

How not to do a Performance Benchmark

Jaqui Lynch
Solutions Architect
Forsythe Technology Inc.
lynchj@forsythe.com



1

AGENDA

- Common mistakes
- What is a benchmark
- Why bother
- What free UNIX benchmarks are there
- Moving forward



2

Common mistakes

- No idea what a benchmark really is
 - No real reason to do one
 - No statement of work
 - No properly defined scope
 - No definition of success
 - No standardized output reporting
 - No planning
 - Insufficient time allocated
-
- Basically which kind of benchmark is being performed and what is the purpose?

3



What is a benchmark?

1. Sizing test or study
 2. Proof of Concept
 3. Functionality test (does box perform as specified)
 4. Test of new or current technologies
 5. Competitive comparisons
-
6. Make sure you understand which one you are talking about as it makes a difference to the testing

4



Why bother?

- Sizing studies are not always correct
- Burned before with bad sizing (for whatever reason)
- Need to test that application, operating system or hardware performs the way you expect
- Proof of Concept
 - Does this application have all the functionality I need?
- Competitive Bake offs
 - Different hardware vendors
 - Different applications
- Test of new or current technologies
 - Should I go to the latest technology or stay with the one I am on now?

5



What performance benchmarks can I run?

- CPU and/or Memory
 - Dhrystones
 - Stream
 - Unixbench 4.0.1
 - TPC-h (may need a license)
 - Custom C or other code
 - nstress
- Network
 - httpperf
 - iPerf
 - Netperf
- Java
 - Specjvm2008
 - SpecjAppserver2004 (need a license)
- Oracle
 - Swingbench
- Check out the following sites for other benchmarks
 - <http://lbs.sourceforge.net/>
 - <http://www.netlib.org/benchmark/>

6



Dhrystones

- C benchmark written initially in 1984
- Replaced by SpecInt (you pay for this one)
- Measures integer performance (no floating point)
- V2.1 (1988) fixes many of the problems with over optimization
- Designed more for PCs as it really tests out core speed (GHz)
- Not designed to test throughput on technologies like SMT
- Has some unusual code that is not representative of commercial workloads
- Does not test instruction fetch that well as it is a very small tight piece of code

<http://www.netlib.org/benchmark/dhry-c>

<http://en.wikipedia.org/wiki/Dhrystone>

Some Dhrystone results

<http://performance.netlib.org/performance/html/dhrystone.data.col0.html>

<http://www.roylongbottom.org.uk/dhrystone%20results.htm>

7



Stream

- Sustainable Memory Bandwidth in high performance computers
- Measures it in MB/sec
- Uses datasets larger than any available cache
- Designed for testing for large vector style applications
- Available in C or Fortran
- Requires some work to set up if doing multi-processor testing
 - Setup OpenMP, MPI or pthreads
- <http://www.cs.virginia.edu/stream/>

8



Unixbench 4.0.1

- CPU and file I/O tests
- High level testing
- C compiler needed

- Download from:
<http://www.tux.org/pub/tux/benchmarks/System/unixbench/unixbench-4.0.1/>

9



TPC-H

- Requires a purchase for the software
- Decision support benchmark
- Suite of business oriented ad hoc queries
- Concurrent data modifications
- Illustrates decision support systems that:
 - Examine large quantities of data
 - Execute complex queries
 - Answer critical business questions
- Reports by database size
 - 100GB, 300GB, 1000GB, 3000GB, 10000GB and 30000GB
 - QphH – composite query per hour performance metric
 - Price/QphH
 - Watts/KQphH

<http://www.tpc.org/tpch/default.asp>

10



Other TPC Benchmarks

- Both require a purchase for the software
- TPC-C
 - OLTP performance
 - Multiple transaction types
 - Complex database structure
 - Based on an order entry application
 - <http://www.tpc.org/tpcc/default.asp>
- TPC-E
 - Online transaction processing (OLTP) workloads
 - New benchmark
 - Simulates OLTP at a brokerage firm
 - Uses a database to model trades, customer inquiries, market research, etc
 - <http://www.tpc.org/tpce/default.asp>
- Obsolete
- TPC-D

11



httperf

- Web workload generator
- Measures various HTTP workloads
- Measures generic web server performance
- Tests HTTP/1.1 and SSL
- Current version as of Dec 2009 is v0.9.0

