

文章编号: 1001-0920(2012)01-0087-06

一种基于在线辨识的 Web 服务自适应接纳控制

赵吉, 孙俊, 须文波

(江南大学 物联网工程学院, 江苏 无锡 214122)

摘要: 设计并实现了一种基于量子行为粒子群算法(QPSO)系统模型在线辨识的 Web 服务自适应接纳控制, 根据系统模型的变化在线调节比例积分控制器参数. 通过接纳时间比反馈控制机制, 调整控制周期内服务器接纳请求的时间长度, 进而实现接纳控制. 通过仿真实验, 并与多种不同控制方法进行比较, 所得结果表明, 在线辨识自适应控制能够在服务器过载的情况下更有效地控制系统资源, 进一步提高了服务质量.

关键词: 在线辨识; 自适应控制; PI 控制器; 接纳控制; 量子行为粒子群算法

中图分类号: TP39

文献标识码: A

An adaptive admission control of Web service based on system model online identification

ZHAO Ji, SUN Jun, XU Wen-bo

(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China. Correspondent: ZHAO Ji, E-mail: queenji97@yahoo.com.cn)

Abstract: An adaptive admission control of Web service based on system model online identification using quantum-behaved particle swarm optimization(QPSO) is designed and implemented, which dynamically adjusts parameters of PI controller according to the changes of system model. In order to achieve admission control, a session-based admission time-ratio feedback control mechanism is introduced, which manages the amount of requests that the server can accept during the control period. The simulation results of Apache Web server compared with several control methods show that the adaptive control based on online system identification is able to control the system resources of the server more effectually in case of overload and further improves the service quality.

Key words: online identification; adaptive control; PI controller; admission control; quantum-behaved particle swarm optimization

1 引言

随着电子商务在 Web 平台上的广泛应用, Web 服务的 QoS 控制受到越来越多的关注. 在高负载下, Web 服务器的吞吐量会下降, 响应时间会增加, 从而造成网站的客户丢失和收益降低, 其原因是超载的 Web 服务器几乎一半的处理能力都消耗在中止/拒绝新的请求上. 防止服务器过载的一个主要方法是接纳控制^[1], 通过主动拒绝过量请求来控制请求速率, 使服务器实际接受的请求接近但不超过其最大处理能力, 这样系统在过载时仍可维持较高的处理量和较好的服务质量, 其控制机制的实现得到了广泛的研究.

随着控制理论在工业领域中取得巨大成功, 人

们尝试着将控制论引入 Web 服务 QoS 控制中^[2-6]. 但由于网络应用的复杂性, 为网络服务系统建立一个具有较强通用性的解析模型是非常困难的. 已有的经典控制理论的研究主要基于系统的线性模型, 并且要求对系统模型进行显式刻画; 对于复杂的网络服务系统, 在有效性和灵活性方面仍有不足, 当负载变化剧烈时适应性不强, 影响了 QoS 控制的效果^[7]. 自适应控制理论具有自我调节功能, 允许控制器在操作条件和负载发生变化时自动改变, 很适合应用于 Web 服务的控制中. 文献[8]提出了一种基于卡尔曼滤波的会话流自适应控制算法的 Web 服务器自适应模型, 但它仅给出了仿真结果, 并没有在实际的 Web 服务器环境中进行验证. [9]提出了一种基于 QoS 的自适应控

收稿日期: 2010-08-26; 修回日期: 2010-11-06.

基金项目: 国家自然科学基金项目(60703106, 60474030).

作者简介: 赵吉(1980—), 女, 博士生, 从事进化计算、人工智能的研究; 须文波(1946—), 男, 教授, 博士生导师, 从事人工智能、计算机控制技术等研究.

