获取对应的RegionServer,根据行键从这两张内置表中查询数据的region,然后查找到对应的行键,从而进行数据检索。但是HBase的存储业务表之间没有直接的关联,而且单一索引很可能造成数据查询变成全表扫描,因此当查询海量数据时,对关联数据或复杂条件的支持较差。目前,针对这一问题,已经发展了ITHBase(带索引的事务性HBase)、IHBase(indexed HBase,是HBase的扩展,用于支持更快的扫描)等二级索引方式。ElasticSearch是一种基于Lucene的实时大数据搜索引擎,与适用于独立应用的Solr相比,更适用于云计算环境。参考文献[3]中采用对交通流数据进行存储,但其存储行键设计在实时查询性能上经实验检测存在较大的问题,参考文献[4]和参考文献[5]引入了大数据的实时处理办法进行日志的处理,本文通过进一步研究和改进,引进ElasticSearch进行数据索引,设计其将必要条件与其行键关联起来,形成二级索引<sup>[6]</sup>,从而加快数据搜索速度。

## 3 海量交通数据的存取架构设计

海量交通数据的存取架构是各个企业目前亟待解决的问题,并发采集数据、海量数据存储及海量数据的检索等问题,成为企业车联网进程中的阻力。

### 3.1 系统架构

整体架构主要从采集、存储和检索3个方面进行设计改进并实现。整体架构如图1所示。

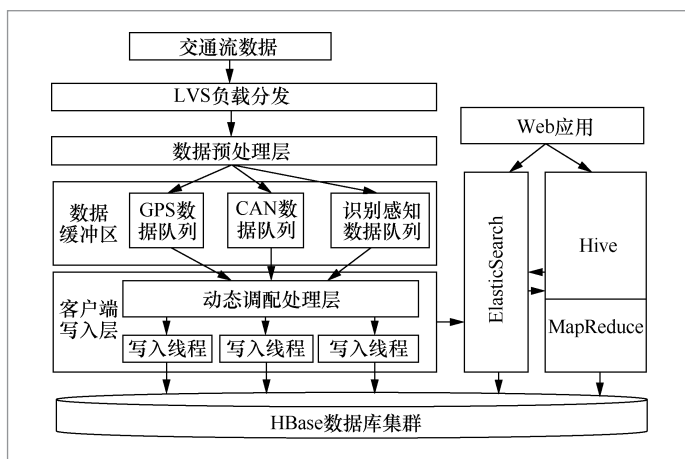


图1 海量交通流数据分布式存取架构

当数据从交通流中获取时,由LVS进行IP转发负载均衡,实现车辆高并发数据的接收工作;接收服务器收到数据以后,对数据进行预处理操作,分别按照业务类型转入缓冲区,通过HBase客户端进行数据写入操作,同时将数据索引<sup>[7]</sup>建立到ElasticSearch中,根据数据的状况动态调配处理层进行数据线程数的设置,保证高并发写入速度;在Web应用中读取数据时,根据业务需求的不同设置不同的线程数量。

## 3.2 关键设计

### 3.2.1 LVS负载均衡

结合实际生产环境,研究采用速度快、开销少的直接路由的工作模式以及IPVS技术进行算法调度<sup>[8]</sup>,配置4台服务器分别作为LVS以及3台数据处理设备,操作系统采用Centos 6。

如图2所示,配置3台数据处理服务器,然后配置对应的LVS,在数据由车载终

端发送到路由时,由路由直接转发到LVS,LVS通过调度算法计算将数据发送到合适的数据处理服务器,数据处理服务器接收数据以后,直接通过路由响应终端。主要实现步骤如下。

- 配置数据处理服务器,禁止地址解析协议(address resolution protocol, ARP)请求。

- 设置LVS子网掩码与数据处理服务器一致,开启报文转发功能,增加网卡VIP记录“ipvsadm -A -t 10.8.10.177:20080 -s rr -p 2000”,含义为各个车载访问地址及其端口号,设定2 s为超时时间,调度为等比例轮询调度(round robin, RR)算法。

- LVS服务配置数据处理服务器,按照“ipvsadm -a -t 10.8.10.177:20080 -r 10.8.10.162:20080 -g”指令分别配置3台数据处理服务器的访问地址端口及直接路由模式。

