P.O.Box 8718, Beijing 100080, China	Journal of Software, Oct. 2005,16(10):1708-1716
E-mail: jos@iscas.ac.cn	ISSN 1000-9825, CODEN RUXUEW, CN 11-2560/TP
http://www.jos.org.cn	Copyright © 2005 by The Editorial Department of *Journal of Software*

并发Java程序同步操作的有效删除

吴 萍, 陈意云, 张 健

吴 萍1, 陈意云1, 张 健2
1(中国科学技术大学 计算机科学技术系,安徽 合肥 230027)
2(中国科学院 软件研究所 计算机科学重点实验室,北京 100080)
作者简介: 吴萍(1978－),女,安徽东至人,博士生,主要研究领域为程序分析,软件工程,形式化方法;陈意云(1946－),男,教授,博士生导师,主要研究领域为程序设计语言理论和实现技术,形式化描述技术,软件安全;张健(1969－),男,博士,研究员,博士生导师,主要研究领域为自动推理,约束求解,形式化方法.
联系人: 吴 萍 Phn: +86-10-62562796, E-mail: cynthia@mail.ustc.edu.cn, http://www.iscas.ac.cn

Received 2004-05-18; Accepted 2005-03-11

Abstract
Synchronization operations make a huge expense for concurrent Java programs. This paper proposes an effective and precise static analysis algorithm for the redundant synchronization removal. The algorithm consists of two phases-basic analysis and inter-thread temporal analysis. Both phases take the effect of control flow relation and thread control relation into count. This paper also constructs a Java compiler-JTool and implements the algorithm on it. To deterministic single-threaded programs, the removal ratio reaches 100% and to multi-threaded programs, the removal ratio is higher than the existing analysis tools.

Wu P, Chen YY, Zhang J. Effective synchronization removal in concurrent Java programs. *Journal of Software*, 2005,16(10):1708-1716.
DOI: 10.1360/jos161708
http://www.jos.org.cn/1000-9825/16/1708.htm

摘要
同步操作是并发Java程序非常大的一部分开销.在现有程序分析方法的基础上,提出了一种精确而有效的冗余同步操作的静态删除方法.该方法分为基本处理和线程间时序分析两个阶段,充分考虑了控制流结构和线程交互时序对同步删除的影响.构造了一个Java编译器JTool,并在其上实现了同步删除算法.对于确定的单线程程序,同步删除率达到100%;对于多线程程序,同步删除率高于现有的分析工具.

基金项目：Supported by the National Natural Science Foundation of China under Grant Nos.60173049, 60421001 (国家自然科学基金); the National Science Fund for Distinguished Young Scholars of China under Grant No.60125207 (国家杰出青年科学基金)

References:

[1] Aldrich J, Chambers C, Sirer E, Eggers S. Static analyses for eliminating unnecessary synchronization from Java programs. In: Proc. of the 6th Int'l Symp. on Static Analysis. London: Springer-Verlag, 1999. 19-38.

[2] Fitzgerald R, Knoblock TB, Ruf E, Steensgaard B, Tarditi D. Marmot: An optimizing compiler for java. Software-Practice and Experience, 2000,30(3):199-232.

[3] Bacon DF, Konuru R, Murthy C, Serrano M. Thin locks: Featherweight synchronization for Java. In: Proc. of the Conf. on Programming Language Design and Implementation (PLDI'98). New York: ACM Press, 1998. 258-268.

[4] Ruf E. Effective synchronization removal for Java. In: Proc. of the Conf. on Programming Language Design and Implementation (PLDI 2000). New York: ACM Press, 2000. 208-218.

[5] http://www.flex-compiler.csail.mit.edu

[6] Blanchet B. Escape analysis for object-oriented languages: Application to Java. In: Proc. of the 14th Annual Conf. on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'99). New York: ACM Press, 1999. 20-34.

[7] Bogda J, Hoelzle U. Removing unnecessary synchronization in Java. In: Proc. of the 14th Annual Conf. on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'99). New York: ACM Press, 1999. 35-46.

[8] Choi JD, Gupta M, Serrano M, Sreedhar VC, Midkiff S. Escape analysis for Java. In: Proc. of the 14th Annual Conf. on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'99). New York: ACM Press, 1999. 1-19.

[9] Choi JD, Gupta M, Serrano MJ, Sreedhar VC, Midkiff SP. Stack allocation and synchronization optimizations for Java using escape analysis. In: Proc. of the ACM Sigplan Trans. on Programming Languages and Systems (TOPLAS 2003). New York: ACM Press, 2003,25 (6):876-910.

[10] Salcianu A, Rinard M. Pointer and escape analysis for multithreaded programs. In: Proc. of the Symp. Principles and Practice of Parallel Programming (PpoPP 2001). New York: ACM Press, 2001. 12-23.

[11] Wilson RP. Efficient, context-sensitive pointer analysis for C programs [Ph.D. Thesis]. Stanford University, 1997.

[12] Muth R, Debray S. On the complexity of flow-sensitive dataflow analyses. In: Proc. of the Symp. on Principles of Programming Languages (POPL 2000). New York: ACM Press, 2000. 67-80.

[13] von Praun C, Gross TR. Static conflict analysis for multi-threaded object-oriented programs. In: Proc. of the Conf. on Programming Language Design and Implementation (PLDI 2003). New York: ACM Press, 2003. 115-128.