

## 改善 Hadoop 文件处理效率的技术研究

樊 超, 凌 捷

(广东工业大学 计算机学院, 广东 广州 510006)

**摘 要:** 提出一种改善 Hadoop 文件处理效率的方法, 在 Hadoop 中添加一个小文件处理模块 SFPM, 根据文件名为海量小文件建立二级索引, 同时采用预加载技术将索引提前存入缓存, 可提高文件查找访问效率; 在合并文件时, 采取舍弃多余空间的策略, 避免将一个文件拆分存储在两个 block 上, 减少了文件访问时间开销. 实验结果表明该方法能有效减轻 NameNode 的负荷, 提高小文件读写效率.

**关键词:** 分布式文件系统; 小文件; 处理效率; 多余空间; 索引

中图分类号: TP311.5

文献标识码: A

文章编号: 1000-7180(2014)07-0125-04

## Study on Improving the Efficiency for Dealing with Hadoop Small Files

FAN Chao, LING Jie

(College of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China)

**Abstract:** An approach of improving the efficiency for dealing with hadoop small files is proposed in this paper, adding a small file processing module(SFPM) to Hadoop, create indexs of two level according to the file names, and use the preload technology to make the indexs into cache in advance, in order to improve the efficiency of file research and access. When merging files, the strategy of giving up the extra memory space is taken so as to avoid to split one file stored in two blocks, which will reduce the time overhead of file access. The experimental results show that this method can reduce the load of NameNode and enhance its work efficiency of read and write files.

**Key words:** distributed file system; small file; processing efficiency; extra space; index

### 1 引言

在云计算日趋成熟的今天, Hadoop 作为一个能够对大量数据进行分布式处理的软件框架, 得到了广泛的关注和应用. 它具有可靠、可扩展、高效、容错等特性. Hadoop 实现了一个分布式文件系统 HDFS( Hadoop Distributed File System ), 适合那些有着超大数据集的应用程序, 并已经成为海量存储集群上部署的主流文件系统.

在 HDFS 的 block 上文件大小默认为 64 MB, 小于这个数值的文件即称之为小文件<sup>[1-2]</sup>. 海量的小文件会给 Hadoop 的扩展性和性能带来很多问题. 巨大的需求和消耗严重制约了集群的性能和扩展.

同时, 对海量小文件的访问操作, 将需要在各个从属节点 DataNode 之间频繁的转换, 使得访问大量小文件速度远远小于访问几个大文件, 极大地降低了访问效率. 最后, 大量以至难以预计的时间将会耗费在启动和释放一个 task 任务上, 因为每一个文件无论大小都需要要占用一个 slot 任务槽, 这将导致海量小文件的处理效率远远低于处理同等大小的文件<sup>[3]</sup>.

针对 HDFS 处理海量小文件效率低下问题, 许多有效的解决方案也应运而生. 其中主流解决方法涉及 HAR、SequenceFile、MapFile 等, 这些方案都能有效的减少 NameNode 的内存开销, 但是同时它们的不足之处都是难以满足低时延访问的要求.

收稿日期: 2013-09-30; 修回日期: 2013-11-22

基金项目: 广东省战略性新兴产业核心技术攻关项目(2012A010701004); 广东省自然科学基金重点项目(S2012020011071); 广东省教育部产学研合作项目(2012B091000037, 2012B091000041)

文献[4]提到 BlueSky 系统 (<http://www.bluesky.cn/>), 其中主要为 PPT 文件. 提出将同属于一个课件的文件合并成一个大文件, 以提高小文件存储效率<sup>[4]</sup>.

文献[5]中的 WebGIS 系统是由 Web 和地理信息系统(GIS)结合而产生的一种新系统. 在分布式文件系统中数据通常以 KB 为单位被切分成小文件来存储, 目的在于尽可能地减少服务器和浏览器之间传输的数据量. 文中利用 WebGIS 中数据相关性特征, 对那些保存相邻地理位置信息的小文件进行合并, 并且为这些小文件建立索引, 成功的完成了该系统要求高效的实现小文件存取<sup>[5]</sup>这一任务.

## 2 一种改进的小文件处理方法

考虑到 HDFS 的主/从架构以及名称节点 NameNode 在处理海量小文件时负载严重的问题, 提出一种改进的方法, 添加一个小文件处理模块 (Small File Processing Module, SFPM). 该模块主要有如下功能: (1) 判断文件是否为小文件, 若是, 则合并小文件; (2) 为小文件建立目录索引和文件索引; (3) 采用预加载机制, 提高文件访问效率.

