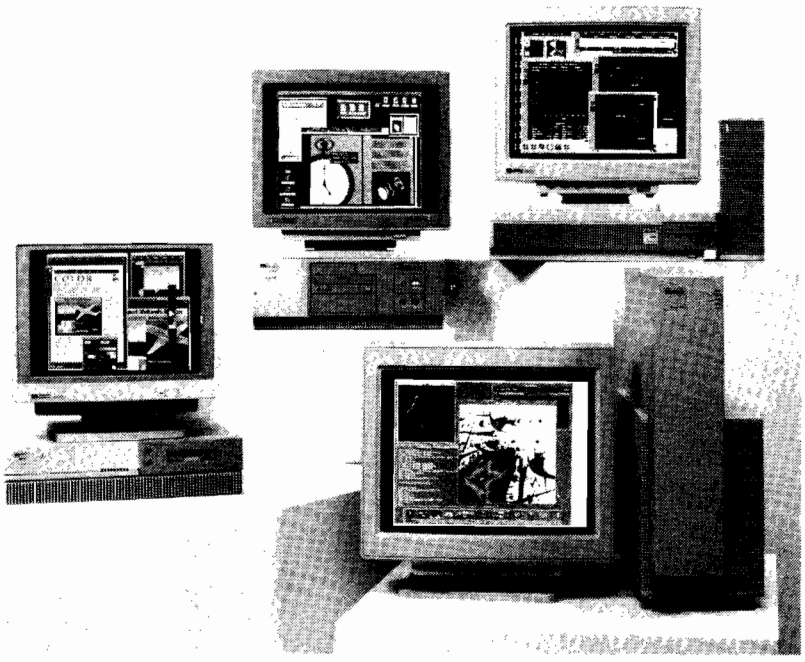
HEWLETT PACKARD

HP Apollo 9000 Series 700 Performance Brief

November 1992



Introduction

Chapter 1 - SPEC CPU Benchmarks

Chapter 2 - Dhrystone

Chapter 3 - Linpack

Chapter 4 - Whetstone

Chapter 5 - X11perf

Chapter 6 - Graphics Performance Characterization Committee Benchmarks

Chapter 7 - Traditional Graphics Benchmarks

Chapter 8 - HP 700/RX X-stations Performance

Chapter 9 - Advanced Optimizing Compilers for PA-RISC

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Appendix A

Appendix B

Appendix C

Appendix D

Performance Quick Reference Card

Introduction

This document provides detailed information on selected performance features of Hewlett-Packard's Series 700 workstations and X-stations. Within you will find results of CPU and graphics benchmarks and a valuable chapter on the performance contribution of compiler technology with RISC based computer systems.

With this document HP provides detailed performance test results on new PA-RISC workstation systems which include the new microprocessor technology based on HP's PA7100 RISC implementation. HP offers workstations with a wide range of price, performance, and expandability options. Since their introduction in March of 1991, the HP Apollo 9000 Series 700 workstations have maintained a position as some of the most affordable high-performance computer systems and X-terminals available.

This performance brief is designed to assist you in making a more informed workstation system purchase. Computer system performance enhances productivity and can substantially contribute to the success of your business. A clear understanding of the performance of the systems you are considering is an important part of the decision process. Numerous performance tests have been run that cover the major components of the system. This document compiles the results of those tests and explains what they mean and how they should be used.

Ideally, you would test each system with performance tests that produce workloads representative of your computing environment. This is an expensive, difficult and time-consuming task for both customer and vendor. The results presented herein may be used to narrow the field of systems under consideration, even if you may choose to do your own more extensive benchmarking.

All of the measurements for the HP systems have been made by Hewlett-Packard engineers except where explicitly indicated. Test environment details are given in the appendices. Every effort has been made to verify that the comparisons to non-HP systems presented in this brief are valid at the time of printing and represent the best known performance of all systems. Many of the sections contain benchmark results for non-HP systems which have been taken from reliable and publicly available sources. The sources of the competitive data are referenced where appropriate.

About this document

This performance brief is organized into several sections of performance test results augmented by appendices that provide supporting details such as system configurations and benchmark descriptions. We have focused on a subset of those system components that are the primary contributors, or limiters, to overall application performance: processor, graphics and compilers/optimizers. Future editions of this document will cover disk, networking, and other aspects of system performance.

Your application or system usage, even if similar to the benchmarks, is expected to differ from any contrived performance test.

Every effort has been made to assure the accuracy of the data in this document. HP can not be held responsible for errors or omissions.

This document supersedes all previous HP Apollo 9000 Series 700 performance briefs.

PA7100 Overview

The PA7100 CPU is the first PA-RISC microprocessor to integrate the floating point coprocessor onto the same chip that contains the RISC integer core and memory management hardware. A single component replaces the two VLSI chips required for previous PA-RISC processors.

Compared to the PA7000 (PA-RISC 1.1) processor, the PA7100 offers enhanced performance for all applications while maintaining strict binary compatibility. Applications which make heavy use of floating point calculations will especially benefit. Added performance was achieved through a combination of design evolution and improvements in integrated circuit technology. Significant gains were made in four specific areas:

1. **Clock Frequency.**
   Boosting the maximum clock rate from 66 MHz to 99 MHz results in a near 50% performance gain across all processor intensive applications. Improvements in HP's CMOS26 (0.8 micron) integrated circuit process largely account for this gain.
   Like its PA7000 predecessor, the PA7100 supports enormous external primary cache memory arrays when compared to competing microprocessors. Advances in static random access memory speed is therefore also an important factor. Workstations and servers based on the PA7100 processor are currently available at clock speeds ranging from 33 MHz to 99 MHz.

2. **Exceptional Floating point performance.**
   Competitive microprocessors with integrated floating point units generally suffer a performance disadvantage compared to multi-chip implementations such as the PA 7000 processor. The PA7100 chip reverses this trend by providing industry leading performance in a single chip format. The following table summarizes the comparison between the PA7100 and the PA7000 implementation:

	Clock Frequency (MHz)	Add or Multiply Latency (cycles / time)	Peak MFLOPS	Load/store Bandwidth (peak Mbytes/sec)	Super-scalar ?
PA7000	66	3 / 45nS	66	528	N
PA7100	99	2 / 20nS	200	800	Y

3. **Superscalar Execution.**
   The communication channel between the PA7100 CPU and its Instruction Cache has been doubled in width compared with its PA7000 predecessor. This enables the 7100 to achieve instruction level parallelism. In this scheme multiple consecutive instructions are fetched by the CPU and simultaneously dispatched to independent integer and floating point processors.

4. **Cache and memory access optimizations.**
   A variety of new features were included on the PA7100 to reduce performance penalties associated with the memory hierarchy. For example, the PA7000 processor incurs a potential penalty of two clock cycles for the execution of any store to the data cache. In the PA7100 design, a technique known as a "Pipelined Store" reduces the maximum penalty to a single cycle.
   The interface between the cache and main memories has been improved as well, resulting in an improvement in parallelism over the PA7000 processor. Through a mechanism known as "Stall-on-Use" the CPU is able to process both an instruction stream and a cache miss (main memory transaction) simultaneously. The CPU cycles saved translate directly to improved performance.
   Hardware support for the virtual memory system has been improved as well. The PA7100 incorporates a "hardware table walker", or hardware TLB, which is capable of servicing a TLB miss in a fraction of the time required by the PA7000 processor. This provides a noticeable performance advantage for a class of applications with very large working sets, notably database and OLTP environments.

Series 700 Systems Overview

This performance brief introduces Hewlett-Packard's new generation of high-performance PA RISC workstations with the PA-RISC 7100 microprocessor. HP PA-RISC workstation family, with full binary compatibility, now spans a range of performance from 22 to 80 SPECint92 (40 to 124 MIPS). The table below provides a ready reference to the processor type, clock speed, cache size and SPEC92 performance for the systems covered in this performance brief. Boldface type in the table denotes the new systems.

HP Apollo 9000 Series 700 Systems

	Processor and Floating Pt	Processor Clock Speed (MHz)	Cache Size instr/data (Kbytes)	SPEC int92	SPEC fp92
705C	PA 7000 - HP custom	33	32/64	22.7	39.3
710C	PA 7000 - HP custom	50	32/64	32.7	56.4
715/33C	**PA 7100 - integrated**	**33**	**64/64**	**24.2**	**45.0**
715/33 t / s[1]	**PA 7100 - integrated**	**33**	**64/64**	**23.3**	**44.2**
715/50C	**PA 7100 - integrated**	**50**	**64/64**	**36.5**	**72.1**
715/50 t / s[1]	**PA 7100 - integrated**	**50**	**64/64**	**35.3**	**66.8**
720CRX	PA 7000 - HP custom	50	128/256	38.5	66.1
730CRX	PA 7000 - HP custom	66	128/256	52.0	86.7
745i, 747i	**PA 7100 - integrated**	**50**	**64/64**	**36.0**	**72.0**
735CRX	**PA 7100 - integrated**	**99**	**256/256**	**80.0**	**150.6**
750CRX	PA 7000 - HP custom	66	256/256	51.1	84.8
755CRX	**PA 7100 - integrated**	**99**	**256/256**	**80.0**	**150.6**

1. Systems with the t/s designator are upgrades to the model 425t desktop, 425s deskside and model 433s deskside systems. Throughout this document, chart labels with the 's' suffix denote the Series 400 to 700 upgrades, e.g. 715/33s.

Series 400 Upgrade to PA RISC 7100

The HP Series 400 workstations, based on the Motorola 68040 processor, are now field upgradable to PA RISC-based systems. Upgrade capabilities vary depending upon the model of Series 400. The series 425 and 433 are upgradeable to 33 MHz or 50 MHz PA 7100 processors yielding from 4.9 to 7.4 times the floating point performance of a model 425s.

The table below provides an overview of performance increases for selected measures. For example, the SPECmark89 rating increases almost 4 times (3.6x) when upgrading from a 425s to a Series 700 model 715/33s (column 5, row 4).

Series 400 to PA 7100 Upgrades

workstation model	dhry-stone MIPS	dhrystone MIPS increase	SPEC mark '89	SPEC mark '89 increase	Linpack double MFlops	Linpack double MFlops increase
425t/s	22	--	11.8	--	1.7	--
433t/s	30	--	14.9	--	2.2	--
715/33t/s from 425	41	1.9	42.8	3.6	8.3	4.9
715/50t/s from 425	62	2.8	65.0	5.5	12.6	7.4
715/33t/s from 433	41	1.4	42.8	2.9	8.3	3.8
715/50t/s from 433	62	2.1	65.0	4.4	12.6	5.7

Please consult your HP sales representative for specific and up-to-date upgrade information.

For comprehensive table of systems and performance data, refer to the Performance Quick Reference Card in the back of this document.

Chapter 1 - SPEC CPU Benchmarks

About SPEC

The organization known as SPEC, the Standard Performance Evaluation Corporation, has as its charter "the development of benchmarks that measure overall computer system performance". SPEC is a non-profit corporation whose members are primarily computer system manufacturers. HP is a founding member of the organization and strongly supports SPEC's goals of "leveling the playing field" for system performance evaluation.

SPEC currently has 19 members including virtually all the major computer manufacturers: AT&T/NCR, Bull S.A., Compaq, Control Data Systems, Data General, Digital Equipment (DEC), Fujitsu Ltd., HP/Apollo, IBM, Intel, Intergraph, MIPS, Motorola, NeXT, Siemens-Nixdorf Information Systems, Silicon Graphics, Solbourne, Sun Microsystems and Unisys.

SPEC Benchmarks and Metrics

The original SPEC benchmark suite, released in 1989, consisted of 10 CPU-intensive benchmarks. Execution time of the benchmarks is expressed as a ratio to a DEC VAX 11/780 running the Ultrix 3.1B operating system. The geometric mean of these ratios across the ten benchmarks yields the composite metric *SPECmark*.

In January 1992 SPEC announced two new suites of processor-intensive benchmarks intended to replace the venerable and now popular 1989 suite. Each of these two new suitesare known as Cfp92 and Cint92 for component floating point and component integer respectively. Each has its own composite metric: SPECfp92 or SPECint92. The new metrics are intended to replace the SPECmark, SPECfp, and SPECint metrics.

The two suites consist of a total of 20 benchmarks, twice the number in the original suite. There are 6 integer (non-floating point) benchmarks and 14 floating point benchmarks. Metrics from the two suites are not combined and therefore there is no SPECmark92 rating.

For purpose of differentiation and to avoid confusion, SPECmark is now known as SPECmark89. Given the rapid advances in hardware and software technology, the year of release is appended to the new metrics anticipating replacement suites in the coming years.

On the following pages are charts that depict the results of the new and old SPEC CPU-intensive benchmarks on HP and other workstation-class systems. Results from non-HP systems have been taken from the SPEC newsletters or provided by the vendors themselves.

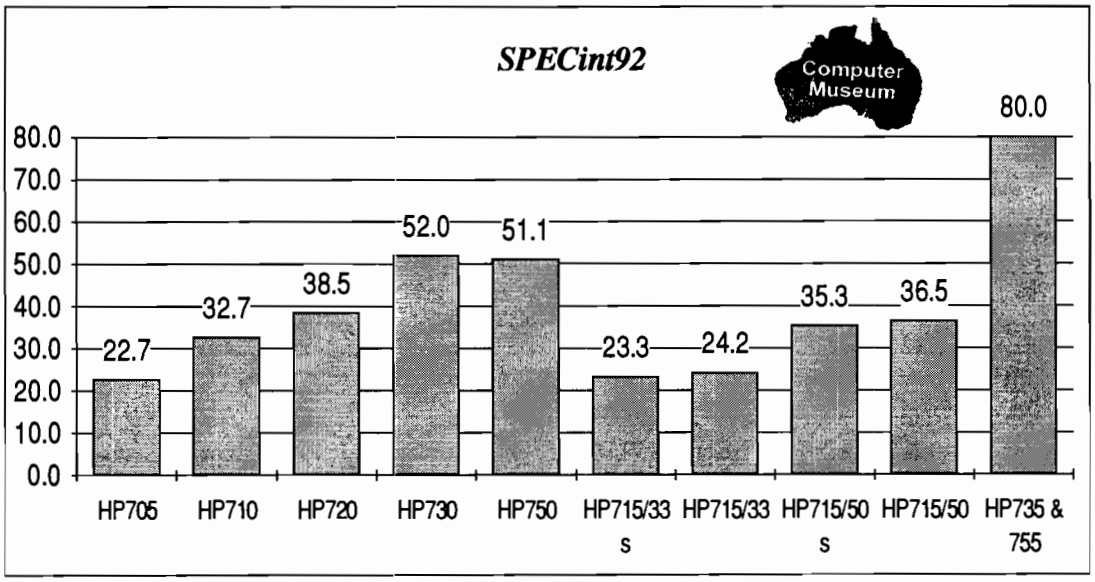
Descriptions of the benchmarks can be found in Appendix A.

For more information on SPEC or to subscribe to the *SPEC Newsletter*, contact:

NCGA
The National Computer Graphics Association,
Suite 200
2722 Merrilee Drive
Fairfax, VA 22031-4499
SPEC administrator Dianne Dean,
phone 703-698-9600 Ext 318.

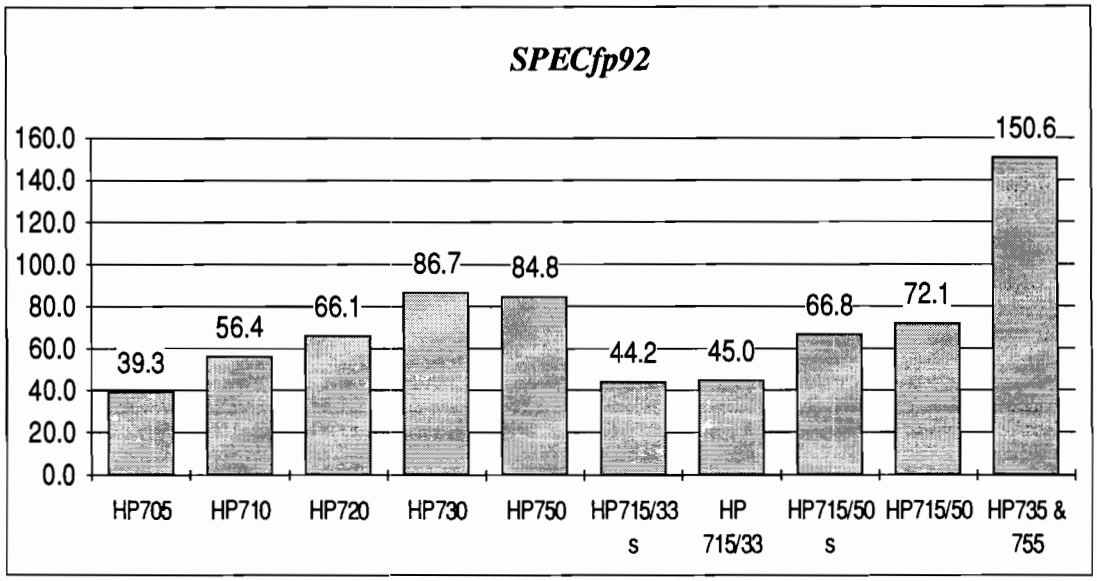
Cint92 suite

SPEC's new integer suite is comprised of 6 real-world application benchmarks. Cint92 represents several application areas: circuit theory, a LISP interpreter, logic design simulations, text compression algorithms, spreadsheet, and software development.



