I2PC

Illinois-Intel Parallelism Center

Distinguished Speaker Series

Paul Petersen
Intel Corporation

The Hardest Part of Parallel Programming is Understanding the Limitations of your Serial Algorithms

Thursday, November 17, 2011

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

Serial algorithms typically run very inefficiently on parallel machines. This may sound like an obvious statement, but it is the root cause of why parallel programming is considered to be difficult. The current state of the computer industry is still that almost all programs in existence are serial. To address this situation, Intel has created Parallel Studio, and in particular Parallel Advisor.

This talk will describe the techniques used in Parallel Advisor to provide a developer with the tools necessary to understand the limitations of the existing serial algorithms. One the limitations are known the developer can refactor the algorithms and reanalyze the resulting code to see if it could run effectively on parallel hardware. Almost all implementations of serial algorithms are serial for a reason, and the tools available in Parallel Advisor help the user expose these reasons so that appropriate rewrites can be done.

Bio

Paul Petersen is a Sr. Principal Engineer in the Software and Solutions Group (SSG) at Intel. He received a Ph.D. degree in Computer Science from the University of Illinois in 1993. After UIUC, he was employed at Kuck and Associates, Inc. (KAI) working on auto-parallelizing compiler (KAP), and was involved in the early definition and implementations of OpenMP. While at KAI, he developed the Assure line of parallelization/correctness products, for Fortran, C++ and Java. In 2000, Intel Corporation acquired KAI, and he joined the software tools group. At Intel, he worked with the tools group to create the Thread Checker products, which evolved into the Inspector and Advisor components of the Intel® Parallel Studio. Inspector uses dynamic binary instrumentation to detect memory and concurrency bugs, and Advisor uses similar techniques along with performance measurement and modeling to assist developers in transforming existing serial applications to be ready for parallel execution.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN