

一种应用于多媒体通信的实时调度算法

陈 向, 雷 航

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】随着网络技术和计算技术的发展,多媒体通信作为一种重要的应用领域,获得了越来越多的应用,而其中一个重要研究主题就是实时调度的效率。传统的实时调度算法有着良好的理论模型,但是由于忽略了许多系统开销,其性能有待提高。该文针对多媒体通信的周期性和实时性提出了一种延迟抢占的策略,与传统立即抢占的模型相比,取得了更好的实际效果,提高了CPU的利用率。

关键词 延时抢占; 多媒体通信; 抢占策略; 实时调度; TCP/IP

中图分类号 TP302 文献标识码 A

A Real-Time Scheduling Algorithm with Delayed-Preemption Feature for Multimedia Communication

CHEN Xiang, LEI Hang

(School of Computer Science and Engineering, Univ. of Electron. Sci. & Tech. of China Chengdu 610054)

Abstract With the rapid deployment of multimedia applications, the need for real-time scheduling in embedded system continues to increase. However, the strict preemptive nature of traditional scheduling algorithms such as Rate-Monotonic (RM) and Earliest-Deadline-First (EDF) leads to inefficiencies specially in application to schedule protocol processing at high speeds or multimedia encode/decode computation. This paper presents a modified RM scheme called RM with Delayed Preemption (DP-RM). The analysis shows that our scheme solves some scheduling efficiency problems.

Key words delayed-preemption; multimedia-communication; preemptive policy; real-time scheduling; TCP/IP

随着无线和有线通信技术的迅猛发展,利用不断改善的网络技术作为多媒体数据传送和共享的载体,使得“无所不在的计算技术”这一由许多科学家和信息技术从业者所共同提出的概念越来越成为可能。而许多终端载体在现有网络技术条件下如何优化自身的处理能力,最大化地发挥和利用现有的技术成果成为一个重要的研究课题。

多媒体通信的应用主要有以下两个方面的要求:(1) 要求通信和对信息的处理是连续且实时的,网络本身和终端设备必须提供某些技术来保证此类网络多媒体在不可预期的网络和系统中正常运行。(2) 多媒体通信属于对网络速度和设备计算能力反应很敏感的应用,因此在传输速度和编解码计算能力方面有很高的性能要求,如何在现有条件下有效地使用计算机资源是非常重要的。

对于多媒体计算,操作系统最重要的支持就是任务的实时调度。现有的实时调度算法,无论是基

于静态优先级的如比率单调(Rate Monotonic, RM)调度,还是给予动态优先级的最早到期优先(Earliest Deadline First, EDF)调度,虽然有很好的理论模型,在一些系统中的使用也取得了良好的效果,但是它们严格实施抢占的本质导致了其被直接运用到多媒体通信系统中的模型不完善。本文主要研究多媒体通信的优化方法。

1 传统实时调度算法分析

多媒体通信系统由于具有实时计算的特点,可以将其归入实时系统。一般来说,实时系统中的任务可以分为周期性的任务(Periodic)和非周期性的任务(Aperiodic)两类。在通常情况下,这两类实时任务可能会共同存在于同一个实时系统中。其中,周期性任务是系统要处理的主要任务,占用处理机的时间较多;而非周期性任务则是系统为了处理一些意外情况或紧急事件所需要执行的任务,占用处理机

时间较少。实时系统的任务调度就是对这两类任务的调度,使它们满足各自的任务时限。本文根据多媒体和网络协议处理的特点,主要考虑的是周期任务的调度。而调度的根本则是抢占正在执行的任务转而执行更高优先级的任务。

抢占多任务带来的代价包括CPU带宽的浪费和内存的总开销^[1]。当一个任务退出运行时,实时操作系统需要保存它的运行现场信息,插入相应的队列,并依据一定的调度算法重新选择一个任务使之投入运行,这一过程所需时间称为任务切换时间。任务切换时间又称为上下文切换时间,是评价一个实时操作系统最重要的技术指标之一。比较著名的商用实时操作系统,如QNX、VxWorks、LynxOS等对上下文切换、中断延迟、获得或释放信号量的延迟的要求都非常小(一般为几微秒),从而能提高实时操作系统运行的及时性和高效性。衡量实时操作系统好坏的主要特点之一就是要求其核心能够支持快速多任务切换、抢占式任务调度等。因此,在保证一定调度性能指标(如截止期错失率)的条件下,尽量减少任务切换是选择调度算法时需要考虑的。

