ABSTRACT
A growing proportion of Science, Technology, Engineering & Mathematics (STEM) research is increasingly dependent on Cyberinfrastructure (CI). CI has experienced rapid progress in enabling technologies – hardware, storage, networking, middleware, tools, libraries – but much slower improvements in workforce development. Currently, CI consumers tend to lag substantially behind CI capabilities. This paper describes a series of linked efforts to address the gap between the workforce and the technology.

Categories and Subject Descriptors
K.3.2 [Computer and Information Science Education]: Computer Science education; D.1.3 [Concurrent Programming]: Parallel programming; F.1.2 [Modes of Computation]: Parallelism and concurrency; G.1.0 [General]: Parallel algorithms

General Terms
Education

Keywords
Parallel programming, cluster computing, computational science, scientific computing, education, curriculum, undergraduate faculty, graduate students, undergraduate students

1. INTRODUCTION
A growing proportion of Science, Technology, Engineering & Mathematics (STEM) research is increasingly dependent on Cyberinfrastructure (CI), which has experienced rapid progress in enabling technologies – hardware, storage, networking, middleware, tools, libraries – but much slower improvements in workforce development, with extant and emerging CI consumers lagging substantially behind CI capabilities. The dawn of the Petascale era has coincided with the rise of new technologies with vast improvements in computing performance – for example, the Graphics Processing Unit (GPU) as a computing engine (the recently released NVIDIA Tesla C2050 would have rivaled the fastest supercomputer in the world in June 1996 and roughly the 500th fastest in November 2003 [1]). But these emerging techniques and technologies require that practitioners acquire a substantial increase in CI sophistication to make effective use of these revolutionary capabilities.

The discipline that in principle is best positioned to teach High Performance Computing (HPC) topics, Computer Science (CS), while increasingly committed to teaching parallel computing because of the rise of multicore and many-core technologies, at many institutions lacks a reward structure for reaching the breadth of STEM disciplines that need this capability. In particular, many CS departments have neither a history of, nor a reward structure for, teaching service courses that are primarily aimed at non-majors, and as a result, parallel computing and related topics are typically taught as upper division undergraduate and/or graduate CS courses, with deep prerequisite chains that are appropriate primarily for CS majors and minors.

As a result, at many institutions – perhaps most – the need for both faculty and students to learn HPC and Computational Science & Engineering (CSE) must be met by alter-
native means. To address this growing need, the National Computational Science Institute (NCSI) has developed a tutorial workshop series, primarily presented as summertime week-long intensives, focusing on (a) Computational X for X Educators, where X is any of several STEM disciplines (e.g., Biology, Chemistry, Physics, Engineering), and (b) Parallel Programming & Cluster Computing (PPCC), which retains a focus on teaching educators, but broadens the target audience beyond the computing disciplines to all STEM disciplines (a typical PPCC workshop audience is roughly half from CS and half from other disciplines). Sponsored by a mix of NCSI, the SC Conference’s Education Program (for example, during SC07–SC09) and, beginning in 2010, the Blue Waters Undergraduate Petascale Education Program [9], along with a variety of co-sponsors, these tutorial workshops have served a broad audience across many STEM disciplines.

2. WORKSHOPS

Over the past 8 years, a series of these week-long tutorial workshops, typically about 10 per year with about 20 attendees each, have been offered at institutions across the United States and internationally. The faculty we serve are primarily educators; that is, they are learning the material for use in the classroom. The bulk of our participants are undergraduate faculty from STEM disciplines. We also engage some high school science teachers and have begun to work with faculty from the humanities, arts, and social sciences. Whenever possible, we encourage small teams, preferably interdisciplinary, to participate, with the goal of facilitating local support from their institution(s), which ultimately can lead to an increased likelihood of sustained curricular change.

A study of the NCSI tutorial workshops [11] concluded the following: (a) there is a lack of community coherence among faculty interested in CSE – they have interest but do not know where to find answers or to get tools – and (b) the community among which these efforts is most likely to have impact is the “community of learners”; that is, the faculty and students involved in introductory undergraduate education [15]. A key difference between CS faculty and domain STEM faculty is that CS faculty tend to be interested in designing parallel computing courses, whereas domain STEM faculty tend to be interested in incorporating HPC examples into undergraduate domain STEM courses. In each of the PPCC workshops, CSE is presented in the context of its use in STEM education. In a recent review, Yasar and Landau listed HPC, including the use of clusters and other parallel architectures, as a key component of any CSE program [17]. Our parallel computing workshops have been designed to address the role of parallel computing in HPC and CSE.

