

## 实时系统中弹性调度策略

杨志邦<sup>1</sup>, 徐 成<sup>1</sup>, 周 旭<sup>2\*</sup>, 朱雪庆<sup>1</sup>

(1. 湖南大学 信息科学与工程学院, 长沙 410082; 2. 嘉兴学院 数理与信息工程学院, 浙江 嘉兴 314001)

(\* 通信作者电子邮箱 yangzhibang2006@126.com)

**摘 要:**弹性调度面向负载可变的实时系统,通过动态调整任务属性以满足系统的灵活性要求,是一种高效的任务调度策略。针对弹性调度研究中的成果及问题,概述了弹性调度的研究背景,从任务模型、调度模型以及调度算法三个方面对弹性调度的国内外研究进展进行综述,探讨当前研究中存在的问题,并对弹性调度未来研究工作进行分析 and 展望。

**关键词:**实时系统;系统性能;弹性调度;任务模型;调度算法

**中图分类号:** TP302.7 **文献标志码:** A

### Elastic scheduling in real-time systems

YANG Zhi-bang<sup>1</sup>, XU Cheng<sup>1</sup>, ZHOU Xu<sup>2\*</sup>, ZHU Xue-qing<sup>1</sup>

(1. College of Information Science and Engineering, Hunan University, Changsha Hunan 410082, China;

2. College of Mathematics Physics and Information Engineering, Jiaxing University, Jiaxing Zhejiang 314001, China)

**Abstract:** Elastic scheduling is designed for real-time systems with variable load. It adjusts the task attributes dynamically to meet the flexibility requirements of system, and it is an effective task scheduling strategy. Concerning the research results and problems of elastic scheduling, an overview of elastic scheduling was given out, and the research progress of the elastic periodic task model, scheduling model and the elastic scheduling algorithm were analyzed. The existing problems in research were investigated, and possible research areas in future were suggested.

**Key words:** real-time system; system performance; elasticity scheduling; task model; scheduling algorithm

## 0 引言

实时系统对任务的响应时间有严格要求。传统的实时系统多为硬实时系统,任务属性在运行过程中不发生改变。然而,随着实时系统在生活中越来越普遍地得到应用,大量的应用需要系统在用户的输入或环境改变等突发情况下仍然能够正常工作。如流媒体服务器,它的任务量根据用户请求而发生改变,需要能够在不同环境下为用户提供较好的音视频服务。这样的实时系统需要具有更多的灵活性,能够提供某种机制,以保证系统在过载的状态下能发挥最佳性能水平,在轻量负载的情况下合理利用系统的资源<sup>[1]</sup>。

在具有一定灵活性的实时多任务系统中,设计高效的实时任务调度方案是保证任务顺利完成的关键。实时系统对于任务临时过载问题,通常有两种处理方案<sup>[2]</sup>:一类是直接丢弃一些不重要的任务,使得系统可正常运行,如非精确计算;另一类是减少任务的资源利用率,使整个系统负载在可承受范围内,也就是弹性调度。对于大多数的实时系统,采用弹性调度降低任务运行质量比非精确计算中直接丢弃任务要好<sup>[3]</sup>,从而成为研究的重点<sup>[2-16]</sup>。

弹性调度最初的研究对象是实时弹性周期任务,任务周期可以在一定时间范围内连续变化,并且在该变化范围内,任务执行结果都可以被系统接受。随着弹性调度应用越来越广泛,弹性调度的研究对象在不断扩展,但是仍遵循其基本条

件:任务可以在一定范围内对任务的某属性进行调整,并且在该范围内的任务执行结果都是有效的<sup>[4]</sup>。

弹性调度算法通过周期调整算法改变任务周期,从而选择合适可接受任务周期,以实现系统整体资源的合理利用或满足系统的顺利运行<sup>[5]</sup>。随着弹性调度发展,弹性调度算法不仅仅局限于对弹性周期进行调整,也包括对弹性任务的其他属性进行调整<sup>[6]</sup>。

## 1 弹性任务模型

弹性调度最初由 Buttazzo 等<sup>[7]</sup>提出,其对弹性任务定义如下。

**定义 1** 弹性任务  $\Gamma_i$  可由五元组表示:  $\Gamma_i = (C_i, T_i, T_{i\max}, T_{i\min}, e_i)$ 。 $C_i$  是任务  $\Gamma_i$  的执行时间,在此定义中,弹性任务的执行时间是固定的,即任务最坏执行时间(Worst-Case Execution Time, WCET)。 $T_i$  是任务  $\Gamma_i$  其当前任务周期,可在区间  $[T_{i\min}, T_{i\max}]$  内连续变化,  $T_{i\min}$  是任务  $\Gamma_i$  的最小周期,  $T_{i\max}$  是任务  $\Gamma_i$  的最大周期。超过任务最大周期或者任务最小周期,系统性能将不被接受。 $e_i \geq 0$  是任务  $\Gamma_i$  的弹性系数,用来表明改变此弹性任务周期的难易程度,弹性系数越小,则改变任务周期越难。在某种程度上,  $e_i$  也表示了任务的优先级,  $e_i$  越小的任务,其压缩的空间有限,运行属性改变不大,执行优先级较高;而  $e_i$  越大的任务,在系统资源不够时为首先被压缩的任务,执行优先级就越低。

收稿日期: 2011-07-29; 修回日期: 2011-09-22。

**基金项目:** 国家自然科学基金资助项目(60973030); 湖南省科研条件创新专项(2010TT1002); 嘉兴市科技计划项目(2011AY1003)。

**作者简介:** 杨志邦(1984-),男,湖南邵阳人,博士研究生,CCF会员,主要研究方向:嵌入式系统、任务调度; 徐成(1962-),男,湖北蕲春人,教授,博士,主要研究方向:嵌入式系统; 周旭(1983-),女,江苏宿迁人,讲师,硕士,主要研究方向:并行计算、任务调度; 朱雪庆(1985-),女,湖南张家界人,硕士研究生,主要研究方向:任务调度。

弹性任务中由于任务的周期可以改变,从而其占用系统资源的比例就会发生变化,这就引入了资源利用率的概率。

**定义2** 任务  $\Gamma_i$  资源利用率  $U_i = c_i / T_i$ , 任务  $\Gamma_i$  最小资源利用率  $U_{i\min} = c_i / T_{i\max}$ , 最大资源利用率  $U_{i\max} = c_i / T_{i\min}$ ,  $U_i$  可在  $[U_{i\min}, U_{i\max}]$  范围内连续变化。系统资源利用率  $U$  是系统中所有实时任务的资源利用率之和, 即  $U = \sum U_i$ 。任务利用率额定值  $U_d$ , 表示系统所能接受的最大系统资源利用率, 一般满足  $U_d \geq U$ 。

在不同应用环境中, 所考虑的因素不同, 弹性任务的定义及形式化描述也有所差别。定义1给出了弹性任务的基本定义, 其包含了弹性任务最基本的特性, 也即弹性任务的某个执行属性是可变, 该属性具有相应的弹性系数, 表示改变该属性的难易程度。后续研究者们基本特性基础上, 对弹性任务进行了相应的扩展。

文献[8-9]放宽基本弹性任务的执行时间, 认为任务的最坏执行时间在运行过程中可以动态变化, 存在一个上边界, 其任务模型定义为一个四元组  $\Gamma_i = (C_i^w, T_0, T_{i\max}, e_i)$ 。其中  $C_i^w$  为  $\Gamma_i$  执行时间的上限, 随着任务周期的变化而变化;  $T_0$  是任务的最小周期, 也是正常运行时的周期; 另外两个参数同定义1。

文献[10]对任务截止期限限制进行考虑, 研究者假定任务执行时间不变, 但存在一个小于等于任务周期的任务截止期, 该截止期在任务周期结束前到来。定义弹性任务的六元组为:  $\Gamma_i = (C_i, D_i, T_i, T_{i\min}, T_{i\max}, e_i)$ 。其中:  $C_i$  是任务  $\Gamma_i$  最坏执行时间;  $D_i$  是任务  $\Gamma_i$  的截止期,  $D_i \leq T_i$ 。

文献[11]认为任务截止期可变, 为保证任务在其截止期到来前完成, 直接调整任务截止期, 而任务周期不变, 任务截止期在一定范围内变化, 其弹性任务模型定义为一个六元组:  $\Gamma_i = (C_i, D_i, T_i, D_{i\min}, D_{i\max}, e_i)$ 。其中:  $D_{i\min}$  为实时任务最小截止期,  $D_{i\max}$  为最大截止期,  $D_i$  在  $[D_{i\min}, D_{i\max}]$  范围内连续可变; 弹性系数  $e_i$  表示任务截止期的可变程度。

文献[12]将弹性任务调度应用于控制系统设计, 考虑到控制系统中任务执行时间存在较大变化, 任务模型定义为:  $\Gamma_i = (C_i, T_{\text{worst}}, T_i, T_{i\min}, T_{i\max}, e_i)$ 。其中  $C_i$  是任务的实际执行时间, 是动态变化的,  $T_{\text{worst}}$  是任务最坏执行时间,  $T_{\text{worst}} \geq C_i$ 。

文献[13]扩展了线性弹性任务模型, 提出了一个非线性弹性任务模型, 任务执行时间预先确定, 即采用最坏任务执行时间, 对任务资源利用率直接进行调整。其弹性任务  $\Gamma_i$  定义为一个三元组:  $\Gamma_i = (U_0, U_{i\min}, e_i)$ 。其中  $U_0$  是弹性任务  $\Gamma_i$  的最大任务资源利用率;  $U_{i\min}$  是弹性任务  $\Gamma_i$  的最小资源利用率;  $e_i \geq 0$  是任务  $\Gamma_i$  的弹性系数, 依赖于每个任务资源利用率。

文献[14]将弹性任务调度应用于流媒体服务中, 通过调整流媒体服务质量等级来调整系统负载, 其弹性任务定义为:  $\Gamma_i = (C_i, e_i, Q_{i\min}, Q_{i\max})$ 。其中:  $C_i$  表示任务  $\Gamma_i$  的关键性,  $e_i$  表示任务  $\Gamma_i$  的弹性系数,  $Q_{i\min}$  表示流媒体任务  $\Gamma_i$  的最低服务质量需求,  $Q_{i\max}$  表示任务  $\Gamma_i$  最大服务质量需求。服务质量  $Q_i$  在  $[Q_{i\min}, Q_{i\max}]$  范围内变化, 都是为任务请求者所接受的。

文献[15]中流媒体服务质量等级为一系列离散值, 即弹性任务的周期为几个离散值。其弹性任务定义为:  $\Gamma_i = (n, P_{i1}, P_{i2}, \dots, P_{in}, W_i)$ 。其中:  $n$  代表任务拥有的任务周期数,  $P_{ij}$  代表弹性任务  $\Gamma_i$  的第  $j$  级作业周期,  $j \in (1, n)$ , 按由大到小顺序排列。每个周期对应任务  $\Gamma_i$  的一个服务质量级

别, 较高服务质量对应较小周期。  $W_i$  代表任务关键性。弹性任务模型不包括任务执行时间, 通过采样任务的作业次数作为调度策略参数。

文献[16]将弹性任务调度应用于数控技术中, 考虑混合任务集的调度。其弹性任务定义为:  $\Gamma_i = (T_{i\min}, T_{i\max}, T_i, P_i)$ 。该定义中前面的三个参数表示弹性任务  $\Gamma_i$  的周期及其周期变化范围,  $P_i$  为弹性任务  $\Gamma_i$  的优先级。

随着实时系统的广泛应用, 弹性任务在实际生活中越来越普及, 在不同领域中弹性任务定义也各有不同。在实际的应用中, 可以根据系统中什么因素可以变化, 从而将其定义为弹性变量, 从而定义弹性任务模型。总体说来, 弹性调度任务模型越来越有其专用性, 弹性任务定义越来越广泛, 更接近实际任务属性。

## 2 弹性调度模型

弹性任务调度模型主要研究实时任务周期可以在一定范围内连续变化的实时系统, 任务在这一周期范围内的执行结果都是可用的<sup>[17]</sup>。研究者们针对弹性任务调度的特性, 提出了明确调整任务参数的方案, 模型简单, 在实时系统中得到较广泛的应用。

弹性任务调度的基础模型一般采用两层调度算法<sup>[18-19]</sup>, 如图1所示。

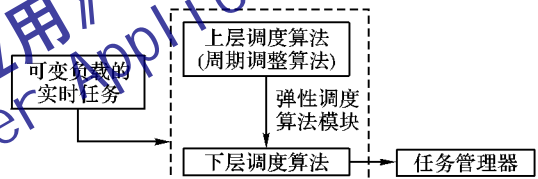


图1 弹性调度基础模型

图1中间虚线框部分为弹性任务调度算法主体部分, 分为两层调度算法。上层调度算法其主要任务是根据当前实时系统负载情况调整任务周期属性, 使其满足下层调度算法的可调度条件, 称之为周期调整算法; 下层调度算法一般采用经典实时调度算法, 较常用的为单调速率 (Rate Monotonic, RM) 算法和最早截止期优先 (Earliest Deadline First, EDF) 算法, 根据调整后的任务周期或资源利用率等任务属性, 确定任务优先级并进行调度。

基于弹性调度模型的上层调度算法对不同的应用环境其调整弹性任务周期等属性的控制性能指标不同, 其调整策略也有较大差别<sup>[7-11]</sup>。下层调度算法一般采用经典 RM 或者 EDF 算法。RM 算法是基于静态优先级的调度算法, EDF 算法是动态优先级调度算法, 具体选择哪种算法将根据实际应用确定。在许多研究中, 下层调度算法均采用 EDF 算法<sup>[8-10, 13]</sup>。

一些研究者将弹性调度模型和反馈模型结合起来, 在任务调度过程中使用了动态反馈弹性调度模型<sup>[12, 14-16, 20-21]</sup>, 虽然研究者们都是针对具体应用环境设计的动态反馈弹性调度模型, 各种模型都有其各自的特点, 但是这些调度模型也有其共性, 即任务管理器将任务执行情况等信息动态反馈给弹性调度模型, 指导弹性任务调度算法进行周期调整。如图2。

动态反馈弹性调度模型与弹性调度模型的弹性任务调度算法部分基本一致, 均采用分层调度算法, 上层算法通过对任务周期属性进行调整, 改变影响任务资源利用率的参数, 进而调整系统负载, 然后将调整属性后的任务交给下层经典调度

算法进行调度并执行。这两个模型的不同之处在于:动态反馈弹性任务调度模型动态地反馈任务执行情况给上层调度算法,将任务执行情况作为调整任务属性的指导参数,从而更好地了解系统运行情况,尽可能好地控制系统整体性能。

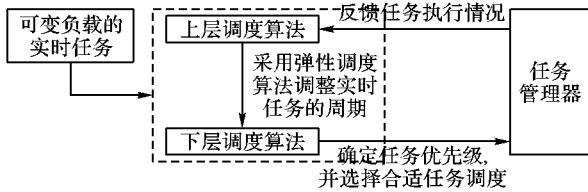


图2 动态反馈弹性任务调度模型

动态反馈弹性调度模型针对不同应用其反馈指标不同。文献[12]反馈系统负载情况以及采样周期内的最大执行时间,上层调度算法根据这些反馈信息对任务周期进行调整。文献[14]中反馈机制提供实时任务运行情况以及实时任务动态插入或退出等信息,根据系统中实际负载情况判断是否调用周期调整算法,如果需要进行调整,则根据任务关键性对任务服务质量进行调整,否则正常运行。文献[15]通过反馈采样周期内的任务实例执行总数、作业丢失率以及系统资源占用率来了解系统运行情况,调整任务资源利用率以降低系统的作业丢失率。文献[16]采用闭环反馈将处理器实际负载情况以及实际任务丢失率反馈给弹性调度算法,进而调整任务周期以保证继续运行。

### 3 弹性任务调度算法

弹性调度模型中最重要的就是任务调度算法,其采用分层调度的方法。上层调度算法的主要工作是对影响任务资源利用率的参数进行调整,以保证系统总资源利用率满足经典调度算法的可调度条件;下层调度算法采用经典实时调度算法对任务分配优先级、调度次序等进行调度执行。经典实时调度算法的研究已经很成熟,其可调度条件非常明确。因此,弹性任务调度的重点和关键是对上层算法进行研究,也即通过合理调度以调整任务资源利用率。

弹性任务调度算法的关键问题是上层算法对影响实时任务资源利用率的参数进行调整,根据任务资源利用率定义  $U_i = e_i / T_i$ , 影响任务资源利用率的参数主要是任务执行时间和任务周期。假定实时任务执行时间在系统运行过程中保持不变,则弹性任务调度算法可看作是求解满足系统可调度性条件的任务集合中各个任务的周期选择问题。

#### 3.1 调度算法基础理论

弹簧模型的调度算法最先由 Buttazzo 等<sup>[7]</sup>提出,其来源于弹簧模型,系统中任务集合可看作是若干弹簧连接在一起,共同作用的弹簧系统。当弹簧系统受到外力  $F$ , 其中每个弹簧均压缩来承担一部分外力,直到外力  $F$  被各个弹簧所分担,系统达到稳定状态,各个弹簧根据自身弹性系数  $k_i$  对弹簧长度  $L_i$  进行调整。当弹性压缩至其最短长度,也就是该弹簧所能承担的最大压力。当外力  $F$  消失时,每个弹簧再次对各自长度  $L_i$  进行调整,以达到另一个稳定状态。如图3。

弹性任务调度算法将系统中每个任务比作弹簧系统中的一个弹簧,任务资源利用率  $U_i$  为弹簧长度  $L_i$ , 每个任务的重要性  $e_i$  对应于弹簧的弹性系数  $K_i$ 。当系统资源利用率  $U$  超过额定值时,即弹簧系统总长度超过  $L_{\max}$  时,系统中的资源请求超过系统的负载能力,导致系统不能被经典调度算法所调度。此时,需要对弹簧施加外力,即保证整个弹簧系统的总长度不

超过额定值。即对每个任务按照其任务重要性  $e_i$  对任务资源利用率  $U_i$  进行压缩,将任务周期延长,使得系统总资源利用率低于额定值。同理,当系统总资源利用率  $U$  低于其额定值时,即弹簧处于有外力作用下,则对各个任务资源利用率按其重要性进行调整,将任务作业周期缩短,直到系统总资源利用率  $U$  接近其额定值或者所有任务都处于其最大资源利用率、最小作业周期,从而保证合理利用系统资源。

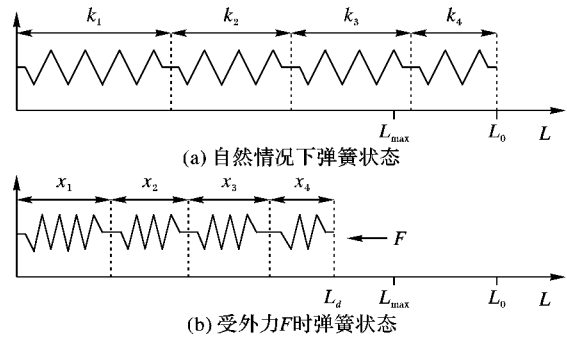


图3 弹簧模型

对于一个基于弹簧模型的实时弹性任务集,通常存在多个可行周期分配方案,不同可行解之间可能会存在系统性能的差异。基于下层的静态优先级调度算法,可以将求解满足可调度性条件的任务周期问题规约为一个最优化问题。弹性调度进而可表述为在满足系统可调度性约束条件下,通过最优化一个特定形式的控制性能指标函数,以获得系统最优性能。针对不同系统特性的应用环境,不仅其选用的控制性能指标函数存在差异;而且其系统可调度条件的限制约束也各不相同,这取决于弹性任务调度算法中的下层调度算法以及系统资源的其他限制条件;各个任务自身限制条件也视具体情况而定。

#### 3.2 弹性调度相关算法

文献[9]将输入看作一个周期可调的具有弹性特征的任务流,通过在线预测任务执行时间,实现了一个对系统输入进行自适应的任务调度算法,从而降低任务错失率,提高系统资源利用率。但该算法需要增加额外的在线监测机制,这也加大了系统开销。

文献[20]的上层采用贪心算法调整任务周期,该算法的计算复杂度低,但是贪心算法容易造成任务资源利用率发生较大波动,另外任务执行时间在运行过程中保持不变,虽然简化了问题难度,但也在一定程度上造成了系统资源的浪费。

文献[22]在基于弹簧模型的弹性任务调度算法基础上,放宽对任务属性的限制,认为任务作业时间在系统运行过程中动态变化,对现有算法进行改进。文中虽然考虑到实时任务执行时间的动态变化,但没有研究任务截止期对系统性能的影响。

文献[23]在弹簧模型的弹性任务调度算法基础上,结合弹性任务调度框架中资源限制的处理机制,考虑任务执行过程中系统资源受限的情况,以最优化任务采样率为控制性能指标,分别研究了满足 EDF、RM 可调度条件的调度算法,提出了资源受限的弹性任务调度算法。该算法计算过程较复杂,而且对于受限资源的分析不充分。

文献[25-26]都是针对可变任务负载的实时系统,以最优化整体性能为控制性能指标函数,在给定任务集下找到任务可选周期集合,设计一个最佳搜索算法选择任务周期,以便下层 RM 算法进行调度。但以最优化为目标的算法潜在的计



算复杂性高,仅适用系统设计阶段。

文献[27]在能耗限制的实时系统中应用,以系统中总能耗最小为控制性能指标,将任务能耗表示为任务周期的函数,实现系统能耗和性能的综合调度。但文中考虑的能耗模型较为简单,未对该问题进行深入探讨。

