Performance Application Programming Interface (PAPI) provides a consistent interface (and methodology) for hardware performance counters found across a compute system: i.e., CPUs, GPUs, on- and off-chip memory, interconnects, I/O system, file system, and energy/power. PAPI enables software engineers to see, in near real time, the relationship between software performance and hardware events across the entire compute system.

STANDARD FEATURES
- Standardized Performance Metrics
- Easy Access to Platform-Specific Metrics
- Multiplexed Event Measurement
- Dispatch on Overflow
- Overflow & Profiling on Multiple Simultaneous Events
- Bindings for C, Fortran and Matlab
- User Definable Metrics derived from Platform-Specific Metrics
- Support for Virtual Computing Environments
- Performance Counter Monitoring at Task Granularity for the PaRSEC Dataflow Runtime

SUPPORTED ARCHITECTURES
- AMD
  - X86
- ARM
  - Cortex A7, A8, A9, A15, X-Gene (ARM64), Raspberry Pi
- Cray
  - Gemini and Aries
  - interconnects | RAPI, power
- IBM
  - Blue Gene Series, 0, 5D-Torus, I/O system, EMON power on BG/Q
  - Power Series, PCP-support for POWER9

Vendor HW/Kernel Extension
OS/Kernel

Infiniband
- Nehalem, Westmere, Sandy Bridge, Ivy Bridge, Haswell, Haswell-EP, Broadwell, Skylake(X), Knights Corner, Knights Landing, Knights Mill | RAPI: power capping, Power on Xeon Phi | Intel RAPI (power/energy), power capping capabilities (via PAPI_write())

Lustre
- Tesla, Kepler, Maxwell, Pascal, Volta; support for multiple GPUs | NVIDIA GPU support for power/energy reading; power capping capabilities (NVML)

Virtual Environment
- VMWare, KVM

IN COLLABORATION WITH
U.S. Department of Defense
U.S. Department of Energy
ECP
National Science Foundation

WITH SUPPORT FROM
AMD
CRAY
IBM
Intel
NVIDIA

FIND OUT MORE AT https://icl.utk.edu/papi
Extending PAPI for ECP Applications

The Exascale Performance Application Programming Interface (Exa-PAPI) project is developing a new C++ Performance API (PAPI++) software package from the ground up that offers a standard interface and methodology for using low-level performance counters in CPUs, GPUs, on/off-chip memory, interconnects, and the I/O system, including energy/power management. PAPI++ is building upon classic-PAPI functionality and aims to strengthen the path to exascale with a more efficient and flexible software design—one that takes advantage of C++’s object-oriented nature, preserves the low-overhead monitoring of performance counters, and adds a vast testing suite.

In addition to providing hardware counter-based information, a standardizing layer for monitoring software-defined events (SDE) is being incorporated that exposes the internal behavior of runtime systems and libraries, such as communication and math libraries, to the applications. As a result, the notion of performance events is broadened from strictly hardware-related events to include software-based information. Enabling monitoring of both hardware and software events provides more flexibility to developers when capturing performance information.

In summary, the Exa-PAPI++ team is preparing PAPI support to stand up to the challenges posed by exascale systems by:

1. Widening its applicability and providing robust support for exascale hardware resources;
2. Supporting finer-grain measurement and control of power, thus offering software developers a basic building block for dynamic application optimization under power constraints;
3. Extending PAPI to support SDEs; and
4. Applying semantic analysis to hardware counters so that the application developer can make sense of the ever-growing list of raw hardware performance events that can be measured during execution.

The team will be channeling the monitoring capabilities of hardware counters, power usage, and SDEs into a robust PAPI++ software package. PAPI++ is meant to be PAPI’s replacement—with a more flexible and sustainable software design.

RECENT PUBLICATIONS

Jagode, H., A. Danalis, H. Anzt, J. Dongarra
**PAPI Software-Defined Events for in-Depth Performance Analysis**

**Software-defined Events through PAPI**
24th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIP), in conjunction with 33rd IEEE International Parallel & Distributed Processing Symposium (IPDPS), May 20-24, 2019, Rio de Janeiro, Brazil, pp. 1-10, 2019.

**Counter Inspection Toolkit: Making Sense out of Hardware Performance Event**

FIND OUT MORE AT
https://icl.utk.edu/exa-papi

EXASCALE COMPUTING PROJECT
https://exascaleproject.org

SPONSORED BY

INNOVATIVE COMPUTING LABORATORY

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.