Center for High-Performance Computing @ the University of Utah

News

University of Utah Fall, 2004

Article

Efficient Usage of the Arches Metacluster

by Martin Cuma

Scientific Applications Programmer, Center for High Performance Computing, University of Utah

At the beginning of the year, CHPC aquired and installed a new 900+ CPU supercomputer system called Arches, which is becoming the main workhorse for the researchers using our systems. The Arches system is a metacluster, consiting of several cluster systems that are coupled together at the scheduling level. The justification for splitting the whole machine into sub-clusters is to target specific applications requirements with specific hardware thus improving the cost-effectiveness of the whole solution. All currently implemented parts of the cluster contain 1.4 GHz AMD Opteron CPUs.

There are currently four Arches clusters that are functional. The largest, *delicatearch*, consists of 512 CPUs in 256 compute nodes each with 2 GB of RAM and Myrinet network interconnect. This cluster is intended for large parallel applications with high communication demands.

The second cluster, *marchingmen*, contains 328 CPUs in 164 nodes, 2 GB RAM and Gigabit Ethernet interconnect. Due to the slower interconnect, its target applications are serial or embarrasingly parallel programs that don't require lots of communication.

The third cluster, *tunnelarch*, has 96 CPUs in 48 nodes and twice the memory of *marchingmen* (4 GB). This cluster is intended to be used for large memory demanding jobs and database searches (e.g. BLAST).

Finally, the last cluster, *landscapearch*, is a condominium style cluster which is built from nodes purchased by researchers, similar to the way our older cluster, *icebox*, was built. The nodes may or may not have Myrinet, but all of them have Gigabit Ethernet. Also, the memory configuration and CPU speed will vary in the future based on the purchaser requirements.

There will be a fifth component coming online soon: a visualization cluster which will enable researchers to visualize their results on a large screen.

Each of the forementioned clusters has two login nodes that serve as a gateway to the system.

All Arches nodes mount users home directories and two types of scratch file systems for temporary data storage:

/scratch/serial is a single NFS mounted file server that is optimized for fast throughput from a single CPU application. Its intended use is I/O from a single processor run or a parallel run that uses only a single process for I/O.

/scratch/parallel, which will be deployed soon, is a parallel file system based on PVFS2 [1] that allows for fast throughput from multiple processes. Although it's ideally suited for parallel runs doing I/O in parallel, serial I/O performance will benefit from the distributed file management built into PVFS2.

The Arches metacluster is currently running the SuSE Linux 9.0 x86-64 Linux distribution on a custom 2.6 kernel with standard packages installed (e.g. GNU compilers, text based tools, etc.). Although the x86-64 architecture supports execution of programs compiled on 32 bit Linux (e.g. Icebox), it is recommended that users rebuild their applications for this architecture in order to achieve best performace. CHPC would be happy to assist with porting users' applications from other systems to Arches. We have gained a considerable amount of expertise in doing this since the deployment of Arches.

Login and batch use

At present, users have to specifically log into each cluster in order to use it. However, in the future we intend to have a single interface for access and job management via a grid-based scheduler.

All Arches clusters can be accessed using a secure shell (SSH) client. The full path to the cluster login node must be specified, e.g. *ssh delicatearch.chpc.utah.edu*. Since all the clusters mount the same user home directories, CHPC sets a UUFSCELL environment variable that defines which cluster one is on. For example, on *delicatearch* this variable is set to *delicatearch.arches*. This enables one to create specific initializations for each cluster, e.g set different paths. On our Arches User's Guide website [2], we are doccumenting sample login scripts that set paths and variables for the main program packages installed on the clusters. More info on the functionality of some of these packages will be provided later in this article.

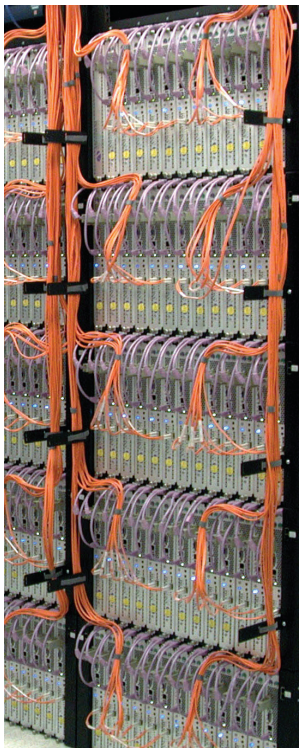
Logging into a cluster means logging into the login node which serves as a gateway for program development, compilation, data analysis, etc. Login nodes are not

designed to run calculations. For that, one has to request time on the compute nodes.