Chantem 等在文献[10-11]中提出了一种任务截止期小于等于任务周期广义弹性任务调度算法<sup>[11]</sup>(Generalized Elastic Scheduling Algorithm, GESA)。该算法的下层调度算法依然采用 EDF 经典调度算法,而上层则采用启发式算法迭代求解近似最优解,根据任务截止期与周期的关系分成两种情况进行求解。该算法简化了求解最优解的复杂性,使得系统可以在线求得可行解,不仅适用于系统过载情况,同样适用于系统轻负载情形,该算法可以很好地根据当前负载情况,通过调用该算法求得当前系统的近似最优任务周期集合。但算法求解过程中的对任务执行时间直接取为最坏执行时间,造成了一定的系统资源浪费。

文献[4]综合考虑任务截止期与任务周期的相互关系,将任务截止期看作是任务周期的函数,提出了一种同时调整任务周期和任务截止期的周期选择算法,以保证系统的正常运行。该算法的基本思想与 GESA 类似,以静态值的迭代方法求得动态值的近似结果。

文献[12]在 GESA 基础上,针对任务个数不变的实时系统,将对任务周期的调整直接作用于对任务截止期进行调整,提出了一种对任务截止期进行调整的弹性任务调度算法。该算法以最小化任务截止期限的变化方差为性能指标函数,以满足 EDF 可调度条件和自身限制条件为约束,提出了实时任务中调整任务截止期的弹性调度算法。该算法能较好地处理任务个数确定的实时系统,但对任务个数变化的实时系统不能很好处理。

文献[16]将弹性调度算法应用于数控系统,采用闭环反馈控制模型,针对硬实时、弹性任务以及非实时任务的混合任务集合,在任务截止期等于任务周期的情况下,提出了一种基于贪心算法的弹性任务调度算法。该算法下层调度采用 EDF 调度算法,上层周期调整算法采用贪心算法,根据任务优先级对任务周期进行调整。当任务集合处于过载情况,求得系统中超过额定值的资源请求,从最低优先级任务开始,延展其周期至最大任务周期,依次对较高任务进行调整,直到系统的过载部分消除或所有弹性任务达到最大周期结束。当系统轻负载时,求得轻负载时资源利用率与额定资源利用率的差距,从最高优先级开始,压缩任务周期至最小周期,依次对较低优先级任务进行调整,直至满足最低的任务资源利用率要求或者所有弹性任务均达到最小周期。该算法处理简单,对于一些复杂应用场合并不适用。

文献[17]运用优化理论,提出了反馈—前馈实时调度算法,利用反馈控制来响应任务作业时间在系统运行过程中的动态变化,利用前馈控制处理新加入的系统负载,从而能适应系统中动态的负载变化。通过反馈机制,反馈任务执行时间以及系统中负载情况,同时以最优优化系统整体耗费成本为控制性能指标,自适应地调整弹性任务周期。这种基于反馈机制的周期调整,通过在线求解非线性规划问题获得可行解,作者提出了二次近似的线性目标函数,将非线性规划问题近似用线性规划问题来在线解决周期选择问题。将任务资源利用率变化造成的系统整体耗费成本作为控制性能指标,自适应

地调整任务周期。但算法计算过于复杂,求解时间较长。

总体来说,各研究者基于弹簧模型,针对不同的应用环境,提出了适用不同应用场合的弹性任务调度算法。各种算法由于考虑的问题不尽全面,存在各自的优缺点,但在相应的应用中都取得了一定的效果。

### 3.3 算法的性能评价指标

弹性任务调度算法根据具体应用环境不同,有多种性能评价标准。其中研究者们一般都会采用的评价标准有如下几种:

#### 1) 调度成功率。

调度成功率指的是被某算法调度成功的任务集合数目与所有参加调度的任务集合数目的比值。调度成功的任务集合是指系统可顺利调度并执行完成的任务集合,也就是满足系统可调度条件的任务集合。

调度成功率是对调度算法最直接有效的评估方式。弹性任务调度算法很重要的功效是处理任务过载的情形,算法对过载任务集合的调度成功率将直接体现算法优越性。

#### 2) 资源利用率。

资源利用率指的是实时任务执行时间与实时任务周期之比,用于描述任务占用系统资源的比率。系统中所有任务资源利用率之和为系统资源利用率。任务资源利用率越高,单位时间内可调度的任务实例越多,系统吞吐量越大。

