DOI:10.19358/j.issn.1674-7720.2014.12.026

一种基于文件预测的分布式缓存模型

张胜利,陈莉君
(西安邮电大学 计算机学院,陕西 西安 710061)

摘 要:在分布式存储中,客户端的数据访问请求并非完全随机,它是由程序或者用户的行为驱动,因此文件访问顺序是可以预测的;服务器端收到的访问请求在时间轴上也非平坦分布,因此服务器有时繁忙有时空闲。为此,提出了一种基于文件预测的分布式缓存模型,在客户端预测将要访问的文件,并利用服务器空闲时间传输预测文件。

关键词:分布式存储;文件预测;缓存技术
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1674-7720(2014)12-0069-04

A distributed caching model based on file prediction

Zhang Shengli, Chen Lijun
(School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710000, China)

Abstract: In distributed storage, data requests in client which driven by programs or users behavior are not completely random, so the requests can be predicted; the requests received by the server are not flat on the timeline, therefore the server sometimes busy or idle. Proposed a distributed caching model based on file prediction, predicts the files which will be requested in client and transfers predicting files during the idle time of server.

Key words: distributed storage; file prediction; cache technology

近年来,数据成爆炸式增长,传统存储方式已无法满足数据增长速度的要求,在此现状下,分布式存储技术得到了快速发展。限于成本与科技等原因,多数分布式存储都是利用大量廉价 PC 搭建而成[1],与传统的单机存储一样,在分布式存储系统中 I/O 也是制约其整体性能的一个瓶颈,因此相继提出了分布式缓存系统。

典型的分布式缓存系统如 Oracle coherence[2]、Memcached[3]、Terracotta[4],在弹性资源供给、可用性与可靠性、敏捷性与自适应性、多承租、数据管理、数据安全与隐私等方面已设计得较为完善,同时也有其不足之处:对数据的迁移是以容量均衡为目标而缺少对热点数据的处理;访问频率低的客户端缓存数据被换出导致资源劫持等[5]。

热点数据不均衡造成服务器之间接收到访问请求的不均衡,而客户的行为也有时间局域性,例如工作时间访问工作相关数据多,非工作时间访问娱乐数据多,这导致服务器收到的访问请求在时间上分布不平坦。预取可改善系统 I/O 的两个主要性能指标[6]:利用异步预取在程序使用文件之前将文件准备就绪,可对应用程序隐藏磁盘 I/O 延时;在服务器空闲时间使用预取可以提升服务器的使用率。

数据的访问请求并非完全随机,它是由用户或程序的行为驱动,存在特定的访问模式。当用户执行程序访问数据时,连续访问的一系列数据之间必然存在一定的关联[7],因此在客户端构建文件预测模型对将要使用的文件进行异步预取是可以实现的。为此提出了一种基于文件预测的分布式缓存模型 DLSDCM(DLS based Distributed Cache Model),在客户端建立由经典的文件预测模型 LS(Last Successor)改进而成的文件预测模型 DLS(Double Last Successor),并利用服务器的空闲时间进行预测请求数据的传输。此模型建立在其他分布式缓存系统之上,完善其资源劫持、热点数据分布不均衡等方面,同时也可作为单独的缓存模型使用。

1 文件预测模型

数据的访问请求并非完全随机,它是由用户或程序的行为驱动,用户执行某种应用程序去访问数据,连续访问的不同文件之间必然存在一定的关联。可构造出一种文件预测模型,通过对数据本体间的内在联系或者历

史访问记录进行分析并构造出预测数据库，依据预测数据对预测文件进行异步预读并缓存。当应用程序使用这些数据时，便可大幅度减少数据的访问延时，同时也减少了服务器空闲时间，提升了网络使用率。本文主要研究 LS 预测模型。

1.1 LS 文件预测模型

当用户访问一系列数据时，或多或少会重复上一次的访问顺序，因此 LS 模型是最常用也是最简单的文件预测模型，被多数预测系统采用。Linux 内核采用的预取算法亦是根据上次及本次的读请求进行顺序模式的匹配[8]。

但是 LS 文件预测模型在交替访问文件时就会完全失效，例如第一次访问顺序为文件 A、文件 B；第二次访问顺序为文件 A、文件 I；第三次又重复第一次顺序为文件 A、文件 B。对于这样的交替访问，使用 LS 模型预测文件 A 的后继则完全失效。若将预测的文件数扩大为 2 个，即对于每个文件同时预读其上一次访问的后继文件和上上一次访问的后继文件，则可避免交替访问的预测失效，据此本文提出了 DLS 文件预测模型。

1.2 DLS 文件预测模型

DLS 文件预测模型一次可预测 2 个文件：上次访问顺序中的后继和上上次访问顺序的后继，预测命中的概率将会有很大提高。图 1 所示为 DLS 模型对文件 A 后继的预测，图中文件 A、B、I、U、D 代表独立的文件，而非顺序文件。

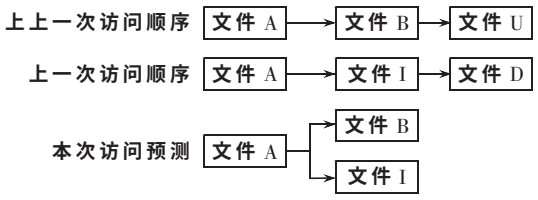

图 1 DLS 文件预测模型

由于一次预测 2 个文件，因此数据传输时也会有 2 个文件同时传输，参考使用概率图来预测未来文件访问的文件预测模型[9]，分别记录文件 B 和文件 I 的预测命中次数，并依命中次数来决定两个文件传输时各自占用的带宽比例，例如，记录中文件 B 命中了 40 次，文件 I 命中了 60 次，此时文件 B 占 40%带宽，文件 I 占 60%的带宽比例进行传输。

1.3 LS 和 DLS 两种文件预测模型对比

文件预测模型很多，每种预测模型都有其最适用的环境。下面从理论上对比 LS 文件预测模型和 DLS 文件预测模型在 DLSDCM 中的适用性。以图 1 为例对比 LS 文件预测模型和 DLS 文件预测模型的命中率和有效使用率：假设文件 B 和文件 I 命中率都是 20%，文件 B 和文件 I 的大小均为 1 MB。表 1 为服务器 I/O 空闲不大于传输 1 MB 数据所用时间的情况。

表 1 为使用 LS 文件预测模型和 DLS 文件预测模型的有效传输数据量相同，但是使用 DLS 文件预测模型却

表 1 DLS 与 LS 对比 1

模型	命中率/%	总传输数据量	有效传输数据量/KB
LS	20	1 MB	200
DLS	20+20	500 KB+500 KB	100+100

有 40%的概率传输有效数据，即在服务器 I/O 空闲小于等于传输 1 MB 数据时间的情况下，DLS 文件预测模型与 LS 文件预测模型相比优势在于稳定性高。表 2 为服务器 I/O 空闲可传输 2 MB 数据时的情况。

表 2 为两种模型的预测命中率不变，DLS 文件预测模型总传输数据量和有效传输数据量均为 LS 文件预测模型的 2 倍。由于 DLSDCM 是使用空闲网络时间进行数据的预传输，并不占用必要读请求数据的传输时间，因此服务器 I/O 空闲大于空闲临界值(传输 LS 预测文件全部数据的时间)时，DLS 文件预测模型相对于 LS 文件预测模型有较大的优势。

表 2 DLS 与 LS 对比 2

模型	命中率/%	总传输数据量/MB	有效传输数据量/KB
LS	20	1	200
DLS	20+20	1+1	200+200

由此可见，在 DLSDCM 中 DLS 文件预测模型较之 LS 文件预测模型在服务器 I/O 空闲不大于空闲临界值时，具有命中稳定性高的优势；而当在服务器 I/O 空闲大于空闲临界值时，DLS 文件预测模型优势就逐渐明显，这个优势在服务器空闲时间足够传输 DLS 文件预测模型所预测的两个文件时达到最大。

