一种应用于大规模存储系统的数据分布算法

郑 胜 李 通
(武汉工程大学电气信息学院 武汉 430205)

摘 要 随着大数据时代的到来,PB级、EB级甚至ZB级数据集出现,存储系统的建设需要根据业务的发展,逐渐进行扩展。不同性能存储设备的加入、旧设备的退出以及多设备同时失效等问题的出现对传统存储系统数据分布算法提出严峻挑战。设计了一种新的hash映射算法,该算法引入节点权重和多副本,并考虑节点失效和节点过载情况,能够适应存储系统扩容、节点失效、节点过载的动态环境。该算法能从概率上保证系统伸缩时,数据对象及其副本分布在不同的节点上,以及在节点间保持概率上分布的均衡性和迁移数据量最优;针对系统运行过程中节点失效和节点过载,该算法也进行了有效处理,提高了系统的可用性和性能。通过数学分析和实验验证了该分布算法自动适应存储系统的伸缩变化,保证了数据分布均匀性和对节点失效和过载的有效处理。

关键词 分布式文件系统,在线扩展,数据映射,数据迁移
中图法分类号 TP301.6 **文献标识码** A

Data Placement Algorithm for Large-scale Storage System

ZHENG Sheng LI Tong
(College of Electrical and Electronic Engineering, Wuhan Institute of Technology, Wuhan 430205, China)

Abstract With the era of big data coming, t PB and EB even ZB-level dataset makes storage system scalable. Traditional data distribution algorithm was confronted with serious challenge because of different performance storage devices added and the old ones quitted, even multiple devices failed simultaneously. A new hash mapping algorithm was proposed which supports the node weight and multi-replica and also considers node failure and node overload. The algorithm can adapt dynamically to change of storage nodes and promises data even distribution probabilistically for different performance nodes. Besides, the one can effectively deal with node failure and node overload which can improve the availability and performance of the system.

Keywords Distributed file system, Scalability, Data placement, Data migration

1 引言

数据集的高速增长,使得人们越来越关注大规模存储系统的问题。与磁盘阵列这种小规模存储系统不同,大规模存储系统是一个需要运行多年、逐步建设和扩展的动态系统,必须允许不同性能节点的加入、退出,同时大规模存储系统包含成千上万的不同性能的存储节点,节点失效和节点过载是一种常态。因此,大规模存储系统中数据对象的分布、迁移和定位成为研究的热点和难点。这里数据对象可以是传统文件系统的块,也可以是面向对象存储的文件系统中的存储对象。

文件系统中数据对象的分布有两种方式,一种是以元数据映射表的方式,分布和记录数据对象的分布;另一种是以映射函数的方式,分布和定位数据对象。传统文件系统采用第一种方式,但是当系统达到PB级、数据对象达到10^9、映射表将达到GB级时,元数据的修改和查找效率十分低下,不能适应大规模存储系统扩展的需要。采用映射函数的方式分布和定位数据对象,只需少量信息就可计算出数据对象分布的位置,极大地减少了元数据量并提高了数据对象查找的效率。因此国内外都在积极研究和开发这类数据映射方法。

这类映射算法的基础是以对数据对象的ID号通过hash运算来映射数据对象存放的位置,但是增加或减少节点时,系统中的所有数据对象必须重新进行映射,这将导致大量的数据迁移。SCADDAR方法[1]和LH* 算法[2]在存储节点同构的环境下,系统扩展后,能够移动尽量少的数据对象,但这类算法没有考虑节点性能差异以及为了提高可靠性数据需要多副本情况下数据的均衡分布和迁移问题。刘仲[3]提到的动态区间映射的方法,如果不考虑节点失效造成的区间分割的增加,就能够有效地限制区间链表的长度。然而,节点失效在大型存储系统中每天都有可能发生,因此将造成区间链表过大,使得空间开销和检索链表的时间开销过大,而且在元数据服务器和成千上万的客户端之间保持区间链表的一致性比较困难。Honicky[4,5]提出的RUSH算法和穆飞[6]借鉴一致性hash提出的算法,支持异构环境下的数据映射,考虑了存储节点的权重差异以及数据对象多副本分组映射,能在系统节点数变化时迁移的数据对象概率上保证迁移最优。但是这类算法在数据映射过程中出现节点失效以及节点过载时都没有

郑 胜(1980—),男,博士,讲师,CCF会员,主要研究方向为分布式系统、网络存储,E-mail:zhengsheng2006@gmail.com;李 通(1989—),男,硕士,主要研究方向为网络存储。

进行相关的处理。

本文基于 hash 映射算法提出了一种数据分布算法，它支持异构环境下的多副本分节点均匀映射，并将失效节点进行标记保留在系统中，在访问到该节点上的数据时，才将该数据对象进行迁移，将重建工作分摊到各次访问中，而不是集中在一个时段，避免了节点失效引起的大量的数据迁移和重建所需要花费大量的时间，同时当客户端访问的数据对象映射到的存储节点负载过大时，在客户端通过映射算法，将访问调度到含有对象副本、热度较低的存储节点上，有效地解决了热点数据迁移引发的网络负载和数据分布不均匀问题。

2 算法的相关定义

定义 1 对象 ID 是标识数据对象的全局唯一标识符。ID 为 x 的数据对象用 x 表示。

定义 2 r 表示对象 x 的副本编号，R 表示对象 x 的副本个数，$r\in[0,R-1]$。

定义 3 存储系统中数据对象用集合 $X=\{x_0,x_1,\cdots\}$ 表示，$|X|$ 表示存储系统中数据对象个数。X 的子集用 X_i 表示，$|X_i|$ 表示子集中数据对象个数。

定义 4 存储系统中全体存储子集的集合用 S 表示，$|S|$ 表示存储系统中存储子集的个数。在按需扩展的集群存储系统中，$|S|$ 对应扩容的次数。

定义 5 存储节点的性能差异用权重 w 来表示，对应存储节点的权重集合 $W=\{w_0,w_1,\cdots,w_n\}$。

3 算法设计

系统扩展过程中，会不断加入不同特性的存储设备。新的存储设备通常具有更快的速度和更大的存储容量。为了更充分地利用资源，应该在这些设备上分配更多的数据对象，因此，根据存储节点的处理器速度、内存大小和存储容量等方面，综合考虑为不同性能的存储节点设置不同的调节因子权重 w。权重可以根据具体情况具体对待。例如，如果存储节点其他方面性能相近，但容量相差较大，我们就可根据容量来设置这个权重。

大规模存储系统的节点失效是常态，必须采用一些冗余机制提高系统的可靠性。通常方法是副本机制和纠删编码算法[7]。纠删编码算法采用生成校验块的方式，提供对节点故障的冗余，和副本机制相比，在提供相同的可靠性时，可有效节省存储空间，但是出现节点故障时，需要进行恢复运算，效率比副本机制低，所以对于性能要求很高的文件系统通常采用副本机制。在支持副本的数据映射算法中，除了需要考虑全体数据对象（主本和副本）分布的均匀性、在线扩展迁移数据量，还必须考虑主本和副本之间、副本和副本之间不能放置在同一存储节点上。

RAID5 算法随着磁盘的容量达到 TB 级，出现磁盘故障时，重建需要 1 天，甚至更长时间，使系统的可靠性大为降低。大规模存储系统中，存储节点成千上万，节点失效经常发生，存储节点容量甚至高达数十 TB，数据映射算法必须考虑有效减少重建所花费的时间。

由于热点数据的存在，使得分布式存储系统各个节点存在负载不均衡的情况。并行磁盘系统内部磁盘和数据块的热度采用启发式的方法将热点数据迁移到热度较低的磁盘来均衡磁盘间的访问[8]。这类方法如果热点数据热度很大，把这些热点迁移到热度小的磁盘上时，又使这些迁入热点数据的磁盘热度很高，而且数据的迁入、迁出会造成网络拥塞，也破坏了数据分布的均匀性。

根据以上分析，我们设计如下算法：

输入：对象的 ID

输出：客户端可以访问的对象的位置

1. n_i：第 i 次扩容前已有的设备数
2. m_i：第 i 次扩容新加入的设备数
3. r：副本的编号，$0\leqslant r\leqslant R-1$，R 为最大副本数
4. c：扩容次数的累计值
5. i←c
6. for r←0 to R−1
7. while i≠0
8. {srand(OID)

▷为随机数发生器的种子赋值

$n_i'\leftarrow\sum_{j=0}^{i-1}w_jm_j$

9. $m_i'=w_im_i$

▷选择一个和 $n_i'+m_i'$ 互质的数，而且 $p\leqslant n_i'+m_i'$

10. z =random(i)

▷ 随机数发生器跳跃 i 步，直接产生第 I 个随机数 z，满足 $0\leqslant z<(n_i'+m_i')$

11. $o_r\leftarrow(z+r*p)\bmod(n_i'+m_i')$
12. if failed(o_r)
13. $r'=r'+1$
14. r=r−1
15. else if ($o_r\geqslant n_i$) and (! overload(o_r))
16. return o_r

▷ overload 函数判断 o_r 节点是否过载

17. else
18. i←i−1
19. end for

4 算法证明与特性分析

4.1 算法证明

4.1.1 主本与副本放置分析

定理 1 对任意正整数 a 和 n，如果 $d=\gcd(a,n)$[9]，则在模 n 的加法群 Z_n 中，由 a 生成的子群和由 d 生成的子群有以下性质：

1) $\langle a\rangle=\langle d\rangle=\{0,d,2d,\cdots(\frac{n}{d}-1)d\}$

2) 由 a 生成的子群的阶数为 $|\langle a\rangle|=\frac{n}{d}$

证明：由扩展的 EDCLID(a,n)[9]可生成满足 $ax'+ny'=d$ 的整数 x' 和 y'，因此，$ax'\equiv d(\bmod n)$，因此 $d\in\langle a\rangle$。

由于 $d\in\langle a\rangle$，因此 d 的所有倍数均属于 $\langle a\rangle$，故有 $\langle d\rangle\subseteq\langle a\rangle$。

对于任意的 m 属于 $\langle a\rangle$，则有某个整数 x 满足 $m=ax \bmod n$，所以对某个整数 y 有 $m=ax+ny$。同时，$d|a$ 而且 $d|n$，则对于任意的整数 x 和 y 都有 $d|ax+ay$，所以 $d|m$，$m\in\langle d\rangle$，并且 $\langle a\rangle\subseteq\langle d\rangle$。

由此可得：

$\langle a\rangle=\langle d\rangle$

因为在0到$n-1$之间恰有n/d个d的倍数，所以$|\langle a\rangle|=n/d$。

故命题得证。

定理2 由式$a(k)=ka \bmod n$(k为整数)，生成序列：$a(1),a(2),a(3),\cdots,a(1),a(2),a(3),\cdots$，其周期为$T=|\langle a\rangle|$。

证明：因为$a(|\langle a\rangle|)=e$，所以对于任意k有$a(|\langle a\rangle+k|)=a(|\langle a\rangle|)\oplus ak=ak$，故$a(k)$的周期为$|\langle a\rangle|$。

由于p和$n_i'+m_i'$互质，则$\gcd(p,n_i'+m_i')=1$

由定理1和定理2可知，

1)$\langle p\rangle=\langle 1\rangle=\langle 0,1,\cdots,n_i'+m_i'\rangle$

2)$|\langle p\rangle|=n_i'+m_i'$

对于任意一个$r\in[0,m_0)$，这里我们要求$R\leqslant m_0$，因此$r*p \bmod n_i'+m_i'$映射区间$[0,n_i'+m_i']$上的不同值。

4.1.2 *数据分布均匀性和迁移的数据量分析*

设r的取值区间为$[0,R-1)$，R小于m_0，则可以根据r的值将全部数据对象分为R类，分别为主本类、第一副本类、第二副本等等。

当$r=0$时，$v=z \bmod(n_i'+m_i')$，其数据分布的均衡性和迁移的数据量最优证明如下：

·数据迁移量分析

假定每次扩容的存储设备性能相同。在第i次扩容前，n_i个设备中数据对象的总数为$|X_i|$，扩容后设备总数为n_i+m_i，要使扩容到n_i+m_i后数据分布保持均匀和一致性，则至少要迁移$(|X_i|*m_i)/(n_i+m_i)$个数据对象到新加入的m_i个设备中。

设对象x映射到$[n_i,n_i+m_i)$记为事件A，对象映射到$[0,n_i-1]$记为事件$\overline{A}$，每个对象的映射相互独立，n_A是$|X_i|$个对象映射中事件A的发生次数。映射到$[0,n_i-1]$上的对象不发生迁移，因此n_A也即是本算法中发生迁移的数据对象。每次映射中A事件发生概率为$\frac{m_i}{n_i+m_i}$，则由贝努利大数定理可得：

$$\lim_{|X_i|\to\infty} p\{|\frac{n_A}{|X_i|}-\frac{m_i}{n_i+m_i}|<\varepsilon\}=1 \tag{1}$$

故在$|X_i|$比较大时，本算法中迁移的数据量n_A依概率收敛于$(|X_i|*m_i)/(n_i+m_i)$，数据的迁移从概率上保证了最优。

·数据分布均匀性分析

对于数据分布的均衡性，我们采用数学归纳法来证明：

1)当n_i为0时，数据对象通过映射函数映射到m_0个存储设备中，因为每个对象映射到每个存储设备的概率相等，为$\frac{1}{m_0}$，由式(1)知，数据对象均匀分布在m_0个存储设备中。

2)假定第i次扩容，n_i+m_i个设备中数据分布保持均匀和一致性。对于第$i+1$次扩容，扩容前的$n_{i+1}=n_i+m_i$中的每个设备数据分布是均匀的，即每个设备中分布的对象数近似为$|X_{i+1}|/(n_i+m_i)$。对于每个设备上这$|X_{i+1}|/(n_i+m_i)$个对象，我们假设映射到$[n_{i+1},n_{i+1}+m_{i+1})$的事件，记为事件$A$，映射到区间$[0,n_{i+1}-1]$的事件记为$\overline{A}$，每个对象的映射相互独立，$n_A$是$|X_{i+1}|/(n_i+m_i)$个对象映射中事件$A$的发生次数，每次映射中事件$A$发生的概率为$m_{i+1}/(n_{i+1}+m_{i+1})$，则由贝努利定理可得：

$$\lim_{|X_{i+1}|\to\infty} p\{|\frac{n_A}{|X_{i+1}|/(n_i+m_i)}-\frac{m_{i+1}}{n_{i+1}+m_{i+1}}|<\varepsilon\}=1 \tag{2}$$

由式(2)可知在$|X_{i+1}|$比较大时，对于n_i+m_i设备中的任何一个，在加入新的节点时，迁移的数据量依概率收敛于：

$$\frac{|X_{i+1}|}{n_i+m_i}*\frac{m_{i+1}}{n_{i+1}+m_{i+1}}$$

故每个设备上分布的对象数量通过式(3)计算。

$$\frac{|X_{i+1}|}{n_i+m_i}-\frac{|X_{i+1}|}{n_i+m_i}\times\frac{m_{i+1}}{n_{i+1}+m_{i+1}}$$

$$=\frac{|X_{i+1}|}{n_i+m_i}\times\frac{n_{i+1}}{n_{i+1}+m_{i+1}} \tag{3}$$

$$n_{i+1}=n_i+m_i \tag{4}$$

联立式(3)和式(4)求解可得，当$|X_{i+1}|$非常大时，每个设备上分布的对象数目从概率上收敛为$\frac{|X_{i+1}|}{n_{i+1}+m_{i+1}}$，因此数据分布从概率上始终保证了均衡性。

任意一副本类k属于$[1,R-1)$，$v=z+k*p \bmod(n_i'+m_i')$，因为k与$n_i'+m_i'$互质，则由定理1可知$z\in\langle p\rangle$并且$k*p\in\langle p\rangle$，故可将v表示为：$v_k=p^i\oplus p^j=p^{i+j}$，又由定理2可知：$p^{i+j}\in\langle p\rangle$，所以可以将$v_k$记为$p^l$，$l\in[1,n_i'+m_i')$。对于任意给定的质数$p$，$v_k$记为$p^{i+J}$，$l=i+J$或者$l=i+J-T$，因为$z$为一等概率的随机变量，即可推知$i$为一等概率随机变量，所以$l$为一等概率随机变量，故$v_k$为一等概率的随机变量。因此，其数据分布的均衡性和迁移的数据量最优的证明同上。

设主本总数为N，对每类对象来说，在每个设备上分布的数量为$\frac{N}{n_i'+m_i}$，故每个设备分布的对象总数为$\frac{NR}{n_i'+m_i}$。对每类对象来说从原有的n_i'个设备迁移到m_i'个新设备的对象个数为$N(\frac{1}{n_i'}-\frac{1}{n_i'+m_i})$，故迁移的总数据量为：

$$RN(\frac{1}{n_i'}-\frac{1}{n_i'+m_i})。$$

4.2 算法特性分析

在考虑到系统中存储节点权重差异的前提下，该算法中仅仅只有失效节点上的数据被移动到其他节点上，使得迁移数据量最优，而且对于在失效节点上数据对象不论是主本或者是副本，映射到其他$n_i'+m_i'-1$个节点上的概率为$\frac{1}{n_i'+m_i'-1}$，设失效节点上主本和副本的总数为$N_{failiure}$，由贝努利定理可知从失效节点迁移到每个节点上的数据对象数依概率收敛于$N_{failure}*\frac{1}{n_i'+m_i'-1}$，因此每个节点分布的存储节点保持了概率上分布的均衡性，从而使系统的负载达到均衡。该算法将失效节点进行标记并保留在系统中，在访问到该节点上的数据时，才将该数据对象进行迁移，这样有效地将重建工作分摊到各次访问中，而不是集中在一个时段进行重建，避免大量的数据迁移和重建花费大量的时间。除此之外，算法实现了当客户端访问的存储对象映射到的存储节点热度过大时，将客户端的访问调度到含有对象副本、热度较低的存储节点上，避免了热点数据的迁移所引发的问题。

5 实验验证

我们根据文中算法设计程序，模拟生成海量数据对象以及将数据对象通过该算法映射到存储节点上，验证算法的相关特性。

5.1 数据对象分布

实验环境是通过随机数发生器产生 100000 个对象 ID，通过文中算法初始分布到 10 个节点上，每个对象副本的个数为 3 个，然后分两次加入新的节点组，一次加入 5 个，其权重为初始节点的 1.5 倍，一次加入 10 个，其权重为初始节点的 2 倍。图 1 给出原始 10 个节点的数据分布情况，图 2 给出了扩展 5 个权重为 1.5 的 OSD 的数据分布情况，图 3 给出了增加 10 个权重为 2 的 OSD 的数据分布情况。

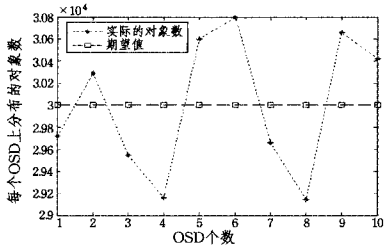

图 1 原始 10 个节点的数据分布情况

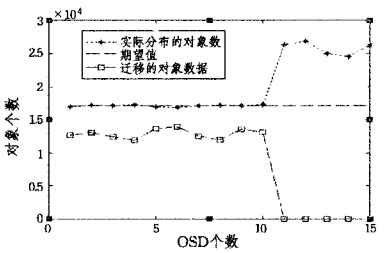

图 2 扩展 5 个权重为 1.5 的 OSD 的数据分布情况

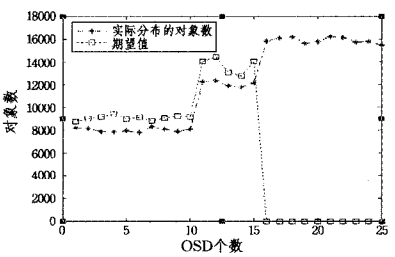

图 3 增加 10 个权重为 2 的 OSD 的数据分布情况

5.2 过载节点负载调度

为了测试我们系统能否有效地处理过载节点，我们将文献[10]中的算法和本文算法进行实验对比。实验中 20 个客户端对 6 个存储节点中的对象发起请求，对某一数据对象的请求按泊松分布来模拟。对于本文算法，我们还需要一个采集 OSD 节点负载信息的采集器，采集器每隔 5 秒收集一次各个节点的信息。图 4 给出了两种算法在 I/O 请求的平均响应时间对比。由图可以看出没有采用负载均衡的算法，响应时间波动比较大，主要原因是当产生热点对象时，请求的队列过长，使得响应时间变长，从而使平均响应时间出现较大波动；采用本文算法有效地处理了热点目录，使得平均响应时间趋于稳定。

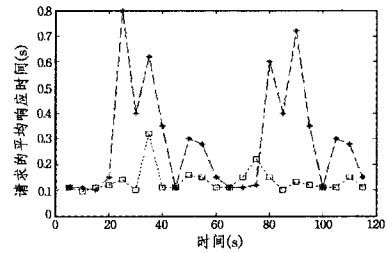

图 4 文献[10]算法和本文算法访问 OSD 的平均响应时间对比

结束语 本文讨论了一种适应于大规模系统扩展过程中节点性能差异以及保证可靠性时，引入多副本情况下的数据分布算法，该算法在保证概率上分布均衡性和迁移最优的情况下，还能够支持存储节点权重的差异，以及同一副本不能分布在同一节点上的要求。对于节点失效这类情况，我们不是进行简单的重建，而是利用我们算法的特性，将重建分摊到访问该节点上数据的时刻，有效地提高了系统的可用性。除此之外，客户端访问的数据对象映射到的存储节点负载过大时，在客户端通过映射算法，将访问调度到含有对象副本、热度较低的存储节点上，有效地解决了热点数据迁移引发的网络负载和数据分布不均匀问题。

参 考 文 献

[1] Goel A, Shahabi C, Yao D S, et al. SCADDAR: An efficient randomized technique to reorganize continuous media blocks [C]// Proc of the 18th Int Conf on Data Engineering(ICDE 02). Piscataway, NJ: IEEE, 2002: 473-482

[2] Litwin W, Risch T. LH * g: a high-availability scalable distributed data structure by record grouping[J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(4): 923-927

[3] 刘仲，周兴铭，等. 基于动态区映射的数据对象布局算法[J]. 软件学报，2005，16(11)：1886-1893

[4] Honicky R J, Miller E L. A fast algorithm for online placement and reorganization of replicated data[C]// Dongarra J, ed. Proc. of the 17th Int'l Parallel & Distributed Processing Symp. Nice: IEEE Computer Society, 2003

[5] Honicky R J, Miller E L. Replication under scalable hashing: A family of algorithms for scalable DecentRalized data distribution [C]// Proceedings of the 18th International Pallel & Distributed Processing Symposium. Santa Fe, NM, 2004

[6] 穆飞，薛巍，舒继武，等. 一种面向大规模存储系统的数据副本映射算法[J]. 计算机研究与发展，2009，4603：492-497

[7] 罗象宏 舒继武. 存储系统中的纠删码研究综述[J]. 计算机研究与展，2012，49(1)：1-11

[8] Peter S, Gerhard W, Peter Z. Data partitioning and load balancing in parallel disk systems[J]. The VLDB Journal, 1998(7): 48-66

[9] 潘承洞，潘承彪，等. 初等数论(第三版)[M]. 北京：北京大学出版社，2013

[10] 郑胜，郝毫毫. 基于贝努力大数定律的数据分布算法[J]. 计算机工程，2009，10(19)：59-61