Third International Workshop on Software Engineering for High Performance Computing (HPC) Applications

Jeffrey C. Carver
Mississippi State University
carver@cse.msstate.edu

1. Background

High performance computing systems are used to develop software in a wide variety of domains including nuclear physics, crash simulation, satellite data processing, fluid dynamics, climate modeling, bioinformatics, and financial modeling. The TOP500 website (http://www.top500.org) lists the top 500 high performance computing systems. The diversity of government, scientific, and commercial organizations present on this list illustrates the growing prevalence and impact of HPC applications on modern society.

Recent initiatives in the HPC community, such as the DARPA High Productivity Computing Systems program, recognize that dramatic increases in processor speed and memory access times do not necessarily translate into increases in development productivity. While the machines are getting faster, the developer effort required to fully exploit these advances can be prohibitive. There is an emerging movement within the HPC community to define new ways of measuring high performance computing systems that take into account not only the low-level hardware components, but the higher-level costs of producing usable HPC applications. This movement creates an opportunity for the software engineering community to apply our techniques and knowledge to a new and important application domain.

Furthermore, the design, implementation, and maintenance of HPC software systems can be significantly different from the systems more typically studied in the software engineering community.

1. The requirements often include sophisticated mathematical models and may take the form of an executable model, such as Matlab, that must be implemented on the proper platform.
2. Often these projects are exploring unknown science making it difficult to determine a concrete set of requirements *a prioiri*.
3. The software development process, or "workflow," for HPC application development may differ profoundly from traditional software engineering processes. For example, one scientific computing workflow, the "lone researcher", involves a single scientist developing a system to test a hypothesis. Once the system runs correctly once and returns its results, the scientist has no further need of the system. This approach contrasts with more typical software engineering lifecycle models, in which the useful life of the software is expected to begin, not end, after the first correct execution.
4. “Usability" in the context of HPC application development may revolve around optimization to the machine architecture so that computations complete in a reasonable amount of time. The effort and resources involved in such optimization may exceed initial development of the algorithm.

2. Workshop

This is the third in a series of workshops focusd on the relationship between traditional software engineering and software engineering that is focused specifically on applications for high performance supercomputers. This workshop brings together researchers from the software engineering community with researchers and practitioners from the high performance computing system application community. The organization of the workshop allows participants to share perspectives and present findings from research and practice that are relevant to the HPCS application development. A significant portion of the workshop is devoted to discussion of the position papers with the goal of generating a research agenda to improve tools, techniques, and experimental methods for HPC software engineering in the future.

29th International Conference on Software Engineering (ICSE'07 Companion)
0-7695-2892-9/07 $20.00 © 2007 IEEE