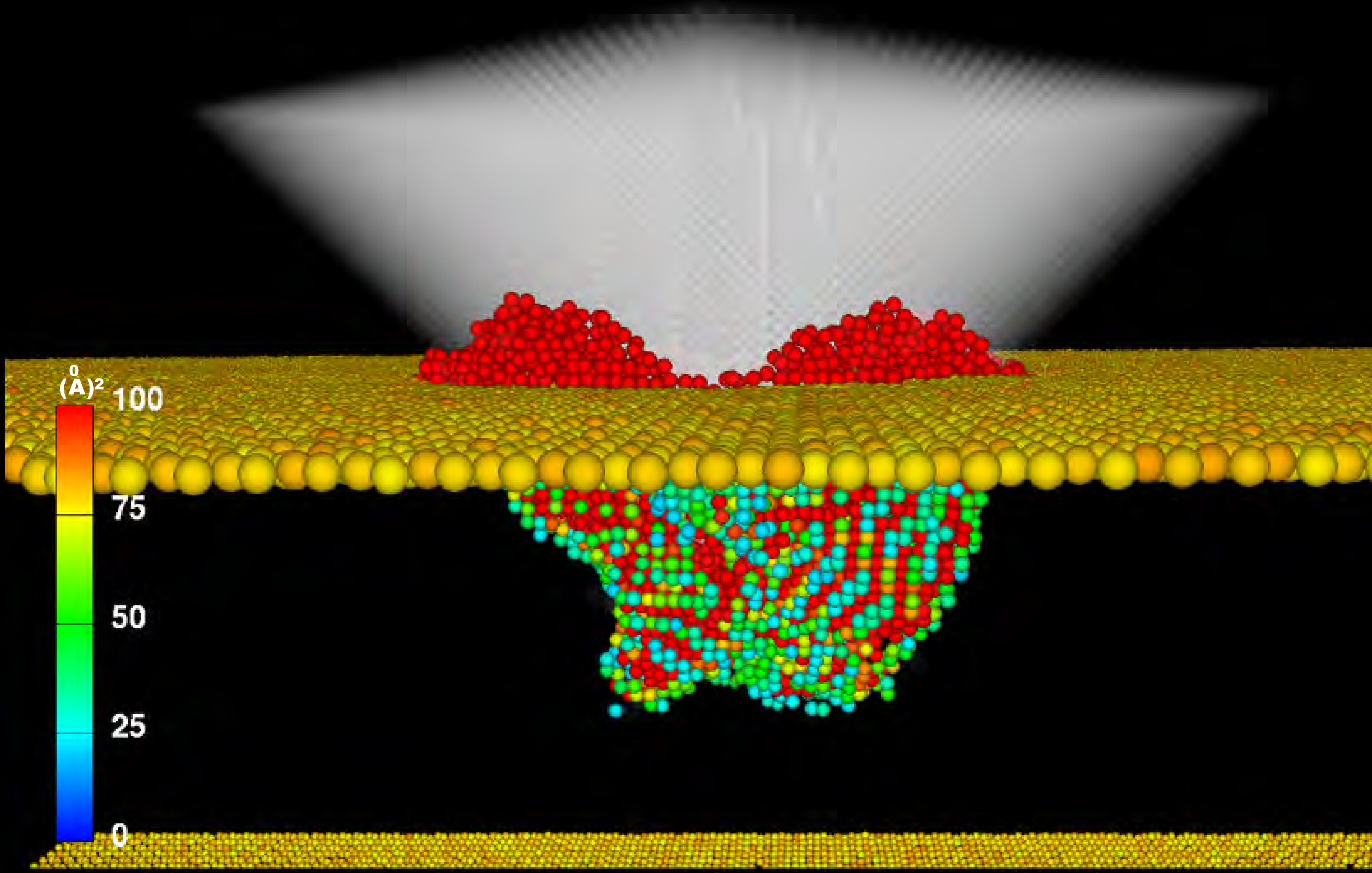
Navigator

NAVO MSRC Fall 2001



News and information from...

The Naval Oceanographic Office Major Shared Resource Center

The Director's Corner

Steve Adamec, NAVO MSRC Director

Moving Ahead at the NAVO MSRC

Fall is arriving after what has been a monumentally busy and tragic summer. A record audience of approximately 400 folks attended the 2001 Users Group Conference in Biloxi, Mississippi in June. The Major Shared Resource Center (MSRC) Technology Insertion for FY01 (TI-01) planning and approval activities were completed, with substantial computing upgrades to be delivered shortly to the Aeronautical Systems Center and Engineer Research and Development Center MSRCs. The MSRC TI-02 planning and approval activities have subsequently begun, with major upgrades to be delivered to the Applied Research Laboratory and NAVO MSRCs by June of 2002. The follow-on contract award for the Defense Research and Engineering Network (DREN) has been delayed slightly, but final award is anticipated for this fall. The new DREN capability promises to be much more capable and cost-effective—hats off to Mr. Rodger Johnson of the High Performance Computing Modernization Office (HPCMO) and the entire DREN team who are making it happen. And finally, the long-awaited follow-on contract for the new Programming Environment and Training (PET) program was awarded this summer and will be fully implemented at the start of FY02. Dr. Leslie Perkins of the HPCMO and her PET technical advisory team have done an outstanding job of taking the successes and lessons learned from the initial five-year PET program and incorporating them as the building blocks for an expanded HPC Modernization Program-wide PET effort.

Finally, no words can adequately describe the anguish and sorrow we all feel as a result of the tragic events of September 11 in New York, Washington, D.C., and Pennsylvania. The United States Department of Defense (DoD) will play a major role in the response to this tragedy—it is my fervent hope that the past eight years of support by the DoD HPC Modernization Program will, in meaningful ways, make DoD's mission more effective, much quicker, and substantially safer for those who must go in harm's way in response to these attacks.

About the Cover:

Virtual environment built by the NAVO MSRC Visualization Center for the Concurrent Computing Laboratory for Materials Simulation at Louisiana State University. This application allows the researchers to visualize a million atom simulation of an indentor puncturing a block of gallium arsenide.

See pages 14-15 (Centerfold) for additional information.

The Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC): Delivering Science to the Warfighter

The NAVO MSRC provides Department of Defense (DoD) scientists and engineers with high performance computing (HPC) resources, including leading edge computational systems, large-scale data storage and archiving, scientific visualization resources and training, and expertise in specific computational technology areas (CTAs). These CTAs include Computational Fluid Dynamics (CFD), Climate/Weather/Ocean Modeling and Simulation (CWO), Environmental Quality Modeling and Simulation (EQM), Computational Electromagnetics and Acoustics (CEA), and Signal/Image Processing (SIP).

NAVO MSRC
Code N7
1002 Balch Boulevard
Stennis Space Center, MS 39522
1-800-993-7677 or
help@navo.hpc.mil

NAVO MSRC Navigator
www.navo.hpc.mil/navigator

NAVO MSRC Navigator is a biannual technical publication designed to inform users of the news, events, people, accomplishments, and activities of the Center. For a free subscription or to make address changes, contact NAVO MSRC at the address above.

EDITOR:
Gioia Furness Petro, petrogio@navo.hpc.mi

DESIGNERS:
Patti Geistfeld, pgeist@navo.hpc.mil
Kerry Townson, ktownson@navo.hpc.mil
Lynn Yott, lynn@navo.hpc.mil

Any opinions, conclusions, or recommendations in this publication are those of the author(s) and do not necessarily reflect those of the Navy or NAVO MSRC. All brand names and product names are trademarks or registered trademarks of their respective holders. These names are for information purposes only and do not imply endorsement by the Navy or NAVO MSRC.

Approved for Public Release
Distribution Unlimited

Contents

The Director's Corner

2 Moving Ahead at the NAVO MSRC

Feature Articles

4 A Proteomics Approach to a Malaria Vaccine

7 MetVR: Using Virtual Reality for Meteorological Visualization

10 Towards Prediction of Aircraft Spin

13 OpenMP Parallel Implementation of SWAN

Scientific Visualization

14 CCLMS Challenge Project

18 Updates from NAVO MSRC

19 Enhancing NAVO MSRC Visualization Capabilities

Programming Environment and Training

20 DoD User Group Conference 2001 Sights

22 NAVO MSRC PET Update

Navigator Tools and Tips

23 Running X Windows with Parallel Codes on HABU Under LoadLeveler

The Porthole

24 A Look Inside NAVO MSRC

Upcoming Events

27 Conference Listings

A Proteomics Approach to a Malaria Vaccine

Giri Chukkapalli, Amitava Majumdar, and Robert Sinkovits
Scientific Computing Department, San Diego Supercomputer Center, La Jolla, CA
Capt Daniel Carucci, MC, USN
Malaria Program, Naval Medical Research Center, Silver Spring, MD
John Yates and Laurence Florens
The Scripps Research Institute, La Jolla, CA

Anopheline mosquito responsible for transmission of malaria

According to the World Health Organization, malaria currently affects 300-500 million people worldwide. More than 90 percent of these cases occur in sub-Saharan Africa and are responsible for over one million deaths each year. The battle against malaria has been hampered by the emergence of drug-resistant strains of Plasmodium falciparum, the parasite responsible for the majority of malaria infections. Further complicating the use of antimalarial drugs is the fact that most cases are concentrated in the world's poorest countries, thereby imposing the additional requirement that newly developed treatments be easily affordable.

As an alternative to antimalarial drugs, work is actively being done on the development of a malaria vaccine. Due to the complicated multistage life cycle of P. falciparum (see Figure 1), this is a difficult task. Using the recently completed draft of the P. falciparum genome, researchers at the Naval Medical Research Center (NMRC) in the laboratory of Daniel Carucci are systematically approaching this problem by studying the proteins that are expressed at various stages of the parasite's development.[1,2] The study uses a new approach to the analysis of protein mixtures that proteolytically digest the proteins present at each stage of the life cycle and then analyzes the resulting peptides to determine the proteins present, as shown in Figure 2.

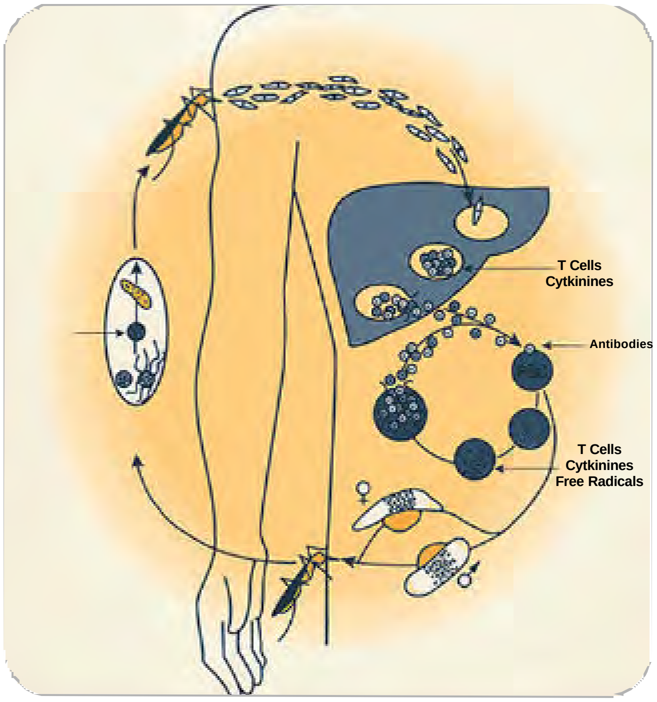


Figure 1. Life cycle of the malaria parasite P. falciparum.

Tandem Mass Spectrometry (MS/MS) is the primary experimental tool used for the protein identification. The first stage is used to isolate peptides having a selected mass to charge ratio. The isolated peptide ion is then subjected to collisionally activated dissociation, and the second stage of the MS/MS is used to measure the masses of the fragments yielding a characteristic sequence ion fingerprint as shown in Figure 3. The Sequest program,[3] developed in the laboratory of John Yates while at the University of Washington and marketed by ThermoFinnigan, is used to identify the proteins by comparing the MS/MS output to the appropriate protein or DNA database (Figure 4). It is this analysis step that is the current bottleneck in the analysis of the P. falciparum parasite.

NAVO MSRC PET Tiger Team collaboration between the San Diego Supercomputer Center (SDSC), The Scripps Research Institute, and NMRC had been initiated earlier this year to reduce or eliminate this computational bottleneck. Currently, the analysis of the P. falciparum proteome on a single-processor machine takes about 30 days of computer time. By taking a two-pronged approach—single processor optimization and code parallelization—we expect to be able to reduce the time to solution on the largest NAVO MSRC machines to less than 30 minutes. Sequest is the most widely used software for the analysis of MS/MS data and is estimated to be used by more than 500 laboratories. Improvements to Sequest should be expected to not only accelerate the malaria vaccine project, but also have the potential to impact a much broader proteomics community.

