多核下一种线程调度算法的研究与实现

林 英,孟 正,康 雁,于 倩
(云南大学 软件学院,云南 昆明 650091)

摘 要:随着多核处理器的出现,多核系统线程调度算法成为了一个重要的研究方向,基于DAG表示并行任务在多处理机上进行处理的研究由来已久。文中提出一个基于DAG及Petri网的调度算法,通过把DAG转换为Petri网,希望以直观的方式表达需调度任务的并发、顺序、冲突、同步等关系。该算法充分考虑调度任务之间的并行性,使得并行任务能够并行调度在不同的处理器上,从而有效缩短任务图的调度长度。结果表明,通过有效挖掘Petri网的并行性,能够得到具有较好并行性的任务调度序列,通过合理分配该任务调度序列,可以得到较好的调度性能。

关键词:多核;线程调度;有向图环图;Petri网
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2013)10-0019-04
doi:10.3969/j.issn.1673-629X.2013.10.005

Research and Implementation of a Thread Scheduling Algorithm in Multi-core Environment

LIN Ying, MENG Zheng, KANG Yan, YU Qian
(College of Software, Yunnan University, Kunming 650091, China)

Abstract: With the emergence of multi-core processors, thread scheduling algorithm in multi-core environment has been becoming an important research direction, and the research of the parallel tasks based on DAG in the multiprocessing machine for processing has a long history. By converting DAG to Petri net, proposed a scheduling algorithm based on DAG and Petri net, and hoped to express concurrent, sequential, conflict and synchronization relationships of scheduling tasks in a intuitive way. Through scheduling parallel tasks on different processors, it can effectively shorten the scheduling length. The results showed that through effectively mining the parallelism of Petri nets, can obtain better parallelism task scheduling sequence, by reasonable distribution of the task scheduling sequence, can get a better scheduling performance.

Key words: multi-core; thread scheduling; DAG; Petri net

0 引 言

多核处理器的出现,大大提高了处理器的性能,满足了人们对处理器性能提高的要求,成为了商业化处理器的发展趋势,但也给体系结构、软件、功耗和安全性设计等方面带来了巨大挑战,多核系统线程调度算法就是其中一个重要的研究方向[1]。

调度算法的好坏决定着多核处理器性能的发挥[2],因为如果调度不当,则很有可能在某一时刻,一些内核的负载极重而另外一些内核却极为空闲,进而将并行的优点完全抹煞,甚至比串行的效果还差。一种不当的线程调度算法将会使整个任务执行过程复杂繁琐、效率低下,因此,采取有效的策略去调度线程和平衡负载,已经成为人们的研究热点。

从调度的时机来看,任务调度可以分为静态调度和动态调度。静态调度指调度完全由编译器在编译时决定,程序各个任务运行时间、通信、数据依赖以及同步等在编译时就已知;动态调度则指由调度程序根据运行时情况动态地分配不同的任务到各个处理器,以求尽量降低总运行时间,同时减少程序本身带来的开销[3]。静态调度按照调度方法分类,目前有四种比较常见的类型[4],分别是有向无环图(Directed Acyclic Graph, DAG)、分级任务图(Hierarchical Task Graph, HTG)、任务交互图(Task Interaction Graph, TIG)以及Petri网。

在并行处理领域,针对DAG来表示并行任务在

收稿日期:2012-12-12 **修回日期**:2013-03-17 **网络出版时间**:2013-05-09
基金项目:云南省教育科学研究基金项目(2010Y250,2012C108)
作者简介:林 英(1973-),女,云南人,副教授,CCF会员,研究方向为信息安全、软件工程。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20130509.1100.056.html

多处理机上进行处理的研究已经由来已久[5]。基于 DAG 的任务调度主要有两种方法,分别是基于表调度(list)的方法以及基于聚集(clustering)的方法,也有很多研究是把两种方法结合起来[4]。ISH,ETF,MCP 等都是基于表调度的算法,EZ,LC,MD,DSC 等都是基于聚集的调度算法。Petri 网由于具有图形化表达的形式语义、基于状态的流程描述方式及丰富的模型分析方法,又具有严格的数学基础,能自然地描述并发、同步、资源冲突等系统特性,并且自含执行控制,也常被用来建模任务的调度问题[6],特别是用来解决对柔性制造系统(FMS)的调度[7-8]。文中研究的主要思路就是在 DAG 的基础上,通过把 DAG 转换为 Petri 网,希望以直观的方式表达需调度任务的并发、顺序、冲突、同步等关系,并以此展开任务调度方法的研究。实验结果表明,该方法通过有效挖掘 Petri 的并行度,能够得到具有较好并行性的任务调度序列,通过合理分配该任务调度序列,可以得到较好的调度性能。

1 相关定义

定义 1:DAG 可以表示为一个二元组, $G=(V,E)$。

●节点 $V=\{v_1,v_2,\cdots,v_i,\cdots\}$ 表示图中顶点 v 的集合,v 在 DAG 中表示任务,文中用 v_i 来表示线程 i,其中 v_i 的权重 $w(v_i)$ 表示线程 i 的计算开销,$|V|$ 表示线程的个数;

● $E=\{e_{ij}\mid v_i,v_j\}\subseteq V\times V$ 表示图中有向边 e 的集合,e 在 DAG 中表示任务之间的通信和依赖关系,$|E|$ 表示边的个数。由于文中主要考虑的是片上多核环境下的线程调度问题,因此假定通信开销为零,所以主要用 e 来表示线程之间的依赖关系。边 (v_i,v_j) 中,前一个节点称为父节点,后一个节点称为子节点。一个 DAG 的优先关系体现在一个节点不能在获得它的父节点的信息之前执行。

●DAG 中没有父节点的节点称为入口,没有子节点的称为出口。

定义 2:(Petri 网[9]):三元组 $N=(P,T;F)$ 称为 Petri 网的充分必要条件时满足:

● $P=\{p_1,p_2,\cdots,p_n\}$ 是一个有限库所集(place set);

● $T=\{t_1,t_2,\cdots,t_n\}$ 是一个有限变迁集(transition set);

● $P\cap T=\varnothing, P\cup T\neq\varnothing$;

● $F\subseteq(P\times T)\cup(T\times P)$,$F$ 是 N 上的流关系(flow relation),其元素叫弧;

● $\mathrm{Dom}(F)\cup\mathrm{Cod}(F)=P\cup T$。

其中,

$\mathrm{dom}(F)=\{x\in P\cup T\mid \exists y\in P\cup T:(x,y)\in F\}$

$\mathrm{cod}(F)=\{x\in P\cup T\mid \exists y\in P\cup T:(y,x)\in F\}$

在 Petri 网的图形表示中,库所用圆来表示,而变迁则用短的线段或小方块来表示,库所和变迁之间是通过有向弧来连接。

上述所定义的网只是 Petri 网的结构部分,作为一个 Petri 网,还有另一个要素:标识。设 $N=(P,T;F)$ 是一个 Petri 网,映射 $M:P\rightarrow\{0,1,2,\cdots\}$ 称为网 N 的一个标识(marking)。四元组 $(P,T;F,M)$ 称为一个标识网(marked net)。

时延 Petri 网(Timed Petri Nets,TdPN)是在基本 Petri 网的变迁节点引入表征时间的变量,是由 C. Ramchandani 最先提出[10-11]。它不仅能描述系统在逻辑层次上的关系,而且能够适度地表征系统在时间层次上的关系。时延 Petri 网定义如下:

定义 3:$\mathrm{TdPN}=(\mathrm{PN},D)=(P,T;F,M,D)$

其中,PN 是标识网; $D=(D_1,D_2,\cdots,D_m)$ 为一组与变迁节点相联系的时间参量。通常时间 Petri 网中的各个时间参量 $D_i(i=1,2,\cdots,m)$ 为一组确定性的常数。

定义 4(库所不变量)[9]:设 $N=(P,T;F)$ 是一个网,$|P|=m$,$|T|=n$,$\boldsymbol{D}$ 是网 N 的关联矩阵。如果存在非平凡的 m 维非负整数向量 $\boldsymbol{X}$ 满足 $\boldsymbol{DX}=0$,则称 X 为网 N 的一个库所不变量,即 P 不变量(P-Invariant)。

2 一种多核处理器下线程调度方法

2.1 DAG 转换为时延 Petri 网

DAG 转换为时延 Petri 网的算法如下:

算法 1:DAG 转换为时延 Petri 网。

输入:DAG 表示为 $G=(V,E)$,其中假设节点 V 已拓扑排序,即入口节点为 v_1,出口节点为 v_n,其中 $n=|V|$;

