

# 基于内容分类的集群负载均衡算法<sup>①</sup>

郑 祺, 周广平

(浙江科技学院 信息与电子工程学院, 杭州 310023)

**摘 要:** 集群技术为 Web 服务带来了新的解决方案, 其核心思想是负载均衡策略。在分析已有方法的基础上, 提出了一种基于内容分类的集群负载均衡算法。该算法通过对用户请求分类后均匀地分配给各节点服务器, 以使每个节点服务器得到的各类请求量大致相同。同时算法引入了一个反馈环节, 通过负载权值的等效变换和进入临界状态后的动态权值调整来防止节点负载倾斜。实验证明该算法能有效提高集群系统的整体性能。

**关键词:** 集群; 负载均衡; 内容分类; 动态反馈; 权值

## Content-Classification Load Balancing Algorithm in Cluster

ZHENG Qi, ZHOU Guang-Ping

(School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China)

**Abstract:** The cluster technology brings a new solution for the WEB service, the key point is load balancing strategy. Based on the existing algorithms, a content-classification load balancing algorithm is proposed. By using content classification, user requests could evenly distribute to the nodes in the cluster, so that each node would get roughly the same amount of various types of requests. To prevent the load skew, a feedback mechanism was introduced, furthermore, the weight of each node interval was adjusted by using an equivalent load-alternant and would multiplicative decrease in critical area. The test shows this algorithm could improve the performance of the cluster system.

**Keywords:** cluster; load balancing; content-classification; dynamic feedback; weight

## 1 引言

近年来, 随着 Internet 的飞速发展以及电子商务、多媒体技术等广泛应用, 网络信息流量正呈指数增长, 这对 Web 服务器的处理能力提出了严峻的挑战, 仅仅依靠提高单个服务器的性能已经无法满足 Web 服务发展的需要。集群服务器以其高可扩展性、高可靠性和高性价比, 为 Web 服务器系统带来了新的解决方案<sup>[1]</sup>。

Web 集群系统一般由局域网内通过高速网络连接的一组通用服务器(这些服务器可以是同构的, 也可以是异构的, 也称为节点服务器)构成, 这个系统对外相当于一台高性能的服务器。Web 集群一般由一台特殊的服务器(请求分配服务器 dispatcher, 也可称控制器)

接受请求, 并按照某种策略动态地分配到各节点服务器上进行处理。集群的负载均衡策略是提高集群整体性能的关键, 其目的是根据处理机的性能来分配与其相称的任务, 以最小化应用程序的执行时间, 最大限度的利用各节点的处理能力, 从而提高集群系统的整体性能。

## 2 相关工作

随着集群的大规模应用, 目前对于负载均衡算法的研究较多。经常使用的负载均衡算法包括两大类: 一类是静态负载均衡算法, 常用的如轮转法(Round-Robin)、最小连接数法(Least-Connection)等, 这些算法都没有考虑到用户请求任务间的差异和集群内各节点

① 收稿时间:2010-08-20;收到修改稿时间:2010-10-17

服务器的性能差异及实际负载情况,无法充分利用各节点服务器的处理能力,并不能有效地解决集群内各服务器间的负载均衡问题。另一类是动态负载均衡算法,可以根据各节点服务器的实际情况动态分发用户请求,近几年的研究工作主要针对动态负载均衡算法的研究,基于内容的负载均衡算法是研究的热点之一。典型的算法如 LARD (Locality Aware Request Distributes)<sup>[2]</sup>、CAP(Client Aware Policy)<sup>[3]</sup>、WARD (Workload Aware Request Distribution)<sup>[4]</sup>、IQRD (Intelligent Queue—based Request Dispatcher)<sup>[5]</sup>等。LARD 将集群中的节点服务器按服务进行分类,每台节点服务器只对某些类型的请求进行响应,只有当负载出现明显不均衡时,才会对请求进行再分配;CAP 算法是一种基于客户端内容的负载均衡算法。其核心思想是通过控制器识别客户端请求的类型,然后将同类请求均匀地分配给各节点服务器,以使每个节点服务器得到的各类请求量大致相同,从而达到负载均衡;WARD 依据访问模式来划分请求,按服务频率和文件大小分类,目标是最大化集群总的内存提供服务的请求数,最小化转发的次数,减少开销,使用 TCPHandof 转发技术;IQRD 将静态内容按请求文件的大小分类,将动态内容按平均执行时间分类,使用 WRR 策略调度,动态的将每个节点服务器剩余的处理能力反馈给负载均衡模块以调节相应的权值<sup>[6]</sup>。

在深入研究各种基于内容的集群负载均衡算法的基础上,本文的后续部分将提出一种基于内容分类的动态反馈负载均衡算法 CAWF(Client Aware With Feedback),该算法采用了一种相对简单的方法来进行权值的等效变换并对进入临界区的节点采取加速递减权值的策略来合理分配用户请求,以平衡节点的负载,能够较好解决不同负载状况下的负载均衡问题。

### 3 CAWF算法

算法设计思想:控制器在收到用户请求后,首先根据其 URL 等应用层信息进行分类,然后发往相应队列等候分发,为了满足异构集群的需求,队列内的请求分发采用加权轮转法(WRR),以使每个节点服务器得到的各类请求数大致与其权重相似,为了尽可能避免节点负载的不均衡情况,控制器会周期性地监测各节点服务器的真实负载状况,通过采

集各节点服务器的响应时间,控制器可等效计算出各个节点服务器当前的负载权值,然后根据实际负载权值调整分发比例。

算法框图如图 1,其中:Server(i)代表节点服务器 i;class(i)表示用户请求的分类类别,Q(i)为对应的队列;Ti 为节点服务器 i 的响应时间;Sch 为根据负载权值计算出的分发比例。

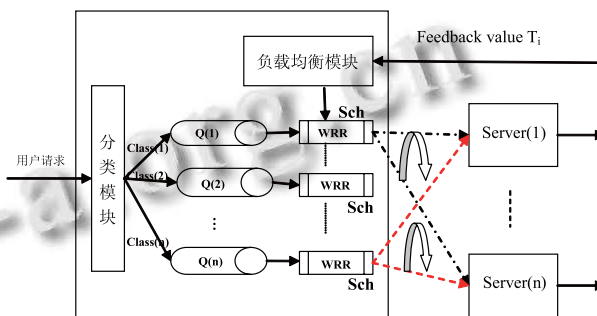


图 1 CAWF 算法框图

#### 3.1 请求分类

由于 WEB 网站中既有大量静态页面,又大量使用了动态嵌入对象技术和数据库操作任务,这使得不同请求任务所消耗的系统资源差异可能很大,为了达到负载均衡的目的,需要先将用户请求进行分类。借鉴 CAP<sup>[3]</sup>的做法将 Web 服务分成 4 种类型:

- 1) 发布型: 主要提供静态信息(如 HTML 页面及嵌入的对象),以及轻微动态网络发布服务
- 2) 事务型: 结果来自动态数据库查询,且查询条件一般由用户通过 HTML 页面动态提供。它通常需要进行密集的磁盘访问,因此也称磁盘密集型服务。
- 3) 电子商务型: 提供静态、动态和安全信息传输,主要面向电子商务应用。出于安全原因,一些动态生成的数据需要通过安全途径进行传送,大部分情况下会使用 SSL 协议。加解密操作会消耗大量的 CPU 资源,而进行数据库访问时又需要密集磁盘访问,因此这类服务也称为磁盘 / CPU 密集型服务。
- 4) 多媒体型: 提供音频和视频流媒体服务。这类服务一般通过特殊的服务器和网络协议连接。

#### 3.2 权值计算

相关研究表明,在基于 Intel 的 UNIX/LINUX 服务器环境中,各种硬件资源对应用程序运行的影响比例如下<sup>[7]</sup>:

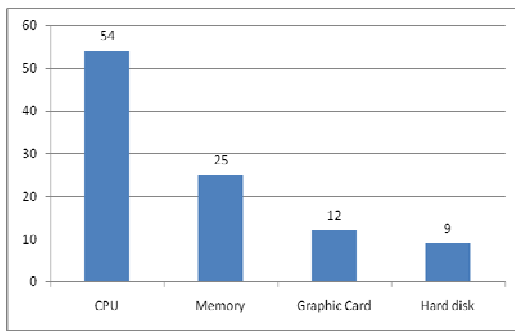


图 2 硬件资源对应用程序的影响比例

根据上述研究结果，在计算节点的初始权值时主要考虑以下参数：CPU 处理速度、内存容量、系统 I/O 速率和网络带宽，各节点初始权值的计算公式如下：

$$Load(N_i) = R_1 K_i \frac{L_{CPU}(N_i)}{BASE_{CPU}} + R_2 \frac{L_M(N_i)}{BASE_M} + R_3 \frac{L_{IO}(N_i)}{BASE_{IO}} + R_4 \frac{L_{NET}(N_i)}{BASE_{NET}} \quad (1)$$

其中  $L_f(N_i)$  表示节点  $N_i$  某一参数的当前值，公式(1)中依次表示为：CPU 处理速度、内存容量、系统 I/O 速率和节点的网络接口速率， $K_i$  表示节点  $i$  的处理器数量。 $BASE_f$  表示某一参数的基准值，依次为基准 CPU 处理速度、基准内存容量、基准 I/O 速率和基准网络接口速率，这些基准值可根据各节点的实际情况统计确定。 $R_i$  是为每个参数设定的一个可调系数，可用以区分各种硬件资源的重要程度，其中  $\sum R_i = 1$ 。一组典型设置可采用：(0.4、0.2、0.1、0.3)。

考虑到各节点权值的更新需要周期性的对节点进行参数采集和计算，这需要消耗一定的系统及网络资源且存在计算系数选择困难的问题，因此拟采用一种相对简单的方法来进行权值的等效变换，以降低系统的资源消耗，达到提高集群效率的目标。

通过对单台服务器的响应状况(图 3)和响应时间(图 4)进行观察<sup>[8]</sup>：

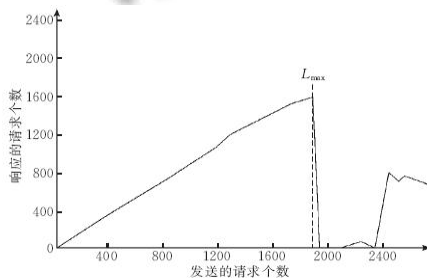


图 3 发送的请求个数与响应的请求个数关系图

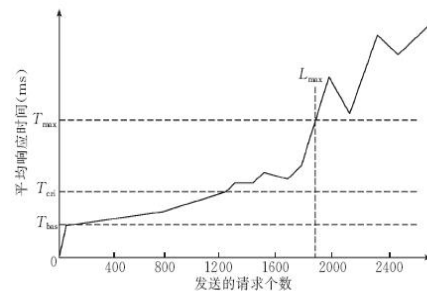


图 4 发送的请求个数与平均响应时间关系图

可以发现当发送的请求量达到一定值  $L_{max}$  时，服务器响应的请求数量会突然下降为零并持续一段时间，这是系统软件为防止死机而采取的一种措施，也称“假死”现象，它会对服务器的性能产生严重的影响，因此需要尽量避免这种情况的发生。一旦服务器出现“假死”现象时，其响应时间会急剧增加，会超过  $T_{max}$ ，这种情况下可认为此时服务器的可用资源为零。从图 3 还可看出，当一个最基本的请求任务加载到空载服务器时，服务器的响应时间为  $T_{bas}$ ，如果忽略该请求的影响，则可认为  $T_{bas}$  就是服务器维持运转的基本负载<sup>[8]</sup>。不妨将  $(T_{max} - T_{bas})$  定义为服务器可对外提供的服务能力，设  $T_{now}$  为服务器当前的响应时间，则可用  $(T_{max} - T_{now})$  来表示该服务器当前的剩余服务能力。由此得出节点当前权值的等效变换如下：

$$W'(N_i) = Load(N_i) \times \frac{T_{max}^i - T_{now}^i}{T_{max}^i - T_{bas}^i} \begin{cases} T_{now}^i \leq T_{bas}^i, T_{now}^i = T_{bas}^i \\ T_{now}^i \geq T_{max}^i, T_{now}^i = T_{max}^i \end{cases} \quad (2)$$

从图 3 可观察到当请求数达到某一值时，其对应的响应时间的变化会出现抖动现象，表明系统即将进入饱和状态，可将此区域定义为系统进入饱和状态前的临界区，从响应时间上看为  $T^{cri} \sim T^{max}$  区间<sup>[8]</sup>。为抑制服务器进入饱和状态，可通过适当降低已进入临界区的节点的权值来降低其下阶段的工作负载。调整公式如下：

$$W(N_i) = W'(N_i) \times K, K = \begin{cases} 1, T_{now}^i \leq T_{cri}^i \\ \alpha(1 - \frac{T_{now}^i}{T_{max}^i}), T_{now}^i > T_{cri}^i \end{cases} \quad (3)$$

$K$  为调节系数，仅当节点进入临界区后才产生作用，节点负载越重则  $K$  值越小，以使临界深度大的节点加快速度，避免进入饱和状态。 $\alpha$  值可用来调节递

减速度，通常取 2。

在实际应用中，管理员还可根据实际情况给每个节点设定一个阈值，当权值小于阈值时，可为该节点已进入“假死”状态，控制器将不会再向该节点分发任何请求，并向管理员发出节点过载警告<sup>[9]</sup>。

