

垃圾收集器实时化的研究

张 宁¹, 张春宇², 杨 霞¹, 熊光泽¹, 廖 勇¹

(1 电子科技大学 计算机科学与工程学院, 四川 成都 610054; 2 攀枝花大学 计算机学院, 四川 攀枝花 617000)

摘 要: 用垃圾收集器进行动态内存管理对大规模软件系统的开发具有重要意义, 为把垃圾收集器应用到嵌入式实时系统中, 对垃圾收集器实时化进行了研究, 提出了动态执行垃圾收集器任务以及用最短时限优先(EDF)算法调度垃圾收集器的算法, 仿真实验和内存分析表明, 与以往的实时垃圾收集器相比, 所提出的算法可以进一步降低系统内存需求, 提高系统的灵活性, 并对分布式垃圾收集器进行了研究, 性能分析表明基于关键引用验证的分布式垃圾收集器更适用于大规模分布式系统。

关键词: 垃圾搜集器; 实时系统; 算法; 调度

中图分类号: TP316.2; V557+.1

文献标识码: A

文章编号: 1000-7180(2010)01-0034-04

Research of Real-time Garbage Collection

ZHANG Ning¹, ZHANG Chun-yu², YANG Xia¹, XIONG Guang-ze¹, LIAO Yong¹

(1 School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China; 2 School of Computer Science and Technology, Panzhihua University, Panzhihua 617000, China)

Abstract: Dynamic memory management with garbage collection is very important for development of large-scale software system. To apply garbage collection into embedded real-time system, real-time garbage collection is researched in this paper. Dynamic execution of garbage collection and scheduling garbage collection by EDF are proposed. Memory analysis and simulation results show that proposed algorithms can reduce memory requirement and enhance flexibility of system compared with previous work. Distributed garbage collection is also researched and performance analysis shows that proposed distributed garbage collection based on critical reference verification fits for large-scale distributed system.

Key words: garbage collection; real-time; algorithms; scheduling

1 引言

近年来,高级开发语言 JAVA 和 C# 成为主流,它们都广泛采用了一种自动内存管理技术,被称为垃圾收集器(GC).垃圾收集器可以自动判断并回收无用的内存对象进行重新分配,由于不需要 C 语言那样人工回收内存,这样就克服了人工内存管理所导致的潜在危险,例如内存泄露、指针悬挂、内存碎片等等,大大减少了系统出错的可能性。

然而,传统的 GC 在执行时需要停止所有进程的执行,这显然是不适合实时系统的.为了将这些高级开发语言应用于大规模嵌入式实时系统的开发,

国外对垃圾收集器的实时化做了许多研究,文中描述了实时 GC 的发展过程,在对单处理器 GC 和分布式 GC 的实时化上提出了一些算法,具有较好的性能。

2 GC 实时化的发展现状

2.1 单处理器 GC 实时化的发展

近十年来,人们提出了可以保证硬实时任务时限的 GC 模型,称为基于时间的 GC. GC 被作为一个单独的实时任务参与调度,GC 的执行不再由任务的操作触发,而在系统分配的时间片内执行. Bacon^[1]等人提出了一种渐进的标记清除算法. Kim^[2]

收稿日期: 2009-01-06; 修回日期: 2009-03-06

基金项目: 国家“八六三”计划项目(2007AA01Z131)

等人设计了一种基于零星任务服务器(SS)的调度算法,如图1所示,其中 R_{GC} 是 GC 的响应时间.该算法利用实时系统中经典的非周期服务器策略,使用非周期服务器提供的时间片来执行 GC,并使其优先权最高.图1表示 GC 执行时间为5,周期为10,非周期服务器时间片为2. Robertz^[3]等人提出了时间触发的 GC, GC 只有一个必要的参数就是时限,也只有时限可以控制 GC 的执行过程,因此,他的算法可以动态调整时限,具有很好的可移植性.

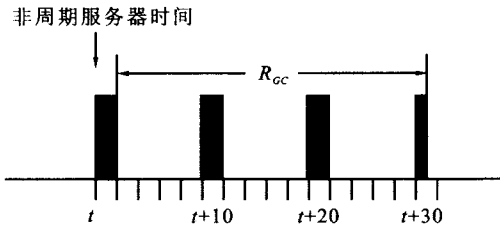


图1 Kim 的 GC 模型 ($T_s = 10, C_s = 2, C_{GC} = 5$)

2.2 分布式实时 GC

分布式系统由许多结点组成,通常每个结点上的无用对象由本地 GC 回收,而分布于多个结点上的无用对象即该对象还被其他结点上的对象引用,则需要由全局 GC 判断并回收. GC 同样可以用标记清除等算法从根结点开始依次标记所有由根可达的活跃对象,并回收标记为无用的对象. 结点之间通过消息传递 GC 所需要的引用信息.

