EXA-PAPI

The Exa-PAPI project is developing new performance counter monitoring capabilities as well as power management support for novel and advanced ECP hardware, and software technologies. Exa-PAPI builds upon classic-PAPI functionality and strengthens its path to exascale with 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.

ECP SCOPE

Exa-PAPI is preparing PAPI support to stand up to the challenges posed by exascale systems by:

GOAL 1 Widening its applicability and providing robust support for exascale hardware resources.

GOAL 2 Supporting finer-grain measurement and control of power, thus offering software developers a basic building block for dynamic application optimization under power constraints.

GOAL 3 Extending PAPI to support Software-Defined Events that originate from the ECP software stack and are treated as black boxes (e.g. communication and math libraries, runtime systems, etc.).

GOAL 4 Applying semantic analysis to hardware counters so that the application developer can better make sense of the ever-growing list of raw hardware performance events.

In summary, the Exa-PAPI team is channeling the monitoring capabilities of hardware counters, power usage, software-defined events into a robust PAPI software package for exascale-level systems.

PERFORMANCE COUNTER MONITORING CAPABILITIES

SUPPORTED ARCHITECTURES

AMD CPU: up to Zen3 Power for Fam17 GPU: MI50/60/100, power, temp	arm Cortex, Cavium ThunderX, ARM64, uncore	CRAY THE SUPERCOMPUTER COMPANY Gemini and Aries interconnect, power	IBM Blue Gene Series, Q: 5-D Torus, I/O System, EMON power, energy	IBM Power 5,6,7,8,9 Power monitoring support
IBM Power9 NEST event support via Performance Co-Pilot (PCP) PAPI component	intel Westmere, Sandy/Ivy Bridge, Haswell, Broadwell, KNC, KNL, KNM, Sky/Kaby/Cascade/Ice Lake	intel GPUs: Gen9	intel RAPL (power/energy), powercap, Linux	INFINIBAND TRADE ASSOCIATION
lustre	NVIDIA Tesla, Kepler, Maxwell, Pascal, Volta, Turing, Ampere	NVIDIA Power monitoring and capping support (NVML), NVLink	KVM Virtual Environment	vmware Virtual Environment

ECP PROJECTS AND 3RD PARTY TOOLS APPLYING PAPI

ECP DTE (PaRSEC) UTK http://icl.utk.edu/parsec/	**ECP LLNL-ATDM (Caliper)** LLVM github.com/LLNL/caliper-compiler	**ECP SNL-ATDM (Kokkos)** SNL https://github.com/kokkos	**ECP Proteas (TAU)** University of Oregon http://tau.uoregon.edu/
ECP HPCToolkit (HPCToolkit) Rice University http://hpctoolkit.org	**Score-P** http://score-p.org	**Vampir** TU Dresden http://www.vampir.eu/	**Scalasca** FZ Juelich, TU Darmstadt http://scalasca.org/
PerfSuite NCSA http://perfsuite.ncsa.uiuc.edu/	**Open|Speedshop** Open|SpeedShop https://openspeedshop.org/	**SvPablo** RENCI at UNC www.renci.org/research/pablo	**ompP** LMU Munich http://www.ompp-tool.com/

SOFTWARE-DEFINED EVENTS IN PAPI

KEY POINTS ABOUT SDEs

- New measurement possibilities:
  Tasks stolen, matrix residuals, partial results reached, arguments passed to functions
- Any tool can read PAPI SDEs:
  SDEs from a library can be read with PAPI_start()/PAPI_stop()/PAPI_read().
- Low overhead:
  Performance critical codes can implement SDEs with zero overhead.
- Easy to use, with rich feature set:
  Pull-mode & push-mode, read-write counters, sampling/overflowing, counters, groups, recordings, statistics, thread safety, custom callbacks

SUPPORT FOR GPUs: INTEL (AURORA EARLY ACCESS)

Added PAPI capabilities for monitoring Intel GPUs:

- GPUs hardware events, and
- Memory performance metrics (bytes read / written / transferred from / to L3).
- Monitoring of GPU and memory performance counters aids developers in producing more efficient code: by profiling the utilization of the latest GPU resources and diagnosing performance bottlenecks.

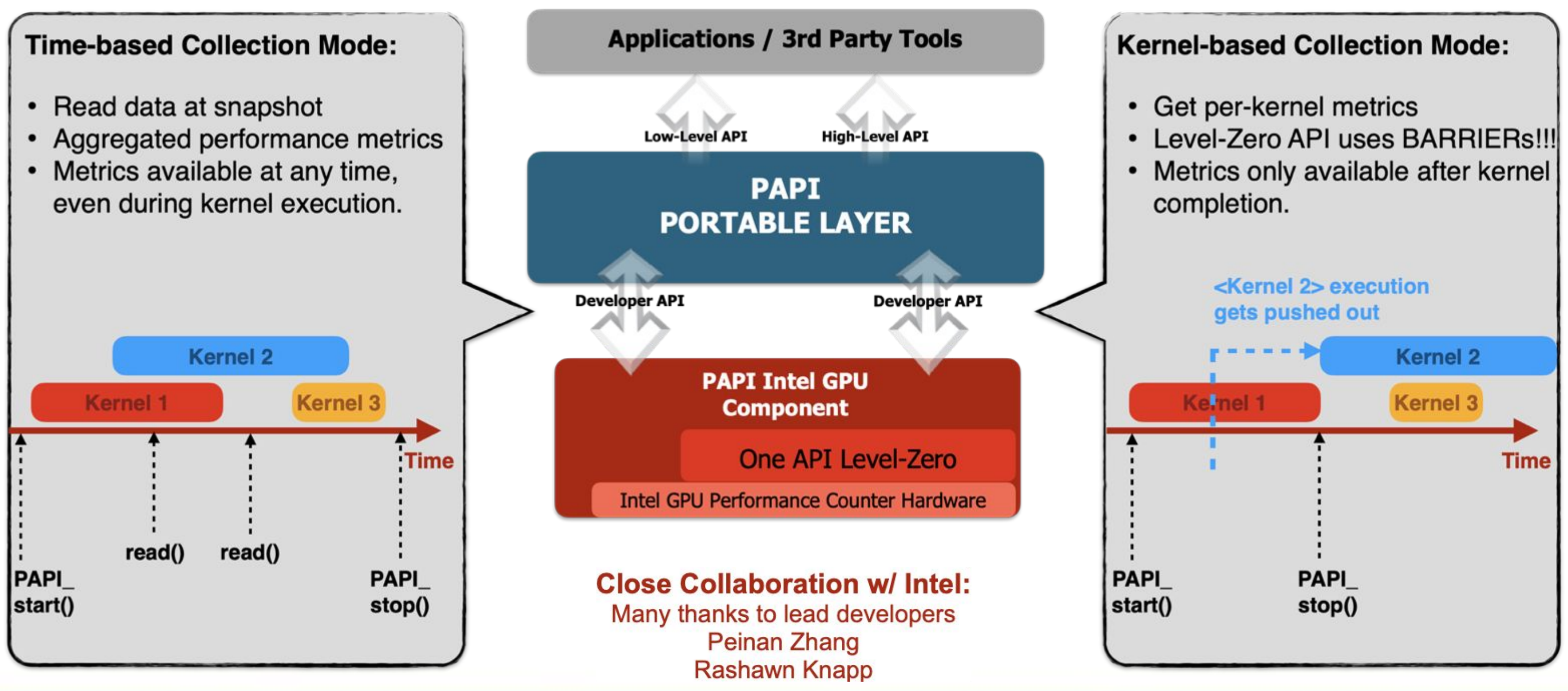
Scope and Objectives:

- A mechanism to collect Intel GPU performance metrics via PAPI's well known API

Approach:

- Enable through PAPI component framework
- Access Intel GPU performance metricsvia Intel One API Level-Zero Interface
- Tricky part: Two different collection modes via one PAPI component

oneAPI



Time-based Collection Mode:

- Read data at snapshot
- Aggregated performance metrics
- Metrics available at any time, even during kernel execution.

Kernel-based Collection Mode:

- Get per-kernel metrics
- Level-Zero API uses BARRIERs!!!
- Metrics only available after kernel completion.

Close Collaboration w/ Intel:
Many thanks to lead developers
Peinan Zhang
Rashawn Knapp

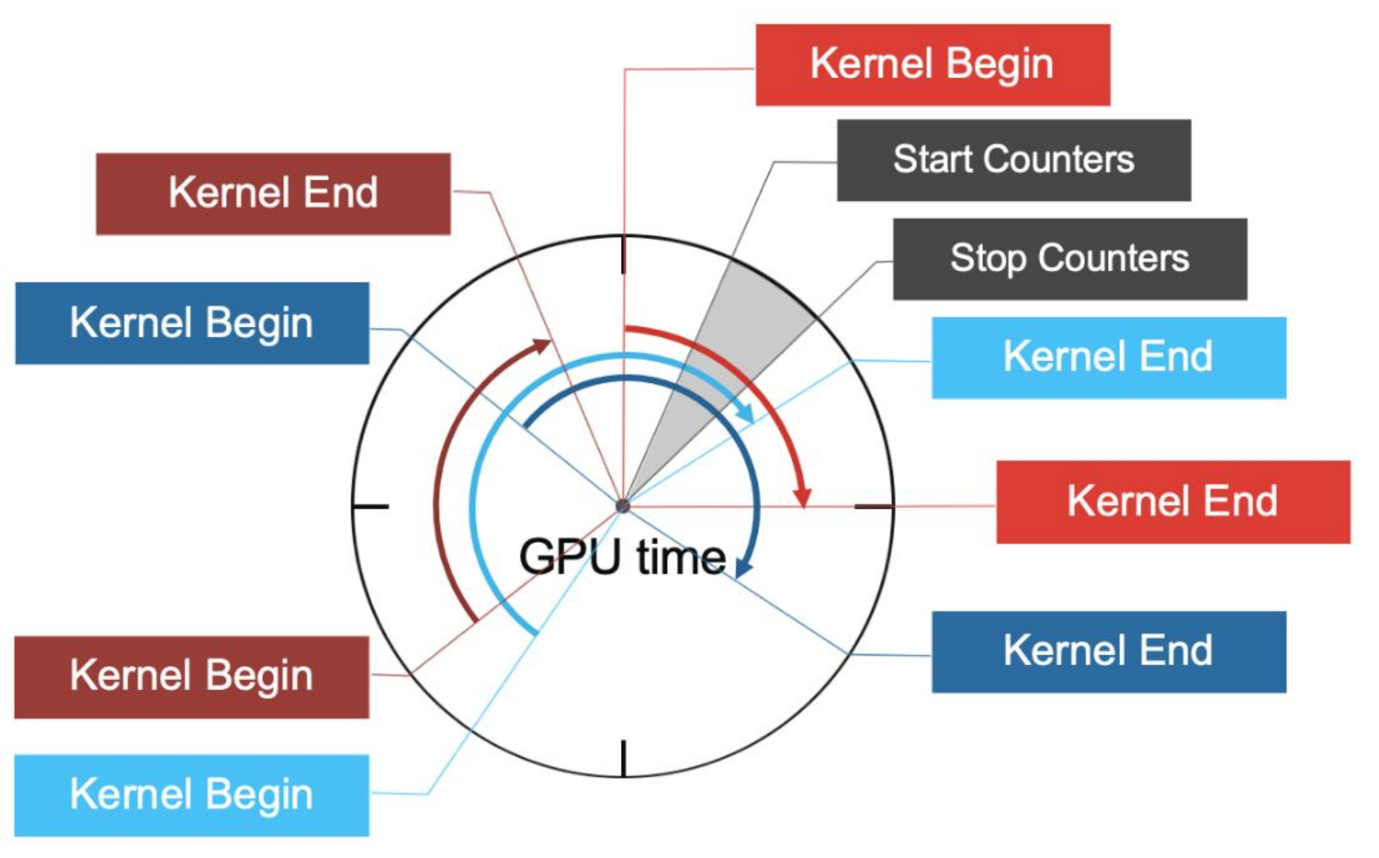
SUPPORT FOR GPUs: AMD (FRONTIER EARLY ACCESS)

Extended PAPI capabilities for monitoring AMD GPUs:

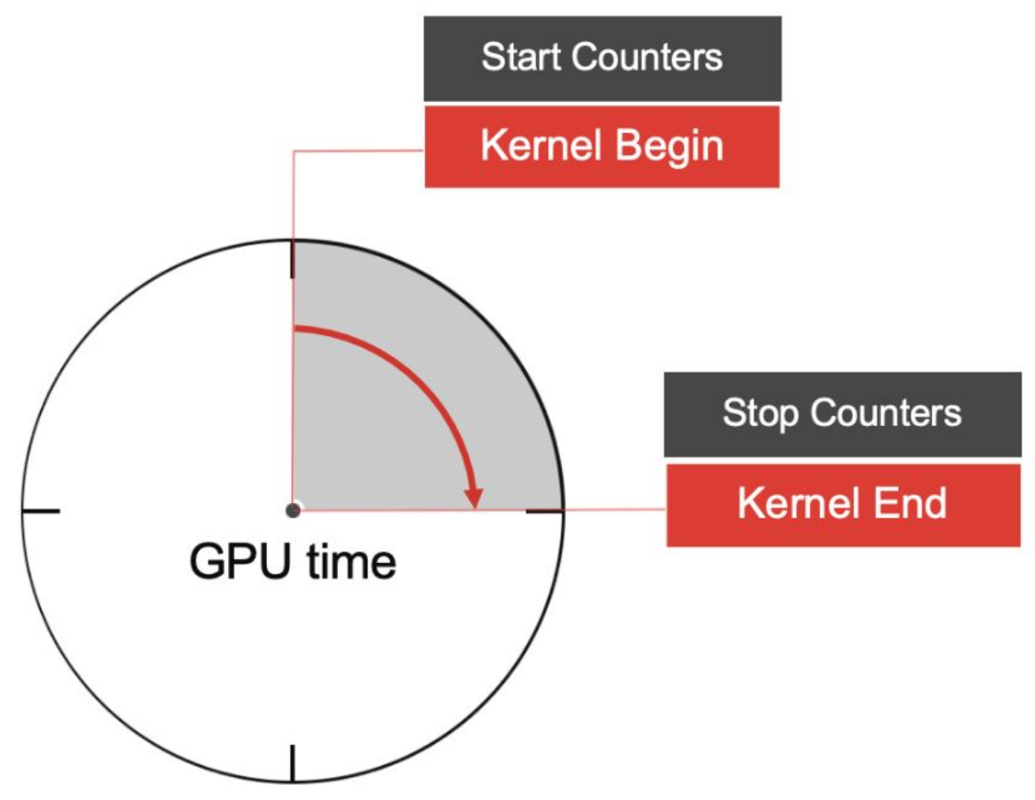
- Previous AMD GPU monitoring was limited to "Sampling mode" only.
- Latest PAPI rocm component supports both "Sampling" and "Intercept Mode".

→ Features and approach are equivalent to PAPI's Support for Intel GPUs
(see details on the left)

Sampling Mode



Intercept Mode



Multi-threaded Support for Sampling Mode:

- Events on multiple GPUs **can** be independently monitored by different threads
- Events on single GPU **cannot** be independently monitored by different threads

Example:

```
setenv("ROCP_HSA_INTERCEPT", "0", 0);
PAPI_library_init(...);
PAPI_thread_init(omp_get_thread_num);
#pragma omp parallel num_threads(3)
{
    PAPI_create_eventset(&eventset);
    PAPI_add_event(eventset, ...);
    PAPI_start(eventset);
    hipSetDevice(omp_get_thread_num());
       /* launch work on GPU */
    PAPI_stop(eventset, counters);
}
```

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.

ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

U.S. DEPARTMENT OF ENERGY | Office of Science

ECP EXASCALE COMPUTING PROJECT

NNSA National Nuclear Security Administration

SOFTWARE AVAILABLE AT
http://icl.utk.edu/exa-papi/