

分布式存储系统中数据副本管理机制

徐小龙^{1,2}, 邹勤文¹, 杨庚³

(1. 南京邮电大学 计算机学院, 江苏 南京 210003;

2. 信息安全国家重点实验室(中国科学院软件研究所), 北京 100190;

3. 宽带无线通信与传感网技术教育部重点实验室, 江苏 南京 210003)

摘要: 分布式存储系统需要完善的数据副本创建、部署、选择、定位和一致性管理机制以保证分布式计算环境中的数据的安全、可用、可靠、可扩展性和服务的高效、连续性。文中全面分析与研究了国内外对分布式存储系统中的副本管理机制研究现状, 重点对副本创建、副本定位、副本一致性维护和副本撤销机制进行深入研究, 并从数据可用性、节点负载均衡、数据一致性和带宽消耗等性能指标进行了分析。文中的研究成果对于分布式存储系统的合理设计与构建具有良好的参考价值。

关键词: 数据副本; 副本创建; 副本定位; 数据一致性; 副本撤销

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2013)02-0245-05

doi: 10.3969/j.issn.1673-629X.2013.02.063

Data Replication Management Mechanisms for DSS

XU Xiao-long^{1,2}, ZOU Qin-wen¹, YANG Geng³

(1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. State Key Laboratory of Information Security(Institute of Software,
Chinese Academy of Sciences), Beijing 100190, China;

3. Key Lab of Broadband Wireless Communication & Sensor Network Technology of
Ministry of Education, Nanjing 210003, China)

Abstract: distributed storage systems need the data replication management mechanism about the data creation, placement, selection, position and consistency, in order to ensure data secure, available, reliable, scalable and services efficient and consistent in distributed computing environments. In this paper, a comprehensive analysis has been made about the domestic and overseas research results of the data replication management mechanism for distributed storage systems in depth. Furthermore, an analysis is made from the aspects of the data availability, the load balancing of nodes, the data consistency and the bandwidth consumption, etc. The research result has a good reference value for the rational design and construction of future distributed storage systems.

Key words: data replica; replica creation; replica placement; data consistency; replica revocation

0 引言

分布式存储系统(Distributed Storage Systems)是基

于存储服务器集群(Cluster)和分布式文件系统,将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作,并通过各种相应的应用软件或应用接口,共同为用户提供高可用、高可靠的数据存储和业务访问功能的存储资源系统。为了保证数据安全、可用、可靠、可扩展性和服务的高效、连续性,分布式存储系统需要完善的数据多副本创建、部署、选择、定位和一致性管理机制。随着互联网中的用户对资源的需求量日益增多,如果仅有一份数据,则需要该数据的用户都须到同一个节点上读取它,网络容易出现拥塞,而处理能力有限的节点也会因为访问数量太大而宕机。然而,创建多份数据副本,并将它们合理分布在多个服务器节点上,分担处理访问请求的任务,可以有效降低节

收稿日期:2012-06-12;修回日期:2012-09-16

基金项目:国家“973”重点基础研究发展计划项目(2011CB302903);国家自然科学基金资助项目(60873231);国家教育部高等学校博士学科点专项科研基金资助课题(20093223120001,20113223110003);中国博士后科学基金资助项目(2011M500095);江苏省博士后科研资助计划项目(1102103C);江苏省自然科学基金(BK2011754,BK2009426);江苏省科技支撑计划(BE2009158);信息安全国家重点实验室开放课题(03-01-1)

作者简介:徐小龙(1977-),男,副教授,博士,博士后在站,CCF会员,主要研究方向为计算机软件、分布式计算、信息安全、Agent技术等。

点失效率,减少用户响应时间。

文中详细分析了目前国内外对分布式存储系统中的副本管理机制研究现状,重点对副本创建、副本定位、副本一致性维护和副本撤销机制进行深入的研究,并从数据可用性、节点负载均衡、数据一致性和带宽消耗等性能指标进行了系统的分析。

1 副本创建

某一节点上的数据被频繁访问使得该服务器节点负载过重时,或出于提高可靠性的考虑时,可将数据复制一份或多份副本并存储到其它节点上。

1.1 副本数量的设置

副本数量对分布式存储系统的可用性的影响很大,创建太少容易产生数据热点问题,延长访问时间,太多则会造成无谓的存储空间浪费。很多存储系统复制的默认数据副本数是 3 份,即在数据投入使用时复制 3 份它的副本,之后根据具体情况来创建和撤销副本。

文献[1]根据副本复制的数量可将副本复制方法分为 3 种:均匀复制,所有数据对象复制相同数量的副本;比例复制,复制数量与被访问频率成正比;方根复制,复制数量与被访问频率的方根成正比。方根复制在平均查询距离和副本利用率方面具有较理想的性能表现。文献[2]经模拟实验得出当副本的生命周期较长和副本密度较高时更能体现方根复制方法的优势。虽然副本复制的数量一般被认为应该正比于原数据大小的平方根,而文献[2]的研究结论表明,副本复制的数量应该反比于原数据大小的平方根。

1.2 副本复制策略

副本复制策略分为路径复制、源请求复制、邻居节点复制、随机复制和优先级复制五种^[3,4]:

(1)路径复制。发送副本给请求路径上的所有节点。优点是实现原理简单,方便数据的查找;缺点是创建的副本数量供过于求,且增加了副本的一致性维护的开销。

(2)源请求复制。只发送副本给请求节点。LAR (Lightweight Adaptive Replication,轻量级自适应的复制方法)算法^[5]是美国马里兰大学研究人员提出的经典源请求复制算法,其主要思想是:当访问请求到达目的节点时,若目标节点未过载,则能读取数据,若目标节点处理能力不够,将创建一份新副本,而且如果请求节点未过载,才把新创建副本发给该请求节点,并告知请求路径上所有节点该请求节点上也有该数据副本。优点是对于目的节点来说,减少了副本的复制数量;缺点是请求路径上有该副本且达到复制阈值的节点都存一份副本到请求节点上,易造成请求节点过载。

(3)邻居节点复制。对网络数据都保存访问历史记录,节点将被频繁访问的副本新建一份发送给频繁请求的节点的邻节点,当请求节点再次访问该数据时,可以到其邻居节点直接读取数据了,从而减少了请求的跳数。该方法缺点在于历史记录预测会有一定概率的失误。

(4)随机复制。随机选择一个或多个节点来存放副本,有随机选择的对象是请求路径上的节点和整个网络的节点两种策略,后者主要运用多哈希函数和关联哈希两种方法。多哈希函数的优点是可以动态调整副本的数目;副本被高度分散了,有益于负载均衡;缺点是管理多个哈希函数是个复杂的工作。关联哈希的优点是明显减少了访问时延;缺点是产生较大的副本数量和系统开销。

(5)优先级复制。请求发生就向已经有副本的节点发送所需副本,直至饱和,再选择别的节点来存储副本。优点是减少了存放副本的节点数,减低了节点的维护开销;缺点是存放副本的节点易过载,容易出现新一轮的访问热点问题。

通过比较这 5 种副本分布方法,可以发现路径复制和优先级复制方法不够灵活、效率相对较低,其它 3 种方法可以在大多数分布式网络环境下使用并能解决热点问题。

1.3 典型副本分布方法

文献[6]提出了一种渐进优化的选举和分区合并算法来存储多个副本,以求得目标区域中的最佳存储节点。方法假设要存储 n 个副本,先将拓扑结构划分为多个区域,每个区域都有一个服务节点,即该区域内最适合放置副本的节点,然后根据选举法,选举过程中,考虑了客户的分布情况、访问频率、通信时延和节点的处理能力四个因素,每次淘汰一个区域,并有调整剩余区域的环节,经过多次的选举淘汰区域调整,最终将整个网格划分为 n 个区域,这 n 个区域的服务节点就是最佳存储节点。

