

一种适用于实时系统的过载控制策略 HP-OMS*

夏家莉¹, 王文乐^{1†}, 曹重华¹, 韩增波²

(1. 江西财经大学软件与通信工程学院, 南昌 330032; 2. 中国软件与技术服务股份有限公司 CEC 移动技术实验室, 北京 102200)

摘要: 为了提高实时系统的任务成功率和资源利用率, 提出了一种过载控制策略 HP-OMS。HP-OMS 利用周期任务在每个超周期内保持其执行状态不变的特点, 并与特定调度算法相结合, 通过拒绝不可能完成的作业, 消除了任务调度中作业级联抢占问题, 提高了系统资源的有效利用率。实验结果显示, OMS 适用于处理静态或者动态调度任务集, 应用了 OMS 过载控制策略的调度算法能够明显提高系统性能。

关键词: 实时系统; 周期性任务; 过载控制; 作业级联抢占

中图分类号: TP311.131 **文献标志码:** A **文章编号:** 1001-3695(2013)06-1678-04

doi:10.3969/j.issn.1001-3695.2013.06.019

HP-OMS: overload management model for real-time systems

XIA Jia-li¹, WANG Wen-le^{1†}, CAO Zhong-hua¹, HAN Zeng-bo²

(1. School of Software & Communication Engineering, Jiangxi University of Finance & Economics, Nanchang 330032, China; 2. CEC Mobile Technology Laboratory, China National Software & Service Co., Ltd., Beijing 102200, China)

Abstract: In order to improve the mission success rate and resource utilization of real-time systems, this paper proposed a overload management mode (HP-OMS). HP-OMS analyzed the relationship of preemption and abort between jobs by using the feature that the periodic transactions keep their state in their hyper-period and combining with the special algorithms. Though rejecting those jobs which can't catch their deadline, HP-OMS can eliminate the cascade preemption of the jobs. Also, the system resource could be efficiently utilized. The results of the simulation show that, the OMS can adaptively deal with the statistic or dynamic task set. Additionally, by applying OMS, the scheduling algorithms can obviously improve the performance of the real-time system.

Key words: real-time system; periodic task; overload management; cascade preemption of jobs

大多数无人值守的工业控制很多实时系统的任务是以周期任务为主, 并且这些周期任务通常为硬实时或固实时任务^[1]。然而当系统过载时, 很难保证任务的每次执行都能顺利完成^[2]。任务的很多作业在执行过程中, 由于受到其他任务作业的影响而夭折, 从而导致任务成功率的降低以及系统资源的浪费。为确保系统任务的成功率, 提高系统资源的利用率, 已有的工作提出接纳控制与预分析两种策略。

接纳控制通过拒绝不可能完成的任务, 保留其资源用于其他任务的执行, 增大其他任务的作业顺利完成的机会。黄文广等人^[1]提出一种基于价值收益的接纳控制协议, 只有当新任务所带来的收益不低于其造成的损失时, 系统才接纳该任务; Atlas 等人^[3]提出一种基于空闲时间偷窃算法的接纳机制, 该机制在不造成周期任务错过其截止期的前提下, 尽可能早地接纳可以完成的非周期任务; 夏家莉^[4]提出了三种接纳策略, 即基本接纳策略、抢占接纳策略和空隙接纳策略, 针对系统的不同需求选取不同的策略; Niz 等人^[5]针对混合任务中高紧迫性任务被抢占并错过截止期的问题, 提出了结合紧迫性计算和空闲度分析的控制算法, 有利于解决紧迫性颠倒问题。

除了采用接纳控制的方法之外, 预分析技术也能起到优化系统性能的作用。刘云生等人^[6]针对系统任务在运行过程中动态改变的特点, 提出了一种“三段法”预分析, 增加了在任务执行过程中的动态分析能力。本文提出了一种过载管理策略 (overload management strategy, OMS), 利用周期任务在超周期内保持其执行状态不变的特性, 通过任务预分析和接纳控制, 分析系统中作业^[7,8]之间的抢占与夭折关系, 将系统中的作业逐渐确认为稳定完成作业和绝对夭折作业。当使用基于 OMS 的调度策略进行作业调度时, 放行那些稳定完成作业, 而拒绝执行绝对夭折作业, 解决了在级联抢占过程中部分作业因无法获得足够的系统资源而夭折的问题, 从而实现了系统性能优化。

1 HP-OMS 策略分析

1.1 HP-OMS 模型

假设 $RT = \{\tau_1, \tau_2, \dots, \tau_n\}$ 为实时系统中的任务集合, $\tau_i = \{T_{i1}, T_{i2}, \dots, T_{ij}\}$ 为任务 τ_i 的作业集合, HP 表示超周期的大小。

每个作业 T_{ij} 具有以下属性: P_{ij} 表示任务 τ_i 的周期; C_{ij} 表示

收稿日期: 2012-09-27; **修回日期:** 2012-11-05 **基金项目:** 国家自然科学基金资助项目(60763002); 江西省自然科学基金资助项目(2008GZS0021)

作者简介: 夏家莉(1965-), 女, 湖北荆州人, 教授, 博导, 主要研究方向为实时数据库、实时系统、软件工程; 王文乐(1985-), 男(通信作者), 博士研究生, 主要研究方向为实时系统、实时调度(wwle1985@126.com); 曹重华(1977-), 男, 讲师, 博士研究生, 主要研究方向为实时数据库、软件工程; 韩增波(1986-), 男, 硕士, 主要研究方向为实时系统。

作业 T_{ij} 的执行时间; D_{ij} 表示作业 T_{ij} 的相对截止期; A_{ij} 表示作业 T_{ij} 的相对到达时间; PRI_{ij} 表示作业 T_{ij} 的优先级; e_{ij} 表示作业 T_{ij} 的已执行时间; b_{ij} 表示作业 T_{ij} 的相对开始执行时间; f_{ij} 表示作业 T_{ij} 的相对结束时间。其中,“相对”是指相对于每个超周期的开始时刻而言。针对该模型,本文假定以下三个前提:a) 系统中任务间相互独立,即除 CPU 外,任务间无冲突资源与相互依赖关系;b) 任务优先级具有唯一性;c) 任务每次执行的截止期不超过该任务下次的到达时间,即 $D_{ij} \leq A_{i,(j+1)}$ 。

1.2 HP-OMS 分析

通常系统过载会加剧资源(特别是 CPU)竞争的程度,大大增加作业的抢占次数,从而增大作业级联抢占出现的概率。为了解决作业级联抢占问题,HP-OMS 分析作业之间抢占与夭折的关系,并给出如下定义。

定义 1 夭折作业集。由于执行 T_{ij} 而造成夭折的作业集合称为 T_{ij} 的夭折作业集,用 AJS_{ij} (aborted job set) 表示。显然, AJS_{ij} 可由式(1)表示。

$$AJS_{ij} = \{ T_{mn} \mid (PRI_{mn} < PRI_{ij}) \wedge (D_{mn} < (f_{ij} + (C_{mn} - \max(0, b_{ij} - A_{mn})))) \wedge ((A_{mn} + C_{mn}) > A_{ij}) \} \quad (1)$$

同时,作业 T_{ij} 也可能被高优先级的作业所抢占,本文称这些作业是 T_{ij} 的抢占作业,由此得到定义 2。

定义 2 抢占作业集。在 T_{ij} 执行期间抢占其执行的作业集合称为 T_{ij} 的抢占作业集,用 PJS_{ij} (preempt job set) 表示。显然, PJS_{ij} 可由式(2)表示。

$$PJS_{ij} = \{ T_{mn} \mid (PRI_{mn} > PRI_{ij}) \wedge (D_{mn} > A_{ij}) \wedge (A_{mn} < f_{ij}) \} \quad (2)$$

PJS_{ij} 中所有作业的到达时间都小于 f_{ij} 。然而那些到达时间在 (f_{ij}, D_{ij}) 的作业在以后超周期内也可能对 T_{ij} 的执行造成影响,因此得到定义 3。

定义 3 影响作业集。它是指在 T_{ij} 的执行周期内,能够影响其执行的所有作业的集合,称为 T_{ij} 的影响作业集,用 EJS_{ij} (effect job set) 表示。显然, EJS_{ij} 可表示为

$$EJS_{ij} = \{ T_{mn} \mid (PRI_{mn} > PRI_{ij}) \wedge (D_{mn} > A_{ij}) \wedge (A_{mn} < D_{ij}) \} \quad (3)$$

由于 $f_{ij} \leq D_{ij}$, 故 $PJS_{ij} \subseteq EJS_{ij}$ 。系统过载时,作业之间的抢占势必造成某些作业不能满足其截止期。根据作业的完成情况,HP-OMS 将作业分为稳定完成作业、浮动作业和绝对夭折作业三类。

定义 4 稳定完成作业。如果作业 T_{ij} 能够顺利完成,并确保在以后任何超周期内保持 (b_{ij}, f_{ij}) 不变,则称 T_{ij} 为稳定完成作业,其集合用 SFS (steady finish job set) 表示。

