Architecture and Application: The Performance of the NEC SX-4 on the NCAR Benchmark Suite

Steven W. Hammond, Ph.D. and Richard D. Loft, Ph.D.
Scientific Computing Division
National Center for Atmospheric Research*
Boulder, Colorado 80307

and

Philip D. Tannenbaum
HNSX Supercomputers, Inc.
The Woodlands, Texas 77381

August 1, 1996

* NCAR is sponsored by the National Science Foundation. Opinions and findings expressed in this paper are those of the author(s) and not necessarily those of the National Science Foundation.

Abstract

In November 1994, the NEC Corporation announced the SX-4 supercomputer. It is the third in the SX series of supercomputers and is upward compatible from the SX-3R vector processor with enhancements for scalar processing, short vector processing, and parallel processing. In this paper we describe the architecture of the SX-4 which has an 8.0 ns clock cycle and a peak performance of 2 Gflops per processor. We also describe the composition of the NCAR Benchmark Suite, designed to evaluate the computers for use on climate modeling applications. Additionally, we contrast this benchmark suite with other benchmarks. Finally, we detail the scalability and performance of the SX-4/32 relative to the NCAR Benchmark Suite.

1 Introduction

As stated in the UCAR Request For Proposal, RFP B-10-95P, understanding and predicting climate, particularly climate variation and possible human-induced climate change, presents one of the most difficult and urgent challenges in science. This is because changes in climate, whether anthropogenic or due to natural variability, involve a complex interplay of physical, chemical, and biological processes of the atmosphere, oceans, and land surface. As climate system research seeks to explain the behavior of climate over longer and longer time scales, the focus necessarily turns to behavior introduced by these processes and their interactions among even more detailed models of climate subsystems. Clearly, use of computer models to study these interactions is a critical element in furthering our understanding of earth’s climate systems.

However, a major, limiting factor to advancing the state-of-the-art in climate change research has been the lack of dedicated computing cycles and high performance machines to run closely coupled atmospheric and ocean models simultaneously. Only by substantially augmenting the speed with which these models execute on a given computing system, can we hope to enhance our understanding of the components that make up our complex global climate system.

To meet the computing needs of climate prediction the National Center for Atmospheric Research (NCAR) conducted a competitive procurement which sought to acquire one or more supercomputers. NCAR announced on May 20, 1996 that it intended to procure such equipment which included four 32-processor SX-4 systems from NEC. In the sections that follow we discuss the architecture of this system, the composition of the NCAR Benchmark Suite, and the performance of the SX-4 relative to these benchmarks.

2 SX-4 Architecture

A building block of the SX-4 is the "node". A single node of the SX-4 consists of up to 32 processors with a single shared memory and uniform memory access time. A *p* processor SX-4 is denoted SX-4/*p*. Each processor has a peak performance of 2 GFLOPS or 64 GFLOPS peak performance per node. A block diagram of a single node SX-4 is shown in Figure 1.

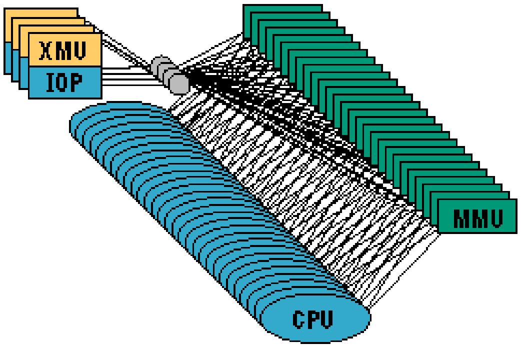

Figure 1: SX-4 Single node architectural diagram.

Up to 16 of these shared memory processing nodes can be combined using a crossbar (IXS). The IXS has a bisection bandwith of 128 GByte/s and supports global hardware addressing. A full SX-4 configuration then consists of 512 processors with a total memory bandwith of more than 8 TByte/s. In particular, an SX-4/512 has 8 Tbyte/s of bandwidth between node memories and arithmetic pipelines, and 128 GByte/s bisection bandwith to other node memories, and 192 GByte/s from node memories to I/O. Figure 2 shows a two node system connected with the IXS.

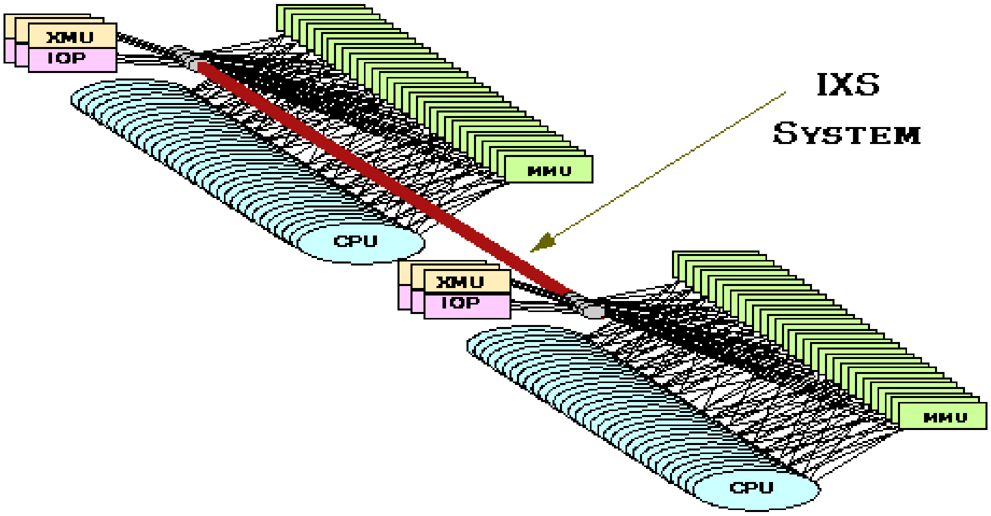

Figure 2: SX-4 multiple node architectural diagram showing two nodes connected with the IXS.

The SX-4 is a 64 bit machine; all performance specifications assume 64 bit data. The vector and scalar units support 32 and 64 bit operands; the scalar unit also supports 8, 16, and 128 bit operands. Each processor has hardware implementations to support three floating point data formats -- IEEE 754, Cray, and IBM. IEEE 754 support includes basic 32 and 64 bit, and extended precision 128 bit word sizes. Cray compatibility mode conforms to Cray 64 and 128 bit data formats. IBM compatibility mode conforms to IBM 32, 64, and 128 bit data formats. Hardware support is also provided for 32 and true 64 bit integers. As an example, two 32-bit integers can be accurately multiplied by the processor. Hardware performance is identical with all 64-bit formats. Floating point format selection is made on a program by program basis at compile time.

Finally, the SX-4 utilizes 0.35 micron CMOS chip technology to provide an air-cooled system which uses a fraction of the power required by conventional ECL implementations. For example, an SX-4/32 has a power requirement of 123 KVA compared to a 16 processor CRI C90 which requires over 400 KVA.

An SX-4 system consists of the following major components.

1. Central Processor Unit (CPU)
2. Main Memory Unit (MMU)
3. Extended Memory Unit (XMU)
4. Input Output Processor (IOP)
5. Internode Crossbar for MultiNode Systems (IXS)

The characteristics of each are described below.

