

April, 2003  
Vol. 13, No. 2

# TeraWord

The Newsletter of High-Performance Computing and Communications in Nevada

## Then & Now: NSCEE Performance

### Contents

Then & Now:  
NSCEE Performance .1

Research Activities  
at NSCEE  
NOAA .....2  
E-Health .....2

Help Desk:  
Onyx 3800  
Single Processor  
Tuning Guide . . . . .3

Onyx 3800  
Multiprocessor  
Tuning Guide . . . . .3

NSCEE History . . . . .4

Articles Invited . . . . .4

The LINPACK Benchmark\* is widely accepted in both the computer industry and user community as the initial measure of floating point performance of a compute server. Jack Dongarra, University of Tennessee and Oak Ridge National Laboratory, developed this floating-point performance benchmark that involves solving a dense system of linear equations. All performance numbers reflect arithmetic performed in full 64-bit precision.

The **Best Effort** test case is for solving a system of equations of order 1000,

with no restriction on the method or its implementation. In fact, vendors are allowed and encouraged to completely rewrite the solver to optimize performance and achieve as high an execution rate as possible.

**Theoretical Peak** is based not on an actual program run, but on a paper computation and is determined by counting the number of floating-point additions and multiplications that can be completed during the cycle time of the machine. NSCEE's Cray Y-MP had a cycle time of 6ns. During

one cycle the results of both an addition and a multiplication could be completed in (2 operations/1 cycle) x (1 cycle/6 ns), approximately equal to 333 Mflops on a single processor. NSCEE's Cray had 2 CPUs, hence the 667 Mflops.

The increase in performance between NSCEE's first supercomputer and the current SGI Onyx 3800 is remarkable, especially considering that the Cray Y-MP, when it arrived on campus in 1990, was one of the 10 fastest, high-performance machines in the world. §

Resource	Year	Number of CPUs	Memory	Disk Space
Cray Y-MP 2/216	1990	2	128 MB	24 GB
Convex C-220	1992	2	256 MB	16 GB
SGI Origin 2000	1998	14	1,320 MB	79 GB
SGI Onyx 3800	2003	32	32 GB	3.6 TB

1,000 Megabytes (MB) = 1 Gigabyte (GB)  
1,000 Gigabytes = 1 Terabyte (TB)

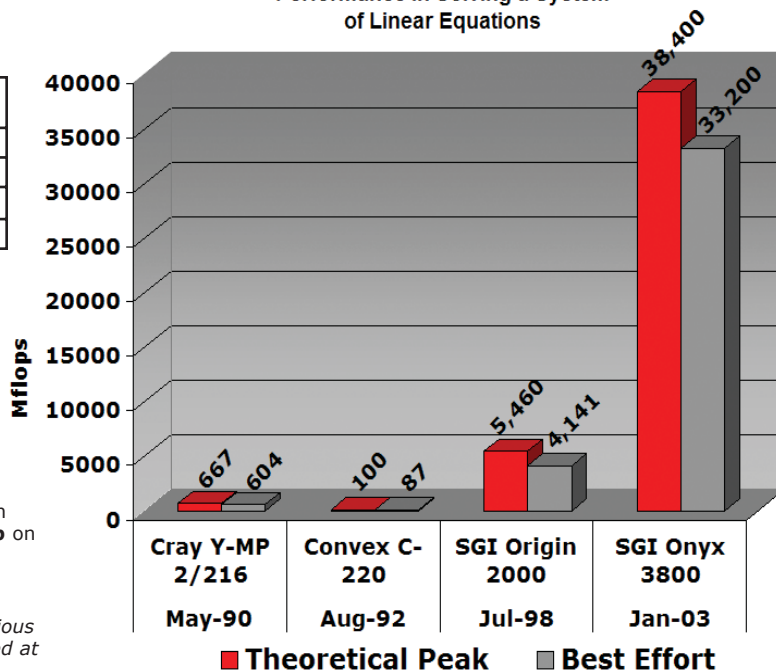
The term **Mflops** stands for millions of floating-point operations completed per second. The **Mflops** shown in the graph are based on the actual number of CPUs in each of NSCEE's supercomputers.

### Special Note: Costing Then & Now

Based on institutional pricing, the cost in 1990 of one Mflop on NSCEE's Cray Y-MP was **\$11,000** compared to **\$16 per Mflop** on NSCEE's current SGI Onyx 3800!

