ML-WorkloadDistribution

- Qilin: exploiting parallelism on heterogeneous multiprocessors with adaptive mapping

  Chi-Keung Luk, Sunpyo Hong, Hyesoon Kim
  MICRO 42 Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture
  2009

  Heterogeneous multiprocessors are increasingly important in the multi-core era due to their potential for high performance and energy efficiency. In order for software to fully realize this potential, the step that maps computations to processing elements must be as automated as possible. However, the state-of-the-art approach is to rely on the programmer to specify this mapping manually and statically. This approach is not only labor intensive but also not adaptable to changes in runtime environments like problem sizes and hardware/software configurations. In this study, we propose *adaptive mapping*, a fully automatic technique to map computations to processing elements on a CPU+GPU machine. We have implemented it in our experimental heterogeneous programming system called*Qilin*. Our results show that, by judiciously distributing works over the CPU and GPU, automatic adaptive mapping achieves a 25% reduction in execution time and a 20% reduction in energy consumption than static mappings on average for a set of important computation benchmarks. We also demonstrate that our technique is able to adapt to changes in the input problem size and system configuration.

- Cross-architecture performance predictions for scientific applications using parameterized models

  Gabriel Marin, John Mellor-Crummey
  SIGMETRICS '04/Performance '04 Proceedings of the joint international conference on Measurement and modeling of computer systems
  2004

  This paper describes a toolkit for semi-automatically measuring and modeling static and dynamic characteristics of applications in an architecture-neutral fashion. For predictable applications, models of dynamic characteristics have a convex and differentiable profile. Our toolkit operates on application binaries and succeeds in modeling key application characteristics that determine program performance. We use these characterizations to explore the interactions between an application and a target architecture. We apply our toolkit to SPARC binaries to develop architecture-neutral models of computation and memory access patterns of the ASCI Sweep3D and the NAS SP, BT and LU benchmarks. From our models, we predict the L1, L2 and TLB cache miss counts as well as the overall execution time of these applications on an Origin 2000 system. We evaluate our predictions by comparing them against measurements collected using hardware performance counters.

- Design principles for end-to-end multicore schedulers

  Simon Peter, Adrian Schüpbach, Paul Barham, Andrew Baumann, Rebecca Isaacs, Tim Harris, Timothy Roscoe
  HotPar'10 Proceedings of the 2nd USENIX conference on Hot topics in parallelism
  2010

As personal computing devices become increasingly parallel multiprocessors, the requirements for operating system schedulers change considerably. Future general-purpose machines will need to handle a dynamic, bursty, and interactive mix of parallel programs sharing a heterogeneous multicore machine. We argue that a key challenge for such machines is rethinking scheduling as an end-to-end problem integrating components from the hardware and kernel up to the programming language runtimes and applications themselves.

We present several design principles for future OS schedulers, and discuss the implications of each for OS and runtime interfaces and structure. We illustrate the implementation challenges that result by describing the concrete choices we have made in the Barrelfish multikernel. This allows us to present one coherent scheduling design for an entire multicore machine, while at the same time drawing conclusions we think are applicable to the design of any general-purpose multicore OS

- Mapping parallelism to multi-cores: a machine learning based approach

  Zheng Wang, Michael F.P. O'Boyle
  PPoPP '09 Proceedings of the 14th ACM SIGPLAN symposium on Principles and practice of parallel programming
  2009

  The efficient mapping of program parallelism to multi-core processors is highly dependent on the underlying architecture. This paper proposes a portable and automatic compiler-based approach to mapping such parallelism using machine learning. It develops two predictors: a data sensitive and a data insensitive predictor to select the best mapping for parallel programs. They predict the number of threads and the scheduling policy for any given program using a model learnt off-line. By using low-cost profiling runs, they predict the mapping for a new unseen program across multiple input data sets. We evaluate our approach by selecting parallelism mapping configurations for OpenMP programs on two representative but different multi-core platforms (the Intel Xeon and the Cell processors). Performance of our technique is stable across programs and architectures. On average, it delivers above 96% performance of the maximum available on both platforms. It achieve, on average, a 37% (up to 17.5*times*) performance improvement over the OpenMP runtime default scheme on the Cell platform. Compared to two recent prediction models, our predictors achieve better performance with a significant lower profiling cost.

  Cited:

  E. Ipek, B. R. de Supinski, et al. An approach to performance prediction for parallel applications. In Euro-Par'05, 2005.

  S. Long, G. Fursin, et al. A cost-aware parallel workload allocation approach based on machine learning. In NPC '07, 2007.

Qilin: exploiting parallelism on heterogeneous multiprocessors with adaptive mapping

Authors: Chi-Keung Luk Intel Corporation, Hudson, MA
Sunpyo Hong Georgia Institute of Technology, Atlanta, GA
Hyesoon Kim Georgia Institute of Technology, Atlanta, GA

Published in:

· Proceeding
MICRO 42 Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture
©2009 ISBN: 978-1-60558-798-1 doi>10.1145/1669112.1669121

Heterogeneous multiprocessors are increasingly important in the multi-core era due to their potential for high performance and energy efficiency. In order for software to fully realize this potential, the step that maps computations to processing elements must be as automated as possible. However, the state-of-the-art approach is to rely on the programmer to specify this mapping manually and statically. This approach is not only labor intensive but also not adaptable to changes in runtime environments like problem sizes and hardware/software configurations. In this study, we propose *adaptive mapping*, a fully automatic technique to map computations to processing elements on a CPU+GPU machine. We have implemented it in our experimental heterogeneous programming system called*Qilin*. Our results show that, by judiciously distributing works over the CPU and GPU, automatic adaptive mapping achieves a 25% reduction in execution time and a 20% reduction in energy consumption than static mappings on average for a set of important computation benchmarks. We also demonstrate that our technique is able to adapt to changes in the input problem size and system configuration.

Cross-architecture performance predictions for scientific applications using parameterized models

Authors: Gabriel Marin Rice University, Houston, TX
John Mellor-Crummey Rice University, Houston, TX

Published in:

SIGMETRICS

· Proceeding
SIGMETRICS '04/Performance '04 Proceedings of the joint international conference on Measurement and modeling of computer systems
©2004 ISBN:1-58113-873-3 doi>10.1145/1005686.1005691

· Newsletter
ACM SIGMETRICS Performance Evaluation Review Homepage
Volume 32 Issue 1, June 2004 doi>10.1145/1012888.1005691

This paper describes a toolkit for semi-automatically measuring and modeling static and dynamic characteristics of applications in an architecture-neutral fashion. For predictable applications, models of dynamic characteristics have a convex and differentiable profile. Our toolkit operates on application binaries and succeeds in modeling key application characteristics that determine program performance. We use these characterizations to explore the interactions between an application and a target architecture. We apply our toolkit to SPARC binaries to develop architecture-neutral models of computation and memory access patterns of the ASCI Sweep3D and the NAS SP, BT and LU benchmarks. From our models, we predict the L1, L2 and TLB cache miss counts as well as the overall execution time of these applications on an Origin 2000 system. We evaluate our predictions by comparing them against measurements collected using hardware performance counters.

Design principles for end-to-end multicore schedulers

Authors: Simon Peter — Systems Group, ETH Zurich
Adrian Schüpbach — Systems Group, ETH Zurich
Paul Barham — Microsoft Research, Cambridge
Andrew Baumann — Systems Group, ETH Zurich
Rebecca Isaacs — Microsoft Research, Cambridge
Tim Harris — Microsoft Research, Cambridge
Timothy Roscoe — Systems Group, ETH Zurich

Published in:
· Proceeding
HotPar'10 Proceedings of the 2nd USENIX conference on Hot topics in parallelism

As personal computing devices become increasingly parallel multiprocessors, the requirements for operating system schedulers change considerably. Future general-purpose machines will need to handle a dynamic, bursty, and interactive mix of parallel programs sharing a heterogeneous multicore machine. We argue that a key challenge for such machines is rethinking scheduling as an end-to-end problem integrating components from the hardware and kernel up to the programming language runtimes and applications themselves.

We present several design principles for future OS schedulers, and discuss the implications of each for OS and runtime interfaces and structure. We illustrate the implementation challenges that result by describing the concrete choices we have made in the Barrelfish multikernel. This allows us to present one coherent scheduling design for an entire multicore machine, while at the same time drawing conclusions we think are applicable to the design of any general-purpose multicore OS.

Mapping parallelism to multi-cores: a machine learning based approach

Authors: Zheng Wang The University of Edinburgh, Edinburgh, United Kingdom
Michael F.P. O'Boyle The University of Edinburgh, Edinburgh, United Kingdom

Published in:

· Proceeding

PPoPP '09 Proceedings of the 14th ACM SIGPLAN symposium on Principles and practice of parallel programming

©2009 ISBN: 978-1-60558-397-6 doi>10.1145/1504176.1504189

· Newsletter

ACM SIGPLAN Notices - PPoPP '09

Volume 44 Issue 4, April 2009 doi>10.1145/1594835.1504189

The efficient mapping of program parallelism to multi-core processors is highly dependent on the underlying architecture. This paper proposes a portable and automatic compiler-based approach to mapping such parallelism using machine learning. It develops two predictors: a data sensitive and a data insensitive predictor to select the best mapping for parallel programs. They predict the number of threads and the scheduling policy for any given program using a model learnt off-line. By using low-cost profiling runs, they predict the mapping for a new unseen program across multiple input data sets. We evaluate our approach by selecting parallelism mapping configurations for OpenMP programs on two representative but different multi-core platforms (the Intel Xeon and the Cell processors). Performance of our technique is stable across programs and architectures. On average, it delivers above 96% performance of the maximum available on both platforms. It achieve, on average, a 37% (up to 17.5*times*) performance improvement over the OpenMP runtime default scheme on the Cell platform. Compared to two recent prediction models, our predictors achieve better performance with a significant lower profiling cost.