基于轮转周期的动态反馈负载均衡算法

许少华,夏智伟
(东北石油大学 计算机与信息技术学院,黑龙江 大庆 163318)

摘 要:网络管理系统中管理端逐步采用分布式集群构架,通过负载均衡算法调度客户端请求,并将客户端请求分配给多个事务节点进行并行处理。为进一步提高集群系统服务的性能,文中在研究以往负载均衡算法的基础上,提出了一种基于轮转周期的动态反馈负载均衡算法。该算法设计了一种基于剩余资源动态权值的节点剩余负载能力计算方法的动态反馈机制;并在动态反馈负载均衡算法的一个采样周期内引入轮转周期对客户端请求均衡分配。通过实验比较分析,该算法能获得更好的负载均衡效果。

关键词:负载均衡;加权轮转;轮转周期;动态反馈
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2013)06-0063-04
doi:10.3969/j.issn.1673-629X.2013.06.016

A Dynamic Feedback Load Balancing Algorithm Based on Round Cycle

XU Shao-hua, XIA Zhi-wei
(College of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: The server of NMS, gradually adopting distributed cluster architecture, uses load balancing algorithm to schedule client requests and assigns client requests to more than one transaction node to parallel processing. On the purpose of improving the performance of cluster system, based on the research of load balancing algorithm on previous, propose a dynamic feedback load balancing algorithm based on the round cycle. The algorithm designs a dynamic feedback mechanism which is the remaining load capacity calculation method based on the remaining resources dynamic weights; introduce weight round cycle within a sampling period of dynamic feedback load balancing algorithm to distribute client requests. Through the experiment comparison and analysis, the algorithm can achieve a good load-balancing effect.

Key words: load balancing; weight round robin; round cycle; dynamic feedback

0 引 言

网络管理系统是指通过对网络资源和网络行为进行有效的测量、分析和控制,最大限度地增加网络设备的利用率,改善网络性能,保证服务质量和网络安全性的一组软件。当前网络管理系统广泛采用管理—代理构架。随着计算机技术的发展,在当前网络管理系统中,代理端运行于客户机中用于采集客户端的运行状态信息并将信息转化为可通用的 XML 数据,管理端运行于服务器中用于获取代理端的采集信息、还原 XML 数据、分析过滤数据并对最终结果入库。随着计算机网络规模的扩大,网络管理系统中代理数量快速增长,传统的集中式管理端已无法及时处理日益增长的客户端请求,管理端采用分布式集群构架成为大型网络管理系统的发展趋势。基于分布式集群的网络管理系统通过负载均衡算法调度客户端数据请求,将客户端请求均衡合理分配给多个事务节点进行并行处理,从而减少客户端等待响应时间,提高系统的吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

1 系统模型

分布式集群网络管理系统的逻辑架构如图 1 所

收稿日期:2012-08-31 **修回日期**:2012-12-02 **网络出版时间**:2013-05-14
基金项目:黑龙江省研究生创新科研项目(YJSCX2011-120HLJ)
作者简介:许少华(1962-),男,教授,博士生导师,主要研究领域为神经网络、智能信息处理;夏智伟(1988-),男,硕士研究生,研究方向为人工智能与数据挖掘。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20130514.1711.016.html

示。调度节点接收到客户端(Agent)的数据处理请求后,将任务分配给相应的事务节点;然后客户端与事务节点建立连接,并将数据传送给事务节点;最后由事务节点对数据进行还原、分析、入库处理等。

