

按序选取的并行存储系统数据分布方式研究

张虎, 伍卫国, 董小社, 钱德沛, 戴罗庚
(西安交通大学计算机科学与技术系, 710049, 西安)

摘要: 针对采用轮转方式的并行存储系统在增加新存储节点后的重均衡操作中移动数据量大、资源消耗高的问题, 提出了按序选取的数据分布方式. 它通过多次模拟数据重均衡操作来完成并行存储系统的数据分布, 从而使得系统更易于节点扩展. 基于此, 又提出了优化按序选取方式, 这样可以通过改变数据抽取规则来提高按序选取方式的访问并行度. 理论分析和模拟实验表明, 采用所提方式的并行存储系统, 数据重均衡操作均可达到零无效移动率, 从而最大程度地减少数据的移动量, 降低系统的资源耗费.

关键词: 并行存储系统; 数据分布; 重均衡; 轮转方式; 按序选取

中图分类号: TP302.8 **文献标识码:** A **文章编号:** 0253-987X(2007)08-0899-04

Data Layouts of Parallel Storage System Based on Ordinal Selection

Zhang Hu, Wu Weiguo, Dong Xiaoshe, Qian Depei, Dai Luogeng
(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Aiming at the problem that when new storage nodes are added to a parallel storage system by using Round-Robin method, the rebalancing operation needs to move a large quantity of data among storage nodes and therefore consume plenty of system resource, a new data layout called ordinal-selection was presented. The data layout of parallel storage system was completed by using rebalancing operations of simulation data repetitiously so as to make the nodes extension of the system more easily. Furthermore, an optimized ordinal-selection layout was also presented to increase the system's parallel degree by changing the data selection rules. The theoretical analysis and experimental results show that for a parallel storage system employing one of these two data layouts, the rebalancing operation can achieve zero ineffectual-move-ratio, therefore the remove of the data can be decreased farthest and the system resource consumption is decreased.

Keywords: parallel storage system; data layout; rebalance; round-robin; ordinal selection

并行存储系统通常采用轮转方式(MF)完成从逻辑数据块到存储节点的映射, 如独立磁盘冗余阵列子系统以及大多数的并行文件系统. 轮转方式在数学上称之为取模函数, 应用取模函数作为数据分布方式的并行存储系统, 当其节点数目增加时, 必须经过重均衡操作, 将所有节点上的原有数据重新分布, 以保证取模函数的一致性, 保证新加入的节点存储空间和服务能力充分利用. 这种重均衡操作需要

耗费大量的 I/O 带宽和计算能力, 甚至需要停止数据服务. 为此, 有些系统, 如 GPFS^[1]、Zebra^[2], 均在后台提供在线数据重均衡操作, 但由于数据量巨大, 所以依然会影响到数据服务.

本文提出了 2 种新型数据分布方式, 消除了并行存储系统在重均衡操作中的无效数据移动, 最大程度地减少了重均衡操作的数据量, 有效地保证了并行存储系统的容量均衡和负载均衡.

1 理想数据分布方式

以优化重均衡操作为目标,通过抽象一个模型来描述具有零无效移动率的理想并行存储系统的数据分布方式.在该模型中,将数据分布方式看作是一个从逻辑数据块号到存储节点号的哈希函数 H ,若存储系统有标号为 $0, 1, \dots, N-1$ 的 N 个节点,逻辑数据块的标号为 s ,则 H 可定义为

$$H(N, s) = r, \quad (r \in \{0, 1, \dots, N\}, s \in \{0, 1, 2, \dots\}) \quad (1)$$

当新增 n 个节点之后,系统节点总数为 $N+n$,新加入的节点标号则分别为 $N, N+1, \dots, N+n-1$. 作为一个理想的数据分布方式,其函数 H 必须满足以下 3 个条件.

(1) 各节点间存储容量均衡,即函数 H 必须保证数据块被散列到任意一个存储节点上的概率是均等的. 因此,对于给定的 N ,应满足

$$\forall r, \quad r \in \{0, 1, \dots, N-1\}$$

$$\sum_l P(l) = \frac{1}{N}, \quad l \in \{L \mid H(N, L) = r\} \quad (2)$$

(2) 满足最大并行度. 依照文献[3-4]的数据分布方式优劣判断准则知,一个存储系统只有满足最大并行度,才能获得最好的性能. 因此,最大并行度也是理想数据分布方式的要求之一. 然而,只有轮转方式才能满足最大并行度,并且可以证明,任何满足最大并行度的数据分布方式本质上都等同于轮转方式.

