文章编号:1001 - 9081(2005)04 - 0889 - 02

# 单调比率(RM)调度算法及应用

叶 明,罗克露,陈 慧
(电子科技大学 计算机科学与工程学院,四川 成都 610054)
(yemingcn@163.com)

**摘 要**:介绍了任务死线不大于其周期的任务集调度条件分析及算法实现。这种约束条件放松,有利于周期与非周期任务混合模型调度。同时,分析了以往调度算法中单调比率调度算法约束条件,并指明了计算时间复杂度的缺点。因而,在 RM 算法基础之上提出一种实时系统调度算法及实现流程图,并对提出的现场级实时调度算法进行了对比测试。

**关键词**:实时系统;单调比率;单调死线;调度理论;调度时间
**中图分类号**: TP393 **文献标识码**:A

# Application of monotonic rate scheduling algorithm

YE Ming, LUO Ke-lu, CHEN Hui
*(College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu Sichuan 610054, China)*

**Abstract:** Scheduling condition and algorithm analysis of task sets which task deadline is not less than its period was introduced. This restrict conditions is slackening, and it is good for scheduling of mixed model of period and no-period task. At the same time, scheduling restrict conditions of original RM algorithm was analyzed and the shortcoming of computing time complexity was pointed out. Then a real-time scheduling approach and implementing flow-chart based on RM algorithm was presented, and a strict contrast test of the real-time scheduling approach was given.

**Key words:** real-time systems; Monotonic Rate(RM); monotonic deadline; scheduling theory; scheduling times

## 0 引言

现场级网络调度算法在实时系统中有着广泛的应用,按消息到达模式来分,这类算法包括周期消息调度算法和非周期消息调度算法两种。周期消息调度算法主要包括单调比率算法 Monontic Rate(RM),截止期最早优先(Earliest Deadline First,EDF)算法,基于距离约束的风车调度算法(DCTS SR),价值密度最大优先(Highest Value Density First)算法。非周期消息调度算法主要包括轮询服务器(Polling Server,PS),优先级交换(Priority Exchange,PE)和延缓服务器(Deferrable Server,DS)三种。由于 RM 实现简单,实际应用面广,故本文重点讨论单调比率调度理论及应用。在实际实时系统中它不但能够满足周期任务调度,同时也是一种有效的非周期任务调度算法。

## 1 单调比率(RM)调度理论

假定调度任务满足如下关系:

$Computation\ time \leqslant Deadline \leqslant Period(T_i)$

对于每一个任务 $\tau_i$($\tau_1$ 优先级最高,$\tau_i$ 优先级最低):

$C_i \leqslant D_i \leqslant T_i$

单调比率任务优先级顺序在概念上同单调死线优先级顺序相似。分配给任务优先级大小同死线长度是成反比的。因此,具有最短死线的任务被分配最高优先级,最长死线任务分配最低优先级。当任务周期等于其死线时,缺省任务优先级顺序同单调死线顺序一样。

如果在时间间歇$[t_0,t_1]$内有任务释放发生,调度会失败。因此,可以建立任务 $\tau_i$ 表明其调度条件的一系列方程如下所示:

(1) $\dfrac{I_i^0}{t_0} + \dfrac{C_i}{t_0} \leqslant 1$

where $t_0 = \sum_{j=1}^{i} C_j$

(2) $\dfrac{I_i^1}{t_1} + \dfrac{C_i}{t_1} \leqslant 1$

where $t_1 = I_i^0 + C_i$

(k) $\dfrac{I_i^k}{t_k} + \dfrac{C_i}{t_k} \leqslant 1$

where $t_k = I_i^{k-1} + C_i$

and where $I_\gamma^x = \sum_{z=1}^{\gamma-1} X_i^z \left\lceil \dfrac{x}{T_z} \right\rceil C_z$

如果任务 $\tau_i$ 满足上述任意一个方程,那么它就是可调度的。显然,如果对每一个任务 $\tau_i$ 和方程 $k$,其 $t_i > D_i$,方程计算就会终止。在这一时间点任务 $\tau_i$ 是不可调度的。

上面一系列方程可以用下面的可调度性判定算法来简化表达。

```
foreach τ_i do
    t = Σ_{j=1}^{i} c_j
    continue = TRUE
    while [continue] do
        IF [ I_i^t / 2 + C_i / t ≤ 1 ]
            continue = FALSE
```

**收稿日期**:2004 - 08 - 30;**修订日期**:2004 - 11 - 27 **基金项目**:四川省科技厅重点科技项目基金资助(02GG006 - 037)

**作者简介**:叶明(1975 - ),男,四川绵阳人,硕士研究生,主要研究方向:实时嵌入式系统、图像处理; 罗克露(1948 - ),女,四川成都人,教授,主要研究方向:实时系统、CAD; 陈慧(1973 - ),女,四川攀枝花人,博士,主要研究方向:实时通信.

```
    /* τi is scheduable */
    else t = I_i^t + C_i
    endif
    if[ t < Di ]
      /* τi is unscheduable */
    endif
  endwhile
endfor
```

## 2 单调比率调度算法的应用

虽然上面推导出 RM 调度算法的充分必要条件，但该可调度判定算法具有指数时间复杂性，约束条件较强，随着任务数增加，该算法就必须进行大量复杂计算，造成系统的响应时间减慢，实时性能较差。同时，在单调比率调度方法中用非周期服务器保证其死线的方法主要有两个缺点：1）每一个零星任务要求一个额外的周期服务器任务；2）要求系统内核保持任何周期内剩余时间轨迹。对于任务数较多的一些特定现场网络实时系统，RM 可调度性判定算法并不适用。由此在 RM 调度算法基础之上提出了一种新的实时调度算法（Hard Real-time Communication Scheduler，HRTCS）。该方案的优点是：任务集通过基础判定和仿真判定后，不经运算就可判定该任务集是可调度的，这样就保证了调度的实时性。

HRTCS 算法的基本思想如下：

先收集消息建立待调度消息集合，接着对建立的消息集合进行基础判定、仿真判定，判定通过后利用 RM 调度算法调度生成总线调度表。

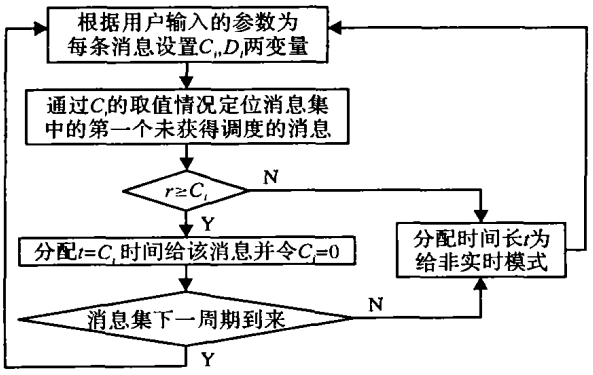


图 1 HRTCS 调度算法流程

由于本文重点研究硬实时调度算法，故在此给出判定通过后该调度算法的重要数据结构，实现流程图如图 1 所示。

```
int RMSchedule( MsgArray * pMsgArray, int MsgCount, BusArray
  * pBusArray, int BusCount)
```

功能：将消息集（原始消息集按照 Freq 从小到大排序）通过调度算法转化生成总线调度表。

参数：pMsgArray 表示指向消息集合数组的指针，MsgCount 表示消息集合的大小，pBusArray 表示总线表数组指针，BusCount 表示总线表数组的大小。

返回值：成功返回值大于零为生成总线表项的数量，PROC_FAIL 调度失败。

数据结构：

```
struct MsgArray
 { int MsgDefID;                    //任务消息定义号
   int PType;                       //任务消息类型
   double ExeTime;                  //任务消息执行时间
   double Freq;                     //任务消息周期
 }
struct BusArray
 { int MessageSerrialNo;            //消息序列号
   int MsgDefNo;                    //消息定义号
   double Starttime;                //消息开始时间
   double Endtime;                  //消息结束时间
 }
```

## 3 仿真分析

### 3.1 HRTCS 调度加速比与调度时间开销

为了评估 HRTCS 算法在不同负载下的性能，对算法在任务集任务数 $n$ = 50，100，200，300，400，500 情况下，分别测试了其加速比。从图 2（a）不难看出任务负载不大时，采用 HRTCS 调度任务，其调度性能表现相当优异。同时对任务集任务数 $n$ = 20，40，60，200，分别测试了其调度时间（单位/μs）。从图 2（b）也能分析得出随着任务集任务数不断变大，该调度算法调度时间开销也依次增加。

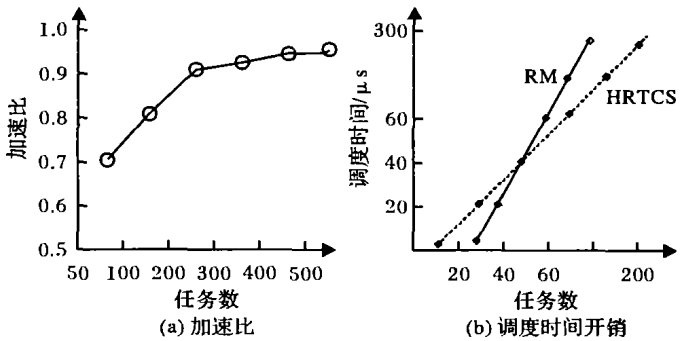


图 2 加速比和调度时间开销

### 3.2 HRTCS 可调度性判定时间开销

从图 3 可以看出，采用 HRTCS 进行在线可调度性判定并生成总线调度表，即使一个包含大量消息的消息集，如果仅存在在线的仿真判定开销，那么该开销是可以接受的。

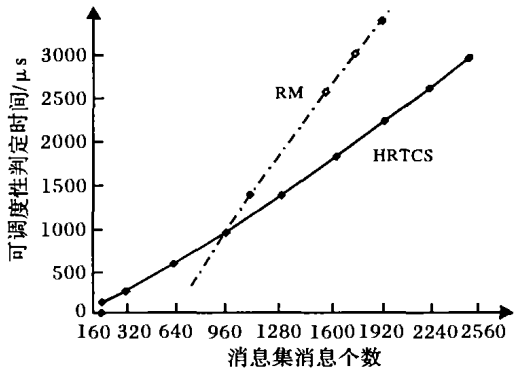


图 3 可调度判定时间开销

## 4 结语

为了允许死线低于周期的任务能够被调度，讨论推导了单调比率任务调度的基本约束条件，降低这种约束条件进而推动单调比率调度理论应用与发展。本文讨论了任务在什么情况下是可调度的。

首先推论了一系列可调度条件方程。为了决定任务可调度性，每一个任务均需要一个简单方程对其判断。这就决定发展一个充分而必要的可调度性条件。与任务周期和计算时间相关的复杂性给这种方法也带来复杂问题。当决定任务可调度性约束的方程数量减少时，该方法的复杂程度也相应降低。无论该任务集优先级分配规则如何，这种可调度性测试能够决定任何固定优先级任务集的可调度性。

文章编号:1001 - 9081(2005)04 - 0891 - 03

# 一种改进的网络拓扑发现方法

邱建林[1],何 鹏[1,2]
(1. 南通大学 计算机科学与技术学院,江苏 南通 226007;
2. 苏州大学 计算机科学与技术学院,江苏 苏州 215006)
(qiu. jl@ ntit. edu. cn)

**摘 要**:在对基于 ICMP 的网络拓扑发现、基于 ARP 的网络拓扑发现和利用 SNMP 访问 MIB(管理信息库)路由表的网络拓扑发现三种方法的分析研究基础上,提出了一个经过改进的网络拓扑发现方法,此方法能够准确、完整、高效地发现网络主干拓扑和子网内的设备,并详细描述了网络拓扑发现的数据结构和算法。

**关键词**:拓扑发现;简单网络管理协议;管理信息库;ICMP,地址解析协议
**中图分类号**: TP393.02 **文献标识码**:A

# Ameliorate algorithm for network topology discovery

QIU Jian-lin[1], HE Peng[1,2]
(1. *College of Computer Science and Technology, Nantong University, Nantong Jiangsu* 226007, *China*;
2. *College of Computer Science and Technology, Suzhou University, Suzhou Jiangsu* 215006, *China*)

**Abstract:** Three kinds of centralized automatic discovery algorithms for network topology were introduced, which include building network toplogy base on ICMP or ARP protocol and utilizing SNMP protocol to visit in MIB( management information base) to construct network toplogy. Then, a kind of better network toplogy algorithm was proposed based on the three algorithms which can discover network topology accurately, unabridged and efficiently. And the data structure and steps of this algorithm was described detailedly.

**Key words:** toplogy discovery; SNMP; MIB; ICMP; ARP

网络的拓扑结构发现是网络管理的基础,对监视整个网络,获取完整的网络信息,保证网络高效、稳定地运行非常重要。

本文对现今流行的几种网络拓扑发现技术进行了分析,对这些技术的优缺点和适用环境进行了一些探讨,介绍了一种基于 ICMP、ARP、SNMP 改进的网络拓扑发现方法,并给出了实现模型。

## 1 基于 ICMP 路由表的拓扑发现方法

ICMP[1] 是提取路由信息的重要工具,人们常利用ICMP echo reply 消息来检测网络设备的活动状态和可达性。利用 ICMP time exceed 和 port unreachable 消息,以及 IP 协议中 TTL 字段可以发现给定主机的路由信息。如果对一个网段内所有可能的 IP 地址依次执行"Ping"操作,根据应答就可以发现该网段内所有当前活动的设备。然后对"Ping"通的 IP 地址逐一执行"Traceroute"操作,记录每一次操作的结果,根据前面的操作结果,再分析所得到的信息,从而得到整个网络拓扑的情况。

基于 TCP/IP 协议的网络设备几乎都支持所有的 ICMP 协议,该方法实现起来较容易。在发现网络拓扑的同时,检测

**收稿日期**:2004 - 09 - 06 **基金项目**:江苏省高校自然科学基金资助项目(03KJB520103);江苏省现代教育技术资助项目(2004 - METR - 27);南通市科委资助项目(C3018、C3024)

**作者简介**:邱建林(1965 - ),男,江苏南通人,副教授,主要研究方向:算法分析与设计、网络安全、逻辑综合 VLSI 的 CAD; 何鹏(1980 - ),男,江苏南通人,硕士研究生,主要研究方向:计算机网络安全.

但是,RM 可调度判定算法有自身的缺点,在一些特定现场实时系统中该调度算法并不适用,进而我们提出一种新的硬实时调度算法(HRTCS)方案,并对该算法同 RM 进行了严格对比测试。经实验说明该方案系统开销较小、可实施性好,易于移置到其他硬实时通信系统中。

**参考文献**:

[1] TINDELL KW, BURNS A, WELLINGS AJ. Allocating hard real-time tasks: an NP-hard problem made easy[J]. Real-Time Systems, 1992, 4(2): 145 - 165.

[2] SHA JPL, DING Y. The Rate - Monotonic Scheduling Algorithm: Exa Characterization and Average Case Behavior[A]. Proceedings of IEEE Real-Time System[C]. 1989. 166 - 171.

[3] BARUAH SK, MOK AK, ROSIER LE. Preemptively Scheduling Hard-Real-Time Sporadic tasks on One Processor[A]. Proceedings of IEEE Real-Time System Symposium[C]. Orlando, Fla, 1990. 182 - 1 - 90.

[4] BAKER TP. A Stack-Based Resource Allocation Policy for Real-time Processors[A]. Proceedings of IEEE Real-Time System Symposium [C]. 1990. 191 - 200.

[5] KRISHNA CM, SHIN KG. Real-Time System[M]. 北京: 清华大学出版社, 2001.

[6] Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines[A]. Proceedings of IEEE Real-Time Systems Symposium[C]. 1990. 201 - 210.

[7] BURNS A. Scheduling Hard Real - Time Systems: S Review [J]. IEEE, 1991, 112 - 119.