### 3.3 请求分发

控制器根据各节点的响应时间可得到各节点的当前权值，然后采用加权轮转法(WRR)对各个队列中的用户请求进行调度，即根据节点权值的大小按轮转方式将用户请求分发到各节点上，权值越高的节点得到的用户请求就越多，每个节点所分配到的用户请求数按其权值占节点权值总和的比例来确定。

## 4 仿真试验

衡量集群负载均衡算法有两个主要评价指标：平均响应时间和平均吞吐量。我们使用了 NAT 方式构造了一个 LVS 集群<sup>[10]</sup>，内部网络连接采用 100Mb/s 的交换机，集群由 1 台控制器和 4 台节点服务器构成，硬件配置见表 1，另使用 4 台 PC 作为客户端对集群进行模拟加压测试，使用的工具是 Linux 下的 Httpperf，请求事件流为符合负指数分布的泊松事件流，后台负载模型为电子商务型站点，包含 30%的静态请求、30%的简单动态请求和 40%的其它各种混合请求(主要以加解密计算和数据库查询为主)，系统加解密采用 RSA 和 Triple-DES 算法，校验使用 MD5 算法。选择的对比测试算法为加权轮转法(WRR)和 CAP。

表 1 设备硬件参数

设备类型	CPU	内存	硬盘
控制器	2×Xeon 2.8G	2G	73G
节点 1	2×Xeon 2.8G	2G	73G
节点 2	Xeon 2.4G	1G	80G
节点 3	P4 3G	1G	80G
节点 4	P4 3G	1G	80G

仿真对比测试结果如图 5、图 6 所示。

从仿真测试结果可以看出，当负载很轻时，CAWF 算法的性能与 CAP 算法的性能基本相似，但略低于 WRR，这主要是由于前两种算法需要先对用户请求进行解析分类后再转发，系统开销相对较大，而 WRR 算法不需要了解用户请求的具体内容，可直接进行转发，系统开销较小。随着负载的逐步增大，CAWF 算法的性能开始优于 CAP 算法，并明显优于 WRR 算法。

进入重载状态后，CAWF 算法的性能已明显优于 CAP 算法，而 WRR 算法性能最差，具体分析其主要原因在于 WRR 算法未考虑用户请求间的差异以及节点的真实负载状况，很容易出现节点负载的不均衡，从而导致系统整体性能的急剧下降；CAP 算法只考虑了用户请求间的差异，但忽略了节点的真实负载状况，也会导致节点负载的不均衡；CAWF 算法则充分考虑到这两种差异，仿真试验表明，算法可较好的解决集群内各节点服务器间的负载均衡问题，从而有效提高集群的整体性能。

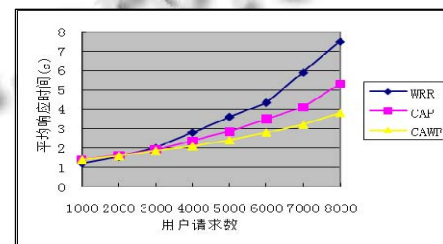


图 5 平均响应时间对比

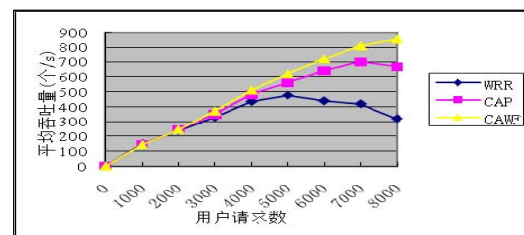


图 6 平均吞吐量对比

## 5 结语

集群负载均衡策略是提高集群整体性能的关键，本文在分析了几种现有的基于内容的集群算法后，提出了一种改进算法 CAWF。该算法充分考虑了用户请求间的差异以及集群内部各节点服务器间的性能差异，通过控制器识别客户端请求的类型，然后再将同类请求均匀地分配给各节点服务器，以使每个节点服务器得到的各类请求量大致相同，同时引入了反馈环节，周期性的通过负载权值的等效变换和进入临界状态后的加速递减权值来合理分配用户请求，以平衡节点的负载，充分利用各节点的资源。仿真测试表明该算法具有良好的负载均衡能力。

### 参考文献

1 Chen LC, Choi HA. Approximation Algorithms for Data

(下转第 74 页)



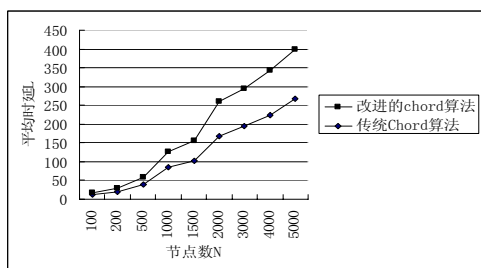


图 6 第一组实验结果对比图

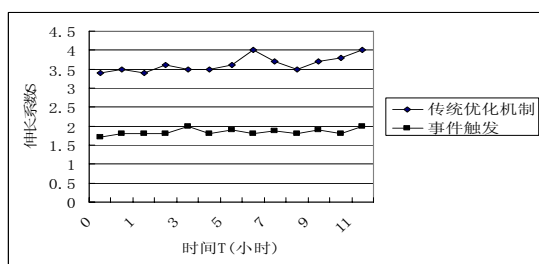


图 7 第二组实验结果对比图

## 6 总结

本文提出了一种构建移动环境下 P2P 网络拓扑结构的方法，这种方法与协议无关，适用于各种结构化 P2P 网络，能有效的构建与物理层网络相匹配的覆盖层网络拓扑。此外，本文还提出了处理移动 P2P 网络中节点移动的方法，显著地提高了物理网络拓扑与覆

盖网拓扑匹配程度。实验表明，通过本方法构建的移动 P2P 网络是一个高效的覆盖网络，网络搜索性能优越，拓扑匹配度高。

本文提出的方法未考虑移动设备的异构性，下一步我们将就降低网络开销及负载均衡等问题展开研究。此外，移动 P2P 网络还存在着诸多的研究方向如资源发现策略、数据分发及安全及隐私问题，笔者也将在以后的工作对上述问题展开研究。

## 参考文献

- 1 斯太门兹等.王玲芳,陈焱译.P2P 系统及其应用.北京:机械工业出版社,2008.
- 2 欧中洪,宋美娜,战晓苏,宋俊德.移动对等网络关键技术.软件学报,2008,2(19):1-4,8-9.
- 3 程久军,李玉宏,程时端,马建.移动 P2P 系统体系结构与关键技术的研究.北京邮电大学学报,2006,4(29):1-2.
- 4 邱彤庆,陈贵海.一种令 P2P 覆盖网络拓扑相关的通用方法.软件学报,2007,2(18):1-3.
- 5 Krcro S, Cleary D, Parker D. P2P Mobile Sensor Networks. Proc. of the 38th Annual Hawaii Int'l Conf. on System Sciences, 2005: 1-9.
- 6 Szer H, Tekkalmaz M, Krpeolu. A Peer-to-Peer File Sharing System for Wireless Ad-hoc Networks. Proc. of the 3rd Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2004), 2004: 1-9.
- 5 Sharifian S, Motamedi SA, Akbari MK. A content-based load balancing algorithm with admission control for cluster web servers. Future Generation Computer Systems, 2008,24(8): 775-787.
- 6 谢红薇,谢显宇.基于内容的网络集群负载均衡算法模型.计算机应用与软件,2010,27(1):131-133.
- 7 Youn C. Performance Improvement of Cluster System by Server Status Information. Proc. of the Fourth Annual ACIS International Conference on Computer and Information Science. 2005: 282-287.
- 8 郭成城,晏蒲柳.一种异构 Web 服务器集群动态负载均衡算法.计算机学报,2005,28(2):179-184.
- 9 郑祺.一种基于 WEB 集群的负载均衡算法研究.计算机系统应用,2009,18(7):76-79.
- 10 马双良,张英敏,宋丽君.基于 LVS 和计算任务的实时集群负载均衡方法.计算机工程与设计,2007,28(20):4934-4937.

(上接第 50 页)

- Distribution with Load Balance of Web Servers. Proc. of IEEE International Conference on Cluster Computing, 2001:274-281.
- 2 Vivek S, Mohit A, Gaurav Banga, et al. Locality-aware request distribution in cluster based network servers. ACM SIGPLAN Notices, 1998,33(11):205-216.
- 3 Casalicchio E, Colajanni M. A Client Aware Dispatching Algorithm for Web Clusters Providing Multiple Services. Proc. of the 10th International World Wide Web Conference. Hong Kong, 2001:535-544.
- 4 Cherkasova L, Karlsson M. Scalable Web Server Cluster Designwith Workload-aware Request Distribution Strategy WARD. Proc. of the IEEE International Workshop on Advanced Issues in E-Commerce and Web-Based Information Systems. San Jose, CA, 2001: 212-221.