\*An up-to-date version of Jack Dongarra's "Performance of Various Computers Using Standard Linear Equations Software", is located at <http://www.netlib.org/benchmark/performance.ps>

Performance in Solving a System of Linear Equations

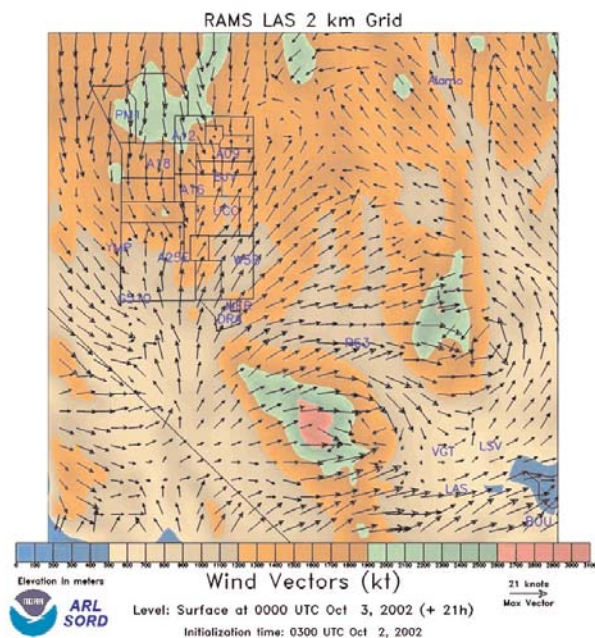


# Research Activities at NSCEE

This issue of TeraWord continues and completes the discussion from the January, 2003 newsletter (Volume 13, No. 1) of two of NSCEE's many on-going projects that use NSCEE compute resources.

## NOAA/ARL Collaborates with NSCEE on Weather Simulation Model

The National Oceanic and Atmospheric Administration's (NOAA) Special Operations and Research Division (SORD) of the Air Resources Laboratory (ARL) in Las Vegas runs a high resolution weather simulation model on the UNLV's SGI Supercomputer daily. ARL/SORD uses the UNLV supercomputer capability to run atmospheric simulations at resolutions of 32, 8 and 2 km, covering the Southwestern US down to Las Vegas.



At the beginning of each model simulation, the NSCEE Computer connects to computers at ARL Headquarters in Washington, DC, and downloads model initialization data fields generated by the National Centers for Environmental Prediction (NCEP). The NSCEE computer creates initial data files for the model using a single processor. The 37-hour simulation, 3 resolution/domain model run begins at 10 pm each night using 12 processors of the NSCEE SGI and takes approximately 7 hours to complete. Once complete, the University computer again connects to ARL Headquarter computers in Washington, DC, and downloads model results. Locally, the ARL/SORD computers connect to the NSCEE and download the model result data files. These data files are then processed into graphics that are posted to the ARL/SORD website

[http://www.sord.nv.doe.gov/home\\_models.htm](http://www.sord.nv.doe.gov/home_models.htm)

for use by forecasters and operation support meteorologists.

## UNLV E-Health Initiative

The on-going telemedicine and distance education program represents a model program for the electronic delivery of health care and education.

### Technical Challenges and Benefits

The capability of network technologies to support health-care applications depends on whether the relevant technical needs are met and whether the operational aspects of the systems involved are understood and manageable. Ongoing efforts to enhance the capabilities of networking technology will produce many benefits for the healthcare community. The research community will provide enhanced QoS guarantees and improved security algorithms for healthcare information management. They will expand broadband access options for consumers, and protect consumer privacy.

The technologies planned for deployment across the Internet in the near future will not fully meet the needs of critical healthcare applications. In particular, QoS may not meet the need for dynamically variable service between communicating entities. Security technologies may not provide for the widespread issuance of certificates to healthcare consumers. And the new technology will not necessarily provide the degree of reliability needed for mission-critical health applications. Although much can be done with the technologies currently planned, additional effort will be needed to make network technology even more useful to the healthcare community. This initiative will review current efforts to improve the capabilities of network technology and will evaluate them on the basis of the needs of the healthcare sector.

### Benefits

Telemedicine will certainly improve overall healthcare services for patients with limited access due to their geographic location in rural or certain urban areas, or due to restricted mobility as a result of disability or other physical restriction. The new technology will also level the playing field of socioeconomic status, which currently substantially determines patients' access to care. Indeed, telemedicine has the clear potential to overcome physical, geographic and economic barriers and to lead to improved healthcare.

## A Storage Definition

**HSM:** *Hierarchical Storage Management.*

The process of **automatically** storing data on the lowest-cost devices that can support the performance required by the applications. To users, **data storage never fills** and **file access, regardless of location in the storage hierarchy, is completely transparent.** The software automatically manages multiple levels of storage hierarchy. The operating environment of NSCEE's StorageTek PowderHorn is based on HSM.

# Help Desk

## Onyx 3800 Single Processor Tuning Guide

This version of Help Desk continues and completes the January, 2003 TeraWord (Volume 13, No. 1), **Onyx 3800 Single Processor Tuning Guide**. Refer to it for a discussion of these first steps:

- One: Get the Right Answer,
- Two: Use Existing Tuned Code, and
- Three: Find Out Where to Tune.

### Step Four: Find the Optimum Compilation Flags

Don't rely on default options. Turn off debugging flags, try these optimization flags:

Option	Effect and Purpose
-Ofast=ip35	Macro for compiler group's top picks for Onyx 3800 platform speed.
-O3	Enable maximum optimizations (includes LNO).
-mips4	For R14K, R12K, R10K, R8K,R5K chips only.
-OPT:IEEE_arithmetic=3 -OPT:roundoff=3	See <i>f77(1)</i> or <i>cc(1)</i> for discussion. Check that answers are still correct after applying these.
-IPA=on	Enable inter-procedural analysis. Changes relative times of compile vs. link; see <i>ipa(5)</i> .
-OPT:alias=<name>	Disambiguate pointer references. Use <name>=disjoint or restrict whenever possible. See <i>cc(1)</i> .

### Step Five: Tune Cache Performance

- Use Loop Nest Optimizations (via -O3 or -Ofast) to enable loop fusion, interchange, cache blocking array padding and prefetching.
- Use stride-1 accesses whenever possible.
- Group together data used at the same time.
- Use LNO directives to fine-tune its actions (see *f77(1)* and *cc(1)*).
- Use larger page sizes to reduce TLB misses.

## Onyx 3800 Multiprocessor Tuning Guide

### Step One: Tune Single Processor Performance

Complete steps 1 through 5 as described above in the **Onyx 3800 Single Processor Tuning Guide**.

### Step Two: Parallelize Code

Choose parallelization methodology:

- Automatic parallelization: -pfa or -pca (see *MIPSpro Power Fortran Programmer's Guide*, *IRIS POWER C User's Guide*)
- MP Library directives: -mp (see *MIPSpro Fortran Programmer's Guide*, *C Language Reference Manual*)
- Other libraries: MPI, PVM, pthreads (see *Topics in IRIX Programming*)

Profile code to monitor degree of parallelization:

- Vary number of threads (e.g., for MP library, use `setenv MP_SET_NUMTHREADS`)
- Measure wall clock time to determine speedup (e.g., use `timex(1)`)
- `perfx -a -mp` prints all counts for each thread
- SpeedShop generates an output file for each thread

CPU activity may be displayed with `gr_osview(1)` and `top(1)`, memory usage with `gmemusage(1)`, hardware configuration with `hinv(1)` and `topology(1)`.

### Step Three: Identify Bottlenecks

- Is load balance OK?  
e.g., to check balance of floating point operations  
`perfx -a -mp prog args |& grep "floating point"`
- Are there a lot of secondary cache misses?  
`perfx -a -y prog args`

If so, false sharing or data placement may be a problem.

- Is there false sharing?  
Check `perfx` output to determine if interventions and/or invalidations are a large fraction of secondary cache misses.

### Step Four: Fix False Sharing

If false sharing is a problem, use SpeedShop to monitor stores to shared cache lines:

```
setenv _SPEEDSHOP_HWC_COUNTER_NUMBER 31  
ssrun -prof_hwc prog args
```

Revise data structures or algorithms to remedy the problem.

### Step Five: Tune for Data Placement

For *libmp* programs:

For well-parallelized *libmp* programs which generate a lot of secondary cache misses but do not scale well, determine sensitivity to data placement. Try as few steps as needed to achieve satisfactory performance:

- 1) Try round-robin page allocation  
(`setenv _DSM_ROUND_ROBIN on`)
- 2) Try page migration (`setenv _DSM_MIGRATION on`) with round-robin

If these techniques don't solve scaling problems, the program needs to be modified to make sure the data are properly distributed.

- 3) Make sure data initializations are parallelized. This relies on "first touch": data are stored in memory of processor which first accesses a page of data.
- 4) Ensure proper data placement via `C$DISTRIBUTE` and `C$PAGE_PLACE`.
- 5) For fine-grain data distributions (chunks smaller than a page), consider `C$DISTRIBUTE_RESHAPE` directive.

The *MIPSpro Fortran Programmer's Guide* and *C Language Reference Manual* describe the data distribution directives and the following environment variables:

Variable	Use and Possible Values	Default
<code>_DSM_VERBOSE</code>	Set (any value) to get runtime report of memory and thread placement.	not set
<code>_DSM_ROUND_ROBIN</code>	Cause pages to be allocated cyclically	not set
<code>_DSM_MIGRATION</code>	Set ON to enable migration of explicitly placed data, ALL_ON to enable migration of any data.	OFF
<code>_DSM_PPM</code>	Processes per memory (node), set to 1 to use only one CPU per memory (but reduce numthreads also).	2
<code>_DSM_FOP</code>	Enable inter-thread barrier synchronization using <code>fetch+op</code> instructions.	not set
<code>_DSM_MUSTRUN</code>	Set (any value) to lock threads to processors, Not recommended in time-sharing environments.	not set
<code>_DSM_OFF</code>	Set (any value) to disable distribution.	not set
<code>PAGESIZE_STACK, DATA, and _TEXT</code>	Set virtual pages sizes in KB (e.g., 64). May reduce number of TLB faults.	16

Continued from page 3

For non-libmp programs, use dplace

```
dplace [-place placement_file] [-data_pagesize n-bytes] [-stack_pagesize n-bytes] [-text_pagesize n-bytes] [-migration_threshold] [-propagate] [-mustrun] [-v[erbose]] program [program-arguments]
```

MPI 2.0 placement:

```
setenv MPI_NP <n>
mpirun -np $MPI_NP /usr/sbin/dplace [dplace args] ./a.out [args]
```

Scalable placement\_file for MPI 2.0

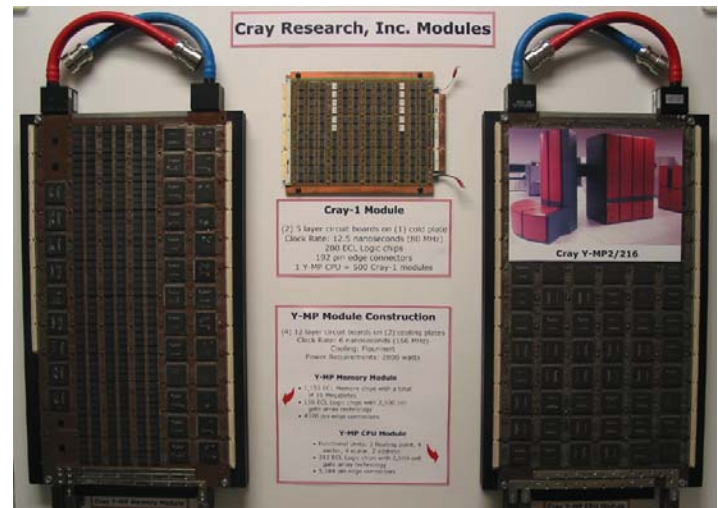
```
# SET UP COMPACT MEMORIES
memories ($MPI_NP+1)/2 in topology cube
# NUMBER OF THREADS
threads $MPI_NP+1
# IGNORE LAZY THREAD 0
distribute threads 1:$MPI_NP across memories
```

## A Compute Definition

**NUMA:** *Nonuniform Memory Access.*

A form of Shared Memory Processing (SMP), also known as shared memory cluster (SMC). A hybrid between symmetrical multiprocessing and clustering. Arranges multiple processors into small groups of processors, all of which communicate with each other. Designed to extend scalability beyond the traditional SMP system bottlenecks. Memory is logically shared. NUMA system architecture is the basis of NSCEE's SGI Onyx 3800.

## NSCEE History



Pictured above are a Memory Module (left) and a CPU Module (right) from a Cray supercomputer, identical to NSCEE's first supercomputer, the Cray Y-MP 2/216. The modules were liquid cooled by an inert carbon, flourinert, as indicated by the red and blue in- and out-take hoses at the top of each board. The blue circulated cooled liquid in and the red carried heated liquid away. When NSCEE's Y-MP arrived at UNLV in 1990, it was one of the 10 fastest machines in existence! Three years later, in 1993, it had moved to 401st. The cost of *each* of these modules, in 1990 dollars, was about \$1M. NSCEE's Cray had only 24 GB of disk storage compared to the 3.6 TB (3,600 GB) of disk storage with our current SGI Onyx 3800. What now seem like enormous limitations (in CPUs, disk storage, memory, etc.), were remarkable achievements in their time and make the Cray Y-MP still revered as one of the truly great compute engines ever to be built. How fortunate we are to have this legacy!

### Articles Invited

The National Supercomputing Center for Energy and the Environment invites you to contribute articles on your work on high-performance computers and especially our resources. Please submit your articles to:

TeraWord	email	
UNLV/NSCEE	teraword@nscee.edu	
4505 Maryland Parkway		
Box 454028	Phone	Fax
Las Vegas, NV 89154-4028	(702) 895-4153	(702) 895-4156

TeraWord is published by the National Supercomputing Center for Energy and the Environment. Materials of interest in the newsletter may be reprinted, provided acknowledgement of the source is included. Hardware and software products mentioned in this publication are trademarks of their respective companies. The use of their names does not constitute an endorsement or approval by the NSCEE or the University of Nevada Las Vegas.

# NSCEE

## National Supercomputing Center for Energy and the Environment

### High-Performance Computing and Communications in Nevada

National Supercomputing Center for Energy and the Environment  
4505 Maryland Parkway, Box 454028  
Las Vegas, NV 89154-4028

**UNLV**  
UNIVERSITY OF NEVADA LAS VEGAS

Visit us at [www.nscee.edu](http://www.nscee.edu)