2011年 7月 第 7期
Vol. 32 No. 7 2011

分布式文件系统的动态负载均衡算法

张聪萍, 尹建伟

(浙江大学 计算机科学与技术学院, 杭州 310027)
E-mail: pczhang1987@126.com

摘 要: 为了解决分布式文件系统的负载均衡问题, 研究了多种负载均衡算法, 分析各种算法的优缺点. 综合了静态权重轮询算法和动态负载均衡算法两类算法的优点, 提出一种自适应的综合动态负载均衡算法, 它实现简单、降低了获取反馈信息的开销且不失实时性. 经过试验比较, 该算法能有效地降低平均响应时间和提高吞吐量, 负载的分配更为均衡.

关 键 词: 分布式文件系统; 负载均衡; 静态权重轮询; 动态负载均衡
中图分类号: TP311 文献标识码: A 文 章 编 号: 1000-1220(2011)07-1424-03

Dynamic Load Balancing Algorithm of Distributed File System

ZHANG Cong-ping, YIN Jian-wei
(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

Abstract: In order to solve load balancing of the distributed file system, I did some research into the load balancing algorithms, and analyzed the advantages and disadvantages of them. Combining static weighted round robin algorithm and dynamic load balancing algorithm, this paper proposed an adaptive comprehensive dynamic load balancing. It is simple, low cost and effective. Aftermore tests, I found that the algorithm can effectively reduce the average response time and increase throughput. It is better than static weighted round robin algorithm and dynamic load balancing algorithm.

Key words: distributed file system; load balancing; static weighted round robin; dynamic load balancing

1 引 言

随着互联网的发展和用户宽带接入的普及, 一些大型网站产生了海量的数据存储, 采用普通的用户访问方式, 这些海量数据占用昂贵的存储设备, 给服务器和带宽带来巨大压力. 为了解决这种问题, 提出了分布式文件系统.

所谓分布式文件系统, 就是将数据分散存储在多台通过网络或通信线路连接起来的相互独立而又相互合作的计算机(又称节点)上. 分布式文件系统采用可扩展的主从式系统结构, 利用多台存储服务器分担存储负荷, 它不但提高了系统的可靠性、可用性和存取效率, 还易于扩展. 然而, 对于分布式文件系统, 由于各节点处理能力存在差异, 当系统运行一段时间后、某些节点分配的任务很多(称为重载), 而另外一些节点却是空闲的(称为轻载或空载). 要避免这种现象发生. 必须采取负载均衡措施.

2 相关工作

为了使多个节点能很好的共同完成任务, 消除或避免现有网络负载分布不均、数据流量拥挤、反应时间长的瓶颈, 人们提出了多种负载均衡算法, 主要分为两大类: 静态负载均衡算法和动态负载均衡算法. 静态负载均衡算法主要有轮询(Round Robin)、比率(Ratio)、优先权(Priority)等[7,8].

◦ 轮询, 将请求依次顺序循环地连接每个服务器节点;

◦ 比率, 给每个服务器节点分配一个加权值为比例, 根据这个比例, 把用户的请求分配到每个服务器;

◦ 优先权, 给所有服务器分组, 给每个组定义优先权, 把用户的请求, 分配给优先级最高的服务器组(在同一组内, 采用轮询或比率算法, 分配用户的请求); 当最高优先级中所有服务器出现故障, 才将请求送给次优先级的服务器组. 这种方式, 实际为用户提供一种热备份的方式.

这些算法实现简单, 但是并没有考虑到各节点的实际负载情况, 从而不能很好的取得负载均衡. 而动态负载均衡算法引进了一种动态反馈机制, 各服务器节点向负载均衡节点周期地反馈其实时负载情况, 负载均衡节点根据此信息分配客户请求. 但是, 此类算法的问题在于收集各个服务器节点的实时负载信息会导致额外的开销. 另外, 如果反馈信息不及时可能会使分配算法无效, 从而破坏负载均衡.

针对以上问题以及分布式文件系统的实际应用环境, 提出一种自适应的综合动态负载均衡算法. 它综合地使用静态负载均衡算法和动态负载均衡算法, 在负载均衡服务器上采用实现简单的算法, 在存储服务器节点上利用其自我监测能力实现自适应的反馈机制, 这种反馈机制降低了获取存储服

收稿日期: 2010-04-12 收稿日期: 2010-05-10 基金项目: 国家自然科学基金项目(60703042)资助; 国家"八六三"高技术研究发展计划项目(2007AA01Z124)资助; 浙江省自然科学基金项目(Y106045)资助; 国家核高基重大专项项目(2009ZX01043-003-003)资助. 作者简介: 张聪萍, 女, 1987年生, 硕士研究生, 研究方向为分布式系统、海量存储系统; 尹建伟, 男, 1974年生, 博士, 博士生导师, 研究方向为分布式计算与平台软件、服务计算与软件系统结构、信息集成、信息安全.

