



# 基于存储熵的存储负载均衡算法

周渭博<sup>1,2\*</sup>, 钟 勇<sup>1</sup>, 李振东<sup>1,2</sup>

(1. 中国科学院 成都计算机应用研究所, 成都 610041; 2. 中国科学院大学, 北京 100049)

(\* 通信作者电子邮箱 zhouweibo@casit.com.cn)

**摘 要:**在分布式存储系统中,一般都是以磁盘空间利用率(DU)来判断各存储节点的负载均衡程度,当所有节点的磁盘空间利用率相等时,是整个分布式存储系统的存储负载均衡点。但是在实际的应用场景中,磁盘I/O速率比较低的存储节点和可靠性比较低的存储节点往往成为影响整个存储系统数据读写性能的瓶颈,因此在异构分布式存储系统中,特别是各存储节点磁盘I/O速率和可靠性差异较大的分布式存储系统中,如果仅仅以磁盘空间利用率作为存储负载均衡的判定条件,则其数据的读写效率必然受到限制。从读写效率的角度提出一种度量分布式存储系统中存储负载均衡的新思路。根据负载均衡理论和熵理论给出存储熵(SE)的定义,并提出一种基于存储熵的负载均衡算法,该算法通过系统负载判定、单节点负载判定和负载迁移实现了对分布式存储系统存储负载的量化调整,并通过实验与基于磁盘空间利用率的负载均衡算法进行了对比分析,验证了该算法对分布式存储系统中存储负载具有良好的均衡性,有效地控制了系统负载失衡的问题,提高了分布式存储系统的整体读写效率。

**关键词:**存储熵;存储负载均衡;读写效率;分布式存储系统

**中图分类号:** TP301 **文献标志码:** A

## Storage load balancing algorithm based on storage entropy

ZHOU Weibo<sup>1,2\*</sup>, ZHONG Yong<sup>1</sup>, LI Zhendong<sup>1,2</sup>

(1. Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu Sichuan 610041, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** In the distributed storage system, Disk space Utilization (DU) is generally used to measure the load balance of each storage node. When given the equal disk space utilization to each node, the balance of storage load is achieved in the whole distributed storage system. However, in practice, the storage node with relatively low disk I/O speed and reliability becomes a bottleneck for the performance of data I/O in the whole storage system. Therefore in heterogeneous distributed storage system and specially the system which has great differences in disk I/O speed and reliability of each storage node, the speed of data I/O is definitely limited when disk space utilization is the only evaluation criteria of storage load balance. A new idea based on read-write efficiency was proposed to measure the storage load balance in the distributed storage system. According to the definition of Storage Entropy (SE) given by the theory of load balance and entropy, a kind of load balance algorithm based on SE was proposed. With system load and single node load determination as well as load shifting, the quantitative adjustment for storage load of the distributed storage system was achieved. The proposed algorithm was tested and compared with the load balance algorithm based on disk space utilization. Experimental results show that the proposed algorithm can balance storage load well in the distributed storage system, which effectively restrains the system load imbalance and improves the overall efficiency of reading and writing of the distributed storage system.

**Key words:** storage entropy; storage load balancing; read-write efficiency; distributed storage system

## 0 引言

面对海量数据的存储和计算需求,通常采用分布式模型,这种模型将待处理的数据存储到通过网络连接的若干个地理位置不同的存储节点上,其典型代表是 Apache 开源的 Hadoop<sup>[1]</sup>。随着海量数据的不断增长,分布式存储系统<sup>[2]</sup>往往通过横向扩展的方式,不断增加新的存储节点,使其规模越来越大;同时为了节省成本,整个集群中的存储节点大多采用可靠性较低的设备,而且新旧节点之间差异明显,这必然导致节点故障越来越频繁<sup>[3]</sup>,这也给整个分布式存储集群带来了

存储和计算资源的不确定性。上述情况给存储分配和任务调度带来了极大的挑战。因此如何解决分布式存储系统中存储资源的均衡和计算任务的均衡问题,是当前分布式系统的研究的热点之一。

## 1 相关研究

### 1.1 负载均衡

负载均衡<sup>[4]</sup>是分布式模型中一个比较重要的研究内容,主要研究方向包括存储负载均衡和计算负载均衡,其中存储负载均衡解决的是存储资源的均衡分布问题,而计算负载均衡

收稿日期:2017-02-23;修回日期:2017-04-26。 基金项目:四川省科技支撑计划项目(2014GZ0013)。

作者简介:周渭博(1981—),男,山东烟台人,博士研究生,CCF 会员,主要研究方向:大数据; 钟勇(1966—),男,四川岳池人,研究员,博士生导师,CCF 会员,主要研究方向:软件工程、大数据; 李振东(1990—),男,宁夏银川人,博士研究生,主要研究方向:数据挖掘。



衡解决的是计算资源的均衡分布问题。

国内外的相关研究中,对计算负载均衡方面的研究比较多,而存储负载均衡方面的研究比较少。在分布式存储系统中,谈到各存储节点的负载均衡,一般都是以存储空间利用率或其他资源利用率<sup>[5-6]</sup>来判断各存储节点的负载均衡程度,在基于资源利用率的负载均衡算法设计过程中,一般是以所有节点的磁盘空间利用率(Disk space Utilization, DU)相等作为整个分布式存储系统的最佳均衡点;而在实际的应用场景中,新加入集群的节点与原节点相比,新节点的磁盘I/O速率和存储可靠性一般都比原节点要高很多,这种差异导致原节点往往会成为整个分布式存储系统数据读写性能的瓶颈。传统的以磁盘空间利用率的方法来判定整个分布式存储系统的负载均衡,实际上是以损失新节点的数据读写效率为代价的。因此本文尝试从保证数据最佳读写效率的角度提出度量分布式存储系统负载均衡的新思路。

## 1.2 熵

在热力学领域,熵是表征系统内混乱程度的物理量,整个系统中热量总是从高温物体自发地传到低温物体,最终达到热平衡状态,此时系统熵值达到最大值<sup>[7]</sup>。在信息学领域,信息熵常用来衡量一个随机信息量的大小,若信息的不确定性越大,则信息量越大,信息熵就越大;信息的不确定越小,则信息量越小,信息熵也越小<sup>[8]</sup>。

分布式存储系统中存储资源的负载均衡问题从本质上可以理解为系统内部存储的均衡问题。文献[9]通过分析分布式系统与热力学系统的内在相似性,从熵的角度探讨了分布式系统均衡负载问题与能量的关系;文献[10]利用最大熵和熵增原理构建了熵优化模型,并提出了分布式系统中资源动态加权评估模型。

## 2 存储熵

本文根据熵和信息熵理论,结合分布式存储系统资源负载均衡问题,提出存储熵(Storage Entropy, SE)的概念和定义,并给出相应的负载均衡算法。

### 2.1 名词约定

1) 数据总量  $DN$ ,即系统中所有节点存储的数据量累加数。设分布式系统中有  $n$  个节点,第  $i$  个节点的存储数据量为  $DN_i$ ,则该系统中的存储数据总量为:

$$DN = \sum_{i=1}^n DN_i \quad (1)$$

2) 磁盘 I/O 速率  $DR$ ,即磁盘平均读写速率。第  $i$  个节点中磁盘的读写速率为  $DR_i$ ,则磁盘平均读写速率为:

$$DR = \sum_{i=1}^n DR_i \quad (2)$$

3) 节点可靠性  $R_i$ ,即第  $i$  个节点中磁盘能够正常运行的概率。

$$R_i = 1 - \frac{T_i}{2 * T_{life}} \quad (3)$$

磁盘属于电子设备,其失效率符合“浴盆曲线”<sup>[11]</sup>,考虑到磁盘的早期失效期时间比较短,为了简化计算,将其与恒定失效期合并,并通过式(3)的方法近似计算,此时可以理解为磁盘出厂未使用时其可靠性为 100%,当达到硬件的使用寿命

时,其可靠性为 50%,当达到 2 倍的硬件寿命时,其可靠性为 0%。其中,  $T_{life}$  表示磁盘寿命(一般为 3 万小时);  $T_i$  表示磁盘已经使用时间。

4) 节点读写时间  $DT_i$ ,即第  $i$  个节点中磁盘可靠的情况下读写数据所需要的时间:

$$DT_i = DN_i / DR_i \quad (4)$$

5) 节点平均读写时间  $\overline{DT_i}$ 。读写第  $i$  个节点的数据时,可能出现读写成功和读写失败两种情况,将该节点的可靠性视为读写成功的概率,则可以计算出第  $i$  个节点的平均读写时间,因此综合考虑磁盘可靠和不可靠两种情况定义读写数据所需要的平均时间:

$$\overline{DT_i} = DT_i * R_i + DT_{max} * (1 - R_i) \quad (5)$$

其中  $DT_{max}$  表示节点不可靠情况下读写数据所需要的时间。

6) 基于读写效率的节点负载率  $L_i$ 。将第  $i$  个节点的数据平均读写时间在整个集群中的比重定义为基于读写效率的节点负载率  $L_i$ 。

$$L_i = \frac{\overline{DT_i}}{\sum_{i=1}^n \overline{DT_i}} = \frac{DT_i * R_i + DT_{max} * (1 - R_i)}{\sum_{i=1}^n (DT_i * R_i + DT_{max} * (1 - R_i))} \quad (6)$$

### 2.2 存储熵定义

分布式存储系统中资源负载均衡的目标是将存储的数据分发到  $n$  个存储节点上,如果采用随机分发的方式,那  $n$  个节点就会有不同的负载情况,需要寻找一种方法来度量这种具有明显不确定性的负载程度。本文根据熵理论给出存储熵的定义,其目的是通过存储熵来度量分布式存储系统中各个存储节点的负载均衡程度。

**定义 1** 在分布式集群系统中,如果系统中有  $n$  个存储节点,则分布式系统的存储熵  $H$  定义为:

$$H = - \sum_{i=1}^n (L_i * \lg(L_i)) \quad (7)$$

存储熵的概念是熵理论在分布式存储系统中资源均衡领域的一种延展,其具有熵的一切性质,根据熵增原理和最大熵原理<sup>[12]</sup>可以证明:存储熵的熵值总是从最不平衡状态的最小值  $H_0$  逐渐增加,直到完全均衡状态的最大值  $H_{max}$ 。通过计算存储熵的熵值,可以简单高效地度量分布式存储系统整体资源负载均衡的程度。

**定理 1** 当  $L_1 = L_2 = \dots = L_n = 1/n$  ( $i = 1, 2, \dots, n$ ) 时,存储熵  $H$  取得最大值,且  $H_{max} = \lg(n)$ 。

**证明** 由式(6)可知:

$$\sum_{i=1}^n L_i = 1 \quad (8)$$

此时问题转变成求  $H$  的条件极值,因此可以构造拉格朗日函数:

$$L = - \sum_{i=1}^n (L_i * \lg(L_i)) + \lambda \left( \sum_{i=1}^n L_i - 1 \right) \quad (9)$$

分别对  $L_i$  ( $i = 1, 2, \dots, n$ ) 求偏导,可以得到:

$$- \lg(L_i) - \frac{1}{\ln 2} + \lambda = 0 \quad (10)$$

由此可以获得拉格朗日函数的驻点  $L_1 = L_2 = \dots = L_n$ ,并将其代入式(8),可以得到  $L_1 = L_2 = \dots = L_n = 1/n$ ,此时  $H$  取得最大值,根据式(7)可以计算出  $H_{max} = \lg(n)$ 。证毕。

根据定义 1 和定理 1 可知,  $H$  的取值范围是  $[0, H_{max}]$ ,因



此可以通过存储熵 $H$ 的大小来判断分布式存储系统中各存储节点的负载均衡的程度:当系统中负载完全均衡时,存储熵的熵值最大,且最大值 $H_{\max} = \lg(n)$ ;当负载不均衡时,存储熵的熵值会逐渐变小,直到等于0。

存储熵函数曲线是一个 $n$ 维的图像,无法形象化描述,文中仅给出 $n=2$ 和 $n=3$ 两种情况下,存储熵的函数曲线。

当 $n=2$ 时, $L_1 + L_2 = 1$ ,令 $L_1$ 为横坐标轴,熵值 $H$ 为纵坐标轴,通过MatLab绘制出存储熵的二维函数曲线如图1所示。

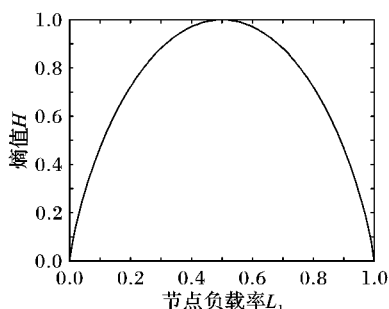


图1 二维熵函数图像

Fig. 1 2D entropy function image

当 $n=3$ 时, $L_1 + L_2 + L_3 = 1$ ,令 $L_1$ 为 $x$ 坐标轴, $L_2$ 为 $y$ 坐标轴,熵值 $H$ 为 $z$ 坐标轴,通过MatLab绘制出存储熵的三维函数曲线如图2所示。

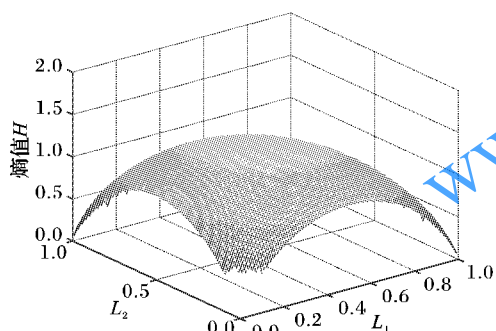


图2 三维熵函数图像

Fig. 2 3D entropy function image

通过对图1和图2的观察可以发现:不论是二维熵函数还是三维熵函数,当所有分量取值相等时函数取得最大值,以此类推, $n$ 维熵函数的最大值应该满足:

$$L_1 = L_2 = \dots = L_n = 1/n \quad (11)$$

### 3 基于存储熵的负载均衡算法

根据上述存储熵的定义,要计算其熵值 $H$ ,要求出所有节点负载率 $L_i$ ,而在节点已知的情况下,要求出该节点的数据读写时间 $DT_i$ 和 $DT_{\max}$ 以及节点可靠性 $R_i$ 。其中, $DT_i$ 可以通过节点存储的数据量 $DN_i$ 和磁盘I/O速率 $DR_i$ 来计算; $DT_{\max}$ 可以通过人工设定的方式确定,比如令 $DT_{\max} = 3DT_i$ ;  $R_i$ 可以根据磁盘已使用时间和磁盘的使用寿命来计算。因此,在存储熵的计算公式中,需要采集的数据包括每个节点存储的数据量 $DN_i$ 和磁盘的已使用时间 $T_i$ ,其他数据都可以通过设备参数(如 $DR_i$ 和 $T_{\text{life}}$ )或人工设定(如 $DT_{\max}$ )来确定。

本文提出的基于存储熵的负载均衡算法就是根据存储熵的熵值判断分布式存储系统中存储负载的均衡程度,当该熵

值小于一定的阈值,则进行负载迁移。该算法包括三个组成部分:判断系统存储负载均衡程度,判断单节点存储负载均衡程度,执行存储负载迁移操作。

#### 3.1 系统存储负载判定

系统存储负载是否均衡的判定是本文算法的关键环节,如果系统存储负载判定是均衡的,则无需进行后续的单点存储负载判定和存储负载迁移工作。由于执行一次存储负载迁移需要占用一定的计算资源和存储资源,并且系统开销比较大,为了避免频繁地进行存储负载迁移操作,所以算法中设置了一个均衡阈值,只有当存储熵的熵值 $H$ 小于该阈值时,表明整个分布式存储系统中存储负载不均衡情况达到了相对严重的程度,需要进行存储负载迁移。

系统存储负载情况判定的过程包括:

- 1) 设定均衡因子 $\alpha$  ( $0 < \alpha < 1$ ),可以据此得到均衡阈值 $H_\alpha = \alpha H_{\max}$ 。
- 2) 收集每个节点存储的数据量 $DN_i$ 和磁盘已经使用时间 $T_i$ ,并根据式(6)计算出每个存储节点的负载率 $L_i$ 。
- 3) 根据式(7)计算出整个分布式存储系统的实际存储熵值 $H$ 。
- 4) 判定系统负载是否均衡。如果 $H \geq H_\alpha$ ,说明整个系统负载均衡程度在可以忍受的范围内,不需要作任何调整;如果 $H < H_\alpha$ ,说明整个系统负载失衡比较严重,需要进行负载迁移。

#### 3.2 单节点存储负载判定

如果整个分布式存储系统中存储资源的负载均衡情况不理想,需要进行存储负载迁移操作,那就需要计算出哪些存储节点是超负载,哪些存储节点是空负载,并按照其存储负载程度构建超载节点有序列表和空载节点有序列表。

单节点负载判定的过程包括:

- 1) 计算每个节点的负载率 $L_i$ 。
- 2) 计算系统平均负载 $L_{\text{avg}}$ 。

$$L_{\text{avg}} = \frac{1}{n} \sum_{i=1}^n L_i \quad (12)$$

- 3) 创建超载节点有序列表和空载节点有序列表。如果 $L_i > L_{\text{avg}}$ ,说明该存储节点超载,就将该存储节点按大根堆序插入到超载节点队列中;如果 $L_i < L_{\text{avg}}$ ,说明该存储节点空载,就将该存储节点按小根堆序插入到空载节点队列中。

#### 3.3 存储负载迁移

存储负载迁移的目的是将负载较多的存储节点上的数据迁移到负载较少的存储节点上,使整个分布式存储系统整体达到资源的负载均衡。

存储负载迁移的具体过程包括:

- 1) 计算每一个超载存储节点需要迁移出的数据量:

$$\Delta DN_i = \left[ \left[ (L_i - L_{\text{avg}}) * \sum_{i=1}^n (DT_i * R_i + DT_{\max} * (1 - R_i)) - DT_{\max} * (1 - R_i) \right] * DR_i \right] / R_i \quad (13)$$

- 2) 计算每一个空载存储节点需要迁移进的数据量:

$$\Delta DN_i' = \left[ \left[ (L_{\text{avg}} - L_i) * \sum_{i=1}^n (DT_i * R_i + DT_{\max} * (1 - R_i)) - DT_{\max} * (1 - R_i) \right] * DR_i \right] / R_i \quad (14)$$





3) 根据存储节点需要迁移的数据量,按序循环迁移数据,具体算法流程如图 3 所示。

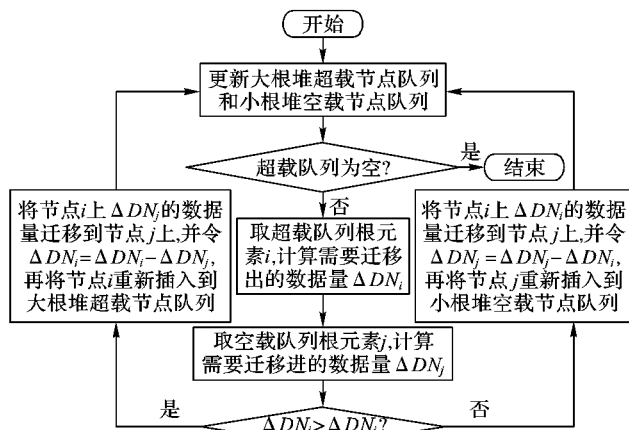


图3 数据迁移流程

Fig. 3 Data migration process

基于存储熵的负载均衡算法按照系统存储负载判定、单节点存储负载判定和存储负载迁移三个步骤调整分布式存储系统中各个存储节点间存储的数据量,以达到均衡存储负载的目的。这种追求全局均衡的机制可以有效控制各节点间的资源均衡程度,能够最大化地提高整个分布式存储系统的数据读写效率。

## 4 实验与分析

### 4.1 实验环境

为了验证算法的正确性和优越性,利用虚拟机搭建一个由双硬盘共 20 个节点构成的 Hadoop 集群进行测试。整个 Hadoop 集群由 2 个 NameNode 节点和 18 个 DataNode 节点组成,其中 8 个 DataNode 节点位于硬盘 A 上,其他节点位于硬盘 B 上,每个节点均为内存 512 MB,硬盘 8 GB,操作系统为 CentOS 6.5, Hadoop 版本为 2.6.0。

硬盘 A 与硬盘 B 分别通过鲁大师硬件检测工具进行测试,测试结果显示硬盘 A 与硬盘 B 的数据传输率分别是 482 MB/s 和 574 MB/s,可靠性分别是 78.12% 和 93.03%。

表 1 硬盘测试数据

Tab. 1 Hard disk test data

磁盘类型	数据传输率/(MB·s <sup>-1</sup> )	已工作时间/h
A:西数	482	13 128
B:希捷	574	4 184

### 4.2 实验设计与分析

实验中设置 Hadoop 集群的副本数量为 1,依次上传 5 个测试文件,每个测试文件的大小分别是 5.75、11.5、17.25、23 和 28.75 GB,根据下面提出的两种不同的实验方案,分别使用 Hadoop 自带的基准测试软件包测试其文件读操作所耗时间。

方案 1 基于磁盘空间利用率的负载均衡算法(简记为 DUA)<sup>[13]</sup>。

将整个测试文件分成的 Block 平均分配到 18 个 DataNode 中,无法平均的 Block 按照随机顺序存储到对应的 DataNode 中,测试读取整个文件所用时间。

方案 2 基于存储熵的负载均衡算法(简记为 SEA)。

按照基于存储熵的负载均衡算法将测试文件存储到硬盘 A 和硬盘 B 对应的 DataNode 中,测试读取整个文件所用时间。

方案 1、2 文件读取时间的对比结果如表 2 所示。

表 2 文件读取时间对比

Tab. 2 Comparison of file reading time

文件大小/GB	文件读取时间/s		文件大小/GB	文件读取时间/s	
	DUA	SEA		DUA	SEA
5.75	112.69	83.49	23.00	312.60	253.88
11.50	177.74	146.20	28.75	387.98	312.60
17.25	253.88	202.99			

表 2 中两种算法读取文件的时间开销均是经过多次重复实验所获得的测试数据均值,实验过程中,基于磁盘利用率的负载均衡算法采用 Hadoop 系统默认的数据均衡阈值(10%),基于存储熵的负载均衡算法也设置数据均衡阈值为 10%,该测试数据中包括了数据均衡和数据迁移所需要的时间开销。

上述两种算法各存储节点存储量和各存储节点存储空间利用率对比结果分别如图 4 和图 5 所示。

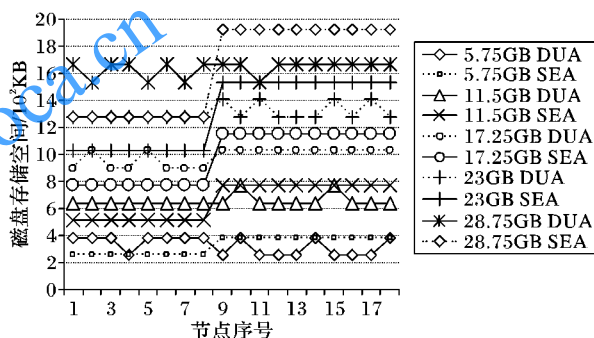


图 4 节点存储量对比

Fig. 4 Comparison of node storage

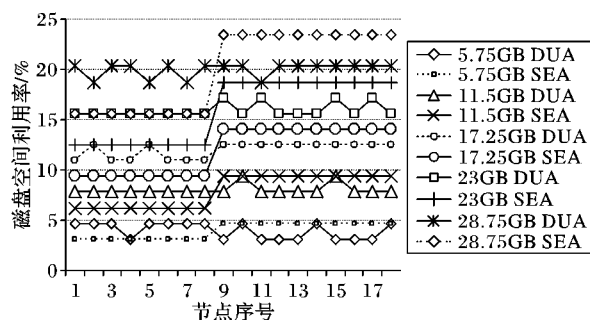


图 5 磁盘空间利用率对比

Fig. 5 Comparison of disk space utilization

对比方案 1、2 的实验结果可以明显看出:基于存储熵的负载均衡算法在各存储节点磁盘 I/O 速率差异较大的分布式存储系统中,其数据读取效率比基于磁盘空间利用率的负载均衡算法高很多。在磁盘空间利用率方面,基于存储熵的负载均衡算法不如基于磁盘空间利用率的负载均衡算法,但实际上,这是因为上述实验中每个 DataNode 节点的硬盘空间都被固定地分配了 8 GB,如果将硬盘 B 所对应的硬盘空间调整为 12 GB,则两种算法磁盘空间利用率对比如图 6 所示。

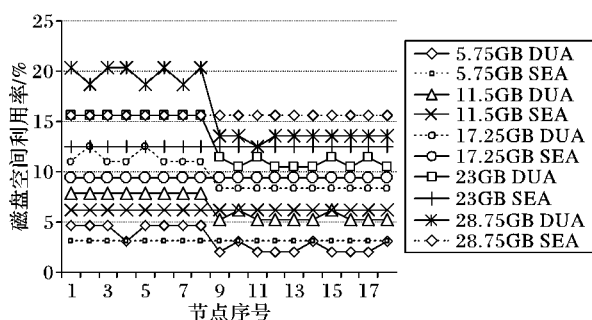


图6 磁盘调整后的空间利用率对比

Fig. 6 Comparison of space utilization after disk capacity adjustment

通过图5和图6的对比可以看出:调整硬盘B所对应的存储节点的磁盘空间大小后,不但使得整个存储系统读取文件的效率达到最优,而且也能保证整个存储系统磁盘空间利用率最优。

在实际应用场景中,随着海量数据的不断增长,分布式存储系统往往需要通过横向扩展的方式不断增加新的存储节点,这些新加入节点的磁盘I/O速率和可靠性一般都比原有存储节点高,如果盲目地增加新节点,有可能造成资源浪费。本文提出的基于存储熵的负载均衡算法追求数据读写效率最优,当存储熵达到最大值时,集群中所有存储节点的数据读写时间应该是一致的,因此可以根据节点负载率计算公式来推导和计算新加入节点最合适的存储容量,使之与原有存储节点匹配,从而保证整个存储系统达到配置最合理。

## 5 结语

针对分布式存储系统中的存储均衡问题,本文从数据读写效率的角度提出了基于存储熵的负载均衡算法,并将该算法与基于磁盘空间利用率的负载均衡算法进行了对比实验,验证了本文算法可以有效地提高整个分布式存储系统的读写性能。事实上,如果考虑到磁盘空间利用率的问题,可以根据磁盘I/O速率和可靠性,利用节点负载率计算公式计算并调整新节点的磁盘容量,这样既可以保证整个分布式存储系统的读写效率,也能够充分利用磁盘空间资源,该方法可以在集群规模扩展的过程中,解决新加入节点和原有节点配置差异大的问题。不过,本文提出的基于存储熵的负载均衡算法主要讨论和解决分布式存储系统整体的存储均衡问题,目的是实现分布式存储系统整体的读写效率最优,并不能实现单个文件的读写效率最优。

## 参考文献 (References)

- [1] 陈吉荣,乐嘉锦. 基于Hadoop生态系统的大数据解决方案综述[J]. 计算机工程与科学, 2013, 35(10): 25-35. (CHEN J R, LE J J. Reviewing the big data solution based on Hadoop ecosystem[J]. Computer Engineering & Science, 2013, 35(10): 25-35.)
- [2] 文莎. 分布式文件系统综述[J]. 软件导刊, 2015, 14(11): 27-29. (WEN S. Survey on distributed file system[J]. Software Guide, 2015, 14(11): 27-29.)
- [3] 黄昌勤,李源,吴洪艳,等. 云存储系统中数据副本服务的可靠性保障研究[J]. 通信学报, 2014, 35(10): 89-97. (HUANG C Q, LI Y, WU H Y, et al. Modeling and maintaining the reliability of data replica service in cloud storage systems[J]. Journal on Communications, 2014, 35(10): 89-97.)
- [4] 董继光,陈卫卫,吴海佳,等. 基于动态副本技术的云存储负载

- 均衡研究[J]. 计算机应用研究, 2012, 29(9): 3422-3424. (DONG J G, CHEN W W, WU H J, et al. Load balancing study in cloud storage based on dynamic replica technology[J]. Application Research of Computers, 2012, 29(9): 3422-3424.)
- [5] 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. (LIU K, XIAO L, ZHAO H Y. Research and optimize of cloud data load balancing in Hadoop[J]. Microelectronics & Computer, 2012, 29(9): 18-22.)
- [6] 罗鹏,龚勋. HDFS数据存放策略的研究与改进[J]. 计算机工程与设计, 2014, 35(4): 1127-1131. (LUO P, GONG X. Research and improvement of data placement strategy for HDFS[J]. Computer Engineering and Design, 2014, 35(4): 1127-1131.)
- [7] CHENG X, CHEN Q, HU G, et al. Entropy balance for the closed system undergoing thermodynamic processes[J]. International Journal of Heat and Mass Transfer, 2013, 60: 180-187.
- [8] DE VOS A. The expectation value of the entropy of a digital message[J]. Open Systems & Information Dynamics, 2002, 9(2): 97-113.
- [9] 董静宜,王鹏,陈磊,等. 云计算集群系统负载均衡算法的熵判定值[J]. 成都信息工程学院学报, 2010, 25(6): 580-583. (DONG J Y, WANG P, CHEN L, et al. The entropy value of load balance algorithm about cloud computer cluster[J]. Journal of Chengdu University of Information Technology, 2010, 25(6): 580-583.)
- [10] 左利云,曹志波,董守斌. 云计算虚拟资源的熵优化和动态加权评估模型[J]. 软件学报, 2013, 24(8): 1937-1946. (ZUO L Y, CAO Z B, DONG S B. Virtual resource evaluation model based on entropy optimized and dynamic weighted in cloud computing[J]. Journal of Software, 2013, 24(8): 1937-1946.)
- [11] 马纪明,万蔚,曾声奎. 基于浴盆曲线故障率函数的FFOP预计方法[J]. 航空学报, 2012, 33(9): 1664-1670. (MA J M, WAN W, ZENG S K. FFOP prediction method based on bathtub-shaped failure rate function[J]. Acta Aeronautica ET Astronautica Sinica, 2012, 33(9): 1664-1670.)
- [12] 耿志强,姬威,韩永明,等. 基于维度最大熵数据流聚类的异常检测方法[J]. 控制与决策, 2016, 31(2): 343-348. (GENG Z Q, JI W, HAN Y M, et al. Data stream clustering algorithm based on the maximum entropy of data dimension and its applications for anomaly detection[J]. Control and Decision, 2016, 31(2): 343-348.)
- [13] 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. (LIU K, NIU W L. An improved data load balancing algorithm for Hadoop[J]. Journal of Henan Polytechnic University (Natural Science), 2013, 32(3): 332-336.)

This work is partially supported by the Science and Technology Support Plan of Sichuan Province (2014GZ0013).

**ZHOU Weibo**, born in 1981, Ph. D. candidate, His research interests include big data.

**ZHONG Yong**, born in 1966, Ph. D., professor. His research interests include software engineering, big data.

**LI Zhendong**, born in 1990, Ph. D. candidate, His research interests include data mining.