(3) 满足零无效移动率. 通过对轮转方式的考察发现,重均衡操作在所移动的数据块中,只有其中一部分被移动到了新增节点上,其他的则在原有节点间相互移动. 数据移动产生了数据流量,但对新节点与原有节点之间的容量均衡来说它不起作用,可视作无效移动. 当增加的节点数目与原节点数目不存在倍数关系时,那么总数据移动中绝大部分是无效的数据移动. 因此,如果能最大限度地削减无效移动,就能使得重均衡操作中的数据移动量最小.

定义 1 在重均衡操作中,无效移动产生的数据量占移动数据总量的比率称之为无效移动率. 零无效移动率则表示,在节点增加后的数据重均衡操作中,数据块只从原有节点移动到新增节点,原有节点间不产生数据移动,因而数据重均衡操作所移动的数据量应该是最少的. 满足零无效移动率的函数

$$H(N+n, s) = \begin{cases} H(N, s) \\ N, N+1, \dots, N+n-1 \end{cases} \quad (3)$$

从式(3)可以看出,在新增节点之后,对于任意数据块 s , $H(N+n, s)$ 的取值要么等于 $H(N, s)$, 要么落在 $[N, N+n-1]$ 区间.

同时满足以上 3 个条件的数据分布方式才可称之为一个理想的零无效移动率的数据分布方式. 可以证明,条件(1)是条件(2)的前提,而条件(3)与条件(2)是不可能同时满足的(除非对新增节点数目加以限定^[5]). 所以,同时满足上述 3 个条件的数据分布方式是不存在的,应该分别从尽可能小的无效移动率和尽可能接近 N 的并行度这 2 个目标来寻找次优解.

2 按序选取数据分布方式

基于第 1 节给出的数学模型和寻找次优解的思路,本文提出了 2 种完全满足条件(1)和条件(3),但不满足条件(2)的数据分布方式,即按序选取(OS)和优化按序选取(IOS).

2.1 按序选取

在重均衡操作中,若能保证每增加一个节点的零无效移动率,就可以实现增加任意数目节点的零无效移动率. 按序选取的数据分布方式就是按照这样的思路设计的:一个采用按序选取方式的并行存储系统,若其有 N 个存储节点,那么从一次数据块号到存储节点号的映射过程,可以看作是存储节点数目从 1 开始,每次增加一个存储节点的 $N-1$ 次数据重均衡的过程,而每次数据重均衡操作都保证只从每个原有节点的数据块中按照规则抽取一定比例的数据块移动到新增节点之上,但原有节点间不移动数据块.

采用的抽取规则是将每个原有节点的数据块按顺序编号,设定一个抽取间隔,抽取间隔的值与当前系统节点的数目相同,然后依照当前的编号每间隔抽取一个数据块并移动到新增节点之上. 移动到新增节点上的数据块则以同间隔抽取的为一组,再以原有节点号大小排序,生成新的顺序号. 因此,从一次数据块号到节点号的映射过程实质就是计算该数据块在 $N-1$ 次重均衡操作后的最终所在节点位置的过程. 按序选择分布方式如图 1 所示.

从图 1a 中可以看出,原系统有一个节点,又增加了一个节点,则当前节点数目为 2,因此抽取间隔为 2. 然后,在 Node0 的数据块中,每 2 个数据块抽取其中的一个移动到新增节点 Node1 上. 在图 1b 中,抽取间隔为 3,则在 Node0 和 Node1 上每 3 个节

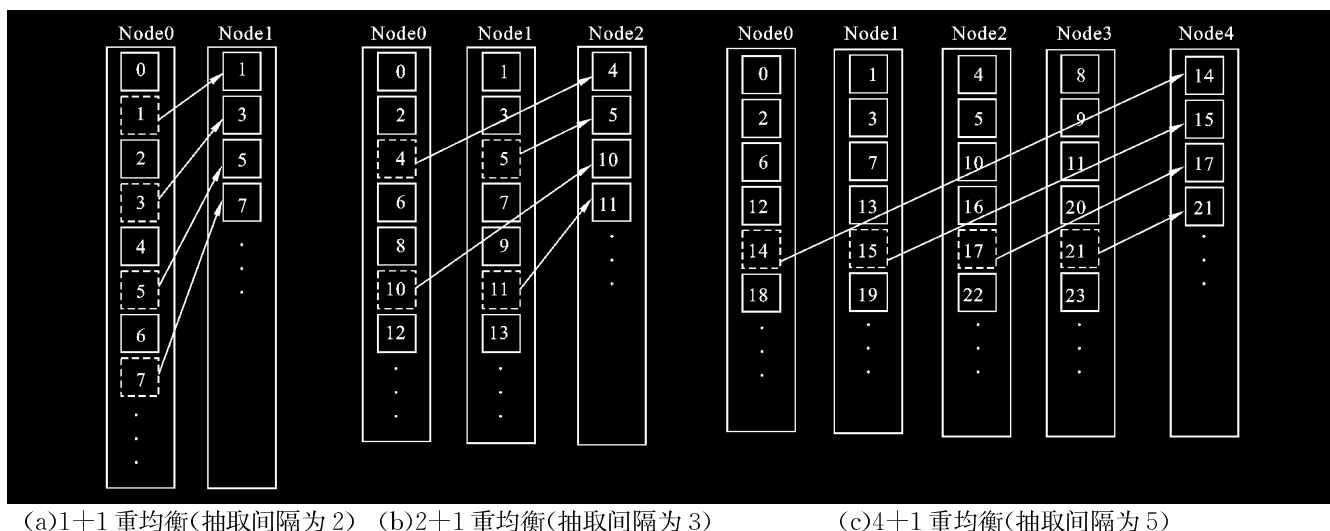


图1 按序选择分布方式图示

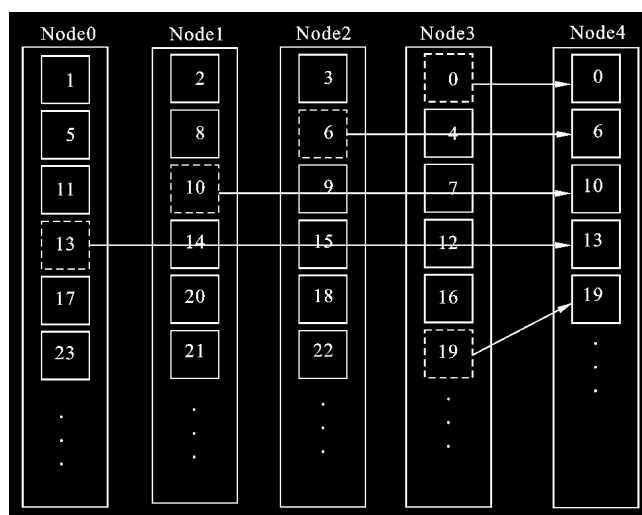
点抽取一个数据块移动到 Node2 上,因此顺序号同为 3 的数据块 4 和数据块 5 分别移动到 Node2 上,那么新顺序号分别为 0、1,顺序号同为 6 的数据块 10、数据块 11 则在 Node2 上的新编号分别为 2、3. 在图 1c 中,节点数量从 4 增加到 5,那么以抽取间隔为 5 来完成重均衡操作.

2.2 优化按序选择

通过观察按序选择方式发现,在这种方式下,很容易形成逻辑编号相连的 2 个数据块分布在同一个节点之上,根据并行度^[3]的定义,这样的数据分布会影响访问并行度.为此,应采用优化按序选取方式来解决序号选取问题,即在一个抽取间隔内沿对角线抽取数据块,抽取间隔的取值与按序选择方式是一致的,如图 2 所示.节点 Node0 在第 1 个间隔内抽取序号为 3 的数据块 13,节点 Node1 抽取序号为 2 的数据块 10,依次类推,则 Node3 抽取序号为 0 的数据块 0.抽取到新节点后的数据块,按照原所属节点与新增节点的距离排序,距离越近序号越小.因此,从 Node3 上抽取的数据块 0 在 Node4 上的新序号为 0,而从 Node0 上抽取的数据块 13 在 Node4 上的新序号为 3.可以看出,优化按序选取方式减少了连续数据块同处于一个节点的情况,从而其并行度要优于按序选择方式.

3 测试与分析

为了验证 2 种按序选取方式的效能,分别从数据块的分布是否均衡、数据重均衡操作的无效移动率大小、映射效率高低以及并行度等 4 个方面进行分析,并与轮转方式进行比较.



4+1 重均衡(抽取间隔为 5)

图2 优化按序选取方式图示

3.1 存储容量均衡

在按序选取方式的映射过程中,模拟的 $N-1$ 次数据重均衡操作均选择当前节点数目为抽取间隔,这保证了在每次重均衡后所有节点的数据块数目大致均等,也使得 2 种按序选取方式具有周期性.周期为从节点数 1 到节点数 N 的所有数的最小公倍数,在一个完整周期内,所有节点上的数据块数目相等.因此,根据式(2),2 种按序选取方式可保证存储节点间的容量均衡.

3.2 无效移动率

由于零无效移动率是本文中的 2 种分布方式的设计目标,这 2 种方式就是依照不产生无效移动的重均衡操作来分布数据的,因此这 2 种方式均可完全达到零无效移动率.

以 4 个节点的并行虚拟文件系统(PVFS)^[6]为测试平台,测试了 2 种按序选取方式下的数据重均

衡操作的开销. 测试方法为:生成 100 GB 数据,分别以轮转方式和 2 种按序选取方式作为数据分布方式,并测试增加节点后的数据重均衡时间. 当系统新增 1 个节点时,应用 2 种按序选取方式的重均衡时间比轮转方式的重均衡时间约减少了 75%;当增加 2 个节点时,重均衡时间约减少 50%,与减少的数据量成正比,这与无效移动率的分析结果相吻合.

3.3 映射过程的性能

对于某些资源有限的并行存储系统来说,从逻辑数据块号到存储节点标号的映射过程的时间和空间消耗至关重要. 映射过程如果过多地占用 CPU 和内存空间,将会直接导致系统性能恶化. 2 种按序选取方式均模拟 $N-1$ 次重均衡操作, N 为系统内的节点数目,因此映射过程的复杂度为 $O(N)$,所耗时间与系统内的节点数目呈线性关系,节点数目越大,映射所需要的时间就越长. 考虑到这 2 种方式均具有周期性,可事先计算一个完整周期的映射结果,后续的映射过程可用查表法完成. 因此,2 种按序选取方式的映射效率均不会成为数据访问的瓶颈.

3.4 并行度分析

对于并行存储系统,并行度是系统性能的关键,而数据分布方式将直接影响数据访问的并行度. 根据文献[3]中最大并行度的定义,可编写一个模拟程序来测定这 2 种按序选取方式的并行度,并与轮转方式比较,结果如图 4 所示.

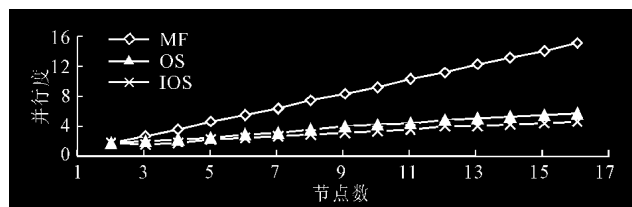


图3 3种方式下系统的平均并行度比较

从图 4 可以看出,轮转方式可以达到最大并行度,而本文的 2 种选取方式均在并行度上有一定的损失,其中优化按序选取方式的并行度要优于按序选取方式. 在实际应用中,2 种按序选取方式的并行度损失并不像图 4 所示的那样,这是因为:①模拟程序的假定过于极端,在真实环境中较难遇见;②真实系统的具体参数和数据访问模式具有多样性. 比较常见的、由对等节点组成的 PVFS 的测试结果显示这 3 种数据分布方式下的性能差距并不大,而对于某些系统,如硬件的 RAID 子系统,按序选取方式的并行度会受到较大影响,此时可采用文件间并行等其他途径来消减并行度带来的性能损失.

4 结 论

本文通过研究并行存储系统在节点扩展时的数据重均衡操作问题,提出了无效移动率的概念以及达到零无效移动率的可能途径,并从改变并行存储系统数据分布方式的思路出发,提出了按序选取和优化按序选取的数据分布方式. 分析和测试表明,所提方式均能达到 0 无效移动率,使得数据重均衡操作所移动的数据量为最小,同时还可以有效地保证存储空间均衡. 但是,这 2 种方式均是以牺牲一定程度的访问并行度为代价的,其中优化按序选取方式的并行度优于按序选取方式. 由于理论分析的假设条件过于极端,所以在实际应用中并行度损失并不像理论分析所显示的那样. PVFS 系统的实测表明,本文的这 2 种数据分布方式具有一定的实用价值.

后续工作将主要研究具体的并行存储系统如何消减和隐藏因并行度导致的性能损失,寻找并行度更好的数据分布方式.

参考文献:

- [1] Schmuck F, Haskin R. GPFS: a shared-disk file system [C]//Proceedings of the Conference on File and Storage Technologies. Monterey, USA: USENIX, 2002.
- [2] Hartman J, Ousterhout J. The zebra striped network file system [J]. ACM Transactions on Computer Systems, 1995, 13(3):274-310.
- [3] Holland M, Gibson G. Parity declustering for continuous operation on redundant disk arrays [C]//Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM Press, 1992.
- [4] Alvarez G, Burkhard W, Stockmeyer L, et al. Declustered disk array architectures with optimal and near-optimal parallelism [C]//Proceedings of the 25th Annual ACM/IEEE International Symposium on Computer Architecture. New York: ACM Press, 1998.
- [5] 刘军, 杨学军, 王俊伟, 等. 动态盘阵 D/H 分布与基于控制理论的在线重构 [J]. 软件学报, 2005, 16(5): 1028-1038.
Liu Jun, Yang Xuejun, Wang Junwei, et al. D/H placement and on-line data reorganization based on control theory in dynamic disk array [J]. Journal of Software, 2005, 16(5):1028-1038.
- [6] Carns H, Ligon B, Ross B, et al. PVFS: a parallel file system for Linux clusters [C]//Proceedings of the 4th Annual Linux Showcase and Conference. New York: ACM Press, 2000. (编辑 苗凌)