HPCinsights

from the DoD Supercomputing Resource Centers

Spring 2009

Six Centers Meld into a Single Operational Unit

HPC Insights is a semiannual publication of the Department of Defense Supercomputing Resource Centers under the auspices of the High Performance Computing Modernization Program.

Publication Team

AFRL DSRC, Wright-Patterson Air Force Base, OH
Joan Henley
Chuck Abruzzino

ARL DSRC, Aberdeen Proving Ground, MD
Debbie Thompson
Brian Simmonds

ARSC DSRC, Fairbanks, AK
Debra Damron
Mary Haley

ERDC DSRC, Vicksburg, MS
Rose J. Dykes

MHPCC DSRC, Maui, HI
Jeff Schmidt
Betty Duncan

Navy DSRC, Stennis Space Center, MS
Christine Cuicchi
Lynn Yott

HPCMPO, Lorton, VA
Deborah Schwartz

MANAGING EDITOR
Rose J. Dykes

DESIGN/LAYOUT
Betty Watson

COVER DESIGN
Miguel Valenciano

Contents

DoD Supercomputing Resource Centers

By Brad Comes, HPC Centers Project Manager

A goal to modernize the high performance computing (HPC) capabilities of the Department of Defense (DoD) via the creation and sustainment of the High Performance Computing Modernization Program (HPCMP) was established more than 15 years ago. A recent development toward this goal includes the incorporation of two former Allocated Distributed Centers—the Maui High Performance Computing Center and the Arctic Region Supercomputing Center—into the fold of what has been historically known as the Major Shared Resource Centers. Along with this comes the renaming of all six centers to DoD Supercomputing Resource Centers (DSRCs). Although each Center has its own unique attributes, the melding of the six Centers into a single, cohesive, and effective operational unit is yet another step toward fostering an improved set of services for the DoD HPC user community. An example of the melding is this new publication, *HPC INSIGHTS from the DoD Supercomputing Resource Centers*, representing all six DSRCs.

More and more research and development (R&E) and testing and evaluation (T&E) teams are pursuing their objectives through the use of multiple HPC systems located at multiple Centers. This has created a tug of two forces for the Centers' service model. One force is to maintain the collective strength of the HPC Centers via their diversity. The other force is to provide the user with a productive work environment that supports leveraging what they have learned and developed at one Center for use at another Center. Several initiatives are underway that work to balance these forces.

One initiative is the continued support of a group called the Baseline Configuration Team. This group is constantly soliciting input from the user community with respect to differences between the Centers that, if harmonized, would increase user productivity. One can visit the Baseline Configuration Team's Web site at http://www.afrl.hpc.mil/consolidated/bc to provide input (*User Input*), view the baseline configuration (*BC Policies and Projects*), and review the Centers' compliance with these policies (*Compliance Matrix*). This is a living set of policies. Checking back often for updates is recommended.

Another initiative is the formation of the Consolidated Customer Assistance Center (CCAC). There were some startup pains associated with standing up this capability, but it continues to grow and improve and always will. For example, CCAC recently implemented a single customer ticketing system that is shared by all six Centers. It is envisioned that the CCAC will continue to serve a larger and larger role in providing single-point access to services and information to the user community. CCAC has an informative cross-center Web site: http://www.ccac.hpc.mil/.

Brad Comes
HPC Centers Project Manager

Last, but certainly not least, is the User Advocacy Group (UAG). This group of individuals, appointed by a Service or Agency to represent the HPC user in the HPCMP, meets three or four times every year and maintains an active dialog via e-mail. The HPCMP listens and responds to what users have to say via this forum, as members of the UAG are effective at representing user HPC interests. Users can contact the UAG via e-mail at hpc.uag.members@hpcmo.hpc.mil and should always consider leveraging this group as an effective means toward getting their concerns and requirements addressed.

The three above-mentioned services are by no means an all-inclusive list of ways users can express their requirements. HPC users can check out the HPCMP Web site at www.hpcmo.hpc.mil for other vehicles.

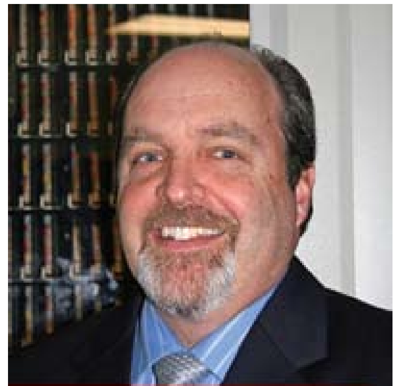
Setting the goal of modernizing HPC capabilities within the DoD through the formation of the HPCMP in 1994 has certainly contributed toward accelerating the development and transition of defense technologies into superior warfighting capabilities. However, the HPCMP cannot rest. To retain this superior capability, the HPCMP must continue to foster the development and deployment of leading-edge HPC capabilities and computational techniques in support of advanced computational environments for the DoD R&D and T&E communities. An important part of this continued success is effective communication between the HPC user community and the HPC services community. In support of this goal, users should continue to express their requirements, concerns, and successes via one or more of the above-mentioned forums.

Air Force Research Laboratory DoD Supercomputing Resource Center

By Frank Witzeman, Director

In the past year, the Aeronautical Systems Center (ASC) Major Shared Resource Center (MSRC) completed a major transition from one Air Force organization to the Air Force Research Laboratory (AFRL). There was some concern about the transformation and consolidation of the four MSRCs and two Allocated Distributed Centers in Alaska and Hawaii into the six centers called DoD Supercomputing Resource Centers (DSRCs). This strategic move by the HPCMP will result in a renewed enterprise devoted to providing a balanced set of resources and capabilities to the user community, and the AFRL DSRC will do its part to make the enterprise a success.

The AFRL DSRC will continue to serve existing DoD customers and investigate opportunities to support a broader AFRL user base at the "entry-level" of high performance computing. The AFRL DSRC has growing requirements for floor space, electrical power, and cooling. For those who have visited the facility, the noticeable age of the building emphasizes its inability to contain the weight and size of the newer systems and forces reevaluation of the building's ability to maintain the constantly changing environmental controls required for high performance computers. One of the several alternatives for the AFRL DSRC is building a new facility that is flexible enough to meet future requirements.



Frank Witzeman
Director, AFRL DSRC

There has been a partnering relationship with the Aeronautical Systems Center to help design and build a new Information Technology Complex under an Air Force military construction project. To date, plans are progressing to start construction in 2010 and be ready by late 2012 to occupy the new facility. The target is about 10,000 sq ft of modern data center space for the AFRL DSRC use, approximately doubling the current floor space.

In the near term, the focus will be the decommissioning of Eagle SGI Altix 3700 later this year and Falcon HP XC Opteron in the spring of 2010. The new 2010 HPCMP Technology Insertion system will complement the existing Hawk SGI Altix 4700 with approximately 20,000 additional processor cores.

Gotcha Project Success Story

By Tessa Welton, AFRL DSRC

Introduction

The Gotcha radar exploitation system is an unprecedented, all-hour, radio-frequency surveillance system. It will provide many military advantages in wide-area urban environments and overcome the limitations of current surveillance systems such as Angel Fire and Night Hawk. This real-time system will save the lives of warfighters by providing all-hour, all-weather surveillance at high altitudes. Were the Gotcha radar system available today, it would have significant impacts on urban battlefield operations.

Objectives

Several years ago, Edmund Zelnio started to consider the vast opportunities in the realm of synthetic aperture radar (SAR)-based, wide-area surveillance that could be used long-term in expansive weather conditions and circumstances. It was through this exploration that the Gotcha project idea was conceived. Although radar technology had the capabilities for such wide-range surveillance, the computational power needed to process such a large amount of data simply did not exist in the late 1990s and early 2000s.

In 2003, the project team members started running synthetic data using the Xpatch code on the DSRC HPC machines. In 2004, they were then able to show the Scientific Advisory Board how the DoD's mission could benefit from the Gotcha project. When the SGI Altix 4700 Hawk HPC system came into production in 2007, the Gotcha project was able to benefit from the large architecture.

To process 1 sq km diameter ground spot SAR data in real-time, approximately 1100 cores of Hawk were required. In addition to the massive amount of computational power that the AFRL DSRC has provided, the Gotcha project has utilized the personnel resources and applications available at the DSRC. These both have allowed the project to process and present the data in a video format that is readable to the human eye. Dr. Majumder has worked closely with Dr. Bracy Elton, Gary Sivak, John Carter, and Lloyd Slonaker in various capacities to accomplish his project goals. Dr. Brent Andersen has also been extremely helpful to them from a systems architecture standpoint on Hawk.

To process the data produced by SAR, it is necessary to use the highest possible computational power and bandwidth in addition to the most advanced data storage available. Because of the requirement for massive computational power and the need for dedicated equipment, the Gotcha project real-time exploitation leader,

Computer graphic of a possible scenario: From a remote location, a soldier accesses Gotcha data on a secure hand-held device from an aircraft scanning the surrounding countryside

Dr. Uttam K. Majumder of AFRL/RYAS, submitted a Dedicated HPC Project Investment (DHPI) proposal. DHPIs support projects that require unique high performance computing resources that are installed, operated, and maintained by the recipient.

The Gotcha project hosts a set of unique HPC storage requirements. For example, the project has a 10-TB partition in the workspace of each of the Falcon and Hawk systems. This houses a portion of the January 2006 radar data. The Gotcha project is also placing its own 18-TB fast storage system at the AFRL DSRC.

Impacts to the DoD Mission

Although the DSRC has the computational power needed for the Gotcha project in terms of number of processors, it currently lacks in the high rate of speed necessary to process the project data in real-time. The project needs high bandwidth in a short amount of time. Because of the strong partnership currently in place with the DSRC, the Gotcha DHPI system will be housed at the AFRL DSRC. The system will have 2048 cores and cost about $2.2 million.

This system should go into production in Spring 2009. Having this dedicated system will fulfill the extremely high bandwidth needs of 4 GB/s and the 2-7 Tflop/s processing speed that will process the large amount of data in real-time. By Fall 2010, the Gotcha project is expected to have a real-time live demonstration of radar surveillance of a 5 sq km diameter ground spot. An application called Rapid Link will connect from the ground to the air at 4 Gb/s and connect it to the data system to show it in real-time. Using this new DSRC system will allow continued development and a dry run of the demonstration before Fall 2010 in order to find and correct any potential glitches in the program.

Benefits to the Warfighters

The hope for the Gotcha project is for it to someday be able to process radar surveillance over a 40 sq km diameter ground area in real-time and to even be able to send this in high-resolution video to a hand-held device. Although it is a long-term goal, the ability to give a soldier a clear view of his surroundings during any hour of the day and through any type of weather could very well revolutionize the intelligence of combat and warfare.

Conclusion

Thanks to the AFRL DSRC, the project team can see a clear path of how to address the problems of computational power. By helping with the Gotcha project, the DSRC is making a big impact that will help the warfighters with this exciting surveillance technology. This is currently the largest project in existence on signal and image processing. The needs of this project push the upper boundaries of storage and processing power. The needs are the highest available version of everything computational currently in existence and beyond. There are no limits to the possible reach of this project as far as geographical coverage, speed, and data processing.

Army Research Laboratory DoD Supercomputing Resource Center

By Charles J. Nietubicz, Director

Charles J. Nietubicz
Director, ARL DSRC

Scientific Computing, Supercomputing, or High Performance Computing—whatever the current terminology—the need to do computational analysis has long been an element of the Research, Development, Testing, and Evaluation communities. The need for a computational analysis capability goes back to the early days of digital computing in 1946 with the inception of ENIAC, the first Government electronic digital computer, and its use for producing artillery firing tables. The difference today is that this capability is an ESSENTIAL element of the research and engineering work being undertaken at Department of Defense (DoD) laboratories and with partners in industry and academia. The long tradition of excellence in the advancement and application of computing technologies, which started with the people and computing systems from the Army Ballistic Research Laboratory, continues today with the outstanding staff and capabilities at the U.S. Army Research Laboratory DoD Supercomputing Resource Center (ARL DSRC) located at the Aberdeen Proving Ground, Maryland.

The ARL DSRC is proud to be an integral part of the DoD High Performance Computing Modernization Program (HPCMP) today. We are equally pleased to have helped develop some of the early program components to include the initial program plan that went to Congress, the IDREN and DREN concepts (including DREN Program Managers); the early access systems; the software component; and the provision for the first Program Manager. So as proclaimed in our logo, we have a Tradition of Innovation and an Attitude for Success.

The tradition of innovation, which has been enabled by the HPCMP, has allowed us to bring to you, the researchers and scientists, the best available hardware, software, networking, people, and solutions. In 2003, the ARL DSRC took the lead by introducing, evaluating, testing, and installing a 256-processor commodity cluster from Linux Networx. This led to an almost exponential rise in capability at the ARL DSRC and for the HPCMP in general with the installation of a 2048-processor JVN system and a 2304-processor Stryker system in 2005. Both were decommissioned at the end of March 2009, but marked a significant leap in price/performance within the HPCMP. MJM and Humvee, acquired 2 years later, continue to provide reliable, outstanding service to the user community. The most current increase in capability has come from the installation of MRAP, a 10,400-processor core system, using the AMD "Barcelona" quad-core 2.3-GHz Opteron Processors. MRAP, placed into production April 13, 2009, has 41.6 TB of memory, 500 TB of RAID disk, and requires 700 KW of power and 195 tons of cooling.

We are excited about the upcoming Technology Insertion 2009 installations this summer of two SGI systems. The test and development system has been installed and is being used to help ensure that the production systems have a smooth transition to operations. The two production systems will use the recently announced Intel XEON 5500-processor family, a.k.a. Nehalem, and will offer significant improvements in performance, especially in memory bandwidth. The two systems will have 10,752 and 6656, 2.8-GHz processor cores, respectively. The systems are expected to be online and productive within the November 2009 time frame.

After 12 years of providing you ARL DSRC updates, status, successes, activities, and pictures of our staff through the LINK magazine, we have embarked on a new forum — *HPC Insights*. We will be contributing articles of interest from the ARL DSRC and hope you will find this new forum interesting and informative. Please take the opportunity in this introductory edition to read about the pioneering work of Dale Shires and his team in looking at GPU technology and an article from Joshua Devine on an LM21 event looking at the acquisition and management of commercial-off-the-shelf (COTS) software across the program.

As the ARL DSRC Director, and speaking for our outstanding and talented Government and contractor staff, I would like to thank you for your cooperation and feedback and for your dedication and use of the HPCMP resources. Together we stand united in helping to provide superior technology in support of our most important assets, the men and women of our Armed Forces, who every day are on the front line of our Nation's defense.

LM21 PDK Event at ARL DSRC for Enterprise Software Management

By Joshua Devine, ARL DSRC

The Army Research Laboratory DoD Supercomputing Resource Center (ARL DSRC) hosted a 5-day LM21 Pre-Design Kaizen (PDK) event on January 26, 2009, for the purpose of planning a centralized, coordinated software procurement, and management process. LM21 stands for Lockheed Martin in the 21st Century and is the premier program for implementing Army Lean Six Sigma (LSS) practices to achieve operating excellence throughout the entire organization. In order to gather all the appropriate input to develop a best practice for the HPC community, we had participants from each of the six DSRCs. These participants encompassed the knowledge base, experience, and talent needed for a successful event outcome. We also had HPCMPO representation, which enabled us to maintain the vision, scope, and clear direction of our efforts. In total, we had 17 participants representing a cross-disciplined skill-set, some of whom included systems administrators, configuration management analysts, procurement/logistics analysts, application managers, and core business leadership.

ARL DSRC Director Charlie Nietubicz kicked off our event detailing the Program's current issues with the lack of a rapid and flexible software solution that meets the dynamic needs of our end users. After the initial welcome and introductions were made, the event became a 5-day flurry of activity with a fast-paced schedule that included an outbrief on the final day, detailing our solution. With 17 participants to include many different ranks and positions within the HPC community, the euphemism "herding cats" quickly comes to mind. Fortunately, we had two excellent facilitators, Casey Bretti from the Air Force Research Laboratory (AFRL) DSRC and Kirk Judd from Lockheed Martin, who helped the group maintain clear focus, direction, and purpose of the LM21 PDK event. The facilitators helped everyone feel comfortable with brainstorming and critiquing ideas without attacking individuals.

The group was placed in a stressful position by trying to consolidate each Center's software and configuration management process into a unified robust model by the end of the week. Rather than breaking for lunch and interrupting our momentum, food was conveniently ordered and brought to our conference room. Doing this allowed us to maintain our pace throughout the day, debating each topic and determining the pros and cons of each data point. Butcher paper lined the walls where ideas were fleshed out; Post-it® notes were used to map the flow of our current and ideal future processes; and large flip pads were filled with our brainstormed ideas. It was as if a small tornado touched down on a cluttered desk and glued all of the paper to the wall. To add levity, Casey and Kirk had an ELMO doll that they would raise in the air if the debate wandered too far off topic. It stood for "Enough, Let's Move On!" There were other methods to ensure a smooth outcome. Rules were set on the first day to help coordinate communication: one person speaks at a time, silence implies consent, and all team members are equal. Occasionally, toward the end of the intense work week, the room would grow quiet, but the climate would quickly change as each participant would add their pertinent input into the process of developing our Get to Excellence (GTE) plan.

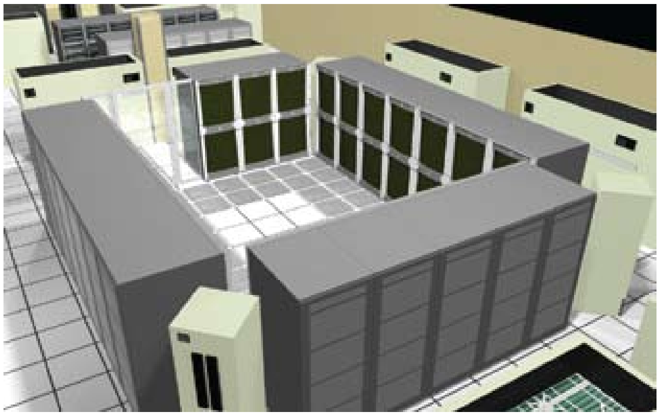
Our outbriefing included a presentation of our GTE plan and solution for designing and implementing a dynamic consolidated configuration and software management process to our cochampions of our LM21 PDK event Charlie Nietubicz and Jeff Gosciniak, and cosponsors Mike Knowles and Ed Williams. Tasks were assigned to key personnel, to include expected completion dates for each task, weekly updates, and a scheduled date for the final review of our implementation. Everyone was pleased with the event outcome and had high expectations for the implementation of our plan. The journey to complete these tasks will involve each LM21 participant and his/her respective DSRC site involvement for a robust and scalable solution for the HPC community.

Based on the level of expertise, congeniality, and motivation of each member, I feel confident that we will work well to complete our mission. Everyone on the team agreed that the communication among such diverse disciplines was surprisingly engaging, efficient, and pleasant. I would like to extend a special thank you to Eia Dobbins at ARL for coordinating our lunches, supplies, and visitor arrangements. Although it was a large undertaking, Eia's efforts were greatly appreciated and orchestrated seamlessly. It was a pleasure working with each person that attended and to see such a high level of cooperation, and I consider this experience a standard to judge future events.

New Systems Bring 17,500 Cores to ARL DSRC

By Brian Simmonds, Outreach Team Lead, ARL DSRC

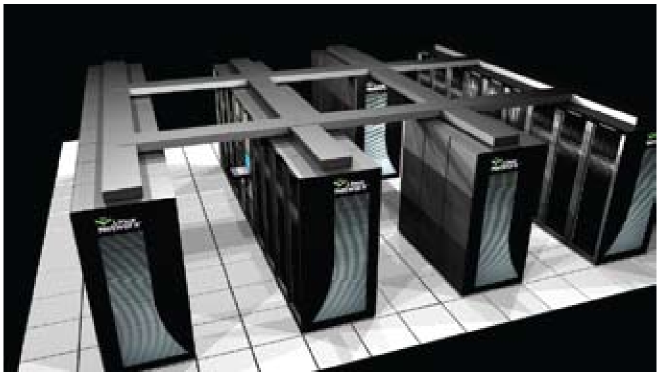
The ARL DSRC, as part of the Technology 2009 acquisition, was awarded three SGI Altix ICE 8200 linux clusters. The first is a 10,752-core system (2688 quad-core processors) with 32 terabytes (TB) of system memory and 600 TB of local disk storage. The second smaller system will have 6656 cores with 52.2 TB of system memory and 400 TB of local storage. The third 96-core system will be a test and development system that is used to support the other two larger systems by providing a nonproduction system to test configurations, applications, and tuning procedures.



Rendering of 10,752-core SGI Altix ICE 8200 system

All three systems use the new 2.8-GHz Intel Nehalem processor. The Nehalem is Intel's next-generation microarchitecture that uses 45 nanometer circuit technology for more efficiency and another technology called QuickPath to improve distribution of data to and from the four cores contained on each chip.

The system that the larger of the Altix systems replaces, JVN, is part of the TI-04 acquisition. The JVN system consists of 36 racks that contain 1024 processors (2048 cores), 4 TB of system memory, and 50 TB of local disk space. The new Altix is also 36 racks, but incorporates 10,752 cores, 32 TB of system memory, and 600 TB of local disk. That is 8704 cores, 28 TB of memory, and 550 more TB of disk space in the same footprint. This is achieved through increased density of the hardware. Each of the 21 Altix compute racks contain 128 quad-core processors (512 cores) as well as InfiniBand switchgear and networking modules. JVN has four racks dedicated solely to networking switches.



Rendering of Linux Networx Evolocity II system JVN, acquired as part of TI-04

You would think with this increased density comes increased power and cooling requirements, but this is not the case. JVN required 585 kilowatts (KW) of electrical power to operate and generated approximately 2000 BTUs per hour of heat to be dissipated. The Altix system will require 600 KW and will output 2050 BTUs per hour of heat, respectively, which is remarkable considering the considerable increase in compute power.

At the time of TI-04, the state-of-the-art interconnect network was Myrinet, a proprietary interface that operates at a speed of 10 Gbits/s. In the 5 years that have passed, the new high-end networking is InfiniBand. This technology was also part of the TI-06 systems (MJM, HUMVEE). InfiniBand is an open-source system that is continually updated and improved, and the latest iteration is rated at 20 Gbits/s. The Altix has a dual-channel system that doubles the rate in a 3D Torus topology vs. the fat tree topology of MJM. This interconnect network is the backbone of a cluster, as it is the mechanism that the nodes use to pass information between themselves as well as the path the data takes to and from the disk systems.

These new systems will enhance the Center's ability to support a greater range of computational problems with more complexity and will increase the total processing power to over 350 teraFlops (trillion floating point operations per second). The systems are due to arrive in June and are expected to be available to users by September.

Arctic Region Supercomputing Center
DoD Supercomputing Resource Center

Frank Williams, Director

A 3-hour drive north of the Arctic Region Supercomputing Center (ARSC) DSRC facilities on the Fairbanks campus of the University of Alaska is the Arctic Circle. Alaska's geographic location at the top of the world has long played an important, strategic role for the Nation.

The U.S. Army Air Corps' Ladd Field in Fairbanks (now Fort Wainwright) was the transfer point for almost 8,000 aircraft delivered to Soviet pilots during World War II. The Alaska-Siberia route was quicker than the longer journey from the Eastern U.S. to the Russian Front. This geographic centrality continues to shape DoD strategies today.

The Butrovich Building on the University of Alaska campus in Fairbanks as seen on a crisp, clear late autumn day. ARSC supercomputers and storage facilities are located on the lower floor of the building

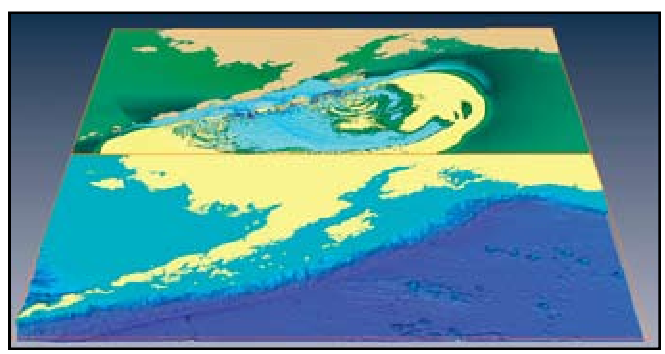
The ARSC DSRC has embraced the role within the DoD HPCMP as the sole provider of "open" research computing capabilities. The Center's geographic location provides focus to many of its major research efforts.

Foreign nationals who are members of DoD HPCMP projects are allowed to participate in research on ARSC DSRC systems. Results of research performed using ARSC DSRC resources may be presented in open, peer-review journals without compromising strict information assurance (IA) protections.

Frank Williams
Director, ARSC DSRC

These unique relationships provide feedback from the academic perspective that has broader implications for connecting the research, computing, and military communities. For example, the scientific paper "Kuril Islands Tsunami of November 2006: Impact at Crescent City by Distant Scattering" coauthored by University of Alaska Professor of Physical Oceanography Zygmunt Kowalik and ARSC DSRC HPC Specialist Tom Logan was published in the January 2008 *Journal of Geophysical Research*, VOL. 113, C01020. With its partner, the Northwest Alliance for Computational Science and Engineering at Oregon State University, the ARSC DSRC has created the Tsunami Computational Portal, a Web-based tool for use of major tsunami models and for sharing bathymetry https://tsunamiportal.arsc.edu. Consequently, civilian and military planners benefit from understanding the anticipated impacts of tsunamis, including specific onshore and at-sea destructive risks. And the HPCMP has gained experience in providing a wide community of users both shared access to data and convenient ways to make multiple runs when investigating parameter sensitivity.

The ARSC DSRC is a recognized leader in research of new and next-generation computing technologies.



A snapshot from a tsunami simulation using the COMCOT model from Cornell University shows a hypothetical event in the Aleutian trench off the Gulf of Alaska. This scenario has been used as a benchmark to compare the different tsunami models included in the portal. In the foreground, you can see the ocean bathymetry, while the background image displays the wave propagation (Image courtesy of Tom Logan)

One area of this leadership is that the Center makes a small cluster of IBM QS22 compute blades available for computational research on hybrid technologies. The cluster, which is not an allocated resource in the HPCMP, is called Quasar. On Quasar, each of 12 compute blades provides two Cell Broadband Engine (BE) processors, each with eight synergistic processing elements (SPEs). The PowerXCell 8i processors in Quasar are improved from those found in Playstations and elsewhere.

Quasar is an IBM QS22 system running Red Hat Enterprise Linux 5

Quasar fits in a single IBM BladeCenter H chassis, yet has a theoretical single precision peak performance of more than 5 teraflops. It has the Torque batch system, a Lustre file system, and other features that are typically found on much larger systems. During 2009, ARSC DSRC Quasar users will determine the extent to which this high theoretical performance can be achieved in practice, including for workloads of interest to the HPCMP.

