

Dramatic Changes in Java Troubleshooting! —Pursuing Uncompromising Low-overhead Implementations

Yasumasa Suenaga

Abstract

This article focuses on the work of one of the researchers who contributed to a Feature Article in this issue.

Keywords: HeapStats, OSS, Java

In April 2013, the NTT Open Source Software (OSS) Center released HeapStats, a Java Virtual Machine (JVM) monitoring and analysis support tool, as open source software (OSS). This attracted a great deal of interest from various people inside and outside the NTT Group. We also had the gratifying experience of being awarded the Prize for Excellence among Java technologists representing Japan at the Java Day Tokyo 2013 event. Installations of HeapStats within the NTT Group are increasing and have begun to produce good results in analyzing the sources of problems when and where they occur.

The NTT OSS Center provides support and problem analysis for organizations adopting various types of OSS. I primarily handle problem analysis related to Java, and I quite often encounter bugs related to regions of Java heap memory that are not being released properly by garbage collection (GC), the mechanism that recovers unused memory. These are called memory leaks. Analyzing such problems used to be very difficult because to identify the cause, it was necessary to follow the state of the Java heap memory over the time sequence leading to the memory leak. However, in most cases, there was no information indicating when the memory leak occurred, and to get such information, appropriate settings had to be set before waiting for the leak to be reproduced.

Because of this, we began developing HeapStats as a way of gathering information that can be analyzed immediately, without waiting to reproduce the problem.

Java provides an application programming interface (API) for collecting information, so we initially used it to create a prototype. However, when we tried to use it, we found that it caused a large drop in system performance. From past experience supporting development projects, we knew we could not allow any drop in overall system performance when gathering information about problems that occur with low probability. In the field of motor sports, a mechanic uses his utmost skill in tuning his machine to ensure high performance in various environments. Similarly, our goal with HeapStats was to operate with low overhead in various environments (applications). Therefore, we considered every detail down to the central processor unit (CPU) architecture in developing it. We studied the source code of OpenJDK (an open-source implementation of the Java Platform), but also analyzed the JVM executable binary by following processing at the CPU instruction level, and we thoroughly investigated the movement of data within JVM by checking the actual placement in memory and the instruction sequences using a debugger. As a result, we synchronized HeapStats perfectly

with the GC, and in doing so, we established a method for gathering information with the least possible load on the system. Furthermore, we selected CPU instructions more efficiently by introducing SIMD (single-instruction multiple-data) instructions and branch prediction, which are difficult to support using only compiler optimization options, and we repeatedly tested and fine-tuned the implementation. As a result, HeapStats operates efficiently and fully uses

the features of the CPU.

The key to growth for HeapStats is to increase the number of users and developers, so we plan to promote it to Java technologists around the world. We will also strive to improve the features and quality of HeapStats, which will contribute to improving the quality of systems that use Java. My goal is to see engineers around the world turning to HeapStats for Java memory analysis.



Yasumasa Suenaga

Senior Expert, OSS Promotion Unit, NTT Open Source Software Center.

He received the B.E. in information and communication engineering from Tokyo Denki University, Tokyo, in 2006. He joined the NTT Group in 2006. He is a committer of IcedTea (HeapStats), and an author of the JDK9 development community. In April 2014, he moved to NTT Comware.
