Forward : 1st JILP Workshop on Computer Architecture Competitions (JWAC-1): Cache Replacement Championship

Joel Emer, Aamer Jaleel

The Journal of Instruction-Level Parallelism

1st JILP Workshop on Computer Architecture Competitions (JWAC-1):

Cache Replacement Championship

Held In Conjunction with ISCA'2010

Saint Malo, France

Forward

Following are the proceedings of the first annual Journal of Instruction-Level Parallelism Workshop on Architecture Competitions (JWAC-1). Each incarnation of this planned series of workshops is intended to provide a forum for the presentation of the best algorithms to implement one of the various important components of modern processors. It is further hoped that the workshop's organization as a contest will bring out a collection of the best algorithms for each year's topic.

The JWAC competitions are guided by the desire to provide a maximum of benefit for researchers and industry. This is accomplished in a variety of ways. First, the workshop organizers provide a standard framework for implementing algorithms for the competition. This framework provides a common baseline design and produces metrics of interest, typically performance, for the contestant's algorithm. Having a common framework that alternative algorithms can easily plug into is designed to reduce the barrier of entry for contestants. Second, the evaluations of the metrics for the algorithms are conducted across a standard set of workloads to provide a common and hopefully representative characterization of the alternatives. Together with the common baseline-design framework, the common workloads provide an opportunity to make a fair and balanced comparison among the alternative designs. Note these workloads are not provided to the contestants during the competition to avoid statistical problems related to 'designing to the workload'. Finally to allow for more comprehensive evaluations and results representative of the best algorithms known, novelty is not required for participation in the contest - if FIFO cache replacement is best we want to know that too.

This year's JWAC competition has been on cache replacement algorithms for last-level caches. While the design of cache replacement algorithms dates back to the classical age of computer architecture, there has been a resurgence of interest in the area. The Moore's Law driven increases in cache sizes and especially the proliferation of shared caches resulting from the multi-core trend has piqued both industrial and academic interest in the topic. The result has been a number of cache replacement-related innovations in topics ranging from the application of machine learning techniques to provisioning for quality of service among processes and processors. Therefore, this year's contest has categories for the best algorithms for both uni-processor and multi-processor systems.

For JWAC-1, the organizing committee provided contestants a framework to evaluate the performance of their cache replacement ideas. Details about the simulator can be found on the JWAC-1 website: http://www.jilp.org/jwac-1/. Contestants downloaded the framework from the website and evaluated their replacement ideas on the following uni-processor and/or multi-processor configurations:

1. Single-core system with a 1MB LLC.
2. 4-core CMP with a 4MB shared last-level cache.

Each contestant was asked to submit a 4-page paper describing the implementation and intuition behind the proposed replacement ideas. The contestants were required to (a) adhere to a fixed hardware budget of 8-bits per cache line plus an additional 1K bits for the entire cache (b) submit their source code with the paper (up to three source code variations of the replacement idea were allowed).

The submissions were evaluated by the workshop organizers on a pre-selected set of workloads that were only available to organizers. Besides the often used SPEC CPU2006 benchmark suite, the organizing committee also evaluated the proposals on real world workloads. For uni-processor studies, a total of 65 workloads were used. The workloads consisted of 22 multimedia and PC game workloads, 14 enterprise server workloads, and 29 SPEC CPU2006 workloads. For multi-processor studies, heterogeneous mix of workloads was used. Seven representative workloads from each category were first selected. All possible 4-core combinations of the seven workloads from each category (7 choose 4 = 35 combinations) were then created. Thus, the organizing committee used 35 heterogeneous mixes of multimedia and PC games, 35 heterogeneous mixes of enterprise server workloads, and 35 heterogeneous mixes of SPEC CPU2006 workloads. Finally, using the 21 representative workloads, the organizing committee created another 35 random combinations of 4-core workloads. The heterogeneous mix of different workload categories are used as a proxy for a virtualized system. In total, the organizing committee used 140 workloads for multi-processor configuration and 65 workloads for uni-processor configurations.

Representative regions for each workload were selected using a SimPoint like methodology. The SPEC CPU2006 workload traces were generated using Pin while the real world workloads were generated from a bus-based hardware tracing system. The real world workloads include both user-level and system-level activity. Each SPEC CPU2006 workload was used with the first reference input set. All applications warmed up the caches for the first 100M instructions and then ran for another 100M instructions. For shared cache studies, all applications in the workload continue executing until the last application finishes the requested instruction count. If an application reaches the end of the trace, the simulator restarts executing the application from the beginning.

Since the focus of this competition is on last-level cache replacement, the submissions were only evaluated at the last-level cache. The smaller levels of the cache hierarchy all used LRU replacement. For the private cache configuration, the performance metric was throughput improvement relative to an LRU managed LLC. For the shared cache configuration, the performance metric was weighted speedup relative to an LRU managed shared LLC. The submissions were then ranked separately for private and shared caches.

Overall, the competition was a tremendous success. We received 26 papers and 35 code submissions. The submissions came from different geographic locations with 12 from Asia, 3 from Europe, and 11 from North America. Due to the large number of submissions, and the desire to have four reviews per submission, the program chair and members of the organizing committee also reviewed papers. Each reviewer was responsible for 10 papers. To ensure a fair review process, the reviewers had no knowledge of the performance ranking of the submissions.

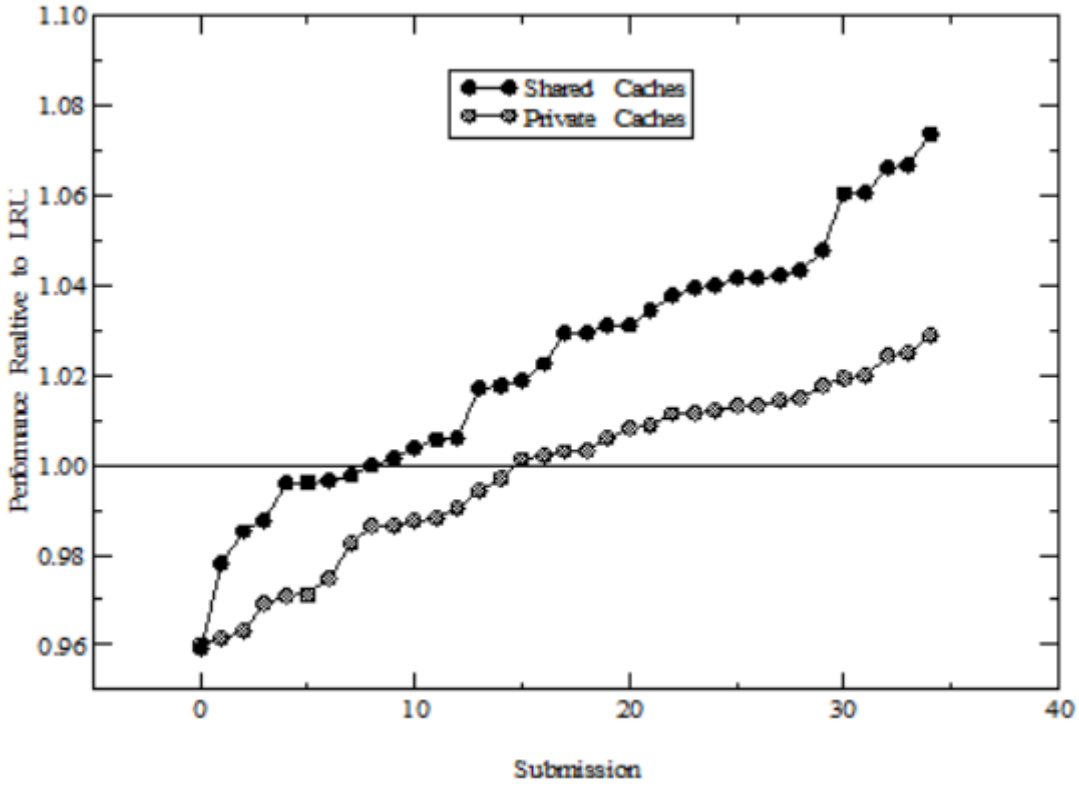

Figure 1: This figure shows the performance of the 27 submissions on private and shared caches.

A total of 10 papers were selected for presentation at the workshop. The selection of papers was primarily based performance ranking but also on paper quality, adherence to competition rules, and a qualitative assessment of logic complexity, which couldn't be incorporated in the cost metric. The accepted papers are listed in the table of contents below. All source code, papers, and results from the infrastructure are publicly available on the JWAC-1 website so that the computer architecture community can benefit from the results. Figure 1 shows anonymous performance scores of all 35 code submissions for uni-processor and multiprocessor configurations. The winners will be announced at the workshop.

Finally, we would like to thank everyone for their hard work putting this program together. Special thanks go to the organizing committee, Alaa Alameldeen and Moin Qureshi, for taking care of the contest logistics and setup. We also thank the program committee for their conscientious evaluation of all the papers. Additional thanks go to Eric Rotenberg for running the contest web site and for JILP sponsorship. Finally, we thank Intel for their financial support for the contest.

Joel Emer, Program Chair
Aamer Jaleel, Organizing Committee Chair

JWAC-1 COMMITTEE

Steering Committee:

Alaa R. Alameldeen, Intel

Eric Rotenberg, NC State

Organizing Committee Chair:

Aamer Jaleel, Intel

Organizing Committee:

Alaa R. Alameldeen, Intel

Moinuddin Qureshi, IBM

Program Chair:

Joel Emer, Intel

Program Committee:

Doug Burger, Microsoft

Mainak Chaudhuri, IITK

Aamer Jaleel, Intel

Gabriel Loh, Georgia Tech

Moinuddin Qureshi, IBM

Yan Solihin, NC State

TABLE OF CONTENTS