Compute Server Performance Results

I.E. Stockdale[1] and John Barton[2]

Report NAS-94-004
November 1994

barton@nas.nasa.gov
Scientific Computing Branch
NAS Systems Division
NASA Ames Research Center
Mail Stop 258-6
Moffett Field, CA 94035-1000

Abstract

Parallel-vector supercomputers have been the workhorses of high performance computing. As expectations of future computing needs have risen faster than projected vector supercomputer performance, much work has been done investigating the feasibility of using Massively Parallel Processor systems as supercomputers. An even more recent development is the availability of high performance workstations which have the potential, when clustered together, to replace parallel-vector systems.

We present a systematic comparison of floating point performance and price-performance for various compute server systems. A suite of highly vectorized programs was run on systems including traditional vector systems such as the Cray C90, and RISC workstations such as the IBM RS/6000 590 and the SGI R8000. The C90 system delivers 460 million floating point operations per second (FLOPS), the highest single processor rate of any vendor. However, if the price-performance ratio (PPR) is considered to be most important, then the IBM and SGI processors are superior to the C90 processors. Even without code tuning, the IBM and SGI PPR's of 260 and 220 FLOPS per dollar exceed the C90 PPR of 160 FLOPS per dollar when running our highly vectorized suite.

---

1. Computer Sciences Corporation, NASA Contract NAS 2-12961, Moffett Field, CA 94035-1000
2. Scientific Computing Branch, NAS Systems Division, Moffett Field, CA 94035-1000

1.0 Introduction

The exploitation of high performance computer systems for advanced aeroscience research and development has been a primary focus of the Numerical Aerodynamic Simulation (NAS) facility since its inception. In a series of competitive procurements, NAS has chosen parallel-vector supercomputers manufactured by Cray Research, Inc. (CRI). This has satisfied the needs of NAS users by providing fast large memory machines. The current NAS supercomputer system is a 16 processor CRI C90 with eight Gigabytes (GB) of main memory. This system sustains a floating point operation (flop) rate of 3-4 billion flops per second on the daily workload of user computer codes (Ref. 1). This success in providing high throughput in a workload, as opposed to quick single-job turnaround, has been important in fulfilling the goals of the NAS program.

There are concerns that the parallel-vector architecture cannot supply the performance gains which the coming decade's large-scale engineering and scientific problems will require. Computer centers such as NAS are investigating the ability of Massively Parallel Processor (MPP) systems to take over the supercomputing workload. This work is motivated by the hope that superior floating point performance may be obtained by harnessing the power of a large number of relatively slow, inexpensive processors to solve single problems. Until the most recent generation of MPP's, each processor has been equipped with a relatively small memory. Thus, both the data and the computations are distributed among the processors.

A new paradigm is motivated by the fact that workstations with advertised 64-bit floating point operation rates of hundreds of Million Flops per Second (MFLOPS) are becoming common. This performance is due to faster clock rates and the ability to issue multiple instructions per clock tick. The bandwidth from main memory to cache has also improved, and cache sizes have increased. Especially important is the fact that these workstations are available with 1 GB or more of main memory for less than $200,000. By way of comparison, the CRI Y-MP installed at NAS during calendar year 1991 had 1 GB of main memory.

Thus, it is now possible, given the appropriate system software, to achieve supercomputer performance on aeroscience *workloads* by building clusters with these high performance workstations. This is different from the goal of most MPP work in that performance is obtained by running many single processor jobs simultaneously, as opposed to running a single job over many processors. Each workstation may be treated as a compute server, and is given its own job to process.

Given these developments, we began a systematic comparison of floating point performance of various compute server systems. The com-

pute servers considered in this study were the high performance workstations from the Digital Equipment Corp. (DEC), Hewlett-Packard (HP), International Business Machines Corp. (IBM), and Silicon Graphics, Inc. (SGI), and the vector processor systems sold by Convex and CRI. This list is a representative sampling of the leading edge vendors for both parallel-vector and workstation technologies.

2.0 Method and Performance Results

The initial study described herein focuses on the performance attainable by a single code running on a single processor. While many other factors must be considered in evaluating computer systems, this provides a useful and well-understood starting point.

The single processor performance is measured by running a suite of computer programs which reflects the usage of the NAS systems. This usage consists primarily of aeroscience applications which perform well on vector architectures. These applications require systems which can run numerically intense calculations with good precision. The suite is very highly vectorized, as is the NAS workload. These performance results should thus provide insight into the general problem of moving workloads from traditional vector architectures to newer systems. This problem is usually considered more difficult than that of moving scalar application programs from mainframes to workstations.

The suite consists of thirty-two highly vectorized floating point intensive applications. These programs comprise benchmarking kernels, pseudo-application codes and applications obtained from NAS users. The numerical techniques represent a range from matrix multiplies and Fast Fourier Transforms in the kernels to Navier-Stokes and Euler solvers in the applications. The applications' outputs were required to agree with reference outputs produced on a CRI Y-MP. Thus, a system had to compute using 64-bit floating point arithmetic.

The performances results presented below are derived from weighted averages of the data for the individual codes. The weighting produces a mix which corresponds to a highly vectorized (>95%) workload on a Y-MP. (The percent vectorization of a workload represents the ratio of the number of vector operations to the total number of operations.) The suite contains codes which range in size from one to over 900 MB.

Below, we present results obtained using both unoptimized and optimized programs. The unoptimized versions were identical to the NAS-supplied versions, except for changes required to use 64-bit integer or floating point arithmetic. DEC, HP, and SGI required such changes, while the other vendors did not. Vendors were allowed to modify the codes to obtain improved performance results. The number and nature of the lines changed for these optimized versions are summarized in Section 3 below. Note that although these programs were highly vectorized, some of them required code changes before the compilers on the parallel-vector systems could take advantage of this fact. The workstation vendors tended to make more performance-enhancing code modifications, although no vendor optimized more than 17 of the 34 codes.

Codes were run both by the vendors and, when possible, at NAS. All code optimizations were performed by the vendors, not by NAS personnel. Section 3 contains further details on both of these issues.

These requirements were tested by examination of the output files and modified source codes returned by the vendor. Several vendors did not provide us with one or both of these, and are so noted in Section 3.

All results were obtained using production compilers. Any combination of documented compiler options was allowed. In the case of CRI, HP, IBM, and SGI, such options were used to invoke production preprocessors for certain codes. Convex and the workstation vendors provided options which either eliminated or reduced code changes required to use 64-bit floating point arithmetic. Workstation compilers also had options which aligned COMMON data along machine word boundaries and/or promoted integers to eight bits.

The average prices of systems used in this work are shown in Table 1, and were determined as follows. The CRI price is one sixteenth of a sixteen processor C90 system. The Convex price is the NASA contract price for a single-processor C3 system. The other prices are weighted averages of the 96MB and 1GB NASA Scientific and Engineering Workstation Procurement (SEWP) prices for single processor systems (*cf.* Table 2). The 128MB SEWP prices were used in those when 96MB prices were not available. The weighting factor was determined from the memory distribution of flops in the suite. Twenty percent of the flops in the suite are found in programs which use 96MB or less of memory. This corresponds to the observed NAS workload (Ref. 2).

TABLE 1. Compute Server Configurations

Configurations tested	Dates on which codes were run	Clock	Single Processor Price (K$)
Convex C3280	Aug. 1993 -April 1994	17 ns	325
CRI Y-MP C90	Mar. 1993 July 1994	4.2 ns	2750
DEC 3000 Model 900	October 1994	275 MHz	106
H-P 9000/755	July - Aug. 1993	99 MHz	110
IBM RS/6000 590	Aug. 1993 - June 1994	71.5 MHz	110
SGI Power Challenge/L R8000	April 1994 September 1994	75 MHz	105

Flop rates and price-performance ratios (PPR) were calculated on a per-code basis. Codes requiring less than 96MB of main memory were reported with a “low memory” price, while those requiring more than

96MB of main memory were reported with a "high memory" price. The averages and standard deviations tabulated below were then calculated. The standard deviations represent the variation of the reported ratios from code to code.

TABLE 2. Prices for low and high memory configurations

Configuration	96MB SEWP Price (K$)	1GB SEWP Price (K$)
DEC 3000 Model 900 (DECchip 21064A)	42	120
H-P 9000/755	39	126[a]
IBM RS/6000 590	46	124
SGI Power Challenge/L R8000	55	116

a. H-P high memory price and performance data is for a 768MB configuration.

Tables 3 to 8 present data in the form, "*y (x)*", where *y* is the optimized result and *x* is the unoptimized result. Figures 1 and 2 summarize the suite flop rates, while Figs. 3 and 4 summarize the PPR results. Note that DEC did not optimize any codes. Convex chose to optimize only three codes. Finally, the HP large memory results are calculated without factoring in the largest memory code. This code required more than 768 MB, which was the maximum available memory.

The best single-processor performance of all machines surveyed was delivered by the C90, with almost fifty percent of the theoretical peak performance. However, the IBM R/S 6000 590 and the SGI R8000 delivered comparable price-performance on unoptimized code. The 590 had a factor of two advantage when optimized performance results, corresponding to 20% of peak performance, were used to form the price-performance ratio. There is more variation in the performance and price-performance ratios for these codes on a workstation than on the two parallel-vector systems tested.

When the "small memory" subset of codes is considered, the price-performance advantage of the workstations is even more pronounced. This is largely due to the fact that the large-memory workstation configurations are significantly more expensive than the small memory configurations.

TABLE 3. Flop rates (MFLOP/sec.): All Codes[a]

Configurations tested	Suite Rate	Lowest Rate	Highest Rate	Standard Deviation
Convex C3280	22 (21)	2.2 (2.2)	110 (110)	21 (21)
CRI Y-MP C90	460 (440)	140 (140)	550 (540)	110 (110)
DEC 3000 Model 900	(15)	(4.3)	(58)	(13)
H-P 9000/755 (*)	11 (7.8)	5.1 (3.3)	52 (52)	12 (12)
IBM RS/6000 590	52 (28)	19 (1.7)	92 (92)	17 (23)
SGI Power Challenge/L R8000	32 (24)	8.5 (4.2)	140 (140)	38 (39)

a. In "*y (x)*", *y* is optimized and *x* is unoptimized.

TABLE 4. Flop rates (MFLOP/sec.): < 96MB codes

Configurations tested	Suite Rate	Lowest Rate	Highest Rate	Standard Deviation
Convex C3280	39 (37)	17 (17)	110 (110)	21 (22)
CRI Y-MP C90	400 (380)	220 (140)	550 (460)	90 (100)
DEC 3000 Model 900	(21)	(11)	(58)	(14)
H-P 9000/755	12 (9.2)	5.1 (5.1)	52 (52)	15 (15)
IBM RS/6000 590	50 (34)	24 (11)	92 (92)	19 (23)
SGI Power Challenge/L R8000	39 (27)	27 (4.2)	140 (140)	36 (38)

TABLE 5. Flop rates (MFLOP/sec.): > 96MB codes

Configurations tested	Suite Rate	Lowest Rate	Highest Rate	Standard Deviation
Convex C3280	20 (19)	2.2 (2.2)	50 (50)	14 (15)
CRI Y-MP C90	470 (450)	140 (140)	550 (540)	140 (110)
DEC 3000 Model 900	(15)	(4.3)	(34)	(7.5)
H-P 9000/755 (*)	11 (7.5)	8.0 (3.3)	31 (31)	5.7 (6.9)
IBM RS/6000 590	52 (27)	19 (1.7)	75 (59)	15 (18)
SGI Power Challenge/L R8000	31 (23)	8.5 (8.5)	140 (140)	39 (40)

TABLE 6. Price-performance ratios (FLOPS/$): All codes[a]

Configurations tested	Suite Ratio	Lowest Ratio	Highest Ratio	Standard Deviation
Convex C3280	66 (65)	6.8 (6.8)	340 (340)	64 (66)
CRI Y-MP C90	170 (160)	51 (50)	200 (200)	42 (40)
DEC 3000 Model 900	(140)	(35)	(1400)	(370)
H-P 9000/755 (*)	100 (71)	64 (26)	1300 (1300)	330 (330)
IBM RS/6000 590	470 (260)	150 (13)	2000 (2000)	510 (550)
SGI Power Challenge/L R8000	300 (220)	74 (61)	2000 (2000)	510 (510)

a. In "*y (x)*", *y* is optimized and *x* is unoptimized.

TABLE 7. Price-performance ratios (FLOPS/$): < 96MB codes

Configurations tested	Suite Ratio	Lowest Ratio	Highest Ratio	Standard Deviation
Convex C3280	120 (110)	52 (52)	340 (340)	65 (68)
CRI Y-MP C90	150 (140)	81 (52)	200 (170)	33 (37)
DEC 3000 Model 900	(510)	(260)	(1400)	(340)
H-P 9000/755	360 (230)	130 (130)	1300 (1300)	380 (390)
IBM RS/6000 590	1100 (750)	530 (240)	2000 (2000)	420 (490)
SGI Power Challenge/L R8000	570 (400)	400 (61)	2000 (2000)	530 (550)

TABLE 8. Price-performance ratios (FLOPS/$): > 96MB codes

Configurations tested	Suite Ratio	Lowest Ratio	Highest Ratio	Standard Deviation
Convex C3280	60 (59)	6.8 (6.8)	160 (160)	45 (43)
CRI Y-MP C90	170 (160)	51 (50)	200 (200)	49 (41)
DEC 3000 Model 900	(120)	(35)	(280)	(63)
H-P 9000/755 (*)	88 (60)	64 (26)	240 (240)	45 (55)
IBM RS/6000 590	420 (220)	150 (13)	610 (480)	120 (150)
SGI Power Challenge/L R8000	270 (200)	74 (74)	1200 (1200)	340 (350)

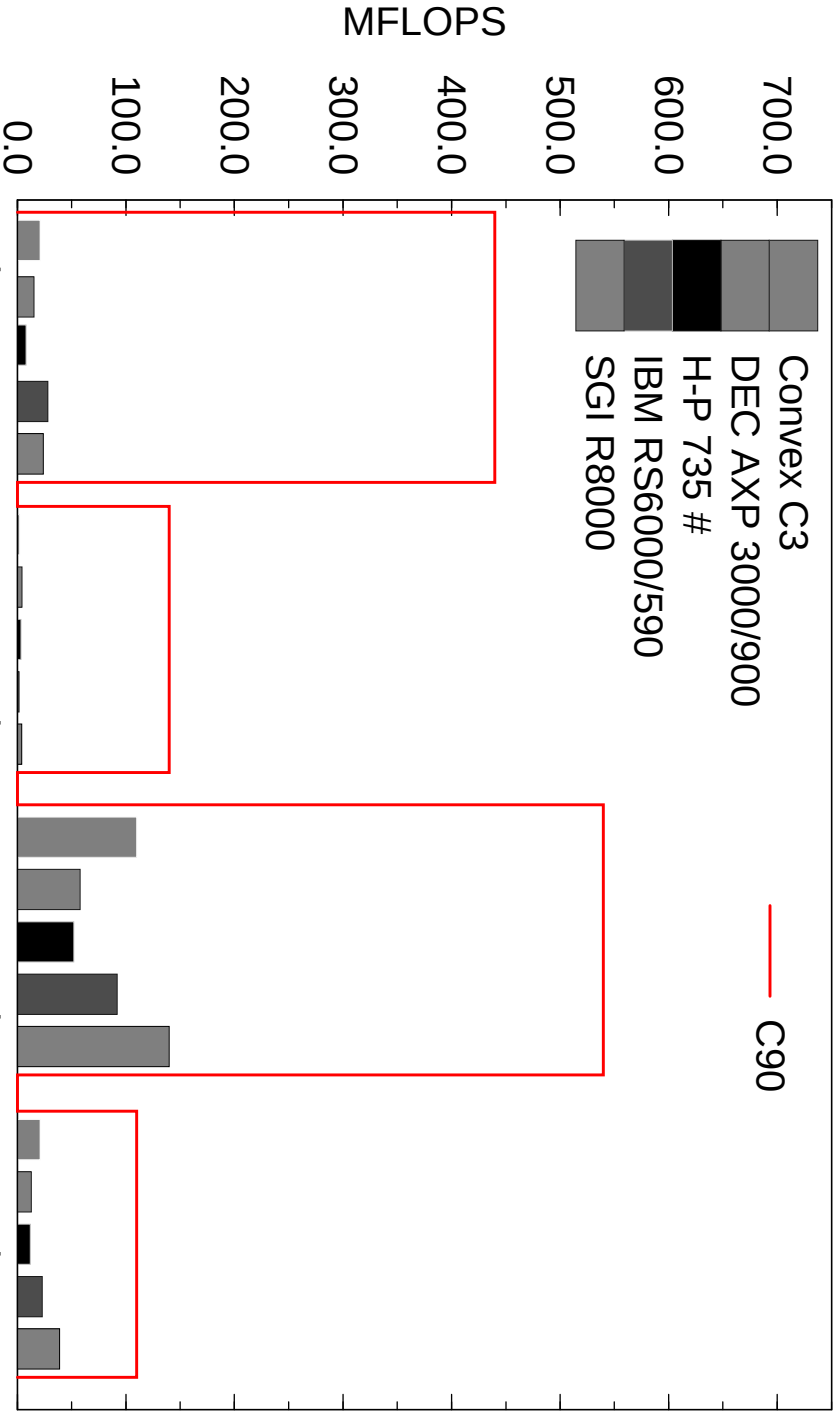


Figure 1: Flop rate for unoptimized codes (# is missing largest memory code)

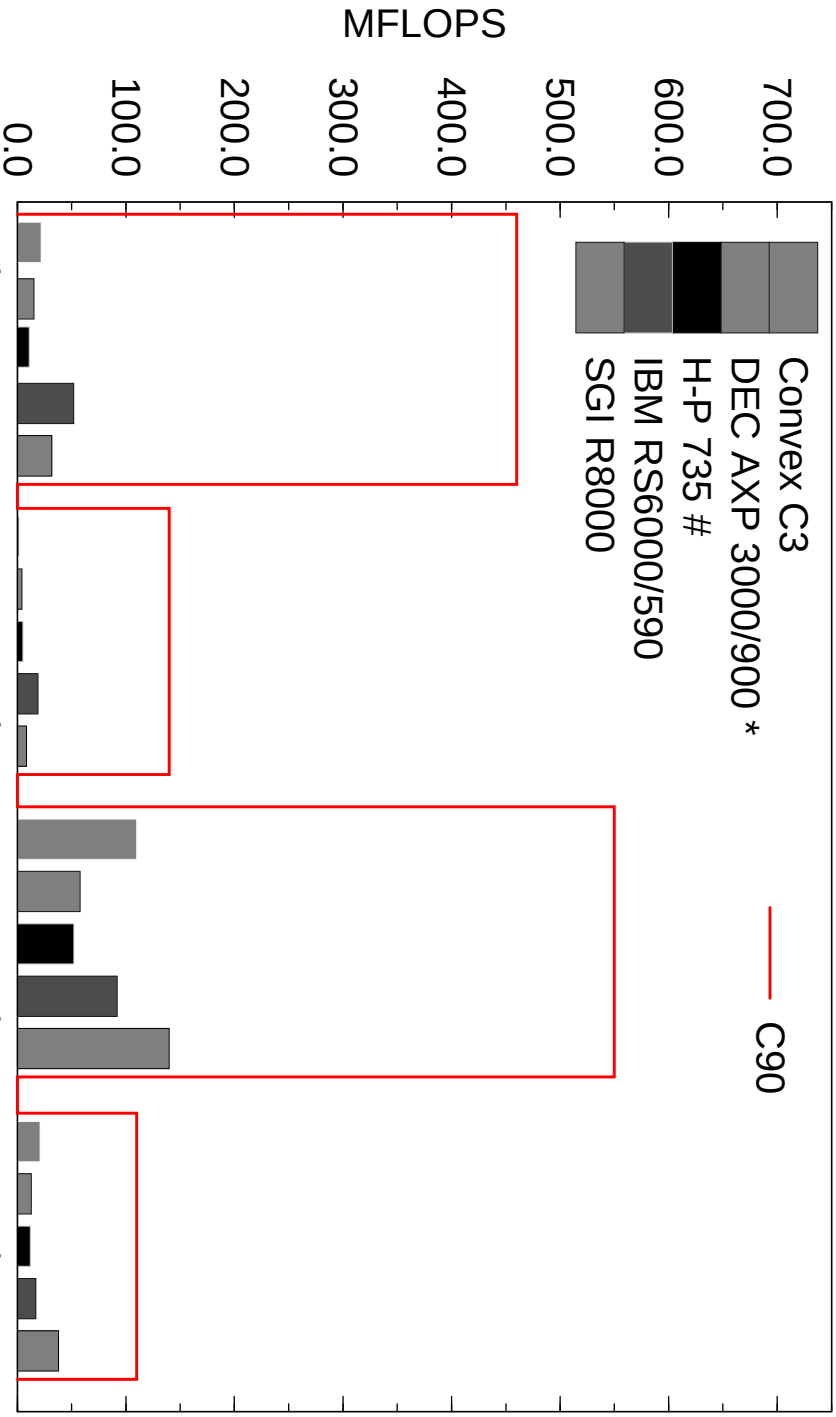


Figure 2: Flop rate for optimized codes (* is unoptimized and # is missing largest memory code)

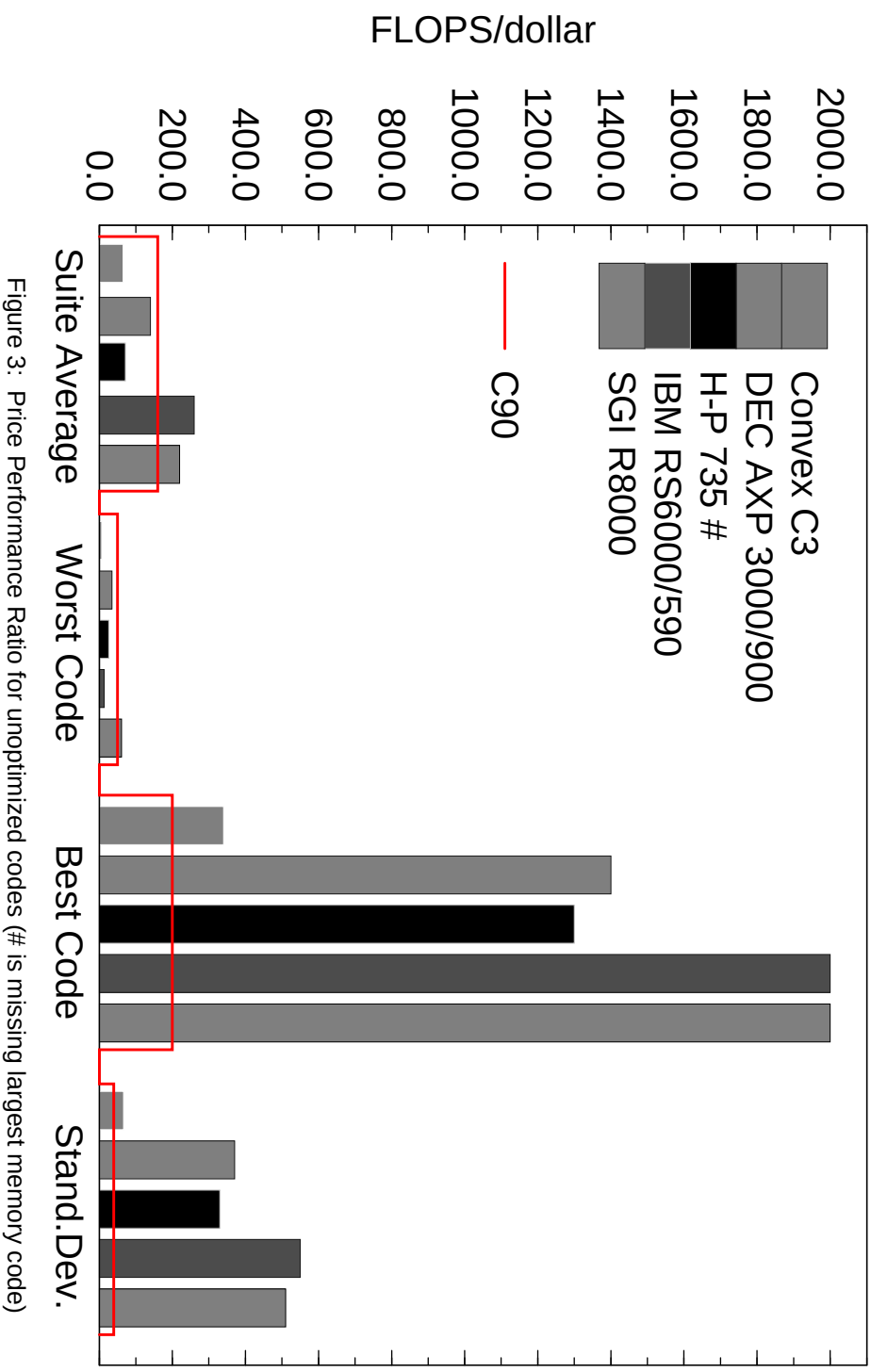


Figure 3: Price Performance Ratio for unoptimized codes (# is missing largest memory code)