Another area of leadership at the ARSC DSRC is in evaluating the scalability and performance of multicore processors. Starting with the 2312-processor Sun Opteron cluster Midnight, and continuing with later generations of multicore processors, the ARSC DSRC has examined the extent to which multicore and multithreaded processing scales up to large HPC workloads. Issues such as contention for cache and main memory, sharing floating point units, and MPI-based commu-

Midnight is ARSC's 2312-processor Sun Opteron cluster with a 68-TB Lustre file system

nication for both on- and off-node communication all play a role. For example, users want to know whether adding more processor cores in a single physical socket is more or less effective than having the same number of cores, but in a system with more physical sockets. ARSC DSRC researchers found that some applications scaled better on multicore sockets than others.

In addition to multicore processors from Intel and AMD, the ARSC DSRC evaluated the Sun T1 and T2+ multithreaded multicore processors, in which chip multithreading is combined with multiple cores, to add processing power to a compute node. Although these processors are geared more towards enterprise workloads (such as databases, e-mail, and other non-HPC applications), ARSC DSRC researchers found that they were able to handle common HPC synthetic benchmarks better than expected. This finding is particularly important for the next generation of Intel X86_64 processors, which enable hyperthreading.

The ARSC DSRC has also performed research on the potential role of Field Programmable Gate Arrays (FPGAs) for HPC workloads. This work includes deployment and evaluation of the Cray XD1 supercomputer, with a set of FPGA nodes, and as well as evaluation of stand-alone FPGA boards. ARSC DSRC researchers and affiliates examined the programming languages and models for FPGAs and available computational libraries. Overall, it was found that FPGAs have some outstanding promise for use as coprocessors in HPC systems, similarly to how vector processors have been used in earlier generations of supercomputers. However, the utility of FPGAs is hampered by limited bandwidth between the FPGA and the CPU. More importantly, the available software stack and programming model for FPGAs creates a barrier to current important HPC applications. ARSC is continuing to play a role in developing next-generation software libraries and hardware environments for FPGAs.

The use of graphics processing units (GPUs) for HPC workloads is emerging as an important research topic for high performance computing. Again, the ARSC DSRC is at the forefront of this research. Several multi-GPU systems using GPUs from nVidia and ATI are undergoing evaluation at the Center. Software including Brook++, CUDA, and the RapidMind software development kit has been in use to determine their suitability for existing HPC applications of interest to the HPCMP. GPUs are similar to Cell BE processors, in that they have parallel lightweight processing elements. They are similar to FPGAs, in that they seem to fit well as coprocessors with the CPU for HPC applications. But they are unique in the large number of processor cores (256 or more, on a single GPU board) and requirement to use the single instruction multiple data (SIMD) programming model. The ARSC DSRC is taking an early look at the OpenCL language and other approaches to making GPUs, which are ubiquitous and powerful in desktop and laptop computers, usable as part of supercomputers.

The elevation of ARSC to DSRC status this year is an acknowledgment of the extremely talented staff the Center provides to the program in the area of hardware and software storage issues. The Center has been tasked with taking the lead on archival data storage issues for the Program and will be transitioning into that role over the rest of the year.

Pinging Pingo—the ARSC DSRC New Cray XT5

By Debra Damron, ARSC DSRC

Supercomputing resources at the ARSC DSRC more than doubled in January 2009 when the Cray XT5 system acquired through the HPCMP Technology Insertion 2008 (TI-08) process became available to users. The new system was one of four procured under a multimillion award to Cray Inc. as part of the TI-08 process.

Following an ARSC DSRC tradition of naming supercomputing systems with arctic themes, the new Cray XT5 is called Pingo, a name suggested by HPC Specialist Brad Havel. A pingo is an earth-covered ice hill formed by the upward expansion of frozen, underground water. Havel won the chance to name the new supercomputer at a staff-sponsored fund-raiser for the Literacy Council of Alaska. Each year the Center holds an auction of employee-donated items and unique HPC-related paraphernalia to benefit the Literacy Council and to cover the ARSC DSRC entry fee of its team, the CRAYons, in the Council's annual spelling bee benefit. Havel's recommendation to christen the new Cray XT5 after a unique cold-climate geophysical phenomenon was approved by the ARSC DSRC executive management team.

The Cray XT5 (Pingo) provides approximately 31.8 theoretical peak teraflops of computing power and has 3456 processor cores on 432 nodes. It has 13.5 terabytes of memory and 150 terabytes of shared high-speed storage. The software and programming environment on Pingo is based on Cray's Compute Node Linux and provides a wide range of functionality for parallel programming.

Cray XT5

Pingo arrived by truck over the Alaska Highway from the Cray Inc. manufacturing facility in Chippewa Falls, Wisconsin, in October 2008. For the next 3 months, HPC specialists at the ARSC DSRC and a select group of DSRC HPC users conducted extensive capability tests, application system testing, and benchmarking as part of the complicated installation process required to bring a new supercomputer online. Pingo's acceptance testing was completed January 8, 2009.

As part of the HPCMP Capability Applications Project (CAP), three competitively chosen projects were the first to access Pingo. CAP allows users with mature

Photo of geophysical phenomenon called Pingo

code to tackle large computational campaigns. This way, both the research itself as well as advances in scaling codes can be accomplished simultaneously.

Dr. James Freericks, Defense Advanced Research Agency (DARPA) and Georgetown University, was one of three HPC users competitively chosen to run code on Pingo during its 3-month acceptance period. He continues to work on Pingo to model the quantum mechanical properties of potassium-rubidium molecules in an optical lattice. Freericks is modeling a new form of matter: quantum dipolar gases (in addition to solid, liquid, gas, and plasma), which he expects to lead to a new type of quantum computer. He expects that understanding of the new type of matter it forms will lead to new ways to perform quantum computation. The quantum dipolar materials hold promise to be more fault tolerant. This basic research may also lead to improved sensors, similar to those currently used in medical scanning, electronic parts evaluation, and oil exploration.

Use of Pingo is appropriate for solving large compute and memory-intensive parallel jobs. The Cray XT5 runs a lightweight Linux operating system (OS) on the compute nodes. Many features that are nonessential to an HPC workload have been removed from this OS. The net result is greater amounts of memory available to user applications and reduced run-time variability. Pingo runs the Cray Linux Environment and the PBS batch system scheduler. User documentation is available at http://www.arsc.edu/support/howtos/usingxt5.html

"We're extremely excited to be able to deliver Pingo and deliver some of the most exciting supercomputing technology in the world for a lot of very important problems for our Nation, for the Arctic region, as well as our military," Ungaro said.

A formal dedication ceremony for Pingo took place in March, on one of the snowiest days of the decade in Fairbanks. Cray Inc. President and CEO Peter Ungaro attended the dedication and noted that Pingo represents another in a long line of Cray supercomputer technology for the ARSC DSRC. A webcast of the event is available at www.arsc.edu/news/pingo_video.html.

Dedication ceremony for Pingo

"ARSC is an especially good place to install our advanced systems because of the facilities preparation and deep-technology expertise the ARSC staff bring to systems hardware and software," Ungaro said.

The first supercomputer at the ARSC DSRC was a 4-processor Cray YMP named Denali, with peak performance of 1.33 gigaflops. That means that in the 16 years since the Center opened, processing capability has increased by a factor of about 25,000.

U.S. Army Engineer Research and Development Center DoD Supercomputing Resource Center

Dr. Robert S. Maier, Acting Director

As Brad Comes said in his introduction, we value the unique character of our individual Centers. We are proud to serve our respective service agencies and our parent research laboratories, and publications such as the *Resource*, *Link*, *Wright Cycles*, and the *Navigator* demonstrate that commitment.

We are now engaged in an experiment to produce a joint publication that demonstrates our commitment to the unified management and governance of our six Centers. Each Center is asked to put aside, for a time, their individual publications, and to join with the others in addressing a greater audience. This publication, HPC Insights, is an expression of our commitment to learn from each other and use our joint strength to better accomplish our missions.

The U.S. Army Engineer Research and Development Center is the proud host of a unique DoD Supercomputing Resource Center, otherwise known as the ERDC DSRC. We publish a semiannual newsletter called the ERDC *Resource*. Over the years, the *Resource* has documented the changes in HPC technology available to the user community (see inset on this page).

Dr. Robert S. Maier
Acting Director, ERDC DSRC

The ERDC DSRC is currently a Cray shop, hosting an XT3 (Sapphire) and an XT4 (Jade), pictured on this page in camouflage. These bad boys have made our reputation as a good place to get big jobs done.

We are about to become a mixed shop with the imminent arrival of an SGI Altix ICE 8200 (Diamond), beautifully rendered by Miguel Valenciano and Kevin George. Our journeyman director, Dave Stinson, describes Diamond in this section. The rendering shows Diamond in its new setting, which is described by our infrastructure czar, Greg Rottman, also in this section.

ERDC Supercomputers

- 1993 – Army Supercomputer Center rechartered as first DoD High Performance Computing Center. Cray Cray C90 and Cray Y-MP
- 1997 – IBM SP, Cray T3E, and Silicon Graphics 2000
- 2000 – IBM Power3 SMP
- 2001 – SGI Origin 3800 and Compaq Alpha Server SC40
- 2002 – Compaq Alpha Server SC45
  Upgrade to Cray T3E – largest unclassified T3E in the world
- 2003 – Two SGI Origin 3900s and Cray X1
- 2005 – Cray XT3, ERDC computing ability at 26 teraflops
- 2007 – Cray XT4, adding in 80 teraflops
- 2009 – SGI Altix ICE 8200

Cray XT4

ERDC Information Technology Laboratory

A New Diamond Coming to ERDC

By David Stinson

The ERDC DSRC is pleased to announce SGI as the winner of the TI-09 competition at the ERDC DSRC. The 15,360-core Altix ICE 8200 is scheduled for delivery in July 2009 and represents SGI's third generation of advanced x86-64 servers and clusters based on 2.8-GHz Nehalem-EP quad-core processors and utilizing Intel's next-generation QPI memory architecture. With two quad-core processor sockets per node operating as an 8-way SMP node and 4 GB of memory per core, each core can share 32 GB of memory. The InfiniBand interconnect switch fabric utilizes a 5D Hypercube topology. All nodes share a common 972-TB (raw) RAID storage system configured with the Lustre parallel file system.

The new machine will be named "Diamond" in keeping with the theme of precious minerals at the ERDC DSRC, and it will employ a significant amount of green technology. Water-chilled doors will remove most of the rack heat, significantly reducing the need for ambient air cooling. The implementation of high-efficiency power supplies promise significant cost savings over the life cycle of the machine. Other features include optical InfiniBand cables for inter-rack connectivity, overcoming space and weight limitations of copper cables and allowing 20 GB/s double data rate (DDR) at distances up to 100 meters with a much lower bit error rate.

Diamond's 172 peak TFLOP rating places it in a tie at number 12 in the world on the November 2008 Top 500 list based on theoretical peak performance. It is expected to be available for production work in October 2009 and will be the first machine to occupy the new 10,000 sq ft computer center at the ERDC DSRC.

Rendering of the SGI Altix ICE 8200 named Diamond

Sustaining Critical Infrastructure

By Greg Rottman, ERDC DSRC Assistant Director

Construction site on April 29, 2009

In preparation for the arrival of the new SGI Altix ICE 8200 supercomputer, the support infrastructure is growing. Just north of the ERDC Information Technology Laboratory, things are moving fast to enhance the infrastructure that will provide the power, cooling, and network connectivity for the Technology Insertion 2009 (TI-09) system and beyond.

The new infrastructure will be capable of supplying up to 8 MW (106 watts) of uninterruptible power when fully configured. Initially, there will be 4 MW of generator and battery backup support. Power will be distributed from the UPS at 13,800 V and will be down converted to 480 near the load, thus reducing the copper wire needed to distribute and also decreasing power losses. The majority of the electrical infrastructure will be outdoor equipment, reducing the cooling load on the chillers by allowing heat to escape into the atmosphere.

One thousand tons of chilled water capacity will have the ability to expand in two 500-ton increments to a total capacity of 2000 tons. The chillers will have variable controllers to run at the minimum of energy consumed to meet the cooling needs of the systems.

Additionally, the ERDC DSRC will enhance its network by upgrading the single-mode and multimode fiber infrastructure to accommodate the anticipated increase in data traffic and provide for the planned future capacity.

Finally, what good is all the power cooling and connectivity if there is not a place to put the new system? The new high performance computing machine room is rapidly maturing toward its final shape. Its overall dimensions are 40 ft wide by 250 ft long, providing 10,000 sq ft of computer space. It will have a raised floor 4 ft deep, providing adequate space for power, cooling, communications, and air movement. The floor will be able to support 625 lb per sq ft or 2500 lb per 2 ft × 2 ft floor tile.

Larger capacity UPS, more cooling capability, larger capacity network, and expanded capacity network translates to the ability to support larger capacity computer systems.

ERDC Hosts HPCMP S/AAA and Security SIG Meetings

By Phil Bucci, ERDC S/AAA and HPC Allocations Officer Support

Over 60 people attended the 2009 annual Service Agency Approval Authority (S/AAA) meeting championed by Cathy McDonald of the HPCMP (March 2-3). ERDC coordinated meeting and hotel accommodations. The agenda included new S/AAA orientation and training, presentations from the six DSRCs, plus recent acquisition and technology insertion plans for SGI ICE 8200 clusters.

The HPCMP Security Implementation Group (SIG) utilized the ERDC Information Technology Laboratory facilities for discussion and presentations to coordinate implementation of DoD Security policies (March 3-4).

The location and timing for these meetings offered a forum where both groups could address a specific change affecting all HPC users. The HPCMP is phasing out SecurID equipment and introducing new devices and processes for kerberized system access. Key SIG members visited the S/AAAs to deliver a tutorial on Common Access Card (CAC) login registration, PKINIT, H-tokens, and V-tokens. The S/AAAs were invited to attend the SIG presentations for further explanation and user migration plans for the new secure access methods.

The S/AAA and SIG Working Groups are members of the HPCMP Community www.hpcmo.hpc.mil.

Maui High Performance Computing Center DoD Supercomputing Resource Center

By David Stinson, Acting Director

David Stinson
Acting Director, MHPCC DSRC

In a short time, one comes to appreciate the contributions of the Maui High Performance Computing Center (MHPCC) to the DoD High Performance Computing Modernization Program (HPCMP). Sure, the scenery is unparalleled, and the staff is extremely competent and friendly; but the Center is coming into its own as a new DoD Supercomputing Resource Center (DSRC) and continues to play a significant role within the HPCMP.

Since its inception in 1993, the MHPCC mission has grown to provide substantial support to the research, science, and warfighter communities (32,000,000 CPU hours in Fiscal Year 2008) in addition to support for the Air Force Research Laboratory Directed Energy Directorate Maui Space Surveillance System. To provide the best resources to users, the MHPCC DSRC is in the process of planning upgrades to its current 5120-processor Dell PowerEdge 1955 system, Jaws. If all plans are successful, the Center could experience as much as a fourfold increase in computational capability before the end of the year. The Center is also investigating technology that will help it to go "green" and contain costs while continuing to provide top-quality service to its customers.

Any suggestions for improving the MHPCC DSRC services are welcome. The Center is committed to providing the best support possible to its customers.

High Performance Computing Software Applications for Space Situational Awareness

By Dr. Chris Sabol, Air Force Research Laboratory, and Bruce Duncan, MHPCC DSRC

The High Performance Computing Software Applications Institute for Space Situational Awareness (HSAI-SSA) is performing its applications software development and optimization, continuing work since its competitive selection by the DoD in Fiscal Year 2005. Work during this past year has continued on multiple subprojects. The HSAI-SSA is one of only nine DoD Institute projects that were selected by the Deputy Undersecretary of Defense (Science and Technology) to focus on and use advanced computational science and high performance computing (HPC) to accelerate solving the DoD's highest priority challenges and make important advances in research, development, test, and evaluation. HSAI-SSA is the only DoD Institute project focused on Space. Its work is performed under the auspices of the HPCMP. HSAI-SSA is led by the Air Force Research Laboratory Directed Energy Directorate (AFRL/RD), and an HSAI-SSA onsite Director is located at the Air Force Maui Optical & Supercomputing Site (AMOS) at Maui. HSAI-SSA core competencies include image enhancement, astrodynamics, nonresolvable satellite characterization, data integration, and HPC.

HSAI-SSA coordinates directly with the research, development, acquisition, and operations and support communities to develop and transition HPC software applications, striving to provide game-changing SSA capabilities. The effort performed by the Institute is overseen by the DoD HPCMP Director, who chairs a Board of Directors (BOD) that is comprised of other senior DoD/SES/General Officer officials. This writing focuses on work performed during 2009 to continue improving space surveillance image enhancement software. These applications and projects include the Physically Constrained Iterative Deconvolution (PCID) image-processing software and Advanced Speckle Imaging Reconstruction Environment (ASPIRE) software. The software applications improvements that the HSAI-SSA has made continue to enhance warfighter capabilities and knowledge and are increasing the role of HPC in SSA.

Introduction

SSA forms the foundation for DoD space superiority. Problem complexity and needs for accuracy and timeliness are continuously growing each year as highlighted in Figure 1. Today's decent knowledge of SSA must transition to a future with even more accurate knowledge, mission identification, and status for decision makers. The mission of the HSAI-SSA is to address the growing needs of SSA stakeholders by developing HPC software applications for SSA. The Institute collaborates directly with experts and end users in the Joint Space Control area and Joint Functional Component Commander (JFCC) for Space and Global Strike to identify those "leverage points" at which HPC can be applied to make a strategic difference in how the mission is performed. The HSAI-SSA has had another successful period of performance during 2009, working with multiple software applications, which span major disciplines including image enhancement, astrodynamics, non-imaging space object identification, and data integration. Two of the HSAI-SSA software applications and projects are highlighted below.

Project Status

HSAI-SSA work has continued, including further development of two software applications that respond to HSAI-SSA Strategic Goal 2 (Image Enhancement) objectives. These applications are running on the MHPCC DSRC Cray XD-1 Linux supercluster named Hoku (star in the Hawaiian language). The PCID image enhancement software and ASPIRE tool for space surveillance have been optimized for faster and more precise operations, while also incorporating functional enhancements needed by the Maui Space Surveillance System (MSSS) image analysts and other users.

PCID is an advanced image enhancement processing application that removes atmospheric and system blurring from observed and measured telescope sensor data to produce high-resolution images. PCID finds the best estimates of the object and point spread functions (PSFs) that maximize the probability of the data given for the object and PSFs. PCID minimizes the error between an estimate of the measurement given for the current object, and PSF estimates (and can employ multiple available minimizers) the real measurement, with the error being minimized using a conjugate gradient search routine. One iteration is defined as finding the minimum of the PCID cost function along a single conjugate gradient search direction. The primary PCID processing steps include algorithm initialization; determination of the initial values for the variables to be estimated (including object pixel intensities, and either PSF pixel intensities or Zernike coefficient values); utilization of these initial values to generate an estimate of the measurement; using the conjugate gradient technique, finding a new set of estimates of the object and PSF values; and then continuing to convergence. AFRL scientists have previously used an information-theoretic analysis to determine that PCID meets theoretical performance limits. While PCID is a nonlinear algorithm that employs significant technical advantages and provides benefits, it necessitates the use of effective HPC and software engineering and optimization.

PCID software development work and successes achieved during 2009 with the latest version of PCID have resulted in increased efficiency of this code. PCID is currently comprised of more than 15,000 source lines of high-order language code (FORTRAN and C++) with 79 files. During 2009, the collective PCID team has continued its development and testing regimen, including alpha and pre-beta testing of PCID Version 8.0. The PCID Version 8.0 baseline code and associated different compilation directives capabilities include the following:

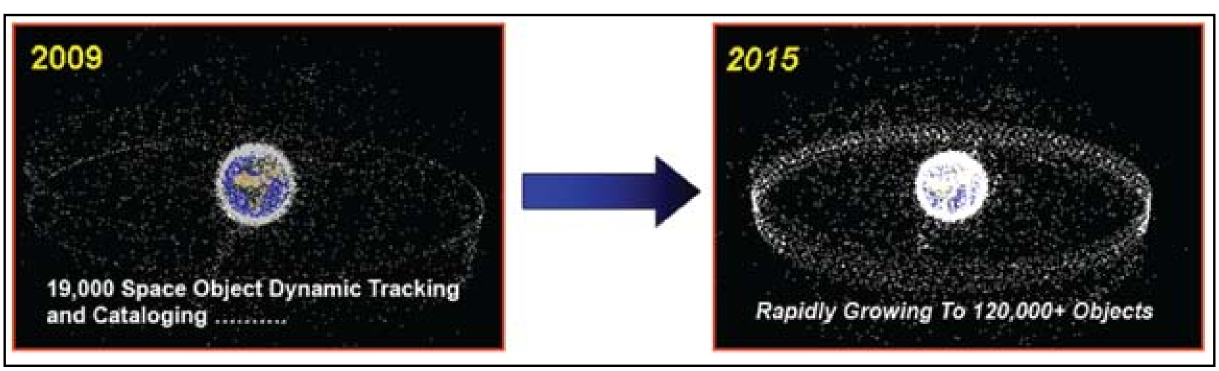


Figure 1. SSA mission area is a complex one that continues to escalate and necessitates supercomputing to meet the challenges presented

- Parallel FFT software and tags.
- FFTW software.
- Intel MKL software.
- Instrumentation software.
- Publish/Subscribe software.

PCID executables include the following:

- Serial PCID binary executable.
- Parallel Master/Worker binary executable.

PCID runs and testing have been performed on four computing platforms at different locations: Hoku, Polaris, Jaws, and Phantom (with different AMD Opteron and Intel clusters). The runs were made using data from three MSSS sensors: GEMINI, AURA, and AOVIS in various computing environments, while also supporting improvements in instrument calibration, utilities, and image frame ensemble generators software. To date, the team has achieved an approximately 15-times additional speedup to augment other speedups previously realized since the inception of the PCID project. Staff members are continuing to analyze minimizers to aid in better image-processing quality and performance in given situations and data set types.

HSAI-SSA work was also performed this year on the next release of ASPIRE software. ASPIRE is being developed for use with PCID and other image-processing software applications by AMOS image analysts and other DoD users. The current version of ASPIRE is comprised of more than 62,500 source lines of code. This includes a mix of Scripts comprised of Dynamic Java, C, Shell Scripts, and Perl. It also includes Web software comprised of HTML, CFML, Dynamic Java Servlets, CGI, PHP, Cascading Style Sheets, and JavaScript. ASPIRE provides an intuitive, Web-based user-friendly graphical user interface (GUI), using default algorithm parameters, and the capability to submit/resubmit and manage multiple processing job submissions asynchronously to select supercomputers. ASPIRE allows users to easily visualize the image-processing results and related information, and it provides many other functions for use by AMOS researchers/developers and image analysts.

Staff continued with developmental test efforts and completion of key calibration, image-conversion utilities, and image preview capability and log file viewers for ASPIRE Version 2.0 on the Hoku and Polaris platforms with MSSS image analysts, closing out tasks for initial performance enhancements in ASPIRE, the Ensemble Generator, and the ability to run PCID V8.0 using ASPIRE (see Figure 2). ASPIRE is also being

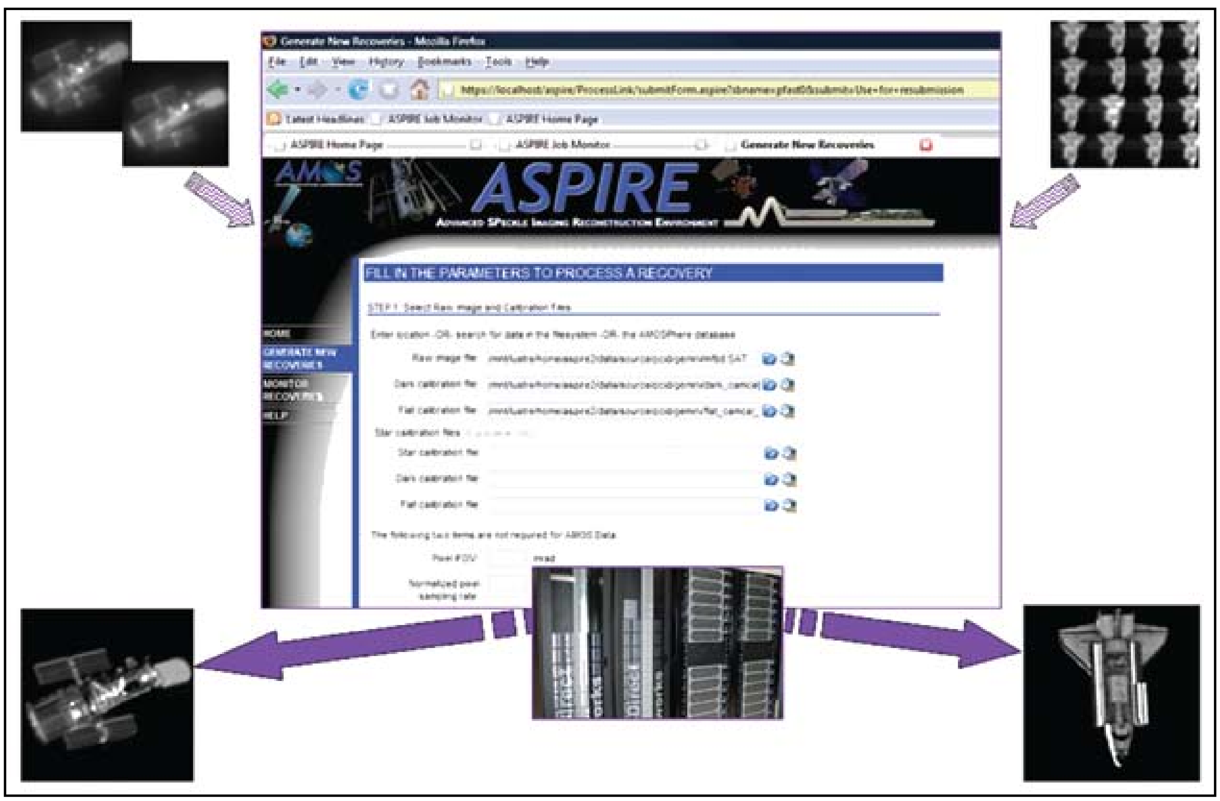


Figure 2. PCID/ASPIRE image-enhancement processing accepts data from sophisticated electro-optical systems and capitalizes on HPC and advanced software to produce high-resolution imagery of space objects utilizing a user-friendly interface and job management system

closely integrated with AMOSphere, which is the data repository for the AMOS/MSSS organization and the location from which the vast majority of data that are generated and processed by the AMOS site is disseminated to customers. One example of this integration is that users who log into ASPIRE will automatically be provided login privileges to AMOSphere and vice versa. Another example is that when a user logged into ASPIRE needs data from AMOSphere, there will be links on the ASPIRE pages to AMOSphere. Additionally, when users who are logged into AMOSphere desire to process the data they are viewing, there is a link to the ASPIRE image-processing page.

