Towards a community of Computer Science educators who teach more parallelism

Elizabeth Shoop
Macalester College (USA)
shoop@macalester.edu

Peter Hinsbeeck
Intel Corporation (Germany)
peter.hinsbeeck@intel.com

Richard Brown
St. Olaf College (USA)
rab@stolaf.edu

1. Rapid change is important

We assert that the recent rise of multi-core and distributed parallel computing presents a challenge to the Computer Science (CS) education community: *CS undergraduate programs need to quickly respond with curricular changes in order to prepare their students for the current state of computation, in which using parallelism is no longer an option, but is an absolute necessity.* CS educators will be able to create this change more rapidly if they can work together and create communities in which they share ideas, course materials, syllabi, and best practices and are able to discuss their use with each other while developing them. Instructors need supportive environments to help them bring about this change, particularly those whose professional background has not previously focused on parallel computing.

We participated in an international ITiCSE 2010 Working Group that, among other things, investigated and discussed strategies for creating such communities and how CS educators might go about making rapid change in their curricula. One section of our report [3] examines current efforts towards this goal as well as some issues that may confront those seeking to improve communities and create new ones. The report also considers the practical issues of bringing about such changes in a CS program and the faculty support that such changes would require. This position paper considers those practical issues and how the community might join together to overcome them.

2. Obstacles to change

As platforms that require parallelism in high-level language programming come to dominate computing, we can reasonably argue that industry now needs CS graduates who possess a significant background in parallelism. This expectation, together with the exciting possibilities that new forms of parallelism enable, provides motivation for CS programs to teach more parallelism. However, changing curriculum is not simple at most institutions, and many desirable curricular innovations compete with the goal of incorporating more parallelism into CS courses. As Dr. Michael Wrinn of Intel points out, a curricular shift towards parallelism was accomplished much more expediently in China, once the country's academic leadership was convinced [12]. For most other countries, and especially in the United States, each individual institution of higher learning must choose whether to proceed with a change in the curriculum and how to accomplish such change in their local circumstances, working from their existing array of courses.

Teaching parallelism at all levels of a CS curriculum, in courses that have not traditionally been associated with parallelism, places significant demands on the faculty teaching those courses. Up to now, parallelism has been a specialty of a small proportion of a CS faculty. The rapidly changing nature of all CS subfields has always placed a substantial burden on CS faculty in keeping up with their academic interests and responsibilities. Each CS professor must make his or her own cost-to-benefit calculation about how much time to invest in each new technology or disciplinary development, including the increasingly prominent developments involving parallelism. It will be important to lower the costs of investment in this transition by working together to affect change.

3. What is needed

An intentionally *supportive online community of peers* is key to helping faculty making the transition toward the parallel future of computing. Such a community will bring parallelism specialists and non-specialists together, and will also connect persons with similar professional interests outside of parallelism, for example, instructors of a given type of course, or persons with common academic interests.

A rich and accessible web repository of materials, ready for incremental insertion into one's course, will be essential for many instructors, but such repositories alone will not be enough. We believe that establishing an *open, non-threatening, sharing community of educators* for supporting this curricular transformation will both facilitate and expedite the necessary changes. The community of educators we envision would be web-based, taking strategic advantage of social computing technologies to make peer support convenient and accessible. It would collectively be a source for key information and services such as shared course materials of many kinds, information about platforms needed to use such materials, technical and non-technical support forums, and announcements about faculty training workshops and other activities of interest. As part of such a community, instructors need to feel that a vast community of others like them struggling to embrace this new technology and adopt it into their curricula will highly value their effort to share information, such as individual lessons and classroom activities, homework assignments, or term projects they have developed, along with the little successes that they have achieved, as well as things that didn't work out.

There is a great deal of literature regarding how online communities function, which we should draw upon when developing new communities for CS instructors. We would do well to pay attention to this work and to enlist the help of our colleagues in these fields when building parallel computing education sites. For example, Lampe and colleagues recently found that there are naturally different types of users on sites with community-generated content—those that primarily wish to get information, and those that wish to provide information—yet a 'sense of belonging' is important to both types of users [8]. Other research has established that a small percentage of users in user-generated content sites invariably generate a great deal of the content[11][4]. A hopeful sign, however, is that when newcomers

who post receive a response, they are much moreπ likely to continue to participate [6][7].

There are practical difficulties to making this community of educators we propose effective. Instructors must be convinced that their efforts are providing a service to their community, and that such service will be recognizable during tenure and promotion reviews. The current wide array of changing platforms and languages means that contributors need to provide details about environments under which shared materials have been run and tested and how to obtain resources needed or adapt materials for other platforms. Instructors and contributors will need to have some access to such resources for testing materials. Instructors must be able to find what they need for their particular course conveniently. Tagging shared content with meta data, either using a controlled vocabulary or user-generated words, will be important for searching and organizing materials on a site. There is a somewhat episodic nature to course development—instructors may put effort into some material, but need to move on to the next project or course before sharing it with others.