2 DLSDCM 设计原理

2.1 缓存模型理论基础

缓存是传输速率相差较大的两种实体(硬件或软件)之间的存储区域，用于存储低速实体中的热点数据或预读数据，以提升系统的反应速度[10]。

传统的缓存模型是在程序请求访问数据之后将数据缓存，基于数据使用的时间局域性，当再次使用数据时便可降低访问延时，这种策略属于被动式缓存。使用预取技术在程序请求访问数据之前将其读入，如果预测命中则将预读数据交由系统缓存管理(以下“缓存”指 DLSDCM 预读缓存，而“系统缓存”指支撑 DLSDCM 的分布式存储系统缓存或分布式缓存系统缓存)，这属于主动式缓存。主动式缓存填充了程序请求访问数据之前的空白时间，对于服务器有空闲时间的情况，是一个提升服务器使用率的策略，对于客户端来说可降低其请求数据的访问延时。

2.2 DLSDCM 的架构

DLSDCM 是基于客户端数据访问请求可以预测和服务器收到数据访问请求在时间轴上非平坦分布这两个前提来构建的。在客户端建立 DLS 预测模型，在服务器端增添一个预读请求队列并默认调度优先级低于 I/O 请求队列，以达到预读数据的传输在服务器 I/O 的空闲

时间进行，提升服务器使用效率[11]的目的（此处亦留下接口可手动修改优先权，调整特定客户端优先级）。DLSDCM 建立在分布式存储系统或分布式缓存系统之上，其缓存模型如图 2 所示。

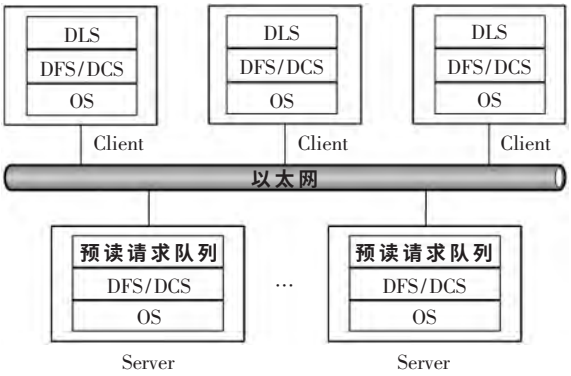

图 2 DLSDCM 缓存模型

3 DLSDCM 的实现

DLSDCM 的实现分为客户端的实现和服务器端的实现。客户端建立 DLS 预测模型，使用 DLS 预测模型维护一份预测数据和一份高访问频率文件记录，每次读请求发生时，通过查询预测数据对读请求文件所对应的预读文件进行异步预读请求操作；高访问频率文件记录作为预留接口为热点数据迁移提供热点数据信息[12]（此功能尚在开发中）；每个客户端在本地维护一个预读缓存，在本地磁盘维护一个缓存目录。服务器端维护调度预读请求队列并默认预读请求队列调度优先级低于 I/O 请求队列，响应客户端发送的预读相关的请求信号。

3.1 DLSDCM 客户端的实现

DLSDCM 客户端在每次读请求发生时，根据 DLS 文件预测模型所预测的数据向服务器发出预测文件的异步预读请求，并将过大的预读数据存储于本地磁盘。参考 Linux 内核的预取算法，其维护了两种类型的状态量：读历史和预读历史，在 DLS 预测模型中使用链表记录文件读历史，而每个文件所对应的节点又包含了节点文件所对应的预读文件和预读命中次数；使用数组记录高频率访问的文件及访问次数，留作热点数据迁移备用接口。客户端结构如图 3 所示。

每次读请求操作首先检查 DLS 预测数据是否有对应文件节点，再分别检查缓存和系统缓存中是否有对应文件，而后再次检查缓存和系统缓存中是否有预读文件，最后再将读请求文件信息写入 DLS。一次读请求的流程图如图 4 所示。

当服务器空闲时间足够、预读文件大小超过预读缓存容量时，执行将预读文件写入本地磁盘缓存目录的操作。缓存与磁盘缓存目录文件保留至下一次预读写操作。

3.2 DLSDCM 服务器端的实现

服务器端主要负责响应客户端的预读请求信号和预读请求队列的调度。在调度方面，服务器视所有客户端为平等优先级（可修改特定客户端的优先级），按照传统的先来先服务策略对预读请求进行调度[8]，只有在 I/O 请求队列为空时才执行预读请求队列调度。预读请求队列每个节点中包含两个文件信息，在被调度传输时同时传输两个文件，并根据相应的比例来分配传输带宽。响应请求信号方面，主要响应客户端的 3 种请求信号是：预读请求、预读请求转读请求、预读终止。对于这 3 种信号的处理如下：

（1）收到预读请求信号：将预读请求文件加入预读请求队列，并标记两个预读请求文件在传输时占据的带宽

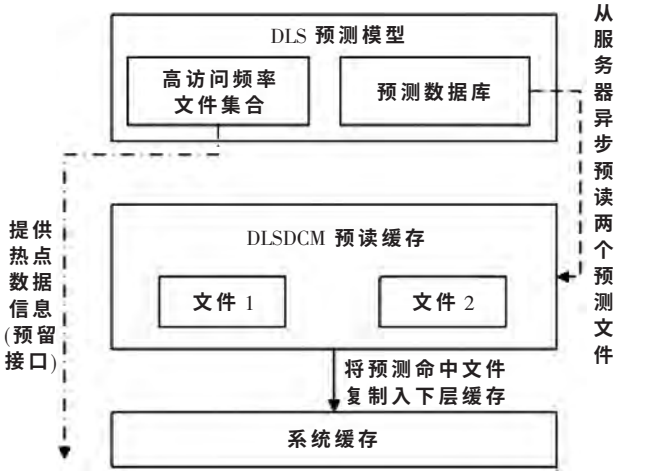

图 3 客户端结构

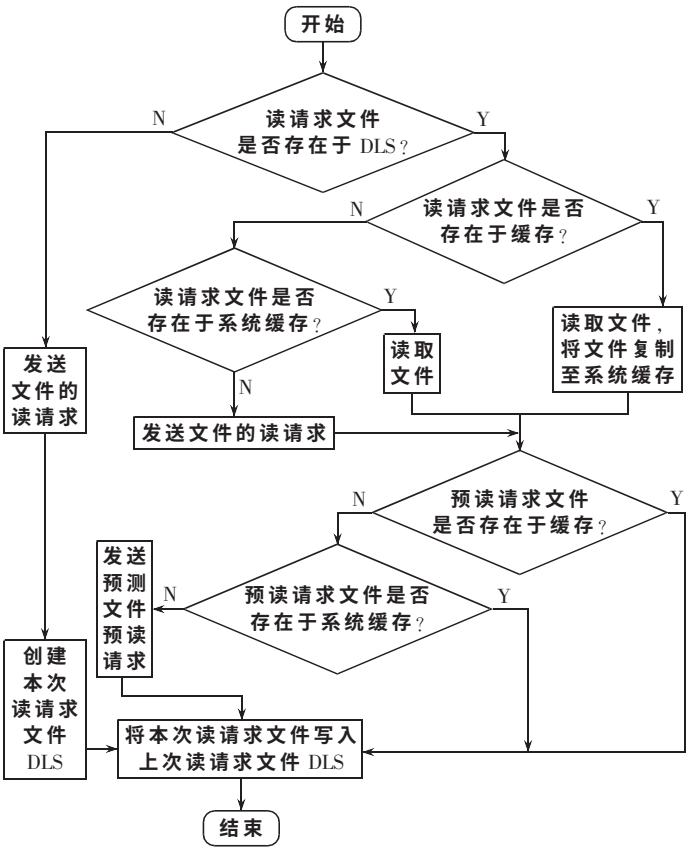

图 4 客户端读请求流程图

比例，队列中两个文件占用一个节点，在被调度时根据比例传输两个文件数据。

(2)收到预读请求转读请求信号：将对应文件加入I/O请求队列；将文件所在的预读请求队列节点删除。

(3)收到预读终止信号：终止数据传输，将预读文件所在节点从预读队列中删除。

DLSDCM的优点在于其可以建立在传统分布式缓存系统之上，在程序使用数据之前异步预读并将预读命中数据传给分布式缓存系统，填补了程序使用数据之前的空白时间；正在进行中的热点数据迁移工作可以填补缓存系统对热点系统分配不均衡的不足，提高了传统分布式缓存效率。

由于数据预读发生在服务器空闲时间，DLSDCM的缺点是预读的命中率问题提升了服务器的无效数据吞吐量，但此缺点并未影响整体分布式存储的性能；DLSDCM对客户端数据访问随机性强和更新速度快的场合适用性较差，对于服务器繁忙时提升效率较低。

参考文献

[1] 邓见光，潘晓衡，袁华强.云存储及其分布式文件系统研究[J].东莞理工学院学报，2012，19(5)：41-46.

[2] Oracle white paper. Platform-as-a-Service pfivate cloude with oracle fusion middleware[EB/OL].(2009-10)[2013-03]. http://www.oracle.com/us/technologies/cloud/036500.pdf.

[3] Wikipedia. Memcached[EB/OL].(2013-03-12)[2013-04]. http://en.wikipedia.org/wiki/memcached.

[4] Terracotta. Terracotta DSO documentation[EB/OL].(2012-08)[2013-04]. http://www.terracotta.org/confluence/display/docs/Home.

[5] 秦秀磊，张文博，魏峻，等.云计算环境下分布式缓存技术的现状与挑战[J].软件学报.2012，19(5)：1787-1803.

[6] SHRIVER E，SMALL C. Why does file system prefetching work?[C]. Proceedings of 1999 USENIX Annual. Technical Conference，1999：71-84.

[7] 刘爱贵，陈刚.一种基于用户的LNS文件预测模型[J].计算机工程与应用，2007，43(29)：14-17.

[8] 吴峰光，奚宏生，徐陈锋.一种支持并发访问流的文件预取算法[J].软件学报，2010，21(8)：1820-1833.

[9] GRIFFIOEN J，APPLETON R. Reducing file system latency using a predictive approach[C]. Proceedings of USENIX Summer Technical Conference，1994：197-207.

[10] BRYANT R E，O'HALLARON D R. Computer systems a programmer's perspective(英文版)[M].北京：电子工业出版社，2006.

[11] 姚念民，鞠九滨.过载服务器的性能研究[J].软件学报，2003，14(10)：1781-1786.

[12] 徐非，杨广文，鞠大鹏.基于peer-to-peer的分布式存储系统的设计[J].软件学报，2004，15(02)：268-277.

(收稿日期：2013-12-30)

作者简介：

张胜利，男，1982年生，硕士研究生，主要研究方向：存储及文件系统。

陈莉君，女，1964年生，教授，主要研究方向：系统软件与安全。

(上接第65页)

相对较轻的处罚。

本文在给出僵尸网络定义的基础上，从Bot、Botmaster、C&C这三个方面研究了僵尸网络是如何躲避检测机制的，使读者对僵尸网络以及僵尸网络是如何躲避检测有进一步的了解。防御者可以从僵尸网络躲避检测机制的反向入手研究如何检测僵尸网络。

参考文献

[1] 诸葛建伟，韩心慧，周勇林，等.僵尸网络研究[J].软件学报，2008，19(3)：702-713.

[2] Xie Yinglian，Yu Fang，ACHAN K，et al. Spamming botnets：signatures and characteristics[C]. Proceeding of ACM SIGCOMM 08，New York：ACM，2008：171-182.

[3] Gu Guofei，PERDISCT R，Zhang Junjie，et al. BotMiner：clustering analysis of network traffic for protocol-and structure-independent botnet detection[C]. Porceeding of the 17th USENIX Security Symposium，Berkeley，CA：USENIX，2008：269-286.

[4] 方滨兴，崔翔，王威.僵尸网络综述[J].计算机研究与发展，2011(8)：702-713.

[5] 康乐，李东，余翔湛.基于SVM的Fast-flux僵尸网络检测技术研究[J].智能计算机与应用，2011(6)：24-27.

(收稿日期：2014-01-18)

作者简介：

杨智兴，男，1988年生，硕士研究生，主要研究方向：信息安全。

(上接第68页)

of Management Sciences，1983，11(1)：91-95.

[7] Lian Zhigang，Gu Xingsheng，Jiao Bin. A Novel particle swarm optimization algorithm for permutation flow shop scheduling to minimize makespan[J]. Chaos，Solitons and Fractals，2008，35(5)：851-861.

(收稿日期：2014-02-23)

作者简介：

张丽萍，女，1987年生，硕士研究生，主要研究方向：嵌入式软件。