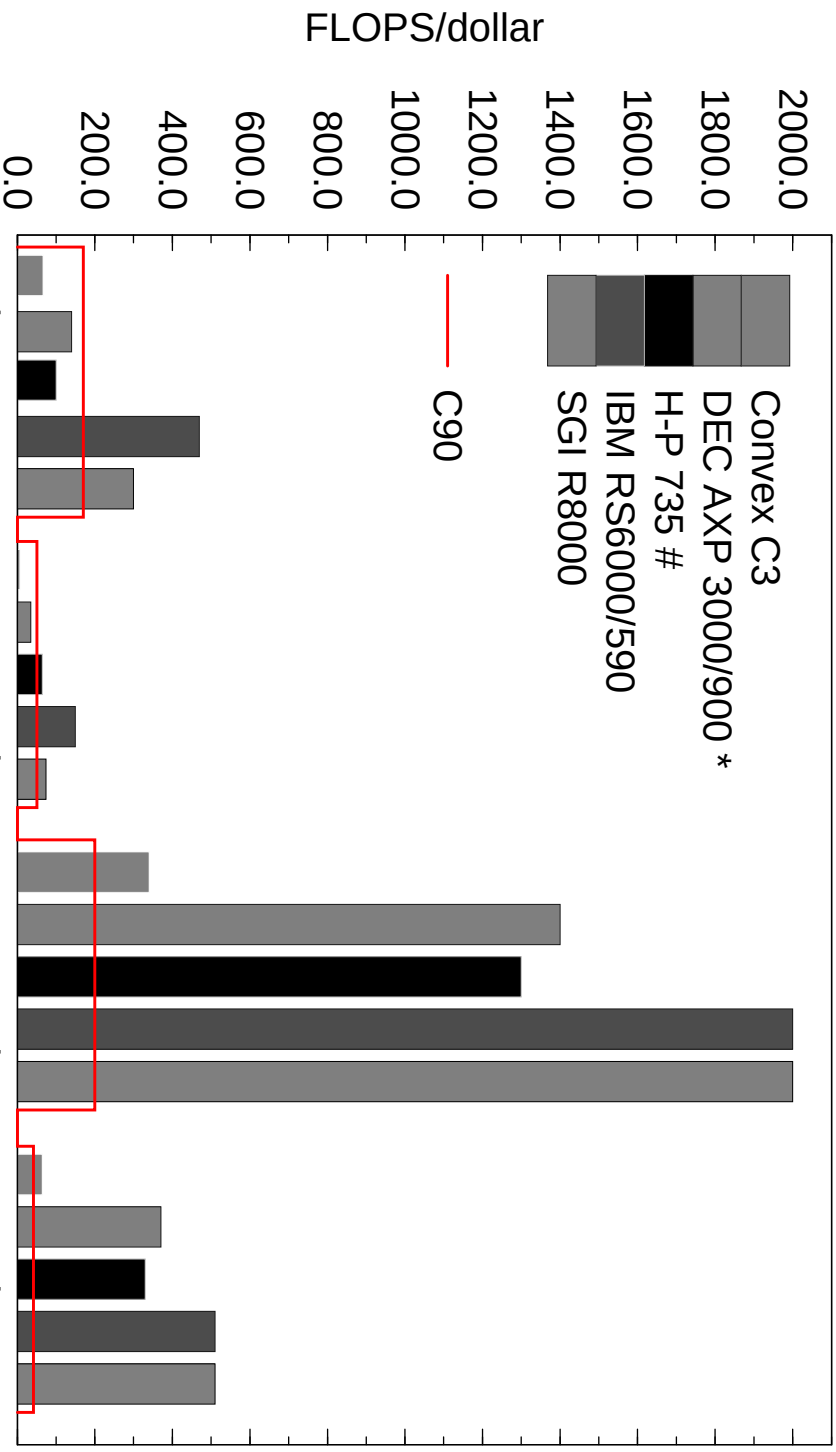


Figure 4: PPR for optimized codes (* is unoptimized and # is missing largest memory code)

3.0 Code Changes

As noted earlier, several vendors modified the benchmark programs to improve performance on their systems. The magnitude of these changes is summarized in Table 9 below. The percent line change was defined to be 100 times the number of changed lines in the source divided by the number of non-empty, non-comment original source lines. Compiler directives count as changed source lines. When blocks of lines are replaced with blocks of alternative lines, the count of changed lines is calculated as the greater of the number of lines removed from the source and the number of lines that replace the removed lines.

TABLE 9. Range of number of lines changed for suite

Configuration	No. of codes optimized	Lines changed (%)
Convex C3	3	0 - 3
CRI Y-MP C90	16	0 - 9
DEC 3000 Model 900	0	0
HP 9000/755	14	not known
IBM RS/6000 590	17	1-50
SGI Power Challenge/L R8000	7	0-50[a]

a. A single code was extensively optimized, see text.

Code changes made by the workstation vendors reduced the number of cache and translation lookaside buffer (TLB) misses. Such optimizations consisted of swapping inner and outer loop indices, reversing array indices, re-arranging complex numbers to improve data locality, and padding arrays in COMMON blocks. A good introduction to such optimizations may be found in Ref. 3.

CRI modified sixteen codes. The hand optimizations included directives to vectorize certain loops where the compiler's dependency analysis failed to allow vectorization. They also reversed loop indices to

allow inner loops to vectorize. Some codes were also tuned to assist the compiler's automatic parallelization utility.

Convex modified three programs to use vectorization directives. No porting code changes were needed. These codes were run both by Convex and by NAS staff.

DEC did not optimize any code in the suite. Nineteen codes were changed for porting reasons. The timings reported here were obtained by vendor personnel.

HP optimized fourteen codes. They commented on the nature of their code changes, but did not supply copies of the modified source code. These programs were all run by HP. Timings for several applications have been verified at NAS.

IBM optimized seventeen codes, with line changes ranging from one to fifty percent. The program with fifty percent lines changed was a Fast Fourier Transform (FFT) code which performed poorly on all cache-based workstations. The next lowest number of lines changed was 23%. Most codes have been run by both IBM and by NAS personnel. In particular, the performance of all but the largest memory codes were verified at NAS.

SGI optimized seven programs. Fourteen codes were changed for porting reasons. Several applications were compiled using an option which directed the compiler to reverse nested loops, eliminating the need for that source code optimization. With one exception, the SGI optimizations were similar to those of other vendors. The exception was an FFT which was extensively rewritten, including a new implementation of the FFT in C. SGI supplied copies of the output files for ten programs. All codes were run by SGI staff.

4.0 Summary

These data show that supercomputer centers such as NAS need to consider clusters of compute servers when acquiring large throughput computer systems. A configuration of 100-200 single-processor workstations may be a practical alternative to a traditional vector-parallel machine, even when the current workload consists of well-vectorized Cray codes.

Of course, several factors other than floating point price-performance ratios must be considered in acquiring a large production system. This report has not addressed such issues as system software and input/output bandwidth. The experience of other sites suggests that the system software issues are manageable (Ref. 4). Additional studies are needed to confirm this.

New studies should assess the combination of system software and hardware by running production workloads on shared memory processors (SMP) and clusters of compute servers. Workloads of message passing codes should be included in these tests. The ability of SMP compilers to produce high-performance programs by automatically parallelizing codes also needs to be investigated. Network and file-system configurations for clusters, particularly in the presence of high performance input/output, also should be evaluated.

Finally, the data provided here is merely a snapshot of a rapidly changing scene. Vendors continue to release improved compilers and processors which promise significant improvements in performance. Compilers are constantly improving, particularly for the RISC processors. Continual effort will be require to keep up with these exciting developments.

5.0 References

[1] Bergeron, Robert J., "The Performance of the NAS HSPs in 4Q93," Report RND-94-005, July 1994, NAS Systems Division, NASA Ames Research Center, Moffett Field, CA 94035.

[2] Dowd, Kevin, **High Performance Computing,** O'Reilly & Associates, Inc., Sebastopol, California, 1993.

[3] Carbon, Duane, private communication, October 1994.

[4] Rinaldo, Frank and Stephen Wolbers, "Loosely Coupled Parallel Computing at Fermilab," **Computers in Physics**, Vol. 7, no. 2 (1993) pp. 184-190.

6.0 Acknowledgments

We would like to thank the vendors for their assistance in providing prices, performance data, and performance tuning.

NAS TECHNICAL REPORT

Title:
Compute Server Performance Results

Author(s):
I.E. Stockdale and John Barton

Reviewers:
"I have carefully and thoroughly reviewed this technical report. I have worked with the author(s) to ensure clarity of presentation and technical accuracy. I take personal responsibility for the quality of this document."

Signed: _____

Name: Bill Nitzberg

Signed: _____

Name: Robert J. Bergeron

Branch Chief:

Approved: _____

Date & TR Number: