

DOI:10.3969/j.issn.1673-4785.201210036
网络出版地址: <http://www.cnki.net/kcms/detail/23.1538.TP.20131030.1508.003.html>

Hadoop 副本放置策略

邵秀丽¹, 王亚光¹, 李云龙¹, 刘一伟²

(1.南开大学 信息技术科学学院, 天津 300071; 2.北京大学 数学科学学院, 北京 100871)

摘 要:分布式文件系统(Hdfs)采用随机的副本放置策略使得系统在运行一段时间后会 出现数据分布不均衡的情况,从而降低数据的可靠性和读取速率.为解决 Hdfs 默认副本放置策略存在的问题,对 Hdfs 副本放置策略进行改进:在副本放置选择时优先考虑存储使用率低的节点.模拟实验一测试了机架数目对于算法的影响,结果显示改进后的副本放置策略中,机架数目对集群的均衡性影响很小,显示出较好的均衡性.模拟实验二测试了随着写入数据的增加,比较了使用改进前后的副本放置策略集群中节点使用率的标准差,证实了改进后的副本放置策略在存储均衡方面较原放置策略有着更好的表现.

关键词:云存储;Hdfs;副本放置;存储均衡;存储节点

中图分类号: TP399 **文献标志码:** A **文章编号:** 1673-4785(2013)06-0489-08

中文引用格式:邵秀丽,王亚光,李云龙,等. Hadoop 副本放置策略[J]. 智能系统学报, 2013, 8(6): 489-496.

英文引用格式:SHAO Xiuli, WANG Yaguang, LI Yunlong, et al. Research on the replica placement strategy of Hadoop[J]. CAAI Transactions on Intelligent Systems, 2013, 8(6): 489-496.

Research on the replica placement strategy of Hadoop

SHAO Xiuli¹, WANG Yaguang¹, LI Yunlong¹, LIU Yiwei²

(1. College of Information Technology Science, Nankai University, Tianjin 300071, China; 2. College of Mathematical Sciences, Peking University, Beijing 100871, China)

Abstract: Hadoop distributed file system applies the random replica placement strategy, which causes unbalanced data distribution after the system has operated for a while, resulting in lowering the data reliability and reading speed. In order to eliminate the defect of the replica placement strategy defaulted by the Hdfs, the strategy was improved. When the placement location of a replica is selected, a node with a low storage and use rate will be considered as a priority. The first simulation experiment tested the effects caused by the number of racks on the algorithm. The results show that, for the improved replica placement strategy, the number of racks has little impact on the equilibrium of the group, the equilibrium is excellent. The second simulation experiment compared the standard difference of the node usage rates between the replica placement strategy groups before and after and found an improvement following the increase of the data input. The results verify that the improved replica placement strategy has better performance with respect to storage equilibrium.

Keywords: cloud storage; Hdfs; replica placement; storage equilibrium; storage node

为提高系统的可靠性,解决不可预知的灾难以及硬件错误对系统造成的损失,云存储系统采用分

布式副本技术来存储数据.

哥伦比亚大学的 Ko 等^[1]提出了一种自稳定、全分布、异步可升级的算法来放置副本,算法的目标是在网络中的结点上放置数据对象的多个副本,从网络中的任意一个结点出发都能够通过最短的路径访问到任意的副本;加州大学伯克利分校的 Chen 等^[2]开

发设计了一个动态、高效及可升级的内容分发网络 SCAN (sealable content aeeess network). SCAN 采用 Testry 进行路由和定位,使用沿路缓存算法进行副本放置;德克萨斯大学的 MadhukarR 等提出了一种协作的缓存放置算法^[3],即给定一组协作的缓存、缓存之间的网络距离以及从每个缓存到每个对象的访问频率的预测,决定在哪里放置对象,从而使平均访问开销最小化;Karger 等^[4]提出了能适应节点数量的动态变化的一致性哈希算法,但它只适用于存储节点同构的情况,当节点的存储容量和处理能力有差异时,数据将不能够均匀地分布到系统当中。

云存储系统的典型代表是 Hdfs^[5],它需将每个存储数据块的副本放置在多个机架的多个节点上,存储数据块的副本放置策略将直接影响数据存储的均衡性以及访问数据块的速度.Hdfs 系统采用随机选择节点的副本放置策略,该策略在系统运行一段时间后会造数据分布不均衡的问题,降低数据的可靠性和读取性能.因此,本文提出了基于节点使用率选择存储节点的 Hdfs 副本放置策略的改进算法,引入了客户端存储阈值,允许副本在放置过程中穿越多个机架,以实现各节点数据存储的相对均衡,实验验证了改进策略的有效性。

1 副本放置策略的相关概念

内容为研究 Hdfs 的副本放置策略,先介绍相关概念如下:

1) 获取集群信息: Hdfs 的 NetworkTopology 类实现对其拓扑结构的操纵,该类中包含添加、删除和获取节点信息等函数.比如, Hdfs 通过调用 NetworkTopology 类的 chooseRandom 来随机获取一个节点的信息,通过调用 getNumOfLeaves 来获取所有节点的数目。

2) 集群拓扑(机架与节点): 将 Hdfs 部署在多台服务器上就形成了一个 Hdfs 的集群.如树状拓扑结构的 Hdfs 集群,树根是一个大型交换机,交换机之下可以是多个二级交换机,可以把每一个二级交换机设置为一个机架,每个机架之下连接多个节点。

Hdfs 管理员可编写脚本文件来配置每个节点属于哪一个机架.在进行机架配置时,应将相同交换机下的节点设置为同一个机架就可实现合理的配置。

一般把组成 Hdfs 集群的每一个服务器称为一个节点,对文件读写的客户端而言,其所在节点称为本地节点,其他节点为远程节点.就某一具体节点而言,称该节点所在的机架为本地机架,其他机架为远程机架。

3) 随机函数: Hdfs 的 NetworkTopology 类中有保存所有节点信息的 ArrayList. Hdfs 在选择副本放置位置时,调用随机选择函数 chooseRandom,从 n 中随机选择一个数对应 ArrayList 中的节点就被选中为副本存储的节点.该函数是只有 2 个参数的重载函数,第 1 个参数是选择节点的范围,它可以是某个机架,默认为整个集群;第 2 个参数是不能选择节点的范围,默认为空,可以设置为某个机架。

4) Hdfs 在进行副本选择过程中,有可能出现参数不合格或内存异常等现象,一旦出现运行异常,chooseRandom 函数就会把异常信息返回客户端该函数的调用者。

2 Hdfs 默认副本放置策略

如图 1 所示, Hdfs 的副本放置策略是将每一个数据项的副本放置在多个节点上.在客户端运行的节点上放置第 1 个副本,在客户端的远程机架上随机选择一个节点放置第 2 个副本,在第 2 个副本所在机架上随机选择一个节点放置第 3 个副本。

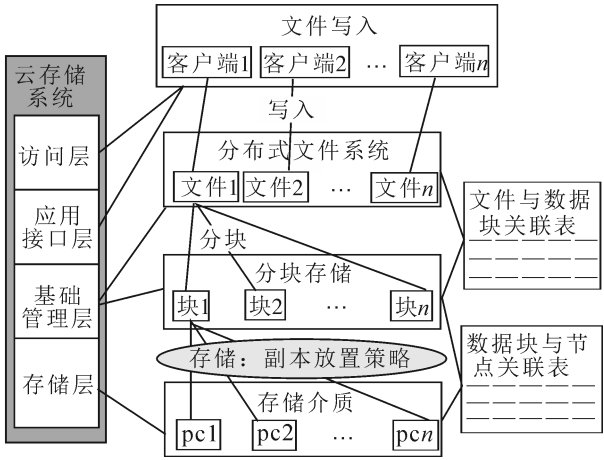


图 1 副本放置策略

Fig.1 The placement policy of duplication

分布式文件系统^[6-7]的副本放置策略确定每一个数据块应该存放的位置,数据块与节点之间的关联被记录在数据块与节点关联表中,数据块最终会被存放在存储层的各个节点上。

2.1 Hdfs 默认副本放置策略的流程

Hdfs 的分块存储文件在选择副本放置位置时,综合考虑了数据存储的可靠性、数据读写的带宽和负载均衡等因素.如将一个数据块所有副本都存储在一个节点上,则存储过程中所占用的带宽是最小的,因为这可以减少数据块的网络传输,但该方案不提供有效的冗余备份,一旦该节点发生故障,则该节点中存

储的这一数据块及其所有副本都会丢失.因此,Hdfs对任意一数据块不在同一个节点上放置多个副本,而是将副本尽可能分散存放^[8-9].图 2 给出了 Hdfs 默认

的副本放置策略流程,其中标注了本文所实现的对副本放置策略的改进工作,Hdfs 默认的副本放置策略选择 3 个节点,可以选择多个节点放置副本.

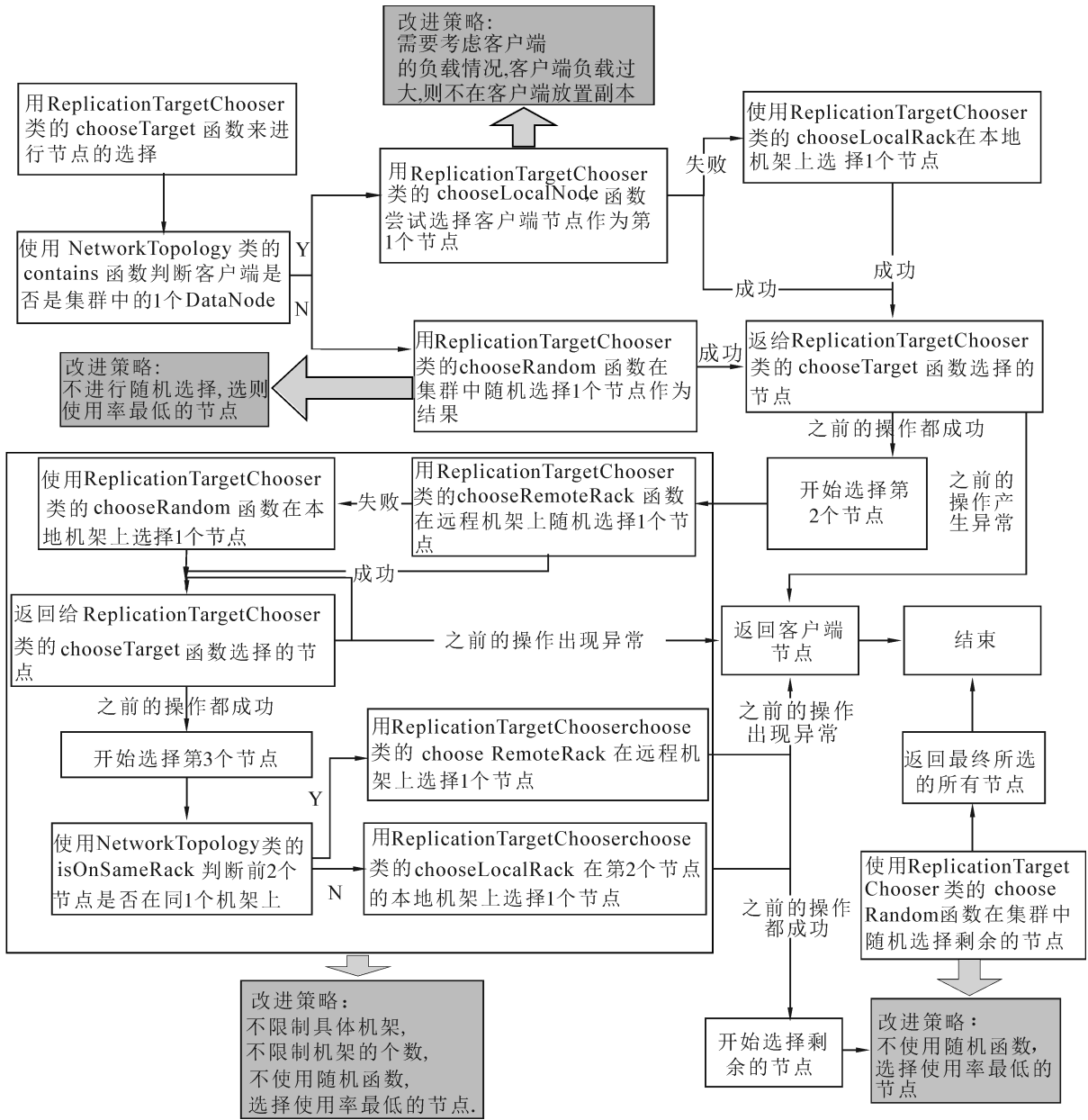


图 2 默认副本放置策略

Fig.2 The flowchart of default replica placement

- 1) HdFs 副本放置策略是调用 ReplicationTarget-Chooser 类的 chooseTargrt 函数来实现的.开始使用 NetworkTopology 类的 contains 函数,contains 函数通过判断客户端所在根节点与集群的根节点是否一致来判断客户端是否在集群中.
- 2) 如果客户端是集群中的一个节点,则调用 ReplicationTargetChooser 类的 chooseLocalNode 函数来尝试选择客户端节点作为第 1 个节点.
- 3) 客户端存储尝试失败时则调用 ReplicationTar-

- getChooserchooser 类的 chooseLocalRack 函数,在客户端节点所在机架随机选择一个节点作为第 1 个节点,然后将这个节点的信息传给 ReplicationTargetChoos-erchooser 类中的 chooseTargrt 函数,且将这个节点的信息记录在 ReplicationTargetChooserchooser 类中的一个 DatanodeDescriptor 类型的数组 results 中.
- 4) 如果客户端不是集群中的节点,则使用 Rep-licationTargetChooser 类的 chooseRandom 函数在集群中随机选择一个节点作为第 1 个节点,且将这个选

择的节点记录在数组 results 中.

5) ReplicationTargetChooser 类的 chooseRemoteRack 函数在第 1 个节点的远程机架上随机选择一个节点作为第 2 个节点.如果在远程机架上选择节点失败,则使用 ReplicationTargetChooser 类的 chooseLocalRack 函数在第 1 个节点的本地机架上随机选择一个节点作为第 2 个节点.将第 2 个节点记录在 ReplicationTargetChooser 类中 DatanodeDescriptor 下的数组 results 中.

6) 选择第 3 个节点,如果前 2 个节点是在同一个机架上,则使用 ReplicationTargetChooser 类的 chooseRemoteRack 函数在前 2 个节点的远程机架上选择一个节点.如果所选择的前 2 个节点并不在同一个机架上面,则使用 ReplicationTargetChooser 类的 chooseLocalRack 函数在第 2 个节点的本地机架上随机选择一个节点作为第 3 个节点,且存储第 3 个节点信息在数组 results 中.

7) 最终将 results 中的所有节点返回给副本选择函数的调用者.

2.2 Hdfs 副本放置策略的缺陷

Hdfs 默认副本放置策略综合考虑了多方面的因素,在可靠性、读写效率、负载均衡方面都做了一定的权衡,是一个比较优秀的副本放置策略,但 Hdfs 采用随机选择的副本放置策略.该策略没有考虑到节点负载的情况,在数据均衡方面比较薄弱,这使数据损坏时需要恢复的数据块数量可能会很多,数据读取的速度会受到影响等问题.

针对这一问题,Hdfs 提供了解决方案——均衡器^[10].均衡器(balancer)是一个 Hdfs 的守护进程,启动之后,它会将数据块从负载较高的节点移到相对

空闲的节点,从而达到重新分配数据块的目的,最终达到整个集群的数据块分布均衡.在数据块重新分配的过程中,均衡器会尽量将一个数据块的复本分散到不同机架,以提高数据块的冗余,降低数据损坏的可能性.

Hdfs 集群的管理员决定是否启动均衡器,启动后,会根据管理员设定的阈值来对集群进行均衡处理.阈值是每个节点的使用率(该节点上已经使用的空间和节点的空间容量之间的比值)和集群的使用率(集群中已使用的空间和集群的空间容量之间的比值)之间的差值,默认的阈值是 10%,管理员在启动均衡器的时候,可以指定阈值的大小.在任何时刻,集群中只能运行一个均衡器.

均衡器虽然可以解决数据块分布不均衡的问题,但是存在着明显的问题:

1) 均衡器对于集群数据块均衡的调节具有滞后性,它必须要在系统的不均衡状况超过阈值之后,才会进行调节.

2) 均衡器的运行和数据块的移动需要耗费一定的资源,很可能一个数据块刚刚写入到集群中,就因为均衡性而被移动,这种情况下集群的资源使用是很低效的.

3 Hdfs 副本放置策略的改进

Hdfs 默认的副本放置策略存在的不足,以及 Hdfs 提供的均衡器存在一些不尽人意的地方,本文提出了对其改进的低使用率优先(low rate first)副本放置策略.

3.1 改进副本放置的流程

图 3 是副本放置改进策略的流程.

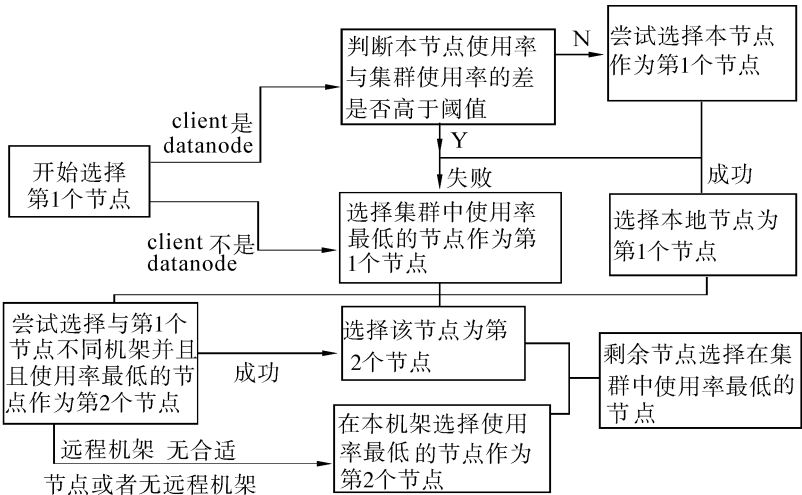


图 3 基于 3 副本放置策略的改进

Fig.3 The improved placement strategy based on three replicas

1) 考虑到数据写入带宽问题,依然在客户端所 在的节点上写入第 1 个副本,但考虑了该节点的负载

情况,即如果本地节点的负载超过了管理员指定的阈值,则选择集群中使用率较低的节点来放置副本.

2)除第 1 个副本在阈值满足的情况下放在本地节点上之外,其余所有的副本放置位置的选择,都是采用优先选择集群中比较空闲的节点的方式,以避免在负载较高的节点上继续存储数据.

3)为提高数据块的冗余,尽可能地将数据存储在至少 2 个机架上,本地机架上存储第 1 个副本,第 2 个副本选择与第 1 个节点不同的机架进行存储.因为 Hdfs 是一次写入、多次读取的设计思想,在数据写入的时候穿越多个机架,虽然写入带宽可能会有所降低,但是提高了集群的数据块分布均衡,有利于文件的读取和程序的运行.

4)为提高数据的冗余,保持每个节点只存储一个副本的规则.Hdfs 的默认副本放置策略是一个节点最多放置一个副本,如果副本的数量超过节点的总数,则集群中最多只放置与节点同样数目的副本.低使用率优先的放置策略依然坚持这个原则,每个节点最多只放置一个副本.

尽管当发生故障时,此策略会影响恢复数据速度,而且每存储一个副本时都需要调用函数获取节点信息,并判断该节点是否可以存储副本,这会降低运行速度及安全性.但考虑到 Hdfs 默认放置策略的副本放置的最终状态很难被控制,它在数据均衡方面的缺点比较明显,而这会带来一系列的问题,比如数据损坏时需要恢复的数据块数量可能会有很多,数据读取的速度可能会受到影响等因素,本文提出的对于 Hdfs 默认副本放置策略的改进方法有相对优势.

3.2 改进策略实现的核心类

副本放置改进策略会优先考虑在使用率比较低的节点上放置数据,这通过对 Hdfs 中负责副本放置节点选择的类 ReplicationTargetChooser 的改进来完成;该类在 Hdfs 中的作用是当有新增数据块或数据块位置变动的时候,NameNode 会调用该类来确定数据块放置的位置. ReplicationTargetChooser 类使用 chooseTarget 函数来选择副本放置的节点,图 4 描述了放置 k 个副本重写 chooseTarget 函数来实现的策略改进.

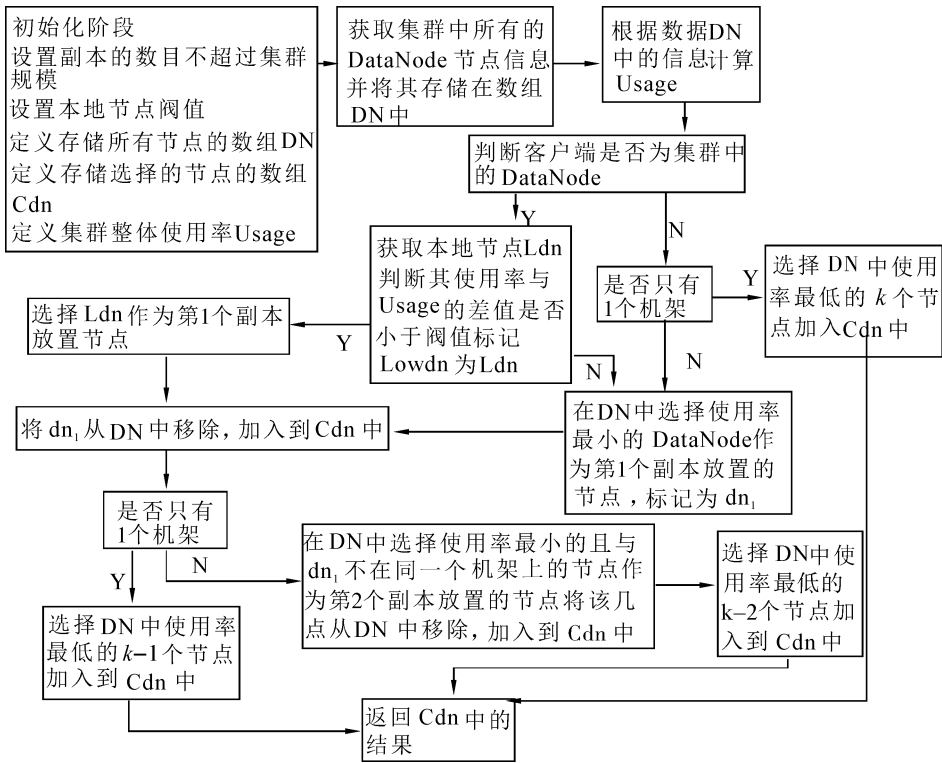


图 4 基于 K 副本放置策略的改进

Fig.4 The improved placement strategy based on K replicas

1)函数的初始化阶段:首先调用 NetworkTopology 类中的 getNumOfLeaves 函数来获取集群的大小,控制副本数目不超过集群的大小,如果设置的副本的数目超过集群的大小,则设置副本数目为集群大小.

2)管理员可设置本地节点阈值,默认值为 0.1,改进后的 Hdfs 在配置文件中为用户设置阈值提供了接口,在 Hdfs.xml 文件中可以通过为 dfs.replication.threshold 设置值来实现阈值的控制,阈值的范

围在 0~1, 0 表示本地节点的使用率必须小于等于集群的使用率才会在本地节点上放置数据块的副本; 1 表示不考虑使用率, 一定要在本地节点上放置数据块的副本。

3) 使用 Configuration 类的 getFloat 函数配置文件中的阈值。但如果用户没有设置阈值或者设置的阈值不合理, chooseTarget 函数依然使用默认阈值进行选择。

4) 在重写的 chooseTarget 函数中需定义一个 DataNodeDescriptor 类型的数组 DN 来存储全部节点的信息, DataNodeDescriptor 是 Hdfs 中用于描述 DataNode 信息的类, chooseTarget 函数可以通过操纵 DataNodeDescriptor 的对象来获取一个节点的信息, 包括节点 ID、节点名称、节点全部存储空间和节点已经使用的存储空间等。另外, 还需定义 DataNodeDescriptor 类型的数组 results 来存储已选择的节点, 同时定义集群的存储空间使用率 Usage。

5) 使用 NetworkTopology 类中的 getLeaf 函数可以获取集群中所有节点的信息, 将返回的所有节点信息存储在数组 DN 中, 然后可以根据 DN 中的信息计算集群的整体存储空间的使用率 Usage。在获取所有节点信息之后, 并不对数组 DN 进行任何处理, 比如排序、建堆等。虽然考虑到后面的算法中需要多次取得 DN 中使用率最小的节点, 但考虑客户端和不同机架, 因此该问题又与经典 TopK 问题相似且稍有不同。一般副本个数 K 默认为 3, 如果在客户端上放置一个副本, 选择另外 2 个副本的计算复杂度为 $O(2N-3)$ 。

6) 初始化后选择节点, 先通过 Hdfs 调用 chooseTarget 函数, 使用 NetworkTopology 类中的 contains 函数判断客户端节点是否在集群中, 如果不在, 则不在客户端上放置副本。否则还需进一步判断客户端节点的使用率与集群使用率的差值, 如果差值小于阈值, 则在客户端上放置第 1 个副本, 否则不在客户端上放置副本。使用 ReplicationTargetChooser 类的 isGoodTarget 判断客户端节点是否可用, 才能确定是否在客户端节点上放置一个数据块。

7) 如果客户端不可用, 则在 DN 中选择使用率最低的节点来尝试放置副本, 如节点不可用, 则将该节点标记为暂时不可选择, 然后继续在其他节点中选择一个使用率最低的节点, 直到选择到合适的节点为止。

8) 机架数目对副本放置节点的算法有一定的影响, 使用 NetworkTopology 类的 getNumOfRacks 函数来获取机架的数目, 则在 DN 中选择一个使用率最小的

节点作为第 1 个副本放置的节点。选择第 1 个节点后, 将其从 DN 中移除, 加入到 results 数组中。

9) 在选择第 2 个节点的时候, 在 DN 中选择使用率最小的节点, 然后使用 NetworkTopology 类的 isOnSameRack 函数判断它与选取的第 1 个节点是否在相同的机架上。如果这 2 个节点不在一个机架上, 则选择这个节点作为第 2 个副本存放的节点, 否则, 重新选择 DN 中其他节点中使用率最小的节点, 直到找到这样的节点为止。选择第 2 个节点之后, 将其从记录未被选择节点的数组 DN 中移除, 加入到记录已选择节点的数组 results 中。

10) 继续上述步骤选择其他节点。

11) 函数执行过程中, 使用 java 中的 try 来尝试运行, 若 chooseTarget 函数的运行没有出现异常, 则最终将存储已选择节点的数组 results 返回给函数的调用者。若执行过程中出现不可处理的异常, 则在 catch 语句中处理异常, 返回客户端节点。

4 实验结果与分析

为了比较 Hdfs 默认的和本文改进的副本放置策略, 本文实现了由 2 部分组成的测试程序: 1) 负责模拟一个节点运行的 DataNode 类, 该类记录了模拟节点的惟一标识、容量、使用量、数据块数量以及机架标识; 2) 模拟系统运行的 NameNode 类, 包括对于 DataNode 的初始化、设置阈值、设置副本放置策略和数据写入等内容的模拟。在模拟的过程中, 并不进行真实的数据的读写, 只是对于数据读写后的结果进行模拟记录。在 NameNode 类中初始化所有的节点, 每个节点初始的容量是 1 T, used 和 blockNum 被设置为 0。该程序模拟写入过程, 设置写入数据块的大小和模拟数据的分块。对于每一个划分的数据块, 程序运行相应的副本放置策略函数, 选择 3 个节点用于放置划分的数据块。然后循环处理直到数据块的写入完成。

写入所有数据后, 可根据模拟节点的使用情况计算不同的副本放置策略的数据存储均衡性。本文使用标准差来衡量副本放置的均衡性, 所使用的数据是所有 DataNode 的使用率, 也就是每个节点的使用容量 used 与总容量 capacity 之间的比值。设集群中一共有 n 个节点, 所有节点的使用率分别为 $X_1, X_2, \dots, X_{n-1}, X_n$ 。节点使用率的平均值 $\bar{X} = (X_1 + X_2 + \dots + X_{n-1} + X_n) / n$, 标准差 $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2}$ 。

本文实验一测试了机架数目对于算法的影响, 分别使用默认的策略和改进后的策略, 模拟测试在 500 个节点上写入大小不同的数据后, 系统的存储

均衡情况,写入文件大小为 100 G 条件下的节点使用率的标准差。

从图 5 中可以看出,对于默认的副本放置策略,机架数目在 3 个以内的时候,机架的个数对于系统的均衡性会有一定的影响,但是差别在 0.02% 以内。当机架数目超过 3 个以后,机架数目对于系统均衡性的影响会在 0.005% 以内。而对于改进后的副本放置策略,机架的数目对于集群的均衡性影响会变得更小,在 0.002% 以内。所以,机架的个数对于 2 种副本放置策略的影响都很小。通过图 5 可以看出,改进后的副本放置策略受到的影响更小,在不同机架个数情况下,都有更好的均衡性。

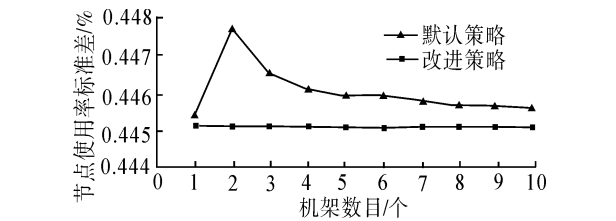


图 5 机架数目对于算法的影响

Fig.5 The impact of the algorithm based on the number of rack

实验二测试随着写入数据的增加,不同副本放置策略下集群存储的均衡性。实验选择在 500 个节点、5 个机架的条件下,分别使用默认的副本放置策略和改进后的副本放置策略,写入 1 G、10 G、100 G、1 T、10 T 和 100 T 的数据,测试集群的副本均衡情况。

根据图 6 显示,使用改进后的副本放置策略进行副本放置位置的选择,集群中数据块的均衡性明显好于使用默认的副本放置策略,在数据量比较小的时候这种优势还不太明显,但是在数据量比较大的时候,改进后的策略的好处就会更加明显。可见,改进后的副本放置策略,在数据块的均衡性方面有更加良好的表现。

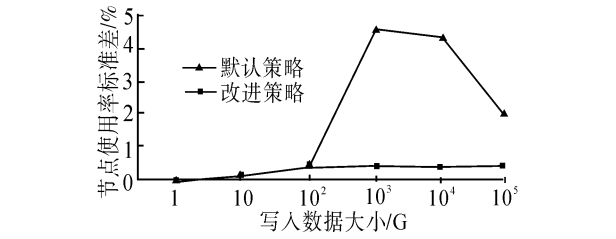


图 6 存储数据量对于算法的影响

Fig.6 The impact of the algorithm based on

本文提出的放置策略需获取集群中所有节点的信息,且将其存储在一个 DataNodeDescriptor 数组

中,从而增加了时间和空间的开销。另外要计算集群的整体使用率与选择集群中使用率较小的节点,增加了线性的开销。

5 结束语

本文基于节点存储率对 Hdfs 中负责副本放置节点选择的类 ReplicationTargetChooser 进行了改进,并部署了简单实际环境进行了实验,由于其实际环境受物理设备和其他异构条件等各种客观因素的影响不大,所以,本文所提方案提高了 Hdfs 的数据块放置的均衡性。但本文所提出的副本放置策略,关注的主要是集群中数据块副本放置的均衡性,所考虑的因素主要是节点的使用率,而没有考虑节点使用的价格、安全性、处理速度等。

参考文献:

[1] KO B J. Scalable service differentiation in a shared storage cache [C]//Proc of the 23rd International Conference on Distributed Computing Systems. Washington, DC, USA, 2003: 184-194.

