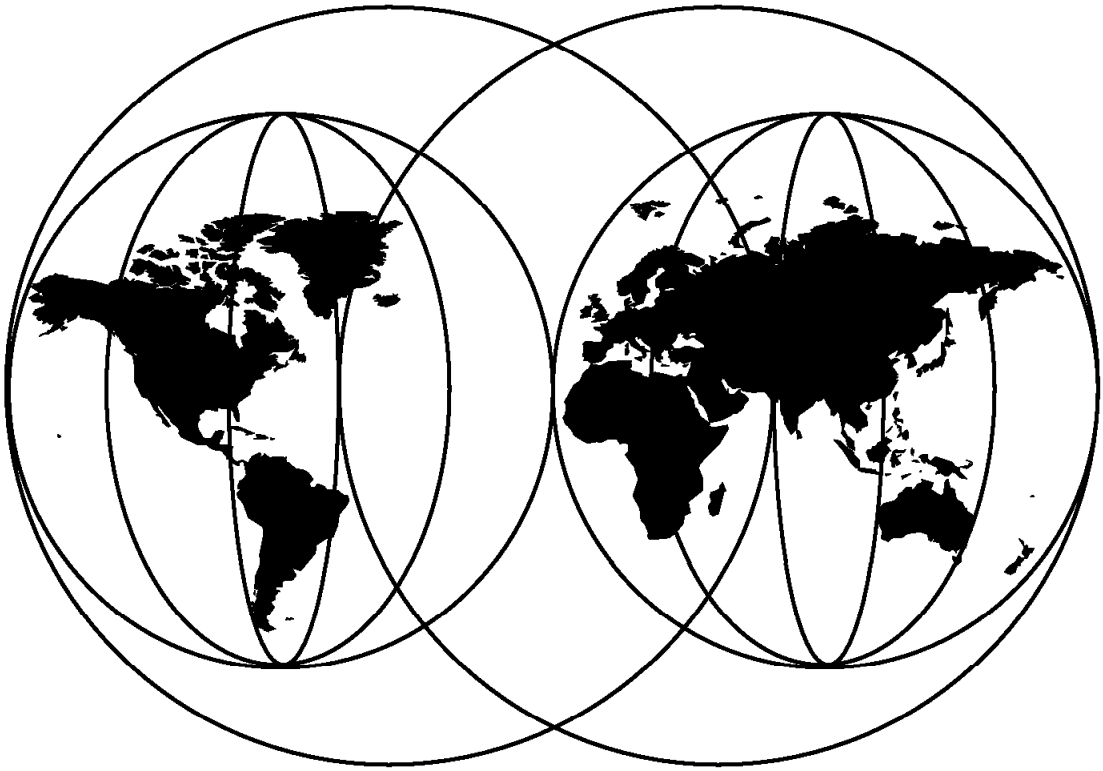
Enterprise Servers: Benchmarketing Perspectives

Moon J. Kim, Stephen Turner, Daesung Chung, Patrick Kappeler, Hans-Dieter Mertiens



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5179-00

International Technical Support Organization

SG24-5179-00

Enterprise Servers: Benchmarketing Perspectives

July 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special Notices” on page 115.

First Edition (July 1999)

This edition applies to Version 2, Release Number 7 of OS/390, Program Number 5647-A01 for use with the IBM 9672 Enterprise Server Generation 5 and later.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This redbook is concerned with the use of benchmarks, both Industry Standard and proprietary, by the major vendors of large servers for marketing purposes.

The injudicious use of such benchmarks may lead to the incorrect sizing of a server, and does not take into account the availability, security, operational and functional differences between different vendors′ products.

Therefore, this redbook assists both customer personnel and IBM personnel involved in server selection and sizing by discussing the strengths and weaknesses of Industry Standard benchmarks, how servers of different architectures may be compared, and where IBM S/390 and SP servers offer benefits not measured by such benchmarks. This redbook applies to Version 2, Release 7 of OS/390 for use with the IBM 9672 Enterprise Server Generation 5 and later.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Moon J. Kim is a Senior Technical Staff Member at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of S/390 Systems and Architecture and of High-Speed Access Network System and Technology. Before joining the ITSO in 1996, Moon worked in the S/390 Architecture department of the Poughkeepsie Development Lab as a system designer and architect.

Stephen Turner is a Consulting Systems Strategist at the ITSO, Poughkeepsie, NY. He has 31 years of experience in the S/390 technical support and marketing field. Before joining the ITSO in 1997, Stephen was a Consulting Systems Engineer in IBM Australia where he worked on technical marketing support for S/390. He has written extensively on comparative analysis of S/390 and benchmarketing.

Daesung Chung is an IT specialist in charge of RS/6000 technical support in IBM Korea. He has nine years of experience in AIX, beginning with RT/6100. His areas of expertise include sizing of large UNIX servers and parallel databases on SP, and he has been involved in numerous RS/6000 and SP benchmark cases. Before joining IBM, he worked as a member of a VLSI design team at a semiconductor manufacturer.

Patrick Kappeler was a computer specialist in the French Air Force before joining IBM France in 1970. He began as a S/370 diagnostic program designer, and has since had several specialist and management positions, including international assignments, all dealing with S/370-390 technical support. He provided hardware and software support for S/390 Parallel Sysplex until late 1996, when he joined the EMEA S/390 New Technology Center in Montpellier, where he now gives consulting and pre-sale technical support to S/390 Growth Initiatives, including e-Business and Server Consolidation

Hans-Dieter Mertiens is a senior technical marketing specialist in IBM Germany. He worked at several IBM installations as a system programmer before joining IBM in 1984. Since 1991 he has worked for the S/390 division, focusing mainly on new function (such as APPC/MVS) within MVS and OS/390. His area of expertise include UNIX Services on OS/390, application porting, and integrating UNIX Services into the traditional OS/390 world. He has participated in several projects at the ITSO in Poughkeepsie and Raleigh, and has coached several porting projects during the last two years.

Thanks to the following people for their invaluable contributions to this project:

Joe Temple
IBM Poughkeepsie

Mary Moore
IBM Poughkeepsie

Gururaj Rao
IBM Poughkeepsie

Todd Boyd
IBM Poughkeepsie

Tom Dewkett
IBM Poughkeepsie

Maryanne Ferraro
IBM Poughkeepsie

Ray Hawryluk
IBM Poughkeepsie

Carl Parris
IBM Poughkeepsie

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 123 to the fax number shown on the form.
- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Overview

Over the brief history of computing, enterprises have used many approaches to sizing the computer systems that they have required in order to run the enterprise.

These approaches have evolved as computing hardware has become less expensive and people costs have increased. Two contrasting approaches are:

- Assume that hardware is “cheap” and purchase a server with much larger capacity and performance than is required - this is typical of many UNIX installations. When the selected server does not have the availability, scalability, security, operational, and functional attributes that are demanded by the application, then this approach can result in much higher people costs in order to provide the necessary attributes.
- Purchase the minimum system necessary to run the application and upgrade the server to a larger model when additional capacity is judged to be required - this is typical of many large IBM S/390 installations. This approach can minimize hardware costs when the vendor offers a simple, short duration upgrade to a larger model. S/390 customers have also utilized the availability, scalability, security, operational, and functional capabilities of S/390 to reduce the people costs of their installations.

Whichever approach is taken, some estimate of the capacity/performance of the various servers on offer from different vendors is required versus the resources required by the application.

This can be relatively simple if the additional capacity is for an existing, production application, but can be exceedingly difficult when trying to acquire a new server for a completely new application.

In the early days of commercial computing, some customers would require the prospective supplier to run a benchmark test using the customer′s data or application. Frequently these benchmarks turned out not to accurately reflect the customer′s actual application requirements and under- or over-sizing occurred. In addition, the costs of such benchmarks to both the customer and vendor quickly became excessive.

Thus today, both customers and vendors are turning to standard benchmarks, run by the vendors, in order to compare different vendor′s offerings for many different workloads and applications.

In many cases, the people responsible for the acquisition do not have a clear understanding of what the benchmark data really means. Benchmarks are widely used incorrectly. Raw performance often does not represent a computer system′s capability to efficiently process actual production workloads.

Most computer systems are designed for a certain purpose or to perform a certain function, in much the same way as a racing car is designed for racing and not for transporting children to school.

Because so many decisions are made based on standard benchmarks, vendors have become very good at tuning them to achieve the best possible results. This tuning may even include changing hardware or software products to recognize and perform better in the benchmark environment.

Frequently these changes may offer no benefit to a customer application, as their sole objective is better benchmark performance.

1.1 What to Measure

There are no absolute rules that say that one system is better than another. The key to assessing the relevance of the benchmark is learning how to evaluate the system in the context of the *real* application. It is important to know if a computing system makes as effective use of processor cycles, memory capacity, and input/output bandwidth in production as it does with the test data. The objective is to look at cost, performance, reliability and serviceability, and not merely at benchmark data, when evaluating systems.

It has long been known that in any performance measurement there will be some bottleneck, for example, processor engine speed, memory size, I/O configuration, application design and so on.

In today′s complex systems with multiple servers, multiple engines per server, and large I/O configurations, determining the exact cause of a bottleneck can be a challenge. When a server′s processor capacity or speed is a bottleneck and is limiting system throughput, upgrading to a faster processor will relieve the problem and that bottleneck problem may disappear. However, as more work is introduced to the system, other problems may appear.

In this book we will consider some of these factors and their affect on benchmarking.

The following performance-related definitions are used in this redbook:

Functional Unit An entity of hardware and software, capable of accomplishing a task.

Processor complex A functional unit that consists of one or more processors, internal memory and I/O interfaces.

Throughput A measurement of the amount of work performed by a system over a given period of time.

Performance System throughput and/or response time.

There are a large number of attributes on which to base a comparison between systems. These features, when ideally configured, assist a system in running faster and more efficiently.

- Processor cycle time
- Number of channels and implementation
- Memory speed
- Degree of parallelism
- Size and nature of the instruction set
- Memory hierarchy
- I/O hierarchy

These attributes contribute to a system′s power and capacity. In the final analysis however, it is how these features support the work that must be performed that determines the power of a processor for its users. A choice has to be made from the measurement criteria for making a benchmark evaluation. The following terms are used to express these measurements:

- *Throughput* is used to express the measured performance expressed in units of work, which may be either jobs or transactions per seconds.
- *External throughput* is the number of jobs or transaction counts completed per elapsed time. Usually the data is used to evaluate the system and cannot be used to measure the processors.
- *Internal Throughput* is defined as the number of completed jobs or transactions per processor busy second. The internal throughput rate is computed using the time that processor is busy. Internal Throughput is External Throughput at 100% utilization. Usually this measurement is used to determine the capacity of the processor.
- *Instruction execution rate* is the ratio of the number of instructions executed to the processor busy time that it took to execute them. The term MIPS (millions of instructions per second) is commonly used. When using MIPS to compare processors, care must be taken to ensure that exactly the same instruction set is being executed to accomplish the work.
- *Response time* is a performance measurement criterion. Response time is a measurement defined by the end user and it is an indicator of service to the end user. The most common definition of response time is the elapsed time from issuing the command to the time that the first response appears on the user terminal. Therefore, the measurement of response time is a system-wide measurement.

1.2 What is A Benchmark

A benchmark is typically thought of as a set of standard tests that are run on a system to compare its performance with that of other system. Benchmarks are used to compare the performance of the same types of computer systems, but they usually reflect how well various subsystem components work within a computer system.

There are three basic types of benchmarks:

1. Measurement of all system performance
   - The most widely used benchmark tests today include the Transaction Processing Council′s (TPC) TPC-C and TPC-D tests, Software AG′s SAP R/3 SD benchmark, and the NotesBench consortium′s benchmarks for Lotus Domino. Each of these attempts to give a measurement of overall system performance. CPU computing power only plays a part of the measurement role. Throughput is the key parameter.
2. Measurement of specific performance
   - Some benchmarks attempt to measure the performance of specific areas such as CPU and disk subsystems. The System Performance Evaluation Council′s (SPEC) Int95, and FP95 are among the commonly used benchmarks in this category. For example, the SPEC Int and FP benchmarks are intended to be a measurement of CPU integer or floating-point computation performance.
3. Measurement for marketing usage
   - Benchmarks can also be devised for marketing purposes. Intel′s ICOMP benchmark is an example of a benchmark program developed by a manufacturer for measuring microprocessor performance. IBM′s Large System Performance Reference (LSPR) benchmarks are used to guide

customers in the sizing of IBM and so-called “plug-compatible vendor ” S/390 systems.

Good benchmarks use a well-defined testing methodology based on real-world use of a system. In addition, the measured data should be deterministic and repeatable. When used appropriately, benchmarks can also provide a means of using the system to tune system parameters, reliability, and system capacity.

Many people tend to assume that benchmarks equate raw performance with productivity: faster-cycle system is always better. They forget to ascertain what the benchmark is measuring. In reality, there is a big difference between performance and productivity, and an equally large variation between different types of benchmarks.

A superior approach for most customers would be to use the following categories to evaluate the hardware and operating system.

- Performance and scalability

  Can the server deliver the required performance and scalability, and how much experience does the vendor have with the size and complexity of the user′s workload in the real world?

- Software vendor enthusiasm

  Can the necessary software be obtained? Will the vendor fix bugs, provide performance tuning and provide new product releases in a timely manner?

- High availability

  Can the system meet uptime and stability requirements?

- Platform architecture longevity

  Will the platform and vendor be viable in five years? What degree of transition and upheaval can be expected during that time frame?

  – Does the vendor provide a comprehensive set of tools to manage the server and PCs connected to the server?

- Maintenance

  Will the vendor provide adequate and reliable hardware and software support?

1.3 Focus of this Redbook

In the remainder of this book, we describe the most commonly quoted Industry Standard benchmarks, look at some of their strengths and weaknesses, and discuss how to compare the performance of servers based on different architectures (for example, S/390, UNIX or NT).

The main body of the book concludes with a discussion of some of the functions and capabilities of two IBM Servers (S/390 and RS/6000 SP) that can have value to a customer and that are *not* measured by any benchmark.

Chapter 2. Industry Standard Benchmarks

The computer industry has always been, and always will be, in need of performance reference data. This data is required either to assess the technical value of a computer design or more practically to use in positioning a vendor′s offerings against the market place.

Designing and running benchmarks are not simple tasks:

- Benchmarks must be representative, or recognized as being representative, of the expected usage of the system.
- Benchmarks must yield reproducible and consistent figures to be of any value to the industry.
- Benchmarks must provide an unbiased basis for comparison between different manufacturers′ products (that is, they must be *manufacturer neutral*).

The vast majority of customers do not have the time, money, or inclination to set up and run a benchmark that reflects (however imprecisely) their unique environment.

Therefore, most organizations, when selecting a server for a new application or for growth in their existing environment, turn to outside sources of benchmark data and try to relate these to their particular needs.

Many manufacturers have set up and published benchmark systems in order to validate the performance claims for their products. However, while many of these may accurately reflect actual user workloads (for example IBM′s S/390 Large Systems Performance Reference, or RAMP/C benchmarks for the IBM AS/400 and RS/6000), the fact that they are specific to a particular vendor makes them difficult to compare to other vendor′s products.

Over the last 10 years, a series of Industry Standard benchmarks have evolved that attempt to provide vendor-neutral data for various workload types. Ideally such benchmarks should be designed by independent third parties or, at the very least, by associations of manufacturers. They should also be run by these same third parties.

Running such benchmarks is a significant undertaking due to the heavy costs and resources needed to set up and run the proper test configuration. In addition, vendors require benchmark results to be ready close to the announcement date of new products.

Therefore, most of the Industry Standard benchmarks are run by the manufacturers themselves and the results are audited and validated by third parties whenever appropriate.

This chapter discusses some of the Industry Standard benchmarks that are commonly encountered in today′s marketplace. These are shown in Table 1 on page 6.

Table 1. Popular Industry Benchmarks

Benchmark	Sponsor	Type	Range of Applicability of Results
TPC C	TPC	OLTP	Commercial online transaction processing
TPC D	TPC	Data Warehouse	Business intelligence
TPC W	TPC	OLTP web serving	e.Commerce
SPECInt95	SPEC	Integer processing	Relative processor speed
SPECweb96	SPEC	Web serving	Web serving
Linpack	Jack Dongarra, University of Tennessee	Linear algebra	Technical computing
Baan	Baan	ERP	Baan ERP
SAP SD	SAP AG	ERP	SAP R/3
Notesbench	NotesBench Consortium	Groupware	Groupware

2.1 Compute-Intensive Benchmarks

The computing requirements of technical and commercial users can be very different. Although technical computing can require very high speed transfer of large amounts of data, the user's main interest is usually in assessing how fast some particular problem can be solved.

Technical benchmarks therefore concentrate on measuring computing speed for integer and floating point arithmetic with very little I/O.

Since this book is primarily oriented to the needs of commercial users, only a few popular benchmarks are discussed here.

2.1.1 Linpack

This well-known standard benchmark is a Fortran program for the solution of a dense set of linear equations by Gaussian elimination. It is distributed by Jack Dongarra of the University of Tennessee[1].

Linpack is a scientific, floating point workload composed of a collection of linear algebra subroutines. Operations are done on a 100 x 100 matrix. Programs will easily fit into the caches of many machines, so it measures only CPU speed and does not measure memory access times, I/O, multiprocessing and so on.

[1] Dongarra, J.J., *Performance of various computers using standard linear equation software*, Computer Science Department, University of Tennessee, Knoxville, CS-89-85, August 29, 1995

The main value of this benchmark is that results are known for more computers than any other benchmark. Most of the compute time is contained in vectorizable DO-loops such as the DAXPY (scalar times vector plus vector) and inner product. Therefore one expects vector computers to perform well on this benchmark. The weakness of the benchmark is that it tests only a small number of vector operations, although it does solve a complete (but small) real problem.

Results are reported as millions of floating point operations per second (MFLOPS), and are regularly published and available by electronic mail.

- LINPACK DP (Double Precision) - n=100 is the array size. The results are measured in megaflops (MFLOPS).
- LINPACK SP (Single Precision) - n=100 is the array size. The results are measured in MFLOPS.
- LINPACK TPP (Toward Peak Performance) - n=1,000 is the array size.

Linpack is not a useful benchmark for assessing commercial performance and capacity.

2.1.2 The Livermore Loops

These are a set of 24 Fortran DO-loops (The Livermore Fortran Kernels, LFK) extracted from operational codes used at the Lawrence Livermore National Laboratory.

They have been used since the early seventies to assess the arithmetic performance of computers and their compilers. They are a mixture of vectorizable and non-vectorizable loops, and they test rather fully the computational capabilities of the hardware, and the skill of the software in compiling efficient code, and in vectorization.

The main value of the benchmark is the range of performance that it demonstrates, and in this respect it complements the limited range of loops tested in the LINPACK benchmark. The benchmark provides the individual performance of each loop, together with various averages (arithmetic, geometric, harmonic) and the quartiles of the distribution. However, it is difficult to give a clear meaning to these averages, and the value of the benchmark is more in the distribution itself.

In particular, the maximum and minimum give the range of likely performance in full applications. The ratio of maximum-to-minimum performance has been called the *instability* or the *speciality*, and is a measure of how difficult it is to obtain good performance from the computer, and therefore how specialized it is.

2.1.3 Whetstone

Whetstone is a synthetic floating-point benchmark, developed in 1976, containing ten modules that perform numerical calculations. The results are very sensitive to the library function calls and the size of the processor cache. Results are Whetstones per second.

2.1.4 Drystones

This benchmark spends a significant amount of time on string functions. It was designed to measure the integer performance of small machines which had a simple architecture and instruction set. Programs are small and fit into cache. RISC machines generally beat CISC machines on this benchmark. Performance is measured in Drystones per second.

2.2 SPEC Benchmarks

The System Performance Evaluation Cooperative (SPEC) was founded in 1988 by a small number of workstation vendors who realized that the marketplace was in need of realistic, standardized performance tests.

SPEC is a non-profit corporation whose membership is open to any company or organization that is willing to support the SPEC goals

SPEC has grown to become one of the more successful performance standardization bodies with more than 40 member companies. SPEC publishes several hundred different performance results each quarter spanning a variety of system performance disciplines.

The goal of SPEC is to ensure that the marketplace has a fair and useful set of metrics to differentiate candidate systems.

The basic SPEC methodology is to provide the benchmarker with a standardized suite of source code based upon existing applications that have already been ported to a wide variety of platforms by its membership. The benchmarker takes this source code, compiles it for the system in question, and then can tune the system for the best results. The use of already accepted and ported source code greatly reduces the problem of making apples-to-oranges comparisons.

2.2.1 Structure of SPEC

SPEC has evolved into an umbrella organization encompassing three diverse groups.

1. The Open Systems Group (OSG)

   OSG is the original SPEC committee. This group focuses on benchmarks for high-end workstations and servers running open systems environments. The OSG Subcommittees are:

   - The CPU committee defines CPU benchmarks, such as SPECint, SPECfp, SPECrates, and so on.
   - The SFS committee defines file server benchmarks such as LADDIS and SPECweb96.
   - The SDM committee defines multiuser UNIX command benchmarks.

2. The High-Performance Group (HPG)

   The HPG is a forum for establishing, maintaining and endorsing a suite of benchmarks that represent high-performance computing applications for standardized, cross-platform performance evaluation.

   These benchmarks target high-performance system architectures, such as symmetric multiprocessor systems, workstation clusters, distributed memory parallel systems, and traditional vector and vector parallel supercomputers.

3. The Graphics Performance Characterization Group (GPC)

The GPC joined SPEC in early 1996. GPC Projects include:

- The Picture Level Benchmark (PLB) project, working on standardized measures to represent “vectors-per-second” and “polygons-per-second” graphics capabilities.
- The X Performance Characterization (XPC) project, developers of the Xmark93, a standardized benchmarking tool for systems running the X Window System.
- The OpenGL Performance Characterization (OPC) project, working on ways of characterizing performance using the OpenGL application programming interface.

2.2.2 SPEC Benchmarks of Commercial Interest

The following SPEC benchmarks reflect the performance of the microprocessor, memory architecture, and compiler of the tested system:

- SPECint95

  A SPEC component-level benchmark that measures integer performance. The result is the geometric mean of eight tests that comprise the CINT95 benchmark suite. All of these are written in the C language. SPECint_base95 is the result of the same tests in CINT95 with a maximum of four compiler flags that must be used in all eight tests.

- SPECint_rate95

  A geometric average of the eight SPEC rates from the SPEC integer tests (CINT95). SPECint_base_rate95 is the result of the same test as CINT95 with a maximum of four compiler flags that must be used in all eight tests.

- SPECfp95

  A SPEC component-level benchmark that measures floating point performance. The result is the geometric mean of ten tests, all written in FORTRAN, that are included in the CFP95 benchmark suite. SPECfp_base95 is the result of the same test in CFP95 with a maximum of four compiler flags that must be used in all ten tests.

- SPECfp_rate95

  A geometric average of the ten SPEC rates from SPEC floating point tests (CFP95). SPECfp_base_rate95 is the result of the same tests as CFP95 with a maximum of four compiler flags that must be used in all ten tests[2].

- SPECwEB96

  The workload simulates the accesses to a Web service provider, where the server supports the home page for a number of different organizations. The primary Metric (SPECweb96) is the peak throughput measured during the benchmark run (reported in operations per second)

  SPECweb96 is targeted at system vendors, software vendors, and customers seeking performance data for Web server purchasing. In its initial release, the benchmark focuses on server performance for static Web pages, measuring the ability of the server to service HTTP requests or *gets*. One or more clients are used by SPECweb96 to send the HTTP requests to the Web server. The software then measures the response time for each request. At

[2] For a more detailed discussion of the these performance benchmarks refer to: *The Benchmark Handbook For Database and Transaction Processing Systems*, Second Edition, edited by Jim Gray, Morgan Kaufmann Publishers, San Mateo CA, 1993

the end of the benchmark run, SPECweb96 calculates a metric based on overall throughput, measured as maximum benchmark operations per second.

The SPECweb96 workload is based on analyses of server logs from a variety of popular Internet servers and some smaller Web sites. To further validate the workload, data from the analyses was compared to logs from Netscape and CommerceNet. Workload files are divided into four classes according to size, from less than 1 KB to slightly less than 1 MB. Access patterns to the files were determined according to the analyses of server logs. The access analyses reflect real-world usage for a Web service provider, with certain files being more popular than others.

Organizations involved in the development of SPECweb96 include CommerceNet, Digital Equipment Corp., HAL Computer Systems, Hewlett-Packard, IBM, Intel, Netscape, OpenMarket,Siemens Nixdorf Informationssysteme, Silicon Graphics, Spyglass, and Sun Microsystems.

Table 2 shows the top five results as of the time of writing:

Table 2. Top Five SPECweb96 Results (as at June 1st 1999)

Configuration	Result	HTTP Version	N-way
HP 9000 N4000	24,139.0	Zeus 1.3.3	8
IBM S/390 XY6	21,591.0	IBM HTTP Server V5.1 for OS/390	10
IBM RS/6000 S7A	20,200.0	IBM HTTP Server 1.3.4	12
IBM RS/6000 S70	19,264.0	IBM HTTP Server 1.3.4	12
Compaq AlphaServer GS140 6/575	14,263.0	Zeus 1.3.0	10

- SPECjvm98

  Measures the performance of Java Virtual Machines. It is applicable to networked and standalone Java client computers, either with disk (for example a PC or workstation) or without disk (for example, a network computer) executing programs in an ordinary Java platform environment. The benchmark requires a Java Virtual Machine compatible with JDK 1.1 API, or later.

  The SPECjvm98 benchmark suite contains eight different tests, five of which are either real applications or derived from real applications that are commercially available. The tests measure the time it takes to load the program, verify the class files, compile on the fly if a just-in-time (JIT) compiler is used, and execute the test. Each test is run several times and two scores are generated: a “worst” score for the slowest time and a “best” score for the fastest. A geometric mean is used to compute a composite score for all tests. Test scores are normalized against a reference machine--a midrange IBM PowerPC 604 with a 133 MHz processor. Higher scores indicate better performance.

The SPECInt and SPECfp benchmarks are a series of simplistic kernels that attempt to measure integer and floating-point performance of a CPU. These are all single-threaded tasks consisting of CPU-intensive programs. Neither the operating system nor the I/O subsystem is stressed.

Therefore, despite SPEC claims to represent commercial workloads, the SPECInt and SPECfp numbers are of most value in comparing the compute-intensive ability of servers, rather than their ability to deliver commercial throughput.

2.3 WebSTONES

The WebSTONE HTTP Web Server benchmark was developed by Gene Trent and Mark Sake at Silicon Graphics.

WebSTONE works in a simple fashion by generating traffic on the World Wide Web′s HTTP protocol to create stressful conditions on a server. WebSTONE records five standard measurements during each test pass:

1. Average (and maximum) tool connect times
2. Average (and maximum) server response times
3. Data throughput rate of transactions
4. Number of pages retrieved from the server
5. Number of files retrieved from the server

Each page that WebSTONE retrieves is weighted by the tester to reflect the actual usage level that page has on the site; the home page should have a high weighting, as it is nearly always retrieved by an end user, while a site′s help pages may have a very low weighting, reflecting their infrequent use. The higher the weighting, the more often WebSTONE will load it during testing.

A WebSTONE benchmark requires both the HTTP server under test and a WebMASTER session on a separate UNIX workstation, which initiates the individual test processes (called Webchildren) and collects the test results when the testing is completed. The Webchildren run on a small number of client workstations (multiple test processes can be run on each client, depending upon the overhead requirements for each). The WebMASTER generates an end-of-test report based on these measurements.

WebSTONE also provides for four different workload mixes:

1. A general modem mix, which attempts to account for users on modem connections, accessing a site with pages that are engineered to be as small as possible to leverage that audience.
2. A general mix, a “middle of the road” workload that assumes page content sizes up to 100 K to maximize site performance.
3. A media-rich mix, a workload that is driven by graphics and digital data content, with page sizes ranging from 20 KB up through multiple megabytes.
4. A general and media-rich mix, which provides coverage for a site with both large and small content pages.

WebSTONE focuses on the raw performance under the existing test network conditions. WebSTONE also does not simulate the performance of the WWW client, which may be a factor for many Web sites (particularly commercial sites that use client-intensive content like VRML).

WebSTONE 2.0 is available as C source and as a ported Windows NT application from SGI.

2.4 NotesBench

The NotesBench Consortium is an independent, non-profit organization dedicated to providing Domino and Notes performance information to customers. The consortium′s primary focus is to reduce customers′ expenses associated with internal benchmark activities, develop a base of performance information, serve as an industry conduit for specifying future benchmarks, and to ensure more rapid and optimized deployment of Domino and Notes.

Lotus NotesBench for Lotus Notes R4 is a collection of benchmarks and documentation for evaluating the performance of Notes R4 servers. The benchmarks model the behavior of Notes workstation-to-server or server-to-server operations. They return measurements to evaluate server performance in relation to the server system′s cost of ownership.

NotesBench supports any platform that can run Notes 4.5 or later including:

- Windows NT on Intel
- WindowsNT on Alpha
- OS/2
- Netware
- Macintosh PowerPC
- SCO UNIX
- HP-UX
- AIX
- Solaris
- OS/390

Since customer usage of Notes covers a vast spectrum of workload scenarios, there are nine different NotesBench workloads. Even though the server product name is now Domino, the databases are still referred to as Notes databases.

2.4.1 NotesBench Benchmarks

The following is a list of each benchmark test; a fuller description of each benchmark can be found in Appendix A, “NotesBench Benchmarks” on page 89. Note that the Idle, Mail and MailDB are the most commonly used metrics:

- Idle
- Mail
- Shared Discussion DB
- MailDB
- Groupware
- Calendaring and Scheduling
- Web Walker
- Replication Hub
- Mail Routing Hub

2.5 SAP SD Benchmark

SAP R/3 is an enterprise resource planning (ERP) application based on a single integrated database. It has cross-industry application modules such as human resources, sales, accounting, and industry-specific application modules (for example oil, banking and manufacturing). Because these application modules all access a single database and the system is based on work processes, SAP R/3 is able to cope with the varied demands that businesses require.

Because SAP R/3 databases tend to grow very quickly, they can become difficult to manage in a fairly short timeframe. Due to the availability needs of these mission-critical systems, the window to back up and reorganize data is often very short.

None of these attributes are measured by any of the SAP benchmarks.

The three-tier nature of SAP R/3 is illustrated in Figure 1. On some platforms, both the application and database servers can reside on the same server (for example, the IBM S/390, or SP).

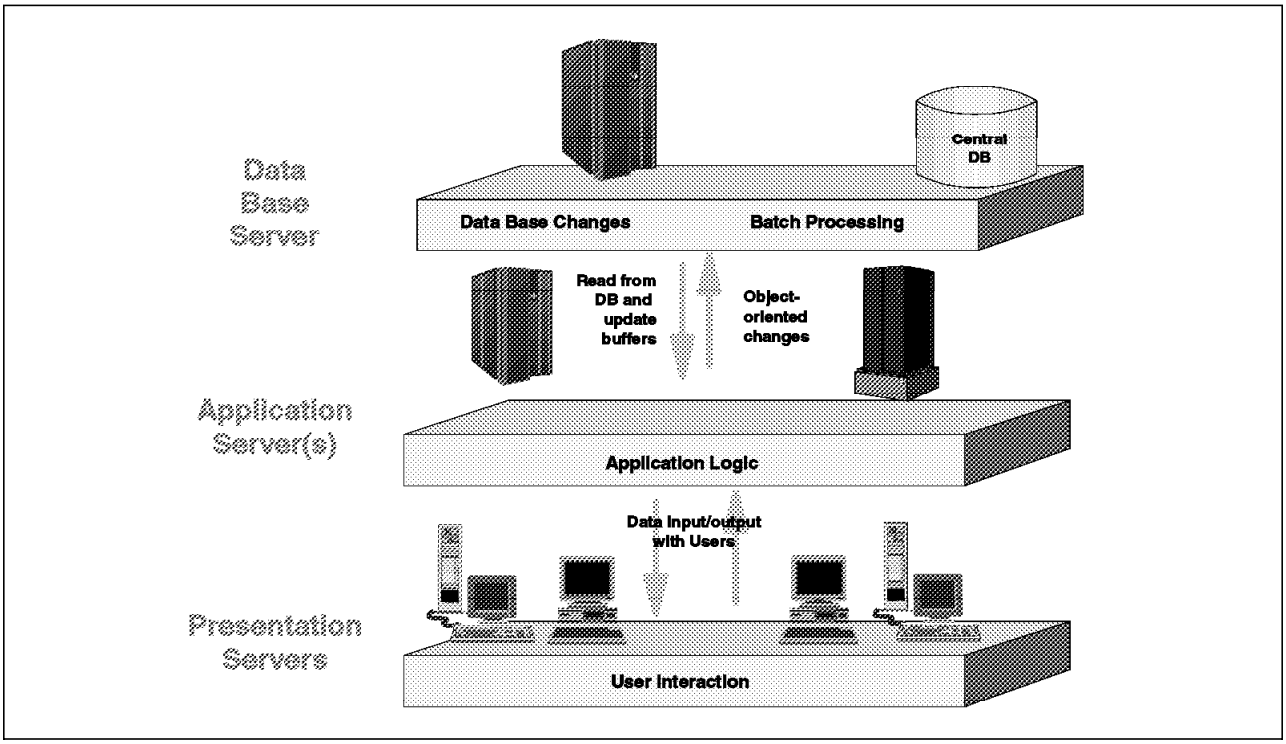


Figure 1. SAP R/3 DB Server on S/390

The SAP Standard Application Benchmarks have been available since the R/3 software Release 1.1H (April 1993). Currently, there are seven different SAP application benchmark modules available.

The first six are dialog benchmarks, with the Human Resources module being a batch benchmark. Results cannot be compared between two different modules.

These are:

- Sales and Distribution (SD)

  The SD benchmark consists of all the following transactions:

  - Create an order with five line items
  - Create a delivery for this order
  - Display the customer order
  - Change the delivery and post goods issue
  - List 40 orders for one sold-to party
  - And finally, create an invoice

- Financials (FI)

  The FI benchmark consists of:

  - Four customer postings
  - The line item display of the last posting
  - The display of the open items of the posted customer (44 items)
  - The balancing of four open items

  At the end of each run, there are exactly 40 open items, which in turn serves as a base for a new run.

- Materials Management (MM)

  The MM benchmark does the following:

  - Creates a purchase requisition for five materials
  - Creates a purchase order with reference to the purchase requisition
  - Creates a goods movement
  - Creates an invoice for the purchase order
  - Posts the invoice

  In order to enable parameter exchange between the transactions, all document numbers are transferred using SGA/PGA (Set Parameter/Get Parameter).

- Production Planning (PP)

  The PP benchmark consists of the following transactions:

  - Creates a production order
  - Change this production order (change amount, release order for production and print order)
  - Create two completion confirmations for the production order (milestone and final confirmation)
  - Post goods receipt for the order

- Warehouse Management (WM)

  The WM benchmark consists of the following transactions:

  - Create a transfer requirement to put three materials on stock

    Use this transfer requirement to create a transfer order
  - Confirm the transfer order for seven items

  These transactions are called again in the same sequence for goods removal.

- Project System (PS)

  The (PS) benchmark consists of the following transactions:

  - Create a project using a project profile
  - Execute the cost planning for the project
  - Approve the project
  - Budget the input necessary for the project
  - Finally, start one report doing cost evaluations.

- Human Resources (HR)

  The HR benchmark is a batch-mode background benchmark

At this time SD, and its derivative SD Parallel, are by far the most popular of the SAP benchmarks, although individual organizations have been known to define their own SAP benchmarks for other modules (HR, for example).

SD tries to simulate a standard sales, financial, and distribution process of a business. That is, it simulates a company taking an order, the scheduling of

manufacturing, shipment, and financial updates. The SD benchmark is intended to be run on many different platforms. The benchmark is run by the different hardware partners of SAP on their own machines.

SAP carefully reviews the run and the results. SAP certifies only those runs which met their criteria. Of course, like many standard benchmarks, these are strongly used in the marketing of the vendor's systems.

SAP R/3 can be implemented in a two-tier or three-tier system. In the two-tier implementation, the *database server* and *application server* run on the same hardware platform. The clients build the first tier and are responsible for handling the presentation layer.

In a three-tier implementation, the database server and the application server are split across different systems. At the time of writing, the S/390 implementation is three-tier with only the database server on the S/390 (the availability of the application server on S/390 in 4Q99 has been announced by SAP AG and IBM). Therefore, only benchmark results which are made using a three-tier environment may be compared with S/390 results.

Similarly, the SD parallel benchmark cannot be compared with the standard SD benchmark[3].

The application server runs the real application logic and submits remote SQL requests to the database server. You can consider the third tier S/390 as just as an SQL engine. The SD workload has about 80 SQL statements per dialog screen. The database server does not see these dialog screens directly, nor does it see how many users are driving the load. The SD benchmark has 15 dialog steps.

A detailed description of this benchmark and other benchmarks can be found in *Standard Application Benchmark Description* on the Internet at:

`www.sap-ag.de/products/techno/media/pdf/`

The SD Parallel benchmark is a significantly different environment to the normal SD benchmark. Although some vendors have not run this benchmark and have attempted to pass it off as trivially different from the normal SD benchmark (and, by implication, suggesting that their results for the SD benchmark are comparable to SD Parallel results), the Parallel SD benchmark is a much more demanding test than the SD benchmark.

Overall the Parallel SD benchmark appears to be much closer to real production workload profiles than strictly partitioned data benchmarks.

The major difference in the Parallel SD benchmark lies in its round-robin scheduling methodology. This results in:

- All business units being spread across all database servers
- Significant inter-system sharing (approximately 50% of transactions versus approximately 5% in the regular SD test)
- A large numbers of Inserts

[3] As of the time of writing, only IBM has published SD Parallel results.

This all probably means that there is a greater overhead (for example, Parallel Sysplex coupling overhead) than would be found in normal SAP R/3 production systems.

Thus the Parallel SD benchmark can be considered a realistic, albeit slightly conservative, measurement for SAP R/3 SD systems.

Table 3 shows the leading results for the SD two-tier benchmark as of the time of writing.

Table 3. Top SAP R/3 SD Two-Tier Benchmark Results (as at June 1st 1999)

Configuration	# Users	Response Time	Database	Version	N-way
Sun E10000 250 MHz	1,410	1.85	Informix 7.21 UC1	3.0E	64
IBM AS/400e 9406-S40	725	1.18	DB2/400 V2R4	3.0F	12
Sun E6000	600	0.74	Oracle 7.1.6	2.2G	24
IBM AS/400e 9406-S40	330	0.77	DB2/400 V4R1	3.0E	12
Digital AlphaServer 8400 5/300	300	1.85	Oracle 7.1.4	2.2C	8

Table 4 shows the leading results for the SD three-tier benchmark as of the time of writing.

Table 4. Top SAP R/3 SD Three-Tier Benchmark Results (as at June 1st 1999)

Configuration	# Users	Response Time	Database	Version	N-way
Sun E10000 336MHz	14,400	0.97	Oracle 8.0.4	3.1H	64
HP V2250	6,750	1.95	Oracle 8.0.4	3.1H	16
IBM AS/400e 9406-S40	6,651	1.88	DB2/400 V4R3	3.0F	12
IBM AS/400e 9406-S40	6,060	0.84	DB2/400 V4R3	3.1H	12
HP 9000 V2250	6,200	1.4	Informix 7.3.0 FC6	3.1H	16

Table 5 shows the leading results for the SD parallel benchmark as of the time of writing.

Table 5 (Page 1 of 2). Top SAP R/3 Parallel SD Results (as at June 1st 1999)

Configuration	# Users	Response Time	Database	Version	N-way
IBM S/390 3 x RY5	8,000	1.58	DB2 for OS/390 V5.1	3.1H	3 x 10
IBM S/390 3 x RY5	6,900	1.69	DB2 for OS/390 V5.1	3.1H	3 x 10
IBM S/390 4 x RX3	3,400	1.50	DB2 for OS/390 V5.1	3.0E	4 x 10

Table 5 (Page 2 of 2). Top SAP R/3 Parallel SD Results (as at June 1st 1999)					
Configuration	**# Users**	**Response Time**	**Database**	**Version**	**N-way**
IBM S/390 3 x RX3	2,570	1.29	DB2 for OS/390 V5.1	3.0E	3 x 10
IBM 5 x RS/6000 SP	1,700	0.69	Oracle Parallel Server 7.3.2	3.0D	5 x 8

2.6 RAMP-C

RAMP-C is a commercial benchmark of approximately the same complexity as TPC-C. There is no industry association that sets standards for the running and reporting of results from this benchmark. Hence, results are not comparable across vendors.

RAMP-C was used extensively by IBM to size and compare its AS/400 and RS/6000 server products. While it was closer to actual customer production environments than many Industry Standard benchmarks, its proprietary nature means that it is no longer used.

2.7 LSPR

Large Systems Performance Reference (LSPR) is a proprietary set of commercial workloads that measure the processor throughput rates of S/390 architected machines. The benchmark is owned by IBM. Results are reported as a set of ratios, one machine to another, for each individual workload.

The following table summarizes several differences between TPC-defined benchmarks and the LSPR set of benchmarks.

Table 6. Major Differences between LSPR and TPC Benchmarks	
TPC	**LSPR**
Cross architecture	S/390 architecture only
Simple OLTP workloads	Suite of workloads representative of mainframe (S/390) computing environments
Industry Standard	Proprietary, owned by IBM although the results are public
Measures performance and price/performance Measures	Measures performance
External throughput-based (measures entire system)	Internal throughput-based (measures the processor)
Too many variables to be useful for capacity planning	Results can be used for processor capacity planning
Synthetic for benchmarking only	Representative mix for S/390

There are several differences which should be specifically noted.

- The biggest difference by far is that with LSPR, the customer gets a set of measurements which shows the relative performance of nearly every S/370

and S/390 system including non-IBM so-called “plug-compatible systems.” Data is provided for multiple operating system environments (OS/390, MVS/XA, VSE/ESA, and VM/ESA) and for S/370 and S/390 processors as much as 10 years old.

- In a S/390 environment, typically several applications run on the same instance of OS/390. Therefore it is of interest how the size of a processing complex must be varied if one type of application requires more computing power. LSPR data allows you to do this.

2.8 The Transaction Processing Council (TPC) Benchmarks

In October 1988, a consultant named Omri Serlin invited computer vendors to form and participate in a council with the mission of developing Industry Standard benchmarks for the online transaction processing (OLTP) environment. This group is known as the Transaction Processing Performance Council (TPC). IBM was one of the first members of the Council, joining in November 1988.

Before the formation of the TPC there were a number of unregulated benchmarks such as TP1 and DebitCredit. TP1 was a benchmark originally developed within IBM that then found its way into the public domain. This benchmark purported to measure the performance of a system handling ATM transactions in a batch mode without the network or user interaction (think-time) components of the system workload (similar in design to what later turned out to be TPC-B). The TP1 benchmark had two major flaws:

1. By ignoring the network and user interaction components of an OLTP workload, the system under test (SUT) could generate inflated performance numbers.
2. The benchmark was poorly defined and there was no supervision or control of the benchmark process.

In the April 1, 1985 issue of Datamation, a group of authors outlined a test for on-line transaction processing which was given the title of DebitCredit. Unlike the TP1 benchmark, the DebitCredit benchmark specified a system-level benchmark where the network and user interaction components of the workload were included. In addition, it outlined several other key features of the benchmarking process that were later incorporated into the TPC process:

- The total system cost should be published with the performance rating. Total system cost included all hardware and software used to successfully run the benchmark, including five years maintenance costs.
- The test should be specified in terms of high-level functional requirements rather than specifying any given hardware or software platform or code-level requirements. This allowed any company to run this benchmark if they could meet the functional requirements of the benchmark.
- The benchmark workload scaleup rules--the number of users and size of the database tables--increased proportionally with the increasing power of the system to produce higher transaction rates. The scaling prevented the workload from being overwhelmed by the rapidly increasing power of OLTP systems.
- The overall transaction rate would be constrained by a response time requirement. In DebitCredit, 95 percent of all transactions had to be completed in less than 1 second.

From 1985 through 1988, vendors used TP1 and DebitCredit--or their own interpretation of these benchmarks--to make performance claims for their products.

In order to bring some credibility to such benchmarks and to introduce an agreed process for “Industry Standard” benchmarks, the Transaction Processing Performance Council (TPC) was formed.

The TPC is a non-profit organization whose mission is to “define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry.”

The first task of the TPC was to standardize the DebitCredit benchmark and provide specifications for standard implementation, measurement and reporting of results. This resulted in the TPC-A benchmark which was generally released in November 1989. Since then, many additional benchmarks have been constructed (see Table 7 on page 20).

Each benchmark is designed to reflect processing in a typical customer environment across a spectrum of workloads.

Manufacturers run the benchmarks at their own expense and need to conform to a set of rules laid down by the TPC. The results of a benchmark require independent auditing and are published in a Full Disclosure Report (FDR) which is written by the vendor. FDRs are available from the TPC via a third-party public relations firm, Shanley PR, which has been contracted to provide administrative and operational support.

Be aware that the usage of the TPC benchmarks must be within the rules of “Fair Use” of TPC results (B.2, “Avoiding Unfair Use of TPC Results” on page 100). IBM as a member of Transaction Processing Council is committed to abiding by these rules. The TPC′s Fair Use policies were adopted in June, 1991.

These require the following criteria in the running and publicizing of TPC benchmarks:

- Fidelity: Adherence to facts; accuracy
- Candor: Above-boardness; needful completeness
- Due Diligence: Care for integrity of TPC results

TPC benchmark selection criteria are as follows:

- Applicability - Is it relevant to a large number of users?
- Comparability - Will it lead to objective comparisons?
- Understandability - Can users/press understand its significance?
- Executability - Can it be run in a reasonable time period and for a reasonable cost?

Note that these benchmarks will measure the performance of both the hardware, and of the operating system and database management subsystem.

Each benchmark generally has two primary metrics:

1. Throughput, which is the maximum throughput expressed in transactions per unit of time (second, minute, hour)
2. The cost of ownership per transaction per second, minute, or hour

The cost of ownership is based on purchase and five years maintenance for all hardware and software required to achieve the maximum throughput rate.

2.8.1 The TPC Benchmarks

Table 7 summarizes various TPC benchmarks that have been developed or proposed since the founding of the TPC in 1988. The same information is shown diagrammatically in Figure 2 on page 21.

Table 7. The TPC Benchmarks

Benchmark	Approval Date	Status	Type of benchmark
TPC-A	October 1989	Obsolete as of June 1995.	Simple OLTP debit/credit.
TPC-B	August 1990	Obsolete as of June 1995.	Simple batch debit/credit.
TPC-C	July 1992	Current. Latest level is Version 3. Version 4 is under development.	Moderate complexity OLTP debit/credit.
TPC-D	April 1995	To be replaced by TPC-H and TPC-R in June 1999.	Data warehouse/decision support queries.
TPC-H		Replaces TPC-D in June 1999	Heavy query oriented decision support - formerly known as TPC-D version 2.0.
TPC-R		Replaces TPC-D in June 1999	Reporting for decision support - formerly part of TPC-D Version 2.1.
TPC-E		Rejected by council vote.	Enterprise computing, concurrent OLTP and batch, software recovery
TPC-S		Never voted upon, effectively obsolete.	Server-only benchmark.
TPC-W		Targetted to be available YE99.	Web serving, e-Commerce.
TPC-c/s		Never voted upon, effectively obsolete.	Version of TPC-C with additional Client/Server function.

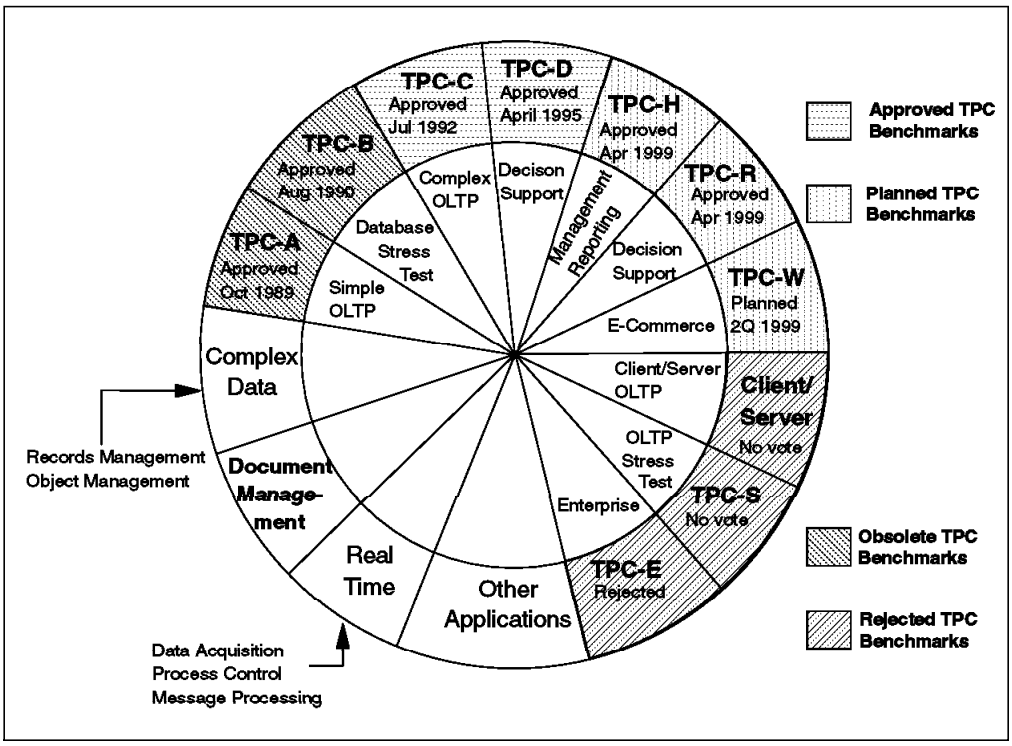


Figure 2. TPC Benchmark Overview

2.8.1.1 TPC-A and TPC-B

As noted, the TPC-A and TPC-B benchmarks are now obsolete. However, the current TPC-C is descended from these benchmarks, so their characteristics are worth discussing.

TPC-A: Using the model and the consensus that had already developed around the DebitCredit benchmark, the TPC published its first benchmark, TPC Benchmark A (TPC-A) in November 1989. TPC-A differed from DebitCredit in the following respects:

- The requirement that 95% of all transactions must complete in less than one second was altered to 90% of transactions must complete in less than two seconds.
- The number of emulated terminals interacting with the System Under Test (SUT) was reduced to a requirement of 10 terminals per Transaction Per Second (tps) and the cost of the terminals was included in the system price.
- TPC-A could be run in a local or wide area network configuration (DebitCredit has specified only WANs).

The production-oriented requirements of the benchmark were strengthened to prevent the reporting of peak, unsustainable performance ratings. Specifically, the ACID requirements (atomicity, consistency, isolation, and durability) were bolstered and specific tests added to ensure ACID viability (see B.2.1.1, “The ACID properties” on page 102).

Finally, TPC-A specified that all benchmark testing data should be publicly disclosed in a Full Disclosure Report.

The first TPC-A results were announced in July 1990. Four years later, at the peak of its popularity, 33 companies were publishing TPC benchmarks and 115 different systems had published TPC-A results. In total, about 300 TPC-A benchmark results were published.

The TPC-A benchmark represents a simple debit/credit banking application. The bank has a number of branches with associated tellers and accounts. Transactions are either a deposit or a withdrawal against an account. The application updates the account, teller, and branch balances, and records the transaction in a history file.

The transaction profile is as follows:

```
Read 100 bytes including Aid, Tid, Bid, Delta from terminal

BEGIN TRANSACTION

Update Account where Account_ID = Aid:
       Read Account_Balance from Account
       Set Account_Balance = Account_Balance + Delta
       Write Account_Balance to Account
Write to History
       Aid, Tid, Bid, Delta, Time_stamp
Update Teller where Teller_ID = Tid:
       Set Teller-Balance = Teller-Balance + Delta
       Write Teller_Balance to Teller
Update Branch where Branch_ID = Bid:
       Set Branch_Balance = Branch_Balance + Delta
       Write Branch_Balance to Branch

COMMIT TRANSACTION

Write 200 bytes including Aid, Tid, Bid, Delta, Account_Balance
to terminal.

Aid, Tid, and Bid are keys to the relevant records/rows.
```

The TPC-A benchmark was originally available in October 1989. Reporting of TPC-A benchmarks ended in June 1995 and listing of TPC-A results ended in December 1995.

2.8.1.2 TPC-B

TP1 (and later TPC-B) was the batch version of DebitCredit, without the network and user interaction (terminals) figured into the workload. A strong block of companies within the TPC felt that the TPC-B model was more representative of customer environments.

Others within the TPC argued that the partial-system model that TPC-B represented would reduced the stress on key system resources and would therefore produce artificially high transaction rates. In addition, while the transaction per second (tps) rates would be artificially high, the total system cost, since the network and terminal pricing would be eliminated, would be artificially low, thereby artificially boosting TPC-B′s price/performance ratings.

TPC-B used the same TPC-A banking transaction but it removed the network and user interaction component of the TPC-A workload. What was left was a batch transaction processing benchmark.

The first TPC-B results were published in mid-1991 and by June 1994, at the peak of its popularity, TPC-B results had been published on 73 systems. In total, about 130 TPC-B tests were published. In all, there were about 2.5 times more TPC-A results published.

TPC-B never received the user or market analyst acceptance that TPC-A did, but many within the Council and the industry perceived a real value in this "database server" benchmark model. This belief fueled later failed TPC efforts to produce benchmarks that were more representative of enterprise computing environments (TPC-E and TPC-S).

2.8.1.3 TPC-C

TPC-C attempted to plug some of the loopholes in TPC-A and to address the criticisms that TPC-A was too simplistic.

In order to do this TPC-C does the following:

- It includes five transactions instead of just one.
- It has an average pathlength of 10 times TPC-A.
- It creates significantly more disk and terminal I/O.
- The terminal costs are no longer part of the total costs. (TPC-A was sometimes referred to as a benchmark to find the cheapest ASCII terminal).

The benchmark itself is intended to represent the transactions of a typical commercial business engaged in selling and distributing a product (see Figure 3 on page 24 for an overview comparison of TPC-C to typical commercial OLTP systems).

The business has warehouses which stock items, and sales districts with customers who place orders. The workload consists of five unique transactions, selected at various frequencies to produce the workload mix. The transactions are:

- NEW ORDER (45%) - Process an order for an average of ten items. Items are selected, stock is updated and packages are created, so that this transaction is update/insert-intensive.
- PAYMENT (43%) - Payment for an order updates the balance for the customer and the sales totals for the warehouse and district, which makes this transaction update-intensive. *This is effectively the TPC-A transaction.*
- ORDER STATUS (4%) - Inquire on the status of a customer's last order. This transaction is read only.
- DELIVERY (4%) - Process the delivery of 10 new orders. This transaction is intended to be executed in batch mode and is update/delete-intensive.
- STOCK LEVEL (4%) - Find the recently sold items whose stock level falls below a threshold. This is a heavy read-only transaction.

With the exception of the Stock Level transaction, 90% of the transactions must complete within five seconds. For the Stock Level transaction, 90% must complete within 20 seconds.

The primary metrics are:

- tpmC - measure of performance in transactions per minute (tpm)
- $/tpmC - measure of price performance, five-year cost of ownership of tested configuration divided by the performance.

As with other TPC benchmarks, there are “other” metrics which are included in each result’s Executive Summary or Full Disclosure Report. However, the previously mentioned primary metrics are the metrics which must be quoted when referring to an individual TPC-C Result

Because the benchmark specification outlines a business model, and does not require any specific hardware or software, companies are free to use any combination of hardware or software in the tested configuration. The limitations basically being that the products used, must be generally available to the public now, or within the next 6 months. This can mean that some tests include hardware and/or software products that are not generally available at the time of the test.

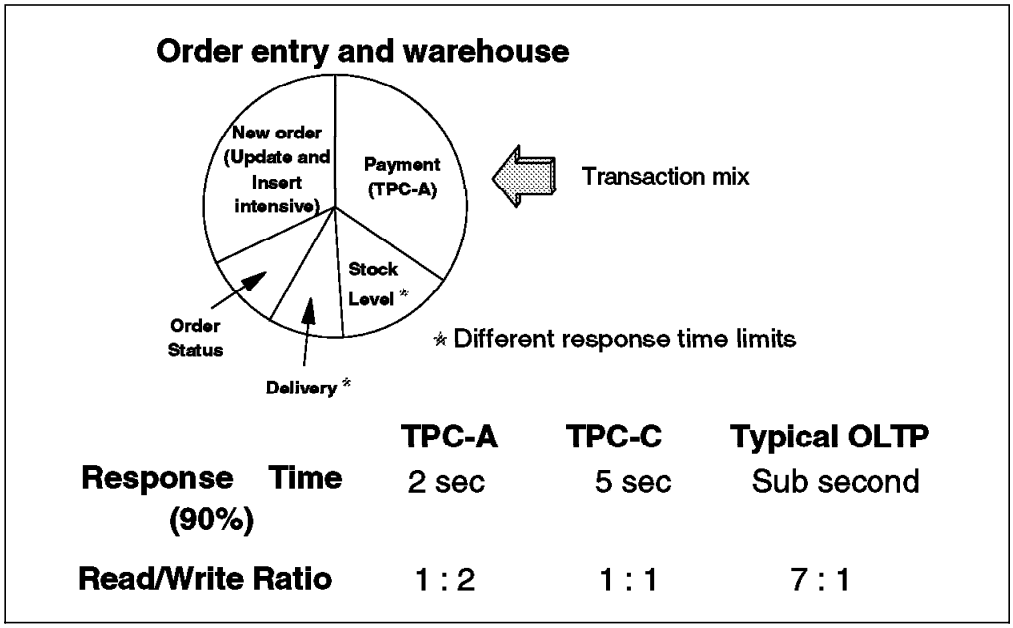


Figure 3. TPC-C Transaction Mix and Comparison to Typical Commercial OLTP

The TPC-C benchmark is a test of a configuration, not an individual system. Vendors have the option of running the TPC-C benchmark as either a standalone system (see Figure 4 on page 25) or in a client/server configuration.

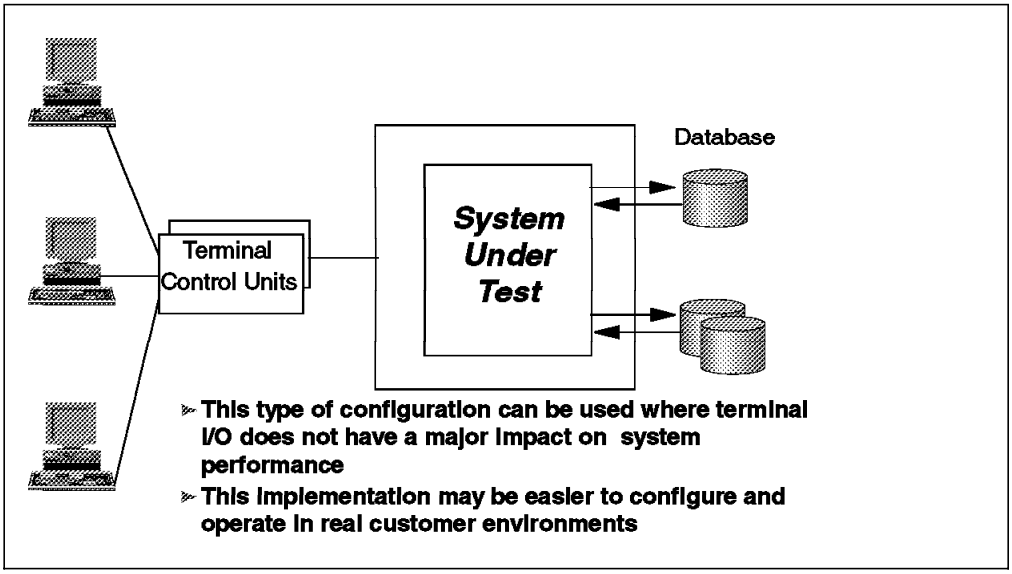


Figure 4. Standalone Processor Measurement Configuration

The front-end systems accept terminal input from non-intelligent terminals and then act as clients, passing on transaction requests to the back-end systems acting as servers. The back-end system is the machine that gets the credit for the test and is nominated in the title of the result (see Figure 5).

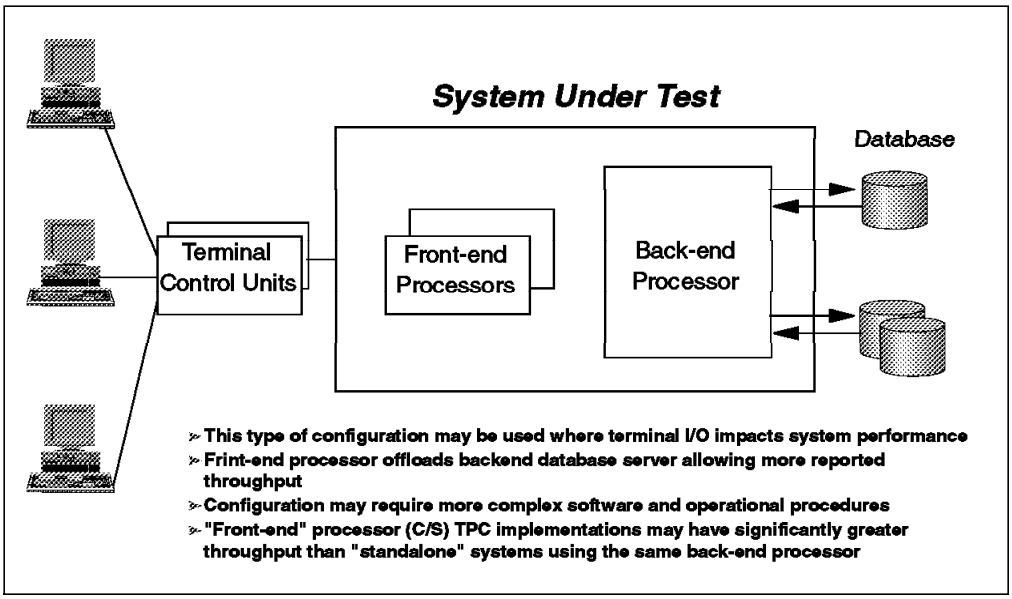


Figure 5. Front End Back - Back End Configurations

The front-end processors are used to minimize the overhead on the server or back-end system, and thus maximize overall performance. It is now rare for a TPC-C configuration to consist of a single system. In fact, a review of the TPC-C results as of May 7, 1999 showed no standalone configurations.

All current results are either client/server or clusters. Table 8 on page 26 shows the top five client/server results at the time of writing:

Table 8. Top Five TPC-C Client/Server Results (As at June 1st 1999)

Configuration	tpmC	$/tpmC	Database	N-way
Sun E10000 400MHz	115,395.73	$105.63	Oracle 8i V8.1.5.1	64
Sequent NUMA-Q 2000 E300	93,900.85	$131.67	Oracle Enterprise Edition 8.0.4	64
HP 9000 V2500	92,832.96	$86.94	Oracle Enterprise Edition V8.1.5.1	32
Sun E6500 336MHz	53,049.97	$76.00	Sybase ASE 11.7.3	24
HP 9000 V2250	52,117.80	$81.17	Sybase ASE 11.5 EBF 7817	16

Table 9 shows the top Five cluster TPC-C results as of the time of writing.

Table 9. Top Five TPC-C Cluster Results (As at June 1st 1999)

Configuration	tpmC	$/tpmC	Database	N-way
Digital AlphaServer 8400	102,541.85	$139.49	Oracle V8.0.5	8 nodes x 12-way
IBM RS/6000 SP Model 309	57,053.80	$147.40	Oracle Enterprise Edition 8.0.4	12 nodes x 8-way
Sun E6000 c/s	51,871.65	$134.46	Oracle Enterprise Edition V8.0.3	2 nodes x 22-way
Compaq ProLiant 6500-PDC/O 1000	33,935.90	$49.99	Oracle Parallel Server 8.0.5	4 nodes x 8-way
IBM RS/6000 R40 SMP	14,285.87	$229.00	Oracle7 V7.3	16

Comparison of different TPC-C results is very difficult because of the variety of combinations of software products that are used. For example, a group of tested systems may use different operating systems, databases, transaction monitors and other software which can be influencing performance, and it is therefore difficult to make accurate relative performance assessments of the hardware. This is unlike other benchmarks, where often the software and programs are standardized and the major variable is the hardware.

In particular it is important to only compare results using the same database management system (preferably at the same release level) and the same configuration (single system, client/server, or cluster). This is because a TPC-C result is a measure of a configuration and not necessarily a single system.

2.8.1.4 TPC-Client Server

The intent of TPC Client Server was to characterize performance in a real Client Server environment where the application is split between client and server. It was built on the TPC-C model with the addition of two new transactions while the database was essentially unchanged.

The transactions had GUI front ends and the goal was to create a significant workload on the clients including the continued processing of transactions while access to the server was down. Performance metrics were to be provided for both the client and the server.

It appears that TPC-C/S was not any more representative of the typical Client/Server implementation than TPC-C. The exploding exploitation of the Web has de facto obsoleted its concept and the project has never been voted upon by the council.

2.8.1.5 TPC-D

TPC-D is an attempt to benchmark one of the newer application areas: that of heavy query processing as encountered in Decision Support and Business Intelligence systems.

TPC-D became available in April 1995. The current revision of the TPC-D Benchmark Specification is version 1.3.1, released to the public on February 20, 1998 (see also 2.8.1.6, “TPC-H and TPC-R” on page 28).

TPC-D is made up of 17 queries which represent a broad range of decision support applications with a wide range of complexity and execution times. The 24/7 query environment simulated includes two database update jobs in addition to the queries.

Since the size of the database used for decision support can vary from the relatively small (1 GB) to the extremely large (many terabytes). The TPC-D benchmark allows for systems to be tested on one or more of several fixed database sizes. These are: 1 GB, 10 GB, 30 GB, 100 GB, 300 GB, 1 TB, 3 TB, 10 TB, and so on.

The primary metrics are:

- Qppd - Query Processing Performance metric provides a measure of raw query execution and shows how fast queries can be executed when all available power of the system is concentrated on a single query at a time.
- QthD - The Query Throughput Metric provides a measure of optimal execution of concurrent queries and shows how many queries the system can process in an hour.
- $/QphD - Queries Per Hour is a measure of price/performance. It is calculated dividing the five-year cost of ownership of the tested configuration by a composite of the two performance metrics.

All three metrics must be reported. The scaling factor (size of database in GB or TB) must be included. The results of different scaling factors cannot be compared.

In addition to the primary metrics, test sponsors are required to report the time taken to create and populate (load) the database.

Table 10 on page 28 shows the top five results for the 300 GB database as of the time of writing:

Table 10. Top Five TPC-D 300 GB Results (As at June 1st 1999)

Configuration	QthD	QppD	QphD	$/QphD	Database	N-way
NCR Worldmark 5200	13,756.2	133,966.8	42,928.80	$440.00	Teradata V2 R3.0	128
HP 9000 V2500	7,540.0	56,243.7	20,593.20	$208.00	Oracle8i V8.1.5.1	32
IBM RS/6000 SP Model 550	6,166.5	10,469.6	8,035.00	$721.00	DB2 UDB 5.2.0	24 x 4-way
Compaq AlphaServer GS140	4,868.7	29,711.6	12,027.40	$192.05	Oracle8i V8.1.5.1	12
Compaq AlphaServer GS140 6/575	3,487.8	8,273.1	5,371.70	$999.00	Informix IDS/AX/XD 8.21 UD2	4 x 10 way

Table 11 shows the top five results for the 300 GB database as of the time of writing:

Table 11. Top Five TPC-D 1 TB Results (As at June 1st 1999)

Configuration	QthD	QppD	QphD	$/QphD	Database	N-way
NCR Worldmark 5200	24,142.0	366,509.5	94,065.3	$273.00	Teradata V2 R3.0	44 x 4-way
IBM RS/6000 SP Model 550	10,661.5	19,137.5	14,284.10	$797.00	DB2 UDB 5.2.0	48 x 4-way
Sun E10000 400 MHz	19,566.3	121,824.7	35,878.10	$283.00	Oracle8i V8.1.5.1.2	64-way
IBM Netfinity 7000 M10	8,166.9	36,872.0	17,353.10	$352.00	DB2 UDB 5.2.0	32 x 4-way
Sun E10000 400 MHz	7,658.6	70,347.3	23,210.70	$438.00	Oracel8i 8.1.5.1.1	64-way

2.8.1.6 TPC-H and TPC-R

TPC-D was developed to represent an ad hoc business environment where users submitted more or less random queries against a data warehouse. The queries were designed to be complex and require significant processing.

Over the life of TPC-D, there have been significant changes in database technology. As a result, the execution times on several TPC-D queries have plunged nearly to zero.

Recently, various database vendors have released new technology which enables the building of structures that contain precomputed query aggregates. Without any changes to query text, queries can access these structures

transparently at run time and quickly calculate the value of needed aggregate expressions.

This technique moves much of the work that has previously been part of the query execution of TPC-D into the database load phase. The council members generally agree that this technology is valuable to end users. For certain types of business environments, this technology does improve query performance many-fold.

The aggregate technology is very useful when users (typically very knowledgeable users like database administrators) know the queries and the domain well in advance, can create auxiliary structures like aggregated columns, and can optimize their databases to run the queries. Many have called this type of environment a “business reporting” environment. The problem is that TPC-D was intended to represent an ad hoc environment in which queries are submitted on a random basis and are not known in advance.

At the February 1999 General Council meeting, the Council decided that TPC-D was one benchmark trying to represent two very different business environments.

1. An environment in which users know the queries very well and can optimize their DBMS to execute these queries very rapidly (business reporting environment)
2. The environment of the original ad hoc environment in which users do not know the queries in advance and the execution times can be very long

The Council voted to issue a mail ballot to the general membership asking it to divide the TPC-D benchmark into the following two benchmarks:

- TPC-R (Business Reporting Benchmark).

  In light of the new DBMS technology application to the TPC-D workload/rules, the TPC has modified the planned TPC-D Version 2.0 and created Version 2.1. TPC-D Version 2.1 eliminates the references to TPC-D′s ad-hoc business model. The language has now been modified to reflect the fact that TPC-D, as it currently stands, is more representative of the business reporting environment. If the TPC-R mail ballot passes, then Version 2.1 of TPC-D will become TPC-R Version 1.0 and TPC-D Version 2.1 will disappear.

- TPC-H (ad hoc benchmark).

  The TPC-D Subcommittee created a new ad-hoc benchmark that restores the “ad hocness” of the original benchmark workload. The Subcommittee used TPC-D Version 2.0 as the baseline specification, but then added language that restricts the use of auxiliary structures (such as indexes and aggregates), as well as regulating the horizontal partitioning of tables.

This mail ballot to the general membership was issued in late February 1999 and a decision is expected by late April 1999. If the mail ballot is approved by two-thirds of the members, TPC-D will disappear and be replaced by TPC-R and TPC-H.

TPC-D Version 2.0.1 (which retains the original ad hoc workload description) remains a valid TPC-D benchmark. Anyone can publish a result on this specification.

TPC-D Version 2.1 will become mandatory on April 11, 1999.

2.8.1.7 TPC-Enterprise

The intention of TPC-E was to reflect the complexity of a system appropriate to a large business or enterprise. TPC-E was based on TPC-C, but with extensions which introduced this complexity.

The attributes that TPC-E was intended to encapsulate are:

- Support of the concurrent execution of multiple transaction types
- Concurrent support of OLTP and batch work
- Tight response time objectives, consistent with real user service levels
- Ability to handle significant I/O loads
- Capability to handle a large and complex database
- Recovery from a significant system failure
- Demonstration of high security

TPC-E represented a departure from previous TPC benchmarks in that it introduced measurement and demonstration of factors over and above performance and price performance.

As has been previously noted, each member of the TPC has only one vote. Vendors who primarily market products to the NT market place saw no marketing value, and significant cost, to running TPC-E.

Consequently, the TPC-E benchmark was rejected by a vote of the Council and will now not be released.

2.8.1.8 TPC-Server

There were two requirements for this benchmark.

1. To reduce the cost of benchmarking

   TPC-S would not have required a terminal network to be measured and priced as part of the configuration. The transaction load could have been simulated by either an internal or external driver.

2. Oracle and hardware vendors who wanted to run Oracle on their processors

   These vendors had unsuccessfully tried to get the TPC Council to relax the isolation requirements on TPC-C.

TPC-S was considering a number of potential modifications, but the major points were the elimination of terminals and networks from the configuration, elimination of user contexts, and the relaxation of isolation. The servers would have only required to have the operating system and DBMS running on the system.

TPC-S has never been voted upon and it appears that this benchmark will now no longer be developed.

2.8.1.9 TPC-W

In January 1998, the TPC announced the formation of a Web Commerce benchmark (TPC-W) which will measure OLTP and browsing performance only of the Web server, excluding the network and human interaction components of the overall system. The TPC worked towards the completion of this benchmark specification by the second quarter of 1999.

TPC-W is designed to represent any business (retail store, software distribution, airline reservation, electronic stock trades, and so on) that markets and sells

over the Internet. It also represents Intranet environments which use Web-based transactions for internal operations. The benchmark will measure the performance of systems supporting users browsing, ordering, and conducting transaction-oriented business activities.

Chapter 3. Strengths/Weaknesses of Industry Standard Benchmarks

The relatively simple nature of the TPC benchmarks ensures that they can easily be implemented by many vendors, and the focus on price and performance generally leads to the cheapest peripherals being used with the lowest function.

While the TPC benchmarks claim to be end-to-end tests of performance, they are not so stressful that they require any advanced functions. Features of IBM disk controllers such as dynamic cache management or record caching are effectively made redundant. Whether the peripherals are ESCON-, parallel-, FICON-, OSA- or SCSI-attached will make little difference. The use of tape is effectively ignored, and real network performance is not even a consideration. Yet we know that all these factors feature very strongly when businesses look for ways to meet the latest demand from their users for improved service levels.

Defining a performance benchmark is a complex process even when only a single company is part of the definition process. The following issues must be addressed:

- What is a *typical* workload; how much contention should there be; should the contention be in the hardware, software, CPU, DASD, or network?
- Is the goal to understand total system performance, or the performance of a single component?
- How will users or transactions be simulated? What kinds of tuning actions are considered reasonable by DBAs and system programmers?
- Does inside knowledge of component performance behavior allow unnatural optimizations?
- How do you quantify the ability of a system to manage its resources in a dynamic environment?
- How can you fairly compare a small system running against a small database against a large system running against a large database?
- How repeatable is the benchmark?
- Can the benchmark be implemented without excessive expenditure of time and resources?
- Will the metric allow for meaningful comparisons between diverse systems?

Compromises to address these issues and many others not mentioned are unavoidable.

Industry Standard benchmarks are defined with the intention of enabling customers to compare the offerings from multiple vendors without the expense of running benchmarks of their own.

In addition, server vendors commonly use such benchmarks as a means of promoting the performance of their offerings versus their competition. For example, the TPC-C benchmark is commonly used by UNIX and NT vendors when making their marketing claims. However, the TPC-C benchmark uses an idealized workload:

- There are only OLTP transactions, one bounded query and no batch.
- The workload utilizes resources at a constant “steady state” rate.
- The I/O is well-behaved and fairly uniform in size.

- Only 1% of the data is skewed so that there is 99% affinity in data, users and work.

Because of these factors there are no bottlenecks or hot spots caused by “user skew,” “workload skew” or “data skew.” Such factors are the norm in actual commercial systems.

3.1 Other Factors Affecting Performance

In practice there are many other factors that are not even considered in Industry Standard benchmarks, let alone measured.

3.1.1 Benchmarks versus Production Workloads

Benchmarks have the following characteristics:

- They are highly tuned.
- Vendor hardware and/or software may be optimized for the benchmark.
- Transaction arrival rates tend to be regular.
- I/O requests tend to be evenly distributed across paths and drives.
- Each transaction usually only has access to a small subset of the database.
- They are oriented around a single application; multiple workloads are usually not included.
- They run at high processor utilizations.
- They optimize I/O activity by having more disk devices than would be found in production.

In contrast, production workloads usually have the following characteristics:

- They are harder to tune unless the operating system has good workload management tools.
- Vendor hardware/software optimization for benchmarks usually offers no benefit.
- Transaction arrival rates tend to be highly skewed.
- I/O requests tend to be highly skewed.
- Transaction database accesses are often scattered across large parts of the database.
- They have multiple concurrent applications. If these are on separate servers, then there will be higher operating costs.
- They run at lower levels of utilizations (particularly for UNIX and NT servers).

Other important items not measured in benchmarks include:

- Operational issues
  - Effectiveness of the automation of the operator interface
  - Speed and reliability of backup and recovery
  - Ability to test readiness and effectiveness of disaster recovery procedures
- Ability of the server hardware and software to survive unplanned outages
- Ability of the platform to accommodate concurrent change, and so avoid planned outages
- Ability of the platform to manage mixed, concurrent applications
- Security and integrity of data

Also, vendors normally publish benchmark results which are in most cases *single* points that do not reveal how their SMP systems scale with increasing numbers of engines. Thus it is rare to find a TPC-C result for more than one SMP system in a single server range[4].

The following are some areas where Industry Standard benchmarks can produce results inconsistent with what is observed in practice.

3.1.2 Resource Affinity

In most TPC-C benchmarks, the vendor, legitimately, will use system parameters to cause the workload (database and transactions) to be partitioned such that transactions to a given portion of the database will always run on the same engine on the SMP. This means that cache contents, for example, are not polluted and large portions of the database can remain in fixed locations in memory.

These affinities result from the use of large private cache memories by UNIX servers. Because of the lower investment by suppliers in developing internal bandwidth, UNIX servers suffer more performance loss when the working set grows due to large numbers of users or when multiple applications cause cache misses.

To make matters worse, UNIX server caches tend to be deep and narrow[5] . While UNIX servers have big caches, data from each memory location has only one place to reside in the cache. Thus, if the CPU needs two pieces of information that use the same cache slot, the data must be moved to and from memory even if the rest of the cache is not being used.

Deep narrow cache arrays can be made from off-the-shelf parts. The resulting caches have good average performance but are more subject to “thrashing” (excessive non-productive rates of movement between memory and the cache), especially when nonsequential memory reference patterns occur. Having more than one program contending for cache space, or having many users on the machine, or an object broker making scattered references to objects in memory, are all examples of situations that can cause thrashing[6].

UNIX servers are therefore forced to choose between preserving cache contents and losing capacity to skew, or using any available processor and losing capacity to cache thrashing. All of these factors compound, creating pathological situations where UNIX server performance does not live up to benchmark results.

In addition using processor affinity, so that each process runs on an assigned engine, significantly reduces the ability of an SMP to survive the failure of a single engine.

---

4 There is a good reason for not providing all data: these measurements are quite expensive, because the systems have to be available and set up, and then the benchmark has to be tuned, which takes considerable time. In benchmarks it is common practice to use database hashing to improve database access times. This requires that the database be re-built and loaded for each SMP size to get the optimal hashing distribution. This requires significant effort, and would be major operational constraint if used in a production environment.

5 For an example, see pages 221 to 235 in *Configuration and Capacity Planning for Solaris Servers*, Brian L. Wong, Prentice Hall, February 1997, ISBN 0-13-349952-9.

6 See also, *Performance Characteristics of a Quad Pentium ProSMP Using OLTP Workloads*, Keeton, Patterson, He, Raphael, and Baker, IEEE 1998.

Providing multiple places for each memory location to map to within a cache uses special hardware, called Set Associative cache design, as is used in the IBM S/390. This design requires greater design effort and higher cost to manufacture. However, the benefit is that resource affinities do not exist, nor are they required in order to achieve high levels of performance in actual commercial processing.

Disk-to-channel and channels-to-processor affinities are also not present in the S/390 hardware or OS/390. These affinities result in queuing penalties when contention or "skew" occurs. The idealized workload found in TPC-C minimizes the skew, eliminating these penalties.

S/390 allows alternate pathing and dynamic I/O scheduling, which result in superior commercial I/O processing, but which also add overhead compared to the idealized benchmark.

3.1.3 Workload Complexity

Fragmentation is the division of the resources among a large number of users. A large number of users are more likely to be using a wide range of resources (processors, cache, memory, I/O paths, devices, and so on) than a small number of users. Also the mix of resources used is likely to vary dynamically, so that any given resource is extremely unlikely to be dedicated to a small number of users. A workload with 1,000 concurrent active users is more fragmented than a workload with 100 concurrent active users run on the same machine.

Fragmentation leads to reduced cache hit ratios and more dependence on high internal bandwidth. As the number of users increases, the speed of data movement becomes as important as processor power. That is, large user counts favor machines with high internal bandwidth. The IBM S/390 is still much faster than even the latest UNIX machines in this area.

For example, four L2 caches share a single 950 MB/sec bus in the SUN UE10000; up to 32 (but typically 16 to 20) L2 buses share a single 2.6 GB/sec bus in the SUN UE6000; whereas three L2 caches share two 2.6 GB/sec buses in the current S/390. This is because of the investment made in technology. The MCM packaging and power/cooling of S/390 results in more, wider connections between chips and shorter bus lengths. As a result, memory latency can be about half that of large UNIX servers, and the aggregate cross-sectional bandwidth of the L2 Cache/Memory in the latest S/390 Enterprise Servers can be an order of magnitude higher[7].

3.1.4 Contentiousness

Contentiousness results from competition for resources. S/390s are most effective at handling contentious, fragmented workloads, with large working sets, that tend towards being I/O-intensive. UNIX servers are best at parallel, homogeneous workloads, which tend towards CPU intensity and small working sets. By way of example, a "light" OLTP application with less than 500 users and partitionable data (minimum skew) looks very good on UNIX. As the number of users goes over 750, the S/390 will look better. If the workload moves toward a larger working set such as in Business Intelligence, the S/390 would also

[7] For an example, see pages 429 to 444 in *Shared-cache clusters in a system with a fully shared memory*, P.Mak et al., IBM Journal of Research and Development, Volume 41 Number 4/5.

improve in standing. When multiple query streams of ad hoc queries are combined with OLTP, the S/390 will look extremely good.

3.2 Interpreting Results

A strength of the standard benchmark approach is the ability to compare the performance of unlike databases, operating systems and hardware architectures.

For example, the TPC-C benchmark or the SAP R/3 SD benchmarks allow performance comparisons to be made between the AS/400 and other operating systems, such as UNIX or Windows NT.

Where the hardware has remained similar and the database differs, some relative performance insight can be gained into the differences between database products.

Such benchmark results can provide a very informative insight into the abilities of the various hardware architectures and operating systems to perform in a number of application areas.

However, because of the many variables involved in individual configurations, we believe that one should use the results for a high-level perspective, rather than as a definitive statement of relative performance between models.

The price/performance metric is an area open to fine tuning by test sponsors. Although a benchmark may be aimed at costing a “realistic” level of support (for example, same day on-site support), there are hardware cases where large disk subsystems are costed with customer diagnosis and replacement of the faulty drive, with the spare drive coming from a 10% spares inventory.

One company may have a higher price/performance, but offer full support. The cheaper company′s price/performance may be a result of that organization costing the customer parts replacement option.

If there is a need to accurately compare the price aspects of results, we strongly recommend that a detailed analysis of the Executive Summary or Full Disclosure Reports be made. Some general issues are server performance, configuration pricing, and configuration complexity.

3.2.1.1 Server Performance

UNIX architectures process terminal I/O one character at a time, which means that each time a key is depressed, a task dispatch interrupting the processor occurs. This can be very expensive. By using front-end processors to handle terminal I/O, the cycles on the server which would normally be taken up handling these tasks are freed up to do other work.

This can result in performance increases on the server ranging from 10-50%! The higher percentages are more likely to be seen with TPC-C due to the heavier terminal I/O load. In effect, the server has become a database call machine with only the operating system and DBMS running on the system. In fact, all the current TPC-C results are client/server measurements.

3.2.1.2 Configuration Pricing

Front-end processors allow the use of very low-priced, low-function terminals: terminal costs range from 30% to 50% of the total price used to calculate price performance. Vendors were motivated to use the cheapest terminal on the market to deflate the cost of the configuration.

Version 3 of the TPC-C specification discontinued the pricing of terminals to address this problem. It is important when comparing TPC-C results to make note of the version. It is not permissible for vendors to compare TPC-C results between V2 and V3.

Front-end processors also allow the use of lower prices on software. TPC specifications require the configuration of development and database software for price/performance calculations. Vendors implementing front-end processors typically price the development software on one of the small front-end processors to avoid the higher license charge on the server processor (which is typically a larger processor requiring a high licence price). If a transaction monitor is also running on the front-end processors, a similar method would also reduce the product price.

DBMS charges can also be reduced due to the method that typical UNIX-related relational database vendors charge for licenses: most charge by number of users. Since the users are logged on to the front-end processors, and not to the processor where the database is running, the license charge can be significantly less in a TPC-C front-end processor configuration (as compared to a real customer environment).

For example, in an early TPC-C result, a system reporting 700+ tpmC would typically require about 600 users. This would necessitate an unlimited user license for most database packages, which could easily cost over $100,000. However, in a front-end configuration, the number of server tasks on the base system would typically be less than 100 (the 600 users logged onto the front-end would not be considered database users), therefore the license charge might be for 128 users (for example, $70,000).

3.2.1.3 Configuration Complexity

Front-end processors can improve the throughput and performance in many customer environments as well as in Industry Standard benchmarks. While the costs of the front-end processors are included in the five-year cost of ownership calculation, the hidden disadvantage may be the cost of implementing and operating such a complex environment.

The TPC price/performance metrics do not include the cost of implementing a complex front-end environment.

Let us show a few examples based on some TPC-A and early TPC-C results:

1. DEC: 4 Node 7000 Model 650 (5 CPUs each) AXP VMScluster c/s rated 3,692.02 tpsA, $4,873/tpsA

   In addition to the 20 back-end CPUs, this measurement had 44 front-end CPUs. Each front-end MicroVAX 3100 Model 90 was rated at 86 tpsA. Thus, the front-ends alone had a combined aggregate capacity of 3,784 tpsA, which was *more* than the total tpsA reported for the full clustered configuration! The throughput of this measurement was slightly better than the 3,504.93

tpsA[8] that IBM achieved with TPF on a single ES/9000 511 air-cooled processor.

2. Tandem: Himalaya K10000-16 c/s, rated 3043.49 tpmC $1,598/tpmC

   This measurement used 32 front-end Intel 486 processors for 16 back-end CPUs.

3. Hewlett-Packard: HP9000 T500 6 way c/s, –2145.83 tpmC, – $972/tpmC

   This measurement used 2 HP9000 Series 800 Model E35 front-ends, each rated at 401 tpmC capacity.

Thus, by using “front-end back-end” configurations, vendors have at best, made it very difficult to compare results across systems and at worst, provided misleading representations of their performance capabilities, their real configuration costs, and the difficulty of configuring and managing the computing environment, all the while suggesting that they are demonstrating their client/server capabilities and superior server price/performance.

In fact, TPC-C can now really only be used to compare the back-end processor as a database server (since one can assume that all the client code is executed on the front-end processors).

It should also be pointed out that many of these so called “client server” configurations are ad hoc configurations which do not have a unified set of application development, system management and administrative tools and facilities.[9]

In addition to these points, it should be emphasized that these front-ends are controlling a large number of clients.

As an example, Compaq published TPC-C results for a ProLiant 7000-6/400-M with 4 Pentium II Xeons running at 400 Mhz. Three front-ends were used, each equipped with 2 Pentium II processors running at 400 Mhz. These are the entry points for the 14750 users which this system should service. In other words, each of the three PCs is the port of entry for 4920 end-user PCs. Most probably such a 1 : 4920 ratio would not be found in a real production environment, where other requirements will lead to a lower ratio.

3.3 Shared versus Non-Shared Architectures

One of the shortcomings of the TPC-C benchmark is the fact that it produces only a single measurement point for an entire system configuration. This opens the door for super-optimized configurations. Because the vendors have prior knowledge of the transaction arrival patterns and database skew, they can optimally partition and place the data closest to the processor that is most likely to access it.

---

[8] $8,310/tpsA

[9] The TPC originally required that systems using front-end back-end configurations be labelled as “c/s” and it referred to them as “client server” configurations. Because of the previously mentioned concerns, these configurations are now officially called “front-end back-end.” The TPC points out that front-end back-end configurations are not comparable to “standalone” configurations, but since they are listed together in the TPC results summary and there are no longer any standalone results, this distinction is usually forgotten.

In TPC-C, there is affinity between a terminal and a warehouse. Only 15% of the payment transactions must access a remote warehouse and 10% of the items ordered by the New Order transaction must be supplied by a remote warehouse. The remainder of the database accesses are effectively random to a given warehouse for a given customer terminal.

Thus TPC-C is highly partitionable. The amount of cross-node message traffic, lock contention and resultant overhead is very limited and generally much less than would be experienced in a more typical production environment where transaction arrival patterns and database accesses are much less predictable and stable.

In addition, the larger number of unique transaction types and the variance in transaction behavior and complexity found in customer shops make it much more difficult to determine the optimal partitioning scheme.

Without debating the pros and cons of non-shared architectures versus shared architectures, it is clear that a single point, highly partitionable benchmark such as TPC-C provides a clear advantage to non-shared architectures.

While non-shared TPC-C implementations avoid demonstrating the real costs of their cross-node communications, the shared architectures must assume the overhead associated with data sharing without being given the opportunity to demonstrate the value of their shared paradigm. Both the cost in pathlength and configuration are incurred with no way to demonstrate added value.

What would happen if there was no affinity between terminals and warehouses or branches, if there were floating and unpredictable hot spots in the arrival patterns and database accesses, if there were more transactions accessing information from across all of the warehouses or branches? What if skews were varied for multiple measurement points without allowing for any changes in system tuning parameters? Would the comparative measurement results between vendors change if some of these variations were benchmarked?

A benchmark should measure how well an operating system, database, transaction monitor, or any other type of load balancer spreads out the work so that all processors are used as efficiently as possible, that is, how well workload distribution can be performed by the system. Unfortunately, in the case of TPC-C, the benchmark definition itself does much of the load balancing for clustered systems. Very little communication passes between nodes, because the vast majority of the processing is local.

In addition to taking advantage of the ease of partitionability, other vendors have taken advantage of the limited number of TPC-C transactions by assigning CPU affinity to individual transactions. In a sense, not only have they partitioned the data, they have also partitioned the incoming transactions. This is only possible in the very limited environment found in a benchmark.

Most vendors are using hashed database access for their benchmarks. This is very efficient and randomizes the accesses. However for large databases that are growing rapidly, this has many operational drawbacks including the need for frequent re-build and load in order to maintain peak performance.

The advantage given non-shared architectures by the TPC-C benchmark is so strong that the results of SMP and other shared implementations should not even be compared to clustered non-shared implementations with this workload.

In addition, real-world performance for these non-shared implementations will most likely be worse when dealing with real-world transaction and data distribution.

3.4 Summary

Based upon the discussions in this chapter and in the remainder of this book, we summarize the strengths and weaknesses of Industry Standard benchmarks.

3.4.1 Strengths of Industry Standard Benchmarks

Industry Standard benchmarks have the following strengths:

- They are standard tests.
- They can compare different architectures.
- They can compare different database management systems and versions (if sufficient data points exist).
- Tuning by vendors can flow through to useful performance for customers (TPC-D reporting, for example).
- Multiple types of workloads can be used to compare to customer applications:
  - OLTP
  - ERP (SAP R/3, Baan, Peoplesoft)
  - Web serving
  - Business Intelligence
  - Technical computing
- At the system level, they can reveal what is needed for an application in production.

No benchmark is perfect. All a benchmark measures is how fast the benchmark runs, not how fast the application will run.

3.4.2 Weaknesses of Industry Standard Benchmarks

Industry Standard benchmarks have the following weaknesses:

- It is difficult to compare the following:
  - Different database management systems.
  - Different versions of DBMSs.
- High utilization in benchmarks not generally achievable in production with UNIX and NT, which makes comparison to systems that can achieve high consistent utilizations difficult.
- The extent of tuning in benchmarks is not generally done by, or economically practical for, most customers.
- Benchmarks use single workloads, when customers are increasingly requiring multiple concurrent workloads.
- Benchmarks use no, (or limited) batch workloads.
- Scalability with the number of engines is not usually visible due to:
  - The cost to the vendor of multiple benchmarks (they use the largest server to gain maximum marketing impact).

  - Even if there are multiple data points, each one is a tuned measurement and does not show actual scalability.
- They run at the system level, and not at the processor level.
- They do not test security capabilities.
- They do not test data integrity functions.
- They do not test backup/recovery capabilities.
- They do not show operational capabilities such as:
  - Ease of operations.
  - Resilience to failure situations.
  - Mixed workloads management according to business priorities.
  - Online database reorganization capabilities.

Many of these customer requirements are discussed in Chapter 5, “Strengths of IBM′s S/390 and SP” on page 69.

Chapter 4. Comparing the Performance of Different Platforms

If we wish to compare the performance of two platforms with significantly different architectures, such as UNIX and S/390, we also need to understand the differing approaches to computing that are inherent in the two platforms.

Both UNIX and S/390 have long heritages, but each has evolved with vastly different mentalities among its users.

UNIX Mentality: The rapid spread of UNIX computing was fueled by the needs of research and educational users for inexpensive computing, and freedom from the disciplines and contention for support of the data centers of the 1970s and 1980s.

The attitudes of UNIX users tend to be that:

- "Hardware is cheap"

  The consequences of this attitude are:

  - Most UNIX servers run at low average processor utilizations
  - There has been little emphasis on performance monitoring or tuning
  - Large memories have been used to overcome low-performing commodity-based I/O devices
  - Most UNIX systems have only a single application per server
  - There has been a tendency not to be concerned with SMP ratios (extra capacity can be gained by buying a new, larger server)
  - With only one application per server, there has been little emphasis on workload management

  Many of the practices above have been reinforced by a belief that:

- "Distributed Client/Server is the direction in computing"

  This architectural approach leads to:

  - A single application per server
  - Little emphasis on workload management
  - Multiple servers
  - Separation of function within applications

S/390 Mentality: S/390, on the other hand, because of its widespread usage in commercial computing, has evolved with a completely different philosophy.

Early users of S/390 (and its predecessors, S/360 and S/370) demanded the maximum utilization of the then very expensive resource in order to run a wide variety of applications. High processor utilization demanded the ability to efficiently run multiple concurrent applications, which in turn resulted in the highly efficient I/O and memory utilization.

The attitudes of S/390 users can be summarized as follows (even though the costs of mainframe computing have been shown to be comparable to those of UNIX and NT, for example see *Selecting a Server - The Value of S/390*, SG24-4812):

- "Hardware is expensive"

  - S/390 servers typically operate at high processor utilizations
  - There is a strong emphasis on performance monitoring and tuning

- Large memories are used to reduce I/O and raise processor utilization
- Most S/390 servers run with multiple concurrent applications
- S/390 scales well for both SMPs and clusters
- There is a major emphasis on both SMP and cluster workload management

As the costs of hardware and software have plummeted, people costs have continued to rise. Minimization of operational and support costs has been a focus area for S/390 for many years:

- "People are expensive"
  - S/390 users have recognized the benefits of consolidating both the number of data centers and servers for many years
  - Automation of operations and workload management has been a trend for over a decade
  - Multiple applications per server in combination with automation and a reduction in the number of software components also reduces people costs.

In this chapter, we discuss some of the factors that impact the performance of a computing system, and we look at the three attributes that are necessary for a server to truly meet the needs of today′s commercial applications.

4.1.1 Differences between UNIX Servers and S/390

In spite of the various marketing claims by the many vendors in today′s marketplace, it is fair to say that similar levels of technology are used by all manufacturers for their server offerings.

The base performance of the processor chips of any given generation are approximately the same no matter whether they are for IBM S/390, AS/400, RS/6000, or Netfinity, or for UNIX vendors such as Sun or HP.

The ability of a processor chip to perform raw calculations also does not appear to depend significantly on whether the instruction set is complex (CISC) or reduced (RISC). The capacity to perform useful commercial work depends on other architectural and design choices and on the nature of the workload itself.

Traditionally, UNIX servers have used commodity packaging and support chips to keep costs low and run operating systems that are tuned to a specific task. As a result they have less internal bandwidth and trade off less of the CPU power for workload management, Reliability, Availability and Serviceability (RAS), integrity and so on.

This means that they can efficiently deliver high instruction rates as long as the workload does not overly stress either the movement of data around the system (for example, either a large working set or a large number of users), or does not have a need for complex workload management (for example, mixed workloads or high utilization).

IBM S/390 servers have used high technology packaging and more complex support chips and have traded off CPU power in the form of pathlength to provide high internal bandwidth, workload management, RAS, integrity, and so on. This means that they can safely and securely deliver service to many users running mixed workloads with acceptable and consistent response times (see also Figure 6 on page 45).

The value of a mixed versus single application environment is customer-dependent and determined by:

- Application
  - Portability
  - User count
  - Working set size
  - Scalability
  - CPU intensity
- Data flow
  - Batch windows
  - Networking in the transaction paths
- Customer background environment
  - Skills
  - Existing systems
  - Beliefs
  - Operational disciplines
- The value of service availability to the enterprise and the cost of outages

4.2 General Performance Issues

There are three major attributes that define how well any system will perform any given workload: processor engine speed, system bandwidth, and workload management.

This is shown diagrammatically in Figure 6.

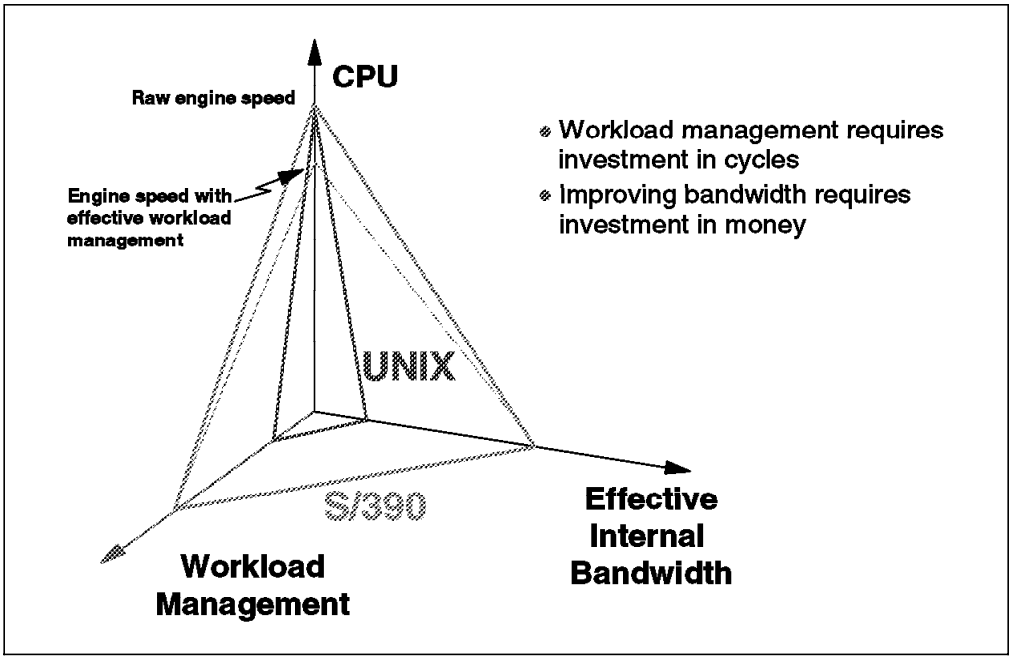


Figure 6. The Three Axes of Performance

Note that the latest IBM 9672 Server (the Generation 6 (G6)) has a cycle time of 637 MHz - faster than almost all UNIX processors. S/390 trades off some of this

raw speed in order to do efficient workload management in the operating system (OS/390).

Increasing the bandwidth of a server requires a large financial investment, which IBM has made in the S/390 over a 30-year period. UNIX vendors still have to make this investment and are working on new I/O architectures (similar to the channel systems of S/360 from the 1960s), I/O devices, and are using very large memories in order to avoid doing I/O.

Obviously, not all attributes are required for the good performance of any given workload. For example, a numerically intensive calculation running as a stand-alone application will benefit little from a high system bandwidth or a workload manager designed for multiple, concurrent applications.

In the past, most commercial UNIX systems have only run one application per server, and have avoided I/O by caching large amounts of data in system memory. Processor power only is needed for such systems and few UNIX systems have either high internal bandwidth or good workload management.

However, the rapidly growing size of single SMP systems, combined with the business need for the inter-working of both existing and new applications, means that the well-balanced system shown in Figure 6 on page 45 is becoming mandatory.

How well do Industry Standard benchmarks measure these attributes?

- Processor speed is measured by most benchmarks, particularly *kernel*-type benchmarks, such as the SPECint benchmarks.
- Bandwidth is not usually measured well by benchmarks, particularly when large in-memory caches are used. I/O throughput is seldom a limiting factor in Industry Standard benchmarks, which are designed to be processor-limited.
- Workload management is not measured by any single-workload benchmark. S/390 trades off some processor power in order to run the workload management software, so that a well-balanced system results.

Thus the overall architecture of a system is extremely important to the achievement of high performance. Some architectural issues are presented in Chapter 5, “Strengths of IBM′s S/390 and SP” on page 69 and Appendix C, “IBM SP MPP Architectures” on page 105.

The following sections discuss some areas which are often raised in performance discussions.

4.2.1 Cycle Time Comparisons

Cycle time in MHz (millions of cycles per second) is often quoted as a measure of chip speed. However, such a raw speed rating does not recognize either how many cycles are needed to perform a processor instruction, or how many instructions are needed to perform a useful computing function (see 4.2.2, “Which Instruction Set is Best” on page 47).

Figure 7 on page 47 shows the relationship between instruction rate (in millions of instruction/second) and MHz for three older IBM server products. Each has a widely different MHz rating but, as we see in 4.2.2, “Which Instruction Set is Best” on page 47, all offer comparable commercial throughput.

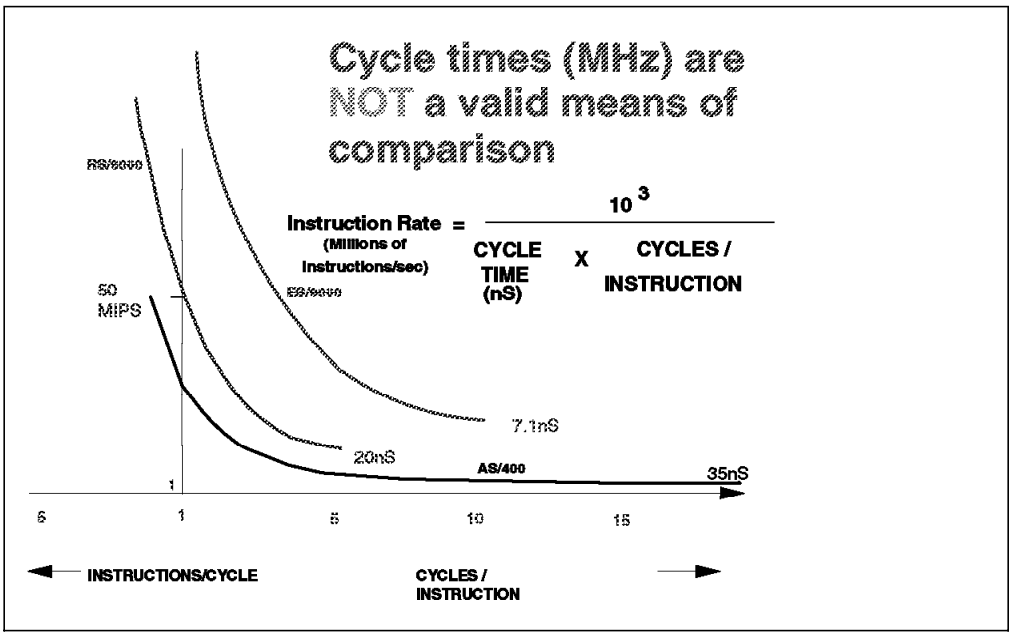


Figure 7. Cycle Times versus MIPS

4.2.2 Which Instruction Set is Best

Over the past three decades, so-called mainframe computers evolved to have a large, functionally rich instruction set called Complex Instruction Set Computers (CISC). The continued development of such systems is expensive both in people time and hardware, so in the 1980s an alternative approach arose. Reduced Instruction Set Computers (RISC) use fewer, less complicated instructions (requiring less hardware) and then use combinations of these basic instructions to achieve the effect of a smaller number of CISC instructions.

- RISC has a simpler instruction set with a fast execution rate.
- CISC has more complex instructions with longer execution times, but more data handling can be achieved in a single instruction.

While there has been much heated discussion concerning the relative merits of CISC and RISC, two facts emerge:

1. Measurements of commercial workloads show that the difference in throughput between the two architectures is minimal. Differences are typically no more than 10% to 20% in both directions.
2. As new applications evolve, RISC instruction sets are increasing in size and complexity, further reducing the differences.

The largest factor in making a difference between RISC and CISC performance is the optimization capability of the compiler used for developing the application code.

Because of the simplicity and the granularity of the RISC instruction set, optimization of machine code at compile time is more easily attainable by software algorithms, as opposed to the CISC instruction set, where several functions are bundled inside monolithic instructions.

On the other hand, CISC compilers have been optimized over a significant period of time and generally produce highly streamlined code. The net result is the small performance differences we mentioned earlier.

Figure 8 on page 48 illustrates these points. On the basis of cycle time alone, the RS/6000 server would appear to outperform both the AS/400 and S/390 servers by wide margin (and may in fact do so for numerically intensive computing).

However, when we look at more commercially-oriented comparisons the difference shrinks to half in the case of TPC-A (well known to be I/O deficient), and to trivial differences in the case of RAMP-C.

This example supports our contention that, for systems built from the same generations of CMOS chips, there will be very little difference in commercial throughput for single engine servers.

Note the last caveat, as additional engines are added to an SMP, the increase in throughput may vary substantially from server range to server range, and with the type of workload being run on it.

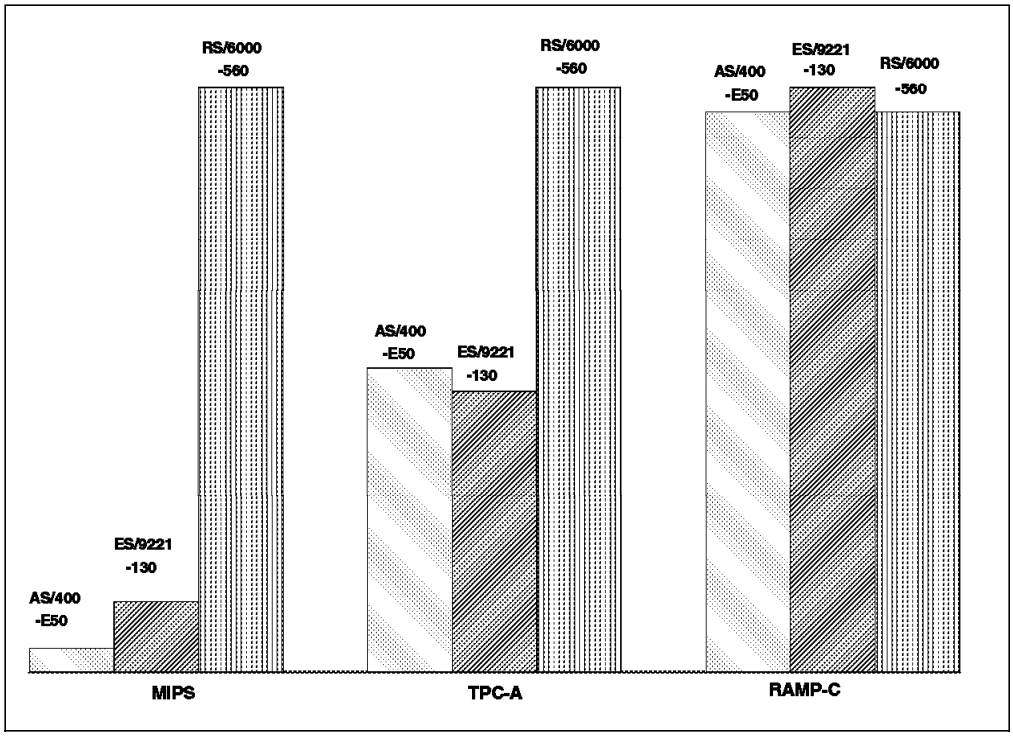


Figure 8. Architecture versus Performance

4.2.3 The Quest for Internal Bandwidth

As of today, high performance objectives are driving the cost of the hardware implementation of a system. This cost is directly related to the packaging material and technology used in the system.

The evolution we went through for the S/390 systems over the past ten years shows the progression from bipolar systems, with large numbers of semiconductors, power and cooling physical devices, towards CMOS technology, where the design and implementation costs are now focusing almost exclusively on chip technology.

One of the major contributors to a high performance level is the system internal bandwidth, which is a measure of how fast data can move from the memory

(memory being the conceptual memory as defined in the Von Neumann model) and the processing unit(s). The quest for bandwidth has driven the way the conceptual memory has been logically and physically implemented using such techniques as caching or interleaving.

When it comes to comparing SMP systems, one of the most important aspects of memory implementation, with respect to the internal bandwidth, is the L2 cache.

- In UNIX systems, the L2 cache tends to be large and private to each processor, implying additional transfer of data between the L2 caches when this data has to be shared among processors.

  These caches are also directly mapped, or uni-dimensional, in that once a position is occupied in a memory set in an address congruence class if the same position is required for another memory set, then the already residing data has to be cast out and replaced by the new data. This should be compared with the set associative technique used in the S/390 L2 implementation.

- In S/390, the L2 cache tends to be smaller than in the UNIX systems, but it is implemented as shared among processors, with some variations in the way this sharing capability is obtained, depending mainly on the S/390 machine generation.

  It is also implemented as “set associative” or “bi-dimensional” in that several memory sets for the same congruence class are held into the L2 cache. This allows the retention in the cache of data belonging to the same position in the same congruence class for different memory sets. The sets are read in parallel and the relevant output is selected as per the examination of the associative array describing the cache contents.

  The net result is that S/390 cache management is generally far more effective than in UNIX servers, and a given cache size can buffer a larger amount of data and instructions than a UNIX cache.

  Obviously this approach is more costly than the UNIX system′s approach, but it is required to match the fragmented workload, and high number of users, found in modern commercial mixed-workload systems.

S/390 internal bandwidth can be an order of magnitude higher than in UNIX servers as follows:

- S/390 9672 - 2 GB/sec/engine
- Typical UNIX servers 200 - 450 MB/sec/engine

4.2.4 Workload Management

UNIX and OS/390 manage workload, and hence throughput, in different ways:

1. UNIX manages the priority of tasks waiting in queues. However, once a task is dispatched, it runs for a fixed allocated time slice. Whereas OS/390, in addition to managing queues according to address space priority, can also modify the amount of dispatching time allocated to a task in order to balance service units consumption against performance objectives.

2. Resource management software has usually been an add-on product (although this is beginning to change with the availability of integrated offerings from HP and Sun, among others).

OS/390 has the integrated System Resource Manager (SRM), which is continuously in operation (therefore contributing to what appears to be a higher overhead for OS/390 in a benchmark environment).

3. There are application-level schedulers available for UNIX. These do not offer the capabilities of the OS/390 Workload Manager (WLM), which offers a variable span of control over the system depending whether it is running in Compatibility Mode or Goal Mode.

   As for SRM, the WLM function may appear as overhead at low system utilization

4. Managing the UNIX system resources in order to maximize throughput typically is done by developing a set of resource affinities, such as restricting a component of the workload (particular transaction types, for example) to a particular processor. The effect in this example is of increasing the probability of cache hits for this processor.

   S/390 has been designed from the very beginning for the sharing of all resources.

4.2.5 Response Time-to-System Utilization Considerations

A major difference in behavior between UNIX and OS/390 systems is shown in Figure 9 and Figure 10 on page 51.

For single tuned workloads, such as are found in benchmarks, the variation in response time with increasing processor utilization is very similar, with both systems being able to run at high utilization levels with acceptable response times.

At low utilizations, UNIX systems will have slightly lower response times because OS/390 has code to handle mixed workloads, error recovery and data protection functions, and functionality that are not used by a single benchmark workload.

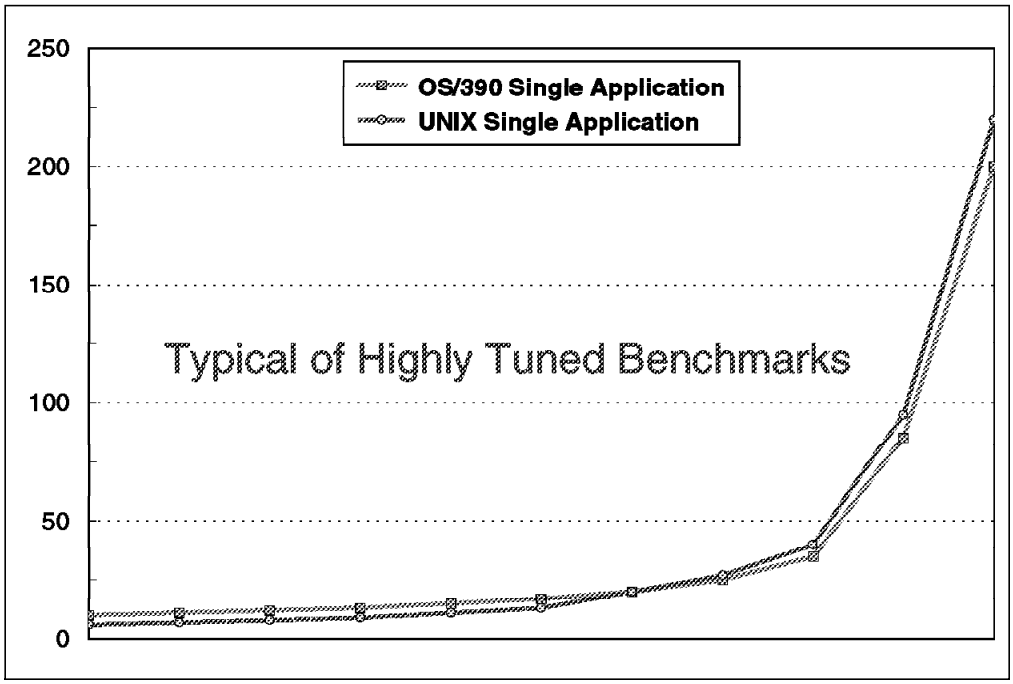


Figure 9. Performance Curves for UNIX and OS/390 for a Single Workload

However, when mixed workloads are run on the same system, or there are large numbers of users, or when transaction arrival rates or data access are skewed, the profiles are quite different.

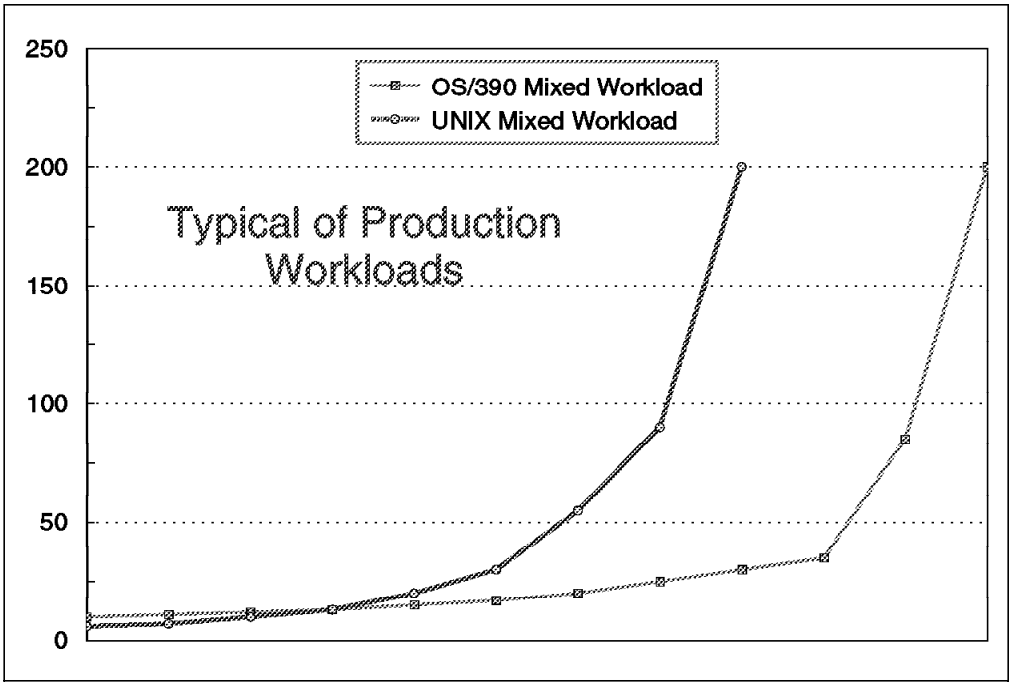


Figure 10. Performance Curves for UNIX and OS/390 for a Mixed Workload

This is the real environment for commercial production systems, not the highly tuned world of benchmarks.

OS/390 can handle such mixed workloads up to very high utilizations with acceptable response times, whereas UNIX systems start to give unacceptable response times at much lower utilizations. Many users report UNIX systems actually failing at utilizations around 60%.

UNIX customers typically run a single application per system at an average utilization over the shift of 20% to 30%. This allows for peaks up to 50% to 70%[10] (see, for example, the workload profiles in 4.3, “Comparing the Performance of S/390 and UNIX Servers” on page 57). The reasons for the differing behavior can be found in the differing design assumptions of OS/390 and UNIX.

As can be seen in Figure 6 on page 45, OS/390 evenly balances the use of the CPU (processor), processor bandwidth and workload management algorithms.

UNIX systems, on the other hand, have traditionally used the processor as a cheap resource and have not developed sophisticated designs for processor internal bandwidth or UNIX management of mixed workloads.

An example of one customer′s experience was presented at an IBM customer conference in 1997. The customer did the following:

[10] Note that this range is for a well managed UNIX OLTP workload. There are many examples of higher or lower utilizations. Some installations may choose to suffer poor resposne times in the quest for higher utilizations.

1. A large UNIX SMP and a S/390 configuration both ran an OLTP workload. Additional engines were added until there was no further increase in performance (with the workload unchanged).

   These peak configurations were:

   - 32 engines for the UNIX SMP configuration
   - 20 engines for the S/390-OS/390 Parallel Sysplex configuration

2. The two configurations were then fixed at:

   - 32 processors for Sun/Solaris, with a typical response time of 3.2 seconds.
   - 20 processors for S/390, with a typical response time of 4.5 seconds.

3. The workload was then increased by a factor of three. The following was observed:

   The two response time curves crossed when the workload was twice its original size. That is, the UNIX SMP response time increased faster than the OS/390 response time.

   At three times the workload, the UNIX SMP response time increased from an initial 3.2 seconds to about 8 seconds, while the S/390 response time increased from the initial 4.5 seconds up to about 5 seconds.

4.2.6 SMPs and Scalability

While there is some presence in the market place of Non-Uniform Memory Access (NUMA)-based servers, the majority of commercial computing today is based upon Symmetric Multiprocessors (SMPs).

What characterizes SMPs is the symmetrical access of each processor engine to memory (equal access time) and all I/O ports (see Figure 11 on page 53).

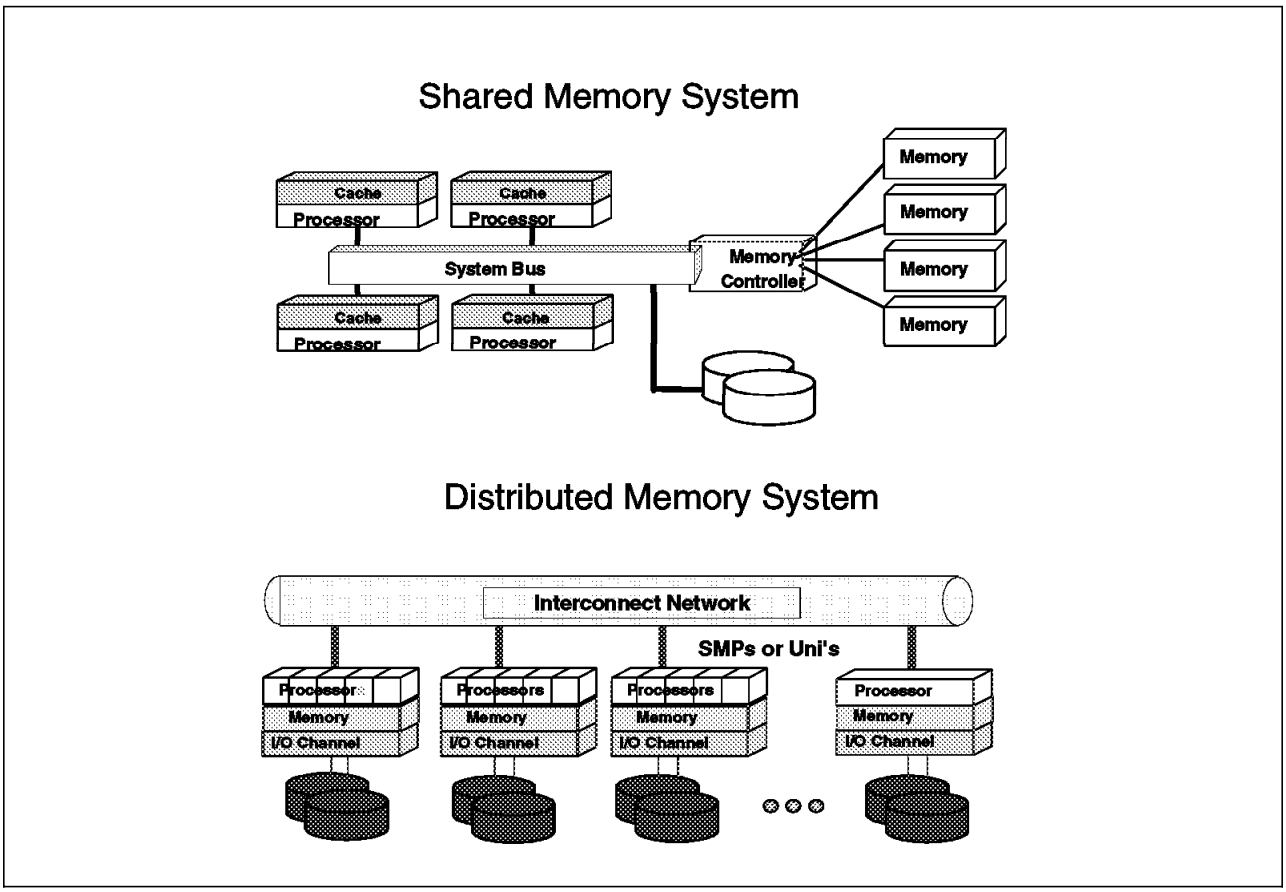


Figure 11. Shared Memory vs Distributed Memory

Ideally one would like to get one engine's worth of additional capacity or performance from the server for each engine added to the SMP.

However, contention for memory access from ever-faster CPUs and disks, plus operating system inefficiencies, limit SMP scaling, so that we observe an ever-decreasing gain per engine for higher numbers of engines.

Some of the sources of this interference are:

- Hardware
  - The need to maintain data integrity between private caches requires additional cycles and/or hardware.
  - Interrupts for the completion of I/O, or task switching caused by the random arrival of higher priority work can cause cache contents to be replaced by access to slower main memory or disk.
- Software
  - The ability (or lack thereof) of the operating system to schedule units of work (tasks, processes, threads) efficiently across engines
  - The ability (or lack thereof) of the operating system to process I/O requests without causing cache cast outs. For example, most UNIX servers still drive I/O with requests directly from memory. S/390 passes this role to channels which operate independently of the processor.

The shape of the SMP curve can be modelled with a simple equation (see Figure 12 on page 54) which assumes a fixed degradation between each pair of server engines. The degradation (also known as the multi-processor or MP effect) can be subdivided into hardware and software portions).

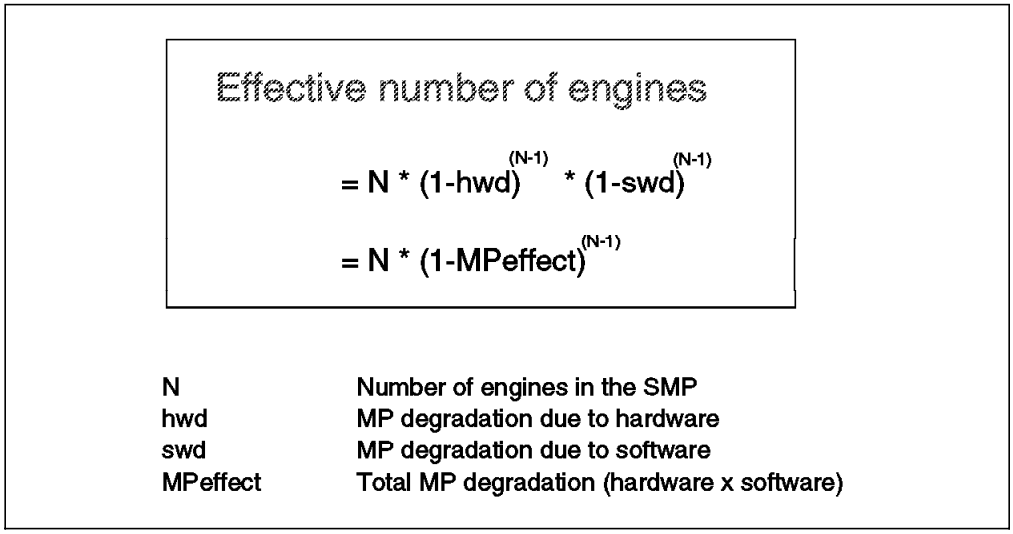


Figure 12. SMP Model

Thus, if one can estimate the MP effect (for example from the performance of a two-way server versus a single engine server, or by curve-fitting measurements of a server range with differing numbers of engines), the performance of a server can be estimated.

Figure 13 on page 55 shows the shapes of the SMP curves for various degrees of degradation.

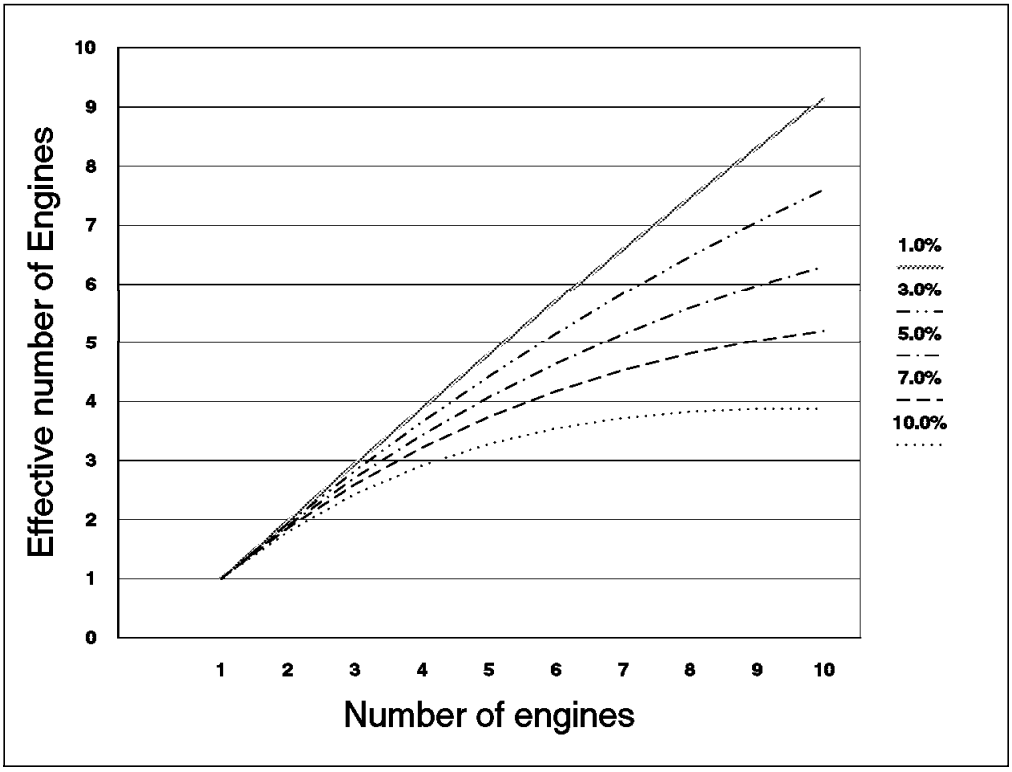


Figure 13. SMP Curves for Different Degrees of Degradation

As can be seen, the degree of degradation can have a marked effect on the shape of the curve and hence the benefit of adding extra engines. The important point that arises from these considerations is that the degree of degradation varies enormously with the type of workload.

Compute-intensive workloads, as are found in engineering-scientific applications, generally have the following characteristics:

- Relatively low I/O content resulting in low disruption to processing (no interference due to process switching, or cache purging)
- Small working sets frequently resulting in major portion of the application residing entirely within processor cache (no delays due to fetching data/instructions from memory)
- Engineering/scientific applications frequently can be broken into units of work that can run in parallel across all engines in the SMP (no interference between engines)

Such applications experience almost linear scalability, as is shown in Figure 15 on page 57.

The traditional batch and OLTP commercial applications, on the other hand, exhibit few of these characteristics and suffer considerable degradation due to SMP effects. In fact, for these workloads adding too many engines may give a negative benefit (that is, less performance with more engines). An example of this effect can be seen by comparing some recent TPC-C results from Hewlett-Packard in Table 12 on page 56.

Table 12. MHz and tpmC Compared for HP 9000 Servers					
Configuration	**MHz**	**MHz Ratio**	**TPC-C $/tpmC**	**TPC-C tpmC**	**Ratio tpmC**
HP 9000 V2200 16-way, Oracle 8.1.5	200	1.000	$103.43	40,794.36	1.000
HP 9000 V2500 32-way, Oracle 8.1.1.5	440	2.200	$87.71	92,832.96	2.276

In this case, the vendor has used the high TPC-C result of the V2500 as proof of industry leadership in OLTP processing. However, as the table reveals, almost all of the improvement over the earlier V2200 number can be explained simply by the use of a faster chip (the architecture of the V220 and V2500 is almost identical). The remainder of the difference may even be due to the differing levels of the Oracle DBMS.

If one fits these two results to an SMP model, it transpires that the resulting curve is entirely consistent with SMP ratios for OLTP on systems with smaller numbers of engines and that this curve does in fact peak between 16 and 32 engines before becoming negative.

An inspection of the SPECweb benchmarks (Figure 14) reveals characteristics similar to OLTP workloads.

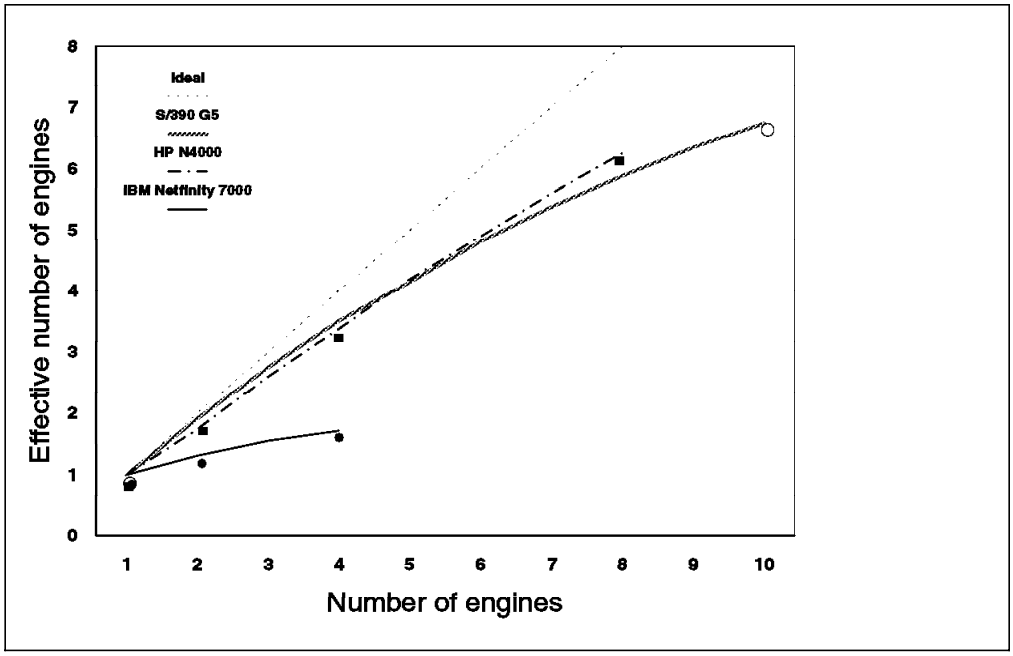


Figure 14. Examples of SPECweb Scalability

Other new application areas, such as Enterprise Resource Planning (ERP) and Business Intelligence, frequently run as multi-tiered applications. For example, SAP R/3 is frequently run with the presentation, application, and database servers on separate servers.

Providing that there are sufficient application servers, this means that the application performance is gated by the application server.

The workload characteristics of application servers and Business Intelligence are such that there is a relatively low I/O content, and there is a high degree of parallelism. Such workloads approach the efficiency of compute-intensive applications and appear to scale well with large numbers of engines.

Figure 15 shows the shapes of the SMP curves typically encountered for a number of types of applications.

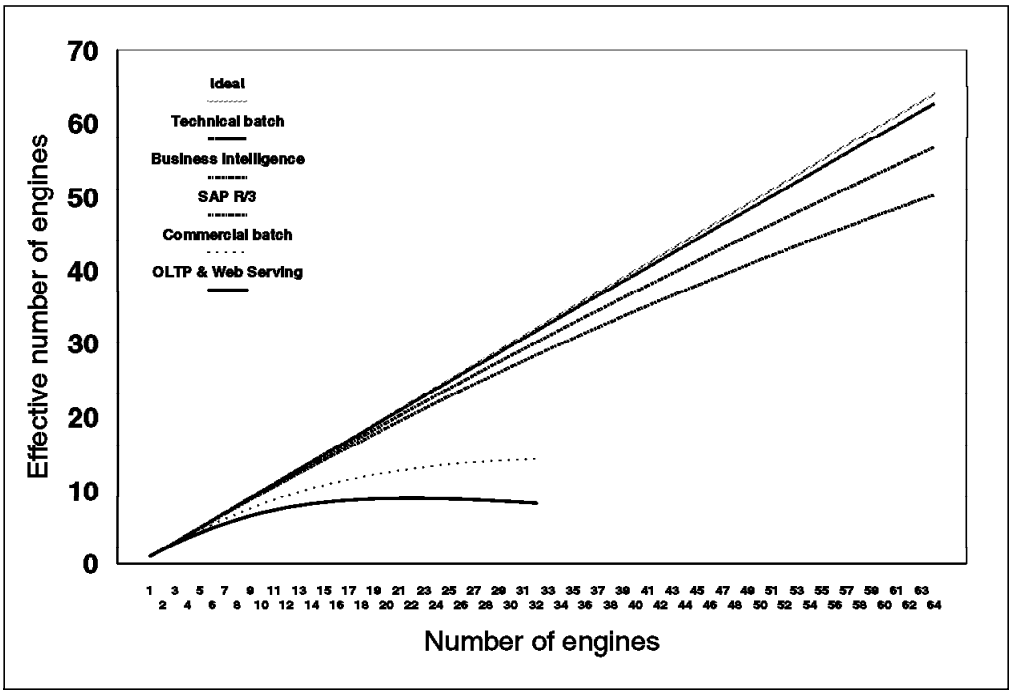


Figure 15. SMP Curves for Different Workloads

It should be noted that these are curves for typical workloads. Not all workloads match these characteristics.

Some applications will not scale on any platform, some will not take advantage of S/390′s capacity advantage, while some will scale only on the S/390 because of intense data sharing.

Implementation issues, such as the use of *fork/exec*, spin locks and memory leaks, can also lead to scaling issues which can effect the comparison of different machines.

4.3 Comparing the Performance of S/390 and UNIX Servers

In addition to the vast number of existing S/390 applications in production today, OS/390 also offers a UNIX environment that is very suitable for the many new applications required by today′s businesses.

OS/390 is UNIX 95 branded and fully, and natively, supports all the standard Application Program Interfaces (APIs). This more recent functionality is supported with all the OS/390 functions for security, reliability, data integrity and system availability of S/390 native applications.

This means that S/390 with OS/390 can run all the major benchmark types discussed in this book. However, as noted, the additional functionality adds to

the pathlength and reduces S/390′s benchmark performance relative to other UNIX systems (see 4.2.5, “Response Time-to-System Utilization Considerations” on page 50).

The value of this extra functionality may be assessed by comparing the different operating environments of S/390 and UNIX.

4.3.1 Application Profile and Operational Considerations on UNIX

The UNIX environment generally has a different approach, that of one application per system. This makes the system architecture both simpler and different, because each system is dedicated to one application and its database.

In this world, if the capacity of a server is exceeded the choices are:

- SMP growth
- Distributed databases with transactions and/or requests shipped between multiple servers
- Clusters of servers

In each option, the result is that the database is partitioned (for example, by engine within an SMP, or by server). The implications of a partitioned database are:

- Additional function/request shipping
- Additional inter-system communications
- Additional single points of failure
- Additional performance impact on scalability

This environment is difficult to tune and therefore servers are often unequally loaded and cannot be driven to maximum capacity.

Some additional questions about partitioned, unshared databases are how to satisfy growth, availability, and backup and recovery. Considerations such as these have led to the sophisticated design of IBM′s DB2 for OS/390 relational database product; see 4.4, “Database Considerations” on page 63.

Let us consider and example of running four different workloads on both UNIX and S/390. The workloads are:

- OLTP (see Figure 16 on page 59)
- Batch (see Figure 17 on page 59)
- Business Intelligence (see Figure 18 on page 60)
- Web serving (see Figure 19 on page 60)

The characteristics of the OLTP workload are:

- Prime shift average utilization - 25%
- Peak utilization - 60%
- Double peak profile typical of in-house OLTP

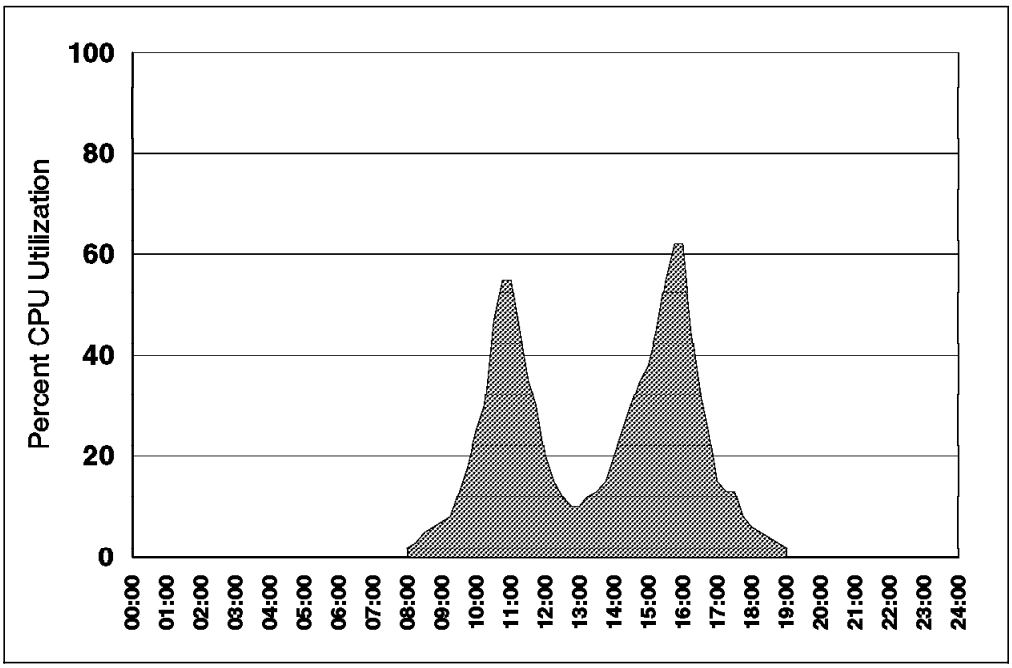


Figure 16. Typical UNIX OLTP Workload Profile

The characteristics of the batch workload are:

- Prime shift average utilization - 20%
- Single threading of batch jobs is assumed
- Large reporting/printing jobs follow completion of BI jobs on BI UNIX instance (see Figure 18 on page 60)
- Database reorganizations/backups performed in second shift

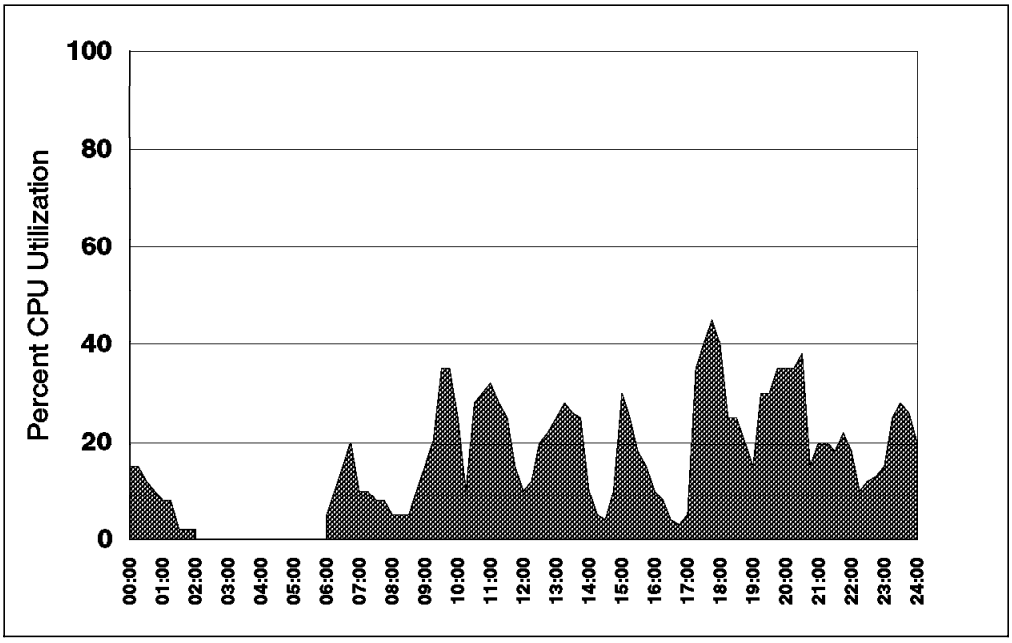


Figure 17. Typical UNIX Batch Workload Profile

The characteristics of the business intelligence workload are:

- Prime shift average utilization - 20%
- Peak utilization - 65%

- 24-hour average utilization approximately 30%
- Small jobs run in prime shift, large jobs in second/third shift

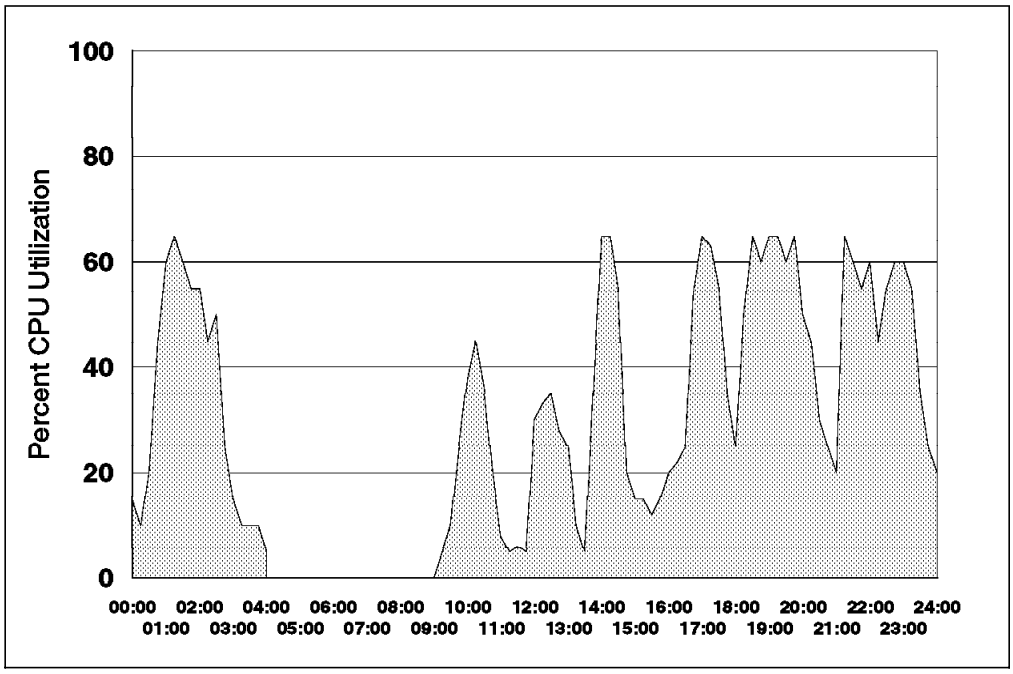


Figure 18. Typical UNIX Business Intelligence Workload Profile

The characteristics of the Web serving workload are:

- Prime shift average utilization - 22%
- Peak utilization - 60%
- In-house and public use in prime shift
- Public use peaks between 22:00 and 24:00
- Usage extends after midnight due to time zones

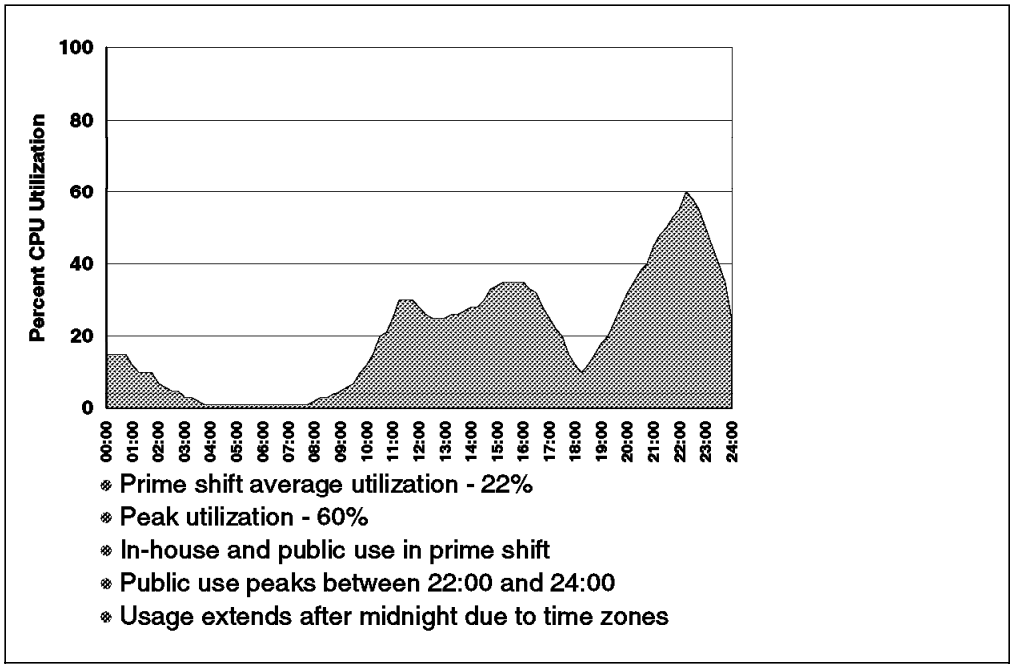


Figure 19. Typical UNIX Web Serving Workload Profile

These examples are considered typical of UNIX servers, that is, an average utilization of 20% to 30% utilization, peaks of 50% to 60% utilization, and a single application per server.

In order to take advantage of the new large SMPs that are in today′s market, and to overcome the poor SMP performance of workloads such as OLTP, batch, and Web serving, some UNIX vendors have turned to physical partitioning of their servers.

In practice, all this achieves is the ability to run several instances of UNIX on a single footprint. There is no resource balancing or sharing between instances. This should be compared to the ability of S/390 to manage multiple workloads on a single image system, as described in 4.3.2, “Application Profile and Operational Characteristics on OS/390.”

4.3.2 Application Profile and Operational Characteristics on OS/390

In a S/390 Enterprise Server environment, customers generally run different application types on the same operating system image. There is no dedicated machine for a specific application. Many applications can run concurrently on the same OS/390 system.

Continuing the example from 4.3.1, “Application Profile and Operational Considerations on UNIX” on page 58, Figure 20 shows the same four workloads running on a single S/390. In this case, the workload which previously required four UNIX servers now runs on a single S/390 with 1.33 times the capacity of one of the UNIX servers.

In addition, there is additional usable capacity on the S/390 (see the “white space” in Figure 20. In an OS/390 environment, it is common to see a processor running at close to one hundred percent of its capacity, and still be providing good service to both online and batch workloads.

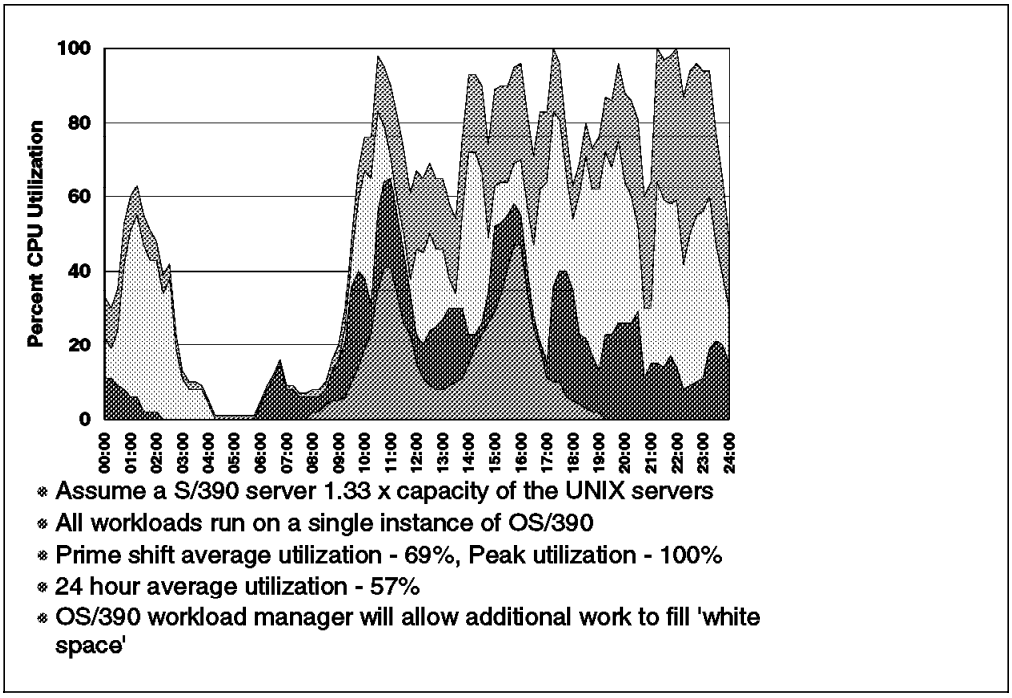


Figure 20. UNIX Workloads Consolidated onto a Single S/390 Instance

Note that there are multiple workload types in this example. These workloads can easily be merged on the same OS/390 server because the OS/390 Workload Manager can manage the resources (CPU, storage and I/O) according to customer policies defined to the OS/390 Workload Manager. Performance definitions or policies can be activated according to fluctuating customer needs. For example, during the day, online activities can be prioritized, and batch activities can be given priority during the night.

OS/390 also provides a secure, reliable environment for multiple concurrent applications by ensuring that no single application can “hog” resources, dominate the use of system resources, or cause other applications to fail. Such an environment is mandatory when many servers and their applications are consolidated to a single system.

What happens if the capacity of the consolidated server is exceeded?

With an OS/390 server, there are two options:

- Increase the number of engines in the SMP, and/or
- Cluster the servers in a Parallel Sysplex

In either case a S/390 solution offers excellent scalability.

The near-linear scalability of a S/390 Parallel Sysplex, together with IBM SMP scalability offers excellent scalability for all types of applications (OLTP, ERP, Business Intelligence, Web Serving, and many others); see, for example, Figure 22 on page 65. This is due to the ability of all systems (engines within an SMP, or servers within a cluster) to access a single copy of each database.

This “shared everything” approach should be contrasted to the “shared nothing” approach that is universally used in UNIX systems.

Shared nothing systems typically partition the databases across individual servers (or server engines within an SMP). When transaction arrival rates (workload skew) or database accesses (data skew) are uniformly spread across the database partitions, this approach can lead to good scalability and high transaction volumes.

In the vast majority of cases, such uniform distributions are only found in benchmark situations. In actual commercial workloads, workload or data skew results in additional overhead in inter-processor, or inter-server, communication in order to access database records in a different database partition.

As a result, while UNIX systems exhibit good benchmark performance, this is frequently not observed in running customer systems where, in order to maintain satisfactory performance, servers typically are only run at 50% to 60% utilization, and each server runs only one single application.

The S/390 shared everything approach, while requiring higher overheads for single, simple applications, can drive multiple, concurrent applications to very high processor utilizations (90% to 100%), and can exhibit near-linear cluster scalability.

4.4 Database Considerations

Many Relational Database Management systems (DBMSs) on UNIX platforms have taken on the role of processor resource management, instead of using the underlying UNIX operating system services. However, the RDBMS is usually unable to manage resource for other workloads in the system and can end up hogging resources. For UNIX users, this is often not perceived to be a problem, since they are accustomed to only running one application per UNIX instance.

The fact that these database systems feel the need to provide resource management is a tacit recognition that most UNIX operating systems offer very low-function workload dispatchers.

This has long been recognized by users of UNIX systems with mainframe type applications. Recently, a number of vendors have commenced to address this deficiency with their first generation of higher function schedulers (for example, the Process Resource Manager (PRM) for HP′s HP-UX, and the Solaris Resource Manager (SRM) for Sun′s Solaris).

A secondary effect of the database management system handling workload prioritization is that they become more tightly bound to the hardware with which they are associated. This in turn makes porting to an alternative platform increasingly difficult.

IBM chose to implement a shared everything design for its S/390 cluster architecture, in order to solve customer demands for high volume, high growth with scalability, and high service availability. As a result, it is possible to manage very large databases (terabyte size) with exceptional performance.

Using IBM′s relational database system, DB2 for OS/390, all data is shared and concurrently accessible by all servers in the cluster. Even though the database is fully shared by all servers, DB2 for OS/390 allows the database administrator to manage the database as if it were partitioned. This means that maintenance functions, such as database reorganization, can be performed independently on each partition, resulting in higher availability of service and fewer scheduled service interruptions.

DB2 has been improved over many years in response to customers′ needs. Business application growth drives transaction volumes and DB2 for OS/390 scales well within an SMP. In addition, near-linear scalable growth beyond the largest S/390 SMP is possible using IBM′s Parallel Sysplex clustering architecture.

In addition to superior performance in OLTP processing, which is required when multiple smaller servers are consolidated to a single, larger server, DB2 for OS/390 on an IBM Parallel Sysplex cluster supports large data warehouse processing for Business Intelligence applications.

DB2 for OS/390 has evolved over the years by increasing the degree of parallelism in its database processing (see Figure 21 on page 64).

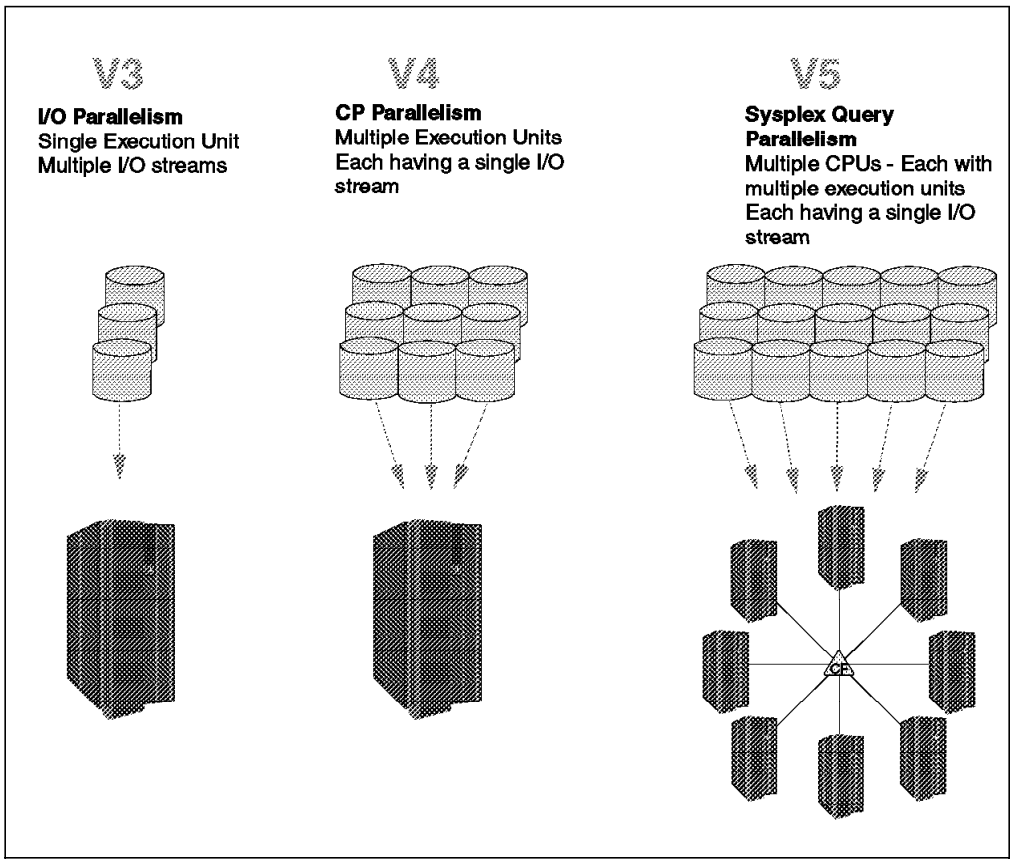


Figure 21. DB2 Evolution Steps

The efficiency of DB2 parallelism and Parallel Sysplex means that DB2 database servers scale very well with both increasing engines in an SMP and across a cluster.

Figure 22 on page 65 shows a scalability of greater than 93% for a processor-intensive business intelligence query.

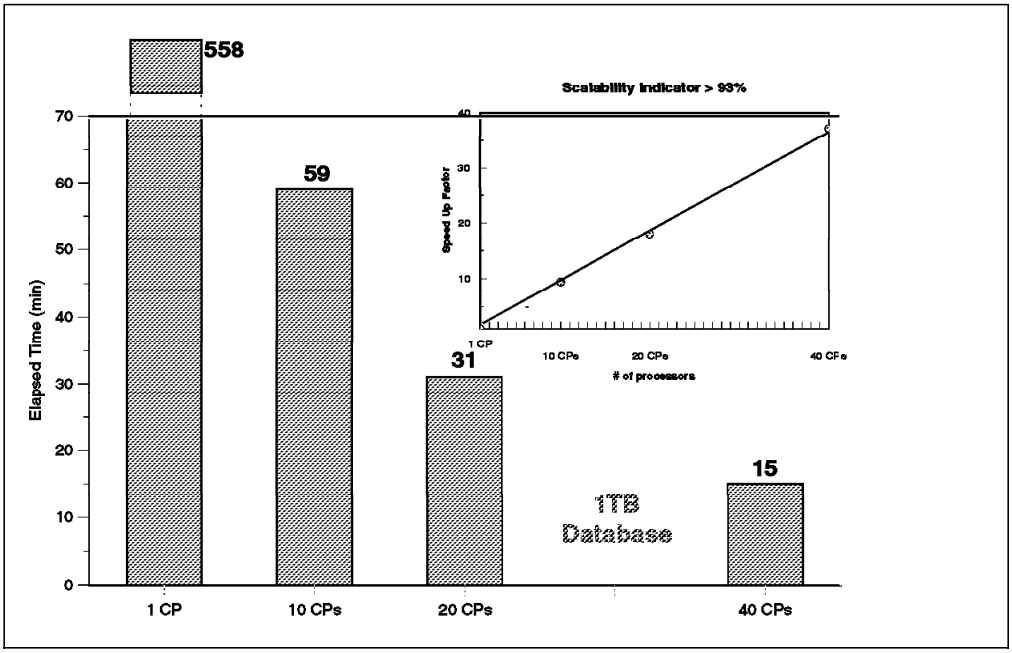


Figure 22. DB2 Version 5 Scalability on Parallel Sysplex

Business intelligence systems have to accommodate a wide variation in the size and importance of the queries submitted. Unlike most TPC-D benchmark results, customers need to be able to run multiple queries concurrently. The OS/390 workload manager enables a mixture of queries of different sizes to scale at greater than 90%, while maintaining the same proportion of query types (see Figure 23).

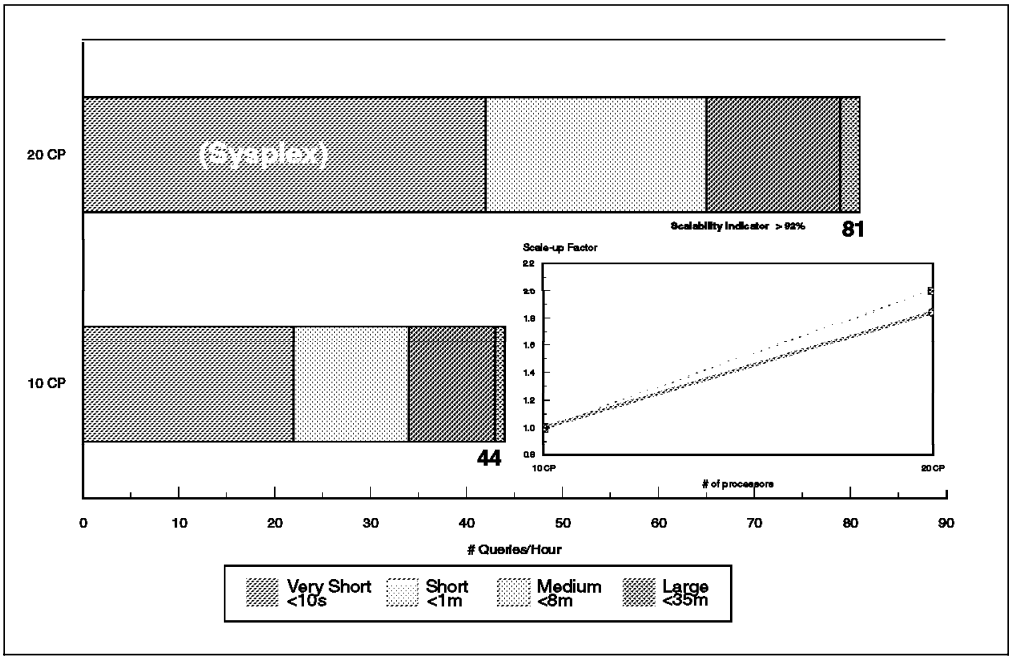


Figure 23. Mixed Query Workload Scalability on Parallel Sysplex

A good reason for server consolidation is to combine multiple instances of “data marts” into a single S/390 server in order to do the following:

- Reduce the effort and time in moving data out of existing operational systems on OS/390 to the data warehouse and data marts for Business Intelligence processing.
- Improve the turnaround time for Business Intelligence application queries.
- Enable efficient processing of the wide spectrum of concurrent query types, including processor- and I/O-intensive queries, and short (a few seconds) to medium (a few minutes) to very long (hours) queries.

DB2 for OS/390 on a Parallel Sysplex provides near-linear (greater than 90%) scalability over a wide range of cluster sizes. The Parallel Sysplex architecture allows the addition of up to 32 OS/390 servers in order to adapt the processor power the customer needs to face business growth.

The Parallel Sysplex architecture provides high service availability. If a processor fails, the remaining processors in a Parallel Sysplex still have access to the database because the database is shared.

4.5 Sizing IBM S/390 Servers

Vendors of UNIX (and NT) servers frequently make extensive use of Industry Standard benchmarks in marketing their products. Generally such standard benchmarks either do not stress the I/O handling capability of the system (for example, SPECint), or only use a uniformly distributed arrival rate for transactions with a uniformly distributed data access pattern (for example, TPC-C). Neither environment is typical for commercial applications. Such benchmarks enable the UNIX/NT vendors to perform extensive tuning, and even optimize their hardware and software designs to exploit the benchmark conditions.

However, as noted earlier in this chapter, real commercial systems do not exhibit such uniform patterns and usually do not achieve benchmark-like performance in production systems. Because such benchmark systems are so tightly tuned, there is a strong dependence in the results on the particular combination of server architecture, UNIX operating system (vendor-dependent) and database system used.

In many cases, only a single benchmark number may be available for one combination of server model (SMP), version of UNIX, and database management system. Direct comparisons between other models in the same server range (for example, with a different number of engines in an SMP), or with other vendors′ results may thus be extremely difficult, if not impossible, to do.

Thus, while Industry Standard benchmarks may give a means of comparing the theoretical performance of different UNIX vendors, they are not very useful in assessing the capacity of a given system for a particular task (server consolidation, for example), nor do they enable easy comparison of different vendors′ systems, or architectures.

S/390 servers, on the other hand, can be readily compared and sized in a realistic fashion by means of IBM-provided measurements.

The IBM Large System Performance Reference (LSPR) is a method designed to provide relative processor capacity for System/390 architecture servers. The IBM S/390 Division runs LSPR benchmarks, which are controlled tests representing different workload environments (for example database processing,

online transaction processing and batch). Each LSPR workload is a unique benchmark and is designed to be a closer match to actual customer environments than most Industry Standard benchmarks (such as TPC-C, SPECint and so on).

In order to size a customer server requirement, the LSPR data that most closely resembles the customer workload is used. Data for all current IBM S/390 servers, and for a large number of non-IBM S/390 architecture servers, is available. The best match to a customer′s capacity needs can then be easily determined for either dedicated or mixed workloads.

In the S/390 server marketplace many people refer to millions of instructions per second (MIPS) as a single-number metric for comparing servers from the same or different hardware vendors.

This metric represents simply the instruction execution rate and does not consider the internal processor design, or the ability of a server to deliver useful work for any given workload. It is particularly invalid for comparisons between servers of different architectures (such as IBM S/390, RS/6000, AS/400, or UNIX vendors such as Hewlett Packard or Sun Microsystems) and running different operating systems (such as OS/390, UNIX and NT).

IBM does not use MIPS as a metric for capacity planning because it is not sensitive to the type of work being processed. In fact the acronym MIPS is considered to stand for “Meaningless Indicator of Processor Speed.” However, the term is widely used and a single number per server can be established for a fixed workload mix and scale relative to a base machine, so long as the limitations of such a single number are understood by the user.

In fact, server capacity may vary significantly depending on the characteristics of the workload. Production workloads may be batch-intensive, online-intensive or a mixture of each. Workloads may also range from CPU-intensive to I/O-intensive.

The LSPR measurements include seven different workloads representing many different customer workload types. Each individual LSPR workload is designed to focus on a single type of activity, such as interactive, online database, or batch.

By matching a customer′s workloads to the LSPR workloads, a good understanding of the capacity of different IBM (and non-IBM S/390) servers for the consolidation can be obtained. Thus, the challenge is now to understand the capacity and performance relationships between UNIX and S/390 servers.

4.6 Resetting the Bar

Industry Standard benchmarks have some value (as discussed in Chapter 3, “Strengths/Weaknesses of Industry Standard Benchmarks” on page 33) in comparing platforms with similar architectures, operating systems and approaches to data sharing.

What we have tried to do in this chapter is to introduce some of the considerations that need to be understood in order to properly compare unlike computer platforms.

The key items to be considered include the following:

- Ensure you understand the workload profile.
- Understand that UNIX systems typically run at 20% to 30% average utilization, while S/390 systems run at 80% to 100% utilization while delivering low response times.
- Understand that UNIX systems typically peak at 50-60% utilization, while S/390 systems operate at, or near, 100% for extensive periods of time.
- Most UNIX systems either need separate servers or partitioning to run multiple concurrent applications, where only a single S/390 server might be required.
- A single S/390 has higher availability than a UNIX cluster.
- Parallel Sysplex provides near-linear scalability for both traditional and new application workloads.

Chapter 5. Strengths of IBM′s S/390 and SP

Clearly no single benchmark test can hope to assess and compare all the possible features and functions of any given server that may be of value to a customer.

Benchmarks are, by definition, intended to compare the performance of different systems in running the same application. However, many factors other than performance have a business importance that often outweighs any performance difference between systems.

Prominent among these are:

- Security
- Application availability to the end user
- The ability to recover from failures
- Fast application restart after failure
- Integrity of information

In addition, no standard benchmark can measure the productivity, or business benefit, of the ability to concurrently run batch and online applications so that the hours of online service are extended by reducing or eliminating the batch window.

IBM′s S/390 and Scalable POWERparallel (SP) servers both contain many features that are of considerable value, but which are not measured by any Industry Standard benchmark.

The remainder of this chapter is devoted to discussing some of the key values of both servers.

5.1 Strengths of S/390

The S/390 platform has evolved over many years of sustained hardware and software development to provide the richest functionality in the industry. Any individual customer or application may only require a subset of these functions, but that subset will often be of enough value to swing the business to IBM.

The TPC benchmarks are designed with enough latitude that they can be implemented on multiple architectures. This generalized approach does not exercise or even recognize the additional S/390 capabilities, which can be broadly categorized as those which contribute to:

- High Performance
- Continuous Availability
- Systems Management

The bottom line is that this additional function in S/390 comes at the cost of an increase in the number of lines of code. As explained in the rest of this chapter, much of this code will be executed during a TPC benchmark without improving TPC transaction throughput.

When comparing S/390 with other architectures, it should be remembered that these alternatives do not provide many of the features which are so often taken for granted by our customers. Within IBM it has been estimated that a typical UNIX Operating System would require an additional two million lines of code to provide a similar level of function to MVS on a S/390 processor. If and when UNIX includes all the functions of MVS, we will have an apples-to-apples comparison. Until then, we should continue to remind our customers of the added benefits which MVS can provide.

In order to ensure high performance with balanced throughput, a number of techniques are used in the S/390 platform.

The key strengths of the S/390 platform are its high availability, security, program and data protection, and its ability to process multiple concurrent applications at high utilizations.

Some of the factors contributing to these strengths are discussed in the following sections.

5.1.1 N-way Multiprocessing

In order to provide very high performance while preserving data integrity and access to shared resources, S/390 and OS/390 include special hardware and software components. These manage the queues of work waiting for shared resources, and do so with very high multiprocessor efficiency and data integrity.

Most alternative platforms now also use large SMPs but do not have the workload management capabilities of S/390.

In many of the standard benchmarks, the benchmark allows the data to be partitioned so that, for example, transactions of a given type always use the same processor engines and access the same fixed partition of the database (so-called CPU affinity).

This means that the overhead (and consequent performance loss) caused by task switching and cache refreshing is minimized. Thus, UNIX and NT vendors are able to obtain very good benchmark numbers at high processor utilizations. Experience shows that UNIX installations typically peak at 50 to 60% utilization, and only average 20 to 30% across a shift.

S/390′s design handles task switching and cache management in such a way that high performance is achieved without the artifact of data partitioning even for high levels of symmetric multiprocessing in both benchmark and production environments.

The marked skew towards updates in the TPC benchmarks favors systems which can perform writes quickly and simply. In a symmetric multiprocessor where there is data sharing between processors, exclusive locks must be obtained for updates. Interprocessor communication is required for each update to preserve local buffer coherency. This overhead puts data sharing systems (such as the S/390) at a marked disadvantage when running the TPC benchmarks, but is of major business value in production systems.

Non-sharing systems, such as many UNIX-based processor implementations, will not show such significant overhead because they do not need to take locks on the data, but they will suffer from all the problems associated with workload

balancing, availability and so on. Of course, these problems are not highlighted by the TPC or other benchmarks.

The same considerations apply when comparing clustered configurations, in which multiple SMPs are interconnected to provide processing power for a single application. In a full data sharing implementation such as IBM′s Parallel Sysplex, there will necessarily be some overhead in ensuring that data integrity is maintained across nodes. In other architectures data is partitioned to be processed by individual nodes and the locking overheads are not incurred. For an application with predictable patterns of data access, such as the TPC benchmarks, the non-shared implementation can easily be tuned to maximize the utilization of all nodes. In a real commercial OLTP application this is rarely the case, and some nodes may be underutilized while others are at full processing capacity. In contrast, workload in a Parallel Sysplex is managed to optimize utilization of all nodes.

The effects of Symmetric Multiprocessing and the capability of other vendors to provide SMPs that perform well is discussed in detail in 3.3, “Shared versus Non-Shared Architectures” on page 39.

5.1.2 Application Processing

S/390 processors are designed to handle a wide variety of large commercial processing requirements. However, if there is one feature which characterizes most large system workloads, it is the significant amount of function in a typical transaction. The majority of large OLTP systems will have a transaction path length measured in hundreds of thousands (or even millions) of instructions. For most customer workloads, application processing will form 50% or more of the workload, with the remainder of the processing being related to data accesses or other operating system services.

In contrast, in the TPC-C workloads, less than 1% of processing is performed by the application. This emphasizes the pathlength cost of the operating system and DBMS in the TPC workloads. As a result, TPC benchmarks favor low function, low pathlength operating systems such as most versions of UNIX, while placing high function, high pathlength systems such as OS/390 at a distinct disadvantage.

It is often said that the only good benchmark is one based on the customer′s own application. A real application may be very different from the TPC benchmark. In one example, a customer measured the performance of platforms from two UNIX vendors using their own customized workloads. The TPC benchmarks indicated that the two systems were within 10% of each other in performance terms. The customer workloads showed a 100% difference. The irony in this case was that the customer benchmark was based on a customized version of TPC-A!

5.1.3 I/O Bandwidth

S/390 processors are capable of handling very high I/O bandwidths through the use of ESCON, and, in some cases, the use of specialized I/O function such as sequential data striping. Even if an application does not require high I/O bandwidth, most customers will use the available channel bandwidth to reduce their channel configuration, and load their disk controllers as heavily as possible.

The current TPC-C benchmark is based on OLTP workloads which require relatively small amounts of data to be read and written for each transaction. As a result, the I/O performance capabilities of the systems under test are hardly stressed.

This is not typical of Enterprise System configurations, which may well have hundreds of gigabytes of data behind a single controller with 4 to 8 channel paths. This S/390 approach is possible because of the high performance characteristics of IBM disks, and has the additional benefits of reduced channel subsystem costs and easier configuration management.

What is clear is that customers are requiring increasingly large I/O bandwidths to handle “new world” applications such as multimedia, video-on-demand and others that handle large data objects. This bandwidth is available on the S/390 range. It may well be restricted in other architectures.

5.1.4 Specialized Functions

There are a number of more specialized functions of the S/390 range which can contribute significantly to overall throughput of a system. These include:

- Data Compression

  The use of can significantly improve performance. The compression of data within the central processor increases the effective storage capacity of the entire storage subsystem. This means that additional data is stored within the processor′s central storage, expanded storage, auxiliary storage, storage control cache, and the disk device itself. Increasing the effective capacity of these storage elements provides significantly higher hit ratios and significantly higher bandwidths on the transport media. Together they result in greatly reduced response times for online transactions and batch jobs.

  It is notable that the S/390 has hardware assists to reduce the CPU overhead of compression to make it economically very attractive. It is also notable that, for UNIX, systems compression is only available through software--with associated costly processing overheads.

- DB2 Sort Assist

  DB2 Sort Assist is a microcode enhancement in S/390 which can significantly improve DB2 sort performance. It is an example of the synergy between IBM hardware and software that can only be produced because IBM develops both products.

  In the UNIX world, the primary software vendors are generally independent from the hardware vendors. Many of the hardware vendors develop their own DBMSs (such as HP′s ALLBASE), but they have a very small installed base. While there are alliances, it is rare to find hardware assists on non-S/390 platforms that benefit the performance of any one DBMS.

5.1.5 Managing Multiple Workloads

The TPC-A workload was restricted to a simple OLTP workload. TPC-C is somewhat broader in that it has five transaction types, but nonetheless it is still restricted to OLTP work. Many OS/390 workloads are also predominantly OLTP-based, but it is quite usual to find other work such as background batch, interactive work, decision support, and utilities running in the same system. It is typical for new OLTP applications to start out by providing straightforward OLTP function. But over time user requirements soon alter the workload profile so that

it can include interactive and batch elements as well. It is also true that the proportion of batch, interactive, and decision support work will grow in proportion to an increase in the amount of OLTP work.

IBM has provided customers with the tools to ensure that multiple workloads can be assigned an appropriate amount of resource, dependent on priority. The primary product for workload management has been the Systems Resource Manager (SRM). This ensures that mixed workloads receive an appropriate amount of resource, but it does require some tuning. More recently the Workload Manager component of OS/390 has enabled customers to specify targets based on real service level agreements. The Workload Manager dynamically allocates resource to all the workloads to ensure they meet their targets. In a Parallel Sysplex, this management is taken one step further by the subsystem managers ensuring that work is divided between multiple OS/390 images (an example is the CICSPlex Systems Manager (CPSM)).

Inevitably, the management of the multiple workloads by the system requires some small additional processing overhead. The TPC-C benchmark requires absolutely no management of resources between contending workloads, and so takes no advantage of the OS/390 capability to manage heterogeneous workloads. The lack of the requirement for workload management, combined with lax response time requirements, tends to favor smaller and simpler operating environments.

5.1.6 Scalability

The scalability of real customer systems is a function of the hardware and software capabilities of the system on which they are running.

The scalability of hardware depends largely on the amount of data sharing in a workload. In an SMP, the more sharing there is, the more interprocessor communications there will be. In a shared node system (that is, where multiple SMPs are coupled) as in a Parallel Sysplex, the more data sharing there is, the more internode communication there will be. The effect, in either case, is to create overhead and thus reduce the overall performance of the system.

S/390 systems have continually improved SMP performance so that overheads are small enough that customers can take advantage of the benefits of shared data systems (high availability, performance, and scalability). Similar data-sharing considerations apply in the Parallel Sysplex environment, where up to 32 OS/390 images may share data.

In a non-shared architecture, interprocessor communications and data sharing are reduced to the minimum. This often requires that physical disk is only directly attached to one processor and only that engine can work with data on the attached disk. For real customer workloads this inevitably leads to performance hot spots, as the high activity on one area of a database cannot be handled by the processor to which it is attached. Performance in these non-shared systems rapidly plateaus as additional processors are added.

However, for a benchmark workload, it is possible to partition data so that there is little or no skew of access to the data. Each individual processor to which the data is attached can be driven to capacity--with very little communication needed between processors. These systems can show an apparently very high throughput, but they are totally unrepresentative of real customer workloads. A more detailed discussion of the benchmarked performance of shared and

non-shared architectures and their scalability is given in 3.3, “Shared versus Non-Shared Architectures” on page 39.

5.1.7 High Availability

Quite apart from the performance capabilities of IBM S/390 products, there are many other features which assist in the delivery of availability and systems management for customer applications.

In many cases considerations such as the availability of a system are just as important to our customers as the price and performance of a system. After all, if a system is down, nobody cares about response times!

Until recently, continuously available systems were required by only a few industries such as banking. Now, however, the increased pressure to remain competitive in the 1990s has forced the requirement for systems to run 24 hours a day, 7 days a week, in many other industry areas. Retail outlets are opening for longer hours, all weekend, and during holidays. The computing systems that support the business must be available to support these longer hours.

Another example is the telecommunications industry, which is increasingly focusing on new “content services” to customers. These applications might include home shopping, home banking, and video-on-demand to name just a few. If, for any reason, these services are unavailable at any time, the companies running them will suffer a direct loss of revenue.

There are many other examples but the net effect is that an increased number of systems are required to be available all day, every day.

The IBM S/390 range and Parallel Sysplex have been architected to meet the stringent demands of these enterprise applications. This is achieved through a combination of hardware and software function. Inevitably this additional function increases the cost of a transaction, but there is no recognition of this in the TPC benchmarks.

It is not our intention to provide an exhaustive view of the availability features of S/390 and Parallel Sysplex in this redbook, but it is worthwhile drawing out some salient points.

When we look at availability, it is worth considering four areas:

1. Error detection and correction, to preserve data integrity
2. Hardware and software fault tolerance, to minimize unscheduled outages
3. Fast recovery, to reduce downtime associated with unscheduled outages
4. Dynamic hardware, microcode, and software changes, to reduce the need for scheduled outages

5.1.7.1 Error Detection and Correction

Data integrity has been key to the design of IBM processors since the introduction of the first S/360 machines. There are many ways in which error detection and correction techniques can help preserve data integrity. Just one area is in memory. Fault Tolerant Dynamic Memory Arrays provides a technique for toleration of errors in central storage (memory) and expanded storage (an IBM S/390 unique second hierarchy of memory).

Today′s processor storage is extremely reliable. To provide this reliability, the first line of defense is Error Correction Code (ECC), which corrects memory

errors. Errors that are not corrected by the hardware ECC are processed and corrected by a complement/recomplement procedure. If this type of error persists, the memory location of the page containing the error is de-allocated from the active configuration used by the operating system. A *hard* error is the result of a permanent failure of an array chip. The impact of such an error is minimized by “array chip sparing” which logically substitutes a spare memory chip from the same memory card for the failing chip in the active configuration. If error thresholds are exceeded, data is moved to these spare chips automatically and processing continues.

Path sparing is a feature which improves redundancies within the Central Storage subsystem on S/390 processors. Path sparing adds to the existing redundancies already built into this memory element through the dynamic fault-tolerant memory array chips. Path sparing provides extra data and control paths to each of the memory array chips. Not only can the S/390 processor transparently recover from the loss of a memory array chip, it can also transparently recover from the loss of a data or control path to a memory array chip. Again the design goal is to provide survivability to the operating system and applications being supported on the processor in the unlikely event that a component failure should occur.

In the worst case, if these availability techniques fail, the system will stop in order to ensure that data integrity is maintained.

While alternative mainframe implementations offer ECC techniques, the majority do not offer memory or path sparing. S/390 has been architected to ensure that data integrity is maintained at all times. It is not unusual to hear of customers “reconstructing”data (rather than recovering it) on other platforms.

Data integrity takes circuitry and processor cycles which get included with application processor time during benchmarks. It may not help improve benchmark performance, but the vast majority of customers would prefer to have confidence that their key business data is sound and reliable.

5.1.7.2 Fault Tolerance

There are numerous features of the S/390 range which provide very high availability through fault tolerance. In general it is possible to suffer an outage with little more than some degradation in performance. This is generally not true of alternative platforms.

Some examples are:

- Alternate CPU Recovery (ACR)

  ACR enables OS/390 executing on a CPU that fails, to be dynamically recovered on an alternate CPU in a tightly coupled multiprocessor. This capability is only possible because the system is architected to broadcast the failure, and recover the executing task--a procedure that requires the presence of recovery code in OS/390.

  No such capability generally exists on RISC processors. The loss of a processor will invariably cause a complete SMP system outage. No TPC benchmark requires any demonstration of a capability to continue processing despite the event of a processor failure, or of the performance that might be expected following such a failure.

- Subsystem Storage Protection (SSP) and CICS Transaction Isolation

The S/390 architecture enables high availability through the isolation of code and data held in memory. It does this through the provision of storage keys and subspaces. This architecture has enabled the implementation of facilities such as SSP and Transaction Isolation. These can greatly reduce outages of systems due to the accidental overwriting of application or subsystem areas in storage.

There is no point in implementing these features in a TPC benchmark because their provision is a trade-off between improved application availability and a slight overhead in performance. Since TPC only measures performance, any overhead needs to be eliminated. Yet many customers are prepared to make this trade-off in their own application environment to ensure high availability. It is notable that most RISC architecture platforms have no equivalent to the S/390 storage protect keys and therefore, while they have error detection and correction, they provide little in the way of protection function.

- Parallel Sysplex

  The IBM Parallel Sysplex provides new levels of application availability through the provision of “software fault tolerance” as well as hardware fault tolerance. In a tightly coupled multiprocessor or SMP, all hardware is under the control of one copy of the operating system, and the application is dependent on the availability of that one copy. In the Parallel Sysplex, an application runs horizontally across multiple copies of OS/390 and its subsystems. The failure of one software system will not necessarily lead to a failure of the application. This only works because each separate system can independently access all the data associated with the application. This data sharing inevitably leads to some elongation of transaction pathlength.

In contrast, most alternate platforms are relatively unsophisticated. Some high availability clusters (such as the HACMP/6000) do not suffer an application outage with the failure of a component. However, in many other instances clustering simply provides the capability to recover the system on the remaining components in the cluster which have not failed.

The net value of functions such as these is amply demonstrated by the leadership of S/390 processors in independent availability surveys. Where figures for mean time between failure (MTBF) and mean time to repair (MTTR) are published by the vendors of alternate platforms, they simply demonstrate the gulf between what the alternatives can provide and the “industrial strength” of an OS/390-based platform.

5.1.7.3 Fast Recovery

The TPC-C benchmark does not test recovery performance. The TPC-E benchmark required that the benchmark system be crashed when operating at full load. The time between initiation of recovery and the system working again at 95% of full load would then have been measured and reported. However, as previously mentioned, TPC-E was rejected by the council, and thus there is no standard benchmark which tests recovery performance.

The error recovery facilities built into the S/390 hardware, OS/390, and the database subsystems, mean that S/390 systems very seldom crash and corrupt the database.

However, in the rare event of failure, the combination of logging facilities in the database subsystem, and the superior speed of tape and disk processing in

S/390 compared to typical UNIX systems, means that restoration of full service is a rapid process.

5.1.7.4 Dynamic Configuration Changes

Nobody likes catastrophic application outages. They cause a lot of pain. But surveys show that they only account for 10% of the time that a system is unavailable. The remaining 90% of downtime is due to planned and scheduled outages. These invariably consist of a change to the system, whether it is for an application upgrade, system software upgrade, or a hardware configuration change.

S/390 has gradually incorporated more function over the years to ensure that as many as possible of these changes can be made dynamically. For instance, in a S/390 system, it is possible to hot-plug a disk into the I/O subsystem.

The IOCDS can be dynamically updated using HCD to reflect the addition of a disk, and the new disk can be dynamically incorporated into an SMS volume pool, ready to be used by application data. All of this can be achieved without any application outage.

The picture in a UNIX environment is very different. An outage will be required to physically attach and define the disk to the system. In many partitioned systems, a database reorganization and tuning exercise is likely to be needed to ensure that the processor to which the disk is attached can handle the additional workload. The combination of changes may require an extended outage every time new capacity is required.

Users of alternative platforms have few options for dynamic changes. They are still in a world where a software version upgrade is a conversion rather than a migration.

5.1.8 Systems Management

One of the issues that we have always faced with the S/390 platform is that the heads in an IS department can easily be counted. For a distributed system (or even a centralized alternative platform), the costs are often hidden because the tasks associated with managing systems are often carried by the user.

As a result, there has always been pressure to reduce data center headcount costs. IBM has assisted with this process by automating many of the mundane tasks and processes like performance tuning and data management. Of course, the automation of these functions has required more processing power, and the trade-off has been a reduction in IS headcount.

The TPC benchmarks take no account of the fact that these functions are available in OS/390, despite the fact that they reduce the overall cost of computing and provide functions which contribute to the overall “industrial strength” of OS/390.

5.1.9 Resource Utilization and Performance Management

The Workload Manager (WLM) component of OS/390 has already been mentioned in terms of its capability to manage resource in a single OS/390 system. Changes to parameters directing WLM decisions can be made dynamically, without stopping any application or subsystem software.

The advantages of a single large pool of resource which can be flexibly assigned to different systems has long been acknowledged by our customers. To help manage multiple systems in a large single footprint or complex, we have provided the partitioning functions of PR/SM. The available processor, memory and channels can be split between multiple different workloads on a single central electronic complex. As the resource requirements for these workloads change, the allocation of resource can also be dynamically altered.

The capability to manage multiple heterogeneous workloads on a single machine in this way is not measured by any Industry Standard benchmark.

The capability to allocate resource in this way is almost unheard of on alternative platforms. At best, there may be some software function that allows a defined pool of processors to be allocated to an application. At worst, a new application will require a totally new hardware footprint.

This lack of resource management invariably leads to over-configuration of UNIX systems, and poor resource utilization. Usually, individual applications will require individual processors, and while one footprint is lightly loaded, another may be short of capacity.

The ability of the system to manage resources in the most effective manner also reduces the dependence on systems programmers to tune systems for performance. Following publication of TPC-C results on their Model H40, HP admitted that three engineers had spent three months tuning the benchmark. They claimed that “This is in line with how much time an MIS shop would spend tuning an application of this size.” Spending nine man-months to tune one simple OLTP application is far from normal in a S/390 environment.

5.1.10 Storage Management

The S/390 platform is known for its excellent disk and tape storage management facilities. Most customers use some degree of this automated hierarchical storage management to ensure that storage subsystems are well-utilized, and to control the flow of data up and down the hierarchy.

Data is managed through the use of Automatic Class Selection routines and data sets will be tagged to ensure they receive the appropriate levels of service in terms of performance and availability.

Implementation of such a strategy shows major savings in headcount and data storage. It can also help to meet service levels for data retrieval--both active and archived. The benefits of such sophisticated data management techniques can often only be fully appreciated when applied to a relatively complex customer system. This is not the case with Industry Standard benchmarks, which are relatively simple, and where data placement is likely do be performed manually to ensure the very best performance.

There has been an interesting trend in recent years that centralized data management tools such as ADSM have become increasingly popular for managing distributed platforms. This has often happened, not because of a requirement for remote backup of data, but because the platforms being backed up simply do not have adequate facilities locally.

5.1.11 Security

No level of security is required by any of the Industry Standard benchmarks. However, the protection of certain data or fields of data is normal in most customer systems. Every time the data is accessed, there will be some overhead as the security manager is exercised.

The provision of even reasonable levels of security inevitably means some processing overhead, but the trade-off here is security against performance. Very often this trade-off depends on the level of customer confidence required in a system. It is not surprising to find that most financial institutions base their systems on S/390.

5.1.12 Summary of S/390 Strengths Not Measured by TPC

Table 13 summarizes the strengths of S/390 and OS/390 that are not measured by the TPC, or other Industry Standard benchmarks.

Table 13 (Page 1 of 2). S/390 Strengths Not Measured by TPC

S/390 Function	Significance	Availability on UNIX
Memory management	S/390 systems require less memory than UNIX databases. Benchmarks do not stress the paging system in the same way as found in production.	UNIX paging systems are typically not as good as S/390.
I/O bandwidth	TPC benchmarks do not stress I/O systems. S/390 has capability of handling the continuous high I/O rates which are found in production.	UNIX I/O systems do not have as high a bandwidth as S/390.
Data compression	Hardware data compression can reduce disk space requirements significantly without incurring the extremely high processor usage of software compression.	Software compression available, but not used widely.
DB2 Sort Assist	Limited sorting in TPC benchmarks. SORTS are a major component of most commercial workloads. DB2 Sort Assist improves sort throughput significantly.	Not available on UNIX systems.
Multiple concurrent workloads	Limited workload management management in TPC benchmarks. Commercial demand mixed workloads.	Limited workload management available on most UNIX systems.
Fault tolerance	Not measured by any benchmark. Loss of a processor engine in UNIX usually causes loss of the system, while S/390 has mechanisms to ensure that the application continues on the remaining engines.	Limited fault tolerance on most UNIX systems, unless specifically special fault-tolerant designs.
High availability	Not measured by any benchmark.	Single S/390 systems generally are more available than UNIX high availability clusters.
Peripheral device performance	Benchmarks focus on total, highly tuned system performance. Extra I/O is often used to maximize throughput.	S/390 I/O devices have superior performance and availability to most UNIX devices.
Parallel Sysplex performance, scalability and availability	Most benchmarks do not measure any of the capabilities of the IBM Parallel Sysplex (exceptions are TPC-D and SAP R/3 Parallel SD).	Parallel Sysplex capabilities are not available on any UNIX system.

Table 13 (Page 2 of 2). S/390 Strengths Not Measured by TPC		
S/390 Function	**Significance**	**Availability on UNIX**
Systems management	Systems management functions, such a problem and change management, software management, software distribution, network management, and so on, are not assessed in any benchmark.	S/390 systems management are generally acknowledged to be superior to most UNIX systems.
Backup and recovery	Not measured in benchmarks. S/390 times are usually significantly faster than in UNIX systems due to better I/O handling, better system and sub-system design and parallel handling of I/O.	S/390 backup and recovery are generally acknowledged to be superior to most UNIX systems.
Data integrity	Not measured in benchmarks. S/390 has protection mechanisms to prevent loss or corruption of data either deliberately or by accident.	S/390 data integrity is superior to most UNIX systems.
Security	Not measured in benchmarks. S/390 and OS/390 have earned some the highest security ratings of any general purpose computing systems. Hacking into S/390 is almost unknown, while it is a serious exposure in UNIX.	S/390 security is better than UNIX.
Cryptography	Not measured in benchmarks. This is a growing, mandatory requirement for electronic business over the Internet. Only S/390 has in-board, hardware cryptographic capability. This allows fast, low cost encryption/decryption.	Not available in UNIX.

5.2 Strengths of RS/6000 SP

The advantages of a platform often vary with the characteristics of applications to be executed. In this section the architectural strengths of the IBM SP are reviewed from the application′s point of view.

5.2.1 Server Consolidation

The RS/6000 SP provides rich features for server consolidation. Multiple node types, which include SMP nodes, and thin and wide uniprocessor nodes, give the freedom of choice to users. The performance of a server can be increased more granularly based on users′ requirements.

The SP Switch has a 150 MB/sec bandwidth uni-directional (300 MB/sec bi-directional) and provides substantial scalability for parallel applications. As mentioned on page 113, SP provides a way to integrate the 12-way S70 and S7A machines into SP. Then the SP system will become a key building block of a server consolidation configuration.

In addition, as part of IBM′s strategy to build enterprise clusters using SP technology, the IBM Netfinity is supported as an NT node under the RS/6000 SP management domain. This support allows the SP control workstation to replace consoles for multiple nodes. It is also planned to support Netfinity connections to the SP through the SP′s high-speed switch. The SP-NT solution′s targeted

application areas include two-tier UNIX/NT hybrid applications and LAN consolidation.

Figure 24 shows the RS/6000 cluster of the future.

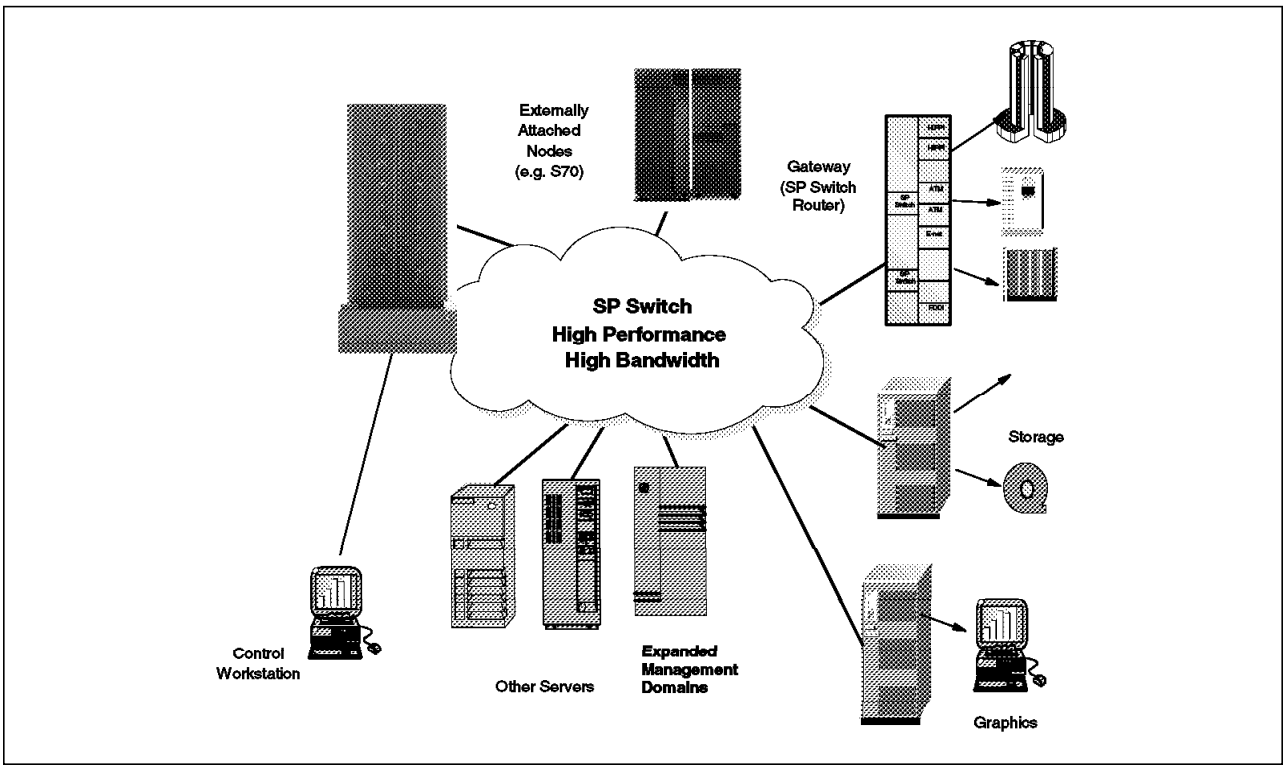


Figure 24. Future of the RS/6000 SP Cluster

The unique features of RS/6000 SP for server consolidation are:

- Inherent clustering architecture

  RS/6000 SP is built upon clustering architecture by its nature, minimizing additional investment in implementing enterprise-level clusters. This inherent clustering architecture helps to improve the availability of SP, compared to an SMP.

  In an SMP system, there is a limitation in availability due to its architecture, since a single copy of the operating system can represent a single point of failure.

- Single point of control

  The RS/6000 SP has a single control console that can be used to manage all the nodes within the SP (up to 512 instances of AIX).

- Scalability

  A customer often cannot forecast how large a system will be needed in the future for a given application. To minimize costs, the new application may initially be run on a small sized system. However, there needs to be a way to expand to a much larger sized system as the application grows.

  The SP can start with as few as one node and can be expanded to 512 nodes. The unpredictability of the business size becomes more evident in the e-business area, where a small company, once it succeeds, deals with customers worldwide through the Internet.

- Floor space saving

  The SP contains multiple AIX instances within a single frame. Therefore, consolidation of multiple UNIX servers onto a single SP can result in major savings in floor space.

- SP Switch

  Many modern client/server applications often consist of a large database server and multiple application servers, with heavy traffic between the database server and the application servers. As more applications move toward e-business they will require graphics, video/audio data as well as text, resulting in heavier message traffic between the servers. The SP Switch has the capacity and speed to handle this traffic. It can be also utilized as a dedicated, high-speed network to do backup/restore operations across nodes.

- Partitioning

  The physically partitioned, shared-nothing SP architecture guarantees the isolation of a node failure from the rest of the system. New nodes can be added without the need to power down and reboot the entire SP system. A failing node can be powered down and serviced without impact to the operation of the rest of the system. Unaffected applications can remain in operation, running undisturbed by the service action on the failing node.

  System partitioning within the SP allows us to divide the system into logically separate SP systems. This gives the user the ability to isolate SP environments from each other. For example, the user can create development, test and production partitions and thereby protect the production environment from interference by development and test activities.

5.2.2 ERP Using SAP R/3

SAP R/3 is well suited to the SP system for a number of reasons. For instance, it is usually beneficial to isolate some components of SAP from others. Isolating the R/3 application servers by running them on separate SP nodes can guarantee service levels to users and also provide redundancy within SAP R/3. The other benefits are:

- SP Switch

  In a 3-tier R/3 environment, a high-performance communication link between the database servers and the application servers is critical to SAP R/3 performance. The SP Switch is one of the best choices for this purpose, because of its high network bandwidth.

- ESCON attachment

  In many SAP R/3 implementations, migration to R/3 cannot be done all at once. Most users prefer a multistage migration strategy. In this case, some of the important legacy applications remain on mainframes and it is often required to provide a communication path between both systems. The ESCON channel attachment may be desirable in high-volume production environments, and SP can be directly attached to ESCON channels for this purpose.

- Single point of control

  As illustrated in Figure 25 on page 83, a typical implementation of SAP R/3 consists of a single large database server such as RS/6000 S70 and many smaller application servers. RS/6000 SP suits ideally in this case because a

large number of application servers can be managed by an SP′s control workstation.

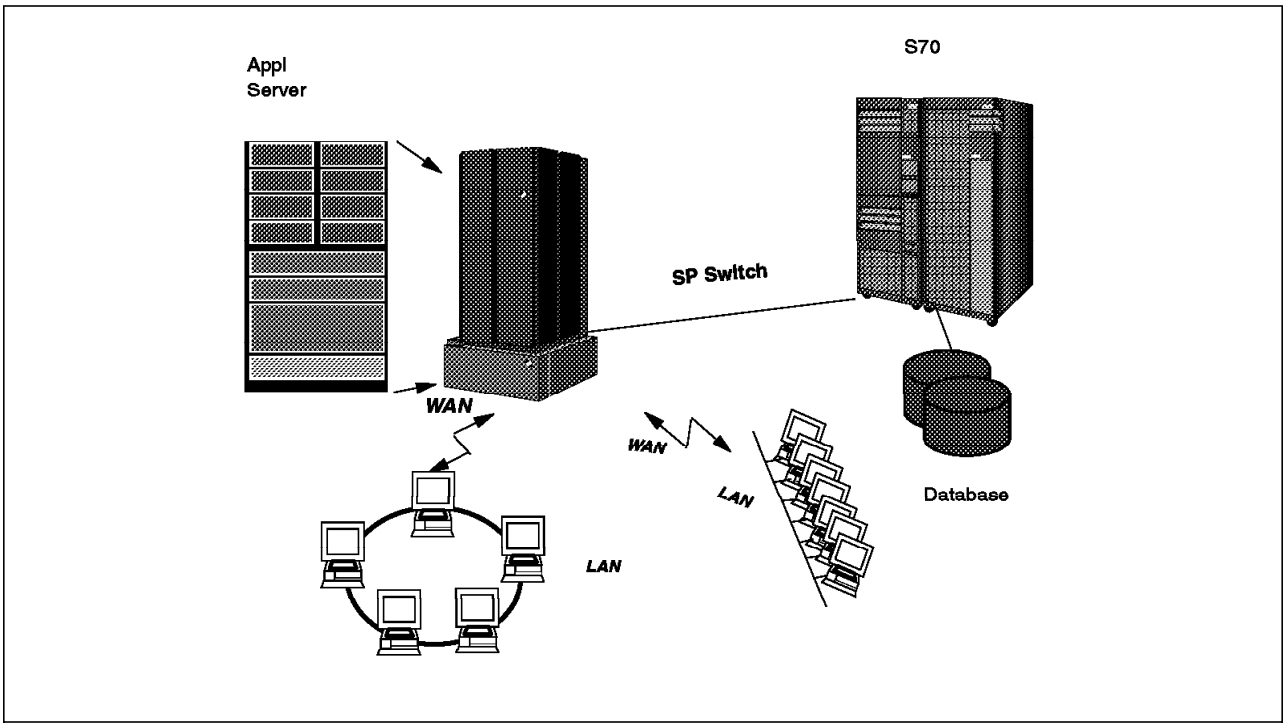


Figure 25. A Typical SAP R/3 Implementation with RS/6000 SP as Application Server

5.2.3 Enterprise Data Warehouse

Data Warehousing applications normally include analyzing large amounts of data and this can be very compute-intensive, requiring much more CPU power than the traditional OLTP applications. The advances in parallel database technology make it possible to process data in a data warehouse in such a way that scanning of a whole table can be done in parallel using multiple processors in a system.

The same parallel processing technique can be applied to join and sort operations. The idea is that since scan, join and sort operations consume much CPU power and memory, a large query operation is divided into many smaller subqueries, and those are dispatched to available processors. As the number of available processors is increased, the degree of parallelism can also be increased.

The technique is illustrated in Figure 26 on page 84.

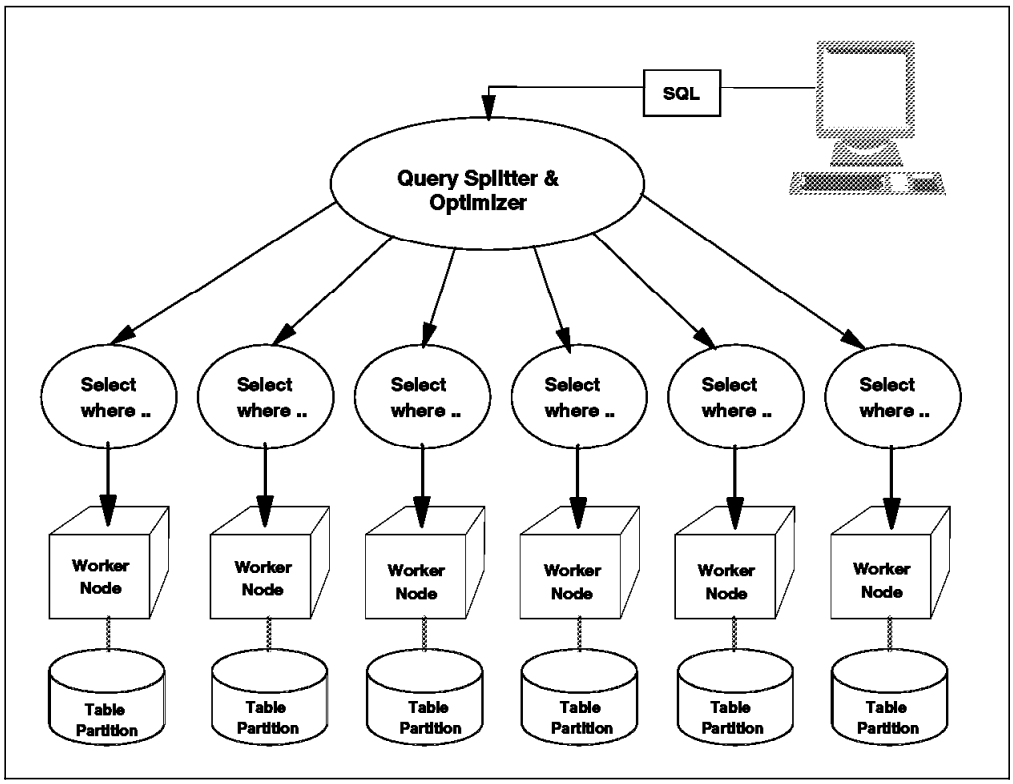


Figure 26. Example of Parallel Scan Used in Parallel Databases

Although most of the new SMP platforms can handle this kind of parallel processing, there will be a point where any single SMP solution will not be powerful enough to support a large data warehouse system. This can be because of either I/O or CPU bottlenecks.

The solution is to parallelize the query across multiple nodes of an SP, or multiple systems in a S/390 Parallel Sysplex.

The advantage of the the SP is its superior scalability. The larger the size of a database, the more processing power tends to be required, in terms of both the number of processors and the size of available memory.

The scalability of SP's MPP architecture allows it to compete very successfully with large SMPs especially in the case of large databases, where SMPs can have a limitation on the number of processors and the maximum size of memory. See Table 10 on page 28 and Table 11 on page 28 for the top five TPC-D 300 GB and 1 TB results.

5.2.4 Investment Protection

This benefit is not limited to any single application type and can be applied to all types of applications.

By design, large-scale SMP implementations are *symmetric*. Every processor in an SMP system must operate at the same speed (cycle time) as every other processor in the system.

In general, customers cannot add new technologies (such as faster processors or faster backplanes) to an existing SMP system in order to mix and match older

and newer technologies as they attempt to protect their investment in any given level of technology.

Implementing the next generation of technology to support a new or growing workload requires the simultaneous replacement of *all* processors, and is likely to demand some substantial amount of scheduled downtime (note, however, the exception of the S/390 Parallel Sysplex, where nodes can be upgraded or replaced without interruption to the application).

For example, let us suppose a customer purchased a 400 MHz SMP this year. Most vendors have announced plans for 500 MHz and faster processors in the near future. If the user wishes to take advantage of faster processors, he would be forced to replace *all* the current processors in the existing SMP, or to replace it completely with a new machine. The larger the configuration of the SMP box is, the bigger the cost and disruption the user will have to suffer.

Due to the shared-nothing architecture of SP, new technologies may be seamlessly added to the existing configuration as they become available without replacing or upgrading existing nodes, and may not even require any system down time.

5.2.5 Real Benchmark Experience

As we discussed in the previous chapters, there are always pitfalls in the standard benchmark suites, in spite of their virtues. Those standard benchmarks are highly tuned and conducted by dedicated specialists. In this section we discuss how a real production system can deviate from the Industry Standard benchmark and why this kind of distortion can happen.

To illustrate, we look at a detailed real benchmark case between IBM and a customer. In 1997, a customer wanted to have a benchmark test of SAP R/3 on the two competing platforms, where one was an IBM RS/6000 SP and the other a high-end UNIX box from a prestigious vendor (referred to as the “UNIX vendor” from now on). The purpose of the test was to measure the performance of database servers under the SAP R/3 environment.

The customer′s need was to reduce the turnaround time of his monthly cost analysis and cost allocation process, which used to be time-consuming, but had firm monthly deadlines that had to be met without fail. The SAP R/3 module to be used was CO (Cost Optimization). The task given to CO was a batch-oriented job and it would sometimes take more than 6 hours on the customer′s development system. Because the standard CO module of R/3 was used for the benchmark test and the tuning parameters of the R/3 system were kept the same on both systems, we could exclude the assumption that the benchmark code was prejudiced or highly tuned to the RS/6000.

The environments are compared in Table 14 on page 86.

Table 14. The Benchmark Environment			
Environment	**Component**	**IBM**	**UNIX Vendor**
H/W	DB server	8-way 604 SP high node (112 MHz)	4-way SMP (at its maximum)
	Application server	Several 135 MHz POWER2 wide nodes	Same as IBM
	Disks	IBM SSA disks	RAID 5 disks with large cache memory
S/W	O/S	AIX 4.1.4	*Kept confidential intentionally*
	SAP R/3	3.0D	Same as IBM
	Oracle	7.2.3	Same as IBM

When we compared the tpm-C values between the two boxes, the UNIX vendor's result was almost three times the IBM SP 8-way 604 high node result.

But the benchmark result was the reverse, as is shown in Figure 27.

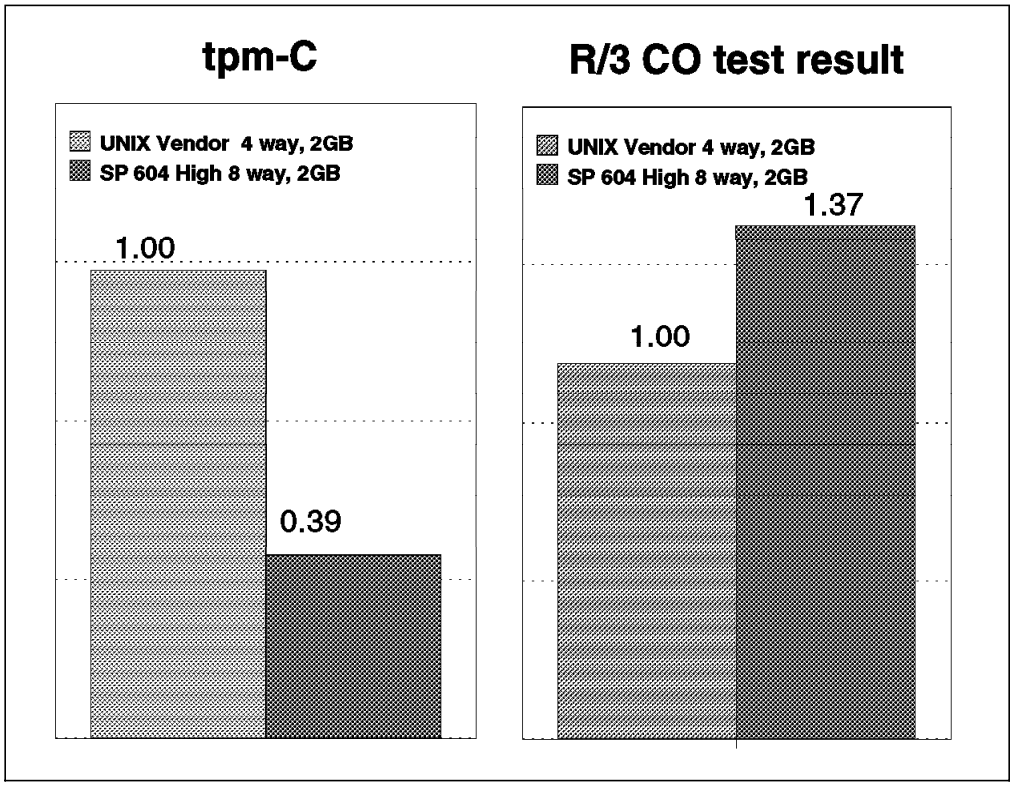


Figure 27. Customer Benchmark versus TPC-C

How could this happen? According to observations made during the test, the given tasks were more I/O-oriented than CPU-oriented. So even if the processors of UNIX vendor had higher performance than the SP nodes, it was highly probable that they did not contribute much to the overall performance. In addition, the vendor's system might have suffered from *write penalty of RAID 5*, while the SSA disks did not.

The lesson learned from this case is that the TPC-C result can be a misleading indicator for this kind of batch-oriented, I/O-intensive job. TPC-C does not tell us

anything about batch performance. It must be noted that almost all TPC-C test runs use more disks and larger memory sizes than a typical production system in order to ensure that the only bottleneck is the CPU, even though the benchmark is supposed to emulate typical commercial workloads.

IBM experience, based upon decades of dealing with commercial workload management, tells us that the really valuable system is one that is well-balanced in terms of processor speed, processor-to-memory bandwidth, cache size and I/O bandwidth. But “well-balanced” is again an abstract expression, and no standard benchmark exists to measure “well-balancedness.”

To conclude, the standard benchmarks which are published provide us probes into a system from various viewpoints, and the results of these must be regarded as indicators of certain aspects, not as a barometer of the whole picture.

5.3 Total Cost Of Ownership

Industry Standard benchmarks attempt to measure and compare platform price and performance. They do not pretend to measure all of the functional capabilities of an architecture, even though they may be very important in platform choice.

The implication of the price comparison in the TPC benchmarks is that all costs have been considered. This is plainly not true. The TPC disclosure reports include a cost per transaction that is based on:

- Processor costs
- Disk costs
- Cost of other peripherals
- Software costs
- Maintenance
- Planned downtime
- Unplanned downtime

It is well understood that these items form only a small proportion of the whole-life costs associated with any IT platform. In addition to the preceding costs, these whole-life costs will include:

- Application development
- Application maintenance
- Systems integration
- Software distribution
- Network tariffs
- Network management
- Support staff
- Change and problem management
- User training and support
- Security and access management
- Facilities management

The sum of these costs will vary from platform to platform, but it is well known and documented that these costs are far higher in a networked or decentralized application than in a centralized solution.

It is also true that many of the costs of a decentralized solution will apply to an application running on a platform which is centrally located but networked in

character (for instance, a centrally located set of multiple loosely coupled systems running the same application).

These additional overheads are well-enough documented elsewhere that they need not be repeated here. Suffice it to say that they will greatly outweigh the price/performance element as measured by Industry Standard benchmarks.

Further references to the total cost of computing can be found in the bibliography in this book.

That list is far from exclusive. Recently, there has been a proliferation of consultant's papers comparing the total costs of decentralized systems with traditional mainframe systems. While the details of these papers may differ, they reach the common conclusion that IBM Enterprise Systems offer the cheapest overall cost of computing.

Appendix A. NotesBench Benchmarks

The nine different NotesBench workloads attempt to provide the using organization with data that will assist in sizing and planning for the extremely diverse set of workloads that may exist in a Domino environment.

A.1 Idle

This workload establishes an upper bound on the number of sessions (which do nothing) that a Domino server can support. After establishing sessions between client and server, the test carries out no Notes transactions in the sessions. Other than the resources required to start a session, no other resources are used in the test. The resultant capacity metric is the maximum number of sessions that can concurrently exist.

The script says the following:

1. Wait for other scripts to finish initialization (pause 0 to 3 minutes).
2. Open the desired number of sessions.
3. Wait for other systems (if any) to open additional sessions (pause 20 to 30 minutes).
4. Close all opened sessions.

A.2 Mail

The following script models an active user who is reading and sending mail. It contains an average of 15 minutes of waiting, so an average user will execute this script no more than four times per hour. For each iteration of the script, there are five documents read, two documents updated, two documents deleted, one view scrolling operation, one database opened and closed, one view opened and closed, and some miscellaneous operations. In sending mail messages, each user sends a mail message to three recipients no more than once every 90 minutes. Max users is the output of the test based on each user performing this workload, along with response time and transactions per minute.

This script says the following:

1. Open mail db.
2. Open a view (mail in-basket) containing 100 mail messages initially.
3. Read view (in-basket for 5 to 10 seconds (wait).
4. Open five documents in mail file (notes) and read each (wait for 10 to 20 seconds while reading each).
5. Categorize two of the documents (file in a notelog).
6. Compose two new mail memos or replies (wait 1 to 2 minutes for each).
7. Send a 1K mail memo to three recipients every sixth time this script is executed/per user (mail sent no more than once every 90 minutes). This memo is sent to random users selected from a Name and Address Book (NAB) like the CALLUP directory. The NAB contains entries for the maximum number of users being tested, that is, 2100 for a run attempting to get 2100 concurrent/active users with <5 second response time.

8. Look at the view (in-basket) and mark a few documents for deletion (wait 5 to 10 seconds).
9. Delete the documents marked.
10. Close the view (mail in-basket).
11. Pause 5 to 15 minutes for a meeting in the office.
12. Repeat.

A.3 Shared Discussion DB

This workload models a server for active users, who are only performing heavy shared database operations. The test includes view operations in a shared database, navigation of unread documents, additions, and updates to documents in a shared database. It applies especially to sites that heavily utilize the collaborative features of Domino. The results for the Shared Discussion Database test are:

- Throughput of completed Notes operations
- Maximum users supported
- Average response time at maximum capacity

The throughput for this test is a capacity metric. It is the maximum number of active users that can be supported before the average user response time becomes unacceptable. The following script models an active user who is reading, composing, and updating documents in a shared database; it says the following:

1. Pause at a random interval so multiple processes are staggered well (pause 0 to 3 minutes).
2. Loop.
    a. Open a discussion database.
    b. Open the current view.
    c. Wait 5 to 10 seconds to peruse the view.
    d. Page down the view two times, and spend 3 to 10 seconds to read each window.
    e. Set the unread list to a randomly selected 30 documents.
    f. Open the next three unread documents and read each for 10 to 30 seconds.
    g. Update three documents with some changes.
    h. Close the view.
    i. Pause at desktop for 4 to 8 minutes for a meeting in the office.
    j. End loop.

A.4 Mail DB

This script models an active user who is reading mail, sending mail and reading a shared database. It contains an average of 15 minutes of waiting, so an average user will execute these events no more than four times per hour. For each iteration of the script, there are eight documents read, two documents updated, two documents deleted, four view scrolling operations, two databases opened and closed, two views opened and closed, and some miscellaneous operations. In sending mail messages, each user sends one mail message to three recipients approximately once every 90 minutes. Max users is the output of the test based on each user performing this workload, along with response time and transactions per minute. This script says the following:

1. Open the mail db.
2. Open a view (mail in-basket) containing 100 mail messages initially.
3. Read the view (in-basket for 5 to 10 seconds (wait).
4. Open 5 documents in the mail file (notes) and read each (wait for 10 to 20 seconds while reading each).
5. Categorize two of the documents (file in a notelog).
6. Compose two new mail memos or replies (wait 1 to 2 minutes for each).
7. send a 1 K mail memo to three recipients every sixth time this script is executed/per user (mail sent no more than once every 90 minutes). This memo is sent to random users selected from a Name and Address Book (NAB) like the CALLUP directory. The NAB contains entries for the maximum number of users being tested, that is, 2100 for a run attempting to get 2100 concurrent/active users with <5 seconds response time.
8. look at view (in-basket) and mark a few documents for deletion (wait 5 to 10 seconds).
9. Delete the documents marked.
10. Close the view (mail in-basket).
11. Pause at the desktop for 4 to 8 minutes for a meeting in the office.
12. Open a discussion database.
13. Open the current view.
14. Look at the view (wait 5 to 10 seconds).
15. Page down the view two times and spend 3 to 10 seconds to read each window.
16. Set the unread list to a randomly selected 30 documents.
17. Open the next three unread documents and read each for 10 to 30 seconds.
18. Close the view.
19. Pause at the desktop for 4 to 8 minutes for a meeting in the office.
20. Repeat.

A.5 Groupware

This script models a server for experienced Notes users who are sending large mail messages, adding documents with attachments to shared databases, performing full-text searches, and replicating changes from their local machine to the server. The Groupware test includes the Mail DB activity *plus* the following:

- Mail messages with 300 KB body fields
- Mail messages with 500 KB attachments
- Users that replicate with the system under test
- Users that execute full-text searches of a shared discussion database.

Groupware is a capacity test for Notes users that process large amounts of information. This workload models sites that use the most resource-intensive features of Notes. It can be used to establish a worst-case lower bound on the maximum number of users a server can support at a desired response time.

This script is similar to the Mail DB script, but includes more operations and is more resource-intensive. With Mail DB:

- Each view scrolling operation is for 40 rows rather than 20 rows.
- One of the two mail messages that are sent includes a large body field, which models a message with a large imported file. The other message has a large attachment.
- The script makes changes to a local database and pushes those changes to the server.
- The script performs a full-text search of a discussion database.

The Groupware workload contains an average of 15 minutes of waiting, so an average user will execute these events no more than four times per hour. Max users is the output of the test based on each user performing this workload, along with response time and transactions per minute. The script says the following:

1. Open the mail db.
2. Open a view (mail in-basket) containing 100 mail messages initially.
3. Read the view (in-basket for 5 to 10 seconds (wait).
4. Open five documents in mail file (notes) and read each (wait for 10 to 20 seconds while reading each).
5. Categorize two of the documents (file in a notelog).
6. Compose two new mail memos or replies (wait 1 to 2 minutes for each).
7. Send a 300 K mail memo to three recipients every sixth time this script is executed/per user (mail sent no more than once every 90 minutes). This memo is sent to random users selected from a Name and Address Book (NAB), like the CALLUP directory. The NAB contains entries for the maximum number of users being tested, that is, 2100 for a run attempting to get 2100 concurrent/active users with <5 second response time.
8. Look at the view (in-basket) and mark a few documents for deletion wait 5 to 10 seconds).
9. Delete the documents marked.

10. Close the view (mail in-basket).
11. Pause at the desktop before switching to a local database (wait 10 to 30 seconds).
12. Change to a local database.
13. Add two notes to the local replica, then push the changes to the server.
14. Pause at the desktop for 1 to 5 minutes for a meeting in the office.
15. Change to a server discussion database.
16. Open a discussion database.
17. Open the current view.
18. look at view (wait 5 to 10 seconds).
19. Page down the view two times and spend 3 to 10 seconds to read each window.
20. Perform a full-text search of the view for either of two random words.
21. Perform another full-text search of the view.
22. Set the unread list to a randomly selected 30 documents.
23. Open the next three unread document and read each for 10 to 30 seconds.
24. Add a new document with a 500 K attachment.
25. Close the view.
26. Pause at the desktop for 215 TO 455 seconds for a meeting in the office (so script is executed no more than four times per hour).
27. Repeat.

A.6 Calendaring and Scheduling

This workload stresses a Domino server's ability to process Calendaring and Scheduling (C&S) invitation requests. Each invitation is sent to multiple recipients. This causes the following work to be performed:

- Domino Mail Router sends invitation requests to the recipient's system under test (SUT) Free Time Database. Any C&S activity, such as the creation of appointments, alarms, or invitations, requires an update to the Free Time database located on the user's home mail server.

The resultant capacity metric for a C&S and mail-only server is the maximum number of users that can be supported before the average response time becomes unacceptable. The results for the C&S test are:

- Throughput of completed Notes operations
- Average response time at maximum capacity
- Maximum C&S users supported

The following script models an active user who is creating invitation requests. The user will create an invitation with a message size of 1 K and send that invitation to three recipients. The invitation date is a randomly chosen date between today and 90 days from today. The invitation time is always some future time from the child's system time. The invitation specifies a meeting duration of 15 minutes. The script says the following:

1. Pause a random interval so multiple processes are staggered well (pause 0 to 3 minutes).
2. Loop: Invite three people from the local Name and Address Book to a meeting that will last 15 minutes (this sends a 1 K mail message to each recipient; this is executed only once in every six times through the loop).
3. Pause 1 to 2 minutes.
4. End loop.

A.7 Web Walker

This workload determines the maximum number of concurrent HTTP users that a Domino HTTP server can support. This workload simulates a Web user′s browsing a Web site built on top of Domino. The user will peruse each link on the selected test database retrieving the *full content* of each Web page, thereby providing a more realistic load against the server. Many HTTP tests simply connect and disconnect (never transmitting any data such as graphics). The resultant capacity metrics are:

- The maximum number of HTTP server sessions supported.
- Total number of HTTP server timeouts (that is, server not responding).
- Total number of Drop connections (connection was lost in the middle of a transmit).
- Average time to fetch the full content of a Web page (this is different from rendering the graphics of a page).
- Average number of bytes per page.
- Total pages walked.

  The script says the following:

- Loop.
  1. Web-walk against a standard Notes database (supplied) served by Domino as HTTP.
  2. End loop.

A.8 Replication Hub

A *replication hub* is a Notes server that exists to propagate changes in Notes databases among a collection of other servers. The workload for a replication hub consists of replicating changes to user databases. Typical replication hubs also have some amount of replication load for the Name and Address Book (NAB) database, but that is not included in the NotesBench replication hub workload. The results for the replication hub test are:

- The throughput metric: the number of documents replicated per unit of time
- The average response time
- The number of spoke servers supported

The test procedure uses a hub-and-spoke topology. A number of driver systems serve as source and destination spokes, and the system under test serves as a single replication hub. The spoke servers modify local replicas databases and then replicate those changes to the hub server. Each test script user makes

changes to one local database and then replicates those changes to the server. The server runs the replicator and updater, but no other server programs. The test executes the following database modifications on the spoke systems: additions, updates, categorizations, and deletions. The script says the following:

1. Pause at a random interval so multiple processes are staggered well (pause 0 to 3 minutes).
2. loop
    a. Update the local database replica of the shared discussion database. Note that there are no pauses between operations since these functions are done to the local database.
    b. Change a few fields in some random notes.
    c. Delete a few random notes.
    d. Add some new notes with random data.
    e. Replicate.
    f. Pause before doing another cycle (pause 5 to 10 minutes).
3. End loop.

A.9 Mail Routing Hub

A *mail routing hub* is a server that exists to route messages to other servers (a "pure" router) and possibly to deliver messages to local users. The workload for a mail routing hub (the system under test) consists of receiving messages from source driver systems and routing or delivering each message to a destination system. The test results for the mail routing hub are:

- The throughput metric: the number of messages processed (routed or delivered) per unit of time
- Messages transferred to recipients per hour
- Message bytes transferred to recipients per hour

There are two kinds of mail routing hubs. One is a pure router forwarding messages to other servers with no local delivery. The other kind of routing hub routes messages to other servers as well as delivering messages to local users. The selection of destination systems determines the amount of local versus remote delivery. For local delivery, make the system under test a destination system.

The following script generates the input message traffic to the system under test for both kinds of mail hub tests. The test selects message recipients with a uniform distribution from the People view in the driver's Name and Address Book.

Note that this script is not intended to be a realistic simulation of any kind of human message generation. Instead it is intended to generate a continuous input stream of messages to the system under test acting as a mail hub router.

The result metric of the Mail Routing Hub test is the maximum throughput of the router, measured on the system under test. It is up to the testing organization to configure the number of source drivers that produce the optimum load on the test system.

1. Pause a random interval so multiple processes are staggered well (pause 0 to 3 minutes).
2. Loop.
    a. Send a 1 K mail message to three recipients.
    b. Pause 5 to 10 seconds.
3. End loop.

Appendix B. More Information about the TPC

The Transaction Processing Council (TPC) is a non-profit organization whose mission is to “define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry.”

Benchmarks are selected by the members of the TPC according to the following criteria:

- Applicability - Is it relevant to a large number of users?
- Comparability - Will it lead to objective comparisons?
- Understandability - Can users/press understand its significance?
- Executability - Can it be run in a reasonable time period and for a reasonable cost?

Each benchmark generally has two primary metrics:

- Throughput - maximum throughput expressed in transactions per unit of time (second, minute, hour)
- Cost of ownership per transaction per second, minute, or hour

The cost of ownership is based on purchase and 5 years maintenance costs for all hardware and software required to achieve the maximum throughput rate.

The council membership consists mostly of system vendors and many database vendors. However, there are a small number of users and consultants.

The members of the TPC include[11]:

- Acer
- Angara Database Systems
- BEA
- Bull
- Compaq
- Data General
- Dell
- EDS
- EMC
- Fujitsu
- Hewlett-Packard
- Hitachi
- IBM
- Informix
- Intel
- ITOM International Corporation
- Microsoft
- Mylex
- NEC
- Netscape
- OKI Electric
- Oracle
- Progress Software

[11] See *http://www.tpc.org* for the current membership list.

- Pyramid
- SCO
- Sequent
- Siemens
- Silicon Graphics
- Sun
- Sybase
- Toshiba
- Unisys
- White Cross Systms

While the TPC 37 members, the day-to-day work is handled by a number of subcommittees and workgroups with members from a subset of the participating vendors.

These are:

- TPC Steering Committee (SC)

  The SC consists of 5 representatives who are elected by the full TPC Council and oversee the overall direction of the TPC.

- Technical Advisory Board (TAB)

  The TAB is a standing committee formed to hear arguments on both sides of interpretation/compliance issues and to make recommendations to the Council, which then makes binding decisions at regular meetings. In addition, the TAB is responsible for providing analysis, definition, and recommended resolution to requests for specification interpretations, and compliance questions of published results. The 7 TAB members are elected by the full TPC Council.

- Public Relations Subcommittee

- Technical Subcommittees

  These subcommittees are responsible for taking an initial draft benchmark specification and driving it through the development and approval process to completion.

Membership is on a voluntary basis. Each TPC member is entitled to one voting representative on each of the technical subcommittees.

Most motions of the Full Council, SC, and Subcommittees require a two-third majority to pass, with only one vote per member company. Thus IBM, with representatives to the TPC from Austin, Poughkeepsie, Rochester, and Toronto, representing a very broad product line, has the same vote as EMC, which just sells disk, or as a PC vendor supporting only Windows NT.

B.1 Benchmarking Versus Benchmarketing

By the spring of 1991, the TPC was clearly a success. Dozens of companies were running multiple TPC-A and TPC-B results. Not surprisingly, these companies wanted to capitalize on the TPC′s cachet and leverage the investment they had made in TPC benchmarking. Several companies launched aggressive advertising and public relations campaigns based around their TPC results. In many ways, this was exactly why the TPC was created: to provide objective measures of performance. What was wrong, therefore, with companies wanting to brag about their good results? What was wrong is that there was often a large gap between

the objective benchmark results and their benchmark marketing claims: This gap, over the years, has been dubbed “benchmarketing.”

Out of these Council debates emerged the TPC′s Fair Use policies adopted in June, 1991. When TPC results are used in publicity, the use is expected to adhere to basic standards of fidelity, candor, and due diligence. These qualities together add up to and define Fair Use of TPC results and are defined as follows:

Fidelity Adherence to facts; accuracy

Candor Above-boardness; needful completeness

Due Diligence Care for integrity of TPC results

With the creation of a good review and fair use process, and with dozens of companies publishing regularly on the TPC-A and TPC-B benchmarks, the TPC may be forgiven for lapsing into a self-satisfied belief that the road ahead was smooth.

That sense of well-being was torpedoed in April, 1993 when the Standish Group, a Massachusetts-based consulting firm, charged that Oracle had added a special option (discrete transactions) to its database software, with the sole purpose of inflating Oracle′s TPC-A results. The Standish Group claimed that Oracle had “violated the spirit of the TPC” because the discrete transaction option was something a typical customer would not use and was, therefore, a benchmark special. Oracle vehemently rejected the accusation, stating, with some justification, that they had followed the letter of the law in the benchmark specifications.

The benchmarking process, which sprang from the discredited TP1 and DebitCredit days, has always been treated with a fair degree of skepticism by the press. So the Standish Group′s charges against Oracle and the TPC attracted broad press coverage. Headlines like the May 17, 1993 issue of Network World were not uncommon, “Report Finds Oracle TPC results to be misleading; says option discredits TPC -A as benchmark.”

B.1.1.1 New Anti-Benchmark Special Prohibition

The TPC benchmark rules had always required companies to run the benchmark tests on commercially available software. However, after the Standish Group charges, the Council realized that it had no real protection from companies that purposely designed a benchmark special component into their commercially available software. In other words, this special component could be buried in some obscure corner of overall product code and only be used when the vendor wanted to run a TPC test.

If the TPC was formed to create fair, relevant measures of performance, then yes, the benchmark special was a violation of the TPC′s spirit and thus had to be prohibited.

In September, 1993, the Council passed Clause 0.2, which contains the sweeping prohibition against benchmark specials that has become a bedrock of the TPC process to ensure, relevant benchmarks

Specifically prohibited are benchmark systems, products, technologies or pricing, whose primary purpose is performance optimization of TPC benchmark results without any corresponding applicability to real-world applications and environments. In other words, all “benchmark special” implementations that

improve benchmark results but not real-world performance or pricing, are prohibited.

Clause 0.2 in TPC-A and TPC-B went into effect in June, 1994. Oracle decided not to test its discrete transaction option against the new anti-benchmark special rules in the specifications and withdrew all of its results by October, 1994. Let it also be noted that Oracle remains a TPC member and strong supporter of the organization.

B.2 Avoiding Unfair Use of TPC Results

The Transaction Processing Performance Council (TPC) has written policy regarding the fair use of TPC results. Violating any of these leads to public censure of the violating vendor by the council. Serious and repeated violations can lead to monetary fines and the potential for issuance of negative press release by the TPC.

TPC members are encouraged to publicize their involvement in the TPC, including the publishing of TPC results. However, all members must follow the guidelines for publishing TPC information. These guidelines include:

1. All references to TPC benchmark names should be accompanied by the TPC trademark (for example, TPC Benchmark C, TPC-C, or derivative).
2. No TPC member shall publish results that imply or suggest that they are TPC-compliant, when they are not. TPC-compliant results are those results where a full disclosure report has been submitted in advance to the TPC Administrator. Therefore, “estimate” or “extrapolated” TPC results are not allowed by TPC policy.

The Policy For Fair Use of TPC Results was adopted at the 16th TPC General Meeting, and amended at the 25th General Meeting in 4/93): “The TPC actively encourages Test Sponsors to widely distribute their TPC benchmark results in publicity. This is, after all, the culminating benefit and purpose of conducting TPC benchmarking.”

This “Policy for Fair Use” states how TPC results may be fairly used in publicity. The TPC label is intended to be applied to only fully legitimate TPC results, used in a fair manner. Reliance on the TPC, its benchmarks, and the large collection of TPC benchmark results is directly dependent on this.

Publicity includes press releases, advertisements, commercials, and any and all marketing materials, literature, and collateral that are obtainable without a nondisclosure statement. Publicity includes spoken as well as written communication; for example, a spokesperson speaking in an open forum such as a press conference is bound by the policy.

When TPC results are used in publicity, the use is expected to adhere to basic standards of fidelity, candor, and due diligence, the qualities that together add up to, and define, Fair Use of TPC Results.

Because TPC results are protected by the TPC Trademark, this policy applies to all parties who use TPC results, including but not limited to members of the TPC. The intent is simple: if you want to use the TPC name, you are requested to follow this policy. Otherwise, do not mention or imply the TPC.

Fair Use is required for all publicity that uses TPC results extracted from TPC Full Disclosure Reports. This applies to publicity that makes explicit TPC references as well as to that which makes implicit references, insofar as a reasonably aware reader would connect it to the TPC.

If publicity uses TPC results for a system, it must explicitly include the "TPC Performance Pair" for that system. This consists of the throughput and price/performance of the system, as taken from the TPC.

- It is Fair Use for publicity to:
  - Use TPC results, as long as the following conditions are met:
    - The TPC Performance Pair is included.
    - A Full Disclosure Report for the results is complete and on file with the TPC Administrator.
    - Appropriate attribution is given to the TPC Trademark.
  - Freely compare and contrast sets of TPC Performance Pairs, from whatever set of Test Sponsors.
- It is Unfair Use for publicity to:
  - Use estimated results that refer to the TPC or TPC workloads or to compare them to TPC results.
  - Use less than the complete TPC Performance Pair; i.e., throughput without price/performance, or vice versa.
  - Display or use one part of the TPC Performance Pair without the other.
  - Display TPC results without specifying that these results are current as of a certain date; if applicable, a single date can be displayed for an entire range of results.
  - Use TPC results, from whatever source, unless the relevant Full Disclosure Report is on file with the TPC Administrator.
  - Use TPC results without an attribution to the TPC Trademark.
  - Use TPC Partial Information without showing the associated TPC Performance Pair.
  - Compare one system′s total price to the partial price of another system, or to compare partial price. TPC results cannot be generated with less than the entirety of the configured system.
  - Show TPC results with non-TPC results in a manner that may cause the reader to believe that non-TPC results are in fact TPC results.
  - State or imply that another Test Sponsor′s TPC results are considered invalid or in any way suspect by the TPC, unless the TPC has publicly stated this to be the case. All TPC internal review proceedings are confidential, and attempts to disclose these review proceedings or their results without TPC authorization are unfair.
  - Compare the price or price/performance of the TPC results when the currencies used in the results are not alike (e.g. dollars vs. pounds, or pounds vs. francs).
  - Compare price or price/performance of TPC results when the comparison based on a direct conversion of different currencies (e.g.,

converting dollars into pounds is based on a simple dollars-to-pounds conversion rate published in a newspaper).

- Refer to a withdrawn result without specifically stating that the result is withdrawn and no longer represents an official TPC result.

The following list of simple rules illustrate the above points:

1. Do not label or refer to estimated results as TPC results.
   Wrong:
   1250 TPM (for estimated tpmC results) is not 1250 tpmC (for published results).
2. Do not compare estimated results to published TPC-A/B/C results.
   Wrong:
   RS/6000 R30 = 1250 TPM, HP 9000 H70 = 1290.9 tpmC.
3. Do not use use just the performance or price/perf alone; use the pair.
   Wrong:
   RS/6000 59H = 1122.3 tpmC RS/6000 390 = $ 768/tpmC
   Right: RS/6000 59H = 1122.3 tpmC, $ 968/tpmC.
4. Do not compare one system′s total price to the partial price of another system.
   Wrong:
   RS/6000 59H = 1122.3 tpmC, $ 968/tpmC; entry price $ 50K
   HP 9000 H70 = 1290.9 tpmC, $ 961/tpmC; total cost $ 940K
5. Do not show TPC results and non-TPC results together on the same page or chart in order to lead readers to think non-TPC results are in fact TPC results.
   Example: Chart with estimated TPM for IBM systems and published TPC-C for competitive systems.
6. Do not refer to a withdrawn TPC result without explicitly stating that it was withdrawn.
7. Do not use TPC results without an attribution to the TPC trademark.

B.2.1.1 The ACID properties

The Atomicity, Consistency, Isolation and Durability (ACID) properties of transaction processing systems must be supported by the System Under Test (SUT) during the running of the TPC benchmarks.

These properties are defined as:

Atomicity Database transactions must be atomic; the system must perform all individual operations on the data, or ensure that no partially completed operations leave any effects on the data (that is data integrity must be preserved).

Consistency Each transaction must take the database from one consistent state to another, assuming that the database was originally in a consistent state.

Isolation Operations of concurrent database transactions must give results which are indistinguishable from results that would be obtained if each transaction was serially executed.

Durability The system must be able to preserve the effects of committed transactions and ensure database consistency after recovery from a failure.

These properties define various categories and levels of isolating a database transaction from a variety of programming and system faults. ACID tests are included in the benchmark to insure that the system tested supports a minimum level of fault protection required by a business. This insures that streamlined systems that perform well on a benchmark but are unreliable in production are not used. The ACID tests are conducted by the auditor and are a test of functionality, not of performance. Failure to pass one or more of the ACID tests will disallow publication of the results.

Appendix C. IBM SP MPP Architectures

To gain a broad understanding of the technologies being exploited these days to build high-end servers, it is helpful to also understand some of the most common architectures in use in the computer industry.

These architectures are:

- RISC and superscalar architecture
- Symmetric Multiprocessor (SMP) technology
- Massively Parallel Processing (MPP) technology

These concepts are explained and compared in the following sections.

C.1 Introduction to RISC Technology

The following sections provide an overview of RISC technology.

C.1.1 Pipeline Architecture

A *pipeline* is a hardware feature, similar to an assembly line, designed to increase instruction throughput through internal parallelism. Different units of the CPU perform, in parallel, the various operations required for fetching, decoding, and executing instructions. Several instructions can be executed in the CPU at the same time. The instructions go along the pipeline stages, in synchronization with the CPU clock. This means that, if everything goes well, each time a new instruction enters the pipeline, an older one is exiting it. This results in one instruction per pipeline stage and per cycle. Thus, although the time it takes to complete each instruction is not directly affected, pipelining increases the overall rate at which instructions complete.

When pipelining works as intended, performance is optimized. However, there are some potential problems: branch instructions and data conflicts. A pipeline normally holds a number of instructions in different stages of execution. Consider the case where one of these is a conditional branch, dependent on the condition code to be produced by a not-yet-executed instruction coming through the pipeline. Should it later turn out that the branch is to be taken, the system has to discard all the instructions prefetched after the branch and continue from the branch target address instead. A “bubble” in the pipeline develops, leading to wasted CPU cycles.

A true data dependency arises when an instruction entering the pipeline needs the result still to be produced by an instruction further ahead in the pipeline. This case cannot be resolved by register renaming, the technique employed to avoid data conflicts. The younger instruction simply has to wait on the older one to produce the result.

While true data conflicts are uncommon, branches are frequently encountered. In fact, branch instructions constitute about 20 percent of the instructions in most computer architectures. Branch target prediction as used in the RS/6000 alleviates the problem to a certain degree. The basic problem that remains is that very complex software, like kernel code and database systems, suffers a slowdown of CPU speed in the pipeline because of the high percentage of conditional branch instructions that are typical for these environments. Simpler applications are less affected by this problem.

C.1.2 Superscalar Architecture

Next came the idea of making several pipelines in order to implement further parallelism. This is called *superscalar architecture*. The instructions had to be distributed between the different pipelines and no more sequential treatment was possible. That is why compilers are so important in the RISC superscalar architecture: complexity no longer lies in the instruction itself, but in the compiler. But the advantage of a compiler is its ability to be optimized continuously, quickly, and much more easily than hardware code. Superscalar implies several independent execution units, like branch units, fixed-point units or floating-point units.

Superscalar allows more than one instruction to complete in a clock cycle. The objective is to achieve the highest number of instructions per cycle.

While the superscalar architecture aims at issuing more than one instruction per cycle, this goal is achieved only when the proper mix of instructions and data is sent through the pipeline. Some benchmarks will perform at several instructions per cycle, but the throughput might go down to less than one in other applications. This has nothing to do with instruction length, since the processor can handle the large percentage of floating-point instructions typical of technical and scientific applications. Actually, this instruction mix promotes parallelism, since load and store operations and loop counting are handled by the fixed-point unit. The challenge for superscalar, highly pipelined RISC architectures lies in complex commercial applications that use the fixed-point and branch units only. These applications tend to have very short sequential execution paths and poor locality.

Figure 28 shows a model of a three-pipelined architecture, the independent processor units being the branch, the fixed-point, and the floating-point processor units.

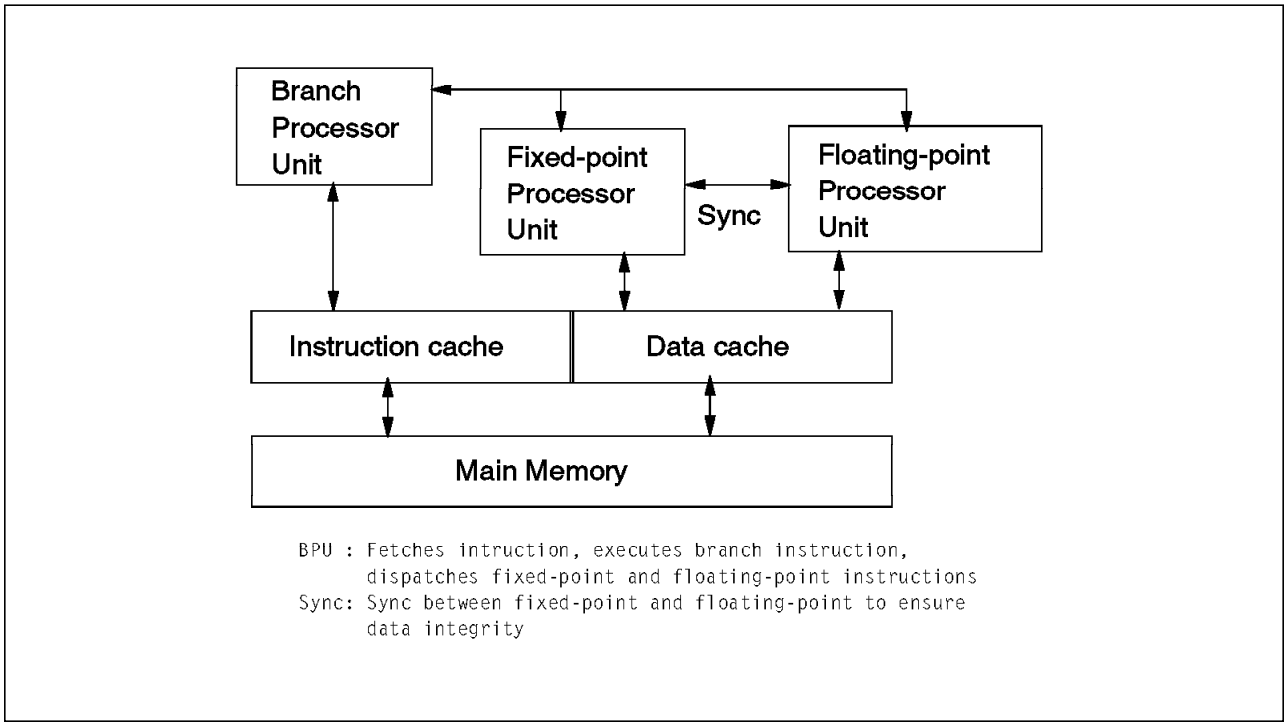


Figure 28. Pipelined Architecture

C.2 IBM RS/6000

The following sections describe IBM RS/6000 concepts and architecture.

C.2.1 POWER Architecture

This architecture was developed to address the mid-range requirements of the IBM AIX family supporting engineering-scientific and commercial application environments. It allows balanced performance between fixed-point and floating-point and provides a near-vector computer performance in numerically-intensive applications, while still performing well on scalar codes.

These features are achieved by allowing parallel execution of distinct functional units. This enables the execution of as many instructions per cycle as the processor implementation and the compiler technology allow. Compound function instructions execute in a single cycle in their respective units, but they perform two or more basic RISC instructions.

The performance results achieved with the compiler technology and the POWER implementation were a good starting point for the future development of the POWER microprocessor family. Compared to CISC contemporary microprocessors, the POWER architecture has reduced the number of cycles on the instruction set. Analysis of compiled code for various commercial and scientific applications has shown a path length for the instruction set that is less than or equal to those of CISC processors. The performance of this implementation approached that of large traditional mainframes and the early generations of supercomputers.

C.2.2 PowerPC Architecture

This architecture uses the POWER design as a base, exploiting its single-chip capabilities and reducing the system′s cost. The ability to maintain compatibility across a broad spectrum of single-chip and multichip implementations is part of IBM′s “Palmtops to Teraflops” strategy.

The PowerPC architecture was developed focusing on changes in the existing POWER architecture that would allow more aggressive superscalar implementations, allow higher clock rates, and require less silicon. The IBM, Apple, and Motorola alliance defined some goals for the architecture:

- Make it simple enough to allow a processor design that has a very short cycle time.
- Minimize effects that hinder the design of aggressive superscalar implementations.
- Enable a broad range of implementations, from low-cost controllers to high-performance processors.
- Include multiprocessor features.
- Define a 64-bit architecture that is a superset of the 32-bit architecture, providing binary compatibility for 32-bit applications.

The PowerPC 604 microprocessor family is a 32-bit implementation of the PowerPC architecture. The early 604 implementation reaches new levels of performance by issuing *four instructions per cycle*, thus achieving balanced execution of integer and floating-point operations.

The 604 uses a superscalar design to provide six independent execution units: one branch unit, three fixed-point units, one floating-point unit, and a load/store unit. The 604 chip also uses dynamic branch prediction techniques to enhance instruction pre-fetching, as well as speculative execution techniques to take advantage of the improved instruction pre-fetching and multiple execution units.

The PowerPC 604e followed PowerPC 604. The enhancements to PowerPC 604e include:

- Size of instruction and data cache have been doubled
- Higher clock frequencies
- A built-in performance monitor

C.3 Architecture of RS/6000 S70

The Model S70 is IBM′s RS/6000 symmetric multiprocessing (SMP) server for high-end commercial performance. In addition, it is ideally poised for the future in that it supports both 32-bit and 64-bit commercial applications running concurrently. It uses the newly designed PowerPC RS64 processor and is offered in 4-, 8-, and 12-way configurations. The S70 is the first RS/6000 system to exploit the new 64-bit large memory addressing capabilities of AIX 4.3.

- Performance features
  - RS64 processor

    The RS64 processors have been heavily optimized for commercial workloads. Target environments are characterized by heavy demands on system memory, both in the form of large working sets and latency-sensitive serial dependencies. As a result, the design of the RS64 processors, which run at 125 MHz, has focused on *large cache sizes* and *datapaths having high bandwidth and low latency*.

    The RS64 processor has separate 4-way associative 64 KB caches for instructions and data. It contains an L2 cache controller and a dedicated 16B interface to a private external 2-way set associative 4 MB L2. The L2 interface runs at full processor speed and provides 2 GBs of bandwidth. The RS64 processor also has a separate 16B system bus interface.

    The processor core was designed to optimize branch processing, and as part of the branch preprocessor function includes a 4K entry branch history table and integrated four entry link stack. The RS64 has a total of four execution units, and can sustain a decode and execution rate of 3 instructions per cycle. To support customer reliability and availability requirements, all the arrays in the RS64 have redundancy and ECC, which enable full fault detection and correction.

    The S70 processor card has four processors with associated L2 cache contained on each card. There are 4 MB of L2 per processor. Each processor card has the four processors on a single SMP system bus and that bus is presented to the S70 backplane.

  - High-speed backplane

    The S70′s four system data buses run at 83.33 MHz. Three buses connect to the processor cards and one bus connects to the I/O hub chip. The backplane uses a new switch-based memory controller complex. The complex contains 6 chips. One chip acts as the data flow control chip for 4 data switch chips. There is an additional system bus arbiter chip.

Crossport traffic is queued if needed. Each processor card system bus is directly connected to a port of the switch-based memory controller complex. Each data path is 128 bits wide. Addressing is via a separate 64-bit data path.

The memory controller complex is mounted on the two-sided active backplane. The processors and memory are inserted as cards. The I/O subsystem is connected to the complex via the system bus to the Remote I/O (RIO) interface I/O hub chip. The I/O hub chip is also mounted on the active backplane.

The memory controller complex has 6 unique data ports. Each of the three processor cards has its own port. One port is connected exclusively to the I/O hub chip. Two ports connect to the memory. The first port connects to memory quads A, C, and D. The second port connects to quads B and E. The ports run independently of one another, which allows for another dimension in concurrent operation. Each of the two memory ports is 64 bytes wide. The ports run at 2:1 mode with a 24 nanosecond cycle. The two memory ports together have *an aggregate bandwidth of 5.33 GB/s.*

One of the design point optimizations of this backplane is the capability of handling the next generation of processors. *The upgrade is designed to allow the replacement of only the processors books and retain an overall balanced system*.

– RIO Connections

The entire I/O subsystem is connected via the system bus to the RIO hub interface chip, which is mounted on the S70 backplane. Four RIO connections are supported with a single hub chip. RIO connections are scalable high-speed point-to-point interfaces designed for low latency, high-bandwidth connections between two boards or boxes. Each RIO bus supports 500 MB total, or 250 MB in each direction concurrently. Because the CEC enclosure contains no I/O, RIO cables connect the CEC to the I/O drawers. The RIO connections are set up as loops. The I/O hub chip directs the traffic around the loop in the most optimal way for performance, and will redirect traffic if there are link errors.

These RIO connections are the key to allowing an expandable number of I/O drawers that are physically separated from the CEC. This feature in turn also enables the high number of PCI buses and slots.

• Improved reliability

The S70 system consists of two physical enclosures. The first is the central electronics complex (CEC) unit which contains the RS64 processor cards, the memory controller complex backplane, the memory cards, the power supplies and the cooling units. The CEC cards are packaged in “books” for added reliability. Book packaging is described as cards that are sandwiched between two sheets of metal for protection, proper insertion and retention, and optimum air flow.

The S70 is the first RS/6000 to use Predictive Failure Analysis to monitor the error rates on DASD, memory, processors, L1 and L2 caches, and remote I/O. This is also the first time for an RS/6000 system to allocate one out of every fourteen memory chips for use as replacement in the event of hard or soft mainstore failures. Redundancy is also provided for data traffic between the CEC and the I/O drawer via a looped cable. If the cabling to the I/O

drawer is experiencing errors, the hardware is designed to redirect the I/O traffic through the alternate path on the loop.

The architecture of the RS/6000 S70 is shown in Figure 29.

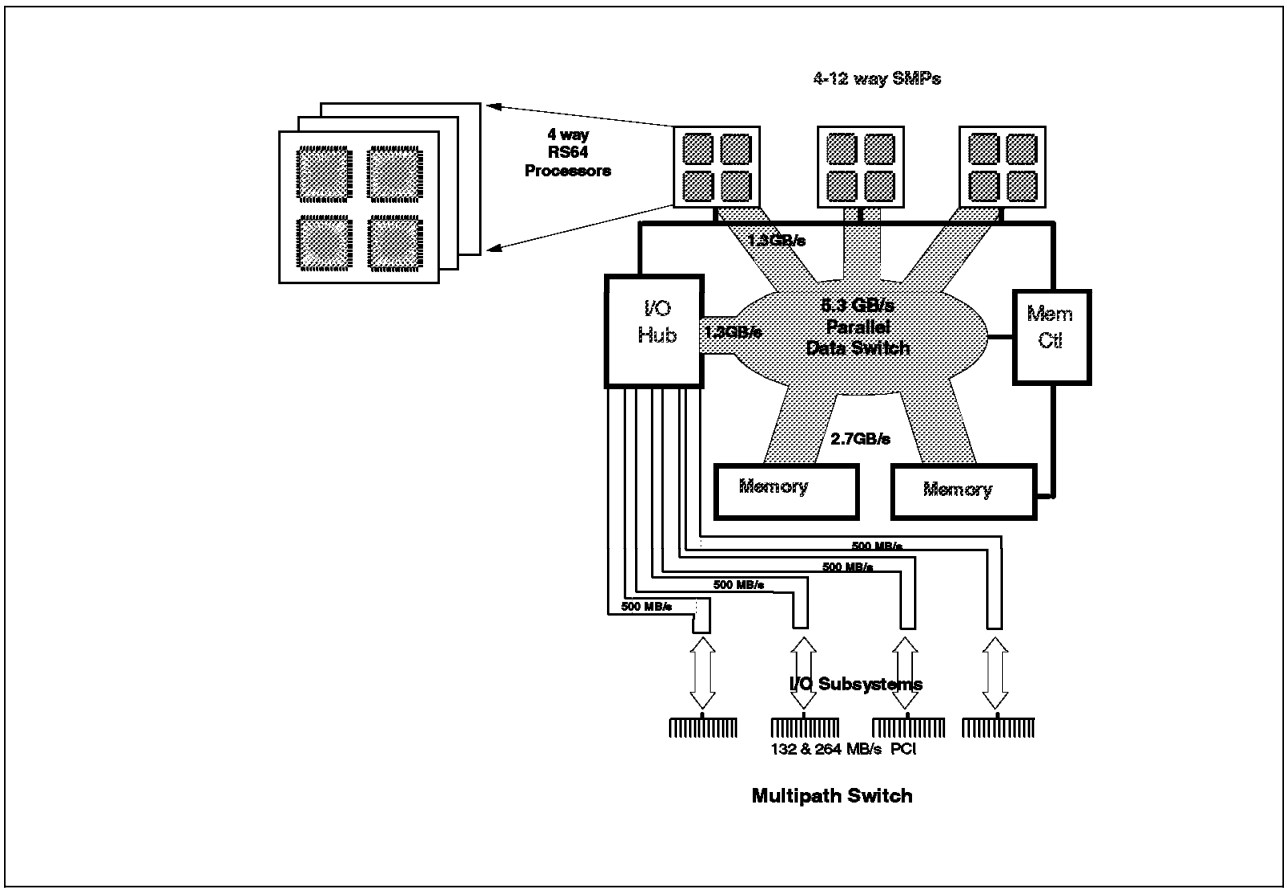


Figure 29. S70 Architecture

C.4 Introduction to IBM SP MPP Technology

The RS/6000 SP architecture is distinct from other platforms discussed in this chapter. It adopts the shared-nothing programming model, while the others use the shared-memory programming model, which is also called SMP. The shared-nothing programming exploited by SP is often referred to as a Massively Parallel Processor (MPP).

The RS/6000 SP is a general-purpose scalable parallel system based on a shared-nothing message-passing architecture. The latest PowerPC and P2SC (POWER2 and POWER3 SuperChips) RS/6000 processors are used for SP nodes, interconnected by SP the Switch, which is a packet-switched network for interprocessor communication.

Each node contains its own copy of the standard AIX operating system. The most distinct feature of the RS/6000 SP is that it is a “shared-nothing” system, unlike other SMP servers compared in this chapter which are “shared memory” systems. Since the “shared memory” system is touched on at the beginning of this chapter and is discussed in more detail in the following sections, only the “shared nothing” system is discussed here.

Distributed memory architecture is also known as *shared nothing* architecture. All processors have their own memory and disks. Because the address spaces handled by each processor are distinct, uniprocessor programs have to be changed to use the parallelism of the system. They must pass messages across an interconnect in order to use multiple processors.

C.4.1 Parallel Programming Models of RS/6000 SP in Commercial Computing

It has often been assumed that developing parallel applications on a distributed memory system like RS/6000 SP would be very difficult, but that is not true, especially in regard to commercial applications. Most of the modern client/server-based commercial applications in the UNIX environment are largely based on a few key subsystems:

- RDBMS such as DB2 UDB for AIX, Oracle, Sybase, and Informix
- TP monitors such as IBM TXSeries for AIX, Encina, and Tuxedo
- Web servers

Of these, RDBMSs are the core components that need to be ported to make the best use of RS/6000 SP′s parallel capability. The other middleware components are easily adapted to utilize parallel functions. Porting these few primary databases to run in parallel on an RS/6000 SP system provides the basis for enabling a host of applications that utilize these subsystems. Many commercial applications do not need to be modified to run in a parallel environment since they utilize and request services from a few key subsystems. In this sense, the solution is actually less complex than the technical computing area in which most of the individual applications have to be individually enabled for the scalable parallel environment. It has to be noted that most of the leading RDBMS vendors have already been supporting the shared-nothing programming model of the RS/6000 SP.

C.4.1.1 332 MHz SMP Node

This node′s outstanding performance is largely due to advances in cache and memory access. The system may look similar to other enterprise servers, but it is unique in design when compared to either the 2-way F40 or to 1-8 way J50 and R50 SMPs. It does not use the crossbar switch technology found in the J50 and R50, but instead uses a fast processor-memory bus, as well as new cache controller and Synchronous DRAM (SDRAM) technology. The use of SDRAM permits the memory subsystem to operate at a higher clock frequency than traditional DRAMs and at a higher effective bandwidth.

This design incorporates high-speed processor and memory buses, a more efficient 8-way associative L2 cache design, a high-performance memory-I/O controller, and fast SDRAM. The memory-I/O controller has a separately clocked, 64-bit mezzanine I/O bridge bus to which three independent PCI buses are connected. This unique design partitions system logic into a high-speed processor-memory portion and a lower-speed industry standard I/O portion. This allows for greater throughput on the system bus and lower manufacturing costs, because the standard PCI devices can be utilized.

Summarizing the state-of-the-art technology exploited:

- Innovative bus design

  Separation of high-speed processor-memory bus from narrower but cost-effective Mezzanine I/O bus
- Use of fast SDRAM (10ns)
- 1.3 GB/sec peak memory bandwidth

- 8-way set associative L2 cache (256 KB per processor)

The Silver node architecture is shown in Figure 30.

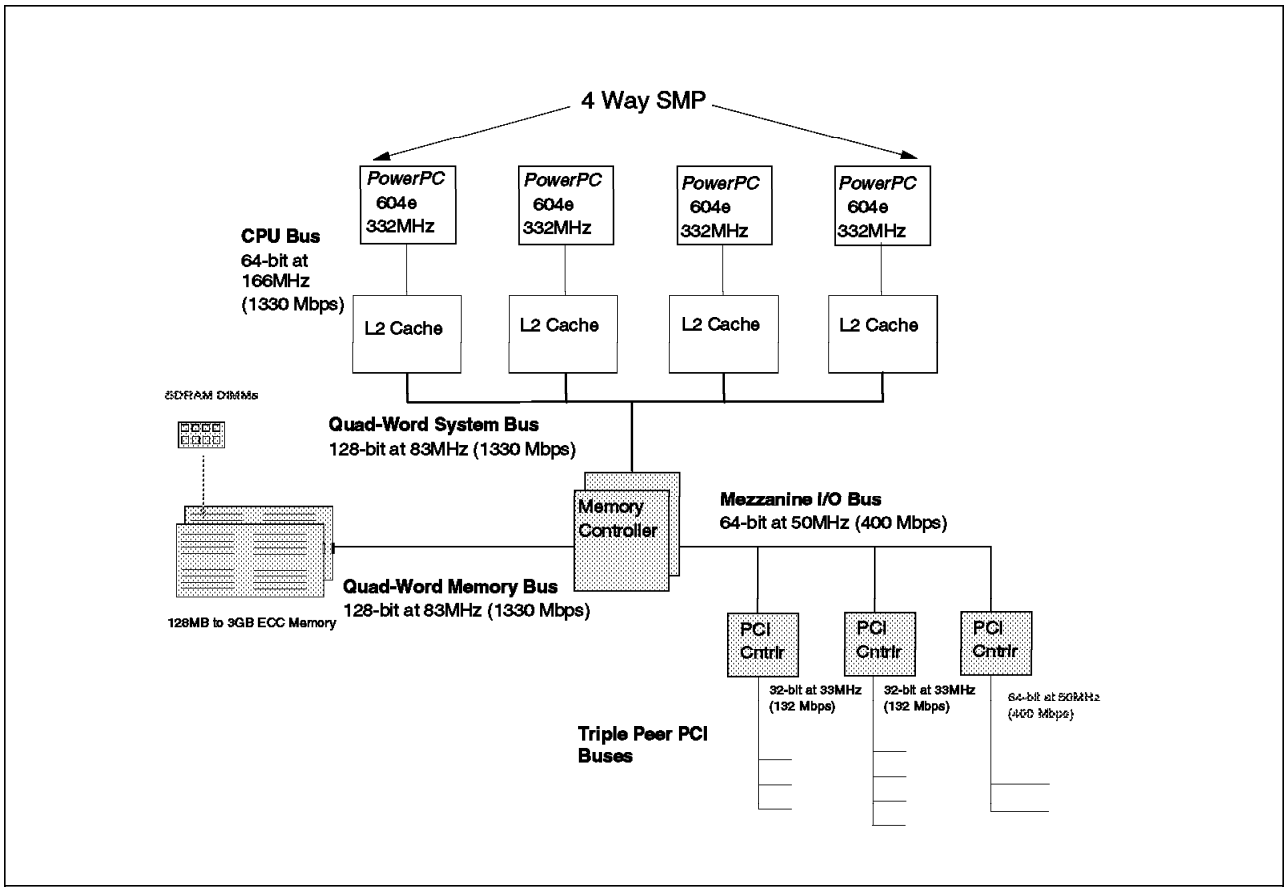


Figure 30. POWER3 SMP Node Block Diagram

C.4.1.2 SP Switch Router

The SP Switch Router is a high-performance I/O gateway, enabling direct network attachment of users′ company-wide network to the high-speed SP Switch network. The SP Switch Router is ideal for:

- SP system users who need faster I/O than provided by node I/O adapters, for example, applications that must quickly move large amounts of data between the SP system and external networks.
- Commercial computing installations that require high-speed access to database servers, such as the RS/6000 S70.
- e-business applications such as Web serving.
- Installations using the SP system for server consolidation.

The performance levels achieved by SP Switch Router are derived from the unique combination of the following:

- High-throughput non-blocking crossbar switch.
- Each media card is dedicated to IP forwarding and contains the full route table of up to 150,000 in its memory.
- Hardware-assisted route table lookup within 2.5 microseconds.
- Built-in intelligence for Layer 3 switching.

C.4.1.3 S70 as an SP Node

The S70 can be attached to the SP Switch via a new switch adapter. If the SP user wants to use an S70 for single-node performance or capacity that exceeds that available in the SP system, then S70 can be integrated into an SP environment. While physically remaining outside the SP frame, the S70 will be treated just as an internal node and be managed from the SP control workstation. The SP Switch will support connection to S70.

The obvious benefits of this newest feature are:

- Unprecedented performance growth of an SP node.
- Able to run 64-bit applications as well as 32-bit applications
- Multiple S70 nodes can support large parallel databases
- Single point of control by PSSP
- S70 will be used the same as internal nodes

As an example, in the case of a large and powerful database server being required, such as ERP or server consolidation, this will make it possible to utilize the best of the RS/6000 SP and S70 systems.

Appendix D. Special Notices

This publication is intended to help IBM and customer personnel who have the task of selecting a server for a new application and who need to better understand Industry Standard benchmarks, including their range of applicability, their strengths and weaknesses, and the production environment characteristics that they do not measure.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names

are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
CICS
DB2
ESCON
IBM
Netfinity
OS/390
POWER Architecture
POWERparallel
PR/SM
RS/6000
S/390
SP
VM/ESA
400
AS/400
CICSPlex
ES/9000
HACMP/6000
MVS/XA
OS/2
Parallel Sysplex
Power PC 604
PowerPC Architecture
Predictive Failure Analysis
S/370
S/390 Parallel Enterprise Server
TXSeries
VSE/ESA

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and Proshare are trademarks of Intel Corporation in the U.S. and/or other countries. (For a complete list of Intel trademarks see *www.intel.com/dradmarx.htm*).

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 119.

- *Understanding IBM RS/6000 Performance and Sizing*, SG24-4810
- *RS/6000 Model E30, F40, F50 and H50 Handbook*, SG24-5143
- *Selecting a Server - The Value of S/390*, SG24-4812

E.2 Redbooks on CD-ROMs

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at http://www.redbooks.ibm.com/ for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037

E.3 Other Publications

The following publications are also relevant as further information sources and can be obtained from:
http://www.vasari.com/s390order/

- *ITG White Paper: Strategies for Scalability: Challenges of Large Scale Internet/Intranet Computing*, GF22-5013
- *ITG White Paper: Cost Implications of Platform Choice*, GF22-5029
- *ITG White Paper: Cost of Messaging*, GF22-5041
- *ITG White Paper: Web Server Solutions*, GF22-5048
- *ITG White Paper: The Cost of Scalability*, G326-3056
- *IDC White Paper: Strategic Fit of Enterprise Servers in Large Organizations*

E.3.1 External References

- *UNIX Server Consolidation-Observations and Recommendations*, D.H. Brown Associates, Inc., February 6, 1998
- *IBM Move to Lower Barrier to Clustered Server Solution*, D.H. Brown Associates, Inc., May 28, 1998
- *The Benchmark Handbook for Database and Transaction Processing Systems*, D.H. Brown Associates, Inc., May 28, 1998
- "Managing The Costs Of Enterprise Computing," Datamation, March 15, 1994
- *Starfire: Extending the SMP Envelope*, pp39 - 49, Jan./Feb 1998 issue, IEEE Micro

E.3.2 References Available to IBM Personnel Only

- *Positioning IBM Servers (vs HP)*, Arne Kruithof, IBM EMEA
- *Convex/HP Exemplar vs IBM SP*, D.H.Brown Associates

These papers can be found at:

`http://emeami.dk.ibm.com/compass/hpcompas.nsf`

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download, or order hardcopy/CD-ROMs redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the redbook fax order form to:

In United States:	e-mail address: usib6fpl@ibmmail.com
Outside North America:	Contact information is in the ″How to Order″ section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl/

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the ″How to Order″ section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl/

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the ″How to Order″ section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl/

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbook Fax Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

U

W

ITSO Redbook Evaluation

Enterprise Servers: Benchmarketing Perspectives
SG24-5179-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
__Customer __Business Partner __Solution Developer __IBM employee
__None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

_____

_____

_____

_____

_____

SG24-5179-00
Printed in the U.S.A.

SG24-5179-00