### 2.1 小文件合并

在 SFPM 中为小文件建立分层索引, 第一级索引为目录索引, 以文件名为关键字, 通过 hash 实现多个小文件的区分并将结果依次排列. 第二级索引为文件存储位置信息, 记录每个小文件所处的 DataNode 号、block 块号以及偏移量, 通过一系列存储信息找到该文件的所处的节点位置、块号以及具体位置. 二级索引的建立, 能够快速又准确地定位目标文件的存储信息, 为文件的读写操作提供了便利.

#### 2.1.1 索引文件结构

改进的方法为了减少文件查找时间提高访问效率, 给小文件建立了两层的索引结构: 目录索引和文件索引<sup>[6]</sup>. 如图 1 所示, 一级目录索引的结构可定义为三个字段: (fileName, fileLen, indexNum). 以文件名 fileName 为关键字来区别并存储小文件, fileLen 表示文件的大小, indexNum 则指示了该文件在二级索引的存储位置. 设有一级目录索引 (test.txt, 4, 3), 则表示有名为 test.txt 的文件, 大小为 4 kB, 其对应的二级文件索引存放位置为 3.

|          |         |          |
|----------|---------|----------|
| fileName | fileLen | indexNum |
|----------|---------|----------|

图 1 一级目录索引

二级文件索引的结构可定义为四个字段: (dataId, blockId, start, offset). 其中 dataId 代表了文件存储在哪个节点; blockId 代表了该文件所在节点的存储 block 块号; start 指出了文件存储的起始位置地址; offset 则表示文件的偏移量, 如图 2 所示.

|        |         |       |        |
|--------|---------|-------|--------|
| dataId | blockId | start | offset |
|--------|---------|-------|--------|

图 2 二级文件索引

假设有一级目录索引 (example.txt, 11, 0), 则以该文件名为关键字可快速找到其对应二级文件索引 (1, blk\_826540629399449945, 0000, 1024), 即可知道该目标文件存放在数据节点号为 1、blockId 为 826540629399449945 的块上, 通过其起始位置和偏移量便可以很方便访问到文件的内容. 采用以上二级索引结构, 可提高文件的查找效率, 减少了每个小文件的读写操作都必须经过主节点 NameNode 的概率, 降低了 NameNode 的负载<sup>[7]</sup>.

#### 2.1.2 小文件合并的策略

这里定义小于 16 MB 的文件为小文件. 为提高文件的访问速度和读取效率, 将小文件合并成大小为 16 MB 或接近 16 MB 的大文件进行存储. 由于文件大小各不相同, 合并之后的文件不可能每次都是正好为 16 MB, 最后一个存储的文件必定会大于剩余的空间大小 (以下称之为多余空间). 此时有两种处理选择, 是将该文件拆分为两部分分别存储在两个 block 上, 二是另选一个 block 来存储. 为避免文件横跨两个 block 降低文件查找和访问效率, 文章采取的策略是舍弃多余空间而将文件存储在另一 block 之上<sup>[8]</sup>.

假设如图 3 所示, 已有文件 A (4 MB)、文件 B (2 MB)、文件 C (3 MB) 和文件 D (5 MB) 四个文件依次存储在 block\_0 之上, 此时有文件 E (5 MB) 需要存储, 由于该 block 多余空间不足 5 MB, 故另选一个新的块 block\_1 来存储文件 E, 而不是将文件 E 拆分为大小为 2 MB 和 3 MB 分别存储在 block\_0 和 block\_1 上. 此时, block\_0 上所剩 2 MB 空间即为多余空间<sup>[9]</sup>.

### 2.2 文件的读写过程

添加了 SFPM 模块的整个系统结构如图 4 所示: SFPM 部分负责客户端的请求、合并小文件、建立索引、预加载以及 HDFS 交互功能. 而 HDFS 部分主要用于数据存储, 处理 Client 的存储和读取请求.

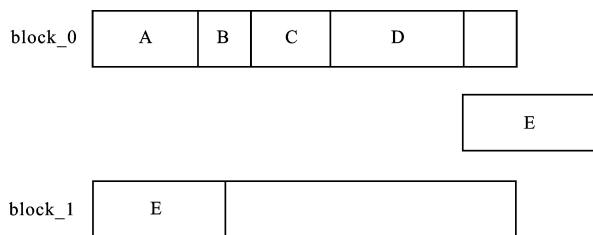


图3 文件合并示意图

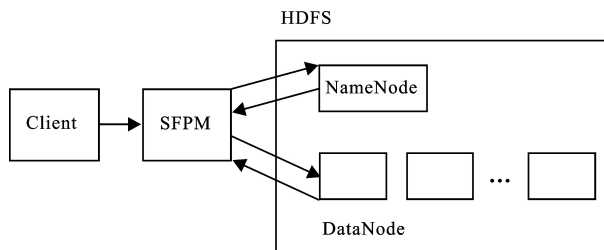


图4 SFPM 结构

### 2.2.1 文件写入操作

文件的写操作是将文件写入到 DataNode 的 block 中,更新 NameNode 及相关 DataNode 中元数据信息.由于建立了目录索引,属于同一目录的文件最好写入相同的 block 中,这样将会提高 MapReduce 分布式计算的速度.过程如下:

(1)SFPM 组建文件目录,根据文件名将其排序,建立一级目录索引.

(2)寻找块和 DataNode 来存储文件:

①搜索第一个使用频率最高的 DataNode 节点的缓存,若找出的 block 是一个新 block,则将后续的文件依次存入;若找到的 block 已存有文件且未存满,则将属于同一个目录中的文件存入.

②如果缓存列表为空或找到的 block 均以填满无法满足继续存储要求,则 Client 将发送一个文件写入请求到 NameNode 节点,NameNode 则会按照正常处理流程,在收到存储请求后会将包含 DataNode 及 block 位置信息的所有副本返回给 Client.完成文件写操作之后,建立二级文件索引,并更新一级目录索引.

(3)将索引与文件一同传输到 DataNode.传输结束后,DataNode 将二级索引加载到内存.而在 SFPM 中,则更新目录索引后同样加载到内存.

(4)通知用户存储成功.

### 2.2.2 文件读取操作

文件的读取操作是将存储在 DataNode 节点 block 上的文件读取并返回,由于 HDFS 是基于“一次写入,多次读取”的模型设计的,因此相比文件的写入,提高文件读取效率意义更加重大.主要过程

如下:

(1)根据文件名查询缓存,如果命中,将结果返回给 Client;

(2)查找一级目录索引,获取其对应二级文件索引信息,以便获取该文件的存储具体细节;

(3)在获得二级文件索引中的信息(dataId, blockId, start 和 offset)之后,Client 向 DataNode 发出读取的文件请求,不需要经过 NameNode 节点即可顺利完成文件的读取操作.

(4)将结果返回给用户.

在步骤(2)中未找到相关文件名或搜索到的 block 失效,则需要按照一般情况来处理:Client 首先向 NameNode 发出读取文件请求;NameNode 将文件存储的信息包括 block 号和偏移量等所有副本的信息返回;Client 根据这些信息默认选择距离最近的副本完成文件的读取<sup>[11]</sup>.

## 3 实验结果与分析

### 3.1 实验环境

本文的实验环境是一个包含三个节点的 Hadoop 集群<sup>[12]</sup>,其中一台机器作为 NameNode 节点,另外两台机器作为 DataNode 节点.每个节点的机器基本配置为:Windows XP SP2 系统,并且已经安装 Cygwin;硬件条件为 Intel Pentium(R) Dual-Core CPU 2.70 GHz,2 GB 内存,160 GB 硬盘;Hadoop 0.20.2 版本;实验网络环境为 10 Mb/s.

### 3.2 结果与分析

本文实验对比的主要指标为写入、读取小文件所需时间开销,使用本文的方法与原 HDFS 默认的处理小文件的方法分别进行实验,并对二者的结果进行了对比,通过对时间开销分析,证明添加 SFPM 模块的方法在提高 HDFS 文件读写效率上起到了一定的改善作用.

#### 3.2.1 实验数据

由于文件的大小影响着 HDFS 的性能,特别是对 HDFS 内存的开销,所以选择不同大小的文件进行测试效果更加明显.本实验选取 10 000 个小文件,大小在 1 kB 到 256 kB 之间不等,总共约 0.35 GB.其中文件大小在 16~32 kB 和 32~64 kB 的部分分别占总文件的 41%和 36%,这两部分占据了整个文件的主要部分.所有文件比例分布如图 5 所示.

#### 3.2.2 文件读写性能对比

本实验分别对文件的写入和读取所用时间进行对比.将 10 000 个小文件分别进行 5 组实验,第一

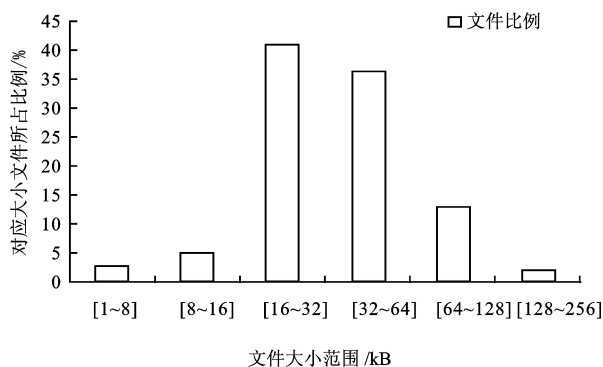


图5 文件大小分布

组取 2 000 个文件,第二组取 4 000 个文件,以此类推每组增加 2 000 个文件.对每组实验分别进行 7 次,记录 7 次实验结果,去掉最大值和最小值,将剩余的 5 个值取平均值即为最后的结果.实验对比的双方分别是添加 SFPM 模块后的 HDFS(以下表示为 pro\_HDFS)和未作改动之前的 HDFS(以下表示为 ori\_HDFS).对文件的写入和读取操作分开实验分别作时间开销的统计,更益于区分对比.

#### (1) 文件写入时间

如图 6 所示,pro\_HDFS 写入文件的时间比 ori\_HDFS 下要少很多.由于正常情况下每个文件的写入都需要先向 NameNode 发出写请求以获取 DataNode 的存储 block 块信息,海量的小文件使得写请求的次数剧增,同时也将耗费更多的时间.通过合并文件减少了文件个数,生成的二级索引降低或避免了 NameNode 超负载的情况,文件写入操作的时间必将大大减少.同时,预加载技术也会相应的降低时间消耗.

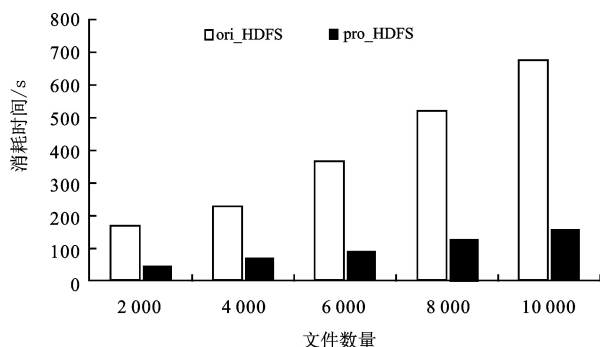


图6 文件写入所用时间

#### (2) 文件读取时间

如图 7 所示,对比 ori\_HDFS 和 pro\_HDFS 二者读取文件操作所消耗的时间,可发现后者所用时间明显较少.在文件数量到达 8 000 到 10 000 左右时,时间减少可达到大约一半. SFPM 模块为文件建

立二级索引,读取文件操作避免的正常情况下先要向 NameNode 发出读取文件请求的过程,通过目录索引和文件索引即可快速找到目标文件.此外,采用的预加载技术将索引提前放入缓存中,也将在一定程度上降低查找时间,提高访问效率.

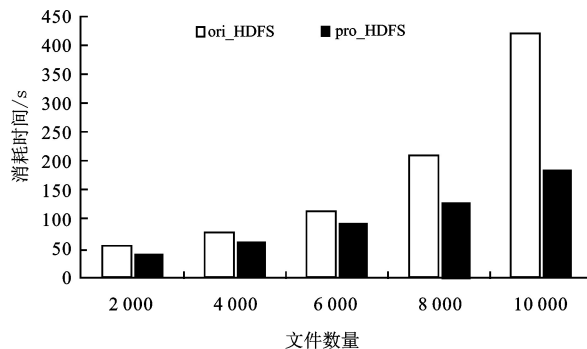


图7 文件读取所用时间

## 4 结束语

提出一种添加小文件处理模块 SFPM 的方法.利用文件之间的相关性,分别生成目录索引和文件索引,旨在解决 NameNode 处理海量小文件时内存开销过大和效率偏低的问题.将二级文件索引存放在相关 DataNode 之中并预加载到内存,减少索引加载和查找带来的时间消耗.并且在合并小文件时,采取放弃多余空间的策略,避免将一个文件拆分存储到两个 block 上,在一定程度上能降低查找文件所用时间开销.这种方法的不足之处在于索引的组建和存储将会造成内存的消耗和时间的延迟,对 Hadoop 整体的处理效率会有一定的影响.

## 参考文献:

- [1] TOM WHITE. Hadoop: The Definitive Guide[M]. 2nd ed. O'Reilly Media, Inc, 2011.
- [2] 吴永发,曾碧.一种面向移动终端的云存储方法研究[J].微电子学与计算机,2013,30(6):48-56.
- [3] Cheng Xingguo, Xiao NanFeng, Huang Faliang. Research on HDFS-based web servercluster[C]//2011 International Conference on E-Business and E-Government (ICEE). Shanghai, China; IEEE, 2011: 1-4.
- [4] 栾亚建,黄翀民,龚高晟,等. Hadoop 平台性能优化研究[J]. 计算机工程, 2010, 36(14): 262-264.
- [5] Grant M, Saba S, Wang J. Improving metadata management for small files in HDFS[C]//The 2009 IEEE International Conference on Cluster Computing and Worker shops, New Orleans; IEEE, 2009.

(下转第 132 页)

综上所述,本算法不但能够比较全面地找出意见领袖,而且能够将网络社团进一步细化,给出细化后各子社团的划分及其意见领袖,并且错误率也比较低,且具有较强的实用性。

## 6 结束语

意见领袖的筛选对于网络舆情的分析与监控具有重要的现实意义。我们通过抽取样本建立模型将网络中的链接划分为核心链接和非核心链接,迅速将网络的骨干进行划分,并依据划分后的网络骨干将网络进一步细化成多个子社团,给出了子社团的划分及其意见领袖。但是本方法在大型网络社区中进行意见领袖挖掘的效果还有待于完善,因此如何提高大型网络社区中的意见领袖挖掘方法的时间复杂度和空间复杂度,将是以后研究的重点方向。

## 参考文献:

- [1] Lazarafeld P F, Berelso B, Gaudet H. The people's choice: how the voter makes up his mind in a presidential campaign[M]. New York: Columbia University Press, 1948.
- [2] 倪邦文. 中国网络青年意见领袖的构成、特征及作用

[J]. 中国青年研究, 2011(9):05-09.

- [3] Ahn Y Y, Bagrow J P, Lehmann S. Link communities reveal multiscale complexity in networks[J]. Nature, 2010, 466(7307):761-764.
- [4] Grolmusz V. A note on the page rank of undirected graphs[J]. CoRR, 2012(2):1205-1260.
- [5] Das Sarma A, Rahaman Molla A, Pandurangan G, et al. Fast Distributed PageRank Computation[J]. Internet Mathematics, 2012, 2(3):251-273.
- [6] 高俊波, 杨静. 在线论坛中的意见领袖分析[J]. 电子科技大学学报, 2007, 36(6):1249-1252.
- [7] Zhai Z W, Xu H, Jia P F. Identifying opinion leaders in bbs[C]//Proc of WI-IAT'08, Sydney, Australia: IEEE, 2008:398-400
- [8] 樊兴华, 赵静, 方滨兴, 等. 影响力扩散概率模型及其用于意见领袖发现研究[J]. 计算机学报, 2013, 36(2): 360-367.

## 作者简介:

申艳光 女, (1970-), 教授. 研究方向为数据挖掘、隐私保护.  
李金伟 男, (1983-), 硕士研究生. 研究方向为数据挖掘、社团挖掘.

(上接第 128 页)

- [6] Dong Bo, Qiu Jie, Zheng Qinghua, et al. A novel approach to improving the efficiency of storing and accessing small files on hadoop: A case study by power point files [C]//International Conference on Services Computing, Miami, FL: IEEE, 2010:65-72.
- [7] 郭琦娟, 陈通照. 一种动态更新索引结构的设计与实现[J]. 计算机系统应用, 2008(12):76-69.
- [8] 史长琼, 唐铭, 张大方, 等. 基于 Trie 树的哈希表[J]. 计算机应用, 2010, 30(8):2193-2196.
- [9] Liu Xuhui, Han Jizhong, Zhong Yunqin, et al. Implementing WebGIS on hadoop: A case study of improving small file I/O performance on HDFS[C]// Proc of the 2009 IEEE Conf on Cluster Computing and Workshops, New Orleans, LA: IEEE, 2009:1-8.
- [10] 泰冬雪. 基于 Hadoop 的海量小文件处理方法的研究[D]. 沈阳: 辽宁大学, 2011.
- [11] 余思, 桂小林, 黄汝维, 等. 一种提高云存储中小文件件

存储效率的方法[J]. 西安交通大学学报, 2011, 45(6): 59-63.

- [12] 刘琨, 肖琳, 赵海燕. Hadoop 中云数据负载均衡算法的研究及优化[J]. 微电子学与计算机, 2012, 29(9): 18-22.

## 作者简介:



樊超 男, (1989-), 硕士. 研究方向为网络与信息安全技术, Hadoop 云计算.

凌捷 男, (1964-), 博士, 副院长. 研究方向为网络与信息安全技术.