此时配置完毕,该方案具有较强的可扩展性,一方面可以根据监控车辆的负载

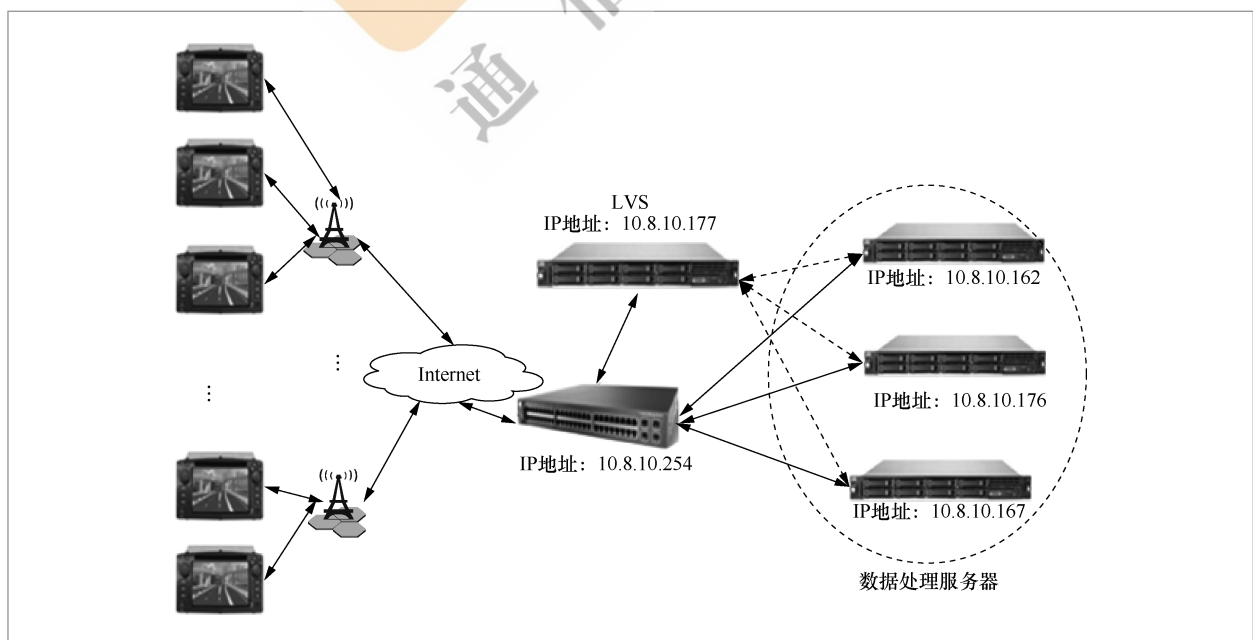


图2 系统LVS负载均衡架构



情况增加或者减少数据处理服务器的数据,另一方面可以根据缓冲和写入线程的情况保证数据写入时延在秒级以内。此外,可以参考参考文献[2]进行LVS备份方案的设定。

### 3.2.2 多源存储设计

交通流数据从LVS分发到各个数据处理服务器以后,由接收端应用进行预处理,按源分类进行缓冲存储操作,通过监控程序根据各个数据源中的数据大小判定启动不同数量的写入线程,默认写入线程为3个,当数据量过大时将增加写入线程。写入线程主要负责写入HBase数据和构建ElasticSearch索引工作。

该架构设计主要分为GPS数据、CAN数据和多媒体感知数据,然后根据每种数据的结构不同分别存储到不同的缓冲区中。按每秒接收10万条GPS和CAN数据计算,3台数据处理服务器每台每秒需接收 $(100\ 000 \times 12\ \text{kB}/3) / (1\ 024 \times 1\ 024) \approx 0.39\ \text{MB}$ 数据,GPS和CAN数据缓冲区大小预设4 MB,多媒体数据缓冲区大小设置为16 MB,即每台数据处理服务器根据数据源由20 MB内存作为数据缓冲区,从而允许将一定数量的数据高效地进行存储。对于分组数据,采用临时缓冲区先行进行缓冲,组合成整个分组后再进入缓冲区,若数据大小大于缓冲剩余空间大小,则直接启动额外的写入线程进行存储。各个文件结构设计如图3所示。

以GPS数据为例,其在缓冲结构中

存储,采用时间值进行散列计算后存储到固定个数的队列中,如2016年1月13日13:42:57,设置 $t$ 为20 161 452 663 777 771,进行求余计算,计算式为 $f(t)=t\%n$ 。设置 $n$ 为3,即GPS数据源采用3组队列,队列索引分别为0、1、2,则 $f(t) = 20\ 161\ 452\ 663\ 777\ 771\%3$ 求余结果为1,因此将该数据放置在索引为1的队列。

缓冲数据构建完毕以后,由写入层以“先进后出”的原则读取数据进行持久化存储。在HBase数据存储中,数据基本上按时间顺序存储,直接存储会造成写入热点问题,即多个线程均指向一个HBase集群节点写入的情况,因此采取以下优化策略防止热点和全表扫描。

- 在行键前加入散列前缀,利用如下计算方式计算:  $\text{byte pre}=(\text{byte}) (\text{Long.hashCode}(t) \% < \text{regionservers no.}>)$ ,将通过前缀加上时间戳散列值的方式产生的不同的数据分发到不同的RegionServer上。

- 取消自动写入,根据实验设置写入缓冲区大小为20 MB,能提高千万级数据同时插入的效率。

- 预分配region,建表时直接使用预分配region,避免单个region灌入数据。

通过以上措施,达到了良好的实现效果,加入索引提高查询的性能,一定会降低写入数据的性能,本文兼容性地考虑两方面,并进行了综合实现。

### 3.2.3 索引方案改进

HBase本身主键构建了B+树进行索

GPS	数据ID	接收时间	监测时间	经纬度	海拔	速度	方向	车牌号	
CAN	数据ID	接收时间	监测时间	电池充/放电电流	故障代码	单体电池电压	温度	...	车牌号
Media	数据ID	摄像头NO	车辆颜色	车牌号	车速	监测时间	地点	部门	文件流

图3 数据结构

引<sup>[9]</sup>,称为一级索引,其对基于主键数据的查询效率很高;然而对于非主键字段的查询效率却很低,对HBase大数据量的访问,仅仅通过MapReduce和扫描器处理是不能达到令人满意的效果的。其主要缺陷在于二级索引构建困难,重新构建表结构进行索引往往需要双重查询,而且难以维护索引数据与原数据的同步性。随着ITHBase、IHBase以及华为技术有限公司(以下简称“华为”)的hindex项目的诞生,二级索引方案和效果不断提升,本文基于华为二级索引方案进行改进,结合ElasticSearch通过多层索引和直接索引由业务引擎共享的方案来实现高性能索引改进方案。

#### (1) 基于HBase表的二级索引方案设计

基于HBase表的二级索引方案设计主要采用索引表和数据表共存共享的方式构建,通过HBase的协处理器(coprocessor)构建与数据表相同和类似的索引,索引表行键设计为“数据表StartKey+IndexName+Value+数据表行键”的方式。通过使索引表和数据表拥有相同的StartKey并重写均衡集群类(balance cluster)控制索引表的分

配,使其索引表和数据表构建在相同的RegionServer上,并且在region分裂时也能同步进行分裂,这样可以使得协处理器非常快速地在RegionServer上计算出相应的索引数据。IndexName和Value对应的为HBase数据中单列的值,如针对车辆监测点进行的数据查询,Value对应为监测点的数值,从而通过索引查询出某个检测点所有的行键数据,进一步查询数据。

如图4所示,实线代表数据指向,虚线代表数据块指向,索引表数据对数据表数据的行键进行进一步改造,增加索引名称和索引数据。当数据进行分裂时,其对应的索引同步进行分裂,并且使用保持数据的StartKey起始一致。

#### (2) ElasticSearch构建

通过索引表的方式构建二级索引以后,数据的查询依然停留在依靠HBase数据表本身的能力去优化查询速度。本文在构建索引数据表基础上同步构建ElasticSearch及缓存索引数据。如图5所示,当负责写入的线程进行写入操作时,通过协处理器同步处理索引表数据,然后通过观察者模式同步索引数据到ElasticSearch中,并且根据多源数据特

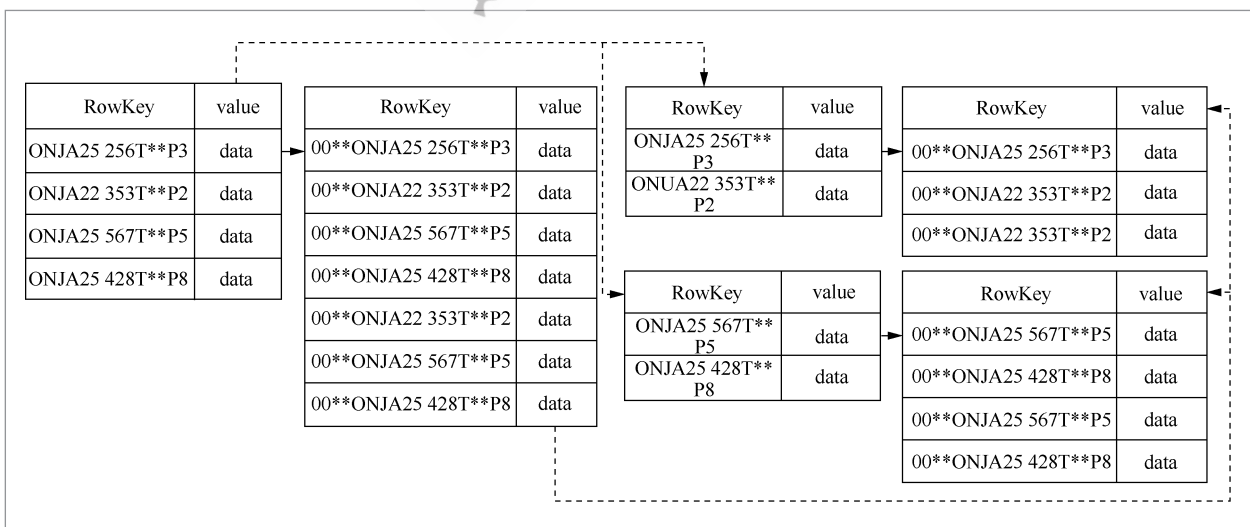


图4 数据分区分裂表

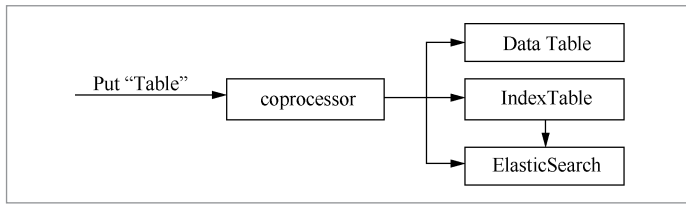


图5 数据写入过程

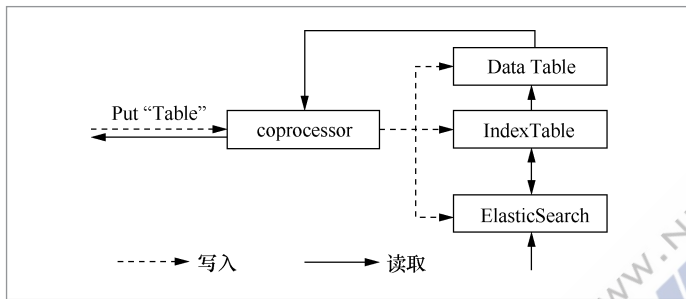


图6 数据读取过程

性，将实时查询的数据添加到内存索引缓冲区。

首先，开启协处理器，通过HBase Shell激活协处理器的观察者(observer)；通过继承基类BaseRegionObserver，重写postPut和postDelete方法。把生成的JAR包配置到写处理器中，即可实现数据的同步。在实际数据操作过程中，交通流数据几乎不会发生更改，但是会持续写入，因此在由观察者数据同步时采用了ElasticSearch的缓冲池批处理操作，当达到限值时进行同步写入操作。此外，设置其分片值、缓存类型为软引用(soft reference)，并调整其最大缓存值等进行ElasticSearch调优。