文献[7]提出一种网格环境下的多副本后向预测调度的算法。方法与邻居节点复制策略有些相似,也是根据已收集的历史数据来预测合适的存储节点,不同的是在发生负载失衡情况之前将副本直接存储到选出的节点而不是它们的邻居节点。

1.4 数据迁移方法

网络系统的一个重点问题是如何实现负载均衡,通过新副本的添加或撤销能达到这一目的,另外一种常用的方法则是数据迁移。虚拟节点技术^[8]的核心思想就是数据迁移。数据虚拟节点是存储数据文件、路由定位的基本单元,一个物理节点可管理多个虚拟节点。若一个物理节点过载,则将其管理的部分虚拟节

点转移给其它物理节点管理,数据也将随之转移。

虚拟节点技术有一对一、一对多和多对多这三种策略。虚拟节点策略的缺点在于实现复杂。由于复制技术本身已包含分布策略,且虚拟节点技术必须是在拥有足够数量的副本才能实现,所以虚拟节点技术更适合于与复制技术结合使用。

2 副本定位

节点访问数据性能表现的优劣很大程度上受到数据定位策略的影响,即如何快速定位出目标数据所在节点的位置,然后读取数据。

传统的基于覆盖网(Overlay network)的副本定位算法虽然在不同程度上解决了副本定位效率、负载均衡和可扩展性等问题,但目标节点不能很好地满足特定应用的服务质量需求^[9]。文献[9]提出一种多维度服务质量约束的副本定位方法,通过采用分层和对等的混合定位机制,在高效定位的同时,还保证目标节点提供有效的服务质量。方法基于区域内分层、区域间对等的覆盖网拓扑结构,运用了区间路由算法、副本信息发布算法、站内副本算法、区内副本定位算法和区间副本定位算法等五个算法,使大量副本定位在本区域完成,从而有效降低了定位延迟,以满足特定应用的多维度服务质量规约作为副本定位标准,有效地保障了目标节点的服务质量。

文献[10]提出了一种用于分布式系统多副本对象访问控制的分层结构分布式互斥实现方法,该方法利用了系统的自组织特性,对节点采取分层式管理方式,如图1所示。

第1层每行只有一个节点,采用动态令牌控制方式,每行的第一个节点都带领着此行的其它节点,以保证在出现节点间逻辑图不一致时互斥操作的顺利完成。

第2层为每行的多个节点,采用基于允许的分布式互斥算法。要求各节点采用相同的聚类规则和代表节点产生规则,系统中每个节点都保持一个系统分层结构逻辑图,并随着系统运行而得到及时更新。对系统进行互斥访问,需要得到各子系统代表节点所组成第2层所有节点的同意。当一个进程需要对系统中的对象进行互斥访问时,该进程先读取本节点保存的系统分层逻辑结构图,并向保存在数组中的表头节点发送访问请求。

3 数据一致性

数据一致性是指复制源相同的多个副本之间数据一致,分为弱数据一致性和强数据一致性。数据各副本最终达到一致即可满足弱数据一致性,强数据一致

性则要求数据各副本任何时候都要求一致。由于多任务并行执行、网络延迟不可预测和修改对象的不确定性等原因,分布式系统中的数据一致性维护过程比较复杂。

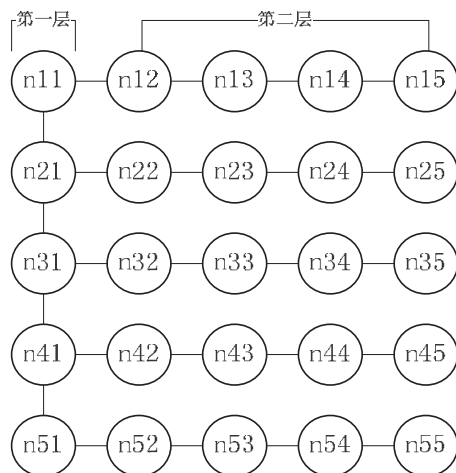


图1 25个节点分布式系统全活跃分层聚类逻辑图

3.1 Paxos 算法

Paxos 算法^[11~13]是由 Leslie Lamport 提出的一种基于消息传递的数据一致性算法,用于解决分布式系统中的一致性问題,是目前为止公认最为有效的经典数据一致性算法。

