Y | CRC

Cyberinfrastructure User Support

Andrew Sherman

Yale University

Senior Research Scientist, Yale Center for Research Computing

Senior Research Scientist, Department of Computer Science

ACI-REF Virtual Residency 2016
Thu August 11, 2016

Goals for this session

- What is CI, and how does it differ from conventional IT?
- CI user categories, and how to support them
- Some of the human aspects of CI support (i.e. politics, conflicts)
- Policies, education, outreach, collaborations, and networking

These slides are based on material from Mehmet (Memo) Belgin (GA Tech), modified by Henry Neeman, and are used with permission. Numerous edits have been made.

Yale Center for Research Computing

- Free-standing center reporting to Deputy Provost for Research (dotted lines to the medical school and ITS); created in July 2015
- Who we are (~15 FTEs)
  - 2 Faculty Directors (Arts & Sciences; Medical School)
  - Executive Director
  - ACI-REFs (6+): 2 research faculty; 5+ others; aligned to specific clusters
  - HPC Engineering/System Administration Team (6)
  - Director of Research Services (education, communications)
- Who we aren't (ITS)
  - Desktop or Lab Support
  - Campus Network Operations (Science Network & DMZ is shared)
  - Data Center Operations (power, etc.)
  - Security & Authentication Services

YCRC Responsibilities

Cyberinfrastructure

- 5 HPC clusters (~17K cores??)
- HPC data storage (~8 PB)
- Research data management
  - Integration with campus-wide “Storage@Yale” active & archive tiers
  - Some integration with lab and instrumentation storage
- Science Network & DMZ

Research & Teaching Support

- Dedicated support (YCGA, G&G)
- HPC software & algorithm installations, tuning & consultation
- Support for science & engineering software applications
- National infrastructure assistance
- Grant preparation
- Faculty recruitment (startup pkgs)
- HPC support for classes

Education & Training

- Parallel Computing (credit class)
- Research Computing Workshops
  - Getting Started Bootcamps
  - Python, Parallel R, GIS
  - Group/Dept. Bootcamps
  - XSEDE & vendor workshops
- User groups

Outside Community

- CASC (http://www.casc.org)
  - Working groups on ”beyond hardware” and regulated data
- XSEDE Campus Champions (2)
- ACI-REF (CaRC); ACI-REF-VR
- Northeast BigData Hub
- LCI

What the Heck is CyberInfrastructure (CI), Anyway?

- **Components**
  - **Computing systems**
  - **Data storage systems**
  - **Advanced instruments and data repositories**
  - **Visualization environments**
  - **High Speed Networks**
  - **People**
- **Purpose**
  - Enable scholarly innovation and discoveries not otherwise possible

Based on Indiana University's definition

Differences between CI and Conventional IT

- Primary target is **performance**
- Usually relies on conventional IT services (by a separate team)
- More focus on supporting end-users than services
- Uses common IT technologies in uncommon ways
- May mix shared and dedicated resources in one entity
- Requires specific middleware and software layers
- Requires code compilations using complicated mechanisms
- May require specific knowledge about the application/science
- Has irregular usage patterns, which may become obvious and troubling to users

Outline

- **Part I: CI user expectations, categorization and commonalities**
- Part II: Policies, Politics, Conflicts and Personality Management
- Part III: Education, Outreach, and Networking

Faculty (a/k/a Principal Investigator) Expectations

- Typical Roles
  - Research entrepreneur & teacher
  - Manager and funder of CI users
    - Often knowledgeable about CI but doesn't use it directly (that pleasure is reserved for students & postdocs!)
    - May own or pay for resources and services (but shared resources may be free at some institutions)
- Expectations:
  - CI resources are reliably up and running on 7x24 basis
  - Students and collaborators have fair (?) access to CI resources required to carry out research or classroom assignments on time
  - Assistance available as and when needed
  - Regular usage and expense reports (especially for storage)

“Actual CI User” Expectations

- Typical Roles
  - Some “hands on” faculty
  - Usually students, postdocs, or others who are not permanent
  - Permanent research staff or research faculty
  - External collaborators
- Expectations
  - 7x24 access to CI resources (and short job wait times, of course)
  - “Insider” relationship to CI staff for advanced users
  - Ultra-fast learning curve
  - Simple and instant solutions to complex problems
  - Applications running much faster than on their desktops (not always possible!)
  - Help diagnosing/fixing problems that may be externally controlled
  - Answers that match their level of knowledge

CI User Categories

- Three broad categories:
  - Novice
  - Intermediate
  - Advanced
- Difficult to identify a user's category without any prior interaction
- The language used in requests is a good indicator
- Replies to follow-up questions also reveal the level of proficiency
- If uncertain, assume “novice” (*but don’t make it obvious*!)

Category 1: Novice Users

- Characteristics
  - Little experience with Linux or command-line environments
  - May use Matlab, Mathematica, and sometimes R (or even Excel)
  - May have limited knowledge of a scripting language like Python
  - Rarely any inkling about parallelism
- Generate up to 40-50% of support requests. Common examples:
  - Desktop setup (especially for Windows)
  - Login procedures (ssh keys, two-factor authentication, etc.)
  - Finding software on the cluster(s)
  - Finding help and documentation
- Most requests are straightforward, but some “simple-sounding” ones may take a lot of work (or be impossible)

Support Activities for Novice Users

- Up-to-date website with reasonable documentation for novices
- Getting-started presentation or on-line tutorial (possibly customized for the user's desktop OS)
- Linux 101 workshop with software suggestions (e.g., easy editor)
- Friendly ticket system for requests, questions, and assistance
- Walk-in office hours
- Make it easy to find software, manage environment & run jobs
  - Tools like Lmod
  - Cross-cluster standardization of environment, job scheduler, etc.
  - Provide annotated template submission scripts
- Software installation assistance
- Help with tools to move data to/from clusters

Category 2: Intermediate Users

- Characteristics
  - Have prior Linux cluster experience; can create job scripts, but may not understand system-wide impact of their actions
  - Varying degrees of proficiency in Python, C, Fortran, R, etc.
  - Use workflows involving multiple domain-specific packages
  - Often notice and report HW or system problems
  - May use web search to try to overcome difficulties
- Generate up to 30-40% of support requests. Common examples:
  - Assistance with complex software installations
  - Assistance with performance issues
  - Help with complex job scripts, job arrays, or parameter studies
  - Special requests (“bending the rules”), such as job priority or quota

Effective Support for Intermediate Users

- “Teach them to fish”: Offer advanced, possibly domain-specific, workshops; take advantage of XSEDE or vendor offerings; Software Carpentry or Data Carpentry may be valuable for some users
- Build strong individual working relationships since these users often serve as local trainers & “experts” for their groups.
- Be transparent in discussions, since they can distinguish fact from speculation (and will probably put your advice to the test).
- Admit when you don’t know something. *You aren’t expected to know everything*! But then try to find out and follow up! (Network!)
- Help them find solid, high-quality on-line information (vendor sites, user forums, etc.) pitched at the proper level.
- Assist or do complex software installations, especially those involving parallel codes or significant optimizations. Help with code development/debugging/tuning may pay big dividends later.

Category 3: Advanced Users

- Characteristics
  - May be hands-on faculty, research staff, or advanced students
  - Experience with and access to multiple clusters (including XSEDE, etc.)
  - Technically proficient in scripting or programming languages
  - Develop and/or use parallel applications
  - Develop complex workflows and job scripts
  - Always trying new things; willing to experiment with new software
- Generate up to 10-15% of support requests. Common examples:
  - Installation of complex software & tools ("It's just 1 Python module!")
  - Requests bordering on R&D
  - Special requests/treatment (often outside of normal channels)
  - Help with special hardware (e.g., GPUs)
  - Bugs found in hardware, 3rd party applications, or libraries

Effective Support for Advanced Users

- Apply all support techniques for intermediate users here, too.
- Communicate and meet regularly with them. Happy advanced users and their faculty advisors/PIs may often be your strongest advocates at your institution.
- Treat advanced users as peers; they may know as much or more than you do about research computing.
- As appropriate, involve them in hardware acquisitions and ACI grant proposals.
- Collaborate! Resolving many of the complex problems they encounter may require close cooperation among ACI-REFs, system administrators, and others.
- Be flexible. Make small rules exceptions when they won't impact others. However, watch out for slippery slopes.