Conclusion

Near-term goals for PCID/ASPIRE are to pass beta testing and MSSS user-acceptance testing and be available for daily use in the MSSS, contributing to the space awareness mission for the United States Strategic Command (USSTRACOM) and other key organizations. HSAI-SSA is hardening PCID from a research-grade code to operational software with orders-of-magnitude speed increases and has developed ASPIRE to support it. PCID and ASPIRE have already demonstrated the ability to produce better images than the software and tool currently in use for MSSS daytime imaging operations and products. The HSAI-SSA continues to achieve good progress in developing HPC SSA applications for the warfighter. The team is continuing to increase performance on other projects and applications (as highlighted in Figure 3). HSAI-SSA continuously employs proven and improved agile software development and engineering practices to demonstrate portability, scalability, speedup, optimization, and performance improvements. HSAI-SSA is focused and is bringing the appropriate technical expertise together with the appropriate computing resources and techniques at the critical time to develop, test, and transition HPC software applications into the critical SSA arena for national defense.

Acknowledgments

The authors express added appreciation and thanks to the Air Force Research Laboratory Directed Energy Directorate; Air Force Maui Optical and Supercomputing (AMOS) site and scientific staff; the HSAI-SSA staff and associates/partners, including the AFRL Information Directorate; and the MHPCC DSRC contractor team.

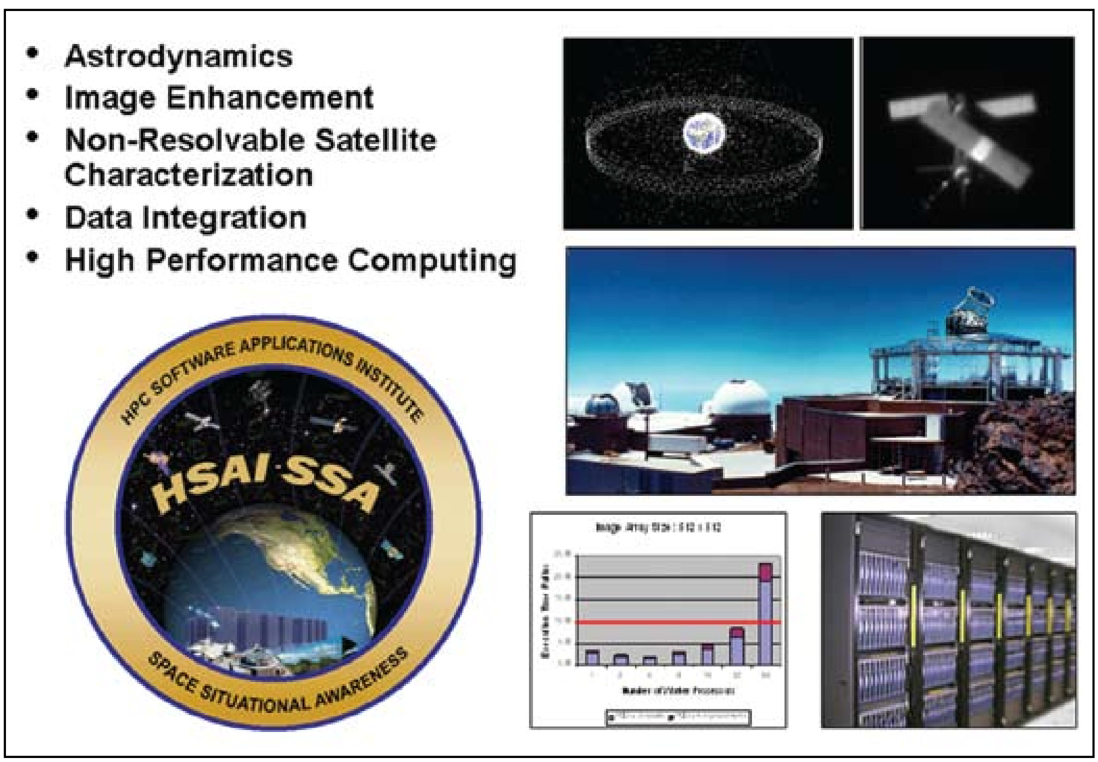


Figure 3. HSAI-SSA has several key competencies that address SSA needs and involve many engineering disciplines including orbital mechanics, statistical estimation, global tracking, imaging operations, space object identification and characterization, multispectral sensors, and modeling and simulation

Navy DoD Supercomputing Resource Center

By Tom Dunn, Director


Tom Dunn
Director, Navy DSRC

In 1994, the Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC) was established as one of four such MSRCs in the DoD High Performance Modernization Program (HPCMP) after several years of service to the Navy as the Primary Oceanographic Prediction System (POPS) Center. Since then the Center has provided premier supercomputing resources and expertise to its HPCMP and Navy operational users, rendering it one of the top 14 sites in the world in overall supercomputing capability since the Top500 list-keeping began in 1993.[1]

In 2007, the Center moved under the auspices of the Commander, Naval Meteorology and Oceanography Command (CNMOC) and in January 2009 received a new moniker as one of six designated DoD Supercomputing Resource Centers: Navy DSRC. A retrospective of the Center's first 15 years of service is available in the following article.

Spring of 2009 finds the Navy DSRC's two new systems, the Cray XT5 EINSTEIN and the IBM P6 DAVINCI, settling into service. At press time, Simultaneous Multi-Threading (SMT) is being enabled on DAVINCI to allow users to increase their code performance by up to 20 percent. Last fall the Center's archive server, the Sun M5000 named NEWTON, seamlessly took over the considerable workload borne by Jules and Vincent (Sun F12000s) for more than 6 years. It is expected that both IBM P5+ systems BABBAGE and PASCAL will transition to the PBS Pro scheduler at the beginning of Fiscal Year 2010. And as the HPCMP overall moves toward a more collaborative environment, the Navy DSRC staff has joined those at the other Centers in strengthening cross-center team efforts to provide the most robust high performance computing resources possible to users throughout the program.

For 15 years, the diverse needs of the Navy meteorological and oceanographic community and the HPCMP Research and Development (R&D) users have been the primary focus of this Center's day-to-day and forward-looking activities. The Navy DSRC is pleased to continue that user-centric focus and commitment now and in the years to come.

[1] Top500 Supercomputing Sites, http://www.top500.org/topsites/results.

Navy DSRC Retrospective: 1990 – 2009

By Christine Cuicchi, Computational Science and Applications Lead, Navy DSRC

Since 1990, supercomputing has been an integral part of the Navy's Meteorology and Oceanography (METOC) community time-critical, or operational, forecasting and nowcasting needs. Over the years, this function at Stennis Space Center has transitioned from a small, localized Center serving the needs of a small user base to a world-class supercomputing center hosting large-scale high performance computing, networking, and storage in support of the DoD warfighter and the High Performance Computing Modernization Program.

1990: Charting a Path to Excellence

Established in 1990 at the Naval Oceanographic Office (NAVOCEANO) at Stennis Space Center, Mississippi, the Primary Oceanographic Prediction System (POPS) center was anchored by an eight-processor Cray YMP and an aggregate peak computational capability of 3.2 gigaflops (GFlops). At the time, the Center was connected through SURANet to the outside world. POPS served to support the Navy's research and development (R&D) and high performance computing (HPC) requirements, providing a foundation for the extensive METOC modeling and time-critical operational forecasting system in place today.

1992-1993: Modernizing the Warfighter

By 1992, the advantages of combining the independently developed HPC resources of the various service branches of the DoD were evident, and the DoD HPC Working Group began the monumental task of unifying these resources. In 1994 the NAVO Major Shared Resource Center (MSRC) became one of four Major Shared Resource Centers to be established under the auspices of the DoD HPC Modernization Program (HPCMP).

Support of the METOC community's 24-7 forecasting requirements put the Center in a unique position within the HPCMP, as near-continuous system availability remained a critical element in the Center's planning and development.

1994-1999: Diversifying HPC Capabilities

A wide array of HPC platforms graced the Center computer floor throughout the period, including Cray SV1 and T932 platforms, SGI Origin 2000s, and a Sun HPC10000 computational server. By the end of 1999, users had amassed 160 TB of data on the Mass Storage Archive Systems. The Center also readied itself for Performance Level 3 upgrades, including an upgrade of its Defense Research and Engineering Network (DREN) connection from an OC-3 connection (155 megabits per second, or Mpbs) to an OC-12 connection capable of carrying 622 Mpbs.

Cray SV1 Poseidon/Zeus/Trident/Athena

2000-2001: Cresting the Wave of HPC Technology

A 33 percent expansion of the Cray T3E, SEYMOUR, and the installation of the 1336-core, 2.0-teraflop (TFlop) IBM P3 named HABU were Center highlights in 2000. Existing SGI O2Ks were merged and expanded to a single 256-core system, and two new nodes were added to the existing Cray SV1 shared memory nodes, making the Cray SV1 the largest parallel vector machine offered at the Center.

The Center had enjoyed a 1000-fold increase in aggregate peak computing capability—3.2 TFlops—in the 10 years since POPS was established with 3.2 GFlops capability. Two Sun E10000 storage servers, JULES and VINCENT, were added to tackle the data storage requirements of an expanding user base. HABU's debut in 4th place on the Top500.org list of most powerful supercomputing systems in the world remains the highest mark attained by any HPC system in the entire DoD HPCMP.

IBM P3 Habu, 2.0 TFlops

2002: A Decade of Progress

The year marked 10 years since the DoD HPC Working Group, the HPCMP's predecessor, determined and solidified the plans for a DoD-wide HPC environment. It also saw the Center's first foray into the Technology Insertion (TI) process, whereupon HPCMP Centers greatly enhance computing capabilities either through the upgrade of existing systems or the acquisition of newer HPC offerings. A hardened Remote Storage Facility was completed to accommodate users' rapidly growing data storage needs. The 1184-core IBM P4, MARCELLUS, one of the largest P4 systems in existence at the time, was installed and debuted in 6th place on the Top500 list. The Center now boasted an aggregate peak computing capability of 8.3 TFlops.

2003: Solidifying Support of the Warfighter

MARCELLUS was upgraded to a total of 1408 cores, garnering a 20 percent increase in peak capacity. This additional power was paramount in providing direct support to the Iraqi Freedom effort via NAVOCEANO operational modeling and forecasts, which assisted DoD efforts to determine appropriate shipping channels for supply ships carrying military equipment and humanitarian supplies into Iraq, and support for ground troops and Special Operations Forces.

The Center was recognized as one of the Top Ten Most Powerful HPC Sites from 1993-2003, joining the likes of Lawrence Livermore National Laboratory, Pittsburgh Supercomputing Center, Oak Ridge National Laboratory, and Japan's Earth Simulator Center.

2004: Bringing Disaster Response to Bear

TI-04 upgrades included installations of the 2944-core IBM P4+, KRAKEN, one of the most powerful systems ever purchased from IBM at the time, and the 512-core IBM P4+, ROMULUS. The two systems brought an additional 24 TFlops aggregate peak computing capability to the Center. With a combined 30 TFlops of computing capability in-house, the Center upgraded its DREN connection from an OC-12 (644 Mbps) to an OC-48 connection (2.4 Gbps).

The post-9/11 world had highlighted a need to protect users' data from disastrous events both natural and man-made. The HPCMP's response was to develop comprehensive Disaster Recovery plans and facilities, in which the Navy DSRC became a driving force. Disaster Recovery sites became operational in November 2004, storing second and third copies of data from six HPCMP Shared Resource Centers.

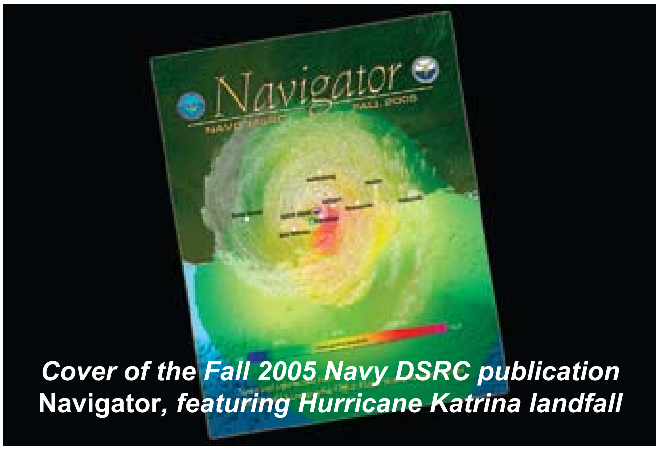
The Navy DSRC was again called upon to support the warfighter in near real-time, providing highly available HPC capabilities to NAVOCEANO oceanographers who were asked to assist the rescue effort after the Indonesian tsunami of December 26, 2004. The oceanographers used a number of computational models to forecast conditions that affected rescue efforts and recovery operations in Sumatra, Sri Lanka, and the Maldives, including survivor and flotsam drift, water temperatures, and surf conditions.

2005: Rising to the Challenges--Expected and Unexpected

The Capabilities Applications Projects (CAP) effort was established by the HPCMP to allow select computational applications to run on significant portions of HPC systems, allowing the solution of large-scale, meaningful problems in a short time. The first such CAP efforts were hosted on KRAKEN.

In late August, the eye of Hurricane Katrina made landfall just south of Stennis Space Center and then passed directly over the Navy DSRC facilities. While the Center itself sustained little damage, all DSRC staff members were affected personally to some degree, as numerous staff suffered significant damage to their homes. Thanks to the work of dedicated Center staff prior to, during, and after the storm, no significant system outages or data loss occurred, and satellite-based connectivity to DREN was quickly established.

The Navy DSRC continues to be proud of its ability to remain in service to its users after a disaster of such magnitude and is humbled by the concerned response received from many in the HPCMP community.



Cover of the Fall 2005 Navy DSRC publication Navigator*, featuring Hurricane Katrina landfall*

2006-2007: Growing Steadily in Advanced Technologies

The Center spent much of 2006 preparing facilities and support staff for the installation of TI-06 systems: a 3072-core IBM P5+ named BABBAGE for unclassified use and the 1920-core IBM P5+, PASCAL, for classified computing. In the spring of 2007, both systems were put into production and pushed the Center's aggregate peak computational capability to 61 TFlops. As 2007 drew to a close, so did the TI-08 acquisition process, which would herald the return of a Cray platform to the Center's IBM-dominated HPC arsenal.

2008: Building a Computational Powerhouse

Much of the first half of 2008 was dedicated to significant facilities upgrades in preparation for the arrivals of DAVINCI, a 4768-core, 80-Tflop IBM P6; EINSTEIN, a 12,872-core, 117-TFlop Cray XT5; and NEWTON, a Sun M5000 archive server. EINSTEIN and DAVINCI

IBM P6, Davinci, 90 TFlops

debuted in the Top500 at 26th and 38th, respectively. KRAKEN and ROMULUS retired after several years of reigning as two of the most venerable platforms in the HPCMP. Despite this retirement, the Center's aggregate peak computing capability had grown to 233 TFlops—quadruple its previous capability.

2009: Continuing a Commitment to Our Users

The Navy DSRC ushered in the new year under its new name, and soon afterwards, DAVINCI was increased to a powerful 5312 cores and 90 TFlops. With an aggregate 243 TFlops of peak computing capability, a robust OC-48 DREN connection, expansive first copy data storage and Disaster Recovery facilities, and well-tested recovery and contingency plans in place, the Navy DSRC stands poised to provide exceptional high performance computing capabilities to the demanding computational problems of our users.

Cray XT5, Einstein, 117 TFlops

Using Simultaneous Multi-Threading (SMT) Under PBS on DAVINCI

By John Skinner, Navy DSRC

Simultaneous Multi-Threading(SMT) is an AIX feature available for Power5- and Power6-based systems. Under SMT, the P6 (Power6) doubles active threads on a processor core by implementing a second onboard "virtual" processor enabled in the chip architecture. The basic concept of SMT is no single process can use all processor execution units at the same time, so a second thread should be able to use unused cycles on the same physical processor core.

DAVINCI has 32 physical cores per compute node. Since the nodes have SMT enabled, it appears as if 64 virtual CPUs are on each node. Running 64 tasks per node should generally provide good efficiency for pure MPI codes. However, we recommend testing your own application on 32 up to 64 tasks per node (assuming per-task memory usage allows 32 or more tasks to all fit in local on-node memory - currently about 54.4 GB). If your program is hybrid, MPI + OpenMP, experiment with various combinations of task counts and thread counts so that you total 64 threads per node, rather than 32.

To use SMT, no source code changes are required in Fortran/C/C++ codes, but we recommend various changes to your PBS scripts, as described in this article. With these changes, you may be able to boost performance by 20 percent or more on some codes. It is often possible to get at least some speedup by exploiting SMT, and little extra effort is required to try using SMT on DAVINCI.

An MPI-only PBS job running 128 tasks over four 32-way nodes can be modified to utilize SMT on two 32-way nodes by changing the "-l select" PBS parameter (see below), or you can continue to use 4 nodes with SMT by specifying that job starts 256 tasks (64 per each node), assuming your MPI application is scalable. This latter method may be preferable if total wallclock time used is a primary consideration.

1. Run 128-task MPI executable on four 32-way SMT-enabled nodes (32 tasks per node) with task-affinity layout of tasks controlled via local "launch" tool:

```
#!/bin/ksh
#PBS -l select=4:ncpus=32:mpiprocs=32
#PBS -lplace=scatter
#PBS -lplace=excl
#PBS -q standard
#
export MP_SHARED_MEMORY=yes
export MEMORY_AFFINITY=MCM
export UL_MODE=PURE_MPI
export UL_TARGET_CPU_LIST=AUTO_SELECT
export MP_EUILIB=us
export MP_DEVTYPE=ib
export MP_LABELIO=yes
#
cd $WORKDIR
/usr/bin/poe /site/bin/launch ./myMPI.
exe
```

2. Run same 128-task MPI executable on two 32-way

SMT-enabled nodes (at 64 tasks per node) with task-affinity layout controlled via local "launch" tool:

```
#!/bin/ksh
#PBS -l select=2:ncpus=64:mpiprocs=64
#PBS -lplace=scatter
#PBS -lplace=excl
#PBS -q standard
#
export MP_SHARED_MEMORY=yes
export MEMORY_AFFINITY=MCM
export UL_MODE=PURE_MPI
export UL_TARGET_CPU_LIST=AUTO_SELECT
export MP_EUILIB=us
export MP_DEVTYPE=ib
export MP_LABELIO=yes
#
cd $WORKDIR
/usr/bin/poe /site/bin/launch ./myMPI.
exe
```

3. Compare results and turnaround times. If code runs correctly under SMT at 64 tasks per node using half as many nodes and walltime is not more than 100 percent higher than running 32 tasks on twice as many nodes, you will burn less of your allocation hours. This is because walltime charges on DAVINCI will be charged at "walltime_ used*32*nodes_used" whether you run 64 tasks or 1 task on each node assigned to your PBS job. You may also increase your chances of running jobs more often and having more jobs become backfill candidates, since less nodes would be needed for each run.

An SMT-aware MPI+OpenMP hybrid job that runs 4 MPI tasks on four nodes, with each MPI task spawning 64 OpenMP threads on a node, is listed below. Task affinity for this code is controlled by the local "launch" tool.

Note:

Under AIX 5.3, there is a known defect that causes performance problems in hybrid applications when application reads stdin redirected from a file, e.g.: poe myHYBRID.exe < namelist. Workaround is to set MP_STDINMODE=0 in job's environment. This is important for best performance under SMT.

```
#!/bin/ksh
#PBS -lselect=4:mpiprocs=1:ncpus=64:ompt
hreads=64
#PBS -lplace=scatter
#PBS -lplace=excl
#PBS -q standard
#
export MP_SHARED_MEMORY=yes
export MP_EUILIB=us
export MP_DEVTYPE=ib
export MP_LABELIO=yes
export OMP_NUM_THREADS=64
export UL_MODE=HYBRID
export UL_TARGET_CPU_LIST=AUTO_SELECT
#
cd $WORKDIR
/usr/bin/poe /site/bin/launch ./myHY-
BRID.exe
```

Summary

SMT should help achieve better throughput for parallel PBS jobs and provide better performance for total walltime charged to a job.

Each DAVINCI compute node has about 54.4 GB of local user memory. If your code cannot fit more than 32 tasks in this memory on a node, you will not be able to run more than 32 tasks per node, but you can still take advantage of SMT and task affinity by using the launch tool.

More information about SMT, setting up PBS jobs to use task affinity, and a general AIX optimization overview are in the document "How to Get Your Program to Run Faster on AIX Clusters" at http://www.navo.hpc.mil/usersupport/IBMP6_SMT_Training/. In particular, please read sections "Simultaneous Multithreading" and "Process and Thread Affinitization" for more information.

Example PBS scripts and compilation examples are located on DAVINCI under the /site/HPC_Examples directory.

Data Analysis and Assessment Center

By Michael Stephens, UDAAC Lead, and John Vines, CDAAC Lead

Michael Stephens
UDAAC Lead

John Vines
CDAAC Lead

The Data Analysis and Assessment Center (DAAC) helps users of the Department of Defense High Performance Computing Modernization Program (HPCMP) perform the alchemy that turns raw simulation data into the pure gold of understanding and insights.

DAAC resources are available at no cost to HPCMP users. These resources are not just the hardware and the supported software dedicated to data analysis; they also include the most valuable resource of all—the data analysis expertise of the classified and unclassified DAAC staffs. The CDAAC, which is collocated with the Army Research Laboratory (ARL) DoD Supercomputing Resource Center (DSRC), is responsible for the classified data analysis needs, while the UDAAC responsibility covers all the unclassified data analysis within the program. The UDAAC is physically homed alongside the U.S. Army Engineer Research and Development Center (ERDC) DSRC.

The uses of data analysis or data visualization vary widely among the large HPCMP user community. Consequently, the DAAC offers a wide range of data analysis services. These services are characterized by three broad categories: community, collaborative, and custom.

The community service is primarily for researchers that are either new to visualization and wish to get started with minimal effort or want to learn more about how to effectively visualize their data. The embodiment of this service is the DAAC Web site, daac.hpc.mil. There users will find a wealth of information about supported software tools, such as ezVIZ, EnSight, and ParaView. Additionally, they will find an ever-growing number of tutorials, how-to guides, and a gallery of past projects that might get them thinking about different ways of presenting their data. Lastly, but perhaps the most important of all, they will find a way of directly contacting DAAC staff members to ask questions and get help.

The collaborative service is for people who want one of the DAAC staff members to help in visualizing their data. At this level, a staff member will work closely with the researcher in deciding on what he/she wants to communicate with the visualizations, recommending tools and methods to produce the visualizations, and generating a work flow so that the researcher can generate these on his/her own. This is geared for both day-to-day visualizations that scientists and engineers might use as a means of validating their simulations and for producing images and animations that are suitable for an audience of peer scientists, such as shown at conferences when presenting their work.

Finally, the custom service is for users that require high-quality images or animations of their data for which they also need to show the data in context or natural environment. For example, rather than just showing the flow data in a combustion chamber, the user first shows an M1A1 Abrams tank, peels the armor off to see the tank's engine, and finally moves into the combustion chamber itself before showing the visualizations of the simulated data. Often this level of service requires conceptual images or animations to demonstrate an idea or to explain the research problem to an audience that is highly interested yet possibly nontechnical. Think Discovery Channel. These kinds of visualization are very effective for high-level briefings to upper management and sponsors. Users can take a look at the DAAC gallery, daac.hpc.mil/gallery, for some examples of what these custom visualizations are like.

Researchers Investigate Ground Vortices Using Visualization Tools

By David Longmire, UDAAC

Prevention of foreign object damage (FOD) to jet engines is a high-priority operation on every flight line in the United States military. Small rocks or other debris can cause major damage to jet engines and endanger personnel and the mission. The FOD problem multiplies in a field environment where military transports and similar aircraft often land and take off on dirt runways while supplying military units with men and material.

Boeing Company researchers Drs. Arvin Shmilovich and Yoram Yadlin have harnessed the power of the supercomputer to help simulate the turbulent conditions that contribute to jet engine damage in austere conditions and examine new ways to prevent it.

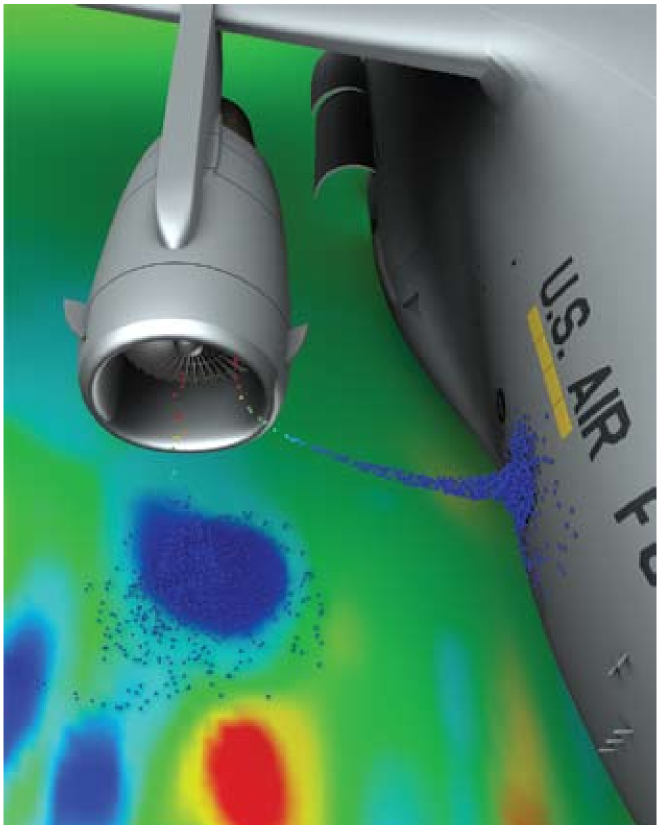
Their project "CFD (Computational Fluid Dynamics)-Base Analysis Method for Assessment of Inlet Vortex Reduction Concepts" is performed under the Boeing IRAD program. Partial sponsorship is provided by the Air Force Research Laboratory using the High Performance Computing Modernization Program (HPCMP) computing resources at the ERDC DSRC.

The researchers were investigating how ground vortices, which resemble mini tornados, can carry debris from the ground into the engine and cause engine failures on aircraft operating in harsh field conditions. Shmilovich and Yadlin utilized real-time data visualizations to provide insight into the ground vortex phenomenon and to view simulated solutions.

"Numbers – nowadays there are way too many numbers. We need to find different ways to look at these things. That's what visualization adds to our research," Shmilovich said.

"We use visualization as a development tool to look at new things and be able to learn what's going on in the data," explained Yadlin. "Sometimes I am not even sure what I am looking for. I am investigating in real-time looking at my data interactively."

ERDC DSRC's Cray XT3 was utilized for project data computation, which ran over a 5-day period in a sequence of batch jobs on 48 processors. The researchers easily ran their CFD simulations from the Boeing Company office in Huntington Beach, California, on the ERDC DSRC Cray XT3 machine located in Vicksburg, Mississippi, but found it necessary to transfer their data to the west coast facility to be able to do their investigative, interactive visualization.



Shown is a top view of a vortex formation of an inboard jet aircraft engine. The tornado-like flow is capable of dislodging sizable foreign objects off the ground (for example, rocks, chunks of ice, or asphalt), causing foreign object damage (FOD), which may lead to engine failure. A simulated solution is shown on page 26

DAAC produced high-definition visualization movies from the researcher's data, which were used for presentation at technical conferences highlighting Boeing's FOD research activities.