[2] CHEN Yan. SCAN: a dynamic, scalable, and efficient content distribution network [C]//Proceedings of the International Conference on Pervasive Computing. Zürich, Switzerland, 2002: 282-286.

[3] KORUPOLU M R. Placement algorithms for hierarchical cooperative caching [C]//Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms. PA, USA, 1999: 586-595.

[4] 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 [C]//ACM Symposium on Theory of Computing. CA, USA, 1997: 654-663.

[5] BORTHAKUR D. The hadoop distributed file system: architecture and design [EB/OL]. [2012-11-08]. <http://hadoop.apache.org/core/docs>.

[6] GUY L, LAURE E, STOCKINGER H, et al. Replica management in data grids, GGF5 [R]. Global Grid Information Document, 2002.

[7] STONEBRAKER M, ABADI D J, De WITT D J, et al. MapReduce and parallel DBMSs: friends or foes [J]. Communication of the ACM, 2010, 53(1): 64-71.

[8] 魏青松, 卢显良, 侯孟书. AdpReplica: 自适应副本管理机制 [J]. 计算机科学, 2004, 31(12): 34-36.

WEI Qingsong, LU Xianliang, HOU Mengshu. AdpReplica: adaptive replica management mechanism [J]. Computer Science, 2004, 31(12): 34-36.

[9] 杨曙峰. 分布式并行文件系统的副本管理策略 [D]. 成都: 电子科技大学, 2003: 23-31.

YANG Shufeng. The Copies' management strategy of distributed parallel file system[D]. Chengdu: University of Electronic Science and Technology of China, 2003: 23-31.

[10] BORDAWEKAR R, LANDHERR S, CAPPS D, et al. Experimental evaluation of the Hewlett-Packard Exemplar file system[C]//ACM Sigmetric SPerformance Evaluation Review. [S.l.]: 1997.

作者简介:



邵秀丽,女,1963 年生,教授,博士生导师,天津市科委授予聘书专家,参与或主持国家自然科学基金,天津市自然科学基金、重点工程项目 40 余项。获得天津市科学进步奖 5 项,在国内外核心刊物上发表文章 90 余篇。



王亚光,女,1988 年生,硕士研究生,主要研究方向为网络安全与云计算等.发表学术论文 2 篇。



李云龙,男,1985 年生,硕士研究生,主要研究方向为软件工程与云计算.发表学术论文 3 篇。

第 11 届全球智能控制与自动化大会(WCICA2014)

The 11th World Congress on Intelligent Control and Automation(WCICA2014)

The 11th World Congress on Intelligent Control and Automation (WCICA 2014) will be held in Shenyang, China, from June 27 to 30, 2014. WCICA 2014 is technically sponsored by IEEE Robotics and Automation Society, IEEE Control Systems Society, National Natural Science Foundation of China, the Chinese Association of Automation, and the Chinese Association of Artificial Intelligence.

WCICA 2014 features plenary keynotes and plenary panel discussion sessions by the world leading researchers as well as awards to honor outstanding papers presented at this Congress. The awards include Best Paper on Theory, Best Paper on Applications, Best Student Paper, Best Poster Paper, Best Paper on Biomedical & Biosystem Related Areas, SUPCON Best Paper on Industrial Automation, and AIAG Best Paper on Supply Chain Related Topics.

Moreover, we are in the process to arrange post-conference publication of a selected group of accepted papers at WCICA 2014 in more than ten leading international journals and top Chinese journals as special issues, among them are IEEE/ASME Transactions on Mechatronics, IEEE Transactions on Robotics, IEEE Transactions on Control Systems Technology, IEEE Transactions on Industrial Electronics, IEEE Transactions on Systems, Man, and Cybernetics, Part B, and Acta Automation Sinica.

It is our great pleasure to invite you to submit your original research papers to the Congress. WCICA 2014 also welcomes proposals for organizing Focused Theme Sessions on the conference topics, Tutorials and Workshops on emerging topics. You are invited to submit focused theme session proposals to Prof. Simon X. Yang (syang@uoguelph.ca) before Dec. 1, 2013. Please refer to the Congress website for details.

Website: <http://2014.wcica.info/index.shtml>.

Tel: 02483681047-8013 .

E-mail: wcica2014@ gmail.com.