MICHIGAN STATE UNIVERSITY

Cyberinfrastructure User Support

Dr. Dirk Colbry

Director of HPC Studies

Department of Computational Math, Science and Engineering

Michigan State University

ACI-REF Virtual Residency

Thursday August 3, 2017

SPARTANS WILL.

CMSE

Successful User Support

1. Solve Problems
2. Build Relationships

Technical Knowledge

Communication!

Goals for this session

- Who am I, and why am I leading this session?
- What is CI, and how does it differ from conventional IT?
- CI user expectations, categorization and commonalities
- Policies, politics, conflicts and personality management
- User support tools such as Education, Outreach, and Networking

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

Outline

- **Part I: Who am I?**
- Part II: Understanding CI users
- Part III: Tools for Interacting with Users

Institute for Cyber-Enabled Research

- Computing Resources
- User Training
- Research Support

ICER

iCER Hardware Resources

FREE*

- Large Memory Nodes (up to 6TB!)
- Lots of Medium memory nodes (128GB)
- GPU Accelerated cluster (K20, K80)
- PHI Accelerated cluster (5110p)
- Over 1000 nodes, 300 accelerators, 17,500 computing cores
- 2PB high speed parallel scratch file space
- 1.5PB replicated file spaces
- Access to large open-source software stack & specialized bioinformatics VMs

Computational Consultant

- XSEDE Campus Champion
- One-on-one consulting
  - Planning
  - Debugging
  - Training
  - Grant Writing
- Office Hours
- Ticket / Email Support
- Training
- Networking
- Communications / Outreach
- Software Development

XSEDE
Extreme Science and Engineering
Discovery Environment

CAMPUS
CHAMPIONS

software carpentry

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

SPARTANS WILL.

CMSE

Director of the HPCC (Administration)

- Hardware component of iCER
- Supervising HPC System Administrators
- Manage Yearly Hardware Budget
- Lots-o-meetings

Computational Mathematics Science and Engineering

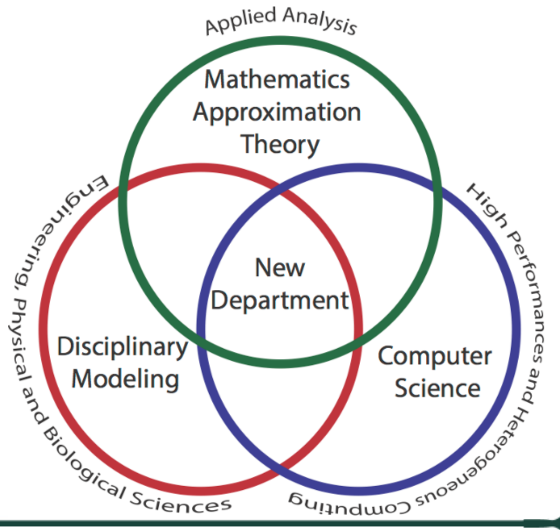
Computational science addresses the construction of mathematical models, quantitative analysis techniques and using computers to analyze and solve scientific problems.

Foundation:

Discipline leverages:

1. Application knowledge
2. Computer science
3. Mathematics

To develop new methods for investigating complex problems through computation.



CMSE Est. 2015

- Designed to be interdisciplinary
  - Joint College administration (Natural Science, Engineering)
  - Most faculty have joint appointments, spanning campus
  - Incentives for cross-discipline and cross-college research collaborations
- Growing rapidly
  - Targeting 25-30 FTEs, including lots of **new hires**
  - Research foci: data science, large-scale, HPC

Director of HPC Studies in CMSE

- Help build new department
- Curriculum development for graduate and undergraduate classes
  - CMSE 201/801: Computational Modeling
  - CMSE 202/802: Tools for Computational Modeling
  - Numerical Linear Algebra
  - GPU Program
  - Large scale programming
  - Scientific Image Analysis
- Oh yeah, and I get to work with researchers again

Outline

- Part I: Who am I?
- **Part II: Understanding CI users**
- Part III: Tools for Interacting with Users

What is CyberInfrastructure (CI)?

- Components
  - Computing systems
  - Data storage systems
  - Advanced instruments and data repositories
  - Visualization environments
  - High Speed Networks
  - People

Where do we see CI Hardware?

- Anything more advanced than your desktop
- Local/Regional resources
  - Lab, Department, Institution (iCER)
- National resources
  - NSF (XSEDE, Blue Waters), DOE, many others
- Commercial Resources (cloud computing)
  - Amazon, Azure, Liquid Web, Others

Why do researchers use CI hardware?