Cfp92 suite

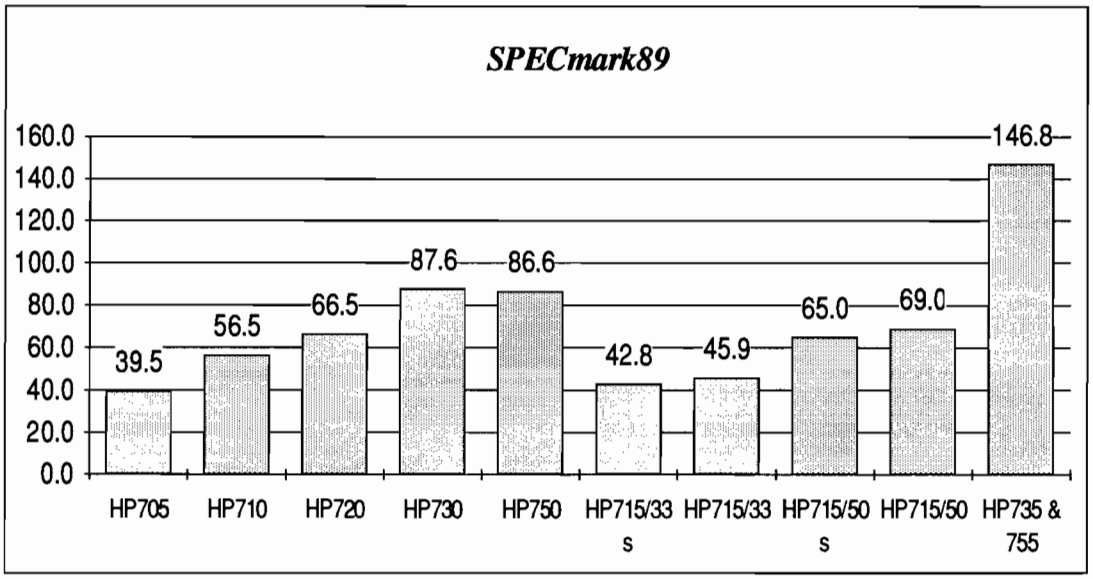
SPEC's new floating point suite is comprised of 14 real world application benchmarks, five of which are single precision. Cfp92 represents application areas in: circuit design, Monte Carlo simulation, quantum chemistry, optics, robotics, quantum physics, astrophysics, weather prediction, and other scientific and engineering problems.



SPECmark89

A SPECmark, now SPECmark89, is the geometric mean of the SPEC ratios for the 10 benchmarks that comprise the 1989 SPEC release 1 suite. A SPECratio is the ratio of the elapsed execution time for a given benchmark relative to that benchmark's execution time on a VAX 11/780 running the ULTRIX 3.1B operating system.

The SPEC release 1 benchmarks, which are weighted towards floating point, are predominantly CPU-intensive. It is the SPEC members' intent and desire to offer a set of benchmarks that encompass CPU, I/O, memory, graphics, and other system components but such code is simply not yet available, portable, or feasible to use.



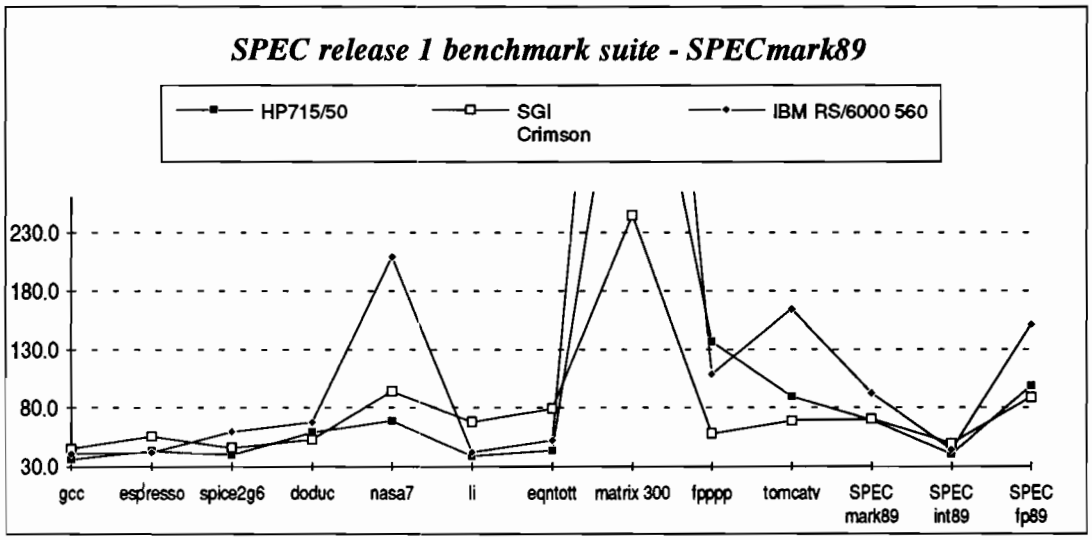
SPEC and its members are not advocates of the "one number theory" for system performance rating. The rationale for creating the SPECmark, as a single number rating, is that if SPEC did not create a single number then the press and vendors would create various ways of summarizing results thereby diluting the value of otherwise tightly controlled metrics.

It is recommended that you determine whether any of the benchmarks represents the type of work you do, now or in the future. You are encouraged to examine the results of each benchmark rather than rely on composite metrics for anything other than a rough selection criteria.

HP and Competitive Workstations

The chart below provides an example of the variability of results across somewhat similar RISC-based platforms. All have a 50MHz clock speed (the SGI Crimson-MIPS R4000 processor reportedly runs at 100MHz internally). Each system has a different processor architecture, compiler/optimizer technology and significantly different system costs.

Floating point variations are most noticeable with the benchmarks nasa7, fpppp and tomcatv. This chart also exemplifies why SPEC has chosen to drop matrix300 from its 1992 floating point suite. The HP model 715/50 achieves a SPECratio of over 314 and the IBM RS/6000 560 over 808.



On the following pages we highlight the 89 and 92 SPEC suites in detail with contrast to non-HP systems for low and high-end workstation systems.

The results for the individual benchmarks on all HP systems, for both SPEC 92 suites and the 1989 suite, can be found in Appendix B. The information in Appendix B is intended to satisfy the requirement of SPEC reporting rules for measured results on any systems available within 6 months of SPEC result publication.

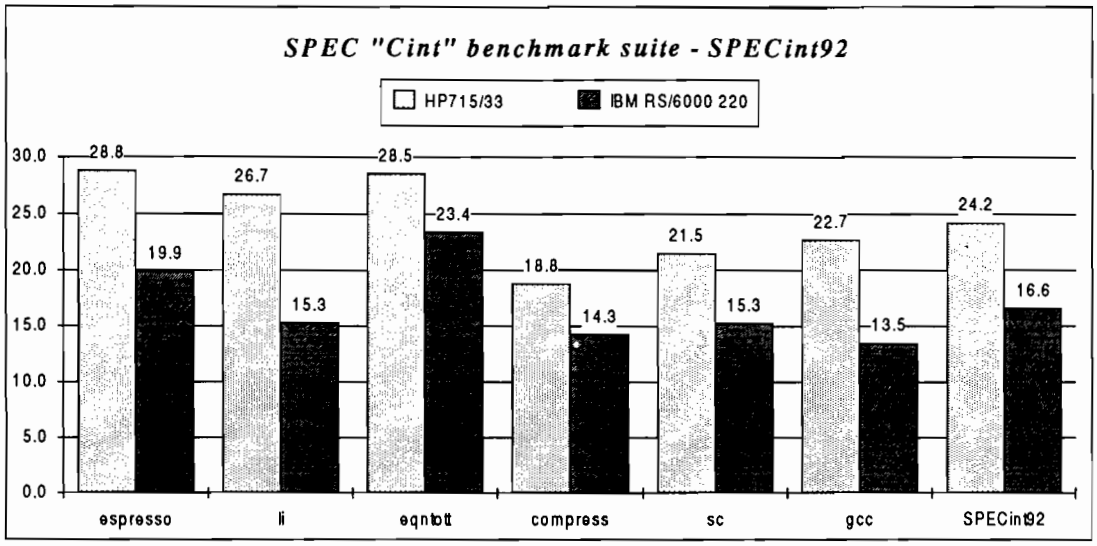
HP and IBM Workstations

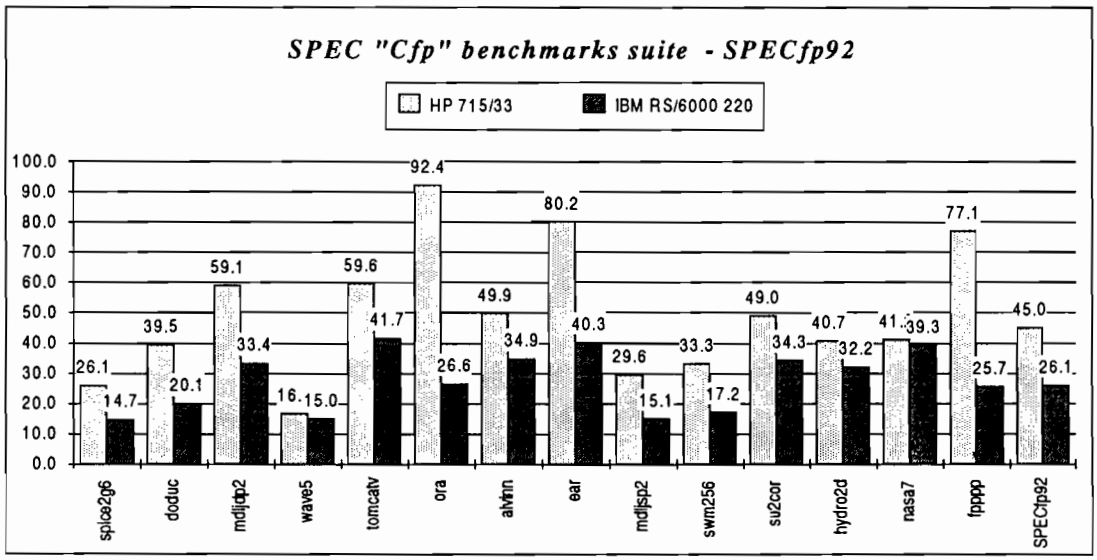
We have used the SPEC individual benchmarks with their composite metrics to compare two sets of systems in the charts on the following pages:

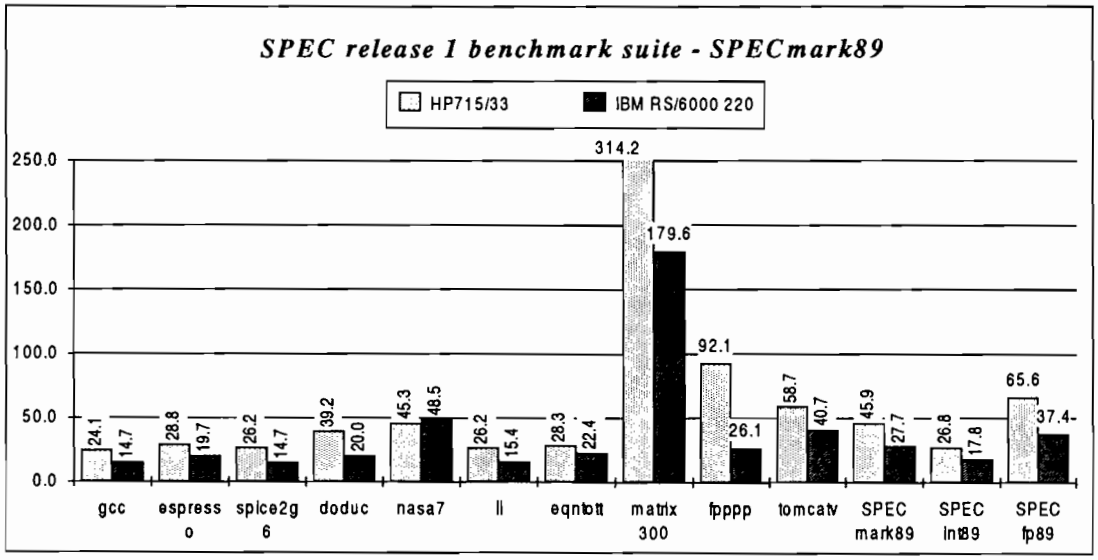
1) Low-end workstations: The HP model 715/33 and the IBM RS/6000 model 220. Both are systems with clock rates of 33 MHz. We have used the latest results, made available September 22, 1992, for the IBM 220 with AIX 3.2.3.

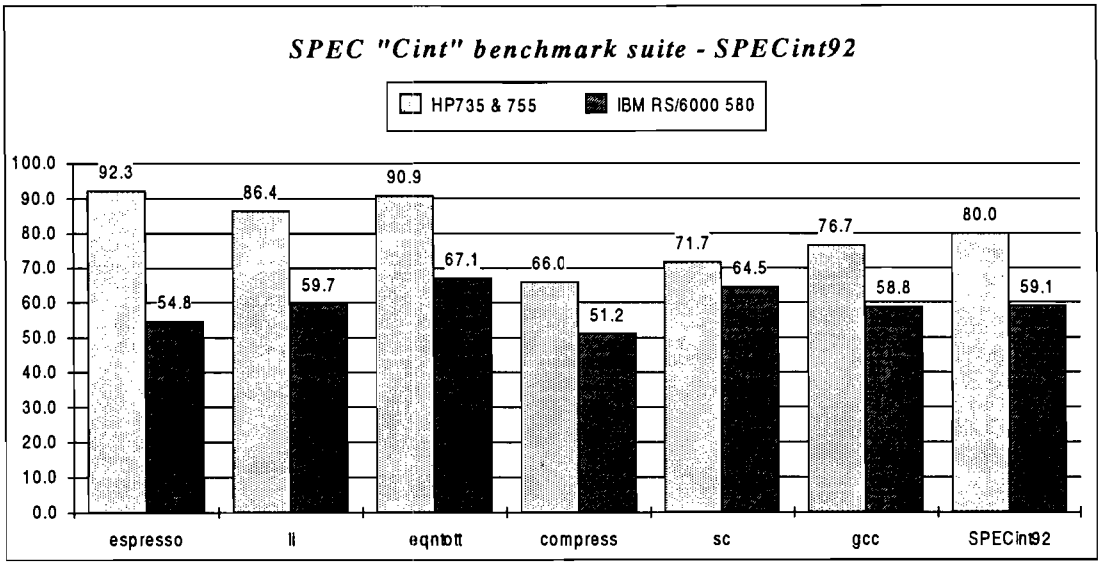
2) High-end workstations: The HP model 735 and the IBM RS/6000 model 580. The HP 735 and HP 755 both have clock rates of 99 MHz while the IBM 580 clock rate is 62.5 MHz.

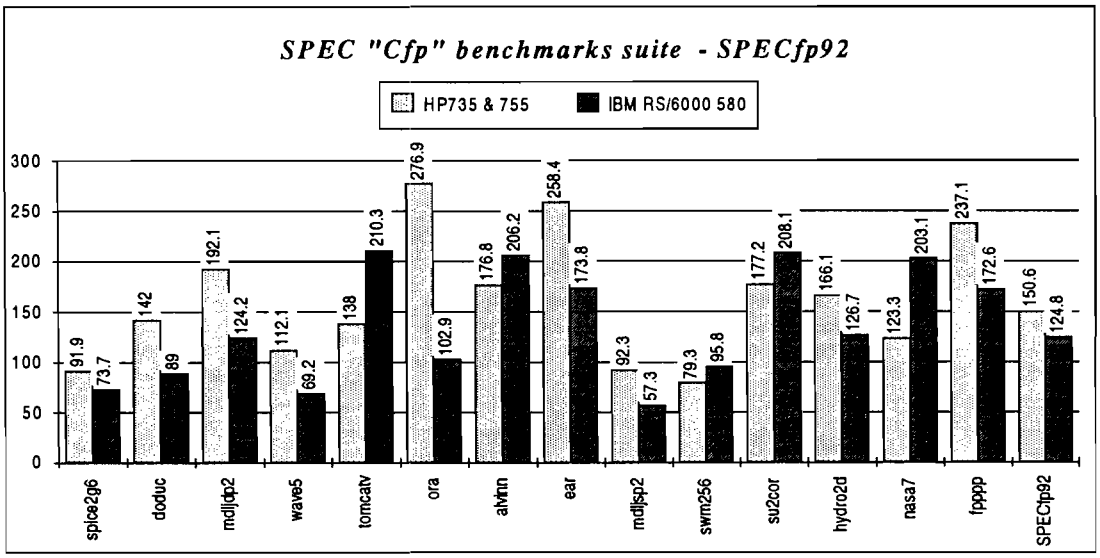
The IBM 580 system was announced September 22, 1992 with hardware available in December of 1992.

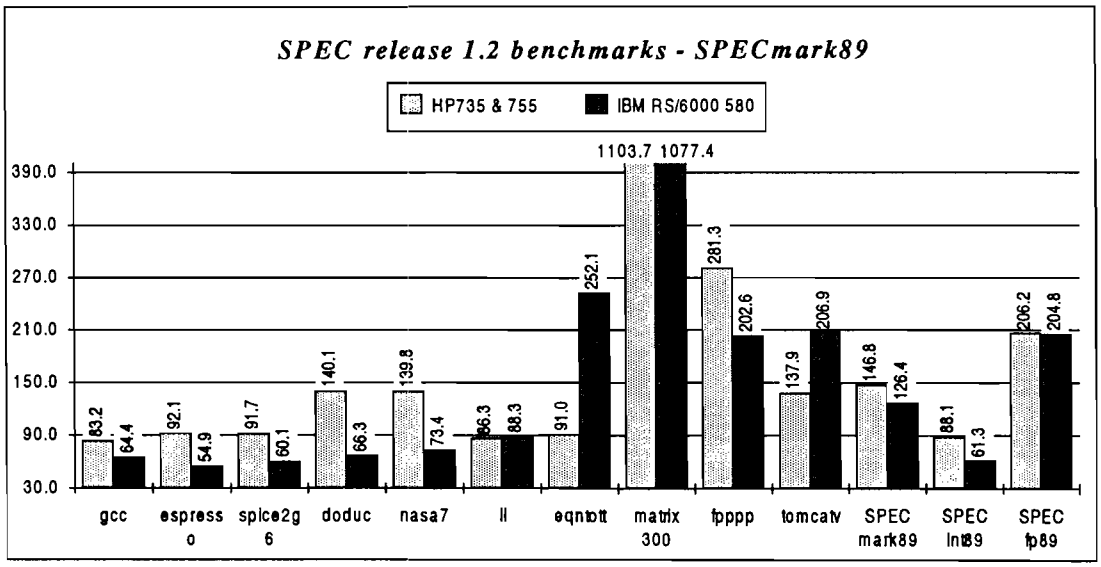

SPEC "Cint" benchmark suite - SPECint92
HP715/33
IBM RS/6000 220
30.0
25.0
20.0
15.0
10.0
5.0
0.0
28.8
19.9
26.7
15.3
28.5
23.4
18.8
14.3
21.5
15.3
22.7
13.5
24.2
16.6
espresso
li
eqntott
compress
sc
gcc
SPECint92


SPEC "Cfp" benchmarks suite - SPECfp92
HP 715/33
IBM RS/6000 220
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
26.1
14.7
39.5
20.1
59.1
33.4
16.
15.0
59.6
41.7
92.4
26.6
49.9
34.9
80.2
40.3
29.6
15.1
33.3
17.2
49.0
34.3
40.7
32.2
41.
39.3
77.1
25.7
45.0
26.1
spice2g6
doduc
mdljdp2
wave5
tomcatv
ora
alvinn
ear
mdljsp2
swm256
su2cor
hydro2d
nasa7
fpppp
SPECfp92


SPEC release 1 benchmark suite - SPECmark89
HP715/33
IBM RS/6000 220
250.0
200.0
150.0
100.0
50.0
0.0
24.1
14.7
28.8
19.7
26.2
14.7
39.2
20.0
45.3
48.5
26.2
15.4
28.3
22.4
314.2
179.6
92.1
26.1
58.7
40.7
45.9
27.7
26.8
17.8
65.6
37.4
gcc
espress o
spice2g 6
doduc
nasa7
li
eqntott
matrix 300
fpppp
tomcatv
SPEC mark89
SPEC int89
SPEC fp89


SPEC "Cint" benchmark suite - SPECint92
HP735 & 755
IBM RS/6000 580
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
92.3
54.8
86.4
59.7
90.9
67.1
66.0
51.2
71.7
64.5
76.7
58.8
80.0
59.1
espresso
li
eqntott
compress
sc
gcc
SPECint92

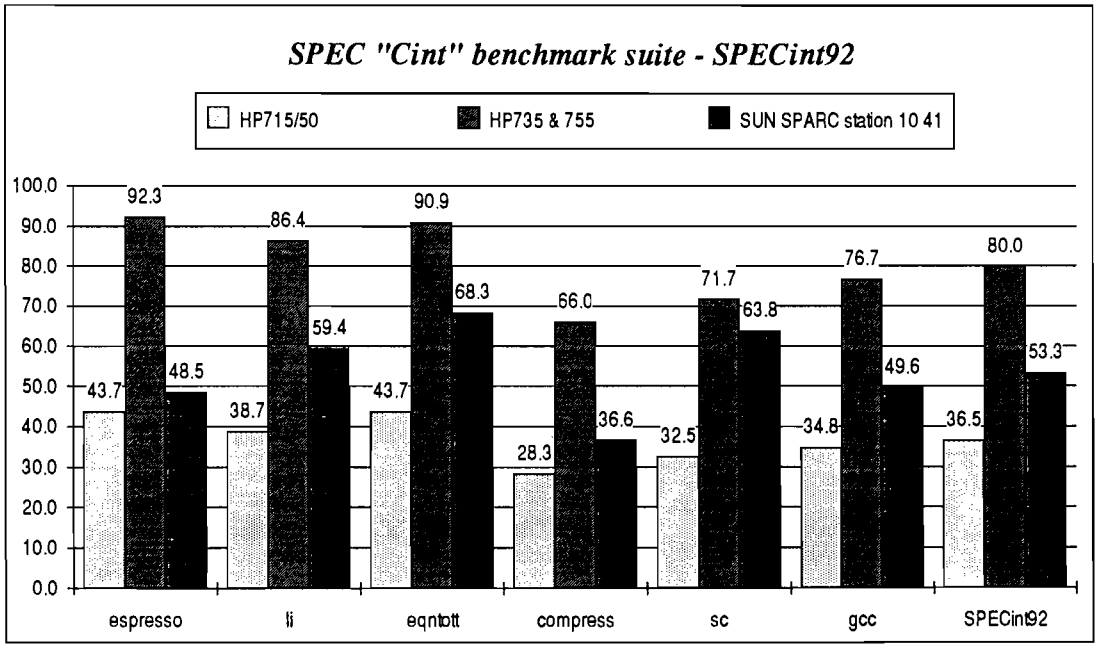

SPEC "Cfp" benchmarks suite - SPECfp92
HP735 & 755
IBM RS/6000 580
300
250
200
150
100
50
0
91.9
73.7
142
89
192.1
124.2
112.1
69.2
138
210.3
276.9
102.9
176.8
206.2
258.4
173.8
92.3
57.3
79.3
95.8
177.2
208.1
166.1
126.7
123.3
203.1
237.1
172.6
150.6
124.8
spice2g6
doduc
mdljdp2
wave5
tomcatv
ora
alvinn
ear
mdljsp2
swm256
su2cor
hydro2d
nasa7
fpppp
SPECfp92

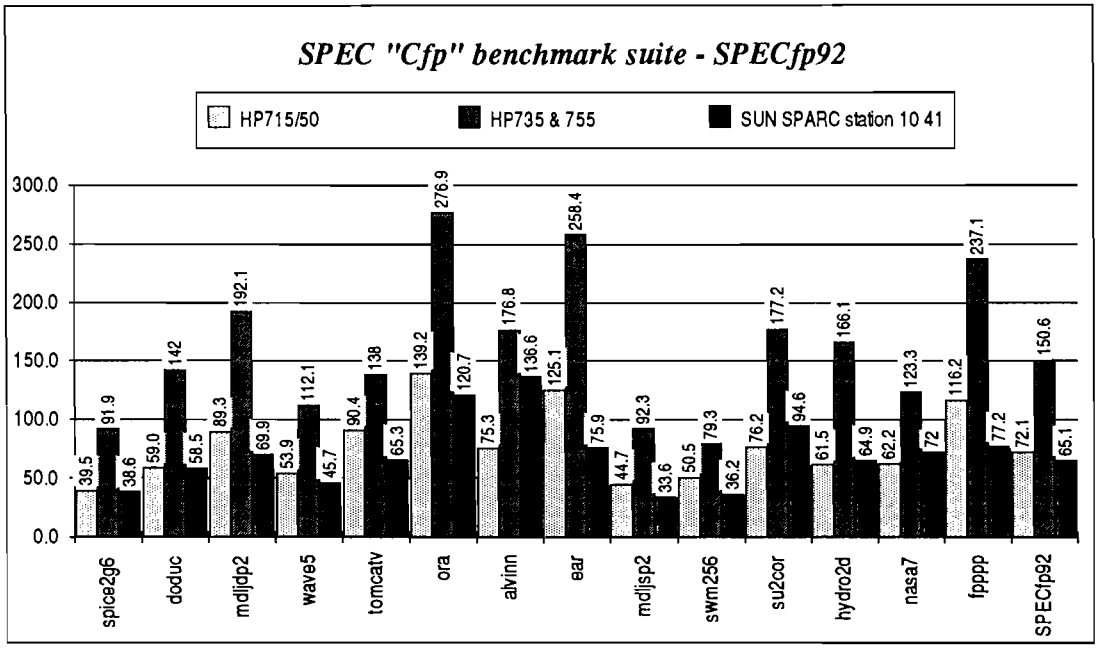

SPEC release 1.2 benchmarks - SPECmark89
HP735 & 755
IBM RS/6000 580
390.0
330.0
270.0
210.0
150.0
90.0
30.0
83.2
64.4
92.1
54.9
91.7
60.1
140.1
66.3
139.8
73.4
86.3
88.3
91.0
252.1
1103.7
1077.4
281.3
202.6
137.9
206.9
146.8
126.4
88.1
61.3
206.2
204.8
gcc
espresso
spice2g6
doduc
nasa7
li
eqntott
matrix300
fpppp
tomcatv
SPECmark89
SPECint89
SPECfp89

HP & Sun Workstations

The two charts below compare the HP models 715/50 and the HP 735/755 workstation systems with Sun's new SPARCstation 10 model 41.

Sun has not reported SPARCstation 10 results for SPEC release 1 benchmarks and therefore there exist only estimates for SPECmark89. See the Performance Quick Reference Card at the back of this document.





Chapter 2 - Dhrystone

About Dhrystone

Dhrystone is a popular integer benchmark that was developed in 1984 by Dr. Reinhold Weicker to reflect a programming environment. The frequency of the different types of high-level language statements in the benchmark is representative of this environment. Originally distributed in Ada, Dhrystone was rewritten in C by Rick Richardson. The C version results are the most recently reported values and are the values used here.

This benchmark is often used to measure processor and compiler efficiency. The results of the Dhrystone benchmark can reflect the effectiveness of a particular hardware and compiler combination for software development applications. In general this benchmark does not provide a good indication of application performance as it focuses on integer computation and fits completely in system cache.

The level of compiler optimization drastically affects the Dhrystone results. Currently, any level of optimization may be used, as long as program procedures are not in-lined, and results are appropriately labeled with the level of optimization. Results reported without qualification are understood to be without explicit register declarations.

Care must be taken to differentiate between the results for Dhrystone versions 1.x and 2.x, as the results can vary widely. Version 1.1 contains sequences of "dead code" (code segments that calculate results which are never used later in the program). This provides opportunity for compilers to identify the dead code and eliminate those instruction sequences from the program. With fewer code lines to execute, these compilers allow a system to complete the program in less time, resulting in a higher Dhrystone rating. The instruction sequence in version 2.0 has been modified to force execution of all instructions.

Dhrystone results are reported in dhrystone instructions executed per second (dhrystones/sec) or thousands of dhrystone instructions per second (Kdhrystones/sec). Dhrystone MIPS can be calculated by dividing the Dhrystone 1.1 result by 1757, the established Dhrystone rating of a VAX 11/780.

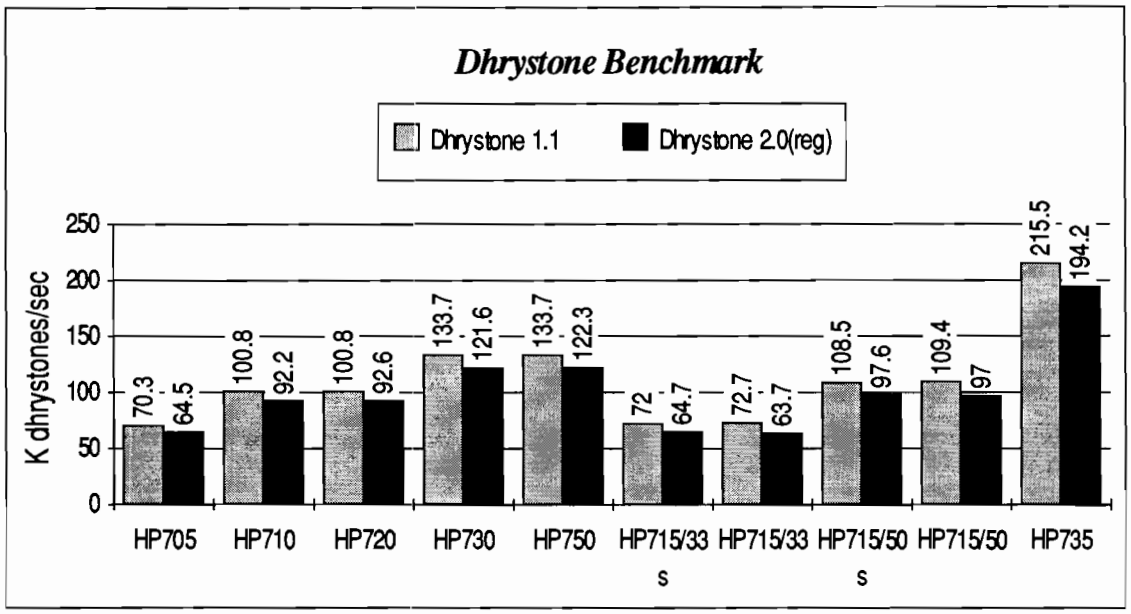


Figure 2.1

Chapter 3 - Linpack

About Linpack

The Linpack benchmarks are designed to provide information on the relative performance of computer systems using Basic Linear Algebra Subroutines (BLAS) to solve dense systems of linear equations. Linpack provides information on the overall hardware and compiler efficiency in this type of floating point compute-intensive environment. Using Linpack to predict general floating-point performance could be misleading.

Results for the Linpack 100x100 benchmark have been included in this document because it has become one of the most widely used benchmarks to predict relative performance for scientific and engineering applications where floating-point computations are prevalent. Developed at Argonne National Laboratories, Linpack is a FORTRAN benchmark that solves a 100x100 system of linear equations, emphasizing floating point addition and multiplication. Linpack is easily vectorized. The results are reported in millions of floating point operations per second (MFLOPS).

There are many different variants of the Linpack benchmark based on the different Basic Linear Algebra Subroutine (BLAS) libraries and depending on whether the internal program loop is "rolled" or "unrolled". The most frequently reported version is the compiled FORTRAN BLAS with rolled inner loop, and the Series 700 results included in this brief use this version. The measurements reported here may differ from the numbers used in many of the Series 700 marketing documents. This is due to conservative reporting in the other documents.

A version of Linpack which solves a 1000 x 1000 matrix allows for any level of optimization, including modification of the benchmark source and hand optimization of the source and BLAS. This represents an extreme corner case environment, and the results are heavily influenced by the skill of the person doing the optimization. Only the 100 x 100 results are included here since the more stringent run rules provide for more consistent results across various systems.

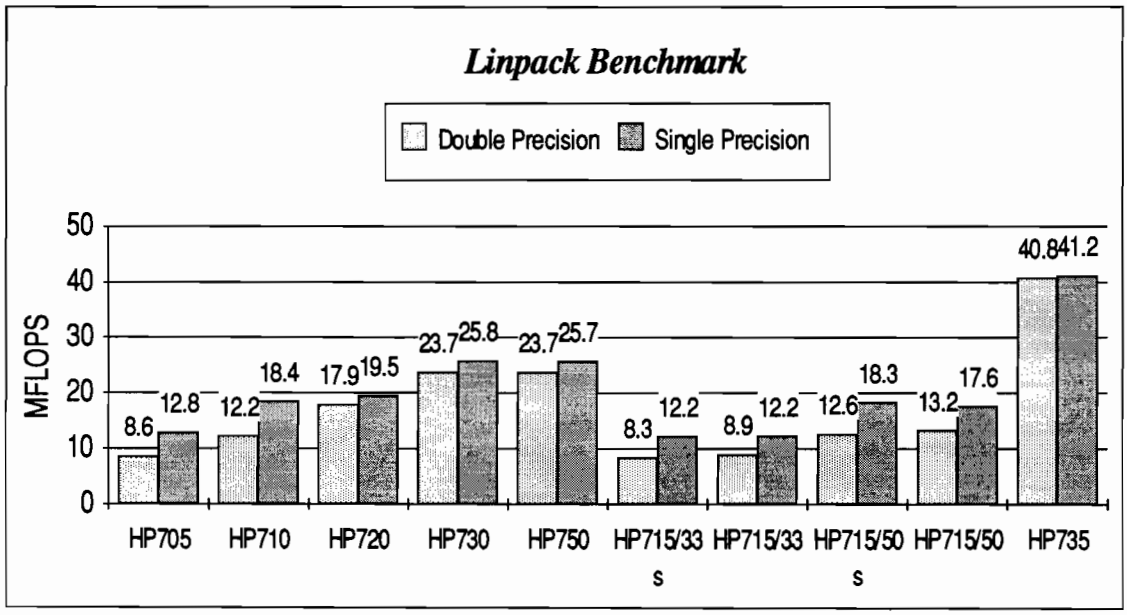


Figure 3.1

Chapter 4 - Whetstone

About Whetstone

The Whetstone benchmark is designed to represent an average program instruction mix for small engineering/scientific applications. It is written in FORTRAN with the instruction mix derived from statistical analysis of 949 ALGOL 60 programs. These instructions include floating point and integer calculations, transcendentals, array manipulation, and conditional jumps. The Whetstone benchmark is designed to defeat many compiler optimizations, including vectorizing.

Whetstone is run in single-precision and double-precision with results expressed in thousands of Whetstone instructions per second (KWIPS).

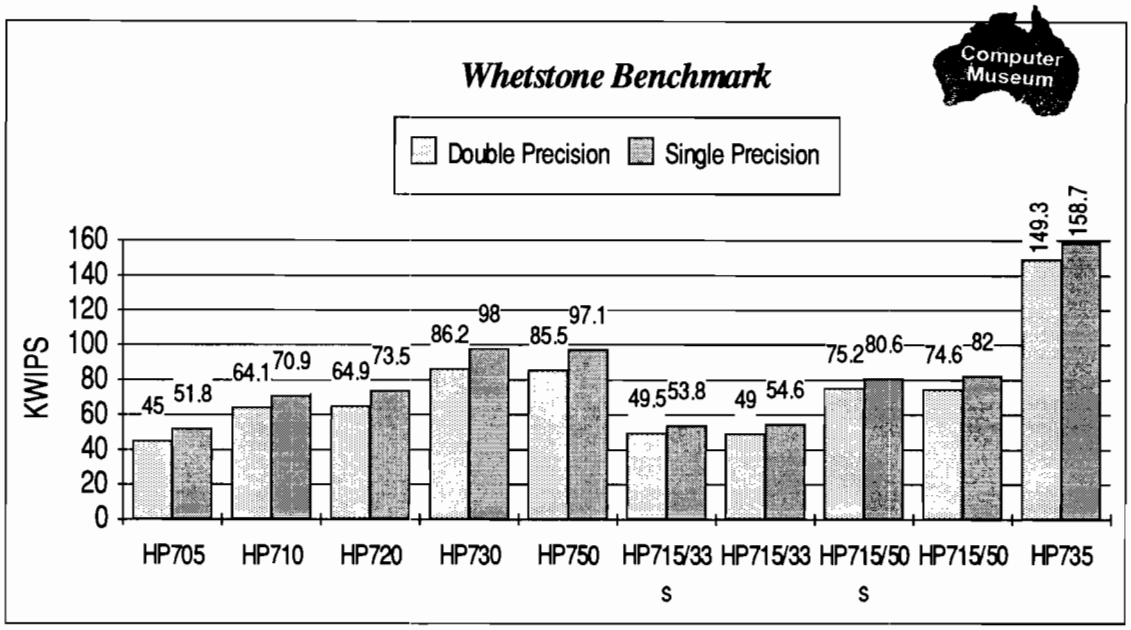


Figure 4.1

Chapter 5 - X11perf

X11perf 1.2 is a public domain benchmark available with the R4 release of the X11 Window System. X11perf measures a fairly broad array of X11 Window System operations with particular emphasis on window manipulation and certain graphics operations. X11perf also provides an accurate client-server synchronization technique based on reading a pixel back from the display. This helps to insure that the primitive has actually been drawn on the display rather than just placed into a graphics pipeline waiting for the hardware to get around to it.

All measurements reported in this section were made by Hewlett-Packard engineers on quiet systems with minimal network activity. While these tests were being run no other X clients were active. Systems tested were configured with at least 16 MB RAM, and were running HP-UX 9.0 with X11R5. X11perf results are not affected by increasing the amount of RAM in the system.

X11perf results for the Hewlett-Packard 700/RX X-stations can be found in Chapter 8.

Overall Operation

The Overall Operation result is the geometric mean of all the x11 perf 1.2 subcategories as defined by *Digital News & Review* (see Appendix D for complete listing). This measurement provides a view of general X Window System performance.

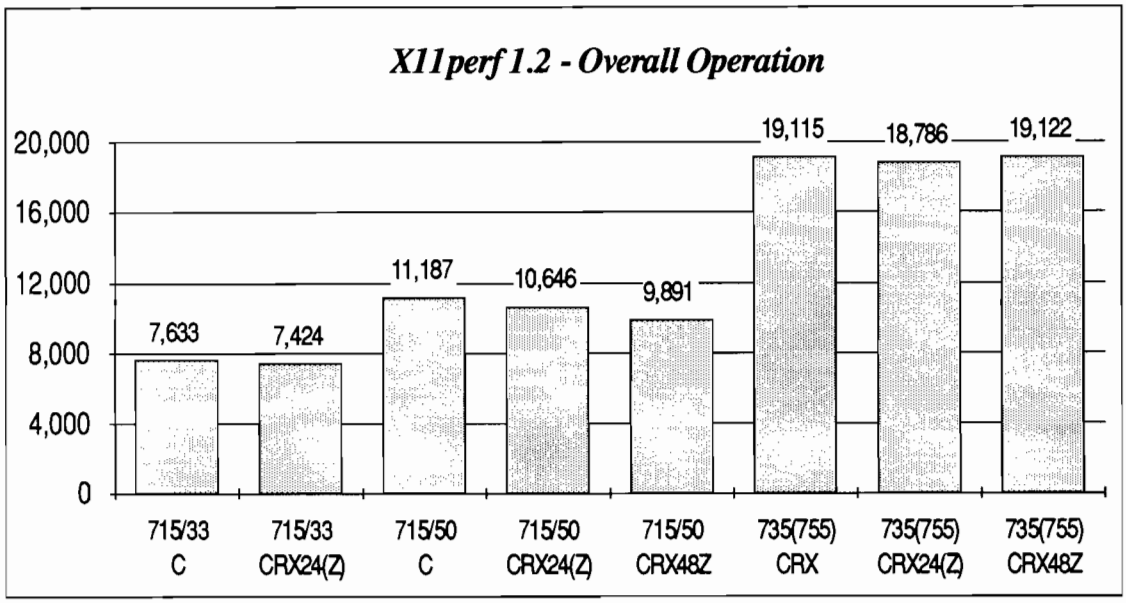


Figure 5.1

Table 5.1 shows selected individual performance measures taken from the X11perf benchmarks.

	715/33 C	715/33 CRX24(Z)	715/50 C	715/50 CRX24(Z)	715/50 CRX48Z	735(755) CRX	735(755) CRX24(Z)	735(755) CRX48Z
X11 Dots/sec	1,060,000	1,050,000	1,610,000	1,600,000	1,440,000	2,870,000	2,540,000	2,610,000
X11 Vec/sec (10-pixel line)	570,000	312,000	860,000	605,000	803,000	971,000	652,000	1,240,000
X11 Rect/sec (10x10 rectangles)	312,000	263,000	341,000	283,000	547,000	287,000	301,000	757,000
X11 Char/sec (Times-Roman 10 font, 80-character line)	186,000	197,000	296,000	279,000	279,000	410,000	360,000	483,000

Table 5.1

Chapter 6 - Graphics Performance Characterization Committee Benchmarks

About the GPC and PLBs

The Graphics Performance Characterization (GPC) committee is composed of the major computer workstation vendors and graphics IC manufacturers. This committee has a goal of providing a standardized and controlled method of measuring and reporting application oriented graphics performance. *The GPC Quarterly Report* provides information about the goals of the committee, a description of the individual GPC benchmarks, the pros and cons of the picture level benchmark (PLB) techniques, configuration details, and test results for several vendors.

Below are the **preliminary** GPC results for Hewlett-Packard systems. They are expected to be close to the official numbers which will be verified and published in an upcoming *GPC Quarterly Report*. The Hewlett-Packard numbers were measured using the Starbase graphics library. However, Hewlett-Packard implementation of the industry standard HP-PHIGS graphics library would be expected to produce similar results. The GPC 3D Solids benchmarks require the PowerShade (2.0 performance version) software option with the at least 16MB of RAM.

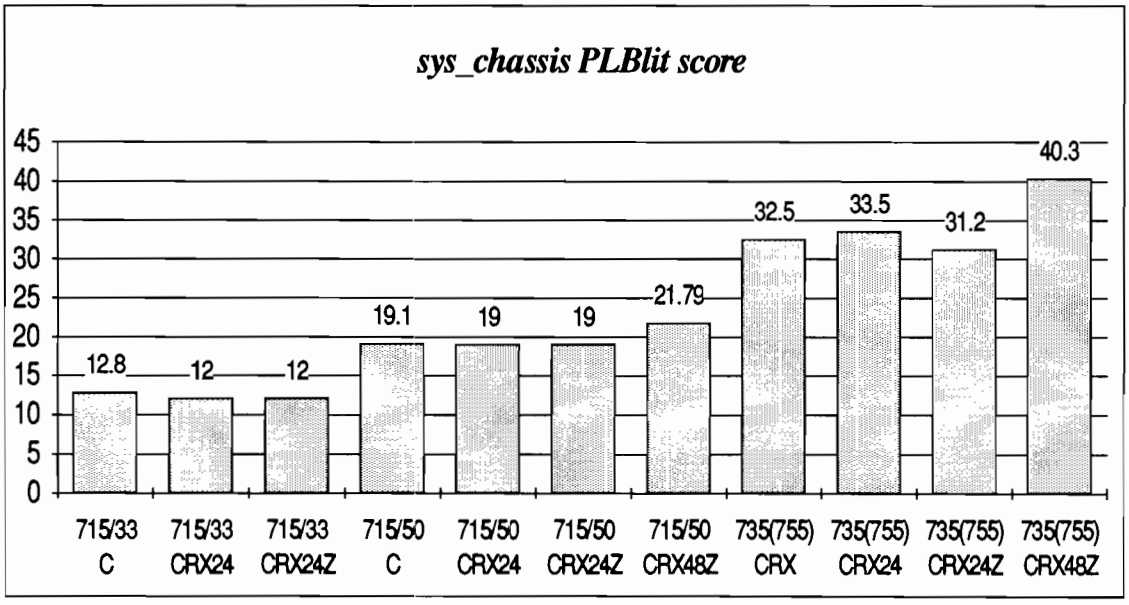
The numbers listed here use the "literal" method of reporting the PLB results (PLBlit). The "literal" method provides an apples-to-apples comparison of different graphics platforms executing equivalent graphics functionality. The other method allows for optimizations which the end-user may or may never see in an application and is not reported here. The PLBlit scores are derived from a normalizing factor divided by the elapsed time for a test. This results in bigger numbers being better.

Each of these benchmarks is designed so that its result will indicate the graphics performance that can be expected in a particular application area. The primitives and attributes displayed are indicative of the type and mix one would expect in a real application.

Results

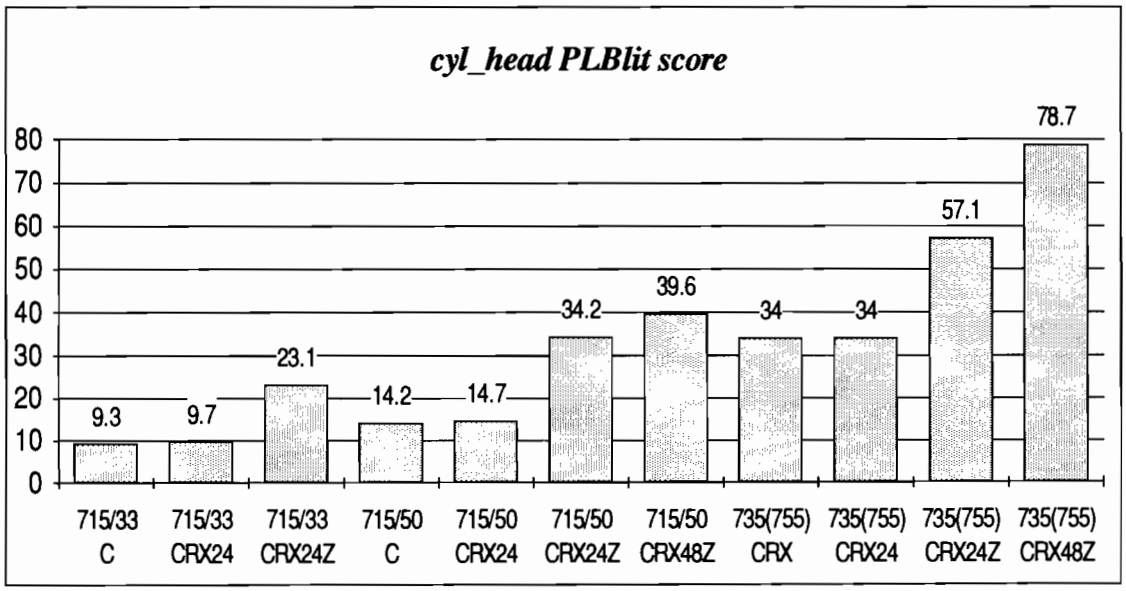
GPC 3D Wireframe: sys_chassis

The sys_chassis PLB is a wireframe model of a computer chassis. It is representative of the type of data required for three dimensional (3D) wireframe applications. 3D wireframe models are typical in mechanical computer aided design applications. The test rotates, pans, zooms, and views the model from different perspectives. The chassis model consists of 6,107 3D solid polylines and 158 3D dashed polylines. The number of vertices per polyline ranges from 2 to 11 with a total of 19,064 vectors. The test displays 500 frames.



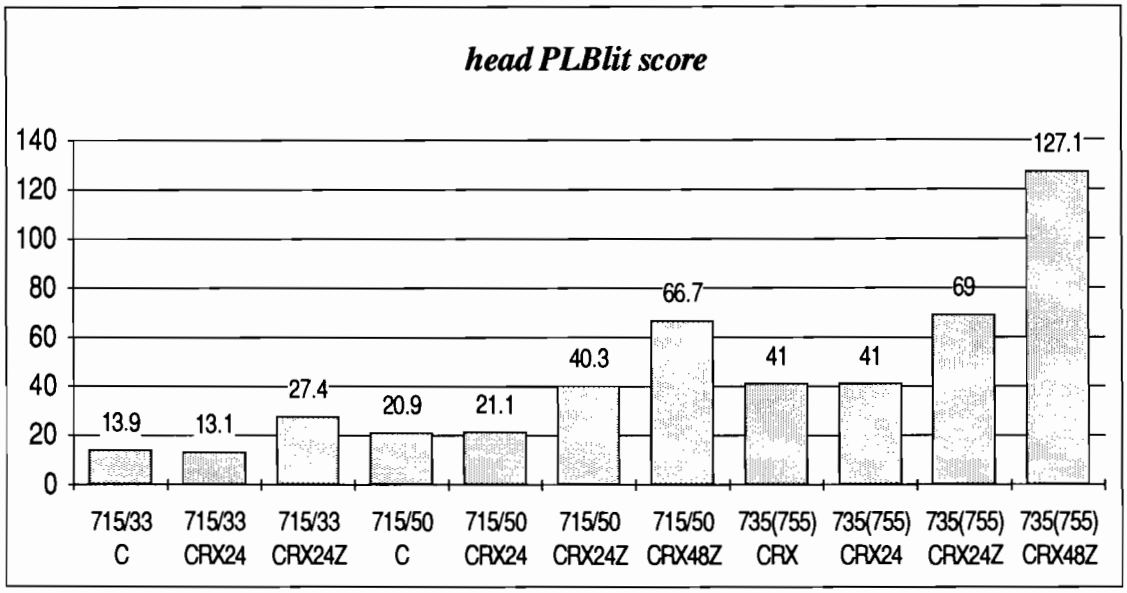
GPC 3D Solids: cyl_head

The cyl_head PLB is 3D solid model of an automobile engine's cylinder head. The head, which is rotated, translated, and zoomed during the test, consists of 3,621 3D polygons, and 32 3D fill area sets. All of the colors are true RGB colors, and all of the polygons are Gouraud shaded. This benchmark uses three light sources with ambient, diffuse, and specular components.



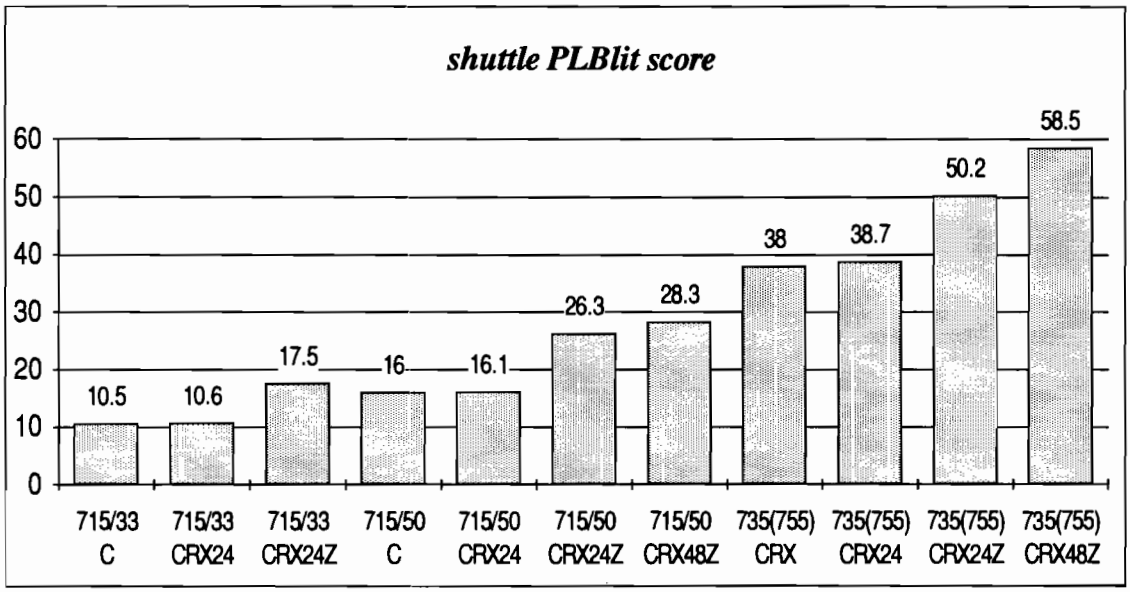
GPC 3D Solids: head

The head PLB is a 3D human head modeled with data generated by a laser scanner. This type of data is typical for application areas such as animation and biomedical. This benchmark generates 240 frames. The data set has nearly 60,000 triangles rendered using triangle strips. There are four directional light sources.



GPC 3D Solids: shuttle

The shuttle benchmark depicts the rendezvous of a space shuttle and a satellite. The shuttle and satellite are modeled using quadrilateral meshes, triangle strips, and polygons. There are a total of 3,355 facets. Three light sources are used. This benchmark provides an example of low end simulation.



Chapter 7 - Traditional Graphics Benchmarks

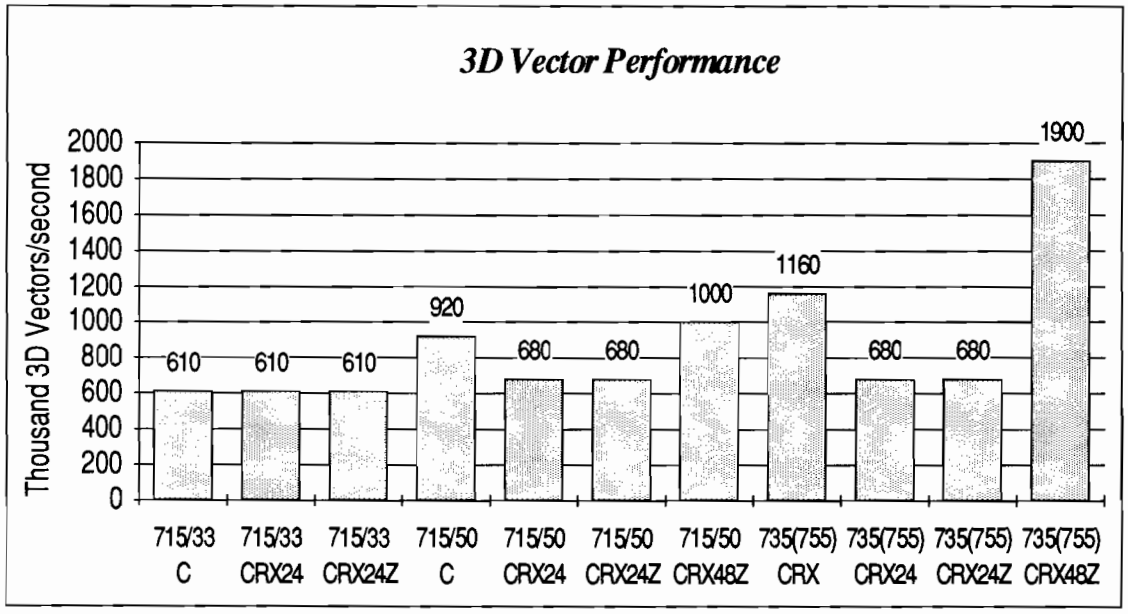
About the Benchmarks

Before the advent of more application oriented GPC standard benchmarks, the typical way to measure and compare graphics performance was to look at the drawing speed for specific graphics primitives. These measures are still used to a large extent in the industry, which is why they are included here. This section looks at 3D vectors/sec, triangles/sec, and light sourced quadrilaterals/sec through the Starbase API using PowerShade (2.0 performance version). HP-PHIGS results should be similar

It is difficult to make accurate comparisons between competing systems using individual benchmarks. This is because each vendor tends to measure the performance under different conditions. Often these conditions are not clearly indicated by the vendor. These different conditions include such items as the size of the primitive and the features that are being used.

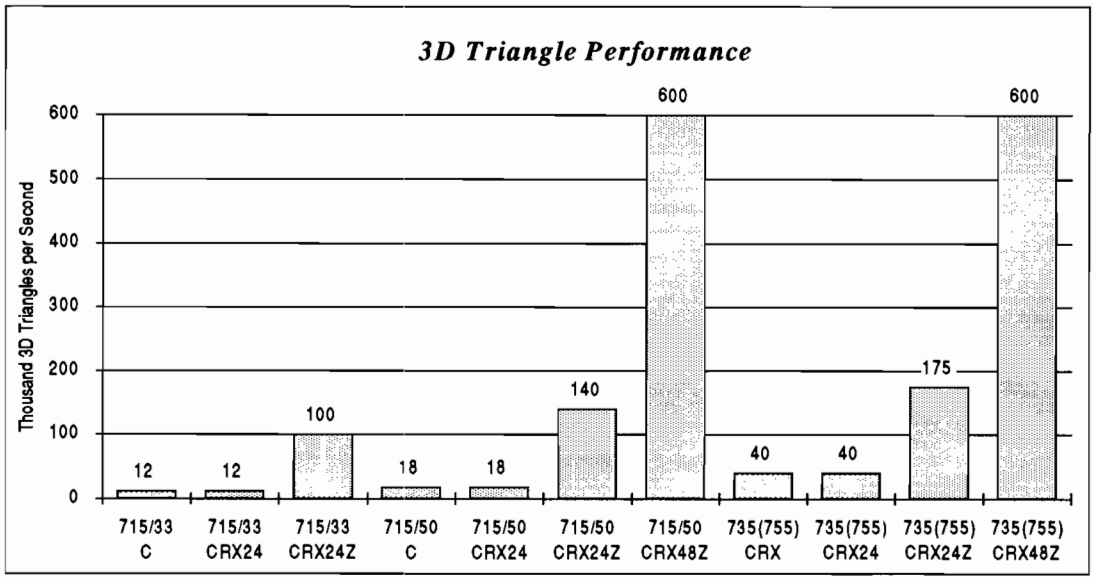
The 3D vector benchmark and 3D light sourced quad benchmark give some indication of how a wireframe or shaded model might perform in an application. The 3D triangle benchmark is included here because it is the only polygonal benchmark commonly provided by SGI. The conditions are similar to the ones defined by SGI. It is not known how any application would make use of long constant color triangular strip primitive.

Results



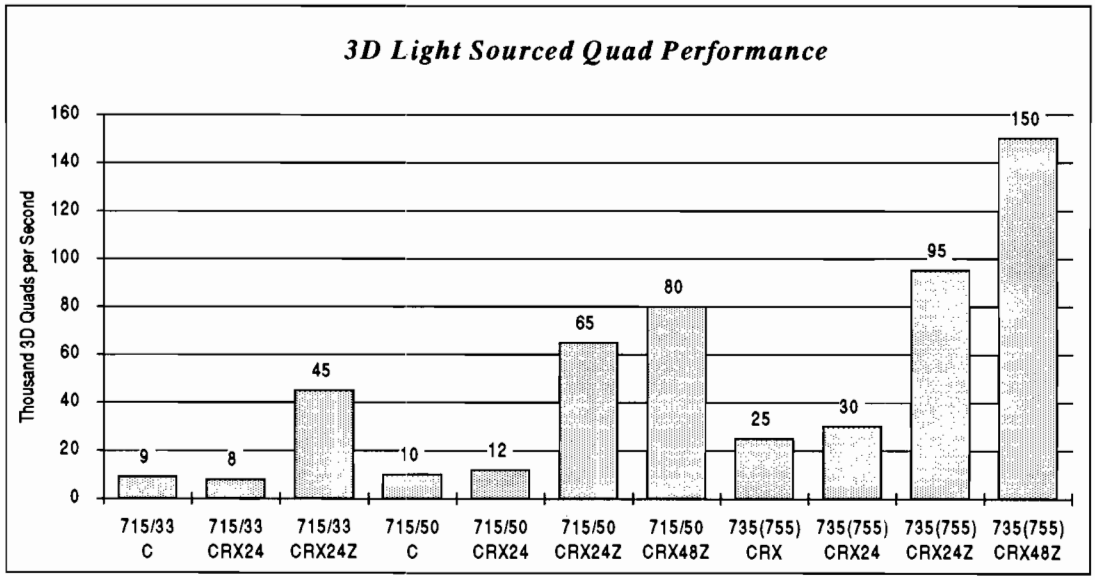
BENCHMARK CONDITIONS:

10 pixel ,constant color, connected, varied orientation,transformed, and clip checked.



BENCHMARK CONDITIONS:

50 pixel triangles in triangular strips, constant color, random orientation, transformed, clip checked, and Z-buffered.



BENCHMARK CONDITIONS:

10x10, 100 pixel independent quadrilaterals, gouraud shaded, random orientation, transformed, clip checked, perspective projection, 1 directional light source + ambient, and Z-buffered.

Chapter 8 - HP 700/RX X-stations Performance

Introduction

This chapter describes the performance of HP 700/RX series of X-stations. Also included in this section is information that will assist in configuring X-stations with the B.04 release of the X server software.

Several new features have been added to HP X-station software that enhance performance and usability. HP's X-stations now offer support for local window managers, terminal emulators and a local HP VUE-like graphical user interface known as VUE/RX. These new features are all local X clients that use both the CPU and memory of the X-station. Additional memory is required to support the enhanced capabilities.

The benchmarks described below were run on the HP 9000 model 720 with HP UX 8.07.

xStone Results

The xStone numbers are one way to compare X-station performance. They are the results of a series of weighted benchmark tests for X servers called from a performance test known as *xbench*. The weights have been scaled to give a monochrome SUN 3/50 running untuned MIT release 3 code an xStone rating of 10000. The weights have been based on the X11 operations the developers of the test found to be used most often.

One can expect that an X-station with an xStone rating of 80,000 will execute eight times as fast as the reference SUN system. Since the test is the same for all the X-stations, it can be used to approximate the relative performance of each X-station .

The Figure 8.1 shows the current benchmark results for all the HP 700/RX stations and some non-HP high-end 19-inch, 1280 x 1024 X-station products. These results are based on the Beta B.04 release of the software and are subject to change with the final release of software.

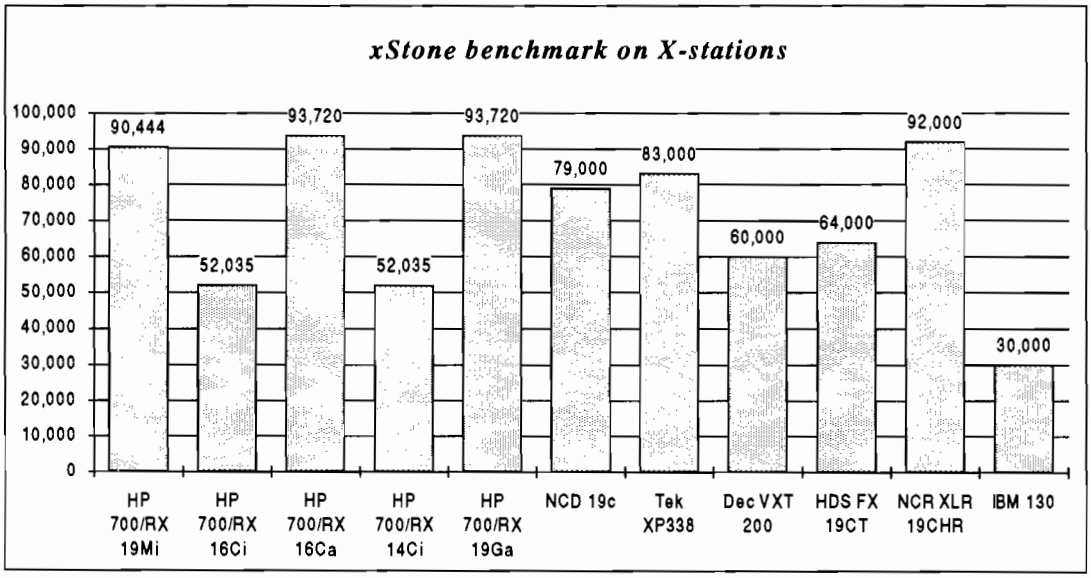


Figure 8.1

X-station X11perf Results

Another benchmark that is used to compare X11 performance on X-stations and workstations is X11perf. The X11perf benchmark is described in Chapter 5 and the operations covered by X11Perf are listed in Appendix D.

These results are based on the Beta B.04 release of the software and are subject to change with the final release of software.

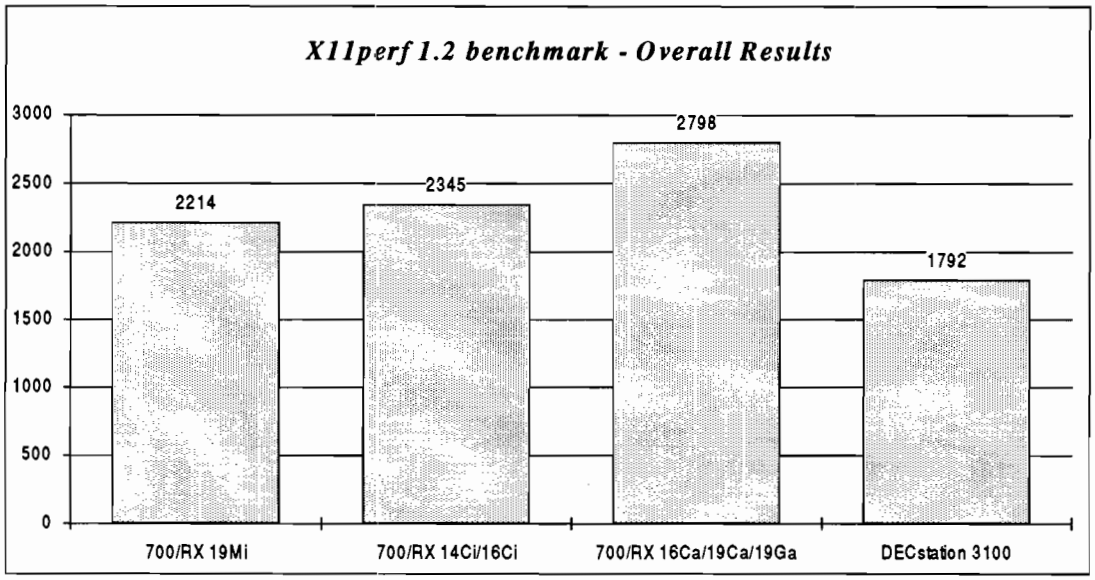


Figure 8.2

X11Perf Composite Results

	19Mi	14Ci, 16Ci	16Ca, 19Ca 19Ga	DECstation 3100
General Graphics	2193	2500	2989	1537
Terminal Emulation	15533	11577	19601	11498
Window Management	1466	869	977	852
X Specific Operations	1900	2291	2513	2512

Table 8.1

Note: Digital Review reports X-station numbers relative to the DECstation 3100 workstation. Results quoted in that magazine are referred to as DXUP's. This is the ratio of the overall geometric mean of the X-station to the DEC 3100. DXUP values can be calculated from Figure 8.2 and Table 8.1.

The B.04 Release of the X Server Software

New Features

There are several new features included in the B.04 software release that will have an effect on the host system as well as the HP 700/RX-station. The features include the ability to run a local window manager, local terminal emulators and a local user environment.

Use of these features will require additional X-station memory, but will reduce the host system memory and system resource requirements. The net effect is that a given host computer system can support more X-stations. The number of X-stations per host system is highly application-dependent.

X-Station Memory Utilization

The following tables will help to estimate the required X station memory.

Local Window Manager [1]	Kbytes Needed
TWM [2]	392
MWM [2]	1656
VUE/RX [3]	2433
VUE WM [4]	2873

Local Clients [1]	Kbytes Needed
Hpterm first innvocation	1126
Hpterm second invocation	324
Xterm first invocation	644
Xterm second invocation	185

X Server [1]	Kbytes Needed
Monochrome 19Mi	2400
Mid Range 16Ci	2600
High Performance 19Ca, 19Ga	2800

1. These results are based on a Beta B.04 release of the software and are subject to change with the final release of software.
2. Alternate window managers; not used on the X-station if either VUE option is selected.
3. New GUI environment with multiple workspace manager that is similiar to HPVue but is not host dependent.
4. The HPVue window manager only runs on the HP 700/RX-station. The other parts of HPVue such as File Manager runs on the host system.

For example, to run VUE/RX plus 3 HPTERM windows the X-station would require a minimum amount of memory equal to 2433 (VUE/RX) + 1126 (HPTERM 1st) + 648 (HPTERM 2nd x 2) = 4207 Kbytes. To this number, the X server code and memory requirements for the applications must be added. Therefore, it is recommended that a total of 10 Mbytes of X-station memory be installed in the HP 700/RX when running local clients.

Chapter 9 - Advanced Optimizing Compilers for PA-RISC

The Optimization Performance Edge

Compiler optimization technology unlocks the performance potential of the PA-RISC architecture. The RISC philosophy has fundamentally changed the role of the compiler. As RISC moves to strike a balance between hardware and software that exploits the best of each technology, the resulting simple, high-performance instruction set gives the compiler more opportunity to apply optimizations that dramatically improve performance. In fact, this opportunity is more of a responsibility. The effectiveness of RISC depends on the compiler's ability to create the most efficient instruction sequence by appropriately rearranging the program steps. Without these optimizations, many applications will execute at a performance level far below their potential.

Indicative of this synergy, PA-RISC depends heavily on the compiler's optimization technology to maximize performance. The PA-RISC compilers offer the level of optimization sophistication required to deliver industry-leading RISC performance. As illustrated in Figure 9.2, the significant performance boost over unoptimized code reflects optimization techniques designed to extract the performance and efficiency inherent in the PA-RISC architecture. For example, on the Model 720 the HP-UX 9.0 optimizations increase the SPECmark from 21.6 to 66.5 (207%), the MIPS rating from 36 to 57 (58%), and the MFLOPS rating from 2.9 to 17.9 (517%).

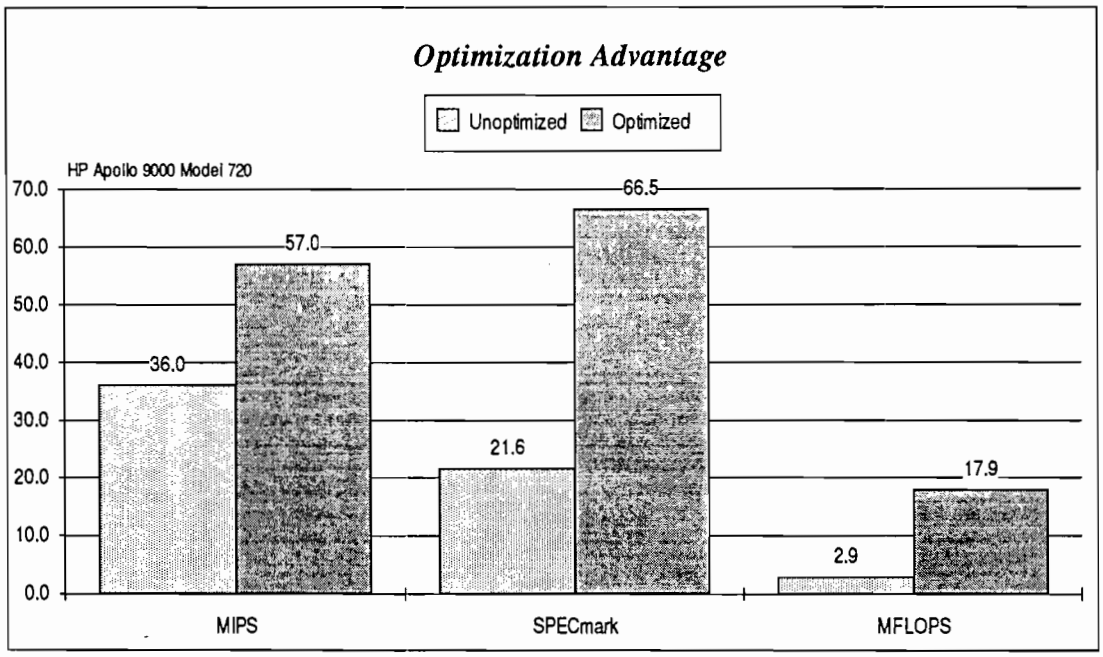


Figure 9.1

Advanced Optimization Technology

The PA-RISC compilers implement a comprehensive set of optimizations geared towards delivering superior performance on a large class of applications running on the PA-RISC processor. Recent optimizer enhancements include the following:

1. Improved Instruction Scheduling: The scheduling heuristics were enhanced to enable the scheduler to generate more optimal schedules, especially systems based on the new superscalar PA7100 processor.
2. Improved Register Allocation: The register allocation algorithm was enhanced to minimize the number of spill instructions generated when the allocator runs out of machine registers.
3. Improved Software Pipelining: Numerically intensive applications often have loops that contain long-latency operations, such as memory accesses and floating-point operations. The compilers use a scheduling technique called software pipelining which overlaps operations from multiple loop iterations in order to hide the long latencies. Enhancements for HP-UX 9.0 include greater integration with register reassociation.
4. Improved Register Reassociation: This optimization reassociates calculations in array subscript expressions to produce more candidates for code motion and strength reduction. Enhancements include increasing the scope beyond just simple innermost loops, as well as greater integration with software pipelining.
5. Improved Branch Optimization: Additional branch optimizations were implemented to minimize the branch penalty along the most frequent path taken. Based upon information from heuristics used to estimate which branches are taken most often, code is repositioned so as to eliminate some of these branches.
6. Profile Based Optimizations: Using an application's execution-profile data, the optimizer rearranges code to improve the instruction cache locality as well as to reduce the number of branches taken. Improving the locality of frequently invoked subroutines results in better utilization of high-speed cache memory.

The continuous enhancement of HP's compilers, in concert with hardware and system software advancements, results in increased performance for each release.

For example, Figure 9.2 shows performance on a Model 720 for successive releases of HP-UX. The SPECmark benchmark performance increased approximately by 36% from HP-UX release 8.01 to 8.05. Performance improved by 12% from HP-UX release 8.07 to 9.0 due to optimizer enhancements.

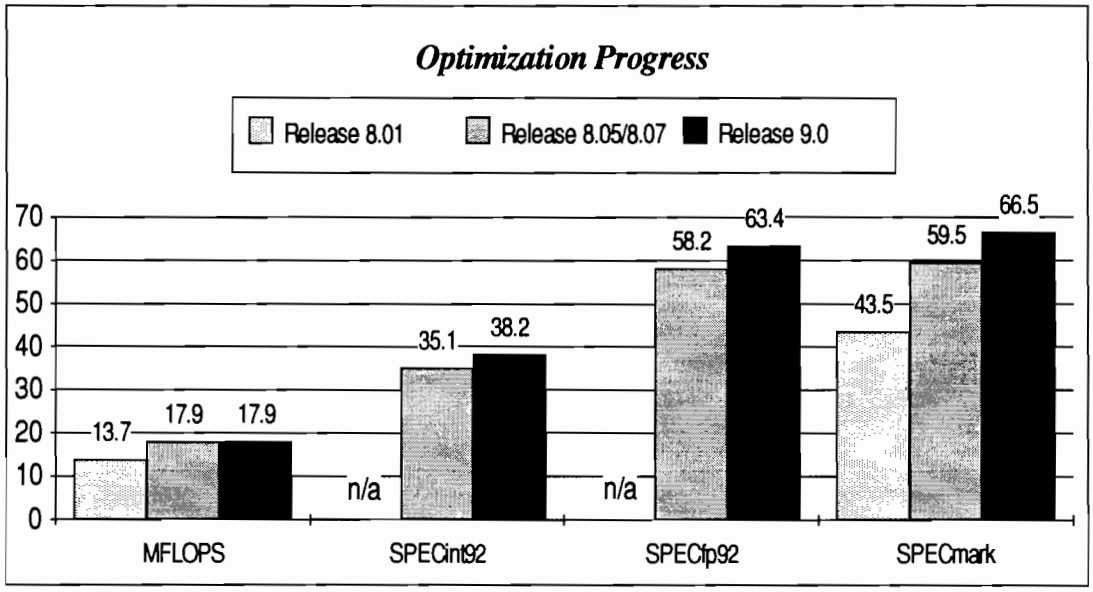


Figure 9.2

FORTRAN Optimizing Preprocessor

The FORTRAN Optimizing Preprocessor analyzes the program by gathering information about the use of data and the nature of the control flow in the program. It then uses this information in conjunction with parameters that describe the machine (such as cache capacity, cache line size, number of registers) in order to restructure source code to achieve data locality.

For example, one restructuring technique called "blocking" treats array operations as multiple operations on blocks of the array, where the size of the block is chosen carefully to match the machine characteristics. This technique can significantly reduce the number of memory loads and cache misses. A number of commonly used numerical algorithms, such as matrix multiplication, benefit substantially from this technique.

Activating Optimizations

The default for PA-RISC compilers is to not optimize your program. Optimizations are activated by specifying particular command-line options. As the analysis required to optimize programs requires additional time and space, the PA-RISC compilers support various levels of optimization for you to control trade-offs between compile-time overhead and code performance improvements.

Table 9.2 provides a summary of optimization options for the C and FORTRAN compilers. For further details, see the Reference Manual and Programming Guide for the five programming languages supported on PA-RISC: C, C++, COBOL, FORTRAN and Pascal.

Table 9.2
C and FORTRAN
Optimization Options Summary

Option	Description
+O1	Instruction scheduling and a subset of optimizations performed on small subsections of code.
-O or +O2	Includes '+O1', plus optimizations performed over the entire scope of each subroutine. This is the generally recommended level for all modules.
+O3	Includes '+O2', plus linker optimizations to improve global variable accesses.
+OP	Invokes the FORTRAN Optimizing Preprocessor (as described in FORTRAN Optimizing Preprocessor).
+I ,+P	Enables profile-based optimizations (as described in Advanced Optimization Technology).

Compiling and Optimizing for Different PA-RISC Architectures

As the PA7000 architecture of the Series 700 workstations is slightly different than the PA-RISC 1.0 architecture of many of the HP 9000 Series 800 systems, code generation and instruction scheduling can be optimized for each architecture. The compilers support the following compile-time options to control these algorithms:

+DS1.0 Instruction scheduling optimized for PA-RISC 1.0.

+DS1.1 Instruction scheduling optimized for PA-RISC 1.1.

+DA1.0 Instruction set of PA-RISC 1.0.

+DA1.1 Instruction set of PA-RISC 1.1.

The default code generated for Series 700 is '+DS1.1 +DA1.1'; and the default code generated for Series 600/800 is '+DS1.0 +DA1.0'. While code generated with any combination of these options will run on the Series 700, the reverse is not true. Code for Series 800 systems based on PA-RISC 1.0 (that is, all models, except the recently introduced 8*7 models) must be compiled using the PA-RISC 1.0 instruction set ('+DA1.0').

A system model number can be used instead of the implementation number (that is, '+DA855' instead of '+DA1.0'). If the program is to be run on one particular target system that is a different HP9000 model than the system used for compiling, then +DA and +DS should be used with the target system's model number. If the program is to be run on many models of the HP9000 (including Series 800 systems) then '+DA1.0' can used to ensure portability and +DS used with the model number of the fastest hardware system.

Appendix A

SPEC benchmark descriptions

SPEC Benchmark Suites

The Standard Performance Evaluation Corporation , (SPEC) is a nonprofit organization formed to develop methods for fairly and effectively characterizing computer system performance. The tests that make up the SPEC Benchmark Suite were selected to provide a mix of integer and floating-point instructions likely to be found in real-world applications.

SPEC has recently released the SPEC92 suites; Cint92 and Cfp92. Both suites build on the popular 1989 release 1.0 of the SPEC Benchmarks by providing better metrics for evaluating systems targeted for commercial and scientic markets. Cint92 is composed of six real-world integer applications. Cfp92 is composed of 14 real world floating-point applications, five of which are single precision.

The SPEC92 suite includes two integer tests and nine floating-point tests not found in the SPEC89 suite. Matrix300 is not included in the Cfp92 suite. All test programs have been modified to address performance issues raised by the current and near-future generations of high-performance CPUs. In addition, the SPECmark summary is no longer included in the results report. This avoids the confusion that results from mixing results of tests designed to measure different system aspects.

The sections below describe the benchmark components of both the new '92 suite and the original release 1 suite now sometime known as SPEC89.

SPECint92

Cint92, which measures integer compute performance, is composed of tests from six different application areas: circuit theory, LISP interpreter, logic design, text compression algorithm, spread sheet, and software development.

Results of the Cint92 suite are reported as the composite metric SPECint92. SPECint92 is the geometric mean of the normalized value of individual test results. A normalized value is the ratio of the measured speed (elapsed time) to the speed measured on a DEC VAX 11/780.

Six Tests of the Cint92 Suite

- espresso: a collection of tools for the generation and optimization of Programmable Logic Arrays. It performs heuristic Boolean function minimization. It is categorized as a CPU-bound integer-intensive application.
- li: a LISP interpreter written in C to solve the 8 Queens problem. It is CPU-intensive.
- eqntott: translates a logical representation of a boolean equation into a truth table. It is an integer-intensive benchmark written in C. It primarily performs sorting; in fact, nearly 95% of its time is spent in the library routine qsort.
- compress: a standard UNIX utility that reduces the size of the named files using adaptive Lempel-Ziv coding. The SPEC version compresses and decompresses a 1 MB file twenty times. It is a CPU-intensive benchmark that performs some I/O.
- sc: a C language benchmark that performs operations typically found in a spreadsheet environment. Such things as cursor movement, data entry, data movement, file handling, row and column operations, operations on ranges, and numeric expressions and evaluations are represented here.
- gcc: a CPU integer-intensive benchmark written in C. It uses the GNU C compiler to convert 19 preprocessed source files into optimized assembly language output.

SPECfp92

Cfp92, which measures floating-point compute performance, is composed of tests from 14 different application areas: circuit design, Monte Carlo simulation, quantum chemistry, optics, robotics, quantum physics, astrophysics, weather prediction, and other scientific and engineering problems. Five of these tests are single precision; the remainder are double precision. Results of the Cfp92 suite are reported as the composite SPECfp92. SPECfp92 is the geometric mean of the normalized value of individual test results. A normalized value is the ratio of the measured speed to the speed measured on a DEC VAX 11/780.

Fourteen Tests in the Cfp92 Suite

- spice2g6: an analog circuit simulation and analysis application written mostly in FORTRAN. It performs most of its operations in double-precision floating-point and makes some use of complex floating point data. Used heavily in the EDA (Electronic Design Automation) markets, spice2g6 is a large program that causes high cache miss rates. More than 80% of assignments are memory to memory.
- doduc: a non-vectorizable floating-point FORTRAN benchmark, using double precision floating-point numbers. It is a Monte Carlo simulation of the time evolution of a thermohydraulic model for nuclear reactor components. It does little I/O, has many short branches and loops, and executes code spread over many subroutines.
- mdljdp2: a double-precision FORTRAN benchmark that represents a quantum chemistry application. It solves the equations of motion for a system of 500 interacting atoms. At each step in the calculation, the positions and velocities of each particle in the model system are used to calculate the configuration energy and pressure through equations of statistical mechanics.
- wave5: a large FORTRAN single-precision floating-point scientific benchmark. It solves Maxwell's equations and particle equations of motion on a Cartesian mesh with a variety of field and particle boundary conditions. It involves 500,000 particles on 50,000 grid points over five time-steps.
- tomcatv: mesh generation program, is a highly vectorizable double-precision floating-point benchmark written in FORTRAN. It generates two-dimensional boundary-fitted coordinate systems around general geometric domains such as airfoils and automobiles. The benchmark favors superscalar and vector processors. It tends to cause high data cache miss rates.
- ora: a CPU-intensive FORTRAN double-precision floating-point scientific application. It traces rays through optical systems composed of spherical and plane surfaces.
- alvinn: is a C language single-precision floating-point robotic application. Alvinn trains a neural network to process video and distance input and generate directional information for vehicle travel. It has 1220 input units representing two input retinas and produces 35 output units.
- ear: C language single-precision floating point benchmark, simulates the human ear. It takes a sound file as input and produces a spectrum of the time dependent representation of the incident sound.
- mdljsp2: the single-precision version of mdljdp described above.
- swm256: a FORTRAN benchmark using single-precision floating-point arithmetic. The program solves a system of shallow-water equations using finite difference approximations on a 256 X 256 grid.
- su2cor: is a vectorizable FORTRAN program doing double-precision floating-point computations in quantum physics. Masses of elementary particles are computed in the context of the Quark-Gluon theory. Data is computed using a Monte Carlo method borrowed from statistical mechanics.

- hydro2d: a vectorizable FORTRAN program doing double-precision floating-point computation. An application from astrophysics, hydrodynamical Navier-Stokes equations are solved to compute galactic jets.
- nasa7: a collection of seven floating-point kernels. These FORTRAN kernels, operating on double-precision data, are: MXM matrix multiply; CFFT2d complex radix 2 fast Fourier transform on two-dimensional array; CHOLSKY Cholesky decomposition in parallel on a set of input matrices; BTRIX block tridiagonal matrix solution along one dimension of a four-dimensional array; GMTR Y Gaussian elimination resulting from a vortex method solution; EMIT creates new vortices according to certain boundary conditions; and VPENT A inverts 3 matrix pentadiagonals in a highly parallel fashion.
- fpppp: performs the two-electron integral derivative that occurs in the Gaussian series of programs used in quantum chemistry. It uses double precision-floating point numbers and is written in FORTRAN. It is difficult to vectorize because it contains very large basic blocks. It does very little I/O.

SPEC89

The integer (non-floating point) benchmarks are:

- gcc - based on the GNU C compiler version 1.35 distributed by the Free Software Foundation. This benchmark measures the time it takes for the GNU C compiler to convert 19 pre-processed source files into optimized Sun-3 assembly language output.
- espresso - one of a collection of tools for the generation and optimization of Programmable Logic Arrays (PLAs). Written in C this benchmark is representative of many EDA applications.
- li - a LISP interpreter written in C. This benchmark measures the time required for li to solve the 9-queens problem.
- eqntott - written in C, translates a logical representation of a boolean equation to a truth table.

The floating point benchmarks are:

- spice2g6 - an analog circuit simulation and analysis application that is heavily used in computer aided electrical engineering. Spice is written in FORTRAN.
- doduc - a non-vectorizable, scalar FORTRAN benchmark. It is a Monte Carlo simulation.
- nasa7 - a collection of seven FORTRAN kernels.
- matrix300 - a vectorizable FORTRAN scientific benchmark. This code performs various matrix manipulations.
- fpppp - a quantum chemistry benchmark written in FORTRAN.
- tomcatv - a vectorized mesh generation FORTRAN program, tomcatv is part of Prof. W. Gentzsch's benchmark suite.

Appendix B

SPEC individual benchmark results and system configurations

SPECint92

	HP705	HP710	HP720	HP730	HP750	HP715/33s	HP715/33	HP715/50s	HP715/50	HP735 & 755
espresso	29.2	41.7	44.2	59.4	58.5	28.4	28.8	43.0	43.7	92.3
li	25.4	37.0	40.4	53.8	53.7	26.2	26.7	39.4	38.7	86.4
eqntott	29.2	41.5	45.3	62.1	60.4	27.6	28.5	41.8	43.7	90.9
compress	18.1	26.1	34.3	45.0	44.2	17.2	18.8	26.2	28.3	66.0
sc	18.9	27.3	34.9	46.1	46.6	20.5	21.5	31.1	32.5	71.7
gcc	18.8	27.1	33.7	47.9	45.6	22.2	22.7	33.5	34.8	76.7
SPECint92	22.7	32.7	38.5	52.0	51.1	23.3	24.2	35.3	36.5	80.0
CPU/MHz	PA-RISC 1.1/35	PA-RISC 1.1/50	PA-RISC 1.1/50	PA-RISC 1.1/66	PA-RISC 1.1/66	PA7100/33	PA7100/33	PA7100/50	PA7100/50	PA7100/99
FPU/MHz	Custom/35	Custom/50	Custom/50	Custom/66	Custom/66	integrated	integrated	integrated	integrated	integrated
Cache	32/64 (I/D)	32/64 (I/D)	128/256(I/D)	128/256(I/D)	256/256(I/D)	64/64(I/D)	64/64(I/D)	64/64(I/D)	64/64(I/D)	256/256(I/D)
Memory (MB)	32	64	64	64	64	32	16	32	32	64
Disk Subsystem	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI internal Quantum LPS525s	1 - SCSI external 2213A 660MB	1 - SCSI internal Quantum LPS525s	1 - Fast-wide SCSI-II 2GB
O/S & Version	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0
C Compilers	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19
Fortran Compilers	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00
Date	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92

SPECfp92

	HP705	HP710	HP720	HP730	HP750	HP715/33s	HP 715/33	HP715/50s	HP715/50	HP735 & 755
spice2g6	25.0	36.0	50.5	65.3	63.0	23.5	26.1	35.4	39.5	91.9
doduc	34.7	50.1	57.8	79.1	76.9	37.7	39.5	57.6	59.0	142.0
mdljdp2	48.3	68.9	74.2	98.7	97.3	58.1	59.1	87.6	89.3	192.1
wave5	26.9	38.9	43.5	58.1	56.7	34.4	16.7	52.3	53.9	112.1
tomcatv	48.1	68.5	74.0	95.0	90.8	51.6	59.6	78.2	90.4	138.0
ora	89.4	127.3	129.9	171.4	168.3	92.3	92.4	139.2	139.2	276.9
alvinn	38.9	55.8	67.1	88.6	87.3	46.8	49.9	71.1	75.3	176.8
ear	57.4	82.7	92.6	122.5	122.4	74.1	80.2	111.8	125.1	258.4
mdljsp2	22.5	32.2	33.5	44.5	44.0	29.4	29.6	44.3	44.7	92.3
swm256	25.8	37.0	38.9	50.0	48.4	29.5	33.3	44.6	50.5	79.3
su2cor	45.0	65.3	92.6	118.6	113.7	43.7	49.0	66.3	76.2	177.2
hydro2d	34.9	50.0	74.9	97.9	95.7	35.7	40.7	54.1	61.5	166.1
nasa7	37.4	55.0	67.1	85.5	83.5	35.6	41.2	53.9	62.2	123.3
fpppp	56.1	80.1	89.1	118.1	117.0	76.9	77.1	115.3	116.2	237.1
SPECfp92	39.3	56.4	66.1	86.7	84.8	44.2	45.0	66.8	72.1	150.6
CPU/MHz	PA-RISC 1.1/35	PA-RISC 1.1/50	PA-RISC 1.1/50	PA-RISC 1.1/66	PA-RISC 1.1/66	PA7100/33	PA7100/33	PA7100/50	PA7100/50	PA7100/99
FPU/MHz	Custom/35	Custom/50	Custom/50	Custom/66	Custom/66	integrated	integrated	integrated	integrated	integrated
Cache	32/64 (I/D)	32/64 (I/D)	128/256(I/D)	128/256(I/D)	256/256(I/D)	64/64(I/D)	64/64(I/D)	64/64(I/D)	64/64(I/D)	256/256(I/D)
Memory (MB)	32	64	64	64	64	32	16	32	32	64
Disk Subsystem	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	1 - SCSI internal Quantum LPS525s	1 - SCSI external 2213A 660MB	1 - SCSI internal Quantum LPS525s	1 - Fast-wide SCSI-II 2GB
O/S & Version	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0
C Compilers	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19
Fortran Compilers	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00
Date	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92

SPEC89

	HP705	HP710	HP720	HP730	HP750	HP715/33s	HP715/33	HP715/50s	HP715/50	HP735 & 755
gcc	20.0	28.7	37.6	52.3	49.5	23	24.1	36.0	36.0	83.2
espresso	29.4	42.3	44.5	58.6	59.0	28.3	28.8	42.4	43.2	92.1
spice2g6	25.0	35.8	49.8	65.2	63.2	23.5	26.2	35.6	39.8	91.7
doduc	34.8	48.8	58.4	77.3	77.9	37	39.2	56.8	59.0	140.1
nasa7	40.9	61.0	76.0	100.1	95.8	38.9	45.3	58.8	68.9	139.8
li	25.5	36.4	40.4	53.6	53.5	25.7	26.2	39.1	38.5	86.3
eqntott	29.1	41.4	47.0	61.8	61.5	27	28.3	40.6	43.5	91.0
matrix 300	195.0	272.6	320.9	415.1	422.9	293.8	316.4	443.6	476.3	1103.7
fpppp	62.6	90.2	102.0	133.3	135.0	90.4	92.1	135.6	136.9	281.3
tomcatv	48.3	69.5	73.2	94.3	91.0	51.1	58.7	77.4	89.5	137.9
SPEC mark89	39.5	56.5	66.5	87.6	86.6	42.8	45.9	65.0	69.0	146.8
SPEC int89	25.7	36.8	42.2	56.4	55.7	25.9	26.8	39.5	40.2	88.1
SPEC fp89	52.5	75.3	89.9	117.5	116.1	59.8	65.6	90.6	99.0	206.2
CPU/MHz	PA-RISC 1.1/35	PA-RISC 1.1/50	PA-RISC 1.1/50	PA-RISC 1.1/66	PA-RISC 1.1/66	PA7100/33	PA7100/33	PA7100/50	PA7100/50	PA7100/99
FPU/MHz	Custom/35	Custom/50	Custom/50	Custom/66	Custom/66	integrated	integrated	integrated	integrated	integrated
Cache	32/64 (I/D)	32/64 (I/D)	128/256(I/D)	128/256(I/D)	256/256(I/D)	64/64(I/D)	64/64(I/D)	64/64(I/D)	64/64(I/D)	256/256(I/D)
Memory (MB)	32	64	64	64	64	32	16	32	32	64
Disk Subsystem	1 - SCSI external 2213A 660MB	1 - SCSI external 2213A 660MB	2 - SCSI Quantum 400MB	1 - SCSI external 2213A 660MB	2 - SCSI Quantum 400MB	1 - SCSI external 2213A 660MB	1 - SCSI internal Quantum LPS525s	1 - SCSI external 2213A 660MB	1 - SCSI internal Quantum LPS525s	1 - Fast-wide SCSI-II 2GB
O/S & Version	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0	HP-UX 9.0
C Compilers	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.16	HP92453-01 A.09.19	HP92453-01 A.09.16	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19	HP92453-01 A.09.19
Fortran Compilers	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00	B2408 A.09.00
Date	Sep-92	Sep-92	Aug-92	Sep-92	Aug-92	Sep-92	Sep-92	Sep-92	Sep-92	Sep-92

Appendix C

Configuration and Compilation Details

The Linpack, Dhrystone, and Whetstone benchmarks were run on the configurations listed in Appendix B.

SPECmark Release 1.2b

001.gcc1.35/M.hp.700
```
OPT=+O3
EXTRA_CFLAGS = +Obb763 +ESlit
EXTRA_LDFLAGS =-Wl,-aarchive
EXTRA_LIBS =/usr/old/malloc3c.o
```
008.espresso/M.hp.700
```
OPT = +O3
EXTRA_LDFLAGS =-Wl,-aarchive
EXTRA_LIBS = /usr/old/malloc3c.o
```
013.spice2g6/M.hp.700
```
EXTRA_FFLAGS = -w +OP3 -WP,'-nv -ind=-1' +Obb1000
EXTRA_LDFLAGS =+OP3 -Wl,-aarchive
```
015.doduc/M.hp.700
```
OPT=+OP
EXTRA_FFLAGS = -w -WP,'-nv -inline -ind=3 -inff=.'
EXTRA_LDFLAGS =-Wl,-aarchive
```
020.nasa7/M.hp.700
```
OPT=-O +OP3
EXTRA_FFLAGS = -w -WP,'-aggressive=a'
EXTRA_LDFLAGS =-Wl,-aarchive
```
022.li/M.hp.700
```
OPT=+O3
EXTRA_LDFLAGS =-Wl,-aarchive
```
023.eqntott/M.hp.700
```
OPT=-O
EXTRA_LDFLAGS =-Wl,-aarchive
```
030.matrix300/M.hp.700
```
OPT=+OP4
EXTRA_FFLAGS =-w +Obb600 -WP,'-inlr'
EXTRA_LDFLAGS =-Wl,-aarchive
```
042.fpppp/M.hp.700
```
OPT = +OP
EXTRA_FFLAGS = -w +Os
EXTRA_LDFLAGS =-Wl,-aarchive
```
047.tomcatv/M.hp.700
```
OPT=+OP3
EXTRA_FFLAGS = -w
EXTRA_LDFLAGS =-Wl,-aarchive
```

Cfp92

013.spice2g6/M.hp.700
EXTRA_FFLAGS=+OP3 +Os -WP,'-nv -ind=-1' +Obb1000
EXTRA_LDFLAGS=+OP3 -Wl,-aarchive
015.doduc/M.hp.700
OPT=+OP
EXTRA_FFLAGS=-w -WP,'-nv -inline -ind=3 -inff=.'
EXTRA_LDFLAGS=-Wl,-aarchive
034.mdljdp2/M.hp.700
OPT=+OP
EXTRA_FFLAGS=-w
EXTRA_LDFLAGS=-Wl,-aarchive
039.wave5/M.hp.700
OPT=+OP3
EXTRA_FFLAGS=-w
EXTRA_LDFLAGS=-Wl,-aarchive
047.tomcatv/M.hp.700
OPT=+OP3
EXTRA_FFLAGS=-w
EXTRA_LDFLAGS=-Wl,-aarchive

048.ora/M.hp.700
OPT=+OP3
EXTRA_FFLAGS=-w
EXTRA_LDFLAGS=-Wl,-aarchive
052.alvinn/M.hp.700
OPT=-O
EXTRA_CFLAGS= -Aa +e
EXTRA_LDFLAGS= -Wl,-aarchive
056.ear/M.hp.700
OPT=-O
EXTRA_CFLAGS= -Aa -J -D_HPUX_SOURCE -Dfloat=double
EXTRA_LDFLAGS= -Wl,-aarchive
077.mdljsp2/M.hp.700
OPT=+OP
EXTRA_FFLAGS=-w +Os
EXTRA_LDFLAGS=-Wl,-aarchive
078.swm256/M.hp.700
EXTRA_FFLAGS=-w
EXTRA_LDFLAGS=-Wl,-aarchive
089.su2cor/M.hp.700
OPT=+OP3
EXTRA_FFLAGS=-w
EXTRA_LDFLAGS=-Wl,-aarchive
090.hydro2d/M.hp.700
OPT=+OP3
EXTRA_FFLAGS=-w
EXTRA_LDFLAGS=-Wl,-aarchive
093.nasa7/M.hp.700
OPT=+O3
EXTRA_CFLAGS= -DTIMES
EXTRA_FFLAGS=-w -WP,'-aggressive=a' +OP3
EXTRA_LDFLAGS=-Wl,-aarchive +OP3
094.fpppp/M.hp.700
OPT=+OP
EXTRA_FFLAGS=-w +Os
EXTRA_LDFLAGS=-Wl,-aarchive

Cint92

008.espresso/M.hp.700
OPT=+O3
EXTRA_LDFLAGS=-Wl,-aarchive
#The following line will only work with HP-UX 8.07 and later
EXTRA_LIBS=/usr/old/malloc3c.o
022.li/M.hp.700
OPT=+O3
EXTRA_LDFLAGS=-Wl,-aarchive
023.eqntott/M.hp.700
OPT=+O3
EXTRA_LDFLAGS=-Wl,-aarchive
026.compress/M.hp.700
OPT=-O
EXTRA_LDFLAGS=-Wl,-aarchive
072.sc/M.hp.700
OPT=+O3
EXTRA_LIBS=-lcurses -lm
EXTRA_LDFLAGS=-Wl,-aarchive
SIMPLE=-DSIMPLE
085.gcc/M.hp.700
OPT=+O3
EXTRA_CFLAGS=+Obb763
EXTRA_LDFLAGS=-Wl,-aarchive
#The following line will only work with HP-UX 8.07 and later
EXTRA_LIBS=/usr/old/malloc3c.o

Dhrystone 2

```
TIME_FUNC = -DTIMES #                   # Use times(2) for measurement
HZ = 100        #               # Frequency of times(2) clock ticks
STRUCTASSIGN= #                         # Compiler can assign structs
ENUMS= #                        # Compiler does have enum type
OPTIMIZE = +O3 -J #                     # Optimization Level
LFLAGS=         -Wl,-aarchive #                 #Loader Flags
```

Dhrystone 1.1

```
CFLAGS = +O3 -J $(FFPA)
FFLAGS = +O3 -J $(FFPA)
```

Linpack

PA7000 Systems

```
CFLAGS = -O
FFLAGS = +OP3 -Wl,-aarchive -w
```

PA7100 Systems

```
CFLAGS = -O
FFLAGS = +OP3 -Wl,-aarchive -WP,-nv -w
```

Appendix D

X11perf Details

The x11perf benchmark consists the following individual X component tests. The four separapte categories are as set forth by *Digital Review* magazine in May 1990. This methodology has been followed in reporting the HP 700/RX X-station test results.

The Overall Operations category includes all of the tests included under 'General Graphics', 'Terminal Emulation', Window Manipulation', and 'X-specific Operations.' A complete listing of the individual tests in each category is included below.

General Graphics		
Dot	10x1 wide line	10-pixel ellipse
10x10 rectangle	100x10 wide line	100-pixel ellipse
100x100 rectangle	100x10 wide dashed line	100-pixel dashed ellipse
10x10 stippled rectangle	100x10 wide double-dashed line	100-pixel double-dashed ellipse
100x100 stippled rectangle	10-pixel circle	10-pixel wide ellipse
10x10 opaque stippled rectangle	100-pixel circle	100-pixel wide ellipse
100x100 opaque stippled rectangle	100-pixel dashed circle	100-pixel wide dashed ellipse
10x10 4x4 tiled rectangle	100-pixel double-dashed circle	100-pixel wide double-dashed ellipse
100x100 4x4 tiled rectangle	10-pixel wide circle	10-pixel partial ellipse
10x10 161x145 tiled rectangle	100-pixel wide circle	100-pixel partial ellipse
100x100 161x145 tiled rectangle	100-pixel wide dashed circle	10-pixel filled ellipse
10-pixel line segment	100-pixel wide double-dashed circle	100-pixel filled ellipse
100-pixel line segment	10-pixel partial circle	10-pixel fill chord partial ellipse
100-pixel line segment (1 kid)	100-pixel partial circle	100-pixel fill chord ellipse
100-pixel line segment (2 kids)	10-pixel solid circle	10-pixel fill slice partial ellipse
100-pixel line segment (3 kids)	100-pixel solid circle	100-pixel fill slice ellipse
10-pixel dashed segment	10-pixel fill chord partial circle	Fill 10-pixel/side triangle
100-pixel dashed segment	100-pixel fill chord partial circle	Fill 100-pixel/side triangle
100-pixel double-dashed segment	10-pixel fill slice partial circle	Fill 10x10 trapezoid
10-pixel line	100-pixel fill slice partial circle	Fill 100x100 trapezoid
100-pixel line	Fill 100x100 opaque stippled trapezoid	Fill 10x10 stippled trapezoid
10-pixel dashed line	Fill 10x10 tiled trapezoid	Fill 100x100 stippled trapezoid
100-pixel dashed line	Fill 100x100 tiled trapezoid	Fill 10x10 opaque stippled trapezoid
100-pixel double-dashed line	Fill 10-pixel/side complex polygon	
	Fill 100-pixel/side complex polygons	

Terminal Emulation
Char in 80-char line (6x13)
Char in 80-char line (TR 10)
Char in 30-char line (TR 24)
Char in 20/40/20 line (6x13, TR 10)
Char in 80-char image line (6x13)
Char in 80-char image line (TR 10)
Char in 30-char image line (TR 24)
Scroll 10x10 pixels
Scroll 100x100 pixels

Window Management
Copy 10x10 from window to window
Copy 100x100 from window to window
Copy 10x10 from pixmap to window
Copy 100x100 from pixmap to window
Copy 10x10 from window to pixmap
Copy 100x100 from window to pixmap
Copy 10x10 from pixmap to pixmap
Copy 100x100 from pixmap to pixmap
Copy 10x10 1-bit deep plane
Copy 100x100 1-bit deep plane
PutImage 10x10 square
PutImage 100x100 square
GetImage 10x10 square
GetImage 100x100 square

X-specific Operations		
X protocol	Destroy window via parent (4 kids)	Resize window (4 kids)
NoOperation	Destroy window via parent (16 kids)	Resize window (16 kids)
GetAtomName	Destroy window via parent (25 kids)	Resize window (25 kids)
GetProperty	Destroy window via parent (50 kids)	Resize window (50 kids)
Change graphics context	Destroy window via parent (75 kids)	Resize window (75 kids)
Create and map subwindows (4 kids)	Destroy window via parent (100 kids)	Resize window (100 kids)
Create and map subwindows (16 kids)	Destroy window via parent (200 kids)	Resize window (200 kids)
Create and map subwindows (25 kids)	Hide/expose window via popup (4 kids)	Resize unmapped window (4 kids)
Create and map subwindows (50 kids)	Hide/expose window via popup (16 kids)	Resize unmapped window (16 kids)
Create and map subwindows (75 kids)	Hide/expose window via popup (25 kids)	Resize unmapped window (25 kids)
Create and map subwindows (100 kids)	Hide/expose window via popup (50 kids)	Resize unmapped window (50 kids)
Create and map subwindows (200 kids)	Hide/expose window via popup (75 kids)	Resize unmapped window (75 kids)
Create unmapped window (4 kids)	Hide/expose window via popup (100 kids)	Resize unmapped window (100 kids)
Create unmapped window (16 kids)	Hide/expose window via popup (200 kids)	Resize unmapped window (200 kids)
Create unmapped window (25 kids)	Move window (4 kids)	Circulate window (4 kids)
Create unmapped window (50 kids)	Move window (16 kids)	Circulate window (16 kids)
Create unmapped window (75 kids)	Move window (25 kids)	Circulate window (25 kids)
Create unmapped window (100 kids)	Move window (50 kids)	Circulate window (50 kids)
Create unmapped window (200 kids)	Move window (75 kids)	Circulate window (75 kids)
Map window via parent (4 kids)	Move window (100 kids)	Circulate window (100 kids)
Map window via parent (16 kids)	Move window (200 kids)	Circulate window (200 kids)
Map window via parent (25 kids)	Moved unmapped window (4 kids)	Circulate Unmapped window (4 kids)
Map window via parent (50 kids)	Moved unmapped window (16 kids)	Circulate Unmapped window (16 kids)
Map window via parent (75 kids)	Moved unmapped window (25 kids)	Circulate Unmapped window (25 kids)
Map window via parent (100 kids)	Moved unmapped window (50 kids)	Circulate Unmapped window (50 kids)
Map window via parent (200 kids)	Moved unmapped window (75 kids)	Circulate Unmapped window (75 kids)
Unmap window via parent (4 kids)	Moved unmapped window (100 kids)	Circulate Unmapped window (100 kids)
Unmap window via parent (16 kids)	Moved unmapped window (200 kids)	Circulate Unmapped window (200 kids)
Unmap window via parent (25 kids)	Move window via parent (4 kids)	
Unmap window via parent (50 kids)	Move window via parent (16 kids)	
Unmap window via parent (75 kids)	Move window via parent (25 kids)	
Unmap window via parent (100 kids)	Move window via parent (50 kids)	
Unmap window via parent (200 kids)	Move window via parent (75 kids)	
	Move window via parent (100 kids)	
	Move window via parent (200 kids)	

Performance Quick Reference Card

The next page contains listings of workstation class systems, popular performance measures, and system specifications. Please be aware that the performance results undergo rapid change.

The page has been printed separately for your convenience and is included in this kit.

*UN*X Systems - Performance Quick Reference Card*										HP makes no guarantee as to the accuracy of this information					November, 1992
HP/Apollo 9000 Workstation Systems	**SPEC int92**	**SPEC fp92**	**dhry-stone MIPS**	**SPEC mark '89**	**SPEC int '89**	**SPEC fp '89**	**Linpack dbl MFlops**	**X11 lines K vectors per sec**	**X11Perf Overall**	**GPC sys-chassis (literal)**	**GPC cyl-head (literal)**	**CPU/FPU (Clock rate) MHz**	**Cache Instr/ Data KBytes per CPU**	**RAM base-max MBytes**	**I/O Slots**	**OS & version**
425t/s	11.2		22	11.8	12.9	11.0	1.7			n/a	nr	MC 68040/int (25)	int 4/4 (s - ext 128)	8-64t -128s	s-4, t-0	UX 8.x
433s	14.1		30	14.9	16.6	13.8	2.2				nr	MC 68040/int (33)	int 4/4 (s - ext 128)	8-128	s-4, t-0	UX 8.x
705C UX 8.x	21.9	33.0	40	34.6	24.6	43.4	8.4		4,840	13.6	5.8	PA 7000/HP7000 (33)	32/64	8-64	--	UX 8.x
705C UX 9.0	22.7	39.3	40	39.5	25.7	52.5	8.6					PA 7000/HP7000 (33)	32/64	8-64	--	UX 9.0
710C UX 8.x	31.6	47.6	57	49.7	35.4	62.4	12.2	580	7,292	19.3	8.1	PA 7000/HP7000 (50)	32/64	16-128	--	UX 8.x
710C UX 9.0	32.7	56.4	57	56.5	36.8	75.3	12.2					PA 7000/HP7000 (50)	32/64	16-128	--	UX 9.0
715/33CRX24Z	24.2	45.0	41	45.9	26.8	65.6	8.9	312	7,424	12.0	23.1	PA 7100/int(33)	64/64	8-192	1 EISA opt.	UX 9.0
715/33 t/s (400 upgde)	23.3	44.2	41	42.8	25.9	59.8	8.3					PA 7100/int(33)	64/64	8-64t-128s	t-4 EISA	UX 9.0
715/50CRX24Z	36.5	72.1	62	69.0	40.2	99.0	13.2	605	10,646	19.0	34.2	PA 7100/int (50)	64/64	16-256	1 EISA	UX 9.0
715/50 t/s (400 upgde)	35.3	66.8	62	65.0	39.5	90.6	12.6					PA 7100/int (50)	64/64	16-128	t-4 EISA	UX 9.0
720CRX24Z UX 8.x	36.4	58.2	58	59.5	39.5	78.3	17.9	600	7,804	21.2	35.3	PA 7000/HP7000 (50)	128/256	16-256	1 EISA	UX 8.x
720CRX24Z UX 9.0	38.5	66.1	58	66.5	42.2	89.9	17.9					PA 7000/HP7000 (50)	128/256	16-256	1 EISA	UX 9.0
730CRX24Z UX 8.x	47.8	75.4	76	76.8	51.2	100.6	22.0	623	10,118	27.1	43.7	PA 7000/HP7000 (66)	128/256	16-128	1 EISA	UX 8.x
730CRX24Z UX 9.0	52.0	86.7	76	87.6	55.7	116.1	23.7					PA 7000/HP7000 (66)	128/256	16-400	1 EISA	UX 9.0
745i, 747i	36.0	72.0	62	69.0	40.2	99.0	13.0					PA 7100/int (50)	64/64		4 EISA	UX 9.0
735CRX48Z	80.0	150.6	124	146.8	88.1	206.2	40.8	1240	19,122	40.3	78.7	PA 7100/int (99)	256/256	32-400	1 EISA	UX 9.0
750CRX UX 8.x	48.1	75.0	76	77.5	51.5	101.6	23.7	915	10,904	27.1	43.7	PA 7000/HP7000 (66)	256/256	32-	4 EISA	UX 8.x
750CRX UX 9.0	51.1	84.8	76	86.6	55.7	116.1	23.7					PA 7000/HP7000 (66)	256/256	32-	4 EISA	UX 9.0
755CRX48Z	80.0	150.6	124	146.8	88.1	206.2	40.8	1240	19,122	40.3	78.7	PA 7100/int (99)	256/256	64-768	4 EISA	UX 9.0
Compaq Dskpro 486/33	18.2	8.3										i80486DX/33	int-8I&D,ext-128I&D	?	?	SVR 4.0
Data Gen'l Aviion/4605	26.1	nr										MC88100/int (33)	64/32	?		DG/UX 5.4.1
DECstation 5000-20	13.7	14.8	22	16.3	13.5	18.4	2.4					R3000A/R3010A(20)	64/128	?	2	Ultrix v4.2a
DECstation 5000-133	20.1	23.5	34	26.5	23.3	29.0	5.9			13.6pxg+	16.7pxg+	R3000A/R3010A(33)	64/128	-128	3	Ultrix v4.2a
DECstation 5000-240	27.3	29.9	43	32.4	27.9	35.8	6.0		3,256	13.7pxg+	17.6pxg+	R3000A/R3010A(40)	64/64			Ultrix v4.2a
Intel Xpress 486DX50	30.1	14.0										i80486/int(50)	int-8I&D ext-256I&D	?		SVR 4.2 beta
IBM RS/6000 220	16.6	26.1	30	27.7	17.8	37.4	6.5		2,000	29.5 GTO	11.5 GTO	POWER 3308/int (33)	int 8 I&D	?		AIX 3.2
IBM RS/6000 530H	28.5	64.6	nr	62.2	29.3	102.7	20.2					POWER 3363/int (33)	int 8/64	?		AIX 3.2
IBM RS/6000 580	59.1	124.8	nr	126.4	61.3	204.8	38.1		*announced 9/22/92*			POWER 6264/int (62.5	int 32/64	64-1GB	8	AIX 3.2.3
IBM RS/6000 970	47.8	101.0	nr	103.4	49.3	169.6	30.7					POWER 5064/int (50)	int-32/64	512	7-8	AIX 3.2.1
SGI Indigo R3000 Elan	22.4	24.2	30	26.3	23.6	28.4	4.2	258	4,339			R3000A/R3010A(33)	ext 32/32	-96	2	SunOS 4.1.2
SGI Indigo R4000 Elan	57.6	60.3	85	70.2	61.1	77.0	16.0	389	6,384			R4000/int (100)	int 8/8, ext 1MB	-384	2	IRIX 4.05
SGI Crimson Elan	58.3	61.5	85	70.4	60.6	77.8	16.3	389	6,384			R4000/int (100)	int 8/8, ext 1MB		4	IRIX 4.05
Sun SPARCstation IPX	21.8	21.5	29	24.4	21.7	26.5	4.2		2,670	10.8	nr	FJMB86903/int (40)	ext 64 I&D	-128	2	SunOS 4.1.2
Sun SPARCstation 2	21.8	22.7	29	25.0	21.7	27.4	4.3			17.4 GT	11.9 GT	CY7C601/int (40)	ext 64 I&D	?		
Sun SS10 model 30	44.1	52.8	86	57.3 (est.)		nr	10.6	262				TMS390Z50 / (36)	int-20/16	512	4	SunOS 4.1.3
Sun SS10 model 41	53.3	65.1	96	71.2 (est.)		nr	17.2					TMS390Z50 / (40)	int-20/16, ext-1MB	512	4	SunOS 4.1.3
								HP systems with C, CRX/GRX graphics unless otherwise noted.						nr=no report		

Printed in U.S.A.
5091-5750E 10/92