传统的调度算法在理论的基础上(特别是任务的可调度性)都有一个比较好的表现。但是由于它们是建立在任务切换开销、任务间共享资源开销等实际应用环境中,忽略了某些需要考虑因素的情况下得出的,而这部分开销在通信处理过程中占有较大的比例,需要做出某种扩展或是优化。这种对于抢占时机的修改和优化是本文的分析重点,称其为“抢占策略”。

2 DP-RM调度算法及其运行模型

本文的工作主要是针对网络多媒体应用中的协议处理和数据运算。多媒体运算和协议处理针对每个连接实体可以被模型化为周期的活动,在每个周期里需要一些计算和通信的资源。在实现方案中,每个进程的周期活动被实现称为一个实时周期任务(Real-Time Period Task, RTPT)^[2],每一个RTPT有一个相对固定的周期和计算时间,把一个RTPT作为一个可以被调度的基本单元。操作系统在下个周期到来之前做一个可调度性方面的测试,从而判断是否所有的RTPT都可以在它们的死限(Deadline)到来之前完成。内核在每个RTPT周期开始的时候将该RTPT放入运行队列使其可以运行,操作系统选择某个RTPT运行则按照其优先级进行,优先级的判定依据选择的调度策略。调度器使用RM调度策略,即周

期最小的RTPT可以获得更高的优先级。

DP-RM调度算法的关键特点就是充分利用协议处理和多媒体计算循环处理批量连续数据的特性。每个循环处理对应一个RTPT,循环根据其要求有自己的周期,该周期可以被定义为RTPT的周期,而任务的强占只是在每个RTPT一次循环处理结束时发生。在实际应用中,每次循环处理可以被对应于处理一个(或多个)协议数据单元(PDU)或者一帧(或多帧)的多媒体数据。为了实现这种抢占机制,每个RTPT与内核的通信共享内存中的某个数据结构。内核设置这个数据结构中的某个标志从而通知一个运行中的RTPT现在有一个更高优先级的RTPT正等待运行。调度器在每个循环处理的结尾检查与该RTPT对应的数据结构中的相应位是否被设置。如果被设置,则让出CPU的控制权,如果该RTPT在当前周期所需要的处理还没有结束,那么调度器将会把它重新放入可运行队列,当该RTPT的优先级再次满足调度条件时将会再次得到运行。由于一个运行中的RTPT可以延迟被强占直到该次循环的末尾,而基本的调度策略是基于RM的,因此称这种延迟强占(Delayed Preemption)的RM调度算法为DP-RM。DP-RM调度算法有两方面的优点:

(1) 既然RTPT的每次运行都至少完成一个循环的处理操作,采用DP-RM调度发生抢占的次数必定比立即强占的策略要少。事实上,DP-RM通过减少这种非自愿的任务切换次数提高了CPU的利用率。

(2) DP-RM并不立即抢占一个RTPT的运行。RTPT运行实体并不需要对仅仅是在自身循环体内使用的共享数据进行保护锁操作。在实时系统中,操作系统需要对锁操作进行特殊的某些处理,从而预防优先级反转(Priority Inversion)^[3]的发生。而每个锁操作也是一个系统调用。对于Pentium系列的机器运行BSD实现方式的TCP/IP协议,一个系统调用的开销大约是770处理器时钟周期(280指令周期)^[4],而每个PDU需要对某些共享变量进行操作,所以保护锁操作实际上需要针对每个PDU进行。既然每个PDU的处理仅需花费几百条指令,那么对于每个PDU的保护锁获得和释放操作所带来的开销将大量增加对于每个PDU的处理时间。另外,由于单位时间需要处理PDU的数目随着传输率的提高线性增加,导致了并发控制的开销也会线性增加,这对于高速的多媒体通信的影响是非常明显的。对于RTPT的实现,借鉴了文献[2,5]的实现方式,包括安全问题、系统调用接口、其他的效率问题、还有协议的

性能测试方法等。

由分析可以得知DP-RM提高了OS的运行效率,因为:(1)它减少了针对每个PDU所需的保护锁的系统调用开销。(2)减少了任务上下文切换的次数。

由于DP-RM修改了任务的抢占策略,所以需要做一个可调度性测试来考虑DP-RM延迟抢占带来的一些问题。

3 相关性分析

本文将周期的活动(一个RTPT或者一个实时线程)当作一个任务来考虑。一个任务有一个相对固定的计算时间 C 和一个周期 T ,该任务的计算资源利用率即为 C/T ;系统的整个计算资源利用率为所有任务的利用率之和,计为 U 。首先,文献[6]对于传统实时调度算法RM和EDF的分析,RM对拥有最小周期的任务赋予最高的优先级,EDF对距离自身Deadline最近的任务赋予最高的优先级。文献[6]中的分析都是假设在一个单CPU的情况下,任务彼此间独立且任务的Deadline等于自身的周期,同时假定一个高优先级的任务的就绪将会立即抢占当前运行任务的执行。本文称这种调度器为“理想化”的调度器,该调度器假设任务的上下文切换时间为0。

表1 相关的分析结果

模型类型	RM利用率边界值	EDF利用率边界值
理想的抢占模型	$n(2^{1/n}-1)$	1.0
非抢占模型	N/A	Pseudopolynomial test
带阻塞的抢占模型	$n(2^{1/n}-1)-\max(B_i/T_i)$	$1-\max(B_i/T_i)$

表1总结了不同情况下的RM和EDF调度算法的可调度性测试分析结果,相关的分析推导过程可以参见文献[7]。表中第一行对应于立即抢占的情况,这也是最广泛使用的一种分析模型。每一个表项的值代表为了能够保证所有任务的可调度(所有任务可在自身Deadline到来之前完成)CPU的利用率的上限值。可以看到,EDF能够支持100%的CPU利用率,而RM相比之下只能支持 $n(2^{1/n}-1)$,对于一般的任务数 n ,利用率的值大约为69%。然而,RM由于简单和方便实现,是应用最广的一种调度算法。表中第二行则是针对没有抢占发生的情况,目前尚没有一个可调度性测试针对RM无抢占的情况。针对无抢占的EDF算法,本文借鉴了文献[4]的研究成果。表中第三行则是考虑到抢占阻塞的情况,对RM和EDF的可调度性测试结果。本文的阻塞是一个低优先级任

务阻塞了一个高优先级任务的运行,如文献[8]中由于一个当前应该被抢占的低优先级任务正处在某个临界区而阻塞抢占的情况。与理想调度器(表中第一行)比较可以发现,抢占阻塞降低了CPU的利用率的上限值,减少量等于所有任务 B_i/T_i 的最大值,其中 B_i 为任务 i 被阻塞的时间; T_i 为任务 i 的周期。

但是表1的研究成果不能直接应用到延迟抢占模型(Delayed Preemption)中。理想调度器模型由于假设任务抢占不可被阻塞,因此不能用于本文的分析。而非抢占的模型由于可能会极大地影响任务集的可调度性,其本身的使用就是一个有争议的问题。粗看表中第三行阻塞抢占的例子似乎可以包含延迟抢占模型,但是这种由于系统开销带来的抢占阻塞是建立在阻塞期间,没有任何任务能运行这样一种研究模型,如果将它直接运用到本文的研究模型,实际上是降低了该模型的CPU利用率的上限值。理论上这种低估大约有50%^[5],因此在研究中一定要仔细区分这两种不同的阻塞抢占。本文用Mechanism Blocking来对应于这种系统开销带来的阻塞。抢占策略带来的阻塞称为Policy Blocking,这也是本文区别于前人工作的重要之处。

理论上,由于延迟抢占策略推迟了抢占发生的时间,造成了某些满足可调度性的任务集CPU利用率上限值的下降,但这种分析是基于一种无阻塞调度机制的。在实际情况中,由于延迟调度策略减少了任务上下文的切换次数,与传统的立即抢占策略相比,实质上增加了CPU利用率的上限值,这可用下面的实验来论证。

本文创建了12个进程,其中每个进程包含了6个循环处理线程(RTPT),每个线程的周期依次为40、50、60、70、80、90 ms。考虑两种抢占策略:立即抢占和延迟抢占(只到当前循环运行结束)。每个RTPT的运行时间为0.54 ms,总利用率为:

$$\sum_{i=40,50,\dots,90} (0.54/i) \times 16 = 0.84$$

实验平台是PC386处理器同时运行BSD实现方式的TCP/IP协议栈,软件的时钟频率为1 000 Hz,图1所示为两种抢占策略的完成时间和死限的关系,图1a中负值代表在死限之后还需要得到的处理时间。不难看出,在40、50、60、70、80、90 ms的周期值的情况下,两种抢占策略都是可调度的,即可以在死限到来之前完成运行。但是在90 ms的周期值的情况下,立即抢占被发现不能满足死限的要求,说明由于减小任务上下文切换的时间实际上提高了CPU的利用率,本文的模型获得了良好的效果。

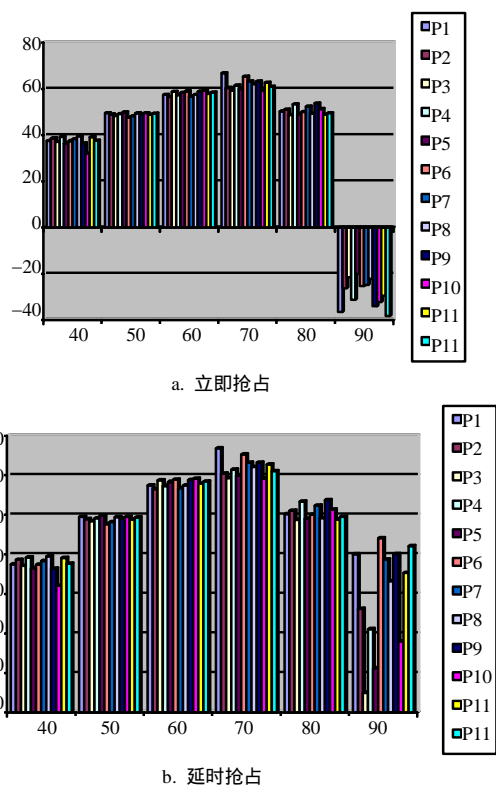


图1 CPU利用率的提高基于延迟抢占

4 结论

本文实现了一个DP-RM的实时调度改进算法,该算法减少了任务切换的时间和保护共享数据结构所带来的开销。为了量化这种优化技术,在PC386/BSD&TCP/IP实验平台模拟了多媒体应用。实验结果显示:基于1 ms的时钟中断,立即抢占方式在某些任务周期下,不能满足实时的可调度条件,而同样的实验却在延迟抢占的情况下获得了良好的

结果(可调度的);DP-RM所带来的延迟抢占的思想对于提高调度效率、满足高速的多媒体通信有着很好的实用价值。不过由于本文的实验建立在仿真的情况下,尚未在具体的多媒体应用中实现,实验结果和分析比较理想化,还有待进一步的分析和研究。

参考文献

- [1] SAKSENA M, WANG Y. Scalable real-time system design using preemption thresholds[C]// In: Jeffay K, ed. Proc. of the 21st IEEE Real-Time Systems Symp. Los Alamitos: IEEE Computer Society Press, 2000: 25-34.
- [2] GOPALAKRISHNAN R, PARULKAR G M. Real-time upcalls: a mechanism to provide real-time processing guarantees[C]// Tech. Rep. WUCS-95-06. St.Louis: Washington University, 1995: 88-102.
- [3] SHA, L, RAJKUMAR, R, LEHOCZKY J P. Priority inheritance protocols: an approach to real-time synchronization[J]. IEEE Transactions on Computers, 1990, 39(9): 34-48.
- [4] CHEN J B, YASUHIRO E, KEE C. The measured performance of personal computer operating systems[C]// 15th ACM SOSP. Colorado, United States: Copper Mountain, 1995, 169-173.
- [5] GOPALAKRISHNAN R, PARULKAR G M. A real-time upcall facility for protocol processing with QoS guarantees[C]// (Poster) 15th ACM SOSP, Colorado. United States: Copper Mountain, 1995: 112-130.
- [6] LIU C L, LAYLAND J W. Scheduling algorithms for multiprogramming in a hard-real_time environment[J]. JACM, 1973, 20(1): 12-20.
- [7] KRISHNA C M, KANG G S. Real-time systems[M]. New York: McGraw-Hill Companies, Inc, 1997.
- [8] KATCHER D, STROSNIDER A H. Engineering and analysis of fixed priority schedulers[J]. IEEE Transactions on Software Engineering, 1993, 19(9): 920-934.

编辑 漆蓉