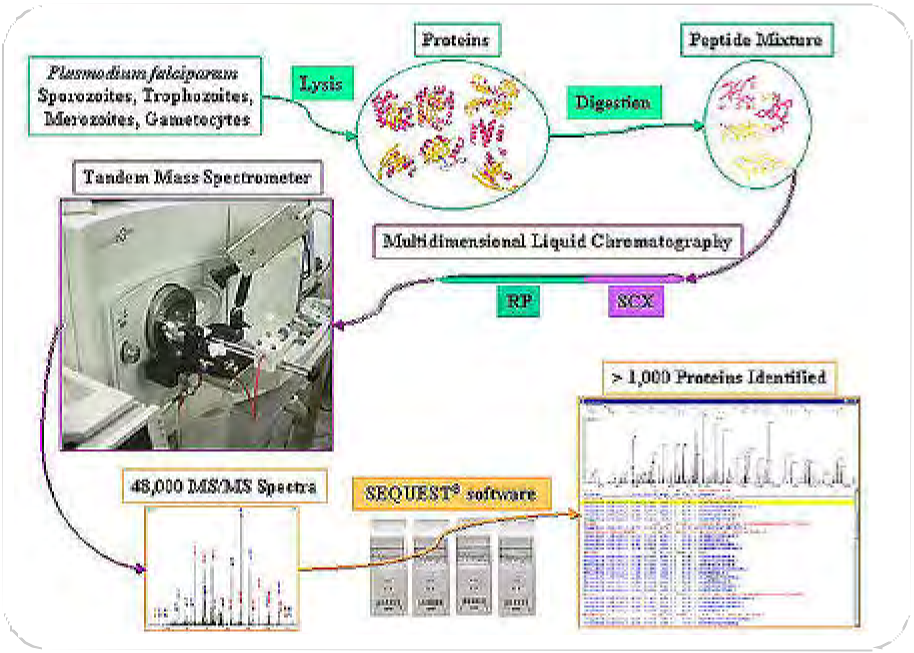


Figure 2. Experimental setup used to determine proteins expressed by P. falciparum. The Sequest program is used in the analysis of the MS/MS spectrum.

Single-Processor Optimization

The goal of single-processor optimization is to reduce the time to solution for a calculation carried out on a single processor. Since the serial code is used as the starting point for the parallel application, this also boosts the performance of the parallel version of the code. A number of general and special techniques were applied as described below.

As would be expected with a proteomics application, a significant amount of time is spent performing string operations. One of the code modifications that had a large impact on performance involved the replacement of calls to the C string library routines with logical tests on pre-computed arrays of integers. Normally, library functions give much better performance than user-developed routines, but in this case the full functionality of the library routine was not required. The C strchr () function returns the location of the first occurrence of a character (typically a residue of the peptide sequence) in a string (for example, a list of protein cleavage sites). Since the test strings are not modified after the initialization stage of the program, a great deal of computational overhead can be avoided by testing the strings for the occurrence of all characters in the amino acid alphabet (normally 20 letters for the naturally occurring amino acids, plus optionally additional characters for unknown or nonstandard amino acids) at the start of the calculation.

Another optimization that had a large impact on the code performance was the precalculation of loop-invariant quantities. (A loop invariant is an expression that appears within a loop, but whose value remains constant across iterations of the loop.) While most compilers are quite effective at recognizing loop-invariant quantities, the presence of function calls within the loop may force the compiler to make more conservative choices. These choices may be required because the functions may have the side effect of modifying the variables that are used in the expression. Other optimizations such as the elimination of redundant logical expressions, loop pealing, force reductions, and replication of key loops for special cases further reduced run time. Timing comparisons for the original and optimized versions of the code are given in Table 1.

Code Parallelization

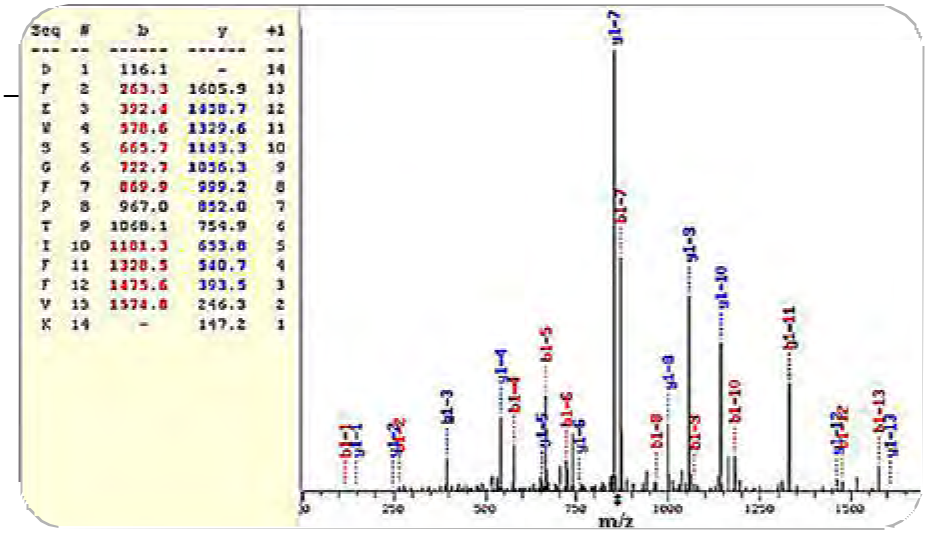
While the benefits of single-processor optimization are undeniable, the biggest potential impact on reducing the time to solution arises as a result of code parallelization. Since the analysis of the P. Falciparum proteome involves the independent processing of roughly 100,000 input files, it is expected that a parallel version of the code should be capable of linear scaling up to thousands of processors.

The optimized version of the serial Sequest code was used as the starting point for a parallel implementation developed using the Message Passing Interface (MPI). In the parallel code, input files are handed out in a round robin fashion to processors rather than dividing up the list of input files at the start of the simulation. This makes no

	Original	Optimized	Speedup
Test 1	83	45	1.84
Test 2	137	63	2.17
Test 3	202	111	1.82
Test 4	381	173	2.20
Test 5	269	139	1.93

Table 1. Performance of original and optimized scalar version of Sequest code. The five test cases stress various parts of the Sequest code and are representative of the types of calculations most frequently performed using Sequest. Test 5 is based on parameters most relevant to the malaria vaccine project. All timings were obtained on a single 375-MHz IBM Power3 processor using aggressive optimization (-O3).

Article Continues Page 6...



Seq	#	b	y	+1
D	1	116.1	-	14
F	2	263.3	1605.9	13
E	3	392.4	1458.7	12
W	4	578.6	1329.6	11
S	5	665.7	1143.3	10
G	6	722.7	1056.3	9
F	7	869.9	999.2	8
P	8	967.0	852.0	7
T	9	1068.1	754.9	6
I	10	1181.3	653.8	5
F	11	1328.5	540.7	4
F	12	1475.6	393.5	3
V	13	1574.8	246.3	2
K	14	-	147.2	1

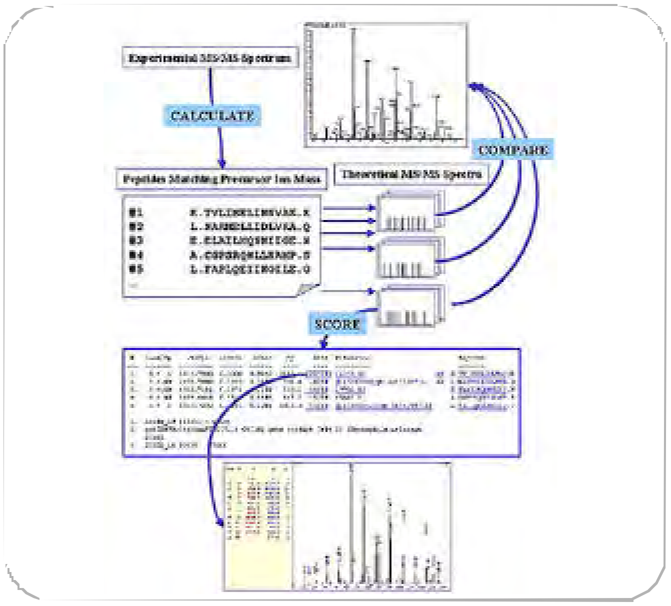
Figure 3. Typical mass spectrum originating from the second stage of MS/MS.

difference in the case where the time to process each input file is identical, but avoids load-balancing problems when there is significant variation in the processing times. Benchmarks carried out on the IBM SP at SDSC show that it takes about 25 sec to process 17 input files on 8 CPUs and about 300 sec to process 1700 input files on 64 CPUs. These tests show that speedup is nearly linear for the parallel Sequest code.

In addition to problem decomposition, attention was also focused on efficient I/O. The original version of the code read in the complete protein or genome database for each input file. For the analysis of the P. falciparum, the database is 30 MB in size, and there are on the order of 100,000 input files. The total I/O requirements for this run would be 3 TB, enough to potentially saturate the I/O network and slow down the simulation. By reading in the database just once per MPI task, the amount of I/O for this case run on 1000 processors could be reduced by two orders of magnitude. We are currently testing a scheme in which the database would is read in once by a single processor and then broadcast to all the other processors.

Impact on Research

We expect that the combination of code parallelization and single-processor optimization should lead to a reduction in time to solution from 30 days to less than 30 minutes. This dramatic improvement in code performance should accelerate the progress of the malaria researchers and allow them to easily perform new analyses of the P. falciparum genome as improved versions of the protein or genome databases become available. Since Sequest is a general-purpose tool in use by hundreds of MS/MS labs around the world, the work that we have done has the potential to dramatically impact an entire proteomics community.



Experimental MS/MS Spectrum
CALCULATE
Peptides Matching Precursor Ion Mass
Theoretical MS/MS Spectra
COMPARE
#1
#2
#3
#4
#5
SCORE

Figure 4. Flowchart of Sequest operation. The key step is the comparison of the sequence ion fingerprint to the DNA or protein database for protein identification.

Acknowledgements

The research described was funded by Naval Medical Research Center Work Units 61102AA0101BFX and 611102A0101BCX and the U.S. Army Medical Research and Materiel Command Contract (DAMD17-98-2-8005). Sequest optimization and parallelization was made possible through support by the NAVO MSRC PET program. The opinions expressed are those of the authors and do not reflect the official policy of the Department of the Navy, Department of Defense, or the U.S. Government.

References

1. Carucci, D.J., “Malaria research in the post-genomic Era” [In Process Citation], *Parasitol Today,* 2000, 16:434-8.
2. Hoffman, S.L., Rogers, W.O., Carucci, D.J., Venter, J.C., “From Genomics to Vaccines: Malaria as a Model System” [Comment], *Nat. Med.*, 1998; 4:1351-3.
3. Eng, J.K., McCormack, A.L., Yates III, J.R., *J. Am. Soc. Mass Spectrom*. 1994, 5, 976-989.

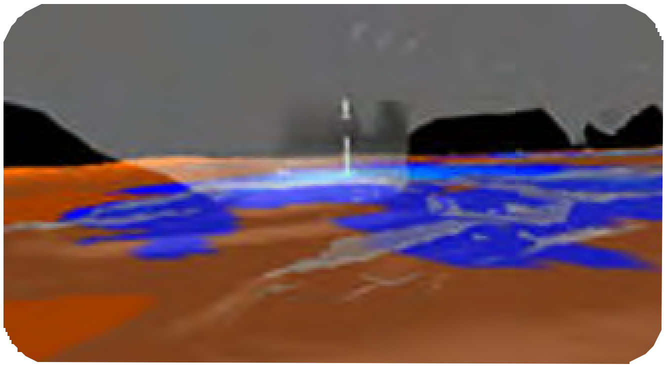
MetVR: Using Virtual Reality for Meteorological Visualization

Sean Ziegeler and Robert J. Moorhead, ERC, Mississippi State University
Paul J. Croft and Duanjun Lu, Department of Physics, Jackson State University

Traditional methods for displaying weather products are generally two-dimensional (2D). It is difficult for forecasters to get the entire picture of the atmosphere using these methods, as the atmosphere is three-dimensional (3D).

The problems apparent in 2D with comparing and correlating multiple layers are overcome by adding a dimension. However, simply using a 3D approach is not enough. Visualization in 2D has a capacity for analysis of small-scale but important features. This capacity is lost when transitioning to 3D.

We propose that 3D's advantages can be incorporated with 2D's small-scale analysis by using an immersive virtual environment, also known as virtual reality. Currently, we are developing such an application that we call MetVR.


A view from the VR Juggler simulator of a simulation of Hurricane Dennis. Rainfall is shown on the terrain. The isosurfaces show potential energy.

The Problem

Traditional 2D Visualization

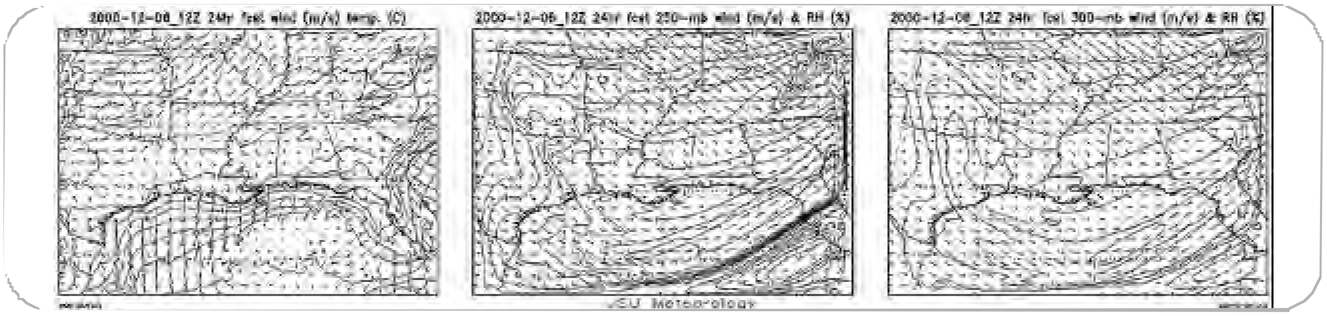
Traditional 2D visualization of multilayer, time-series data makes it difficult to see all layers and time steps in a single image. Animation is the most obvious solution to visualizing the time-series aspect of the data. However, we still have the problem that inherently 3D data, which captures the essence of atmospheric behavior, is being reduced to 2D.

The figure at the bottom of the page illustrates how multiple layers are usually displayed in 2D. To be able to analyze phenomena that span several layers, one would have to imagine each image superimposed over the other. This could be difficult to imagine for many layers and would be further complicated with time-series data.

Traditional 3D Visualization

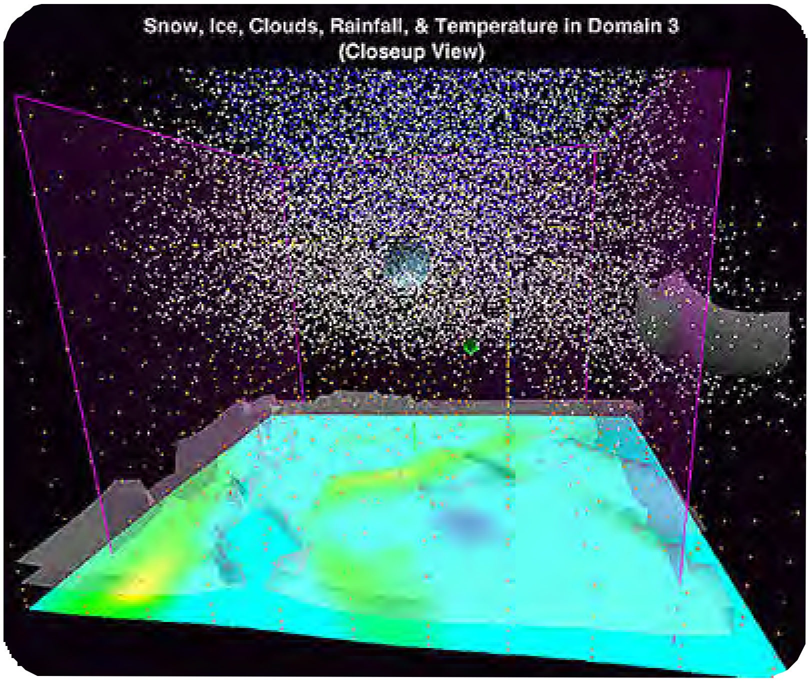
Using 3D visualization and animation, we can easily view multilayer, time-series data sets in a unified manner. Interactive 3D visualization exploits the mind's ability to grasp complex environments. For example, instability of atmosphere is crucial for the development of storms. The stability of the atmosphere is dependent upon vertical structure. Three-dimensional visualization techniques can provide a visual understanding of vertical structure in any part of the domain.

Traditional "desktop 3D" visualization loses an important feature of 2D: the ability to closely examine small-scale, but important, phenomena within the data set. This is an


2000-12-08_12Z 24hr fcst wind (m/s) temp. (C)
2000-12-08_12Z 24hr fcst 250-mb wind (m/s) & RH (%)
2000-12-08_12Z 24hr fcst 300-mb wind (m/s) & RH (%)
JSU Meteorology

Three layers from an MM5 model output data set: surface layer, 250 mb, and 300 mb. Each displays wind vectors with a second variable shown as isolines.

Article Continues Page 8...



A view from the VR Juggler simulator. The particles indicate snow (white) & ice (blue). Rainfall is shown on the terrain, and clouds as white isosurfaces.

easy task in 2D. We simply zoom into the area of an image in which we are interested. If we were to zoom into the center of an image, for example, the surrounding data would not occlude our view of that area. Also, if we wanted to compare that area to another area, zooming out and panning around the image is easy. However, with desktop 3D, we are generally viewing a volume of data from outside the volume. Although it is possible to implement 3D navigation such that the user is placed inside the data, it is still difficult for the user to analyze the data as if inside the data set.[1]

The Solution

Our solution is to implement an immersive Virtual Environment (VE) that we call MetVR. Besides our primary goal, there are additional advantages to applying VE technology to meteorological visualization:

- ☞ Improved human-computer interaction[2]
- ☞ More degrees-of-freedom for navigating the data [2]
- ☞ A "visual paradigm" more closely related to the real world[3]
- ☞ Stereoscopy for improved depth perception[2]
- ☞ Wider field-of-view[2]

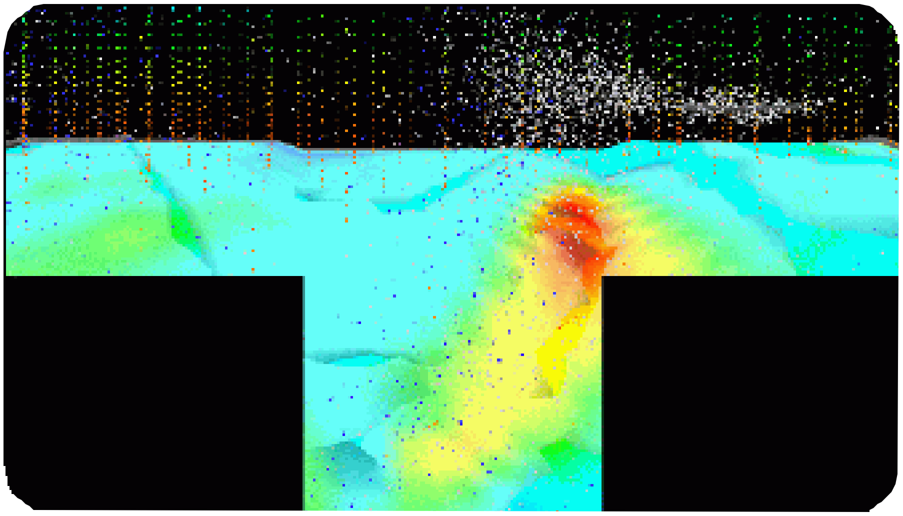
The application itself is built upon the VR Juggler libraries with OpenGL as the graphics Application Programming Interface. We chose VR Juggler because it is open source and more portable than other competing VE libraries. VR Juggler is also written in C++, and we are using an object-oriented design strategy.

Improvements upon Previous VEs

Two other VE systems, Cave5D and vGeo, can cache data on disk, but MetVR is optimized for multivariable, time-series data sets. Thus it streams in the data as needed. MetVR is faster and more flexible. We also feel that the visuals are much richer. Finally, MetVR is free and user extendible. To use vGeo, one must purchase both the CaveLibs and vGeo itself.

The Results

With MetVR at a usable phase in its development, we could compare its effectiveness with traditional 2D and 3D applications. The evaluation involved meteorologist testing the application in the Computer-Assisted Virtual Environment (CAVE) to determine its effectiveness and usefulness.



This view is a simulation of the view inside the CAVE. Once again, the variables are the same as in previous figures. The left, center, and right images show the left, center, and right walls of the CAVE, respectively. The bottom image is the floor of the CAVE.

The evaluators had considerable experience with common 2D and 3D visualization software. We allowed them to explore the data set and noted any comments and questions. When the users were finished evaluating, we asked a series of evaluation questions.

Small-Scale Analysis

With their experience in traditional 2D and 3D visualization, the users did indeed find that small-scale features were easier to see and analyze in the VE. They could identify regions of interest from a distance, and then simply navigate to the area for further inspection. One specific advantage that a user noted was that the VE shows the tops and bottoms of cloud layers when he navigated to a specific region. This is an important feature in storm analysis since the location and shape of clouds indicate if a storm is or will be present (i.e., low altitude, thin clouds usually mean a storm).

Additional Advantages

The users found the navigation more natural than with a desktop system. They could simply point to the location with a wand and press a button to fly there. The users thought that they could easily complete common analysis tasks using the given interface.

As expected, the users found that immersion in the meteorological data in the VE was an excellent metaphor for actual weather. They found the stereoscopy to be very useful in discriminating between near and distant features—a much more difficult task using desktop 3D. Also, the wider field-of-view allowed them to see more data simultaneously.

Conclusions and Future Work

Using an immersive VE, we were able to combine 3D's greater ability for multilayer analysis with 2D's capacity for small-scale analysis. We could qualitatively identify how our immersive VE achieved this goal. We believe that such an advantage, in addition to the other listed advantages in using immersion, could lead to improved interpretation of meteorological data. Such an improvement would bring about better weather-related products and increased capability in decisionmaking. Our conclusion from this study is that a VE could be properly tailored for atmospheric analysis, and further development of such a VE would be valuable.

For future work, we plan to do a comprehensive study comparing the abilities to see small-scale features so that we can quantify the improvements gained by immersion. We also plan to implement improvements to MetVR based on users' recommendations. To help indicate the user's location within the dataset, we will add a "you-are-here" overview map with an icon indicating the user's location inside the dataset.

For the particle density fields, spheres instead of points may be desirable. However, this is unrealistic, considering the rendering time necessary to render the required number of spheres. Instead, we propose using a single billboarded polygon and texture mapping the polygon with the image of a sphere. Additional features will include new tools and enhancements to tools.

References

1. Sarathy, S., Shujace, K., Cannon, K., “Visualization of Large Complex Datasets Using Virtual Reality,” IEEE Information Technology: Coding and Computing, 522-526, March 2000.
2. Youngblut, C., Johnson, R., Nash, S., Wienclaw, R., Will, C., “Review of Virtual Environment Interface Technology,” IDA Paper P-3186, Institute for Defense Analyses, 1801 N. Beauregard Street, Alexandria, VA 22311-1772, March 1996.
3. Cruz-Neira, C., “Applied Virtual Reality,” Course 14 Notes, In Proceedings of SIGGRAPH '98 (Orlando, FL, July 1988), ACM Press.

Acknowledgements

This work was done as part of the High Performance Visualization Center Initiative funded by the Department of Defense High Performance Computing Modernization Program and administered by the NAVO MSRC.

Towards Prediction of Aircraft Spin

Kyle D. Squires, Arizona State University
James R. Forsythe, United States Air Force Academy
Kenneth E. Wurtzler, William Z. Strang, Robert F. Tomaro, Cobalt Solutions, LLC
Philippe R. Spalart, Boeing Commercial Airplanes

Most of the flow fields encountered in Department of Defense applications occur within and around complex devices and at speeds for which the underlying state of the fluid motion is turbulent. While Computational Fluid Dynamics (CFD) is gaining increased prominence as a useful approach to analyze and ultimately design configurations, efficient and accurate solutions require substantial effort and expertise in several areas. Geometry description and grid generation, numerical solution of the Navier-Stokes equations, and efficient postprocessing are all key elements.

While advances have taken place in areas such as grid generation and fast algorithms for solution of systems of equations, CFD has remained limited as a reliable tool for prediction of inherently unsteady flows at flight Reynolds numbers. Current engineering approaches to prediction of unsteady flows are based on solution of the Reynolds-averaged Navier-Stokes (RANS) equations. The turbulence models employed in RANS methods necessarily model the entire spectrum of turbulent motions. While often adequate in steady flows with no regions of reversed flow, or possibly exhibiting shallow separations, it appears inevitable that RANS turbulence models are unable to accurately predict phenomena dominating flows characterized by massive separations. Unsteady, massively separated flows are characterized by geometry-dependent and three-dimensional (3D) turbulent eddies. These eddies, arguably, are what defeat RANS turbulence models of any complexity.

