计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
49(Suppl.): 276-280, 2012

分布式文件系统副本一致性检测研究

田怡萌　李小勇　刘海涛
(上海交通大学信息安全工程学院　上海　200240)
(tym@sjtu.edu.cn)

Replica Consistency Detection in Distributed File System

Tian Yimeng, Li Xiaoyong, and Liu Haitao
(*School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai* 200240)

Abstract　In large-scale distributed file systems, replica consistency control not only requires consistency protocols, but also needs to detect and repair inconsistencies in time. This paper focuses on the on-line detection of replica consistency. We discuss almost all inconsistency problems in object-based large-scale distributed file system. Then we elaborate three key issues of on-line consistency detection, that is, how to decrease the detection influence on system performance, ensure detection timeliness and accuracy. According to the three key issues, we present relevant solutions. Simulation of store status reduces the detection load to the system; Pre-detection enables the system to detect inconsistencies timely; Detection of static data can effectively avoid false detection and ensure the accuracy of detection.

Key words　distributed file system; object storage; replica consistency; consistency protocols; on-line detection

摘　要　在大规模分布式文件系统中,副本一致性的实现既需要一致性协议模型的保障,也需要系统能够及时检测出已产生的不一致并予以修复.这里主要研究副本一致性的在线检测技术,全面分析了基于对象存储的大规模分布式文件系统中副本不一致的所有可能情况,深入阐述了在线一致性检测所面临的3个关键问题,即如何减小检测过程对系统性能的影响、保证检测的及时性、以及保证检测的准确性.针对这3个关键问题,总结提出了相应的解决方法.其中,模拟存储状态的方法,减轻了检测过程给系统带来的负荷;预检测的方法,增强了检测出系统不一致错误的及时性;检测静态数据的方法,有效避免了误检和漏检,保证了在线检测的准确性.

关键词　分布式文件系统;对象存储;副本一致性;一致性协议;在线检测

中图法分类号　TP316.4

分布式文件系统中广泛采用数据复制技术来提高系统的可靠性及其性能.通过存放多个副本来保证数据的安全可靠,当一个副本被破坏后,只需使用另一个副本便可以正常地访问数据.此外,当系统规模较大时,采用复制技术在不同节点上存储同一文件的副本有助于实现系统的负载均衡,从而提高了整体性能.然而,复制技术为系统提供快速响应能力和错误恢复能力的同时,也引入了副本管理和一致

收稿日期:2012-01-04
基金项目:上海市科学技术委员会重大基金项目(10DZ1500200)

性控制的新难题.

通常,分布式文件系统会采用一致性协议模型来避免不一致的出现,例如强一致性协议[1]、乐观一致性协议[2-3]等.目前对于一致性控制的相关研究大多围绕如何设计最优的一致性协议模型,平衡分布式系统的可用性和一致性[4-6].通过一致性协议模型来保障一致性的这种策略,主要是为了预防不一致错误的产生,本文称之为预防性策略.

在大规模分布式文件系统中,节点、部件数量庞大,网络通信及系统软硬件故障的出现不再是小概率事件.如果采用严格一致性协议模型,保证分布式事务操作的原子性,就会增加系统的复杂度,降低性能.如果采用非严格的一致性模型,如乐观一致性模型等,允许系统出现短时间的不一致,就会因为故障的频繁出现导致不一致问题不能及时处理,最终造成错误累积.因此,在大规模分布式文件系统中,不仅需要一致性协议模型来保障副本的一致性,还有必要实现一种检测机制,对系统进行持续监控,及时检测已经出现的不一致错误,并快速修复.这种一致性保障策略,本文称之为修复性策略.目前国内外对这方面的研究还比较少,未见论述、研究该问题的参考文献.

本文以我们设计开发的碧海分布式文件系统(blueocean file system, BOFS)为例,全面分析基于对象存储的大规模分布式文件系统中副本不一致的现象,指出实现在线检测面临的关键技术问题,并提出了相应的解决方法.

1 副本不一致问题描述

本节以我们开发的碧海分布式文件系统为例,对系统中可能出现的副本不一致现象进行描述.