While there have been variations for one and two day presentations, the majority of our PPCC workshop experiences have been week-long intensive workshops, essentially identical to those we plan to leverage for the CI-TEAM project described below under Future Work. Each PPCC workshop consists of three main HPC informational areas: pedagogy, protocols, and parallel machines. This corresponds to the ACM/IEEE-SC Computing Curricula 2001’s [12] recommendations for cluster computing in the classroom: teaching System Architecture, Programming, and Algorithms and Applications [6].

In the pedagogy-focused components, participants engage in activities oriented towards the key concepts in parallel computing: speedup, efficiency, the interplay between communication and computation, the impact of the storage hierarchy on performance, and algorithms. The learner-centered activities have a “run one, modify one, build one” structure that allows participants to focus on becoming (in many cases) re-acquainted with Unix before becoming (also in many cases) re-acquainted with programming before actually having to produce real code. This approach of focusing on problems that participants solve in an increasing progression of skills and difficulty, using completed materials before learning how to create them from scratch, has been successfully applied in the past to the teaching of computer science [8]. In addition, we include activities to show how parallel algorithms and concepts could be illustrated metaphorically, using a series of group oriented “line dances” of various parallel algorithms, where participants act as processing elements, leveraging a kinesthetic component to enhance understanding and retention of parallel algorithm behaviors.

We primarily focus on using MPI to teach message passing protocols, given its predominance over other protocols (e.g., PVM, Linda and sockets) in STEM community application software, though we also spend a modest proportion of the contact time on OpenMP, which is widely available and commonly used with C, C++ and Fortran. We initially present MPI in terms of the six basic MPI routines: MPLInit, MPLFinalize, MPLComm_rank, MPLComm_size, MPLSend and MPLRecv. We separately discuss concepts of data reduction, collective communication, data decomposition, and profiling, as well as other MPI routines, sometimes with the full group, sometimes in optional breakout “mini-sessions”, which attendees choose, from among multiple options (listed below), based on their stated interests.

In addition to required sessions covering pedagogy, protocols, and parallel machines, all participants in the week-long workshops are required to complete a project, which each participant presents to the entire group at the end of the week, typically focusing on parallel code that the participant(s) created or modified, or on curriculum design for HPC and parallel computing education at their institution, or related topics.

2.1 Workshop Content

The general format of the workshops is to start on Sunday evening, with introductory presentations and a tour of the local HPC facility if available. During the week there are four 90 minute instructional blocks and an evening lab each day (except on Wednesdays, when the afternoon and evening are free time, to give participants a breather midway through the workshop). The workshops finish mid-day on Saturday. An exception is the Undergraduate Introduction to Petascale Computing, which is presented over a two week period but on Monday through Friday only, with weekends off (and no Wednesday afternoon break). The last day is devoted to participant presentations. Each day is a mix of lecture, open-ended lab exercises, and discussion.
We maintain a high instructor-to-participant ratio in order to facilitate the hands-on nature of the workshop materials. The evening labs are left open for the participants to work on lab exercises and projects with support and guidance from the instructors. Participants complete pre, daily, and post workshop surveys. The daily feedback is considered by the instructors as they are available, this facilitates adjusting the delivery or content a bit to accommodate the particulars of a given group. Example straw man agendas are below, but these are guidelines rather than the actual agendas used in practice, which tend to be fluid because of the need to respond to participants’ specific backgrounds and goals.

Introduction to Parallel Programming & Cluster Computing
- Day 1 - HPC Overview, Supercomputing center tour
- Day 2 - Time to Science, Cluster Computing Overview, Introduction to MPI, point-to-point communication
- Day 3 - Storage Hierarchy, Parallel Paradigms and Decompositions, MPI, collective communication, blocking and non-blocking communication
- Day 4 - Bootable Cluster CD [10], Parallel Debugging
- Day 5 - Scientific Libraries, OpenMP, Mini-sessions
- Day 6 - GPGPU, Mini-sessions
- Day 7 - Project Presentations, Where to go from here

