

文章编号:1001-9081(2010)10-2735-03

多处理器全局单调比率的可调度性分析

石林勇, 晏立

(江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

(shilingyong@126.com)

摘要:针对全局单调比率(RM)调度的多处理器系统中最高优先级任务数量少于处理器数量时, Bertogna 等给出的最坏情况计算任务受到的干涉过于悲观, 证明了任务受到最高优先级任务的干涉不会出现最坏情况, 它受到非最高优先级任务干涉仍可能出现最坏情况。分析得出了任务受到最高优先级任务干涉的最大可能值, 由此得到了一个更紧的可调度性判断条件。实验结果表明, 提出的方法提高了判为可调度的任务集数量。

关键词:实时系统; 多处理器; 全局调度; 可调度性分析; 干涉

中图分类号: TP316.2 **文献标志码:** A

Schedulability analysis of global rate-monotonic scheduling algorithms on multiprocessor platforms

SHI Lin-yong, YAN Li

(School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang Jiangsu 212013, China)

Abstract: In the Rate-Monotonic (RM) global scheduling for multi-processor system, when the highest-priority task is fewer than the number of processors, the worst-case calculation interference given by Bertogna is too pessimistic. The authors proved that the interference of the task with highest priority on a task was not so pessimistic. But the interference of the task without highest priority on a task may be as pessimistic as the worst case presented by Bertogna, et al. The possible maximum interference of the tasks with highest priority was given by analysis, and then a tighter schedulability policy was put forward. The experimental results show that the proposed schedulability policy increases the amount of detected task set.

Key words: real-time system; multiprocessor; global scheduling; schedulability analysis; interference

0 引言

可调度性判定是实时调度算法的关键问题。可调度性判定就是判定给定的 n 个实时任务在应用某种调度算法的前提下能否产生一个可行调度。过去的研究都是集中在单处理器上, 然而, 单处理器上得到的大多数结果并不能扩展到多处理器平台上。本文研究了 m 个相同处理能力处理器组成的多处理器上调度周期任务的问题。解决这个问题有两种方法, 划分任务处理法和全局调度程序处理法。划分任务方法首先离线地将任务划分到各个处理器上, 然后用单处理器上的方法来处理划分给它的任务。与划分方法不同的是, 全局调度程序法维护一个唯一的全局就绪队列, 有处理器可用时就从该队列中取出任务并分配给它执行。划分方法中的任务不能在不同处理器上执行, 而全局调度法则允许同一任务在不同的处理器上执行。近几年, 多处理器平台上使用全局调度算法调度周期任务和偶发任务的研究取得了一系列成果。Goosen 等人^[1]研究了全局最早时限优先(Earliest Deadline First, EDF)调度周期任务的问题, 得到了一个基于利用率的充分判定条件; 为了克服 Dhall 效应的影响, 他们提出了改进的 EDF 策略。Baruah 等人^[2]研究了不同计算速度的处理器组成的多处理器平台上, 全局 RM 算法调度周期任务的问题, 得到了处理器计算带宽和任务利用率之间满足可调度的关系。Baker 等人^[3-4]从任务错过截止期的角度出发, 对使用全局 EDF 和固定优先级(Fixed Priority, FP)的情况进行了研究, 得

到了任务集可调度时, 每个任务都必须满足的充分条件。Bertogna 等人^[5-6]基于 Baker 的思想, 进一步研究该问题, 得到了更紧密的判断条件。此外 Bertogna 等人还在文献[7]中研究了使用任意保持工作算法调度周期任务和偶发任务的问题, 得到了一个充分必要的可调度条件, 由于没有理想复杂度的方法计算任务受到的干涉, 使用干涉上界来替代, 从而得到了一个充分的可调度条件。

本文基于文献[7]的思想, 对使用全局 RM 调度策略, 且系统中最高优先级任务数量不多于处理器数量的情况进行了研究, 证明了任务受到最高优先级任务的干涉不会出现文献[7]中指出的最坏情况, 并分析得到了任务受到最高优先级任务干涉的最大值, 从而得到了一个更紧的可调度性判断条件。最后通过实验将得到的方法与文献[7]给出的 FP 判定方法进行了比较。

1 系统模型

在 m 个处理能力相同的处理器组成的多处理器平台上, 使用全局 RM 策略调度 n 个周期任务组成的任务集 τ 。任务集 τ 中的每个任务 τ_k 使用最坏计算时间 C_k , 相对截止期 D_k 以及周期 T_k 来表示。任务 τ_k 是作业 J_k^j 的无穷序列, 作业 J_k^j 使用到达时间 r_k^j , 绝对截止期 d_k^j , 计算时间 c_k^j , 以及完成时间 f_k^j 来表示。它们之间满足 $C_k \geq c_k^j, r_k^j \geq r_k^{(j-1)} + T_k, d_k^j = r_k^j + D_k, D_k \leq T_k$, 任务的利用率为 $U_k = C_k/T_k$ 。本文的研究基

收稿日期: 2010-04-14; 修回日期: 2010-06-09。

作者简介: 石林勇(1985-), 男, 广西桂林人, 硕士研究生, 主要研究方向: 实时调度; 晏立(1951-), 男, 江苏镇江人, 教授, 主要研究方向: 实时数据库。

于以下的假设。

1) 任务间相互独立,不存在任何优先约束,不共享资源,任务不阻塞自己,任务切换开销忽略不计或者已经考虑在最坏计算时间中,任务到达即可执行。

2) 允许抢占任务,以及允许任务在不同处理器间迁移,并且抢占开销和任务迁移开销要么可以忽略不计,要么已经考虑在最坏执行时间中了。

3) 最高优先级任务的数量 n' 不多于处理器的个数 m , 即 $n' \leq m$ 。

4) 任意时间 t 为整数,并且表示整个区间 $[t, t+1)$ 。

5) 任务集 τ 中的任务按照优先级非递增(即周期非递减 $T_i \leq T_{i+1}$) 的顺序来编号,这样 τ_1 为最高优先级任务, τ_n 为最低优先级任务。

2 可调度性分析

本文的可调度性判定是基于文献[7]的思路,使得作业 j_k^i 错过它的截止期,所有的 m 个处理器必须执行其他作业超过 $D_k - C_k$ 时间。如果能够说明,对于每一个作业 j_k^i , 任务集都无法在区间 $[r_k^i, d_k^i)$ 形成如此多的任务负载,那么就能保证任务集的可调度性。文献[7]给出了基于这种思想的一个充分必要条件。在给出定理之前先介绍干涉与工作负载的概念。

定义 1^[7] 任务 τ_k 在区间 $[a, b)$ 的工作负载 $W_k(a, b)$ 为按照某种调度策略,任务 τ_k 在区间 $[a, b)$ 执行时间的总和。

定义 2^[7] 任务 τ_k 在区间 $[a, b)$ 受到的干涉 $I_k(a, b)$ 为任务 τ_k 在区间 $[a, b)$ 中就绪可执行,但是由于高优先级任务的缘故无法执行的所有时间片。任务 τ_i 对任务 τ_k 在区间 $[a, b)$ 的干涉 $I_{i,k}(a, b)$ 为任务 τ_k 在区间 $[a, b)$ 中就绪可执行并且 τ_i 正在执行而 τ_k 不能执行的所有时间片。令 $\bar{I}_k$ 表示任务 τ_k 的所有作业中受到最坏情况的干涉,对应的作业记为 j^* 。令 $\bar{I}_{i,k} = I_{i,k}(r_k^{j^*}, d_k^{j^*})$ 。

定理 1^[7] 任务集 τ 在 m 个相同处理器组成的多处理器上是可调度的,当且仅当对于任务集中的每个任务 $\tau_k \in \tau$, 满足条件 $\sum_{i \neq k} \min(\bar{I}_{i,k}, D_k - C_k + 1) < m(D_k - C_k + 1)$ 。

由于没有合理复杂度的方法计算出任务受到的干涉,定理 1 无法直接用来判断任务集是否可调度,但可以使用任务受到干涉的上界来代替。这个上界应该越逼近真实值越好。定理 1 并没有指明使用某种调度算法,也就是说该定理适用于任意的调度算法。下面讨论使用全局 RM 调度策略,且任务集中的任务满足假设 3) 的情况下,任务可能受到的最大干涉。

2.1 全局 RM 调度的工作负载分析

由任务工作负载和干涉的定义可知对于 $\forall i, k, a, b$, 有 $I_{i,k}(a, b) \leq W_i(a, b) \leq b - a$ 。这说明 τ_i 的工作负载为任务 τ_i 对 τ_k 的干涉的上界。下面的问题是如何寻找这样的最大工作负载,即在什么情况下 τ_i 在区间 $[a, b)$ 可获得最大的总执行时间。文献[7]指出, τ_i 的第一个在区间 $[a, b)$ 中的作业尽可能晚地执行,并且在区间开始时刻开始执行;其他在该区间中的作业尽可能早执行,即尽可能多地向区间 $[a, b)$ 内汇聚。 τ_i 在区间 $[a, b)$ 最大工作负载情况如图 1 所示。然而,对于使用全局 RM 策略调度,且任务集 τ 中最高优先级任务个数 n' 不多于处理器数量 m 时,最高优先级的 n' 个任务并不会出

现这种最坏的情况,定理 2 说明了原因。

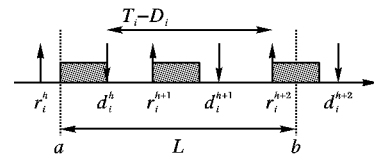


图 1 任务 τ_i 在区间 $[a, b)$ 的最大工作负载的情况

定理 2 在 m 个相同处理能力处理器组成的处理器平台上,使用全局 RM 调度周期任务和偶发任务集 τ , 如果最高优先级任务的个数 $n' \leq m$, 那么这 n' 个最高优先级任务到达就执行,且执行期间不被抢占。即这些任务的作业完成时间为 $f_k^i = r_k^i + c_k^i$ 。

证明 使用反证法来证明。假设 τ_k 为这样一个最高优先级的任务,存在 τ_k 的一个作业 j_k^i 在 r_k^i 到达,然而不能在 $f_k^i = r_k^i + c_k^i$ 完成。RM 是固定优先级算法,任务一旦分配了优先级则它与其他任务间的优先级就不会改变。由于最高优先级任务的个数 $n' \leq m$, 不可能是相同优先级任务的执行而使得 τ_k 等待而不能执行。又因为假设任务自己不阻塞自己,且 RM 是保持工作的,因此说明了存在更高优先级的任务,然而这与 τ_k 是最高优先级矛盾,定理得证。证毕。

定理 2 说明最高优先级的这 n' 个任务是严格周期性执行的,因此不可能出现一个作业尽可能晚执行,其他作业尽可能早执行的最坏情况。下面分别讨论使用全局 RM 调度的这 n' 个最高优先级任务与剩余任务在长度为 L 的区间 $[a, b)$ 中最大的工作负载。

2.2 最高优先级任务的最大工作负载

令 $\tau_{i'}$ 为最高优先级任务, L 为区间 $[a, b)$ 的长度,由于最高优先级任务是严格周期性执行的,那么当 $\tau_{i'}$ 的作业在区间 $[a, b)$ 开始时刻 a 到达并执行时,可得到 $\tau_{i'}$ 在该区间中的最大工作负载。在区间 $[a, b)$ 中最多有 $\lfloor L/T_{i'} \rfloor$ 个 $\tau_{i'}$ 的作业位于该区间中。那么 $\tau_{i'}$ 在 $[a, b)$ 中的最大工作负载可由下式表示:

$$\psi_{i'}(L) = \lfloor L/T_{i'} \rfloor C_{i'} + \min(C_{i'}, L - \lfloor L/T_{i'} \rfloor T_{i'}) \quad (1)$$

2.3 非最高优先级任务的最大工作负载

由于任务 τ_i 不是最高优先级的,它的执行期间可能受到更高优先级任务的干涉,那么它就可能出现文献[7]中指出最坏的情况,如图 1 所示,它在区间 $[a, b)$ 的最大工作负载可由下式计算,

$$\varpi_i(L) = N_i(L) C_i + \min(C_i, L + D_i - C_i - N_i(L) T_i) \quad (2)$$

$$\text{其中 } N_i(L) = \left\lfloor \frac{L + D_i - C_i}{T_i} \right\rfloor.$$

2.4 可调度性判定

由于使用全局 RM 调度,任务的优先级相对于其他任务是不变的,最高优先级的任务不会受到其他任务的干涉,根据本文的模型,它们总是可调度的,因此只需要考虑剩下的 $n - n'$ 个非最高优先级任务即可。而非最高优先级任务只会受到优先级不低于它的任务的干涉。使用式(1)和(2)来代替任务 τ_k 受到其他任务对它的真实干涉。下面给出一个具有更紧的干涉界限的充分判断条件,

定理 3 在 m 个相同处理能力的处理器构成的多处理器平台上,使用全局 RM 策略调度任务集 τ , 系统中最高优先级任务的个数 $n' \leq m$, 如果对于任务 $\tau_k \in \tau (n' + 1 \leq k \leq n)$ 满

足条件:

$$\sum_{i=1}^{n'} \min(\psi_i(D_k), D_k - C_k + 1) + \sum_{j \geq n'+1, j < k} \min(\varpi_j(D_k), D_k - C_k + 1) < m(D_k - C_k + 1) \quad (3)$$

那么该任务集 τ 是可调度的。其中 $\psi_i(D_k)$ 与 $\varpi_j(D_k)$ 由式(1)和式(2)给出。

证明 任务 τ_k 受到最高优先级任务 $\tau_i (n' \geq i \geq 1)$ 的干涉满足关系 $\bar{I}_{i,k} = I_{i,k}(r_k^{i*}, r_k^{i*} + D_k) \leq W_i(r_k^{i*}, r_k^{i*} + D_k) \leq \psi_i(D_k)$ 。任务受到其他任务 $\tau_j (k > j \geq n' + 1)$ 的干涉满足关系 $\bar{I}_{j,k} = I_{j,k}(r_k^{j*}, r_k^{j*} + D_k) \leq W_j(r_k^{j*}, r_k^{j*} + D_k) \leq \varpi_j(D_k)$, 那么 τ_k 受到优先级的不低于它的任务的总干涉满足关系:

$$\sum_{i \neq k} \min(\bar{I}_{i,k}, D_k - C_k + 1) \leq \sum_{i=1}^{n'} \min(\psi_i(D_k), D_k - C_k + 1) + \sum_{j \geq n'+1, j < k} \min(\varpi_j(D_k), D_k - C_k + 1) < m(D_k - C_k + 1)$$

由定理1可知该任务集 τ 是可调度的。 证毕。

定理3给出的判定条件只要求判定 $n - n'$ 个非最高优先级任务是否满足条件,而不是对所有的 n 个任务都进行判定,因此提高了计算性能。此外它针对 RM 调度的最高优先级任务的执行特征,使用比文献[7]给出的干涉上界更逼近于真实值的干涉上界来进行判断,理论上可检测到更多的可调度任务集。

3 实验与结果分析

实验按如下方式来随机生成任务:任务利用率服从期望为 $\sigma_u = u$ 的指数分布, u 为任务的平均利用率;周期服从 $[1, 2000]$ 上的均匀分布,截止期服从 $[0, T_u]$ 上的均匀分布。按如下过程进行实验:首先生成一个任务总利用率较小的任务集 τ' 并检验该任务集是否满足文献[11]给出的可行任务集的必要条件,若满足则使用给出的方法对其进行可调度性测试,然后向 τ' 中添加任务以增大其总利用率,得到一个新的任务集并重复上一步的检验和测试;如果不满足文献[11]给出的条件说明该任务集是不可行的,那么重新抽取任务,并重复上面的过程直到生成 50000 个任务集时结束。将本文提出的判断方法与文献[7]提出的 FP 策略的可调度判定方法(简记为 B-FP)进行了比较。如图2显示了 $m = 4, \sigma_u = 0.3$ 的实验结果,横坐标表示任务集总利用率,纵坐标表示检测为可调度的任务集数量。

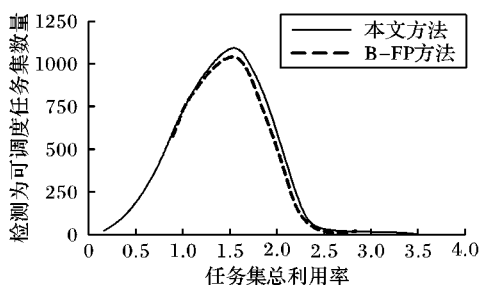


图2 $\sigma_u = 0.3, m = 4$ 的实验数据

此外,还使用 $m = 2, 8, 16$, 以及 $\sigma_u = 0.5$ 重做了实验,得到的结果与图2类似,因此不再给出这些的结果。由图2可以看

到,本文方法和 B-FP 方法检测到的任务集数量比较接近,本文方法检测到的稍微多些。由于本文的实验是按优先级严格单调调度的,也就是说最高优先级任务只有1个,因此显示不出更明显的优势。而且两者都是使用最坏的干涉来代替任务实际受到的干涉,而任务实际受到的干涉并不是如此的悲观,这可能造成了很多的可调度的任务集被判为不可调度。

4 结语

本文研究了相同处理能力的处理器组成的多处理器平台上,使用全局 RM 策略调度周期任务和偶发任务的问题。对于系统中最高优先级任务个数小于处理器数量的情况,得到了一个更逼近真实值的干涉上界,从而得到了更紧迫的充分判断条件,实验结果也显示本文方法提高了检测为可调度的任务集数量。当使用不同的调度策略时,可根据具体调度策略的特征进一步缩小受到的干涉上界,这有助于增大检测到的可调度任务集数量。

参考文献:

- [1] GOOSSENS J, FUNK S, BARUAH S. Priority-driven scheduling of periodic task systems on multiprocessors [J]. Real-Time Systems, 2003, 25(2/3): 187-205.
- [2] BARUAH S K, GOOSSENS J. Rate-monotonic scheduling on uniform multiprocessor [J]. IEEE Transactions on Computers, 2003, 52(12): 966-970.
- [3] BAKER T P. Multiprocessor EDF and deadline monotonic schedulability analysis [C]// RTSS 2003: Proceedings of the 24th IEEE International Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 2003: 120-129.
- [4] BAKER T P. An analysis of fixed-priority schedulability on a multiprocessor [J]. Real-Time Systems, 2006, 32(1/2): 49-71.
- [5] BERTO GNA M, CIRINEI M, LIPARI G. Improved schedulability analysis of EDF on multiprocessor platforms [C]// ECRTS 2005: Proceedings of the 17th Euromicro Conference on Real-Time Systems. Washington, DC: IEEE Computer Society, 2005: 209-218.
- [6] BERTO GNA M, CIRINEI M, LIPARI G. New schedulability tests for real-time tasks sets scheduled by deadline monotonic on multiprocessors [C]// Proceedings of the 9th International Conference on Principles of Distributed Systems. Washington, DC: IEEE Computer Society, 2005: 306-321.
- [7] BERTO GNA M, CIRINEI M, LIPARI G. Schedulability analysis of global scheduling algorithms on multiprocessor platforms [J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 20(4): 553-566.
- [8] BARUAH S, GOOSSENS J. Deadline monotonic scheduling on uniform multiprocessors [C]// Proceedings of the 12th International Conference on Principles of Distributed Systems, LNCS 5401. Berlin: Springer-Verlag, 2008: 89-104.
- [9] CIRINEI M, BAKER T P. EDZL scheduling analysis [J]. Real-Time Systems, 2008, 40(3): 264-289.
- [10] BARUAH S K. Optimal utilization bounds for the fixed-priority scheduling of periodic task systems on identical multiprocessors [J]. IEEE Transactions on Computers, 2004, 53(6): 781-784.
- [11] BAKER T P, CIRINEI M C. A necessary and sometimes sufficient condition for the feasibility of sets of sporadic hard-deadline [C]// RTSS 2006: Proceedings of the 27th IEEE International Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 2006: 178-190.