

C
001.642
SAM

A TOOL FOR TESTING JAVA PROGRAMS

A PROJECT REPORT PRESENTED BY

SISIRA J. B. SAMARASINGHE
✓

to the Board of study in Statistics & Computer Science of the

POST GRADUATE INSTITUTE OF SCIENCE

in partial fulfillment of the requirement

for the award of the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

of the

UNIVERSITY OF PERADENIYA

SRI LANKA

2005

603392

ABSTRACT

A TOOL FOR TESTING JAVA PROGRAMS

Sisira J. B. Samarasinghe

Post Graduate Institute of Science

University of Peradeniya

Sri Lanka

Profiling is an important step in software development. We use the term profiling to mean, in a broad sense, the ability to monitor and trace events that occur during run time, the ability to track the cost of these events, and the ability to attribute the cost of the events to specific parts of the program. For example, a profiler may provide information about what portion of the program consumes the most amount of CPU time, or about what portion of the program allocates the greatest amount of memory.

Here we discuss a general purpose, portable and extensible approach for obtaining comprehensive profiling information from the Java Virtual Machine (JVM).

Profilers based on this framework can uncover unnecessary object retention which leads to memory leaks, CPU usage hot spots, heavy memory allocation sites, contended monitors and thread deadlocks. Memory leaks can arise in Java applications due to unintentional object references. For example, an application may fail to remove callback listeners or close streams, preventing these objects from being garbage collected. Even minor leaks can grow to major problems in deployed systems. The main objective of this project is to suggest a reliable solution to address the above problems.

In this profiling process, Java Virtual Machine Profiler Interface (JVMPPI) / Java Virtual Machine Tool Interface (JVMTI) is programmed to provide information on heap dump event. A native agent is designed to capture these events and send them to a heap analysis tool via TCP socket through the Network. Heap analyzing tool is used to analyze and graphically visualize event specific data to get an understanding of what are the loaded classes and count of objects from each class. The system uses advanced Java features such as JVMPPI/JVMTI to obtain the information from Java VM, Java Native Interface (JNI) to incorporate native codes written to receive information from the JVM and HTML to build a graphical user interface to visualize results.