由于分布式垃圾收集器更加复杂,并且分布式系统中影响实时性的因素很多,例如消息传递的时间花费的不确定性使分布式 GC 的实时化变得更为复杂,其中具有代表性的是 Rudalics^[4]提出的实时分布式 GC 模型,他的模型基于拷贝算法.事实上,由于在他的算法中,一次全局 GC 的循环需要全部结点的本地 GC 依次完成后才能结束,因此如果结点数较多,全局 GC 的执行时间就会过长,从而导致某些结点中的内存可能耗尽.因此, Rudalics 的算法也不适合于大规模分布式系统.

3 GC 实时化的研究

3.1 减小实时 GC 的系统内存需求

由于嵌入式实时系统往往运行在资源有限的环境中,因此如何保证系统实时性的条件下减小系统的内存需求成为一个重要的研究方向. Kim 等人在文献[2]中对实时 GC 的系统内存需求(基于拷贝算法)进行了详细的分析,式(1)计算了最坏情况的内存需求量,

$$M = 2 \left(\sum_{i=1}^n \pi_i A_i + L_k^* \right) \quad (1)$$

式中, π_i 是 GC 响应时间(R_{GC})内任务 M_i 的执行次数, A_i 是任务 M_i 的最坏情况内存需求, L_k^* 是系统中活动对象所占内存的最大值, n 是周期性实时任务的数目,从式(1)可以看出,系统的内存需求取决于 R_{GC} . 当其他参数不变时, R_{GC} 的大小实际上取决于 GC 执行时间(C_{GC}).

$$C_{GC} = c_1 \left(\frac{R + L_k^*}{\text{sizeof}(\text{word})} \right) + c_2 L_k^* + c_3 \epsilon + c_4 M \quad (2)$$

在 C_{GC} 的计算公式(式(2))中,除了 L_k^* ,其他都为定值,这表明 L_k^* 是决定 GC 的最坏情况执行时间的主要因素. Kim 等人在文献[2]中详细分析了 L_k^* 的最大值,减小了 L_k^* 的理论上界,因而减小了系统的内存需求. Xian^[5]在 Kim 分析的基础上提出了精确参数分析的方法,通过精确分析调整参数配置,使 R_{GC} 的值达到最小,与 Kim 的算法相比, Xian 的方法减小了系统内存需求.

文中在文献[6]提出的算法同样是基于 Kim 的拷贝 GC 和内存分析方法,根据式(1)和式(2),可以看出,减小 L_k^* 就可以减小系统内存需求.

$$L_k = \sum_{M_i \in \text{active}(t_i)} A_i + \sum_{M_j \in \text{inactive}(t_i)} \alpha_j A_j \quad (3)$$

如式(3)所示,内存需求量是由所有任务的内存量相加而得,而当任务未被调度时,其内存占用降低为 $A_i * \alpha_i$. 文献[7]提出 GC 的执行是个动态的过程,应当优先处理未调度任务的内存,由于任务在动态地并发执行,因此当 GC 在下一个时间片运行时,原先活跃的任务可能已经运行完,进入了未调度状态. GC 按这样的顺序依次处理各任务所占用的内存,可以减小 L_k^* ,按上面的公式,自然就减小了 GC 的执行时间,最终达到减小系统内存的目的.

3.2 用 EDF 调度 GC

在以前对 GC 的研究中,调度分析通常假设系统中没有非周期性任务.然而,在现实的实时系统中,特别是当系统需要和外界交互时,往往会遇到不可预知的情况发生,因此零星的非周期性任务应当被考虑进去.这些非周期性任务同样具有硬时限,而传统的基于 RM 算法的调度策略难以处理这样的情况.提出了用 EDF 调度 GC 的策略,详细的分析在文献[8]中.

相比基于 RM 的调度模型,用 EDF 调度 GC 更具灵活性,当一个非周期性任务到达系统时, EDF

调度系统可以动态地调整垃圾搜集器的周期以保证所有的实时任务包括非周期性的实时任务都能够满足其时限. 只要系统里的任务数有限, 用 EDF 来调度 GC 系统, 就会比 RM 调度具有更好的灵活性. 另一方面, 用 EDF 调度 GC 能进一步减小系统内存需求量.

考虑和 Xian^[5]中同样的一组任务集(见表 1).

表 1 任务集($U_p=68.3\%$, $C_{GC}=3$)

任务	C_i	T_i	D_i	A_i	α_j
M_1	2	10	10	488	0.53
M_2	4	30	30	528	0.46
M_3	10	50	50	800	0.38
M_4	15	100	100	1296	0.57

图 2 显示了与 Xian 的算法相比, 如果使用 EDF 调度系统, 可以降低 9.8% 的内存需求. 因为在 EDF 调度下, CPU 的带宽可以得到充分利用, GC 的响应时间 R_{GC} 可以达到最小值, 而根据式(1)可知, 系统内存由 R_{GC} 决定, 因此用 EDF 调度 GC 可以进一步减小内存需求.

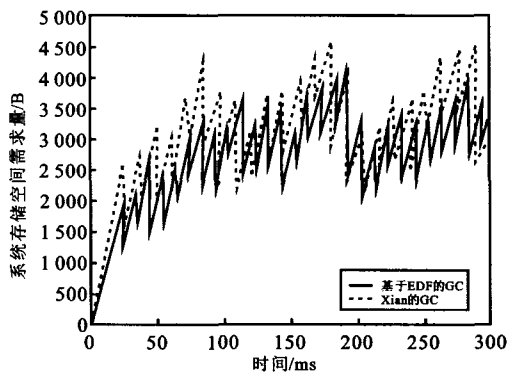


图 2 基于 EDF 调度的 GC 和 Xian 的 GC 的内存需求对比

3.3 新的分布式实时 GC 算法

分布式 GC 比单处理器的 GC 要复杂得多, 本地垃圾可以用本地 GC 回收, 而回收跨结点的循环垃圾成为分布式 GC 的一个重要问题. 目前存在的许多算法有的太过复杂开销太大^[7], 有的则需要很长时间才能判断并回收循环垃圾. 文中简要介绍的一种新的分布式实时垃圾算法, 每个结点用 Kim 的 SS 实时调度算法调度 GC 和实时任务(模型如图 1 所示), 每个对象维护一张引用表, 记录了该对象的引用信息. 每个结点的本地 GC 循环结束后, 都周期性地向它引用的结点发送包含了引用信息的信息. 如果对象的引用项为零, 则表明该对象没有任何引用指向它, 自然被 GC 回收. 该算法最重要的部分在

于可以最短时间判断并回收跨结点的循环垃圾.

该算法对循环垃圾的判断是基于关键引用, 关键引用指的是当删除该引用后即出现循环垃圾的引用, 如图 3 所示, 如果删除引用 $root \rightarrow O$, 则导致 O 等对象都成为循环垃圾, 而删除 $B \rightarrow O$ 和 $C \rightarrow O$ 则不会, 因此可以判断引用 $root \rightarrow O$ 是关键引用. 因此, 每个结点除了周期性传递引用信息给引用结点外, 还同时传递时间戳, 每次本地 GC 循环结束时, 根对象更新时间戳. 这样, 当时间戳随着引用依次传递时, 就能保证关键引用一定是时间戳最大的引用.

然而, 反之时间戳最大的引用并不都是关键引用, 有可能还有其他的根间接引用该对象, 因此, 当时间戳最大的引用被删除时, 需要启动判断过程来验证该引用是否为关键引用. 如果是, 则 GC 回收包含该对象的循环垃圾.

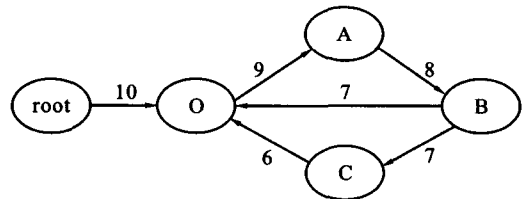


图 3 分布式 GC 循环垃圾模型

如图 3 所示, 数字表示引用的时间戳, 当引用 $root \rightarrow O$ 被删除时, 因为其时间戳为 10 最大, 则启动判断, 判断的标准是: 如果该引用所到达的所有对象都被他们自己所引用, 即没有来自外部对象的引用, 则该引用为关键引用. 于是 O 发送消息给 A 询问引用它的 B 和 C 是否可到达, A 发送询问消息给 B, B 发送消息给 O 时, O 就知道 B 是可到达的, 同样也知道 C 可到达, 则 O 知道所有引用它的对象都可由它到达, 可判断删除的引用是关键引用. 有两种情况将终止询问过程, 一是 O 的所有引用被删除, 则它明显成为需回收的垃圾; 二是当有其他根间接引用它时, 不断更新的时间戳将最终超过启动询问时的时间, 则表明 O 不是循环垃圾, 询问可终止^[9-10]. 这种算法之所以是实时的, 因为一方面本地采用 Kim 的 SS 实时调度策略, 可以保证实时任务的时限, 另一方面, 循环垃圾可以在最短时间内被判断并回收, 这样就保证了某些结点不会因为垃圾存放时间过长而导致内存耗尽. 不过由于分布式系统的复杂性, 该算法还只能说是软实时的, 精确的调度分析和内存分析将是未来进一步研究所需要做的工作.