4. The larger context

We must urgently meet the objective of bringing more parallelism into CS curricula at all undergraduate levels. We assert that developing a supportive online community of educators is a *need*, not merely a desire, for attaining that goal as quickly as we must. The usual publication cycle, through conference papers, workshop presentations, and ultimately textbook incorporation, is already under way, and this too is necessary. But it is not sufficient, for several reasons. First, we expect the software environments for parallel computing to evolve rapidly as academic researchers and industry developers find new strategies for wielding parallel computing resources. This rapid evolution is suggested in [2]. The scale of this dynamic evolution of software environments seems likely to outpace the publication cycle. A similar evolutionary/speed argument may also apply to the body of knowledge of parallel computing itself.

The scale of this particular curricular change is the most substantial reason why curricular change as usual will be too slow. If every undergraduate CS student at every academic level must rapidly gain exposure to parallelism, then CS educators must develop all of those curricular resources in a short amount of time. Also, that majority of CS instructors who are not parallelism specialists must expediently acquire the background to teach appropriate parallel computing material well. An effective supportive community represents our best hope for accomplishing such systemic change so quickly. If this effort succeeds, perhaps it will serve as a model for future large-scale curricular change.

This desired community of educators may take time to develop, yet we don't have much time. Intel Corporation's Academic Community initiative [1] has shown great leadership towards this goal, in that Intel has invested resources towards creating a site where educators can share and review materials. Other sites are also seeking elements of this type of community, such as a wiki sponsor by OpenSparc called "Sharing Teaching Material for Concurrent Computing"[5]. Two authors of this progress report have recently launched a site designed to provide materials and solicit discussion and new submissions [10]. There is not yet enough widespread participation in these sites to form the community we are advocating.

Much work remains to discover and implement ever more effective support mechanisms that will engage and serve the emerging heterogeneous populace of CS educators who will soon begin teaching parallelism in new curricular contexts. The CS education community must be willing to take the time to develop and participate in online communities by contributing, returning to answer questions and discuss approaches, and updating material as situations change. In addition to participating online, educators with experience need to be willing to conduct and organize workshops demonstrating how they have been able to incrementally change their curriculum. It is only with this type of community effort from at least a core group of people that we will be able to make needed change fast enough to keep up with one of the more significant developments in our field– perhaps it starts with the help of those of us participating in this workshop.

5. References

[1] Academic - Intel® Software Network. *http://software.intel.com/en-us/academic/?cid=cim:ggl|academic_us_brand|ks112A7|s*. Accessed: 08-06-2010.

[2] Asanovic, K., Bodik, B. et al. 2006. *The Landscape of Parallel Computing Research: A View from Berkeley*. Technical Report #UCB/EECS-2006-183. EECS Department, University of California, Berkeley.

[3] Brown, R., Shoop, E. et al. 2010. Strategies for Preparing Computer Science Students for the Multicore World. *ACM SIGCSE Bulletin*. under review, (2010).

[4] Butler, B.S. 2001. Membership Size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures. *Info. Sys. Research*. 12, 4 (2001), 346-362.

[5] ConcurrentComputing < CourseMaterial < TWiki. *http://wiki.opensparc.net/bin/view.pl/CourseMaterial/ConcurrentComputing*. Accessed: 06-26-2010.

[6] Joyce, E. and Kraut, R.E. 2006. Predicting Continued Participation in Newsgroups. *Journal of Computer-Mediated Communication*. 11, 3 (2006), 723-747.

[7] Lampe, C. and Johnston, E. 2005. Follow the (slash) dot: effects of feedback on new members in an online community. *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work* (Sanibel Island, Florida, USA, 2005), 11-20.

[8] Lampe, C., Wash, R. et al. 2010. Motivations to participate in online communities. *Proceedings of the 28th international conference on Human factors in computing systems* (Atlanta, Georgia, USA, 2010), 1927-1936.

[9] Mitchell, S.M. and Lutters, W.G. 2006. Assessing the Value of Computer Science Course Material Repositories. *Conference on Software Engineering Education and Training Workshops* (Los Alamitos, CA, USA, 2006), 2.

[10] Parallel Computing in the Computer Science Curriculum. *http://csinparallel.org*. Accessed: 08-03-2010.

[11] Priedhorsky, R., Chen, J. et al. 2007. Creating, destroying, and restoring value in wikipedia. *Proceedings of the 2007 international ACM conference on Supporting group work* (Sanibel Island, Florida, USA, 2007), 259-268.

[12] Wrinn, M. 2010. Suddenly, all computing is parallel: seizing opportunity amid the clamor. *Proceedings of the 41st ACM technical symposium on Computer science education* (Milwaukee, Wisconsin, USA, 2010), 560-560.