制中间件,通过在线辨识提高对软件系统的建模准确性和控制效果,说明了在线辨识自适应控制在软件系统的适用性,但并没有对 Web 服务器进行自适应控制.[10]提出了一种基于在线辨识的 Web 服务器自适应参考模型,能维持较好的比例延迟保证,但并没有对 QoS 的相关性能进行比较。

利用接纳时间比来控制接纳控制^[6],本文设计并实现了一种基于系统模型在线辨识的自适应控制,根据网络服务系统模型的变化调节控制参数,以动态调整 QoS 的性能控制。

2 基于控制论的接纳时间比控制

从控制论的观点看,一个典型的反馈控制系统包含 3 个基本组成部分:观测器、控制器和执行器^[11]。观测器负责采集受控系统的运行信息;控制器根据观测器采集的数据、控制模型及控制目标,在线推演下一步的控制动作;执行器则是控制动作的实施者,直接作用于受控系统。

基于控制论的接纳时间比^[6]的具体表达式为

$$ATR = t_{adm}/T_{ctrl} \quad (1)$$

其中: T_{ctrl} 为反馈控制系统的控制周期, t_{adm} 为控制周期内接纳请求(包括新会话请求)时间。在其余时间新会话请求将被拒绝。接纳时间比控制的基本思想是周期性侦测服务器 QoS 性能指标,如 CPU 利用率或响应时间,控制器根据实际观测值与期望值之间的误差及其历史信息计算下一周期的 ATR,并在该值确定的时间间隔—— t_{adm} 期间,服务器接收所有请求,在余下时间新会话请求将被视为过量负载而被拒绝。通过不断调整 ATR,防止系统过载,使得服务器的实际负载基本维持最大的服务能力,Web 服务器的 QoS 性能指标接近期望值。

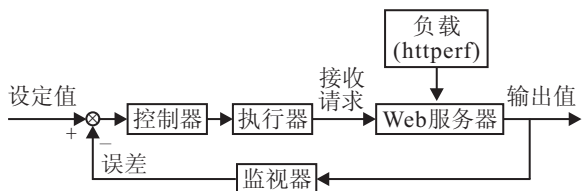


图 1 接纳时间比反馈控制系统结构图

在图 1 中,设定值是被控量(资源利用率或 QoS 指标)的期望值,输出则是相应的测量值。PI 控制器是控制系统的核心部分,其主要目的是维持 Web 服务器的输出值在期望值附近,并最终收敛于期望值。在每个控制周期内,将根据前一周期得到的测量值与期望值之间的误差反馈到 PI 控制器。控制器利用当前和历史误差信息,生成下一控制周期的控制变量 ATR,并输入执行器。执行器在每个控制周期末,从 PI 控制

器获取 ATR, Web 服务器按修正后的 ATR 具体实施接纳、拒绝操作,在本次控制周期内的输出更接近期望值。具体实施过程如下:

1) $ATR = 0$ 时,即 $t_{adm} = 0$,接纳标志为 0。在本次控制周期内,Web 服务器拒绝所有新会话的请求,只接收已处理会话的请求。

2) $ATR = 1$ 时,即 $t_{adm} = T_{ctrl}$,接纳标志为 1,当前控制周期内所有会话请求均可被接收处理。

3) $0 \leq ATR \leq 1$ 时,当前控制周期开始至 $(1-ATR) \times T_{ctrl}$ 时间段内,接纳标志为 0;余下时间段直到本次控制周期结束,接纳标志为 1。

3 系统模型

典型的 Web 服务器处理过程分为以下几个步骤: 1) 客户端向服务器请求建立传输控制协议(TCP)连接; 2) 服务器收到请求后生成处理进程/线程,并为连接分配资源; 3) 客户端收到连接建立成功后,发送 http 请求; 4) 服务器收到请求后先进行解析,然后进行相应的处理操作,处理完毕后将结果页面返回客户端。