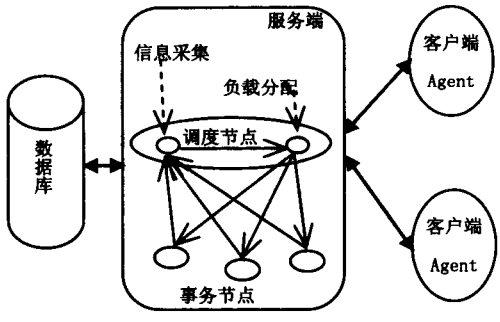

图1 分布式网络管理系统

2 负载均衡算法概述

负载均衡算法根据是否考虑服务器动态性能可以分为静态负载均衡算法和动态负载均衡算法。常用的静态负载均衡算法包括:轮转法(Round Robin)、加权轮转法(Weight Round Robin)、比率法(Ration)、优先权法(Priority)。静态负载均衡算法有简单性、高效性和运行可靠性等优点,但是由于事务请求的差异性,静态负载均衡算法容易失效,从而导致服务器负载失衡。常用的动态负载均衡算法[1~5]包括:最少的连接法(Least Connection)、最快响应法(Fastest)、预测模式[6](Predictive)、动态反馈负载均衡算法等。其中,最小连接法考虑到事务节点的实时负载状况,调度节点将新来的事务请求总是分给当前已有连接数量最少的事务节点。但当事务节点综合性能差别较大时,该算法也会造成事务节点间负载的不均衡。最快响应法是指调度节点对各事务节点发出一个请求(例如 Ping),然后根据各事务节点对请求的最快响应时间来决定哪个事务节点来响应客户端的事务请求。此种均衡算法能较好地反映事务节点的当前运行状态,但最快响应时间仅仅指的是调度节点与事务节点间的最快响应时间,而不是客户端与事务节点间的最快响应时间。预测模式通常采用 BP 网络预测方法,该方法的唯一缺点是耗时[7,8]。动态均衡算法通过实时分析事务节点间的负载状况和任务量,选取合适的节点提供服务,动态分配任务,相对静态均衡技术更好地符合了分布式服务的要求,但它的缺点是,在平均分配任务方面不如静态负载均衡,并且未考虑到采集实时信息消耗的时间以及采集任务对事务节点的负载影响。综合静态负载均衡和动态负载均衡的优缺点,文中提出一种基于轮转周期的动态反馈负载均衡算法。该算法包括两部分:基于剩余资源动态权值的节点剩余负载能力计算方法和基于轮转周期的负载均衡算法。

3 算法描述

3.1 事务节点剩余负载能力的计算

节点的剩余负载能力反映节点当前的负载,节点剩余负载能力越小,当前节点的负载越大。常用的节点剩余负载能力[9]的计算方法描述如下:假设影响节点剩余负载能力的资源集合 $RS=\{RS_1,RS_2,\cdots,RS_n\}$,在一个采样周期 T 内各种剩余资源的集合为 $RS'=\{RS'_1,RS'_2,\cdots,RS'_n\}$($RS'_i$ 表示一种资源的剩余值),根据不同资源对节点剩余负载能力的影响不同分配不同的资源权值 W_i,计算出该节点的剩余负载能力:

$$F'=RS'_1*W_1+RS'_2*W_2+\cdots+RS'_n*W_n$$

该方法易于理解,计算方便,但是只是静态地反映出各种资源对剩余负载能力的不同影响。在实际的运行情况中,当资源的剩余值越小时,该资源对剩余负载能力的影响越大,也就是说资源的剩余值对剩余负载能力的影响是动态变化的。因此综合考虑方法实现的难易以及剩余资源的动态影响,文中设计出一种基于剩余资源动态权值的节点剩余负载能力计算方法:

$$F'=RS'_1*W'_1+RS'_2*W'_2+\cdots+RS'_n*W'_n$$

其中 W'_i 表示剩余资源动态权值,RS'_i 表示资源的剩余值,$W'_i=\dfrac{\dfrac{C_i}{RS'_i}}{\sum_{i=0}^{n}\dfrac{C_i}{RS'_i}}$($C_i$ 为常数)。

通过上述方法,首先通过调度节点获取事务节点的各资源剩余值,然后根据该事务节点的所有资源剩余值计算出各个资源的动态权值,最后根据加权求和公式计算出事务节点的剩余负载能力。

3.2 基于轮转周期的负载均衡算法

负载均衡算法是分布式系统中调度节点的核心算法,它的主要任务是合理和透明地在事务节点之间重新分配系统负载,以达到系统整体的综合性能最优[10,11]。一般加权轮转负载分配算法的基本思想是:在一个采样周期 T 内,调度节点计算出各个事务节点的剩余负载能力,并通过加权法计算出各个事务节点的负载权重;当新的事务请求到达时,调度节点根据一定的权重轮转算法将事务分配给指定的事务节点[12]。为方便算法描述,假设系统中有三个事务节点。

一般加权轮转算法伪代码如下:

```
//首先为轮转算法分配一定范围
int s=50;
//a,b,c 分别表示 A,B,C 三个节点的负载权重
double a,b,c;
while(True){
//当没有事务到达时,线程阻塞
Task t=ReceiveTask();
//产生一个小于 s 的随机非负正整数 r=CreateRandomInt
```

```
(s);
    //事务转移
    if(s<(int)(s*a)) Transmit(t,A);
    else if(s<(int)(s*(a+b))) Transmit(t,B);
    else Transmit(t,C);
    }
```

加权轮转算法一般在静态负载均衡算法中使用。在事务处理分配中,高权重事务节点会比低权重值的事务节点分配到更多的事务。在静态负载均衡算法中,加权轮转算法在负载较轻的系统中表现较好,但如果运用在负载较重的系统上或偶尔计算量较大的事务被分配到同一台服务器上,系统会出现负载不平衡的现象。文中将加权轮转算法引入动态负载均衡算法中,解决在静态负载均衡算法中出现的负载不均衡现象。但在实际使用中,由于事务节点负载权重的动态变化导致事务分配算法失效,也就是短周期内的随机数存在不均匀分布。例如上述一般加权轮转算法中,当 $r < 10$ 时事务被转移到事务节点 A;而在下一个周期 T 内,当 $r < 20$ 时事务被转移到事务节点 A。如果前一个周期内随机数大部分小于 10,后一个周期内随机数大部分小于 20,这样由于短周期 T,导致两个周期内大部分事务都被分配到事务节点 A。由此,文中设计出一种基于轮转周期的加权轮转算法,该算法很好地克服了短周期内的随机数不存在均匀分布所引起的问题。该改进加权轮转算法核心思想是:在一个采样周期 T 内,用事务处理数来规定一个轮转周期 n(如下伪代码描述中轮转周期等于 50),该轮转周期在正常负荷下所用时间 t 应该不大于采样周期 T;在一个轮转周期内,当事务节点接受一个事务后,用数组记忆法将事务节点的权重减少 $1/n$,在一个轮转周期内事务节点按照负载权重均匀分配,从而保证在一个采样周期 T 内,系统的负载均衡分配。

基于轮转周期的加权轮转算法伪代码如下:

```
//a,b,c 分别表示 A,B,C 三个节点的负载权重
double a,b,c;
//首先为轮转算法分配数值空间,当数组元素等于 1、2、3 时分别对应事务节点 A、B、C
int status[50]=InitStatus(status,a,b,c);
//事务计数器 i
int i=0;
while(True){
//在规定的轮转周期后对 temp 重新赋值
if(i==0)ArrayCopy(temp,status);
//当没有事务到达时,线程阻塞
Taskt=ReceiveTask();
//产生一个小于 50-i 的随机非负整数
r=CreateRandomInt(50-i);
//事务转移
if(status[r]==1)Transmit(t,A);
else if(status[r]==2)Transmit(t,B);
else Transmit(t,C);
//将 status 中 r 和 49-i 位置中元素对换,在下次事务请求中 r 元素被排除,也就是使其对应的事务节点的权重减少 1/50;
switch(status,r,49-i);
i=(++i)%50;
}
```

