

The Impact of Object Oriented Characteristics of Middleware Benchmarks

Position Paper for OOPSLA 2003
Middleware Benchmarking Workshop

Paul Brebner

CSIRO ICT Research Centre

ANU Canberra

Australia

Paul.Brebner@csiro.au

1. Objects and Design: Benchmark Design Choices

Distributed Object Oriented Platforms such as J2EE provide a number of design alternatives which impact development, maintenance and performance. It is becoming recognised that speed is not everything [1,2], and other impacts such as development and maintenance costs are also important. In previous benchmark development we have explored the performance and scalability impact of different designs (e.g. Stateless Session Beans, CMP and BMP Entity Beans). However, we did not explore in detail the development and maintenance differences, although we did explore (for example) the impact of tool and middleware support on deployment effort [3].

Many benchmarks are written to a high-level specification which allows wide variability in implementations, and if left up to the vendors, code which may be highly optimised, but time consuming to develop and maintain. The SPEC series of benchmarks, however, are code based. This has provided some challenges for the new Enterprise Services benchmark [4], as it must provide fair comparison of at least two current platforms, J2EE and .NET. However, the recommended design patterns are not always the same across platforms, or even comparable or supported. Issues such as how to ensure or measure selected non-functional characteristics across platforms have been explored, and include the use of design level attributes for each “component” (e.g. whether a component must be represented as a data object), and declarative specification of consistency requirements. However, more work is needed to provide a more robust theoretical and empirically validated (or at least, illustrated) framework to compare and measure development and maintenance metrics and costs within, and even across, platforms. As an initial exercise, preliminary attempts at quantifying development and maintenance costs of two designs of the StockOnline

benchmark [5] will be presented for a number of test scenarios.

2. Objects and Performance: J2EE Application and JVM profiling

We have been working for a number of years on Java based middleware, particularly J2EE, and discovered that the interaction of application design, Application Server, and choice of JVM brand and type, garbage collection algorithms and settings, EJB caching strategies and settings, number of JVMs and number of CPUs per server are all likely to have a critical impact on application scalability (E.g. [6]).

By factoring out Application Server differences from the equation, we have recently made progress on a model for throughput estimation for EJB applications which takes into account different designs [7].

We have also started work on white box analysis of the interaction of J2EE application design, application servers, and JVMs. So far we have developed a portable JVM profiling tool which allows the collection, storage and analysis of detailed JVM data, including all object creation and destruction, movement within heap spaces, and garbage collection [8]. Initial analysis of “object cost” and “object lifetimes” allowed us to gain a better understanding of the performance peculiarities of previous EJB caching results [9], and provided interesting insights into the performance impact of different application designs, application server and JVM settings. The JVM profiler works with any JVM which supports the JVM profiling API (JVMPI) [10], and will be further explained, and the source code provided, to workshop attendees.

REFERENCES

- [1] Gorton, Liu, Brebner, “Rigorous Evaluation of COTS middleware technology”, IEEE Computer March 2003.
- [2] Vinoski, “The Performance Presumption”, IEEE Internet Computing, March/April 2003.
- [3] Brebner (Ed.), “Evaluating J2EE Application Servers”, 2.1, MTE Report series, CSIRO Publishing, June 2002,
<http://www.cmis.csiro.au/paul.brebner/pubs/MTE%20App%20Server%20Comparison%20v2.1.pdf>
- [4] SPECAppPlatform, <http://www.spec.org/appPlatform/>
- [5] StockOnline, <http://forge.objectweb.org/projects/stock-online/>
- [6] Invited talk at BorCon2001, *How to choose an Application Server*, Paul Brebner, 2001, in proceedings of the 4th Annual Borland Conference Asia Pacific, 2001, Melbourne.
- [7] Brebner, Gosper, “The J2EE ECperf benchmark: transient trophies or technology treasures?”, *Concurrency and Computation: Practice and Experience*, in press, 2003.
- [8] Brebner, Smith, “Enterprise Java Application Profiling: Object lifetimes and garbage collection in J2EE Applications”, CMIS Technical Report 03/55, February 2003.
- [9] Brebner, Ran, “Entity Bean A, B, C's: Enterprise Java Beans Commit Options and Caching”, Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, Germany, November 2001, LNCS 2218, Springer-Verlag, pp 36-55.
- [10] Sun Microsystems, Inc. *Java Virtual Machine Profiler Interface (JVMPI)*.
<http://java.sun.com/j2se/1.3/docs/guide/jvmpi/index.html>