"The services that the DAAC provided, and which we surely appreciate, were for our high-end presentations," said Shmilovich. "On our day-to-day research, we utilize visualization almost every day, but we don't need that high a quality for our every day visualizations."

DAAC visualization scientist Richard Walters worked with Shmilovich and Yadlin to produce their visualizations. Walters, with the aide of DAAC animation artists Kevin George and Miguel Valenciano, inserted a realistic 3-D, textured model of a representative transport aircraft into the finished visualization to enhance the visual impact of the science.

"The visualizations produced by the DAAC successfully demonstrated the problematic flow field characteristics associated with routine runway operational procedures," Walters said. "These 'problematic' visualizations, which demonstrated the formation of inboard engine vortex regions, were complemented with visualizations that offered proposed solutions.

"Two proposed solutions were visualized, showing high-frequency air injection through small nozzles to break up the vortex regions. Creative efforts were taken to visually show free-flowing particles to help explain the science," Walters said. "This required several stages of a highly tailored data processing 'pipeline.' The first stage required secondary particle trace data to be calculated from the simulated CFD data. This was accomplished in EnSight," Walters explained.

"The results of the calculation were collected in an ASCII table containing the particle positions and velocity magnitudes of all time-based traces. Next, the tabular numerical values were converted to geometry - thus, spherical glyphs were created and separated in time-based files for each calculated x, y, z position. These files were converted to native rendering package format. Finally, the geometry associated with the aircraft boundary conditions of the simulation was also exported from EnSight, textured, and merged into the final scene with key frames to produce the optimum camera path that told the final story," Walters said.

Shmilovich said they use several different tools to visualize their data. "These movies were done in EnSight, but we have other visualization tools that we employ to help us quickly visualize our data. We don't always need movies; sometimes snapshots are good enough."

"What we are using now at least 90 percent of the time," Yadlin added, "is the commercial software packages EnSight and Tecplot. We know these tools very well."

Ground vortex activity develops during high-engine power setting, low speed, and static-ground operation of aircraft with turboprop or turbojet engines that are mounted close to the ground. Being able to simulate

The pulse jets device uses high-pressure air to alternatively eject fluid from two nozzles mounted underneath the engine nacelle close to the nacelle lips. The intermittent high-frequency ejection provides turbulent mixing to prevent the formation of a coherent vortex. The particles are colored from 0.5 Mach (red) to 0.0 Mach (blue)

and visualize the vortex and develop possible solutions is an important part of the research effort.

"The rotational flow field induced by the ground vortex is the cause for kicked up dust and dirt, which can become entrained in the airflow that is drawn into the engine inlet. The tornado-like flow is capable of dislodging sizable foreign objects off the ground such as rocks, chunks of ice, or asphalt, causing FOD that may lead to engine failure," Shmilovich said.

"This ground vortex problem hinders the ability to land aircraft in underdeveloped field environments and perform essential ground maneuvers on unimproved terrain," added Yadlin.

CFD simulations running on the supercomputers at the ERDC DSRC were used to facilitate the development of ground vortex alleviation/elimination concepts. Though none of the concepts have made their way to Boeing aircraft yet, Shmilovich said that the Boeing pulsed system concept had been proven effective by in-lab testing at the Georgia Tech Research Institute.

"The pulse jets device developed by Smith and Dorris uses high-pressure bleed air from the compressor, which is ejected alternatively from two nozzles mounted underneath the engine nacelle close to the nacelle lips. The intermittent high-frequency ejection provides turbulent mixing to prevent the formation of a coherent vortex," Shmilovich said.

Another of Boeing's vortex disruption methods was proposed by Shmilovich as a result of the project research. A sprinkler jet system for vortex alleviation uses continuous ejection through a moving nozzle mounted on the nacelle lip in order to provide wide area coverage. This system reduces the risk of vortex ingestion in realistic situations when the vortex moves rapidly.

"The current CFD simulations focus on the evaluation of the sprinkler system for full airplane configurations. The control system has been incorporated into a model that includes all relevant aircraft components for adequate representation of the flow during airplane ground operations," Shmilovich said.

In the sprinkler jet system, the high-pressure bleed air from the aircraft's engine compressor is piped to a valve located inside the engine cowl and close to the nacelle lip. This valve is connected to the nozzle, which is deployed during low speed and high-engine power operations. During actuation the nozzle swivels according to a prescribed motion at a high frequency.

"The slew motion of the ejected fluid disrupts the global flow field in front of the engine and prevents the formation of vortices," Shmilovich said.

Simplified SSH Tunnels for ParaView Client/Server

By Joel P. Martin, Mississippi State University (MSU); Rick Angelini, Army Research Laboratory; Rhonda Vickery, MSU

ParaView has definitely found its place for visualizing the results of many HPC codes. It is extendable and scalable. It can be run on everything from a laptop to a large cluster. Utilizing these clusters, however, can often be challenging. Modern, general-purpose visualization tools such as ParaView use a client/server architecture. That is, the "client" portion of the application runs on the local workstation, fully utilizing the graphics-processing capability of the local hardware. The "server" side of the application generally resides on the HPC system where the data were calculated (or stored). Here, data-intensive operations can be run in parallel, fully utilizing the capabilities of the hardware where the data were generated. The critical component to make this solution work properly is the availability of appropriate network connectivity.

However, with increased awareness for network security, firewalls and other mechanisms have been put into place that often restrict the availability of TCP ports required to implement this client/server architecture. It is possible to make the appropriate network connections using SSH port forwarding commands; however, these commands can be quite cryptic and convoluted. Additionally, there is no easy mechanism to start up the ParaView server in parallel on the remote high performance computer that is aware of these SSH sockets.

To simplify this process, the authors have developed a ParaView connection definition (.pvsc) file and an LSF (Load Sharing Facility for High Performance Computing) script that allow the user to quickly and securely connect a ParaView client to the computational nodes of MJM at the Army Research Laboratory DoD Supercomputing Resource Center (ARL DSRC).

When should a cluster be used?

Although running ParaView in client/server mode can be a little more complicated initially, the benefits make it worthwhile if any of the following conditions exist:

- Data are too large to move. It may be the case that the data are so large that it will take hours or days to move it to the desktop from the server where it was generated, or the desktop workstation does not have sufficient disk capacity to house the computed data set.
- Data are too large to load into the memory of the machine. The user may have tried loading data in ParaView on the desktop, and it failed because it ran out of memory.
- Client has minimal rendering resources. The desktop may be a thin client or have an outdated graphics card. In rare cases, the graphics card may not be hardware accelerated.

If the data do not quite fit the conditions above, the user will have to experiment to see if running ParaView in client/server mode is beneficial. The user should keep in mind that there will be latency added by transmitting large images over a network from a remote server and overhead because of server side communication on multiple cluster nodes. Since most of the HPC clusters do not have user-accessible hardware rendering, all images are generated in software before being sent to the client. Once connected in client/server mode, several parameters can be adjusted to improve rendering efficiency. The user can refer to https://daac.hpc.mil/wiki/Paraview_Client-Server_Mode for more information.

Requirements

The technique described below was developed to solve a particular issue on the ARL DSRC systems and can be easily transferred to other HPCMP resources. This technique assumes that the user has the following:

- Account on MJM at the ARL DSRC.
- ParaView 3.4.0 running on a Windows, Linux, or Mac machine.

One-time Setup

The first thing that needs to be addressed is to provide the ParaView client with a definition of how to define the connection to the remote HPC resource. This is accomplished through the use of a ParaView connection definition file. Once this file is loaded into the ParaView client, the definition will be saved into a permanent configuration file. The specific steps are as follows:

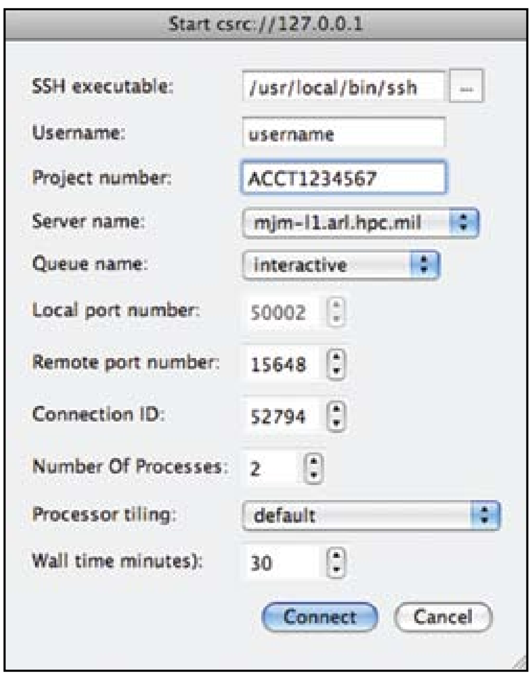
1. Obtain a copy of the ParaView connection definition file that usually has a .pvsc file extension. This file can be downloaded as paraview_batch.pvsc from https://daac.hpc.mil/wiki/ParaView or on systems at ARL that receive the standard ParaView distribution; the file is /usr/cta/CSE/Misc/arl-hpc.pvsc.
2. Register the ParaView connection file with ParaView through the following steps:
   a. In ParaView, click File → Connect → Load Servers.

b. Choose the .pvsc connection file identified in step 1 and click OK.

c. You should now see a screen that looks like this:

Connecting to the Server

1. Get a Kerberos ticket.
2. Start ParaView. Once running do the following:
   a. Click File → Connect.
   b. Click "ARL Clusters", then "Connect".
3. A list of user-configurable options will appear.



a. Set the SSH Executable location by typing it in the box or using the ... button. The actual path to SSH will be site-dependent; however, you will want the HPCMP-kerberized version of SSH (for instance, on ARL systems, the path is /usr/krb5/bin/ssh). On Windows, you will need to find plink.exe. It will likely be in C:\Program Files\HPCMP\Kerberos or a similar location.

b. Set your username and project number. These will be saved for the next time you use ParaView.

c. Set the queue name. The options are debug, urgent, staff, high, challenge, cots, interactive, standard-long, standard, and background. You will likely not have access to all of these queues. For information on run times and priorities for each queue, visit http://www.arl.hpc.mil/System-Software/lsf.html#limits.

d. Local port number cannot be changed because of a bug in ParaView 3.4.0. If another service is running on port 50002 on your local machine, you will need to edit the servers.pvsc file that is part of your ParaView configuration. When the bug in ParaView is resolved, this option will be enabled.

e. Remote port number and Connection ID can usually be ignored. If you are having trouble connecting to a server, it may be useful to try a different remote port number, as someone else may already be using the port you requested.

f. Number of processes sets the number of CPUs that ParaView can use. Since MJM has multiple processors per node, there will be multiple ParaView instances on each node.

g. If you are memory-bound, it may be useful to change processor tiling from default. This will reduce the number of CPUs/node that is used.

h. Wall time sets the length of time you want to run ParaView. There is a tradeoff in how long you request. If you request a long wall time, your job will take longer before it starts. If you request a short wall time, your program may be terminated before you are finished. Remember that you can use File → Save State if you are getting close to the end of your requested wall time. You could then start a new connection and use

i. File → Load State to pick up where you left off. This will also work if you need to increase/decrease the number of CPUs for your session.

4. Click connect. The server can take 10 seconds to several minutes or more to connect. This is based mostly on the current load of the server and your priority in the queue.

Debugging Problems

Suggestion:
Debug and interactive are the only queues that allow an interactive shell. This may be useful if you are troubleshooting the connection, as additional debugging information may be available.

- Error:
  ssh_askpass: exec(/usr/local/ossh/libexec/ssh-askpass): No such file or directory. Host key verification failed.
  Solution:
  This means you have never logged into this machine with SSH before. Use Putty or SSH from the command line to access this machine. Respond 'yes' when asked if you want to continue to connect to the machine. Try the ParaView connection again.
- Error:
  Permission denied (gssapi-with-mic).
  Solution:
  This can be a couple of things. The most likely one is that you need to request/renew your Kerberos ticket. It can also mean that you do not have an account on the machine to which you are trying to connect.

Under the Hood

Some of you are probably trying to figure out what is happening to make all this work. Once you have decided on the connection options and clicked 'connect', the ParaView client starts listening to the local port number. An SSH connection is then opened to the server name that is chosen, and the LSF script is submitted to the scheduler. The SSH connection also includes a tunnel that connects the local port number on the client to the remote port number on the server name.

When the LSF scheduler decides it is time for your job to run, a script is run on the first cluster node. This script sets up a tunnel back to the login node that you picked and starts pvserver on all the nodes you were allocated by the LSF. Because this second tunnel is set up, there is a pathway all the way from the cluster nodes to the client machine. This lets pvserver connect to a port on its node, which sends data to the cluster's login node, which sends data to your desktop. Because of the tunnels, any firewalls preventing TCP/IP traffic from the server to your desktop are avoided.

Conclusion

The technique defined assists users with the cumbersome task of making a connection with a client/server application between a desktop workstation and a remote DSRC asset. This technique uses SSH tunneling to pass all network communication required for the application to run properly. This means it does not require any special assistance from system administration staff to allow particular TCP ports to communicate outside of the local network. Through the use of a ParaView connection definition file, the incantations required to establish the SSH tunnel are hidden from the end user, allowing ParaView to be securely and efficiently utilized by the HPC user community.

References

[1] http://www.arl.hpc.mil/SystemSoftware/lsf.html#limits

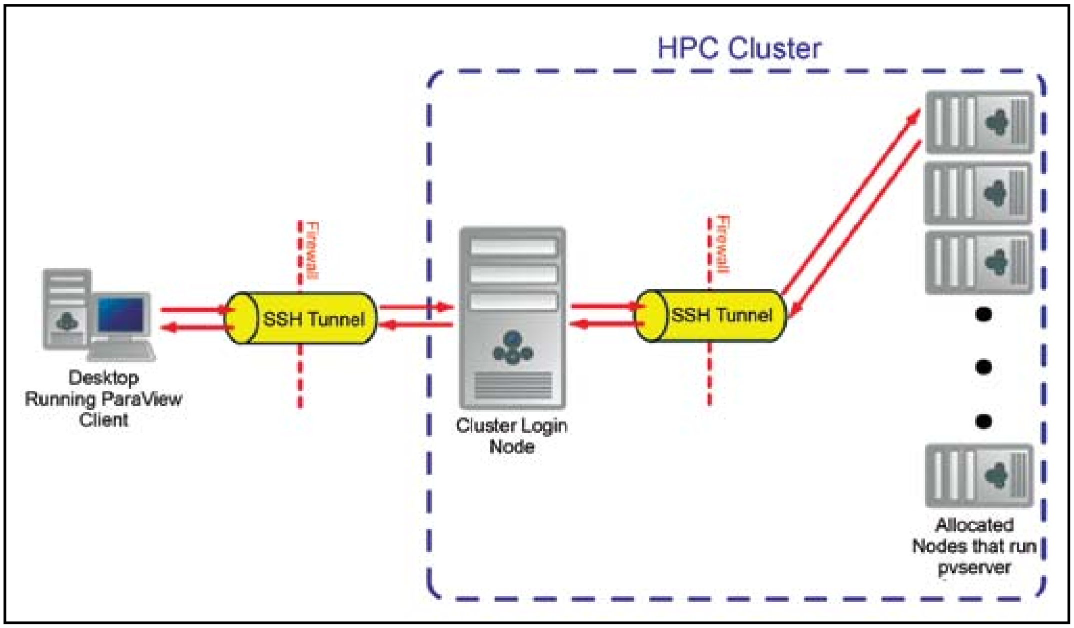
[2] http://www.arl.hpc.mil/SystemSoftware/modules.html

[3] http://www.slac.stanford.edu/comp/unix/package/lsf/currdoc/html/lsf6.1_ref/O_envars.html

[4] http://www.open-mpi.org/community/lists/users/att-5929/mpirun.lsf

[5] http://www.open-mpi.org/community/lists/users/att-5929/openmpi_wrapper

[6] https://visualization.hpc.mil/wiki/ParaView



Advanced Computing for Battlefield Applications: Real-Time SIRE Radar Image Processing

Dale Shires, Song Park, and Lam Nguyen, U.S. Army Research Laboratory; David Richie, Brown Deer Technology; and James Ross, High Performance Technologies, Inc.

As part of the Army Research Laboratory (ARL) Advanced Computing Strategic Technology Initiative, researchers at the ARL Computational and Information Science Directorate (CISD) and Sensors and Electron Devices Directorate (SEDD) are collaborating on technologies that will bring powerful computing to forward-operating environments and missions. Several changes in the high performance computing (HPC) industry have made this research timely and critical. First, scalar processors have for the most part followed the von Neumann architecture where a CPU follows a fetch, execute, and store instruction path. Mapping computationally complex algorithms to this sequence can have a limited effectiveness (many algorithms achieving only 10-30 percent of peak theoretical efficiency). Second, the "free ride" on computer performance gains in the HPC world appears to be nearing an end. Dynamic power for a complementary metal oxide semiconductor (CMOS) chip is proportional to the product of load capacitance, the square of voltage, and switching clock frequency. Therefore, higher clock speeds of CPUs lead to more and more generation of heat.

To compensate, chip manufacturers began shifting to more cores clocked at lower speeds. This technique, known as voltage scaling, provides a throughput increase by using parallelism with no increases in power requirements.[1] Parallelism is now the norm for developers who can no longer count on increasing capabilities and speeds of single processors.[2] Furthermore, the commodity computing sector began to focus more on HPC using the technologies they were developing. Specialized cores designed for graphics processing were opened up to developers in new application programming interfaces (APIs). Also, the increasing density of configurable semiconductor devices has led to an emergence of reconfigurable computing and tools to promote its use. All of these changes have led to a large number of options for those seeking high performance in commodity solutions.

This research is assessing how this changing landscape in computing can be harnessed for maximum deployable Army benefit. Hybrid computers are becoming more common in HPC settings. Cray's XD1 system included an option for reconfigurable computing using Field Programmable Gate Arrays (FPGAs) as co-processors closely connected to the CPUs. Multicore clusters are also coming online with nodes incorporating programmable general-purpose graphics processing units (GPUs). Likewise, we are investigating heterogeneous systems including multicore processors, FPGAs, GPUs, and hybrid systems such as the Cell processor. These options can allow for smaller footprint systems with tremendous speedups compared with traditional large clusters of von Neumann general-purpose CPUs. Our goal is to combine the asymmetric capabilities of these various cores into a complete HPC system. The primary focus is on battlespace applications to move HPC closer to the soldier and field commander. Doing so opens up new possibilities to increase the capabilities of the soldier and the systems being used in many application areas (e.g., sensors and intelligence applications).

Algorithm development for applications needing improvements in speed and fidelity will be critical. Each of these accelerators comes with a different development approach, and determining which one to target is not always easy. FPGAs require a customized design approach with the best performance coming from low-level hardware design languages such as VHDL. They are best suited for integer- and bit-based operations. GPUs use a multithreaded approach with fast context switching between threads that are executing and those that must stall waiting for memory operations. Multicore systems must be programmed using message-passing protocols such as MPI or shared memory paradigms such as OpenMP. Interfacing all of these and merging them into a coherent runtime system is not an easy task and is an active research component at ARL.

We have had several successes with the various asymmetric cores we have investigated, but the remainder of this monograph discusses one in particular that has mapped well to a heterogeneous approach. To support the U.S. Army vision for increased mobility, survivability, and lethality, SEDD has designed and developed the forward-looking ultra-wideband (UWB)

[1] "Multi-Core Microprocessor Chips: Motivation and Challenges," Dileep Bhandarkar, Intel Corp., May 2006.

[2] "Discovering Multi-Core: Extending the Benefits of Moore's Law," Geoff Koch, Technology @ Intel Magazine, July 2005.

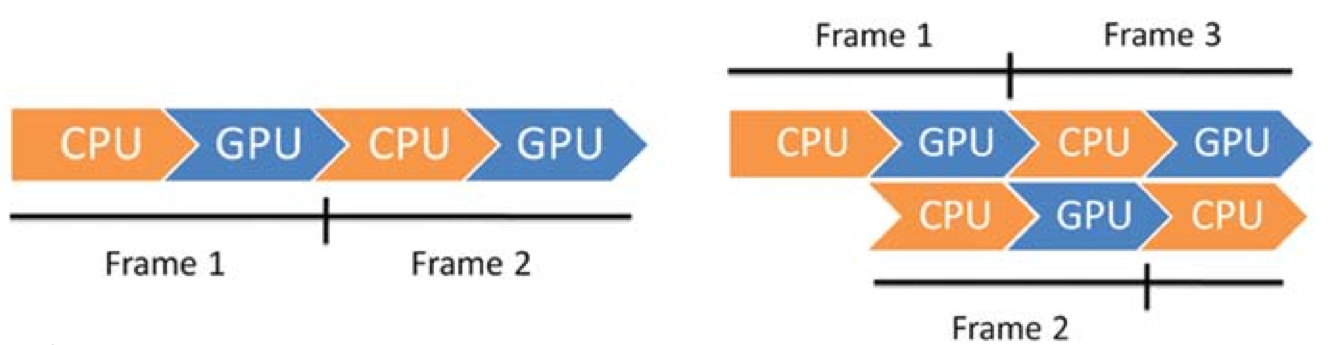
synthetic aperture radar (SAR). The radar is based on time-domain wideband impulses and uses a data acquisition technique called Synchronous Impulse REconstruction (SIRE). Several signal processing algorithms are being developed to allow the system to discriminate targets from other objects.

The computational requirements for the radar processing are significant. Including data acquisition rates and integration using forward motion, along with parameters such as platform speed and subimage size, the operation count quickly exceeds 20 million operations per second. Because of the high floating-point requirements of this code, we decided to target a multicore CPU combined with the NVIDIA GTX 8800 GPU hardware and the associated Compute Unified Device Architecture (CUDA) for speedup. The 8800 GTX has 128 cores clocked at 1.35 GHz with a FLOP rate of about 345 GFLOPs. CUDA provides a "C"-like programming interface where kernels are written to run on the device (GPU) and are callable from the host (CPU). CUDA has effectively opened up the GPU for direct use by the application developer. The 8800 GTX uses a fast context switching design for threads executing on the 128 compute cores. Accordingly, it is best to use a fine-grain approach to parallelism. For example, one pixel of each 100x250 frame is assigned to a thread. If a thread stalls waiting for a computation to complete (such as a division operation requiring several clock cycles), the runtime system will context switch to a new thread so the core can continue to work.

Our starting point was an initial code written in Matlab. Matlab is widely used in the signal and image processing (SIP) community and is best at providing a wide-range of functionality and easy-to-use scripting language. However, execution speed is not one of its strong points. This code was carefully profiled and analyzed, and algorithms were written in a procedural fashion using the C programming language. Critical sections of the code were identified, the primary function being a backprojection routine that accounted for over 80 percent of the total runtime.

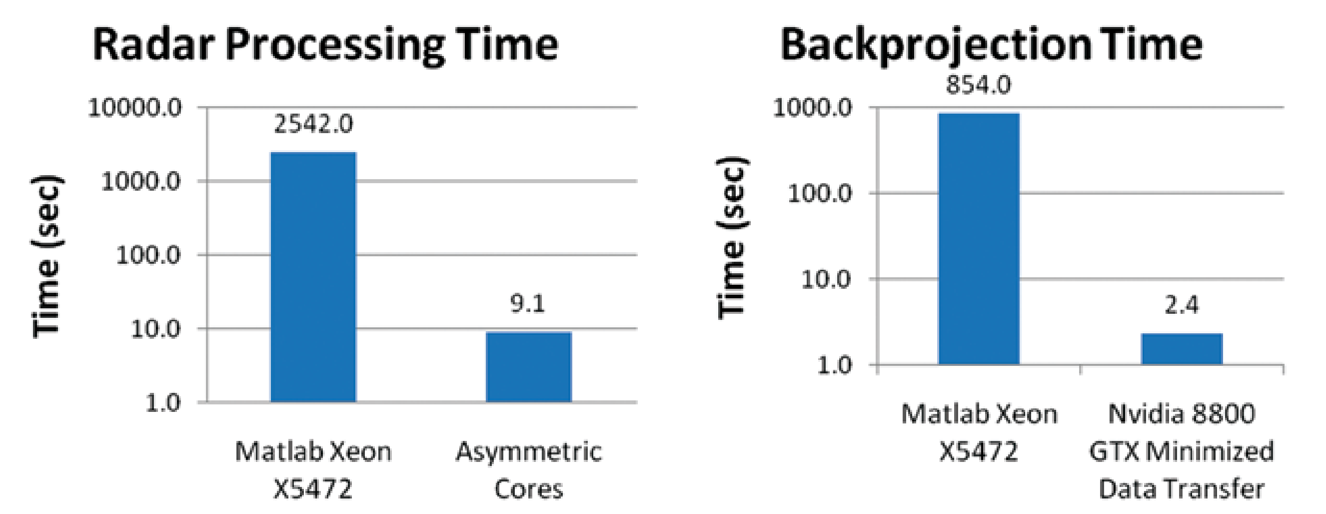
The team set out to optimize the radar processing in an approach to utilize both multicore CPU assets coupled with GPU accelerators. The first step involved developing and testing backprojection algorithms for the GPU. In phase 2, this combined CPU-GPU approach was analyzed, and bottlenecks involving data transfers over the PCI bus were identified. Researchers then discovered ways to reuse data resident on the GPU from prior transfers since radar frame data were overlapping. This optimization reduced the size of per-frame data transfers by about 80 percent. Phase 3 involved identifying ways to pipeline the CPU and GPU interaction. Phases 1 and 2 used an approach where the CPU became idle after calling the GPU. Control would then go back to the CPU after the GPU completed its work (Figure 1a). A new approach was developed using threaded control on the CPU that allowed it to immediately begin processing new frame data after the GPU was invoked (Figure 1b).

The combined CPU-GPU asymmetric solution has yielded an incredible amount of increased performance. Figure 2a shows the overall computing time of the original Matlab processing time and the asymmetric solution developed by the team. A distance of roughly 274 meters of data acquisition and processing requires about 42 minutes to process in Matlab. When put in terms of real-time processing speeds, this means the radar would have to travel less than 0.5 MPH in order to keep up with processing demands. However, the new approach would accommodate a radar system with speeds up to 68 MPH. Realistically, other physical constraints on the system will limit it from approaching this newly computed maximum speed. However, the new approach allows for a system that will provide far greater real-time speeds. Perhaps more importantly



(a) Serialized execution
(b) Optimized pipelined approach

Figure 1. CPU-GPU control flow



(a) Total radar processing runtime *(b) Exclusive backprojection runtime*

Figure 2. Reduced total runtime through asymmetric core solution

is the new flexibility for engineers to include other enhanced detection and signal and image processing algorithms that were not possible with the real-time constraints of the system.

As we progress in this research, we will be looking at using more asymmetric cores (such as a combined FPGA, GPU, and CPU approach) in algorithms or parallelizing across numerous asymmetric cores. This will involve detailed algorithm analysis and selection of cores based on code signatures. The difficulties here are predominately load balancing, selection of either task or data parallelism, and a way to synchronize and perform data transfers. A simplified example of this is the Folding at Home project at Stanford where individuals can offer their idle processors (whether it is Windows PCs, Linux workstations, Playstation3s, etc.) to a largely data parallel application. These issues will need to be addressed as various streaming and multithreaded architectures find their way to fielded units. To be fully effective, this will involve some research on work distribution and load balancing among the components.