在 Paxos 算法中,节点被分成了三种类型,Proposer、Acceptor 和 Learner,且每个节点可以有多种角色。保证数据的一致性要满足以下三个条件:

1、决议只有在被 Proposer 提出后才能被批准;

2、每次只批准一个决议;

3、只有决议确定被批准后 Learner 才能获取这个决议。

为了达到这三个条件,需要满足下面几个约束条件:P1:每个 Acceptor 只接受它得到的第一个决议;P2a:一旦某个决议 v 得到通过,之后任何 Acceptor 再批准的决议必须是 v ;P1 和 P2a 有矛盾,主要体现在节点因失效参加不了决议,所以提出来第三个条件 P2b;P2b:一旦某个决议 v 得到通过,之后任何 Proposer 再提出的决议必须是 v ;P2b 不易通过技术实现,所以提出蕴含 P2b 的 P2c;P2c:如果一个编号为 k 的提案具有值 v ,那么存在一个“多数派”,它们中没有谁批准过编号小于 k 的任何提案,或者它们进行的最近一次批准具有值 v 。

在这些约束条件的基础上,将一个决议的通过分为两个阶段^[12]:

1. 准备阶段:

a. Proposer 选择一个提案编号 k 并将 prepare 请求发送给 Acceptor 中的一个多数派;

b. Acceptor 收到 prepare 消息后,如果提案的编号

大于它已经回复的所有 prepare 消息,则 Acceptor 将自己上次的批准回复给 Proposer,并承诺不再批准编号小于 k 的提案。

2. 批准阶段:

a. 当一个 Proposer 收到了多数 Acceptor 对 prepare 的回复后,就进入批准阶段。它要向回复 prepare 请求的 Acceptor 发送 accept 请求,包括编号 k 和根据 P2c 决定的 value(如果根据 P2c 没有决定 value,那么它可以自由决定 value);

b. 在不违背自己向其他 Proposer 的承诺的前提下,Acceptor 收到 accept 请求后即批准这个请求。

为了减少决议发布过程中的消息量,Acceptor 将这个通过的决议发送给 Learner 的一个子集,然后由该 Learner 去通知所有其他的 Learner,即所有副本都将执行一样的更新。

3.2 协同计算系统的副本一致性维护

协同计算环境由计算机网络技术、通讯技术、多媒体技术和群件技术共同构成,使不同地域、不同时间、不同文化背景的人们能够协调一致地为某项任务共同工作。

文献[14]指出基于无结构 P2P 网络的文件共享系统中的数据一般是静态的,数据一致性维护的工作量较小;然而在基于无结构 P2P 网络的新型协同计算系统中,数据具有较强的动态性,这种模式下的数据要求可被参与工作的人频繁更改,同时保持强数据一致性。最简便的方法是集中式方法^[15~17]:由一个或几个副本节点保存所有成员信息,当发生更新时,就由这些节点来向其它节点发送更新信息。集中式方法的优点在于发送更新快,但它的可扩展性差,易引起单点失效问题。基于 Gossip 的组管理协议^[17]的容错能力和扩展性较好,但基于 Gossip 协议的数据一致性维护方法不能保证副本数据的强一致性,还会出现大量冗余数据^[18]。如以树状结构组织节点,更新信息冗余较少且更新快,但容错能力不高。

文献[14]结合以上三种方法的优点,提出一种基于分割树的无结构 P2P 网络数据一致性维护方法。该方法采用 Chord^[16]作为副本组管理协议,对 Chord 环所代表的 ID 空间不断分割,动态地建立更新消息传播树,从而达到副本数据的强一致性、更新信息冗余少、容错性能高的目的。

