开放式实时系统中的自适应调度方法

邹 勇 淮晓永 李明树

(中国科学院软件研究所互联网软件技术实验室 北京 100080)

摘 要 首先针对开放式实时系统,讨论了自适应实时调度的需求情况和自适应技术应用上的关键问题;提出了适用于硬实时调度需求的调度参数自适应调整机制;重点面向软实时调度需求,提出了一种基于模糊控制策略的自适应调度方法,它致力于动态跟踪调度对象的负载变化,并把截止期错过率控制在期望值附近.相对于现有方法,更适合于解决开放式实时系统中的自适应调度问题.

关键词 实时;开放系统;调度算法;自适应控制;截止期错过率

中图法分类号 TP316

The Adaptive Scheduling Approaches to Open Real-Time Systems

ZOU Yong HUAI Xiao-Yong LI Ming-Shu

(*Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

Abstract The adaptive scheduling problems for open real-time systems are discussed in this paper. In an open real-time system, tasks are loaded dynamically and their schedulability is tested on line. At the mean time, how much CPU bandwidth the tasks need may not be known as a priori. This feature urges us to develop adaptive real-time scheduling approaches which can allocate resources to tasks dynamically at run time. According to the scheduling requirements of the open real-time system, adaptive scheduling approaches are presented respectively for hard real-time tasks and soft real-time tasks. Firstly, an approach based on feedback control is presented for the hard real-time tasks scheduling. Secondly, for soft real-time applications, another adaptive scheduling approach is presented. The second one is based on fussy control technology. It can control the deadline missing ratio near the expected value while sampling the applications' performance. These adaptive scheduling approaches are developed for open real-time system, with the merits that they can schedule different kinds of real-time tasks and adapt to their dynamic resource requirements through tuning the schedulers.

Keywords real-time; open system; scheduling algorithms; adaptive control; deadline missing ratio

1 引 言

现有的许多实时调度研究专注于如何通过处理机带宽预留、可调度性检验等手段来确保实时系统避免错误(如错过截止期、系统过载等)的发生.这种调度方式是开环的,即系统的调度策略和参数一旦确定之后,在运行中不再根据反馈信息对它们进行调整.因而需要预知实时任务与调度相关的确切信息,只能适用于实时需求及系统模型确定的封闭式

收稿日期:2002-05-09;修改稿收到日期:2003-03-17.本课题得到国家自然科学基金重大项目(69896250-3)和国家"八六三"高技术研究发展计划(2001AA113201)资助.**邹 勇**,男,1976 年生,博士研究生,主要研究方向为实时系统、操作系统. E-mail: zouy@itechs.iscas.ac.cn. **淮晓永**,男,1973 年生,博士,主要研究方向为实时系统、智能系统、构件软件. **李明树**,男,1966 年生,研究员,博士生导师,主要研究方向为实时系统、智能软件工程.

实时环境[1,2].而具有自适应功能的实时系统则致力于在运行时进行动态的调整,以控制实时错误的发生频率.这往往需要闭环的调度策略,即采集当前调度的输出和实时需求满足情况,并作为反馈量与期望值比较;再由控制算法产生控制量,动态调整调度器等部件以适应实际运行环境或负载变化,从而满足期望的实时性能指标[1,3].

"开放式实时系统"的理论与方法在近年受到人们的重视,也是一个研究热点.在这样的系统中,多种调度策略并存,并且非相关实时应用可单独进行开发和验证,而不必关心系统中同时存在的其它实时应用[4,5].它的特点更需要自适应技术的支持:

(1) 开放式实时系统中,任务可动态加载,并进行可加载性测试.因此,新到达的应用往往未知其所需的资源量;

(2) 开放式实时系统的软件环境若需适应异构的硬件平台,则其运行环境的速度差异对调度参数的要求会不同;

(3) 调度对象所包含的任务可能会随着时间的变化而有较大的负载变化.

这里的"调度对象"是指被系统调度的各种任务、任务组或其它成分.自适应技术与开放式实时系统相结合是实际需求造成的必然趋势,然而,目前的自适应实时调度研究还不能满足这方面的需要.例如近年提出的 FC-EDF 方法是典型的闭环反馈控制调度[1],它可根据系统总负载变化情况,通过准入控制和服务等级调节使系统运行满足一定的实时性能指标.通过增加处理机利用率的采样和控制环节,人们又对它进行了改进[6],使系统输出更加稳定.但是这些方法都只能依据系统的全局负载变化进行自适应调度,而无法对开放式实时系统下的各个应用程序作针对性的局部控制,也不适合开放式实时系统中多种调度策略并存的情况.

本文首先分析了开放式实时系统的典型结构和关键理论,并讨论了调度对象模型;其次针对硬实时调度,提出了一种调度参数自适应调整机制;然后重点面向软实时应用的自适应实时调度需求,提出了一种基于模糊控制技术的详细解决方案,并做了充分的讨论和模拟实验研究;最后总结全文.

2　开放式实时系统调度机制及关键问题

2.1　调度对象模型和相关概念

开放式实时系统的调度对象类型不再是单一的,而是具有层次性特征.文献[4]提出了调度对象的层次模型,本文以下的讨论将基于这个模型.

首先,任务是指完成某一特定功能的软件实体,它是实时调度中的基本单位.任务在其生命期中的某一次执行称为该任务的一个作业.其次,在实际中,一个完整的应用往往是由一组任务共同构成的,我们把组成一个应用的任务的集合称为任务组.

从应用开发者的角度看,一个特定的任务组有其适合的调度方法与之对应,允许系统中的各个任务组可选择使用不同的调度器将极大方便应用开发,并取得好的应用效果.这也是开放式实时调度的主要功能特性之一,因此有以下定义.

定义 1.　在开放式实时系统中,把应用开发者确定的一个任务组及其相应的调度器称为一个任务组-调度器对(Task-Set and Scheduler,TSS).

实际上,没有配置调度器的任务组也可看作 TSS 的一种特殊情况,即认为它们有一个不做任何事的调度器.所以,TSS 的概念是任务组或任务的超集.

综上,开放式实时系统的调度对象可分为三个层次:任务、任务组、TSS.

目前已有几种实时调度方法研究与开放式实时系统相关,它们基于服务器调度策略[4,7,8].这里的"服务器"是指系统调度机制所创建的特殊任务,它为调度对象提供资源服务.服务器为它所关联的调度对象提供所需的运行资源,可看作一个具有确定速率的虚拟处理机.若干个服务器及其调度对象可共存于系统中.

从服务器的角度看,调度对象的范围包括单独的任务、任务组和 TSS.以下若未作特殊说明,假设具有以下特性:

(1) 各个调度对象之间互相独立;

(2) 调度对象的每个任务没有不可抢占区;

(3) 非实时应用作为具有非实时调度器的 TSS 对待.

定义 2(截止期错过率(Deadline Missing Ratio,DMR)).　假设在某一段确定的时间内,错过其截止期的作业数为 n,正常完成的作业数为 m,则

$$DMR=\frac{n}{m+n}.$$

DMR 是描述实时任务集在实际运行中,其实时需求满足情况的一个重要衡量值.

定义 3(服务器速率).　不失一般性,假设总的系统处理机的速度为 1,当把服务器 S_i 看作一个虚

拟处理机时，服务器速率 σ_i 指 S_i 的速度相对于系统处理机速度的比值[9].

这个比值表示了服务器占用系统处理机总带宽的比例，所以有 $0<\sigma_i\leq 1$.

2.2 研究现状及有待解决的关键问题

目前，基于服务器调度策略的开放式实时系统一般采用双层的调度框架[4,7,9]. 它实际上是致力于为系统中的每个非相关应用提供一个具有确定速率的虚拟处理机，这是此类调度方法的关键所在. 它从理论上可以确保：若一个 TSS 在专有处理机上运行时可调度，则在具有相同速率的服务器上运行时也可调度[4,9]. 显然，在这样的开放式实时系统中，服务器所分配的速率大小是决定它上面运行的 TSS 能否满足实时性能指标的关键因素.

目前这方面的研究是基于已知实时应用所需的确切服务器速率的假设. 但在实际中存在这样一些关键问题：对于新到达的应用程序，它对服务器速率大小的需求往往是未知的；对于要移植到异构硬件平台上的应用程序，也未知其在新环境中所需的服务器速率大小；对于已在系统中运行的应用程序，若它的负载需求随时间变化，那么当前的运行条件可能将不能满足需求，导致它的 *DMR* 产生不可预测的变化. 所以，如何解决这些问题，并使开放式实时系统的调度具有自适应功能，是理论应用于实际所需解决的重要问题.

3 面向硬实时应用的服务器速率自适应方法

3.1 问题描述

开放式实时系统中，一方面，具有硬实时需求的 TSS 要求其所有作业的截止期都得到满足，因此在它本身的正确性已经被保证的条件下，还需要足够的服务器速率；另一方面，为它分配的服务器速率过大将会造成系统资源的浪费. 所以，在 TSS 加载后的一段时间里，自适应地调整分配给它的服务器速率具有重要意义.

不同类型的服务器有不同的支持算法，因此它的速率自适应方法也可能不同. CUS(Constant Utilization Server)和 TBS(Total Bandwidth Server)[9]是可支持硬实时应用的两种服务器，这里即针对它们提出了一种服务器速率自适应方法.

调度对象的已知条件：

(1) 作业截止期的期望值.

(2) 每个作业的最坏执行时间(Worst Case Execution Time，WCET)的预估值.

(3) 存在一个小于 1 的服务器速率值，使 TSS 的局部调度器可调度它的任务组.

3.2 CUS 和 TBS 服务器的自适应调整

设 S_i 为系统中的一个 CUS 或 TBS 服务器，假设在 t 时刻，作业 $J_{i,j}$ 是 S_i 的就绪队列头部的作业；$J_{i,j}$ 的截止期期望值为 $sd_{i,j}$；$J_{i,j}$ 的 WCET 预估值为 $e_{i,j}$，它减去 $J_{i,j}$ 已执行的时间后，剩余的部分为 $e'_{i,k}$；S_i 的速率当前值为 $\sigma_{i,k}$，S_i 的当前截止期为 $d_{i,k}$. 根据 CUS 和 TBS 服务器的支持算法，$\sigma_{i,k}$ 与其它参数之间有以下关系[9]：

$$\sigma_{i,k}=e'_{i,k}/(d_{i,k}-\max\{t,\ d_{i,k-1}\}) \tag{1}$$

假设在 t 时刻之前 $J_{i,j}$ 未被执行过，则我们可用 $e_{i,j}$ 和 $sd_{i,j}$ 分别替换式(1)中的 $e'_{i,k}$ 和 $d_{i,k}$，用式(2)计算，可得到 S_i 的速率在以 WCET 预估值为依据的情况下的值：

$$\sigma'_{i,k}=e_{i,j}/(sd_{i,j}-\max\{t,d_{i,k-1}\}) \tag{2}$$

但是因为此速率值 $\sigma'_{i,k}$ 并不是由 WCET 实际值计算得到，所以需要在实际运行中进行测量和调整. 设 $J_{i,j}$ 的 WCET 实际值为 $ae_{i,j}$，若 $e_{i,j}<ae_{i,j}$，则显然会造成 $e'_{i,k}$ 偏小，从而也使 $\sigma_{i,k}$ 的值偏小；反之，则会造成 $\sigma_{i,k}$ 的值偏大. 我们通过反馈控制机制来解决这个问题.

每当一个作业执行完毕时，我们对它当前的 WCET 实际值 $ae_{i,j}$ 和实际完成时刻 $f_{i,j}$ 进行采样，若

$$sd_{i,j}-f_{i,j}<0,$$

则说明 S_i 的当前速率不能满足需求，需要进行调整. 而此时必有

$$e_{i,j}-ae_{i,j}<0.$$

所以，首先要调整 $e_{i,j}$，使它与 $ae_{i,j}$ 相等；其次，根据调整后的值，用式(2)计算新的服务器速率值 $\sigma'_{i,k}$. 调整量 $\Delta\sigma_{i,k}$ 便可以用式(3)得出：

$$\Delta\sigma_{i,k}=\sigma'_{i,k}-\sigma_{i,k} \tag{3}$$

如图 1 所示，作业队列头部的作业 $J_{i,j}$ 由 S_i 服务，当它执行完毕时，系统采样它本次运行实际所用时间，得到 WCET 实际值 $ae_{i,j}$；同时也采样它的实际完成时刻 $f_{i,j}$. 然后，$ae_{i,j}$ 和 $f_{i,j}$ 作为反馈值分别与 WCET 预估值 $e_{i,j}$ 和截止期的期望值 $sd_{i,j}$ 相比较，若偏差显示需要调整，则计算控制量 $\Delta\sigma_{i,k}$，对服务器 S_i 的当前速率进行调整. 与通常所见的周期性采样、控制情况不同，这里的采样时间并不是一个固定的周期，而是在每个作业完成的时刻做一次采样.

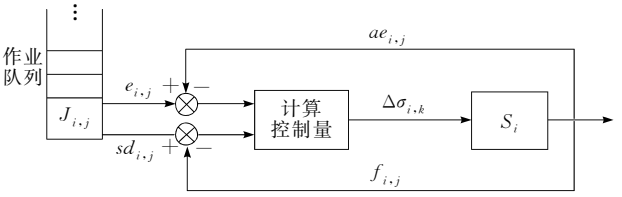

图 1　服务器速率自动调节

需特别注意的是,服务器速率的预设值可以小于或等于实际需求值,但在自适应调节时,应满足 $\Delta\sigma_{i,k}\geq 0$. 这是因为在硬实时要求下,S_i 的速率值必须满足它所服务的作业队列的最大需求. 在经过一段时间的运行与调整之后,S_i 的速率值将由一个较小的预设值单调增到一个稳定值,它是所有已运行作业所需求的最大值. 此时可根据具体情况认为该稳定值是实际需求值.

以上对 CUS 和 TBS 服务器速率的自适应调整方法基于反馈控制机制,它自动化了硬实时任务的资源需求量化求解过程,从而节省人工实验和调整所带来的时间和成本代价.

4　面向软实时应用的自适应调度方法

4.1　应用背景及实际需求

目前,在开放式实时系统的应用中,具有软实时需求的情况日益增多. 例如,多媒体信息传输和解码,网络服务器应用中对服务请求的实时响应等. 这类实时需求基本特点是,要求 *DMR* 被控制在一个小的期望值附近,即实现满意的实时,并不要求严格确保每个作业的截止期. 另外,许多软实时应用的负载需求随时间动态变化并不可预测. 例如,MPEG4 解码运算中,各帧数据所耗的处理机时间不同,而且可能在某些帧出现远离平均值的峰值[8];再如,单位时间内网络服务请求的数量往往是随机的,造成所需的系统资源量也随时间变化.

对这一类实时应用,若以它们的最大峰值为依据来分配服务器速率,则会造成多数情况下的系统资源浪费. 因此,理想的解决方法是:根据负载的动态变化,自适应地进行优化的资源分配.

4.2　控制器的选择

传统的 PID 控制器可有效地解决一些自适应调度问题[3]. 但是在开放式实时系统中,若干个不同类型的服务器同时运行,它们又为多种类型的应用提供实时服务,这使得被控对象具有动态性和不确定性. 所以为系统建立精确的数学模型不实际,也很耗时. 另外,控制参数的自整定也是一个不易解决的问题. 所以,PID 控制器在这里的应用受到局限.

我们采用不依赖于具体被控对象模型的模糊控制器[10]来解决开放式实时系统的自适应调度问题. 它根据 *DMR* 的偏差和偏差变化率,通过基于控制规则的决策计算求得一个控制量,然后调节服务器速率,或者进行运行等级调整和准入控制,从而实现控制基上的自适应实时调度. 模糊控制器无需了解被控对象的数学模型,并且无论被控对象是线性的还是非线性的都能有效控制,具有良好的适应性[10]. 所以,它相对于 PID 控制器更适合开放式实时系统的自适应调度.

4.3　系统原理和结构设计

4.3.1　模糊控制器

基本模糊控制器的结构如图 2 所示. 当前的偏差 $e(k)$、偏差变化率 $ec(k)$ 是根据被控对象的输出采样值和期望值计算而得,经过隶属函数进行模糊化得到模糊输入信号,然后依据控制规则进行模糊逻辑推理,推理求得的结论再通过最大隶属度法、重心法等精确化方法[10],求得精确的控制量 $u(k)$,进而执行控制动作,使得系统的偏差快速、稳定地趋向于 0.

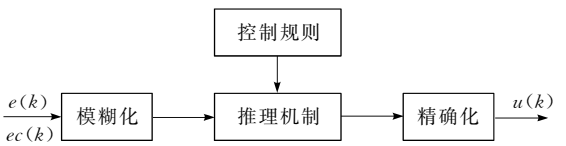

图 2　基本模糊控制器

考虑到实时调度的快速性要求,这里通过对图 2 的基本模糊控制器进行离线的模糊逻辑推理运算,得到一种查表结构的快速模糊控制方案. 如图 3 所示,决策表在离线时就已获得,而在运行时,只需根据当前的偏差 $e(k)$、偏差变化率 $ec(k)$,通过量化因子 k_e,k_{ec} 转换到决策表论域来查找决策表,再通过比例因子 k_u 转换到实际控制量论域,即可得出实际控制量 $u(k)$.

