WHITE PAPER

Benchmark overview SPECjbb2005

Version 1.2
February 2006

Pages 7

Abstract

This document presents the benchmark SPECjbb2005 (Java Server Benchmark) which was developed and published by the Standard Performance Evaluation Corporation (SPEC).

SPECjbb2005 evaluates the performance of server side Java by emulating a three-tier client/server system (with emphasis on the middle tier). It is generally accepted by the industry. While it is not a full blown OLTP benchmark, it is a very good stand-in for business applications. It is also a very good functional and performance test for Java platforms.

The benchmark exercises the implementations of the JVM (Java Virtual Machine), JIT (Just-In-Time) compiler, garbage collection, threads and some aspects of the operating system. It also measures the performance of CPUs, caches, memory hierarchy and the scalability of shared memory processors (SMPs). The benchmark concept realized in SPECjbb2005 allows to make across-the-board manufacturer comparisons based on the benchmark results.

Contents

SPECjbb2005 – An Overview

The „Standard Performance Evaluation Corporation“ or briefly SPEC is an organization specialized in the development and issue of standardized benchmarks to evaluate the performance of computer systems. Members of the SPEC consortium are the leading companies of the computer industry such as Fujitsu. One of the benchmarks issued by SPEC and the successor to SPECjbb2000 (retired 01/2006) is SPECjbb2005. The purpose of this document is to provide an overview of this benchmark and to explain its results.

Objective of the Benchmark

When comparing systems, in principle the individual customer application would be the measure of all things. However, due to shortage of time or for complexity or other reasons you often do not have the opportunity to test a variety of possible systems with a customer application in order to find out which is the most suitable one. Benchmarks have been developed to nevertheless provide customers with objective criteria when choosing their system. These are standardized tools, measuring the speed of individual system components or the speed of the entire system.

JBB stands for Java Business Benchmark. The benchmark emulates a three tier client/server system, which is typical for modern business processing applications:

- Clients or drivers are generating load
- The middle tier implements the business logic and updates a database
- The database tier makes these updates permanent

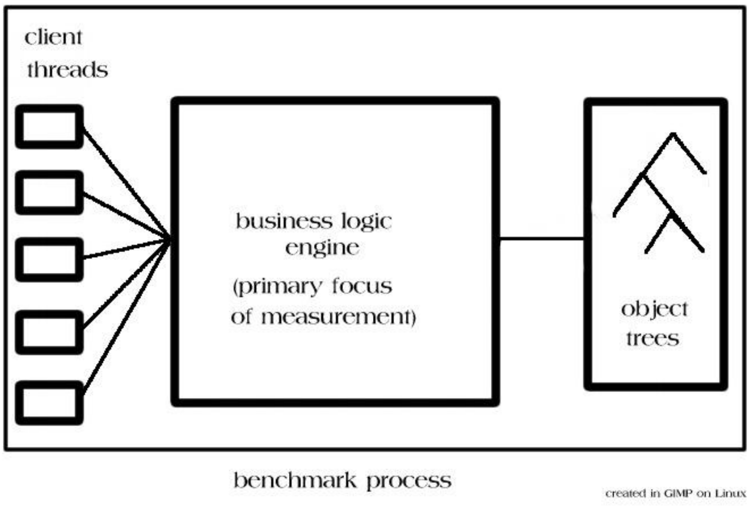
SPECjbb2005 measures the performance of a Java based middle tier. In order to be self-content it brings the two other tiers along in rudimentary form. They are running on the same host as the middle tier and are also written in Java. The great advantages are that the benchmark can be run on a single server without setting up large hardware installations and SPECjbb2005 results can be compared straightforward across different system architectures.

On the hardware side SPECjbb2005 measures processor and memory performance. It executes neither disk nor network I/O. On the software side it measures the performance of the Java Virtual Machine (JVM), just-in-time compilation (JIT), garbage collection, user threads and some aspects of the operating system.

Components of the Benchmark

SPECjbb2005 contains three logical components running on one single host:

- The client tier consists of driver threads, which generate operations without think times to maximize utilization
- The dominant middle tier implements the business logic inspired by the TPC-C database transaction benchmark, modeling the order processing of a wholesale supplier
- The database tier is replaced by in-memory Java Collection objects, and transaction logging is implemented using XML



SPECjbb2005 requires only the operating system and a Java Virtual Machine (JVM) supporting J2SE 5.0 features.

The scaling unit is a warehouse of about 25 MB of Java objects. Exactly one Java thread per warehouse executes operations against the objects. The business operation types are taken from TPC-C:

- new order entry
- payment
- order status inquiry
- delivery
- stock level supervision
- customer report

However at this abstract conceptual level the similarity between TPC-C and SPECjbb2005 ends. Their results are by no means comparable.

The application code of the benchmark was written as a highly scalable parallel application. How well it scales in a particular configuration depends largely on the capabilities of the underlying hardware and software components.

SPECjbb2005 Test Sequence

A full compliant benchmark run consists of a sequence of measurement points with an increasing number of warehouses (and thus an increasing number of threads).

- The benchmark run starts at 1 warehouse up to 8 warehouses (minimum) or up to 2*_MaxWh_ warehouses, whichever is higher.

- _MaxWh_ is the expected peak number of warehouses, which is usually defined by the number of processors in the SUT (System Under Test). If _MaxWh_ is not equal to the number of processors, it must be documented in the result report.

- The sequence must increment by 1 warehouse and 1 thread respectively for each measurement point.

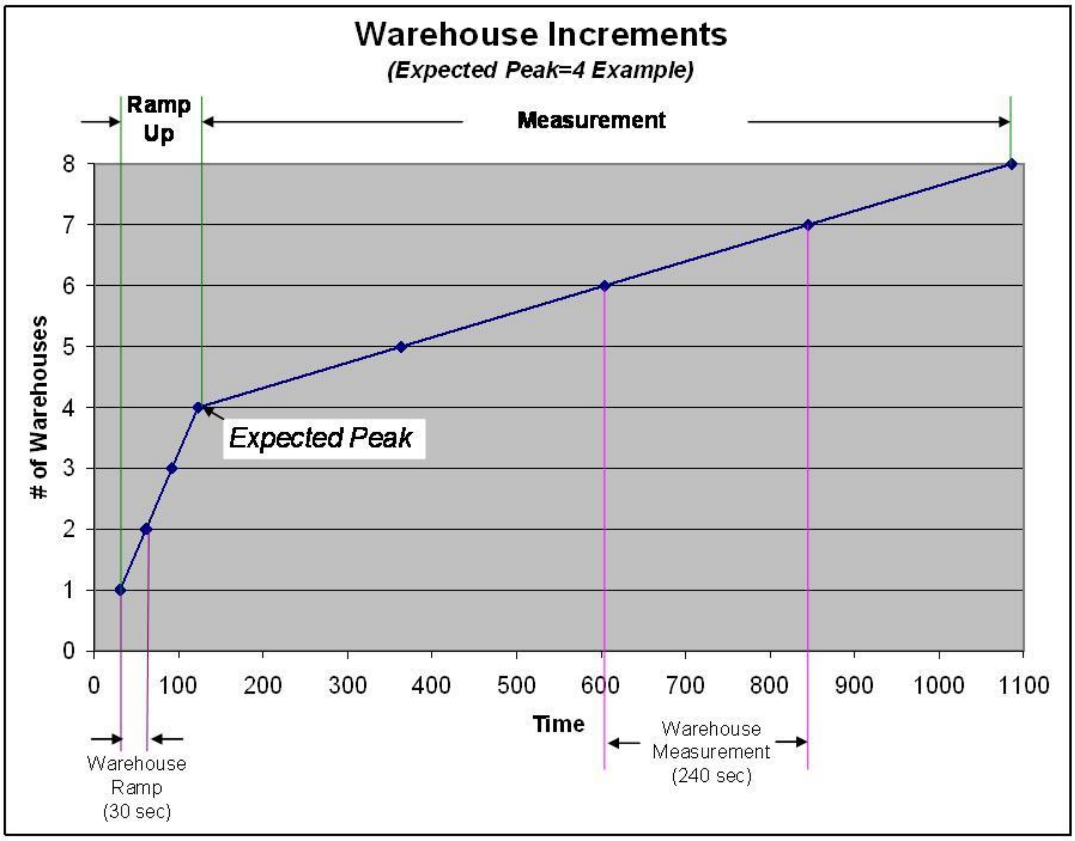
- The test may be configured to run beyond 2*_MaxWh_ warehouses. The points beyond the 2*_MaxWh_ point will appear in the report, but will not be used to calculate the metric.

- For numbers of warehouses from 1 to the expected peak number _MaxWh_ -1 of warehouses, there is a 30-second measurement period. For larger numbers of warehouses, the measurement period is four minutes (240 seconds).

- The ramp-up time for each measurement point, which is the time between the start of the run of a single measurement point and the beginning of the actual measurement, is 30 seconds.

- All JVM instances run these measurement periods together. It is ensured by the benchmark that the measurement intervals for all instances occur during the time all instances are running.

The following graph shows the test sequence of a compliant SPECjbb2005 benchmark run. The expected peak number of warehouses _MaxWh_ is 4 in this example.



Measurement Results and their Interpretation

SPECjbb2005 has two performance metrics:

- **SPECjbb2005 bops** (business operations per second), which is the overall rate at which the different business operations are performed per second

- **SPECjbb2005 bops/JVM**, the first metric divided by the number of JVMs running during the benchmark

Both performance metrics must be stated when SPECjbb2005 benchmark results are compared to each other.

The metric **SPECjbb2005 bops** is calculated as follows:

The calculation of the overall SPECjbb2005 bops metric is based on the calculation of the throughput of a measurement period of work done at a given number of warehouses. For each measurement point of a benchmark run the following performance data is logged:

```
Calculating results

Minimum transactions by a warehouse = 470224
Maximum transactions by a warehouse = 472731
Difference (thread spread) = 2507 (0.53%)

===============================================================================
TOTALS FOR: COMPANY with 2 warehouses
................ SPECjbb2005 0.06 Results (time in seconds) ................
                   Count     Total    Min     Max     Avg        Heap Space
    New Order:    414459    223.27   0.000   2.084   0.001       total  254.1MB
      Payment:    285744     75.24   0.000   2.082   0.000        used  219.7MB
  OrderStatus:     28574      9.42   0.000   0.043   0.000
     Delivery:     28575     61.09   0.001   0.068   0.002
  Stock Level:     28574     11.51   0.000   0.042   0.000
  Cust Report:    157029     81.42   0.000   0.132   0.001

           throughput =    3928.64 bops

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

The table shows information about the minimum and maximum amount of work done by the threads/warehouses at that point. The following numerical results for all six transaction types during the measurement interval are printed:

- Count is the total number of transactions
- Total is the total response time in seconds
- Min is the minimal response time in seconds
- Max is the maximal response time in seconds
- Avg is the average response time in seconds
- Heap Space shows the total and the actually used heap space of the JVM
- throughput is the sum of all Counts of each type of transaction divided by the measurement interval in seconds.

In this example the throughput is computed as follows:

throughput = (414459 + 285744 + 28574 + 28575 + 28574 + 157029) / 240 sec = 3928.98 bops

This small difference to the throughput value in the table is caused by the more accurate benchmark internal timer working in milliseconds.

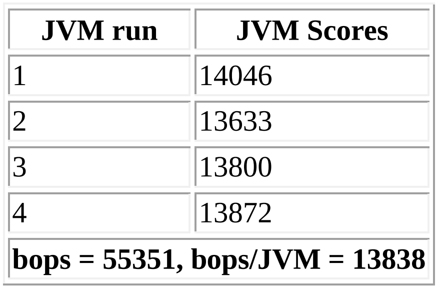
The overall SPECjbb2005 bops metric for a compliant run is now calculated as follows:

The total throughputs of all the JVM instances for all the points from *MaxWh* warehouses to 2**MaxWh* inclusive warehouses are averaged. This average is the metric ”SPECjbb2005 bops”.
For all the missing points in the range *MaxWh*+1 to 2**MaxWh*, the throughput is considered to be 0 bops in the metric computation.

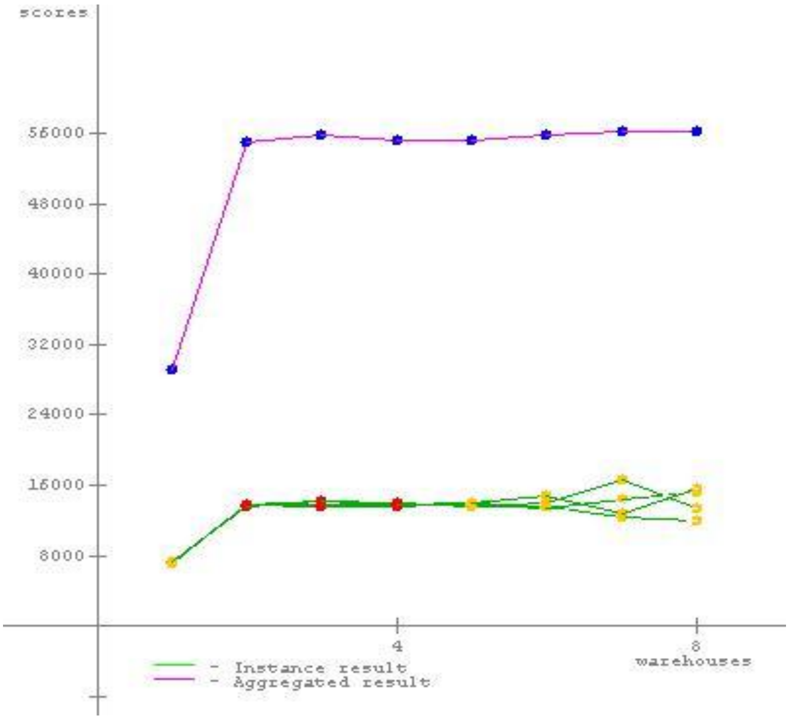
The second multi-JVM metric **SPECjbb2005 bops/JVM** is obtained by dividing value of bops for the run by the number of JVM instances used in the run.

The reporting tool contained within SPECjbb2005 produces a graph of the throughput at all the measured points with warehouses on the horizontal axis and throughputs on the vertical axis. All points from 1 to the maximum of 8 or 2*_MaxWh_ are reported. Missing points in the range _MaxWh_+1 to 2*_MaxWh_ will be reported to have a throughput of 0 bops. The points being averaged for the metric will be marked on the report. Additional graphs for all individual JVMs are also included in the graph.

The following example is an extract from a SPECjbb2005 result sheet . It shows the result of a compliant SPECjbb2005 run with 4 JVMs.



JVM run	JVM Scores
1	14046
2	13633
3	13800
4	13872
bops = 55351, bops/JVM = 13838	



The most significant influences on the performance of the benchmark are:

- the number of processors and processor characteristics
- memory subsystem
- system bus
- the operating system capabilities
- the Java runtime environment (JVM)
- address space support (32-bit vs. 64-bit)

SPECjbb2005 should not be used to size a Java server configuration, because it is based on a specific workload. There are numerous assumptions made about the workload, which might or might not apply to other user applications. Also, all the operations that would be database operations are memory operations in SPECjbb2005. SPECjbb2005 is a tool that provides a level playing field for comparing JVM products in a server environment.

Literature

PRIMERGY Systems	http://ts.fujitsu.com/primergy
PRIMERGY Performance	http://ts.fujitsu.com/products/standard_servers/primergy_bov.html
SPECjbb2005	http://www.spec.org/osg/jbb2005

Contact

PRIMERGY Hardware
PRIMERGY Product Marketing
mailto:Primergy-PM@ts.fujitsu.com

PRIMERGY Performance and Benchmarks
PRIMERGY Performance and Benchmarks
mailto:primergy.benchmark@ts.fujitsu.com

Delivery subject to availability, specifications subject to change without notice, correction of errors and omissions excepted.
All conditions quoted (TCs) are recommended cost prices in EURO excl. VAT (unless stated otherwise in the text). All hardware and software names used are brand names and/or trademarks of their respective holders.

Published by department:

Enterprise Products
PRIMERGY Server
PRIMERGY Performance Lab
mailto:primergy.benchmark@ts.fujitsu.com

Internet:
http://ts.fujitsu.com/primergy
Extranet:
http://partners.ts.fujitsu.com/com/products/servers/primergy