Workshop 3 Summary:

Workshop 3 included a high level look at high performance computing in universities, along with key questions that should be addressed at the institutional level. This was followed by several presentations that laid out the positives and negatives of maintaining your own system; the kinds of work that is done at relatively small institutions in chemistry, physics, and computational fluid dynamics; the value of collaborations and consortia; and support for computational science.

Key points:

- Let the question of what work our faculty want to do drive the decision about what hardware and associated support we set up.
- Breakthroughs can happen at any level of computing power, as long as the computing power is a good match for the problem being addressed (with respect to bandwidth, memory space, and speed).
- It is possible to build and manage your own cluster resource, but it brings with it a large systems administration burden that takes away from doing the research that led you to want the computer in the first place.
- Institutions have to really be committed to this direction before they start investing in cyberinfrastructure.

Workshop 3 Overview:

Workshop 3, held on October 27, 2009, explored applications of high performance computing in science and engineering disciplines. It included the following presentations (most are available at http://cs.union.edu/~barrv/Grants/Teagle/teagle-overview.html#W3):

- Fran Berman, Rensselaer Polytechnic Institute, discussed applications of HPC and cyberinfrastructure within the academic setting, and raised questions that she suggests should be addressed by any institution considering local cyberinfrastructure capability.
- Carol Parish, University of Richmond, discussed her experience setting up and maintaining her own cluster computers as well as her experience as part of the MERCURY consortium.
- Janine Shertzer, The College of the Holy Cross, discussed her experience with national external computing resources, computing resources at schools of collaborators, computing resources that she maintained at Holy Cross.
- Ray LeBeau, University of Kentucky, discussed his experience with a series of clusters that he has built and maintained locally.
- Bob Panoff, Shodor Foundation, discussed educational opportunities across the 9-16 spectrum, including undergraduate petascale education program and materials for
incorporating HPC into various science curricula.

Fran Berman, Rensselaer Polytechnic Institute:

Berman started with a perspective on what people are doing now with regard to large hard problems, like what the Milky Way looks like. Science that could never have been done 100, 50, or even 20 years ago is being facilitated by new tools and technologies.

These tools and technologies are also leading to changes in 21st century education, such as the EPICS program at Purdue and the UCSD Teacher Tech & Science program.

Berman defined cyberinfrastructure (CI) as “all the tech stuff you use, glued together to enable research and education.”

Rather than ask the question “What kind of CI do I need?”, have to ask the question “What kind of work do I want to do?” and let the answer to that drive the CI decision.

Berman set out an old pyramid of computing power, ranging from high end workstations at the bottom to teraflop machines at the top, pointing out that today the lowest level is achieved by desktop computers providing 10s of gigaflops; the mid-range level is now achieved by small clusters providing 100s of gigaflops; computing center supercomputers are providing 1000s of gigaflops; and leadership class machines are no longer in the teraflop range, but are now providing petaflops. She argued that research breakthroughs can take place at any level of this computing power pyramid, not just at the top level.

Berman also addressed the issue of “bragging rights”, noting that in order to sustain bragging rights for equipment, you have to replace it often, requiring constant infusion of new money. But you can have bragging rights for what you do with the equipment, since you can still make good use of a resource that is at the bottom of the list. Also important to remember that the top 500 list (www.top500.org) is based on one benchmark – not all problems will be solved in a way that aligns well with that benchmark. She also pointed out that large cool equipment is available to the U.S. academic community, particularly installations funded by U.S. research and educational agencies.

RPI sees its supercomputer as a regional and national resource. Only 50% of users are from RPI. Does not yet but may start to charge for cycles. Cyberinfrastructure is not free!

Challenges in building and delivering cyberinfrastructure:

- the “ilities” → scalability, interoperability, reliability, capability, etc.
- data storage
- physical infrastructure
- support models may vary depending on whether your focus is computing CI (requires series of one-time funding) or data CI (have to have consistency)

How much CI should institutions support locally?

- what does the user community need?
- how will the facility support the mission and strategic goals of the institution
- what is cost effective to support?
• is there budget for the continuing costs for maintenance, operation, management of the facility?
• is there enough space, power, bandwidth, data storage to support your usage mode
• can you recruit and retain expert staff to administer your facility
• how much equipment refresh will be necessary over time, and are there resources to do that?
• what business model should you use to run your facility?

Suggestions:
• develop campus people who can help your faculty use national facilities, rather than rely on the people who are at the national facilities
• change local metrics of success since you cannot measure the success of local support people the same way you would other faculty members.
• what about consortium level people?
• consider condo-cluster model which is great training for undergraduates and could work for a consortium
• but would have to explore issues of scheduling
• condo cluster not as hard to administer as a large supercomputer.
• look at curriculum – what level of computing is required to do an outstanding job in the computational part of the curriculum?

Carol Parish, University of Richmond:

Parish has managed a number of cluster computers at the University of Richmond as well as been a founding member of the Mercury Consortium (Molecular Education and Research Consortium in Undergraduate computational chemistRY).

