I2PC Illinois-Intel Parallelism Center

Distinguished Speaker Series

Kathleen Knobe

Intel Corporation

Concurrent Collections (CnC): Application parallelism via coordination

Thursday, March 29, 2012

4 - 5 PM CST

3405 Siebel Center

For more information about I2PC research, education, and activities, visit:
i2pc.cs.illinois.edu

Attend this talk remotely at:
http://media.cs.illinois.edu/live/I2PClive.asx

A chat interface has been set up for questions:
http://i2pc.cs.illinois.edu/chat

Abstract

Explicitly parallel languages and explicitly serial languages are each over-constrained, though in different ways. Concurrent Collections (CnC), on the other hand, maximizes the scheduling freedom for a given target (efficiency) and also among distinct targets (portability). The domain expert writing a CnC program focuses on the meaning of the application, not on how to schedule it.

To prepare an application for parallel execution, we first need to answer two questions: "How should the data and computation be divided into chunks that are potential parallel?" and "What are the scheduling constraints among these chunks?" A CnC program specifies exactly this information. The resulting program is "ready for parallelism." CnC isolates the work of the domain expert (interested in finance, chemistry, gaming…) from the tuning expert (interested in load balance, locality, scalability, …) This isolation minimizes the need for the domain expert to think about all the complications of parallel systems. CnC is a coordination language that specifies the required orderings among potentially parallel chunks of application. As a coordination language it must be paired with a computation language. Intel® Concurrent Collections for C++ supports C++ programs.

The talk will include an introduction to the CnC domain specification, an overview of an entirely separate approach for specifying the tuning of the domain spec and performance results for the Intel distributed CnC/C++ system.

Bio

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 in 1997, she joined Digital Equipment's Cambridge Research Lab (CRL). She stayed through the DEC/Compaq/HP mergers and when CRL was acquired by Intel. She currently works in the Software Solutions Group / Developer Products Group) at Intel.

In addition to CnC, her major projects include the Subspace Model of computation (a compiler internal form for parallelism), Data Optimization (compiler transformations for locality), Array Static Single Assignment form (a method of achieving for array elements the advantages that SSA has for scalars), Weak Dynamic Single Assignment form (a global method for eliminating overwriting of data to maximize scheduling flexibility), Stampede (a programming model for streaming media applications).

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN