

# RTSIM : 一种实时调度模拟框架

金国哲<sup>1</sup>, 陈蜀宇<sup>2</sup>

(1. 重庆大学计算机学院, 重庆 400044; 2. 重庆大学软件学院, 重庆 400044)

**摘要:** 针对现有的模拟工具无法同时满足实时可调度性分析、调度结果的图可视化显示及算法动态可替代性等问题, 利用 python 的 Matplotlib 库与动态加载技术, 提出一种实时调度模拟框架——RTSIM。模拟实时调度算法的调度过程, 并验证算法的可调度性是一项较复杂的问题。实验结果表明, RTSIM 能够较好地完成对实时调度算法的验证和模拟。

**关键词:** 实时调度; 模拟; 可调度性

## RTSIM : Real-time Schedule Simulation Framework

JIN Guo-zhe<sup>1</sup>, CHEN Shu-yu<sup>2</sup>

(1. School of Computer, Chongqing University, Chongqing 400044;  
2. School of Software, Chongqing University, Chongqing 400044)

**【Abstract】** Aiming at the problems of current simulation tools are not able to provide schedulability analysis, graphical show of schedule result and ability of dynamic substitution of algorithm simultaneously, this paper puts forward a real-time schedule simulation framework called RTSIM by using python's matplotlib, dynamical loading technique. Simulating the process of real-time schedule and testing schedulability is a complicated issue. Experimental result shows that RTSIM can test and simulate real-time schedule algorithm well.

**【Key words】** real-time schedule; simulation; schedulability

### 1 概述

为了满足实时系统中各个任务的实时性要求, 当提出新的实时调度算法时, 都要检查此算法是否满足特定的实时性要求。目前为止, 各国学者已经提出检查实时性要求的各种模型或工具。RM 算法可以帮助实时系统设计者预测一个任务集的调度行为, 同时可以分析任务集的可调度性<sup>[1]</sup>。Rapid-RMA<sup>[2]</sup>作为比较完整的工具提供了对各种经典实时调度算法和可调度性分析。TkRTS<sup>[3]</sup>工具针对不同的实时调度算法提供可调度性分析功能。YASA<sup>[4]</sup>工具则只提供对实时调度的模拟功能。MAST<sup>[5]</sup>系统虽然同时提供了对实时调度的模拟功能和可调度性分析功能, 但是如果增加新的调度算法, 那么就需要更改整个系统, 因此, 系统的可扩展性较差。

为了解决之前同类系统的各种不足, 本文提出 RTSIM 实时调度模拟框架, 并设计和实现了整个系统。

### 2 实时调度理论基础

在实时调度模型中可以把一个任务  $T_i$  描述为三元组:

$$\langle D_i, P_i, C_i \rangle$$

其中,  $D_i$  表示任务的相对时限, 即任务在就绪之后的  $D_i$  时间之前应该完成执行并产生一个有价值的结果;  $P_i$  表示任务的周期;  $C_i$  表示任务在最坏情形下无中断执行的时间。

对于给定的实时任务集, 本文主要讨论 2 个问题: 实时任务集的调度模拟和实时任务集的可调度性分析。

针对实时任务集的调度模拟问题, 可以举例说明如下: 给出一个任务集  $\{T_1, T_2, T_3\}$ , 其中,  $T_1$  为  $\langle 6, 6, 2 \rangle$ ;  $T_2$  为  $\langle 8, 8, 2 \rangle$ ;  $T_3$  为  $\langle 12, 12, 4 \rangle$ 。

利用 RM 算法对此任务集进行调度, 任务集的调度结果如图 1 所示。

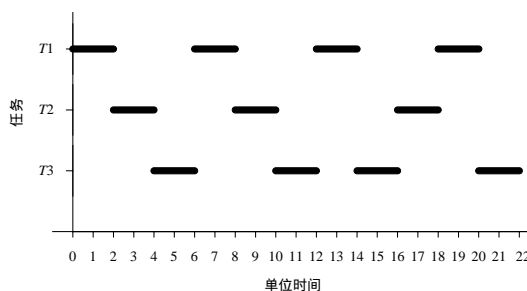


图 1 任务集调度结果

针对实时任务集的可调度性问题, 根据 RTSIM 框架系统的需要, 介绍 RM 和 EDF 2 种算法的可调度性分析方法。

首先定义 CPU 利用率:

$$U = \sum_{i=1}^n \frac{C_i}{P_i}$$

其中,  $n$  为任务个数。

**定理 1** 采用 RM 调度的周期性任务集, 如果其 CPU 利用率  $U$  不大于  $n(2^{1/n} - 1)$ , 则该任务集是可调度的。

同时, 文献[1]证明了以下定理:

**定理 2** 对于给定的任务集(由  $n$  个任务组成), 采用 EDF 调度, 其任务集可调度的充分必要条件是:  $\sum_{i=1}^n \frac{C_i}{P_i} \leq 1$ 。

在 RTSIM 框架系统中可以预定义一个任务集, 每个任务

**作者简介:** 金国哲(1983 - ), 男, 硕士研究生, 主研方向: 网络安全, 实时系统; 陈蜀宇, 教授、博士生导师

**收稿日期:** 2009-12-20 **E-mail:** kimguozhe@163.com

的  $C_i$  和  $P_i$  都是可知的。因此，利用定理 1 和定理 2 的结论给出的公式，可以分别分析采用 RM 调度和 EDF 调度的任务集的可调度性。这种分析是提前分析，如果一个任务集调度模拟的计算量很大，则运用上面的公式提前计算可调度性具有十分明显的优势。

### 3 RTSIM 框架系统的设计与实现

根据功能把 RTSIM 分为 3 个子系统：用户界面，控制，实时调度模拟引擎。

- (1)用户界面：负责设置参数，显示结果。
- (2)控制：负责把用户界面发过来的请求按照请求类型转发给实时调度模拟引擎。
- (3)实时调度模拟引擎：此子系统是整个系统的核心，包括任务管理模块、资源管理模块、调度算法管理模块、消息管理子模块。

下面重点介绍实时调度模拟引擎的几个模块以及它们之间的关系。

#### 3.1 任务与资源管理

根据上述讨论的实时调度理论结合 RTSIM 的需求，抽象出的任务如下：

- taskID：任务 id。
- readyTime：任务进入就绪状态的时间。
- runTime：任务在最坏情况下无中断执行时间。
- deadLine：任务的相对时限。
- period：任务周期。
- priority：任务优先级。
- curRunTime：任务当前运行时间。
- state：任务状态，包括运行、阻塞、就绪。

资源是对外围设备、文件等实体的抽象，主要由信号量控制对资源的并发访问。

图 2 为描述任务、资源、GlobalTimer 之间的关系的静态类图。

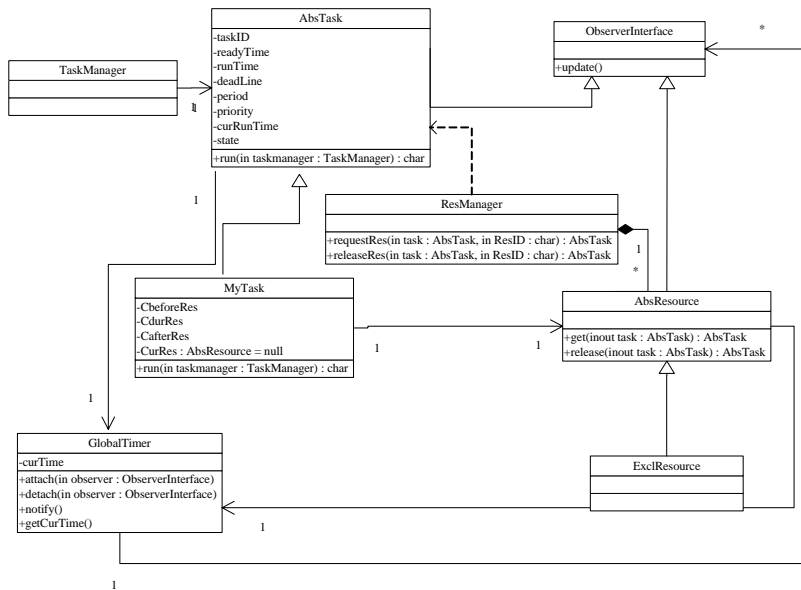


图 2 任务与资源管理类图

任务和资源中所说的所有时间概念可以理解为 Global Timer 类产生的时间计数单元。GlobalTimer 与其他类之间的交互过程中使用了 Observer 模式。

#### 3.2 调度算法管理

为实现调度算法的动态可替换性，本文使用了 Strategy 模式。Strategy 模式将算法封装到一个类(SchedulerManager)，通过组合的方式将具体算法的实现在组合对象中实现，再通过委托的方式将抽象接口的实现委托给组合对象实现。调度算法管理类图如图 3 所示。

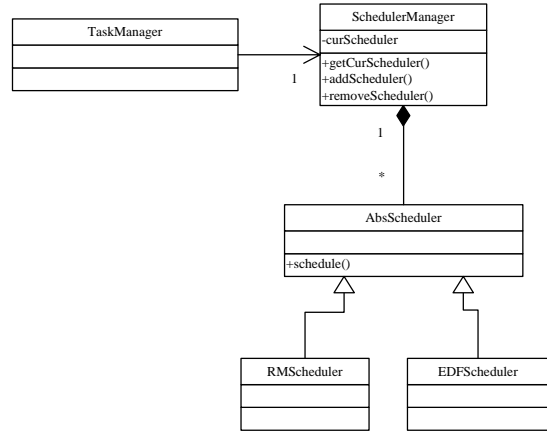


图 3 调度算法管理类图

TaskManager 类就可以通过 SchedulerManager 类动态地选择所需要的类，而 SchedulerManager 类则保存和管理所有具体算法的实例。

#### 3.3 消息管理及调度模拟的核心过程

在实时调度模拟过程中，如果出现正在运行的任务请求的资源得不到满足或超过任务规定的时限等事件发生的时候，则会调用消息管理模块来处理消息。图 4 为消息处理模型的静态类图。

实时调度模拟过程中的核心流程如下：

- (1)如果用户设置了直接用公式计算任务集的可调度性，则转到(2)，否则转到(3)。
- (2)利用第 2 节的定理 1(RM)和定理 2(EDF)给出的结论计算可调度性，如果可调度，则转到(3)，否则转到(8)。

(3)开始实时调度模拟。

(4)调用 3.2 节中介绍的 Scheduler Manager 中的当前调度算法(RMScheduler)的 schedule()方法，选择下一个需要运行的任务，进入运行状态(调用 Task Manager 中的 curTask.run()方法)。

(5)进入 curTask.run()方法，依次判断当前运行任务是否释放资源，当前请求的资源是否满足，是否超过当前运行任务的时限，用户定义的调度模拟时间是否已到，是否有新的任务需要加入到就绪队列，当前运行的任务是否是就绪队列中优先级最高的任务。如果其中的任何一项条件判断为是，那么从 curTask.run()方法中返回相应的事件的代码，转入(7)，否则，转入(6)。

(6)更新 3.1 节中介绍的 GlobalTimer 的 curTime 为 curTime+1(过了一个单位时间)，并调用 GlobalTime 的 notify()方法，通知所有注册过的类，并且转到(5)。

(7)根据 curTask.run()中返回的事件代码调用消息管理模块处理相应的消息,如果返回的事件代码是用户定义的调度模拟时间已到事件,转到(8),否则转到(3)。

(8)结束实时调度模拟,显示模拟结果。

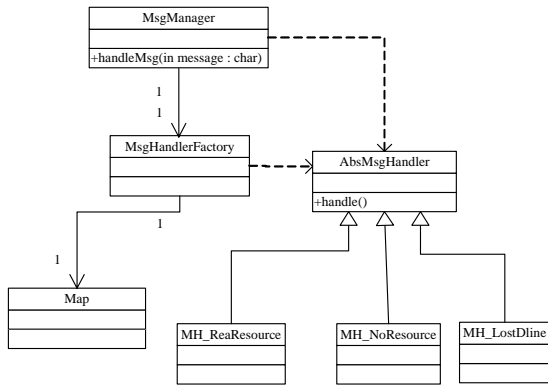


图 4 消息管理类图

### 3.4 相关细节实现

图 5 为实时调度模拟框架 RTSIM 的截图。

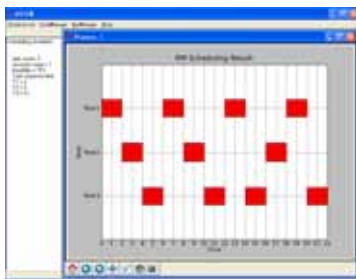


图 5 RTSIM 截图

利用 python 编写系统在算法可替代性的设计,无需重新编译整个系统就可以将新算法动态加入到 RTSIM 中,并对算法进行实时调度实验。最后利用 python 的 Matplotlib 库来编写调度结果图表化显示模块。

### 4 结束语

RTSIM 较好地实现了其主要的 3 个功能,达到了本框架设计的要求。下一步需要研究的工作是:(1)利用 XML 描述任务、资源等数据,以提高灵活性和扩展性。(2)提供对多处理机的支持。

#### 参考文献

- [1] Liu C L, Layland J W. Scheduling Algorithms for Multiprogramming in a Hard Real-time Environment[J]. Journal of the Association for Computing Machinery, 1973, 20(1): 46-61.
- [2] Pacific T. Rapid-RMA: The Art of Modeling Real-time Systems[EB/OL]. (2007-12-03). <http://www.tripac.com/html/product-rrm.html>.
- [3] Migge J. Real-time Scheduling: A Trajectory Based Model[D]. Nice, France: University of Nice Sophia Antipolis, 1999.
- [4] Blumenthal J, Hildebrandt O J. Framework for Validation and Analysis of Real Time Scheduling Algorithms and Scheduler Implementation[EB/OL]. (2003-10-20). <http://yasa.e-technik.uni-rostock.de/>.
- [5] Harbour M G, Garca J G, Gutierrez J P, et al. MAST: Modeling and Analysis Suite for Real-time Application[C]//Proc. of the 13th EUROMICRO Conference on Real-time Systems. New York, USA: [s. n.], 2001.

编辑 索书志

(上接第 277 页)

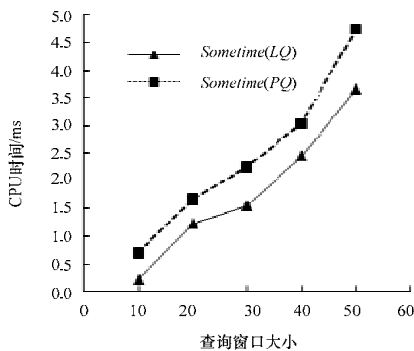


图 5 Sometimes 模式下的 CPU 时间

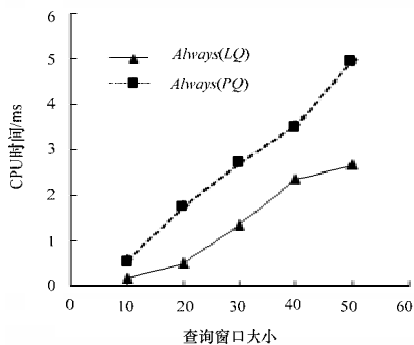


图 6 Always 模式下的 CPU 时间

### 5 结束语

本文利用已有的空间轨迹不确定性模型对网络移动对象的轨迹进行了建模,并进行了点查询和概率范围查询。实验表明该方法能够有效提高查询结果的正确性且占用较少的 CPU 时间。后续的研究工作包括将概率方法运用到受限网络中移动对象的索引等方面,对不确定性进行更深入的研究。

#### 参考文献

- [1] Byunggu Y. A Spatiotemporal Uncertainty Model of Degree 1.5 for Continuously Changing Data Objects[C]//Proc. of SAC'06. Dijon, France: [s. n.], 2006.
- [2] Byunggu Y, Kim S H, Alkobaisi S, et al. The Tornado Model: Uncertainty Model for Continuously Changing Data[C]//Proc. of DASFAA'07. Bangkok, Thailand: [s. n.], 2007.
- [3] Ding Zhiming, Guting R H. Uncertainty Management for Network Constrained Moving Objects[C]//Proc. of DEXA'04. Zaragoza, Spain: [s. n.], 2004.
- [4] Almeida V T, Guting R H. Supporting Uncertainty in Moving Objects in Network Databases[C]//Proc. of the 13th Int'l Workshop on Geographic Information Systems. Bremen, Germany: [s. n.], 2005.
- [5] Trajcevski G. Probabilistic Range Queries in Moving Objects Databases with Uncertainty[C]//Proc. of MobiDE'03. San Diego, California, USA: [s. n.], 2003.

编辑 金胡考