Information Science and Technology Seminar Speaker Series

Kathleen Knobe
Rice University

Concurrent Collections for Both Productivity and Parallel Performance

Wednesday, June 10, 2015

3:00 - 4:00 PM

TA-3, Bldg. 1690, Room 102 (CNLS Conference Room)

Abstract: Parallel programming is hard. One reason is that it requires that we consider many different concerns at once. By isolating the concerns we can improve both productivity and performance.

In general, applications express both computations and orderings among the computations. Some orderings are absolutely required by the semantics (dependences), some are for tuning (for a specific goal and platform) and some are purely arbitrary. To modify the application as it evolves, the required orderings must be deduced. During this process, unnecessary orderings might mistakenly be seen as required (it's ineffective) or required orderings might mistakenly be seen as arbitrary (it's error-prone). Furthermore the process takes time (it's inefficient).

CnC's philosophy is to separate the three concerns: chunks of computations, the required orderings among them (the domain spec) and additional constraints for tuning purposes (the tuning spec). This separation leads to a simpler, more effective and more efficient development process. CnC is not a parallel programming model. There is no thinking about parallelism involved. But it isn't a serial model either. There are no arbitrary serial orderings. We refer a CnC programming as dependence programming. The CnC domain spec is exactly a dependence graph.

Because it is high level and declarative, we can and have implemented a variety of quite different execution styles for CnC. With respect to tuning, not only can a single domain spec be used, unchanged, with multiple tuning specs, in fact, it can be used, unchanged, with many different styles of tuning.

The talk will introduce the CnC concepts, show various tuning approaches, describe the current state of the project and our future plans.

(This work was partially supported by the DOE X-Stack project.)

Biography: Kathleen Knobe worked at Compass (aka Massachusetts Computer Associates) from 1980 to 1991 designing compilers for a wide range of parallel platforms including Thinking Machines, MasPar, Alliant, Numerix, and several government projects. In 1991 she decided to finish her education. After graduating from MIT (with Bill Dally) in 1997, she joined Digital Equipment's Cambridge Research Lab (CRL). She stayed through the DEC/Compaq/HP mergers and when CRL was absorbed into Intel. She is now a senior research scientist at Rice University.

For more information contact the technical host Susan Mniszewski, smm@lanl.gov, 667-0790
or Christoph Junghans, junghans@lanl.gov, 665-2278.

Hosted by the Information Science and Technology Institute (ISTI)