

# LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

LA-UR -88-2391

CONF-8011-1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

LA-UR--88-2391

DE88 014338

**TITLE:** PERFORMANCE COMPARISON OF THE CRAY-2  
AND CRAY X-MP/416 SUPERCOMPUTERS

**AUTHOR(S):** Margaret L. Simmons  
Harvey J. Wasserman

**SUBMITTED TO** Supercomputing '88 Conference  
Orlando, Florida, November 14-18, 1988

#### DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

By acceptance of this article, the publisher recognizes that the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.



**Los Alamos** Los Alamos National Laboratory  
Los Alamos, New Mexico 87545

2010 RELEASE UNDER E.O. 14176

# PERFORMANCE COMPARISON OF THE CRAY-2 AND CRAY X-MP/416 SUPERCOMPUTERS

Margaret L. Simmons and Harvey J. Wasserman

Computing and Communications Division  
Group C-3, MS B265  
P. O. Box 1663  
Los Alamos National Laboratory  
Los Alamos, New Mexico 87545  
mls@lanl.arpa (505) 667-1749  
hjw@lanl.arpa (505) 667-2136

## Abstract

The serial and parallel performance of one of the world's fastest general purpose computers, the CRAY-2, is analyzed using the standard Los Alamos Benchmark Set plus codes adopted for parallel processing. For comparison, architectural and performance data are also given for the CRAY X-MP/416. Factors affecting performance, such as memory bandwidth, size and access speed of memory, and software exploitation of hardware, are examined. The parallel processing environments of both machines are evaluated, and speed-up measurements for the parallel codes are given.

**Keywords:** performance, benchmark, supercomputer.

## 1. Introduction

In 1965, the first Cray Research Incorporated CRAY-2 supercomputer was installed at the National Magnetic Fusion Energy Computational Center (NMFECC). Since that time this series of machine has undergone many changes, both in hardware and software. This paper evaluates some of these changes by observing their effect on a series of computationally intensive benchmark codes. We measured the performance of three models of the CRAY-2 that differ in their common memory hardware. The first two models we measured had common memory implemented with dynamic random-access memory (DRAM) with chip access times of 120 and 80 nanoseconds (ns). These machines are Serial 2003, located at the University of Minnesota, and Serial 2011, located at the Air Force Weapons Laboratory in Albuquerque, New Mexico. The third model, Serial 2012, located at Cray Research, uses static random-access memory (SRAM) with a chip access time of 55 ns.

In Section 2, we present a brief outline of the architectural and functional features of the CRAY-2, with emphasis on those features that affect performance. For comparison, corresponding architectural features from another Cray Research product, the X-MP/416, are included. Later sections present benchmark data with single-processor and multiprocessor results discussed separately.

## 2. Comparison of Architectures

The CRAY-2 is a general-purpose parallel/vector supercomputer system. There are four central processing units (CPUs), each with vector and scalar capabilities. Up to 256 million words of dynamic CMOS memory gives the CRAY-2 one of the largest memory capacities of any supercomputer on the market today. For a schematic of the mainframe configuration, see Figure 1. For comparison, the X-MP is also a 4-CPU machine, each CPU having vector and scalar capabilities, but with a common memory of up to 16 million 64-bit words of static bi-polar memory.

### 2.1 CPUs

The CPU clock period on the CRAY-2 is 4.1 ns, while on the X-MP/416, the CPU clock period is 8.5 ns. The effect of this difference is not always as large as it at first seems. Instructions can issue from the instruction buffer on the X-MP every clock period (CP), while on the CRAY-2 the rate is one every other clock period. This gives the CRAY-2 an effective clock period of 8.2 ns with respect to instruction issue, nearly equal to that on the X-MP. After an appropriate start-up, however, arithmetic results are produced every CP on both machines.

The CPUs on both machines contain three sets of registers that serve as source and destination for computations in the functional units. These are address registers, scalar registers, and vector registers, referred to as A-, S-, and V-registers. In addition, the CRAY-2 has 16K words of local (or fast) memory that can be used by these registers as temporary storage. The access time between local memory and A and S registers is 5 and 4 CPs, respectively. The access time for V-registers is 8 CPs + length of the vector. Instead of local memory, the X-MP has an extra set of 72 temporary storage registers called B- and T-registers. Access times for these registers is 1 CP. The V-registers on the X-MP have no corresponding temporary registers. In addition to these registers, the CRAY-2 has eight semaphore flags to enable synchronization of common memory during multitasking. Only one of these semaphores can be assigned to a "job." In contrast, the X-MP has five sets of shared registers (shared among the four CPUs) including 32 semaphores. Arithmetic on both machines is done in fully se-

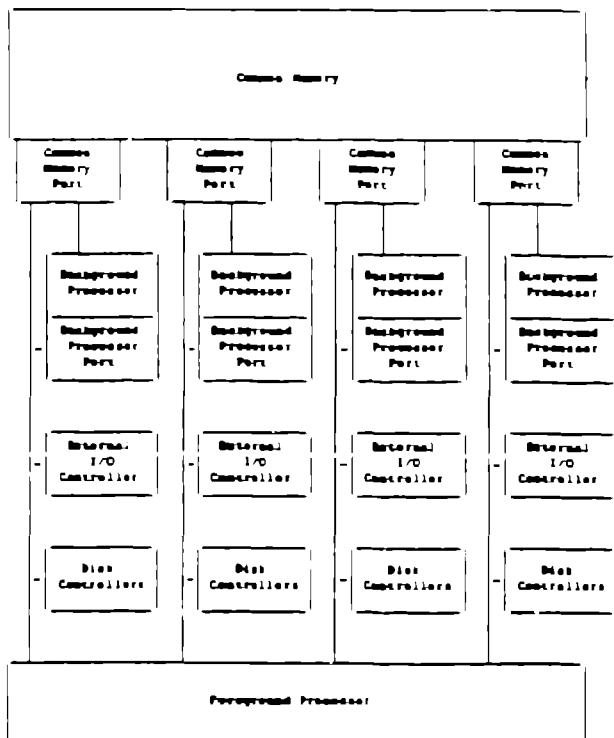


Figure 1. CRAY-2 mainframe configuration.

mented (pipelined) functional units. This pipelining allows the functional units, some of which can also operate in parallel, to deliver a result every clock period, after a suitable start-up time. Chaining, which allows the output of one arithmetic operation to serve as the intermediate input to a subsequent operation, is not available on the CRAY-2. There are also different numbers of functional units on the two machines: the X-MP has 14 while the CRAY-2 has 9, including the reciprocal square root unit. Table I gives some representative times for arithmetic operations. For a more complete explanation of the CPU organization, see (1).

## 2.1 Memory

As mentioned previously, the CRAY-2 has up to 256 million 64-bit words of common, or shared, memory, interleaved up to 128 ways. The memory is organized into quadrants with 32 banks in each quadrant. Each quadrant has a data path to four common memory ports, one for each processor. The four quadrants are accessed by the four processors in "phase time." This means that each processor can access one particular quadrant every fourth clock period. The quadrants are accessed in a round robin fashion, that is, processor 1 can access quadrant 1

Table 1. Scalar Floating Point Operation Types

Operation	CRA Y-2	X-MP/416
Add	76 ns (19 CP)	51 ns (6 CP)
Multiply	76 ns (19 CP)	99.5 ns (7 CP)
Divide	88 ns (22 CP)	119 ns (14 CP)

at cycle 1, quadrant 2 at cycle 2, and so on, back to quadrant 1 at cycle 5. This arrangement has important implications for strided memory references, as discussed below. With one port per processor, the CRAY-2 can do one load or one store at a time. The X-MP, with its four ports per processor, can do two vector loads, one vector store, and one IO memory reference simultaneously.

While the large size of the memory on the CRAY-2 is an asset, the memory cycle time on the early CRAY-2 models of 234 ns (57 CPs) was slow enough to be a detriment. Cray's first solution to the memory speed problem was pseudobanking, a technique to allow access to a physical memory in less time than the memory chip cycle time. Cycle time is made up of two parts, access time and off-chip time. The logic chips are busy for a time equal to the access time, while the memory chips are busy for an additional time equal to the off-chip time. Pseudobanking uses the simple trick of addressing alternate planes of chips within the module. This can be done in a time equal to the access time, effectively reducing the cycle time by nearly half. Using this approach the effective cycle time on the 256-Mword DRAM decreased from 57 CPs to 33 CPs [2]. Pseudobanking is only needed on CRAY-2s with DRAM. Later solutions have involved the use of faster memory chips and static rather than dynamic memories. The X-MP uses chips with a scalar memory access time of 14 CPs (119 ns).

Several factors affect the rate of data transfer between common memory and the vector registers on the CRAY-2.

The first is the rate of instruction issue for vector reads and writes. With only one common memory port per processor, each read or write instruction must wait for the port to be free before it can issue. If one word transfers per clock period, then the next instruction can issue  $VL + 8$  CPs later [2], where  $VL$  is the requested vector length. The minimum transfer time per word,  $T(\min)$ , is approximated by

$$T_{\text{min}} = 0.7L + 0.1V L \quad CPs \quad (1)$$

For a vector length of 64, this time is  $72/64$  or 1.125 CP/word, giving a maximum transfer rate of 217 Mword/s (assuming a 4.1 ns clock). This rate is highly optimistic and assumes no quadrant or bank conflicts.

Memory conflicts also reduce the data transfer rate. Memory conflicts on the CRAY-2s occur at two levels: quadrant conflicts and bank conflicts. Quadrant conflicts are caused by the difference in the rate at which memory-request addresses arrive at the port and the rate at which the port can process these requests. Recall that each memory quadrant on the CRAY-2 can be addressed by each processor only once every 4 CPs. The time between memory requests to the same quadrant is called the quadrant period. Quadrant periods of four cause no conflicts to occur, while periods of two or one do cause conflicts. Vectors with odd strides, including strides of one, have a quadrant period of four, and thus cause no conflicts. Even strides, however, cause conflicts of varying severity. The worst case is a stride divisible by four, here the quadrant period is one. Even in the absence of other conflicts, memory quadrant conflicts can cause performance degradation.

Bank conflicts, like quadrant conflicts, are caused by attempts to access data in the same bank within too small a time period. The bank conflict effect is a function of bank cycle time and number of banks. See Table II for a list of cycle times for the machines we tested. When a bank conflict occurs, the address in the quadrant buffer requires more than the 4-CP quadrant access time to clear. Memory backup then occurs because these quadrant buffers remain full until the requested bank is free.

### 3. Description of Benchmark Programs

The Computing and Communications Division at Los Alamos National Laboratory maintains a set of portable benchmark programs representing characteristic tasks that a large supercomputer would be required to run at the Laboratory. This benchmark set has been run on a wide range of both scalar and vector machines [3-9]. A database is maintained containing results of past runs of these programs on a variety of computers. A report from the National Research Council has characterized supercomputer benchmarks in terms of a hierarchy [10]. Using their characterization, the Los Alamos benchmark set consists of tests at the levels of hardware demonstration programs, basic routines, and stripped down applications. A description of the codes can be found in Ref. [11]. The programs described there are coded in ANSI Fortran for portability and typically can be run on a new machine with little or no change. Execution rates will be indicative of the potential initial usefulness of a new machine.

### 4. Single-Processor Results

The benchmark of CRAY-2 Serial 2003 took place in July 1987, while Serial 2011 was measured in October 1987, and Serial 2012 was measured in January 1988. The X-MP results were obtained in November 1987. Two of the CRAY-2s, Serial 2003 and 2012, ran UNICOS, a UNDX-like operating system. Serial 2011 and the X-MP/416 ran the Cray Timesharing System (CTSS). All measurements were made during dedicated time on a single processor.

#### 4.1 Comparison of Three Types of CRAY-2 Hardware

Table III shows the effect of the faster memory hardware on our benchmark codes. The results for the 80-ns DRAM CRAY-2 are not always consistent with results from the other two machines. That is, the times for some benchmarks increase in going from the 120-ns CRAY-2 to the 80-ns CRAY-2. We believe this is due to different implementations of the CFT77 compiler, and, in particular, the implementation

Table III. Comparison of Benchmark Execution Times (in seconds) for CRAY-2 (One Processor) Showing Effect of Hardware\*

Code	120-ns DRAM (SN 2003)	80-ns DRAM (SN 2011)	55-ns SRAM (SN 2012)
FFT	10.7	10.5	9.6
GAMTEB	5.3	7.5	4.7
SCALGAM	104.6	100.9	92.2
LSS	9.5	9.5	8.9
MATRIX	61.9	59.4	57.3
INTMC	20.8	20.6	18.6
HYDRO	79.7	79.6	68.8
WAVE	186.6	203.0	174.6
ESN	NR	21.5	22.9
MCNP	NR	94.1	77.8

\*CFT77 version 1.3 compiler.

of CFT77 under CTSS on Serial 2011. For this reason, and also because we wish to illustrate the maximum performance gain that could be realized from faster memory, in this discussion we focus on the difference in performance between the 120-ns DRAM CRAY-2 and the 55-ns SRAM CRAY-2S. Speedups because of the static memory are in the range 7-16%. The two scalar codes SCALGAM and GAMTEB show identical speedups of 13%. HYDRO, which is nearly 100% vectorizable, shows the largest speedup. Note that a twofold change in memory chip access time should not yield anything close to a twofold speedup in the codes. The more pertinent hardware feature is the memory latency, which is the time to do loads from common memory. On the DRAM machine, the scalar access latency is 59 CPs, while on the SRAM machine, the latency for scalar loads is 43 CP. Thus, the maximum speedup we could observe here is about 37%. That the maximum observed speedup is still smaller than this may suggest that the compiler could hide some of the memory latency, perhaps by more use of the local memory.

#### 4.2 Comparison of Compilers

Table IV shows a comparison of execution times on the fast memory CRAY-2S (Serial 2012) using the two CFT77 compilers (1.3) and (2.0). Version 2.0 yields a dramatic improvement on some of the codes. The FFT code speeds up by a factor of 2.3 relative to CFT77 1.3. All previous versions of CFT77 vectorized several loops in FFT conditionally; the repeated execution of the conditional code at run time caused much slower execution rates. In FFT the conditional code is generated because a loop bound is passed as an argument to a subroutine. These loops are now fully vectorized in version 2.0.

Using CFT77 2.0, HYDRO speeds up by 35%. HYDRO contains one minor loop that was conditionally vectorized with CFT77 1.3 and now fully vectorized with CFT77 2.0. HYDRO also has three loops, in the time-consuming subroutines VSETUV and VQTERM, that had not vectorized at all in version 1.3 and now vectorize in version 2.0. Each of these three loops did not vectorize with version 1.3 because of function references in conditional blocks. However, another loop in HYDRO that

Table II. Memory Characteristics for Three Tested Models of the CRAY-2

Model	Chip Access Time (ns)	Memory Access Time (CP)	Memory Type	Memory Size (Mword)
2003	120	37	Dynamic	256
2011	80	48	Dynamic	256
2012	55	41	Static	128

**Table IV. Comparison of Benchmark Execution Times (in seconds) for Serial 2012 CRAY-2S (55 ns Memory, One Processor) Showing Effect of Compiler**

Code	CFT77 1.3	CFT77 2.0	Speed-Up
FFT	9.6	4.1	2.34
GAMTEB	4.7	4.4	1.04
SCALGAM	92.2	92.8	0
LSS	8.9	8.9	0
MATRIX	57.3	57.0	0
INTMC	18.6	18.2	1.02
HYDRO	68.8	51.1	1.35
WAVE	174.6	109.7	1.6
ESN	21.5	20.4	1.05
MCNP	77.8	78.1	0

\*Four thousand source particles.

consumes a considerable portion of the execution time does not vectorize with any Cray Research compiler, although we know of other compilers that are successful in vectorizing this loop.

The WAVE code also derives impressive gain from use of CFT77 2.0 on the CRAY-2S. The 60% decrease in execution time results from full vectorization of over 50 loops that had been conditionally vectorized in version 1.3. However, 16 loops in WAVE remain conditionally vectorized.

So far we have focused on CRAY-2 results obtained with the CFT77 compiler. A version of the Cray Research CFT compiler, called CFT2, is also available on the CRAY-2. A comparison of benchmark execution times for code produced using both compilers is given in Table V. The CFT77 2.0 compiler produces much better code in all cases.

#### 4.3 Comparison of CRAY-2S with CRAY X-MP/416

In this section we examine the performance of only the CRAY-2S with that of the CRAY X-MP/416 results. We used a pre-release of CFT77 2.0 (BF185) on a CRAY X-MP/416 running the CTSS operating system at Los Alamos National Laboratory (LANL). (The version of CFT77 2.0 we used on the CRAY-2S was also a pre-release, BF184.)

**Table V. Comparison of Benchmark Execution Times (in seconds) for CRAY-2 (One Processor) Using CFT2 and CFT77 Compilers on SN-2012**

Code	CFT2	CFT77 2.0
FFT	4.9	4.1
GAMTEB	10.0	4.4
SCALGAM	144.7	92.8
LSS	12.4	8.9
MATRIX	*1.5	57.0
INTMC	20.6	18.2
HYDRO	107.0	51.1
WAVE	*	109.7
ESN	27.0	20.4
MCNP	*	78.1

\*Did not compile

#### 4.3.1 Primitive Vector Operations

First we examine performance of selected elementary vector operations, listed in Tables VI and VII. These tables contain rates, in MFLOPS, for various elementary vector operations as a function of vector length. All operations were carried out with unit stride except for the second and third operations in both tables. The X-MP/416 tests used bidirectional memory.

For the non-strided, non-scatter/gather operations in Tables VI and VII, the differences between the two machines at vector length 1000 can generally be reconciled with the rate at which each machine is capable of producing results. For example, on the first operation,  $V = V + S$ , we expect comparable rates, and we observe 83 MFLOPS for the CRAY-2S and 97 MFLOPS for the X-MP. As another example, on the fourth operation,  $V = V * V$ , we expect the asymptotic rate on the CRAY-2S to be less than that of the X-MP by about a factor of 1.5; at vector length 1000, the observed ratio is 1.76 (83 MFLOPS for the CRAY-2S and 90 MFLOPS for the X-MP). However, the CRAY-2S compiler has to unroll all these loops (to a depth of four) to achieve this performance. At shorter vector lengths the X-MP is faster than the CRAY-2 by about a factor of 2.

Comparison of the first and second operations in Table VI shows that, as expected, the CRAY-2S suffers no performance degradation with odd strides. (The X-MP should not show any degradation with stride; it does, and this appears to be an anomaly with CFT77 2.0.) However, with stride 8, performance on the CRAY-2S is about one-fourth of the non-strided rate. The minimum time for memory transfer on the CRAY-2S is slightly more than 1 CP/word. However, with stride 8 all words of data reside in the same quadrant. Therefore, the minimum transfer time, delayed by quadrant conflict only, is about 6.5 CP/word. With a stride of 8, there are no bank conflicts on the machine we used.

Scatter/gather operations, the last two rows in Tables VI and VII, are much more efficient on the X-MP than they are on the CRAY-2, over the entire range of vector lengths. The gather operation on the CRAY-2 is subject to a special hardware delay so that references are allowed roughly once every 4 CPs.

#### 4.3.2 Benchmark Codes

A comparison of the current CRAY-2S results with the CRAY X-MP/416 for the rest of the benchmark codes is shown in Table VIII. Two sets of results are given for the X-MP: one from a pre-release of CFT77 2.0 and one from the production compiler, CFT 1.14. The first thing to notice in Table VIII (comparing columns two and three) is that on the X-MP, CFT77 2.0 now produces better code than CFT 1.14 (with no compiler options) for all but one benchmark. The only (minor) exception is MATRIX, for which CFT 1.14 with the BTREG option (shown in parentheses in Table VIII) is slightly faster than CFT77 2.0.

The X-MP has a significant performance advantage over the CRAY-2S on seven of the ten codes. Of the seven, four are highly vectorizable: HYDRO, LSS, MATRIX, and WAVE. In

Table VI. Rates (MFLOPS) on the CRAY-2S for Selected Vector Operations as a Function of Vector Length (Single Processor, CFT77 2.0)

Operation	10	50	100	200	1000
$a(i) = b(i) + s$	10	37	44	47	83
$a(i) = b(i) + s(i=1,n,23)$	9	35	42	46	84
$a(i) = b(i) + s(i=1,n,8)$	8	16	16	16	18
$a(i) = b(i)*c(i)$	8	28	33	36	51
$a(i) = b(i) + s * c(i)$	15	54	65	72	113
$a(i) = b(i)*c(i) + d(i)*e(i)$	19	60	66	76	90
$a(i) = b(j(i)) + s$	7	19	21	22	29
$a(j(i)) = b(i)*c(i)$	8	28	29	34	35

Table VII. Rates (MFLOPS) on the CRAY X-MP/416 for Selected Vector Operations as a Function of Vector Length (Single Processor, CFT77 2.0; Bidirectional Memory)

Operation	10	50	100	200	1000
$a(i) = b(i) + s$	17	75	75	83	97
$a(i) = b(i) + s(i=1,n,23)$	11	39	44	49	59
$a(i) = b(i) + s(i=1,n,8)$	11	39	43	48	58
$a(i) = b(i)*c(i)$	14	68	70	71	90
$a(i) = b(i) + s * c(i)$	27	136	145	143	173
$a(i) = b(i)*c(i) + d(i)*e(i)$	32	114	126	126	143
$a(i) = b(j(i)) + s$	14	54	58	49	57
$a(j(i)) = b(i)*c(i)$	13	44	43	43	48

HYDRO, LSS, and MATRIX, the predominant loop length is about 100. The VECOPS data in Tables VI and VII showed that the X-MP ran loops at vector length 100 nearly twice as fast as the CRAY-2S did. In WAVE, the predominant loop length is 256. WAVE also involves many gathers for which, as shown above, the X-MP is superior.

Interestingly, in contrast with the VECOPS data, the X-MP is only about 5% faster on the FFT code, a highly vectorized code with short vector lengths on which the X-MP should be fastest.

An important aspect of vectorization on the CRAY-2S concerns the way in which arrays are dimensioned. Because of quadrant conflicts that can have a noticeable effect on performance, arrays with even dimensions will suffer performance degradations relative to arrays with odd dimensions. This fact is highlighted in the performance of the codes LSS and MATRIX relative to the X-MP. Both codes spend most of

their time in SAXPY, and both have loop lengths of 100. Yet MATRIX runs nearly 63% faster on the X-MP than it does on the CRAY-2S, whereas LSS runs about 45% faster on the X-MP. In MATRIX, two of three critical arrays have even dimensions, while in LSS, all critical arrays have odd dimensions. Thus, relative to the X-MP, one must be far more careful of program array dimensions on the CRAY-2S.

The relationship between the X-MP and the CRAY-2S on codes not overwhelmingly vector in nature is harder to explain. Of the two Monte Carlo photon transport codes, one, SCAL-GAM, runs about 28% faster on the X-MP, while the other, GAMTEB, runs about 18% faster on the CRAY-2S. ESN, a totally scalar code, runs about 28% faster on the X-MP. But on MCNP, the X-MP is only 7% faster than the CRAY-2. The reason for this is not clear.

Table VIII. Comparison of Benchmark Execution Times (in seconds) for CRAY-2S  
(Serial 2012) and CRAY X-MP/416 (Single Processor)

Code	CRAY-2 (CFT77 2.0)	X-MP/416 (CFT77 2.0)	X-MP/416 (CFT 1.14)
FFT	4.1	3.9	4.3
GAMTEB	4.4	5.2	7.6
SCALGAM	92.8	72.5	88.4
LSS	8.9	6.1	11.6 (6.7)
MATRDX	57.0	34.9	54.7 (33.2)
INTMC	18.2	12.1	40.2
HYDRO	51.1	39.8	48.9
WAVE	109.7	86.0	111.2
ESN	20.4	15.9	18.2
MCNP**	78.1	73.2	76.7

\*Time in parenthesis is with bidirectional memory and OPT = BTREG for the CFT 1.14 compiler.

\*\*Four thousand source particles started.

#### 4.3.3 X-MP/416 External Storage Performance

The larger central memory on the CRAY-2 is an important asset for this machine. However, the X-MP can be equipped with an external Solid-state Storage Device (SSD) that can also offer potential for large codes. An obvious question is: if a problem can be programmed with an "out-of-core" algorithm, how does the X-MP with SSD perform relative to the same problem run "in-memory" on the CRAY-2?

The WAVE code can be so programmed. We ran a job requiring about 20 Mwords of storage on the CRAY-2 (Serial 2011, 80-ns memory). We ran the same code on an X-MP/416 running CTSS and equipped with a 512-Mword SSD using one channel (1250 Mbyte/s). Both machines used the CFT77 version 2.0 compiler. The X-MP/416 version transferred to the SSD in block sizes of 204800 words. The CRAY-2 ran the job in 461 seconds, while the X-MP required 355 seconds CPU time and 360 seconds elapsed (wall-clock) time. Although we did not run this code on the 55-ns CRAY-2S, we can approximate what the performance will be. Using the CFT77 version 2.0 compiler, the standard WAVE benchmark runs about 12% faster on the 55-ns CRAY-2S than it does on the 80-ns CRAY-2, so the best CRAY-2S time for the 19-Mword job would be about 411 seconds. This value is still larger than the X-MP wall clock time. Note that although I/O to the SSD does not require particularly difficult coding (as might I/O to a disk) other than insuring a large block size, the CRAY-2 version requires no extra coding.

#### 5. Multitasking Results

The four processors of the CRAY-2 can simultaneously be brought to bear on a single job through the multitasking environment. We ran our large Monte Carlo transport code, MCNP, in this environment on the Serial 2011 CRAY-2 at the Air Force Weapons Laboratory in February 1988. The compiler was CFT77 2.0, the operating system was CTSS, and the

multitasking library was Multilib. We ran a problem size of 60000 source particles. For comparison, we ran the same problem on an X-MP/416 at Los Alamos using the CFT77 2.0 compiler, the CTSS operating system, and a multitasking library that is a local system. We used a parallelization method called macrotasking developed at Cray Research and adapted for CTSS on the CRAY-2 by the NMFECC. This method operates at the granularity level of the subroutine.\* Multitasking runs on both machines were done during dedicated time. The times are given in Table IX. Note that the X-MP is about 40% faster than the CRAY-2 for one to four processors. The serial times differ by 34%, which is comparable to the differences observed for the other scalar serial codes.

Speedup is defined as

$$S = T_s / T_n \quad (2)$$

where  $T_s$  is the serial execution time and  $T_n$  is the execution time using  $n$  processors. The speedups for MCNP are plotted in Fig. 2. The CRAY-2 shows a speedup of 3.53 for four processors, while on the X-MP, speedup is 3.65. This difference might be attributed to several factors, one of which is the availability of only a single semaphore per job on the CRAY-2. The X-MP has 32 semaphores available to a job. Another factor affecting speedup is the implementation of synchronization primitives. The Los Alamos system has implemented spin-wait locks while the Cray Research/NMFECC implementation is somewhat less efficient. Since Monte Carlo algorithms are considered to be ideal candidates for parallel processing, one might expect a speedup for four processors that is somewhat closer to four. One reason that we do not see this for this set of runs is that the time spent in the serial sections, such as the setting up of the problem, is constant and independent of the

\*Another approach, called macrotasking, requires the granularity of the DO loop and can be much more efficient. However, when MCNP does the common nested DO loops, a modest amount of reprogramming would be required in order to take advantage of macrotasking.

Table IX. Multitasking Execution Times (in Seconds) for MCNP on the CRAY-2 (Serial 2011) and the CRAY X-MP/416

	CRAY-2	CRAY X-MP/416
	TOTAL	TOTAL
1 Processor	1343.3	963.6
2 Processor	697.4	486.3
3 Processor	477.2	330.9
4 Processor	351.5	253.0
Serial	1240.9	923.8

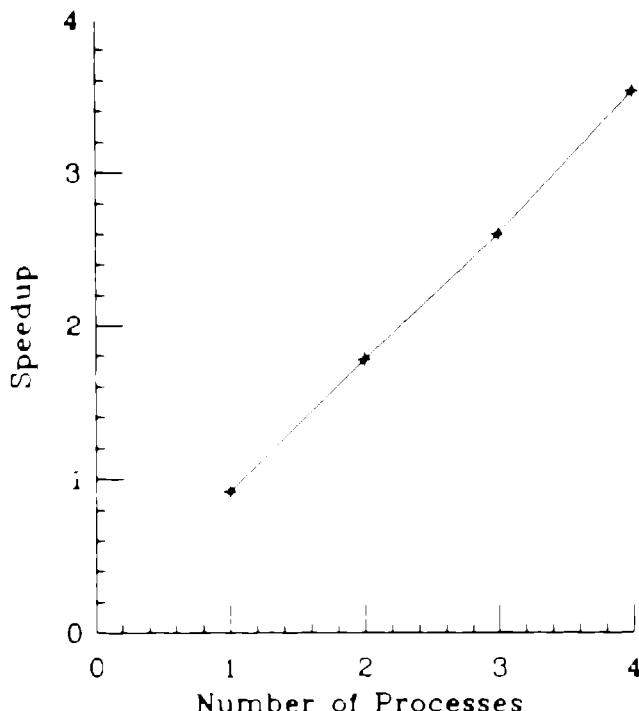


Figure 2. Speedup on a CRAY-2.

number of source particles. This means that as more and more processors are brought to bear on a problem of fixed size, the serial portion takes a larger percentage of the time. This is, of course, Amdahl's law [12].

If we interpret Ware's model [13] (of Amdahl's law) of vector performance as applying to multiprocessor performance, we can also define speedup as

$$S = [(1 - f) + (f/p)]^{-1} \quad (3)$$

where  $f$  is the fraction of the code that can be executed in parallel and  $p$  is the number of processors. For MCNP, which is about 98% parallel, we get a predicted speedup of 3.77 for four processors. This is quite a bit higher than our measured speedup of 3.53 on the CRAY-2S.

There are several reasons for the difference in these two speed-ups. One is the effect of multiprocessor synchronization overhead [14]. Another is the additional time required for system overhead in the multiple processor runs. The serial version of MCNP, for example, is not stack based and so incurs no overhead associated with stack management.

## 6. Conclusions

The faster memory chips on recent models of the CRAY-2 provide some improvement on our benchmarks, but do not, by themselves, allow the CRAY-2S to perform better than the X-MP/416 in single-processor mode. This is because too much of the memory bottleneck on the CRAY-2 is due to factors other than chip access time.

The biggest improvements we have observed during the evolution of the CRAY-2 are derived from compiler changes, not hardware changes. In particular, HYDRO and WAVE, two benchmark codes that closely resemble production codes at the Laboratory, benefit significantly from the combination of new hardware and a new version of CFT77 on the CRAY-2S.

The X-MP has a clear performance advantage over the CRAY-2S on our codes that are highly vectorized. However, the difference between these machines is less clear on codes that are not overwhelmingly vector. The significant factor here appears to be the longer memory latency on the CRAY-2. Although the CRAY-2 provides more central memory than the X-MP, we have shown that on one code that takes advantage of the X-MP SSD, the faster processor and high I/O rates can overcome the lack of X-MP memory.

In multitasking mode, the CRAY-2 performs about as well as the X-MP on the problem that we ran. While the overall times are not as fast as the X-MP, the speedups are comparable. The overhead observed for the problem we ran could be reduced either by running a larger problem or by the use of more efficient synchronization (multitasking).

## Acknowledgments

This work was performed under the auspices of the United States Department of Energy. We are grateful to Dr. Larry Rapagnani of the Air Force Weapons Laboratory for a generous allocation of resources on the CRAY-2 (Serial 2011). Ann Hayes, Rebecca Koskela, Olaf Lubeck, and James Moore of Los Alamos National Laboratory helped collect some of the benchmark data. We thank Ken Lord of Cray Research for his assistance during all phases of the benchmark process.

## References

1. Frank C. Kumpf and Tung M. Nguyen, "Performance Comparison of the CRAY-2 and CRAY X-MP on a Class of Seismic Data Processing Algorithms," *Parallel Computing* (1986) 559-570.

2. Robert Nutt, "CRAY-2 Memory," private communication, June, 1985.
3. R. G. Brickner, H. J. Wasserman, A. H. Hayes, and J. W. Moore, "Benchmarking the IBM 3090 with Vector Facility," *Los Alamos National Laboratory Unclassified Release LAUR-86-3300* (1986).
4. J. H. Griffin and M. L. Simmons, "Los Alamos National Laboratory Computer Benchmarking 1986," *Los Alamos National Laboratory Unclassified Report LA-10151-MS* (1984).
5. O. Lubeck, J. Moore, and R. Mendez, "A Benchmark Comparison of Three Supercomputers: Fujitsu VP-200, Hitachi S810/20, and CRAY X-MP/2," *IEEE Computer* 18 (1985), 10-29.
6. O. Lubeck, J. Moore, and R. Mendez, "The Performance of the NEC SX/2 and CRAY X-MP Supercomputers," *Los Alamos National Laboratory Unclassified Release LAUR-87-227* (1987).
7. M. L. Simmons and O. Lubeck, "Benchmark of the Convex C-1 Mini Supercomputer," *Los Alamos National Laboratory Unclassified Release LAUR-86-2890* (1986).
8. M. L. Simmons and H. J. Wasserman, "Los Alamos National Laboratory Computer Benchmarking 1986," *Los Alamos National Laboratory Report LA-10898-MS* (1987).
9. H. J. Wasserman, M. L. Simmons, and A. H. Hayes, "A Benchmark of the SCS-40 Computer: A Mini Supercomputer Compatible with the CRAY X-MP/24," *Los Alamos National Laboratory Unclassified Release LAUR-87-659* (1987).
10. "An Agenda for Improved Evaluation of Supercomputer Performance," National Research Council (National Academy Press, Washington D.C., 1986).
11. Harvey J. Wasserman, "Los Alamos National Laboratory Computer Benchmarking 1988," Los Alamos National Laboratory technical report to be published.
12. G. Amdahl, "The Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities," AFIPS Conf. Proc. 30, 1967.
13. W. Ware, "The Ultimate Computer," *IEEE Spectrum*, March 1972.
14. B. L. Buzbee, "The Efficiency of Parallel Processing," *Computer Design*, June 1984.

REPRODUCED FROM  
BEST AVAILABLE COPY