High Throughput Computing On Campus

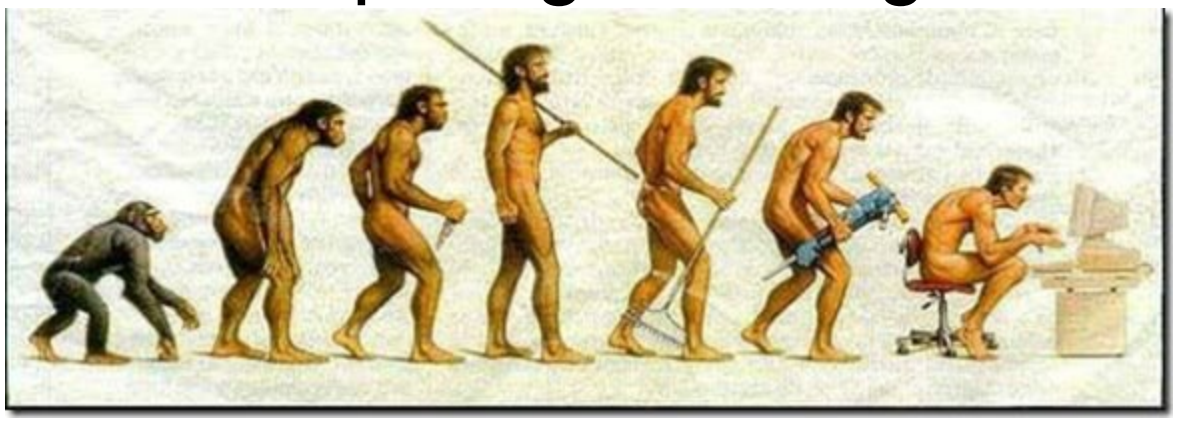
High Throughput Computing:
How we got here, Where we are

Condor Week 2010, Madison, WI
April 2010

Todd Tannenbaum, Greg Thain, Zach Miller
{tannenba, gthain, zmiller } @cs.wisc.edu
Center for High Throughput Computing
Department of Computer Sciences

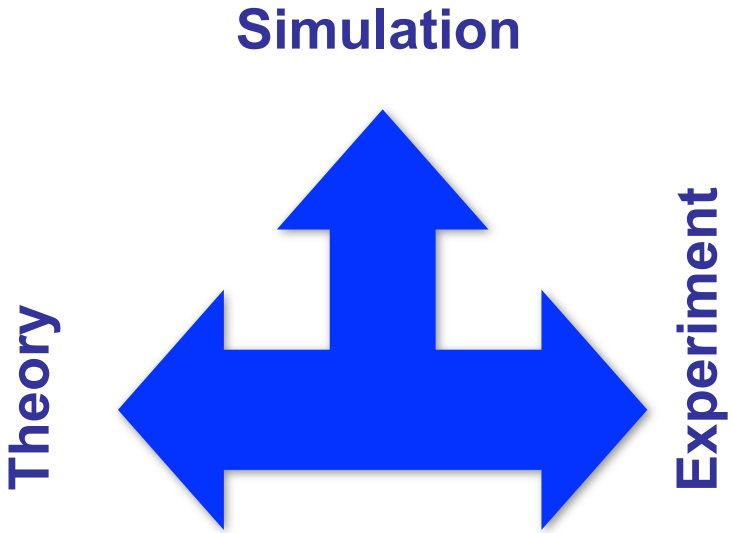
Agenda

- Brief history of HTC computing
- High Throughput Computing on Campus
  - What is available
  - How to get started
- Examples of how other research groups teamed up w/ the CHTC to tackle their scientific computing challenges



Why we need compute power

- Three-fold scientific method
  - Theory
  - Experiment
  - Computational analysis
- Cycles as “fertilizer”



=

The basic problem

- Speed of light
- 1 ft / nanosecond
  - (roughly)

- 3 GHz CPU =
  - 1/3 foot per clock

Parallelism: The Solution

– Parallelism can cover latency with bandwidth

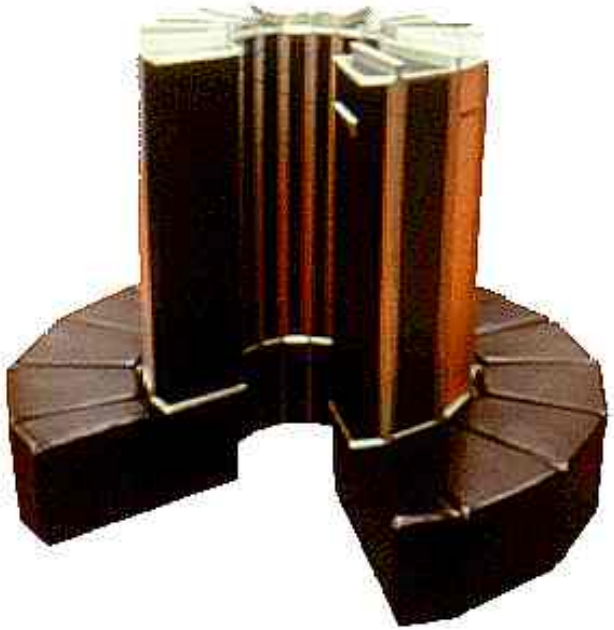
– But at what level should parallelism be?
  - This is basic question for last 40 years

  - Programmer visible?
  - Compiler visible?
  - Instruction level?
  - Machine level?
  - etc. etc. etc.

Some History…

Start with the past…

- Early 70's : Seymour Cray asks why one by one?
- Vector Supercomputers born
- Mid 70s thru 80s, Cray's vector supercomputers set records
- SIMD machines, explicit parallelism



Cray-1 ~ 1976

Parallel Machines

- Late 80's – Early 90's
- Dozens to thousands of cheap(er) CPUs (such as SPARC)
- Ex: KSR and Thinking Machines
- Difficult to program
  - Shared Memory?
  - Message Passing?
  - Multiple Threads?

Up to Today: 20-teens?

- MegaWatts, not MegaFlops
- Massively parallel
  - Millions of cores
- Massively expensive
  - Megabucks
- Massively hard to program
  - Explicit parallelism via MPI

IBM Blue Gene

Meanwhile, back in the lab, another kind of computing was developing...

Initial Condor Development

- Started in 1986
- Cycle-scavenging
  - Also called "Opportunistic"
- Desktop Workstations
- Parallelism via processes
  - (Pleasantly parallel)

SETI@home

- Started 1999
- “Volunteer” computing
- Similar to HTC, but different focus, scale
- Process level parallelism

Birth of off-the-shelf compute clusters

- Beowulf Project started in 1994
- Off-the-shelf commodity PCs, networking
- Parallelism via processes

Late 90s and beyond: Why Commodity Clusters?

- Massive cost/performance gain of CPU and network technology.
- The industry started to provide fully assembled subsystems (microprocessors, motherboards, disks and network interface cards).
- Mass market competition has driven the prices down and reliability up for these subsystems.
- The availability of open source software, particularly the Linux operating system, GNU compilers and programming tools, MPI / PVM message passing libraries, workload managers such as Condor, PBS.
- The recognition that obtaining high performance, even from vendor provided, on parallel platforms can be hard work
- An increased reliance on computational science which demands high performance computing.

All of these factors contributed to...

The rise of Distributed Ownership

- Due to dramatic decrease in the cost-performance ratio of hardware, powerful computing resources fell into the hands of individuals, groups, departments, …
  - Huge increase in the aggregate processing capacity owned by the organization
  - Much smaller increase in the capacity accessible by a single person

The fall of High Performance Computing

- High Performance - Very large amounts of processing capacity over short time periods (FLOPS - Floating Point Operations Per Second)
- High Throughput - Large amounts of processing capacity sustained over very long time periods (FLOPY - Floating Point Operations Per Year)

FLOPY ≠ 30758400*FLOPS

The rise of
High Throughput Computing

Process Level parallelism for long timeframes

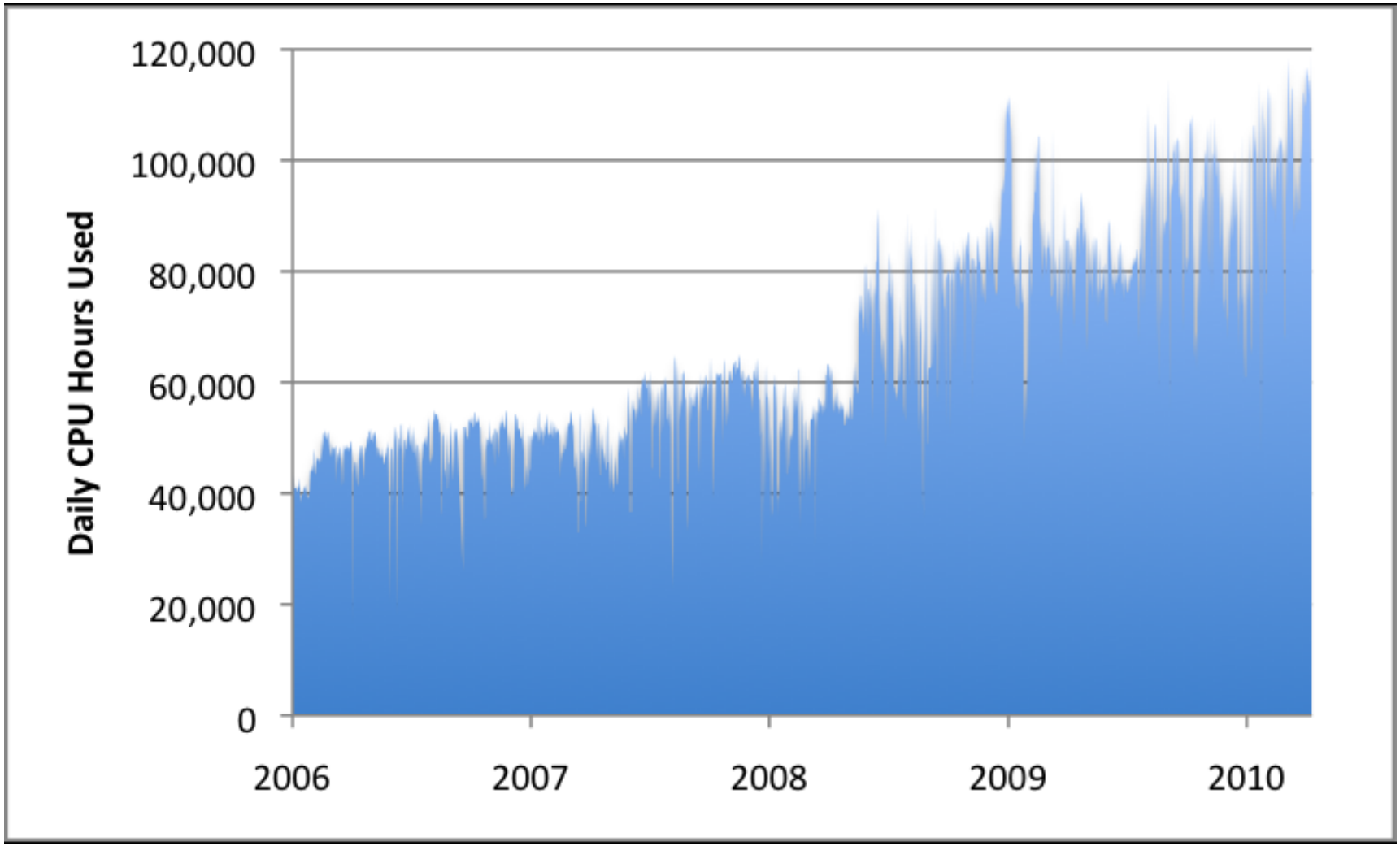
Opportunistic computing

Perhaps at different scales

So given that background...

In August 2006, the UW Academic Planning Committee approved the Center for High Throughput Computing (CHTC). The College of L&S then staffed positions for the center.

Since 2006: 101M CPU Hrs (12kYrs)



About the CHTC

- Goal
  - Offer the UW community computing capabilities that enable scientific discovery.
- How
  - Work with research projects and the campus on the organization, funding, creation, and maintenance of a *campus computing grid*
  - Strong commitment to the *engagement* model; be ready to work with research groups and applications
  - Involve UW in national and international grid efforts.

About the CHTC, cont.

- Who
  - Steering Committee
    - Guri Sohi, CS Department Chair
    - Miron Livny, Condor Project PI
    - Todd Tannenbaum, Condor Project Technical Lead
    - Juan de Pablo, Professor
    - David C. Schwartz, Professor
    - Wesley Smith, Professor
  - Staffing
    - Primarily comes from the projects associated with the CHTC

About the CHTC, cont.

- What
  - Adminster UW Campus distributed computing resources
  - Linked with world-wide grid via OSG
- Low ceremony engagements

About the CHTC, cont

- Force multiplier for UW Campus
- Can help with *grantsmanship*
- Some help with HTC best practices
  - We won't write your code, but will do much to help
  - Can play a matchmaker role, hooking you up w/ others on campus

The rest of this presentation

- Introduce some a few HTC resources in operation today on our campus
  - Grid Laboratory of Wisconsin (GLOW)
  - Open Science Grid (OSG)
- So who currently uses this stuff? Examples…
- How to get involved…

So you can be like this student

Adam Butts, who is now with IBM Research. On the Thursday before his PhD defense on Monday, he was still many simulations short of having final numbers. GLOW provided him with an on-demand burst of 15,000 CPU hours over the weekend, getting him the numbers he needed:

```
From: "J. Adam Butts" <butts@cs.wisc.edu>
Subject: Condor throughput

The Condor throughput I'm getting is really
incredible.  THANK YOU so very much for the extra
help.  My deadline is Monday, and if I'm able to meet
it, you will deserve much of the credit.  The special
handling you guys have offered when a deadline must be
met (not just to me) is really commendable.  Condor is
indispensable, and I'm sure it has a lot to do with
the success of (at least) the architecture group at
UW.  I can't imagine how graduate students at other
universities manage...
```

Or like this student

Adam Tregre, a physics graduate student, has used over 70,000 hours of compute time. He reports:

In "Neutrino mass limits from SDSS, 2dFGRS and WMAP," (PLB 595:55-59, 2004), we performed a six-dimensional grid computation, which allowed us to put an upper bound on the neutrino mass of 0.75 eV at two sigma (1.11 eV at three sigma). We calculated CMB and matter power spectra, and found the corresponding chi^2 value at approximately 106 parameter points, a feat only made possible by GLOW. In an ongoig analysis, we employ a Markov Chain Monte Carlo analysis to determine what type of cosmomological model (standard LCDM, inflation, varying dark energy equation of state, etc.) provides the best fit for current CMB, LSS, and SN observations. GLOW provides an excellent tool for the many parameter fittings and different starting points necessary for such an analysis.

Lets start our tour with the Condor Project

The Condor Project

Distributed Computing **research** project in the Comp Sci dept performed by a team of faculty, full time staff and students who

- face **software/middleware engineering** challenges,
- involved in national and international **collaborations**,
- interact with **users** in academia and industry,
- maintain and support a distributed **production** environments
- and educate and train **students**.

Funding (~ $4.5M annual budget)

Condor Project has a lot of experience... (Est 1985!)

What is the Condor software?

- Open source distributed computing software developed and maintained by the Condor Project
- Condor converts collections of distributively owned computers into a distributed high-throughput computing (HTC) facility.
  - It is the cornerstone technology behind the UW Campus Grid (and many others).
- Condor manages both resources (machines) and resource requests (jobs)

Federating Condor Pools

- Across campus
  - Simple to set up “flocking”
  - If not enough machines in your pool, your job will “flock” to a friendly pool.
  - You are responsible for finding friends.
- Across the world
  - Condor interoperates with many evolving “grid protocols”
    - Remote machines do not need to be running Condor

The Story of GLOW: The Grid Laboratory of Wisconsin

How GLOW got started

Seven departments on Campus needed computing, big time.

- Computational Genomics, Chemistry
- High Energy Physics (CMS, Atlas)
- Materials by Design, Chemical Engineering
- Radiation Therapy, Medical Physics
- Computer Science
- Amanda, Ice-cube, Physics/Space Science
- Plasma Physics

Diverse users with different conference deadlines, and usage patterns.

GLOW

- GLOW Condor pool is distributed across the campus at the sites of the machine owners.
  - 1800 cores
  - 100 TB disk
  - Over 25 million CPU-hours served
  - Contributed to ~50 publications
  - Clusters' owner always has highest priority

Why join together to build the GLOW campus grid?

very high utilization

more diverse users = less wasted cycles

simplicity

All we need is Condor at campus level. Plus, we get the full feature-set rather than lowest common denominator.

collective buying power

We speak to vendors with one voice.

consolidated administration

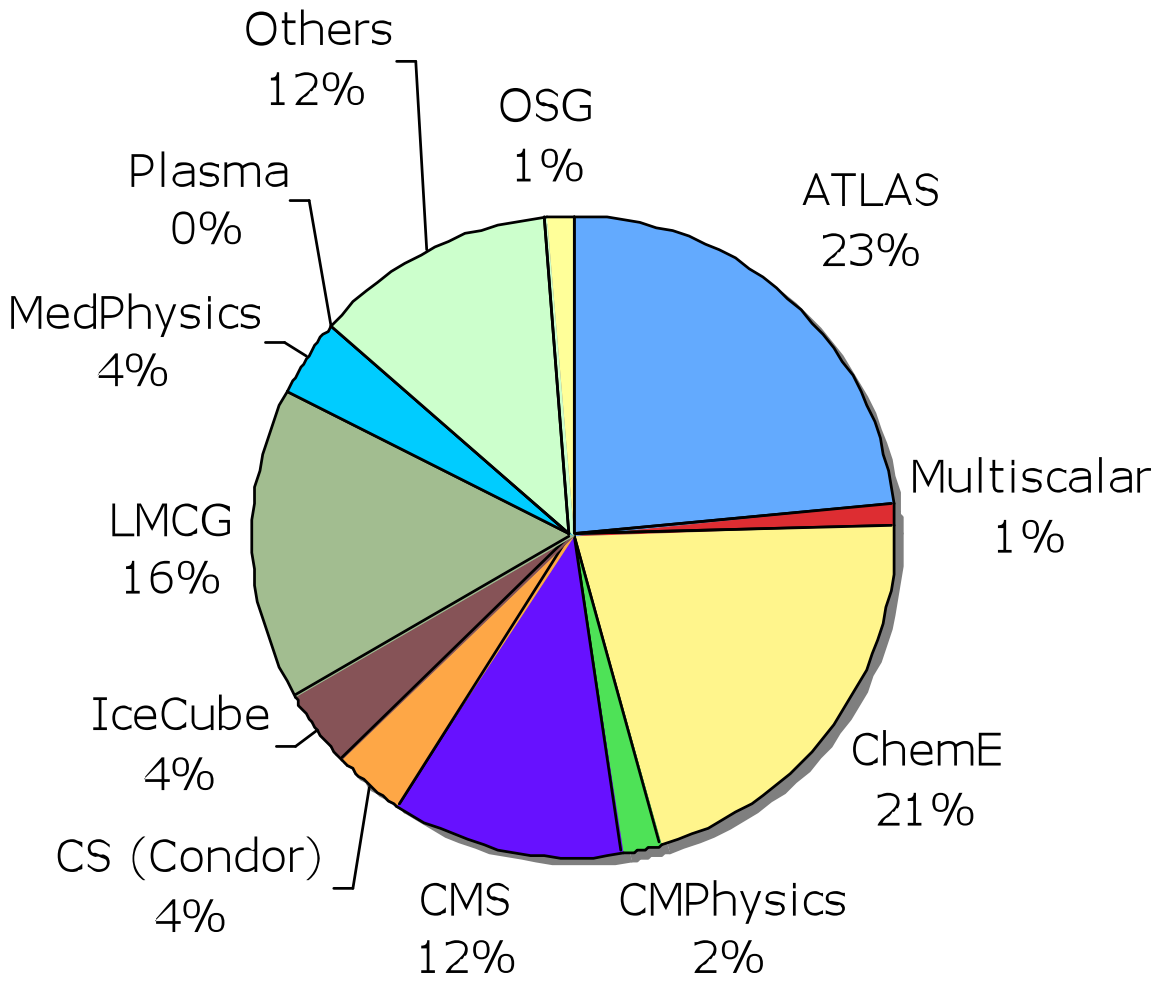
Fewer chores for scientists. Fewer holes for hackers.

More money for research, less money for overhead.

synergy

Face-to-face technical meetings between members.
Mailing list scales well at campus level.

Fractional Usage of GLOW



Molecular and Computational Genomics

Availability of computational resources serves as a catalyst for the development of new genetic analysis approaches.

Professor David C. Schwartz: *"We have been a Condor user of several years, and it is now unthinkable that we could continue our research without this critical resource."*

Physics

The Large Hadron Collider under construction at CERN will produce over 100PB of data over then next few years. About 200 “events” are produced per second, each requiring about 10 minutes of processing on a 1GHz Intel processor.

Professors Sridhara Rao Dasu, Wesley H. Smith and Sau Lan Wu: “*The only way to achieve the level of processing power needed for LHC physics analysis is through the type of grid tools developed by Prof. Livny and the Condor group here at Wisconsin.*”

Chemical Engineering

In computation fluids and materials modeling research, a single simulation commonly requires several months of dedicated compute time. Professor Juan de Pablo's group leverages Condor and GLOW to satisfy this demand.

Professor Juan de Pablo: *"The University of Wisconsin is a pioneer in [grid computing]. A case in point is provided by a number of outstanding, world-renowned research groups within the UW that now rely on grid computing and your team for their research."*

Why not GLOW across campus?
UW Campus Grid

Many federated Condor pools across campus, including CHTC and GLOW resources

Condor installs at UW-Madison

- Numerous Condor Pools operating at UW, several at a department or college level:
  - CHTC Pool: ~1000 cpu and growing
  - GLOW Pool: ~1800 cpu and growing
  - CAE Pool: ~ 1200 cpu
  - Comp Sci Pool: ~ 1215 cpu
  - Biostat Pool: ~ 132 cpu
  - DoIT InfoLabs Pools: ~200 cpus and growing
  - Euclid Pool: ~2000 cpu
- Condor *Flocking*
  - Jobs move from one pool to another based upon availability.

Usage from *just* the CS Pool

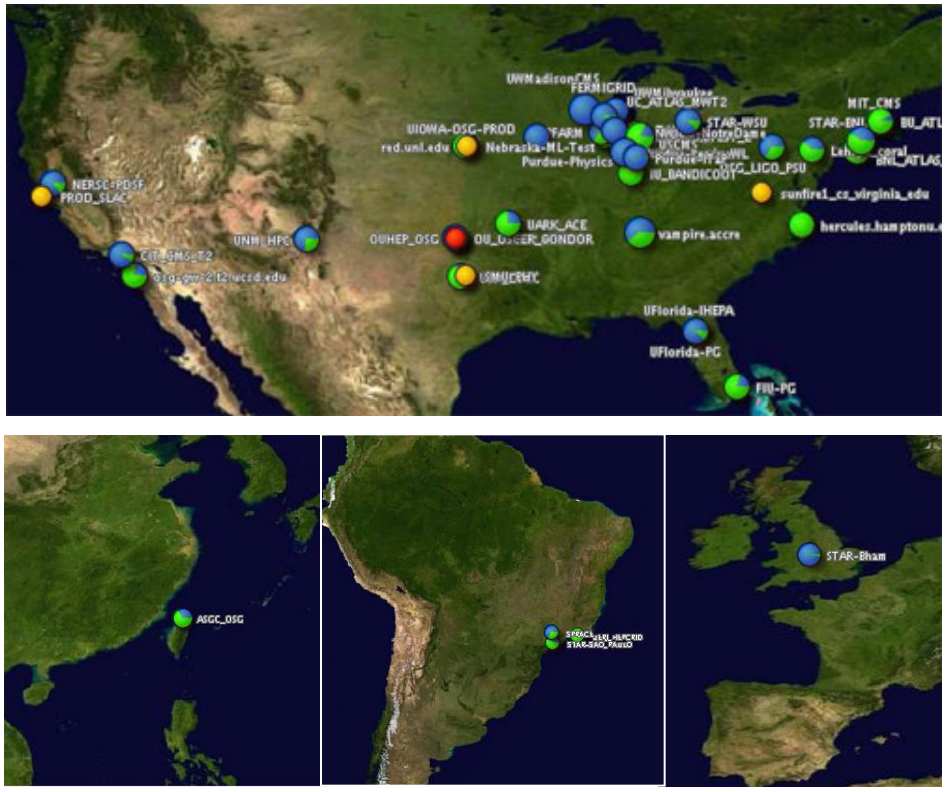
User	Hours	Pct
payseur@cs.wisc.edu	313205.7	6.1%
dang@cs.wisc.edu	302318.1	5.9%
skolya@cs.wisc.edu	284046.1	5.5%
lyen@cs.wisc.edu	259001.3	5.0%
liyinlee@cs.wisc.edu	211382.4	4.1%
bmrb@cs.wisc.edu	166260.7	3.2%
rabosch@cs.wisc.edu	150174.0	2.9%
szhou@lmcg.wisc.edu	121764.7	2.4%
nater@cs.wisc.edu	107089.7	2.1%
osg_cmsprod@hep.wisc.edu	106475.1	2.1%
yu-chi@cs.wisc.edu	103529.2	2.0%
corneliu@cs.wisc.edu	98983.6	1.9%
....		
TOTAL	5142471	100%

Open Science Grid (OSG): Linking grids together

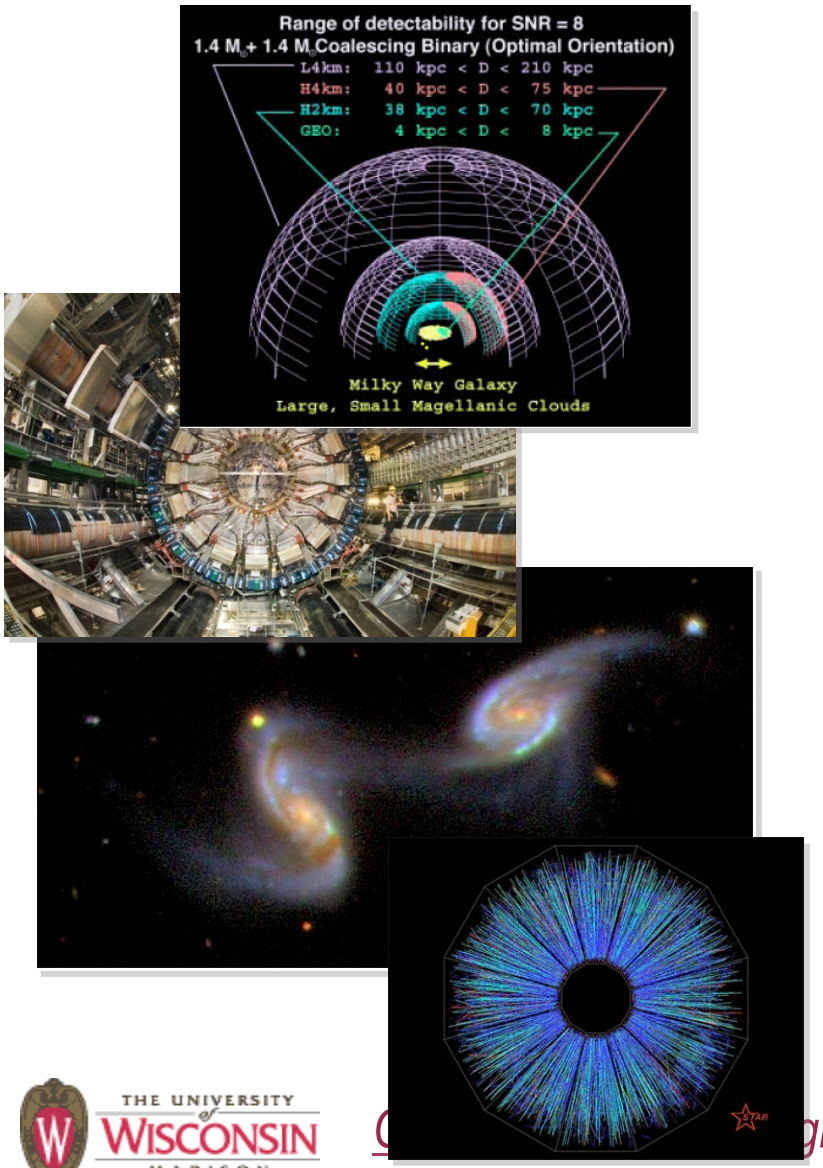
The Open Science Grid

A framework for large scale distributed resource sharing
addressing the technology, policy, and social requirements of sharing

- OSG is a consortium of software, service and resource providers and researchers, from universities, national laboratories and computing centers across the U.S., who together build and operate the OSG project. The project is funded by the NSF and DOE, and provides staff for managing various aspects of the OSG.
- Brings petascale computing and storage resources into a uniform grid computing environment
- Integrates computing and storage resources from over 50 sites in the U.S. and beyond



Principal Science Drivers

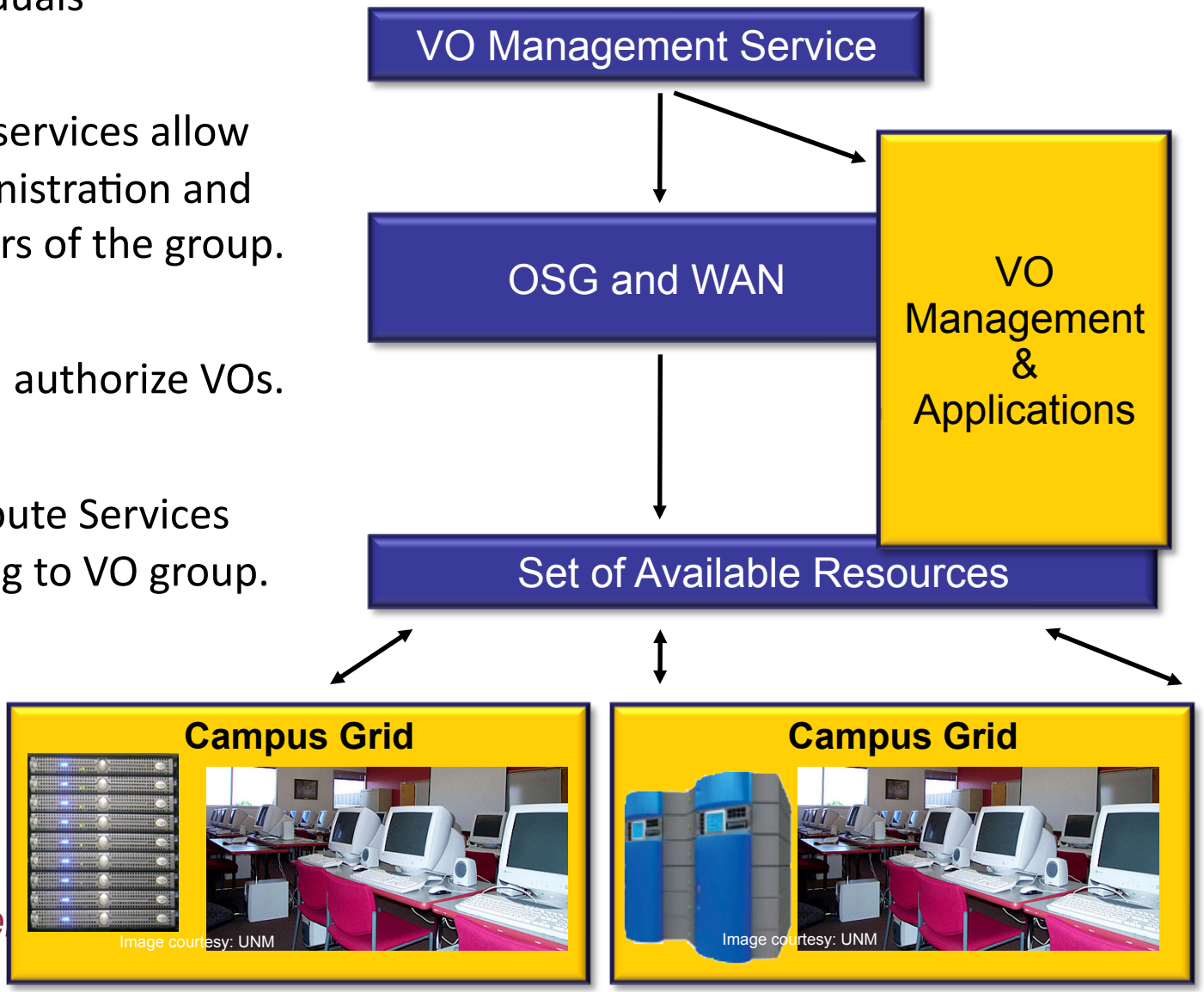


- High energy and nuclear physics
  - 100s of petabytes (LHC) 2007
  - Several petabytes 2005
- LIGO (gravity wave search)
  - 0.5 - several petabytes 2002
- Digital astronomy
  - 10s of petabytes 2009
  - 10s of terabytes 2001
- Other sciences emerging
  - Bioinformatics (10s of petabytes)
  - Nanoscience
  - Environmental
  - Chemistry
  - Applied mathematics
  - Materials Science

THE UNIVERSITY of WISCONSIN MADISON

Virtual Organizations (VOs)

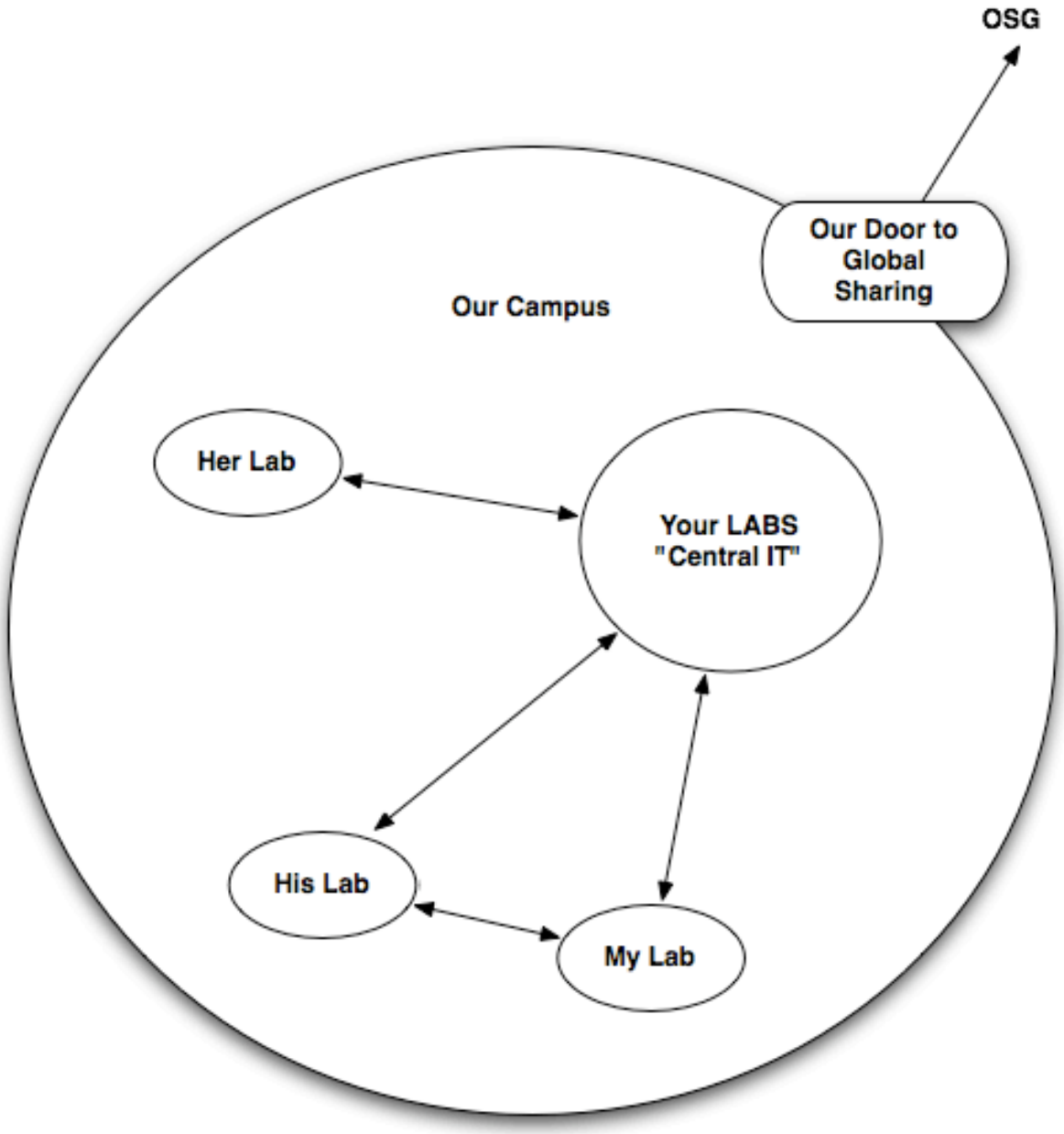
- The OSG Infrastructure trades in Groups not Individuals
- VO Management services allow registration, administration and control of members of the group.
- Facilities trust and authorize VOs.
- Storage and Compute Services prioritize according to VO group.



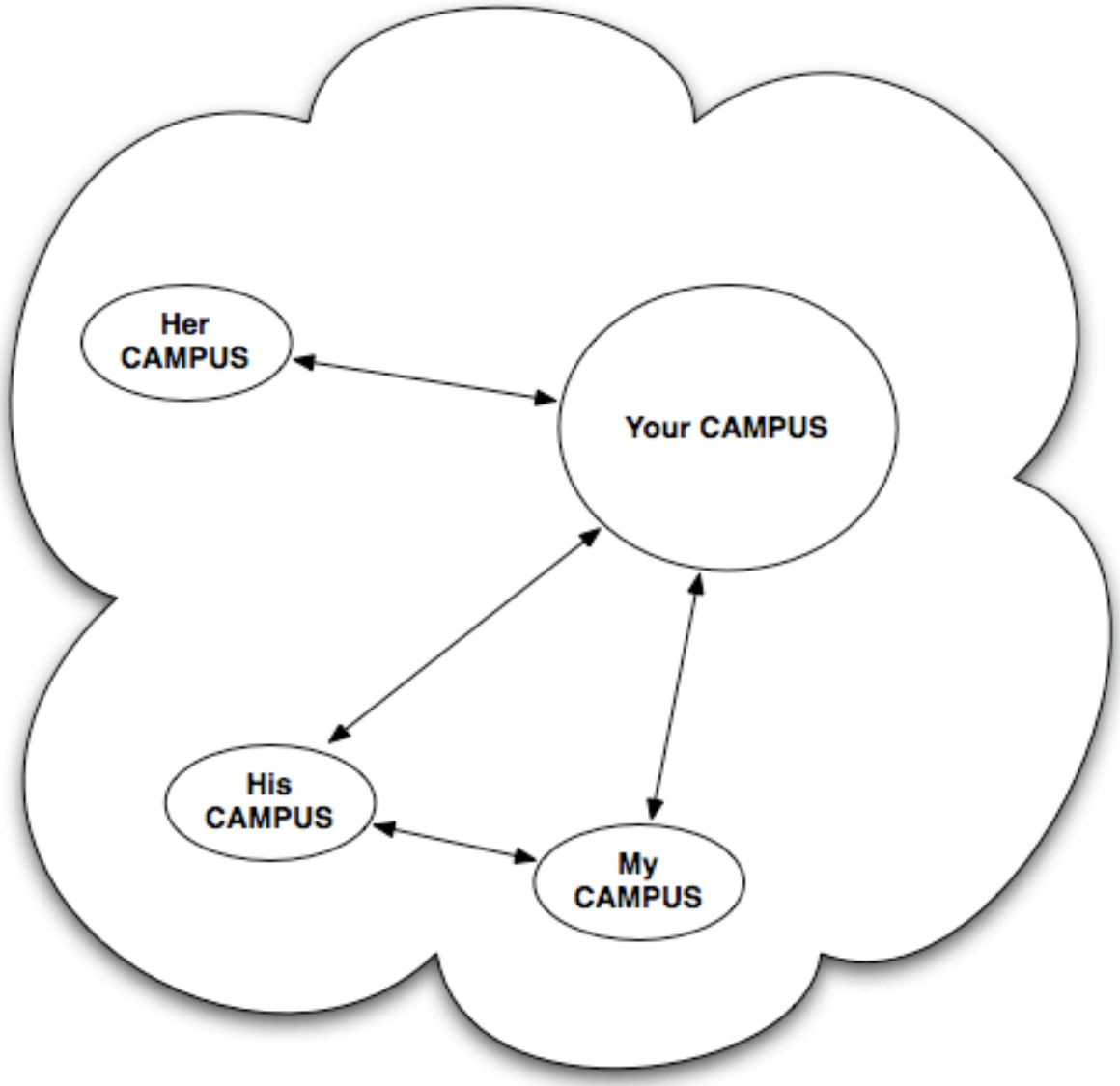
Current OSG Resources

- OSG has more than 50 participating institutions, including self-operated research VOs, campus grids, regional grids and OSG-operated VOs
- Provides about 10,000 CPU-days per day in processing
- Provides 10 Terabytes per day in data transport
- CPU usage averages about 75%
- OSG is starting to offer support for MPI

Simplified View: Campus Grid



Simplified View: OSG Cloud



Why should UW facilitate (or drive) cross-campus resource sharing?

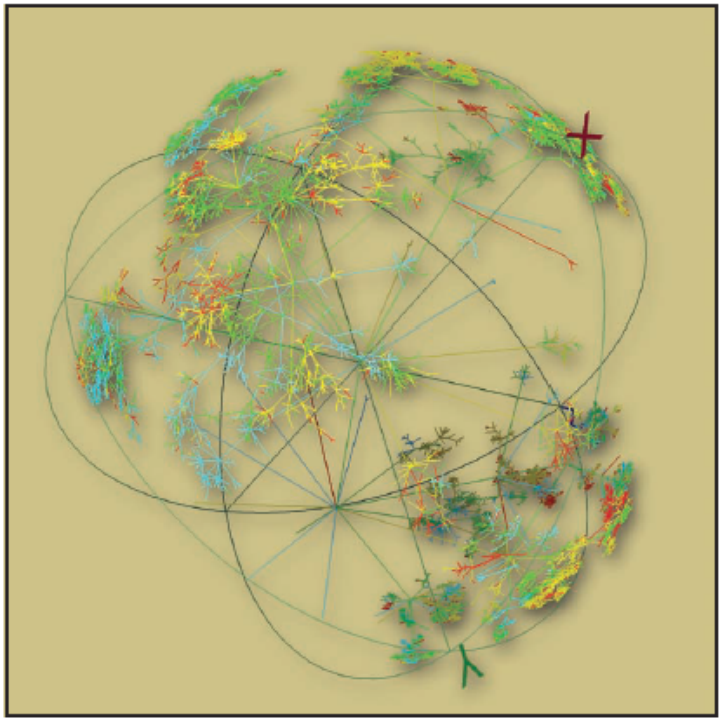
Because it's the right thing to do

- Enables new modalities of collaboration
- Enables new levels of scale
- Democratizes large scale computing
- Sharing locally leads to sharing globally
- Better overall resource utilization
- Funding agencies

Open Science Grid

At the heart of the cyberinfrastructure vision is the development of a cultural community that supports peer-to-peer collaboration and new modes of education based upon broad and open access to leadership computing; data and information resources; online instruments and observatories; and visualization and collaboration services.

Cyberinfrastructure Vision for 21st Century Discovery



National Science Foundation
Cyberinfrastructure Council
March 2007

CI_Vision_March07.pdf - Adobe Acrobat Professional

File Edit View Document Comments Forms Tools Advanced Window Help

An effective computing environment designed to meet the computational needs of a range of science and engineering applications will include a variety of computing systems with complementary performance capabilities. NSF will invest in leadership class environments in the 0.5-10 petascale performance range. Strong partnerships involving other federal agencies, universities, industry and state government are also critical to success. NSF will also promote resource sharing between and among academic institutions to optimize the accessibility and use of HPC assets deployed and supported at the campus level. Supporting software services include the provision of intelligent development and problem-solving environments and tools. These tools are designed to provide improvements in ease of use, reusability of modules, and portable performance.

[1] A picosecond is 10^{-12} second
[2] A petascale is 10^{15} operations per second with comparable storage and networking capacity

e density for iron oxide (FeO) within the local density approximation, with ors represent the spin density, showing the antiferromagnetic ordering.

8.50 x 11.00 in

THE UNIVERSITY of WISCONSIN MADISON

Center for Hig

What level of know-how, experience, and influence does the CHTC and UW-Madison have in OSG?

Notice any similarities?

Open Science Grid PI

UW CHTC Director

Great! How can I get involved?

What can you do to get started

- If you have batch ready jobs, we can get you going quickly. Talk to us!!
- If you are developing software, don't go "High Performance", go AGILE
  - Example: Parallel tightly coupled -vs- independent tasks, or master-worker
  - Example: Octave -vs- Matlab
  - Example: Java -vs- C or Fortran?
- Don't think operations per second, think how many different environments can my program run?

How does it work?

- Any PI at UW can get access
- Email chtc@cs.wisc.edu
- Small blurb about your work
- Initial engagement
  - Talk about your problem
  - We can help recommend/suggest resources and tools
  - We can help get you going (scripting, DAGMan files, etc)
  - Create accounts and/or install in your lab

Examples

Wisconsin Center for Education
Social Science Computing Coop
SiPHT (Workflows!)
MPI the HTC way

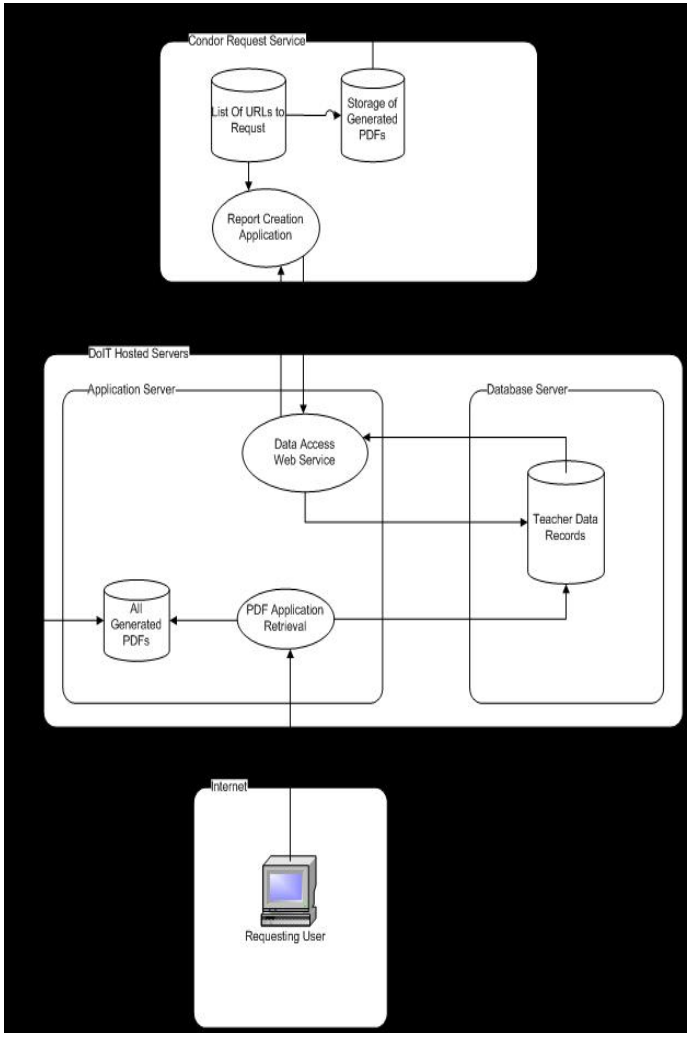
Teacher Evaluation Reports

NYC School System

- UW Wisconsin Center for Education Research
- Problem of large scale individual-level reporting.
- "The current bottle neck that has not yet been Condorized is the production of reports or the individual graphics that might populate dynamically generate reports to serve very large populations. *The immediate need is to deal with the problem of taking over a week to generate the 18,000 PDF reports that we need to provide individual teacher and building leader results to teachers in the New York City public school system.*"

Teacher Evaluation Reports

NYC School System



- Partioned so each job runs for an hour (generates 100 reports) enabling us to …
- Runs opportunistically on Windows at CAE and DoIT
- Each run used to take a week per run – now completes in 2 hours.

Social Science Computing Cooperative

"We will produce estimations of causal effects for various samples of the Texas Higher Educational Opportunity Project through propensity score matching techniques. These estimated causal effects are then tested through statistical simulations of various parameters each run at 500 iterations."

The problem is they have a small cluster (34 nodes) and this project had *400 Stata jobs and many of them ran over a day. The application, being commercial, could not be run in the standard universe and exceeded their limits in cluster size and CHTC's limits for runtime.*

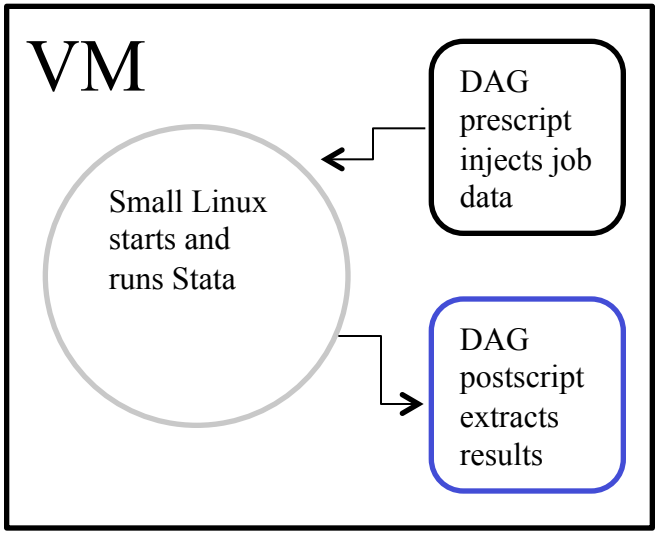
Social Science Computing Cooperative

Solution was to run Stata within a virtual machine and to run that machine as the actual job via Condor's VM universe. In Condor, VM universe jobs will checkpoint/ restarted when they get evicted.

Social Science Computing Cooperative

Solution was to run Stata within a virtual machine and to run that machine as the actual job via Condor's VM universe.

We built a DAG with 400 jobs in the VM Universe which restarted as needed.

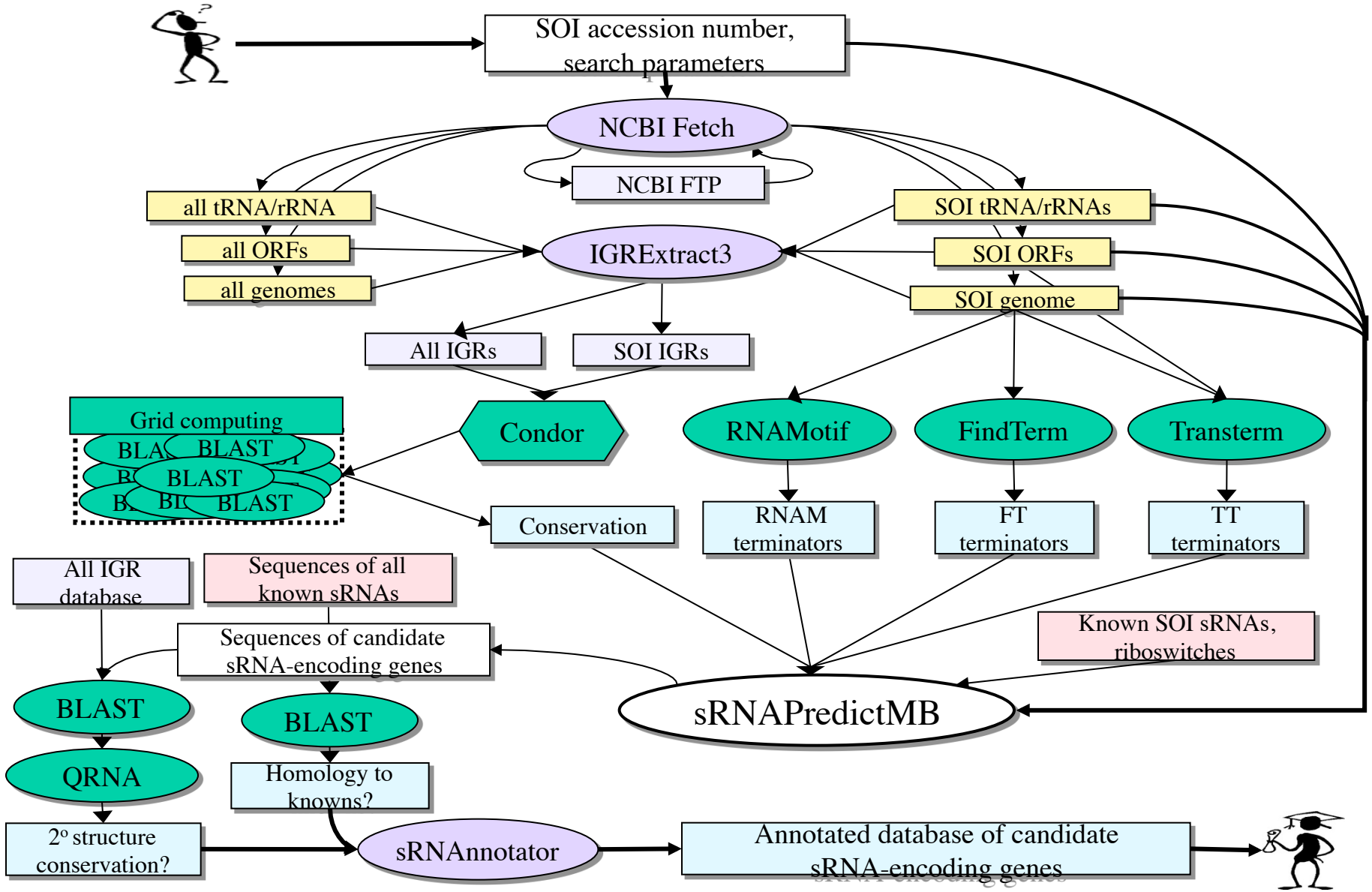


SIPHT Portal

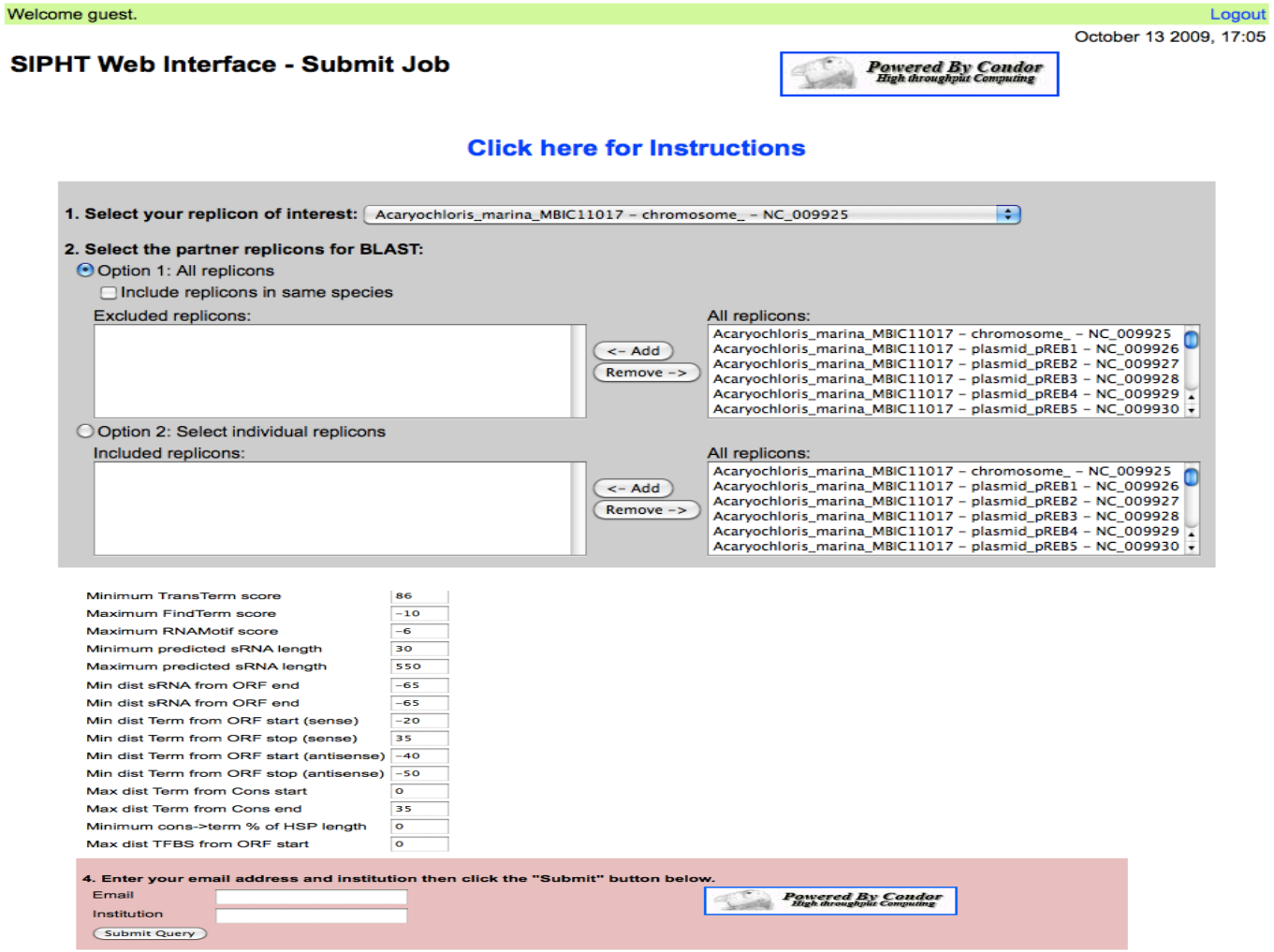
- Collaboration with Harvard Medical, UW, and ISI
- Relatively complicated workflow, managed by DAGMan and Pegasus
- Made “simple” using a web portal that emails the results to the end user

SIPHT Portal

- High throughput computational tool utilizing workflow management
- Conducts kingdom-wide predictions and annotations of intergenic sRNA-encoding genes.
- Searches for putative sRNA-encoding genes in all 1640 bacterial replicons in the NCBI database.


SOI accession number, search parameters
NCBI Fetch
NCBI FTP
all tRNA/rRNA
all ORFs
all genomes
IGRExtract3
SOI tRNA/rRNAs
SOI ORFs
SOI genome
All IGRs
SOI IGRs
Grid computing
BLAST
BLAST
BLAST
Condor
RNAMotif
FindTerm
Transterm
Conservation
RNAM terminators
FT terminators
TT terminators
All IGR database
Sequences of all known sRNAs
Sequences of candidate sRNA-encoding genes
Known SOI sRNAs, riboswitches
sRNAPredictMB
BLAST
BLAST
QRNA
Homology to knowns?
2º structure conservation?
sRNAnotator
Annotated database of candidate sRNA-encoding genes

Web portal interface



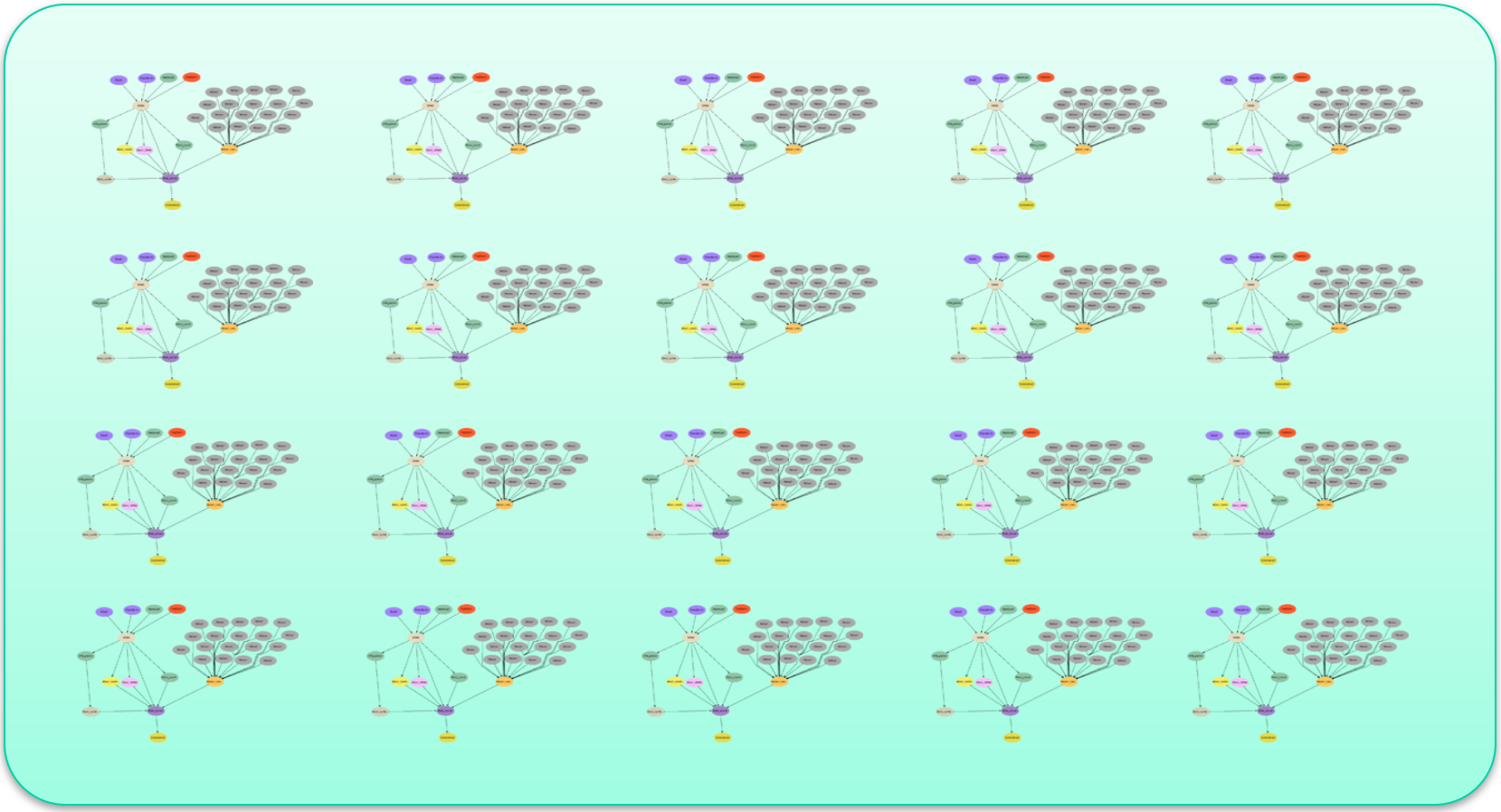
THE UNIVERSITY of WISCONSIN MADISON

Center for High Throughput Computing

SIPHT Portal

- In addition to the web interface, SIPHT is also used on a large scale from the command line
- 1640 replicons run for about 2 hours each
- Refinements in the code require recomputing all data

Annotate all ROI (1640) (recursive DAX)

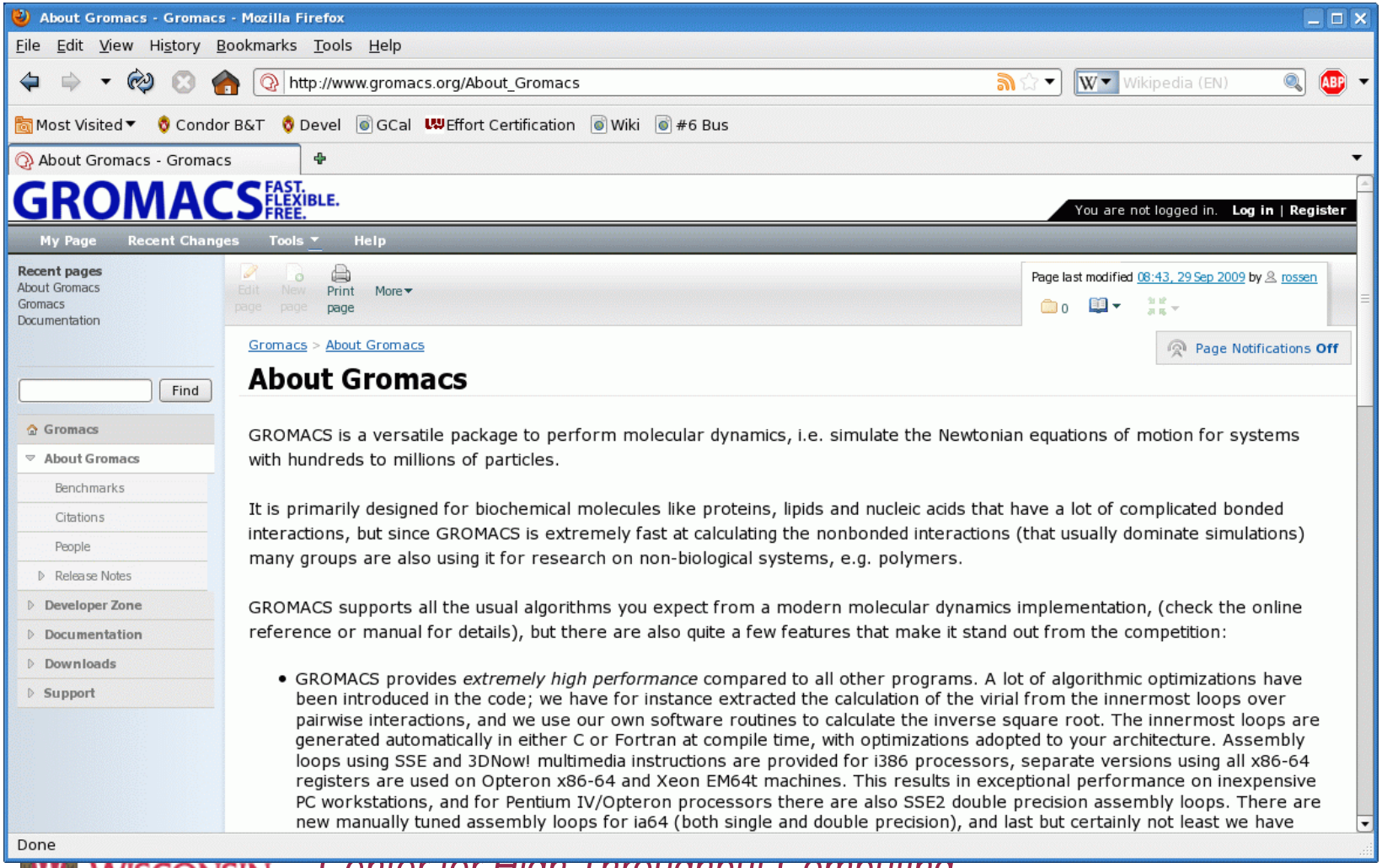


SIPHT Portal

- Single script to do all the work
- First implementation used DAGMan alone
- Second pass uses Pegasus to do higher level planning
- Third pass (yet to be done) will do parameter sweeps and cache results

CHTC working w/ Chemistry Dept

Gromacs


About Gromacs - Gromacs - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.gromacs.org/About_Gromacs
Most Visited Condor B&T Devel GCal Effort Certification Wiki #6 Bus
About Gromacs - Gromacs
GROMACS FAST. FLEXIBLE. FREE.
You are not logged in. Log in | Register
My Page Recent Changes Tools Help
Recent pages
About Gromacs
Gromacs
Documentation
Edit page New page Print page More
Page last modified 08:43, 29 Sep 2009 by rossen
Gromacs > About Gromacs
Page Notifications Off
About Gromacs
Find
Gromacs
About Gromacs
Benchmarks
Citations
People
Release Notes
Developer Zone
Documentation
Downloads
Support
GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles.
It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the nonbonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.
GROMACS supports all the usual algorithms you expect from a modern molecular dynamics implementation, (check the online reference or manual for details), but there are also quite a few features that make it stand out from the competition:
GROMACS provides extremely high performance compared to all other programs. A lot of algorithmic optimizations have been introduced in the code; we have for instance extracted the calculation of the virial from the innermost loops over pairwise interactions, and we use our own software routines to calculate the inverse square root. The innermost loops are generated automatically in either C or Fortran at compile time, with optimizations adopted to your architecture. Assembly loops using SSE and 3DNow! multimedia instructions are provided for i386 processors, separate versions using all x86-64 registers are used on Opteron x86-64 and Xeon EM64t machines. This results in exceptional performance on inexpensive PC workstations, and for Pentium IV/Opteron processors there are also SSE2 double precision assembly loops. There are new manually tuned assembly loops for ia64 (both single and double precision), and last but certainly not least we have
Done

30+ day runtime

- Too long, even as HTC
- Step one – compile with SSE support
  - 10x improvement
- Just a Gromac compile-time option
  - Hand-coded assembly, not gcc option

3 days still too long…

- Gromacs also support MPI
- CHTC doesn't have infiniband

- What do to?

Whole machine jobs

- Submit file magic to claim all 8 slots

```
universe = vanilla
requirements = (CAN_RUN_WHOLE_MACHINE =?= TRUE)
+RequiresWholeMachine=true
executable = some job
arguments = arguments
should_transfer_files = yes
when_to_transfer_output = on_exit
transfer_input_files = inputs
queue
```

MPI on Whole machine jobs

Whole machine mpi submit file

```
universe = vanilla
requirements = (CAN_RUN_WHOLE_MACHINE =?= TRUE)
+RequiresWholeMachine=true

executable = mpiexec

arguments = -np 8 real_exe

should_transfer_files = yes
when_to_transfer_output = on_exit

transfer_input_files = real_exe

queue
```

Condor Motto:

If you want it,

Bring it yourself

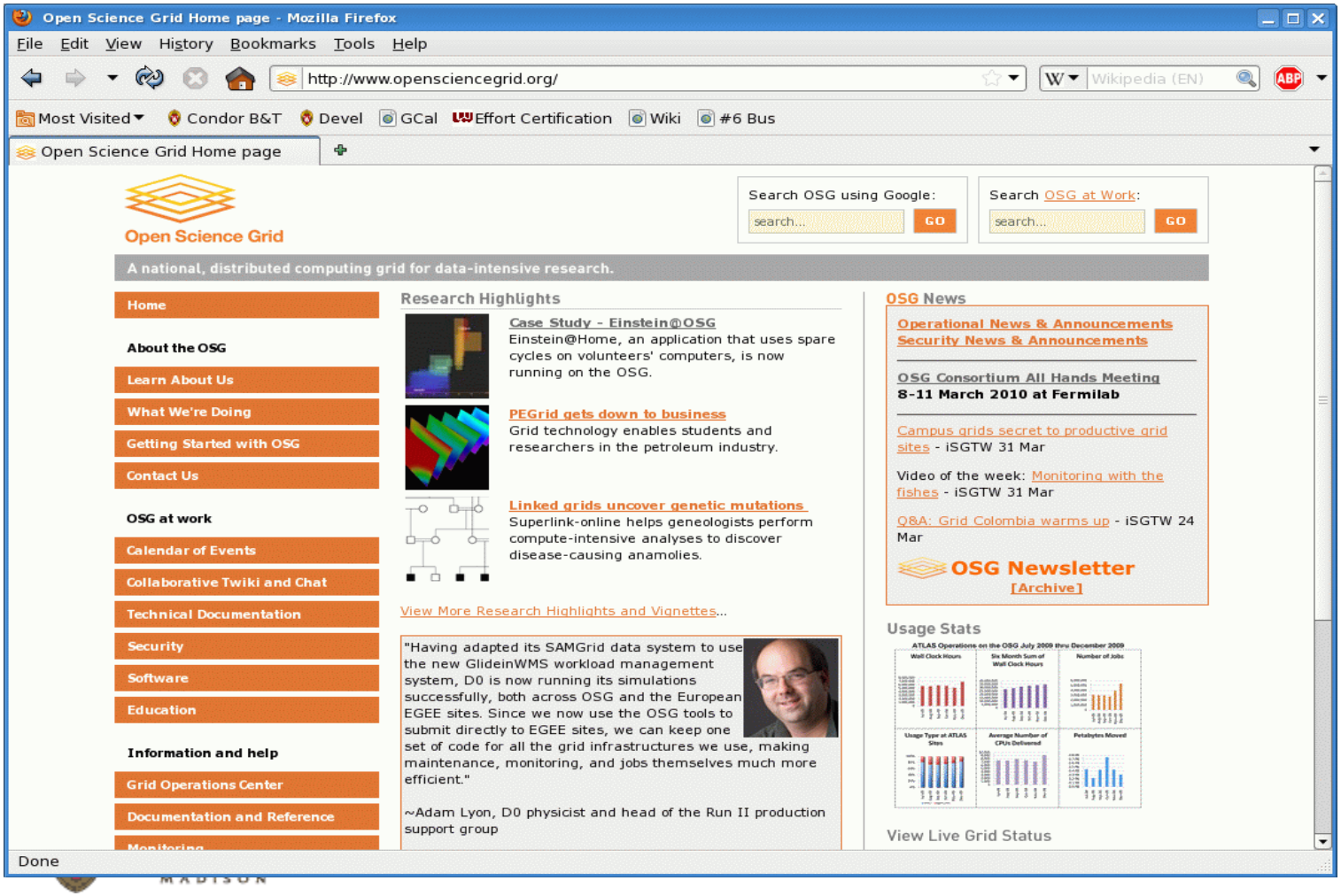
Advantages

- Condor is parallel agnostic:
  - MPICH, OpenMPI, pthreads, fork, etc.
- High-bandwith memory transport
- Easy to debug
  - Ssh-to-job still works
- Access to all machine's memory

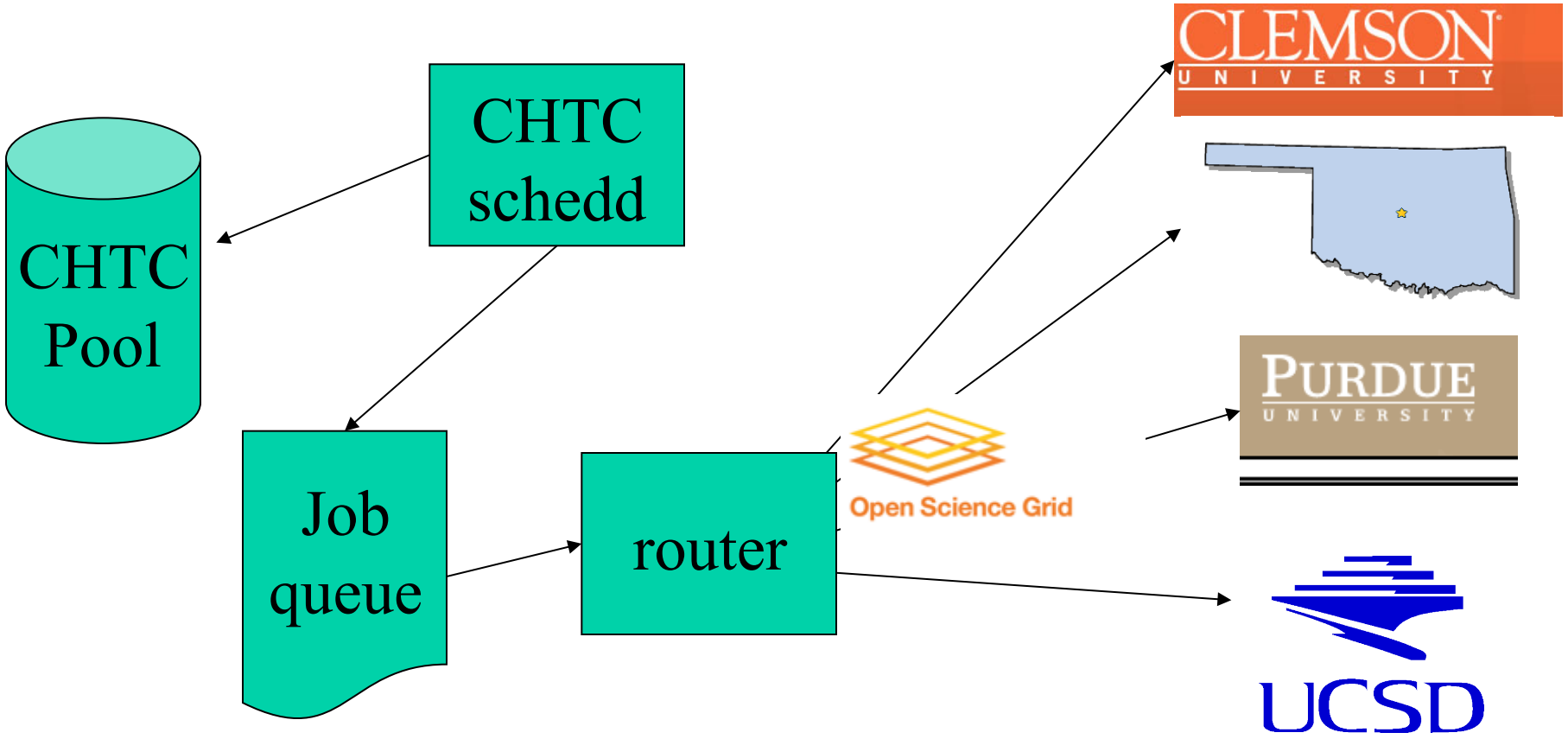
Disadvantages

- Still need to debug parallel program
  - helps if others already have
- Fewer full-machine slots
  - Currently 15, more coming

15 machines not enough: OSG to the rescue



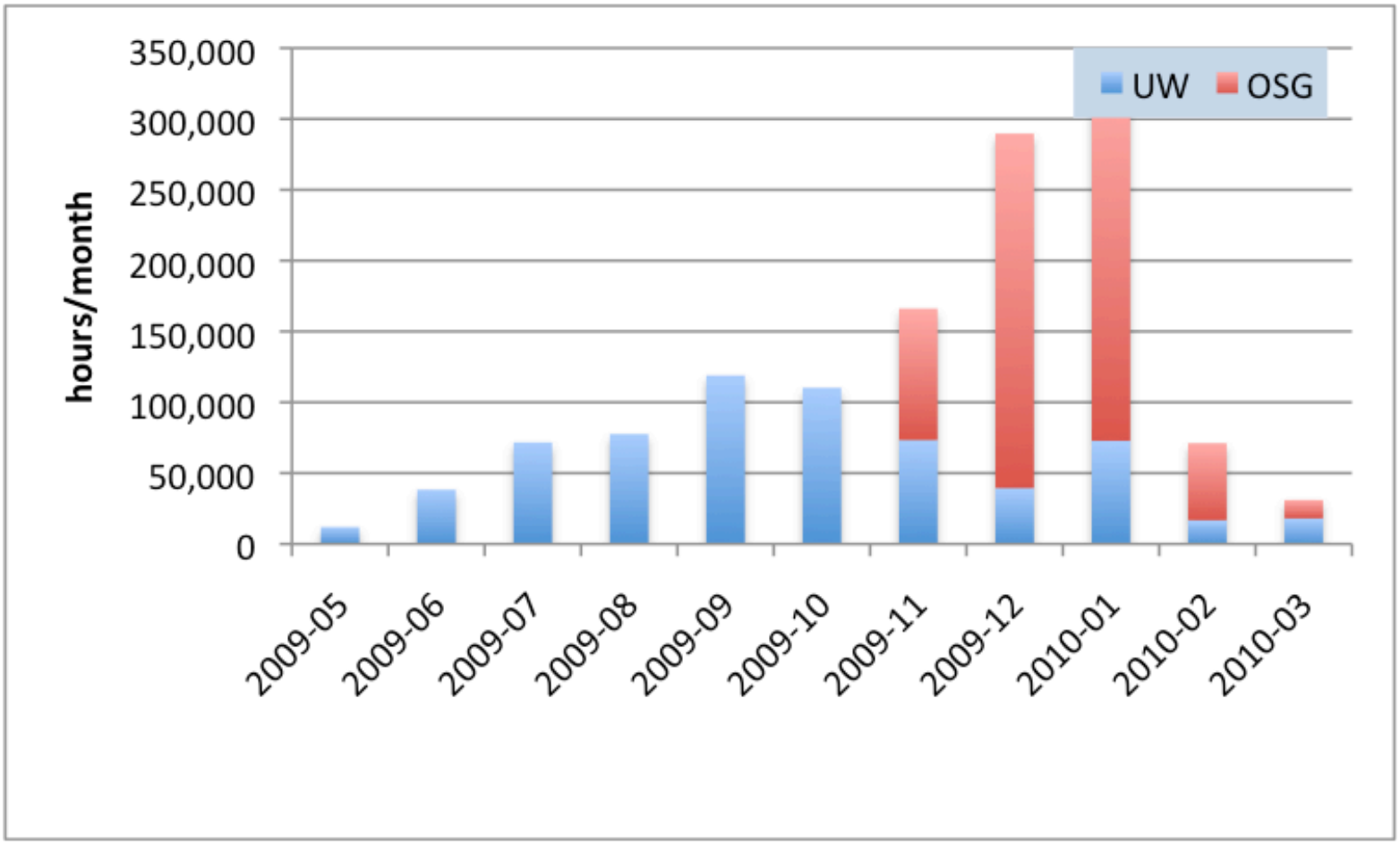
JobRouting MPI to OSG



Restrictions of job-routing

- More diverse hardware resources
  - No prestaged, some AMD…
- Must specify output file
  - transfer_output_files = outputs
- (touch outputs at job startup)

Computational Results



Real Results

- Iron-Catalyzed Oxidation Intermediates Captured in A DNA Repair Monooxygenase, C. Yi, G. Jia, G. Hou, Q. Dai, G. Zheng, X. Jian, C. G. Yang, Q. Cui, and C. He, {\it Science}, Submitted

Real Results

Disruption and formation of surface salt bridges are coupled to DNA binding in integration host factor (IHF): acomputational analysis, L. Ma, M. T. Record, Jr., N. Sundlass, R. T. Raines and Q. Cui, {\it J. Mol. Biol.}, Submitted

Real Results

- An implicit solvent model for SCC-DFTB with Charge-Dependent Radii, G. Hou, X. Zhu and Q. Cui, {\it J. Chem. Theo. Comp.}, Submitted
-

Real Results

- Sequence-dependent interaction of β-peptides with membranes, J. Mondal, X. Zhu, Q. Cui and A. Yethiraj, {\it J. Am. Chem. Soc.}, Submitted

-

Real Results

- A new coarse-grained model for water: The importance of electrostatic interactions, Z. Wu, Q. Cui and A. Yethiraj, {\it J. Phys. Chem. B} Submitted

-

Real Results

- How does bone sialoprotein promote the nucleation of hydroxyapatite? A molecular dynamics study using model peptides of different conformations, Y. Yang, Q. Cui, and N. Sahai, {\it Langmuir}, Submitted

-

Real Results

- Preferential interactions between small solutes and the protein backbone: A computational analysis, L. Ma, L. Pegram, M. T. Record, Jr., Q. Cui, {\it Biochem.}, 49, 1954-1962 (2010)

Real Results

- Establishing effective simulation protocols for $ \beta$- and$\alpha/\beta$-peptides. III. Molecular Mechanical (MM) model for a non-cyclic β-residue, X. Zhu, P. K\"onig, M. Hoffman, A. Yethiraj and Q. Cui, {\it J. Comp. Chem.}, In press (DOI: 10.1002/jcc.21493)

-

Real Results

- Curvature Generation and Pressure Profile in Membrane with lysolipids: Insights from coarse-grained simulations, J. Yoo and Q. Cui, {\it Biophys. J.} 97, 2267-2276 (2009)

Summary

- CHTC supports whole machine jobs
- Consider “small mpi”, or
  - High Throughput of High performance
- See Dan & Greg’s talk on Wednesday

Conclusion

- Computation enables better research.
- There are many opportunities to run jobs on campus.
- The CHTC is here to help you:
  - Get access to campus resources
  - Get access to campus expertise
  - Leverage computing to do better science!

Thank you! Email us at chtc@cs.wisc.edu