弹性任务调度算法另一个很重要的方面是对任务轻负载状况处理,算法对轻负载的任务调整后,系统资源利用率是反映系统资源是否合理利用的一个重要标志。

#### 3) 时间复杂度。

时间复杂度是算法所解决问题规模的函数,随着问题规模增长,算法执行时间的增长率级别。一般而言,算法时间复杂度越小越好,即调度算法占用处理器资源的执行开销越少。

弹性任务调度算法通过降低系统资源利用率处理任务过载情形,如果算法的复杂度过高将会增加处理器的额外负担。

## 4 弹性调度未来挑战

由于实时系统中任务负载的可变性,保证系统能执行至少超出一些由系统设计者预先确定的阈值任务是实时系统中一个很重要的属性。弹性调度因其明确的调整任务参数的方案而在实时系统中得到较广泛的应用。

虽然在弹性任务模型、调度模型以及调度算法等方面已有部分研究成果,但其真正走向成熟应用还有许多问题需要研究。

#### 1) 多处理器弹性调度问题。

已有的弹性调度大都针对的是单处理器系统,任务在同一处理器上可以进行不同任务缩放,从而实现任务可调度性以及系统资源利用率之间的平衡。在多处理器普及的今天,对于在多核上的弹性调度研究变得非常重要。在多处理器环境中,考虑的不仅仅是单个处理器资源,而是对系统整体资源利用率进行综合考虑;同时要考虑处理器之间通信,在任务压缩、迁移等代价方面进行综合权衡。尤其对于异构多处理器系统,各个处理器的计算能力存在差异,同时各处理器对不同任务的处理能力也存在差异,在该环境中进行弹性调度是一个复杂问题,是弹性调度所面临的一个重要挑战。

#### 2) 能耗约束的弹性调度。

能耗越来越成为大家所关注的重要问题,处理器能耗也

越来越突出。现有的关于能耗限制下的弹性调度研究较少,相关研究所考虑的模型也较为简单。现有的处理器大多支持动态电压缩放,通过降低系统频率来降低系统能耗。如何在弹性调度和能耗调度中获取系统平衡,实现能耗约束下的弹性调度,是未来的一个重要研究方向。

### 3) 算法自适应性。

目前的弹性调度大都针对特定的应用环境进行设计。算法对于其相应的应用场合能取得一定的效果,但对于变化的系统环境,算法可能需要重新设计。如何设计具有一定适用性的弹性调度算法,针对不同的应用环境寻求更合适的任务执行时间估计范围,通过有效反馈使系统能够自适应变化的任务输入,通过在线调整系统参数从而实现系统各方面的均衡。如何提高算法的自适应性也是弹性调度需要解决的一个重要问题。

总的来说,弹性任务调度算法的应用领域越来越广泛,弹性任务调度算法也将不再局限于对任务周期的调整,而是会对实时系统中其他可动态调整的相关因素产生影响。研究更好更快的求出实时弹性任务调度算法的解集合,将会是未来实时任务调度的一个重要研究方向。

## 5 结语

随着计算机技术的不断发展,实时系统的应用越来越广泛,提高系统灵活性的要求也不断增强,对实时系统设计提出了新的挑战。弹性任务调度是保证系统性能及灵活性的重要技术,对其进行研究具有重要意义。目前弹性任务调度相关研究并不完善,许多问题亟待解决。本文从任务模型、系统模型以及调度算法等方面综述了近年来关于弹性任务调度研究进展,并展望了进一步的研究方向。

### 参考文献:

- [1] HAMDAR M, RAMANATHAN P. A dynamic priority assignment technique for streams with  $(m, k)$ -firm deadlines [J]. IEEE Transactions on Computers, 1995, 44(12): 1443–1451.
- [2] LIU J W S, LIN K J, ZHAO W. Algorithms for scheduling imprecise computations [J]. IEEE Computer, 1991, 24(5): 58–68.
- [3] NICOLAU G. Weakly hard real-time systems [J]. IEEE Transactions on Computers, 2001, 50(4): 308–321.
- [4] CHANTEM T, WANG X F, LEMMON M D, *et al.* Period and deadline selection for schedulability in real-time systems [C]// ECRTS 2008: Proceedings of the 20th Euromicro Conference on Real-Time Systems. Washington, DC: IEEE Computer Society, 2008: 168–177.
- [5] MOK A K, WANG W. Window-constraint real-time periodic task scheduling [C]// RTSS'01: Proceedings of the 22nd IEEE Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 2001: 15–24.
- [6] 徐丽萍, 王轩, 卢炎生. 具有自适应特性的反射式动态弹性调度模型[J]. 华中科技大学学报, 2007, 35(2): 26–29.
- [7] BUTTAZZO G, LIPARI G, ABENI L. Elastic task model for adaptive rate control [C]// RTSS'98: Proceedings of the 19th IEEE Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 1998: 286–295.
- [8] CACCAMO M, BUTTAZZO G, SHA L. Elastic feedback control [C]// ECRTS 2000: Proceedings of the 12th Euromicro Conference on Real-Time Systems. Washington, DC: IEEE Computer Society, 2000: 121–128.
- [9] BUTTAZZO G, ABENI L. Adaptive workload management through elastic scheduling [J]. Real-Time Systems, 2002, 23(3): 7–24.
- [10] CHANTEM T, HU X S, LEMMON M D. Generalized elastic scheduling [C]// Proceedings of the 27th IEEE Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 2006: 236–245.
- [11] CHANTEM T, HU X S, LEMMON M D. Generalized elastic scheduling for real-time tasks [J]. IEEE Transactions on Computers, 2009, 58(4): 480–495.
- [12] QIN CHENGANG, YU DONG, WU WENJIANG, *et al.* Control system and elastic scheduling co-design [C]// ICIA 2010: IEEE International Conference on Information and Automation. Washington, DC: IEEE Computer Society, 2010: 1142–1148.
- [13] HARADA F, USHIO T, KOUTAKI H, *et al.* Nonlinear elastic task model and its application to adaptive fair sharing control [C]// SICE-ICASE 2006: International Joint Conference. Washington, DC: IEEE Computer Society, 2007: 2400–2403.
- [14] 王轩. 自适应的实时动态调度服务的设计与实现[D]. 武汉: 华中科技大学, 2006: 19–38.
- [15] 陈宇, 戴琼海. 可变负载动态反馈弹性调度模型及其算法研究[J]. 软件学报, 2004, 15(3): 379–390.
- [16] 秦承刚, 于东, 吴文江, 等. 一种面向数控系统的动态反馈调度模型[J]. 小型微型计算机系统, 2010, 31(4): 788–792.
- [17] CERNAN A, EKER J, BERNHARDSSON B, *et al.* Feedback-forward scheduling of control tasks [J]. Real-Time Systems, 2002, 23(1): 25–53.
- [18] GOOSSENS J, MACQ C. Limitation of the hyper-period in real-time periodic task set generation [C]// Proceeding of the 9th International Conference on Real-Time Systems. Washington, DC: IEEE Computer Society, 2001: 133–148.
- [19] BUTTAZZO G. Achieving scalability in real-time systems [J]. Computer, 2006, 29(5): 54–59.
- [20] 秦承刚, 于东, 吴文江, 等. 基于 Lebesgue 采样的动态反馈实时调度模型[J]. 计算机工程, 2010, 36(19): 1–4.
- [21] CHEN YU, DAI QIONGHONG. A improved elastic scheduling algorithm based on feedback control theory [C]// ICSP '04: 7th International Conference on Signal Processing. Washington, DC: IEEE Computer Society, 2004, 2: 1330–1339.
- [22] BUTTAZZO G, ABENI L. Adaptive rate control through elastic scheduling [C]// Proceedings of the 39th IEEE Conference on Decision and Control. Piscataway: IEEE, 2000, 5: 4883–4888.
- [23] BUTTAZZO G, LIPARI G, CACCAMO M, *et al.* Elastic scheduling for flexible workload management [J]. IEEE Transactions on Computers, 2002, 51(3): 289–302.
- [24] MARINONI M, BUTTAZZO G. Adaptive DVS management through elastic scheduling [C]// Proceeding of the 10th IEEE International Conference on Emerging Technologies and Factory Automation. Washington, DC: IEEE Computer Society, 2005: 307–313.
- [25] SETO D, LEHOCZKY J, SHA L, *et al.* On task schedulability in real-time control systems [C]// Proceeding of the 17th IEEE Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 1996: 13–21.
- [26] BINI E, NATALE M D. Optimal task rate selection in fixed priority systems [C]// Proceeding of the 26th IEEE Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 2005: 399–409.
- [27] 桂盛霖, 雷航. 能耗限制的松弛任务实时调度算法[J]. 电子科技大学学报, 2008, 37(4): 598–601.