Apache 服务器是一个广泛使用的开源服务器,具有处理 http 请求的典型结构,其工作进程池由主进程监控。客户端请求进入 TCP 监听队列等待处理器处理,服务器的每个工作进程负责处理与客户端的联通,并且最多可同时处理一个连接请求,只有处理完该请求后才能接受下一个请求。为了可以动态控制服务器,在服务器端实施了一个控制模块,它对 TCP 监听队列提供了一个获取/设置控制量的接口,输入的请求先进入监听队列,接纳控制模块对服务器接受请求队列进行控制,用来接纳或丢弃新的请求。

由系统输入和输出间数学关系(通常为差分、微分方程)表征的动态模型是控制器分析和设计的基础。将开环系统视为一个黑盒模型,通过对输入输出进行采样来推断 Web 服务器模型。单输入单输出(SISO)模型用来拟合开环 Apache 服务器。输入信号是 ATR,输出信号是 CPU 利用率或响应时间。输入信号必须满足持久激励条件,因此使用四阶 M 序列作为输入(ATR)信号。控制周期为 5 s。

虽然 Apache 服务器具有随机性和非线性的特点,但可以选择合适的线性时不变(ARX)模型来描述开环 Apache 服务器。因为如果一个线性模型能充分描述控制输入和系统输出之间的关系,则在一定范围内就能利用线性控制系统理论设计出兼具性能保障,又相对简单的反馈控制器。即使系统可以用一个完整的非线性模型描述,其第 1 步往往是根据系统的线性化来设计控制器的。此外,该模型是专门用于控制

器设计,并不需要系统模型非常精确,同时为了尽可能减少辨识系统的参数以及在线辨识的开销,使用一阶RAX便足够描述Web模型.

基于在线辨识的自适应控制是对过程进行在线辨识,其特点是在闭环条件下进行.闭环系统可辨识性的条件首先要求控制系统是闭环稳定的,因此,为了满足在线辨识自适应控制的闭环可辨识条件^[12]:调节器的阶次大于或等于对象的阶次,选择一阶ARX模型也是适用的,即

$$Y(k) + aY(k-1) = bU(k-1). \quad (2)$$

输入输出相应的 z 变换为

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b}{z-a} = \frac{bz^{-1}}{1-az^{-1}}. \quad (3)$$

对于采用PI控制器的离散系统,控制规律可以表示为

$$u(k) = u(k-1) + (k_p + k_i)e(k) - k_p e(k-1), \quad (4)$$

其中 $e(k) = r(k) - y(k)$,是参考值与输出值的误差.因此PI控制器的传递函数为

$$C(z) = \frac{U(z)}{E(z)} = \frac{(k_p + k_i)z - k_p}{z(z-1)}. \quad (5)$$

4 自适应控制器设计

以往的网络模型都是离线辨识,通过一组数据确定固定的模型参数.在实际情况下,网络的负载大都是变化的,具有固定参数的系统模型无法准确地表示实际的Web服务,因此描述服务器系统的模型参数也要进行相应改变.本文设计了一种自适应控制器,对服务器输入输出数据进行在线采集,周期性地对系统模型进行动态参数辨识,然后根据当前得到的系统模型计算PI控制器参数.

4.1 基于QPSO的系统在线辨识

4.1.1 量子行为粒子群算法

量子行为粒子群算法(QPSO)^[13]是一种新颖的粒子优化(PSO)算法.由于其每一迭代步的取样空间能覆盖整个搜索空间,QPSO算法是全局收敛的.实验表明,QPSO算法的性能优于PSO算法^[14].为了保证算法的收敛性,每个粒子必须收敛于各自的 p 点, $p = (p_1, p_2, \dots, p_d)$,第 i 个粒子 p 点的第 d 维坐标为

$$p = \varphi \cdot p_{id} + (1 - \varphi)p_{gd}, \quad \varphi \sim U(0, 1). \quad (6)$$

在QPSO算法中引入一个全局点 m_{best} 来计算粒子的下一步迭代,它定义为所有粒子的局部最好位置的平均值,即

$$m_{\text{best}} = \frac{1}{M} \sum_{i=1}^M P_i = \left(\frac{1}{M} \sum_{i=1}^M P_{i1}, \dots, \frac{1}{M} \sum_{i=1}^M P_{id} \right). \quad (7)$$

其中: M 为粒子的个数, P_i 为粒子 i 的局部最好位置.