数据读取过程如图6所示，当发起数据读取过程时，首先进行查询。当数据读取时，首先访问ElasticSearch，根据查找到的索引表中的结果，调用协处理器进行数据实际行键查找，访问数据表，从而得到数据，从协处理器返回给客户端。通过该模式进行改进，数据查询效率大幅度增加。

## 4 系统测试

本文主要从并发存储的吞吐量、轨迹回放查询速度进行了测试，从而验证其并发写入和实时读取的性能。

### 4.1 并发存储吞吐量测试

为了充分验证本架构的可行性，分别测试1万到1 000万条数据的插入速度，并且从单机服务、不添加索引和添加索引3方面进行对比测试。图7为测试结果。

其中，添加索引的情况比不添加索引的情况速度明显降低，但插入数据完毕时间总体维持在秒级以内。此外，对数据进行了持续性测试，连续10 h以每秒13万条数据的速度插入(相当于监控了10万辆车每秒上传一次数据)，其插入数据缓冲区平稳保持在某个较低临界值后不再变动。在数据进行索引插入时，能够支持每秒7.84万条数据平稳运行，较不加索引时有一定降低，但整体插入速度能够在秒级以内实现10万级速度的插入。

### 4.2 轨迹回放查询速度测试

轨迹回放主要从整体统计查询和实时响应两方面性能进行体现，图8测试了在PB级数量基数水平上，结果集数量不同时数据查询效率。

当小结果集进行查询时，不添加索引时响应速度在10 s以上，而进行索引时，数据查询速度在1 s以内，速度提升了20倍左右。大结果集(万级)进行查询时，速度提升了9~10倍，实时查询效率及速度大幅度提升。在实际应用中，20 min的轨迹回放约为600条数据，能够实现5 s之内查询，

