

基于网络延时的 CEPH 存储性能优化方法

陈凌剑¹, 王 勇², 俸 皓²

(1 桂林电子科技大学 信息与通信工程学院, 广西 桂林 541004;

2 桂林电子科技大学 广西高校云计算与复杂系统重点实验室, 广西 桂林 541004)

摘 要: 通过分析集群中单节点在不同网络延时下的延时大小与集群存储性能的关系, 提出基于网络时延的权重改进方法. 实验表明所提出的方法可根据存储节点的网络时延对权重进行再调整, 使数据较大概率存入网络时延小的节点. 该方法降低了由单一节点网络延时引起的存储性能损耗, 从而提高了 CEPH 分布式存储系统的存储性能.

关键词: CEPH; CRUSH; 分布式存储; 任务调度; 负载均衡

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2017)06-0084-05

Method to Improve the Performance of CEPH Storage Based on Network Delay

CHEN Ling-jian¹, WANG Yong², FENG Hao²

(1 School of Information and Communication Engineering, Guilin University of Electronic Technology,

Guilin 541004, China; 2 Guangxi Colleges and Universities Key Laboratory of Cloud Computing and

Complex Systems, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: This article propose an new method based on weight of network delay by analysis the relationship of one node's network delay and the storage performance of CEPH. The experiments show that the method could adjust the node's weight depended on the network delay of that node, so that data files could be stored in good network performance with greater probability. This new method reduces the time cost caused by network delay and improves the performance of CEPH storage system in that case.

Key words: CEPH; CRUSH; distribute system; task scheduling; load balancing

1 引言

相关数据显示 2020 年的全球信息量将超过 40 ZB, 而我们每天的全球互联网流量累计达 1 EB. 各个大型网站每天都在产生海量的数据, 如何高效地存储这些海量数据并且满足快速横向纵向扩展的要求是待解决的首要问题. CEPH^[1] 分布式存储系统是满足以上一系列条件的分布式存储系统之一. CEPH 充分利用到了存储设备自身的计算能力, 通过 CRUSH^[2] 算法计算出数据对象的存储位置,

从而消除系统对单一中心节点的依赖, 实现了真正的无中心结构^[3]. 其副本强一致性策略保证了数据对象在 CEPH 存储集群中的有效性, 出错节点的自动故障恢复机制保证了存储集群的可靠性, 去中心化的设计结构使其可扩展性得到了保证. 正是其高有效性、高可靠性、高可扩展性使得 CEPH 分布式存储集群从各个开源的分布式存储系统中脱颖而出, 逐渐获得更多人的关注以及更深入的研究.

典型的分布式存储系统广义上可分为基于中心的分布式文件系统和无中心的分布式文件系统两

种. 其中 GFS(Google File System)、HADOOP 是前者的代表^[4]. GlusterFS、Amazon 下的 Dynamo 存储架构是无中心存储系统的典型代表^[5]. 文献[6]针对 CEPH 集群中元数据服务中 cache 数据丢失的问题提出了 cache 日志备份方案,该方案降通过备份的 cache 日志降低了集群获取元数据的耗时以及存储节点注册到集群的时间,从而达到提升集群的存储性能的效果.

文献[7]提出了基于节点行为的任务调度算法. 文献[8]提出了一种基于文件支持度的动态副本管理算法. 文献[9]提出了一种时间序列技术的副本策略算法. 文献[10]提出了一种通过评估文件未来价值并以此删除文件的策略.

以上算法从不同方面优化了 CEPH 性能,但仍存在一些缺陷.

2 基于网络时延的 CEPH 性能优化

2.1 背景介绍

CEPH 分布式存储集群读写数据对象时采用 Primary-Replica 模型,通过 CRUSH 算法求得数据对象的主 OSD(Object Store Device)节点及副本 OSD 节点. 客户端向集群写某一数据对象时只向该数据对象所对应的主 OSD 节点发送写请求,主 OSD 再负责与副本 OSD 的数据交互,只有在主副本 OSD 均保存完该数据对象后,客户端才会收到写完成消息. 该副本强一致性策略保证了 CEPH 存储集群数据的高可靠性.

客户端-集群写流程如图 1 所示.

