

权重队列实时调度系统的反馈控制

张 军, 邹益仁

(中国科学院自动化研究所综合自动化工程研究中心, 北京 100080)

摘要: 针对工作在不可预测环境下具有不确定性任务特征的实时调度系统的服务区分问题, 利用队列理论对权重队列实时调度系统进行多变量控制建模, 并结合模型预测控制方法将控制问题转化为受限多变量优化问题, 进而基于标准的最小二乘求解器设计预测控制调度框架以及算法。通过调整任务类的权重因子来改变任务类的计算资源分配, 以实现为调度任务类提供相对延迟时间比保证。仿真试验结果表明, 所设计方法在任务到达速率和执行时间变化且不能预知的情况下, 仍然能为系统提供良好的相对延迟时间比控制。

关键词: 实时调度; 延迟; 预测控制; 权重; 服务区分

Feedback Control of Weighted Queueing Real-time Schedule

ZHANG Jun, ZOU Yiren

(Engineering Research Center of Integrated Automation Technology, Institute of Automation, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 To provide service differentiation guarantee for real-time schedule system with uncertain task set characteristic, queueing theory is applied to construct the multivariable control model of a weighted queueing schedule system. Based on model predictive control approach, a predictive control scheduling framework and algorithm are designed which can keep relative delay ratio on target by dynamically adjusting computation resource allocation of scheduled tasks. The simulations results demonstrate that the predictive control schedule can achieve desired relative delay ratio specification even when task execution times and arrival rate are variable and unpredictable.

【Key words】 Real-time schedule; Delay; Predictive control; Weight; Service differentiation

目前, 传统控制理论正被引入到如实时调度、Web 服务器、多媒体数据流等这样的工作在不可预测环境中且具有不确定任务特征的复杂软件应用的 QoS 性能控制研究中, 并且为这一领域的研究建立了良好的理论基础。文献[1]提出了一个基于反馈控制理论的实时调度框架及相应的调度算法, 设计者可以由已建立的分析型方法系统地进行实时系统的设计, 使得在不可预测环境下实时系统的暂态性能和稳态性能都能得到保证。队列服务模型在文献[2, 3]中被用于对服务器的响应时间进行预测, 并与简单的闭环反馈控制回路相结合, 为任务响应时间提供了绝对与相对延迟担保。文献[4]通过采用多输入多输出控制理论对 Apache 服务器进行辨识建模以及相应的控制设计, 并由控制器对服务器可调节参数的在线调整实现对服务器的多个相互关联的资源利用指标的控制。

在复杂软件应用的性能控制研究中, 服务区分是其中一个重要的部分。例如, 在线的电子商务服务器系统为交易类请求所提供的服务质量(延迟、内容等)当然应当需要优于简单的浏览请求。在软件应用中, 不同的服务质量意味着需要为服务分配不同的计算资源。但为取得所期待的服务质量要求而进行的资源分配需要预先掌握系统负载情况及任务执行时特征等信息。因此对于工作在不可预测环境中且具有不确定任务特征的复杂计算机应用, 就需要一种动态的资源分配框架与算法以实现计算资源在运行时的灵活分配, 从而取得所设定的服务区分要求。

本文以实时权重调度队列为对象进行控制建模, 并结合模型预测控制的方法, 在运行时动态优化调整调度任务的权重值, 从而实现了对工作在不可预测环境中且具有不确定任

务特征的实时调度系统的服务延迟时间区分。

1 权重队列调度的控制模型

1.1 控制回路与控制变量描述

假定调度任务集被划分为 n 个类别, 各个类别 $class_i$ 的每个具体任务实例的到达时间间隔与任务实例执行时间长度在运行时都是变化和不可预知的。在本文的实时调度服务区

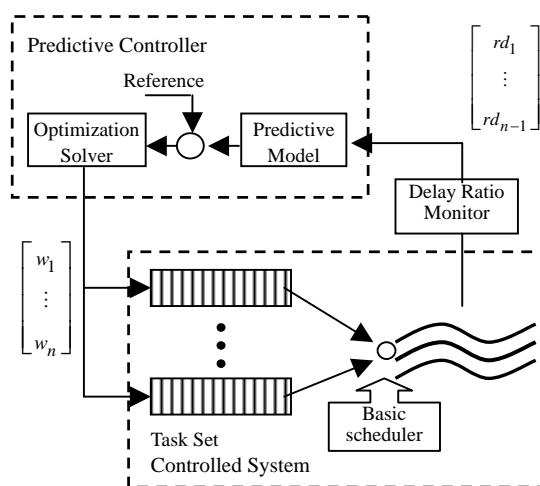


图 1 权重队列调度控制回路

基金项目: 国家“863”计划基金资助项目(2003AA412010)

作者简介: 张 军(1974 -), 男, 博士生, 主研方向: 实时调度, 一体化控制与调度; 邹益仁, 研究员、博导

收稿日期: 2006-03-27 E-mail: jun.zhang@ia.ac.cn

分研究中, 选择的服务质量指标是调度任务的平均统计延迟时间, 即调度任务由其到达系统到其执行完成离开系统之间的时间间隔, 这一指标在如服务器应用中能表征用户对服务质量的直接认定。定义第 i 个任务类的平均统计延迟时间为 ad_i , 系统的受控变量则选取为相邻类别任务的相对延迟时间比 $rd_i = ad_i / ad_{i+1}$ 。实时调度中引入权重因子的目的是为了实现对多个调度任务类进行比例资源分配, 即使得每类任务所能分配到的计算资源比例与其权重因子比例相符合。各个类别 $class_i$ 对应的权重因子定义为 w_i , 因为权重因子直接决定调度任务所能分配到的计算资源数量进而有效影响到系统受控变量 rd_i , 所以选择权重因子 w_i 的变化量 Δw_i 作为系统的控制输入, 系统的控制回路如图 1。

1.2 权重队列调度的控制模型

权重队列调度的相对延迟时间比与所分配的权重系数之间的控制关系是复杂及非线性的, 借助队列理论的帮助来对其进行线性化的控制建模。首先将权重队列调度系统中各个类别的任务的运行过程用队列理论中的 M/M/1 服务模型来近似, 即每一类别任务 $class_i$ 有着其相应的统计到达率 λ_i 和服务率 μ_i , 则根据队列理论, 该类别任务的平均统计延迟时间为

$$ad_i = 1/(\mu_i - \lambda_i) \quad (1)$$

相邻类别任务的延迟时间比为

$$rd_i = (\mu_{i+1} - \lambda_{i+1})/(\mu_i - \lambda_i) \quad (2)$$

任务的服务率被认为与任务所分配的计算资源成比例关系, 通过调整任务的权重因子, 可以改变分配给调度任务的计算资源进而改变任务的服务率。权重队列调度的基本调度策略选取的是 SFQ(start-time fair queueing) 调度策略^[5], 这一调度策略可以保证按照任务的权重因子对系统计算资源进行比例分配。通过选取任务类所分配到的计算资源占资源总量的百分比值作为该任务类的权重因子, 各个类别 $class_i$ 的服务率可以写为 $w_i \mu$, 其中 μ 是单位百分比资源的服务率, 具体值可以通过相应试验获得。这样式(2)可写成

$$rd_i = (w_{i+1} \mu - \lambda_{i+1}) / (w_i \mu - \lambda_i) \quad (3)$$

在任务类到达率给定的情况下, 根据式(3)并且线性化系统, 可得

$$\Delta rd_i = (\partial rd_i / \partial w_i) \Delta w_i + (\partial rd_i / \partial w_{i+1}) \Delta w_{i+1} \quad (4)$$