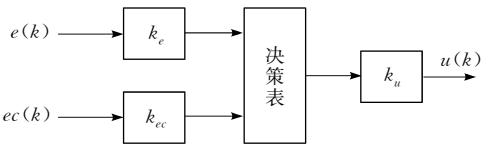

图 3　快速模糊控制

4.3.2　系统结构设计

通过将上述模糊控制机制与开放式实时系统的服务器调度机制相结合,设计了可适用于负载动态

变化的软实时应用的自适应调度方法.其系统结构如图 4 所示.每次采样时间,服务器 S_i 上运行的 TSS 的当前截止期错过率 $DMR(k)$ 被采样,并作为反馈量与期望值 DMR_s 相比较,经过模糊决策模块后产生控制量 $\Delta\sigma$;然后,控制方法选择模块依据 $\Delta\sigma$ 和当前系统资源的总利用率情况选择一种合适的方法进行调整.其中,S_i 速率调整是通过向系统申请或减少计算资源占用率来达到调节目的;运行等级调整是指,当同一任务具有不同负载需求的运行副本时(它们的计算精度也不同),通过使用不同的运行副本(调整运行等级)以使 $DMR(k+1)$ 被控制得更趋近于 DMR_s;运行请求准入控制是通过限制或允许新的运行请求来达到同样目的.

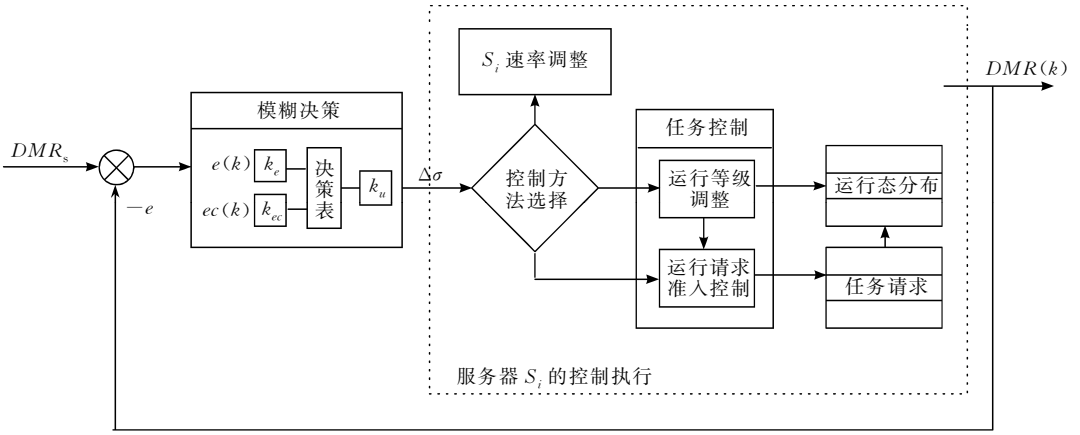

图 4 服务器 S_i 的负载动态变化自适应方案

DMR 是衡量实时应用的时间约束得到满足情况的量化指标,选择 DMR 作为系统的被控变量,既可满足实时性能需求,又可顾及到系统计算资源的充分利用.服务器速率是服务器及其上的 TSS 对系统计算资源占用情况的量化指标,它是决定实时性能的关键因素.因此,模糊决策输出的控制量是对服务器速率的调整值,图 4 中的另外两种控制方法根据调整值是增量还是减量来做相应的控制动作.

4.4 采样间隔计算策略

相邻两次采样时间的间隔值对于保持稳定的控制输出有重要影响.间隔过小,会产生频繁调节和不必要的系统开销;间隔过大又无法及时地跟踪实时应用的负载变化.这里根据不同类型任务组的特点,使用不同的策略计算采样间隔时间:

(1) 对于周期型任务组,当它的超周期(即组内所有任务的周期的最小公倍数)为 T_{SP},一个超周期内执行任务数为 n 时,则采样周期为 $T=\lambda\frac{100}{n}T_{SP}$,其中 λ 是一个比例因子,可以通过调节它来改变控制的频率.在每个超周期中所有任务都可得到一次以上的完整运行,并且各超周期内的运行情况相同(在任务的时间约束都被满足的条件下).

(2) 对于非周期型任务组,可累计到达的作业数目 m 以及其中错过截止期的作业数目 n.当 n 到达一定值时,便计算一次该时间段的 $DMR(k)$.另外设定一个最长采样间隔 $T_{\max}$,只要当前时刻距上次采样时间超过 $T_{\max}$,就做一次采样.这是为了在任务负载需求较低时,可通过降低服务器速率向系统释放计算资源.n 值和 $T_{\max}$ 的值可由用户根据具体的调度对象对控制频率的实际需求来灵活确定.

(3) 对于周期任务和非周期任务混合的复合型任务组以及其它情况可使用策略 2.

4.5 量化因子、比例因子及隶属函数的计算

如 4.3.1 节所述,量化因子 k_e,k_{ec} 和比例因子 k_u 是实现论域间转换的关键.设变量 x 的论域为 $X=[x_L,x_H]$,其模糊隶属函数的模糊集论域为 $[m,n]$,则相应的量化因子 k_e,k_{ec} 采用式(4)计算;比例因子 k_u 采用式(5)计算.采用式(6)把 X 论域的精确值 a 转化为模糊集论域中的元素 b;采用式(7)把模糊集论域中的元素 c 转化为 X 论域的精确值 d.

$$k=\frac{n-m}{x_H-x_L} \tag{4}$$

$$k_u=\frac{x_H-x_L}{n-m} \tag{5}$$

$$b=k\left(a-\frac{x_H+x_L}{2}\right) \tag{6}$$

$$d=k_uc \tag{7}$$

考虑开放式实时系统中调度对象的多样性,以广泛适用性为目的,这里采用一般经验的模糊子集、

隶属函数和模糊规则定义方法.如果服务器只是针对某一类实时应用,则可以根据实际对象特性、实际控制数据,对模糊子集个数、隶属函数形状以及模糊规则集等再进行优化计算.

模糊集和论域取值范围是与实际被控对象相关的,用户可根据具体需求对其进行更改.这里取偏差 E 的模糊集为 $E=\{NB,NM,NS,NZ,PZ,PS,PM,PB\}$,偏差变化率的模糊集为 $EC=\{NB,NM,NS,ZE,PS,PM,PB\}$,控制量的模糊集为 $U=\{NB,NM,NS,ZE,PS,PM,PB\}$.其中,$NB$=负大,$NM$=负中,$NS$=负小,$NZ$=负零(−0),$PZ$=正零(+0),$ZE$=零,$PS$=正小,$PM$=正中,$PB$=正大.相应的,分别将偏差、偏差变化率和控制量的基本论域进行量化,转化为若干个离散的量化等级.取偏差的量化等级为 $e=\{-6,-5,-4,-3,-2,-1,-0,+0,1,2,3,4,5,6\}$,偏差变化率的量化等级为 $ec=\{-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6\}$,控制量的量化等级为 $u=\{-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7\}$.

隶属函数是对模糊概念的定量描述,它表示变量的模糊集论域中的某个值隶属于其模糊集中的某个值的程度.隶属函数的确定方法有模糊统计法、例证法和专家经验法等.正态型的隶属函数是较为常用的,也适合于许多实时应用的实际情况[10].这里的偏差、偏差变化率和控制量的隶属函数为正态分布的,具体函数参数可依据实际情况再作调整,这里不再赘述.

4.6　模糊控制规则和决策表

基于 E,EC 的模糊控制规则的一般描述形式为

IF $e(k)$ is E_i and $ec(k)$ is EC_i THEN $u(k)$ is U_i.

根据 4.5 节中 E,EC 和 U 的模糊集定义以及偏差控制的经验,可以得出 DMR 偏差、偏差变化率和控制量之间的模糊规则,如表 1.建立模糊控制规则表的基本思想是:根据偏差和偏差变化率的情况取相应的控制量.例如,当偏差为 NM,偏差变化率为 NM 时,说明此时偏差为负且有增大的趋势,控制量应取 PB,以尽快消除已有的偏差并抑制偏差变大.

由表 1 的模糊控制规则,可以按照 Min-Max 模糊逻辑推理算法[10]计算,并采用最大隶属度精确化方法,转化求得控制决策表.该表主要是由偏差和偏差变化率的模糊集论域值得到控制量的模糊集论域值,然后再通过比例因子 k_u 由式(7)计算得到实际控制量 $u(k)$.控制决策表可作为二维数组存储在计算机中,系统实际运行时只需根据采样的偏差和偏差变化率查表即可,因此是一种快速控制决策,适合对效率和速度要求较高的领域.