3.3 系统仿真与算法性能分析

系统仿真使用了 3 台 DELL OPTIPLEX 990 构成分布式网络管理系统集群,事务节点配置为 Core i5-2400 3.10GB*4,8GB 内存,1000M 网络适配器。在系统运行期间,调度节点采集事务节点的三项剩余资源:CPU、内存、网络带宽,其中 C_1、C_2、C_3 分别为 3、2、1。同时运行 200 个 Agent 对事务节点进行数据操作请求,每个 Agent 随机请求的数据量的集合为{100k,200k,400k,800k},系统采样周期 T 为 10s,轮转周期为 50。文中以节点剩余负载能力是否均衡来衡量算法性能,系统分别采用静态加权轮转负载均衡算法、基于一般加权轮转的动态反馈负载均衡算法和文中提出的基于轮转周期的动态反馈负载均衡算法进行测试,测试结果分别如图 2、图 3 和图 4 所示。从图中可以看出无论采用何种算法,当系统运行 40s 左右都进入基本稳定状态;静态加权轮转算法由于事务请求的差异性,在负载均衡失衡时很难调整;基于一般加权轮转的动态反馈负载均衡算法在负载均衡失衡时系统很快进行调整,不过由于短周期内的随机数不存在均匀分布,导致事务节点的负载交替震荡;文中提出的基于轮转周期的动态反馈负载均衡算法,3 个事务节点的负载虽然处在交替变化中,但并没有明显的负载不平衡现象,3 个事务节点比较均匀地分担客户端的事务请求,没有出现有哪个事务节点的负载过度高于其他事务节点的情况。很明显文中提出的基于轮转周期的加权轮转算法在系统长期运行期间更加稳定,负载更加均衡。

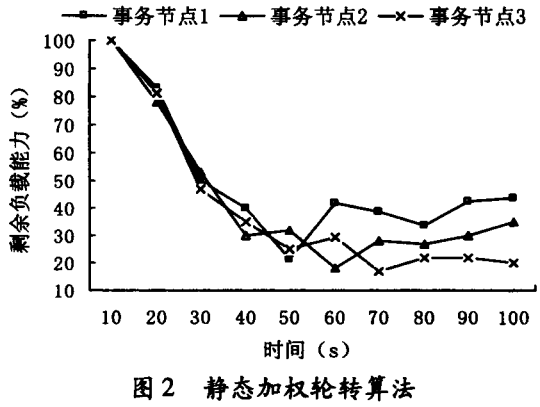

图 2　静态加权轮转算法

4 结束语

文中在静态加权轮转算法的基础上提出轮转周

期,同时结合动态反馈负载均衡算法提出了基于轮转周期的动态反馈负载均衡算法。该算法由两部分组成:基于剩余资源动态权值的节点剩余负载能力计算方法和基于轮转周期的加权轮转算法。该算法结合了静态加权轮转算法的简单性、高效性和动态反馈机制的实时性等优点。经测试,该算法获得了很好的负载均衡效果。

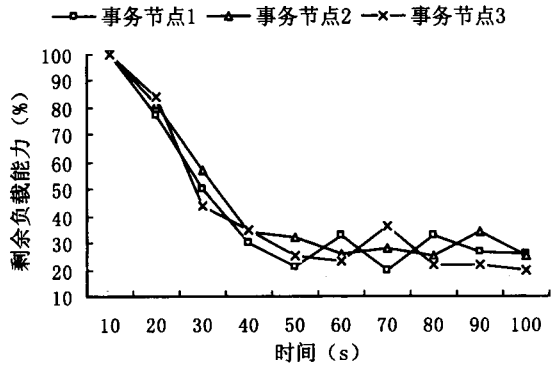

图3 基于一般加权轮转的动态反馈负载均衡算法

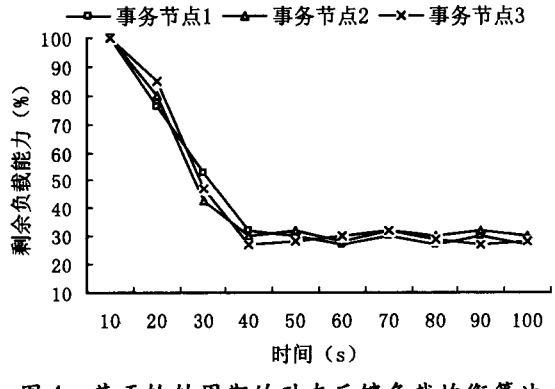

图4 基于轮转周期的动态反馈负载均衡算法

参考文献:

[1] 陈 伟,张玉芳,熊忠阳.动态反馈的异构集群负载均衡算法的实现[J].重庆大学学报,2010,33(2):73-78.

