



用于多媒体的最早截止时间优先算法改进

马永军, 杨帆

(天津科技大学计算机科学与信息工程学院, 天津 300222)

摘要: 对经典实时调度算法之一的最早截止时间优先算法进行研究, 提出一种基于动态 deadline 时间的改进方法, 避免了瞬时过载导致的系统不可预测的行为. 实验表明, 本方法保证了连续多媒体应用的服务质量, 达到较高的资源利用率, 使该算法更加适合多媒体应用环境.

关键词: 多媒体; 实时系统; 任务调度; 服务质量; 最早截止时间优先算法

中图分类号: TP397 **文献标志码:** A **文章编号:** 1672-6510(2012)04-0065-04

Improvement in the Earliest Deadline First Scheduling Algorithm for Multimedia Systems

MA Yongjun, YANG Fan

(College of Computer Science and Information Engineering, Tianjin University of Science & Technology, Tianjin 300222, China)

Abstract: A research on EDF (earliest deadline first) algorithm was conducted, which is in the realm of classical real-time scheduling algorithm. An improved method based on the dynamic deadline time was presented to decrease the number of starved tasks in order to avoid a transient overload which can lead to unpredictable behavior of the system. Experiments show that the quality of service for multimedia applications can be guaranteed in this way. The system can achieve higher resource utilization, and this algorithm is more suitable for streaming media applications.

Key words: multimedia; real-time operating system; task scheduling; quality of service; earliest deadline first algorithm

随着计算机网络和多媒体技术的发展, 实时多媒体系统已应用到社会各个方面. 系统中主要包括两种实时性任务, 即以实时控制为代表的硬实时任务和以多媒体为代表的软实时任务^[1]. 各种任务的主要处理对象是多媒体信息, 其在采集、传输和处理器调度方面都有着实时性的要求.

最早截止时间优先 (EDF, earliest deadline first) 算法是 20 世纪 80 年代提出的, 适用于实时系统的动态优先级调度算法, 它主要用在硬实时系统. EDF 在多媒体任务调度时, 优先级是由截止时间动态决定的, 截止时间越短, 优先级就越高^[2]. 在这种系统中, 要求所有任务集的任务负载 ≤ 1 , 否则系统行为是不可预测的, 可能发生多米诺骨牌现象, 最终导致系统

崩溃. 但对于多媒体系统: 一方面, 一定的数据丢失是可以接受的, 所以多媒体系统是软实时系统; 另一方面, 多媒体任务具有突发性, 这导致任务的运行时间变化较大, 任务的周期难以预测. 所以多媒体系统中的任务容易过载, 导致后续多媒体任务连续丢失, 多媒体系统的服务质量得不到保证. 传统的 EDF 已经不适用于实时多媒体系统, 文献提出的悲观方法和乐观方法两种解决方法^[3]也没有考虑到系统的资源利用率和任务 deadline 率, 存在着一定不足.

本文提出了一种改进的 EDF 算法, 即动态 deadline 时间法, 可以有效地避免多媒体系统任务瞬时过载导致的崩溃, 同时资源利用率也能达到要求, 从而使整体上多媒体系统的服务质量得到保证.

收稿日期: 2012-02-20; 修回日期: 2012-03-14

基金项目: 教育部规划基金资助项目 (12YJAZH091); 国家自然科学基金资助项目 (71141019)

作者简介: 马永军 (1970—), 男, 吉林长春人, 教授, yjma@tust.edu.cn.

1 多媒体的服务质量保证和调度算法

1.1 多媒体的质量保证

服务质量^[4](quality of service, QoS)是指用户可见的服务性能特性的集合. 对于多媒体系统, 它包括时间和内容两个方面.

在时间上, 多媒体系统中存在诸多延迟, 包括压缩延时、网络传输延时、解压缩延时及播放同步延时等. 但是多媒体数据具有时间相关性, 它是与时间有关的数据流, 各个数据单元需要在规定的时限内完成处理, 否则就是无效数据. 所以系统对延迟和延迟抖动的时限都有要求.

在内容上, 多媒体系统常常需要处理包括图像、音频、视频和文本等信息, 处理的数据量较大, 即使经过压缩处理后, 数据的信息量仍然很大. 这就对网络带宽提出了很高的要求. 因为媒体内容和压缩技术不同, 经过压缩得到的数据单元大小不一, 这使多媒体数据具有突发性, 需要多媒体系统能够利用较好的调度算法, 合理地分配各种资源, 从而保证多媒体的服务质量.

1.2 传统任务模型和调度算法

当前的研究方法主要是根据连续媒体周期性的特点, 将其作为传统的周期性任务. 当处理多媒体任务时, 主要用周期性的实时任务来描述, 可以表述为以下实时任务调度模型^[5]:

任务调度的目标 S 可以描述为一个周期性任务集合 $S = \{T_1, T_2, \dots, T_n\}$. 集合中单个任务 T_i 表示为 $T_i = (r_i, p_i, c_i, d_i)$. 其中: r_i 为任务开始执行时刻; p_i 为任务周期; c_i 为执行任务实例的时间; d_i 为任务的截止期限时间.

任务在单个周期内的一次执行就是任务的一个实例, 由于多媒体数据流的内容不是固定的, 不同时间的内容不同, 用同一种压缩算法的媒体任务压缩比也不同, 所以最终的数据大小是不一样的, 处理每个数据 CPU 所花费的时间也不同, 所以任务的不同实例^[6-7]的执行时间不同, 即 c 不同. 任务 T 必须在截止期限 d 内完成, 也就是任务的绝对 deadline 在 EDF 算法中是不变的. 通常的多媒体任务 $d=p$, 故第 i 个实例必须在时刻 $r+p_i$ 之前完成, $r+p_i$ 称为该实例的相对 deadline.

在实时领域内这种任务模型得到普遍应用. 在此基础上, 很多基于这种模型的调度算法被扩展出

来. 比较经典的是 EDF, 这个算法易于理解, 算法的性能也比较好.

EDF 算法是优先级抢占式算法, 各个任务的优先级是动态变化的. 调度进程要先按照等待任务的 deadline 时间计算出任务 deadline 序列, 待处理器在就绪状态时, 它的调度进程在任务序列中选择一个距 deadline 最近的任务进入就绪状态, 该任务的优先级为最高. 如果就绪期间没有新任务到来, 处理器执行这个任务, 如果有新的任务到来, 就得重新计算任务的 deadline 序列, 若新任务的 deadline 时间比已就绪的任务短, 新任务就会抢占正在运行的任务, 重新计算被抢占的任务优先级.

另外, 文献[8]还给出了一个任务集合可调度的充分条件:

假设任务集合 $T = \{T_i = (r_i, c_i, p_i, d_i)\}$ 满足以下条件, (1) 任务的运行周期确定; (2) 任务相对 deadline 和任务周期是相同的; (3) 任务相互独立; (4) 任务的运行时间确定, 是一个常量.

对于 EDF 算法, 若任务集合符合式(1), 即 CPU 的利用率 ≤ 1 , 则所有任务是可以有效调度的.

$$\sum c_i / p_i \leq 1, i = 1, 2, 3, \dots, n \quad (1)$$

1.3 传统周期任务模型存在的问题

这种模型对多媒体任务的描述不够精确, 不能简单地利用在多媒体实时系统上. EDF 算法中任务可调度的一个条件是, 任务的运行时间是固定的. 在多媒体实时系统中任务需要处理数据量是随着时间的变化而变化的, 任务的处理时间无法确定, 所以这种系统中任务集合的处理周期是一个变量, 它不是一个固定周期的任务. 在这种情况下, 这种周期模型的可调度条件不能得到满足, EDF 算法也不能简单地应用在多媒体实时系统上.

在悲观方法^[3]中, 把任务的最大可能运行时间当作任务的绝对 deadline, 该方法下降为传统的 EDF, 系统的处理器足够处理任何的任务, 但是大部分时间处理器是空闲的, 造成了处理器资源的浪费, 资源利用率很低.

在乐观方法^[3]中, 把运行时间的平均值当作任务的绝对 deadline, 在这种情况下, 系统中很多任务的运行时间会大于平均值, 在规定的时限内不能完成这些任务, 引起任务丢失, 造成较高的 deadline 丢失率, 满足不了实时性要求. 因此, 直接使用经典任务周期模型, 以处理传统周期性任务的方法处理多媒体任务, 不符合多媒体实时系统特点, 服务质量得不到保证.

2 最早截止时间优先算法的改进

由以上分析中可知, 多媒体应用允许有一定的数据丢失, 但对系统的实时性有严格的要求.

EDF 算法调度周期性任务时, 如果总的任务负载 ≤ 1 , 则该任务集是可以正确调度的; 当任务集的任务负载 > 1 , 则任务大量丢失, 结果不可预测. 在多媒体应用中, 只要任务 deadline 丢失率达到系统需求, 它就是可以被接受的. 因此, 当系统任务负载较大时, 在系统允许的 deadline 丢失率要求范围内, 可以适当地动态改变任务的 deadline 时间, 使任务能得到正常运行. 通过这种方法, 任务的平均 deadline 丢失率得到了保证, 即保证了任务的服务质量. 算法的基本思想是在系统高负载情况下, 为防止部分任务连续失败, 通过计算前段任务的运行时间确定出新的任务 deadline 时间, 以尽量保证任务能正常运行, 从而减少任务的平均 deadline 丢失率.

在多媒体系统中, 每一条多媒体数据流包括视频流和音频流, 都可以作为一个可调度的实体来进行调度, 把每条多媒体数据流当作周期性任务. 一个多媒体流是由多个帧组成的, 把对每帧的处理作为该任务的一个实例. 在指定时间内如果实例完成则任务成功, 否则任务失败.

假设在多媒体系统中有 n 个多媒体任务, 任务都定义为周期性执行. 建立多媒体任务模型如下:

$$T'_i = (r_i, p_i, c_i, d_i, S_i, L_i), i = 1, 2, \dots, n \quad (2)$$

式中: r_i 为任务 T'_i 开始执行时刻; c_i 为任务 T'_i 执行时间; p_i 为任务 T'_i 运行时间, 即周期; d_i 为任务 T'_i 的 deadline 时间; L_i 为任务 T'_i 的实例个数; S_i 为 L_i 个实例中运行成功的个数.

如果 T'_i 实例的完成时间小于或等于 deadline, 则任务成功, 否则任务失败. 定义任务 T'_i 的成功率为 S_i/L_i , 表示为在任务 T'_i 执行的 L_i 个实例中, 成功的个数为 S_i . 多媒体任务的服务质量由 S_i, L_i 决定, 即多媒体任务调度必须达到要求的最小成功率.

定义 α 为系统设定的最小成功率, 不同的多媒体系统对服务质量要求不同, α 可以按照用户需求设定.

在前 n 个任务中如果 $S_i/L_i \geq \alpha$, 下一个任务 T'_{i+1} 的 deadline 时间为

$$d_{i+1} = \frac{1}{n} \sum_{i=1}^n c_i \quad (3)$$

如果前 n 个任务中 $S_i/L_i < \alpha$, 则下一个任务 T'_{i+1} 的 deadline 时间为

$$d_{i+1} = \frac{1}{n} \sum_{i=1}^n c_i + \beta(1 - S_i/L_i)c_i \quad (4)$$

式中: β 为任务执行时间超出平均执行时间的比例大小^[9], 取值范围为 $0 \sim 1$.

下面给出多媒体任务调度执行步骤:

(1) 当任务 T'_i 到达就绪状态时, 根据式 (5) 计算任务的绝对 deadline 时间 D_i , 把任务 T'_i 排在任务就绪队列中.

$$D_i = d_i - c_i \quad (5)$$

(2) 按照 D_i 由长到短的顺序, 对就绪队列进行排列. 在任务就绪队列中, 任务 deadline 最短的任务优先级设为最高, 其他任务按照队列顺序其优先级依次降低.

(3) 选出优先级最高的任务, 等待处理器执行. 如果等待期间没有新任务到来, 转步骤 (4), 如果有新任务到来转步骤 (1).

(4) 处理器执行该任务.

3 仿真测试及分析

对提出的调度策略在 DRTSS 中仿真, 并与传统的 EDF 方法进行了结果比较, 以验证改进的调度策略的有效性. 实验中, 网络周期性地接收到一定的视频数据, 经过解码和演播, 得到 10 条 MPEG-1 视频流, 然后统计出播放 10 条视频流的任务成功率能达到预期值的概率. 考虑到人的视觉的最低分辨率在 20 帧左右, MPEG-1 视频流标准是 25 帧, 设 $\alpha = 0.8$, 由统计得出 $\beta = 0.5$.

分别模拟传统 EDF 算法和动态 deadline 时间法, 比较结果如图 1 所示.

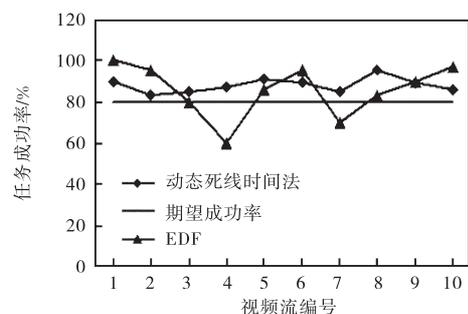


图 1 模拟结果比较

Fig. 1 Comparison of simulation results

图 1 结果显示,采用 EDF 算法的任务中,有 2 个任务的成功率没达到系统要求,不能有效调度. 而基于动态 deadline 时间法调度下,所有任务基本达到了期望成功率,保证了服务质量.

改变实验方法,把模拟的各条视频流任务顺序地加入系统,直到系统达到饱和. 分别模拟 EDF 方法、悲观方法、乐观方法及动态 deadline 时间法,统计同样任务数量的不同调度方法的任务成功率及随着任务的增多,同一调度方法的任务成功率变化情况. 算法性能比较结果见表 1.

表 1 算法性能比较
Tab. 1 Comparison of algorithms

视频流 数量	任务成功率/%			
	EDF	悲观方法	乐观方法	本文算法
1	100	100	100	100
2	100	100	100	100
3	94.0	100	95.1	98.4
4	87.3	93.5	93.2	98.0
5	85.2	89.1	87.3	94.8
6	79.2	42.1	84.7	89.7
7	67.4	25.0	74.5	78.0
8	59.7	54.0	40.4	70.2
9	43.4	10.4	37.5	69.4
10	35.3	7.7	32.4	61.4

由表 1 可看出,在视频流数量相同的情况下,动态 deadline 时间法任务成功率普遍比 EDF 算法要高,随着视频流的增加,算法的任务成功率较 EDF 下降缓慢,同时平均 deadline 丢失率大幅降低,为任务的实时性提供了保证.

与传统 EDF 调度相比,改进算法的任务成功率均达到期望成功率,能调度的任务明显得到提升,改进算法的任务失效率低,能较好地满足多媒体系统所要求的质量保证,更适用于多媒体任务的调度.

4 结 语

本文在对最早截止时间优先算法分析的基础上

提出了以动态 deadline 为概念的改进算法. 理论分析和实验表明,提出的算法部分解决了传统 EDF 在多媒体系统中的任务瞬时过载问题,使系统的服务质量得到保证;同时,在改进算法的调度下,系统最大任务数得到提升,从而使系统资源得到更为充分的利用. 本方法在计算动态 deadline 时会占用少量处理器资源,以后研究方向是尽量减少该部分资源占用,更大地提高资源利用率.

参考文献:

[1] 张占军,韩承德,杨学良. 多媒体应用与操作系统支持[J]. 小型微型计算机系统,2001,22(6):641-643.

[2] Liu C L, Layland J W. Scheduling algorithms for multi-programming in hard-real-time environment[J]. Journal of Association for Computer Machinery, 1973, 20(1): 46-61.

[3] 阮俊波,李红兵,金惠华. 实时连续多媒体任务模型及调度算法[J]. 北京航空航天大学学报,2005,31(8): 863-868.

[4] 王兴伟,张应辉,刘积仁. 分布式多媒体系统服务质量管理机制的研究[J]. 软件学报,1998,9(2):86-90

[5] Finkel R A, Bentley J L. Quad trees: A data structure for retrieval on composite keys[J]. Acta Informatica, 1974, 4(1): 1-9.

[6] 杨学良,张占军. 分布式多媒体计算机系统教程[M]. 北京:电子工业出版社,2002:242-244.

[7] 张炫. 多媒体系统任务调度算法研究[D]. 武汉:华中科技大学,2006.

[8] Leung J Y T, Whitehead J. On the complexity of fixed-priority scheduling of periodic, real-time tasks[J]. Performance Evaluation, 1982, 2(4): 237-250.

[9] 张怡. 分布式多媒体系统 QoS 技术的研究[D]. 长沙:国防科学技术大学,2006.

责任编辑:常涛