于是粒子的迭代方程变为

$$x(t+1) = p \pm \beta \cdot |m_{\text{best}} - x(t)| \cdot \ln(1/u), \\ u \sim U(0, 1), \quad (8)$$

其中 β 被称为收缩扩张系数,调节该值能控制算法的收敛速度.一般而言, β 值在算法运行过程中从1.0线性减小到0.5时可达到较好的效果.算法详细流程可见文献[13].

4.1.2 基于QPSO算法的在线辨识

考虑式(2)描述的网络模型,输入 $U(k)$ 及输出 $Y(k)$ 可观测.该模型欲辨识参数为 $\hat{\theta} = [\hat{a}, \hat{b}]^T$.最小二乘法是一种经典的辨识方法,其结构简单、易于实现.但在具体使用时,不仅占用内存量大,而且不适合在线辨识.文献[15-17]介绍了将遗传算法及其改进算法应用于在线辨识,但算法都比较复杂,收敛速度慢,窗口长度长,辨识精度低.在线辨识需要反映实时情况,因此为了进一步减少辨识过程中算法的运算量,加速在线辨识的速度,提高辨识精度,将QPSO算法^[13]应用于系统在线辨识,粒子群中粒子每一维即是需要辨识的参数.QPSO算法用于系统在线辨识时,适值函数定义为

$$F_i = \sum_{k=1}^M (Y_k - \hat{Y}_k)^2. \quad (9)$$

其中: $Y(k)$, $\hat{Y}(k)$ 分别为实际输出与仿真输出; M 为窗口长度,相当于最小二乘限定记忆法中的数据记忆长度, M 值越大,精度越高,计算量也增加.

虽然式(2)描述的网络模型是一阶ARX模型,比较简单,但对于该系统的辨识还是一个多峰问题,使用最小二乘法进行在线辨识可能会陷入局部最优,因此使用全局搜索的QPSO可以保证全局收敛.此外,使用QPSO算法,其计算速度和最小二乘法的计算速度比较接近,其计算时间是可接受的.

对式(2)描述的系统,基于QPSO算法在线辨识步骤如下:

Step 1: 根据辨识参数个数 n ,按一定规模 N 初始化 n 维粒子的种群;

Step 2: 采样系统实际输出;

Step 3: 根据式(9)计算群体中每一个粒子的 F_i ;

Step 4: 判断收敛或达到最大迭代次数;

Step 5: 根据式(8)进化粒子;

Step 6: 转Step 2.

4.2 自适应控制器

采用间接自校正调节器^[18]对服务器进行控制,在线辨识自适应控制结构如图2所示.首先假设一阶线性时不变(ARX)模型总是适用于建立网络系统模

型. 即网络系统模型的结构保持不变, 但根据实时网络负载变化, 模型的参数将随之发生改变. 式 (2) 仍然代表系统模型结构, PI 控制器仍然用来调节控制变量 ATR.

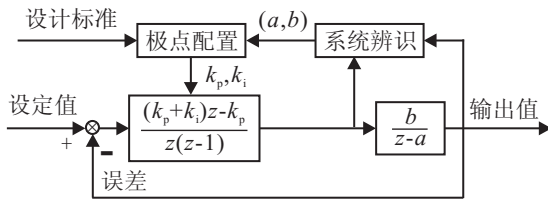


图 2 自适应控制结构框图

在每个控制周期, 根据 4.1.2 节方法利用 QPSO 算法在线辨识系统模型的参数 (a, b) . 更新网络模型参数 (a, b) 后, 将辨识参数输入极点配置模块, 该模块选择合适的 PI 控制器参数 k_p 和 k_i 从而保证闭环系统具有理想的极点值. 将计算得到的增益值传输给控制器模块进行 PI 控制.

