Adding Rigorous Statistics to the Java Benchmarker's Toolbox

Andy Georges Dries Buytaert Lieven Eeckhout
ELIS, Ghent University, Belgium
{ageorges,dbuytaer,leeckhou}@elis.ugent.be

Abstract

Java performance is far from trivial to benchmark because it is affected by various factors such as the Java application, its input, the virtual machine, the garbage collector, the heap size, etc. In addition, non-determinism due to Just-in-Time compilation/optimization, thread scheduling, etc., causes the execution time of a Java program to differ from run to run.

This poster advocates statistically rigorous data analysis when reporting Java performance. We advise to model non-determinism by computing confidence intervals. In addition, we show that prevalent data analysis approaches may lead to misleading or even incorrect conclusions. Although we focus on Java performance, the techniques can be readily applied to any managed runtime system.

Categories and Subject Descriptors D.2.8 [*Software Engineering*]: Metrics—Performance measures; D.3.4 [*Programming Languages*]: Processors—Runtime environments

General Terms Experimentation, Measurement, Performance

Keywords Java, benchmarking, data analysis, methodology, statistics

1. Introduction

Benchmarking is at the heart of experimental computer science research and development. Hence, it is crucial to have a rigorous benchmarking methodology. Otherwise, the overall performance picture may be skewed, and incorrect conclusions may be drawn. In particular, a managed runtime system, such as a Java virtual machine, poses a great challenge to benchmark because there are numerous factors affecting performance, and some of them interact with each other in a nondeterministic way, which is illustrated in a number of research papers [4, 5, 7]. Recent work in Java performance analysis [1, 2] highlights the need for a well chosen and motivated experimental design.

Through an extensive literature survey including 50 papers from the past 7 OOPSLA, PLDI, VEE, ISMM and CGO conferences, we found that there are many prevalent data analysis approaches. Some report a best value of k benchmark runs, others report a mean value of k runs, yet still others report a second best, or the worst performance number as done by SPEC [8]. In this poster we show that these prevalent data analysis approaches often lead to misleading conclusions, and in some cases even incorrect conclusions.

We advocate the use of statistically rigorous data analysis which computes confidence intervals when reporting experimental results. In addition, we develop a toolbox that automates the measurement and collection of startup and steady-state Java performance numbers.

2. A practical statistically rigorous methodology

In [3], we present the proposed statistically rigorous data analysis methodology in great detail. In this poster, we limit ourselves to presenting the key idea and a few highlights, and refer to the full paper version for an elaborate discussion on the methodology and its underlying statistics.

For Java (startup) performance[1], we propose the following approach to computing a confidence interval: (i) measure the execution time of $n > 1$ VM invocations, running a single benchmark iteration per VM invocation, and (ii) determine a confidence interval for the execution time, using the Student $t$statistic [2] $[\bar{x} \pm t_{\alpha/2;n-1}\frac{s}{\sqrt{n}}]$ [6], where $\bar{x}$ denotes the mean, and s denotes the standard deviation of the collected samples; $t_{\alpha/2;n-1}$ can be found in a precomputed $t$statistics table. α is the significance level; $(1-\alpha)$ is called the confidence level. The meaning of a confidence interval is as follows. A 90% confidence interval, i.e., a confidence interval with a 90% confidence level, means that there is a 90% probability that the actual mean of the underlying population, μ, falls within the confidence interval. It is important to note that computing confidence intervals builds on the assumption that the measurements' mean $\bar{x}$ is Gaussian distributed, which is a valid assumption based on the central limit theory, irrespective of the distribution from which the measurements are taken. In other words, the measurements themselves need not be Gaussian distributed in order to apply statistically rigorous data analysis.

OOPSLA'07, October 21–25, 2007, Montréal, Québec, Canada.
ACM 9781595937865/07/0010.

[1] We consider quantifying steadystate performance in the full paper version.

[2] If $n > 30$, one can also use the $z$statistic derived from the normal distribution.