BluePrint for Benchmarking
Success

Barbara Perz/Hans Joraandstad,
Global Customer Benchmarking and
Characterization

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-1363-10
12/11/02, Revision A
Edition: January 2003
Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95045 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form or by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Sun Ray, and Solaris are trademarks, registered trademarks or service marks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the US and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun’s licensees who implement OPEN LOOK GUIs and otherwise comply with Sun’s written license agreements.

U.S. Government Rights —Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.


Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l’utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l’autorisation préalable et écrite de Sun et de ses bailleurs de licence, s’il y en a. Le logiciel détienu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Les parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l’Université de Californie. UNIX est une marque enregistrée aux Etats-Unis et dans d’autres pays et licenciée exclusivement par X/Open Company Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Sun Ray, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d’autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d’autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L’interface d’utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d’utilisation visuelle ou graphique pour l’industrie de l’informatique. Sun détient une licence non exclusive de Xerox sur l’interface d’utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l’interface d’utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L’ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N’EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L’APTITUDE DE LA PUBLIATION A REPODRE A UNE UTILISATION PARTICULIÈRE, OU LE FAIT QU’ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S’APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.
BluePrint for Benchmarking Success

This article provides guidelines for anyone who is assembling a benchmark and expecting to get information back that will help make a decision on which computer to buy. Benchmarking is expensive in both currency and people time. If the benchmark is not well-defined, it is a waste of time for both the customer and the computer vendor.

When defining a benchmark, the person assembling it must consider many things to complete the benchmark successfully. This article groups these items in four categories—goals, practicality, commitment, and rules.

This article contains little specific technical information. The information presented is not Sun-specific. It discusses how the customer can be successful at benchmarking any vendor’s hardware and software.

The audience for this article is project managers who are charged with benchmarking their compute environment on one or more computer vendor’s platforms and Sun™ system engineers who are working with a customer or prospect to define a benchmark.

This article covers the following topics:

- Goals
- Commitment
- Practicality
- Define the Rules
  - Software
  - Hardware
- Types of Runs
- Reporting
- Questions
- Summary
- Checklist
The Situation—You have heard the sales pitches. You’ve read the marketing brochures. You’ve explored the vendor’s website. You’ve checked the industry standard benchmarks. You’ve talked with the references. Now you have decided that a benchmark is the next step to help determine the best vendor for your computing needs. How are you going to do it?

The Solution—This is a complex question that must be answered specifically for your compute environment. While you are planning your benchmark, you must determine the goals, practicality, commitment, and the rules. This article provides you with many items to consider to make the benchmark a success. Most of these items are non-technical, but some technical issues are mentioned. The differences between HPTC (High Performance and Technical Computing) and Commercial (database and related applications) benchmarking are stated where needed.

Goals

The results of the benchmark are not the only criteria for selecting a computer; they are just one of several criteria. What new information do you expect to learn from this exercise? What is the weight that this benchmark holds compared to other criteria in the decision making?

Why are you looking for new compute resources? Are your users complaining about response time? Are engineers idle waiting for analysis to complete? Are you expecting new projects to start up? What do you expect the workload to look like one month after installation? Six months? One year? Can the benchmark be used to simulate the workload you expect to see 18–24 months from now?

The benchmark must reflect the key usage of the computing resource to be filled with this purchase. Typically, you will not benchmark word processing or email applications on a workstation or Sun Ray™ appliance server. That information is available in various published white papers. But you may want to measure the response time of your particular application as it accesses your database on a remote server. Or you may want to determine if the typical engineering design for your organization can be analyzed overnight. If you multiply this by the number of users, will it represent your production environment?

Pretend that you are at the end of the benchmark. Are the results measurable? Can they be readily compared between vendors? A set of basic metrics can include and certainly are not limited to:

HPTC

- Raw CPU speed (MegaFlops)
Commitment

Running a benchmark to simulate some subset of your compute environment requires an investment on your part and on the part of the computer vendor who wants your business. There is the time commitment of your staff, computing resources, and, possibly, travel and per diem. The computer vendor will have similar commitments.

There may be some legal requirements. Talk to your legal department to establish what, if any, rules may be imposed on your benchmark and selection process by your government, state laws, and so forth; for example, European Union rules or the license to use a program that does not give you the right to redistribute it for a benchmark. The presence of such rules may impact your planning and execution of the benchmark process substantially.

You need to determine the part of your compute environment that you want to benchmark. This is not something that can just be thrown together. Planning is necessary.
Are any independent software vendor (ISV) codes required to run the benchmark? Due to business partnerships, the computer vendors most likely have access to the major database software, like Oracle® and DB2, and key HPTC applications, such as MSC.Nastran, FLUENT, and STAR-CD. The computer vendor may request your help in obtaining other licensed software necessary for your benchmark.

Do all of your ISV codes run on the computers built by the computer vendors? For Commercial benchmarks load generators written by ISVs are available for a fee. In general, benchmark centers do not have licenses for these. There are times when the database ISV will actively participate in the planning and execution of benchmarks.

Can the program or set of programs be extracted and run independently from the rest of the environment? The best way to test that is to stage and test it on its own in your own data center or lab. It does not really matter which computer or OS you use. Set aside a computer or two and disk space on an isolated network. Load a new OS. Load your benchmark. Does it run successfully? Does it measure what you intend? Write down what you did, every step. You will be doing this again at each computer vendor visit or you may have to write down the instructions carefully for someone else to follow. Measure the performance. Verify the results and write down how the results should be verified.

You have just run your own benchmark. How long did it take you to set up? How long is the run? Can you do that again in a reasonable amount of time?

This staging and testing step produces several key items:

- Validates that you are benchmarking the right thing
- Allows you to verify that you have all the components necessary to complete the benchmark successfully.
- Provides you with a reference to compare the computer vendors
- Provides a verification reference for the computer vendor
- Provides a performance reference for the computer vendor
- Defines your requirements for the computer vendor

Plan to hold several conference calls with the computer vendor's sales team and the benchmark team prior to the benchmark itself. These calls give everyone an opportunity to learn about the benchmark; what is required, what computer configurations are available, and what the goals are.

For commercial benchmarks, less often for HPTC benchmarks, you should plan to travel to the benchmark centers of each computer vendor. You know your benchmark and your applications best and can speedily install and run them. The computer vendor can set up the computer environment before you arrive based on the conference call planning sessions. The computer vendor will host you with experts who can tune the computer and perhaps even your application to perform optimally. Assemble a team of individuals whose skills are complementary, are familiar with the benchmark and are available to travel. Consider which days during
the benchmark you need the skills of each individual on the team. There is no point to having your Oracle application expert there while the database administrator is setting up the compute environment. You will need to pay your own travel and per diem expenses. Have individuals on call who are not traveling to consult with those who are to resolve any issues that arise.

If you are not traveling to the benchmark center to perform the benchmark yourself with assistance from the experts, the experts will run the benchmark based on your directions. They will have questions as the benchmark progresses. How do you want to answer them? You may want to have several people answer questions about parts of the benchmark that are in their area of expertise or you could have one person funnel the questions to the others. Supply names, phone numbers, email addresses and area of expertise for each person with the benchmark. You might also consider creating a website with benchmark information, downloads, FAQs, and so forth, as a way to communicate with the benchmark experts.

Practicality

While you are deciding the composition of the benchmark, use common sense. How practical it is to assemble and run? Are you willing to do it yourself?

It is not practical to expect that the computer vendor has 20 PCs to act as clients. It is not practical to expect the computer vendor to download three gigabytes of files from a website. It is practical to work with your computer vendors to use the resources that are available effectively.

Do not forget about confidentiality. Do you want to use real, sensitive, and proprietary data as part of the benchmark? If you will be using sensitive, proprietary data, have the computer vendor sign a non-disclosure agreement, unless there is one already in place. As part of the benchmark rules include a statement that all programs, applications, databases, data, and media should be removed, destroyed, or returned at the conclusion of the benchmark.

If you plan to travel to the benchmark center and run the test yourself, you will probably want to limit the benchmark test. Do you really want to ship off a benchmark team for a month to work on this when they need to do their usual jobs, too? Determine how long the tests will take. This is where that pre-staging in your own computer lab will tell you a lot.

How long does the benchmark take? Do runs that last just a few minutes or days make sense? When benchmarks complete too quickly (seconds) or take too long (a day), the opportunities for the computer vendor to tune the computer configuration and the application are limited due to the turnaround time. If you really have to
have long runs, include a shorter run that exhibits the same characteristics as the
longer run, so that the vendors can tune and showcase their computers quickly. Then
the long run can be made with the optimizations of the shorter one.

If you want to benchmark using a three terabyte database, where is the data coming
from? Tapes are not practical. Can you write a program to generate the data? When
making tapes of data, compress the files first. Writing and reading the tapes will be
much faster.

---

**Define the Rules**

Everyone will know what the benchmark is when you define in writing the
benchmark and its goals and rules. If you do not state them clearly and publicly,
you will be asked lots of questions or maybe none, with each computer vendor
creating their own rules. In the absence of information people will invent their own
to fill the gaps.

Clearly state what the main parts of the benchmark are, why these are included and
their respective objectives. What are the goals for each part of the benchmark? Is it
functionality? Performance? Portability?

How is each goal measured? Here are some ideas. For functionality, is the failover
within your allowable time? For performance, is 2x faster than your current compute
environment acceptable? State the current performance and what you expect from
your new system. For portability, what is your tolerance level?

What if your requirements are not met? Do not be overly restrictive in your
specifications, such as avoiding exact specifications of number of CPUs to be used in
fat SMP nodes for throughput tests. Keep your goal in mind, *not* a specific system
from a specific vendor you happen to know.

Everyone involved including your team and the computer vendor’s team must keep
in mind the definition and goals throughout the benchmark. It is easy to get
sidetracked by “It would be nice if we could...” Finish the benchmark and, if there is
time remaining, explore further.

If there are any confidentiality issues, state them in the rules. Computer vendors
may have a collection of benchmark reports available for internal use. If you do not
want a report of your benchmark as part of that collection, state that.
Software

What software is required to run this benchmark successfully?

Which operating system should be used? UNIX®? What tools do you want to use to monitor the compute environment? If you want to use more than the common UNIX tools like sar and vmstat, plan on providing them.

For an ISV code, if it is not a commonly available product like DB2 or ANSYS, identify how the computer vendor can get it or get a copy and license for them. Also indicate the version of the ISV product that should be used. Sometimes there are compatibility issues, particularly between major releases, so knowing the version saves time.

For database benchmarks, table layouts, queries, and middle ware must be identified and documented. To ensure that the tables are loaded correctly, can you provide a query that verifies this along with the correct results of that query?

Does your benchmark include programs that your staff has written? In addition to source code for all the programs, provide some documentation about the program, such as flowcharts and overall descriptions of the program and algorithms used. For every input data set include the corresponding output file(s). The computer vendor will use them to verify that the answers are correct on his computer. If the run time is on the order of hours, is there a smaller data set that shows the same characteristics as the longer one? The computer vendor’s optimization activity will be more successful if the runtimes are not excessive.

Once the computer vendor has your program, what is the vendor allowed to do with it? Can the vendor only make changes necessary to run it successfully (porting)? Can the benchmark expert rewrite parts of the program to improve performance? Can optimized libraries be used? Can directives (like OpenMP) be used? What about assembly code? Is that acceptable as an optimization technique? You probably do not want optimizations that are beyond the expertise of your users.

Hardware

Now that you have described the software environment, what about the hardware? Are you looking for large SMPS or a cluster of thin nodes? What are your availability and operational requirements? For the cluster, what kind of interconnect performance is needed? What is your expected data throughput for this application?
You probably have decided these points already, but, if you are not set on them, the benchmark can influence your decision. You may even solicit advice or 'best' solution from the vendors.

Are there any extreme memory requirements, say, for the performance of the SGA (shared global area) or for allocating scratch space as tmpfs (a memory-based file system)?

For many Commercial benchmarks, the disk configuration is more important for performance than the memory. How much disk space is required for the database? Will you need an online backup of the database to ease the restoration of the database in between runs? What are the requirements for a logging file system? For HPTC, how much space does your out-of-core solver need? What is the configuration of the disks: RAID 5, SAN, just a bunch of disks (JBOD)? Do you have a corporate standard for file systems like UFS or VxFS? If so, is that required in the benchmark? Then state that.

---

**Types of Runs**

The next step in defining the rules is to describe how the benchmark is to be run. The standard often used is one program or query at a time and then try adding processors to check the scalability. That is fine, but does that really simulate your compute environment? Does that make sense if each system only uses one or two processors and you plan on purchasing eight, or 24 or 72 or 102 processors where many programs are running in production? A throughput test may give better answers to your basic questions.

Throughput benchmarks vary depending on your compute environment. Get creative, but make it realistic. Is running the same exact query 20 times simultaneously really how your compute environment used? You need to set a goal, determine the components, and state how the components are started.

First, define a goal for the throughput test. Is the winner the one who completes the most number of runs within a specified time limit? Or is it the one who completes the suite of runs fastest?

Next, determine what the components of the throughput run are. How many processors should each component use? How many times should each component be run? If one of the components has a run time that is a lot longer than all of the others, the time of the throughout run will be the same as that long run. That does not demonstrate much about the computer under a load.
Now, how do you want the components started? Should they be started all at once or staggered? Do you want use a resource management system like Sun ONE™ Grid Engine software or Veridian's Portable Batch System or just let the operating system manage the scheduling? With a resource management system, the computer vendor will set it to run so that the processors are not oversubscribed. With the operating system doing the scheduling, the processors will be over-utilized, if more processors are needed than are available. How is this managed in your production environment?

Do you want the elapsed time for each component recorded? Are you interested in the system-wide CPU utilization, disk usage, and so forth for the duration of the throughput test? State what information you want so you get the same information from all the computer vendors.

What about vendors extrapolating to ‘future’ systems?

---

**Reporting**

Whether you are running the benchmark yourself or the computer vendor is, it is a good idea to set up a benchmark report template as part of the planning step. With a format to follow, consistent information is reported, which eases comparison. Make an annotated spreadsheet to fill in. What should be included in that template?

There should be an itemization of hardware and software configurations including versions actually used. Yes, you specified this earlier (see “Software”), but what if the computer vendor used a different version of Oracle or LS-DYNA? There are often feature and performance differences between versions and sometimes they just do not compare.

What metrics did you decide to use for each component of the benchmark? Is it CPU seconds, elapsed time reported in hours, minutes, and seconds, or is it number of runs completed per hour to name just a few? Identify each component in the template and spell out the metric clearly. Avoid metrics that are specific to one or a few vendors; stick to commonly available metrics.

Double-check that the template matches the rest of the document. There is nothing more confusing than one part of the rules stating one thing and another part contradicting that. Which is right? It leads to questions from every computer vendor.

The benchmark expert may have made changes to your scripts, programs, or setups as part of porting or optimization. Request that any modifications be included as part of the report. This could be in the form of the UNIX `diff` command or just a commentary. You decide what works best for you.
Are there any files that you would like back or to take with you perhaps to recheck the results later? Identify them and the media that you will accept. CDs and 4mm DDS4 tapes are the current technology. Maybe even ftp to an anonymous FTP server will suffice. Give that information. Avoid requesting massive amounts of data to be returned unless necessary for your verification of correct results.

Questions

Earlier in “Commitment” the matter of answering questions was discussed. What did you decide? Explain how you want to answer questions as part of the rules, and which external rules you have to follow, if any (for example, European Union rules).

Summary

Thoroughly planning a benchmark is key to its success. The items covered herein have been gathered from years of benchmarking experience on a variety of computers and operating systems. The guidelines are the same across them all. Good luck with your benchmark, should you decide to accept the challenge.

Acknowledgements

We would like to thank Melanie March, Anna Roots, Ron Larson, Tom Kemper, and Masaru Ikeda for taking the time to read this article and giving us more ideas.
About the Authors

Barbara Perz

Barbara holds a degree from the University of Illinois, Urbana-Champaign, in Mathematics and Computer Science. After graduation in 1975, she began working in the computer industry as a systems engineer teamed with sales for Control Data Corporation, Cybernet. In this position she was involved in customer visits, benchmarks, and special programming projects, including the writing of programs to generate a Biblical concordance and a tool to aid in transliterating Greek/English and Hebrew/English. Barbara joined the headquarters benchmarking team at Floating Point Systems in 1984 where she produced systems engineers’ training materials and newsletters while waiting for turnaround. She also developed and delivered customer training on UNIX and Fortran and C program optimization and parallelization.

As Manager of Technical Sales Support, she continued supporting systems engineers, and her benchmark team consistently completed benchmarks on time. When Floating Point Systems was acquired by Cray Research, she continued in this role and added on “care and feeding of the sales analysts”, including training the sales force, newletters and as a technical contact at headquarters.

Barbara was part of the Cray Research, Business Systems Division, that was acquired by Sun Microsystems in 1996. Again, she produced training sessions for a new sale force introducing the newly acquired product, Starfire. But Barbara decided that running benchmarks is more challenging and fun because the projects are always new and either the hardware or the software is always changing. Barbara has been dedicated to benchmarking at Sun Microsystems for the past 5 years.

Hans Joraandstad

After receiving a university degree in Mathematics, Physics and Computer Science at NTH in Norway in 1971, Hans implemented a compiler for Simula67, the first object oriented programming language, using Pascal as the implementation language. Hans then joined CERN in Switzerland where he headed the group responsible for the CDC7600 OS support and tuning.
In 1980 Hans joined Control Data Professional Services in Norway where he did presale and benchmarking of Cyber205 and ETA10 systems, plus consulting related to the oil production platforms for the North Sea. In 1995 Hans joined AT&T/NCR to participate in some of the first successful data warehouse implementations for banks in Norway.

In 1997 Hans joined Sun and has been running customer HPC benchmarks. Hans has also been involved in optimization of ISV-codes (AVL/Fire and Molpro) and conducting performance optimization classes at customer sites. Hans is an active member of the HPC Aces community at Sun.

CHECKLIST

You can use the following checklist to verify that you have defined the rules for the benchmark you are about to embark on.

- Purpose
- Goals
- Confidentiality
- Software
  - Documentation
  - Operating system
  - ISV Codes including the version
- Source
  - Own code
  - Include small and large data sets
  - Include reference output or verification for each data set
- Tablespace
- Queries
  - As is—no changes, except for porting
  - Optimizations—what is allowed
- Performance
  - Current
  - Target
■ Hardware
  ■ Disk requirements—quantity, RAID level
  ■ Memory
  ■ CPUs
  ■ Large SMPs or a thin node cluster
■ Types of runs
  ■ Standalone
  ■ Scaling
  ■ Throughput mix
  ■ Rules for the runs
■ Reporting
  ■ Report matrix
    ■ Configuration(s)
      ■ Hardware
      ■ Software—include versions
    ■ Performance metrics
      ■ Modifications to scripts, code, queries for porting and optimization
      ■ Extrapolations
  ■ Files, including output files, and other information to be reported
  ■ How to answer incoming questions