<http://code.google.com/p/httperf/>

<http://www.hpl.hp.com/research/linux/httperf/>

12



iperf

- Used to test client/server network
- Tests maximum UDP and TCP bandwidth
- Allows for tuning parameters and characteristics
- Reports bandwidth, delay jitter, datagram loss

<http://sourceforge.net/projects/iperf/>

netperf

- Network performance benchmark
- Tests unidirectional throughput and end to end latency
- TCP and UDP xsockets
- UNIX domain sockets
- IPv4 and IPv6 tests

- Latest version is 2.4.5 as of June 2009
 - <http://www.netperf.org/netperf/NetperfPage.html>

nstress

- <http://www.ibm.com/developerworks/wikis/display/WikiPtype/nstress>
- Made up of
 - ncpu Stresses cores
 - ndisk Stresses disks
 - ndisk64 Used to handle larger files
 - ndiskaio Same as ndisk but uses asynchronous I/O
 - ndiskmio Uses modular I/O
 - nmem Stresses memory
 - npic Tests shared memory and semaphores
 - nlog Generates log type output
 - nfile Stresses JFS logs using creates, deletes and writes
 - ipctest Tests IPC
- Note – this is an older set of programs compiled on AIX v5.1 and AIX v5.3 ML4.
- Make sure you know the differences between the hardware you are testing so you can correctly interpret the results.

15



Specjvm2008

- Free SPEC/OSG Java benchmark
- Measures performance of Java RTE
- Several real life applications and benchmarks
- Focuses on core java functionality
- Low dependence on I/O and no network I/O
- Stresses memory and CPU

<http://www.spec.org/jvm2008/>

Other SPEC benchmarks (require licenses)

SpecjAppServer2004 <http://www.spec.org/jAppServer2004/>

Specjbb2005 <http://www.spec.org/jbb2005/>

S

16



Other Spec Benchmarks

- These ones require licenses
- SpecjAppServer2004
 - Measures performance of J2EE 1.3 application servers
 - <http://www.spec.org/jAppServer2004/>
- Specjbb2005
 - Evaluates servers running typical Java business applications
 - Represents an order processing application for a wholesale supplier
 - <http://www.spec.org/jbb2005/>
- Spec CPU2006
 - CPU intensive benchmark stressing CPU, memory and compiler
 - <http://www.spec.org/cpu2006/>
- SpecjEnterprise2010
 - Extends SpecjAppServer2004
 - Measures full system performance for Java EE 5 or later application servers
 - <http://www.spec.org/jEnterprise2010/>
- SPECvirt_sc2010
 - Tests datacenter servers used in virtualized server consolidation
 - http://www.spec.org/virt_sc2010/

17



Oracle Swingbench

- Stress test Oracle databases
- Order Entry
 - Similar to TPC-C
 - Includes log-ins
 - I like the 200 user test with 20 million customers and 20 million orders
- Calling Circle
- Most people only run order entry
- OE transactions consist of:
 - Select 50%
 - Insert 30%
 - Update 20%
- Reports on
 - TXNs/min low and high
 - TXNs/sec low and high

18



Other tools

- Do not underestimate basic tools
 - dd
 - cpio
 - ftp
 - You can use the above to get some very basic baselines
- Monitoring and other tools
 - nmon
 - iostat
 - vmstat
 - netstat
 - Operating system specific tools like trace, svmon, fcstat
 - Zip and unzip tools
 - Compilers
- Know which ones you will use
- Make sure you know what tools you will monitor with

19



Notes on Sizing Studies

- Tend to only size for the application needs
- Based on data provided by customer
- Often do not include resources for:
 - Virtualization
 - CPU and memory for hypervisor
 - CPU and memory for SEA, virtual ethernet and virtual SCSI
 - Hardware specific memory needs (each IVE active port needs 102MB)
 - Memory required for database buffers (SGA/PGA in Oracle for example)
- They often round to whole cores or whole processor boards
 - Do not take micropartitioning into account
 - Affects consolidation sizing
- Make sure you validate any sizings – they can be off by several cores and lots of memory

20



Moving Forward

- Document which type of benchmark
- What is the purpose?
- What constitutes success or failure?
- Document assumptions and expectations
- Clear definition of scope – scope creep happens all the time
- Clear list of all tests to be run
- Clearly defined test plan for each set of tests
- Rational estimate of time and personnel resources
 - Who will install and configure the LPARs?
 - Who will provide support during the test?
 - DBAs?
- A performance benchmark can take 2 weeks to set up, 4 to 8 weeks to run and then you have to do analysis and teardown
- Understand the costs

21



Benchmark information

- What hardware and software prerequisites are there
 - Cores, disk, network, fibre cards
 - Application versions
 - Compilers? Gcc versus xIC
 - Java versions?
- Virtualized, partially virtualized or dedicated?
 - Do you want to test all of these, combinations or just one?
 - Shared or dedicated cores
 - VIO servers?
- Database
 - Which version?
 - Ensure the same setup and explain plans on ALL LPARs and systems
 - Defaults between Linux and various UNIX will be different so you need to check
- Clustering?

22



Benchmark information

- Software licenses
 - Ensure you have spare licenses for databases, etc for using during the test
- How do you plan to drive the load for each different benchmark
- Will you test over VPN or locally?
- Have a baseline test that you then base tuning on
- Expect to modify tunables and to have to retest after changing them – this adds time so plan for it
- Other
 - One tier or two tier testing?
 - Monitoring tools – make sure they are set up the same everywhere

23



Presenting Results

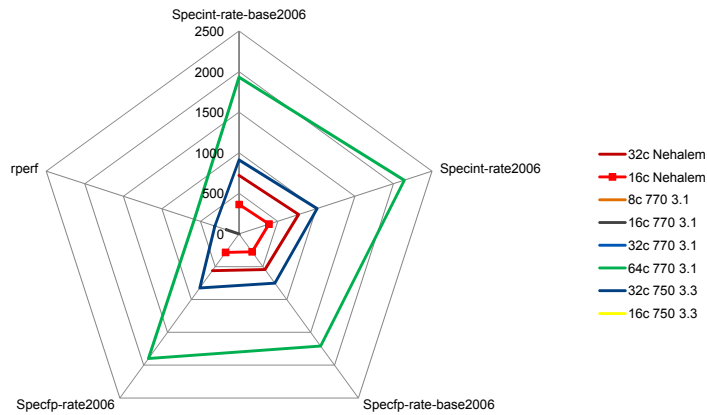
- Report
- Include executive summary at front
- Include tables of data at the end
- Pick graphs that make sense
 - Radar / spider graphs appeal and show a lot of data at once but not always appropriate
 - Use consistent colors for the same things between graphs
- Include assumptions
- Include what you expected, what you saw and why they may have been different
- Include list of hardware, software and any tuning you did

24



Graph Options – Radar/Spider

General Published benchmarks

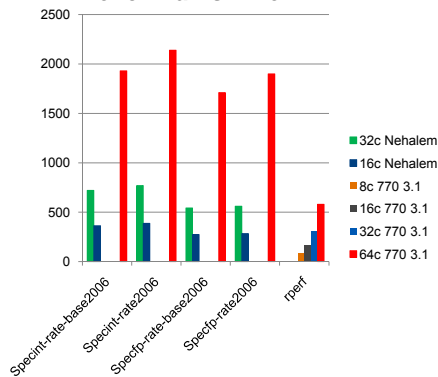


25

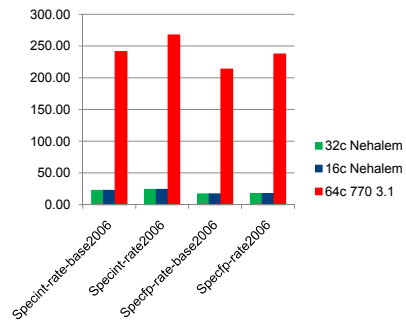


Graphs – Bar - Normalizing per core

General Published Benchmarks - View 2



General Published Benchmarks - normalized per core



26



Summary

- Plenty to choose from
- Most problems arise out of lack of planning
- People regularly underestimate what a benchmark takes in terms of time and resources
- Have a plan
- Have an expectation for the results that you can justify
- Watch for scope creep

- KNOW WHAT CONSTITUTES SUCCESS OR FAILURE