第18卷 第2期 2018年1月
1671—1815(2018)002-0285-05
科 学 技 术 与 工 程
Science Technology and Engineering
Vol.18 No.2 Jan. 2018



引用格式:薛弘晔,田治武,罗香玉,等. 一种不依赖访问热度信息的分布式文件放置算法[J]. 科学技术与工程,2018,18(2):285—289
Xue Hongye,Tian Zhiwu,Luo Xiangyu,et al. A distributed file placement algorithm without depending on popularity information[J]. Science Technology and Engineering,2018,18(2):285—289

# 一种不依赖访问热度信息的分布式文件放置算法

薛弘晔 田治武 罗香玉* 冯 健 王 丹
(西安科技大学计算机科学与技术学院,西安 710054)

**摘 要** 文件放置问题一直是分布式存储领域的研究热点。分布式文件存储系统HDFS随机选择节点完成文件放置,存在访问负载分布不均衡的缺点。研究人员提出大量基于文件访问热度信息的放置算法;但是,文件的访问热度信息是动态变化的,难以准确预测。提出一种不依赖访问热度信息的分布式文件放置算法;该算法仅使用文件的创建时间信息,利用文件已创建时间与访问热度之间的相关性,首先将时间进行区间划分,然后统计出各节点在不同时间区间内所创建文件的数据量,放置过程中保持同一时间区间的数据量在不同节点间大致相同。实验结果表明,该算法不仅可以使各节点的存储负载达到均衡,还能够提升访问负载的均衡,消除因文件访问热度不均而导致的性能瓶颈。

**关键词** 分布式文件存储系统 文件访问热度 文件放置 负载均衡
**中图法分类号** TP311.12; **文献标志码** A

随着大数据时代的到来,海量数据正以前所未有的趋势冲击着整个数据中心。根据IDC(international data corporation)2013年的预测,到2020年,将会有约300亿个设备连接到互联网,数据将以每两年翻一番的速度增长,平均每人将拥有约5.2 TB的数据[1]。面对如此庞大的数据,传统的数据存储管理方法和系统已经无法满足人们的需求。

分布式文件存储系统的研究受到广泛重视。其设计的核心理念是让多台服务器协同工作,完成单台服务器无法处理的任务,尤其是高并发或者大数据量的任务。分布式文件存储系统的代表包括Google file system(GFS)、Hadoop distributed file system(HDFS)、Taobao file system(TFS)、Facebook haystack[2]等。

文件的放置问题是影响分布式文件存储系统整体性能的关键问题之一,也是目前学术领域的一个研究热点。良好的文件放置策略不仅可以提高文件存储效率,同时能够促进访问负载的均衡分布。

2017年6月14日收到 西安科技大学博士启动基金(2015QDJ031)和陕西省教育厅专项科学研究计划项目(15JK1468)资助
第一作者简介:薛弘晔(1960—),男,工学博士。研究方向:网络与高性能计算。E-mail:xuehy60@qq.com。
*通信作者简介:罗香玉(1984—),女,博士,讲师。研究方向:分布式计算。E-mail:luoxiangyu@xust.edu.cn。

目前使用最广泛的分布式文件系统HDFS[3]默认的文件放置策略是随机的数据块放置策略[4]。虽然针对随机放置策略造成节点访问负载不均衡的现象,Hadoop平台内置了一个名为balancer的负载均衡程序,但主要是从存储空间利用率角度考虑,不当的使用反而可能会加重访问负载不均衡的程度[5]。为了提升分布式文件存储系统的整体性能,研究人员提出了基于文件访问热度信息的分布式文件放置方法。但是,文件访问热度信息在文件存储时刻并不可知,并且是动态变化的,难以预测出准确值。

本文提出一种不依赖访问热度信息的分布式文件放置算法。该算法仅需要文件的创建时间信息,利用文件已创建时间与访问热度之间的相关性完成文件的放置。首先将时间进行区间划分,然后统计出各个数据节点在不同时间区间内所存储文件的数据量。放置过程中保持同一时间区间的数据量在不同数据节点间大致相同。实验结果表明,通过该算法完成文件的放置不仅可以使各数据节点的存储负载达到均衡,而且能够提升访问负载的均衡,使得各个数据节点上都分布有高热度的文件。

## 1 相关工作

针对分布式文件存储系统的节点特性,罗鹏等[6]在HDFS原始策略的基础上充分考虑了节点

的差异性,通过采集数据节点的 CPU 负载、内存负载、网络负载和存储空间负载,引入负载因子计算数据的存放成本,以选择一个最优的数据节点存放数据。实验结果表明,该策略优化了集群负载的均衡性,同时也提高了数据块副本的传输性能。罗军等[7]针对 HDFS 在文件放置过程中没有综合考虑各节点服务器的差异性,导致集群负载失衡,以及文件采用随机放置策略造成节点之间网络距离过长等问题,提出了一种基于 SVM 的文件放置策略。实验结果表明该策略能更有效实现负载均衡。

基于文件热度信息方面的研究,张松等[8]基于预测文件的热度信息来提升分布式文件系统文件放置策略,结果表明可以有效减少热点数据的访问冲突,减少热点数据作业的执行时间和网络负载。伍文静等[9]采取基于密度的单次频度分析和基于区间长度的两次频度聚合的方法对文件访问流进行分析,作为确定热点文件的依据,提升文件放置策略。陶永才等[10]在同时考虑节点异构性及文件热度的情况下对文件副本系数进行动态调整来提高系统的负载均衡,降低作业执行时间。王宁等[11]根据用户对内容块兴趣需求,定义了数据块的热度并提出了一种满足用户需求的存储数据块的最小服务成本策略(MCSB)。实验结果表明该策略能够在降低服务成本的同时,有利于系统的负载均衡,并提高该系统的服务性能。

## 2 不依赖访问热度信息的放置算法

提出的不依赖访问热度信息的分布式文件放置算法的基本思想是让每个数据节点内既有老文件,也有新文件,而且各个数据节点之间创建于相同时间区间的文件数据量基本是相同的。

Tang 等[12]发现,不管是普通文件还是具有时效性的文件其访问分布都与时间是有关系的。总体来看文件在刚创建的一段时间其访问量是较高的,随着时间的推移,访问量逐渐下降,到最终很少有人访问,变成了“冷数据”[13]。由此发现文件已存在的时间与文件访问热度之间存在极强的相关性。

因此,为了实现让每个数据节点既存储有热度值高的文件,也存储有热度值低的文件,而且各个数据节点之间热度值相似的文件数据量基本一致,就可以用文件的创建时间来代替。即让每个数据节点既存储有创建时间较短的文件,也存储有创建时间较长的文件,且各个数据节点之间相同时间区间的文件数据量基本相同。

### 2.1 时间区间的划分

利用创建时间进行文件的存储,时间区间的划分非常重要。根据 Tang 等[12]的研究,文件访问热度的变化率在文件创建初期变化较大,随着时间的推移,变化速率越来越小,文件访问热度随时间的变化呈现出对数正态分布的趋势。本文采取指数函数划分的方法,即区间 $i$ 中文件已创建时间介于$2^{i-1}$ ~ $2^i$ d 之间(区间 0 除外,介于 0 ~1 d 之间)。

确定了时间划分的方法,接下来就是确定划分时间的区间数。先来观察一组数:当 $i=10$ 时,$2^{10}=$ 1 024 d,将近 3 年;当 $i=15$ 时,$2^{15}=32\ 768$ d,将近 90 年;当 $i=20$ 时,$2^{20}=1\ 048\ 576$ d,将近 2 873 年。因此,划分的时间区间数在 20 以内就可以满足大多数需要,具体取值可以根据文件系统的情况自行设定。

### 2.2 各时间区间数据量的统计及其动态更新

确定了划分时间的方法、时间划分的区间数以及时间划分的区间长度之后,就可以利用计数器来统计出每个时间区间内的文件数据量。

随着时间的推移,文件所属的时间区间会发生变化,所以各数据节点数据量的更新也是很重要的。数据节点数据量的更新就是统计最新的数据节点存储状态信息,只有获取到最新的节点存储状态信息,文件才能够准确选择目的数据节点。本文采取的更新策略是实时更新加周期更新。所谓实时更新是指当各数据节点存储新文件时及时进行更新;所谓周期更新是指以天为周期,重新计算各个文件所属的时间区间,进而更新各数据节点的数据量。

### 2.3 新型文件放置策略

基于文件已创建时间区间的划分以及各区间数据量的计数,文件放置的具体步骤如下:

(1)获取文件的创建时间。

(2)计算文件已创建的时间,单位为天(即系统当前时间 - 文件创建时间)。

(3)确定文件所属时间区间。

(4)比较各节点相应时间区间的数据量,选取计数最小的节点,若出现多个节点相应时间区间内文件数据量计数相同的情况,则比较这几个节点存储的文件总数据量,取文件总数据量最少的节点作为目的存储节点。

(5)在步骤(4)选择的节点上完成文件的存储。

采取本文的分布式文件放置算法进行文件存储,可以使得各个数据节点的总数据量基本相同;各个数据节点对应相同的时间区间内的文件数据量也基本是一致的。

图 1 所表示的是分布式文件存储系统示例图。

假设该分布式文件存储系统有 4 个数据节点,每个数据节点将时间按照指数函数划分的方法划分成 4 个时间区间,分别存储已创建时间为 0 ~1 d、1 ~2 d、2 ~4 d、4 ~8 d 的文件。假设现在有 20 个大小都是 1 mB 的文件要往该分布式文件系统中存储,存储的文件中 1 ~4 号文件已创建 1 d,5 ~7 号文件已创建 2 d,8 ~12 号文件已创建 3 d,13 ~15 号文件已创建 4 d,16 ~19 号文件已创建 5 d,20 号文件已创建 6 d。完成存储的最终结果如图 2 所示。

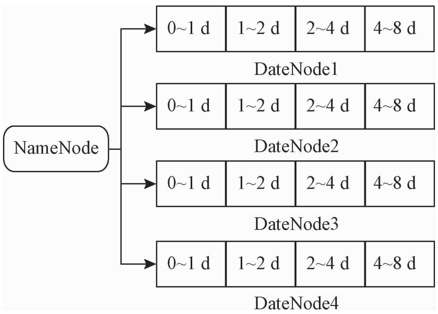


图 1　分布式文件系统示例图

Fig. 1　Sample graph of distributed file system

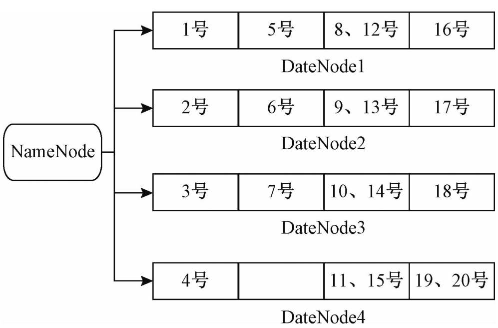


图 2　存储结果示例图

Fig. 2　Sample graph of storage results

由图 2 可以看出,完成存储后,该分布式文件存储系统的 4 个数据节点中文件数据量都是 5 MB,而且,相同时间区间的文件数据量也基本相同。存储结果与本算法的设计目标是一致的。

# 3　实验设计与结果分析

为了验证不依赖访问热度信息的分布式文件放置算法的可行性,本文是基于 Hadoop 平台下的分布式文件存储系统 HDFS 的原理进行抽象仿真实验的。并通过实验与 HDFS 默认的文件随机放置策略进行比较。

## 3.1　实验设计

首先通过 Java 程序模拟仿真 HDFS 分布式文件存储系统;然后根据本文提出的算法思想对模拟分布式文件存储系统的文件随机放置策略进行改写,设计基于本算法的模拟分布式文件存储系统;其次将相同的实验数据分别存储在模拟 HDFS 分布式文件存储系统和基于本算法的模拟分布式文件存储系统中;最终比较二者的存储结果。

实验结果重点关注以下两个方面。

(1) 基于本算法的模拟分布式文件存储系统完成文件存储后各个数据节点的文件数据量是否基本相同,即验证采用本文提出的算法进行文件的存储是否可以实现存储负载均衡。

(2) 各个数据节点内文件的访问热度分布情况,即验证采用本文提出的算法进行文件的存储是否可以解决访问热度较高的文件集中存储在少数节点上的问题。

为了直观地观察各个数据节点内文件访问热度的分布情况,本文根据文件的创建时间与文件的访问热度之间的相关性,由文件的创建时间定量给出相应文件的访问热度值。最终通过计算各个数据节点的文件访问热度值和,将结果绘制成一张柱状图,然后通过柱状图直观的观察不同访问热度的文件在各个数据节点中的分布情况。

## 3.2　实验数据

根据 Tang 等[12]的研究,本文实现了文件访问负载生成器,其中,文件间热度分布服从类 Zipf 分布,文件到达服从泊松分布,同一文件热度随时间演化呈对数正态分布。然后利用文件访问负载生成器随机产生 10 000 条数据,每条数据大小为 1 MB。将随机产生的 10 000 条实验数据的文件创建时间与相应的文件访问热度值绘制在一张柱状图上,其中文件创建时间为横坐标,文件的访问热度值为纵坐标,如图 3 所示。

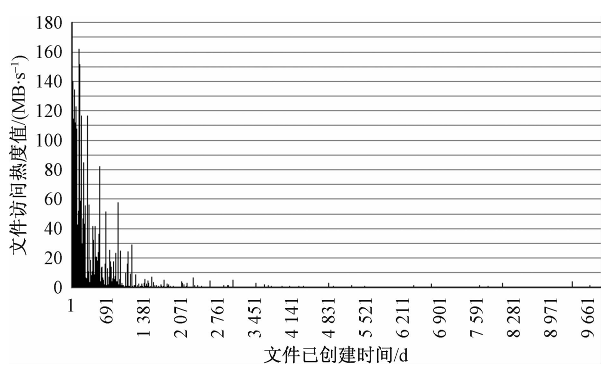


图 3　文件创建时间与热度值关系图

Fig. 3　The relation of file creation time and file popularity

由图 3 可以观察到,实验数据中文件创建时间与相应的文件访问热度值的关系与 Tang 等[12]所研究的文件访问频率(即文件访问热度)随时间的变化趋势基本相似,由此可以确定实验数据是有效的。

### 3.3 实验结果

将 10 000 条实验数据分别存储在模拟 HDFS 分布式文件存储系统和基于本算法的模拟分布式文件存储系统中。实验中设定模拟分布式存储系统的数据节点数为 8 个,时间划分为 10 个时间区间。完成存储后,统计得到基于本算法的模拟分布式文件存储系统各数据节点的文件数据量如表 1 所示。由表 1可以得出,采用本文算法进行文件的存储,可以实现各个数据节点文件数据量基本一致,即可以达到存储负载均衡的效果。

**表 1 基于本算法分布式文件存储系统节点文件分布表**

**Table 1 Based on this algorithm distributed file storage system file distribution in nodes**

| 数据节点 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 文件数据量/MB | 1 249 | 1 250 | 1 251 | 1 250 |
| 数据节点 | 5 | 6 | 7 | 8 |
| 文件数据量/MB | 1 250 | 1 250 | 1 250 | 1 250 |

假设模拟 HDFS 分布式文件存储系统在完成文件存储后又执行了其内部负载均衡程序 balancer,即使其达到存储负载均衡后,再比较默认系统与基于本算法系统的文件热度分布情况。

利用公式 $H_i = \sum P_j M_j$ 分别计算两种放置算法下存储系统各数据节点 $i$ 的文件访问热度值和。其中,$H_i$ 为节点 $i$ 的文件访问热度值和;$P_j$ 为节点 $i$ 上文件 $j$ 的访问频率;$M_j$ 为文件 $j$ 的数据量。由此得到如图 4 和图 5 所示的柱状图。

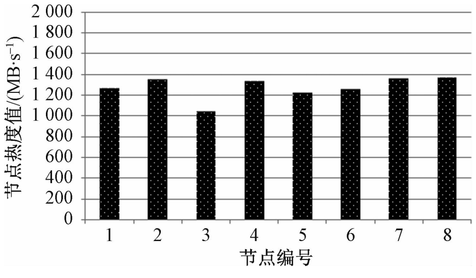


图 4 基于本算法的系统各数据节点文件访问热度值和

Fig. 4 The sum of file popularity value of each node based on our placement algorithm

由图 4 和图 5 对比可知基于本算法的分布式文件存储系统各数据节点的文件访问热度值和比较平稳,即不同访问热度值的文件在各个数据节点中的分布是均衡的。但模拟 HDFS 分布式文件存储系统各数据节点的文件访问热度值和相差比较明显,即不同访问热度值的文件在各个数据节点中的分布是不均衡的。

以上实验结果表明,采用本文提出的算法进行文件的存储可以实现各个数据节点的文件总数据量

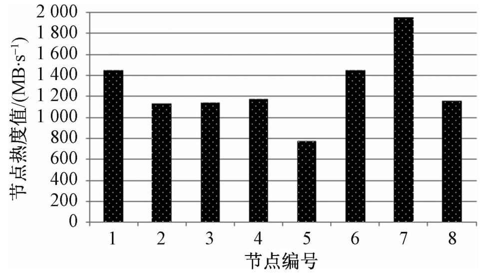


图 5 模拟 HDFS 系统各数据节点文件访问热度值和

Fig. 5 The sum of file popularity value of each node based on the placement algorithm of HDFS

基本相同,各个数据节点对应相同的时间区间内的文件数据量也基本一致的目标。这不但保证了各数据节点存储负载的均衡,而且也能够提升访问负载的均衡,消除因文件访问热度不均而导致的性能瓶颈问题。

## 4 结语

本文针对基于文件访问热度信息以及 HDFS 默认的随机分布式放置算法的缺陷,提出了一种不依赖文件访问热度信息的分布式文件放置算法。该算法不但可以实现存储负载均衡,而且也可以提升各数据节点的访问负载均衡,消除因文件访问热度不均而导致的性能瓶颈问题。目前仅针对同构系统进行了研究,下一步工作是提出数据节点异构情况下的不依赖访问热度信息的算法。

### 参 考 文 献

1 张广彬, 盘 骏, 曾智强. 数据中心 2013:硬件重构与软件定义. 2014

Zhang Guangbing, Pan Jun, Zeng Zhiqiang. Data center 2013: hardware refactoring and software definition. 2014

2 杨传辉. 大规模分布式存储系统: 原理解析与架构实战. 北京: 机械工业出版社, 2013

Yang Chuanhui. Large-scale distributed storage system: principle analysis and structure of actual combat. Beijing: Machinery Industry Press, 2013

3 Shvachko K, Kuang H, Radia S, *et al.* The hadoop distributed file system. IEEE, Symposium on MASS Storage Systems and Technologies. IEEE Computer Society, 2010:1—10

4 王永洲, 茅 苏. HDFS 中的一种数据放置策略. 计算机技术与发展, 2013; (5): 90—92

Wang Yongzhou, Mao Su. A data placement strategy in HDFS. Journal of Computer Technology and Development, 2013; (5):90—92

5 杨俊杰, 廖卓凡, 冯超超. 大数据存储架构和算法研究综述. 计算机应用, 2016; 36(9): 2465—2471

Yang Junjie, Liao Zhuofan, Feng Chaochao. Research on large data storage architecture and algorithm. Journal of Computer Application, 2016; 36(9): 2465—2471

6 罗 鹏, 龚 勋. HDFS 数据存放策略的研究与改进. 计算机工

程与设计，2014；35（4）：1127—1131
Luo Peng，Gong Xun. Research and improvement of HDFS data storage strategy. Journal of Computer Engineering and Design，2014；35（4）：1127—1131

7 罗　军，陈仕强. 基于支持向量机的 HDFS 副本放置改进策略. 计算机工程，2015；41（11）：114—119
Luo Jun，Chen Shiqiang. Implementation strategy of HDFS replica placement based on support vector machine. Journal of Computer Engineering，2015；41（11）：114—119

8 张　松，杜庆伟，孙　静，等. 基于预测的云计算热点数据副本因子决策算法. 计算机与现代化，2015；（2）：62—66
Zhang Song，Du Qingwei，Sun Jing，*et al.* Decision-making algorithm based on cloud computing hotspot data replica factor based on forecasting. Journal of Computer and Modernization，2015；（2）：62—66

9 伍文静，程耀东，汪　璐，等. 面向本地分布式存储系统的动态副本策略. 计算机工程与应用，2010；46（12）：21—24
Wu Wenjing，Cheng Yaodong，Wang Lu，*et al.* Dynamic replica strategy for local distributed storage systems. Computer Engineering & Applications，2010；46（12）：21—24

10 陶永才，张宁宁，石　磊，等. 异构环境下云计算数据副本动态管理研究. 小型微型计算机系统，2013；34（7）：1487—1492
Tao Yongcai，Zhang Ningning，Shi Lei，*et al.* Research on dynamic management of data replicas of cloud computing in heterogeneous environments. Journal of Chinese Computer Systems，2013；34（7）：1487—1492

11 王　宁，杨　扬，孟　坤，等. 云计算环境下基于用户体验的成本最优存储策略研究. 电子学报，2014；42（1）：20—27
Wang Ning，Yang Yang，Meng Kun，*et al.* A customer experience-based cost minimization strategy of storing data in cloud computing. Acta Electronica Sinica，2014；42（1）：20—27

12 Tang W，Fu Y，Cherkasova L，*et al.* Modeling and generating realistic streaming media server workloads. International Journal of Computer & Telecommunications Networking，2007；51（1）：336—356

13 柴云鹏，杨　楠. 冷数据集中的流媒体存储系统节能方法. 计算机科学，2012；39（10）：148—151
Chai Yunpeng，Yang Nan. Cold data concentration：energy saving method for streaming media storage systems. Journal of Computer Science，2012；39（10）：148—151

# A Distributed File Placement Algorithm without Depending on Popularity Information

XUE Hong-ye，TIAN Zhi-wu，LUO Xiang-yu*，FENG Jian，WANG Dan
（School of Computer Science and Technology，Xi'an University of Science and Technology，Xi'an 710054，China）

［**Abstract**］ File placement has always been a research hotspot in the field of distributed storage. The distributed file storage system HDFS places files by randomly selecting nodes，which leads to imbalance in accessing load. Researchers have proposed a large number of placement algorithms based on file popularity. However，file popularity is dynamically changing，and is difficult to accurately predict. A distributed file placement algorithm was proposed without depending on file popularity. According to the creation time of file and the correlation between creation time and file popularity，the algorithm firstly divides the time interval，and then counts the data of each node in different time intervals. It keeps the data of different nodes in the same time interval roughly the same. Experimental results show that the algorithm can balance not only the storage load，but also the access load on each node，and it eliminates the performance bottleneck caused by the uneven distribution of file popularity.

［**Key words**］ distributed file storage system　　file popularity　　file placement　　load balance