定义 5 绝对夭折作业。如果 T_{ij} 在以后任何一个超周期内总是因固定的抢占作业集而夭折,则称 T_{ij} 为绝对夭折作业,其集合用 AAS (absolute abort job set) 表示。

若 T_{ij} 既不是稳定完成作业,也不是绝对夭折作业,则称 T_{ij} 为浮动作业,其集合用 FS (float job set) 表示。

根据前面的几个定义,可以得到如下定理。

定理 1 $\exists T_{mn}, (T_{mn} \in PJS_{ij}) \wedge (T_{mn} \in FS)$ 是 $T_{ij} \in FS$ 的充分条件。

证明 因为 $T_{mn} \in FS$, T_{mn} 的执行时间不固定;又因为 $T_{mn} \in PJS_{ij}$; 所以 T_{mn} 会影响 T_{ij} 的执行; 所以 (b_{ij}, f_{ij}) 不能确保不变, 即 $T_{ij} \in FS$ 。

定理 2 $\forall T_{mn}, (e_{ij} = C_{ij}) \wedge (PJS_{ij} \subseteq (SFS \cup AAS))$ 是 $T_{ij} \in SFS$ 的充分条件。

证明 过程如同定理 1。

根据定义 4, 当 T_{ij} 被确定为稳定完成作业后, 其在后面的所有超周期内保持 (b_{ij}, f_{ij}) 不变, 因此 AJS_{ij} 中的所有作业在以后的所有超周期内都会夭折, 故可得到下面的推论。

推论 1 若 $T_{ij} \in SFS$, 则 $AJS_{ij} \subseteq AAS$ 。

然而推论 1 并不能确定出所有的绝对夭折作业, 存在一些作业不属于其他任何作业的夭折作业集, 但由于受到多个高优先级稳定作业的影响, 始终无法得到充足资源, 因此这些作业也属于绝对夭折作业, 给出定义 6。

定义 6 极大使用时间。作业 T_{ij} 在时间段 (A_{ij}, D_{ij}) 所能使用的最大时间, 称为 T_{ij} 的极大使用时间, 用 BUT_{ij} (biggest used time) 表示, 其计算公式为

$$\forall T_{mn}, (T_{mn} \in SFS) \wedge (T_{mn} \in EJS_{ij}) \wedge (f_{mn} > A_{ij}) \wedge (b_{mn} > D_{ij}) \\ BUT_{ij} = D_{ij} - A_{ij} - \sum (\min(f_{mn}, D_{ij}) - \max(A_{ij}, b_{mn})) \quad (4)$$

HP-OMS 使用极大使用时间来确定浮动作业是否为绝对夭折作业, 由此得到推论 2。

推论 2 若 $BUT_{ij} < C_{ij}$, 则 $T_{ij} \in AAS$ 。

在诸如 LSF 的颠簸调度算法中, 由于作业之间的互相抢占而产生无法计算作业的集合, 所以 HP-OMS 不支持颠簸调度算法。而在无颠簸调度算法中, 总存在一些顺利完成作业 T_{ij} 的 $PJS_{ij} = \emptyset$, 故满足定理 2。

不难看出, HP-OMS 具有以下两条性质:

性质 1 HP-OMS 的 SFS 中至少包含级联抢占链中优先级最高的作业。

性质 2 HP-OMS 中任何一个作业最终都会被确定为稳定完成作业或者绝对夭折作业。

2 HP-OMS 设计

系统执行前, HP-OMS 将所有周期任务分解成作业, 由于此时系统并未对任何作业进行确认, 因而将所有作业初始为浮动作业。然后以作业为基本单位进行调度, 不断地将浮动作业确定为稳定完成作业或绝对夭折作业。

HP-OMS 的调度可分为静态预分析、放行检测、作业调度和作业确定四个阶段。本文根据任务集的是否变化, 将实时系统分为静态系统和动态系统两类。

2.1 静态预分析

静态预分析的目的是为系统的调度提供优化系统性能的知识, 其对应于作业编译时的处理^[6]。在本阶段, 首先将所有的周期任务分解成作业, 同时提取作业的相关属性。为每个作业创建三个队列, 即 AJQ (aborted job queue)、PJQ (preempt job queue) 和 EJQ (effect job queue), 分别存放该作业的夭折作业集、抢占作业集、影响作业集。HP-OMS 分别采用式(1)~(3)计算三个集合。然而式(1)(2)中的 b_{ij} 和 f_{ij} 是动态属性, 必须放宽两个公式的条件。

由于 $f_{ij} \leq D_{ij}$, $(C_{mn} - \max(0, (b_{ij} - A_{mn}))) \leq C_{mn}$, 经放宽后, 式(1)转变成式(5):

$$ASJ_{ij} = \{ T_{mn} \mid (PRI_{mn} < PRI_{ij}) \wedge (D_{mn} < (D_{ij} + C_{mn})) \wedge ((A_{mn} + C_{mn}) > A_{ij}) \} \quad (5)$$

因为 $PJS_{ij} \subseteq EJS_{ij}$, 则可将式(2)放松为式(3)。HP-OMS 采用式(3)计算作业 PJQ、EJQ, 采用式(5)计算作业 AJQ。

2.2 放行检测

放行检测的目的是将满足推论 2 的浮动作业确定为绝对

夭折作业,同时放行所有可能完成的浮动作业。为判断是否放行新作业,HP-OMS 将剩余极大使用时间作为放行的门檻。剩余极大使用时间的定义如下。

定义 7 剩余极大使用时间。作业 T_{ij} 在时间段 (t, D_{ij}) 所可能使用的最大时间,称为 T_{ij} 的剩余极大使用时间,用 $LBUT_{ij}$ (leave biggest used time) 表示,其中 t 为当前时刻,其计算公式为

$$\forall T_{mn}, (T_{mn} \in SFS) \wedge (T_{mn} \in EJS_{ij}) \wedge (A_{mn} > b_{ij})$$

$$LBUT_{ij} = D_{ij} - t - \sum (\min(f_{mn}, D_{ij}) - \max(A_{ij}, b_{mn})) \quad (6)$$

放行检测中,根据式(4)(6)分别计算 BUT_{ij} 和 $LBUT_{ij}$ 。若 $BUT_{ij} < C_{ij}$, 则拒绝 T_{ij} , 确定该作业为绝对夭折作业;若 $(BUT_{ij} \geq C_{ij}) \wedge (LBUT_{ij} < C_{ij})$, 则拒绝 T_{ij} , 不更改该作业类型;反之,放行 T_{ij} 。

2.3 作业调度

如第 1 章所述 HP-OMS 支持无颠簸调度算法。由于调度过程中,该策略并没有引入新的调度算法,因此本文不再讨论。

2.4 作业确认

作业确认的目的是确认顺利完成的浮动作业 T_{ij} 的状态。该阶段在 T_{ij} 完成时刻提取动态信息 f_{ij} , 根据式(2)计算 PSJ_{ij} 。若结果满足定理 2, 则将 T_{ij} 确定为稳定完成作业,同时根据式(1)计算 AJS_{ij} , 并将 AJS_{ij} 中的所有作业放入确定为绝对夭折作业中。

根据前文对 HP-OMS 四阶段的介绍,总结其实现算法如下:

Notations:

- $C_{ij}, f_{ij}, D_{ij}, BUT_{ij}, LBUT_{ij}, PJS_{ij}, AJS_{ij}, SFS, AAS$ have been depicted in the paper
- AA_{ij} denotes the job is Absolute Abort Job
- SF_{ij} denotes the job is Steady Finish Job.

Procedure: Realse_test (T_{ij})

begin

if ($AA_{ij} = 1$) return REJECT;

if ($SF_{ij} = 1$) return ADMIT;

compute $BUT_{ij}, LBUT_{ij}$;

if ($BUT_{ij} < C_{ij}$) { $AA_{ij} = 1$; return REJECT; }

else

if ($LBUT_{ij} < C_{ij}$) return REJECT;

else return ADMIT;

end

Procedure: Schedule (T_{ij})

begin

if (Release_test (T_{ij}) == ADMIT) {

T_{ij} executes;

if (($e_{ij} = C_{ij}$) && ($SF_{ij} \neq 1$)) {

Compute PJS_{ij} ;

if ($PJS_{ij} \subseteq (SFS \cup AAS)$) {

$SF_{ij} = 1$; Compute AJS_{ij} ;

$\forall T_{mn}, (T_{mn} \in AJS_{ij}) : AA_{ij} = 1$; }

end

2.5 性能分析

实时系统中最重要的是满足任务的时间约束,因此本文主要在时间复杂度上对 HP-OMS 进行分析。静态预分析发生于系统初始之时不会影响系统的真正运行,另外,调度的性能与

具体的算法有关,故本节主要分析放行检测和作业确定两阶段的时间复杂度。

放行检测阶段时间主要消耗在计算 BUT_{ij} 和 $LBUT_{ij}$ 上。由式(4)(6)可知,计算两者需要遍历一次 T_{ij} 的 EJQ , 计算时间复杂度为 $O(N_EJQ_{ij})$ 。其中, N_EJQ_{ij} 为 T_{ij} 的 EJQ 中作业的个数。

作业确定阶段时间主要消耗在计算 PJS_{ij} 和 AJS_{ij} 上。确定 T_{ij} 为稳定完成作业需遍历一次 T_{ij} 的 PJQ ; 若 T_{ij} 为稳定完成作业, 则又需遍历一次 AJQ 来计算 AJS_{ij} 。因此,进行一次作业确认的时间复杂度为 $O(N_PJQ_{ij} + N_AJQ_{ij})$ 。其中, N_PJQ_{ij} 为 T_{ij} 的 PJQ 中作业的个数, N_AJQ_{ij} 为 AJQ 中作业的个数。

综上所述,一个浮动作业运行最坏情况下所带来的时间开销为 $O(N_EJQ_{ij} + N_PJQ_{ij} + N_AJQ_{ij})$ 。由于实时作业通常是“短”作业,该值通常是一个非常小的值。一个稳定完成作业或绝对夭折作业无须经过放行检测和作业确认阶段,所带来的时间开销为 0。因此,当系统稳定后,系统性能达到最优化。

作业级联抢占使得系统不能在一个超周期内将所有作业都确定完毕,因此系统经过几个超周期才能进入稳定状态,这也是评价 HP-OMS 的一个性能指标。该指标取决于系统中最长级联抢占链的长度,而该长度又与调度算法的类型相关。

表示 1 $TS_i \rightarrow TS_j$ 是指任务 τ_i 的作业被任务 τ_j 的作业所抢占。

静态优先级调度算法中,最坏情况下存在 $TS_1 \rightarrow TS_2 \rightarrow \dots \rightarrow TS_n$ 这样的抢占链,其长度为 n , 其中 n 为系统中优先级的个数。由 HP-OMS 策略分析可知,每个超周期内,系统至少能确定两个优先级层上的作业的类型,因此系统最多执行 $(\lceil n/2 \rceil)$ 个超周期后便可进入稳定状态。

无颠簸的动态优先级调度中,由于任务的优先级动态改变,存在 $TS_n \rightarrow TS_{n-1} \rightarrow \dots \rightarrow TS_1 \rightarrow TS_{n-1} \rightarrow \dots$ 这样的级联抢占链,其长度不可估计。然而动态优先级算法通常会减少系统的抢占次数,从而不会产生较长的级联抢占链。

3 HP-OMS 实现

前面针对 HP-OMS 的分析建立于静态系统上,而在系统的实际运用中,某些条件可能发生改变,使得新周期任务加入调度任务队列,或者某些正在运行的任务中途停止。为适应调度任务集的动态改变,HP-OMS 将超周期分为两类,即系统超周期和调度超周期。系统超周期是指系统内所有任务集合的超周期,用 SY-HP 表示;调度超周期是指可调度任务集的超周期,用 SC-HP 表示。显然, $SY-HP = m \times SC-HP$, 其中 m 为不小于 1 的自然数。SY-HP 在系统运行期间保持不变,而 SC-HP 随调度任务集的改变而改变。因此,系统进入稳定状态更准确的说法为:系统调度作业集中所有作业都被确定为稳定完成作业或绝对夭折作业。

为使 HP-OMS 适用于动态系统,首先在静态预分析阶段计算 SY-HP 和初始的 SC-HP,使用 SY-HP 分解所有任务,在放行检测、作业调度和作业确认三个阶段使用 SC-HP 来操作作业。动态系统下,调度作业集不等于全部作业集,当任务加入或者中途退出时,系统发生两个变化:a) SC-HP 发生改变;b) 调度作业集发生改变,作业 T_{ij} 可调度的条件为 $j \times P_i \leq SC-HP$ 。

调度作业集的改变影响到作业的抢占关系,必须重新分析被影响作业之间的抢占关系,为此,HP-OMS 给出以下定义。

定义 8 重新确定作业集。由于任务 τ_i 的加入或者退出,需要重新确定类型的作业的集合,称之为 τ_i 的重新确定作业集,用 ADS_i (again define job set) 表示。显然, ADS_i 包括所有优先级低于 PRI_i 的可调度作业。何时重新开始确定 ADS_i 的作业类型是适应调度作业集动态改变的关键。

假设 τ_i 加入后第一次的相对开始执行时间为 FET (first execute time), 在当前 SC-HP 的 $(0, FET)$ 内, 不存在 τ_i 的作业执行, 然而 τ_i 在下一个 SC-HP 的 $(0, FET)$ 内执行。由于作业级联抢占的存在, τ_i 在 $(0, FET)$ 内的执行可能影响到 FET 时刻后 ADS_i 中某些作业的执行, 因此不能在 τ_i 第一次执行之时就开始重新确定 ADS_i 中的作业。重新确定的时间点 ADT (again define time) 需要满足一个条件: 在无新任务加入的情况下, 在 ADT 时刻, τ_i 与 ADS_i 的所有当前作业都必须结束。显然 ADT 应该是 τ_i 加入后第一个 ADS_i 与 τ_i 的超周期的结束时刻。在 (FET, ADT) 时间段, 系统不对 ADS_i 中任何作业进行确定, 而其他作业依然使用 HP-OMS。ADT 时刻后, 重新开始对 ADS_i 中的作业进行确定。注意, 在 (FET, ADT) 阶段, 虽然不对 ADS_i 的作业进行确定, 但放行检测依然执行。当一个执行任务 τ_i 中途退出调度任务队列时, 其处理过程与新任务加入时相同。

因调度作业集改变而带来的额外开销是 $O(N_{ADS} + N)$ 。其中, N_{ADS} 是重新确定作业集中作业的个数, N 为可调度任务队列中任务的个数。 $O(N_{ADS})$ 用来取消重新确定作业集中作业的类型标志, $O(N)$ 用来计算 SC-HP 和 ADT。

4 实验仿真

实验在 CPU 为 AMD Athlon 2.0 GHz、内存为 1 GB 的 PC 机上运行, 实验程序采用 C 语言实现。实验中 HP-OMS 的基本调度算法分别采用静态优先级调度策略 HVF (价值高优先) 和动态优先级调度策略 EDF (最早截止期优先)。将应用了 HP-OMS 的 HVF 和 EDF 算法分别称为 HPHVF 和 HPEDF。为了更好地说明 HP-OMS 的优势, 实验中随机生成三组任务, 每组任务的系统负载因子为 2.0。分别采用四种算法对产生的任务进行调度, 结果取三次实验的平均值。

4.1 性能评价指标

本文主要从以下四个指标进行分析:

a) MDP (miss deadline percent)。假设 total_number 和 success_number 分别表示一个超周期中作业的总数量和成功完成作业的数量, 则

$$MDP = \frac{\text{total_number} - \text{success_number}}{\text{total_number}} \times 100$$

b) TV (total value)。系统在一个超周期内实现的总价值, 其计算公式为

$$TV = \sum V_{ij}, e_{ij} = C_{ij}$$

c) AU (actual utilization)。系统在一个超周期内顺利完成作业所使用的时间与超周期 (HP) 的比率。在不考虑调度开销的情况下, 可以按照下式计算:

$$AU = \frac{\sum C_{ij}}{SP} \times 100 \quad e_{ij} = C_{ij}$$

d) VPP (valid preempt percent)。指系统在一个超周期内的有效抢占率, 其计算公式为

$$VPP = \frac{VPT}{TPT} \times 100$$

其中: VPT 为有效抢占次数, TPT 为总抢占次数。

4.2 性能分析

4.2.1 静态系统中的性能分析

图 1(a) 中 x 轴上的 HVF 表示传统 HVF 的一个超周期, HPHVF- n 表示 HPHVF 的第 n 次超周期; (b) 中的 EDF、HPEDF- n 具有同样的含义; (c) 中 HP- n 是指系统第 n 个超周期。

图 1(a) 和 (b) 对 HVF 和 EDF 分别与相应的改进策略在 MDP、AU、VPP 三个指标方面进行比较, 图 1(c) 在 TV 指标上进行比较。可以看出, HP-OMS 对于两种调度算法的性能都有了很大的改进。这是由于 HP-OMS 通过拒绝那些不可能完成的作业, 破坏了系统中的作业级联抢占链, 从而使得更多的作业能够顺利完成。当系统进入稳定状态后, 其放行的作业都为稳定完成作业, 则消除了无效抢占, 故 VPP 达到理想的 100%。

另外, 随着系统超周期数的增加, 系统性能改进的程度越来越小。这是由于系统中作业级联抢占链中长度的不同, 使得 SFS 的增长速率随超周期数的增加而减小; 又因为 HVF 中产生的作业级联抢占链长度小于 EDF 所产生的作业级联抢占链, 使得相对于 HPEDF、HPHVF 能够更快地使系统达到稳定状态。

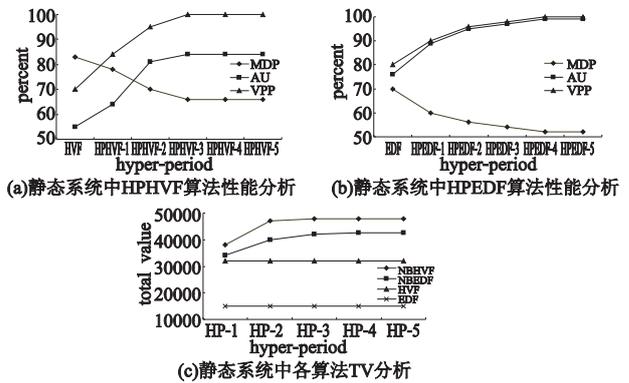


图 1 静态系统中四种算法的性能比较

4.2.2 动态系统中的性能分析

图 2 中的 HP- n 与图 1(c) 中的类似, 然而 HP- n 的大小并非保持不变, $HP-3 = m \times HP-2 (m > 1)$ 。实验依然选择静态系统实验中的任务组, 不同之处在于该次实验设计一个新任务在 HP-3 中的某一时刻开始执行。

图 2(a) 比较了动态系统下四种算法的 MDP 性能。在前三个超周期内, 四种算法保持与静态系统一样的执行情况。当 HP-3 内新任务开始执行后, 新任务的加入造成 CPU 竞争更加激烈, 从而造成四种算法的 MDP 都上升。造成 HPHVF 与 HPEDF 的 MDP 上升的另一原因是新任务的加入使得两种算法重新确定某些作业的类型, 并且在 (FET, ADT) 时间段内对这些作业采用传统的调度算法执行。由于实验中新任务到达时间位于 HP-3 的中间某一时刻, 因此新任务的执行影响到 HVF、EDF 算法在 HP-3、HP-4 两个超周期内的执行情况。HP-4 后, HVF 与 EDF 算法中的 MDP 保持不变。而由于 RST 时刻后, 系统恢复到 HP-OMS 中执行, 因此 HPHVF 与 HPEDF 的 MDP 在 HP-4 开始下降。图 2(b) ~ (d) 分别在 AU、VPP、TV 三个方面进行了比较, 其指标变化趋势如同 MDP, 即新任务加入后, 系统出现一段波动。在图 2(d) 中, 由于加入新任务使得调度超周期增大数倍, 因此在 HP-3 后系统总收益呈直线提高。对比图 1 和 2, 动态系统中任务的执行情况大致类似于静态系统, 不同之处是当新任务加入时, 动态系统会产生一段时间的波动。

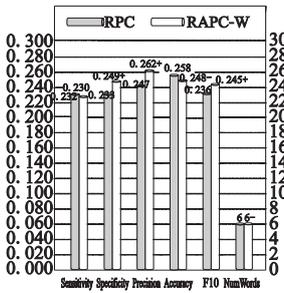


图7 Emotion上的对比结果

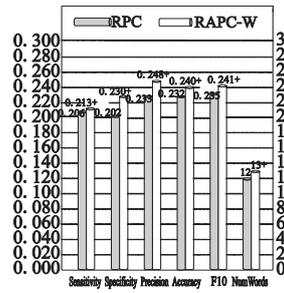


图8 Yeast上的对比结果

4 结束语

本文在多标签数据集划分方法 RPC 的基础上,提出了一种新方法 RPC-W,在标签组合时引入了 WordNet 校验,通过在 UCI 知识库以及 KEEL 提供的数据集上的一系列实验,验证了本文所提方法的有效性。需要指出的是,本文在对标签组合引入 WordNet 校验时,只是进行了简单的不相关词过滤,并未有效利用数据集中的上下文信息。因此,如何利用数据集中的上下文信息并结合 WordNet 进行标签组合的过滤,是下一步值得深入研究的工作。

参考文献:

[1] SOUMAKAS G T. Multilabel classification: an overview[J]. *International Journal of Data Warehousing and Mining*,2007,3(3):1-13.

[2] SANTOS A M. Analyzing classification methods in multi-label tasks [M]. Berlin:Spring-Verlag,2010:137-142.

[3] SCHAPIRE R E. Boostexter: a boosting based system for text categorization[J]. *Machine Learning*,2000,39(2-3):135-168.

[4] ZHANG Min-ling, ZHOU Zhi-hua. A K-nearest neighbor based algorithm for multi-label classification[C]//Proc of IEEE International

Conference on Granular Computing, 2004:718-721.

[5] ZHU Sheng-huo,JI Xiang,XU Wei, et al. Multi-labelled classification using maximum entropy method[C]//Proc of the 28th Annual International ACM SIGIR Congerence on Reserch and Development in Information Retrieval. New York: ACM Press,2005:274-281.

[6] CHANG C C. A library for support vector machine[EB/OL]. (2010-01-10). <http://www.csie.ntu.tw/~cjlin/libsvm>.

[7] TSOUMAKAS G, DIMOU A, SPYROMITROS E, et al. Correlation-based pruning of stacked binary relevance models for multi-label learning[J]. *Machine Learning*,2009,54(1):5-32.

[8] TROHIDIS K. Multi-label classification of stack binary relevance models for multilabel classifiers [J]. *Eurasip Journal on Audio Speech and Music Processing*,2011,4(1):4-6.

[9] READ J. A pruned problem transformation method for multi-label classification[C]//Proc of New Zealand Computer Science Reasearch Student Conference. 2008:143-150.

[10] TSOUMAKAS G, KATAKIS L, VLAHAVAS L. Mining multi-label data[C]//Proc of Data Mining and Knowledge Discovery Handbook. 2010:667-685.

[11] SECHIDIS K, TSOUMAKAS G, VLAHAVAS I. On the stratification of multi-label data [C]//Proc of European Conference on Machine Learning and Knowledge Discovery in Databases. 2011:145-158.

[12] READ J. Classifier chains for multi-label classification [J]. *Machine Learning Journal Springer*,2011,85(3):333-359.

[13] LESK M. Automatic sense disambiguation using machine readable dictionaries: hoto tell a pine cone from an ice cream cone[C]//Proc of the 5th Annual International Conference on Systems Documentation. New York: ACM Press,1986:24-26.

[14] JIANG J J, CONRATH D W. Semantic similarity based on corpus statistics and lexical taxonomy[C]//Proc of International Conference on Research in Computational Linguistics Manchester. 1997:19-23.

(上接第 1681 页)

实验表明,不论在静态系统还是在动态系统中,HPHVF 与 HPEDF 的性能都分别优于其对应的传统算法。

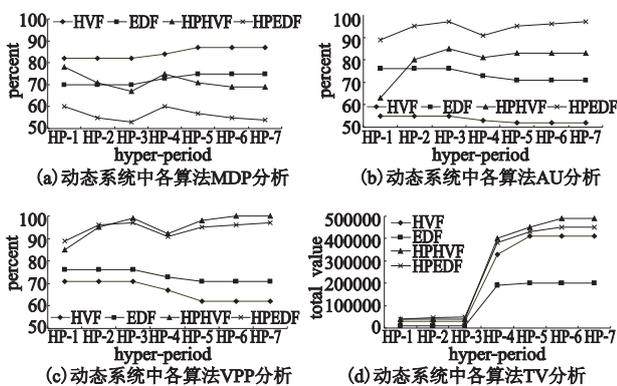


图2 动态系统中四种算法的性能比较

5 结束语

系统过载加剧了资源竞争的程度,增大了作业级联抢占出现的概率,从而增大了任务的截止期错失率,降低了系统性能。有选择地执行作业,可以合理限制系统负载、减少无效抢占、降低资源竞争程度、提高资源利用率和系统收益。HP-OMS 通过确定作业的类型,拒绝执行那些无法完成的作业,使得系统性能得到最优化。HP-OMS 的主要创新点在于:解决了过载系统

中作业级联抢占的现象;该策略能够适应调度任务集的动态改变。

参考文献:

[1] 黄文广,于士齐.周期多帧任务的固定优先级调度算法的调度分析[J]. *计算机研究与发展*,2001,38(2):240-245.

[2] NAGY S, BESTAVROS A. Value-cognizant admission control for RT-DB systems[C]//Proc of the 16th IEEE Real-time System Symposium. Washington DC: IEEE Computer Society,1996:230-239.

[3] ATLAS A, BESTAVROS A. Slack stealing job admission control [R]. Boston, MA:Boston University,1998.

[4] 夏家莉.适用于嵌入式实时数据库系统的接纳控制机制 IACM [J]. *计算机学报*,2004,27(3):295-301.

[5] De NIZ D, LAKSHMANAN K, RAJKUMAR R. On the scheduling of mixed-criticality real-time task sets[C]//Proc of the 30th IEEE Real-time Systems Symposium. Washington DC: IEEE Computer Society, 2009:291-300.

[6] 刘云生,李国徽.基于预分析的实时任务处理[J]. *计算机研究与发展*,2002,39(12):1728-1734.

[7] CHENG A M K, FENG Chen. Predictive thermal management for hard real-time tasks[J]. *ACM SIGBED Review*,2006,3(1):35-40.

[8] SCHRANZHOFER A, CHEN Jian-jia, THIELE L. Timing analysis for TDMA arbitration in resource sharing systems[C]//Proc of the 16th IEEE Real-time and Embedded Technology and Applications Symposium. Stockholm: IEEE Computer Society,2010:215-224.