Outline

- Part   I: CI user expectations, categorization and commonalities
- **Part II: Policies, Politics, Conflicts and Personality Management**
- Part III: Education, Outreach, and Networking

Policies

- Have well-defined written policies. These set everyone’s expectations and avoid misunderstandings.
- Publish policies in places easy to find (online). Require PIs to accept your policies and make PIs responsible for the behavior of their students, postdocs, and staff.
- Be prepared to explain the reasoning behind each policy item.
- Make policies strict (conservative), but consider exceptions as needed (but avoid slippery slopes!)
- Encourage users to openly discuss and criticize the policies.
- Don’t hesitate to update policies to stay relevant.
- Build trust and effective communication with decision makers.
- Seek delegation privileges to speed things up.
- Influence, but don’t make, policies for resources you don’t own.

Scheduled Maintenance

- Set regular schedule, with multiple advance announcements.
- Unscheduled downtimes are no excuse for skipping maintenance
- Provide a summary of completed tasks after maintenance.
- Have clear goals; plan ahead in great detail:
  - Work with your vendors
  - Team member / task associations
  - Estimated task duration
  - Critical paths and fallback plans
- Prepare for potential problems during/after maintenance days
- Show best effort for minimal impact
  - Configure the scheduler to have no running jobs
  - Disable user access to resources during the maintenance activities
  - Assist users in moving work to alternative clusters when possible

Politics and Conflicts

- Tricky but inevitable
- No magic formula, need case-specific creative solutions
- Biggest challenge: conflicts due to limited resources
  - Configure systems to match your policies.
  - Collect and store data for past and present usage.
  - Provide users with tools to browse data/statistics for their accounts.
  - Run regular audits to defuse problems before they explode.
  - Consider a scavenge queue for pre-emptible jobs

Tiers of Conflict

- Internal to a group/department: Usually easier to solve with communication and informal agreements. Sometimes a good job scheduler can help (e.g., multi-level fairshare). Provide advice, but get the PI or chair to take the lead and own the resolution.
- Between groups/departments: Can get messy, but may be avoidable if you stick to your policies. Be even-handed; don’t show favoritism. Get all agreements in writing!
- Between users and CI support staff: Have clear policies handy as a basis for declining unreasonable or impossible requests, and keep solid statistics/data as evidence. As above, be even-handed; don’t show favoritism. Get all agreements in writing!

Personality Management

- Some users are more difficult than others. That’s life!
- Don’t take things personally; report harassment; never retaliate
- Users don’t mean to be difficult; but may be under great pressure and extremely frustrated
- If you make a mistake, take responsibility and offer an apology.
- Show empathy and sincerity
- Acknowledge that:
  - you understand the user’s concerns;
  - you are aware of its particular impact on the user.
- Be sensitive to cultural differences and language difficulties.
- Use humor appropriately, and avoid being awkward or insulting.
- Communicate frequently while working on any issue

Outline

- Part I: CI user expectations, categorization and commonalities
- Part II: Policies, Politics, Conflicts and Personality Management
- **Part III: Education, Outreach, and Networking**

Trainings and Tutorials

- Research Computing Workshops
  - Getting Started Bootcamps
  - Python, Parallel R, GIS
  - Group/Dept. Bootcamps
  - XSEDE & vendor workshops
  - Software Carpentry; Data Carpentry; SC Tutorials & Workshops
- Special Topics
  - Parallel Computing
  - Debugging/optimization of codes (including parallel)
  - System architecture specific details
  - Advanced use of common tools (Scientific Python, Parallel MATLAB)

Group Consultations

- Mini-orientations for new groups (“On-Boarding”)
- Use group meetings for feedback & to resolve internal conflicts
- Resolution of technical problems that are specific to a group
- Technical feedback to assist in policy making and system purchases
- Introduce services to new groups interested in getting resources

Collaborations with Researchers and Vendors

- Researchers helping researchers
- Crucial for staying relevant: What is your faculty planning?
- Collaborative grant writing
- Collaborative projects/papers (acknowledgements or co-authors)
- Support for classes and workshops
- Developer/vendor collaborations
  - Bug tracking and fixes
  - HW/SW information, evaluation of new systems and technology
  - Pilot studies & benchmarks

Some External Groups for Staff Training & Networking

- ACI-REF; ACI-REF-VR; CaRC
- XSEDE Campus Champions (national & regional)
- CASC (http://www.casc.org)
  - Working groups on ”beyond hardware” and regulated data
- Educause
- LCI (aimed at HPC system administration)

THANKS FOR YOUR ATTENTION!

QUESTIONS?

ANDREW.SHERMAN@YALE.EDU