表 1　模糊控制规则表

E	U						
	$EC=NB$	$EC=NM$	$EC=NS$	$EC=ZE$	$EC=PS$	$EC=PM$	$EC=PB$
NB	PB	PB	PB	PB	PM	ZE	ZE
NM	PB	PB	PB	PB	PM	ZE	ZE
NS	PM	PM	PM	PM	ZE	NS	NS
NZ	PM	PM	PS	ZE	NS	NM	NM
PZ	PM	PM	PS	ZE	NS	NM	NM
PS	PS	PS	ZE	NM	NM	NM	NM
PM	ZE	ZE	NM	NB	NB	NB	NB
PB	ZE	ZE	NM	NB	NB	NB	NB

4.7　控制方法的选择算法

如图 4 所示,这里提供了三种控制方法:(a)服务器速率调整;(b)运行等级调整;(c)运行请求准入控制.

方法(a) 根据控制量的值对服务器速率大小进行调整;

方法(b) 若控制量为正,则说明 DMR 过大,需减小一个运行等级;若控制量为负,则说明 DMR 小于期望值,需增大一个运行等级;

方法(c) 若控制量为正,则说明 DMR 过大,停止接受新运行请求;若控制量为负,则说明 DMR 小于期望值,允许接受新运行请求.

后两种方法是前一种方法无法实施时的补充方案.具体选择算法如下:

1. 根据控制量计算服务器速率的调整值 $\Delta\sigma$;
2. 若 $\Delta\sigma=0$,则结束;
3. 若 $\Delta\sigma<0$,则转 8;
4. 若 $\Delta\sigma$ 大于系统剩余处理机带宽,则转 6;
5. 用方法(a)进行服务器速率调整,结束;
6. 若运行等级大于最低级,则使用方法(b),结束;
7. 使用方法(c),结束;
8. 若运行请求准入控制为关闭状态,则放开,结束;
9. 若运行等级小于最高级,则使用方法(b),结束;

10. 使用方法(a)进行服务器速率调整,结束.

以上算法的主要思路是:

(1) 当 *DMR* 过大时,首先用增加服务器速率的办法,即优先使用方法(a),若不可行则再通过降低服务质量(用方法(b))控制 *DMR*. 在以上两种方法仍不能满足 *DMR* 需求时,最后采用停止接受新任务运行请求的方法(用方法(c));

(2) 当 *DMR* 过小时,首先使用后两种方法(用方法(b)和(c))提高服务质量,达到最高的运行等级后再降低服务器速率(用方法(a)).

这样设计的好处是一方面可保证在系统资源充足时能提供最好的服务质量,并且服务器速率也不会过大;另一方面在系统资源不足时,也可保证一定等级的服务质量下的实时性能.

5 模拟实验

当系统负载和调度对象所需处理机带宽都在不断变化时,调度对象的 *DMR* 与其期望值的偏差变化情况是衡量自适应实时调度方法的一个重要标准.

我们设计了模拟实验方案,利用仿真工具对第 4 节所提出的自适应调度方法的实际效果做了实验. 实验中,假设系统中有多个实时任务并发运行,被模拟的调度对象为一个软实时周期任务,它有 3 个运行等级. 每个运行等级在各个周期内所需的执行时间在一定的范围内随机变化,相应的所需要占用的处理机带宽也在一定范围内随机变化,具体的参数见表 2.

表 2 调度对象的运行参数

运行等级	处理机带宽需求变化范围
1	0.12～0.16
2	0.16～0.20
3	0.20～0.24

各个运行等级的周期均为 1ms,*DMR* 期望值设定为 0. 调度对象的运行等级是随系统所能提供的处理机带宽的变化而变化的. 若系统所能提供的处理机带宽不能满足调度对象达到 *DMR* 期望值的需求,则运行等级降低,反之,则升高.

在所设定的实验条件中,可分配给该任务的处理机带宽也是不断变化的(因为在实际中系统总负载可能是动态变化的),因而可供调度对象使用的处理机带宽也不确定. 模糊控制器根据这些变化因素和由此引起的 *DMR* 变化来调节任务的服务器速率.

图 5 为控制结果,即 *DMR* 随时间的变化情况. 可见,由于服务器速率不断的适应系统可供给的处理机带宽和任务对资源需求的动态变化,再加上运行等级控制的作用,*DMR* 总是被控制在期望值附近. 模拟研究的结果表明,使用面向软实时应用的自适应调度方法,可在一定程度上满足调度对象的实时性能需求.