4 副本撤销

引起副本撤销的原因通常有:副本的生命周期结束;副本被访问的频率很低;副本所在节点存储空间不够;副本所在节点的处理能力达到极限。

如果给节点所在副本制定生命周期,则在生命周

期结束时就撤销副本;当副本的需求度不够,即被访问频率很低,应予以撤销;如果节点需要接纳一个新的副本,而本身存储空间不够,则会从已有副本中撤销一个或多个副本,直到能够存储该副本;如果节点的处理能力已达到极限,有时会新建一份副本到其它节点上以分担负载,有时会选择撤销副本。

4.1 周期保存法

文献[1]采用的是周期保存法:在副本创建时就给其一个初始值,每当计时周期到了就减 1,到该值变为 0 时,不论副本的访问频率或它所在节点的利用率的高低都撤销该副本,在未变 0 之前,如果检测到该副本几乎未被访问,则直接撤销。

4.2 最久未访问法

最久未访问法(Least Recently Used, LRU)是一种比较简单直接的撤销方法:每个节点自己维护一个 LRU 队列,队列中包含了节点上的所有文件。新产生的文件和副本被加到队列的尾部。当一个文件被客户端访问时,它也被从 LRU 队列中抽取出来放到队列的尾部。当需要删除一个文件时,从队列首部的文件开始,逐个删除,直至磁盘剩余空间达到新文件的存储要求。

文献[19]认为生成一个副本是代价比较高的一件事,所以建议尽量避免无谓地副本删除。在撤销副本时使用 LRU 方法并结合考虑副本的重要程度来决定撤销对象可以减少得不偿失的副本撤销。该文描述了当迎接新副本而节点存储空间不够时采取的撤销策略。其过程是:将新副本与节点 LRU 队列中的第一个副本开始比较重要程度,第一个重要程度低于新副本的副本将会被撤销,且将新副本放置在队列的队尾,如果比较完后无副本被撤销,则取消新副本的存储。

两种撤销方法中的周期保存方法在国外没有被广泛采用,其中的原因可能是:不同环境下数据的利用率不尽相同,有些数据的访问频率长期保持在一个波动不大的值上,而有些数据可能只是在短期被大量访问,副本生命周期的制定会限制副本被有效地利用。而最久未访问方法则显得更加灵活,可以降低副本被误删的概率。

5 结束语

副本管理问题是现在的研究热点,通过对副本本身进行高效管理和一些相关技术的改进,使大众所处的网络环境能朝更高效、更可用和更安全的方向发展。

副本是一群需要实际存储空间的实体,能引发大量访问请求,继而影响网络中节点的利用率、带宽消耗和数据一致性维护等方面。如何为一个局域网乃至整

个互联网制定出一套实用的副本管理机制是一项重要任务。其中,数据的副本数量、副本的分布、副本的定位和副本的数据一致性对网络的总体性能影响较大,因而是副本问题研究的重点。

参考文献:

[1] 葛建清. 异质结构化对等网络动态副本访问负载均衡策略研究[D]. 上海:华东师范大学,2010.

[2] 冯国富,张金城,陆桑璐,等. 无结构覆盖网络中面向搜索范围最小化的副本分布[J]. 计算机学报,2011,34(4):628-635.

[3] Lv Q, Cao P, Cohen E, et al. Search and Replication in Unstructured Peer-to-Peer Networks [C]//16th International Conference on Supercomputing. New York:ACM Press,2002.

[4] Bassam A A, Chen W, Zhou B B, et al. Effects of Replica Placement Algorithms on Performance of Structured Overlay Networks[C]//Parallel and Distributed Processing Symposium. Nice:IPDPS,2007.

[5] Gopalakrishnan V, Silaghi B, Bhattachar B, et al. Adaptive replication in peer-to-peer systems[C]//Proceedings of the 24th International Conference on Distributed Computing Systems. Japan:IEEE Press,2004.

[6] 蒋砚军,马华东,张 海. 网格中热点服务的多副本部署策略[J]. 北京邮电大学学报(自然科学版),2007,30(2):89-92.