碧海分布式文件系统是基于对象存储的大规模分布式文件系统,其架构如图1所示.存储系统由管理节点(management node, MN)和多个数据节点(data node, DN)组成.系统中的文件被分割成固定大小的数据块(chunk),每个chunk有多个副本以实现数据保护和高可用.DN主要负责这些chunk副本的物理存储.MN主要存储命名空间和文件元数据,其中包括文件与chunk的对应关系以及chunk副本的存放位置.

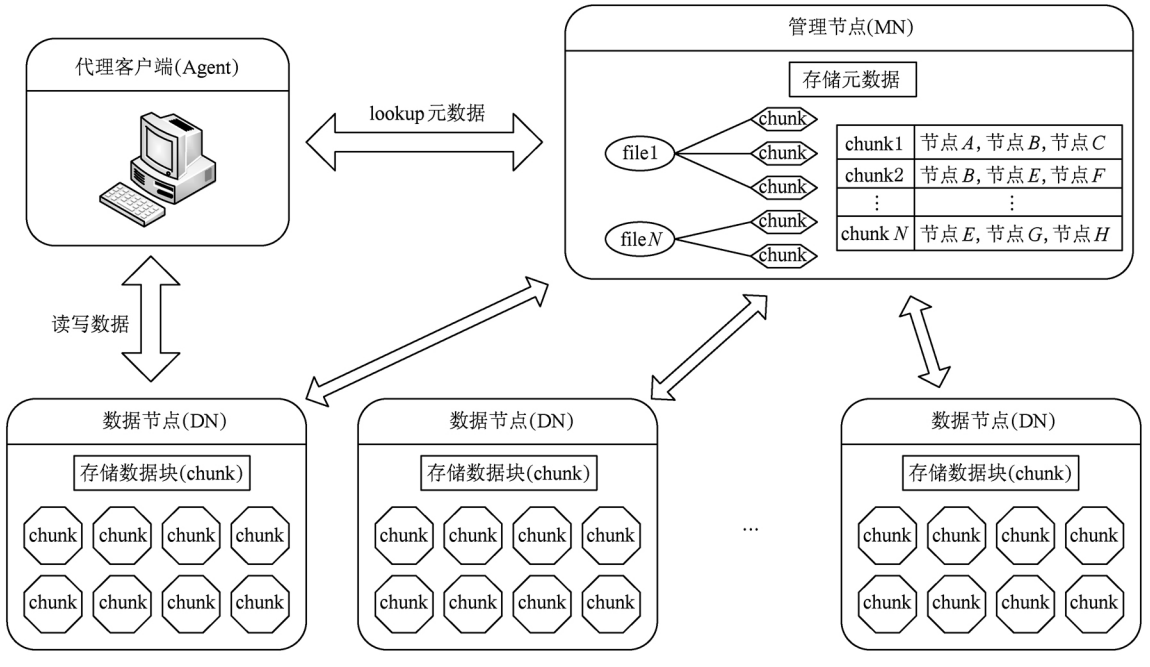

图1 碧海分布式文件系统架构

在碧海分布式文件系统中,其副本一致性问题主要包括2个方面:一是各个副本之间的一致性;二是MN元数据与DN实际存储chunk副本的一致性.

第1类问题主要表现在存储于不同DN上的chunk副本数据内容的不一致.

第2类问题主要表现为以下几种情况.

1) 副本丢失:MN元数据记录chunk副本存储在某个DN上,但该DN并未实际存储该副本;

2) 副本孤儿:DN上存在某个chunk副本,但

MN 上并未有文件指向该副本;

3) 副本位置错误:MN 元数据记录了 chunk 副本存储在某个 DN-A 上,然而实际该副本却存储在另一个节点 DN-B 上.

对于第 1 类副本之间的数据一致性问题,我们采用多个副本同步写入,以及为每个副本保存版本号(version number)的方式解决. 这类问题在本文中不进行深入探讨.

第 2 类不一致问题涉及到 MN 与 DN 之间的交互,比较复杂,也是系统检测不一致的关键. 接下来将着重分析这类问题的解决方案.

2 在线检测的关键问题

本文的主要研究对象是上节提出的第 2 类问题的检测. 系统一致性检测可以有 2 种方式,离线检测和在线检测.

离线检测是指在整个系统停止运行、各组件处于静止状态时进行一致性检测. 其原理类似于 Unix 文件系统中广泛使用的文件系统修复工具 fsck[7]. 这种检测方式比较容易实现,但是对于一个大规模分布式文件系统来说却不现实. 因为离线状态会大大降低系统的业务连续性,给用户带来极大的不便. 因此一致性检测应以系统持续在线运行为前提.

