

Profiling Java on Linux

John Kacur

IBM

jekacur@ca.ibm.com

Abstract

In this paper, I will examine two profilers. IBM's Open Source Performance Inspector and OProfile which contains code that has been officially accepted into the Linux Kernel. Currently OProfile doesn't work with programs that dynamically generate code, such as Python and Java JITs. Various people have proposed patches that record events in anonymously mapped memory regions as raw virtual addresses, instead of the usual tuple of binary image and offset. This information can be post-processed by matching it with the output generated by running a Java program with Performance Inspector's JPROF which uses JVMPI to record addresses of JITted methods. In this paper, I will discuss the details of profiling Java, specifically looking at the inner workings of OProfile and Performance Inspector. I will discuss problems that we have encountered with both tools and our attempts to resolve them. Finally, I will demonstrate profiling a java program to show the kind of information that can be obtained.

Check Online

This author did not provide the body of this paper by deadline. Please check online for any updates.
--

Proceedings of the Linux Symposium

Volume One

July 20nd–23th, 2005
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc.*
C. Craig Ross, *Linux Symposium*
Stephanie Donovan, *Linux Symposium*

Review Committee

Gerrit Huizenga, *IBM*
Matthew Wilcox, *HP*
Dirk Hohndel, *Intel*
Val Henson, *Sun Microsystems*
Jamal Hadi Salimi, *Znyx*
Matt Domsch, *Dell*
Andrew Hutton, *Steamballoon, Inc.*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*

Authors retain copyright to all submitted papers, but have granted unlimited redistribution rights to all as a condition of submission.