Measuring the Impact of Object-oriented Techniques in Grande Applications: a method-level analysis

David Gregg
Dept. of Computer Science
Trinity College Dublin
Dublin 2, Ireland.
David.Gregg@cs.tcd.ie

James F. Power
Dept. of Computer Science
National University of Ireland
Maynooth, Ireland.
James.Power@may.ie

John T. Waldron
Dept. of Computer Science
Trinity College Dublin
Dublin 2, Ireland.
John.Waldron@cs.tcd.ie

ABSTRACT

In this work we seek to provide a foundation for the study of the level of use of object-oriented techniques in Java programs in general, and scientific applications in particular. Specifically, we focus on the use of small methods, and the frequency with which they are called, since this forms the basis for the study of method inlining, an important optimisation technique. We compare the Grande and SPEC benchmark suites, and note a significant difference in the nature and composition of these suites.

1. OVERVIEW

In attempting to improve the performance of Java programs, there has been much concentration on compiler design techniques to increase the speed and efficiency of the Java Virtual Machine (JVM). Since choosing and applying the correct optimisations is so important for the performance of Java applications, it is reasonable to ask whether the same techniques that apply to standard object-oriented applications also apply to scientific applications. It is arguable that many scientific applications may not be written in an object-oriented style, or may not fully exploit all the features of Java, and thus may not be amenable to the same kind of optimisation.

In this study we seek to provide a foundation for the study of the level of use of object-oriented techniques in Java programs in general, and scientific applications in particular. Specifically, we focus on the use of small methods, and the frequency with which they are called, since this forms the basis for the study of method inlining, an important optimisation technique. Unlike many existing techniques which are based on static analysis and static metrics, we study the behaviour of Java programs dynamically, as they execute on the virtual machine. This allows us to build a picture of the behaviour, rather than the architecture of the programs and directly parallels modern optimisation techniques.

In order to measure examples of Java programs we compare the results from the analysis of two benchmark suites. We use the Java Grande benchmark suite [1] to provide an example of larger, scientific applications, and the SPEC JVM98 benchmark suite [2] to represent more standard Java applications. It is not entirely obvious what constitutes a "standard" scientific Java application, and we hope that the results presented here can be used to calibrate the Grande suite, and measure the degree to which it is in fact representative of such applications.

Java Grande/ ISCOPE Seattle, Washington, November 3-5, 2002

2. SUMMARY OF RESULTS

In all cases, we note a significant difference in the nature and composition of the Grande and SPEC suites, with the programs from the Grande suite demonstrating a less object-oriented approach. On average, they make less use of the class library than SPEC applications, make greater use of static methods, and use longer methods.

The work presented in this study raises two important questions in relation to the performance of Grande applications. First, do existing compiler optimisations, targetted specifically at object-oriented programs, deliver satisfactorily for scientific applications that may not make such heavy use of these techniques? Second, are the programs in the Grande suite representative of scientific applications, or will such applications come to resemble object-oriented programs as more of the code base is moved to Java?

We believe that the analysis presented in this study provides a framework in which an answer to these questions can be developed, particularly as more scientific applications in Java become available for study.

This study was carried out as poart of the KWILU project; more details can be found at:

`http://www.kwilu.org/`

3. REFERENCES

[1] M. Bull, L. Smith, M. Westhead, D. Henty, and R. Davey. Benchmarking Java Grande applications. In *Second International Conference and Exhibition on the Practical Application of Java*, Manchester, UK, April 2000.

[2] SPEC. SPEC releases SPEC JVM98, first industry-standard benchmark for measuring Java virtual machine performance. Press Release, August 19 1998. http://www.specbench.org/osg/jvm98/press.html.

Poster presented at the Joint ACM-ISCOPE Conference on Java Grande Seattle, Washington, USA, November 3-5, 2002, pp. 229-229