务器节点负载信息的开销且不失实时性.

3　自适应的综合动态负载均衡算法

3.1　分布式文件系统

分布式文件系统采用可扩展的主从式系统结构, 整个系统由三部分构成: 主服务器、存储服务器以及客户端. 主服务器主要负责元数据管理、存储服务器节点管理、负载均衡、副本管理等; 存储服务器, 又称从服务器, 负责文件的数据的存储、负载状况自我监测、发送心跳信息等, 其系统结构图如图1所示. 首先, 客户端发送请求到主服务器, 主服务器进行元数据查找或创建等操作后, 根据一定的负载均衡算法选择合适的存储服务器地址返回给客户端; 接下来客户端直接同存储服务器进行数据传输操作.

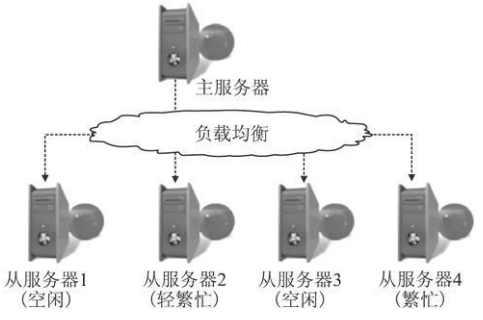

图 1　分布式文件系统结构图

Fig. 1　Architecture of distributed file system

在分布式文件系统中, 客户端主要进行的是文件的上传、下载等操作. 文件上传时, 主服务器根据负载均衡算法选择存储服务器集群中当前性能最优的存储服务器节点. 如果仅限于此, 当遇到高并发下载同一文件时, 则存储该份文件的存储服务器必然会超载甚至崩溃. 解决方案是采用根据文件访问流行度的副本生成策略, 将文件根据访问频率复制 N个(具体 N值取决于副本生成策略)副本存储到其它存储服务器节点上. 在此基础上进行文件下载, 主服务器根据负载均衡算法在多台具有文件副本的存储服务器节点中选择最优节点.

3.2　存储服务器的自适应反馈机制

当前提出的负载均衡技术一般来说都是从服务器的负载状况的定义、如何获取以及获取后如何处理的这三个方面来寻求解决办法[1]. 如何获取以及获取后如何处理均由负载均衡器来处理, 而在分布式文件系统中, 担当负载均衡器作用的是主服务器, 主服务器本身要处理大量的客户请求, 过于复杂的负载均衡策略会大大增加主服务器的负担. 为此, 我们转换考虑的角度, 从存储服务器出发提出一种存储服务器的自适应反馈机制.

在分布式文件系统中, 每个存储服务器节点均有负载状况监测模块监测节点的磁盘剩余空间、网络利用率、CPU利用率等信息. 我们根据这些信息计算出存储服务器的实时负载情况 -负载变量. 为了描述负载变量, 假设存储服务器集群系统由 n台服务器组成, 表示成 $S=\{S_1, S_2, \cdots, S_n\}$. 其中服务器 S_i在某一时刻 t的网络负载为 $Net(S_i)$, CPU负载为 $CPU(S_i)$, 服务器 S_i的负载变量为 $LBV(S_i)$, 则有:

$$LBV(S_i)=[\lambda,\ 1-\lambda]\cdot\begin{bmatrix} Net(S_i) \\ CPU(S_i) \end{bmatrix} \tag{1}$$

其中 λ为负载影响因子, 它取决于 $Net(S_i)$对负载变量 $LBV(S_i)$的影响程度.

将整个负载状况分成三个区间(0, 0.45], (0.45, 0.85], (0.85, 1], 分别对应三种负载现象: 轻载、适载、重载. 相应地, 主服务器的节点管理模块将存储服务器节点分成三组队列, 分别为轻载队列、适载队列、重载队列, 我们分别用集合 LQ, SQ, WQ表示, 其中 $S=LQ\cup SQ\cup WQ$. 其对应关系如下:

$$LBV(S_i)\in\begin{cases}(0, 0.45], & S_i\subset LQ\\ (0.45, 0.85], & S_i\subset SQ\\ (0.85, 1], & S_i\subset WQ\end{cases} \tag{2}$$

存储服务器摒弃了传统的反馈方法: 周期性反馈负载变量 $LBV(S_i)$给主服务器, 而是当 $LBV(S_i)$从一个区间跳至另一个区间时才向主服务器反馈警报信息. 这样大大降低了反馈机制所占用的网络开销, 同时也可及时的向主服务器反映当前的负载状况.

