All Symposium Tutorial
(open to all attendees at no charge)

Tuesday, April 15th at 7:00 PM

Title: *Open64 Compiler infrastructure for emerging multicore/manycore architecture*

Presenters

Organizer: Guang R. Gao, University of Delaware
Barbara Chapman, University of Houston
Tony Linthicum & Anshuman Dasgupta, Qualcomm Inc.
Juergen Ributzka, University of Delaware

Abstract

Open64 was originally developed by SGI and released as the MIPSpro compiler. It has been well-recognized as an industrial-strength production compiler for high-performance computing. It includes advanced interprocedural optimizations, loop nest optimizations, global scalar optimizations, and code generation with advanced global register allocation and software pipelining. It was open-sourced in 2000 after it was retargeted to the Itanium processor. Now, Open64 are accepted by many compiler researchers as a good infrastructure for the research on new compiler optimizing technologies, especially the for the emerging multi-core/many-core architecture.

Presentation Outline

7:00pm – 7:30pm
A Briefly Introduction to Open64
Presenter: Guang R. Gao, University of Delaware
Abstract: Open64 was originally developed by SGI and released as the MIPSpro compiler. It has been well recognized as an industrial-strength production compiler for high-performance computing. It includes advanced interprocedural optimizations, loop nest optimizations, global scalar optimizations, and code generation with advanced global register allocation and software pipelining. It was open-sourced in 2000 after it was retargeted to the Itanium processor. There have been a number of subsequent branches and improvements to the compiler since then. Intel adopted the Open64 compiler for compiler-related research and subsequently released it as the Open Research Compiler (ORC) starting Jan 2002. During this time, Intel drove ORC to outstanding performance and functionality and released ORC 2.1 in the summer of 2003. Later, Pathscale (acquired by Qlogic in early 2006) released a branch of Open64 for the AMD Opteron processor in 2004, bringing the high performance open source compiler to x86-64 developer community. HP has sponsored the Open64 Compiler project for Itanium since November 2005 following the path of ORC for compiler research with additional focus on quality and upgrading C++ language support to stay close to the GCC front end evolution. Open64 has been ported to different architectures and this tutorial will present the work and results on them.

7:30pm – 8:15pm
OpenUH: An Open Source Reference Compiler for OpenMP
Presenter: Barbara Chapman, University of Houston
Abstract: This talk will give an overview of OpenMP and its implementation in the OpenUH (a variant of Open64) compiler. We will also talk about other extensions we have made to Open64 in order to create our branch of Open64 and our experiences in doing so. In particular, this includes features that have enabled us to interact closely with a variety of third-party tools, including TAu, Kojak, and PerfSuite. We will describe how we were able to experiment with language features (extensions to the existing languages Fortran, C and C++) with Open64, and we will introduce the OPen64 loop nest optimizer, as well as the cost model and our enhancements to it as an example of one way in which its optimizations can be influenced. We will also discuss our experiences using Open64 technology in a university environment, both for research and for teaching.

8:30pm – 9:15pm
The Open64 Compiler for QDSP6
Presenters: Tony Linthicum & Anshuman Dasgupta, Qualcomm
Abstract: The discussion will center on our experiences with re-targeting Open64 to a DSP, and why we chose it over the other open source compiler systems available. We will discuss both the strengths and weaknesses of Open64 for the DSP and how we have sought to exploit the former and minimize the latter. We will also delve into the challenges that our particular architecture posed for the machine independent portions of the compiler and how we solved those problems. Enhancements to Open64 driven by the needs of our customer base will also be described.

9:15pm – 10:00pm
Software-Pipelining on Multi-Core Architecture
Presenter: Juergen Ributzka, University of Delaware
Abstract: It is becoming increasingly evident that multi-core chip architecture is emerging as a solution to efficiently amortizing the ever-growing number of transistors on a chip. However the success of such multi-core chips depends on the advances in system software technology, such as compiler and run-time system, in order for the application programs to exploit thread level parallelism out of originally single-threaded applications and to fully utilize the hardware on-chip concurrency. We propose a method which, from a parallel and non-parallel imperfect loop nest written in a standard sequential language such as C or Fortran, automatically generates a multi-threaded software-pipelined schedule for multi-core architectures. The generated schedule already contains all the necessary synchronization instructions and is guaranteed free of deadlocks and buffer overflow. The feasibility of the proposed method within a modern compiler infrastructure has been verified through a pilot implementation in the Open64 compiler and tested on the IBM Cyclops multi-core architecture. Experimental results show that the performance exhibits good scalability even with 100 cores. Our light-weight synchronization mechanism minimizes the dependencies stalls and synchronization overheads without the use of dedicated hardware support.

Brief Bios

Guang R. Gao - Dr. Guang R. Gao received his S.M. and Ph.D. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, in 1982 and 1986, respectively. He is an IEEE Fellow and ACM Fellow. Currently he is a Professor at the Department of Electrical and Computer Engineering at the University of Delaware, where he has been the founder and leader of the Computer Architecture and Parallel Systems Lab. His research interests are parallel computer architecture and computer system, compiler optimizing techniques, programming language.

Barbara Chapman - Ph. D., Queen's University, Belfast, N. Ireland. Currently, she is a full professor in the department of computer science at Univ. of Houston. Her research interests are compiler technology, parallel programming languages, tool support for application development, parallel computing and high-performance computing, computational Grids.

Tony Linthicum - Tony Linthicum is a staff engineer from Qualcomm Inc., and he is now the technical lead of the Open64 compiler team in Qualcomm. Currently, he is adding ability for IPA to deal with mixed object files (Whirl and ELF) in Open64 compiler. Before he joined Qualcomm, he had worked in AMD Inc., Tera Systems Inc., Equator Tech. Inc., SGI Inc., and Convex Inc. Tony has an extensive experience in designing and implementing an optimizing compiler. He designed and implemented the SSA construction pass in Tera's HDL analysis tool. He also developed the call graph construction, inter-procedural analysis, code generator & optimizer for a VHDL compiler when he was in Stage2 Innovations Inc. When he was in SGI, he designed and implemented Mahlke's hyperblock formation algorithm in Pro64 that targets IA64 platform. He was also deeply involved in the design and implementation of global and local register allocator in Pro64.

Anshuman Dasgupta – A member of the compiler team at Qualcomm Inc. He received a B.A. degree in Computer Science from Ohio Wesleyan University and M.S. and Ph.D. degrees in Computer Science from Rice University. His research interests include runtime compilation, register allocation, and binary optimizations. He currently is employed at Qualcomm, Inc. as a senior engineer on the compiler team.

Juergen Ributzka - Juergen Ributzka began his higher education as a Software Engineer at the University of Applied Sciences in Esslingen. His interest expanded to multicore / multi processor systems when joining the University of Delaware for their ECE Doctoral program. Among his work experience is research and implementation of an object classification system used for Parking Spot Localization (PSL) in cars. His interests include, but are not limited to, GPGPU programming, multicore / multiprocessor architectures and runtime systems, high performance compilers and toolchains, architecture benchmarking and software engineering.

Contact Person: gan@ece.udel.edu