- Science takes too long
- Computation runs out of memory
- Need licensed software
- Need advanced interface (visualization/database)
- Lots of file i/o

Purpose of CI

- NOT always maximizing Kflops:
  - Floating Point Operations per Second
- Instead, the goal of CI IS:
  - KSciences / second*
- Doing More Science, Faster
  - Reducing the "Mean time to Science"*
- Enable scholarly innovation and discoveries not otherwise possible

* Not my terms, not sure where it's stolen from…thank you!

Who are our users?

- Institutional Administrators
- Faculty (Principal Investigators)
- Users (Students, Post-Docs, etc.)

Faculty Roles

- Researcher, entrepreneur, teacher
- Manager and funder of CI users
  - Often knowledgeable about CI
  - Often does not use CI 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)

Faculty Expectations

- Reliable CI resources, available 24x7
- Students and collaborators have fair (?) access to CI resources needed to meet deadlines on time
- Assistance available as and when needed
- Regular usage and expense reports (especially for storage)

“Actual CI User” Roles

- Some “hands on” faculty
- Usually students, postdocs, or others who are not permanent
- Permanent research staff or research faculty
- External collaborators

“Actual CI User” Expectations

- 24x7 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 run much faster than on desktops (not always possible!)
- Help diagnosing/fixing problems that may be externally controlled
- Answers that match their level of knowledge

Institution Administration

- Typical Roles
  - Funder and supporter
  - Does not use the HPC
  - Doesn’t know how to use the HPC
- Expectations:
  - Enables more science (i.e., more journal papers)
  - Brings in funding (i.e., more grants) to the Institution
  - Low cost to Institution (i.e., wants total cost recovery)
  - Brings prestige to university (i.e., dog and pony show)

CI User Ability

- Three broad categories:

  Novice ---- Intermediate ---- Advanced
- Difficult to identify a user's category without prior interaction
- The language used in requests is a good indicator
- Replies to follow-up questions also reveal proficiency level
- 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

Category 1: Novice Users

- 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)

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

Category 2: Intermediate Users

- 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

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

Category 3: Advanced Users

- Generate up to 10-15% of support requests. Common examples:
  - Installation of complex software & tools
  - 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

Outline

- Part I: Who am I?
- Part II: Understanding CI users
- **Part III: Tools for Interacting with Users**

Successful User Support

1. Solve Problems
2. Build Relationships

Successful Communication

- Website and on-line documentations
- On-line Video Tutorials
- User Ticketing System
- In-person tutorials and training workshops
- One-on-one Meeting
- User Communities

This is where many relationships start and end!

Making Policies

- Make policies strict (conservative), but consider exceptions as needed (but avoid slippery slopes!)
- Be prepared to explain the reasoning behind each policy item
- 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

Communicating Policies

- Have well-defined written policies
- 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

Ticket/Email Systems

- Ticket systems ensure emails do not get lost
- Ticket systems build a history of your relationship
- Build Trust
  - Follow up with all interactions
  - If you make a mistake, take responsibility and offer an apology
- Closed Tickets are not a measure of success

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
  - Leave time for Testing

Scheduled Maintenance

- 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
- **Test**, **Test**, and do more **Testing** before turning scheduler back on

Conflict Management

- Tricky but inevitable
- **No magic formula,** need case-specific creative solutions
  - Don’t take things personally; report harassment; never retaliate
- Some users are more difficult than others. That’s life!
  - Users don’t mean to be difficult, but may be under great pressure and extremely frustrated

Biggest Conflict Management Challenge

- Conflicts due to limited resources
  - Configure systems to match your policies
  - Collect and store data for past and present usage
  - Provide user tools to see data/statistics for their accounts
  - Run regular audits to defuse problems before they explode
  - Consider a scavenge queue for pre-emptible jobs

Personality Management

- When emotion enters the discussion
  - Stop your current train of thought
  - Show empathy and sincerity
  - Look for the underlying cause of emotion
  - Ask questions to get to the cause
  - Check to see if you have found the cause
- Be sensitive to cultural differences and language difficulties
- **Communicate** frequently while working on any issue

Trainings and Tutorials

- Online videos
- Frequent Hands on Training for Introductory Topics
  - Linux 101, Scheduler 101, etc.
- Focus on your area of expertise
- Leverage National Resources
  - XSEDE, Blue Waters, Software/Data Carpentry, Vendor workshops

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

Thanks for your attention!

Questions?

colbrydi@msu.edu