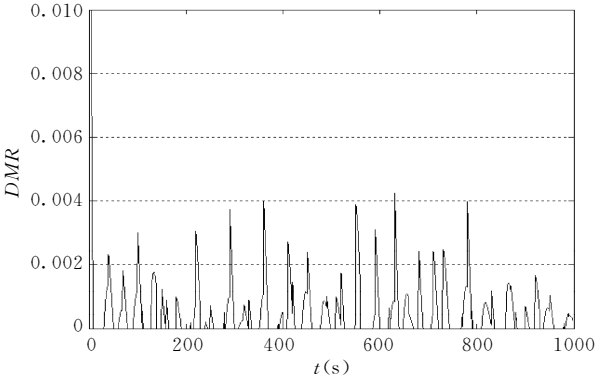

图 5 *DMR* 控制结果

6 结 论

本文针对开放式实时系统对自适应实时调度方法的迫切需求,提出了相应的解决方案. 对硬实时应用的自适应调度问题进行了分析,解决了 CUS 和 TBS 两种类型服务器的速率自适应问题,使硬实时应用的动态加载和异构平台间移植更为方便. 重点提出了面向软实时应用的自适应调度方法. 通过使用模糊控制技术,可在负载动态变化的情况下,自适应地控制软实时应用的 *DMR* 在期望值附近,并可避免服务器过度占用系统计算资源. 实验结果表明该方法是有效的. 开放式实时系统具有了自适应调度功能,从而大大减少了消耗在实时调度参数调整上的代价,方便了实时应用的开发和移植.

参 考 文 献

1 Lu C., Stankovic J. A., Tao G., Son S. H.. Design and evaluation of a feedback control EDF scheduling algorithm. In: Proceedings of the IEEE Real Time Systems Symposium, Phoenix, 1999, 56～67

2 Zou Yong, Wang Qing, Li Ming-Shu. The research and implementing of real-time support of linux kernel. Journal of Computer Reseach & Development, 2002, 39(4): 466～472(in Chinese)
(邹 勇,王 青,李明树. Linux 内核的实时支持的研究与实现. 计算机研究与发展,2002, 39(4): 466～472)

3 Stankovic J. A., Lu Chen-Yang, Son S. H., Tao Gang. The

case for feedback control real-time scheduling. In: Proceedings of the 11th Euromicro Conference on Real Time Systems, York, UK, 1999, 11～20

4 Zou Yong, Li Ming-Shu, Wang Qing. The analysis for scheduling theory and approach of open real-time system. Journal of Software, 2003, 14(1): 83～90(in Chinese)
(邹　勇, 李明树, 王　青. 开放式实时系统的调度理论与方法分析. 软件学报, 2003, 14(1): 83～90)

5 Liu J. W. S.. Real-Time Systems. Upper Saddle River: Prentice Hall, 2000

6 Lu C., Stankovic J. A., Abdelzaher T. F., Tao G., Son S. H., Marley M.. Performance specifications and metrics for adaptive real-time systems. In: Proceedings of the 21st IEEE Real-Time Systems Symposium, Orlando, 2000, 13～24

7 Lipari Giuseppe, Carpenter John, Baruah Sanjoy. A framework for achieving inter-application isolation in multiprogrammed, hard real-time environments. In: Proceedings of the 21st IEEE Real-Time Systems Symposium, Orlando, 2000, 217～226

8 Lipari Giuseppe, Baruah Sanjoy. A hierarchical extension to the constant bandwidth server framework. In: Proceedings of the 7th IEEE Real Time Technology and Applications Symposium, Taipei, Taiwan, 2001, 26～35

9 Deng Z., Liu J. W. S.. Scheduling real-time applications in open environment. In: Proceedings of the 18th IEEE Real-Time Systems Symposium, San Francisco, 1997, 308～319

10 Li Shi-Yong. Fuzzy Control, Neurocontrol and Intelligent Cybernetics. Harbin: Harbin Institute of Technology Press, 1998 (in Chinese)
(李士勇. 模糊控制, 神经控制和智能控制论. 哈尔滨: 哈尔滨工业大学出版社, 1998)

ZOU Yong, born in 1976, Ph. D. candidate. His research interests include real-time systems and operating systems.

HUAI Xiao-Yong, born in 1973, Ph. D.. His research interests include real-time systems, intelligent systems and component software.

LI Ming-Shu, born in 1966, professor, Ph. D. supervisor. His research interests include real-time systems and intelligent software engineering.

Background

Real-time systems are mainly designed to satisfy the timing requirements from the real world applications. A real-time system concerns not only the logic correctness of the computing results, but also the time when the results come out. Nowadays, real-time systems have been widely used in military and civilian industry. However, in many real-time systems, the cases that different kinds of hard real-time, soft real-time and non-real-time applications coexist in one system become more and more popular. This situation makes the application requirements become more complex and causes the proposition of the concepts of open real-time system and adaptive scheduling.