输出:时延 Petri 网 $\mathrm{TdPN}=(P,T;F,M_0,D)$。

BEGIN

$P=\varnothing$; //初始化

$F=\varnothing$; //初始化

WHILE $(v_i,v_j)\in E$ 且未标记 DO

BEGIN

$t_i=v_i$;//把任务转换为变迁

$t_j=v_j$;

$T=T\cup t_i\cup t_j$;

$P=P\cup p_{ij}$;

$F:=F\cup\{(t_i,p_{ij}),(p_{ij},t_j)\}$;

$D(t_i)=w(v_i)$;//节点 v_i 的权重值直接表示为变迁 t_i 的时延值

$D(t_j) = w(v_j)$;//节点 v_j 的权重值直接表示为变迁 t_j 的时延值

标记边 (v_i, v_j) 为“已处理”;

END //END WHILE

$P := P \cup \{p_{start}\} \cup \{p_{end}\}$; //增加起始和结束库所

$F := F \cup \{(p_{start}, t_1)\}$; //连接起始库所与入口变迁

$F := F \cup \{(t_n, p_{end})\}$; //连接出口变迁与结束库所

FOR each $p \in P \wedge p \neq p_{start}$ DO

$M_0(p) = 0$;

$M_0(p_{start}) = 1$; //给出初始状态

END //算法结束

按照上述算法 1,如图 1 所示,将(a)所示的 DAG 图转换后得到(b)所示的时延 Petri 网。

2.2　时延 Petri 网并行性挖掘

文中主要基于 P 不变量来挖掘时延 Petri 网的并行性。根据 FM 算法[12],在不考虑时间因素的情况下,得到图 1 所示时延 Petri 网的 P 不变量如下所示:

$$M(p_{start}) + M(p_{79}) + M(p_{17}) + M(p_{end}) = 1$$

$$M(p_{start}) + M(p_{89}) + M(p_{13}) + M(p_{38}) + M(p_{end}) = 1$$

$$M(p_{start}) + M(p_{89}) + M(p_{14}) + M(p_{48}) + M(p_{end}) = 1$$

$$M(p_{start}) + M(p_{12}) + M(p_{26}) + M(p_{69}) + M(p_{end}) = 1$$

$$M(p_{start}) + M(p_{12}) + M(p_{79}) + M(p_{27}) + M(p_{end}) = 1$$

可以看出,求出的 P 不变量可能有多个,每个 P 不变量在网模型中表示一种可能的分割。如果某些库所不包含在已求得的 P 不变量中,例如库所 p_{15},则直接把这些库所以及起始库所 p_{start} 划分成一个子网。每个 P 不变量中非零元素所对应的库所以及这些库所的外延集合构成这个分割的子网,该子网 $N_i = \bigcup_{|p_i \in P| X(i) > 0|} p_i \cup \cdot pi \cup p_i \cdot i = 1, 2, \cdots, m$。虽然可以按照 P_iiii 不变量将网分割成若干个子网,但分割方案是否合理,还需判定子网是否满足一定的条件。

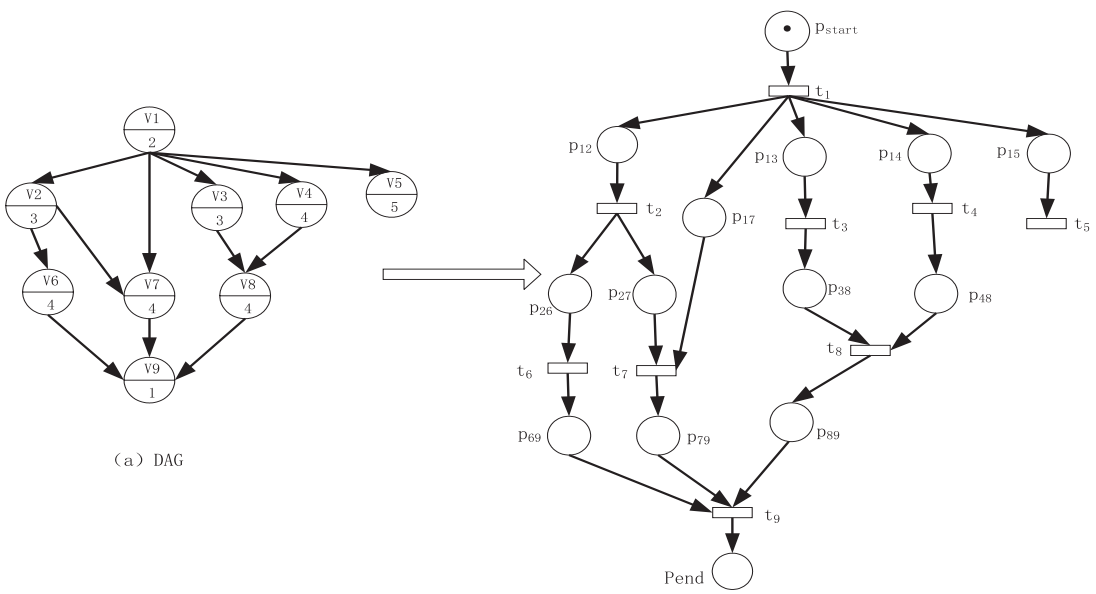
条件 1:子网中至少有一个库所的初始标识不空。该条件保证了从能够划分成的子网是从开始就运行的。