[7] 蒋砚军,马华东,张 海. 网格中多副本数据的后向预测调度算法[J]. 华中科技大学学报,2006,34(ZI):67-70.

[8] Rao A, Lakshminarayanan K, Surana S, et al. Load Balancing in Structured P2P Systems[C]//Proc of International Workshop on Peer-to-Peer Systems. Berlin:IPTPS,2003.

[9] 陈建英,刘心松. 基于多维度 QoS 约束的大规模企业信息副本定位方法[J]. 计算机集成制造系统,2011,17(1):171

-176.

[10] 李美安,刘心松,王 征. 多副本访问控制的分层结构分布式互斥算法[J]. 计算机工程,2006,32(9):112-114.

[11] Lamport L. Paxos made simple [J]. ACM SIGACT News, 2001,32(4):18-25.

[12] Ychellboy. 分布式一致性 Paxos 算法回顾[EB/OL]. 2010-04-05. <http://kb.cnblogs.com/a/1704883/>.

[13] 刘 鹏. 云计算[M]. 第2版. 北京:电子工业出版社, 2011:25-51.

[14] 李振宇,谢高岗,李忠诚. PATCOM:基于分割树的无结构 P2P 系统一致性维护方法[J]. 计算机学报,2007,30(9):1500-1510.

[15] Ratnasamy S, Francis P, Handley M, et al. A salable content addressable network[C]//Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications. San Diego:SIGCOMM,2001:161-172.

[16] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer to peer lookup service for internet applications[C]//ACM SIGCOMM 2001. San Deigo:SIGCOMM,2001:149-160.

[17] Dabek F, Kaashoek M F, Karger D, et al. Wide area cooperative storage with CFS [C]//Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP). Banff: SOSP,2001:202-215.

[18] Datta A, Hauswirth M, Aberer K. Updates in highly unreliable, replicated peer to peer systems[C]//Proceedings of the 23rd International Conference on Distributed Computing Systems. Washington:ICDCS,2003:76-85.

[19] 陈 赓,余宏亮,张 堃. 对等网络中基于位置信息和文件流行度的自适应副本管理机制算法[J]. 计算机学报, 2009,32(10):1927-1937.

(上接第244页)

25(4):260-262.

[2] NVIDIA. NVIDIA CUDA Programming Guide ver. 2.1. 2008 [EB/OL]. 2008. <http://developer.download.nvidia.com/compute/cuda>.

[3] Nguyen H. GPU Gems 3[M]. 杨柏林,陈根浪,王 聪译. 北京:清华大学出版社,2010.

[4] Shreiner D. The Khronos OpenGL ARB Working Group. OpenGL Programming Guide[M]. 李 军,徐 波译. 7th ed. 北京:机械工业出版社,2010:424-427.

[5] Cook D L, Ioannidis J, Keromytis A D, et al. CryptoGraphics: Secret key cryptography using graphics cards [C]//LNCS 3376. [s. l.]:[s. n.],2005:334-350.

[6] 仇德元. GPGPU 编程技术[M]. 北京:机械工业出版社, 2011:19-20.

[7] 维基百科. 高级加密标准[S/OL]. 2012-01-16 [2012-

04]. <http://zh.wikipedia.org/wiki>.

[8] NIST. Advanced Encryption Standard (AES) [S]. USA: FIPS,2001.

[9] 吴亚联,段 斌. AES 密码计算构建的设计及应用[J]. 计算机工程,2005,31(21):181-186.

[10] 曹华平,罗守山,温巧燕,等. AES 算法轮密钥与种子密钥之间的关系研究[J]. 北京邮电大学学报,2002,25(4):47-50.

[11] Wang F, Qiu J, Yang J, et al. Hadoop high availability through metadata replication[C]//Proceeding of the First International Workshop on Cloud Data Management. Hong Kong, China: [s. n.],2009.

[12] Rost R J. OpengGL Shading Language[M]. 北京:人民邮电出版社,2006:43-62.