在线检测是指系统在线运行时,在尽量减小对正常业务影响的前提下,通过实时监测、周期性检测等方法,尽早发现系统中的数据不一致错误,并及时修复,防止错误累积.

上述在线检测的描述性定义中指出了实现在线检测需要面临的 3 个关键问题:

一是尽量减小对系统性能的影响. 在线检测不能为系统带来过多的资源消耗压力,影响系统正常运行的性能.

二是保证检测的及时性. 在线检测需要及时发现系统的不一致错误,防止错误累积,最理想的情况是一旦出现不一致,就能立即被检测出来,但这个实现起来难度较大.

三是保证检测的准确性. 在线检测需要对系统进行全面检测,防止漏检. 也要能够分辨出不一致现象是系统的不一致错误,还是由于数据修改操作未完成而造成的不一致,防止误检.

3 在线检测的解决方案

要实现第 2 类问题的一致性检测,即 MN 元数据与 DN 数据的一致性检测,就需要二者交互信息,将彼此信息进行对比,才能实现全面检测,及时发现不一致错误. 下面将根据上节指出的关键问题,提出在线检测的解决方法.

3.1 模拟存储状态

在线检测时,为实现信息对比,MN 需要通过信息交互获取 DN 上的 chunk 列表. 而对于 DN 来说,如果每次交互前都遍历一遍磁盘获取 chunk 列表,将会对系统的正常运行产生很大的影响.

为了解决这个问题,可以在 DN 的内存中维护一个数据结构,记录该节点上所有 chunk 及其状态信息. 当节点上 chunk 存储状态发生改变时,内存中的数据结构也同步修改,因此这个数据结构可以作为 DN 上 chunk 实际存储的模拟. 信息交互时,DN 只需要遍历这个数据结构即可获取 chunk 列表,这便大大减轻了对系统性能的影响. 同样,在 MN 内存中也维护一个数据结构,用于存储从 DN 上获取的 chunk 位置信息,这个数据结构可以作为在 MN 上保存的对 chunk 实际存储位置的模拟.

这种解决方法本文称之为模拟存储状态. 在其他类似架构的基于对象存储的大规模分布式文件系统中也能看到这种设计思想,如 GoogleFS[8], HDFS[9], KFS[10] 等.

为了更形象地描述,我们画出存储状态的模拟关系图,如图 2 所示. 这种方法实际上是将整个系统的一致性检测工作分成了 3 个部分,分别由各个组件进行分担.

1) DN 内部:保证内存中记录 chunk 状态信息的数据结构与实际存储同步更新,定期进行一致性检测保证二者的一致性.

2) 节点间交互:遍历 DN 内存中数据结构,得到 chunk 列表信息. 将 chunk 列表发送给 MN,保存在 MN 内存中. 二者需要定期实现交互过程,保证 MN 内存中模拟 chunk 位置的数据结构与实际存储位置的一致性.

3) MN 内部:定期将持久存储的文件元数据信息与内存中 chunk 位置的模拟信息进行对比分析,完成一致性检测.

上述 3 部分中,1)和 3)的一致性检测是由节点内执行,具体检测过程可以根据节点当前负载状况进行调整. 对于 2)来说,这个过程既需要 DN 遍历获取 chunk 列表,又要通过网络传输进行信息交互,因此这部分周期性的交互过程对整个系统的影响最大.

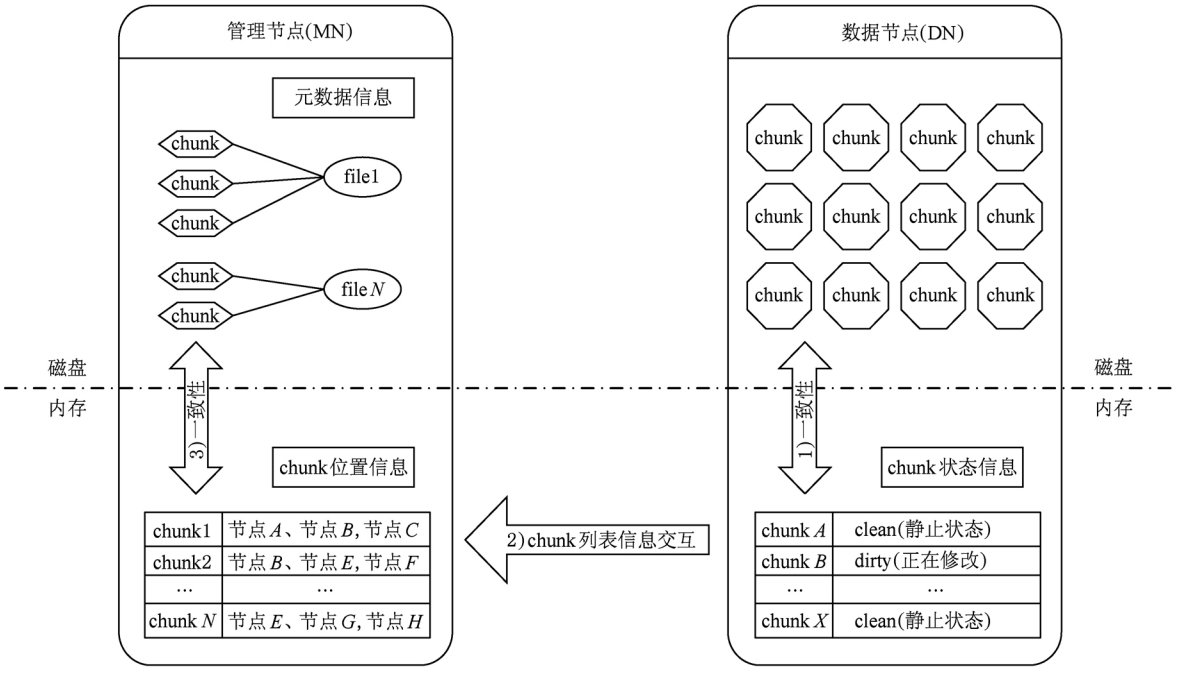

图2 一致性模拟关系图

如何设定周期性交互的时间间隔,权衡考虑检测过程对系统性能的影响和检测的及时性,还需要从理论上进行分析,在实验中加以验证.此外,优化交互过程,最小化交互信息,也是解决问题的一个重要研究方向.

3.2 预检测

3.1节中提到,MN与DN之间周期性交互时间间隔的设定会对系统性能和检测的及时性产生影响.时间间隔如果很大,系统出现的不一致错误就不能及时被发现,积累的错误就会越来越多.如果时间间隔很小,那么频繁的遍历扫描和信息交互会严重影响系统的正常运行.因此,当交互的时间间隔减小到一定程度时就不能再减小了,检测的及时性也就受到了限制.

为了解决这个问题,可以引入预检测的方法.预检测是指通过MN与DN之间频繁地交互简要信息,及时发现系统不一致错误的倾向,一旦发现,便立即启动全面细致的检测程序.在实际设计中,可以通过节点间的心跳(heartbeat)实现预检测的思想.

heartbeat是MN与DN之间频繁进行的信息交互,主要目的是使MN获知DN的运行情况,比如是否宕机、是否过载等,进而进行负载均衡等一系列的调整.

heartbeat信息中也包含对DN上chunk数量的统计,因此可以用于系统一致性的预检测.将chunk统计信息与MN上的元数据进行对比,如果发现不一致,MN便可以立即向DN发送请求,要求交互完整的chunk列表信息,从而启动全面的一致性检测程序.

上述通过chunk数量的对比进行预检测方法有些简陋.可以考虑在heartbeat信息中加入DN上所有chunk ID(chunk的全局唯一标识符)计算出的Hash值,将这个Hash值传送给MN,与MN元数据中记录的所有chunk ID计算出的Hash值进行对比,如发现不一致,则启动全面检测程序.

通过频繁的heartbeat交互进行预检测的方法,可以有效解决全面检测程序中信息交互"时间窗口过大"的问题,增强了在线检测的及时性.

3.3 检测静态数据

由于系统持续在线运行,系统中存在处于动态修改状态的数据,因此MN与DN之间的信息不一致可能有2种情况:一种是因修改操作未完成而造成的不一致,这是正常现象.另一种是系统出现了副本不一致的错误,这也是检测的目标.