Users can't login to the compute nodes directly; they have to request them using a batch queue command. A queue manager, together with a scheduler, determines when this request can be fulfilled.

As in any major computing center, all CHPC systems are queue based in order to maximize job throughput. The Arches clusters run the Torque [3] resource manager for queue management and the Moab [4] scheduler for job scheduling. Torque is a spinoff from PBS and Moab is a spinoff from Maui. Both should be familiar to users of Icebox and both have the same user interface. Basic commands are *qsub* to submit a job, *qstat* or *showq* to see a queue and *qdel* to delete a job.

In order to run a job, one has to create a Torque script (see the Arches User's Guide webpage [2] for details) and submit it to the queue using the *qsub* command. The scheduler then determines the job's priority, queues it up, and when the time comes, runs it. The priority of the job is built from a fairly complex set of rules that sometimes frustrates users who feel that their job is waiting in the queue too long. The main contributors to the job priority are available time credit on the system (if there's no credit, the job falls to the freecycle level which will run only when there is no other job waiting in the queue), fairshare (the less the user runs, the higher priority), parallelism (the more nodes requested, the higher priority) and time spent in the queue (the more, the higher priority). *landscapearch* prioritizing is further complicated by a set of preferences for users who own the nodes.



It is sometimes desirable to log directly into a node to run a short parallel program, e.g. to debug. This is accomplished by requesting an interactive Torque job, which is done by appending flag *-I* to the *qsub* command. For example, to ask for 2 CPUs for one hour, issue command *qsub -I -l nodes=2,walltime=1:00:00*. If there are available nodes, the system comes up shortly with a prompt on the nodes that one asked for and one can use them in the same fashion as an interactive node (e.g. to compile) plus run the program from the command line.

Finally, since Arches was designed to be a "cluster of clusters", all the interactive nodes have access to batch commands on all the Arches clusters. Thus, users can log into one cluster's interactive node and direct batch commands to other clusters. This can be achieved by specifying the full path to the batch commands. For example, while logged into *delicatearch*, one can issue command */uufs/marchingmen.arches/sys/bin/qsub* to submit a job on *marchingmen*. This approach is especially useful when the interactive nodes of certain cluster are not responding.

Compilers, debuggers, profilers

A fast machine is not of much use without good compilers and tools. Compilers create executable programs out of users code and tools help in finding errors (debuggers) or performance bottlenecks (profilers). We have evaluated most of the available compilers and tools for the x86-64 platform and installed those that we believe perform the best.

Arches offers three compiler suites. There's the open source GNU compiler suite which is quite good for C/C++ (*gcc*, *g++*) and Fortran77 (*g77*) but does not include Fortran 90/95 compiler and does not have many code optimization options. Text based tools assist in debugging (*gdb*) and profiling (*gprof*).

We also have a license for the Portland Group (PGI) compilers. These support the C/C++ (*pgcc*, *pgCC*) and Fortran 77/90/95 (*pgf77*, *pgf90*) languages and are quite good performers. Their main advantage is compatibility with many software packages, since the compilers have been around for a long time and are quite mature. PGI also ships several tools that assist in debugging (*pgdbg*) and profiling (*pgprof*).

Finally, we have licensed the Pathscale compilers which are relatively new but show lots of promise. Like PGI, they support C/C++ (*pathcc*, *pathCC*) and Fortran 77/90/95 (*pathf90*) and offer a wide range of code optimization features as well as generally superior performance compared to both GNU and PGI. Since Pathscale compilers are relatively new, they still don't support the OpenMP shared memory parallelization standard and they don't ship with their own tools (GNU's *gdb* and *gprof* can be used as substitutes), but the company plans to release some in the near future. For details on how to use these compilers on the Arches systems, consult our online documentation [5].

