多层一致性哈希的 HDFS 副本放置策略①

席 屏, 薛 峰

(江苏科技大学 计算机科学与工程学院, 镇江 212003)

摘 要: 分布式文件系统 HDFS 采用机架感知的副本放置策略在一定程度上保证了数据的可靠性, 但系统运行一段时间后会出现数据分布不均衡的情况. 虽然使用 Balancer 程序可以对数据进行重分布, 但对数据存储不均衡处理的后置性影响了系统的数据读取速率和可靠性. 采用多层一致性哈希的副本放置策略, 首先通过一致性哈希算法获得数据副本对应的机架位置, 再通过一致性哈希算法获得该机架下对应的数据节点位置并最终成为存储位置. 一致性哈希算法在查找对应位置的过程中采用地址等分和虚拟节点的技术, 提高了查找的效率和分布的均衡性. 该策略在数据均衡存储、上传速率方面较原有策略都有很大的提高, 并且具有数据自适应性的能力.

关键词: 一致性哈希; HDFS; 副本放置; 存储均衡; 自适应性

Replica Placement Strategy Based on Multi-layer Consistent Hashing in HDFS

XI Ping, XUE Feng

(School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003 , China)

Abstract: The HDFS distributed file system, with RackAwareness replica placement strategy, ensures the reliability of the data in a certain extent. But the data distribution will be unbalanced after the system runs for a period of time. Although the usage of Balancer program could redistribute tha data, the postposition of unbalanced treatment of the data storage affects the data read rate and reliability of the system. This paper adopts a replica placement strategy, which is based on multi-layer consistent hashing. At first, we will get the position of the frame which corresponds to replica through the consistent hashing algorithm, and then with the consistent hashing algorithm, we will get datanode position which is under the frame, finally, becoming the storage location. Consistent hashing algorithm uses the equal-sized partitions technology and the virtual node technology in the process of searching the corresponding position, which improves the search efficiency and the balance of distribution. The strategy, used in the data equilibrium storage and the upload rate, has greatly improved than the original one. Besides, it has the ability of replicas adaptability.

Key words: consistent hashing; HDFS; replica placement; storage equilibrium; adaptability

分布式存储是当前应对海量数据存储管理的重要技术之一. HDFS(Hadoop Distributed File System)是分布式系统基础架构 Hadoop 的核心子项目, 由于搭建硬件要求不高, 并且具有数据高传输率和高容错性, 是解决大数据集应用不错的分布式存储技术. HDFS 的数据放置策略在选择数据节点时采用随机方式, 易造成数据分布的不均衡, 影响系统的整体性能. 针对这一问题提出了许多解决方案: 选择数据节点时计算所有节点当前的使用情况, 选择使用率最低的节点进行存储[1]; 或者综合考虑节点存储性能和网络拓扑距离, 比较选择最优数据节点[2]. 但是在选择数据节点前扫描所有集群比较判断的方式增加了数据存储的资源消耗和时间, 没有兼顾到数据均衡和系统性能两个方面.

一致性哈希是分布式哈希表协议的一种实现, 最早是在分布式 cache 里面提出的, 现在更多应用在分布式存储和 p2p 系统[3]. 本文综合一致性哈希算法和

① 收稿时间:2014-05-13;收到修改稿时间:2014-06-20

DHFS 数据副本分散存储原则，采用两次一致性哈希算法来定位数据节点. 在保证同一数据的多个副本存储在不同机架的同时延续了哈希算法快速定位和均衡分布的优势. 同时面对集群或节点的增加删除，一致性哈希算法的存储拓扑可以实现自适应的数据分发和转移，保证数据的安全性和可靠性. 此外，结合一致性哈希的改进技术，采用创建虚拟节点和等分存储拓扑的方法[4]，进一步优化了一致性哈希的分布均衡和数据分发、转移效率.