PI 控制器进行控制时, 必须设置参数 k_p 和 k_i 以达到预期的控制目的, 如零稳态误差和足够小的整定时间. 为了保证闭环系统的稳定性, 同时为了克服自适应控制器对非最小相位系统产生不稳定的情况, 常用的 PI 控制器设计方法是极点配置, 即选择闭环系统的极点来满足某些所需条件. 极点配置自适应控制器的方法可以有效地使系统稳定. 利用极点配置进行控制器设计的步骤如下:

Step 1: 设定闭环系统预期的瞬态性能 (如整定时间);

Step 2: 从性能设定标准确定所需的闭环极点位置;

Step 3: 根据控制方法, 从开环系统模型导出闭环系统模型;

Step 4: 因为设定的闭环极点可从 Step 2 得到, 闭环系统模型是 k_p 和 k_i 的函数, 可从 Step 3 得到, 所以通过待定系数法, 即匹配闭环系统模型的极点与期望闭环极点计算控制增益, 解出 k_p 和 k_i 的值.

5 实验仿真及结果

5.1 实验环境及负载

网络由百兆网互联、一台服务器和一台客户机组成. 客户机配置为 Pentium IV 500 MHz, 512 MRAM, 服务器为 Pentium III 500 MHz, 512 MRAM, 操作系统均为 Linux 9.0. 服务器采用 Apache Web server 2.0.48^[19]. 服务器配置采用 Apache 组推荐的参数设置. 客户机安装 httpperf^[20] 软件产生负载. httpperf 是一款可以持续向服务器发送 http 请求的开源软件. 负载采用 httpperf 中的“-wssesslog”产生 session 请求负载. session 长度按指数分布, 均值为 30, 每个 session 随机

访问 540 个页面文件中的一个, 每次连接产生 10 个请求. Timeout = 10 s, 服务器响应超过 10 s 没有应答, 客户机将重发请求. session 之间的间隔 Think Time = 5 s. 这样的负载请求可以模拟静态和动态网页混合的动态网站.

5.2 系统模型

对系统模型进行在线辨识过程中, 窗口长度 M 设为 15, QPSO 算法参数 β 设为 0.7. 在线辨识是实时辨识, 当前辨识利用前 M 个数据确定本次模型参数, 并得到输出值.

利用最小二乘法 (LS) 进行离线系统参数辨识, 确定系统模型参数后该参数将无法改变. 离线辨识基于 CPU 使用率的系统模型为

$$\text{CPU}_k = -0.8653\text{CPU}_{k-1} + 0.5728\text{ATR}_{k-1}; \quad (10)$$

离线辨识基于响应时间的系统模型为

$$\text{Res}_k = 0.4128\text{Res}_{k-1} + 0.1548\text{ATR}_{k-1}. \quad (11)$$

基于 QPSO 算法系统在线辨识是一个动态的过程, 根据窗口数据确定每一次系统模型的参数. 在自适应控制系统中随着模型参数发生改变, PI 控制器的参数也是变化的. 通过实验结果比较, 可以验证在线辨识自适应控制结果的有效性.

5.3 实验结果

对服务器进行压力测试, 当负载达到 30 sess/s 后, CPU 的使用率超过 90%, 服务器响应时间显著下降, 有效吞吐率也随之降低. 因此该服务器负载能力为 30 sess/s.

实验负载变化如图 3 所示. 图 4 给出了基于 CPU 使用率控制的实验结果. 在没有控制时, 系统的利用率随负载增加而显著增加, 并且还有接近 100% 的点, 即系统达到饱和状态, 此时发生系统崩溃的可能性较大. 施加了 PI 控制, 并规定其 CPU 利用率为 50%^[6], 系统会被控制在规定值附近. 从图 4(a) 可以看出, 实施固定 PI 控制后, CPU 的使用率在 50% 附近波动, 但波动幅度依然较大; 而从图 4(b) 可以看出, 实施自适应控制后, CPU 使用率波动幅度明显变小, 更有效地实现了资源控制.

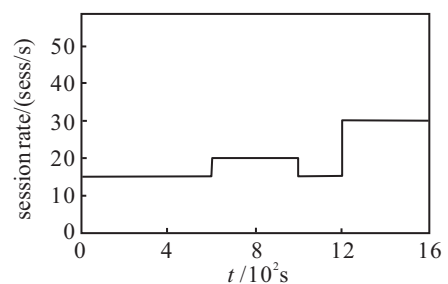


图 3 负载变化曲线

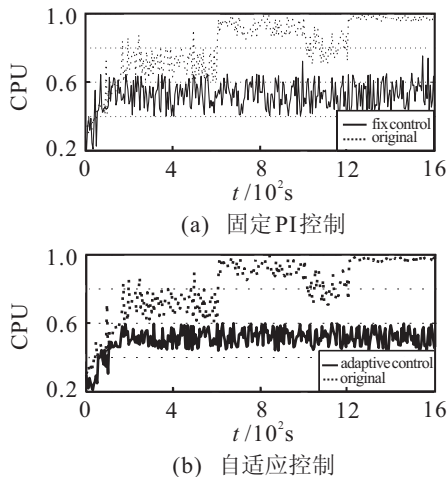


图 4 负载大波动下 CPU 资源控制

接纳时间比控制不但可以实现资源的有效利用和得到较好的服务质量, 还能保证满足用户特定的 QoS 指标. 假定用户要求响应时间不超过 100 ms, 下面验证基于响应时间的控制. 固定 PI 控制, 使用数学模型 (11), 响应时间的设定值为 100 ms, 图 5 是变负载时的响应时间曲线, 负载变化曲线见图 3. 从图 5 可以看出, 自适应控制得到的响应时间曲线更接近设定值附近, 比固定 PI 控制得到了更好的 QoS 指标.

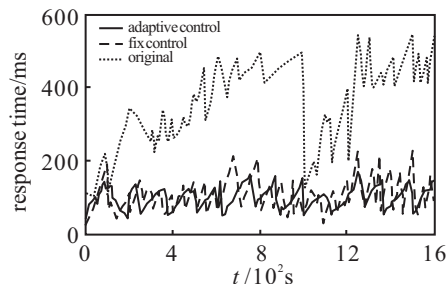


图 5 负载大波动下响应时间控制

表 1 不同方法平均值及方差

方法	CPU 利用率控制/50%		响应时间控制/100 ms	
	Ave	Std	Ave	Std
在线辨识 自适应控制器	0.503 587	0.069 204	99.781 3	28.255 04
离线辨识 固定控制器	0.522 321	0.084 353	107.105 3	40.185 8

从表 1 也可看出, 在线辨识自适应控制器得到的性能控制指标更接近设定值, 方差也比固定控制器得到的值更小.

为了进一步比较不同的控制方法对 Web QoS 性能的控制效果, 图 6 给出了基于 CPU 资源使用率的负载-性能比较图. 此时 CPU 设定值为 90%. 图 6(a) 为完成会话平均长度的比较. 会话平均长度越低, 网站效益的下降越显著. 图 6(b) 为平均响应时间的比较. 图 6(c) 为有效吞吐率的比较. 从图 6 可以看出, 在服务器严重过载的情况下, 自适应控制方法比固定 PI 控

制能更有效地保障服务器性能, 同时, 基于 QPSO 算法在线辨识的自适应控制比其他控制方式 (包括最小二乘法在线辨识的自适应控制方式) 都能得到更优的性能, 显著提高了控制的效果, 进而提高了服务器的性能.

