

# Web 机群服务质量控制自我管理模型

刘涛<sup>1,2</sup>, 曾国荪<sup>1,2</sup>

(1. 同济大学计算机科学与技术系, 上海 201804; 2. 国家高性能计算机工程技术中心同济分中心, 上海 201804)

**摘要:** 基于服务类别的资源请求和服务等级给出一种资源划分策略, 对各类别的服务请求状态进行建模, 使用反馈控制理论的方法对各类别资源进行动态调整, 与资源调整规则相结合, 以适应访问的突发性和保证服务质量要求, 实现了 Web 机群服务质量的自我管理。实验表明自我管理模型能有效支持区分服务质量和资源效用控制。

**关键词:** 服务质量; 自我管理; 控制理论

## Self-management Model for Control Differentiated QoS in Web Clusters

LIU Tao<sup>1,2</sup>, ZENG Guo-sun<sup>1,2</sup>

(1. Department of Computer Science and Technology, Tongji University, Shanghai 201804;

2. Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai 201804)

**【Abstract】** This paper presents a method to partition the resources for each service class with its resource demand and service level agreement requirement, then models different service requests' states of each class. Feedback control theory, together with resource tuning policy is used to regulate resources of each class in order to adapt to paroxysmal requests. This mechanism guarantees QoS of different classes and achieves self-management in Web clusters. Experimental results indicate that the self-managing model could support differentiated QoS effectively and control resources' utility.

**【Key words】** Quality of Service (QoS); self-management; control theory

Internet 在传播信息的数量和范围上一直呈现出爆炸性增长, 其中绝大部分是 Web 应用的增长, 对 Web 服务器提出了更高的要求。通过 Web 前端提供交易与服务的电子商务形式迅猛发展, 使得 Web 服务器成为支持电子商务的核心设施。电子商务应用要求对其用户和服务进行区分优先级别的处理, 提供基于利益的区分服务, 而对 Web 服务器则意味着为付费的高级用户提供比免费用户更好的服务级别保证。目前, IETF 已针对网络传输中的性能保证提出了区分服务质量(Quality of Service, QoS)体系结构, 但是在 Web 服务器领域却没有相应的机制与策略, 无法为 Web 应用提供区分服务和性能保证。因此, 如何在 Web 服务器中实现控制机制与策略, 为不同类型的用户请求提供性能保证和区分服务, 是 Web 发展迫切需要解决的问题。本文采用控制理论的模型与算法构建出了 Web 机群提供区分服务质量保证的体系结构和控制算法。

### 1 相关工作

目前, 对于 Web 服务器支持区分服务的机制与方法开展了一些研究。提供区分服务最直接的方式是维护优先队列, 根据服务需求的不同进行优先调度, 为高优先级的请求提供比低优先级的更好的服务<sup>[1]</sup>。在服务提供者需要控制服务器资源分配来满足服务质量保证的情况下, 这种严格优先级机制不能提供期望的区分服务。本文提出了一种建立在反馈控制理论基础上的优先级机制, 能够根据各优先级服务请求的不同来调节服务器的资源分配, 避免出现负载不均衡的情况。文献[2]提出了一种需求驱动的区分服务策略(DDSD), 对不同

类别的资源周期性地重新划分, 划分方法归结为求解以服务等级为条件的最优解问题。DDSD 将一个周期作为评估粒度的访问控制机制同样不能适应在周期内访问率波动较大的情况。文献[3]提出了可管理服务区分策略的解析方法, 方法建立在动态优先级队列概念基础之上。使用马尔可夫链对请求的到达和分配处理过程进行刻画, 能够避免基于严格优先级的机制的缺点, 为服务提供者提供必要的资源分配方法来满足服务质量要求和不同的服务期望目标。这种基于队列的方法容易受到所采用预测方法的影响, 易出现资源分配不合理的情况。控制理论的方法实现 Web 服务器区分服务控制是一个比较新颖的思路, 有一定的研究进展。文献[4]提出并实现了以控制理论方法为基础的区分内容缓存服务, 能够在代理缓存中调整不同等级的资源配置, 以保证各等级缓存能满足不同的命中率要求。文章按照比例来刻画各等级的命中率要求, 并采用相对命中可能性来代替绝对命中可能性, 采取基于反馈的启发式缓存资源分配方式来调整每个类别的缓存空间分配, 并讨论了控制增益对于性能的影响。

本文集中于 Web 机群提供区分服务质量, 为所有服务请求提供一个分类, 以满足所有请求不同的服务等级约定; 使

**基金项目:** 国家自然科学基金资助项目(60673157); 教育部科研重点基金资助项目(105071); 上海高校网络技术E-研究院资助项目(2003 01-1)

**作者简介:** 刘涛(1979-), 男, 博士研究生, 主研方向: 自主计算, 网络计算; 曾国荪, 教授、博士、博士生导师

**收稿日期:** 2006-12-21 **E-mail:** liutao.tj@gmail.com

得高优先级的请求得到好的服务，即较短的响应延时和较高的响应率；使得 Web 机群在满足所有请求服务质量的同时整体占有的系统资源最少，进行效用管理。提出基于反馈控制理论的 Web 机群区分服务质量自我管理模型，减少人的干预，避免错误的引入。

## 2 自我管理模型

实现区分服务质量通常是对服务请求按照访问用户或被访问内容进行分类，根据重要性设置服务等级，按照特定的区分服务技术来保证相应等级的服务质量。本文采用的方式是按照请求确定相应的服务类别，由分类器将请求分配到具有不同处理能力的类别去，由控制器根据各类别情况的不同来自动调整各类别的处理能力分配。

### 2.1 模型架构

本文提出了以下支持区分服务质量的Web机群自我管理架构，如图1所示，主要由分发器Dispatcher、Web服务器机群和控制器等组成。

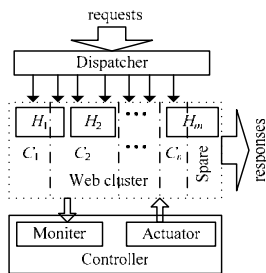


图1 Web机群区分服务质量自我管理架构

Web机群的主要特点是将多个主机逻辑组合，通过发挥所有主机的整体性能来构成一个高性能的虚拟Web服务器。典型Web机群由一个前端控件担任虚拟站点，通过对接收的请求按特定策略进行调度来达到服务需求和负载均衡，前端控件称为Dispatcher<sup>[5]</sup>。由于Web服务中响应数据远多于请求数据，本文采用单向结构作为支持区分服务质量的Web机群分发器。当服务请求到达Web机群后，首先经过分发器Dispatcher，按照不同的用户请求优先级分配到相应的类别中，每个类别代表着处理能力的不同。

将Web机群中的 $m$ 台主机分别表示为 $H_1, H_2, \dots, H_m$ ，将所有的机群资源划分为 $n$ 类，用 $C_1, C_2, \dots, C_n$ 表示，并且按照服务优先级有 $priority(C_1) > priority(C_2) > \dots > priority(C_n)$ ，其各自的服务等级约定，即期望服务质量分别用 $ref_1, ref_2, \dots, ref_n$ 表示。每个类别的资源多少随着服务请求不断变化，资源划分点所在主机也在不断变化，导致一个类别的资源可能不在连续变化的主机上，而是分布于几个不同的主机。在满足所有类别的服务等级约定并预留部分资源后，剩余的机群资源空闲，用Spare表示。

支持区分服务质量自我管理架构中最重要的部分是控制器，由监视器、执行器、比较设定模块、调整控制器等组成。监视器负责对各不同类别的服务请求处理和服务质量保证情况进行实时反馈；比较设定模块根据监视器传入的各类别的状态，比较期望服务质量与时间服务质量，获取偏差，作为控制方程的输入参数，结合针对各类别和各主机资源分配制定的规则进行计算，得出资源调节方案；执行器则根据控制方程的输出对各自类别的资源进行调整，以保证各类别的服务质量符合不同的服务等级约定。比较设定模块接受系统管理员的设定指令，包括对各类别所占资源下限、期望的服务

质量等规则都在此定义。

### 2.2 资源调整算法

本文采用不同处理状态中的服务请求来对资源进行建模。对于不同的服务类别都按如下请求进行处理状态的维护：新增，活跃，空闲，释放。如图2，在控制器不同的采样时间中所有的服务请求在几种状态间相互转化。

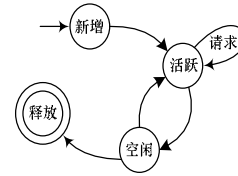


图2 服务请求的状态变化

新增请求表示在采样时间内转发到本类别的服务请求，将对其进行初始化操作，需要的系统资源较多，在下个采样周期内将转为活跃请求。活跃请求表示在初次响应后不断提交新请求的连接，需要的资源较多且稳定，如果规定时间内没有新的请求提交，将转化为空闲请求。空闲请求表示已经响应了的服务请求，并且一段时间内没有新的请求提交，仅保持连接状态的请求；如果规定时间内提交了新的请求就转化为活跃请求；空闲请求最有可能断开或释放，需要的系统资源较少。释放请求表示在采样时间中断开或释放的请求，资源将释放出来归入各类别的资源集中。

对于服务类别 $C_i$ ，其所分配到的资源量用 $R_i$ 表示，并设其资源下限 $LR_i$ ，目的是保证在服务机群满负荷运转时较低优先级的服务请求也能得到一定的服务质量保证。在机群初始化时也将 $C_i$ 的初始资源设为 $LR_i$ ，以实现效用管理。执行器中维护资源调整请求等待队列 $WQ$ ，首先按照服务类别请求的资源调整量 $\Delta r$ 进行区分， $\Delta r < 0$ 的调整请求均处于 $\Delta r > 0$ 的请求前；其次根据提出调整请求的服务类别优先级由高到低的顺序不断调整队列。

执行器资源调整算法，即对任意类别 $C_i$ 的资源调整量 $\Delta r$ 过程如下：

- (1)若 $\Delta r < 0$ ，则该类所调整的资源加入到空闲资源集Spare中，否则转(2)；
- (2)若Spare不为空，则从Spare中调出 $\Delta r$ 到 $C_i$ 中，并计算Spare中剩余资源量，退出，否则转(3)；
- (3)按照 $C_n, C_{n-1}, \dots, C_{i+1}$ 的顺序查找所有资源量大于资源下限的服务类别，放入结果集Usable中；
- (4)若Usable为空，则系统已在满负荷运转，将调整请求推回 $WQ$ 并退出，否则转(5)；
- (5)在Usable中查找 $C_l$ ，使得 $Min(priority(C_l))$ 成立；
- (6)若 $LR_l + \Delta r > R_l$ ，则 $R_l \leftarrow R_l - \Delta r$ ， $R_i \leftarrow R_i + \Delta r$ ，清空Usable并退出，否则转(6)；
- (7) $R_l \leftarrow LR_l$ ， $R_i \leftarrow R_i + (R_l - LR_l)$ ， $\Delta r \leftarrow \Delta r - (R_l - LR_l)$ ，从Usable中消去 $C_l$ ，清除 $l$ 中的值，转(5)。

### 3 QoS 反馈控制方程

采用反馈控制方程来计算各服务类别的资源调整量，需要对各种状态的服务请求所占用的资源进行量化，然后根据各类别不同的服务等级约定比较实际服务质量与期望服务质量，根据差值进行资源调节量的计算，实现自动地区分服务质量保障。

本文使用一段采样时间内处于各种状态的请求来表征本

类占有的资源数。首先对不同状态的服务请求数进行表示： $nN$  为新增请求数； $nA$  为活跃请求数； $nI$  为空闲请求数； $nR$  为释放请求数。

使用状态空间法来对系统进行建模，使用采样时间内处于各种状态的请求作为状态向量，则对于采样时间  $k$  有状态向量  $x(k)$ ；对于控制输入，本类别所分配的资源，用  $u(k)$  表示：

$$x(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{pmatrix} = \begin{pmatrix} nN(k) \\ nA(k) \\ nI(k) \\ nR(k) \end{pmatrix} \quad u(k) = \begin{pmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \\ u_4(k) \end{pmatrix}$$

则对于某一服务类别，其状态空间模型：

$$x(k+1) = Ax(k) + Bu(k), y(k) = Cx(k), ref = F \times r$$

其中  $A, B, C$  均为  $4 \times 4$  矩阵，其值由具体机群中试验得出。 $y(k)$  表示采样时间  $k$  内本类别的实际服务质量向量。 $r$  表示本类别的期望服务质量向量，而  $F$  为权重向量，它与  $r$  的向量积就是本类别的期望服务质量， $F$  为  $1 \times 4$  矩阵。

相对于期望服务质量向量  $r$ ，控制误差为

$$e(k) = r - y(k) = r - Cx(k)$$

动态控制误差为

$$e(k+1) = r - Cx(k+1) = r - C(Ax(k) + Bu(k)) = r - C(AC^{-1}(r - e(k)) + Bu(k)) = CAC^{-1}e(k) - CBu(k) + (I - CAC^{-1})r$$

为了缩短系统的暂态响应时间，使得服务质量尽快准确地收敛于期望值，引入积分控制误差，它表示累积的控制误差。积分控制误差

$$x_I(k+1) = x_I(k) + e(k)$$

可以建立系统的增广状态空间模型为

$$\begin{pmatrix} e(k+1) \\ e_I(k+1) \end{pmatrix} = \begin{pmatrix} CAC^{-1} & 0 \\ I & I \end{pmatrix} \begin{pmatrix} e(k) \\ e_I(k) \end{pmatrix} + \begin{pmatrix} -CB \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} I - CAC^{-1} \\ 0 \end{pmatrix} r$$

则系统的状态空间反馈控制框图如图 3 所示。

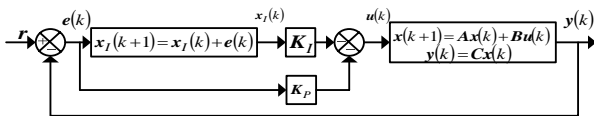


图 3 状态空间反馈控制框图

反馈控制系统中使用了这样的控制方法：

$$u(k) = -K \begin{pmatrix} e(k) \\ e_I(k) \end{pmatrix} = -\begin{pmatrix} K_p & K_I \end{pmatrix} \begin{pmatrix} e(k) \\ x_I(k) \end{pmatrix}$$