There is a fundamental change occurring in the HPC field as throughput architectures gain in importance, market share, and raw floating-point computational capacity. Portable and even hand-held HPC is quickly becoming a reality, and a concerted effort is needed to make these technologies viable to Army forces. We have been able to show a considerable amount of speedup in many applications and kernels. Unfortunately, the performance gains achieved are at times proportional to the amount of effort put into the task. New approaches are holding out some promise of moving these technologies more into the open-source arena, thus allowing researchers more direct understanding and control at low levels. These efforts should greatly facilitate the maturation of compiler and API technologies. The performance of these new hardware approaches is providing a way to field HPC, and we look forward to continued research and development efforts to transition these new technologies to Army applications.

HPC Reveals the Phenomena of Flight on a Small Scale

By Michelle McDaniel and Chuck Abruzzino, AFRL DSRC

Need for Miniaturization

Unmanned aerial vehicles (UAVs) assist many sectors of the population. Currently, police departments use large UAVs in bomb detection, border patrol, and reconnaissance. In addition to these uses, the military conducts seek and destroy missions using armed UAVs. Scientists use them to monitor weather, animal herds, and forest fires. Can the current UAV technology provide a long-term, less conspicuous alternative to human interaction in dangerous situations?

Researchers say yes. Micro air vehicles (MAVs) are small-scale UAVs, typically around 15 cm, that allow a distant viewer access to locations they may be otherwise unable to approach. An example of this would be a hazardous environment, like searching for survivors in a collapsed building, or a clandestine operation in an unsecure location. Current UAVs can perform large-scale operations, and researchers are investigating smaller biologically inspired MAVs (like birds and insects) that will be better able to access tight spaces and be less conspicuous than the larger UAV.

AFRL Takes the Lead

The Air Force Research Laboratory (AFRL) requested an activity group, led by Dr. Leslie Perkins, be started to investigate MAVs. The activity group is focused on two major goals: a 2015 goal of bird-sized MAVs with a life span of hours and a 2030 goal of autonomous insect-sized MAVs with a life span of weeks. These goals are broad enough to allow researchers flexibility in their development, but narrow enough to help direct research. The expansive nature of these goals allows the activity group to engage members of multiple disciplines such as biologists, chemists, and physicists. Each brings their own perspective to the group in the hopes of truly discovering how animals take flight.

In the near term, these goals may be addressed by fixed-wing MAVs that will be able to provide reconnaissance in unsafe war zones. The extended power capability of fixed-wing MAVs lends itself to long-term observation. This ability allows soldiers to infiltrate locations that might be inaccessible and predict targeted locales and either provide a military presence or evacuate the area. This can reduce military and civilian casualties.

In the long term, more novel MAV designs employing flexible flapping wings will be required. Ornithopters, an MAV machine that is held aloft and propelled by wing movements and can be augmented with multiple types of equipment, can allow movement between demolished concrete blocks, make sharp turns when encountered, and provide a hovering capability. This will help emergency medical technicians, firefighters, and police officers save lives that may have been otherwise lost.

Computer graphic of a possible clandestine operation monitored by two miniature robo-bugs equipped with advanced sensors

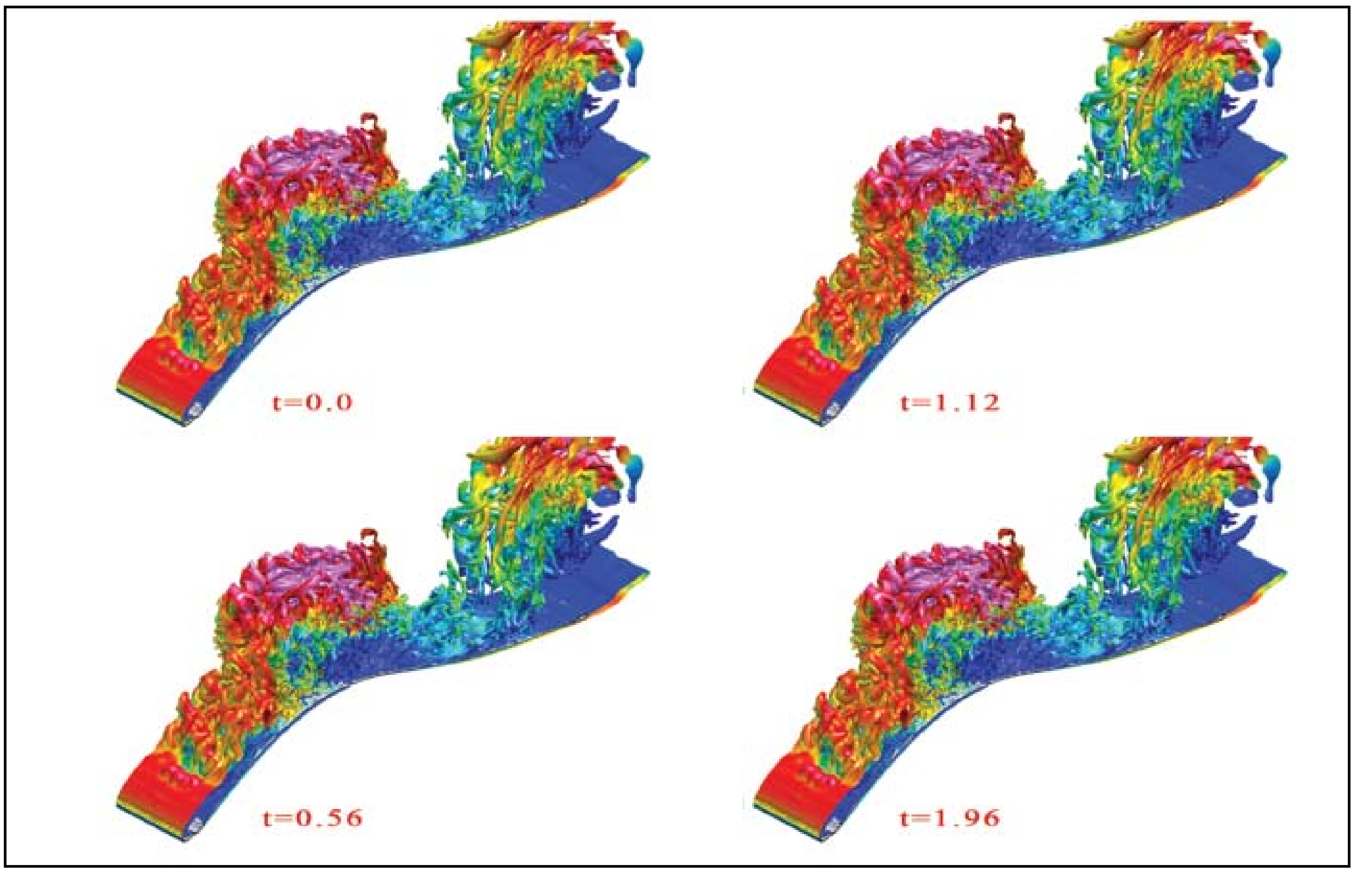
The MAVs also provide significant advantages in adapting to such adverse effects as wind gusts and crosswinds. The MAVs will be more easily able to hide in plain sight for reconnaissance missions. The conceivable use of biologically inspired MAVs in the military and civilian arenas makes research into their flight characteristics more essential than ever. Different types of flight capabilities, like the hovering capabilities of a hummingbird or the flapping capabilities of a bat, require a definitive understanding of how flight occurs before they can be correctly exploited in MAV design.

Research

Dr. Raymond Gordnier is the principal investigator of a challenge project currently utilizing the AFRL high performance computers Eagle, Falcon, and Hawk, and a combination of HPCMP-provided computational fluid dynamics (CFD) software, computational structures and mechanics (CSM) software, and an in-house code to do intricate research into flexible and flapping wings. The challenge project will provide a better understanding of how membrane wings are used for flight and how the membrane is affected by airflow over and around the wing. It will provide the underlying physics of natural flight. Dr. Gordnier has run in-house software (FDL3DI) on many DSRC supercomputers, but has a good historical relationship with AFRL and has achieved successful results. FDL3DI is an AFRL-developed, three-dimensional (3D) block-structured implicit CFD code. Its unique features include high-order discretization schemes coupled with an overset mesh technique for treating complex geometries. The code also incorporates high-fidelity turbulence models and state-of-the-art parallel processing procedures. This CFD solver has been coupled with several nonlinear structural models in order to capture critical fluid/structure interactions resulting from wing flexibility.

MAV Usage

Developing effective biologically inspired MAVs requires that scientists building prototypes have a strong understanding of the impact of the unique flow environments encountered by MAVs to improve their aerodynamic performance. Dr. Gordnier points out that for MAVs, Reynolds numbers are low. Reynolds number quantifies the ratio of the inertia forces to viscous forces in a fluid. In other words, a low Reynolds number means that MAV aerodynamics is more dominated by viscous forces. The magnitude of the Reynolds number also determines whether a flow over a wing will undergo a transition from a regular laminar flow to a highly fluctuating turbulent flow characterized by small-scale vortical flow structures. The transition from a laminar to turbulent flow can drastically change the aerodynamic performance of an MAV. MAVs operate in transitional flow regimes; the flow may be laminar over part of the wing, turbulent over part of the wing, and transitional in between.



Unsteady membrane response of a flexible membrane wing airfoil. Isosurfaces of vorticity magnitude colored by axial velocity at selected time during a cycle, α=14º. The three-dimensional structure of the shedding vortices is represented in these four views. A laminar separated shear layer originates from the leading edge but rapidly succumbs to spanwise instabilities and transitions to turbulent flow. This turbulent shear layer rolls up to form coherent vortical structures that are subsequently shed from the leading edge

This proves to be challenging for traditional CFD solvers with conventional turbulence modeling. Wing motion further complicates the situation since the transition point may travel from the leading edge where the airflow meets the wing, to the trailing edge at the back of the wing. With a higher order scheme, Dr. Gordnier's team is attempting to capture the above motion from first principles. They are not modeling the transition, but calculating it directly from equations. Transitional and turbulent flows are composed of larger scale flow features to small-scale flow structures that must be accurately represented by the computations. Accurately simulating this broad range of scales results in computational demands that far exceed the abilities of a standard desktop computer or even multiple desktop computers networked together to do parallel processing. Many scientists doing this type of research necessary to advance MAV technology have to make concessions to their computing capabilities. Using the HPCs available at the AFRL DSRC removes these concessions and allows the computational results produced to be more comprehensive and give a greater insight into turbulent flows encountered by MAVs.

By exploiting the high performance computational power available from the AFRL DSRC, Dr. Gordnier and his team have been able to produce computational simulations of an increasingly complex and multidisciplinary nature. Completing these highly accurate and lengthy computations in a reasonable time frame was enabled by the massively parallel systems available through the HPC program. To complete the simplest of these computations required meshes of the order of 26 million points with the computations distributed over hundreds of processors. These highly accurate results have given MAV researchers new insights into the physics of MAV flight and provide additional information that includes the impact of membrane flexibility on the lift produced by an MAV-type membrane wing and a detailed description of the vortical flow structure over a wing undergoing a flapping type motion. These types of high-fidelity simulations will allow MAV design challenges, such as adaptation to gusts in an urban canyon environment, enhanced maneuverability, and perching, to be addressed with a more keen understanding of the fundamental aerodynamics involved.

Dr. Gordnier's research is a wonderful example of how high performance computers can benefit researchers entering into a relatively unknown area. Although there is some understanding of how winged animals fly, the true fundamentals are ambiguous. To build realistic MAVs, it is necessary to understand the basics. Without this understanding, the systems developed will not be truly effective and will provide minimum impact to the warfighter.

This HPCMP challenge project "High-Fidelity Computations of Moving and Flexible Wing Sections with Application to Micro Air Vehicles" was run at multiple sites, including the AFRL DSRC by Dr. Raymond E. Gordnier, Principal Investigator (PI), Miguel R. Visbal, and Marshall C. Galbraith, U.S. Air Force Research Laboratory, Air Vehicles Directorate (AFRL/RB), Wright-Patterson AFB, OH.

AFRL DSRC system utilization was as follows: (Falcon) with 535,481hours, (Hawk) with 1,379,591 hours, and (Eagle) with 217.006 hours.

Design of Energetic Ionic Liquids

Jerry A. Boatz, Air Force Research Laboratory, Space and Missile Propulsion Division, Edwards AFB, CA; Yu-Hui Chiu, Air Force Research Laboratory, Space Vehicles Directorate, Hanscom AFB, MA

An essential need of the U.S. Air Force is the discovery, development, and fielding of new, energetic materials for advanced chemical propulsion in space and missile applications. Some of the key factors driving the requirement for new chemical propellants include (a) improved performance in terms of increased specific impulse and density, (b) reduced sensitivity to external stimuli such as impact, friction, shock, and electrostatic discharge, and (c) mitigation of environmental and toxicological hazards (and the resulting costs) associated with currently used propellants.

A class of compounds that can potentially meet these requirements is known as ionic liquids (ILs), which are organic salts with unusually low melting points and vapor pressures. The physical and chemical properties of ILs render them useful for many purposes, most notably as environmentally benign (green) solvents/reaction media but also as catalysts, electrolytes, etc. [1]. From a Department of Defense (DoD) perspective, ILs are being explored as new propellants, explosives, and munitions. The Air Force in particular is interested in ILs for multiple propulsion applications, including their use as monopropellants, bipropellants, and working fluids for electric propulsion [2]. Some of the advantages of ILs relative to current propellant ingredients include increased density, higher specific impulse, reduced sensitivity, and lower toxicity. Furthermore, the properties of ILs can be carefully tuned, for example, via the choice of the component ions.

The overall objective of the Design of Energetic Ionic Liquids challenge project is to address several key technical issues and challenges associated with the characterization, design, and development of ILs as new propellant ingredients. Among these, for example, are a fundamental understanding of the (in)stability of ILs, the intrinsic nature of the short- and long-range structure and interactions between the component ions[2e-f], and identification of the key steps in the initial stages of decomposition and combustion [2a-c]. The research described in this article, which is a subset of the overall challenge project effort, is focused on the characterization of the structures, stabilities, and vibrational spectra of the gas-phase ion clusters of the 1-ethyl-3-methylimidazolium *bis*(trifluoromethyl) sulfonylimide ionic liquid, which has been selected as the propellant for the high-precision positioning demonstration [3]. In order to assess and tune the performance of these types of working fluids, it is necessary to characterize the exhaust plume, which contains a distribution of clusters, including [Emim$^+$]$_j$[Im$^-$]$_k$ (j=k±1), i.e., clusters containing either an excess cation or anion. The results of these calculations serve four main purposes: (1) to calculate the internal energy distribution functions of the ion clusters, which in turn are used to assess their thermal stability and to characterize the relative ion cluster populations, (2) to determine whether or not charge neutralization occurs in the ion clusters, (3) to provide a comparison between DFT and MP2 results, and (4) to establish a baseline for new methods designed for efficient quantum chemical calculations for large systems, such as the Effective Fragment Potential [4] (EFP) and Fragment Molecular Orbital [5] (FMO) approaches.

The computational approach utilizes quantum chemical methods for prediction of ion cluster geometries, interaction energies, and vibrational spectra. Specific ion clusters of interest, which have been detected in vacuum electrospray ionization experiments [6], include the following clusters of the 1-ethyl-3-methylimidazolium cation (abbreviated as [Emim+] and shown in Figure 1a and the bis(trifluoromethyl) sulfonylimide anion (denoted as [Im-] and illustrated in Figure 1b: [Emim$^+$]$_j$[Im$^-$]$_k$; $1 \leq k \leq 3$, j=k or k±l.

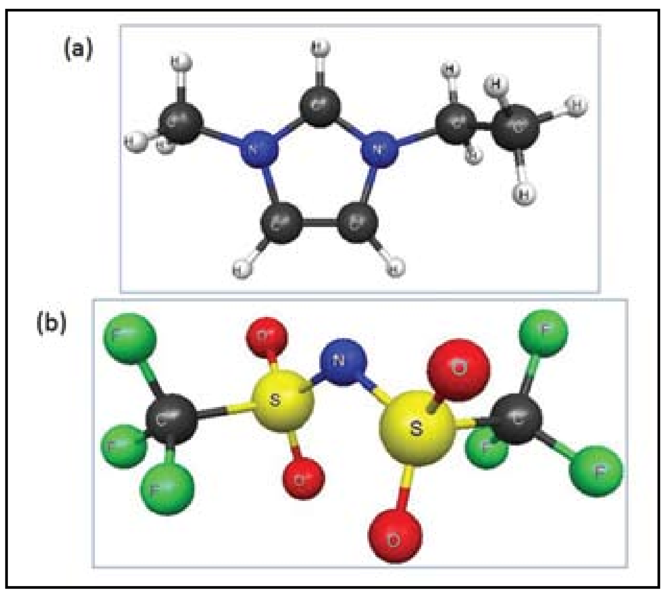


Figure 1. B3LYP/6-311++G(d,p) optimized structures of (a) the 1-ethyl-3-methylimidazolium cation and (b) the bis(trifluoromethyl)sulfonylimide anion. Hydrogen, carbon, nitrogen, oxygen, fluorine, and sulfur atoms are shown in white, black, blue, red, green, and yellow, respectively

Geometry optimizations were performed using density functional theory (DFT) with the hybrid B3LYP functional [7] and the 6-311++G(d,p) basis set [8], denoted as B3LYP/6-311++G(d,p). All structures were verified as local minima, and harmonic vibrational frequencies were obtained via diagonalization of the mass-weighted matrix of energy second derivatives with respect to nuclear coordinates, i.e., the hessian matrix. Geometries, vibrational frequencies, and interaction energies were refined using second-order perturbation theory [9] (MP2, also known as MBPT(2)) with the 6-311++G(d,p) basis set [8], denoted as MP2/6-311++G(d,p). All computations were performed using the GAMESS quantum chemistry code [10].

Since the potential energy surfaces of these systems tend to be rather "flat," it is necessary to use more stringent convergence criteria for geometry optimization of ion clusters in order to obtain reliable structural predictions. Consequently, geometry optimizations of these clusters, which consist of a systematic sequence of energy+gradient evaluations, generally require up to thousands of steps to converge. Therefore, significant computational resources are needed for these types of computations. For example, optimization of the structure of the $[Emim^+]_2[Im^-]_3$ ion cluster required approximately 3 million CPU hours on the Cray XT3 at the ERDC DSRC.

As a starting point, the structure of the $[Emim^+][Im^-]$ ion pair was computed. At the B3LYP/6-311++G(d,p) level, three distinct local minima were found and are shown in Figure 2. The binding energies Eb are given in kJ/mol and indicate the energy required to separate the ions. The DFT binding energies for the three ion pairs are similar. In Figure 2a, the anion is located partially above the ring of the cation, suggestive of an attractive electrostatic interaction between the anion and the five-membered ring π electron density (located above and below the plane of the ring). In contrast, the anion in Figures 2b and 2c is located along the side of the cation ring. The dominant interactions in 2b and 2c are hydrogen bonds, which are weak attractive electrostatic interactions between a hydrogen atom that carries a partial positive charge and another atom bearing a partial negative charge. In 2b and 2c, hydrogen bonding occurs between the hydrogen atoms on the cation and the oxygen atoms on the anion. Note that 2c has an additional hydrogen bond between the cation and the nitrogen atom in the anion. Charge neutralization does not occur in any of the three local minima.

The minima shown in Figure 2 were reoptimized at the MP2/6-311++G(d,p) level and are shown in Figure 3. Note that the binding energies are significantly larger relative to DFT. The most stable structure is shown in Figure 3a, in which the anion is located directly above

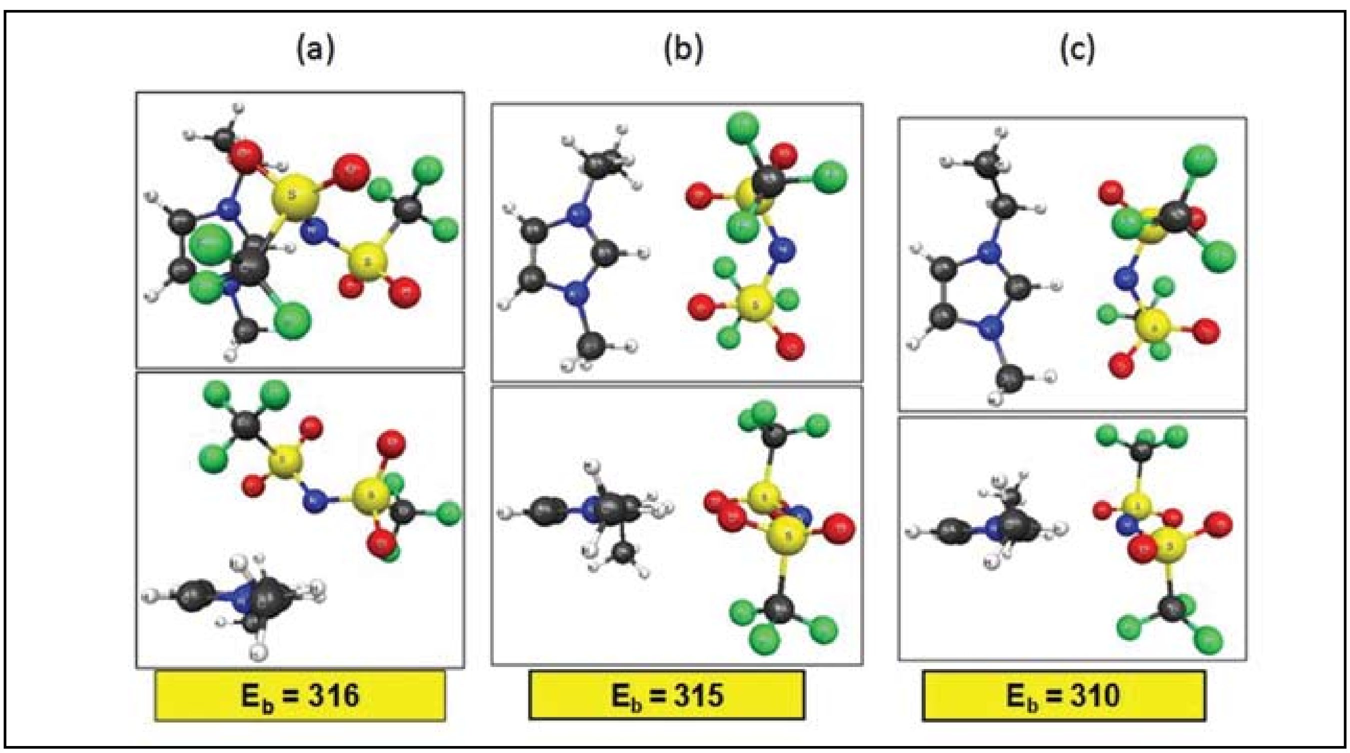


Figure 2. B3LYP/6-311++G(d,p) optimized structures of $[Emim^+][Im^-]$ ion pairs. The binding energy E_b (in kJ/mol) is the energy required to separate the ions. Each panel shows two views of the same structure

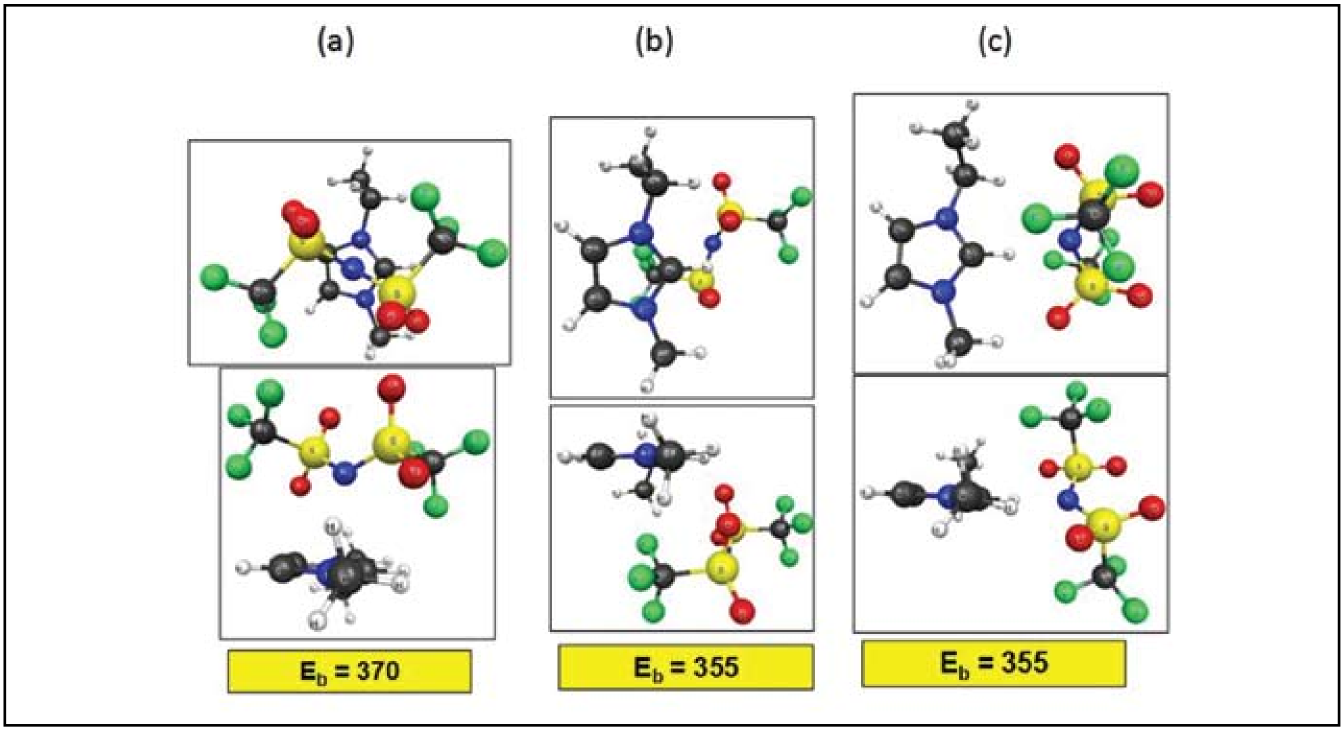


Figure 3. MP2/6-311++G(d,p) optimized structures of [Emim$^+$][Im$^-$] ion pairs. The binding energy E_b (in kJ/mol) is the energy required to separate the ions. Each panel shows two views of the same structure

the cation ring. The primary interaction in 3a appears to be between the nitrogen atom of the anion and the π density of the cation ring. In the structure shown in Figures 3b, the anion is partially located above the cation ring with one of the –CF¬3 groups positioned over the π density of the ring. Finally, in the isomer shown in Figure 3c, the anion is located along the side of the cation ring and is similar to the DFT isomer shown in Figure 2c, indicative of hydrogen bonding interactions between the ions. As in the case of DFT, charge neutralization does not occur in any of the three minima.