综合上述讨论, 得到系统的控制模型为

$$rd(k) = rd(k-1) + E \Delta w(k-1) \quad (5)$$

$$\Delta w(k) = w(k) - w(k-1) \quad (6)$$

$$rd(k) = \begin{bmatrix} rd_1(k) \\ \vdots \\ rd_{n-1}(k) \end{bmatrix}, \quad \Delta w(k) = \begin{bmatrix} \Delta w_1(k) \\ \vdots \\ \Delta w_n(k) \end{bmatrix}, \quad w(k) = \begin{bmatrix} w_1(k) \\ \vdots \\ w_n(k) \end{bmatrix},$$

$$E = \begin{bmatrix} \frac{\partial rd_1}{\partial w_1} & \frac{\partial rd_1}{\partial w_2} & 0 & 0 & \dots & 0 & 0 \\ 0 & \frac{\partial rd_2}{\partial w_2} & \frac{\partial rd_2}{\partial w_3} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{\partial rd_{n-1}}{\partial w_{n-1}} & \frac{\partial rd_{n-1}}{\partial w_n} \end{bmatrix}$$

其中 k 表示为第 k 个采样时刻, 相关偏导数值可通过测量该时刻各任务类的平均到达率值并通过计算得到。

2 预测控制器的设计

2.1 预测控制描述

模型预测控制^[6]的基本过程为: 控制器使用对象模型预测对象在预测域 P 的时间段内的输出; 根据当前时刻的性能

反馈求解受限优化问题以得到能最小化控制评价函数的控制输入策略, 该输入策略包括在将来的控制域 M 的时间段里的控制输入, 但只采用控制策略的第 1 个元素作为系统的控制输入。

控制器评价函数取为

$$\min_{\Delta w(k) \dots \Delta w(k+M-1)} J(k) = \sum_{i=1}^P \square rd(k+i) - c(k+i) \square^2 \quad (7)$$

评价函数表征的是系统的受控输出 $rd(k+i)$ 与输出参考策略 $c(k+i)$ 之间的偏差。输出参考策略 $c(k+i)$ 指系统受控输出由当前值改变为系统受控输出设定值 $rd_s = [rd_{1,s} \dots rd_{n-1,s}]^T$ 的理想轨迹, 本文选择的是常用的一阶指数策略:

$$c(k+i) = \alpha^i rd(k) + (1 - \alpha^i) rd_s \quad (8)$$

其中 α 是设计参数, α 越小, 参考轨迹就能越快达到设定值 rd_s , 即取得所期待的相对延迟时间比。

因为选取任务分配计算资源的百分比值作为权重因子, 所以相关的限制条件为每个任务分配一定大小范围内的百分比计算资源, 且总的计算资源固定为 100%, 写成表达式形式即

$$w_{\min,i} \leq w_i(k) \leq w_{\max,i} \quad (9)$$

$$\Delta w_1(k) + \dots + \Delta w_n(k) = 0 \quad (10)$$

根据上述分析, 可知控制器需要根据式(5)、式(6)描述的系统控制模型来最小化评价函数式(7), 限制条件为式(9)、式(10)。这一预测控制描述可以转化为标准的受限最小二乘求解问题。

2.2 最小二乘求解

标准的最小二乘问题的求解式为

$$\min_x \square Cx - d \square^2 \quad (11)$$

限制条件 $A \cdot x \leq b$, $Aeq \cdot x = beq$ 。

接下来将控制器评价函数式(7)写成向量形式, 为此, 先定义:

$$y(k) = \begin{bmatrix} rd(k+1) \\ \vdots \\ rd(k+P) \end{bmatrix}, \quad r(k) = \begin{bmatrix} c(k+1) \\ \vdots \\ c(k+P) \end{bmatrix}$$

这样, 评价函数式(7)可写为

$$\min_{\Delta w(k) \dots \Delta w(k+M-1)} J(k) = \square y(k) - r(k) \square^2 \quad (12)$$

由对象模型表达式(5), 可以得到

$$\begin{cases} rd(k+1) = rd(k) + E \Delta w(k) \\ \vdots \\ rd(k+P) = rd(k) + E \left(\sum_{j=0}^{M-1} \Delta w(k+j) + (P-M) \Delta w(k+M-1) \right) \end{cases} \quad (13)$$