We also offer several commercial debugging and profiling tools. There is the Totalview debugger which is highly effective at finding errors in both serial and parallel applications. Its usage is the same as on our older platforms, Icebox and Sierra, so the transition to Arches should be very easy. For details on its usage see our online documentation [6] or earlier newsletter article [7].

In addition to Totalview, we have licensed the Intel Trace Collector/Analyzer (ITC/A, f.k.a. Vampir), an MPI profiler with which users should also be familiar from Icebox. It helps to find performance bottlenecks due to bad MPI communication design. One slight problem with ITC/A is that it does not build with 64 bit applications, so in order to use it one has to build his/her application as 32 bit. This is a fairly complicated process and is detailed on our webpage [8].

Since all Arches clusters currently consist of a single architecture, executables compiled on one cluster will run on all the others. The only exception is MPI executables compiled for Myrinet (with MPICH-GM) on *delicatearch* which will not run on clusters with nodes that don't have the Myrinet interconnect (*tunnelarch*, *marchingmen*).

Parallel programming

As we mentioned at the start of the article, the Arches clusters consist of dual processor nodes connected with two different network interconnects. Having dual CPU nodes gives us two layers of parallelism: one on the node level (two CPUs) and one on the cluster level (sets of dual CPU nodes). Since the CPUs inside the node share the memory, one can use shared memory programming schemes, the representative of which is the OpenMP standard which is supported by the PGI compilers. Naturally, because of the two CPU per node limit, only two processor OpenMP runs can be done on Arches which somewhat limits the OpenMP usability.

In order to run on more than two processors, one has to communicate the data over the network between the nodes. Message passing interface (MPI) is a de-facto standard for communication in distributed memory systems. There are several MPI distributions that fit well into the Arches structure, plus the presence of two different interconnects warrants use of different MPI distributions for each of them.

On the two clusters that use just Gigabit Ethernet, *marchingmen* and *tunnelarch*, we offer two MPI distributions, MPICH and MPICH2. The former is the original MPI distribution from Argonne NL which supports MPI standard 1.1 with several extensions. It is now being phased out in favor of MPICH2 which is fully complying with the MPI standard 2.0. We still recommend the use of MPICH but encourage users to experiment with MPICH2 which is in its late beta stage and shows significant performance gains with some applications.

On *delicatearch* and *landscapearch*, the nodes of which contain both Ethernet and Myrinet, we offer the Myrinet version of MPICH - MPICH-GM - in addition to MPICH and MPICH2. Since only MPICH-GM runs over the Myrinet interconnect, users should maximize their efforts to use MPICH-GM with their code on these clusters. Runs over the Ethernet using either MPICH or MPICH2 will not perform as well and should be limited to testing. To do otherwise would decrease the utilization of the clusters and may result in correctional measures against the user. For more details on how to compile with the MPI libraries see our online documentation [9].

Finally, one can write his/her code using mixed OpenMP/MPI parallelism. Although this approach has gained some attention over the past years, we have seen only limited performance gains from the few tests we have done, especially when running with MPICH2 or MPICH-GM which have optimized modules for shared memory communication. Thus at this point we don't see any major benefits in using this approach on Arches.

Conclusions

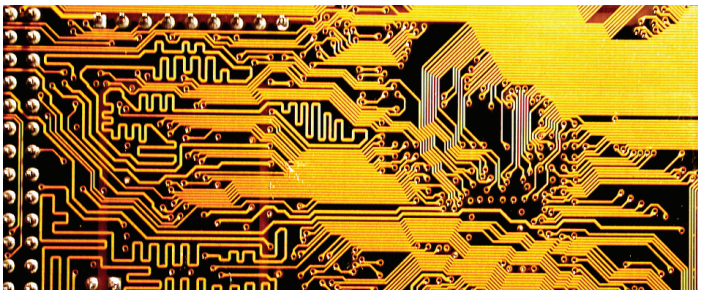
We hope that in this article we gave a solid basic introduction to our new computing resource, the Arches metacluster. For more details and updates we recommend to consult the web-based Arches User's Guide [2]. We will be happy to assist with any questions or problems the users may have while utilizing the Arches clusters.

References

[1] PVFS2 (Parallel Virtual File System 2),
http://www.pvfs.org/pvfs2/
[2] Arches User's Guide,
http://www.chpc.utah.edu/index.php?currentNumber=4.4.6
[3] Torque resource manager,
http://www.supercluster.org/projects/torque/
[4] Moab workload manager,
http://www.clusterresources.com/products/mwm/
[5] CHPC's C and Fortran compilers help webpage,
http://www.chpc.utah.edu/index.php?currentNumber=3.2.50, http://www.chpc.utah.edu/index.php?currentNumber=3.2.110
[6] CHPC's Totalview help webpage,
http://www.chpc.utah.edu/index.php?currentNumber=3.2.360
[7] Tips and tricks for programming on Icebox and Sierra part 3. Debugging, CHPC Newsletter Summer 2003
[8] CHPC's Intel Trace Collector/Analyzer help webpage,
http://www.chpc.utah.edu/index.php?currentNumber=3.2.380
[9] CHPC's MPI help webpage,
http://www.chpc.utah.edu/index.php?currentNumber=3.2.200

FYI

CHPC will be highlighting interesting CHPC-supported projects at the **IEEE/ACM SuperComputing 2004** conference. Of particular interest are projects with a visual emphasis. We will present the projects in posters as well as in video/simulation/visualization form. If you have a project that fits our criteria and would like it to be highlighted at SC04, please contact Sam Liston at *stliston@chpc.utah.edu*.

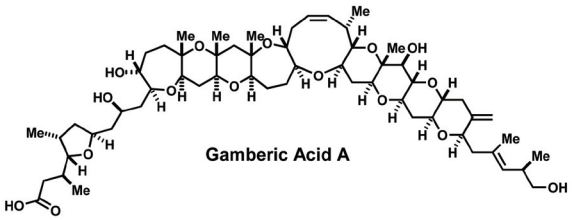


Article

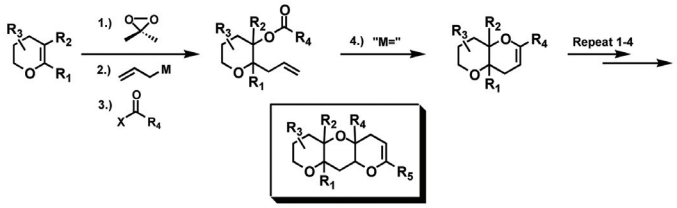
Use of Gaussian03 to Study Transition States and Reaction Chemistry

by Anita M. Orendt
Staff Scientist, Molecular Sciences, Center for High Performance Computing, University of Utah
with Scott W. Roberts and Jon D. Rainier
Department of Chemistry, University of Utah

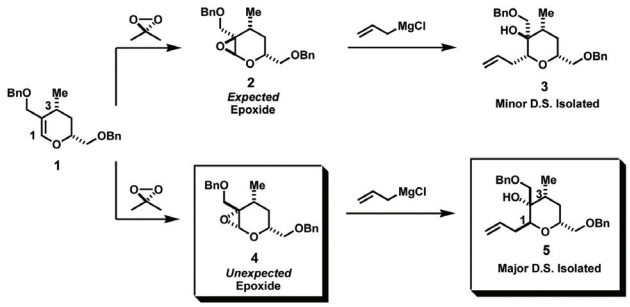
C-glycosides have remained an important area in the synthetic community, due to their interesting biological activity and prevalence in many natural products. One such example of this type of natural product is Gamberic Acid.



The Rainier group has been working on an iterative strategy towards synthesizing the fused polycyclic ether ring systems found in these systems. Central to this synthetic strategy is the development of a one flask enol ether oxidation and nucleophilic addition sequence to build this natural product, steps 1-2 of the following synthetic pathway.