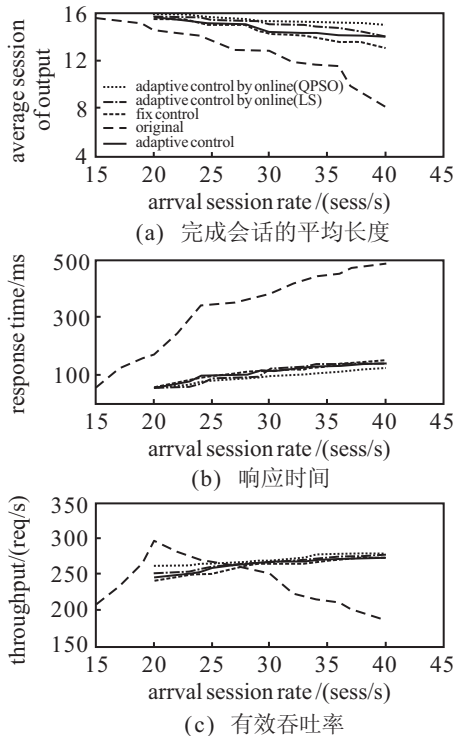


图 6 CPU = 90% 时服务器各项性能比较

图 7 为不同负载下不同自适应控制方法的平均响应时间的对比. 可以看出在不同负载的情况下, 对于设定的响应时间值 (100 ms), 基于 QPSO 算法在线辨识的自适应控制方法的控制效果相对于其他方法是最优的.

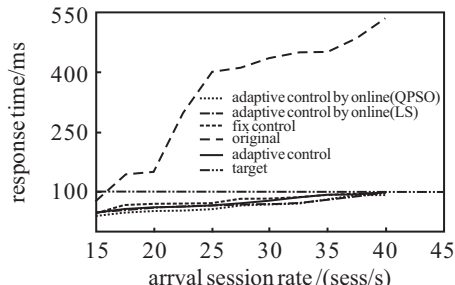


图 7 基于响应时间的在线辨识自适应控制

6 结 论

通过接纳时间比控制机制, 提出并设计了一种基于量子行为粒子群算法在线辨识的自适应反馈控制方法, 实现了动态调整 QoS 的性能控制. 仿真实验表明, 即使在严重过载的情况下, 通过在线辨识网络模型, 可动态调整 PI 控制器参数, 自适应控制系统性能. 该方法可以使系统资源得到合理分配、有效利用, 在不可预测的情况下有效地控制系统负载, 获得性能保

证, 对比不同的控制方法, 该方法能取得更好的服务器性能指标. 然而实际的网络 QoS 不局限于单一的控制性能指标, 往往需要同时满足多个控制目标, 因此, 多输入多输出 (MIMO) 模型的自适应控制系统将是下一步需要研究解决的问题.

参考文献(References)

- [1] Knightly E W, Shroff N B. Admission control for statistical QoS: Theory and practice1[J]. IEEE Network, 1999, 13(2): 20-29.
- [2] Lu C Y, Abdelzaher T F, Stankovic J A, et al. A feedback control approach for guaranteeing relative delays in Web servers[C]. Proc of the IEEE Real-time Technology and Applications Symp. Taipei: IEEE, 2001: 51-62.
- [3] Diao Y, Gandhi N, Hellerstein J, et al. Using MIMO feedback control to enforce policies for interrelated metrics with application to the apache Web server[C]. Proc of Network Operations and Management. Florence, 2002: 219-234.
- [4] Lu Y, Abdelzaher T F, Saxena A. Design, implementation, and evaluation of differentiated caching services[J]. IEEE Trans on Parallel and Distributed Systems, 2004, 15(5): 440-452.
- [5] Lu C Y, Lu Y, Abdelzaher T F, et al. Feedback control architecture and design methodology for service delay guarantees in Web servers[J]. IEEE Trans on Parallel and Distributed Systems, 2006, 17(9): 1014-1027.
- [6] 江滢, 孟丹. 基于接纳时间比控制和比例积分调节器的接纳控制机制[J]. 计算机研究与发展, 2007, 44(1): 65-70.
(Jiang Y, Meng D. Enforcing admission control using admission-time-ratio and PI controller[J]. J of Computer Research and Development, 2007, 44(1): 65-70.)
- [7] 单广志, 林闻, 肖人毅, 等. Web QoS 控制研究综述[J]. 计算机学报, 2004, 27(2): 145-156.
(Shan G Z, Lin W, Xiao R Y, et al. Web quality of service: A survey[J]. Chinese J of Computers, 2004, 27(2): 145-156.)
- [8] 李捷, 刘先省, 皇甫伟. 面向 Web server 的 QoS 自适应控制鲁棒性研究[J]. 计算机工程, 2006, 32(11): 24-28.
(Li J, Liu X X, Huang P W. Robustness of QoS self-regulating scheme for Web server[J]. Computer Engineering, 2006, 32(11): 24-28.)
- [9] 陈伟安. 基于 QoS 的自适应控制中间件研究与应用[D]. 长沙: 湖南师范大学数学与计算机科学学院, 2008.
(Chen W A. A QoS-based adaptive control middleware research and application[D]. Changsha: College of Mathematics and Computer Science, Hu'nan Normal University, 2008.)
- [10] 胡延苏, 戴冠中, 高昂, 等. Web 服务器上比例延迟保证的自适应控制方法[J]. 计算机应用研究, 2010, 27(1): 336-338.
(Hu Y S, Dai G Z, Gao A, et al. adaptive relative delay guarantee scheme for Web server[J]. Application Research of Computers, 2010, 27(1): 336-338.)
- [11] Franklin G F, Powell J D, Workman M. Digital control of dynamic systems[M]. The 3rd ed. New York : Addison-Wesley, 1998: 21-42.
- [12] 谢新民, 丁锋. 自适应控制系统[M]. 北京: 清华大学出版社, 2002: 116-160.
(Xie X M, Ding F. Adaptive control system[M]. Beijing: Tsinghua University Press, 2002: 116-160.)
- [13] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior[C]. Proc of 2004 Congress on Evolutionary Computation. Piscataway, 2004: 325-331.
- [14] Sun J, Xu W B, Feng B. Adaptive parameter control for quantum-behaved particle swarm optimization on individual level[C]. Proc of 2005 IEEE Int Conf on Systems, Man and Cybernetics. Hawaii, 2005: 3049-3054.
- [15] 黄炯, 邬永革, 李军, 等. 基于遗传算法的系统在线辨识[J]. 信息与控制, 1996, 25(3): 171-176.
(Huang J, Wu Y G, Li J, et al. On-line system identification based on genetic algorithms[J]. Information and Control, 1996, 25(3): 171-176.)
- [16] 杨旭东, 张彤, 张家余. 遗传算法应用于系统在线辨识研究[J]. 哈尔滨工业大学学报, 2000, 32(1): 102-104.
(Yang X D, Zhang T, Zhang J Y. Genetic algorithm for on line system identification[J]. J of Harbin Institute of Technology, 2000, 32(1): 102-104.)
- [17] 李孝安, 张小绩, 戴冠中. 一种基于遗传算法与进化编程的系统辨识方法[J]. 控制与决策, 1996, 11(3): 404-407.
(Li X A, Zhang X H, Dai G Z. An approach for system identification based on genetic algorithms and evolutionary Programming[J]. Control and Decision, 1996, 11(3): 404-407.)
- [18] Astrom K J, Wittenmark B. Adaptive control[M]. New Jersey: Prentice Hall, 1994: 51-100.
- [19] Apache Web server. <http://www.apache.org>.
- [20] Mosberger D, Jin T. Httpperf: A tool for measuring Web server performance[C]. SIGMENTRICS Workshop on Internet Server Performance. Madison, 1998: 31-37.