General issues about using cluster computers:
• tradeoff between the number of nodes and the speed of communication between nodes (need high speed interconnects)
• as hardware changes, have to make sure that all packages will continue to be compatible with the new hardware
• freeware is often harder to use than purchased software, and more time has to be spent on student training to use freeware. But if students stay for multiple years then you get more mileage out of the training.

Access to computing:
• need successful grant proposals to get money to get equipment, and heavily used equipment has to be replaced within 3-4 years.
• Have to have enough equipment that nobody is trying to steal cycles from anyone else.

Important to take the systems administration role away from the scientists so that they can actually do their “real” work. Over time Parish dealt with many things that would better be handled by an administrator:
• tracking the latest in hardware developments
• hardware-software configuration and compatibility
• vendor interaction
• cost/performance concerns
• security, backup, OS, power, cooling, space/footprint, noise
• user accounts
• trouble shooting failures

Have to have good IT people and trust them to worry about installation, footprint, etc. Need real Linux support in ITS, with people who can provide the HW/SW support.

Institutional buy in is key – there must be a long term commitment to maintain, repair, replace, provide technical help.

Parish commented on TeraGrid, pointing out that in her experience it was hard to use, different sites on the grid have different log on protocols, interact with algorithms differently. this made it especially difficult for undergraduates to use. Berman pointed out that there are efforts to standardize the access for exactly the reasons Parish raised.

Parish also pointed out the best aspects of the Mercury Consortium: interaction with other researchers, the fact that there is a system administrator, and the annual meetings brought together faculty as well as the students who were also working with faculty on their projects.

**Janine Shertzer, The College of the Holy Cross:**

Using a regional or national resource:
• gives access to fastest/largest machines, with lots of hardware and software support, tech staff, workshops
• but, long queues, even for short jobs; have to apply for CPU time; can't afford to make big mistakes which cost lots of CPU time.

Borrowing resources from other universities:
• access to large/fast machines with HW/SW support; no need to write grant proposals
• but collaborations require common interests; can't get accounts for your students; usually it's just a temporary arrangement.

Having local resources:
• have all the access to the equipment that you want; no need to write more proposals once you've gotten the machine, yo have unlimited time on it, no queues, can easily set up student accounts
• but need money to buy the machine and maintain it and replace it
• and system support is critical

Shertzer raised the possibility of a supercomputer center for liberal arts colleges:
• internal review process for faculty and student accounts
• three year upgrade cycle
• HW and SW support
• but there are questions – who would host it? who would fund the staff and the computers? what sort of governance structure should there be?

How can those of us at liberal arts colleges effectively pool our resources?
Ray LeBeau, University of Kentucky:

LeBeau's main message was that it is possible to “embrace the idea of do it yourself computing”

Motivation to build your own cluster:
- stay within price range
- can do it for $500/node, $300 - $2000 to put the network together; $1000 for server node; student assembly for the cost of pizza; free software
- So can put together a cluster, customized for the software you want to run, for $10K-$25K
- Have total access and control
- No queueing, put students on, can test grid resolution by running the same job multiple times
- can use home grown system to test methodologies before applying for use of more standard shared super computing facilities
- Allows you to look at code performance (different code, different hardware), chance to “doubly optimize” your system.

Procedure to construct a cluster:
- determine funds, requirements, location, A/C, power, likely components
- build test nodes
- order components – all at once so you get exactly what you want and everything is the same
- organize build day
- build day for nodes
- connect nodes
- load cluster software
- burn-in

Challenges:
- suitable location, A/C, power
- freeing up $ for component purchases
- purchasing flexibility
- inventory
- construction time
- hardware compatibility issues

Software challenges:
- compatibility of Linux, MPI, hardware
- closed source code issues

Management Challenges:
- minimize idle nodes
- in his case they use no queuing, so users have to communicate with each other, have fewer users at a time
- can overload server access if there are too many jobs running at once
• bottle neck of communication into cluster from outside

Life Cycle Challenges:
• years 1 & 2 – okay
• year 3 – power supplies and fans start to go
• year 4+ – move into obsolescence, significant node loss
• but can maintain >50% of nodes beyond 5 years if do basic repairs/replacements
• 2-4 person weeks/year average maintenance
• around 6 years – new single computer = ½ the power of the cluster → then it's time to replace the cluster.

Commodity clusters can serve as gateway to shared supercomputers, intro to parallel computing.

Bob Panoff, Shodor Foundation:

Panoff discussed the many ways in which Shodor Foundation is involved in promoting computational science throughout the educational system.

National Computational Science Institute, undergraduate petascale education program.

Materials for building HPC into Biology and Chemistry curricula

Talked about the importance of teaching computing (not teaching programming). Push students to look at visual results, see what they can glean without programming, motivate programming by the desire to get more/better/different results from data.

First piece of computational thinking is quantitative reasoning: fractions, ratios, percents, decimals. Second piece is analogy, analogic thinking.

When bringing computation into the picture, have to ask – are we interested in doing
• more work?
• same work in less time?
• better work?

Ask ourselves as a community: what's the science that we're not doing in the undergraduate curriculum that we will be able to do when desktop computers are 1000 times faster?

Shodor supports curriculum development and internships with NSF $$.