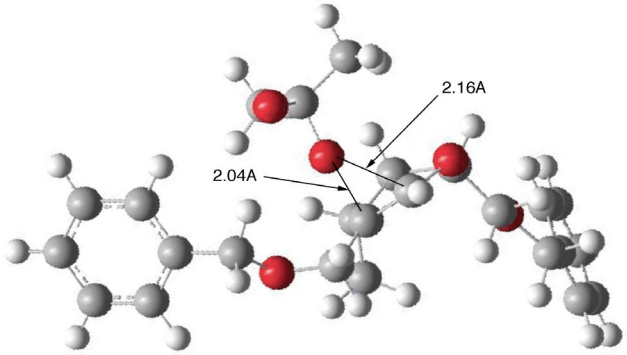
During the course of looking at this type of reaction some unexpected stereochemical results were obtained, with the starting material 1 giving the major product 5 over the expected diastereomer, molecule 3, in a 3:1 ratio.



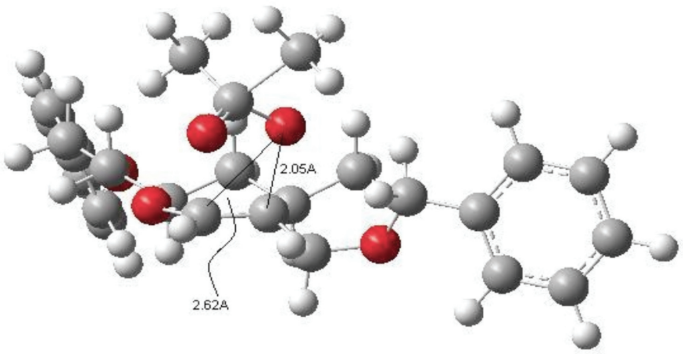
This result can be explained by two possible reaction pathways for the addition of the oxygen atom to the double bond resulting in the formation of the three membered epoxide ring. If the addition proceeds on the face of the double bond opposite of the methyl group (Me, position 3 of the six membered ring) molecule 2 is formed, and therefore the expected product 3, would be obtained. If the addition proceeds on the face of the double bond on the same side of the six membered ring as the Me group on carbon 3, then the epoxide compound 4 and subsequently product 5 would result. This was the major product obtained in the reaction. It was postulated that the first of these reactions would proceed by a concerted, symmetric reaction path. The steric factors in the second reaction, however, would lead to an asynchronous mechanism, where the oxygen atom attached first to one of the carbons of the double bond and then to the other.

There have been numerous theoretical studies on the pathway and transition state for this type of alkene epoxidation reaction in the literature, using a number of epoxidation agents including the dimethyldioxirane used in this case. However all were performed on idealized small olefin systems, usually ethylene, and therefore did not address the different stereochemical possibilities, as there is not any difference between the two faces. The majority of calculations have resulted in a symmetrical transition state.

To support the postulated transition states, calculations were completed on the first step of this reaction, i.e., going from molecule 1 to either molecule 2 or 4. Molecular geometry optimizations were first performed on molecules 1, 2 and 4, using the B3LYP DFT hybrid functional and D95(d,p) basis sets. The relative energy of these three molecules support the fact that molecule 4 was the major product as it was calculated to be 6.7 kcal/mole lower in energy than 2. Subsequently a search was performed for the transition state for each of these products. This search was performed at the same level of theory as the geometry optimizations. The results confirmed the postulation of the two reactions having different types of transition states. The calculations on the reaction that leads to the epoxide 2 resulted in a relatively symmetric transition state, shown below, with the distances between the oxygen atom being added and the two olefinic carbons being 2.04 and 2.16Å. This small difference can be expected due to the difference in substitution of the two olefinic carbons.



The reaction in which the oxygen attack is on the same side of the six memebered ring as the methyl group, which leads to molecule 4, shows a distinctly asymmetric transition state, with distances of 2.05 and 2.62Å. The oxygen atom is being added to the olefinic carbon with the side chain.



In both cases, the fact that the found transitions states corresponded to the reaction of interest was confirmed by doing a frequency calculation. Each molecule showed only one imaginary frequency, and the normal mode of these vibrations clearly shows the oxygen atom moving from the dimethyldioxirane epoxidation agent towards the olefin in molecule 1.

The energy difference between these two transition states is 2.5 kcal/mole with the first, more symmetric transition state being lower in energy. Unfortunately, this energy difference is not consistent with the observed product distribution. Further studies are being completed to understand this difference.

The last step was to follow the reaction path using the IRC keyword in Gaussian03. Starting with the transition state and the known normal mode of vibration of the reaction, the reaction path is followed both directions. This produces both an energy profile of the reaction as well as a set of structures along the reaction coordinate.

All calculations were performed on a single four way node of the sierra cluster. The calculations on the transition state had a total of 640 basis functions. Individual jobs were able to be completed in under the 72 hour job time limit of that system, including both the frequency and the IRC (calculating 24 points along the reaction coordinate) calculations. All jobs were given 3900 mb memory, which is the maximum that Gaussian03 will allow to be allocated on sierra. The optimization to a transition state was a trial and error process during which the number of imaginary frequencies often changed from one. This required new jobs be started repeated (10-15 times were typically in this study) until an optimized first order transition state was located.

FYI

For the latest news, system status, and downtimes, see the CHPC home page: *http://www.chpc.utah.edu/*

Article

CHPC's New Dedicated Cluster Computer Room in Research Park

by Guy Adams

Assistant Director, Systems, Center for High Performance Computing, University of Utah

CHPC's newest dedicated "cluster" computer room is located next to the University of Utah's ITS/Netcom Data Center in Research Park at 585 Komas Drive. The building and the computer rooms contained within are leased through Research Park and Associates.

With the additional computer room, CHPC now has three computer room facilities connected via dedicated network fibers (as opposed to the more conventional approach of one large room, which is common with Supercomputer Centers or National Labs).

Within the new computer room are two smaller rooms, each with a raised floor and walls that are insulated with sound reducing material. These smaller rooms are dedicated for the staging of equipment and a few workstations.

The availability of inexpensive "Commodity Off-the-Shelf" (COTS) clusters has significantly enhanced our ability to purchase hardware for scientific purposes. In our case, the cluster consists of COTS computers racked in a dense "Blade Configuration." Today, we have 65 nodes (130 processors) occupying one standard rack footprint. We have well over 500 (1000 processors) of these blade nodes and racks of COTS disk arrays, and management devices. (See Martin Cuma's article, "Efficient Usage of the Arches Metacluster," earlier in this newsletter for more details about the Arches metacluster).

The increase in the number of computers brings with it many obstacles. For instance, with our latest purchase (the "Arches" metacluster) we bumped into power and cooling issues. The planned figure of 75 Watts per square foot ended up being increased to 300 Watts or more per square foot for the clustered environment.

When Julio Facelli approached Randy Green of Research Park and Associates with the news, Randy was shocked to discover we would use all the power that was

available. After much discussion, it was decided that the room would get one panel with 2000 AMP circuits along with a few extra circuits that can be utilized for other purposes. To put this in perspective, that's more than double the power for the entire rest of the building.

Cooling becomes a limiting factor as to the size of clusters we can optimally run. At today's density and power and cooling consumption, CHPC could house well over 4000 processors, or four equivalent in size to the "Arches" metacluster. The trend in the industry over the next decade is to lower the power and cooling needs of processors. This will allow centers to run even larger clusters on their power and cooling budgets.

Because this cluster computer room is designed for "lights out management," equipment has been installed that will facilitate management from remote locations. For serial console access, every machine can be accessed from a network workstation through "Cyclades Terminal Servers." To power the various clusters up and down, we have included either "vendor included" power management or "Cyclades Power Management" servers. Except for a physical failure of the hardware, all computers can be administered from off-site, including powering them off and on.

For software management of these clusters we have implemented a centralized "Root Boot Server" methodology. This uses PXE ("Pre-eXecution Environment) boot capability of the new motherboards and then uses "NFS ROOT BOOT" to load the OS on each computer. The local disks on each node do not contain the operating system and thus can be used for temporary files. This methodology is set up in a modular fashion so that as our clusters grow or change we can balance the number of nodes per server and add more if needed.

Already there have been huge dividends in undertaking this change of methodology. There have recently been many power outages, which in a thousand processor complex would certainly damage dozens of local disk images containing the operating system (which would then need to be repaired). Now, once we repair the servers, we no longer need to repair the operating system on each node; they simply boot as they power up.

