

Oracle9iAS vs. Microsoft .NET Benchmark Results

March 2002

Introduction

In November 2001, Microsoft announced a benchmark comparing the performance of Sun's Java Pet Store blueprint application, against the performance of the .NET Pet Shop application. The latter is a Microsoft port of the Java Pet Store to C# for the .NET platform. The comparison was based on Oracle's published performance data for a version of the J2EE Java Pet Store application running on the Oracle9i Application Server. Microsoft didn't test the Oracle9i Application Server. Instead, the .NET results were compared to Oracle's published results from June 2001.

On inspecting the Microsoft application code, it became apparent that not only had the application been rewritten in C#, but the application had been fundamentally changed. Although Oracle modified the Pet Store application slightly in its original tests (documented in the performance white paper from June 2001), Oracle used the *exact* same J2EE code and hardware configuration to compare J2EE application servers. Microsoft's comparison is unfair, as it compares two different applications. Microsoft also ran their tests on a different hardware and operating system configuration.

Subsequently, Oracle ran a Java Pet Store benchmark to compare the performance of Oracle9iAS and the .NET Pet Shop. This document summarizes the results of this benchmark. We found that without caching, Oracle9iAS was up to 18 times faster than Microsoft .NET, while using just half the resources. In addition, Oracle was able to scale to a much higher user load on the same hardware. If we compare the same application using Microsoft's output caching and Oracle's Web Cache, the results are even more dramatic. Using Web Cache, Oracle proved to be more than 22 times faster under load, while the middle tier resource utilization was only a small fraction of that consumed by .NET.

In addition to disclosing the results of this benchmark effort in this paper, Oracle is also providing the source code, a description of the environment in which the tests were run, and the test scripts that were used to simulate this environment. The code and test scripts can be downloaded from the Oracle Technology Network.

Overview of Benchmark

In the rewrite of the Java Pet Store application for .NET, Microsoft made some changes that are not specific to the .NET platform, including:

- (i) modification of the SQL query statements to return less data;
- (ii) replacing complex join queries with simple queries of a single table;
- (iii) storing the SQL queries within stored procedures in the database.

Microsoft then compared the results of this .NET application to the results that Oracle published in June 2001, thereby comparing results from two completely different applications.

Oracle recognizes that the Java Pet Store application was designed to be a showcase application that exercises a majority of the J2EE APIs and features. There are a variety of optimizations that can be done to the application design, Java code and the underlying SQL code to improve the performance and throughput of the Pet Store application.

To develop a version of Java Pet Store that was suitable for comparison with the .NET Pet Shop application, Oracle made modifications to the SQL and some of the Java to make it more efficient, but did not change the structure of the application. Oracle9iAS was able to achieve superior performance to .NET without compromising the J2EE blueprint design principles demonstrated in Java Pet Store or using database stored procedures.

In summary, Microsoft published benchmark results based on an unfair comparison between different applications. In reproducing that benchmark, with only minor modifications to the Java Pet

Store, Oracle9iAS proved to be over 22 times faster than .NET in some cases, while using far fewer resources.

Hardware and Software Configurations

For this Oracle9iAS vs. .NET benchmark, Oracle used the same mid-tier and database systems that were used for the benchmark in June 2001. The hardware used by Microsoft was somewhat faster than this hardware.

Machine Configuration

Host	CPU	Memory	OS
Clients	Client 1: 2 x 450 MHz	1 GB	Solaris 2.7
	Client 2: 2 x 450 MHz	1 GB	Solaris 2.8
	Client 3: 2 x 360 MHz	512MB	Solaris 2.6
	Client 4: 2 x 360 MHz	512MB	Solaris 2.6
	Client 5: 2 x 360 MHz	512MB	Solaris 2.6
Mid Tier	2 x 450 MHz	1 GB	Solaris 2.7
Database	4 x 400 MHz	1 GB	Solaris 2.8

The same versions of the software used in the earlier test were used again.

Software

Application Server	Oracle9iAS 1.0.2.2.
JVM	1.3.1_01
EJB Isolation Level	Read Committed
Database Software	Oracle 8.1.7.

Description of Benchmark Test Execution

Oracle replicated the test execution environment and workload using the same hardware that was used in the original benchmark published in June 2001. The tests were repeated for multiple iterations to ensure consistent results.

Test Execution

Ramp Up	10 minutes
Measurement Interval	25 minutes
Think Time	5 seconds - 35 seconds (20 second average)

Oracle ran a series of tests for 500 to 5000 users to compare the results of Oracle9iAS and Microsoft's .NET. The tests were run in two deployment configurations. The first series of tests did not use any caching technology, and can be compared with the .NET results without a cache. The second series of tests uses Oracle9iAS Web Cache, and can be compared with the .NET results with Microsoft's output cache enabled.

With the application changes, the new tests were much lighter than the original tests. Therefore, we were able to run a much higher user load than the original 450 users on the same hardware configuration. Like Microsoft, we increased the number of client systems to emulate more users. *Interestingly, unlike Microsoft, we found that we required only 5 and not 90 clients to drive a high user load.*

Benchmark Results

The following results clearly demonstrate that Oracle9iAS offers much better performance (average response time) and scalability (% CPU utilization) than .NET. In both deployment configurations that Oracle tested (without caching and with caching), Oracle9iAS delivered significantly better results than Microsoft .NET.

The following tables and graphs illustrate our results.

Average Response Time (MilliSeconds)				
Users	Oracle9iAS (No Caching)	.NET (no output caching) ⁽¹⁾	Oracle9iAS with Web Cache	.NET (output cache enabled) ⁽¹⁾
450	35	63	11	34
1000	37	111	10	73
1500	38	Not Tested	13	Not Tested
2000	42	231	13	145
2500	46	479	14	198
3000	50	900	16	360
3500	57	Not Tested	19	1034
4000	69	Not Tested	21	Not Tested
4500	97	Not Tested	25	Not Tested
5000	302	Not Tested	29	Not Tested

⁽¹⁾ - From Benchmarking Microsoft® .NET vs. Sun® Microsystem's J2EE™ Version 1.5, November 2001 at

<http://www.gotdotnet.com/team/compare/Benchmark%20Results%20for%20Microsoft%20.NET%20vs.%20Sun%20J2EE.doc>

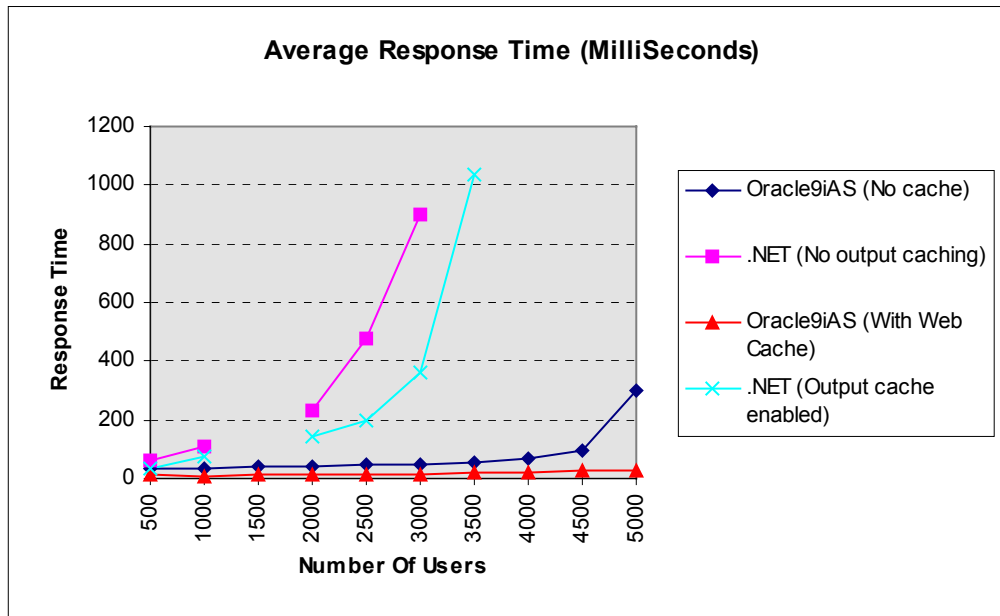


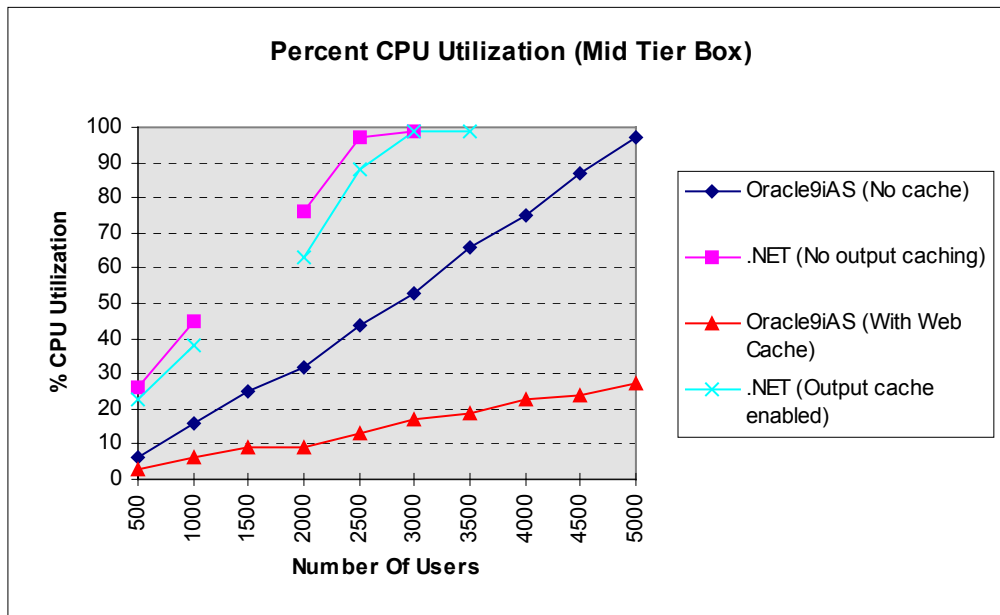
Figure 1: Response Time vs. User Load Graph

Middle-tier CPU Utilization

Middle Tier % CPU Utilization				
Users	Oracle9iAS (No Caching)	.NET (no output caching) ⁽²⁾	Oracle9iAS with Web Cache	.NET (output cache enabled) ⁽²⁾
450	6	26	3	23
1000	16	45	6	38
1500	25	Not Tested	9	Not Tested
2000	32	76	9	63
2500	44	97	13	88
3000	53	99	17	99
3500	66	Not Tested	19	99
4000	75	Not Tested	23	Not Tested
4500	87	Not Tested	24	Not Tested
5000	97	Not Tested	27	Not Tested

⁽²⁾ - From Benchmarking Microsoft® .NET vs. Sun® Microsystem's J2EE™ Version 1.5, November 2001 at: <http://www.gotdotnet.com/team/compare/Benchmark%20Results%20for%20Microsoft%20.NET%20vs.%20Sun%20J2EE.doc>

Figure 2: Middle-tier CPU Utilization Chart



Summary

Based upon these results, it is evident that Oracle9iAS without Web Cache delivers up to 18 times better performance than Microsoft .NET without caching. Oracle9iAS with Web Cache was able to beat Microsoft .NET with caching by a factor of 22. An interesting observation is that Oracle9iAS

without the Web Cache was up to 7 times faster than .NET with caching. Oracle9iAS also scales better by utilizing CPU more efficiently than .NET. In summary, Oracle9iAS and J2EE demonstrate far superior performance when compared to Microsoft .NET.