因此，可以得到系统的闭环模型为

$$\begin{pmatrix} e(k+1) \\ e_I(k+1) \end{pmatrix} = \begin{pmatrix} I - CAC^{-1} \\ 0 \end{pmatrix} r + \left( \begin{pmatrix} CAC^{-1} & 0 \\ I & I \end{pmatrix} - \begin{pmatrix} -CB \\ 0 \end{pmatrix} \begin{pmatrix} K_p & K_I \end{pmatrix} \right) \begin{pmatrix} e(k) \\ x_I(k) \end{pmatrix}$$

其中，反馈增益  $K_p$  和  $K_I$  可以通过经验法或根轨迹法根据实际需要的超调量和稳定时间进行选择，具体方法可以参考文献[6]中的控制分析与设计部分。

#### 4 仿真实验

服务质量要求可以采用请求的响应失败率、响应延时等作为性能指标，直接取决于服务类的资源需求和实际资源量，所以区分服务质量自我管理模型可以通过控制服务类别资源量来满足不同的服务等级约定。本文采用响应延时作为服务质量的指标。实验环境如下，Web 机群间采用 100 M 的以太网连接，机群服务器均采用 Linux2.6.14，调度分发器 Dispatcher 由 2 台服务器担当，后台服务器由 4 台模拟，另有 8 台 PC 作为测试客户端，采用 Scilent 模拟用户请求。

本文实现了 2 个具有不同服务质量需求的服务类别  $C_1, C_2$  的响应延时变化如图 4 所示，3 个服务类别  $C_1, C_2, C_3$  的响应延时变化如图 5 所示。

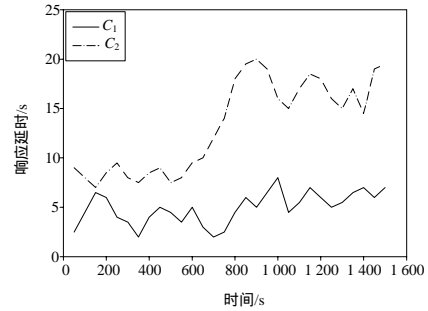


图 4 2 个服务类别的响应延时比较

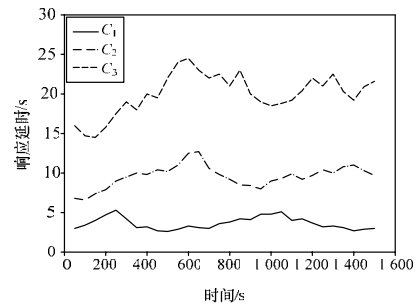


图 5 3 个服务类别的响应延时比较

通过实验可以看出，Web 机群区分服务质量控制自我管理模型能够很好地完成提供不同服务质量需求的工作。

#### 5 结束语

通过建立 Web 服务请求处理状态模型，引入反馈控制理论的状态空间法，建立了 Web 服务机群支持区分服务的自我管理模型。本文的工作由 Web 机群区分服务质量的控制入手，解决了目前服务器管理主要依靠系统管理员主观经验的问题，实现了区分服务质量控制的自我管理。保障网格用户的服务质量需求是当前的研究热点，如何基于服务请求内容使得 Web 机群能够实现资源配置的自优化是下一步的研究方向。

#### 参考文献

- [1] Bhatti N, Friedrich R. Web Server Support for Tiered Services[J]. IEEE Network, 1999, 13(5): 64-71.
- [2] Zhu Hui-Can, Tang Hong, Yang Tao. Demand-driven Service Differentiation in Cluster-based Network Servers[C]//Proceedings of IEEE INFOCOM'01. USA: [s. n.], 2001: 679-688.
- [3] Rashid M M, Alfa A S, Hossain E, et al. An Analytical Approach to Providing Controllable Differentiated Quality of Service in Web Servers[J]. IEEE Transactions on Parallel and Distributed Systems, 2005, 16(11): 1022-1033.
- [4] Lu Y, Saxena A, Abdelzaher T F. Differentiated Caching Services: A Control-theoretic Approach[C]//Proc. of the International Conference on Distributed Computing Systems. Washington D. C.: IEEE Computer Society, 2001: 615-622.
- [5] Cardellini V, Colajanni M, Yu P. Dynamic Load Balancing on Web Server Systems[J]. IEEE Internet Computing, 1999, 3(3): 28-39.
- [6] Astrom K J, Wittenmark B. Adaptive Control[M]. 2nd ed. MA, USA: Addison-Wesley, 1995.