

Education, Outreach, Training

Discussion Lead: George Karypis

Scribe:

Participants:

- Richard Brower
- Dennis Castleberry
- John Cazes
- Jodi Hadden
- Eric Heien
- Chathura Herath
- Gideon Juve
- Murat Keceli
- Suresh Marru
- David McQueen
- Manish Parashar
- Ivan Roderio
- Jay Roloff
- Shawn Shadden
- Ross Walker
- Julie Wernert
- Vineet Yadav

Questions:

Education/Outreach/Training

- Two types of EOT:
 - EOT that you give: what could you give?
 - EOT that you get: what would you like to receive?
 - Whatever we decide should be basic skills training, we should implement it as early as possible. Two reasons: first, experience with the software can be gained over years; secondly, and more importantly, is the non-use of crutch software (the use of which only facilitates further use). People tend to be biased toward the software they use initially, which ideally should be the software they wind up using in their careers. (dc)
- What are our projects doing for project-specific training? Two useful templates discussed this morning:
 - Software carpentry classes
 - TACC "Defensive Program" course series teaches people basics like using the shell up to parallel programming and visualization tools.
 - Several groups had examples of internal discipline, having long-standing teams to train new project members, having a well-designed code base to show new developers how to do things.
- How do we go beyond the previous two approaches? Every university should have #1. How do you encourage #2?
 - Clear documentation of the software can eliminate the need for an oral tradition of software training. (dc)
- More training on SI2 software? Side effects: Force/encourage teams to package, document, design software (if these need improving), to fertilize multidisciplinary connections, to provide templates to other projects on how best to do things.
- How can we best connect to other training efforts?
 - To what end? To consolidate training efforts (is this necessarily a good thing?) or simply to learn how others are doing training? It depends on what we want to achieve. If we want to find out what the best training paradigms are, perhaps we could look at empirical studies on the topic as well, as we should not rely on anecdotal evidence alone. (dc)
- What constitutes effective connections to curricula for our projects?
 - Having training for our software/projects embedded in the curriculum, then covered either by us or someone intimately familiar with the software/project. Perhaps the best candidate to provide training for the software/project may be a faculty member and dedicated user of the software/project with social ties to the group. (dc)
- How can we succeed in bringing underrepresented groups into our efforts in as many roles as possible?
 - Be aware of *stereotype threat*. Underrepresented groups who are negatively stereotyped tend to conform to the stereotype because of performance anxiety associated with the stereotype. A classic example is women who are led to believe that they are genetically destined to perform less well in mathematics; the same holds of verbal skills for men. To combat this, we should try not to bring these stereotypes to mind when we recruit. (dc)
 - The answer could very well depend on the underrepresented group in question. Are we to target any in particular? To bring more women into the technical roles of a project may require one technique (eliminating stereotype threat, educating and being educated about the gender issues that surround women and science) while bringing Alaskan Natives may require broadcasting the message in a certain way. In general, I think individual appeal to the given group would have more impact than lumping all the target groups together in a single broadcast invitation (e.g. "calling all women, African Americans, Alaskan Natives, etc."). (dc)

Notes:

Can be what you get or what you receive

Amber: Workshops for users. Amber developers available to students

Pegasus: Conferences, Extensive manual (w/ tutorial), Mailing lists, have VM images that go with the tutorial to make it easy for new users to try the software

USC has a CS MS degree in High Performance Computing and Simulation that some PhD students in other departments (chemistry, physics, biology, engineering) take in order to acquire computation skills for their research. <http://www.cs.usc.edu/academics/mshighperform.html>

-Try to document questions and solutions that arise

John: Offers basics of coding, running, optimization, (short day long and week long in summer). Mainly intro to concepts. Two day each semester. 75% from UT--larger percentage for weeklong summer course. Hard to do webcast training. Most ppl have code on system that needs to be parallelized.

What type of training is given to sys admin for installing sci codes?

Sometimes developers don't know best way to config codes to run on systems.

Installing on virtual machines, part of the cloud.

Amazon offers training grants 4 times a year.

1). development thru Matlab--undergrads

2) develop in c--graduate students

3) developing own modules

Gabrielle: Scientific computing course: 5 instructors each 4 modules, brief surgery of part of field in 2 lectures to introduce to state of the art. 2 lectures to teach specific software or packages. Coursework: add a module. Basic skills lectures as well. Use NSF facilities.

Taught about linux and version control.

EOT: We don't look at others codes, so we don't learn from them. Need mechanisms for this.

Lots of tools for developing tools. More work should be done in comparing these tools.

New build up programming skills--easier for new departments.

Would it be possible to have other dept do service courses.

Easier to integrate classes at graduate level.

IBM class. Sessions on use of this code.

Titan: Workshops at own institution.

Software engineering should be requirement at graduate level in computational science.

Teach good habits earlier the better.

Can we pass material? Sometimes it's hard without using same hardware.

TAC: Remote users can get into.

XSEDE: Training for particular tools are available, but likely outdated. Would be nice to have.

SI,XSEDE,NSF: Website with examples.

Course experience: Assignment for GPU, scientific computing. There is a confidence barrier to have students overcome to do a "real project"

Graduate student trickle down knowledge.

How to evaluate training? Track the people over time and follow up.

What should be in data management plans for software? Data management plan can be driven by EOT.

Metadata formats for research paper. Potential use for training material?

Google is good for particular searches, but not for generic searches.

Everything in SI2 should be reusable. How can we maintain a list?

XSEDE has software list. Need full infrastructure (training, software, ratings, etc.)

There doesn't seem to be strong connection with SI2 to MSI.