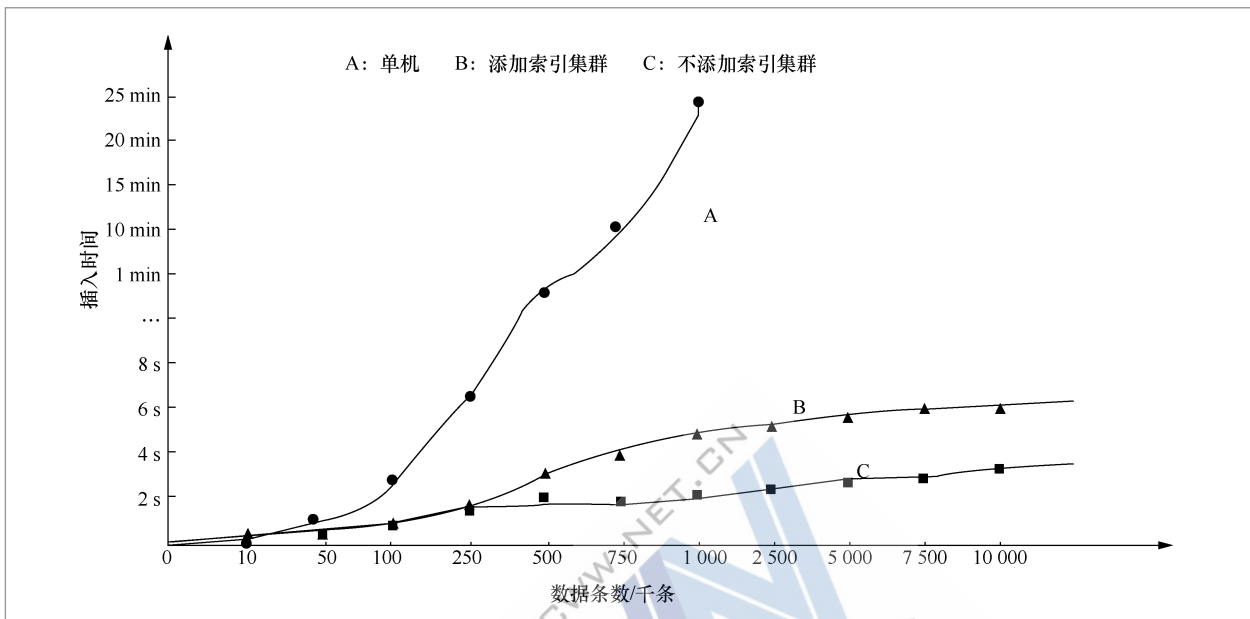


图7 单机、添加索引集群、不添加索引集群数据

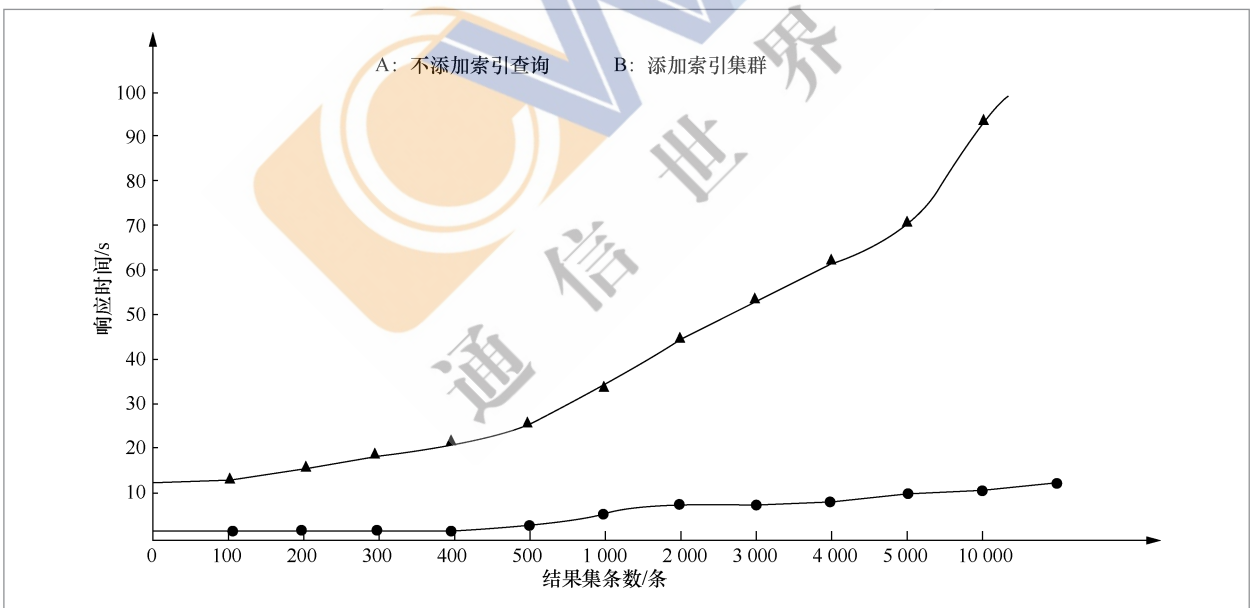


图8 不同结果集数据查询效率

因为数据响应可分段，所以如果以5 min为时间段进行4次查询，能够达到页面较为流畅的效果。

此外，在研究传统数据库（如Oracle）在数据存取过程中的表现时，采用了按照月进行分区、创建复杂查询条件索引、添

加存储过程、避免全表扫描操作（如执行“like”语句）、建立缓存等设计。研究发现索引过多则写入性能下降，而且对非结构化数据存储支持性能不佳，在整体大数据操作编程和配置的复杂度提高。研究测试了直接对GPS数据表进行插入和读取操作



的速度情况:在进行数据插入操作时(对比每分钟插入数据量),百万条以下的数据插入速度Oracle和HBase没有明显差异,但超过百万条以后,Oracle的数据插入速度逐步下降,千万条以上HBase数据插入速度比Oracle快2~7倍;在进行数据读取操作时,HBase千万条以上数据读取速度是Oracle的5~15倍;在同时进行插入和读取操作的过程时,HBase读取速度比Oracle快15~30倍。在相同的硬件和网络环境中分析对海量交通流数据的处理能力,HBase列式数据库无论从插入性能还是读取性能都可以调优到更高水平。