4 结束语

文中介绍了垃圾收集器实时化的发展过程,介绍了在 GC 实时化上做的一些研究,对单处理器 GC 性能进行改进,降低了系统内存需求,提出了 EDF 调度 GC 的思想使系统更具灵活性,同时内存需求更小,在分布式 GC 方面提出了真正适合大规模分布式系统的软实时算法,能以最短时间及时回收所有垃圾。

参考文献:

- [1] 马明全,周明全,耿国华. 分布式大规模交通网络实时系统[J]. 微电子学与计算机,2006,23(12):56-59.
- [2] 赵春云,郭煦,金戟,等. JAVA 语言及其虚拟机技术探讨[J]. 微电子学与计算机,2006,23(12):15-18.
- [3] Baker H G. List processing in real time on a serial computer[J]. Communications of the ACM, 1978, 21(4): 280-294.
- [4] Henriksson R. Scheduling garbage collection in embedded systems[D]. Sweden: Lund University, 1998.
- [5] Kim Taehyoun, Naehyuck Chang, Heonshik Shin. Bounding worst case garbage collection time for embedded real-time systems[C]// Real-Time Technology and Applications - Proceedings. Washington, USA, 2000: 46-55.
- [6] Martin Rudalics. Distributed copying garbage collection[J]. ACM Symposium on Lisp and Functional Programming, 1986(8): 364-372.
- [7] Xian Yuqiang, Xiong Guangze. Minimizing memory requirement of real-time systems with concurrent garbage collector[J]. ACM SIGPLAN Notices, 2005, 40(3): 40-48.
- [8] Zhang Ning, Xiong Guangze. Minimizing GC work by analysis of live objects[J]. ACM SIGPLAN Notices, 2006, 41(3): 20-29.
- [9] 张宁,熊光泽. 用 EDF 调度实时任务和 GC[J]. 航空学报, 2008,29(5):1226-1232.
- [10] Lang B, Queninnec C, Piquer J. Garbage collecting the world[C]// Proceedings of the Nineteenth ACM Symposium on Principles of Programming Languages. New York: ACM Press, 1992: 39-50.

作者简介:

- 张宁** 男,(1975-),博士研究生.研究方向为嵌入式实时系统、动态内存管理.
- 张春宇** 男,(1970-),讲师.研究方向为计算机控制.
- 杨霞** 女,(1978-),讲师.研究方向为实时可信计算、防危安全.
- 熊光泽** 男,(1938-),博士生导师,教授.研究方向为嵌入式实时系统、可信计算.
- 廖勇** 男,(1976-),博士,讲师.研究方向为实时计算、实时调度.

(上接第 33 页)

- [2] Sun M, Chen B, Zhou M T. An ILP approach to mine the association rules on log ontology[C]// Proceedings of ICA-CIA 2008. Chengdu, China: IEEE Press, 2008: 274-278.
- [3] Motik B, Sattler U, Studer R. Query answering for OWL-DL with rules[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2005,3(1):41-60.
- [4] 张玉花,张娜. 面向对象的本体创建方法[J]. 微电子学与计算机,2008,25(7):65-67.
- [5] Hustadt U, Motik B, Sattler U. Reducing SHIQ-description logic to disjunctive datalog programs[C]// Proceedings of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2004). Whistler, Canada, 2004:152-162.
- [6] Francesca A L. An ILP approach to ontology refinement for the semantic web[C]//Proceedings of the 4th European Semantic Web Conference, LNCS, 2007. Berlin: Springer,2007:117-129.
- [7] J'ozefowska J, Lawrynowicz A, Lukaszewski T. A study of the SEMINTEC approach to frequent pattern mining [C]// Proceedings of PriCKL 2007, ECML/PKDD' 2007. Warsaw, Poland, 2007:41-52.
- [8] Francesca A L. Principles of inductive reasoning on the semantic web: a framework for learning in AL-log[C]// Proceedings of the 3rd International Workshop on Principles and Practice of Semantic Web Reasoning. Heidelberg: Springer, Berlin, 2005:118-132.

作者简介:

- 孙明** 男,(1978-),博士研究生.研究方向为知识发现、语义 Web 挖掘以及本体学习.
- 陈波** 男,(1977-),博士.研究方向为知识发现以及粒度粗糙集理论.
- 周明天** 男,(1939-),教授,博士生导师.研究方向为中间件技术、知识发现以及网络计算.