CHPC is moving toward a grid approach with these clusters. As more and more clusters are purchased they can be accessed from single login points. This kind of meta-clustering is an emerging technology in which CHPC is playing a leading role along with Ohio State University and Cluster Resources. CHPC is leading with others in expanding our batch scheduling software with "Moab," "TORQUE" "Globus," and "Silver." The intent is to keep the complexity of these systems abstracted for the scientific users.

This facility is open to other University departments, but only by contacting CHPC and only for clusters. Because this facility is leased, a large portion of the operational costs is paid for out of departmental budgets.

Upcoming Presentations

CHPC has developed a series of courses to help users make the most of CHPC resources. We continuously add to and improve this series, which is presented every fall and winter. We welcome suggestions for new presentations.

The following upcoming presentations will be held in the INSCC Audtitorium at 1:30pm on the scheduled date:

October 7th: Profiling with Vampir/Guideview
October 14th: Chemistry Packages at CHPC
October 21st: Using Gaussian 98/03
October 28th: Mathematical Libraries at CHPC
November 4th: Fast Parallel I/O at CHPC
November 11th: Hybrid MPI-OpenMP Programming

Recent Publications

Bazterra, V. E., O. Ona, et al. (2004). "Modified genetic algorithms to model cluster structures in medium-size silicon clusters." Physical Review A 69(5): 053202.

Berdys, J., I. Anusiewica, et al. (2004). "Damage to Model DNA Fragments from Very Low-Energy (<1 eV) Electrons." Journal American Chemical Society 126(20): 6441-6447.

Hart, K. A., W. J. Steenburgh, et al. (2004). "An Evaluation of Mesoscale-Model-Based Model Output Statistics (MOS) during the 2002 Olumpic and Paralympic Winter Games." Weather and Forecasting 19(2): 200-218.

Izvekov, S., M. Parrinello, et al. (2004). "Effeective force fields for condensed phase systems from ab initio molecular dynamics simulation: A new method for force-matching." Journal of Chemical Physics 120(23): 10896-10913.

Ka, B. J. and G. A. Voth (2004). "Combining the Semi-classical Initial Value Representation with Centroid Dynamics." J. Phys. Chem B 108(21): 6883-6892.

Orendt, A., B. Haymore, et al. (2004). Design, Implementation, and Deployment of a Commodity Cluster for Periodic Comparison of Gene Sequences. High Performance Computing: Paradigm and Infrastructure. L. T. Yang and M. Guo, John Wiley & Sons: (in press).

CHPC Staff Directory

Administrative Staff	Title	Phone	Email	Location
Julio Facelli	Director	556-2426	Julio.Facelli@utah.edu	410 INSCC
Julia Harrison	Associate Director	652-0019	julia@chpc.utah.edu	430 INSCC
Guy Adams	Assistant Director, Systems	554-0125	gadams@chpc.utah.edu	424 INSCC
Joe Breen	Assistant Director, Networking	550-9172	jbreen@chpc.utah.edu	426 INSCC
DeeAnn Raynor	Administrative Officer	581-5253	dee@chpc.utah.edu	412 INSCC
Victoria Volcik	Administrative Assistant	585-3791	vicky@chpc.utah.edu	405-2 INSCC
Teresa Hennigan	Administrative Assistant	581-6440	teresa@chpc.utah.edu	405-3 INSCC

Scientific Staff	Expertise	Phone	Email	Location
James Agutter	Information Visualization	581-8779	agutterja@arch.utah.edu	235 AAC
Thomas Cheatham III	Biomolecular Modeling	587-9652	cheatham@chpc.utah.edu	306 INSCC
Martin Cuma	Scientific Applications	587-7770	mcuma@chpc.utah.edu	418 INSCC
Byron L. Davis	Statistics	585-5604	byron@chpc.utah.edu	416 INSCC
Julio Facelli	Molecular Sciences	556-2426	Julio.Facelli@utah.edu	410 INSCC
Stefano Foresti	Information Visualization	581-3173	stefano@chpc.utah.edu	322 INSCC
Robert McDermott	Visualization	581-4370	mcdermott@chpc.utah.edu	420 INSCC
Anita Orendt	Molecular Sciences	231-2762	orendt@chpc.utah.edu	422 INSCC
Alun Thomas	Bioinformatics	587-9309	alun@gene.pi.med.utah.edu	Research Park

Systems/Network Staff	Title	Phone	Email	Location
Irvin Allen	System Administrator	231-3194	iallen@chpc.utah.edu	405-40 INSCC
Wayne Bradford	Cluster Grid Administrator	243-8655	wayne.bradford@chpc.utah.edu	405-41 INSCC
Erik Brown	Computation Cluster Admin.	824-4996	erik@chpc.utah.edu	405-29 INSCC
Joe Clyde	Network Operations Engineer	558-7661	joe.clyde@chpc.utah.edu	405-38 INSCC
Geoff Fritz	Enterprise Server Administrator	259-3495	gfritz@chpc.utah.edu	405-31 INSCC
Brian Haymore	Lead, Comp. Cluster Admin.	558-1150	brian@chpc.utah.edu	428 INSCC
Samuel T. Liston	Digital Communication & Visualization	232-6932	stliston@chpc.utah.edu	405-30 INSCC
Jimmy Miklavcic	Multimedia, Telematic & Digital Communication	585-9335	jhm@chpc.utah.edu	296 INSCC
Ron Price	Software Engineer & Grid Architect	560-2305	rprice@eng.utah.edu	405-39 INSCC
David Richardson	Computer Technician	550-3788	drr@chpc.utah.edu	405-23 INSCC
Steve Smith	System Administration	581-7552	steve@chpc.utah.edu	405-25 INSCC
Eli Stair	System Manager	558-3099	eli@chpc.utah.edu	405-17 INSCC
Matthew Thorley	Network Assistant	560-3438	ruach@chpc.utah.edu	405-20 INSCC
Alan Wisniewski	Network Engineer	580-5835	quantix@chpc.utah.edu	405-21 INSCC

User Services Staff	Title	Phone	Email	Location
Iris Boanta	Technical Assistant	N/A	iris@chpc.utah.edu	405-10 INSCC
Jason Duhaine	Systems Assistant	N/A	jason@chpc.utah.edu	405-28 INSCC
Eric Hansen	Technical Assistant	N/A	ehansen@chpc.utah.edu	405-27 INSCC
Shawn Lyons	Network Assistant	N/A	slyons@chpc.utah.edu	405-22 INSCC
Paul Hartzog	Technical Assistant	N/A	phartzog@chpc.utah.edu	405-11 INSCC
Beth Miklavcic	Multimedia Design, Digital Video	585-1067	bam@chpc.utah.edu	115 INSCC
Erik Ratcliffe	Graphic & Web Design	N/A	erat@chpc.utah.edu	405-13 INSCC

The University of Utah seeks to provide equal access to its programs, services, and activities to people with disabilities. Reasonable prior notice is needed to arrange accommodations.

UNIVERSITY OF UTAH
Center for High Performance Computing
155 South 1452 East, RM #405
SALT LAKE CITY, UT 84112-0190

Welcome to CHPC News!

If you would like to be added to our mailing list, please fill out this form and return it to:

Vicky Volcik
UNIVERSITY OF UTAH
Center For High Performance Computing
155 S 1452 E ROOM 405
SALT LAKE CITY, UT 84112-0190
FAX: (801)585-5366

(room 405 of the INSCC Building)

Name:
Phone:

Department or Affiliation:
Email:

Address:
(UofU campus or U.S. Mail)

Thank you for using our Systems!

Please help us to continue to provide you with access to cutting edge equipment.

ACKNOWLEDGEMENTS

If you use CHPC computer time or staff resources, we request that you acknowledge this in technical reports, publications, and dissertations. Here is an example of what we ask you to include in your acknowledgements:

"*A grant of computer time from the Center for High Performance Computing is gratefully acknowledged.*"

If you use Arches, please add:

"partially supported by NIH-NCRR grant # 1S10RR17214."

Please submit copies of dissertations, reports, preprints, and reprints in which the CHPC is acknowledged to: Center for High Performance Computing, 155 South 1452 East, Rm #405, University of Utah, Salt Lake City, Utah 84112-0190