1　HDFS数据放置策略

1.1 默认放置策略

HDFS 对数据采用分块存储，默认采用机架感知的数据放置策略，采用冗余技术对数据块进行多份备份，同一数据块的多份副本保存在多个不同机架的数据节点上，每个数据节点只保存同一数据块的一份副本，在不超过负载阀值的前提下数据节点的选择采用随机的方式.

放置策略流程为：客户端上传的数据块优先考虑本地节点存储，若客户端本地不是数据节点，在集群中随机选择一个节点进行存储；随后在远程机架上随机选择一个节点存储数据块；接着判断前两份副本的数据节点是否属于同一个机架，如果属于就在远程机架中随机选择一个节点存储数据块，如果不属于就在第一份副本所在机架的其他节点中随机选择一个存储数据块；其余的数据备份都在集群中随机选择一个数据节点进行保存. 随机选择的数据节点在保存之前都会检查节点是否活动，以及节点负载是否超过设定负载阀值. HDFS 默认一个数据块有三份副本，按照上述流程描述，三份副本存储情况如图 1 所示:

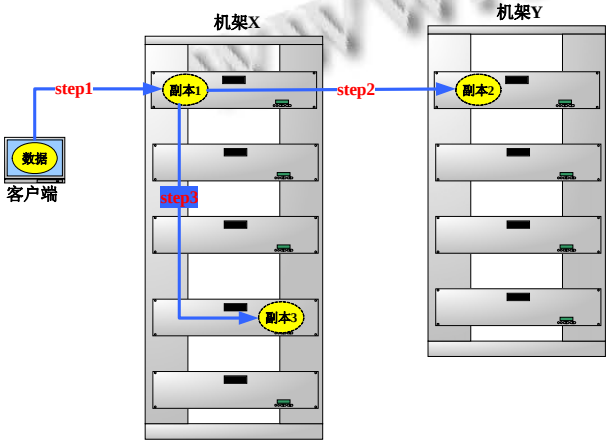

图 1　默认副本放置策略

1.2 默认放置策略的问题

HDFS 的默认放置策略权衡了可靠性、数据存储率以及分布均衡等多方面因素，但随机选择数据节点的方式容易产生以下问题:

① 易造成数据分布不均衡. 默认放置策略试图通过随机方式使数据均衡分散，但对系统中灵活增加和退出的数据节点无法自动调整数据分布，动态情况下系统整体分布并不均衡;

② 名称节点内存资源消耗与数据量呈线性增长关系. 数据块与数据节点的对应关系需要在名称节点(主节点)中进行记录，记录的对应关系为方便快速查询在系统启动时需要加载到名称节点内存. 随着数据量不断增加，名称节点内存资源的消耗会逐渐增加，同时大量记录会降低查找数据效率;

③ 默认放置策略中没有考虑数据节点性能的差异. 集群中数据节点的性能存在差异，以同构为前提对数据副本进行存储，性能高的数据节点无法充分利用，性能低的数据节点容易负载过重，影响整体性能.

针对第一点提出的无法自适应调整数据的问题，HDFS 采用均衡器(Balancer)程序对数据进行转移，最终达到数据分布均衡. 但该程序需要手动触发，对不均衡调整具有后滞性. 调整过程中将负载过重数据节点的数据转移到负载过轻的数据节点，涉及数据节点多且转移数据量大.

2　多层一致性哈希放置策略

2.1 一致性哈希的基本原理

将存储空间抽象成一个环，所有数据节点的唯一标识 key(比如 hostname 或 IP)通过 hash 函数映射到环上，就把存储空间进行了划分，每个数据节点“负责”从其在环上的映射位置沿逆时针方向到下一个数据节点映射位置的区域. 将数据块的标识 key 通过同一 hash 函数映射到环上某个位置，沿顺时针方向找到的第一个数据节点映射位置，即该数据要存放的数据节点[3].

2.2 多层一致性哈希的描述

将集群中所有数据节点按照所属机架进行划分[5]，即二维向量(机架号，数据节点号)可以唯一确定一个数据节点位置. 结合一致性哈希查找对应存储位置的工作原理，首先将集群内所有机架以唯一标识映射到存储空间环，数据块通过 hash 函数映射找到要存储的

机架号, 将该机架的所有数据节点以唯一标识映射到存储空间环, 数据块再次通过 hash 函数映射找到要存储的数据节点号, 最终确定存储位置. 映射过程如图 2 所示:

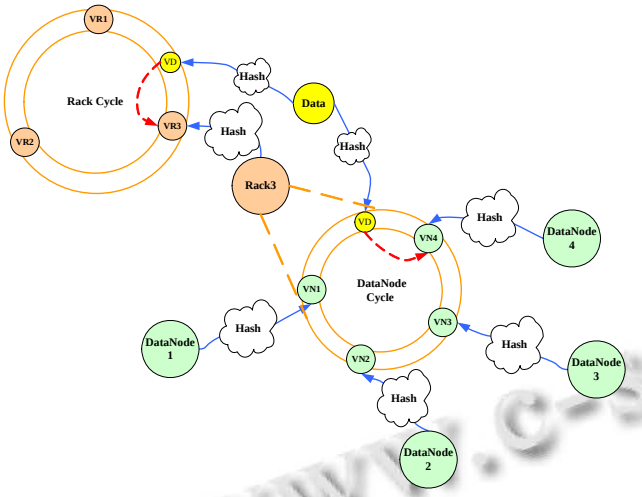

图 2 多层一致性哈希映射过程

直接将机架或数据节点映射到存储空间环, 由于映射少易分布不均匀, 会导致数据分布不均衡. 为机架或数据节点增加虚拟点, 每个机架或数据节点对应多个虚拟点, 将所有虚拟点映射到存储空间环. 在一致性哈希过程中, 当数据查找到某个虚拟点, 即对虚拟点对应的机架或数据节点进行操作. 机架或数据节点对应的虚拟点个数可根据硬件性能(例如: cpu、内存、磁盘、网络等)进行差异化, 在均衡分布数据的同时兼顾了节点性能的因素, 提高系统整体性能. 具体虚拟点个数差异化方法不是本文重点, 在此不做累述.

对存储空间环进行等分, 等分区域个数大于映射到同一存储空间环的所有虚拟点数量总和. 机架或数据节点的虚拟点通过 hash 函数直接映射到存储空间环后, 沿顺时针方向找到的第一个等分点做为该虚拟点在存储空间环的映射位置. 若找到的等分点是其他虚拟点的映射位置, 沿顺时针方向继续查找下一个等分点, 直至找到为止, 这样, 每个虚拟点负责的存储空间环区域由若干等分区域组成. 数据节点按照映射位置所属等分区域的不同对数据分组存储, 当机架或数据节点加入或退出集群时, 根据变化虚拟点涉及的等分区域来决定需要转移和分发的数据范围. 等分存储空间环后, 虚拟点映射和数据存储位置如图 3 所示.

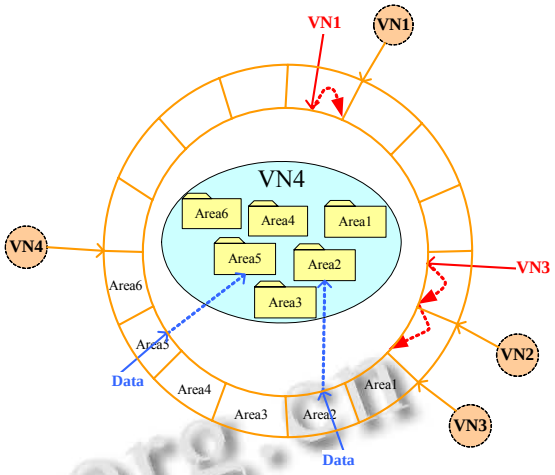

图 3 等分存储空间环中虚拟点映射和数据存储位置

2.3 放置策略相关定义

机架集合 R: $R=\{r_i \mid 1 \le i \le m\}$, r_i 表示第 i 个机架, 其中 m 为集群中机架个数;

数据节点集合 DN: $DN=\{dn_{(i,j)} \mid 1 \le i \le RC, 1 \le j \le n\}$, $dn_{(i,j)}$ 表示第 i 个机架上第 j 个数据节点, 其中 n 为第 i 个机架中数据节点个数, RC 为集群内机架总数, 即 $\| R \| = RC$;

副本集合 BR: $BR=\{br_{(i,j,k)} \mid 1 \le i \le RC, 1 \le j \le DN_i, 1 \le k \le z\}$, $br_{(i,j,k)}$ 表示第 i 个机架第 j 个数据节点的第 k 份副本, 其中 z 为数据块的副本数, DN_i 为第 i 个机架中数据节点总数, 即 $DN_i=\{dn_{(i,*)} \mid 1 \le i \le RC\}$, RC 为集群内机架总数;

存储空间环集合 C: $C=\{c_i \mid 1 \le i \le RC\}$, c_i 表示映射第 i 个机架所有数据节点的存储空间环, CR 表示映射集群所有机架的存储空间环.

2.4 数据放置策略机制描述

为保证数据存储的可靠性, 数据副本尽量存放在不同机架. 数据块 d_j 的所有副本 $\{br_{(i,j,1)}, br_{(i,j,2)}, \ldots, br_{(i,j,k)}\}$ 共 k 个, 通过 hash 函数映射到 k 个不同的机架上进行存储. 首先将数据块 d_j 标识 key 映射到机架存储空间环 CR 上, 沿顺时针找到的第一个机架映射点即第一份副本存储的机架 r_i, 再沿顺时针方向查找 $k-1$ 个不同机架映射点依次存储余下 $k-1$ 份副本, 副本存储的所有机架为 $\{r_i, r_{i+1}, r_{i+2}, \ldots, r_{i+k}\}$.

获得每个机架 r_i 的数据节点存储空间环 c_i，数据块 d_j 标识 key 再通过 hash 函数映射到 c_i 上，沿顺时针找到的第一个数据节点映射点即该副本存储的数据节点 $dn_{(i,j)}$. 基于两次一次性哈希的副本放置策略流程如图 4 所示.

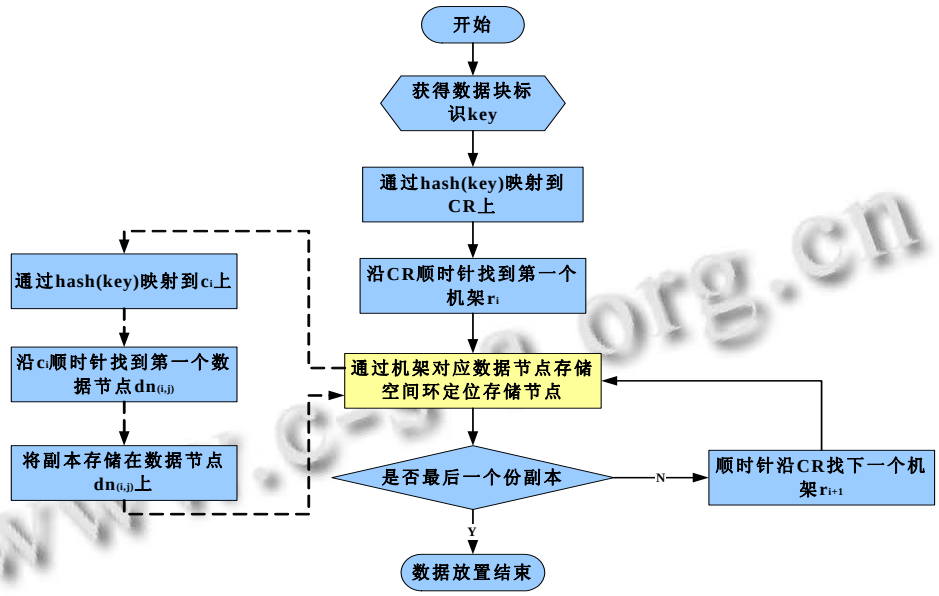

图 4　基于两次一致性哈希的副本放置策略

2.5 数据调整策略机制描述

为了适应集群的灵活可扩展性，当数据节点或机架进入或退出集群时，需要遵循一定策略对数据进行转移和分发，使得调整后的集群数据依然均衡分布. 在集群中可能调整的对象包括数据节点和机架两种，当数据节点进入或退出集群时，只需要将数据在所属机架内进行转移和分发；当机架进入或退出集群时，先确定机架间转移和分发的数据，再确定数据在机架内如何进行转移和分发. 根据两种不同的进入退出对象，有如下两种数据调整策略.

2.5.1 适应数据节点增减的数据调整策略

每个数据块的副本在一个机架上只保存一份，所以一个机架内所有数据节点上的副本可以认为是彼此不同独立存在的. 当一个新数据节点 $dn_{(i,a)}$ 进入机架 r_i，数据节点 $dn_{(i,a)}$ 的标识 key 通过 hash 函数映射到存储空间环 c_i 上某个位置，再查找等分点最终确定在 c_i 的映射点. 以图 5 为例，数据节点 $dn_{(i,a)}$ 的映射点在数据节点 $dn_{(i,1)}$ 和 $dn_{(i,2)}$ 映射点之间，属于数据节点 $dn_{(i,2)}$ 的等分区域 Area1 …… AreaX 将由新数据节点 $dn_{(i,a)}$ “负责”，所有映射到等分区域 Area1 …… AreaX 并保存在数据节点 $dn_{(i,2)}$ 上的副本数据将转移到新数据节点 $dn_{(i,a)}$ 上.

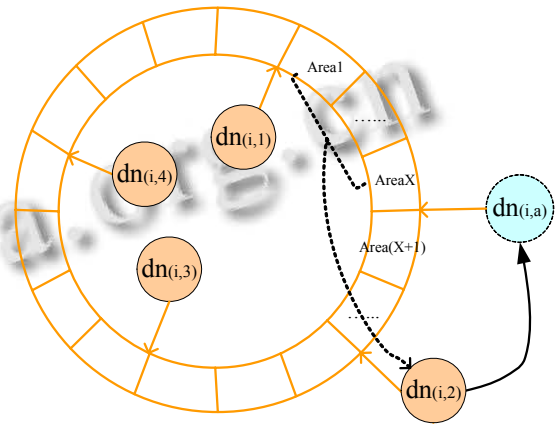

图 5　新增数据节点的数据调整过程

当一个数据节点 $dn_{(i,d)}$ 退出机架 r_i，属于退出数据节点 $dn_{(i,d)}$ 的等分区域将由沿顺时针方向下一个映射点对于的数据节点“接管”. 以图 6 为例，数据节点 $dn_{(i,d)}$ 退出集群，原本属于数据节点 $dn_{(i,d)}$ 的等分区域 Area1 …… AreaX 将由数据节点 $dn_{(i,2)}$ “接管”，在数据节点 $dn_{(i,d)}$ 上映射到等分区域 Area1 …… AreaX 的副本数据将分发给数据节点 $dn_{(i,2)}$.

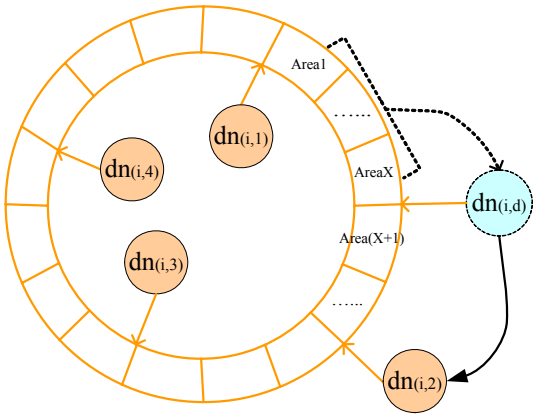

图 6 删减数据节点的数据调整过程

2.5.2 适应机架增减的数据调整策略

一个数据块的 k 份备份会从查找到的第一个机架映射点开始，存储在连续的 k 个机架映射点上，如果一个机架进入或退出集群，会影响映射点位置前 k 个机架或后 k 个机架的数据转移和分发. 机架的数据调整先确定机架间的数据转移和分发，再依据数据节点调整策略对调整机架内的数据节点进行数据备份的转移和分发，在此只对机架间的数据调整策略进行描述，数据节点的数据调整策略不再累述.

新增机架时，当一个机架 r_i 进入集群并映射在 CR 中机架 r_{i-1} 和 r_{i+1} 之间，则映射在机架 r_{i-z} 和 r_{i-z+1}（z 为数据块的备份数)之间等分区域，且通过连续存储 z 份保存在机架 r_{i+1} 的数据备份转移给机架 r_i；映射在机架 r_{i-z+1} 和 r_{i-z+2} 之间等分区域且连续存储保存在机架 r_{i+2} 的数据备份转移给机架 r_i. 以此类推，直到映射在机架 r_{i-1} 和新增机架 r_i 之间等分区域且保存在机架 r_{i+z} 的数据备份转移给机架 r_i，至此，机架间的数据调整完成.

以 HDFS 数据块的默认备份数(3 份)为例. 映射在等分区域 Area1 和 Area2 且存储在机架 r_{i+1} 的数据备份转移给机架 r_i；映射在等分区域 Area3—Area5 且存储在机架 r_{i+2} 的数据备份转移给机架 r_i；映射在等分区域 Area6—Area8 且存储在机架 r_{i+3} 的数据备份转移给机架 r_i，如图 7 所示.

删减机架时，当一个机架 r_i 退出集群，则机架 r_i 中映射在机架 r_{i-z} 和 r_{i-z+1} 之间等分区域的数据备份分发给机架 r_{i+1}；机架 r_i 中映射在机架 r_{i-z+1} 和 r_{i-z+2} 之间等分区域的数据备份分发给机架 r_{i+2}. 以此类推，直到机架 r_i 中映射在机架 r_{i-1} 和删减机架 r_i 之间等分区域的数据备份分发给机架 r_{i+z}，至此，机架间的数据调整完成.

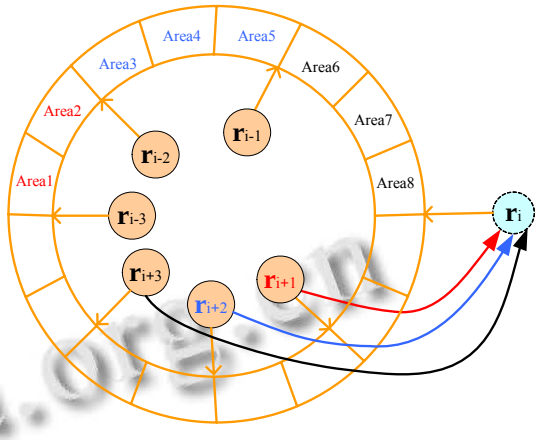

图 7 新增机架的数据调整过程(备份数为 3)

以 HDFS 数据块的默认备份数(3 份)为例. 机架 r_i 中映射在等分区域 Area1 和 Area2 的数据备份分发给机架 r_{i+1}；机架 r_i 中映射在等分区域 Area3—Area5 的数据备份分发给机架 r_{i+2}；机架 r_i 中映射在等分区域 Area6—Area8 的数据备份分发给机架 r_{i+3}，如图 8 所示.

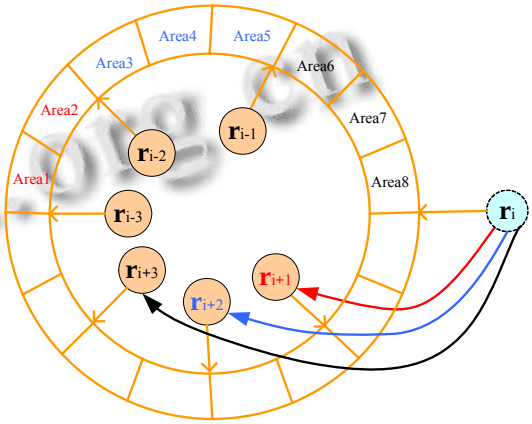

图 8 删减机架的数据调整过程(备份数为 3)

3 实验与结果分析

针对本文提出的多层一致性哈希策略进行均衡性和自适应性的测试. 在局域网中搭建 HDFS 集群，模拟 4 个机架，每个机架分别有 5 个、7 个、10 个、6 个数据节点. 每个数据节点的性能相同. 数据节点的唯一标识采用网卡物理地址+局域网内 IP，映射函数采用基于 MD5 的散列函数 KETAMA. 数据分块大小为

64M, 数据块备份数为 2 份. 上传数据大小为 70G(约 1120 个数据分块).

3.1 均衡性分析

在不考虑数据节点性能差异的前提下, 理论上每个机架分配的备份数与机架内数据节点的个数成正比(机架的数据节点/所有机架的数据节点总和*备份数总和). 图 9 显示了每个机架实际备份数占比与理论备份数占比的对比情况, 从图上来看每个机架获得的备份数与理论值偏差不大, 基本达到了机架间的数据均衡分布.

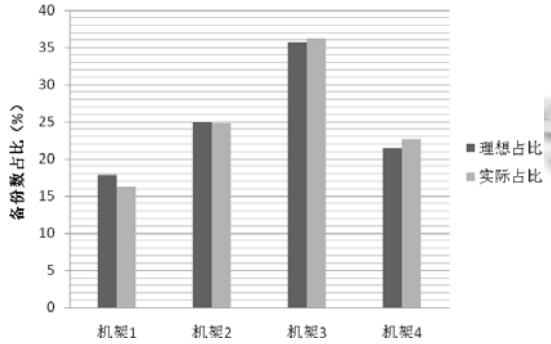

图 9　机架间备份分布

选取机架 2 来说明机架内数据节点备份数分布情况. 不考虑数据节点性能差异, 理论上数据节点会均分分配给该机架的备份数总和(该机架获得的备份数总和/该机架包含数据节点数). 图 10 显示了每个数据节点实际备份数占比与理论备份数占比的对比情况, 实际值对比理论值控制在 2%上下范围内, 偏差不大, 达到了数据节点间的数据均衡分布.

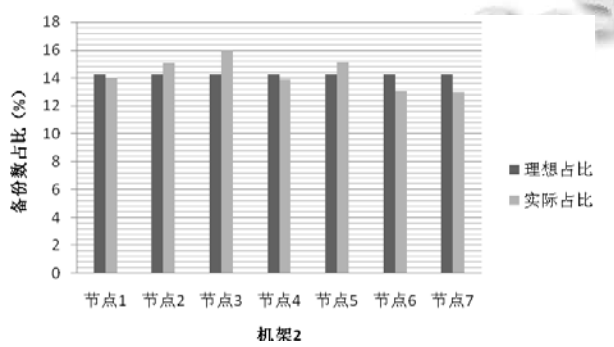

图 10　机架 2 内数据节点间备份分布

3.2 自适应性分析

对机架 2 内数据节点进行增减操作, 先退出数据节点 3, 再加入新数据节点 8.

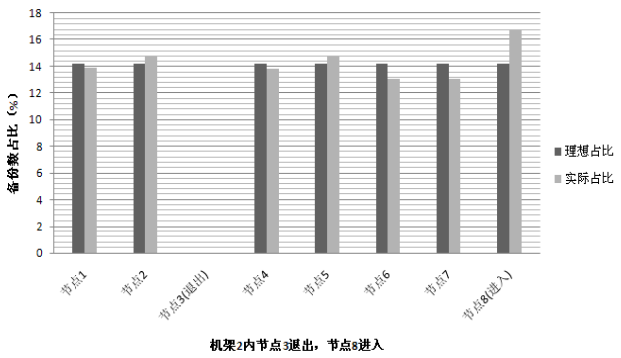

图 11　机架 2 内数据节点增减后备份分布

图中显示数据节点间经过数据转移和分发后实际备份数占比与理论备份数占比的对比情况, 新增节点 8 与理论值偏差最大(约为 2.7%左右), 但没有影响机架内整体的分布均衡.

对集群内新增机架 5, 包含 4 个数据节点.

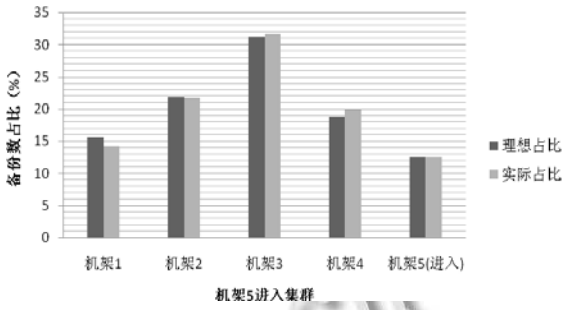

图 12　集群增加机架 5, 包含 4 个数据节点

图中显示机架间经过数据转移后实际备份数占比与理论备份数占比的对比情况, 新增机架 5 与理论值基本一致, 其他机架的偏差也不大, 说明机架间的自适应调整基本保持了数据分布均衡.

4　结语

本文介绍了 HDFS 作为分布式存储被广泛应用的前景, 同时提出默认放置策略存在的问题. 针对问题结合一致性哈希原理提出了基于两层一致性哈希确定数据存储位置的放置策略, 再结合虚拟节点建立和等分区域等技术对一致性哈希进行优化, 在保证数据可靠性的基础上提升了数据的均衡分布, 释放了记录存储地址的资源. 为了适应集群灵活可扩展的实际要求, 制定数据调整策略, 在维持数据分布均衡的前提下, 使数据转移和分发的自适应性成为可能.

参考文献

1 邵秀丽,王亚光,李云龙,等.Hadoop 副本放置策略.智能系统学报,2013,(6):489–496.

2 刘晨光.面向 Hadoop 存储系统的节能优化技术研究[学位论文].武汉:华中科技大学,2012.

3 杨或剑,林波.分布式存储系统中一致性哈希算法的研究.电脑知识与技术,2011,7(22):5295–5296.

4 De Candla G, Hastorun D. Dynamo: Amazon's highly available key-value store. Proc. of the 2lst ACM SIGOPS Symposium on Operating Systems Principles. New York. ACM Press. 2007. 14–17.

5 董继光,陈卫卫,田浪军,等.大规模云存储系统副本布局研究.计算机应用,2012,32(3):620–624.

6 徐婧.云存储环境下副本策略研究[学位论文].合肥:中国科学技术大学,2011.

7 吴昊.基于 HDFS 的分布式文件系统数据冗余技术研究[学位论文].西安:西安电子科技大学,2011.

8 Karger D, Lehman E, Leighton T, et al. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. Proc. of the 29th Annual ACM Symposium on Theory of Computing(STOC'97). New York. ACM Press. 1997.