

SPECjAppServer2004 Performance Testing with JBoss Application Server

Using Hewlett-Packard Integrity
Servers running HP-UX 11i v2

What is the SpecjAppServer Test

- **The Standard Performance Evaluation Corporation (SPEC)** is a non-profit corporation formed to establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers. SPEC develops benchmark suites and also reviews and publishes submitted results from our member organizations and other benchmark licensees.

JBoss & HP Working Together

- Jboss & HP are collaborating on the SpecjAppServer2004 benchmark
- We use the test to develop better performance for JBoss and the HP-UX Java environment.
- Our performance has become much better since we started testing.
- Below we share some of the tuning methods and tuning parameters we have found useful in increasing the Application Server's performance.

What is the SpecjAppServer Test-cont

- SPECjAppServer2004 is an end-to-end application which exercises all major J2EE technologies implemented by compliant application servers as follows:
- The web container, including servlets and JSPs
- The EJB container
- EJB2.0 Container Managed Persistence
- JMS and Message Driven Beans
- Transaction management
- Database connectivity

What is the SpecjAppServer Test-cont

The Spec Organization requires that members only publish results that have been officially submitted and undergone peer review.

Other members review the results (which must include most configuration details of the tests) and can challenge the published results.

All software/hardware used must be available within 3 months.

Only "waived" anomalies in the Specj results are allowed. "Results" are response times to requests and reported errors.

What is the SpecjAppServer Test-cont

- Specj Application Server is based on the J2EE 1.3 (now 1.4) Standard. It requires that the Application Servers be Java J2EE compliant.

(For example, we can have better performance making procedure calls "call-by-reference" rather than "call-by-value" (as the J2EE standard compliance requires).

But since all J2EE applications must be inherently distributed, procedure calls must pass the parameters on the calling stack rather than just as a pointer to the data structures.)

What is the SpecjAppServer Test-cont

- You need to license the test from SPEC.
- JBoss has provided a kit to help run the Specj2004 test with the JBoss AS
SPECjAppServer2004Setup
(best to search spec.org site using "jboss" search term to find the kit's location).

What is the SpecjAppServer Test-cont

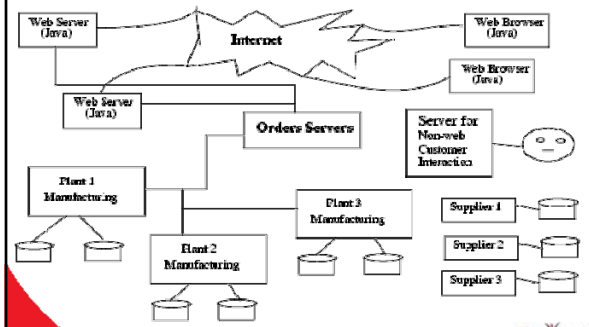
The SpecjAppServer benchmark emulates an automobile manufacturing business.

Five business domains are modeled : Dealer, Manufacturing, Supplier, Customer, and Corporate.

In a company of the size being modeled, it is assumed that each business domain, has separate databases and applications. Most likely, they are implemented on separate computing hardware also. There are producer-consumer relationships between domains in the company and to outside suppliers and customers as well.

The cost of the solution is an important component of the benchmark.

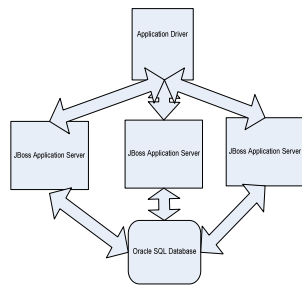
The SpecjAppServer Test Simulation



Basic SpecjAppServer Test Setup



Multi-node SpecjAppServer Test Setup



JBoss Application Server

- We are currently using the JBoss 4.0.4.GA Release.

In the multi-node test, the Application Servers are setup as a cluster.

Also DNS round-robin is used to distribute the load between the 3 SUTs.

HP-UX Hardware

- We use HP-UX rx2620 1.6 GHz 6MB Cache– 2 way Integrity (IA) processors with 6 GB of memory running HP-UX 11i v2 for our Systems Under Test (SUT) JBoss AS's.
- We have a private 1Gb link connecting the SUTs, emulator, and driver.
- We also have a separate private 1Gb link connecting the SUTs to the SQL Database server.
- The emulator is a separate rx2620 2-way system also with 6GB of memory.