条件 2:分割后可以并行的子网,其所有库所外延集合刚好构成网的变迁集 T。该条件保证了划分后的子网能够覆盖原网。

条件 3:除了开始库所及结束库所 p_{start}, p_{end} 以外,这些子网之间都不存在共享库所。

根据求得的 P 不变量以及以上条件,得到一种子网划分方案如图 2 所示。

在得到划分好的子网后,还可以继续挖掘子网内部变迁之间存在的并行性,例如变迁 t_6 和 t_7 当变迁 t_2 实施后可并行,变迁 t_3 和 t_4 当变迁 t_1 实施后可并行。针对每一个子网,我们生成相应的调度序列,例如,子网 1 产生的调度序列为 t_1, t_2, t_6, t_7, t_9;子网 2 产生的调度序列为 t_1, t_3, t_4, t_8, t_9;子网 3 的调度序列为 t_1, t_5。这些子网之间由于可以完全并行,因此在把这些序列分配到不同处理器的时候,可以并行进行,而不需要考虑前后之间是否还存在依赖关系。



(a) DAG

(b) 转换后得到的时延Petri网

图 1　DAG 及转换后得到的时延 Petri 网

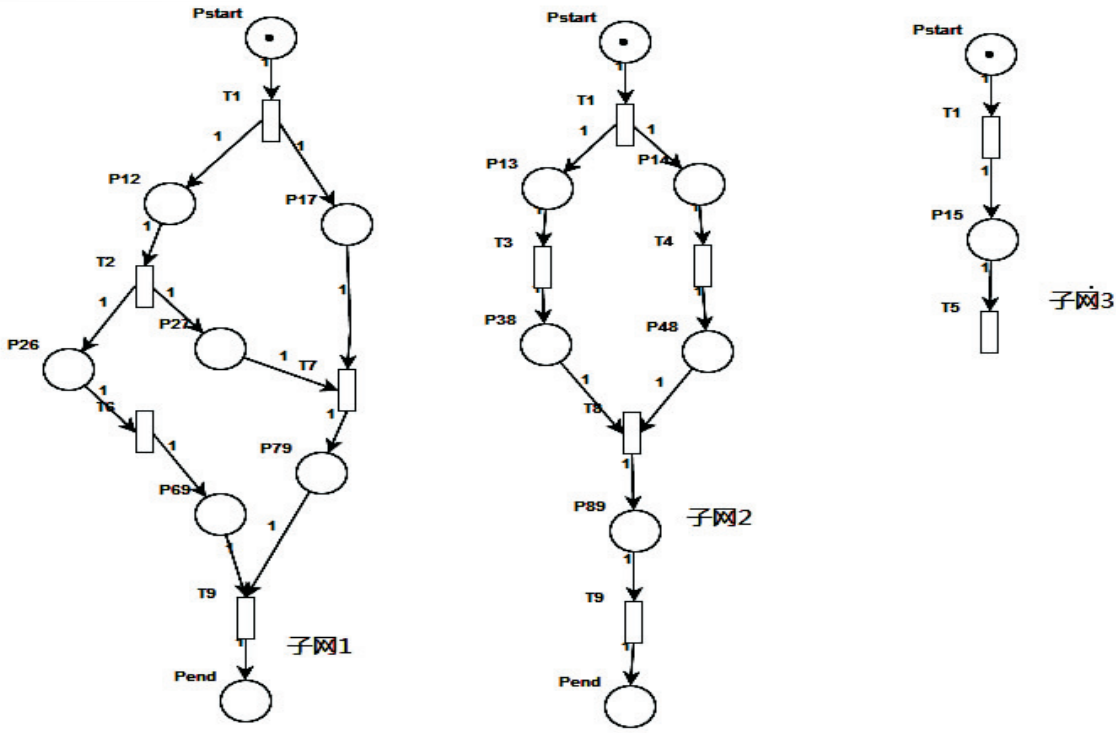


图 2 一种子网划分方案

2.3 处理机分配

对于产生的调度序列,处理器分配遵循以下规则:

(1)对于调度序列上的节点 n, 只要成为就绪节点,就分配到处理器,如果该节点与后一节点存在并行关系,继续调度下一节点到另一处理器,否则转入调度其他序列;

(2)对于调度序列中的一些节点,如已被调度,则继续查询下一节点。调度序列的查询循环依次进行,直至所有的节点都调度完毕。

图 3 给出调度过程示意图。由图 3 可知,其调度长度为 11。与传统的调度算法 ISH、ETF 相比,具有相同的调度长度。

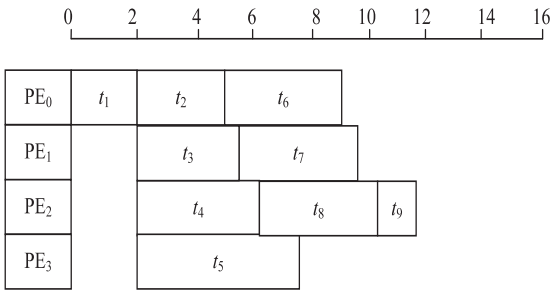


图 3 模型线程调度结果

3 结束语

文中提出一个基于 Petri 网的调度算法。该算法充分考虑调度任务之间的并行性,使得并行任务能够并行调度在不同的处理器上,从而有效缩短任务图的调度长度。文中提出的DAG转换为Petri的方法,能够为形式化分析任务调度打下基础,并且通过基于 P 不变量来划分 Petri 网,通过有效减少 Petri 网的搜索空间,能够为目前很多基于构造 Petri 网可达图的启发式搜索算法提供一种有效地解决状态空间爆炸的研究思路。

参考文献:

[1] Intel Inc. 多核白皮书[EB/OL]. 2007-03-20. http://www.intel.com/multi-core/.

[2] 章 军. 分布式内存多处理机上并行任务静态调度[D]. 北京:中国科学院,1999.

[3] 吴佳骏. 多核多线程处理器上任务调度技术研究[D]. 北京:中国科学院,2006.

[4] 袁 云. 基于多核处理器并行系统的任务调度算法研究[D]. 上海:华东师范大学,2008.

[5] 华强胜. 基于 DAG 模型的高效并行任务调度算法研究[D]. 长沙:中南大学,2004.

[6] 韩 咚,陈 波. 基于时间 Petri 网的多处理机的调度算法[J]. 计算机技术与发展,2007,17(6):15-17.

[7] 杨世强,张海峰,李德信. 基于 Petri 网的 FMS 物流系统建模与仿真[J]. 计算机工程与应用,2008,44(22):226-228.

[8] 黄 波,赵春霞,孙亚民. 基于 Petri 网与动态加权启发策略的 FMS 调度优化[J]. 南京理工大学学报(自然科学版),2010,34(4):482-486.

[9] 袁崇义. Petri 网原理[M]. 北京:北京电子工业出版社,1998.

[10] Holiday M A, Venon M K. A generalized timed Petri net model

(下转第 26 页)

pensateRate 在 0.01 以下,如图 3 所示。

表 2 测试的服务及相应指标值(续)

服务名称	ResponseSpeedOf CustomerService	ConfirmTime OfOrder	NumberOf Hotel	Reputation
BookHotelServiceA	fast	1 ~ 2 天	15 000	9
BookHotelServiceB	very fast	1 ~ 3 天	10 000	8
BookHotelServiceC	fast	1 ~ 3 天	12 000	10
BookHotelServiceD	general	1 天	11 000	9
BookHotelServiceE	fast	1 ~ 2 天	8 000	10
BookHotelServiceF	fast	1 ~ 3 天	9 000	9
BookHotelServiceG	fast	1 ~ 4 天	15 000	8
BookHotelServiceH	fast	1 ~ 2 天	6 000	9
BookHotelServiceI	general	1 天	15 000	7
BookHotelServiceJ	very fast	1 ~ 2 天	10 000	9

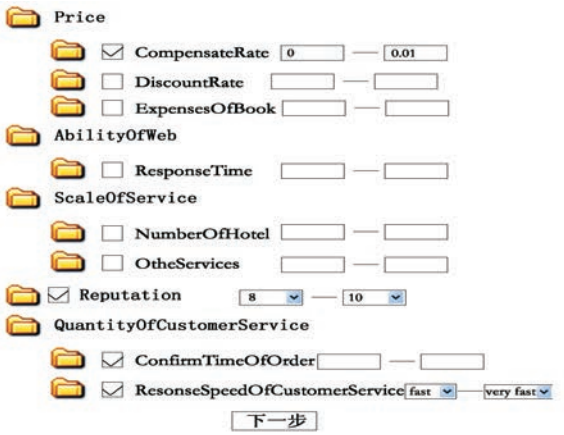


图 3 服务查找的条件输入

经查找后的候选服务如表 3 所示。经检验,查找的结果完全符合服务请求者的要求。

表 3 查找后的候选服务

服务名称	ResponseSpeedOf CustomerService	ConfirmTimeOf Order	Compensate Rate	Reputation
BookHotelServiceA	fast	1 ~ 2 天	0.01	9
BookHotelServiceB	very fast	1 ~ 3 天	0	8
BookHotelServiceC	fast	1 ~ 3 天	0.01	10
BookHotelServiceE	fast	1 ~ 2 天	0.01	10
BookHotelServiceF	fast	1 ~ 3 天	0.005	9
BookHotelServiceJ	very fast	1 ~ 2 天	0	9

5 结束语

文中提出了一个改进的面向 Web Service 的服务质量评价系统模型,并重点研究了基于 Web Service 服务质量的查找方法,使用户可实现个性化和高效率的查找。当用户查找到多个满足条件的记录后,还需要通过服务排序处理从中选择最适合的服务进行调用。

参考文献:

[1] 徐 辉,曹 健. 面向 Web Service 的服务质量评价系统[J]. 微型电脑应用,2010,26(2):1-3.

[2] Tsalgatidou A, Athanasopoulos G, Pantazoglou M. Semantically enhanced discovery of heterogeneous services[J]. IFIP International Federation for Information Processing, 2005, 188: 275-292.

[3] Huang A F M, Lanb C W, Yanga S J H. An optimal QoS-based Web Service selection scheme[J]. Information Sciences, 2009, 179(9): 3309-3322.

[4] Paolucci M, Kawamura T, Payne T R, et al. Semantic matching of Web Services capabilities[C]//Proceedings of the First International Semantic Web Conference on Semantic Web. London: [s. n.], 2002: 333-347.

[5] 高亚春,张为群. 基于 QoS 本体的 Web 服务描述和选择机制[J]. 计算机科学,2008,35(12):273-276.

[6] Xu Ziqiang. Reputation-enhanced Web Services discovery with Qos[D]. Ontario, Canada: Queen's University Kingston, 2006.

[7] 杨胜文,史美林. 一种支持 QoS 约束的 Web 服务发现模型[J]. 计算机学报,2005,28(4):589-594.

[8] 宋顺林,殷荣网. 一种支持 QoS 约束的 Web 服务质量模型[J]. 江苏大学学报(自然科学版),2006,27(5):450-453.

[9] 许 斌. 基于领域的 Web 服务查找方法[J]. 计算机工程,2006,32(20):33-34.

[10] 郑晓霞,王建仁. 基于 QoS 的 Web 服务发现模型研究[J]. 情报科学,2007,25(2):249-253.

[11] 梁 泉,王元卓. 网络计算环境下 QoS 偏好的处理策略及其应用[J]. 计算机应用,2009,29(6):1502-1505.

[12] 唐小燕,李 斌. Web 服务集成中基于 QoS 的服务选择[J]. 计算机应用,2006,26(05Z):242-243.

(上接第 22 页)

for performance analysis[J]. IEEE Trans on Software Eng, 1987, SE13(12): 1297-1310.

[11] Ramchandani C. Analysis of Asynchronous Concurrent Systems by Timed Petri Nets[D]. Cambridge: Massachusetts MIT, 1974.

[12] Martinez J M S. A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri Nets[C]//Proceedings of Second European Workshop on Application and Theory of Petri Nets. Berlin: Springer Publishing Company, 1982.

多核下一种线程调度算法的研究与实现

作者： 林英， 孟正， 康雁， 于倩， LIN Ying， MENG Zheng， KANG Yan， YU Qian
作者单位： 云南大学 软件学院,云南 昆明,650091

刊名： 计算机技术与发展

ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2013(10)