## 5 结束语

LVS解决高并发接收数据的问题,通过多源缓存策略解决数据存储不及时的问题,避免数据分组丢失;同时设计表级二级索引、引入ElasticSearch增加数据查询速度。总体上本文通过设计高并发存储架构和多层索引查询架构,实现了交通流数据的高并发实时监控数据存储和查询,从软件架构上解决了基于Hadoop存储数据对实时计算查询支持度不够的问题。下一步,将集中设计基于多层的热点内存缓存方案,并设计响应的缓存命中策略来实现更高的实时性能。

## 参考文献:

- [1] 王颀帅. 基于LVS的服务器负载均衡技术[J]. 计算机系统应用, 2014, 23(7): 252-255.  
WANG Y S. Server load balancing architecture based on LVS[J]. Computer Systems & Applications, 2014, 23(7): 252-255.
- [2] 刘敏娜, 张继涛. 基于LVS+KEEPALIVED

的高可用负载均衡研究与应用[J]. 自动化技术与应用, 2014, 33(11): 22-27.

LIU M N, ZHANG J T. The study and application of based on the LVS + KEEPALIVED high availability load balance[J]. Techniques of Automation and Applications, 2014, 33(11): 22-27.

- [3] 陆婷, 房俊, 乔彦克. 基于HBase的交通流数据实时存储系统[J]. 计算机应用, 2015, 35(1): 103-107, 135.

LU T, FANG J, QIAO Y K. HBase-based real-time storage system for traffic stream data[J]. Journal of Computer Applications, 2015, 35(1): 103-107, 135.

- [4] 葛微, 罗圣美, 周文辉. HiBase:一种基于分层式索引的高效HBase查询技术与系统[J]. 计算机学报, 2015, 38(35): 1-15.

GE W, LUO S M, ZHOU W H. HiBase:a hierarchical indexing mechanism and system for efficient HBase query[J]. Chinese Journal of Computers, 2015, 38(35): 1-15.

- [5] 白俊, 郭贺彬. 基于ElasticSearch的大日志实时搜索的软件集成方案研究[J]. 吉林师范大学学报(自然科学版), 2014(1): 85-87.

BAI J, GUO H B. The design of software integration for big log data real time search based on ElasticSearch[J]. Jilin Normal University Journal(Natural Science Edition), 2014(1): 85-87.

- [6] 钟雨, 黄向东, 刘丹, 等. 大规模装备监测数据的NoSQL存储方案[J]. 计算机集成制造系统, 2013, 19(12): 3008-3016.

ZHONG Y, HUANG X D, LIU D, et al. NoSQL storage solution for massive equipment monitoring data management[J]. Computer Integrated Manufacturing Systems, 2013, 19(12): 3008-3016.

- [7] SFAKIANAKIS G, PATLAKAS I, NTARMOS N, et al. Interval indexing and querying on key-value cloud stores[C]// The 29th IEEE International Conference on Data Engineering (ICDE), April 8-12, 2013, Brisbane, Australia. New Jersey: IEEE Press, 2013: 805-816.

- [8] 苏命峰, 陈文芳, 李仁发. LVS 集群负载调

- 度算法研究[J]. 长沙大学学报, 2012(5): 72-74.
- SU M F, CHEN W F, LI R F. Research on LVS cluster load scheduling algorithm[J]. Journal of Changsha University, 2012(5): 72-74.
- [9] BAI J. Feasibility analysis of big log

data real time search based on HBase and ElasticSearch[C]// 2013 Ninth International Conference on Natural Computation (ICNC), January 28-31, 2013, San Diego, USA. New Jersey: IEEE Press, 2013: 1166-1170.

## 作者简介



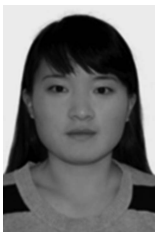
董长青(1980-),男,北京卡达克数据技术中心软件业务本部高级工程师,主要研究方向为大数据、车联网。



任女尔(1990-),女,北京卡达克数据技术中心软件业务本部助理工程师,主要研究方向为大数据、云计算。



张庆余(1991-),男,北京卡达克数据技术中心软件业务本部助理工程师,主要研究方向为软件架构、云计算。



田玉靖(1987-),女,北京卡达克数据技术中心软件业务本部中级工程师,主要研究方向为软件架构、编程模式。

收稿日期: 2016-09-20