Intermediate Parallel Programming & Cluster Computing
- Day 1 - Terascale, Petascale and Exascale Computing, Supercomputing center tour
- Day 2 - MPI review, OpenMP review
- Day 3 - Hybrid MPI/OpenMP programs
- Day 4 - Debugging Hybrid MPI/OpenMP programs
- Day 5 - Scientific Libraries, Large Datasets, Parallel I/O
- Day 6 - GPGPU Review, Hybrid MPI/GPGPU programs
- Day 7 - Project Presentations, Where to go from here

Undergraduate Introduction to Petascale Computing
- Day 1 - Introduction to Petascale Computing
- Day 2 - Introduction to OpenMP, Debugging OpenMP
- Day 3 - Introduction to MPI, Debugging MPI
- Day 4 - Intermediate MPI, Intermediate OpenMP
- Day 5 - GPGPU Programming, Parallel Debuggers
- Day 6 - Hybrid parallel programs; MPI/OpenMP and MPI/GPGPU
- Day 7 - Scientific Libraries for Petascale Computing
- Day 8 - Data Intensive Applications and Visualization
- Day 9 - Benchmarking and tuning
- Day 10 - Project Work and Presentations

Mini-sessions are a participant selected subset of a range of closely related topics. Partway through the workshop the participants’ interests are surveyed, and the most popular subset of the topics are then covered in mini-sessions later in the week. Since the instructors know the interests of the participants by this time, every effort is made to include topics that are germane to that particular group. These are optional sessions; participants can attend as many or as few as they would like, or instead choose to work on lab exercises or their project.

A typical workshop’s initial list might include the following:
- Building a Cluster: Software, Libraries, User Administration, Funding, Infrastructure
- LittleFe: A Portable Cluster for Teaching Parallel Programming & Cluster Computing, Outreach and Training
- Benchmarking and Tuning Parallel and Cluster Software
- Debugging Parallel Code (additional lab)
- MPI Lab: Practicing Send/Recv (additional lab)
- MPI Lab: Practicing Collective Communication (additional lab)
- Instruction Level Parallelism
- Stupid Compiler Tricks
- Multicore Madness
- High Throughput Computing (e.g., Condor)
- Scientific Libraries (additional lab)
- People Parallelism/Line Dancing
- Visualizing Science via the ScienceSim Metaverse
- Curriculum Resources – Computational Science Education Reference Desk (CSERD) [16], HPC University [7], etc.

Computational resources for the workshops come from a number of entities: the modest clusters of Edu-Grid (resources intended for educational use), the Bootable Cluster CD (see below), the large cluster at the OU Supercomputing Center for Education & Research (OSCER) at the University of Oklahoma (OU), and Kean University’s large cluster. As participants become more advanced in their usage and needs, they can graduate to TeraGrid resources, for example through the Campus Champions program. (Two of the authors of this paper are TeraGrid Campus Champions for their home institutions.)

3. CURRICULUM IDENTIFICATION AND DEVELOPMENT

The workshops and outreach activities use curricula from a variety of sources. Additionally, the instructors develop new materials as needed to fill in gaps and extend the reach of the communities’ parallel programming and HPC content. After they are tested and refined by use in the workshops, and a verification and validation pass has been performed on them, these modules are migrated into the community repositories listed below.

In order to fully engage workshop participants, the materials are developed to be:
- highly interactive, using guided-inquiry methods;
- scaffolded;
- relevant to contemporary problems;
- visualized where possible; and
- integrative across science disciplines as appropriate.
The value of using guided-inquiry methods in HPC and computational science is well documented [14]. The curriculum modules and software laboratories developed follow those principles whenever possible. One particular way we support this approach is through the use of “software scaffolding” [13]. As employed here, this technique suggests a framework which enables a workshop participant to first run a program, then open the hood to see how it is built, with an eye towards refining or extending it, ultimately learning enough to build their own.

The following are some of the key sources of curricular components that are used in the NCSI workshops.

3.1 Computational Science Education Reference Desk
The Shodor Foundation’s Computational Science Education Reference Desk (CSERD) [16] is one of the Pathways projects of the National Science Foundation’s National Science Digital Library (NSDL) [5]. CSERD is the premier national dissemination channel for quality reviewed materials for CSE education.

3.2 HPC University
HPC University (HPCU) [7], a collaboration among TeraGrid and a number of partner organizations, is a virtual organization focused on high-quality HPC learning and workforce development activities and resources. HPCU priorities are driven by community needs and requirements. Since 2007, HPCU has been collecting peer-reviewed metadata on curriculum modules, workshops, tutorials and other materials related to parallel programming and HPC.

3.3 Blue Waters Undergraduate Petascale Education Program
The Blue Waters Undergraduate Petascale Education Program (BW–UPEP) [9] is a collaborative effort among the National Center for Supercomputing Applications (NCSA) Blue Waters project, NCSI, and national HPC programs to prepare current and future generations of students with the computational thinking skills, the knowledge and the commitment that are needed for advancing scientific computing through the use of HPC resources and environments. Petascale computing requires a solid foundation of CSE, HPC, and related STEM domains. As such, the community must address the need for the engagement of a larger and more diverse workforce and ensure that CSE and HPC educational materials and practices evolve at the same rate as the science and technology do.

BW–UPEP is taking advantage of faculty expertise to establish best practices, identify and fill curriculum gaps, and modernize the CSE curriculum across all STEM fields. Through the program, faculty are given the opportunity to submit proposals for the development of curricular materials targeted to the computational aspect of their area of expertise. In addition, each of these curriculum development projects will often include proposals for internship positions, giving an eligible undergraduate student the opportunity to work within a Petascale research group and to attend annual science and technology events such as the ACM/IEEE-CS sponsored SC conference and an intensive 2 week workshop held at the NCSA facilities.

The curriculum materials development effort is being conducted in partnership with CSERD and HPCU, in order to broadly disseminate the developed materials through these established digital libraries. Once submitted, the materials will be subject to a formal verification, validation, and accreditation process, ensuring that (a) the correct problem is solved, (b) the problem is solved correctly, and (c) the problem/module is targeted to the appropriate educational level. During the development process, faculty will receive support in the form of access to computing systems, technical and curricular assistance, and a moderate stipend.

3.4 SC Education Program
The SC Conference’s Education Program is an ongoing program designed to teach primarily undergraduate faculty from a broad range of disciplines about HPC and computational methods, and to help them integrate HPC and computational techniques into their classrooms and research programs.

4. BOOTABLE CLUSTER CD
The Bootable Cluster CD (BCCD) [10] is a software platform designed to allow computational science and HPC instruction to concentrate on the education and not the computing infrastructure necessary for it. Originally developed by co-author Gray at the University of Northern Iowa, the BCCD continues to be developed by those close to CSE and HPC education. It is an open-source Knoppix-based [3] live CD that provides many of the tools necessary to teach CSE and HPC (e.g., GNU compiler suite, Intel compiler, multiple MPI implementations, CUDA, Java) that can be run on either a single system or on a network of computers that are automatically configured using custom networking tools to form an ad hoc temporary cluster. At a higher level, a number of curriculum modules suitable for high school and college students have been written around the BCCD, allowing instructors to benefit from a community of users interested in tackling the same problems.

The BCCD is also developed with LittleFe clusters [4] in mind. LittleFe is a complete multi-node portable cluster designed as an “educational appliance”, to reduce the friction associated with teaching parallel programming, cluster computing and computational science in a variety of settings. LittleFes have also proven to be attractive outreach tools. By “liberating” the BCCD onto LittleFe’s system disk users can easily move from temporary to persistent resources. Using the same “liberation” procedure and script the BCCD can be installed onto any system disk or virtual machine environment. The latest generation of LittleFe units support dual core mainboards and CUDA capable on-board chipsets, this allows an educator to support shared memory, distributed memory and hybrid parallel programming models in the classroom or workshop for less than the cost of a high-end laptop.

Development of the BCCD has moved to a Knoppix/Debian-based Linux distribution in version 3. This transition has allowed the system to develop more quickly to meet the rapidly changing requirements of CSE and HPC education,
and allows greater portability across computer architectures. Along with a new underlying Linux distribution, a more robust “liberation” system allows a user to take a booted live CD and copy its entire current state to a hard drive. After liberation, the system is capable of network booting diskless compute nodes, allowing for quick provisioning of a cluster. In order to manage a growing list of software packages, the BCCD ships with the modules shell environment management system, making it straightforward for users to manage a large amount of installed software. Finally, we are working to make the BCCD more accessible to users with disabilities. Accessible software is very important in education as it reduces the number of alternative curricula instructors must maintain, and reduces the barrier to science education for disabled students.