由表达式(13)可以将 $y(k)$ 改写为

$$y(k) = y'(k) + G \cdot v(k) \quad (14)$$

其中:

$$y'(k) = \begin{bmatrix} rd(k) \\ \vdots \\ rd(k) \end{bmatrix}, \quad v(k) = \begin{bmatrix} \Delta w(k) \\ \vdots \\ \Delta w(k+M-1) \end{bmatrix}, \quad G = \begin{bmatrix} E & 0 & 0 & \dots & 0 \\ E & E & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E & E & E & \dots & E \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E & E & E & \dots & E + E \\ E & E & E & \dots & E + (P-M)E \end{bmatrix}$$

将表达式(14)代入评价函数式(12), 可得

$$\min_{v(k)} J(k) = \square Gv(k) - (r(k) - y'(k)) \square^2 \quad (15)$$

以 $v(k)$ 为优化变量 x , 本文的控制器评价函数式(7)可以与标准受限最小二乘式(11)对应为

$C \sim G, d \sim r(k) - y(k)$

系统限制条件(9)根据式(6)可以写为

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \end{bmatrix} \begin{bmatrix} w(k) \\ \vdots \\ w(k+M-1) \end{bmatrix} \leq \begin{bmatrix} W_{\max} \\ \vdots \\ W_{\max} \\ -W_{\min} \\ \vdots \\ -W_{\min} \end{bmatrix} \quad (16)$$

根据式(6), 上式可写为

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \\ -1 & 0 & \dots & 0 \\ -1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & -1 \end{bmatrix} v(k) \leq \begin{bmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} w(k-1) + \begin{bmatrix} W_{\max} \\ \vdots \\ W_{\max} \\ -W_{\min} \\ \vdots \\ -W_{\min} \end{bmatrix} \quad (17)$$

表达式(17)与标准最小二乘的限制条件 $A \cdot x \leq b$ 相对应。

系统限制条件(10)可写为

$$\begin{bmatrix} l & 0 & \dots & 0 \\ 0 & l & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & l \end{bmatrix} v(k) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, l = [1 \dots 1] \quad (18)$$

表达式(18)与标准最小二乘的限制条件 $A_{eq} \cdot x = b_{eq}$ 相对应。

这样, 完成了将预测控制问题转化为标准受限最小二乘求解问题的过程。且在第 k 个采样时刻, $rd(k)$ 、 $\Delta w(k-1)$ 、 $w(k-1)$ 都为已知, 从而可以采用标准的最小二乘求解器来实现预测控制器求解优化变量 $v(k)$ 并得到该采样时刻的控制输入 $\Delta w(k)$ 。

3 性能仿真试验

3.1 试验建立

仿真试验系统由基本 SFQ 权重调度队列和预测控制器两部分组成。基本权重队列调度是采用 C/C++ 实现的伪实时调度系统, 包括延迟时间量测器、到达率量测器以及权重因子调整器, 为避免延迟时间测量的随机噪声干扰而引起不必要的权重调整, 在将测量信号反馈至控制器的回路上引入一个指数加权平均(EWMA)滤波器。预测控制器基于 Matlab 中的标准最小二乘求解器 lsqin 来实现。调度系统与预测控制器通过 Matlab 引擎进行交互。

试验任务集选择的是非周期任务集, 任务集由两个类别的任务负载组成, 任务实例的到达时间间隔以及执行时间长度在运行过程中都是变化和不确定性的, 相关的初始任务参数设置为: 任务类 1, 2 的任务实例到达时间间隔都为在 [1, 50]ms 间均匀随机分布; 任务类 1, 2 的任务实例的执行时间长度都在 [1, 10]ms 间均匀随机分布; 任务类 1 的初始权重因子为 0.70, 即初始分配给该任务类 70% 的计算资源, 任务类 2 的初始权重因子相应的为 0.30。

控制器参数设定的主要考虑因素为在模型不精确及任务执行特征带有不确定性时预测控制器的强健性与快速性的均衡。预测域 P 与控制域 M 的选择对控制的稳定性和快速性有较大影响, 具体控制器参数值的选择可以根据所要求的实际控制策略来得到。在试验中选择 $P=4, M=2$ 。采样时间的选择根据所测量的输出是任务的平均统计延迟时间, 因此需要保证在采样时间内达到一定数量任务实例数, 以准确反映测量的实际变化, 试验中选择的采样时间为 2s。反馈回路上 EWMA 滤波器的参数设置为 0.4, 输出参考策略 $c(k)$ 的设计参数 $\alpha = 0.5$ 。

3.2 仿真试验 1: 变化的到达率

试验 1 模拟的是当运行时任务类的到达率发生变化时系统的响应性能。任务类 1、2 的初始到达时间间隔设定为在 [1, 50]ms 间的均匀随机分布, 在运行至第 100 个采样时刻时, 任务类 1 的到达时间间隔变化为在 [1, 15]ms 间的均匀随机分布。在整个试验过程中, 系统任务类的相对延迟时间比的设定值为 1.00, 任务类 1、2 的执行时间长度为 [1, 10]ms 间的均匀随机分布。试验结果如图 2 所示。由图 2 可见, 对于初始时参数分布相同的任务类 1、2, 任务类 1 因为其较大的起始权重因子而分配到更多的计算资源, 使得两类任务的相对延迟时间比小于设定值 1.00。控制器因此开始增加任务 2 的权重因子并减小任务 1 的权重因子以改变计算资源的分配, 最终使得任务 1 与任务 2 的权重因子都稳定在 0.50 附近, 即平均分配计算资源, 这时任务类的相对延迟时间比也稳定在设定值 1.00 附近, 控制器良好地执行了所设计的控制作用。在第 100 个采样时刻时, 任务类 1 的到达时间间隔发生变化造成任务 1 的负载加大, 在当前的计算资源分配状况下导致任务 1 的延迟时间增大, 任务类的延迟时间比也相应增大超出设定值。控制器根据此时的相对延迟时间比开始增大任务类 1 的权重因子以增大分配给任务类 1 的资源, 并同时相应减少任务类 2 的权重因子, 任务类 1、2 的权重因子被调整为相应的 0.68 与 0.32, 这时的系统输出又重新稳定在设定值 1.00 的附近。可见系统在任务负载发生变化时的控制效果是比较理想的, 达到了设计的目的。

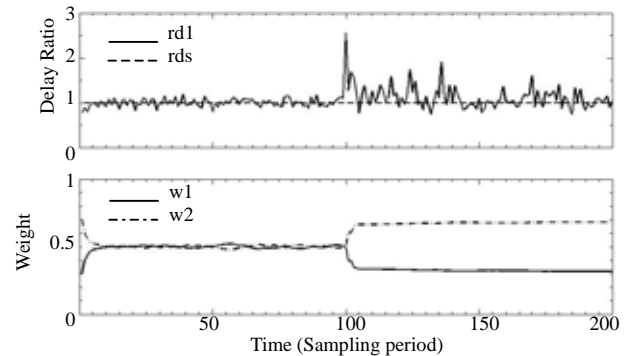


图 2 仿真试验 1: 变化的到达率

3.3 仿真试验 2: 变化的执行时间

在试验 2, 实例的执行时间分布是变化和不确定性的, 初始时任务类 1、2 的执行时间长度都为 [1, 10]ms 间的均匀随机分布, 运行至第 100 个采样时刻时, 任务类 1 的执行时间长度变化为 [1, 20]ms 间的均匀分布。在整个试验过程中, 系统任务类的相对延迟时间比的设定值 1.00, 任务类 1、2 的到达时间间隔都为在 [1, 50]ms 间均匀随机分布。由试验结果图 3 中可以看到, 前 100 个采样时刻内系统的情况与试验 1 类似, 任务类 1、2 的平均分配计算资源, 系统输出稳定在设定值 1.00。第 100 采样时刻, 任务类 1 的执行时间分布的变化使得其输出延迟时间增大, 任务类的相对延迟时间比也相应增大偏离设定值。此时控制器仍然很好地发挥了设计的控制作用, 通过将任务类 1 的权重因子增大至 0.87 以分配更多的计算资源, 并同时减小任务类 2 的权重因子至 0.13 附近, 重新使得任务类的相对延迟时间比稳定在设定值。由此可以观察到系统对于变化的任务执行时间的控制强健性。

(下转第 12 页)