[2] 凌 云,周华锋.面向异构集群系统的动态负载均衡技术研究[J].计算机工程与设计,2008,29(12):3068-3070.

[3] 陈 超,赵跃龙,王文丰,等.基于反馈的改进动态负载均衡策略[J].计算机工程,2010,36(14):34-36.

[4] 田绍亮,左 明,吴绍伟.一种改进的基于动态反馈的负载均衡算法[J].计算机工程与设计,2007(3):572-573.

[5] 周松泉.一种改进的集群动态负载均衡算法[J].计算机与现代化,2012(1):135-139.

[6] 杨 伟,朱巧明,李培峰,等.基于时间序列的服务器负载预测[J].计算机工程,2006,32(19):143-145.

[7] 余燕芳,陆 军.基于改进遗传算法的服务器端负载均衡算法[J].微电子学与计算机,2007,24(7):146-148.

[8] 薛正华,董小社,李炳毅,等.基于BP神经网络的集群负载预测器[J].华中科技大学学报:自然科学版,2007,35(z2):164-167.

[9] 段淮川,胡 平.基于剩余负载率的动态负载均衡研究[J].微电子学与计算机,2010(2):141-144.

[10] Karger D R,Ruhl M. Simple Efficient Load-balancing Algorithms for Peer-to-Peer Systems[J]. Theory of Computing Systems,2006,39(6):787-804.

[11] Sharifian S,Motamedi S A,Akbari M K. A content-based load balancing algorithm with admission control for cluster web servers[J]. Future Generation Computer Systems,2008,24(8):775-787.

[12] Wu Di,Tian Ye,Ng K W. Resilient and Efficient Load Balancing in Distributed Hash Tables[J]. Journal of Network and Computer Applications,2009,32(1):45-60.

(上接第62页)

端进行连接操作。另外为了能够在MapReduce框架中使用这一分区策略,因此对原MapReduce框架中的分区函数接口进行了修改。通过实验分析发现,改进模式的效率明显得到提高,达到了预期的目的。

参考文献:

[1] 钟伟彬,周梁月,潘军彪,等.云计算终端的现状和发展趋势[J].电信科学,2010,26(3):22-26.

[2] 李林娟,张 敏.云计算环境下关联规则挖掘算法的研究[J].计算机技术与发展,2011,21(2):43-46.

[3] 罗军舟,金嘉晖,宋爱波,等.云计算体系架构与关键技术[J].通信学报,2011,32(7):3-21.

[4] 江务学,张 璟,王志明,等.MapReduce并行编程架构模型研究[J].微电子学与计算,2010,27(6):168-170.

[5] 郑启龙,房 明,汪 胜,等.基于MapReduce模型的并行科学计算[J].微电子学与计算机,2009,26(8):13-17.

[6] Jiang Dawei,Tung A K H,Chen Gang. MAP-JOIN-REDUCE:Toward Scalable and Efficient Data Analysis on Large Clusters[J]. IEEE Transactions on Knowledge and Data Engineering,2011,23(9):1299-1311.

[7] Lamel R. Google's MapReduce Programming Model-Revisited[J]. Science of Computer Programming,2008,7(1):208-237.

[8] Ghemawat S,Gobioff H,Leung Shun-Tak. The Google file system[J]. ACM SIGOPS Operating Systems Review,2003,37(5):29-43.

[9] Chang F,Dean J,Ghemawat S,et al. A distributed storage system for structed data[J]. ACM Transactions on Computer System,2008,26(2):1-26.

[10] 史佩昌,王怀民.面向云计算的网络化平台研究与实现[J].计算机工程与科学,2009,31(11):12-13.

[11] 朱 珠.基于Hadoop的海量数据处理模型研究和应用[D].北京:北京邮电大学,2008.

[12] White T. Hadoop权威指南[M].曾大聃,周傲英,译.北京:清华大学出版社,2010.

基于轮转周期的动态反馈负载均衡算法

作者： 许少华， 夏智伟， XU Shao-hua， XIA Zhi-wei
作者单位： 东北石油大学计算机与信息技术学院,黑龙江大庆,163318
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2013,23(6)