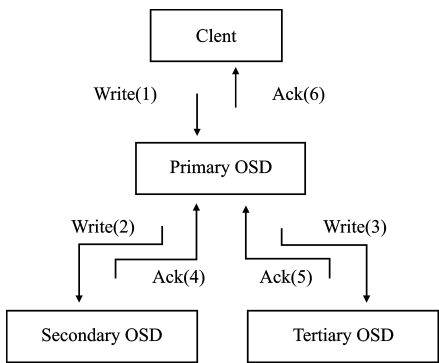


图 1 CEPH 集群写流程

2.2 CRUSH 算法的缺陷

如何使数据尽可能均匀地分布在集群上是分布式算法必须重点解决的一个问题. GlusterFS 采用

弹性哈希的同时引进虚拟卷的概念、Dynamo 采用一致性哈希同时引进虚拟节点的概念,这两个能很好地解决均匀分布的问题. CEPH 中的 CRUSH 算法在解决均匀分布的同时引入了权重参数,在存储数据时集群中权重较大的节点具有更高的概率存储数据,这在数据存储设备容量大小参差不齐情况下有着明显的优势. 数据对象更大概率存储在数据容量大的硬盘上,可避免小硬盘存满而大硬盘剩余容量很多而引起的负载不均衡现象.

CEPH 中的 CRUSH 算法以权重为限制条件达到数据分配概率均匀的目的. 但在实际的运行过程中权重参数并不随着剩余存储容量的变化而变化,当某个节点的剩余容量因某些原因产生巨大变化时,如果数据仍然以原来的权重为限制条件选择存储位置,在该情况下这种选择策略显然不能达到负载均衡的效果.

其次 CRUSH 算法中的权重仅仅与剩余容量有关,没有考虑到机器负载、网络性能等参数带来的影响. 通过对 CEPH 集群副本强一致性的研究可知,主 OSD 节点在客户端与集群的数据交互以及集群内部之间的数据交互中均起着至关重要的作用. 在集群负载较大、网络性能不稳定的情况下,原有的 CRUSH 算法仅与剩余存储容量有关,并不能选择出负载小、网络性能好的存储设备作为数据对象的主 OSD 节点. 在该种环境下通过优化 CRUSH 算法的选择策略可以提高 CEPH 集群的写性能.

2.3 基于节点时延的优化方案

CEPH 集群通过 CRUSH 算法选择存储节点位置时权重大的节点有更高的概率获得数据,而原有 CRUSH 算法中的节点权重参数仅与该节点的剩余容量有关. 当某一节点出现网络延时由于集群的副本强一致性策略,该节点很容易成为存储数据过程的短板节点,从而影响整个集群的存储效率.

针对上述的情况,提出一种基于网络时延的 CEPH 集群优化方案,将网络时延信息加入到各节点权重的计算. 考虑到网络状况具有波动性,过于频繁的修改权重同样会引起集群的数据迁移从而提高集群负载,所以新算法不需要对每一次的网络时延变化做出响应进而修改权重值. 通过对网络性能中的时延参数设定一阈值 T ,可以解决该问题. 表示如下:

$$\begin{cases} W(\text{osd}, n) = W_{\text{new}}(\text{osd}, n) & d(\text{osd}, n) \geq T \\ W(\text{osd}, n) = W(\text{osd}, n) & d(\text{osd}, n) < T \end{cases}$$

式中, $d(\text{osd}, n)$ 表示第 n 个 OSD 节点相对于集群的时延. T 为时延的阈值. 只有当网络时延超过该阈值 T 时集群重新对各节点网络性能进行评估, 降低性能较差节点的权重减少向其分配的数据以达到优化存储的效果.

3 实验

为了测试 CEPH 分布式存储集群中的某一节点在不同网络延时下对整个集群的存储性能影响, 搭建以下测试环境 (CEPH 集群采用双副本存储策略)

硬件信息:

- 处理器 英特尔第四代酷睿 i7-4790 @ 3.60 GHz;
- 内存 16 GB (三星 DDR3L 1600 MHz);
- 硬盘 希捷 ST1000DM003-1ER162;
- 网卡瑞昱 RTL8168/8111/8112.

软件信息:

- 操作系统 Ubuntu 14.04;
- CEPH 版本 9.2.1;
- OSD 系统 XFS.

集群节点信息:

- ClientClientIP: 192.168.1.30;
- OSD.0 OSD/MonitorIP: 192.168.1.1;
- OSD.1 OSDIP: 192.168.1.2;
- OSD.2 OSDIP: 192.168.1.4.

以 5 ms 为步长 0~50 ms 为范围模拟 CEPH 集群中某一 OSD 节点到客户端的网络延时, 在每一个网络延时阶段以 0.1 为步长在 0~1 的范围内调整该 OSD 节点的权重同时保证其余各 OSD 节点权重均为 1, 客户端在每一个权重变化阶段向集群写入 2048 个 2 MB 随机文件共 4 GB, 测试在不同权重下集群存储 4 GB 文件所需的时间 t .

通过比较不同延时、不同权重条件下的集群存储完成时间 t , 拟合出集群在空负载条件下权重与集群某一节点网络延时的关系函数 W .

图 2 显示在 OSD.1 节点存在 50 ms 网络延时的情况下, 随着该节点权重的提升, 向 CEPH 集群存储 4 GB 数据所需的总时间在不断减少, 当节点

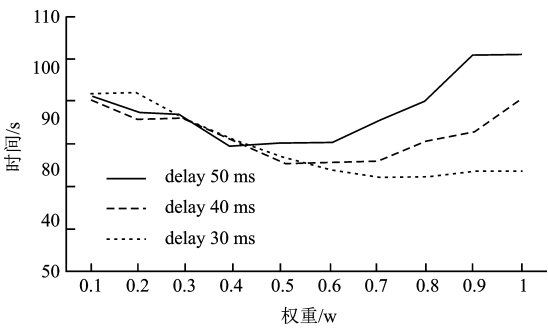


图 2 集群存储性能测试对比

权重上升至 0.4 时, 波形出现拐点集群存储 4 GB 数据所需时间最小. 当权重上升至 0.6 时, 波形出现拐点, 集群存储 4 GB 数据所需时间上升.

在 OSD.1 节点延时 40 ms 的情况下, 随着该节点权重的增大, 在权重为 0.5 时出现拐点, 0.5~0.7 阶段存储效果较优, 当权重大于 0.7 时, 集群存储时间明显增加.

在 OSD.1 节点延时 30 ms 的情况下, 随着该节点权重的增大集群存储所需总时间降低, 当权重达到 0.7 时, 存储所需总时间趋于稳定.

图 3 显示随着 OSD.1 节点权重的增加, 存储在该节点的数据总量在增长, 而存储在其余节点的数据量在降低. 直至权重与其余节点相同时各节点分布数据量大致均匀.

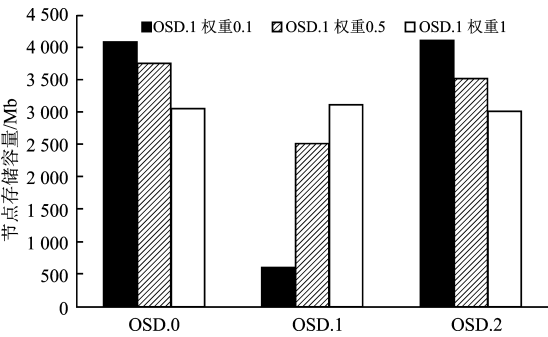


图 3 各节点存储数据对比

该实验表明开始时 OSD.1 节点权重为 0.1 而其余节点权重均为 1, OSD.0 与 OSD.2 两个节点将负责很大部分的数据存储工作而 OSD.1 节点虽然存在了 50 ms 的网络延时, 但在大部分时间内仍处于空闲状态. 此时节点 OSD.0、OSD.2 成为集群存储瓶颈. 随着 OSD.1 节点权重的增大有更多的数据存储到 OSD.1 节点, 分摊了其余节点的存储任务从而降低了总的存储时间. 50 ms 延时下当 OSD.1 节

点权重上升至 0.6 以后波形出现明显拐点,此时随着 OSD.1 节点权重的增加该节点获取的数据量不断增大,在存在网络延时的情况下,处理相同的数据所需的时间更多,因此 OSD.1 节点成为存储集群中的短板节点从而导致集群总的存储数据时间上升.

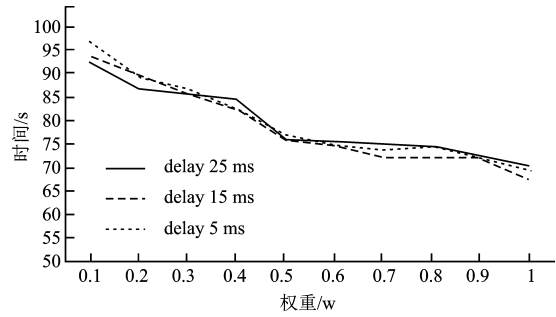


图 4 集群存储性能测试对比

图 4 显示当 OSD.1 节点网络时延为 5 ms 时,随着节点权重的增加,集群存储 4GB 文件所需的总时间线性下降,且没有出现类似图 2 的明显拐点,这表明在集群存储 4GB 文件且 OSD.1 节点存在 5 ms 网络延时的情况下集群存储性能与网络延时并不存在直接关系,随着该节点权重的增大,存储的数据量也相应增大,集群节点的短板效应在减弱.

当 OSD.1 节点网络时延为 15 ms,25 ms 时,集群的存储耗时波形并没有显著变化. 这表明在三个节点双副本策略的 CEPH 集群测试环境下,当某一节点的网络延时小于 25 ms 时,集群存储 4GB 文件所需的时间与网络延时并没有直接关系,网络延时并不是影响集群存储能力的关键因素.

表 1 列出了 OSD.1 节点在不同时延情况下权重与整个集群存储 4G 文件所需总时间的测试数据,节点延时 0~20 ms 阶段当 OSD.1 权重为 1 时存储总时间最少且在 67~69 s 之间波动. 数据显示当节点时延大于 25 ms 时随着时延的减小其最优权重值在不断变大,时延与最优权重大致成反比关系. 当节点时延小于等于 25 ms 时其最优权重为 1. 可以猜想当节点延时大于 50 ms 时其最优权重随之时延的增大而减小,且由于集群副本强一致性策略,节点时延应当存在一个上限阈值当时延超过该阈值时该节点权重为 0 集群存储效果最优.

由表 1 可知在节点延时 30~50 ms 之间其最优权重的变化范围为 0.4~0.8,该范围变化较小,以 0.1 为步长测试并不能得到精确的最优权重变化,

因此只能反应出单节点网络时延与权重值的大致关系,可将其表示为

表 1 不同时延下节点权重与存储效率关系

delay/ms \ Weight	50	45	40	35	30	25	20
0.1	91.4	93.1	90.7	95.8	92.2	92.7	93.6
0.2	87.2	90.9	86	88	91.9	87.3	87.6
0.3	86.3	84.7	86.2	84	86.3	85.8	87.3
0.4	79.5	81.2	80.8	79.9	81.3	84.3	80.9
0.5	80.1	75.9	76	74.5	76.9	76.6	76.6
0.6	80.1	77.3	75.5	73	74.2	75.6	74.6
0.7	85.4	79.7	75.8	75.2	72.3	74.6	76
0.8	90.3	82.9	80.4	77.6	72.2	74.4	71.7
0.9	100.5	90.0	82.5	78	73.6	72.4	72
1.0	100.9	94.8	90.5	82.9	73.7	69.9	68.1

$$W = \begin{cases} 1, & t \leq T_1 \\ \frac{a}{(t+b)} + c, & T_1 < t \leq T_2 \\ 0, & t > T_2 \end{cases}$$

由本次试验可知,在此次测试环境下 $T_1 = 25\text{ ms}$.

图 5 显示在此次实验环境下的最佳权重点的拟合曲线. 结果表明随着节点时延的增加最佳权重点在下降,在一定时延范围内最佳权重和时延的关系符合反比例函数特性,实验结果与实验前的猜测相符. 根据拟合结果节点的延时与权重关系可以表示为

$$W = \begin{cases} 1, & t \leq 25 \\ \frac{19.25}{(t-5.949)} - 0.01149, & 25 < t \leq T_2 \\ 0, & t > T_2 \end{cases}$$

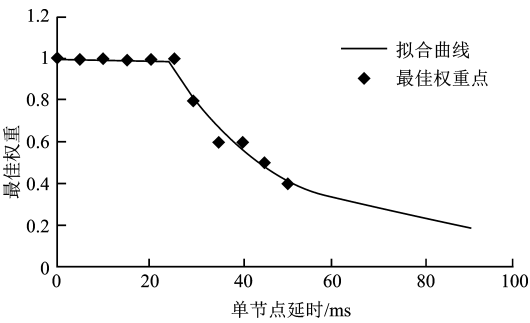


图 5 单节点时延与最佳权重关系

图 6 显示了在单节点网络延时 50 ms 的情况下改进后的算法与原算法的存储性能对比情况. 在 10

次测试情况下,改进前集群存储 4G 数据所需时间在 85~119 s 之间波动,平均消耗时间为 100.9 s 且波动幅度较大.在算法改进后相同的情况下集群存储 4G 数据所需时间在 74~82 s 之间波动,平均存储耗时为 78.7 s 且波动较小.结果表明改进后的算法能有效减少单节点出现网络时延情况下的集群存储耗时,提高集群存储性能.

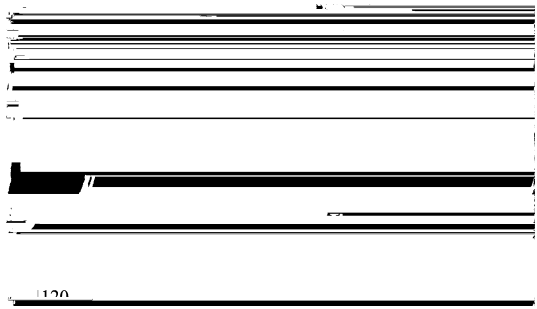


图 6 集群存储性能测试对比

4 结束语

原本的 CRUSH 算法根据剩余存储容量设定权重有效地解决了负载均衡问题,使数据能够尽量概率均匀地分布在集群上,然而在存在网络时延的情况下其存储效率并不能达到最优.基于该种情况,本文研究了三节点双副本策略的 CEPH 分布式存储集群环境下单节点网络延时对集群整体存储性能的影响关系,并给出了该环境下的节点权重拟合公式.实验结果表明通过新权重公式设置节点权重能有效提高 CEPH 集群在该环境下的数据存储效率,缩短存储耗时.下一步考虑对集群规模变大时多节点出现网络延时的数据存储策略问题进行研究.

参考文献:

[1] Weil S A. CEPH: reliable, scalable, and high performance — distributed storage[EB/OL]. [2016-01-31]. <http://www3.nd.edu/~dthain/courses/cse40771/spring2007/papers/ceph.pdf>.

[2] Weil S A, Brandt S A, Miller E L, et al. CRUSH: controlled, scalable, decentralized placement of replicated data[C]// ACM/IEEE Conference on Supercomputing. ACM, 2006:31.

[3] 衣国垒. 基于中心节点架构的大规模数据对象存储系统[D]. 北京:清华大学, 2013.

[4] Garcia H, Ludu A. The Google file system[J]. Acm-Sigops Operating Systems Review, 2003, 37(5): 29-43.

[5] Decandia G, Hastorun D, Jampani M, et al. Dynamo: amazon's highly available key valuestore[J]. Acm Sigops Oper. syst. rev, 2007,41(6):205-220.

[6] Ling Zhan, Fang Xieyun, Li Daping. The research and implementation of metadata cache backup technology based on CEPH file system[C]//IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). IEEE, 2016:72-77.

[7] Jiang Y, Zhou Y, Wang W. Task allocation for dependable multiagent systems in social networks[J]. Parallel & Distributed Systems IEEE Transactions on, 2013, 24(8):1671-1681.

[8] 肖中正, 陈宁江, 贾灵昊, 等. 一种基于文件支持度的动态副本管理机制[J]. 计算机研究与发展, 2016, 53(2):431-442.

[9] Hussein M K, Mousa M H. A light-weight data replication for cloud data centers environment[EB/OL]. [2016-02-06]. https://www.ijircce.com/upload/2014/january/1_A%20Light.pdf.

[10] Mansouri N, Dastghaibyfar G H. Enhanced scheduling strategy in datagrid[J]. Journal of Parallel & Distributed Computing, 2013, 73(4):534-543

作者简介:

陈凌剑 男,(1990-),硕士研究生. 研究方向为云计算、分布式存储. E-mail:250316792@qq.com.

王 勇 男,(1964-),博士,教授. 研究方向为智能计算、泛在网研究及应用、计算机网络技术.

俸 皓 男,(1978-),博士研究生. 研究方向为无线传感器网络、移动计算、嵌入式实施软件等.