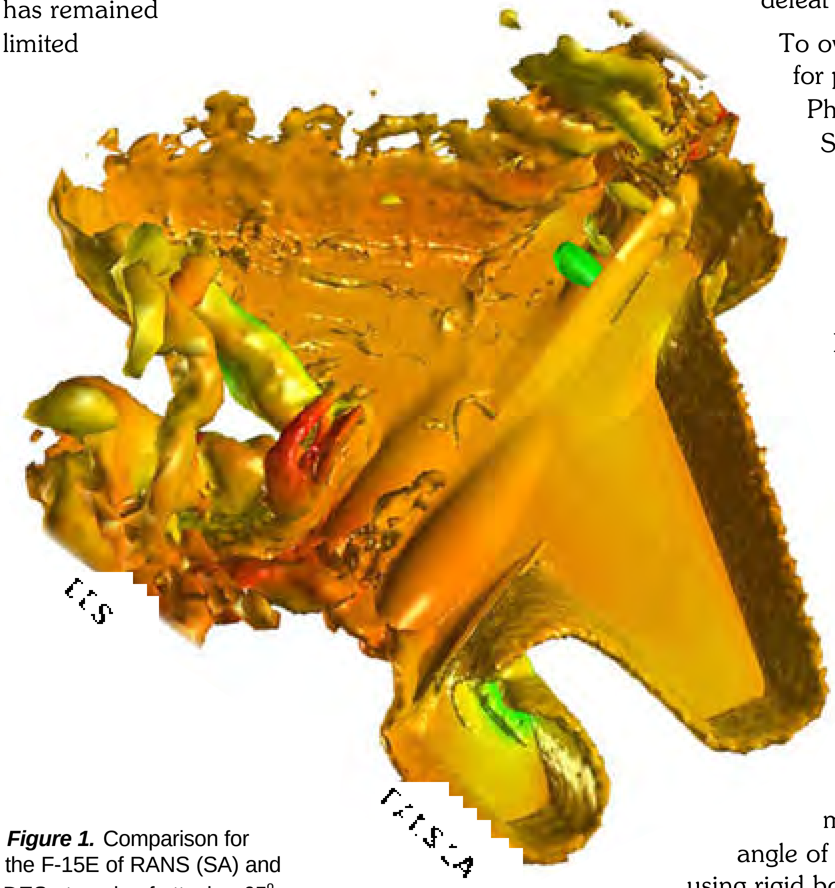


Figure 1. Comparison for the F-15E of RANS (SA) and DES at angle of attack = 65°. Isosurface of vorticity colored by pressure.

To overcome the deficiencies of RANS models for predicting massively separated flows, Dr. Philippe Spalart proposed Detached-Eddy Simulation (DES) with the objective of developing a numerically feasible and accurate approach combining the most favorable elements of RANS models and Large Eddy Simulation (LES). The primary advantage of DES is that it can be applied at high Reynolds numbers, as can Reynolds-averaged techniques, but also resolves geometry-dependent, unsteady 3D turbulent motions as in LES.

Computations were performed of the flow over an F-15E at angle of attack of 65° and zero sideslip. Boeing provided the authors with a stability and control database for the F-15E that was developed from a comprehensive spin-flight test program. Two stable spin conditions were detailed, including data for symmetric and asymmetric fuel loads. The aircraft with symmetric loading maintains a stable spin at 65° angle of attack. Prior to computing the actual spin using rigid body moving grids, the performance of the computational model was investigated at the same fixed

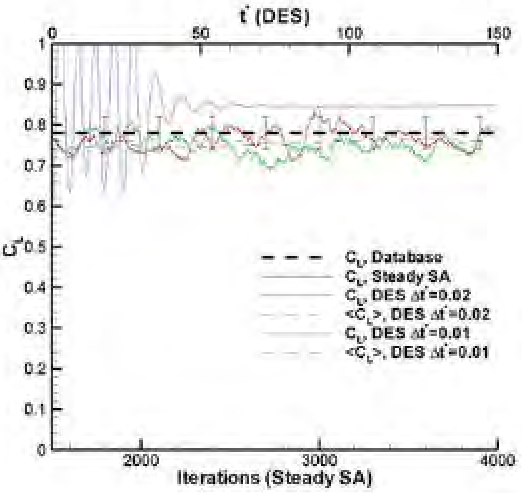


Figure 2. Time and iteration histories of C_L on the F-15E.

angle of attack as for the stable spins. All computations were made matching the flight test conditions at a Mach number of 0.3 and standard day 30,000 feet. This resulted in a chord-based Reynolds number of 13.6 million.

Simulations were performed at the NAVO MSRC Cray T3E. The F-15E runs (5.9 million cells) were computed using up to 512 processors and required 1 to 2 days for complete turnaround. For the DES runs, the F-15E grid was mirrored about the symmetry plane, resulting in 11.8 million cells. These runs required approximately four days for turnaround on 256 processors and were computed using a non-dimensional time-step (using the chord and freestream velocity) of 0.01.

Side by side comparisons of DES and RANS predictions across the symmetry plane are shown in Figure 1. The isosurfaces of vorticity illustrate the capability of DES in "LES mode" resolving the unsteady, geometry-dependent flow features. Force histories of C_L, C_D, and C_M for the two simulation techniques are next compared to the Boeing database in Figures 2, 3, and 4. The largest discrepancy for the RANS result is in the moment coefficient, which is overpredicted (in a negative sense) by 12%. The lift and drag coefficients are overpredicted by 9% and 7%, respectively. DES predictions of these quantities are all within 4% for lift and drag and 7% for moment for both time steps. The expected accuracy of the Boeing database for this angle of attack is anticipated to be around 5%.

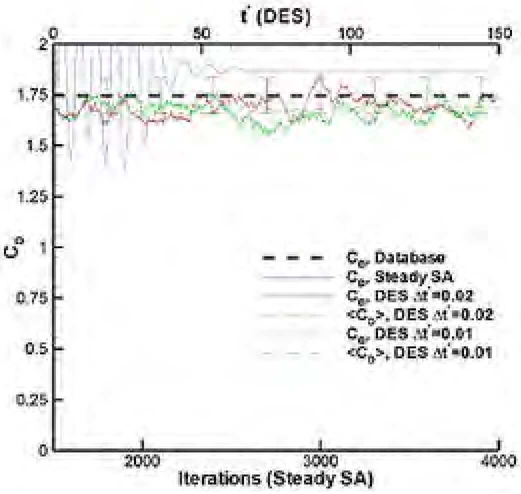


Figure 3. Time and iteration histories of C_D on the F-15E.

To examine the source of the differences between the RANS and DES results, the pressure coefficient from the (steady-state) RANS is compared to the time average of the DES predictions in Figure 5. As evident from the figure, DES yields a relatively flat profile in Cp that is the norm on a wing with separation, while the RANS predicts a more varied pressure distribution. This lends confidence to the accuracy of the DES results in the absence of experimental pressure profiles. In the current study, the computational cost of the time-accurate calculations is roughly an order of magnitude greater than a steady-state run. The unsteady

Article Continues Page 12...

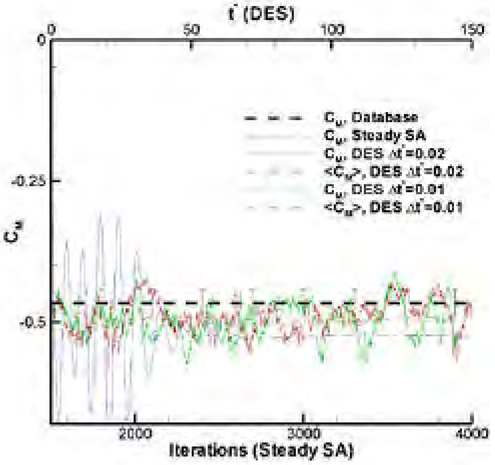


Figure 4. Time and iteration histories of C_M on the F-15E.

RANS calculations carry the same cost as DES if the grid and time step are the same and if the sampling period for calculation of averaged quantities is fixed. The current calculations, although not yet demonstrating grid-convergent solutions, suggest that DES exhibits improved predictions on a grid originally designed for RANS. Although a factor of ten increase in cost may seem large, if Moore's law continues to hold, this factor of ten will be surpassed in less than three and a half years. Thus, facilities that can currently support steady RANS calculations on full aircraft should be able to accommodate DES runs on at least coarse grids.

For the F-15E, the present DES calculations are probably the first applications of a turbulence-resolving technique to full aircraft at flight Reynolds numbers in which turbulent boundary layers on the vehicle were represented without recourse to wall functions (i.e., with grid spacings within one viscous unit at the wall). This technique is being applied to drag analysis of semitruck tractor-trailers. The sharp edges of the truck geometry creates a definite separation line, producing large-scale unsteadiness. The DES solution captures this region while the RANS solution dampens the unsteadiness (Figure 6).

Combining the time for grid generation and solution, a DES calculation could be made within two weeks, given the performance of the HPC machines. In six years, if Moore's law continues, HPC will see its capacity increase by approximately 16 times, allowing rapid solutions for grids nearing 200 million cells. As grids are made more dense, the fidelity of the DES solutions will improve for the current simulation (static aircraft). Alternatively, increased computing capacity will enable the incorporation of additional physical effects into the modeling. The ability of DES to predict unsteady flow phenomenon at high Reynolds numbers opens the possibility to handle more complex problems that require an unsteady solution such as aerolasticity and aeroacoustics.

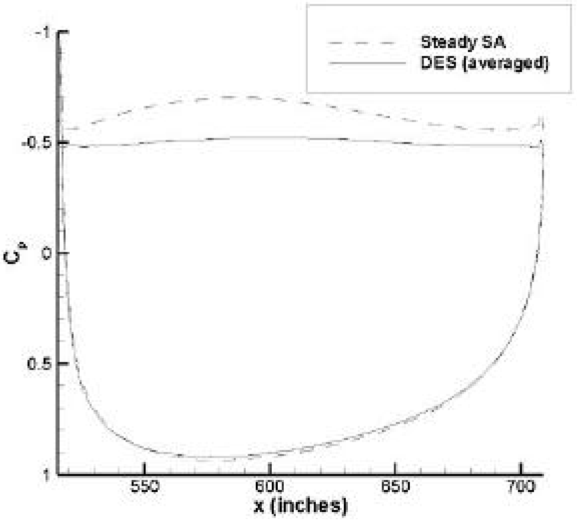


Figure 5. Comparison of the pressure coefficient from the (steady-state) RANS case and the time average of the DES case.

Acknowledgements

This work is partially supported by AFOSR Grant F49620-00-1-0050 (Program Manager: Dr. Tom Beutner). The authors are also grateful for the assistance of Dr. Glen Peters, Dr. Ken Walck, and Dr. Walt Labozzetta of Boeing Military, who provided the stability and control database and the F-15E geometry. Mr. Matthieu Chapelet and Cadet Kory Klismith assisted with grid generation and postprocessing. Cadet Warren Carroll and Cadet Alan VantLand aided in postprocessing the F-15E results.

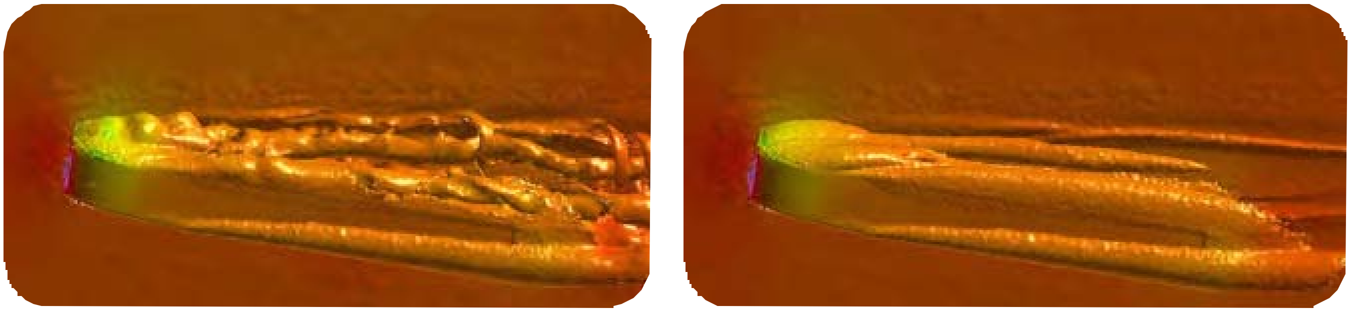


Figure 6. Comparison of DES method (left image) and standard RANS method (right image) applied to sideslip flow past a tractor-trailer truck model.

OpenMP Parallel Implementation of SWAN

John Cazes, NAVO MSRC Programming Environment and Training
Tim Campbell, NAVO MSRC Programming Environment and Training
Erick Rogers, Oceanography Division, Naval Research Laboratory

This article describes a recent collaboration between NAVO MSRC Programming Environment and Training (PET) analysts and Naval Research Lavoratory (NRL) researchers to realize a parallel implementation of an important wave modeling code called Simulating WAves Nearshore (SWAN). A similar successful collaboration involving a 3-D finite element circulation model was described in the Spring 2001 NAVO MSRC *Navigator*. The goal of this project was to modernize the memory management used in SWAN and then produce a parallel code that involved minimal changes in the algorithms used by the model and no changes to the user interface or configuration files. With these goals in mind, we chose to port the model using the OpenMP multithreading directives. Since OpenMP directives are seen as comments by non-OpenMP compilers, with careful planning, this also allowed us to produce a code that was equally suitable for serial architectures as well shared memory architectures. Consequently, the same code can be maintained and distributed to the SWAN community at large regardless of their choice of architecture.

OpenMP is a parallel programming model for shared memory and distributed shared memory multiprocessors that works with either standard Fortran or C/C++. OpenMP consists of compiler directives, which take the form of source code comments, that describe the parallelism in the source code. A supporting library of subroutines is also available to applications. The OpenMP specification and related material can be found at the OpenMP web site: http://www.openmp.org. Online training in OpenMP is part of the NAVO MSRC PET distance learning (http://www.navo.hpc.mil/pet/Video), and links to other online training material can be found at the NAVO PET Parallel Computing Portal (http://www.navo.hpc.mil/Tools/pcomp.html).

In Fortran, OpenMP compiler directives are structured as comments, written as C$OMP or !$OMP. This allows for the design of OpenMP codes that can be compiled for serial as well as parallel architectures with no code changes. An OpenMP program begins as a single process, called the master thread. When a parallel region, which is preceded by either a PARALLEL or PARALLEL DO construct, is encountered, threads are forked, on separate processors if available, to execute the statements enclosed within the parallel construct. At the end of the parallel region, the threads synchronize, and only the master thread remains to continue execution of the program. The PARALLEL DO construct is commonly used to parallelize computationally intensive loops within a program.

Model Description

SWAN[1] is a serially coded wind-wave model (WAM) designed to overcome the traditional difficulties of applying predecessor models (e.g., WAM,[2] presently used at NAVOCEANO) at relatively small scales (e.g., less than 500-m grid spacing). Specifically, it uses a semi-implicit, unconditionally stable numerical scheme for geographic propagation, so that high geographic resolution does not dictate an excessively small (and therefore expensive) time step. In addition to a nonstationary mode, similar to WAM, SWAN can be used in a stationary mode, similar to the STationary WAVE (STWAVE) model used at NAVOCEANO. SWAN includes a more complete description of the physics than does STWAVE, which results in more computation and longer turnaround time. This, coupled with the lack of a parallel version, represents a significant obstacle in model development and transitioning SWAN from a research code to an operational code.

The SWAN code consists of approximately 30,000 logical lines of code with over 200 subroutines written mostly in Fortran 77 with a small number of Fortran 90 constructs added in recent years. The model solves the spectral action balance equation using finite difference techniques in two geographic dimensions (X and Y), in spectral space (frequency and direction), and time. In geographic space, an upwind finite difference is applied, resulting in the state at each grid point being dependent only on the state in the upwave (as defined by the direction of propagation) grid points. This permits the spectral space to be decomposed into four directional quadrants. The whole solution process involves iteratively computing a sequence of four forward-marching sweeps, as illustrated in Figure 1, across the geographical grid, with each sweep utilizing only boundary

Article Continues Page 16...

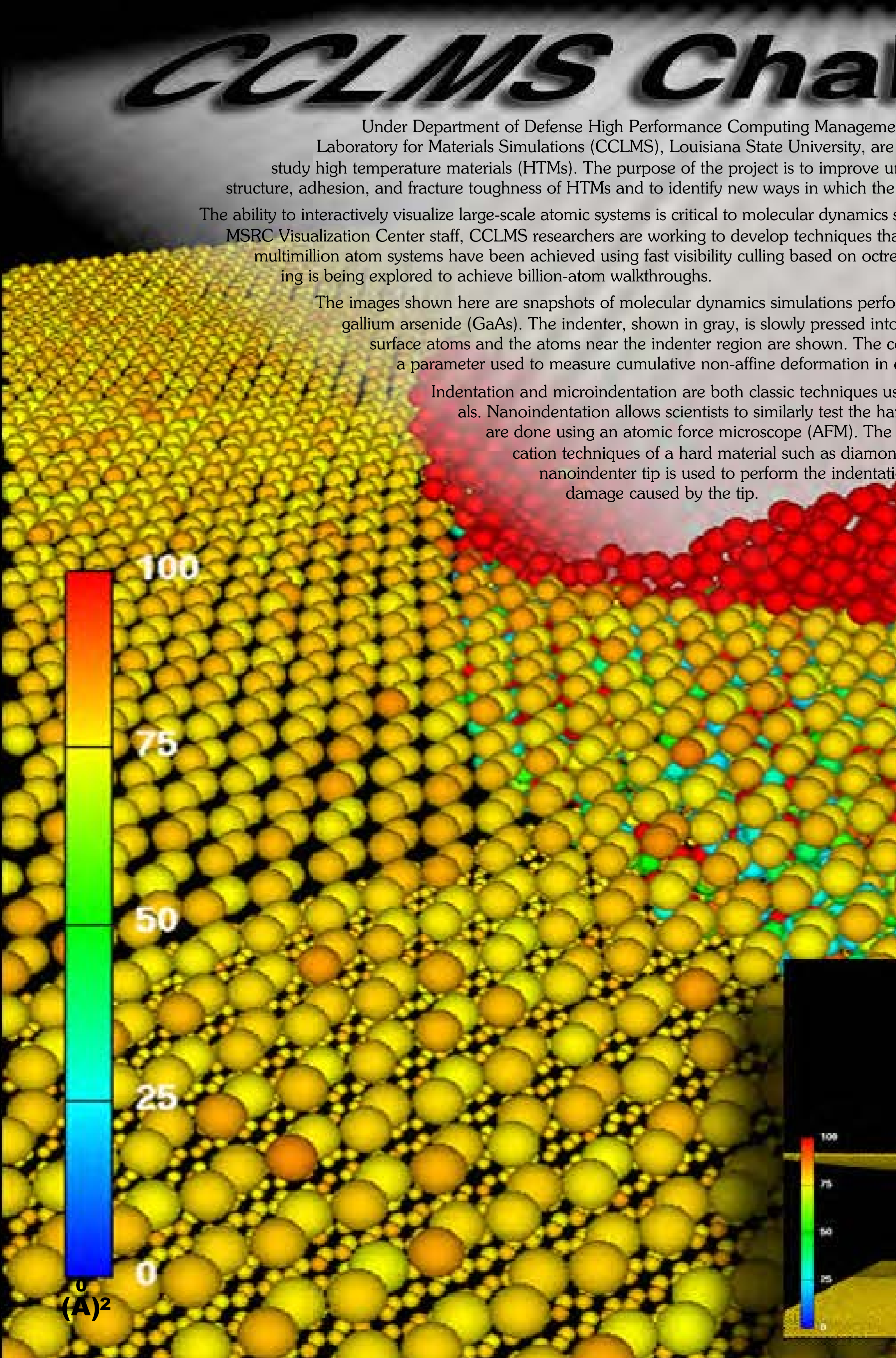
CCLMS Chal

Under Department of Defense High Performance Computing Manageme
Laboratory for Materials Simulations (CCLMS), Louisiana State University, are
study high temperature materials (HTMs). The purpose of the project is to improve u
structure, adhesion, and fracture toughness of HTMs and to identify new ways in which the

The ability to interactively visualize large-scale atomic systems is critical to molecular dynamics s
MSRC Visualization Center staff, CCLMS researchers are working to develop techniques tha
multimillion atom systems have been achieved using fast visibility culling based on octre
ing is being explored to achieve billion-atom walkthroughs.

The images shown here are snapshots of molecular dynamics simulations perfo
gallium arsenide (GaAs). The indenter, shown in gray, is slowly pressed into
surface atoms and the atoms near the indenter region are shown. The c
a parameter used to measure cumulative non-affine deformation in

Indentation and microindentation are both classic techniques us
als. Nanoindentation allows scientists to similarly test the ha
are done using an atomic force microscope (AFM). The
cation techniques of a hard material such as diamon
nanoindenter tip is used to perform the indentati
damage caused by the tip.



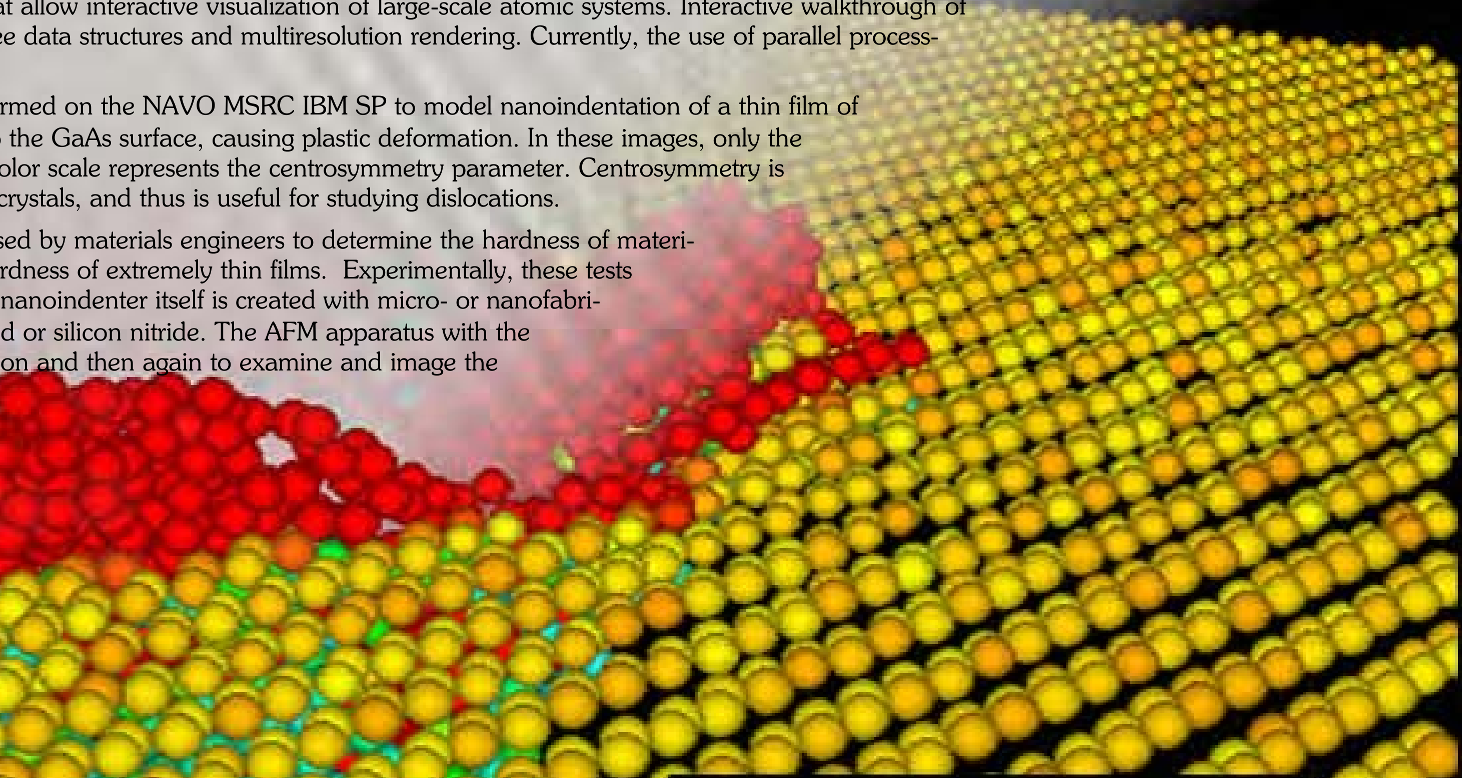
llenge Project

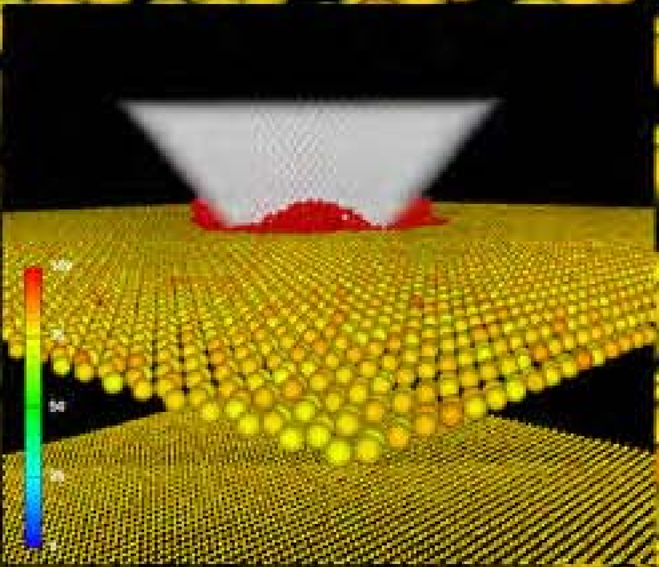
nt Office (HPCMO) support researchers at the Concurrent Computing applying expertise in materials simulations and high performance computing to nderstanding of how atomic level processes determine macroscopic properties such as performance of HTMs can be improved under extreme operating conditions.

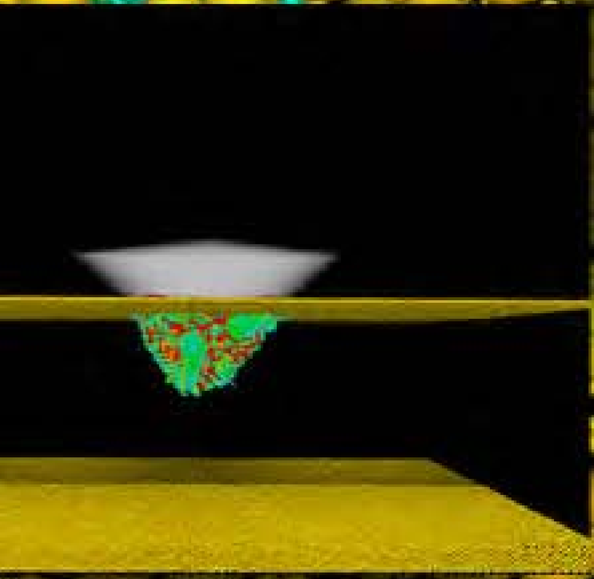
simulations and understanding the atomic processes that occur. Together with the NAVO at allow interactive visualization of large-scale atomic systems. Interactive walkthrough of e data structures and multiresolution rendering. Currently, the use of parallel process-

rmed on the NAVO MSRC IBM SP to model nanoindentation of a thin film of the GaAs surface, causing plastic deformation. In these images, only the olor scale represents the centrosymmetry parameter. Centrosymmetry is crystals, and thus is useful for studying dislocations.

sed by materials engineers to determine the hardness of materi- rdness of extremely thin films. Experimentally, these tests nanoindenter itself is created with micro- or nanofabri- d or silicon nitride. The AFM apparatus with the on and then again to examine and image the







Top. View from above that shows the indenter and the ejected material on the surface of the GaAs film. The structure of the ejected material is consistent with that of amorphous GaAs, indicating that a phase transformation, solid-state amorphization, is occurring underneath the indenter during the indentation.

Left. View from the side of the film that shows the indenter surrounded by the damaged region of the film. The bulk of the plastic deformation occurs around the indenter. However, in this picture we see plastic deformation beginning to appear some distance from the film-indenter contact region (lower center).

Article Continued From Page 13...

conditions or previously calculated points in the upwind directions. This results in a semiimplicit method that is inherently stable.

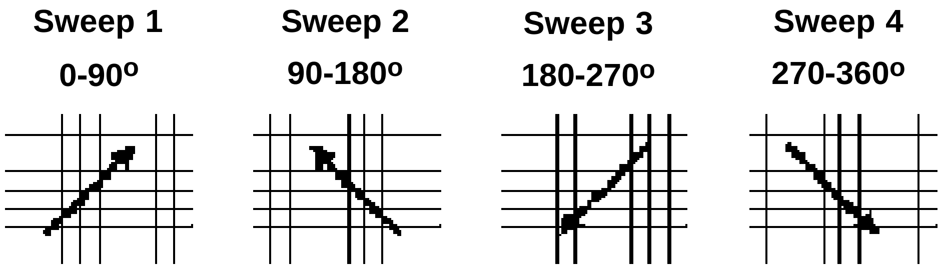


Figure 1. Schematic of geographic sweeping technique used in SWAN.

Approach

Due to the data dependencies in the sweeping technique employed, a simple domain decomposition of the spatial grid is not possible without completely altering the numerical algorithm. Instead, we opted for a shared memory pipelined parallel approach using OpenMP. This approach represents a non-traditional, i.e., not loop-level, way of using OpenMP to port an application to shared memory platforms. This allowed us to introduce parallelism at a fairly high level, requiring changes in only a few subroutines and leaving the large majority of the code untouched.

Since the numerical technique uses an upwind method that is dependent on the sweep direction, each spatial grid point has at most a data dependency in two directions. For example, in the first sweep, execution proceeds in the positive X direction then the positive Y direction (see Figure 1). Therefore, the data dependencies for each grid cell lie in the negative X and negative Y directions. By decomposing along the Y direction, each thread can begin execution as its data dependency in the negative Y direction is satisfied. The dependency in the negative X direction will already have been satisfied because the sweep in the X direction is serial.

The changes required in the SWAN code to implement the pipelined parallel approach for the sweeping loops are straightforward as illustrated with the following code:

```
!$OMP PARALLEL DO SCHEDULE(STATIC, 1)
      DO IY = IY1, IY2, IDY
      DO IX = IX1, IX2, IDX
           DO WHILE( LLOCK(IX,IY-IDY) )
           END DO
           CALL SWOMPU (IX, IY, ...)
           LLOCK(IX,IY) = .FALSE.
      ENDDO
      ENDDO
```

The OpenMP PARALLEL DO directive distributes the Y loop among the available threads. The SCHEDULE clause directs each thread to execute specific iterations of the Y loop with a chunk size of 1. For example, if the loop runs from 1 to 9 and there are 3 threads, thread 1 executes IY = 1,4,7, thread 2 executes IY = 2, 5, 8, and thread 3 executes IY = 3, 6, 9. The STATIC keyword informs the compiler that the IY values assigned to each thread will not change during the execution of the PARALLEL DO. The LLOCK array, which is reset prior to beginning each sweep, is a logical array that ensures that each thread cannot proceed to the next grid point until that grid point's data dependency in the Y direction has been satisfied. After the state at a grid point (IX,IY) is computed by the call to the subroutine SWOMPU, the thread processing that grid point signals it is finished by setting LLOCK(IX,IY) to FALSE.

The code illustration given above appears very simple, and may in fact be somewhat deceiving. What is not shown is the large number of OpenMP directives used to describe the nature of the arrays passed to the SWOMPU subroutine and arrays accessed through COMMON blocks. Each of these data structures must be declared as shared (all threads access the same data) or private (each thread has its own private copy of the data). Data-dependency analysis of this sort for a code as large and complex as SWAN is not trivial.

Results

We verified the correctness of the OpenMP version of SWAN with four different test cases that exercised a number of different options in the model such as time evolution, external currents, and wave refraction. Due to the minimal changes in the SWAN code and the absence of changes to the numerical algorithm, we were able to achieve bit-for-bit reproducibility of the parallel code with the results from the original serial version of SWAN. Results have been verified on the NAVO MSRC Sun E10000, IBM SP, and Origin 2000. Unfortunately, we were unable to execute the code on the Cray SV1 due to the fact that the Cray Fortran 90 compiler does not yet fully support OpenMP.

The speedup of a parallel program on P processors is defined as the single processor execution time divided by the execution time on P processors. If we assume that the amount of computational work at each geographic grid is exactly the same and that there is zero overhead due to thread management, then we can express the ideal speedup, S, for a single sweep as

$$S = (XY)/(X*CEILING(Y/P) + (Y-1)\%P) \qquad (1)$$

where X is the number of grid points in the X direction, Y is the number of grid points in the Y direction, and P is the number of processors.

In Figure 2, we present the results from one test case on a 48x48 spatial grid that includes external currents. The dashed line is the ideal speedup for a single sweep, as expressed in Eq. 1. The jogs in the ideal speedup correspond to points where P is a divisor of Y. The solid line is the actual speedup per iteration of the OpenMP version of SWAN as measured on the NAVO MSRC Sun E1000. We see that for P less than 16, the OpenMP version of SWAN exhibits a jog pattern similar to the ideal and sustains a speedup of at least 80% of the ideal. By 24 processors the speedup saturates. A detailed look at Eq. 1 and the ideal plotted in Figure 2 shows that the speedup gained by increasing the number of processors past Y/2 is very small. Due to the large amount of work per grid point, this test case gave the best performance results. Less computationally demanding test cases showed lower performance. Further analysis is required to determine the sources of overhead and performance bottlenecks.

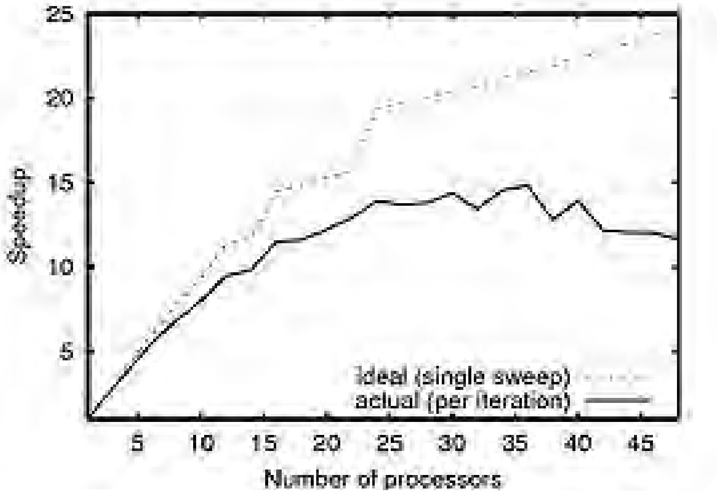


Figure 2. Speedup of test case with external currents on a 48x48 spatial grid. The dashed line is the ideal speedup for a single sweep, and the solid line is the measured speedup. These results were produced on the NAVO MSRC 64 processor Sun E10000.

Conclusion

Most descriptions of OpenMP implementations focus on loop-level parallelization, an approach that we believe would have not only been unduly difficult but also unsuccessful with a code the size and complexity of SWAN. A loop-level approach in SWAN would involve applying OpenMP to the large number of loops that form the various matrix solvers that are invoked for each spatial grid point. By taking into account the data dependencies in the solution procedure, we were able to introduce parallelism at a fairly high level using a small number of OpenMP constructs with minimal changes to the legacy code. This allows other SWAN developers to easily add more modules to enhance the physical model or improve the numerical techniques without affecting the parallelization. The improvements gained from this project may even be instrumental in SWAN becoming approved as an operational code.

References

1. Booij, N., R.C. Ris, and L. H. Holthuijsen, 1999, “A Third Generation Wave Model for Coastal Regions, 1, Model Description and Validation,” *J. Geogr. Res.*, 104, 7649-7666.

2. WAMDI Group, 1988, “The WAM Model - A Third Generation Ocean Wave Prediction Model,” *J. Phys. Oceanogr.* 18, 1775-1810.

Updates from NAVO MSRC

NAVO MSRC Remote Rendering Initiative

Pete Gruzinskas, NAVO MSRC Visualization Center

Christine E. Cuicchi, NAVO MSRC Computational Science and Applications Lead

As computing capabilities within the HPCMP grow, so does the size of the datasets that users wish to visualize. Specialized sci-vis servers and their capabilities are available at selected Shared Resource Centers (SRCs) for use primarily by collocated users, but are not generally available to the overwhelming majority of users (80% at the NAVO MSRC) who are remote to the SRCs where their large datasets may be stored. A potentially viable approach for providing optimal sci-vis application support to remote users is a technique called "remote rendering." With remote rendering, a user would remotely log in to an SRC Onyx2-class sci-vis server system, start an application on that system which accesses locally available/generated SRC data, and securely export and control the rendered/animated display stream to a local client display system such as those found on many users' desktops. This technique may dramatically increase the availability and utility of centralized sci-vis capabilities available at the larger SRCs by providing application transparent access to these costly resources over high-speed (100 Mb/sec or better) TCP/IP-based network connections such as DREN. "Application transparent" simply means that the remote rendering software runs independently from any visualization application and vice versa.

The NAVO MSRC is actively working with its PET and High Performance Visualization Center Initiative (HPVCI) university partners on remote rendering initiatives. Additionally, the NAVO MSRC has been engaged in active dialogue with other SRCs (TARDEC, ARL, and NRL-DC) to explore opportunities for collaborative visualization testbeds throughout the SRCs and to remote user communities from these central locations. The remote rendering software packages VizServer (SGI) and Exceed 3D (Hummingbird) are being evaluated across various combinations of desktop platforms and network connection speeds. This initial testbed effort should be viewed as the first step toward a long-term goal to facilitate the establishment of a remote rendering capability throughout the HPCMP.

For more information on the remote rendering software, see:

- ☞ SGI VizServer - http://www.sgi.com/software/vizserver/overview.html
- ☞ Hummingbird Exceed 3D - http://www.hummingbird.com/products/nc/exceed/3dfeatures.html

Signal and Image Processing & Climate, Weather, and Ocean Modeling Forums

Dr. Robert Melnik, CTA Coordinator

Brian Tabor, Training Coordinator

The NAVO and ARL MSRC PET programs have been co-sponsoring forums in Signal and Image Processing (SIP) annually since 1998. This year they co-sponsored and organized the fourth annual SIP Forum (SIP2001). NAVO PET also organized the first forum in Climate, Weather, and Ocean Modeling (CWO2001). The overall objective of the forums is to bring together select researchers with diverse expertise to share their views and experience in the role of High Performance Computing in each of these Computational Technology Areas (CTA's). The forums also provided an opportunity to review the status of CHSSI programs in each of the CTA's.