Because we have structured the BCCD for non-computational users such as STEM educators, we have designed the typical start process to require nothing more than loading the CD, supplying a password for the non-root “bccd” user, and then pressing the “Enter” key at all the network configuration prompts. If this is the first BCCD system on the network, it becomes the master node for all further BCCD systems that join the network, and provides DNS, DHCP, and NAT services to those nodes. DHCP is locked down so that only systems supplying a custom BCCD tag will receive a response, and the BCCD network is a private RFC–1918 network that typically will not interfere with a production network. Once a node comes online, it exchanges SSH public keys with other hosts on the network, allowing the passwordless logins that MPI jobs require. A custom BCCD command is available that will generate a valid MPI machines file based on the MPI module that the user has loaded, and will supply the correct CPU-count annotation if multiprocessor systems are available.

The BCCD has been used extensively as part of the Parallel Programming & Cluster Computing workshop series produced by NCSI. During these workshops, participants are given hands–on experience with the BCCD where they are introduced to some of the basic concepts behind cluster creation, maintenance, and usage. Throughout the workshop, the BCCD is then used as a platform for developing and running instructional HPC applications. At the end of the workshop, the participants can take the BCCD home on either a CD–ROM or USB flash drive, or download it for free from the BCCD website, and continue the workshop tutorials on their own.

As the BCCD continues to grow and evolve, its development is as much a teaching and learning opportunity as its use. Students and interns new to the fields of CSE and HPC can contribute to the testing and augmentation of the BCCD, in the process learning some of the gritty details involved in building and managing a dynamic cluster. In the past few years, the BCCD has come a long way, but it still has plenty of room for improvement. Along with the usual maintenance tasks inherent in such a project (e.g., software updates, bug fixes, documentation), we are planning significant feature additions such as integrating a job scheduler, implementing a robust and automated regression testing framework, and generating a global configuration system. These kinds of projects would be extremely well suited for any student interested in jumping into the cluster computing arena.

5. MENTORING

The authors’ extensive experience working with HPC users has shown that teaching HPC content is inadequate for full productivity; in addition, personal follow–up is required. At OU, teaching of content is followed by rounds, in which an HPC expert works directly with an individual researcher or research team (one-on-one or one-on-few) to facilitate adoption of HPC into a specific research endeavor. In OU’s original configuration, rounds were conducted by visiting each researcher directly, typically in their office on a weekly basis, using a regular timeslot. This approach met with great success, but proved to have significant scaling limitations, becoming unmanageable when the number of groups receiving rounds in the same semester reached approximately 20. As a result, OU has switched to a help session approach, under which one or more HPC experts spend scheduled time in a centrally located PC lab, with researchers who need HPC help coming on a drop-in basis to get assistance with the computational aspects of their investigations. Initially, these help sessions were conducted exclusively by Neeman, but over time other OSCER staff have taken on this task, to the point where, over the past year, Neeman has provided virtually none of this service, with other OSCER staff – primarily those whose backgrounds are in system-facing rather than user-facing tasks – taking over entirely, with no loss of amenity to OU HPC users. As a result, several of the co-authors of this paper recently submitted an NSF Cyberinfrastructure Training, Education, Advancement and Mentoring (CI–TEAM) grant proposal that would “train the trainers” to create a national community of HPC experts trained to provide rounds services (see below).

6. FUTURE WORK

6.1 NSF CI–TEAM grant proposal

Co-authors Neeman, Gray, Joiner, Panoff and Peck have proposed an NSF CI–TEAM Demonstration project as a follow-on to Neeman’s existing CI–TEAM project (OCI–0636427; “Cyberinfrastructure Education for Bioinformatics and Beyond,” $249,974; 12/01/06 – 11/30/08 with no cost extension to 11/30/10). The extant CI–TEAM project’s primary approach to teaching Cyberinfrastructure (CI) content uses lectures presented via videoconferencing to CI consumers, whereas the new proposed CI–TEAM project is designed to focus on a transformative means of “training the trainers” – future CI mentors – by leveraging: (a) OSCER’s longstanding, successful rounds program, and (b) NCSI’s summer tutorial workshop series.

If successfully funded, the proposed training regime will use an apprenticeship approach, with two years of training and then ongoing mastery. The CI–TEAM team will develop materials and select CI mentor apprentices in Spring of Project Year 1 (PY1). In Summer 1, the selected CI mentor apprentices will attend three NCSI week-long tutorial workshops: (1) Introduction to and (2) Intermediate Parallel Programming & Cluster Computing, and then (3) a new workshop specifically targeting their apprenticeship, focusing on the practicalities of providing CI mentoring to STEM researchers. Fall 1 and Spring 2 will have the CI mentor apprentices attending rounds remotely via videoconferencing,
in Fall 1 typically just observing but helping where possible, and in Spring 2 actively helping under supervision. In Summer 2, the mentors will attend the same tutorial workshops, but as instructional staff, providing both teaching and mentoring. In Fall 2, the CI mentors will conduct their own shared rounds, combining local and remote clientele, and thus will learn from one another’s experiences. In Spring 3, they will conduct their own individual rounds. Goals include:

1. The CI mentor apprentices will learn CI technical content, especially HPC, and, more generally, large scale computing, but also including a variety of practical issues; for example, deploying community software.

2. The CI mentor apprentices will acquire experience and soft skills focused on addressing user needs and concerns, expectations management, and especially providing CI facilitation for users via rounds.

3. The CI mentor apprentices, upon completing the apprenticeship, will have achieved sufficient mastery to train their own apprentices, whether at their local institution or remotely. Thus, considerable cost savings will be realized by participating institutions.

To assess the need for CI mentoring apprenticeships, over 100 institutions considered likely to have users needing CI mentoring were emailed, specifically (a) contacts at many of the member institutions of the Coalition for Academic Scientific Computation (CASC) [2] which represents most of the medium to large scale academic and government HPC centers in the US; (b) recipients of Major Research Instrumentation (MRI) awards for CI (typically clusters) during 2005–9; (c) institutions that the PIs were aware had established HPC centers otherwise not included in (a) or (b).

Upon proposal submission, 26 institutions in 17 states and DC, including 8 institutions in 7 EPSCoR jurisdictions, one Historically Black College or University (HBCU), one Hispanic Serving Institution (HSI), several rural and non–metro institutions, and many institutions with populations of African American, Native American and/or Hispanic students above that of the US population, had expressed interest in participating in the CI mentor apprenticeship program. Regarding consuming the apprentices’ rounds, a similar number of institutions were emailed, mostly institutions with which the PIs were already in contact because of other activities. Of the many institutions contacted, 38 in 11 states and DC, including 24 institutions in 8 EPSCoR jurisdictions, 4 of them HBCUs, 2 of them HSI, and one a high school, had expressed interest in consuming rounds via videoconferencing, for a grand total of 64 institutions in 23 states and DC, including 32 institutions in 11 EPSCoR jurisdictions, and among them 5 HBCUs, 3 MSIs and one high school, had expressed interest in participating in one or both activities at time of proposal submission.

The new workshop for the CI-TEAM program, focused on CI mentoring, would cover topics as follows:

Example Agenda: CI Mentoring Apprenticeship Workshop

- Day 1 - Introduction to Rounds, How to give a Supercomputing center tour
- Day 2 - Deploying Community Software, Benchmarking and Tuning
- Day 3 - Deploying Community Software II, Writing Grant Proposals
- Day 4 - Working with Telepresence Technology, Communication Tools and Techniques
- Day 5 - Finding and Provisioning Computational Resources, Collaboration Tools and Techniques

In addition, each full day will include presentations from real CI users about their research and rounds practice with real CI users.

7. ACKNOWLEDGMENTS

Portions of this material are based upon work supported by the National Science Foundation under the following grants: Grant No. OCI-0636427, “CI-TEAM Demonstration: Cyberinfrastructure Education for Bioinformatics and Beyond,” Grant No. DUE-0202452 “National Center of Excellence for High Performance Computing Technology,” Grant No. OCI-0722790, “Acquisition of Computing Equipment to Enhance Computational Science Research at Kean University,” and Grant No. DUE-0435187, “The Computational Science Education Reference Desk (CSERD)”.

8. ADDITIONAL AUTHORS

Paul Gray, University of Northern Iowa, grayca.uni.edu
Tom Murphy, Contra Costa College, tmurphy@contracosta.edu
and R.M. Panoff, Shodor, National Computational Science Institute, rpanoff@shodor.org

9. REFERENCES

Teaching parallel computing to science faculty: Best practices and common pitfalls. In PPoPP '06:
Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel
programming, pages 239–246, New York, NY, USA, 2006. ACM.


TeraGrid.