The DFT predicted structure of the [Emim$^+$]$_2$[Im$^-$] ion cluster is shown in Figure 4a. Not surprisingly, the cations are arranged on opposite sides of the anion in order to minimize the electrostatic repulsion between their positive charges. Note that the dominant interactions are hydrogen bonds between the cations and the oxygen and nitrogen atoms of the anion with no interactions present between the anion and the π density of the ring cations, similar to the interactions seen in Figures 2b and 2c.

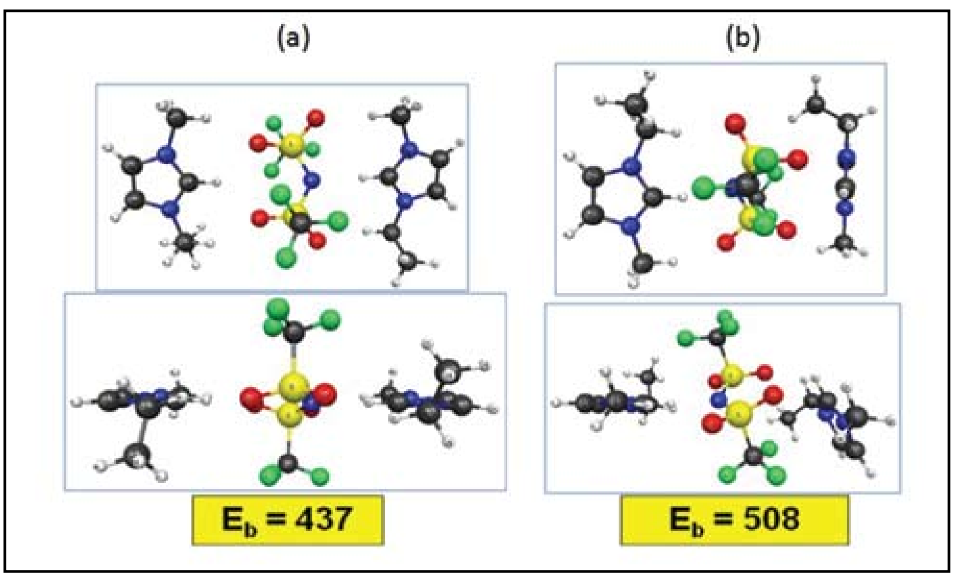


Figure 4. (a) B3LYP/6-311++G(d,p) and (b) MP2/6-311++G(d,p) optimized structures of the [Emim$^+$]$_2$[Im$^-$] ion cluster. The binding energy E_b (in kJ/mol) is the energy required to separate the ions. Each panel shows two views of the same structure

Figure 4b illustrates the MP2 predicted geometry of the [Emim$^+$]$_2$[Im$^-$] ion cluster. As in the case of the DFT structure in Figure 4a, the cations are located on opposite sides of the anion. However, note that one of the cations is positioned to interact with the anion via its π electron density, whereas the other cation is located along the side of the cation ring so that its interactions with the anion are dominantly hydrogen bonding in nature. As in the case of the [Emim$^+$][Im$^-$] ion pairs, the MP2 predicted binding energy (508 kJ/mol) is considerably larger than the DFT value (437 kJ/mol).

Figures 5a and 5b show the DFT and MP2 predicted structures of the [Emim$^+$][Im$^-$]$_2$ ion cluster. As expected, the cation is sandwiched between the anions in both structures. The DFT-predicted geometry shows the presence of hydrogen bonding between the cation and anions, whereas the MP2 structure has the anions interacting with the π electron density of the cation ring. As in the previous clusters, the MP2 binding energy (519 kJ/mol) is larger than DFT (401 kJ/mol), and charge neutralization does not occur.

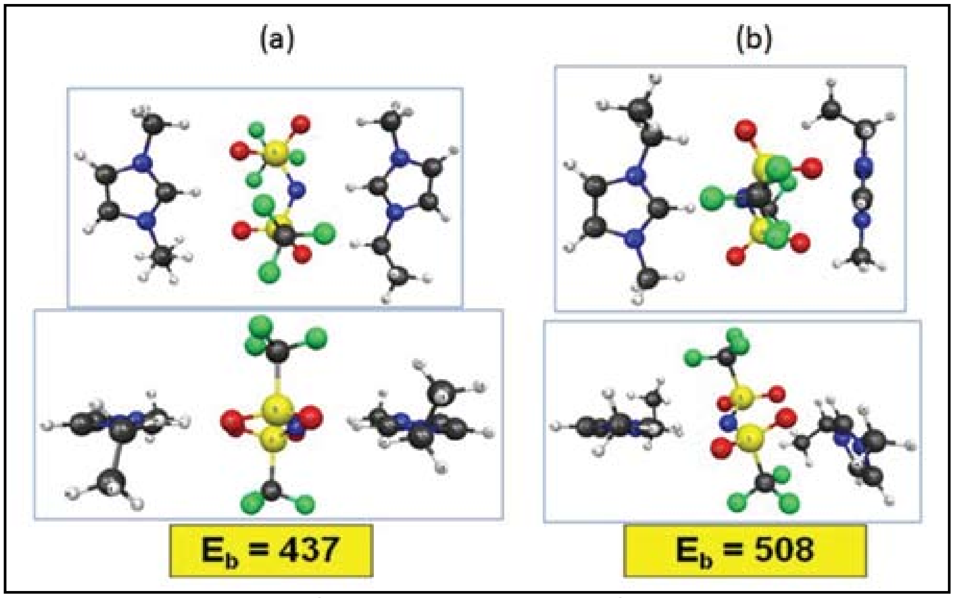


Figure 5. (a) B3LYP/6-311++G(d,p) and (b) MP2/6-311++G(d,p) optimized structures of the [Emim$^+$][Im$^-$]$_2$ ion cluster. The binding energy E_b (in kJ/mol) is the energy required to separate the ions

In summary, the structures, harmonic vibrational frequencies, binding energies, and internal thermal energy distributions of gas-phase ion clusters of the 1-ethyl-3-methylimidazolium bis(trifluoromethylsulfonyl) imide ionic liquid have been predicted using density functional theory (B3LYP/6-311++G(d,p)) and second-order perturbation theory (MP2/6-311++G(d,p)). Comparisons between the DFT and MP2 results show that the latter generally favor interactions between the anion and the π electron density of the cation ring, whereas the DFT geometries tend to favor interionic hydrogen bonding. Furthermore, MP2 predicts significantly stronger interaction energies relative to DFT. Ongoing studies include reoptimization of the DFT-predicted structures of the larger [Emim$^+$]$_2$[Im$^-$]$_3$ and [Emim$^+$]$_3$[Im$^-$]$_2$ clusters at the MP2 level and refinement of the calculated binding energies using coupled cluster theoretical methods (e.g., coupled cluster single and double excitations with perturbative estimates of triples, CCSD(T) [11]). Future efforts will include similar studies of ion clusters obtained from the 1-butyl-3-methylimidazolium dicynamide ionic liquid.

Acknowledgments

A grant of computer time from the DoD HPCMP ERDC DSRC, in conjunction with the Design of Energetic Ionic Liquids challenge project, is gratefully acknowledged. This work was supported by the Air Force Office of Scientific Research (AFOSR) and the Air Force Research Laboratory Propulsion Directorate and the Space Vehicles Directorate.

References

1. a) Rogers , R.D., and K. R. Sheldon. “Industrial Applications to Green Chemistry.” American Chemical Society (ACS) Symposium Series 818, American Chemical Society: Washington, DC, 818(2002).

b) Welton, T. “Room-Temperature Ionic Liquids: Solvents for Synthesis and Catalysis.” *Chemical Review* 99(1999): 2071.

c) Larson, A. S., J. D. Holbrey, F. S. Tham, and C. A. Reed. “Designing Ionic Liquids: Imidazolium Melts with Inert Carborane Anions.” *Journal of the American Chemical Society* 122(2000): 72654

d) Sheldon, R. “Catalytic Reactions in Ionic Liquids.” *Chemical Communications* (2001): 2399.

e) Wasserscheid, P., and W. Keim. “Ionic Liquids - New ‘Solutions’ for Transition Metal Catalysis.” *Chemisty*, International Edition (English) 39(2000): 3772-3789.

2. a) Pimienta, I.S.O, S. Elzey, J.A. Boatz, and M.S. Gordon. “Pentzole-Based Energetic Ionic Liquids: A Computational Study.” *Journal of Physical Chemistry A* 111(2000): 691.

b) Zorn, D.D., J.A. Boatz, and M.S. Gordon. “Tetrazolium-Based Energetic Ionic Liquids.” *Journal of Physical Chemistry B* 110(2006): 11110.

c) Schmidt, M.W., M.S. Gordon, and J.A. Boatz. "Triazolium-Based Energetic Ionic Liquids." *Journal of Physical Chemistry A* 109 (2005): 7285.

d) Cadena, C., and *E.J. Maginn. "Calculating the Enthalpy of Vaporization for I*onic Liquid Clusters." *Journal of Physical Chemistry B* 110(2006): 18026.

e) Wang, Y., and G. J. Voth. *Journal of Physical Chemistry B*. "Tail Aggregation and Domain Diffusion in Ionic Liquids." 110(2006): 18601.

f) Wang, Y., and G. Voth."Unique Spatial Heterogeneity in Ionic Liquids." *Journal of the American Chemical Society* 127(2005): 12192.

g) Wang, Y., S. Izvekov, T. Yan, and G.A. Voth. "Multiscale Coarse-Graining of Ionic Liquids." *Journal of Physical Chemistry B* 110(2006): 3564.

3. Ziemer, J.K.; M. Gamero-Castaˇno, V. Hruby, D. Spence, N. Demmons, R. McCormick, T. Roy, C. Gasdaska, J. Young, and B. Connolly, AIAA 2005-4265, 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Tucson, Arizona, July, 2005.

4. Gordon, M.S.; M.A. Freitag, P. Bandyopadhyay, J.H. Jensen, V. Kairys, and W.J. Stevens. "The Effective Fragment Potential Method: a QM-based MM approach to modeling environmental effects in chemistry" *Journal of Physical Chemistry A* 105(2001), 293.

5. a) Fedorov, D.G. and K. Kitaura. "The importance of three-body terms in the fragment molecular orbital method." *Journal of Chemical Physics* 120(2004):6832.

b) Fedorov, D.G. and K. Kitaura. "Extending the Power of Quantum Chemistry to Large Systems with the Fragment Molecular Orbital Method." *Journal of Physical Chemistry A* 111(2007):6904.

c) Fedorov, D.G.; K. Ishimura, T. Ishida, K. Kitaura, P. Pulay, and S. Nagase. "On the accuracy of the three-body fragment molecular orbital method (FMO) applied to Moeller-Plesset perturbation theory." *Journal of Computational Chemistry 28(2007):1476.*

6. Chiu, Y.-H.; G. Gaeta, D.J. Levandier, R.A. Dressler, and J.A. Boatz. "Vacuum electrospray ionization study of the ionic liquid, [Emim][Im]." *International Journal of Mass Spectrometry* 265(2007):146.

7. a) Becke, A.D. "Density-functional thermochemistry. III. The role of exact exchange." *Journal of Chemical Physics* 98 (1993):5648.

b) Hertwig, R.H. and W. Koch. "On the parameterization of the local correlation functional. What is Becke-3-LYP?" Chemical Physics Letters 268 (1997):345.

c) Stephens, P.J.; F.J. Devlin, C.F. Chabalowski, and M.J. Frisch. "Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields." *Journal of Physical Chemistry* 98 (1994):11623.

d) Vosko, S.H.; L. Wilk, and M. Nusair. "Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis." *Canadian Journal of Physics* 58(1980):1200.

8. a) Clark, T.; J. Chandrasekhar, G.W. Spitznagel, and P.v.R. Schleyer. "Efficient diffuse function-augmented basis sets for anion calculations. III. The 3-21+G basis set for first-row elements, Li-F." *Journal of Computational Chemistry* 4 (1983):294.

b) Hariharan, P.C. and J.A. Pople. "The influence of polarization functions on molecular orbital hydrogenation energies." *Theoretica Chimica Acta* 28(1973):213.

c) Krishnan, R.; J.S. Binkley, R. Seeger, and J.A. Pople. "Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions." *Journal of Chemical Physics* 72(1980):650.

9. a) Møller, C., and M.S. Plesset. "Note on an Approximation Treatment for Many-Electron Systems." *Physical Review* 46(1934): 618.

b) Pople, J.A., J.S. Binkley, and R. Seeger. "Theoretical models incorporating electron correlation." *International Journal of Quantum Chemistry* S10(1976): 1.

c) Frisch, M. J., M. Head-Gordon, and J. A. Pople. "A Direct MP2 Gradient-Method." Chemical Physics Letters 166(1990): 275.

d) Bartlett, R.J., and D.M. Silver. "Some aspects of Diagrammatic Perturbation Theory." *International Journal of Quantum Chemistry* Symposium 9 (1975): 183.

e) Aikens, C. M., S. P. Webb, R. Bell, G.D. Fletcher, M.W. Schmidt, and M.S. Gordon. "A Derivation of the Frozen-Orbital Unrestricted Open Shell and Restricted Closed Shell MP2 Analytic Gradient Expressions." *Theoretical Chemistry Accounts* 110 (2003): 233.

10. a) Gordon, M.S., M.W. Schmidt as cited in Dykstra, C.E., G. Frenking, K.S. Kim, and G.E. Scuseria. Theory and Applications of Computational Chemistry: The First Forty Years. Amsterdam: Elsevier, 2005.

b) Schmidt, M.W., K.K. Baldridge, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, K.A. Nguyen, S.J. Su, T.L. Windus, M. Dupuis, and J.A. Montgomery. "General Atomic and Molecular Electronic Structure System." *Journal of Computational Chemistry* 14 (1993): 1347.

11. a) Purvis III, G.D., and R.J. Bartlett. "A Full Coupled-Cluster Singles and Doubles Model: The Inclusion of Disconnected Triples." *Journal of Chemical Physics* 76(1982): 1910.

b) Raghavachari, K., G.W. Trucks, J.A. Pople, and M. Head-Gordon. "A Fifth-Order Perturbation Comparison of Electron Correlation Theories." *Chemical Physics Letters* 157(1989): 479.

Computer Simulation of the Mechano-Chemical Behavior of Oxide-Coated Aluminum Nanoparticles

By Brian J. Henz, U.S. Army Research Laboratory, and Michael R. Zachariah, Department of Mechanical Engineering and Department of Chemistry and Biochemistry, University of Maryland, College Park

Introduction

Classical molecular dynamics simulations have been used with the reactive force field (ReaxFF) empirical potential in order to accurately simulate the mechano-chemical behavior of oxide-coated aluminum nanoparticles with diameters of up to 25nm. The simulation results show that homogeneously heated oxide-coated aluminum nanoparticles have relatively low internal pressures, mitigated by diffusion of aluminum cations driven by an induced electric field. Heterogeneously laser-heated nanoparticles on the other hand have appreciably higher computed internal pressures, and these pressures are size dependent, possibly resulting in mechanical failure at nanoparticle diameters larger than considered here. These simulations were performed on the Linux Networx cluster, MJM, at the ARL DSRC and the Cray XT5 system, Einstein, at the Navy DSRC.

For metal nanoparticles, it is well-known that they are pyrophoric and have enhanced energy release rates, which make them attractive in propulsion applications [1]. All metal nanoparticles will nominally have a native oxide shell, which for aluminum is approximately 2-3 nm thick. Thus any oxidative reaction or vigorous combustion must proceed by transport of either the aluminum or oxidizer through the oxide shell.

Nanoparticles have been experimentally shown to react at or near the melting point of bulk aluminum, which is 933K [1], whereas larger particles react closer to the melting point of the oxide shell, namely 2327K. The closeness of the reaction temperature to the melting point of pure aluminum indicates that the melting of the aluminum core is the possible initiator of this reaction. Examples of this theory are the melt dispersion mechanism presented by Levitas et al. [2,3], and the cracking and reduced temperature melting of the oxide shell by Puri and Yang [4]. In the reaction described by Levitas et al., the oxide shell ruptures violently from the internal pressure exerted on it because of the expansion of the aluminum core upon melting. Levitas et al. [2,3] states that for rupture to occur, the ratio of the oxide shell thickness to the nanoparticle radius must be larger than 5 for an assumed shell strength of about 1/2 to 1/3 of the theoretical strength for Al_2O_3. In order to simulate a nanoparticle of the size required to observe the melt dispersion process requires over 600k atoms, a large size considering the computational complexity of the ReaxFF.

Simulation Approach and Model Details

In this work, we have chosen to use ReaxFF, an empirical potential from van Duin [5] implemented within the GRASP (General Reactive Atomistic Simulation Program) parallel MD application. The ReaxFF potential has an advantage over traditional empirical potentials in that it is able to accurately simulate the charge transfer that occurs during the oxidation process.

The model systems considered here consist of a 5.6nm diameter core of aluminum with either a 1nm or 2nm thick shell of Al_2O_3, an 8.2nm aluminum core, and a

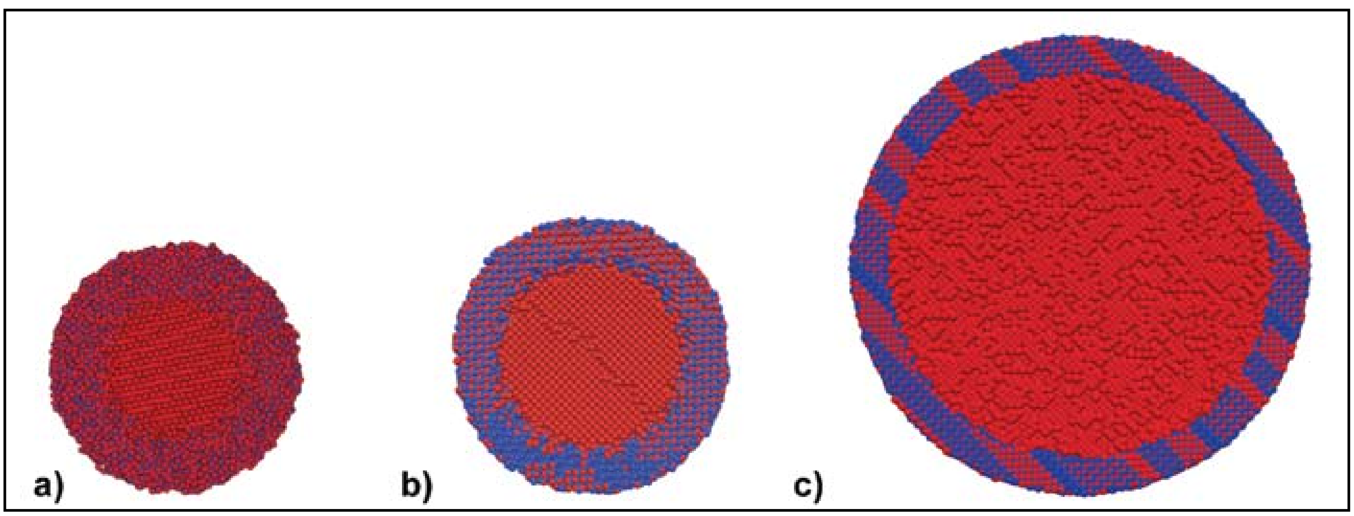


Figure 1. Sample cross sections from a) 5.6nm, b) 8.2nm, and c) 21.0nm nanoparticle cores with oxide shell. Red denotes aluminum atoms, and oxygen atoms are blue

21nm aluminum core both with a 2nm thick crystalline oxide shell as shown in Figures 1a, b, and c, respectively.

After formation and equilibration of the oxide shells, the model systems were heated at a rate of 10^{12}K/s [2,3]. Using the maximum reasonable heating rate is important, as lower heating rates will increase the number of MD simulation time-steps to a level that would be unreasonable to simulate. The limiting factor in decreasing the number of MD steps required is the maximum simulation time-step of 1 fs.

Homogeneous and Heterogeneous Heating Simulation Results

The initial simulations in this work were carried out in a vacuum so that as Al ions reach the outer surface of the oxide shell, they will not be exposed to oxygen molecules for further oxidation reactions. Simulations performed elsewhere indicate that this approximation is valid because the diffusivity of the aluminum cations through the oxide layer is much higher than the diffusivity of the oxygen anions towards the core [6].

During homogeneous heating, a rapid volumetric expansion of the core is observed near the melting point of aluminum. After melting of the aluminum core and reaching 1000K, the oxide shell is still intact; even after holding at this temperature for up to 100ps, we do not observe cracking of the shell. Instead, we observe a large number of the aluminum cations from the core beginning to diffuse to the surface of the nanoparticle, Figure 2.

The simulation results in Figure 2 offer some insight into the mechanism by which oxidation proceeds at elevated temperatures for homogeneously heated nanoparticles. The first observation is that the oxide shell does not crack as one might expect if diffusion were extremely limited or the shell were brittle and stress in the oxide shell because of core expansion were high. This suggests that the shell is more elastic at this scale than observed in the bulk material; the expansion of the aluminum is insufficient to cause failure in the shell; or diffusion of aluminum ions through the oxide shell mitigates the pressure buildup in the core. For comparison with the experimental single particle mass spectrometry method, where a laser is used to rapidly heat the oxide-coated aluminum nanoparticle, we have also simulated heterogeneous heating of oxide-coated aluminum nanoparticles [7,8].

Computed Stress in Oxide Shell

In Figure 3, the pressure is computed for the 8.2nm nanoparticle model with a 2nm thick, crystalline oxide

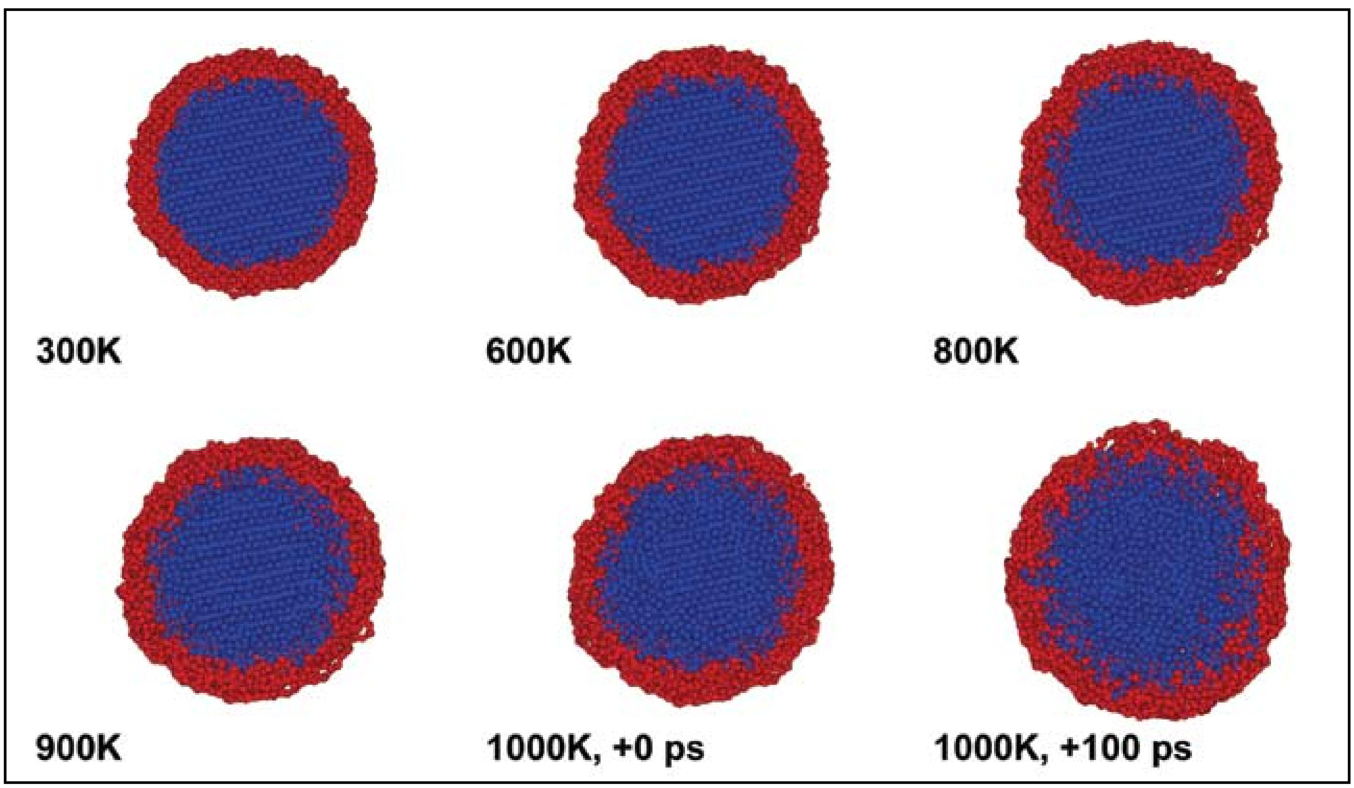


Figure 2. Plot showing diffusion of aluminum cations (blue) through the oxide shell (red) as the temperature increases from 300K to 1000K and held for 100 ps

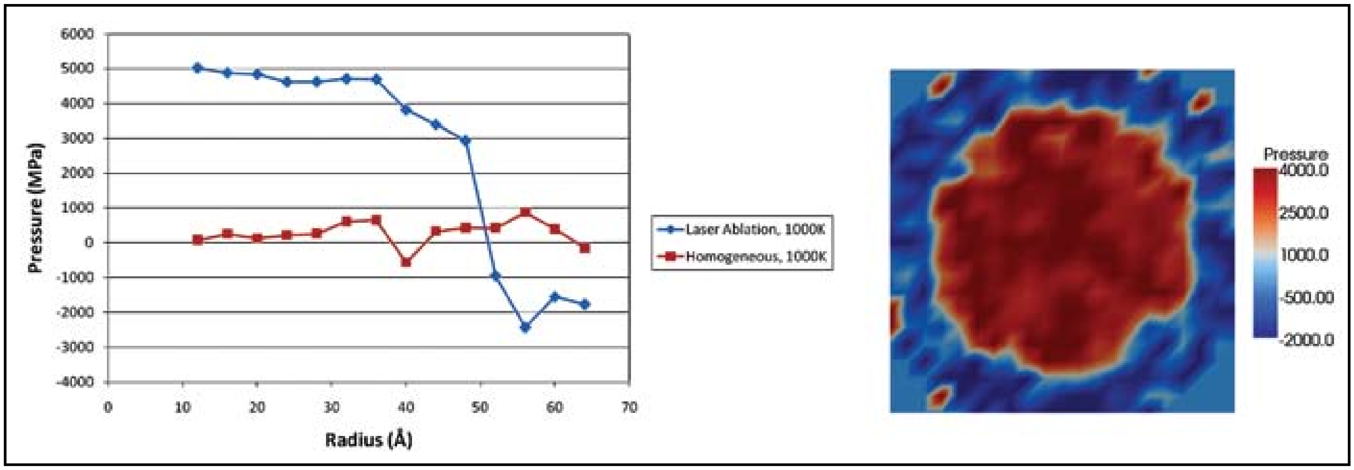


Figure 3. Pressure profile averaged over 8ps at 1000K for homogeneous heating and heterogeneous laser heating of an 8.2nm aluminum core with 2nm crystalline oxide shell. Pressure units are MPa

shell. Immediately evident in these pressure results, Figure 3 (left), is the difference in pressure between the two systems. For the homogeneously heated nanoparticle case, we do not observe the same high magnitude pressures present in the laser-heated particle.

