

基于事件密集度的交通监控视频存储方法

臧继昆 喻 剑

(同济大学计算机科学与技术系 上海 201804)

(同济大学嵌入式系统与服务计算教育部重点实验室 上海 200092)

摘要 利用 HDFS 进行大规模交通监控视频的存储和处理是一种可靠、高效、可扩展的数据存储方案。针对 HDFS 默认的机架感知策略可能造成存储热点这一问题,提出了一种基于事件密集度的交通监控视频放置策略。该策略利用交通视频可按事件类型进行分类这一特征,在数据放置时将数据节点中已存储的各类型的事件视频可能对其造成的负载作为节点的主要评价因素之一,同时结合节点的实时负载、磁盘容量等因素进行综合评价,选择最佳的数据放置节点,从而平衡数据节点的负载。实验表明,基于事件密集度的交通监控视频放置策略可以改善数据节点的吞吐量,提高存储系统性能。

关键词 交通监控视频, HDFS, 交通事件, 数据放置, 吞吐量

中图分类号 TP399 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.5.044

Traffic Surveillance Video Storage in HDFS Based on Event Density

ZANG Ji-kun YU Jian

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

(The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China)

Abstract Utilizing HDFS to store and process large scale traffic surveillance video is a reliable, highly efficient and scalable solution. However, the default rack awareness data placement strategy in HDFS may cause hotspots when placing data. To address this problem, this paper presented a traffic surveillance video data placement strategy based on traffic event density. The characteristic of traffic surveillance videos allows us to classify them according to traffic event types. When placing new data, the proposed strategy predicts the load of each datanode which is influenced by various of traffic events the datanode stores, then combines the instant load and disk capacity to evaluate each datanode, and chooses the most suitable datanode to store new data. Experiments show that the proposed strategy alleviates the hotspot problem and effectively improves the load balancing and throughput in comparison with the default strategy.

Keywords Traffic surveillance video, HDFS, Traffic event, Data placement, Throughput

1 引言

智能交通系统^[1] (Intelligent Transportation System, ITS)在我国的应用越来越普及。为满足 ITS 对交通信息实时、准确、高效的需求,网络视频监控和高清视频监控的应用越来越广泛。大规模高清视频监控系统的部署对视频存储系统的性能提出了更高的要求。以用于违章事件检测等功能的 200 万像素高清摄像头为例,其最高视频码率为 4Mbps/s,那么 8 个路口(每个路口设置 4 个摄像头)的监控区域 30 天内将产生 40TB 的视频数据。系统中的存储设备要对如此大规模的数据实时地进行存储,同时快速响应用户在海量数据中检索特定视频片段(如交通事件)的请求,这对设备的读写性能提出了严峻的挑战。

现阶段网络硬盘录像机 (Network Video Recorder, NVR)和存储区域网络 (Storage Area Network, SAN)的应用虽然能够满足系统对于存储方面的需求,但是系统的部署和

管理维护成本较高,且不利于视频数据分析等应用的拓展^[2]。分布式存储系统为解决上述问题提供了一个很好的思路。Hadoop^[3,4] 分布式架构以其分布式文件系统 (Hadoop Distributed File System, HDFS) 和 MapReduce 分布式计算框架为核心。其中 HDFS 是 GFS^[5] 的开源实现,具有高容错性、高伸缩性等优点,允许用户将 Hadoop 部署在廉价的硬件上建设成本低廉、易于管理的分布式存储系统。同时, HDFS 设计能够很好地支持大文件的操作,使用了文件一次性写入、多次读取这样一个有利于提供高吞吐量的数据访问模型,并且以流的方式对数据进行访问^[6]。监控系统存储访问具有以流媒体方式对存储设备进行访问、多路视频长时间同时写入同一个存储、写后不再修改、带宽稳定性要求高等特征^[7],这些特征符合 HDFS 的设计特点;并且 Hadoop 架构可以利用 MapReduce 进行数据分析和挖掘,因此,利用 Hadoop 架构实现交通监控视频的分布式存储可以提供一个好的解决方案。分布式存储系统中数据的放置策略是一个重要的问题,

到稿日期:2014-06-23 返修日期:2014-09-17 本文受科技部国际合作专项(2012DFG11580)资助。

臧继昆(1989-),男,硕士生,主要研究方向为计算机软件与理论、分布式存储, E-mail: zangjikun@126.com;喻剑(1975-),男,博士,主要研究方向为物联网、云计算、传感器网络、RFID, E-mail: yujian@tongji.edu.cn.

需要在写入读取带宽、可靠性之间进行权衡。HDFS 中的默认数据放置策略为机架感知 (rack awareness) 策略。该策略的优点是能够快速选择数据节点, 保证系统带宽。但由于数据节点的选择是随机的, 这可能会造成过多的数据相对集中在某些节点上从而产生存储热点, 导致系统负载不均衡, 影响系统的吞吐量。虽然交通监控视频数据规模较大, 但实际应用中用户的访问请求往往集中于包含交通事件的视频片段。如果过多的事件视频数据因为随机的放置而被集中存放在某些数据节点, 则这些节点可能会由于负载较高而成为存储热点。因此 HDFS 用于交通监控视频的存储相对于其他数据应用, 采用默认的机架感知策略可能会导致更多的负载不均衡情况。

针对机架感知策略用于交通视频存储易产生存储热点问题, 本文提出了一种改进的交通监控视频存储数据放置策略。本策略充分利用了不同事件视频获得的访问不均匀这一特性, 对交通视频按照事件类型进行分类, 统计用户对不同类型事件的访问差异, 以此作为各类事件视频数据对存储节点造成的负载量度。在放置新数据前计算数据节点中已存储的各类型事件视频对节点的负载影响程度作为节点的负载预估, 同时结合数据节点的当前负载以及节点容量等因素, 在数据节点选择时从数据因素和节点本身因素两方面对节点进行评价, 选择最佳评价节点进行新数据存储, 从而达到更好的负载均衡效果, 预防了存储热点的产生, 有效提高了系统吞吐量。

2 HDFS 中的数据放置策略及相关工作

机架感知策略首先通过 Java 提供的 Random 类中的随机数生成方式, 从机架所含数据节点列表中随机选择某一节点, 然后通过两次判断 (节点的存储空间是否足够; 节点的负载是否超过集群平均负载的两倍) 来判断节点是否符合要求。但是这仅是一种粗略的判定方式, 并不能保证很好的节点间的负载均衡。如节点 A 和 B 位于同一机架中且二者都具有足够的空间来放置数据, 此时 A 比 B 的负载高, 但都没有超过设定阈值, 那么数据可能仍然会被存放在 A 节点上, 假若此时要放的数据引起的负载很高, 那么这将导致 A 与 B 负载差别更大。

针对机架感知策略的局限性以及 HDFS 的负载均衡问题, 国内外已经有很多研究。文献[8]提出了一种根据异构集群中各节点的运算能力来进行数据放置的策略, 该策略考虑到异构集群中不同节点的运算能力差异, 以此作为数据节点的评价依据。文献[9]提出了结合网络拓扑距离信息与结点数据负载信息的数据结点评价方法, 该方法通过设置平衡参数提高了 HDFS 的存储性能。文献[10]提出了一种改进的负载均衡的模型, 其综合了文件大小、文件并发访问时间、文件访问频度、结点的处理能力、带宽、结点的可用存储空间等因素进行负载均衡。文献[11]提出一种综合考虑了网络带宽、节点 I/O 性能以及节点存储空间等因素的基于灰色马尔可夫链预测模型的副本选择策略, 以此在系统可用性和负载均衡性之间寻求一个平衡。文献[12]通过分析微软 Bing 使用的集群工作日志发现不同数据的数据流行度存在很大的差异, 存放大量流行度较高的数据的机器容易成为系统瓶颈, 因此提出了 Scarlett 系统, 通过数据的流行度进行动态副本放置从而减少热点的发生, 提高了集群性能。文献[13]则提出

了一种分布式可适数据放置算法 DARE, 通过不同的数据流行度来放置策略从而提高了数据局部性, 降低了集群的调度延迟。

3 基于事件密集度的数据放置策略

3.1 策略的基本思想

在海量的交通监控视频中, 包含各类事件的视频文件是被检索和访问的主要内容, 而且不同事件视频的用户关注度存在差异。由于 HDFS 的机架感知策略并不将存储文件的内容作为放置时的参考因素, 在存储交通监控视频时, 该策略即使能够使得各节点放置的数据在量上相近, 也可能将关注度较高的事件视频集中放置在某些节点上从而导致这些节点读写密集而成为存储热点。本文提出的基于事件密集度的数据放置策略利用交通视频可按事件进行分类这一特征, 将数据节点所放置的视频数据内容作为数据放置时的主要参考因素之一, 系统在用户访问数据的时候统计用户对不同类型交通事件的关注度以及由此导致的对节点负载的影响程度。在放置新的视频数据时, 预估估计数据节点所存储的文件可能对其造成的负载, 同时结合数据节点的实时负载情况和存储利用率等因素来对节点进行评价, 选择最佳节点进行数据放置, 从而提前进行了负载均衡, 预防由于节点存放过多高关注度的视频文件导致系统负载失衡的情况发生。

3.2 参数定义

现在的交通视频监控检测技术可以实现多车道检测车辆数、车型、占有率、车速、停车时间、停车排队长度等交通信息, 还可以对视频图像场景内的交通状况进行实时智能监控; 而且可以较为准确地识别出交通拥挤、车辆逆行、停车、违章、超速等交通事件的发生^[14]。因此, 鉴于目前交通事件识别技术的限制, 本文将交通视频划分为正常情况、超速、违章、车辆逆行、停车、交通拥挤等事件类型, 智能摄像头在采集交通视频的同时检测视频中的交通事件, 在生成视频文件时赋予其事件类型。

定义 1 (事件类型集合) 根据以上对交通视频的分类, 设 E 为交通监控视频中出现的交通事件类型的集合:

$$E = \{\text{正常, 拥挤, 逆行, 停车, 违章, 超速} \dots\} \quad (1)$$

定义 2 (事件影响度) 对于定义 1 中 E 内不同类型的交通事件, 由于用户对其关注度不同, 因此含有不同事件的文件对于数据节点造成的负载程度不同。将事件视频文件对数据节点造成的相对负载影响称为事件影响度。定义事件影响度的集合为 A 。A 中的元素分别对应于 E 中相应事件的影响度, $|A| = |E|$, 对于 $a \in A, e \in E, a_e$ 表示事件类型 e 的影响度。

定义 3 (事件密集度) 设集群中的数据节点的集合为 D , 数据节点 $d \in D, F$ 为节点 d 中的文件集合, $e(f)$ 为文件 f 的事件类型, $e(f) \in E, a_{e(f)}$ 为文件 f 的事件影响度。定义数据节点 d 的事件密集度为

$$L(d) = \sum_{f \in F} a_{e(f)} \quad (2)$$

定义 4 (目标放置节点) 在本文使用的数据放置策略中, 定义被选中的存放新数据的最佳节点为目标放置节点。设目标放置节点为 g, g 满足如下条件:

- 1) $L_g \leq L_d, d \in D$, 即 g 是所有数据节点中事件密集度最小的节点。
- 2) 设节点 d 的当前时刻的负载为 N_d, N_d 用当前连接到

d 的客户端操作请求数量来表示。如果 1) 中选出的节点为多个, 那么 g 为当前负载 N_d 最小且有足够存储容量的节点。

3) 如果 2) 中同样获得多个节点, 则 d 为从这些节点中按照机架感知策略选出的节点。

事件影响度代表了包含不同类型事件的视频文件由于用户的访问量不同对其所在数据节点负载影响程度。因此, 影响度的确定方法直接关系到数据节点事件密集度的评价准确性, 也决定了改进的数据放置策略的有效性。本文使用 e 类型文件在时间间隔 T 内的访问量与 e 类型文件在系统中的文件总数之比来计算事件影响度, 且每隔时间 T 更新一次, 从而对用户关注度的变化做出快速响应, 确保事件影响度的准确性。事件影响度的确定按如下方式进行: HDFS 中维护各类型视频文件的数量和它们的访问量。每经过设定的更新时间间隔 T , 则更新事件影响度的集合 A , 对于 $a \in A, e \in E$:

$$a_e = \frac{e \text{ 类型文件在 } T \text{ 内的访问量}}{e \text{ 类型文件在系统中的文件总数}} \quad (3)$$

3.3 算法描述

假设在视频文件生成时, 交通摄像头已经检测出视频中所含数据并为其赋予了事件类型, 在此基础上, 本文提出的基于事件密集度的数据放置策略执行流程如图 1 所示。

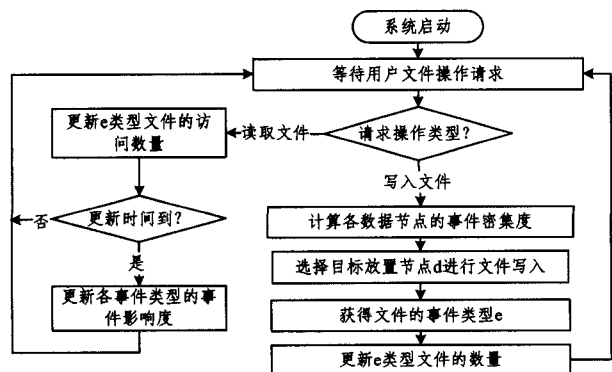


图 1 基于事件密集度的数据放置策略执行流程

描述如下:

- 1) 当系统收到客户端的文件操作请求时, 判断操作为何种类型;
- 2) 如果操作为文件写入操作, 获得系统中即时维护的各事件类型的事件影响度;
- 3) 按照定义 3 计算各个数据节点的事件密集度;
- 4) 按照定义 4 选取目标放置节点来进行新的视频文件的存储;
- 5) 获得要存储视频的事件类型 e , 更新目标节点 e 和系统中 e 类型文件的存储数量;
- 6) 如果操作为文件读取操作, 则更新 e 类型文件的访问量。系统每隔固定的时间 T , 为每种事件类型更新事件影响度。

根据上面的算法描述思想, 算法的伪代码描述如下: 其中 $tmpTargetList \langle DataNode \rangle$ 为存放具有最小事件密集度的数据节点的列表, $targetList \langle DataNode \rangle$ 为存放 $tmpTargetList$ 中具有足够容量、实时负载最小的数据节点的列表, 更新 List 的操作为尝试将数据节点插入到 List 当中, 确保 List 满足上述条件。

1. if 有新文件 f 需要存储 then
2. for each d in D do
3. 计算 $L(d)$;

4. 更新 $tmpTargetList$, 确保其包含的数据节点具有最小的 $L(d)$;
5. end for
6. if ($tmpTargetList.size > 1$) then
7. for each d in $tmpTargetList$ do
8. 获取节点剩余容量 C_d ;
9. if ($C_d > f$ 的大小)
10. 获取 N_d ;
11. 更新 $targetList$, 确保其包含的数据节点具有最小的 N_d ;
12. end if
13. end for
14. if ($targetList.size > 1$) then
15. $t =$ 从 $targetList$ 中按机架感知策略选择的节点;
16. end if
17. else
18. $t = targetList.get(0)$;
19. end else
20. end if
21. else
22. $t = tmpTargetList.get(0)$;
23. end else
24. 将 f 写入 t 中;
25. $e = getEventType(f)$;
26. 增加 t 中 e 类型文件的数量;
27. end if

事件类型影响度更新的算法伪代码如下:

1. if 用户访问文件 f then
2. $e = getEventType(f)$;
3. $increaseAccessCount(e)$;
4. end if
5. if 更新时间间隔 T 时间到 then
6. for each e in E do
7. $a_e = \frac{e \text{ 类型文件在 } T \text{ 内的访问量}}{e \text{ 类型文件在系统中的文件总数}}$;
8. end for
9. end if

4 本文策略的实现

在 Hadoop 中, 数据放置策略会在客户端向 HDFS 中写入数据的时候被调用。以 Hadoop2. 2. 0 为例, HDFS 文件写入的实现如图 2 所示。

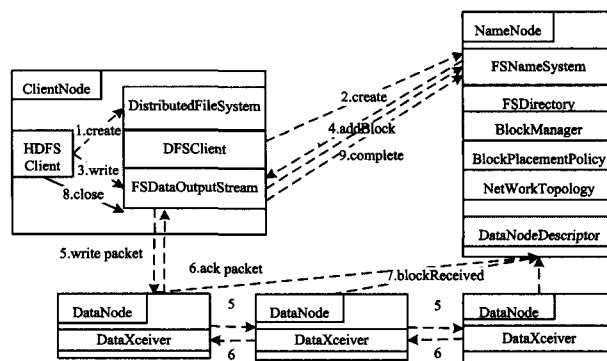


图 2 HDFS 文件写入流程及主要参与类

图 2 中每一部分标示了参与写操作的主要类。首先客户端需要通过 DistributedFileSystem 类获得 HDFS 的对象。然后调用其 create 方法创建用户要创建的文件, 这主要由 DFSCClient 的对象与 NameNode 进行通信实现。FSNameSystem 通过调用 FSDirectory 来操作文件系统的目录树, 建立新文件

的文件头 INode,并返回到此文件的输出流 FSDataOutputStream 给客户端。接着,客户端通过 FSDataOutputStream 向 HDFS 中写入本文件的内容。NameNode 通过 BlockManager 类的对象来进行与数据块相关的操作。BlockManager 中包含了 BlockPlacementPolicy 类的对象,BlockPlacementPolicy 是一个抽象类,通过继承这个类可以实现不同的数据放置策略。HDFS 的机架感知策略在 Hadoop2. 2. 0 中的实现类称为 BlockPlacementPolicyDefault,其 chooseTarget 中利用 NetworkTopology 的 chooseRandom 方法来为新数据块随机选取数据节点。之后被选中的数据节点之间建立管线,FSDataOutputStream 向数据节点中写入数据包。DataNodeDescriptor 是数据节点的描述符,在名字节点中作为数据节点的抽象。收到 DataNode 数据块接收成功消息后,NameNode 将此块的信息更新到 DataNodeDescriptor 中。系统反复进行以上过程将数据写入到 HDFS 中,直到客户端写入完成,关闭输出流,通知 NameNode,写入文件操作完成。

本文在 Hadoop 2. 2. 0 上实现了提出的数据放置策略。首先,在 FSNameSystem 中加入 videoType、filesPerTypeCounter、filesPerTypeAccess、effectParam 变量分别代表事件类型集合、各事件类型文件的数量、各事件类型文件的访问量、各事件类型的影响度。在每次写入文件时,系统获取写入文件的事件类型,如果文件的类型包含于 videoType,那么递增此类型文件的 filesPerTypeCounter。当用户对文件进行读取时,递增对应文件类型的 filesPerTypeAccess。然后,在 FSNameSystem 中加入一个线程类 EffectParamWatcher,它随 NameNode 启动,用于每隔时间 T 更新各事件类型的影响度 effectParam。在 DatanodeDescriptor 中,加入 blocksPerType 变量来记录数据节点存储的各事件类型文件块的数量,在每次数据节点上报成功存储的数据块后,NameNode 增加 DatanodeDescriptor 相应事件的 blocksPerType。通过继承 BlockPlacementPolicy 类实现了 EventBlockPlacementPolicy 来作为数据放置策略实现类。本文在 NetworkTopology 中新增 chooseBest 方法用于在数据节点选择时替换原有的 chooseRandom 方法。chooseBest 方法按照 3. 3 节中的算法流程选出数据块的目标放置节点。

5 实验与结果分析

本文将默认 Hadoop2. 2. 0 与修改后的 Hadoop2. 2. 0 先后部署在实验集群上对所提出的改进策略进行对比实验。实验集群为 1 个机架,包含 3 个数据节点和 1 个名字节点,节点的硬件配置如表 1 所列。

表 1 实验环境

节点	操作系统	CPU	RAM	硬盘
NameNode	Ubuntu12. 10	Intel E7400	4G	160G
DataNode1	Ubuntu12. 10	Intel E7400	2G	160G
DataNode2	Ubuntu12. 10	Intel E7400	2G	160G
DataNode3	Ubuntu12. 10	Intel E7400	2G	160G

受搭建平台规模的局限性,本文将每个文件的大小设定为 64M,与默认块大小相同。设定集群的默认副本数为 1,即每个文件只会存放在一个数据节点。假设写入的数据分为 4 种事件类型: $E = \{A, B, C, D\}$,初始写入时各事件影响度为 $A = \{1, 2, 3, 4\}$ 。实验时首先建立 9 个写入线程,向系统中写入 300 个各类型文件,作为系统已有文件数。各类型文件写

入数量如表 2 所列。

表 2 各事件类型文件的写入读取数量

事件类型	写入数量	读取次数
A	150	60
B	75	120
C	45	180
D	30	240

写入数量的差异代表现实中不同交通事件发生频率的差异。然后重复一遍上述写入操作,但同时再建立 9 个读取线程,连续对各类型文件进行 600 次的读取访问操作,来测试系统的读取和写入性能。读取和写入次数如表 2 所列,各文件访问数量的差异表示实际中用户对不同交通事件视频的访问关注度的差异。本文分别使用 HDFS 默认的机架感知策略与本文提出数据放置策略进行上述实验,并在集群上部署了 Ganglia^[15] 监控系统用于监控集群的各项指标。图 3 所示为通过 Ganglia 在实验中获得的 dfs. datanode. ReadBlockOpAvgTime(读块平均时间)这一指标数据的对比。

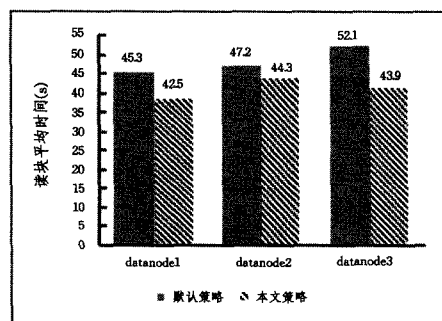


图 3 不同策略下各数据节点的读块平均时间

从图中可以看出本文策略在密集读取操作时相对于默认策略减少了读块平均时间。默认策略的各节点读块平均时间的平均值为 48. 2s,而本文策略为 43. 6s,比默认策略减少了 10%。

图 4 为写性能测试中获得到的 dfs. datanode. WriteBlockOpAvgTime(写块平均时间)这一指标数据的对比。

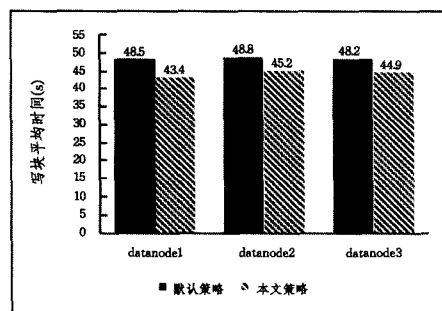


图 4 不同策略下各数据节点的写块平均时间

从图中可以看出,使用本文提出的数据放置策略,系统中每个数据节点的写块平均时间也减少了。默认策略的各节点写块平均时间的平均值为 48. 5s,而本文策略为 44. 5s,比默认策略减少了 8. 2%。因为本文策略在选择节点时需要对节点进行评价,所以目标节点的选择相比默认策略引入了额外的时间开销,具体时间复杂度为 $O(N)$,其中 N 为需要评价的节点数。但是,本文策略降低了过多任务落入同一节点的概率,避免了热点的发生,从而减少了写块平均时间。最后,本

(下转第 229 页)

- [J]. *Enterp Inf Syst*, 2011, 5(2): 169-181
- [5] Jøsang A, Guo G, Pini M S, et al. Combining Recommender and Reputation Systems to Produce Better Online Advice[M]. *Modeling Decisions for Artificial Intelligence*. Springer Berlin Heidelberg, 2013: 126-138
- [6] Cho Young-Sung, et al. Clustering Method using Item Preference based on RFM for Recommendation System in u-Commerce [M]. *Ubiquitous Information Technologies and Applications*. Springer Netherlands, 2013: 353-362
- [7] Yuan X, Lee J H, Kim S J, et al. Toward a user-oriented recommendation system for real estate websites[J]. *Information Systems*, 2013, 38(2): 231-243
- [8] Barragáns-Martínez A B, Costa-Montenegro E, Burguillo J C, et al. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition[J]. *Information Sciences*, 2010, 180(22): 4290-4311
- [9] Cai X, Bain M, Krzywicki A, et al. Collaborative filtering for people to people recommendation in social networks[M]. *AI 2010: Advances in Artificial Intelligence*. Springer Berlin Heidelberg, 2011: 476-485
- [10] Lops P, de Gemmis M, Semeraro G. Content-based recommender systems: State of the art and trends[M]. *Recommender Systems Handbook*. Springer US, 2011: 73-105
- [11] Carrer-Neto W, Hernández-Alcaraz M L, Valencia-García R, et al. Social knowledge-based recommender system. Application to the movies domain[J]. *Expert Systems with Applications*, 2012, 39(12): 10990-11000
- [12] Dunham M H. *Data Mining: Introductory and Advanced Topics* [M]. Pearson Education, 2007
- [13] Zhang Y J, Cheng E. An optimized method for selection of the initial centers of k-means clustering[M]// *Integrated Uncertainty in Knowledge Modelling and Decision Making*. Springer Berlin Heidelberg, 2013: 149-156
- [14] Liu J, Musialski P, Wonka P, et al. Tensor completion for estimating missing values in visual data[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(1): 208-220
- [15] Werner D, Cruz C. A Method to Manage the Precision Difference between Items and Profiles[C]// *2013 International Conference on a Context of Content-Based Recommender System and Vector Space Model. Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2013: 337-344
- [16] Huang X, Wu Q. Micro-blog commercial word extraction based on improved TF-IDF algorithm[C]// *2013 IEEE Region 10 Conference on TENCON*. IEEE, 2013: 1-5

(上接第 224 页)

文评估了集群中数据分布的均匀性;写入读取测试完成后,分别计算了两种策略下集群事件密集度的变差系数。变差系数为所有数据节点的事件密集度标注差与平均数的比值,用以衡量各数据节点存储文件量的离散程度。得到的默认策略下系统的变差系数为 0.0418,而本文策略下的仅为 0.0038,说明文件分布较前者更加均匀。以上实验表明本文提出的基于事件密集度的数据放置策略相比于 HDFS 默认的机架感知策略减少了块的写入时间和读取时间,有效地提高了系统的吞吐量,并且能够将文件更均匀地分布在各个节点上,从而提供了更好的负载均衡。读块时间的减少也有利于系统更及时地响应用户的访问请求。

结束语 随着智能交通系统的普及, Hadoop 等分布式架构将越来越广泛地应用于交通监控系统以应对海量交通监控视频的高效存储以及开放交通监控查询平台等应用对于系统性能的需求。在将 HDFS 用于交通监控视频存储的背景下,本文分析了 HDFS 中默认的机架感知策略的不足,根据交通监控视频的事件分类特征,提出了一种 HDFS 中基于事件密集度的交通监控视频数据放置策略。实验表明本文策略相对于默认策略能够更好地平衡节点负载,提高系统的吞吐量。

参 考 文 献

- [1] 王国锋,宋鹏飞,张蕴灵. 智能交通系统发展与展望[J]. *公路*, 2012(5): 217-222
- [2] 张庆华. 云存储技术在视频监控中的发展与应用[J]. *中国安防*, 2013(8): 53-58
- [3] Borthakur D. The hadoop distributed file system: Architecture and design [J]. *Hadoop Project Website*, 2007, 11: 21
- [4] Shvachko K, Kuang H, Radia S, et al. The hadoop distributed file system [C]// *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 2010: 1-10
- [5] Ghemawat S, Gobioff H, Leung S T. The Google file system [J]. *ACM SIGOPS Operating Systems Review*. ACM, 2003, 37(5): 29-43
- [6] 蔡斌,陈湘萍. *Hadoop 技术内幕* [M]. 北京:机械工业出版社, 2013: 216-217
- [7] 武文斌. 视频监控云存储模型设计[J]. *山西科技*, 2012(3): 35-37
- [8] Xie J, Yin S, Ruan X, et al. Improving mapreduce performance through data placement in heterogeneous hadoop clusters [C]// *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*. IEEE, 2010: 1-9
- [9] 林伟伟. 一种改进的 Hadoop 数据放置策略[J]. *华南理工大学学报:自然科学版*, 2012, 36(1): 152-158
- [10] 刘琨,钮文良. 一种改进的 Hadoop 数据负载均衡算法[J]. *河南理工大学学报:自然科学版*, 2013, 32(3): 332-336
- [11] 徐晓勇,潘郁,丁燕艳. 基于灰色马尔可夫链预测模型的 HDFS 云存储副本选择策略[J]. *计算机应用*, 2012, 31(A02): 39-42
- [12] Ananthanarayanan G, Agarwal S, Kandula S, et al. Scarlett: coping with skewed content popularity in mapreduce clusters [C]// *Proceedings of the sixth conference on Computer systems*. ACM, 2011: 287-300
- [13] Abad C L, Lu Y, Campbell R H. DARE: Adaptive data replication for efficient cluster scheduling [C]// *2011 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2011: 159-168
- [14] 吴萌. *交通监控视频中的异常事件检测* [D]. 北京:北京邮电大学, 2010
- [15] Massie M L, Chun B N, Culler D E. The ganglia distributed monitoring system: design, implementation, and experience[J]. *Parallel Computing*, 2004, 30(7): 817-840