13



Running the SpecJAppServer Test

- The "**SPECjAppServer2004 JOPS**" is the primary SPECjAppServer2004 metric and denotes the average number of successful "jAppServer Operations Per Second" completed during the Measurement Interval. "**SPECjAppServer2004 JOPS**" is composed of the total number of Business Transactions completed in the Dealer Domain, added to the total number of work orders completed in the Manufacturing Domain, normalized per second.
- $\text{SPECjAppServer2004 JOPS} = \text{Dealer Transactions/sec} + \text{Workorders/sec}$

14



Running the SpecJAppServer Test

- The JOPS is a function of the **Injection Rate** of new business transactions requests from the Dealer application in the Dealer Domain being injected into the SUT.
- We generally refer to this term as the "transaction rate" (txRate). The test is scaled up by increasing the transaction rate.
- A successful test has no errors and the JOPS is generally about 1.7 times the txRate.

15



Running the SpecJAppServer Test

- By using the performance tools described below, we monitor the SUTs and the Data Base system.
- When the CPUs are not fully utilized, either multi- or single node, we know that we can support a higher txRate by tuning the application.
- At the transaction rates that can be supported by 2-way systems, even in a multi-node setup, the database should not be a performance limiting factor.

16



Running the SpecJAppServer Test

- The App Server test consists of 4 separate states:
 - o **Trigger Time**
initialize test scripts and invoke the emulator.
 - o **Ramp-up**
populates the data base with initial transaction values.
 - o **Steady-state**
Normal transaction processing over at least a 60 minute period.
 - o **Ramp-down**
The steady-state interval must be maintained for at least 5 minutes after the measurement run.

17



Running the SpecJAppServer Test

- Note that one interesting test is the "Cache Validity Test".
Ramp-up: The driver selects a random item and increments it's price by \$2,000.00. The driver then updates the new price to the database directly using the JDBC interface. The driver also saves a copy of the new price.
Ramp-down: The driver finds the item it has updated during ramp-up by calling the entity bean. Then it gets the price from the entity bean. The price obtained from the entity bean during ramp-down must reflect the previously updated price. Otherwise, the entity bean's state would not have been refreshed by the new values in the database and the test will fail.

18



HP-UX Performance Tools

- **glance** – The Global or Process Summary. More specific data is presented via the CPU, Memory, Disk I/O, LAN, Swap and System Table detail screens.
- **sar** – used to report cpu utilization and disk utilization (especially of the DB server).
- **top** - display and update information about the top processes on the system, ranking processes by CPU utilization.
- **HPjmeter** –
 - o Graphic display of profiling data
 - o Call graphs with call count, or with CPU or clock time
 - o Per thread display of time spent in 9 different states
 - o Per thread or per process display

19



HP-UX Performance Tools

- **HPjmtune** - Java Garbage Collection (GC) visualization tool for analyzing garbage collection activity in a Java program.
 - o using the output of -Xverbosegc file.
 - o users can perform multiple simulations changing -mx, -ms, SurvivorRatio and calls to System.gc to see what effect these have.

20



Running the Spec jApp Server Test

- The construction of the Java Virtual Machine that the JBoss Application Server (and the driver) execute in is very important.
- Tuning the Java options can dramatically increase performance.
- Below are some of the options we have found useful:

21



Performance Java Options

```
JAVA_OPTS="-Xmx2200m -Xms2200m -Xmn1100m \  
-XX:PermSize=96m -XX:SurvivorRatio=14 -Xoptgc \  
-XX: +ForceMmapReserved XX:UseHighResolutionTimer \  
-XX: -StackTraceInThrowable \  
-XX: SchedulerPriorityRange=SCHED_NOAGE \  
-XX: +DisableExplicitGC -XX: -UseFastAccessorMethods \  
-Djava.nio.channels.spi.SelectorProvider:\   
"sun.nio.ch.DevPollSelectorProvider" \  
-XX: +UseSpinning "
```

22



Explanation of Java Options

- **Xmx2200m - Xms2200m** -> set the minimum and maximum Heap Size to 2.2GB to pre-set Heap allocation. Removes constant resizing of JVM by pre-allocating the entire Heap.
- **Xmn1100m** -> Size of Heap for New generation
- **PermSize=96m** -> Size of Heap reserved for persistent objects
- **SurvivorRatio=14** -> <SurvivorRatio> defines the ratio between one of the two survivor spaces and the eden heap.
eden : survivor = <SurvivorRatio> : 1
- **Xoptgc** -> Optimized Heap garbage collection. Also overrides application calls for garbage collection so its only done when absolutely needed. Normally, we only have to do savaging gc and never a full gc during a 60 minute SPEC jAppServer run.