3.3　主服务器的综合负载均衡算法

在 3.2节中, 提到主服务器的节点管理模块将存储服务器节点分成轻载、适载、重载三组队列, 队列中保存了所有有效存储服务器节点的信息. 主服务器根据存储节点反馈的警报信息对节点进行出队、入队等操作.

分布式文件系统支持同构、异构等集群, 在异构集群中实施负载均衡必须要考虑存储服务器节点本身的处理能力. 对于存储服务器节点 S_i的处理能力为 $W(S_i)$, 我们主要根据其本身的硬件配置来计算, 考察的指标主要有 CPU的类型, CPU数量 $N(S_i)$、系统最大进程数 $P(S_i)$、网络吞吐量 $NT(S_i)$、磁盘有效空间大小 $DS(S_i)$、磁盘 IO速率 $DI(S_i)$、内存容量 $M(S_i)$. 其计算公式如下所示:

$$W(S_i)=[\partial_1, \partial_2, \partial_3, \partial_4, \partial_5, \partial_6]\cdot\begin{bmatrix} N(S_i) \\ P(S_i) \\ NT(S_i) \\ DS(S_i) \\ DI(S_i) \\ M(S_i) \end{bmatrix} \tag{3}$$

其中向量 $\partial=(\partial_1, \partial_2, \partial_3, \partial_4, \partial_5, \partial_6)$, $\sum\partial=1$为处理能力影响因子, 它的各个元素值分别取决于对应的指标对 $W(S_i)$的影响程度.

表 1　主服务器的系统配置

Table 1　System configuration of main server

选项	参数值
CPU	Intel Pentium Dual CPU E2160@1.80GHz
内存	2G
网络	1个 100M网卡
操作系统	Windows Server Enterprise 2008 SP1

主服务器的负载均衡算法主要思想为: 在队列与队列间, 我们采用优先权算法, 其中 LQ优先级最高、SQ次之、WQ最低; 在队列内, 我们采用权重轮询均衡算法, 根据服务器的不同处理能力, 给每个服务器分配不同的权值, 使其能够接受相应权值数的服务请求. 例如: 服务器 A的权值被设计成 1, B的权值是 3, C的权值是 6, 则服务器 A, B, C将

分别接受到 10%、30%、60%的服务请求.此种均衡算法能确保高性能的服务器得到更多的使用率,避免低性能的服务器负载过重.根据该算法思想,具体的请求处理过程表述如下:

(1)如果 LQ≠∅,选择集合 M,M⊆LQ;否则,如果 SQ≠∅,选择集合 M,M≠SQ;否则,如果 WQ≠∅,选择集合 M,M⊆WQ;否则,返回失败信息(无有效存储服务器其节点).

(2)计算集合 M中各个节点的的分配请求的概率 P(Si),这是根据节点的处理能力 W(Si)来计算的.如下所示:

$$P(Si)=W(Si)\Big/\sum_{i=1}^{n}W(Si),\quad i=0,1,\cdots,n;\ Si\subseteq M,\ Si\subseteq M \tag{4}$$

(3)根据集合 M中各个节点的概率,将请求分配到 M中一个节点上.

4 算法分析

为了验证算法的有效性,我们在研发的分布式文件系统 JTangFS中进行了实验研究.分别采用静态权重轮询、周期性动态反馈、自适应综合算法来实现 JTangFS的负载均衡,模拟 50-300客户端并发进行下载大小均为 100M的多个文件(配置文件的副本数为 3,即 3台存储服务器器中均有同一文件的副本),然后比较三种算法下的平均响应时间和吞吐量.

测试中采用了 14台 PC机,其中 1台主服务器、3台存储服务器、10台客户端.主服务器的系统配置如上页表 1所示.

三台存储服务器的系统配置分别如表 2、表 3、表 4所示.10台客户端分别为普通 PC机,每台客户端并发模拟 5~30个文件下载进程.实验结果如图 2、图 3所示.

表 2 存储服务器 A的系统配置
Table 2 System configuration of storage server A

选项	参数值
CPU	Intel Pentium Dual CPU E2180 @ 2.00GHz
内存	2G
网络	1个 1000M网卡
磁盘	写速度:70MB/s读速度:77MB/s读写速度:31MB/s
操作系统	Red Hat Enterprise Linux Server release 5.2 Linux version 2.6.18-92.el5

表 3 存储服务器 B的系统配置
Table 3 System configuration of storage server B

选项	参数值
CPU	Intel Pentium 4 CPU 3.06GHz
内存	1G
网络	1个 1000M网卡
磁盘	写速度:45MB/s读速度:47MB/s读写速度:20MB/s
操作系统	Red Hat Enterprise Linux Server release 5.2 Linux version 2.6.18-92.el5

表 4 存储服务器 C的系统配置
Table 4 System configuration of storage server C

选项	参数值
CPU	Intel Pentium 4 CPU 1.60GHz
内存	1G
网络	1个 100M网卡
磁盘	写速度:50MB/s读速度:42MB/s读写速度:18.5MB/s
操作系统	Red Hat Enterprise Linux Server release 5.2 Linux version 2.6.18-92.el5

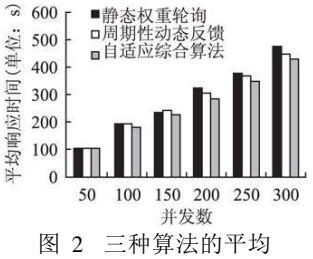

图 2 三种算法的平均响应时间比较

Fig.2 Compare the average response time of three algorithms

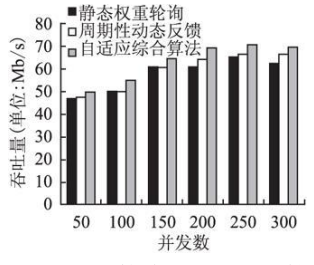

图 3 三种算法的吞吐量比较

Fig.3 Compare throughput of three algorithms

这种性能改善不太明显;当并发数增大一些后,新算法的优势才逐渐表现出来.这种现象说明:在系统的负载较轻时,按照三种算法分配请求都可以达到均衡负载的目的;当各服务器都达到一定的负荷(并发数达到一定数量)后,新算法对负载的分配更为均衡.

5 结束语

针对静态负载均衡算法和动态负载均衡算法所存在的不足,本文综合两者的优点以及算法应用环境提出了一种自适应的综合动态负载均衡算法.它在主服务器端维护三组队列,在队列间采用优先权算法,保证了存储服务器的负载状态比较均衡的在同一队列内;在队列内,采用权重轮询算法,分配请求按各节点的能力均匀分配.在存储服务器中,改善周期性反馈机制,采用区间变换时再反馈,这样大大降低了网络开销,同时也可及时的反映当前的负载状况.

References:

[1] Xue Jun, Li Zeng-zhi, Wang Yun-lan. Development of technology of load balancing[J]. Mini-Micro Systems, 2003, 24(12): 2100-2103.

[2] Colajanni M, Yu P S. A performance study of robust load sharing strategies for distributed heterogeneous web server systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(2): 398-414.

[3] Eric Dean Katz, Michelle Butler, Robert McGrath. A scalable HTTP server: the NCSA prototype[J]. Computer Networks and ISDN Systems, 1994, 27(2): 155-164.

[4] Thomas T Kwan, Robert E McGrath, Daniel A Reed. Ncsa's world wide web server: design and performance[J]. IEEE Computer, 1995, 28(11): 68-74.

[5] Luis Aversa, Azer Bestavros. Load balancing a cluster of web servers using distributed packet rewriting[C]. IEEE International Performance, Computing and Communications Conference, 2000.

[6] Philip S Yu, Daniel M Dias. Analysis of task assignment policies in scalable distributed web-server systems[J]. IEEE Transations on Parallel and Distributed Systems, 1998, 9(6): 585-600.

[7] The guide of load balancing (Part One)[EB/OL]. http://cisco.chinaitlab.com/case/12415.html, 2003-05.

[8] The guide of load balancing (Part Two)[EB/OL]. http://cisco.chinaitlab.com/case/12416.html, 2003-05.

附中文参考文献:

[1] 薛 军,李增智,王云岚. 负载均衡技术的发展[J]. 小型微型计算机系统, 2003, 24(12): 2100-2103.

[7] 负载均衡技术全攻略(上)[EB/OL]. http://cisco.chinaitlab.com/case/12415.html, 2003-5.

[8] 负载均衡技术全攻略(上)[EB/OL]. http://cisco.chinaitlab.com/case/12416.html, 2003-5.