SIP2001

SIP2001, which was co-chaired by Dr. Rich Linderman, US Air Force Rome Laboratory and CTA lead for SIP, and Dr. Keith Bromley, SPAWAR Systems Center, San Diego, was held May 15-16 in Biloxi, Mississippi, with NAVO PET serving as the local host. The specific objectives of CWO2001 were to (1) bring together experts in the SIP community to share their experience in the application of HPC to SIP, (2) identify critical problems in SIP that can benefit from advanced HPC technology, (3) identify existing and emerging HPC technologies that can be applied to the benefit of critical SIP problem areas, including recent accomplishments under the CHSSI program, and (4) develop requirements for PET support of SIP. The SIP2001 Forum brought together 56 researchers and managers from the DOD SIP and MSRC communities. Thirty-two papers were presented in six sessions entitled: General overviews, Enabling Software technologies, Example Applications, CHSSI SIP Project Status, Hyperspectral Imaging Exploitation, and Commercial SIP Processing Technologies. Highlights of the program were an overview of the new DOD HPC Modernization Program by Dr. Leslie Perkins of the High Performance Computing Modernization Office (HPCMO) and an Overview of SIP Technology by Dr. Richard Linderman.

CWO2001

The first forum in Climate, Weather, and Ocean Modeling (CWO2001), which is being chaired by Dr. George Heburn, NRL, Stennis Space Center and CTA Lead for

Article Continues Page 27...

Enhancing NAVO MSRC Visualization Capabilities

Pete Gruzinskas, NAVO MSRC Visualization Center

Success in scientific visualization, like high-performance computing, is a matter of hitting a moving target. That is to say, the technology is constantly changing. To succeed in meeting the mission of the High Performance Computing Modernization Program (HPCMP), these changing technologies need to be seamlessly inserted or adopted by the various components within the program. These changes almost always represent improvement, but changes are always difficult to incorporate into a workplace, even when they ultimately make life easier in the long run.

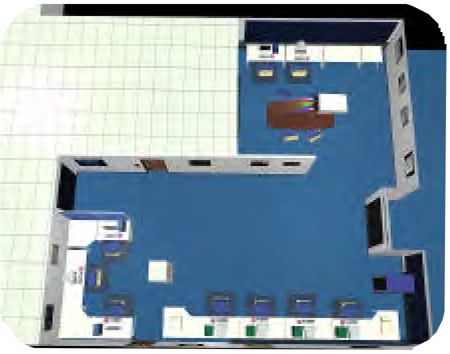


Figure 1. Rendering of the new Visualization Center.

The NAVO MSRC Visualization Center is no exception. We actually welcome change, painful as it may be, to our work process, if it can make the Center's mission easier and/or more cost efficient. Recently, the Visualization Center has undergone some major modifications, which represent more than just change, but a shifting paradigm in the visualization industry and a modified strategy to accommodate this new paradigm into our workflow. The good news about all these shifts and changes is that they will ultimately benefit the Department of Defense (DoD) researchers and their efforts to analyze their data.

The images shown on these pages represent changes to the physical layout of our Visualization Center, but the changes made to the Center go deeper than merely the physical layout (Figure 1). The physical layout compliments the groundbreaking work accomplished by the Visualization staff, by providing a means of collaboration and demonstration in a comfortable, well-equipped setting.

Figure 3. Collaboration Area in use by senior Navy management (including the Deputy Oceanographer of the Navy) for a brief on support provided for *Ehime Maru* salvage/recovery operations.

The new collaboration area, shown in Figure 2, is supported by a BARCO 1209 analog RGB projector. This projector can be driven by an SGI Onyx2, a Windows-based system, a VCR, or other desktops in the Center. Beyond the physical changes, there are enhancements to the Visualization Center in the areas of hardware, software, and networking, which will increase the availability of Center assets to the remote user community (Figure 3).

The Visualization Center hardware base has been diversified from exclusively SGI IRIX to a combination of SUN SOLARIS, LINUX, WINDOWS, and SGI IRIX. This diversification provides a level of compatibility with the various architectures employed by the user community. Additionally, all Center desktops are now equipped with Gigabit Ethernet NICs, and the Onyx2 has been retrofitted with GigE and ATM-OC12. At this time, testing several software packages are being actively tested, which will facilitate remote rendering within the DoD security model. Remote rendering will significantly augment the visualization resources available to the user community by placing large shared-memory partitions, state-of-the-art graphics architectures, software, massive storage partitions, and expertise at their disposal.

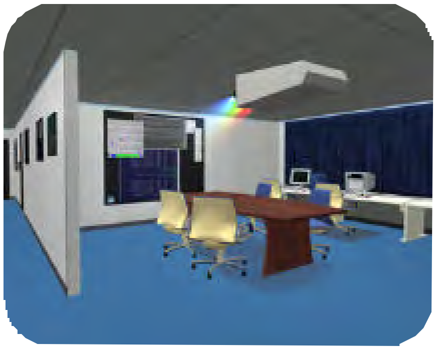
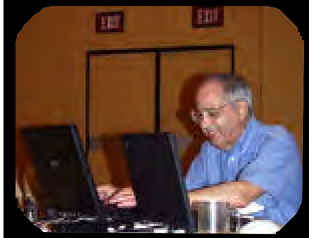


Figure 2. Rendering of the new Collaboration Area.

So, while change can initially be painful, the recent changes to the NAVO MSRC Visualization Center will only enhance its ability to provide state-of-the-art service to the user community.

Questions concerning Visualization Center assets or the use of these assets should be directed to Pete Gruzinskas, 228-688-4027 or gruz@navo.hpc.mil or contact the NAVO MSRC Help Desk at 228-688-7677.

DoD User Group Conference 2001 Sights



The NAVO MSRC, in conjunction with the Shared Resource Center Advisory Panel, hosted the recent 2001 HPCMP Users Group Conference at the Beau Rivage Resort in Biloxi, MS. The focus of the conference was, as always, the users and high performance computing (HPC) support which the program affords them. Each year this conference is hosted by one of the four MSRCs to gather the people involved in the program for an exchange of ideas and information. This year's attendance was close to 400—the largest conference in DoD High Performance Computing Modernization Program (HPCMP) history.

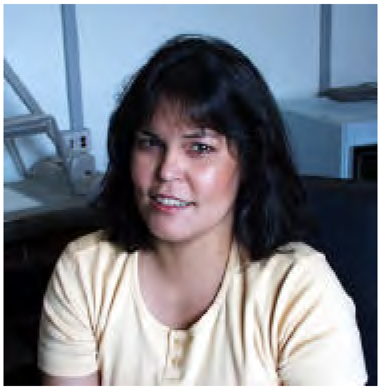
Featured speakers of the conference this year were Dr. Robert Ballard of the Institute for Exploration and Dr. Arthur Hopkins of the Defense Threat Reduction Agency. The conference afforded the users a chance to present their progress on various projects sponsored by the HPCMP. Due to extraordinary demand, an extra session was added to the technical program to accommodate additional user presentations. A variety of tutorials were also provided that covered a wide range of topics such as the Grid, Kerberos, MATLAB Programming Techniques, Pthreads, Advanced FORTRAN 90/95, Introduction to the Cray MTA Architecture, and Numerical Methods for Sparse Systems.

For information regarding the UGC 2001 conference in Biloxi, visit the conference web site at:

http://www.hpcmo.hpc.mil/Htdocs/UGC/UGC01/index.html

NAVO MSRC PET Update

Eleanor Schroeder, NAVO MSRC Programming Environment and Training Program (PET) Government Lead



So PET as we knew it wound down, and the new (and hopefully improved) version of PET is in high gear. NAVO's PET, as Component 1, has three primary support responsibilities with respect to the DoD HPCMP community: Climate, Weather, and Ocean Modeling (CWO); Environmental Quality Modeling (EQM); and Computational Environments (CE).

We welcome to the new NAVO PET family Dr. George Heburn (Mississippi State University integrator component lead), Dr. Jay Boisseau (University of Texas in Austin, CWO POC), Dr. Mary Wheeler (University of Texas in Austin, EQM POC), and Dr. Shirley Moore (University of Tennessee in Knoxville, CE POC).

The first round of projects have been approved and are underway. We have retained some of the staff from the previous PET program, including Andrew Schatzle (Online Knowledge Center/Collaborative Distance Learning Technologies Technologist), Brian Tabor (Education, Outreach, and Training Technologist), and Dr. Phu Luong (EQM onsite at Engineer Research and Development Center (ERDC)). At press time, none of the CWO onsite positions (NAVO and ERDC) or the NAVO EQM onsite position, had been filled.

Please visit the PET web site (www.navo.hpc.mil/pet) to learn more about the new program and the exciting things that we will be doing.

And A Big Thanks....

Not enough can be said about the team of people the NAVO MSRC PET were fortunate to have working with them for the past five years. Their dedication, enthusiasm, and hard work epitomize the successes of that program. The success of the Tiger Team effort was such that it was adopted under the new PET program. Several of the usability tools (such as Web Queue Stats, Resource Allocation Tool, and Resource Allocation Exchange) are being adopted into corporate initiatives such as the Information Environment group. Legion and Globus have banded forces to work toward a strong metacomputing system that will one day be a common computational environment.

So many, many thanks to all our partners whose contributions have made this program a success: Alcorn State, Center of Higher Learning/ University of Southern Mississippi, Duke University, Grambling State, Illinois Institute of Technology, Mississippi State University, MPI Technologies, Morgan State University, North Carolina A&T, Oregon State University, San Diego Supercomputing Center, Syracuse University, Tennessee State, University of Minnesota, University of California at San Diego, and University of Virginia. And a very grateful thank you to the wonderful staff that I've had the privilege of interfacing with since 1997 when I first came on board: Dr. Timothy Campbell, Dr. John Cazes, Dr. Howard Cohl, Dr. Jay Jayakumar, Eruch Kapadia, Shelley Clay, Luke Lonergan, Dr. Robert Melnik, Jack Morgan, Dr. John Pormann, Dr. Gil Rochon, Andrew Schatzle, Dr. Walter Shackelford, Margaret Simmons, Brian Tabor, Gail Van Nattan, Evan Willett, and Chuck Young.

Navigator Tools and Tips

Running X Windows With Parallel Codes on HABU Under LoadLeveler

John Skinner, NAVO MSRC Support Analyst

This article describes a simple method for running multi-node, multiprocessor programs "interactively" on the NAVO IBM SP3 (HABU) internal compute nodes under LoadLeveler, the IBM batch scheduling system. We make use of the X Window system and Kerberized ssh to securely set up an X connection from the internal network used by HABU's compute nodes back to a user's local computer running X Windows.

Parallel jobs on HABU, such as Message Passing Interface (MPI) codes running under the Parallel Operating Environment (POE), have to run on compute nodes and also have to be submitted as batch jobs via LoadLeveler. Since HABU's internal network is locally configured to only talk to HABU's two interactive login nodes and the NAVO Archive Servers, we cannot set our DISPLAY back to our local computer (in the standard X Windows manner) and send an xterm window or other X-based program from a LoadLeveler run back to our home computer. Instead we must make use of Secure Shell (ssh) to handle the required setup.

This method allows us to send xterm windows from a running LoadLeveler job back to our home computer to use like normal login windows. We can then run Unix commands, debug programs with the IBM pedb program (a Graphic User Interface to IBM's pdbx parallel program debugger) or run pdbx itself in command-line mode. We can then start multiple programs in background and rerun our code under POE from the xterm window started within our LoadLeveler batch job.

The following procedure outlines the required steps to use whether you have ssh on your home system or not, and an example LoadLeveler script is also provided.

PROCEDURE

1. Connect to HABU interactively via one of the Kerberized login commands Kerberized ssh, ktelnet, krlogin, or krsh (if possible, use ssh for its ease of use with the X Windows setup):

```
mycomputer% ssh -l skinman habu.navo.hpc.mil
Last login: Tue Aug 29 15:14:33 2000 from
204.222.177.188
f15n13e%
```

2. If using ssh from your local system to logon directly to HABU, continue to Step 3. Otherwise, you need to manually set up and test a valid X Window connection and DISPLAY after logging in, as noted in steps A - F below:

A. Run "xauth list" on your local workstation to list MIT "Magic Cookie" string that your X Server uses to authenticate X Clients that connect to your computer.

An xterm window or any other X-based programs you want to run from within your LoadLeveler job will be the X Clients that need access to this encrypted "cookie" so they can display on your screen:

```
mycomputer% xauth list
mycomputer:0  MIT-MAGIC-COOKIE-1 0578
34173e477d52401161623779276a
```

B. Use xauth with the "add" option to cut and paste the appropriate line from your local X setup into your .Xauthority file on HABU. Be sure to use either your local system's IP number or full domain name when adding this to your .Xauthority file on HABU:

```
f18n13e% xauth add mycomputer.at.my.
domain:0 MIT-MAGIC-COOKIE-1057834173e
477d52401161623779276a
```

NOTE: It should be necessary to execute steps 2.A and 2.B only once, since the information will be saved and valid as long as you don't log off your current interactive login session on HABU.

C. On HABU, set your DISPLAY environment variable to either the IP number or the fully qualified domain name for your local workstation (don't forget the "0.0" part):

```
f18n13e% setenv DISPLAY mycomputer.at.my
domain:0.0
```

or

```
f18n13e% setenv DISPLAY 204.222.177.65:0.0
```

D. Test the connection by starting an X client such as xclock or xterm from your HABU login window and verifying that it displays on your workstation screen:

```
f18n13e% /usr/bin/X11/xterm
```

E. From the same login window on the IBM,

Article Continues Page 26...

A Look Inside NAVO MSRC

We welcome our visitors...

Left:

L-R **- CAPT Tim McGee, Comanding Officer, NAVOCEANO; U.S. Representative Gene Taylor**

Right:

L-R **- Frank Lovato, Security Officer, NAVOCEANO; IGA Yves Desnoes, Director, French Naval Hydrographic and Oceanographic Service (SHOM), and Paul Cooper, Head, NAVOCEANO, International Division**

Left:

L-R **- Rich Peel, Program Manager, National UUV Test and Evaluation Center (NUTEC); Dave Cole, NAVOCEANO; Debbie Hisayasu, Director, NUTEC; Debbie Triplet, NUTEC; Craig Peterson, Director, NAVOCEANO, N9**

Right:

L-R **- CAPT Fred Shutt, Commanding Officer, Coastal Systems Station; Dave Cole;**
LCDR Angel Rivera, Staff METOC Officer,

Left:

L-R **- Cecil Mills, Mississippi Enterprise for Technology; Dr. D.L. Durham, Technical Director, COMNAVMETOCOM; Suzanne Case, Gulf Coast Office Director for U.S. Senator Thad Cochran; Dave Cole**

Right:

Staffers to U.S. Senator Thad Cochran and U.S. Representative Chip Pickering

Left:

Space Campers at work

Right:

L-R - Dave Cole; John Palmer, U.S. Ambassador to Portugal

Left:

L-R - Julie McClean, NAVO MSRC Challenge User; Ludwig Goon, NAVOCEANO Visualization Specialist

Right:

L-R - Dave Cole; CAPT John Kamp, DARPA; Craig Peterson, N9; Larry Raynor, PMS 395

Left:

L-R - LCDR Mykyta, N096; LCDR F. Swett, NAVOCEANO

Right:

L-R - Stuart Holmes, Legislative Assistant for Defense to U.S. Senator Thad Cochran; Clayton Heil, Legislative Director to U.S. Senator Thad Cochran; Dave Cole; CAPT McPherson, Chief of Staff, COMNAVMETOCOM

Left:

L-R - CAPT McPherson,Chief of Staff, COMNAVMETOCOM; CAPT Chris Gunderson, Deputy Oceanographer of the Navy

Right:

Mississippi Students visit NAVOCEANO

Article Continued from Page 23

now ssh locally from HABU to HABU itself:

```
f18n13e% ssh habu.navo.hpc.mil
Last login: Thu Aug 31 10:24:40 2000 from
204.222.177.182
f15n13e%
```

F. List the new DISPLAY setting created by this ssh login so that you can hardcode it into your LoadLeveler batch script.

```
f18n13e% env | grep DISPLAY
DISPLAY=f18n13e.navo.hpc.mil:11.0
```

Notice that ssh has set your DISPLAY variable in this new login session to a new value, which happens to be one of the two IBM login nodes with an additional ssh identifier at the end of the string. The ssh command automatically takes care of setting DISPLAY and the xauth "magic cookie" setup needed for this new login and connects it back to your existing HABU login, where your DISPLAY is still set to "204.222.177.65:0.0" (or "mycomputer.at.my .domain:0.0"). The connection to the X11 DISPLAY is automatically forwarded by ssh in such a way that X programs started from a HABU shell will now go through the encrypted channel, and the connection will be made from HABU to the X Server running on your computer.

Once you close the interactive ssh login session, the new DISPLAY setting that ssh just set for you is no longer valid and can't be used again. For this reason, do not abort this ssh connection until you are through with your pedb debugging work or any other X-based programs that you want to run from within your LoadLeveler script on the IBM SP compute nodes (which is where LoadLeveler jobs run).

3. Add the DISPLAY information created from Step 2 to your LoadLeveler script and llsubmit your job. Be sure to set the DISPLAY variable within your script before you execute any X commands such as Xterm, pedb, or aixterm within the script itself. You can then use your current interactive login window to run llq and /maui/bin/showq to monitor your job until it begins execution and sends the Xterm to your workstation.

SCRIPT

What follows is a sample LoadLeveler script that sets up the correct DISPLAY and starts an interactive xterm session from which you can run any HABU commands, start up a pdbx session, or run pedb and send the debugger window back to your computer. You can run in this manner until you reach the wallclock limit set for your batch job and then llsubmit the job again using the same DISPLAY. Remember that if you kill the login ssh session that first set up the DISPLAY, you will have to re-login with ssh and modify your LoadLeveler script accordingly to point to the new DISPLAY setting.

```
#@ environment = ENVIRONMENT=BATCH
#@ shell = /bin/csh
#@ output = $(jobid).$(stepid).out
#@ error = $(jobid).$(stepid).error
#@ network.MPI = css0,shared,US
#@ job_type = parallel
#@ job_name = my_debug_job
#@ account_no = NA0101
#@ node = 6
#@ tasks_per_node = 4
#@ node_usage = not_shared
#@ wall_clock_limit = 1:00:00
#@ class = batch
#@ queu
    setenv WORKDIR /scr/skinman/my_test
    if (! -e ${WORKDIR}) then mkdir -p $WORKDIR
    endif
    cd $WORKDIR
# Copy needed files to $WORKDIR
    cp $HOME/mpi.exe $WORKDIR
    cp $HOME/mpi-src/*.f90 $WORKDIR
    cp $HOME/input.dat $WORKDIR

# Set DISPLAY to value set by your ssh login to
# HABU or, if logged in via ktelnet/krsh/krlogin to
# HABU, set to the value created by the ssh from
# HABU to itself.
    setenv DISPLAY f18n13e.navo.hpc.mil:11.0

# Start an xterm from this script and send it to your
# local workstation and use it like a normal login
# session.
    echo "trying xterm connection to $DISPLAY..."
    xterm

# You can also run the xterm in background, start up
# the pedb debugger, and use both interactive win-
# dows before quitting the LoadLeveler batch job.
    xterm&; pedb

# You can now run your parallel code under the
# graphical parallel debugger, pedb, and interactively
# step through your program while it is executing on
# multiple processors. You can rerun the job multiple
# times and also run other commands, but you are
# limited to no more than the 6 nodes this
# LoadLeveler script requests when it is queued with
# the llsubmit command.
    pedb ./mpi.exe -procs 24 -labelio yes
# End of sample LoadLeveler script
```

Upcoming Events

Current and Future Trends in Numerical PDEs: Where is the field, and where is it going?

February 8-9, 2002 · University of Texas at Austin, Austin, TX

http://www.ticam.utexas.edu/%7 Earbogast/jim.html

35th Hawaii International Conference on System Sciences (HICSS-35)

January 7-10, 2002 · Big Island, Hawaii

http://www.hicss.org/

(IPDPS 2002) (IPDPS = IPPS+SPDP)

April 15-19, 2002 · Fort Lauderdale, FL

http://www.ipdps.org/ipdps2002/

Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)

Aug 10-13, 2002 · Winnipeg, Manitoba, Canada

http://www.spaa-conference.org/

Focus: MTS Oceans 2002

October 28-31, 2002 · Biloxi, MS

http://www.mtsgulfcoast.org/oceans 2002.html

Signal and Image Processing & Climate, Weather and Ocean Modeling Forums

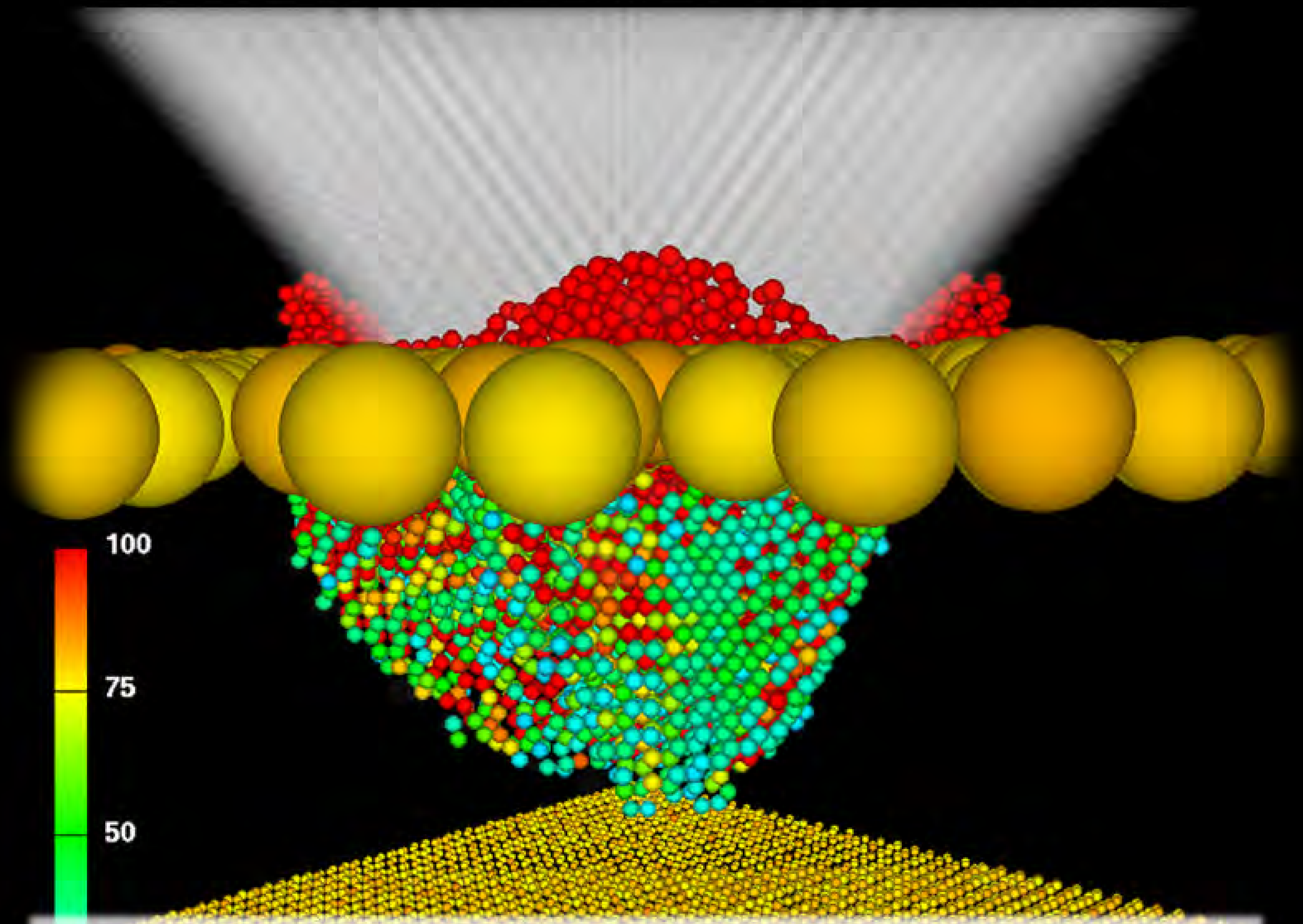
Article Continued From Page 18...

CWO, is to be held on September 18-20 in Gulfport, Mississippi. The specific objectives for CWO 2001 are to provide a forum for the interchange of ideas on (1) the modeling of the earth's physical environment using high-performance computational resources, and (2) the challenges of developing scalable, portable model codes that execute efficiently on computational resources ranging from high-powered workstations and clusters to supercomputers. At the time this article was written, 38 invitees from the CWO community accepted invitations to attend the forum. The Forum program, which can be viewed on the CWO2001 web site, includes 24 papers arranged in seven sessions entitled: General overviews, Atmospheric Modeling, DMEFS (Distributed Marine Environment Forecast System), CWO CHSSI Alpha Test Reviews, Gulf of Mexico, PET Support to CWO, and Ocean Modeling.

Further information on SIP2001 and CWO2001 is available at their respective web sites:

- ☞ http://www.navo.hpc.mil/pet/sip2001/
- ☞ http://www.navo.hpc.mil/conferences/cwo/

Downloadable speaker charts are available at the web site for most of the SIP forum talks and will be for the CWO talks shortly after CWO2001.



Naval Oceanographic Office * MAJOR SHARED RESOURCE CENTER

1002 Balch Boulevard . Stennis Space Center, Mississippi . 39522