2.1 Central Processor Unit

Each SX-4 processor contains a vector unit and superscalar unit. The vector unit (see Figure 3) is built using eight vector pipeline processor VLSI chips. Each vector unit chip is a self contained vector unit with registers holding 32 vector elements. The eight chips are connected by crossbar and comprise 32 vector pipelines arranged as sets of eight add/shift, eight multiply, eight divide, and eight logical pipes. Each set of eight pipes serves a single vector instruction, and all sets of pipes can operate concurrently. With a vector add and vector multiply operating concurrently, the pipes provide 2 GFLOPS peak performance. If a vector divide is also operating at the same time the processor can exceed its peak rating.

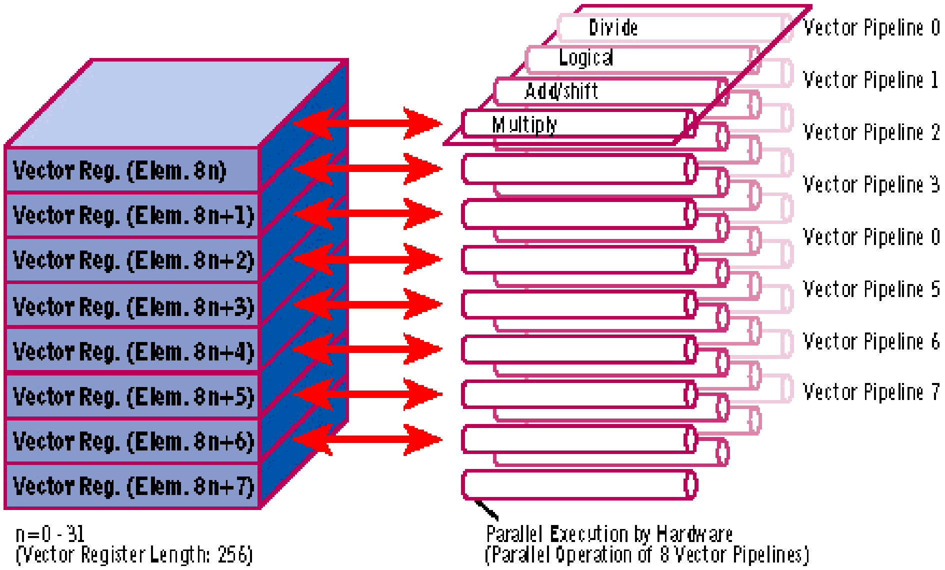

Figure 3: SX-4 vector unit block diagram.

The scalar unit is a superscalar RISC processor with a 64 kilobyte data cache, a 64 kilobyte instruction cache, and eight kilobyte instruction buffer, see Figure 4. All instructions are issued by this superscalar unit which can issue two instructions per clock period. Most scalar instructions issue in a single clock and most vector instructions issue in two clocks. To effectively service the various instruction states, the issue unit can actually initiate 1, 2, 3, or 4 instructions in any given clock. Branch prediction, data prefetching, and out-of-order instruction execution are employed to maximize throughput. Additionally, each processor has access to a set of communications registers optimized for synchronization of parallel processing tasks. Examples of communications register instructions included are test-set, store-and, store-or, and store-add. There is a dedicated set of these for each processor, and each chassis has an additional set for the operating system.

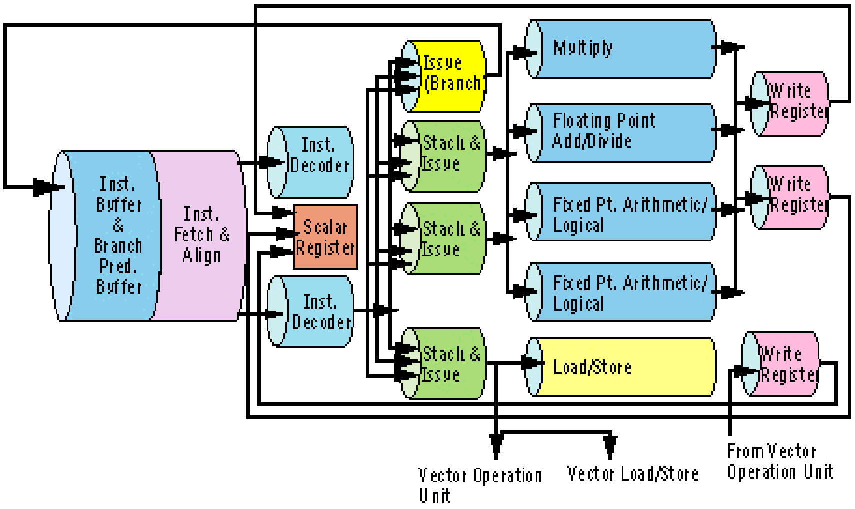

Figure 4: SX-4 scalar unit block diagram.

2.2 Main Memory Unit

The memory and the processors within each SX-4 node are connected by a nonblocking crossbar. Each processor has a 16 Gbytes per second port into the crossbar. The main memory can have up to 1024 banks of 64-bit wide synchronous static RAM (SSRAM). The SSRAM is composed of 4 Mbit, 15 ns components. Bank cycle time is only two clocks. A 32 processor node has a 512 gigabytes per second sustainable memory bandwidth. Conflict free unit stride as well as stride 2 access is guaranteed from all 32 processors simultaneously. Higher strides and list vector access benefit from the very short bank cycle time.

The SX systems are real memory mode machines, but utilize page mapped addressing. The SX architecture does not support demand paging. The page mapped architecture allows load modules to be non-contiguously loaded, eliminating the need for periodic memory compaction procedures by the operating system.

2.3 Extended Memory Unit

SX-4 systems are equipped with high speed semiconductor disks called extended memory units (XMU). The XMU is constructed of 60 nanosecond, 16Mbit DRAM components. A 32-processor node can be configured with up to 32 gigabytes of capacity with 16 Gbytes per second of bandwidth.

Hardware features allow the XMU to be effectively used for direct mapped FORTRAN data arrays. This feature allows processing of large data sets that might not fit into main memory. This feature is supported by compile time options and does not require special programming. XMU can also be used for operating system functions, disk caching, swap, and /tmp space. It is similar in functionality to the CRI SSD.

2.4 Input-Output Processor

Each Input-Output Processor (IOP) has a 1.6 gigabytes per second bandwidth. Up to 4 IOPs can be connected to a SingleNode chassis. The basic high performance channel is HIPPI, with additional support for fast wide SCSI-2. The IOPs operate asynchronously with the processors as independent I/O engines. There is also an Input-Output Multiplexer (IOX) which serves as the boot device and can be used as the operator and maintenance console. It also serves as a channel concentrator to multiplex multiple lower speed channels types into HIPPI.

2.5 Internode Crossbar

The Internode Crossbar (IXS) connects multiple SX-4 nodes through a fibre channel connected crossbar. Connections between nodes have 8 Gbyte per second bandwidths. Each node has an input channel and an output channel which can operate concurrently. The IXS itself is a non-blocking crossbar which can sustain 128 Gbytes per second of bisection bandwidth for a full 16 node system. The IXS is unique as it provides very tight coupling between nodes enabling a single system image for both hardware and software. The IXS also contains global, internode communications registers to enable efficient software synchronization of events occurring across multiple nodes.

2.6 Operating Software

The SX-4 runs the SUPER-UX operating system which is a System V port with additional features from 4.3 BSD plus enhancements to support supercomputing requirements. It has been in production use since 1990. SUPER-UX Release 6.1 supports the SX-4 and provides complete load module compatibility with SX-3 Series systems.

Some of the major operating system enhancements include:

1. automatic, unattended operation facilities
2. checkpoint/restart by user or operator commands
3. enhanced NQS batch subsystem
4. enhanced system configurability and partitioning
5. file archiving management (SXBackStore)
6. high performance file system
7. multilevel security system

Each is briefly described below.

2.6.1 Automatic Operation

SX systems provide hardware and software support to enable operatorless environments. The system can be preprogrammed to power on, boot, enter multi-user mode, and shutdown-poweroff under any number of programmable scenarios. Any operation which can be determined by software and responded to by closing a relay or executing a script can be serviced.

2.6.2 Checkpoint/Restart

NQS batch jobs can be checkpointed by either the owner, operator, or NQS administrator. No special programming is required for checkpointing.

2.6.3 NQS Batch Subsystem

SUPER-UX NQS is enhanced to add substantial user control over work. Recently added commands include qcat which will copy the stdout or stderr file from an executing batch script and present it to the user. NQS queues, queue complexes, and the full range of individual queue parameters and accounting facilities are supported.

2.6.4 Configurability and Partitioning

SUPER-UX has a feature called Resource Blocking which allows the system administrator to define logical scheduling groups which are mapped onto the SX-4 processors. Each Resource Block has a maximum and minimum processor count, memory limits, and scheduling characteristics such that a portion of an SX-4 can be defined primarily for interactive work while another may be designated for static parallel processing scheduling using a FIFO scheme; another area can be configured to optimize a traditional vector batch environment having multiple processors. All processors can be assigned to a single process by properly defining the Resource Blocks.

2.6.5 File Systems

The SUPER-UX native file system is called SFS. It has a flexible file system level caching scheme utilizing XMU space; numerous parameters can be set including write back method, staging unit, and allocation cluster size. Individual files can exceed 2 terabytes in size. The NFS (Network File System) and DFS (Distributed Computing Environment distributed file system) are both supported (DFS from R7.1).

2.6.6 Multilevel Security

The Multilevel Security (MLS) option is provided to support site requirements for either classified projects or restricted access. Security levels are site definable as to both names and relationships. MLS has been in production use since early 1994.

3 Other Benchmark Suites

Before going into the details of the NCAR benchmarks, we first discuss existing benchmark suites and explain why they were inappropriate for our purposes.

3.1 LINPACK

The LINPACK Benchmark [4,5] is a numerically intensive test that has been used for years to measure the floating point performance of computers. LINPACK is a collection of Fortran subroutines which analyze and solve various systems of simultaneous linear algebraic equations. The benchmark consists of solving dense systems of equations for a system of order 100 and 1000. The subroutines are designed to be completely machine independent, fully portable, and to run at near optimum efficiency in most operating environments. LINPACK tends to measure peak performance of a computer and is not intended to evaluate the overall performance of a computer system which was required as part of the NCAR procurement.

3.2 NAS Parallel Benchmarks

The NAS Parallel Benchmarks [1] are designed to characterize the computation and data movement of large scale computational fluid dynamics (CFD) applications. There are five kernel codes and three simulated applications. These benchmarks are unique in that they are specified algorithmically rather than with computer code. Although there is significant commonality between CFD and numerical climate/weather prediction, the differences are such that benchmarks from the NAS suite did not characterize the computational load at NCAR and thus were inappropriate for inclusion in the NCAR Benchmark suite.

3.3 HINT

The HINT benchmark was developed by Gustafson and Snell [8]. HINT stands for Hierarchical INTegration and it uses interval subdivision to find rational bounds on the area in the x-y plane for which x ranges from 0 to 1 and y ranges from 0 to $(1-x)/(1+x)$. This type of benchmark captures characteristics of applications utilizing adaptive refinement methods such as fast N-body solvers. The results of the HINT benchmark are in units called "QUIPS" which are quality improvements per second. The authors assert that MFlops are an inappropriate measurement since they don’t measure how much progress was made on a computation but rather what was done, useful or otherwise.

We have executed the HINT benchmark on single processors of four different systems at NCAR. We have compared the megaQUIPS (MQUIPS) result with the Mflops metric of RADABS, a computational kernel benchmark from the NCAR Benchmark Suite. We ran on a single processor of a SUN Sparc 20, an IBM RS6000/590, a CRI J90, and a CRI Y-MP. The results are shown in Table 1.

Benchmark	**SUN SPARC20**	**IBM RS6K 590**	**CRI J90**	**CRI YMP**
HINT (MQUIPS)	**3.5**	**5.2**	**1.7**	**3.1**
RADABS (MFLOPS)	**12.8**	**16.5**	**60.8**	**178.1**

Table 1: Comparison of the "MQUIPS" metric and the Mflops measurement from the NCAR kernel benchmark "RADABS" for single processor performance of four systems.

The authors claim that the HINT benchmark has highly predictive powers - much better than extant benchmarks. However, we do not find that the MQUIPS metric correlates well with the relative performance for our applications across scalar and vector processors. For example, HINT assigns a value of 3.5 MQUIPS to a SUN Sparc 20 and 3.1 MQUIPS to one processor of a CRI Y-MP. The kernel benchmark RADABS which is a compute-intensive radiation physics routine from a climate model sustains 12.8 Mflops on the Sparc 20 and 178.1 Mflops on the Y-MP. Given that HINT rates both the J90 and the Y-MP below the Sparc 20 and the RS6000, it seems that HINT is better tuned to measuring scalar processor performance than the performance of vector processors.

3.4 STREAM

The STREAM benchmark [11] is a set of four operations that evaluate computer memory bandwidth using four long vector operations. They have unit stride memory access patterns and are designed to eliminate the possibility of data reuse. The COPY benchmark in the STREAM suite is similar to the COPY benchmark in the NCAR suite except that the array size is fixed in the STREAM version. Memory bandwidth for irregular data access patterns is not measured and the array sizes are fixed. In general, there is only a single bandwidth measurement taken instead of testing bandwidth for a range of array sizes.

4 The NCAR Benchmark Suite

The NCAR Benchmark Suite was developed to evaluate computer systems and gain insight into their performance relative to long running, dedicated climate simulations. It consists of thirteen kernels and three complete geophysical simulation codes. The kernels measure specific aspects of system operation such as accuracy of intrinsics, memory to memory bandwidth, processor speed, memory to disk I/O rates, and HIPPI transfer rates. The three applications measure combinations of these and are to be run at multiple resolutions to measure how performance varies as a function of problem size. Included in the application codes is the NCAR Community Climate Model version 2 (CCM2). The applications are also run on different numbers of processors (where appropriate) to evaluate scalability of the machine for a fixed problem size. Together these codes give a comprehensive measure of the capabilities of a computer system with respect to NCAR’s computing environment as well as a computer system’s performance under NCAR’s current and anticipated computational load.

The construction of the NCAR Benchmark Suite is consistent with the recommendations of Dongarra et al [3] who suggest that an effective benchmark suite must accurately characterize the anticipated workload of the system. Additionally, initial tests should start with simple programs and then increase the complexity of the programs that approximate ever more closely the jobs that are part of the work day. The 13 benchmarks can be grouped into seven categories. We list them here and give a complete description for each as well as the performance of the SX-4 on them below:

1. Correctness of basic floating point arithmetic as well as accuracy and performance of intrinsics.
   - PARANOIA: arithmetic operation test
   - ELEFUNT: elementary function test

2. Memory bandwidth tests.
   - COPY: memory to memory
   - IA: indirect addressing speed
   - XPOSE: array transpose

3. Coding style comparison - scalar versus vector processor.
   - RFFT: "scalar" FFT
   - VFFT: "vectorized" FFT

4. Raw performance.
   - RADABS: processor performance

5. I/O to disk system and network.
   - I/O: memory to disk

- HIPPI: HIPPI throughput
- NETWORK: external network evaluation

6. Production mix.
   - PRODLOAD: simulated production job load

7. Complete applications.
   - CCM2: global climate model
   - MOM: F77 ocean model
   - POP: F90 ocean model

For the COPY, IA, XPOSE, RFFT, VFFT, and RADABS benchmark, there is a parameter in the code that the user can set called "KTRIES". This determines the number of times that a particular experiment within the benchmark is conducted. For values of KTRIES greater than one, the best performance for that instance is reported. We have found that the performance curves produced are relatively smooth when KTRIES is set to 5 or greater and yet it still accurately portrays the system capability. In the results reported below, KTRIES was set to 5 for VFFT and 20 for the other benchmarks. The VFFT value was simply a matter of expedience in completing the benchmarks.

In Table 2 we list the characteristics of the machine benchmarked in February, 1996. In particular, one should note that the clock speed of the SX-4/32 benchmarked was 9.2 ns rather than the 8.0 ns available currently.

Clock Rate	**9.2 ns**
Peak FLOP Rate Per Processor	**2 GFLOPS**
Peak Memory Bandwidth	**16 GB/sec/proc**
Disk Capacity	**282 GB**
Main Memory	**8GB**
Extended Memory	**4GB**
Cooling	**air cooled**
Power Consumption	**122.8 KVA**

Table 2: Specifications of the NEC SX-4/32 system used for the benchmark results reported in this paper.

4.1 Floating Point Correctness

We were concerned about the correctness and accuracy of the mathematical operations performed by a computer system. This is especially critical when evaluating experimental or newly developed systems with optimized intrinsic libraries. Verifying correctness of basic arithmetic and accuracy of intrinsic functions in isolation is easier than tracking down the same problem if it occurs in a large application. Therefore, we utilized two benchmarks to perform these tests. The first is PARANOIA, a freeware program developed by Professor Kahan of the University of California at Berkeley. It checks the correctness of the vendor’s implementation of basic floating point arithmetic. The second benchmark is ELEFUNT, based on the code developed by W. J. Cody at the Argonne National Laboratory. The code developed by Cody measured the accuracy of intrinsic functions and we added the performance measurement for the intrinsic functions `EXP`, `LOG`, `PWR`, `SIN`, and `SQRT`.

The correctness and accuracy checks for both PARANOIA and ELEFUNT are essentially pass/fail tests and the SX-4 passed these tests. ELEFUNT also measures the performance of intrinsic functions and reports millions of function calls per second. The performance of the SX-4/1 on ELEFUNT for 64-bit arithmetic is shown in Table 3.

Function	**alog**	**exp**	**pwr**	**sin**	**sqrt**
Performance	34.6	40.7	10.4	39.5	46.7

Table 3: Single processor performance in 64-bit for five different intrinsic functions measured in millions of function calls per second.

4.2 Memory Bandwidth Tests

We have observed that many NCAR modeling codes are memory bandwidth limited. Three benchmarks were developed to measure the memory bandwidth characteristics of the processor:

4.2.1 Matrix Copy (COPY)

The COPY benchmark measures system memory bandwidth for a highly regular (unit stride) memory to memory copy operation:

```
do j=1,M
   do i=1,N
      b(i,j)=a(i,j)
   end do
end do
```

The copy axis length *N* is in the range of 1 to 10^6 and the instance axis length *M* is in the range of 10^6 to 1.

4.2.2 Indirect Address (IA)

The IA benchmark measures system memory bandwidth for an irregular (gather) memory access operation:

```
      do j=1,M
         do i=1,N
            b(i,j)=a(indx(i),j)
         end do
      end do
```

The gather axis length *N* is in the range of 1 to 10^6 and the instance axis length *M* is in the range of 10^6 to 1.

4.2.3 Matrix Transposition (XPOSE)

The XPOSE benchmark measures system memory bandwidth of an array transposition (scatter) operation for an *N* by *N* matrix:

```
      do k=1,M
         do j=1,N
            do i=1,N
               b(i,j,k)=a(j,i,k)
            end do
         end do
      end do
```

The matrix size *N* is in the range of 2 to 10^3 and the instance axis length *M* is in the range of 250,000 to 1.

There is a novel feature in these three benchmarks. The values of *M* and *N* in each benchmark are chosen so that the amount of data being moved in memory is roughly constant. At one extreme there are many small arrays being manipulated and at the other extreme a few large arrays are being operated on. This yields a more comprehensive measurement of the memory bandwidth than a single measurement for a fixed *M* and *N*.

Figure 5 shows the memory bandwidth of a single processor of the SX-4 as reported by the three benchmarks. Note that the performance on the COPY benchmark far exceeds the performance on the XPOSE and IA benchmarks. Also, in the bandwidth we report here, we only count the elements of the array *a* being moved to the array *b* and not the index values used.

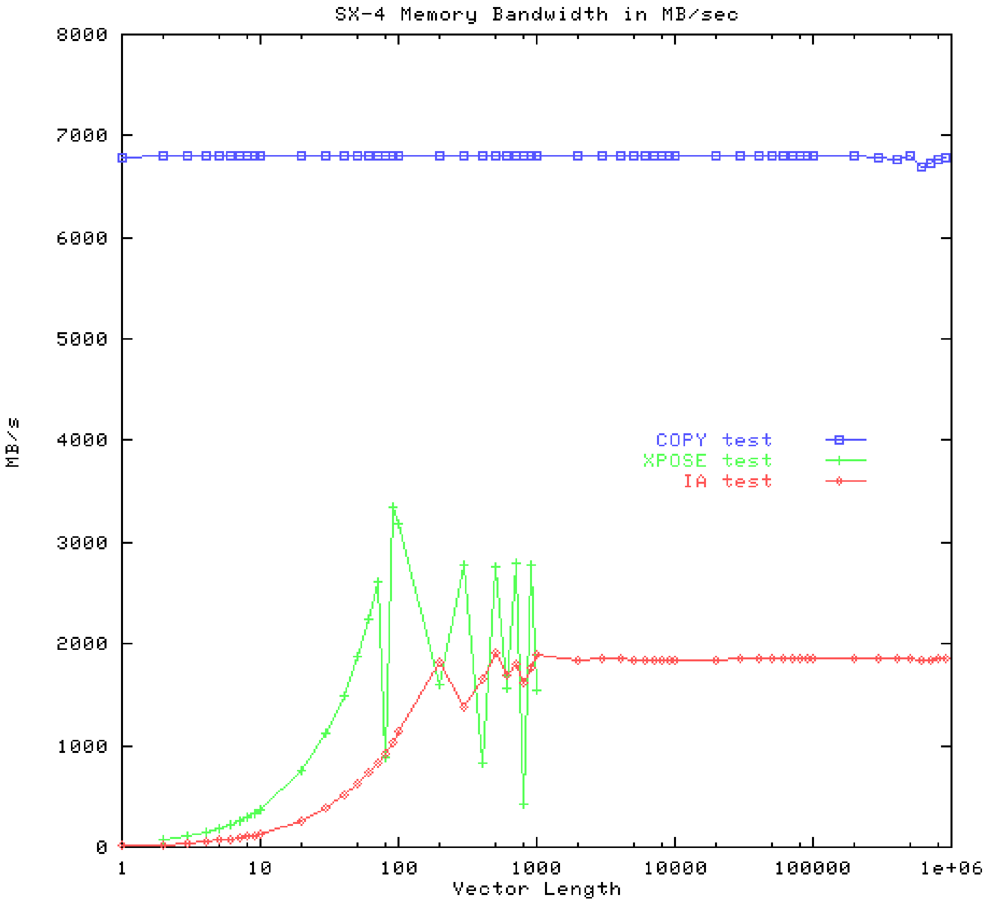

Figure 5: Measured memory bandwidth for three memory benchmarks measured in MB/sec on an SX-4/1.

4.3 Coding Style Comparison

RFFT and VFFT are two different implementations of a Fast Fourier Transform (FFT) algorithm from the FFTPACK library developed by P. N. Swarztrauber at NCAR. RFFT is a ‘‘scalar’’ real to complex FFT written in a style suited to cache-based processors. VFFT is a ‘‘vector’’ real to complex FFT written in a style suited to vector processors. This pair of codes is *not* a test of a vendor’s ability to rapidly compute FFTs. Instead, it is designed to help one understand the impact of loop ordering on processor performance and give guidance to code developers. The only significant difference between the two benchmarks is the order of the loops.

In the RFFT benchmark, the fastest varying axis is the FFT axis. In the VFFT benchmark, the fastest varying axis is the instance axis. The size of the FFT axis to be transformed ranges from 2 to 1280 in length. Three sets of FFT lengths are studied. These include pure power of two axes, a set of axes with one factor of three, the rest being factors of two, and a set of axes with one factor of five, the rest being factors of two. For RFFT, the instance loop is varied so as to maintain a roughly constant number of elements (~1000000) in the overall operation. This prevents the timing results from ranging over several orders of magnitude. The FFT array is dimensioned `a(N,M)`. For VFFT, the inner instance loop is varied to produce a representative range of vector lengths from 1 to 500. The FFT array is dimensioned `a(M,N)`, where `N` is the FFT axis and `M` is the instance axis. The FFT axis length N for each of the benchmarks is given below:

	RFFT	VFFT
$N = 2^n$	n=1,10	n=2,4,6,7,8,9
$N = 3*2^n$	n=0,8	n=0,2,4,6,
$N = 5*2^n$	n=0,8	n=0,2,4,6,8

For RFFT, the number of instances `M` varied from 500,000 to 800 depending on size of `N` and for VFFT the number of instances *M* took on the values *M = 1, 2, 5, 10, 20, 50, 100, 200, 500*.

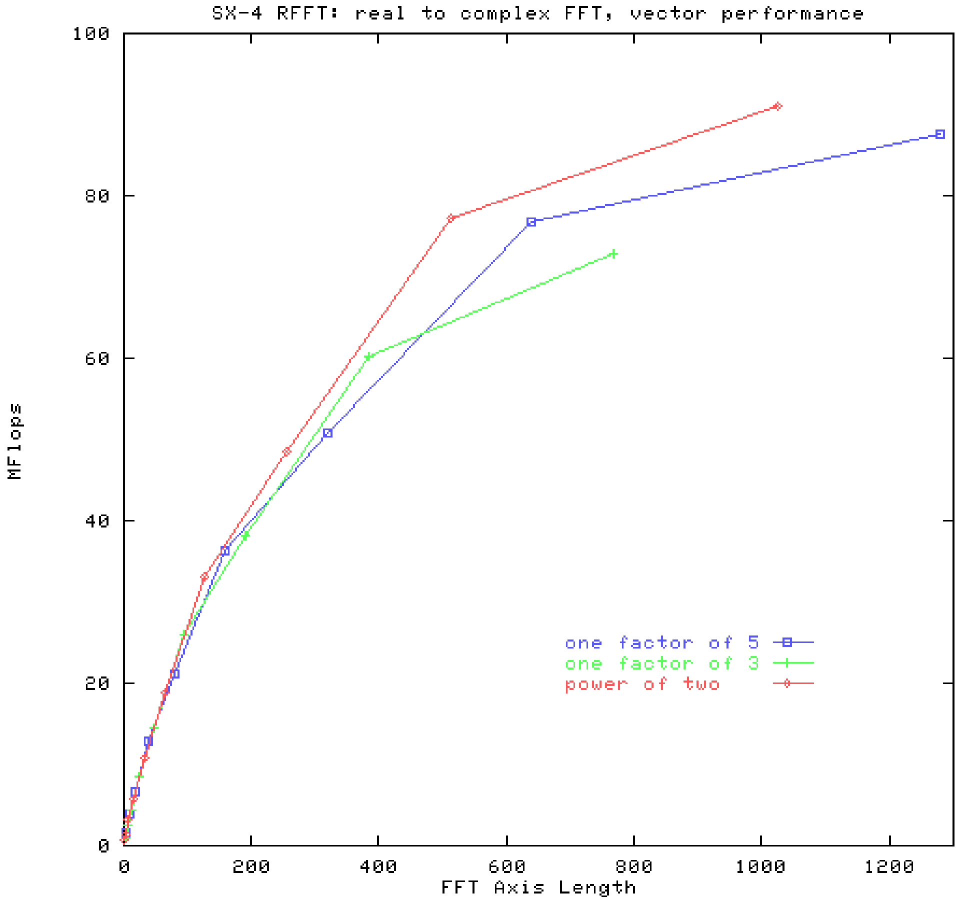

Figure 6: Results of the RFFT benchmark on an SX-4/1 measured in Mflops.

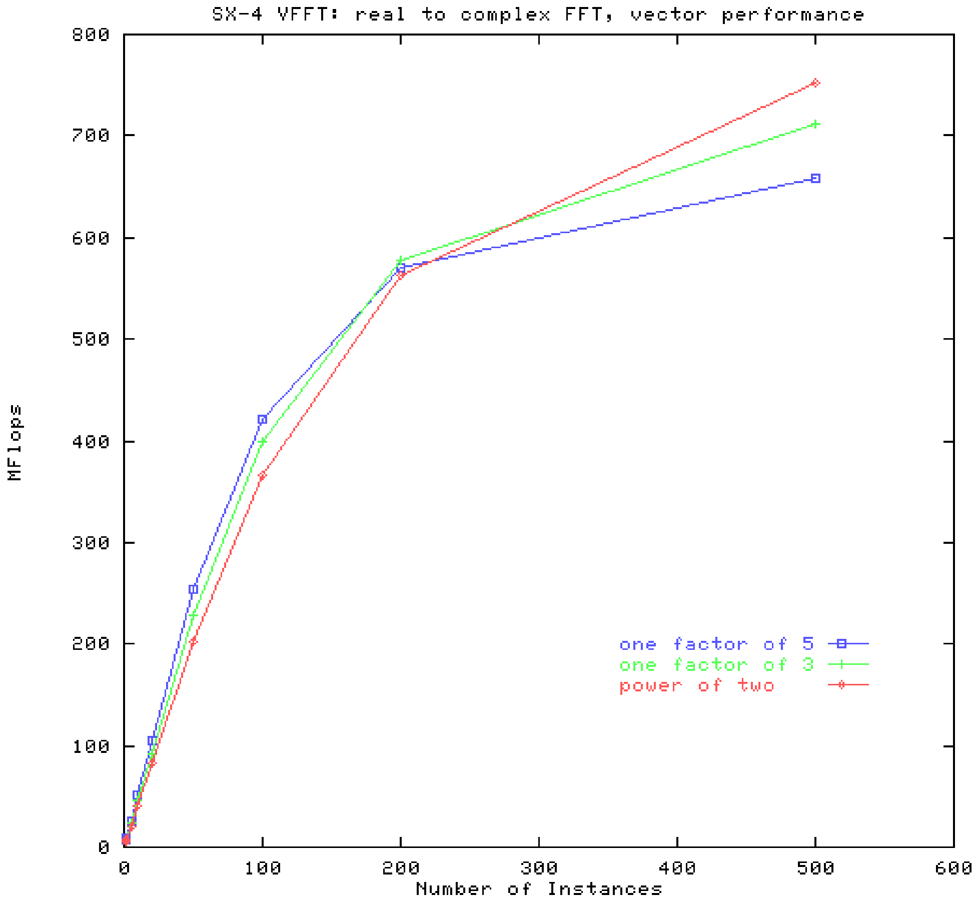

Figure 7: Results of the VFFT benchmark on an SX-4/1 measured in Mflops.

Figures 6 and 7 show the results of the RFFT and VFFT benchmark on the SX-4/1. The VFFT performance results are approximately an order of magnitude faster than those from RFFT. Thus, it is clear that applications developed for the SX-4 will achieve better performance if they are written in a style ammenable to vector processing as opposed to in a style for good cache utilization.

4.4 Raw Performance

The RADABS benchmark is intended to measure the proposed system's floating point performance on the single most time consuming subroutine in NCAR's Community Climate Model version 2 (CCM2). It is a computationally expensive radiation physics routine from CCM2. Much of the time in RADABS is spent in intrinsic function calls (`EXP`, `LOG`, `PWR`, `SIN`, and `SQRT`).

So, the performance on RADABS is related to the performance on the ELEFUNT benchmark. In addition there are numerous complex, multi-line equations that are representative of many of the atmospheric calculations that NCAR performs. This benchmark is to NCAR's climate codes what LINPACK is to numerical linear algebra applications. Our experience is that a computer's performance on RADABS sets an upper bound on CCM2 performance.

The physics calculations in RADABS are performed in a vertical column. RADABS is embarrassingly parallel in the latitude and longitude (i,j) directions. For the purposes of the benchmark, the initial data is identical in each vertical column. The performance demonstrated on this benchmark on the SX-4/1 is

865.9 Cray Y-MP equivalent Mflops.

4.5 Input/Output to Disk and Network

In evaluating a computer system, one needs to consider more than the processor to memory bandwidth and the performance of the processors. Like many other applications, atmospheric and related sciences are data intensive applications. Three benchmarks were developed to test the ability of a system to read and write data between memory and disk, and to test the speed of external network connections. These types of tests are important when one considers purchasing a computer. They insure that one can get data into and out of a computer and that it will interoperate with other computers. We include a brief description of each of the three benchmarks below. The results are not included since they are voluminous and the configuration of the tests is tuned to NCAR's computing environment.

4.5.1 Input/Output (I/O)

The first benchmark in this category is the I/O benchmark. It measures the performance of an attached, conventional disk system (not a solid-state disk) relative to reading initial climate model data and writing climate model output files, checkpoint/restart files, and intermediate data files used during a large simulation. This benchmark is run for multiple climate model resolutions. It writes a simulated header file and a simulated "history tape" file. The history tape file is an unformatted, direct access file so that if run on a multiprocessing system, different processors could write different records representing data associated with a specific latitude.

4.5.2 HIPPI bandwidth (HIPPI)

The second benchmark in this category is a HIPPI benchmark. It is intended to insure interoperability of a computer system with the NCAR Mass Storage System, which is HIPPI-based. It measures the communication bandwidth using HIPPI for single data transfers and multiple concurrent data transfers. It demonstrates the ability of a system to send and receive "raw" HIPPI packets of varying sizes, and to measure the data rate of the HIPPI transfers.

4.5.3 FDDI/IP External Network Benchmark (NETWORK)

The third benchmark in this category is the NETWORK benchmark. It is a shell script that tests system IP capabilities. Briefly, the benchmark starts with a small number of environmental variables that the user must set to be consistent with the test environment. There are two types of tests - data-transfer commands and non-data-transfer commands. Data-transfer commands are to be executed between the benchmarked machine and (if possible) a target machine identical or comparable to the benchmarked machine. Non-data-transfer commands will inherently execute on the benchmarked machine.

4.6 Production Workload (PRODLOAD)

The PRODLOAD benchmark provides a performance measure for overall system performance in a production environment. It consists of different numbers of application codes (CCM2 resolution T42 and T106) and a HIPPI test running concurrently. Groups of applications codes are run one after another to simulate a series of long running simulations that stop and restart. The benchmark measures the effect of system loading by running varying numbers of these applications simultaneously. Finally, two large application codes (CCM2 at T170 resolution) are run concurrently.

We define a "job" to be composed of the HIPPI Benchmark and three copies of the CCM2 executing simultaneously. The CCM2 runs are a 3-day simulation at resolution T106 and two 20-day simulations at T42 resolution. A job is considered complete when all of its components are finished executing. Test one consists of one sequence of four jobs run one after another. Test two consists of two sequences of four jobs run one after another. Test three consists of three sequences of four jobs run one after another. The sequences within a test are run concurrently. The fourth test consists of two CCM2 2-day runs at resolution T170 executing concurrently.

The performance measurement in this benchmark is the wall clock time required to complete the entire benchmark. The start time for the each test shall be when the first job begins. The stop time shall be when the last job has completed. In addition, the start and stop times of individual jobs is considered in order to identify system specific characteristics.

The NEC SX-4/32 completed the PRODLOAD benchmark in 93 minutes and 28 seconds (with the 9.2 ns clock).

4.7 Complete Applications

The final category in the NCAR Benchmark Suite is applications. We include three complete geophysical simulation codes - one atmospheric general circulation code and two global ocean simulation codes. For the NCAR procurement, we permitted the Vendor to choose to benchmark one or the other ocean codes.

The application tests include running the models at different resolutions for a fixed number of processors to evaluate the performance as a function of problem size. Also, system scalability is measured by fixing the problem size and varying the number of processors. Finally, each application has a correctness check that must be passed to verify that the application is running properly as well as fast.

4.7.1 CCM2

CCM2 is an atmospheric general circulation model that has been developed at NCAR and provided to atmospheric scientists for over a decade [2,9,10]. Readers interested in the an introductory discussion of the scientific issues of climate modeling are referred to [14]. CCM2 consists of approximately 40,000 lines of Fortran 77 organized as 232 subroutine and include files which have been optimized for vector processor systems. The vertical and temporal aspects of the model are represented by finite-difference approximations. The spherical harmonic transform (spectral transform) method is employed to compute the dry dynamics of CCM2 [10,14]. It consists of computing the spherical harmonic function coefficient representation of the atmospheric state variables through a series of highly non-local operations. The set of spectral coefficients is typically truncated in some fashion to avoid aliasing. Horizontal derivatives and linear terms involving these variables are calculated in spectral space and are combined to form the dynamical right hand sides in spectral coefficient space. This operation is completely local in spectral coefficient space. The data are then transformed back into grid space where they are used to update the model variables.

For accuracy reasons, the spectral transform calculations are performed on a polar grid which is irregularly spaced in latitude, called a Gaussian polar grid. The calculation of non linear terms in the equations of motion are carried out on this grid, as are the physical parameterizations of CCM2. These ‘‘physics’’ computations involve only the vertical column above each grid point and are thus numerically independent of each other in the horizontal direction. Finally, trace gases, including water vapor, are transported by the wind fields using a shape preserving SLT scheme [12,15]. This transport involves indirect addressing on the Gaussian polar grid.

For spectral climate models such as CCM2 it is canonical to denote the resolution by the truncation wave number and the number of vertical layers in the model discretization. For example, a spectral atmospheric model that uses a 128 longitude by 64 latitude grid and 18 vertical levels is called a T42L18 model. The ‘‘T’’ indicates a triangular truncation of the spherical harmonic coefficients, 42 indicates the maximum longitudinal wavenumber used in the model, and the ‘‘L’’ denotes the number of vertical levels. At present T42L18 is the production resolution of CCM2. Table 4 shows the grid size, nominal grid point spacing, and model time step for five CCM2 resolutions.

Model Resolution	**Horizontal Grid Size**	**Nominal Grid Spacing**	**Time Step**
T42L18	64 x 128	2.8 degrees	20.0 min.
T63L18	96 x 192	2.1 degrees	12.0 min.
T85L18	128 x 256	1.4 degrees	10.0 min.
T106L18	160 x 320	1.1 degrees	7.5 min.
T170L18	256 x 512	0.7 degrees	5.0 min.

Table 4: Typical CCM2 resolutions, grid spacings, and time steps.

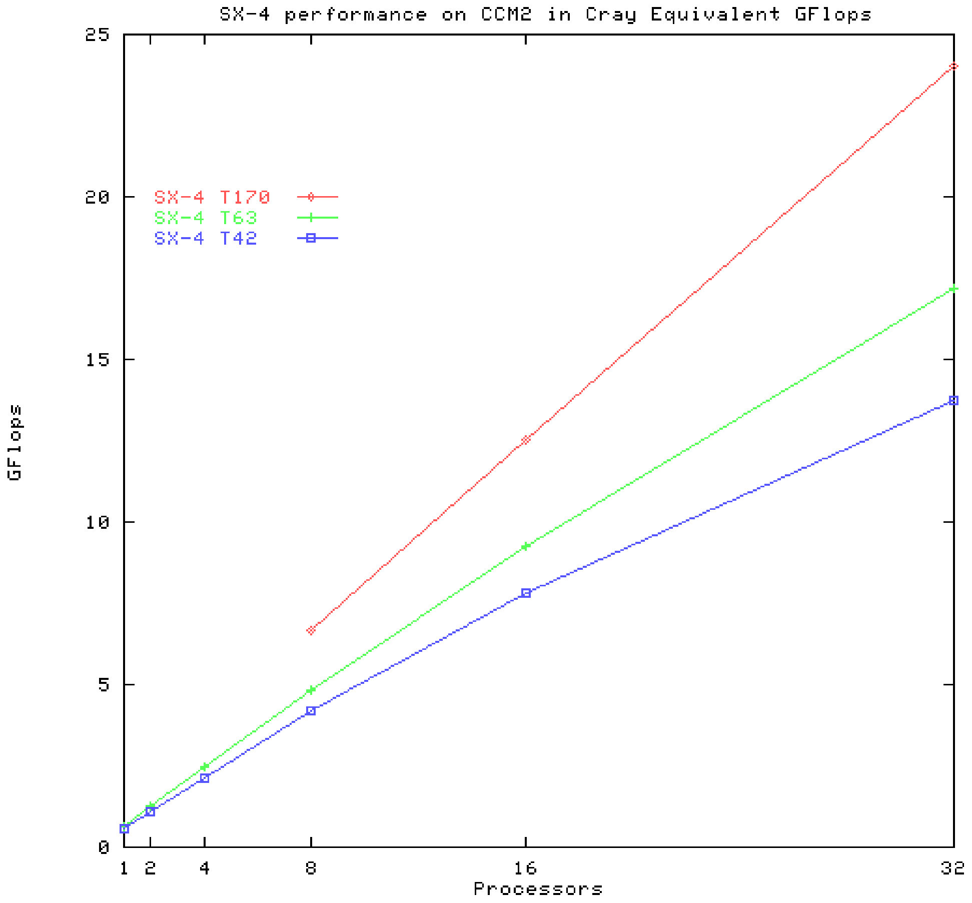

Figure 8: CCM2 performance measured for various numbers of processors in Cray equivalent GFlops for three different model resolutions on the SX-4/32.

Figure 8 shows both CCM2 performance as a function of problem size and scalability of the SX-4/32. Note that the sustained performance of CCM2 at resolution T170L18 on 32 processors is 24 Gflops. This is on a system with a 9.2 ns clock. We anticipate that an additional 15% performance improvement can be realized with some code tuning and running on a system with an 8.0 ns clock. It is clear from Figure 8 that the SX-4 runs most efficiently on long vector problems and that medium and large problems scale reasonably well.

Another test performed using the CCM2 benchmark was simulation of a year of climate at resolutions T42L18 and T63L18. In this test the application writes daily average climate statistics each model day. Approximately 15GB of model data and restart information were written during the T63L18 test. Table 5 shows the results for these two tests on the SX-4/32.

Resolution	**T42L18**	**T63L18**
Time	1327.53	3452.48

Table 5: Time in seconds to simulate one year of climate for resolutions T42L18 and T63L18.

The final test with the CCM2 benchmark is the ensemble test. It measures degradation of performance for running multiple copies of a program relative to running a single instance of the program. In the past we have observed a significant difference between the performance of a single application on a quiescent system and the performance of an application running on a system with other applications. In this test a single instance of a 12-day run of CCM2 at T42L18 was timed on four processors of the system. Then, there was the multiple copies test. We ran eight concurrent 4-processor copies of the single instance code so that all processors were engaged in the computation. Table 6 shows the wall clock time for running a single 4-processor job and for running eight 4-processor jobs. The relative degradation of the job is only 1.89%.

single instance time	**257.13**
multiple instance time	**262.00**
relative degradation	**1.89%**

Table 6: Single and multiple instance times for the ensemble test measured in seconds and relative performance degradation on the SX-4/32.

4.7.2 Modular Ocean Model (MOM)

The MOM benchmark is based on the NOAA Geophysical Fluid Dynamics Laboratory (GFDL) Modular Ocean Model (MOM) Version 1.1. However, there are significant modifications to both the physics and computational aspects of the code from the base GFDL version. The model is a finite difference formulation of the rigid-lid, boussinesq primitive equations on the sphere, formulated in latitude-longitude-depth coordinates. The model predicts temperature, salinity, three components of velocity and a number of related diagnostic quantities (pressure, diffusivities, ...).

The code provided in the NCAR Benchmark suite is configured in the global domain in low and high resolution versions. The code for the two versions is identical; only a few include files (specifying array dimensions and some constants) and the input data files differ. The low resolution version has a nominal horizontal resolution of 3 degrees latitude-longitude with 25 levels in the vertical. It can be used for purposes of familiarization and porting verification. A run of 40 timesteps should take only a few minutes of CPU time on a fast workstation and is used for testing and verification of the model. The high resolution version is used as the benchmark. It has a nominal horizontal resolution of 1 degree latitude-longitude, with 45 levels in the vertical.

CPUs	Time for 350 time steps	Speedup
1	1861.25	1.00
2	-	-
4	696.92	2.70
8	519.74	3.66
16	331.67	5.88
32	226.62	9.06

Table 7: MOM Ocean Model benchmark performance. Time in seconds for 350 time steps and speedup relative to performance on one processor.

Table 7 shows the performance and speedup of the MOM benchmark on the SX-4/32. For each number of processors, we ran for 40 time steps and then for 390 time steps and subtracted the two times to remove initialization time. Also, for expediency in completing the benchmark test, no two processor tests were made. The modest level of scalability is in part due to the fact that the benchmark prints out model diagnostics every 10 timesteps and in part with the algorithms and coding of the application and not with the SX-4. The scalability is consistent with the results that we see on parallel vector machines that reside at NCAR.

4.7.3 Parallel Ocean Program (POP)

The POP benchmark is based on the Parallel Ocean Program developed at Los Alamos National Laboratory by R. D. Smith, J.K. Dukowicz, and R. C. Malone [6,7,13]. It is designed as a portable ocean model for single and multiple processor machines. The model source code is written in Fortran 90 and makes extensive use of the C-preprocessor. Additionally, various physical parameterizations and other options are turned on or off via the preprocessor options for this benchmark. It is a stand-alone code with a free surface formulation and flat bottom topography. The code is portable and scalable and runs on such systems as the CRI T3D and TMC CM-5.

A pre-release of the NEC F90 compiler was used for this benchmark test. At the time, the `CSHIFT` intrinsic did not vectorize. Even so, we observed 537 Mflops on a the 2-degree POP benchmark on one processor of the SX-4.

5 Conclusions

In this paper we described the architecture of the SX-4. It is a shared-memory vector supercomputer manufactured with CMOS components running at an 8.0 ns clock cycle and a peak performance of 2 Gflops per processor. We also described the composition of the NCAR Benchmark Suite, designed to evaluate the computers for use on climate modeling applications. The benchmarks are a mix of kernels and complete applications that measure such aspects as accuracy of intrinsics, memory to memory bandwidth, processor speed, and memory to disk I/O rates. Additionally, we contrasted this benchmark suite with other benchmarks and discussed why they were inappropriate for use in evaluating a computer for use at NCAR. Finally, we detailed the scalability and performance of the SX-4/32 relative to to the NCAR Benchmark Suite. In particular, the SX-4/32 sustained 24 Gflops on CCM2 at resolution T170L18, completed a one year simulation of global climate at T63L18 in 57.5 minutes which included writing approximately 15GB of data, and the ensemble test demonstrated that there is very little degradation of performance under load.

Acknowledgements

Many people at NCAR contributed to the assembly and testing of the NCAR benchmark suite. We would like to thank Jim Hack, Dennis Colarelli, John Dennis, Tony Craig, Frank Bryan, Basil Irwin, John Merrill, Dan Anderson, and Jeff Kuehn. HTML help was graciously provided by webmaster Greg McArthur. Finally, we would like to acknowledge Dave Parks and Per Nyberg for their help in getting the benchmark results.

References

1. D. Bailey, et. al. "The NAS Parallel Benchmarks" , *RNR Technical Report RNR-94-007*, March 1994.

2. L. M. Bath, J. R. Rosinski, and J.Olson. "User’s guide to (CCM2)." Technical Report NCAR Technical Note/TN-382+IA, Climate and Global Dynamics Division, National Center for Atmospheric Research, PO Box 3000, Boulder, CO 80307, 1992.

3. J. Dongarra, L. Martin, and J. Worlton, "Computer Benchmarking: paths and pitfalls," *IEEE Spectrum*, July 1987, pp38-43.

4. J. Dongarra, "The LINPACK Benchmark: An explanation", *Supercomputing*, Spring 1988, pp10-14.

5. J. Dongarra, "Performance of Various Computers using Standard Linear Equations Software in a Fortran Environment", *TR MCSRD 23*, Argonne National Laboratory, March 1988.

6. J. K. Dukowicz, R. D. Smith, and R. C. Malone. "A Reformulation and Implementation of the Bryan-Cox-Semnter Ocean Model on the Connection Machine," *J. of Atmos. and Oceanic Tech.* vol 10, no 2, pp 195-208, 1993.

7. J. K. Dukowicz and R. D. Smith. "Implicit free-surface method for the Bryan-Cox-Semnter Ocean Model," *J. of Geoph. Research* vol 99, no C4, pp 7991-8014, 1994.

8. J.L. Gustafson and Q. O. Snell, "HINT: A new way to measure Computer Performance". *Proceedings of the HICSS-28 Conference*, Wailela, Maui, Hawaii, January 3-6, 1995.

9. J.J. Hack, B.A. Boville, B. P. Briegleb, J.T. Kiehl, P. J. Rasch, and D. L. Williamson. "Description of the NCAR Community Climate Model (CCM2)." Technical Report NCAR Technical Note/TN-382+STR, Climate and Global Dynamics Division, National Center for Atmospheric Research, PO Box 3000, Boulder, CO 80307, 1993.

10. J. J. Hack, J. M. Rosinski, D. L. Williamson, B. A. Boville, and J. E. Truesdale. "Computational Design of the NCAR Community Climate Model." *Parallel Computing*, volume 21, 1995.

11. J. D. McCalpin, "Memory Bandwidth and Machine Balance in Current High Performance Computers," *IEEE Computer Society,* Technical Committee on Computer Architecture Newsletter, December 1995, pp 19-25.

12. P. J. Rasch and D. L. Williamson. "Computational aspects of moisture transport in global models of the atmosphere. " *Quart. J. Roy. Meteor. Soc.*, **119**:1071--1090, 1990.

13. R. D. Smith, J. K. Dukowicz, and R. C. Malone. "Parallel ocean general circulation modeling," *Physica D* **60**:38-61, 1992.

14. W. M. Washington and C. L. Parkinson. *An Introduction to Three-Dimensional Climate Modeling*, University Science Books, pp 181-205, 1986.

15. D. L. Williamson and P. J. Rasch. "Two-dimensional semi-Lagrange transport with shape preserving interpolation." *Mon. Wea. Rev.*, **117**:102-129, 1989.