23



Explanation of Java Options

- **+ForceMmapReserved** – Allocates Heap space for large objects immediately; also uses large 4 MB pages.
- **UseHighResolutionTimer** - diminishing the impact of date/time functions (which are atomic operations in the operating system) used by the JVM executing this program.
- **-StackTraceInThrowable** – Display a stack trace for an exception – useful for debugging.
- **SchedulerPriorityRange=SCHED_NOAGE** – do not lower the process's scheduling priority over time
- **-UseFastAccessorMethods** - Disables server compiler feature (otherwise on by default) to workaround a type profiled inlining bug that will be fixed in upcoming J2SE 5.0 update releases.

24



Explanation of Java Options

- **DisableExplicitGC** - Disable application calls to System.gc(), JVM still performs garbage collection when necessary.
- **-Djava.nio.channels.spi.SelectorProvider: "sun.nio.ch.DevPollSelectorProvider"** – use /dev/poll method of learning which sockets have new data – active sockets are identified by event model, doesn't require active scanning of file descriptors.
- **+UseSpinning** - Enable naive spinning on Java monitor before entering operating system thread synchronization code.



25

Profiling Java Options

- **-verbosegc:file=/tmp/vgc\$\$out** – Java garbage collection output file that describes gc actions. It records:
 - o How frequent is garbage collection? Note time stamps between collections
 - o How much time does gc take?
 - o What is percentage of free heap space?
- **eprof:time_on=1200,time_slice=600,\file=/tmp/eprof.\$\$out** – sets interval and duration of profile data collection interval. Above, 15 minutes for ramp-up to complete, the steady-state can run for 5 minutes, and then collect profiling data for 10 minutes. (The profiling data is analyzed with HPjmeter.)



26

Oracle Database

- The Oracle Database is run on an rx4680 4-way Integrity servers running HP-UX 11i v2.
- Striped across Compaq MSA-1000 Disk Array
- Oracle version 10.1 (with Oracle patches)
- The Specj test supplies the SQL scripts with which we load the database for the test. Some tables are split to increase parallel access.
- The Database performance does not limit our SUT/driver performance.



27

3 node Testing – Load Distribution

- Uses DNS round-robin to spread load between 3 AS SUTs.
- DNS round-robin requires BIND 9.3 (otherwise its just best-effort rather than real round-robin)
- Update the forwarder with:

```
rrset-order { order cyclic; };
```
- Use "hosts_to_named" utility to convert the /etc/hosts file into the appropriate Internet domain name server (named) configuration files.
- For detailed configuration information, go to www.hp.com and search for "OSRA Configuration Guide."



28

3 node Testing – Load Distribution

- Note that the driver Java parameters must include the following options:

```
-Dnetworkaddress.cache.ttl=0
-Dsun.net.inetaddr.ttl=0
```

These options tell the Java environment not to cache the results of a DNS lookup, but to re-ask DNS for the address resolution each time. The round-robin DNS will then cyclically return each address in the specified robin-robin system pool.

Note that DNS address caching must also be disable in the BIND configuration.



29

Multiple Application Servers per node

- You can run multiple application servers on the same node using virtual IP addresses.

On HP-UX, you define a virtual IP address:

```
ifconfig lan0: <num> <ip address>
netmask <ip mask> up
```

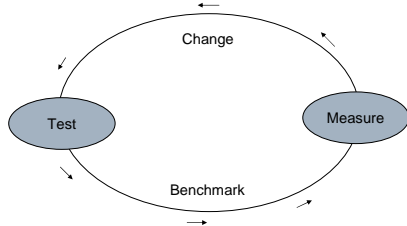
We did not achieve a performance benefit using multiple ASes/node.



30

Tuning Methodology

- Track your changes



31

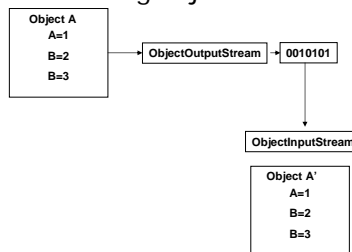
Recent Improvements

- JBoss Serialization
 - ✓ Meta-Data / Reflection intensive
 - ✓ Singleton HashMap for the Metadata
- CMP
 - ✓ New container (cmp2)
- Reduced Contention

32

Java Serialization

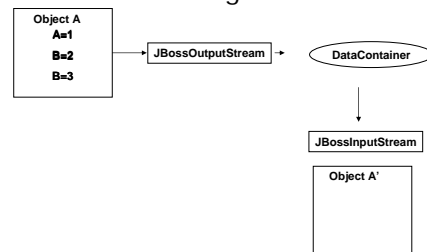
- Cloning objects with serialization



33

JBossSerialization

- Smart Cloning



34

Tuning GC

- GC Verbose Logging
- Different GC models
 - ✓ Single Generation
 - ✓ Multiple Generation
 - ✓ Parallel
 - ✓ Concurrent
- Large Page sizes
- Thread Stack size

35

Identifying Bottlenecks

- Using System Tools
 - ✓ Thread Dumps
 - ✓ iostat
 - ✓ vmstat
 - ✓ top
 - ✓ JDK5 JMX console
 - ✓ JBoss JMX console

36

Questions?

37

Background Slides

38

Spec Java Application Server Objectives

Golden Rules for SPECJAppServer2004 Workload

- **Showcase Enterprise Components**
- **Fully distributed** The business problem should necessitate use of worldwide services and data whose exact location is not known a priori.
- **Redundant & Available Services** Services should be redundant and fully available with the runtime mapping unknown to the application developer.
- **Middleware Focus** Strive to stress the middle-tier (whether this is logical or physical), rather than the client tier or the database server tier.
- **Scalable business domain** As the size of the modeled business grows, the services, data, users, and number of geographically distributed sites should also grow.
- **Universality** An enterprise bean should be capable of being deployed on any vendor's H/W and O/S, as long as a compliant application server is present.
- **Command Credibility Real-world** The performance workload should have the characteristics of real-world systems.
- **Complexity** The workload should capture intra-, extra-, and inter-company business processes.
- **Openness** The workload should be implementable atop all popular full-function J2EE application servers.
- **Neutrality** No workload features are selected with the goal of making any particular product look good (or bad). Fidelity to real-world business requirements is paramount.
- **Conform To Programmer Craft** RAD/IDE application development The workload should be developed as if by a customer using reusable components intended for standardized deployment.
- **Scoped development skills** The implementation should assume Fortune 500 developers who understand business logic, not systems programming.
- **Pattern Spec to Best Industry Practice**
- **Simplicity** The workload should be easy to understand, straightforward to implement, and run in reasonable time with excellent repeatability.
- **Industry-standard Benchmark** The workload should take into account the price of the system it is run on. Aspects of everyday DBMS scalability (e.g., database I/O, concurrency, memory management, etc.) are not.

39

HP-UX Performance Tools -- glance

- **HPjmeter** -- * Multi-window GUI with Mark and Find features * Over 15 metrics, including:
 - Method Call Count - Inclusive and Exclusive Method Times (CPU or Clock)
 - Interactive Call Graph with Call Count, or with CPU or Clock Time, if available
 - Allocated and Residual Objects with Allocation Sites, Object Reference Graph
 - Derived metrics, such as Average Execution Times or Class Times
 - * Per Thread or Per Process display
 - * Lock Contention, Inlining, Thrown Exceptions, and Memory Leaks heuristics
 - * Compare capability for performance improvement tracking

Additional features for the HP-UX version of the HotSpot VM:

- * Low overhead of the Extended Profiling Agent (the -Xeprof option)
- * Many more metrics, including:
 - CPU and Clock Times for all Java methods
 - Lock Contention metrics - Threads Histogram showing the lifetime of Java threads and distribution of thread state over the thread lifetime, including garbage collection and thread starvation
 - more derived metrics * CPU times adjusted to compensate for profiling intrusion

40

Recent SpecJAppServer Results

Note that we evaluate JOPS results for comparison "per core." This means that the number of JOPS is divided by the number of CPUs (or cores) that produced the results.

This is not an official SPECJAppServer measure, but we find it useful in relating results.

Per core results give a indication of the processing efficiency of the SUTs.

High JOPS with many cores, however, also indicate the scaling efficiency of the multi-core system.

Please refer to www.spec.org for more information.

41

Tester	System Title	JOPS	J2EE Server Nodes	J2EE Server CPU Description	J2EE Server Instances	DB Server Instances	JOPS/core
BEA Systems, Inc	BEA WebLogic Server 9.0 HP DL380	1374.11	5	10 cores, 10 chips	5	1	137.41
BEA Systems, Inc	BEA WebLogic Server 9.0 HP DL380	1664.36	6	12 cores, 12 chips	6	1	138.70
HP	BEA WebLogic Server 9.0 on HP-UX rx4640	471.28	1	4 cores, 4 chips	4	1	117.82
HP	BEA WebLogic Server 9.0 on HP-UX rx4640	538.03	1	4 cores, 4 chips	1	1	134.51

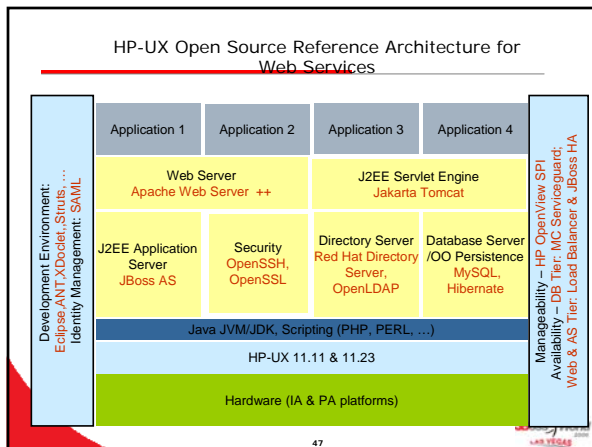
42

Tester	System Title	JOPS	J2EE Server Nodes	J2EE Server CPU Description	J2EE Server Instances	DB Server Instances	JOPS/core
HP	BEA WebLogic Server 9.0 on HP-UX rx4640 Cluster	3734.68	8	32 cores, 32 chips	8	1	116.71
HP	BEA WebLogic Server 9.1 on Red Hat Linux rx4640	542.17	1	4 cores, 4 chips	1	1	135.54
IBM Corporation	WebSphere 5.1 Application Server on eServer xSeries 335 Cluster	1306.44	11	22 cores, 22 chips	11	1	59.38
IBM Corporation	WebSphere 5.1 Application Server on eServer xSeries 335 Cluster	900.32	8	16 cores, 16 chips	8	1	56.27
				43			

Tester	System Title	JOPS	J2EE Server Nodes	J2EE Server CPU Description	J2EE Server Instances	DB Server Instances	JOPS/core
IBM Corporation	WebSphere 6.0 Application Server on IBM eServer OpenPower 720 Cluster	1334.96	4	16 cores, 8 chips (SMT on)	4	1	83.44
IBM Corporation	WebSphere 6.0 Application Server on eServer xSeries 365	1343.47	5	20 cores, 20 chips	5	3	67.17
IBM Corporation	WebSphere 6.0.2.3 Application Server on IBM System p5 550	2921.48	8	32 cores, 16 chips (SMT on)	8	1	91.30
Sun Microsystems Inc.	BEA WebLogic Server 9.0 on Sun Fire T2000 Cluster	3328.80	6	48 cores, 6 chips	6	1	69.35
				44			

Tester	System Title	JOPS	J2EE Server Nodes	J2EE Server CPU Description	J2EE Server Instances	DB Server Instances	JOPS/core
Sun Microsystems Inc.	BEA WebLogic Server 9.0 on Sun Fire T2000 Cluster	4098.77	7	56 cores, 7 chips	7	1	73.19
Sun Microsystems Inc.	BEA WebLogic Server 9.0 on Sun Fire T2000	615.64	1	8 cores, 1 chip	1	1	76.96
Sun Microsystems Inc.	BEA WebLogic Server 9.0 on Sun Fire X4100 Cluster	1781.47	5	20 cores, 10 chips (2 cores/chip)	5	1	89.07
Sun Microsystems Inc.	Sun Java System Application Server 8.1 2005Q1 Standard Edition on V20z Cluster	1201.44	13	26 cores, 26 chips (1 core/chip)	13	1	46.21
				45			

Tester	System Title	JOPS	J2EE Server Nodes	J2EE Server CPU Description	J2EE Server Instances	DB Server Instances	JOPS/core
Sun Microsystems Inc.	Sun Java System Application Server 8.2 Platform Edition on T2000	436.71	1	8 cores, 1 chip, 8 cores/chip (4 threads/core)	3	1	54.59
Sun Microsystems Inc.	Sun Java System Application Server Platform Edition 8.1 2005Q1 URI on V20z Cluster	298.51	3	6 cores, 6 chips (1 core/chip)	3	1	49.75
Sun Microsystems Inc.	Sun Java System Application Server Platform Edition 8.1 2005Q1 URI on V20z Cluster	266.01	3	6 cores, 6 chips (1 core/chip)	3	1	44.34
				46			



- ### HP-UX OSRA web links
- **HP-UX Open Source page:**
<http://h20338.www2.hp.com/hpux111/cache/323704-0-0-0-121.html>
 - **HP Open Source Reference Architecture:**
<http://h20293.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=OSRA>
 - **JBoss Application Server on HP systems:**
<http://h20293.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=JBOSS>
 - **JEMS Support on HP systems:**
<http://h20293.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=JBOSS-SUPPORT>
 - **OSRA Configuration Guide:**
<http://docs.hp.com/en/7733/osra-config-apr13.book.pdf>
 - **HP-UX 11:**
<http://www.hp.com/go/hpux>