在线检测程序须能分辨这2种情况,防止误检,避免漏检,保证一致性检测的准确性.针对这个问题,本文提出2个解决方案,其原理相似,即将在线检测的对象限定于已经完成修改操作的静态数据,而不包括正在修改的动态数据.

第1个方案是一致性检测时,数据信息的对比分析都只涉及当前静止的数据.而对于动态改变的数据,一旦修改完毕固定下来,便加入检测对象队

列,等待下一周期的一致性检测.由于系统中的数据不可能总处于修改状态,因此只要完成修改,检测程序就会将其覆盖,实现全面检测.

第2个方案是当系统启动一致性检测程序时,DN进入"轮休"状态.一旦DN进入"休息"状态,该节点上的任何修改操作都会被阻塞,此时所有的数据都处于静止状态,从而保证系统一致性检测的结果准确可靠.

第1个方案需要对系统当前的静态数据和动态修改数据进行标识,使在线检测可以分辨.第2个方案需要节点轮流停止运行,这可能会引入许多新的问题,因此其可行性还有待验证.

4 总 结

分布式文件系统副本一致性不仅需要一致性协议模型进行预防控制,还需要定期的一致性检测来最终保证.对于大规模分布式文件系统来说,各类故障使副本不一致的错误难以避免.因此只有定期进行检测才能及时发现已经产生的不一致,并予以修复,防止不一致错误不断累积.

考虑到大规模分布式文件系统的业务连续性,需要一致性检测在线运行,这就带来了3个挑战性的问题:尽量减小检测过程对系统性能的影响、保证检测的及时性和检测的准确性.为此,本文提出了相应的解决方法.其中,模拟存储状态的方法将整个一致性检测的过程分散到系统各个组件中进行,使得各个组件都有所分担.预检测的方法能够及时发现系统中不一致的现象,弥补完整一致性检测"时间窗口"的缺陷.检测静态数据的方法主要是为了分辨副本不一致现象的真假,保证检测的准确性.

本文提出的解决方法也适用于与碧海系统架构类似、基于对象存储的大规模分布式文件系统.在后面的工作中,需要通过仿真实验对这些解决方法加以验证,设计更优化的在线一致性检测的解决方案.

参 考 文 献

[1] Stonebraker M. Concurrency control and consistency of multiple copies of data in distributed INGRES. IEEE Trans on Software Engineering, 1979, 5(3): 188-194

[2] James J Kistler, Satyanarayanan M. Disconnected operation in the coda file system. ACM Trans on Computer Systems, 1992, 10(1): 3-25

[3] Terry D B, Theimer M M, Petersen K, et al. Managing update conflicts in bayou, a weakly connected replicated storage system //Proc of the 15th ACM SOSP. New York: ACM, 1995: 172-183

[4] Yu Haifeng, Vahda Amin. Design and evaluation of a continuous consistency model for replicated services //Proc of the 4th Conf on Symp on Operating System Design & Implementation. Berkeley: USENIX, 2000: 305-318

[5] Jehan. Voting with witnesses: A consistency scheme for replicated files //Proc of Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 1986: 606-612

[6] 谈华芳,孙丽丽,侯紫峰.一种多副本一致性控制方法.计算机工程,2006,32(11):52-54

[7] McKusick M K, Joy W N, Leffler S J, et al. Fsck-the Unix file system check program unix system manager's manual-4.3 BSD virtual VAX-11 version. Berkeley, CA: USENIX, 1986

[8] Ghemawat S, Gobioff H, Leung S T. The google file system //Proc of the 19th ACM SOSP. New York: ACM, 2003: 29-43

[9] Shvachko K, Kuang H, Radia S, et al. The hadoop distributed file system //Proc of IEEE 26th Symp on Mass Storage Systems and Technologies. Piscataway, NJ: IEEE, 2010: 1-10

[10] KFS. Kosmos File System. [2011-11-17]. http://kosmosfs.sourceforge.net

田怡萌 女,1988年生,硕士研究生,主要研究方向为存储系统、数据保护.

李小勇 男,1972年生,博士,副教授,主要研究方向为海量存储、高性能网络、嵌入式系统(xiaoyongli@sjtu.edu.cn).

刘海涛 男,1976年生,博士,讲师,主要研究方向为海量存储、数据保护(htliu@sjtu.edu.cn).