PART 1 PAPI for Arithmetic Intensity  Anthony Danalis, Heike Jagode, Daniel Barry

The goal of this work is to create a set of PAPI presets (predefined events) for effortless computation of the Arithmetic Intensity (a.k.a. Computational Intensity), measured as ratio of computation to traffic (Flops / bytes).

Floating-point Operations: ddot, dgemm

FLOPS involve multiple events for capturing operations of different vector length.

Memory Traffic: ddot, dgemm

Traffic to DRAM involves multiple non-trivial source (Intel)/northbridge (AMD)/nest (IBM) events.

PART 2 PAPI’s Counter Analysis Toolkit  Anthony Danalis, Heike Jagode, Daniel Barry

The goal of this work is to create a set of microbenchmarks for illustrating details in hardware events and how they relate to the behavior of the microarchitecture.

PART 3 Modernizing PAPI Infrastructure  Vince Weaver and Van Liu

Improved PAPI Test Infrastructure

- The existing PAPI test suite is used to test the correctness of PAPI before release.
- The hardware and operating systems used by PAPI are always changing, and some of the existing tests were outdated or gave false negatives.
- New counter validation tests were created, which should provide a sanity check when bringing up support for a new processor architecture.

Low-Overhead PAPI read() Support

- Traditionally, PAPI_read() counter reads went through the standard Linux read() system call, which can be slow (around 1,000 cycles).
- x86 hardware supports a user-space rdrpc() instruction that bypasses the kernel and requires 200 cycles (2.5+ speedup).
- Various bugs in the Linux kernel around this interface were found and fixed so that rdrpc() can be enabled by default.

Enhanced Sampling Interface

- PAPI currently has a limited counter-sampling interface that only allows gathering the instruction pointer at regular intervals.
- Modern processors support much richer sampling information, including the cause of cache misses, where in the cache hierarchy the miss happened, and the cycles taken.
- We extended the PAPI sampling interface to provide this additional sampling information.