In order to visualize the steep pressure gradient at the core/shell interface, we have computed the virial stress within 0.6nm voxels over 2ps intervals [9] for the 8.2nm core model. In Figure 3 (right), a cross section of the 8.2nm core heated to 1000K by laser is shown. The high-pressure gradient at the core/shell interface reflects the steep slope observed on the chart in Figure 3. The core is experiencing high pressures because of the volumetric expansion upon melting of aluminum while the oxide shell is under high negative pressure (tensile stress). We do not observe mechanical failure of the oxide shell under these conditions either, but simulations of the large, 21nm core with the 2nm thick oxide shell, are underway to make a more direct comparison with the predictions of Levitas et al. [2,3]

Using the Cray XT5 (Einstein)

When Einstein came online in January 2009, I quickly acted to activate my account in order to take advantage of the over 12k processor cores to simulate the 21nm aluminum core model with its 600k atoms. In our past efforts with Cray systems, we have found them to scale extremely well. The type of scaling that I am concerned with is increasing processor core count while maintaining a constant problem size. This is because we typically have a physical problem we wish to solve, in this case homogeneous heating of a 25nm nanoparticle; and we hope to do it quickly. This scaling is reported in lieu of another scaling analysis often discussed, which is in terms of increasing problem size with number of processor cores, the so-called "efficiency conserved model."

According to the documentation from Cray, the XT systems use a linux kernel on the compute nodes that does not explicitly support loading of shared libraries. In order to get around this limitation, I have developed the script in Figure 4 where the required shared libraries are copied into the temporary execution directory, and the LD_LIBRARY_PATH variable is set to point at this location. In this way, I have been successful in using shared libraries with the Cray XT5 system at the Navy DSRC.

The script in Figure 4 follows the documentation at the Navy DSRC except for the use of the LD_LIBRARY_PATH variable.

This Cray XT5, like other Cray systems we have encountered, appears to scale well for a fixed problem size. In Figure 5, we have plotted the execution time for 500 times-steps of a 92k atom model using GRASP with the ReaxFF potential. Notice that the Cray system continues to scale up to 1024-processor cores, and although it is slower per core, its scalability and large number of available cores allow rapid simulation of large MD models.

In Figure 5, the super linear speedup between 64 and 128 cores on the Cray XT5 is likely due to the storage requirements on each core now fitting into the processor cache rather than main memory. The jump in simulation time at 64 and 256 cores on MJM is repeatable, but a cause is as yet undetermined. Performance results for 512 cores on MJM shows that the system has likely become communication bound at 256 cores, and further performance improvements by increasing the core count is not possible.

```
#! /bin/csh
#PBS -N ReaxFF
#PBS -o Al2O3_8nm_core_2nm_shell_crystalline.1000K.1024.182000.o
#PBS -e Al2O3_8nm_core_2nm_shell_crystalline.1000K.1024.182000.e
#PBS -A ARLAP800
#PBS -l walltime=8:00:00
#PBS -l mppwidth=1024
#PBS -l mppnppn=8
#PBS -q standard
# ----------------------------------
#
# Set appropriate data and output directories variable names
setenv DATADIR $WORKDIR/Al2O3_8nm_core_2nm_shell_crystalline/1000K/timing/1024
setenv OUTPUTDIR $WORKDIR/Al2O3_8nm_core_2nm_shell_crystalline/1000K/timing/1024
# Create the directories
#
mkdir -p $DATADIR
mkdir -p $OUTPUTDIR
#
# Copy executable, shared libraries, and any required input files to your work
directory
# located under the /scr Lustre filesystem.
setenv GRASP_HOME $HOME/CVS/CELST/Support/src/Grasp_reax
cp $GRASP_HOME/Source/Obj_navo_einstein_pathscale/grasp.exe $DATADIR
cp $GRASP_HOME/{fort.4,types.in} $DATADIR
cp $GRASP_HOME/LIBS_pathscale/* $DATADIR
cp $GRASP_HOME/SZTests/papers/Al2O3_8nm_core_2nm_shell_crystalline/{inp.
dat,restart.in} $DATADIR
cd $DATADIR
#
# Set LD_LIBRARY_PATH
#
setenv LD_LIBRARY_PATH ${DATADIR}:$LD_LIBRARY_PATH
#
# Run the MPI job with the CRAY "aprun" command:
#
aprun -N 8 -n 1024 ./grasp.exe
#
# Archive any output from the job to your home directory on EINSTEIN
# or to your home directory on the archive server, NEWTON.
#
# In this example, user "skinman" is homed on NEWTON at
# /u/home/skinman.
#
#/usr/bin/rcp $OUTPUTDIR/* newton:$ARCHIVE_HOME/
#/usr/bin/rcp restart newton:$ARCHIVE_HOME/restart
# ----- End of sample PBS script -----
```

Figure 4. Job submission script developed on Einstein in order to use shared libraries on the Cray XT5

Conclusions

For oxide-coated aluminum nanoparticles with core radius to shell thickness ratios of up to 5, we have found that the oxidation process occurs by rapid diffusion of aluminum cations through the oxide shell driven by an induced electric field as opposed to mechanical failure or melting of the oxide. For the range of nanoparticle sizes investigated here, the simulation results agree well with Levitas's mechanical analysis, i.e., shell failure does not occur for small nanoparticles. Ongoing simulations with radius to shell ratios of over 5 should be able to predict when mechanical failure will occur. The scalability and availability of processing cores on the Cray XT5 system Einstein has allowed us to investigate atomic systems large enough to make direct comparisons with experimental and analytical results.

Acknowledgments

The authors would like to acknowledge the support received by the U.S. Army Research Laboratory DoD Supercomputing Resource Center (DSRC) and the U.S. Navy DSRC. Additional support was provided by the National Institute for Standards Technology (NIST) and the U.S. Army Research Office (ARO).

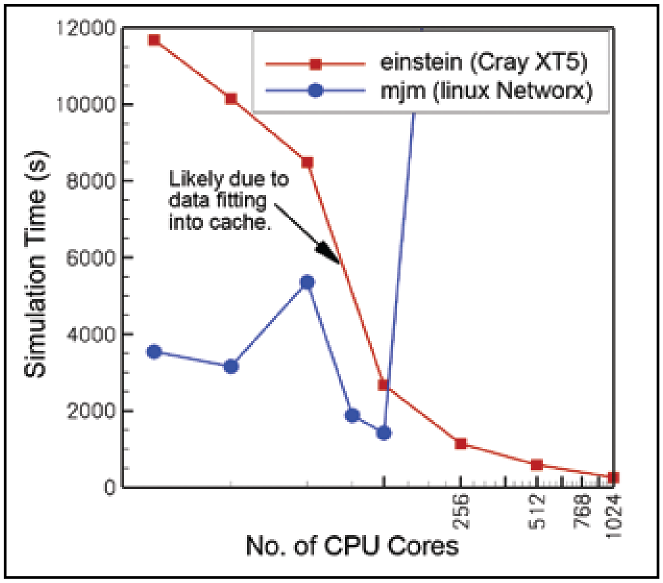


Figure 5. Scalability plots for Linux Networx cluster MJM and Cray XT5 system Einstein

References

1. A. Rai, D. Lee, K. Park, and M. R. Zachariah. "Importance of Phase Change of Aluminum in Oxidation of Aluminum Nanoparticles," *Journal of Physical Chemistry B*, 108, 14793–14795, 2004.

2. V.I. Levitas, B.W. Asay, S.F. Son, and M. Pantoya. "Melt dispersion mechanism for fast reaction of nanothermites," *Journal of Applied Physics*, 89, 071909, 2006.

3. V.I. Levitas, B.W. Asay, S.F. Son, and M. Pantoya. "Mechanochemical mechanism for fast reaction of metastable intermolecular composites based on dispersion of liquid aluminum," *Journal of Applied Physics*, 101, 083524, 2007.

4. P. Puri and V. Yang. "Thermo-Mechanical Behavior of Nano Aluminum Particles with Oxide Layers," 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 7-10 January 2008.

5. A.C.T. van Duin, S. Dasgupta, F. Lorant, and W.A. Goddard III. "ReaxFF: A Reactive Force Field for Hydrocarbons," *Journal of Physical Chemistry A*, 105, 9396–9409, 2001.

6. B.J. Henz, T. Hawa, and M.R. Zachariah. "On the Role of Built-in Electric Fields on Oxidation of Oxide Coated NanoAluminum: ion mobility versus Fickian Diffusion," *Physical Review B*, Submitted for publication.

7. L. Zhou, A. Rai, and M.R. Zachariah. "Component and Morphology Biases on Qualifying Size and Composition of Nanoparticles Using Single Particle Mass Spectometry," *International Journal of Mass Spectometry*, 258, 104–112, 2006.

8. L. Zhou, K. Park, H. Milchberg, and M.R. Zachariah. "Understanding the Interaction of an Intense Laser Pulse with Nanoparticles: Application to Quantification of Single Particle Mass Spectometry," *Aerosol Science and Technology*, 41, 818–827, 2007.

9. T.J. Campbell, G. Aral, S. Ogata, R.K. Kalia, A. Nakano, and P. Vashishta. "Oxidation of aluminum nanoclusters," *Physical Review B*, 71, 205413, 2005.

Predicting "Ocean Weather" Using the 1/12° Global HYbrid Coordinate Ocean Model (HYCOM)

E. J. Metzger, H. E. Hurlburt, A. J. Wallcraft, Naval Research Laboratory; O. M. Smedstad, QinetiQ North America/Planning Systems, Inc.; J. A. Cumming , Naval Research Laboratory; and E. P. Chassignet, Florida State University

Introduction

The development of an advanced global ocean prediction system has been a long-term Navy interest. Such a system must provide the capability to depict (nowcast) and predict (forecast) the oceanic "weather," some components of which include the 3-D temperature (T), salinity (S) and current structure, the surface mixed layer, and the location of mesoscale features such as eddies, meandering currents, and fronts. The space scale of these eddies and meandering currents is typically ~100 km, and current speeds can easily exceed 1 ms^{-1} in the Gulf Stream (Atlantic Ocean) and Kuroshio (Pacific Ocean). Numerical ocean models with sufficiently high horizontal and vertical resolution are needed to depict the 3-D structure with accuracy superior to climatology and persistence (i.e., a forecast of no change). The accelerated development of these prediction systems would not have been possible without the computational resources provided by the DoD HPCMP. Throughout the research and development stages of numerical ocean models and data assimilation techniques, HPC has played a key role. This is especially true with regard to grand challenge projects that allowed development of high horizontal resolution global systems long before it became feasible to run them in an operational environment. In addition, nonchallenge and Capability Application Projects have also provided considerable resources toward advancement of these systems.

The existing two-model operational global ocean prediction system, run daily at the Navy DSRC, is based on the 1/8° Navy Coastal Ocean Model (NCOM) and the 1/32° Navy Layered Ocean Model (NLOM). Unlike NLOM, NCOM has high vertical resolution, but it has medium-range horizontal resolution (~15 km at mid-latitudes near 40°N), which makes it eddy-permitting. The next-generation system is based on a single model, the HYbrid Coordinate Ocean Model (HYCOM) (Bleck, 2002). It was developed as part of a multi-institutional consortium between academia, Government, and private industry. At 2.2 times the horizontal resolution of NCOM, the HYCOM-based system is eddy-resolving, a distinction associated with important dynamical implications for both ocean model dynamical interpolation skill in the assimilation of ocean data and for ocean model forecast skill (Hurlburt et al., 2008). HYCOM is also uniquely designed to allow an accurate transition between deep and shallow water, historically a challenging problem for ocean models. Its generalized hybrid vertical coordinate is a substantial advance over the vertical coordinate system

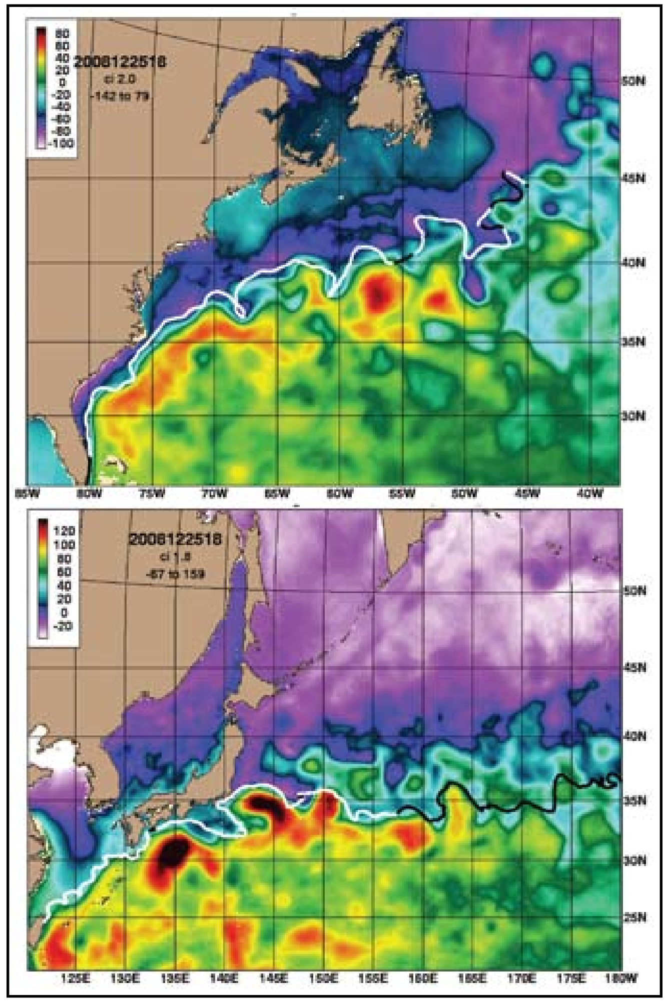


Figure 1. Sea surface height (cm) from the 1/12° global HYCOM prediction system for the Gulf Stream in the Atlantic Ocean (top) and the Kuroshio in the Pacific Ocean (bottom) on December 22, 2008. The ribbon of high gradient color shows the location of these western boundary currents; embedded within the meandering flow are warm and cold core eddies. The currents generally flow parallel to the isolines of height and are strongest where the gradients are the tightest. An independent infrared (IR) analysis of the north edge of both current systems is performed by NAVO and overlain on each image. A white (black) line means the IR analysis is based on data less (more) than 4 days old

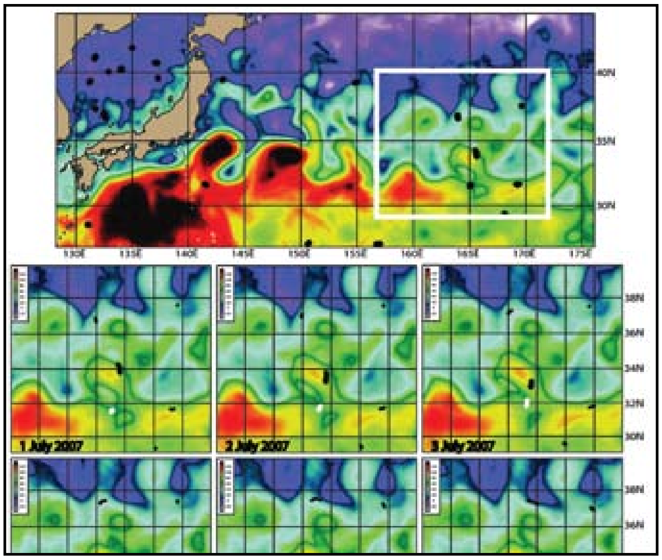


Figure 2. Sea surface height (cm) from the 1/12° global HYCOM prediction system for the Kuroshio on July 1, 2007 (top). Drifting buoy tracks over a 1-day time period are overlain on each panel. The white box defines the focused area of the bottom six panels that span the time frame July 1-6, 2007. A warm core eddy is about to detach from the Kuroshio, and two drifting buoys (highlighted in white and black) are traversing its western and eastern sides

used in NCOM. The HYCOM-based system represents the world's first eddy-resolving global ocean prediction system with both high horizontal and vertical resolution and has been validated against observational data (Metzger et al., 2008). It is scheduled for operational testing in 2009.

HYCOM Description

HYCOM is on a 1/12° global grid (mid-latitude resolution of ~7 km) with 32 hybrid vertical coordinate surfaces. The truly generalized vertical coordinate can be isopycnal (density tracking – often best in the deep stratified ocean), levels of equal pressure (nearly fixed depths – best used in the mixed layer and unstratified ocean) or terrain-following (often the best choice in shallow water). HYCOM combines all three approaches by choosing the optimal distribution at every grid point and time-step. The hybrid coordinate extends the geographic range of applicability of traditional isopycnic coordinate models toward shallow coastal seas and unstratified parts of the world ocean. It maintains the significant advantages of an isopycnal model in stratified regions while allowing more vertical resolution near the surface and in shallow coastal areas, hence providing a better representation of the upper ocean physics.

HYCOM employs the Navy Coupled Ocean Data Assimilation (NCODA) (Cummings, 2005), which is a fully 3-D multivariate optimum interpolation scheme, to assimilate observational data. The data include surface observations from satellites, including altimeter sea surface height (SSH) anomalies, sea surface temperature (SST), and sea ice concentration, plus in situ SST observations from ships and buoys as well as T & S profile data from XBTs, CTDs, and Argo profiling floats. The 3-D ocean environment can be more accurately nowcast and forecast by combining these diverse observational data types via data assimilation and using the dynamical interpolation skill of the model.

The 1/12° global HYCOM-based prediction system has been running daily in pre-operational mode at the Navy DSRC since December 22, 2006. Originally running on the IBM machines (Romulus – Power4+ and then Babbage – Power5+), the system was recently moved to the Cray XT5 (Einstein). Here it is presently configured to use 78 nodes (619 processors) to run HYCOM and produce the NCODA analyses with an additional two nodes set aside for pre- and post-processing.

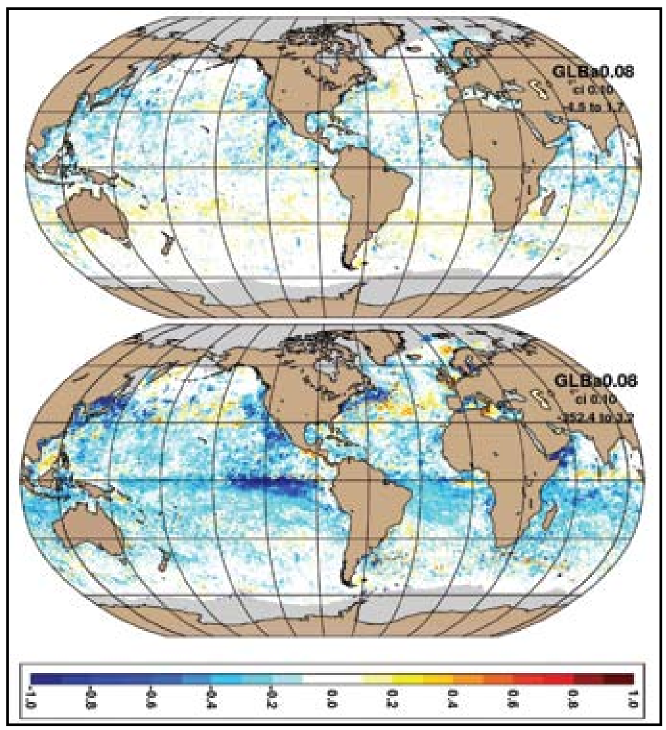


Figure 3. Sea surface temperature (SST) mean error relative to ~33,000,000 MCSST observations from the 1/12° global HYCOM prediction system at the analysis time (top) and for a 3-day forecast (bottom) over a year-long hindcast spanning June 2007-May 2008. Red (blue) colors indicate nowcast and forecast SST that is warmer (cooler) than observed. Values between ±0.1˚C are white. The gray area near the poles is an annual average ice coverage mask

HYCOM efficiently scales to large processor counts and can easily be configured to fit within the allocated resource window. Each day the system starts 5 days in arrears of the nowcast time (to assimilate all available late-arriving observational data) and then runs forward to create a 5-day forecast. It generates 3-D whole domain instantaneous archive files at 00Z each model day that are ~10 Gb.

Real-Time Results

Where possible, the HYCOM-based prediction system is evaluated using independent observations, and some examples follow. Figure 1 shows simulated SSH for the Gulf Stream and the Kuroshio systems. The assimilation of satellite altimeter SSH anomalies is essential to accurately map the circulation in these highly chaotic regions dominated by mesoscale flow instabilities. Infrared-based frontal analyses that show the northernmost edge of the currents are overlain on the panels. They provide an independent analysis of the current positions and clearly indicate the ocean nowcast/forecast system is accurately mapping these western boundary currents. Figure 2 shows an example that uses drifting buoy trajectories to validate the flow field in the Kuroshio. Drifting buoy temperature (but not velocity) is assimilated via NCODA, allowing the trajectory to be an independent validation source. The white box focuses on a warm core eddy about to detach from the Kuroshio, and a pair of drifting buoys is noted on the western and eastern sides. These two drifters pass within a half degree of each other while traveling in opposite directions. Close examination indicates the two buoys are on opposite sides of a saddle point that still connects the main current with the detaching eddy. Thus, the system is able to accurately assimilate the altimeter data and act as a dynamical interpolator. Lastly, SST forecast skill of the system is examined.

Table 1. SST error statistics as a function of forecast length from the 1/12° global HYCOM prediction system compared against ~33,000,000 satellite-based observations. The analysis is performed over a year-long hindcast spanning June 2007-May 2008 and is limited to the area between 45 S – 45 N

	Mean error	RMSE
Analysis	-.02	.36
1-day forecast	-.09	.44
2-day forecast	-.14	.52
3-day forecast	-.18	.60
4-day forecast	-.22	.67
5-day forecast	-.26	.72

Table 1 shows the mean error (bias) and root-mean-square-error (RMSE) as a function of forecast length. The bias and RMSE gradually grow with forecast length. The spatial distribution of the mean error is shown in Figure 3 for the analysis time and a 3-day forecast. In hindcast mode, the global HYCOM system has also demonstrated forecast skill on time scales up to a month for the meandering currents and eddies in some regions (not shown).

Impact

A next-generation ocean nowcasting/forecasting system based on 1/12° global HYCOM is running in real-time at the Navy DSRC. It can accurately depict and forecast such features as western boundary currents and sharp ocean fronts, thus providing improved environmental awareness to the Fleet. Other naval applications include optimum track ship routing, search and rescue, anti-submarine warfare and surveillance, tactical planning, and providing boundary conditions for regional and coastal nested model. HPC resources have played a major role in making this state-of-the-art system feasible, beginning with the preliminary development of HYCOM and continuing all the way through its transition as a pre-operational product.

References

Bleck, R., 2002. An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates. *Ocean Modelling*, 4, 55-88.

Cummings, J.A., 2005. Operational multivariate ocean data assimilation. *Quart. J. Royal Met. Soc*., 131, 3583-3604.

Hurlburt, H.E., E.P. Chassignet, J.A. Cummings, A.B. Kara, E.J. Metzger, J.F. Shriver, O.M. Smedstad, A.J. Wallcraft, and C.N. Barron, 2008. Eddy-resolving global ocean prediction. In: Hecht, M., Hasumi, H. (Eds.), Ocean Modeling in an Eddying Regime, Geophysical Monograph 177. *American Geophysical Union*,Washington, DC.

Metzger, E.J., O.M. Smedstad, P. Thoppil, H.E. Hurlburt, A.J. Wallcraft, D.S. Franklin, J.F. Shriver and L.F. Smedstad, 2008. Validation Test Report for Global Ocean Prediction System V3.0 – 1/12° HYCOM/NCODA: Phase I, NRL Memo. Report, NRL/MR/7320—08-9148.

Simulation of Space Object Time-Resolved Signatures

By Cadet Katharine I. Kalamaroff and Francis K. Chun, Department of Physics, United States Air Force Academy

Cadets and faculty from the United States Air Force Academy along with the Air Force Research Laboratory have developed a high performance computing modeling tool that simulates the time-varying photometric signature from a satellite as observed from any point on the earth. This tool directly supports research in nonresolvable space object identification, which is critical to space situational awareness. The emphasis of our work this year was to adapt a serial version of the modeling tool to run in a parallel environment. We architected an embarrassingly parallel version and benchmarked it on the Cray XD-1 Linux supercomputer (Hoku) at the Maui DoD Supercomputing Center. The high performance computing version will ultimately provide the means to easily and efficiently explore the large parameter search space of conditions that govern a satellite's reflected light.

Introduction

Space situational awareness is concerned with knowing as much about earth-orbiting space objects as possible. When ground-based sensors are large enough or space objects are large in size and low in altitude, it is possible to optically image the space object so that one can resolve distinguishing features. As space objects become farther away or smaller in size though, obtaining high-resolution images becomes harder. However, we can still extract information about a space object from its time-resolved photometric signature. If we could observe the same space object from more than one photometric site simultaneously, more data would obviously be collected; but would there be more information available? What is the optimum location of the observing sites to maximize the amount of information collected? And, what does "maximize information" mean? What is the optimum ratio of inclusive vs. exclusive information (e.g., the blind men and the elephant)? How do you place the sites relative to the satellite's ground track, the terminator, or each other? Is there optimum pass geometry (e.g., angle between terminator and ground track)? To start answering these types of questions, the Air Force Research Laboratory and Air Force Academy developed a photometric modeling software tool.

Modeling Tool

The multisite photometric modeling tool is written in MATLAB and was initially configured for use on a desktop or laptop computer in a serial mode. Through graphical user interfaces, a user can specify a two-line element to simulate a satellite orbit, pick a location for a sensor site, design a simple satellite body in terms of shape and materials, orient the satellite attitude with respect to the earth, and define the model output such as time resolution, graphics, and movies. In its current form, the modeling tool does well in simulating the time-resolved photometric signature of a satellite that a ground sensor would measure. However, using the serial version of the modeling tool to investigate the enormous parameter space that governs a photometric signature would take a long time to complete. Thus a parallel version of the tool is necessary to quickly explore the various combinations of input parameters.

Parallelization Efforts

We designed and developed two versions of the parallel code, each with a different purpose. The first parallel code version examined how the light curve from one satellite would vary over different days of the year when measured at the same sensor site (see Figure 1). The second version compared the light curves from a specific satellite and

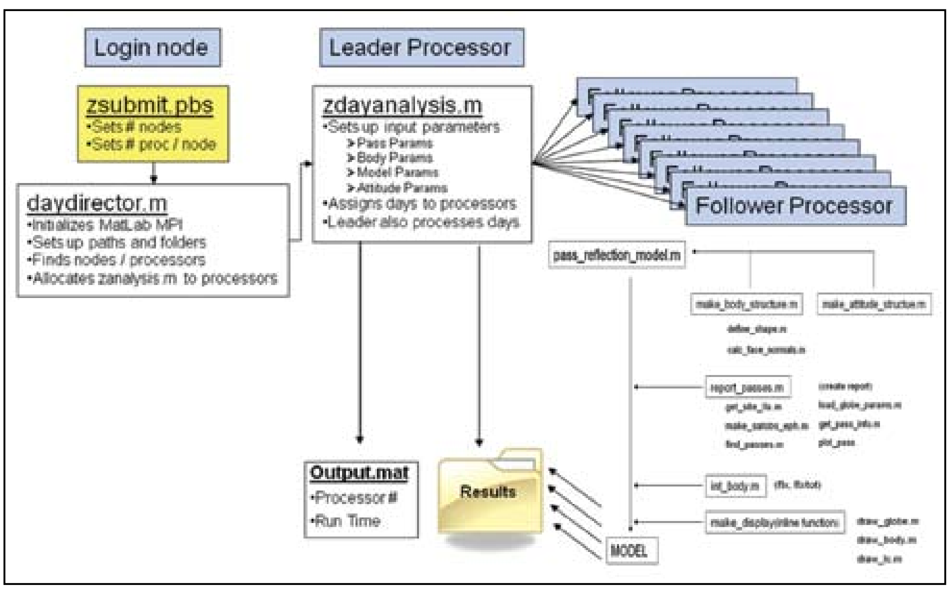


Figure 1. Schematic of version 1 of the parallel code whereby the leader processor assigns days of the year to the follower processors. Version 2 of the code is similar except the leader processor assigns sensor locations to the follower processors

pass at different sensor site locations. Both versions have the basic embarrassingly parallel software architecture with a login node, leader processor, and follower processors. At the login node, there is a "director" program that initializes Matlab MPI, sets up paths and folders, and finds the nodes and processors. The leader processor then reads a text file, which sets up the input parameters for the model run and assigns either the days (version 1) or sensor locations (version 2) to the follower processors. All the follower processors then execute the same program to generate the simulated light curve.

In all this work, we analyzed the load balance and speedup achievable by running in a parallel-processing fashion compared with a serial desktop computer. Figure 2 shows the load balance of version 1 of the parallel code where the number of days of the simulation is allocated over 12 processors. The results are shown for a short run of 31 days (top plot) and a long run of 365 days (bottom plot). For the short run, the maximum difference in processing time is 0.93 minutes, which is 22 percent of the total processing time. For the long run, although the maximum difference in processing time is more (18.83 minutes), it accounts for a smaller percentage (17 percent) of the total processing time. Figure 3 shows the results of the speedup analysis using up to 12 processors for a run of 31 days with approximately one orbital pass per day. The top plot is the processing time per run showing that 1 processor takes 70 minutes to complete the job, whereas 12 processors reduce the time to 10 minutes. The bottom plot shows that same information in terms of speedup where 2 processors are <2× faster and 12 processors are 7× faster. One can see, however, that there appears to be a limit in the speedup of the code using this particular software architecture to about 12 processors. Speedup may be limited by the equitability in the distribution of passes amongst follower processors.

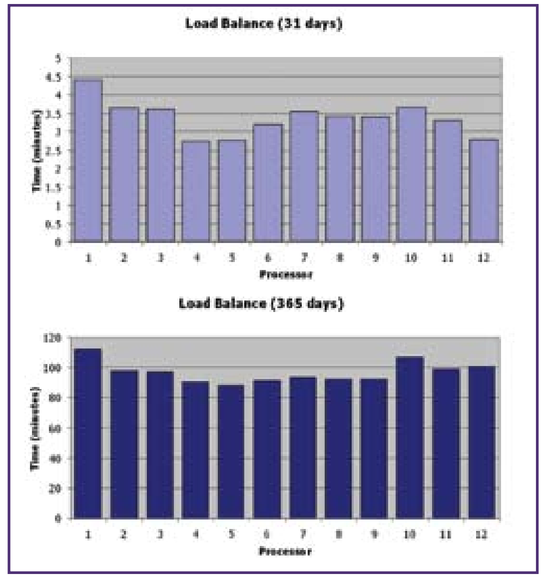


Figure 2. Load balance of version 1 of the parallel code with the number of days allocated to 12 processors. Top plot is for a short run of 31 days; bottom plot is a longer run of 365 days

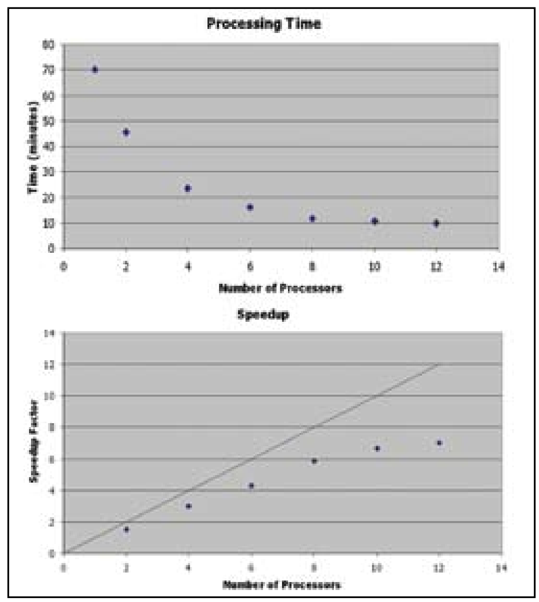


Figure 3. Speedup analysis comparing the processing time for up to 12 processors. Analysis was conducted for 31 days with approximately one orbital pass per day

Conclusion

The Air Force Academy and Air Force Research Laboratory are making significant progress in developing a high performance computing tool to simulate the time-varying light curve from any satellite model as observed from anywhere on the earth. Significantly, cadets from the Air Force Academy are learning about space situational awareness by developing and using this tool. Future work will entail further development of version 2 of the parallel code, the multisite version, as well as development of analysis tools to work with and understand the vast amount of data that this tool will produce.

Acknowledgment

This work is sponsored by the DoD HPCMP, the AFRL Directed Energy Directorate and Air Force Office of Scientific Research. The authors also would like to thank Dr. Concetto Giuliano (AFRL/RDS), Mr. Paul Kervin (AFRL/RDSM), Dr. Doyle Hall (Boeing LTS), and the staff of the MHPCC DSRC for their valuable advice, recommendations, and insights on this work, and for providing the initial serial software tool.

UIT Version 3.0

By Wes Monceaux, ERDC Information Technology Laboratory (ITL); Scotty Swillie, ERDC DSRC; and Keith Rappold, ITL

With the Program-wide move to CAC for Kerberos authentication, major changes were required within the User Interface Toolkit's (UIT) architecture to accommodate the physical interaction with the CAC reader. The UIT anywhere, anytime, Web-only access model will not work with a CAC authentication. This meant a serious redesign of the UIT security access model had to take place to ensure this tool remained available to DoD HPCMP users and developers.

The UIT development team at ERDC accepted this requirement as an opportunity to enhance the usability of the UIT. While continuing to pursue a solution for the continuation of a centralized Web service API that can support multitiered Web applications, the team has also devised a new flavor of the UIT that is more extensible and robust for Web-based client interface applications. This new direction led them to develop a version of the UIT that is available as a local Web service available on each user's machine – Version 3.0. This new version is now available in beta, while the centralized Web-centric version will be rolled out at the end of the fiscal year and will be labeled Version 3.1.

Version 3.0 Overview

Version 3.0 will require a one-time download that will come prepackaged with a small Web server that will run accessible only from the local system. Developers will code against this API the same way that it is coded against now with the exception of user authentication. Authentication will take place via `pkinit` or `kinit` using the DSRC Kerberos client kit, just as it is done for any other application that requires Kerberos. From that point forward, API calls by a user interface to the UIT will just verify that the user has a ticket and will then simply use that ticket to perform actions on the user's behalf on the HPC machines.

UIT version 3.0 will require users to have Java Runtime version 1.5+ and the DSRC Kerberos toolkit installed for their platform.

Here is how the process will work:

1. Users will use their SecurID card or their CAC to authenticate using the DSRC Kerberos `kinit`, `pkinit`, or similar command. Once authenticated, the client Java application running on the user's desktop will then issue commands (using `ssh`) to interact with the HPC systems.
2. The HPC systems will treat the `ssh` connections from the interface no differently than any other connections made by the user.

The UIT will effectively behave just like any of the other DSRC tools (i.e., Filezilla, PuTTY, etc.) and utilize the user's existing Kerberos tickets to interact with the HPC systems.

Version 3.0 of the UIT will interface with the native Kerberos libraries (the DSRC Kerberos Kit) and utilize the same tickets opaquely as any of the other DSRC-supplied tools do (Filezilla, PuTTY, etc.). This makes the UIT oblivious to how a user acquired his Kerberos ticket (CAC, HToken, or SecurID).

Centralized UIT information will still be maintained in the UIT database. This includes all user-specific pIE data and including the systems to which they have access. Also, it includes the configuration information we house for each system to provide a uniform interface to them – i.e., queuing information, system availability, etc. Updates to the UIT will be downloaded and installed automatically when users login.

Version 3.0 Benefits

Aside from CAC authentication, there are other benefits that come with the way Version 3.0 interacts with the API. Direct connections from the user's machine to the HPC machines will now be possible allowing for better performance when manipulating large files. Since the API is now accessed via localhost, method calls should respond much faster than before.

Benefits to moving UIT v3.0 in this direction are as follows:

- Use of CAC and any DSRC-supported authentication.
- Direct file transfer and communication between client and HPC systems.
- Security architecture greatly simplified.
- Users' Kerberos ticket life not limited to 1 hour (depending on network).
- Method calls should respond faster than before, since the API is now accessed via localhost.

Version 3.1 Overview

As with any situation where tradeoffs must be made, there are issues that will need to be addressed. The biggest and most obvious with UIT Version 3.0 is that Web-site-type UIT clients that authenticate users by passing their credentials on to the UIT Web service will not work with Version 3.0. For this reason, we are continuing the development process to include that capability in Version 3.1.

UIT Version 3.1 will make use of existing Kerberos libraries to enable CAC authentication to get a ticket that can be used via the Web. This will allow Web clients to continue to use the UIT in much the same way that it does today. While it will likely require Web developers to make some changes to their code to accommodate the CAC authentication, the rest of the API will work the same as it does today.

The details of this architecture are still being worked out, but we expect to have a beta version available at the end of this fiscal year.

Client Development Support

Two of the Program DSRCs, Maui and ARSC, will soon be offering support for UIT client development for Versions 3.0 and 3.1. We are in the process of determining how that support will be provided, and we will release more information as the details are worked out.

Valgrind's Cachegrind Profiler Tool

By Craig Stephenson, ARSC DSRC

"Cachegrind," according to Valgrind's User Manual, "is a tool for finding places where programs interact badly with typical modern superscalar processors and run slowly as a result." In other words, Cachegrind analyzes an executable's low-level instruction and data operations, reporting the number of cache misses along the way. This information is critical to the performance-minded, as reducing the number of cache misses can equate to substantial improvements in speed. Cachegrind can generate reports as a summary of the entire program, per function, or line by line.

Cachegrind is one of many tools bundled in the Valgrind application, which is installed and maintained in the $PET_HOME directory on all general access DSRC clusters. The default version of Valgrind can be accessed from the following location:

$PET_HOME/bin/valgrind

Running Cachegrind with its default options will produce a summary of the entire program. For example:

```
> valgrind --tool=cachegrind ./example
...
==2950== I   refs:      1,999,515
==2950== I1  misses:        1,110
==2950== L2i misses:        1,090
==2950== I1  miss rate:      0.05%
==2950== L2i miss rate:      0.05%
==2950==
==2950== D   refs:        594,114  (462,989 rd   + 131,125 wr)
==2950== D1  misses:       16,918  ( 15,298 rd   +   1,620 wr)
==2950== L2d misses:        7,868  (  6,641 rd   +   1,227 wr)
==2950== D1  miss rate:       2.8% (    3.3%     +     1.2%  )
==2950== L2d miss rate:       1.3% (    1.4%     +     0.9%  )
==2950==
==2950== L2 refs:          18,028  ( 16,408 rd   +   1,620 wr)
==2950== L2 misses:         8,958  (  7,731 rd   +   1,227 wr)
==2950== L2 miss rate:        0.3% (    0.3%     +     0.9%  )
...
```

(Note: As in this example, much of Valgrind's output is prefixed with the process ID.)

A quick key to interpret this output is as follows:

```
  I/i = instructions
  D/d = data
  I1 - level 1 instruction cache
```

```
D1 - level 1 data cache
L2 = level 2 shared instruction/data cache
rd = data read
wr = data write
```

This example had a level 1 data cache miss rate of 2.8 percent and a level 2 cache data miss rate of 1.3 percent. Not too bad.

A textbook example of the benefits of cache optimization comes from the distinction between row-major and column-major programming languages. To traverse the memory of a multi-dimensional array sequentially in a row-major language such as C or C++, the program should access each row in order. Conversely, the memory of a multi-dimensional array in column-major languages, such as Fortran, is sequenced by columns. Let's use Cachegrind to put this to the test with the following two equivalent programs:

Row-Major/Column-Major Comparison in C

```
----------------------------------------
#include <stdio.h>

void rowMajor()
{
  int A[1000][100][10];
  int i, j, k;

  for(i=0; i < 1000; i++)
  {
    for(j=0; j < 100; j++)
    {
      for(k=0; k < 10; k++)
      {
        A[i][j][k] = i + j + k;
      }
    }
  }
}

void columnMajor()
{
  int A[1000][100][10];
  int i, j, k;

  for(k=0; k < 10; k++)
  {
    for(j=0; j < 100; j++)
    {
      for(i=0; i < 1000; i++)
      {
        A[i][j][k] = i + j + k;
      }
    }
  }
}

int main()
{
  rowMajor();
  columnMajor();

  return 0;
}
----------------------------------------
```

Row-Major/Column-Major Comparison in Fortran 90

```
---------------------------------------------
PROGRAM major
IMPLICIT NONE

  CALL rowMajor()
  CALL columnMajor()

END PROGRAM major

SUBROUTINE rowMajor()
IMPLICIT NONE
  INTEGER, DIMENSION(1000,100,10) :: A
  INTEGER :: i, j, k

  DO i = 1,1000
    DO j = 1,100
      DO k = 1,10
        A(i,j,k) = i + j + k
      END DO
    END DO
  END DO

  RETURN
END

SUBROUTINE columnMajor()
IMPLICIT NONE
  INTEGER, DIMENSION(1000,100,10) :: A
  INTEGER :: i, j, k

  DO k = 1, 10
    DO j = 1, 100
      DO i = 1, 1000
        A(i,j,k) = i + j + k
      END DO
    END DO
  END DO

  RETURN
END
---------------------------------------------
```

Each program performs the same operation using both orders, so viewing a summary of the entire program will not be terribly helpful. To see which of the two functions performs faster, we would be well advised to generate a per-function report using the cg_annotate command, via the following series of commands:

Using a PGI compiler, debugging information can be included without sacrificing optimizations by using the "-gopt" option:

```
> pgcc -gopt -o c_major major.c
```

Then, run the compiled executable using Valgrind's Cachegrind tool:

```
> valgrind --tool=cachegrind ./c_major
```

This will display the program summary in addition to writing a file named cachegrind.out.####, where #### is the process ID. Use this file as a parameter to the cg_annotate program:

```
> cg_annotate cachegrind.out.7902
...
--------------------------------------------------------------------------------
        Ir I1mr I2mr      Dr D1mr D2mr         Dw     D1mw     D2mw  file:function
--------------------------------------------------------------------------------
17,707,009    2    2       2    1    1 1,000,001  62,500   62,500  major.c:rowMajor
17,008,079    1    1       2    1    1 1,000,001 840,460  622,690  major.c:columnMajor
```

According to this function profile, there is no doubt that C is a row-major language. Its row-ordered function produces a mere 62,500 level 1 data cache write misses compared with its column-ordered function's 840,460.

How does the Fortran 90 equivalent of this program fare?

```
> pgf90 -gopt -o f90_major major.f90
> valgrind --tool=cachegrind ./f90_major
> cg_annotate cachegrind.out.11321
...
--------------------------------------------------------------------------------
        Ir I1mr I2mr      Dr  D1mr D2mr         Dw    D1mw    D2mw  file:function
--------------------------------------------------------------------------------
15,606,007    2    2       2     1    1 1,000,001 90,528 62,999  major.f90:rowmajor_
15,006,067    2    2       2     1    1 1,000,001 62,500 62,500  major.f90:columnmajor_
...
```

Notice how Fortran 90's column-ordered D1 write misses (62,500) are the same as C's row-ordered D1 write misses. At first glance, C's column-ordered cache misses (840,460) appear to be significantly more expensive than Fortran 90's row-ordered cache misses (90,528). This, however, is due to the example's array dimensions being biased toward column major languages.

Cachegrind can easily reveal instruction and data bottlenecks in your code's performance, as seen in these examples. I find myself wanting to run Cachegrind on every code I have ever written to put my programming efficiency to the test.

For more information, refer to Valgrind's Cachegrind manual at the following URL:

http://valgrind.org/docs/manual/cg-manual.html

Computational Fluid Dynamics Simulation of Three-Dimensional Airflow over a Vegetated Soil Surface

Phu V. Luong, Robert S. Bernard, and Stacy E. Howington, U.S. Army Engineer Research and Development Center Coastal and Hydraulics Laboratory

The PAR3D code [1] represents a continuation of work that began with the MAC3D code [2], which was developed for simulating flow and transport in deep water. PAR3D retains the original capabilities of MAC3D, but it also incorporates parallel processing, which reduces CPU time and increases allowable resolution (problem size).

Since water and air can both be regarded as incompressible fluids at flow velocities much less than the speed of sound, PAR3D has been extended to facilitate simulations of airflow and heat transfer over irregular terrain. PAR3D is limited to incompressible flow in particular, which means that the fluid density may be altered slightly by temperature but not by pressure. Variations in density create buoyant forces for which PAR3D employs the Boussinesq approximation. Thus, density variations are neglected in the continuity equation but not in the momentum equation.

The code uses a finite-volume scheme with curvilinear marker-and-cell (MAC) grids to compute the transport of mass and momentum through flow regions of arbitrary shape. The MAC designation indicates that vector components are computed normal to the cell faces, while scalar quantities are computed exclusively at the cell centers. The grids are subdivided into multiple components called grid blocks, and each block is assigned to a single processor. Shared information is communicated between processors via the standard message passing interface (MPI) library discussed by Gropp et al. [3].

PAR3D uses a time-marching scheme to solve the Reynolds-averaged Navier-Stokes (RANS) equations. A Poisson equation for pressure, obtained by combining the momentum and continuity equations, is solved iteratively during each time-step to supply the pressure gradient needed for conservation of mass. The influence of turbulence is incorporated via the standard turbulence model developed by Launder and Spalding [4].

In recent applications [1, 5], PAR3D has been used for the simulation of airflow and aerosol dispersion in complex man-made enclosures. The current application of PAR3D involves the simulation of airflow over a hypothetical, irregular, vegetated ground surface. The domain of interest is 10 m wide, 10 m long, and 2 m deep, discretized on a computational grid with a nominal spacing of 4 cm between grid nodes. The grid itself is divided into 50 blocks, with each block residing on a single processor. The bottom boundary for the grid was generated from topographic data for representative semi-arid terrain.

Stones and other objects were included as impermeable obstacles, and plants were incorporated as semi-permeable regions of the grid. Although the scene is synthetic, the surface shape, surface roughness, and vegetation geometries used were derived from field data collections. Realistic inflow boundary conditions are still under development, but the provisional inflow conditions used for this study include a logarithmic velocity profile with uniform turbulence energy. Constant (but distinct) values of temperature were specified along the inflow and bottom boundaries, respectively. Values for all computed variables were extrapolated from upstream values along the outflow boundary. Using a time-step of 0.05 seconds, steady state was achieved after 10 minutes of simulated time.

Spatially variable wind speeds and air temperatures produced by these simulations will drive surface heat exchange functions within soil and vegetation models. The resulting surface temperatures will feed models that approximate atmospheric and sensor effects on infrared signals. The end product is high-resolution, synthetic, infrared imagery of realistic soil surfaces. Such images permit the exploration of meteorological, hydrological, and geological influences on the performance of infrared sensors.

References

[1] R. S. Bernard, P. V. Luong, and M. J. Sanchez, "PAR3D: Numerical Model for Incompressible Flow with Application to Aerosol Dispersion in Complex Enclosures," Technical Report ERDC/CHL TR-07-9, US Army Engineer Waterways Experiment Station, Vicksburg, MS (2007).

[2] R. S. Bernard, "MAC3D: numerical model for reservoir hydrodynamics with application to bubble diffusers," Technical Report CHL-98-23, US Army Engineer Waterways Experiment Station, Vicksburg, MS (1998).

[3] W. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message-Passing Interface, MIT Press, Cambridge, MA, (1996).

[4] B. E. Launder and D. B. Spalding, "The numerical calculation of turbulent flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 3, pp 269-289, 1974.

[5] P. V. Luong, M. J. Sanchez, and R. S. Bernard, "CFD Simulation of Three-Dimensional Airflow and Contaminant Transport in a Complex Building," *Proceedings of the Ninth International Symposium on Fluid Control, Measurement and Visualization*, Tallahassee, FL (2007).

ezHPC Version 3.0

By Keith Rappold, ERDC Information Technology Laboratory (ITL); Scotty Swillie, ERDC DSRC; and Wes Monceaux, ITL

The Program-wide move to CAC for Kerberos authentication brought major changes within the User Interface Toolkit's (UIT) architecture (see UIT article for details). No longer would users be able to depend on their Kerberos SecurID card for access to ezHPC. A ripple effect of these changes also meant that the critical authentication mechanism of the ezHPC client was affected. The ezHPC development team was forced to find an alternate way to authenticate and present the API methods to DoD HPCMP users.

The UIT development team at ERDC took a two-pronged approach to the CAC solution – 1) a local Web service available on each user's machine that interacts with a central API; 2) a centralized Web service that can support multitiered Web applications. After much review, it was determined that the local Web service model would bring some much desired performance enhancements. Because of these added benefits, the ezHPC team chose to go the route of the local Web service for Version 3.0.

Version 3.0 will require a one-time download from the ezHPC site. This will come as a prepackaged, small Web server that will run accessible only from the local system. Authentication will take place via `pkinit` or `kinit` using the DSRC Kerberos client kit, just as it is done for any other application that requires Kerberos. From that point forward, ezHPC API calls to the UIT will just verify that the user has a ticket and will then simply use that ticket to perform actions on the user's behalf on the HPC machines.

Like the previous version of ezHPC, Version 3.0 will require users to have Java Runtime version 1.5+. They will also need to have the DSRC Kerberos toolkit installed for their platform. Also, just like the previous version of ezHPC, users will continue to enjoy the visual, Web-feel of the interface. They will be able to take advantage of all of the time-saving benefits offered in ezHPC – job status, cue status, script management, script generator, one-click file transfers (larger files will be faster), upload/download from local machines – to name a few.

Here is how the new process will work:

1. Users will go to the ezHPC site - https://ezhpc.hpc.mil – and download the local Web service to a directory on their hard drive. This is a one-time-only process.
2. Users will use their SecurID card or their CAC to authenticate using the DSRC Kerberos `kinit`, `pkinit`, or similar command. (Once authenticated, the ezHPC Java application running on the user's desktop will then issue commands (using `ssh`) to interact with the HPC systems.)
3. Users will launch ezHPC v.3.0 from an icon located on their desktop.
4. ezHPC will interact with the local Web service. The local UIT Web service will access the central UIT Web server to obtain all necessary pIE user information.
5. The HPC systems will then treat the `ssh` connections from the interface no differently from any other connections made by the user.

Aside from the simplified CAC authentication, there are other performance benefits that come with the use of ezHPC Version 3.0.

- Direct connections from the user's machine to the HPC machines will allow for better performance when manipulating large files.
- Since the API is now accessed via localhost, method calls should respond much faster.
- Direct file transfer and communication between client and HPC Systems occurs.
- User's Kerberos ticket life is no longer limited to 1 hour (depending on network).
- Users will continue to be able to access the HPC systems via an easy-to-use Web-site-type – point and click – drag and drop – interface.
- A more reliable, robust interface is provided.

While ezHPC Version 3.0 will require some fundamental changes in how users interact with the interface, we believe the improvements gained in performance and reliability far outweigh any minor adjustments.

CCAC's Future is Bright

By Tracey Smith, CCAC

In the AFRL DSRC Fall 2008 Journal, the Consolidated Customer Assistance Center (CCAC) introduced an initiative to provide exceptional, cost-effective user service. The first portion of this initiative brought about recommendations and requirements for improvement, which led the team to investigate Service Management applications. Many applications were reviewed, resulting in the acquisition and implementation of LiveTime, a commercial off-the-shelf Service Management tool. Currently, the application is being used for ticketing only, with plans to introduce a customer portal at the Users Group Conference (UGC) in June. The application provides improved capabilities for both CCAC technicians and users that are far beyond what were available previously.

The primary improvement is that technicians at all of the sites are now using the same application and a shared database to work on unclassified problems. Additionally, an enhanced knowledge management system will provide staff and users with instant access to the collective wisdom of prior technician's resolutions. This provides a starting point for solving new issues. The increase in efficiency allows a greater level of collaboration between technicians at different sites and the user community, resulting in shorter resolution time.

The knowledge base rolled out with LiveTime in March, but it will not be unveiled to the user community until the UGC in mid-June. The delay in announcing the capability to users allows CCAC extra time to build up articles to populate the knowledge base, which will provide a significantly enhanced experience when users first encounter the Portal.

The Portal allows users to enter their problems directly into the system for immediate ticket creation, to check on their ticket status, and to gain access to the combined knowledge base. It also provides an easy method to supply technicians with attachments like log files, screen shots, or documents.

In addition to the unified ticketing system, improved intrasite communications have been established with a secure chat engine (Jabber), which is maintained by the HPCMP. Use of Jabber allows technicians to collaborate on issues, allowing them to solve user concerns more quickly.

Further communication improvements will be addressed by the deployment of an advanced phone system for CCAC. This phone system is still in development, but will allow the six sites to function as one virtual office. The essential requirements for the phone system are the capability to transfer calls to any location in order to provide disaster recovery and 24X7 operations capabilities. It will also allow technicians to more efficiently consult on a particular problem.

Once the ticketing system and the phone system are in place, the plan is to implement the Information Technology Infrastructure Library (ITIL) best practices into CCAC's procedures. ITIL will ensure that the CCAC technicians at all levels are following industry-recognized procedures in dealing with user problems.

This initiative has already had an impact on CCAC by improving the user experience when dealing with the help desk. Once the ticketing system, phone system, and ITIL best practices are in place, CCAC is confident that it is on track to provide exceptional, cost-effective user services.

SC09

November 14-29, 2009, Portland Convention Center, Portland, Oregon
http://sc09.supercomputing.org

Recently, two former Allocated Distributed Centers – the Maui High Performance Center and the Arctic Region Supercomputing Center – were incorporated into the fold of what has been historically known as the Major Shared Resource Centers. Along with this comes the renaming of all six Centers to DoD Supercomputing Resource Centers (DSRCs). Although each Center has its own unique attributes, the melding of the six Centers into a single, cohesive, and effective operational unit is yet another step toward fostering an improved set of services for the DoD HPC User Community.