INNOVATIVE COMPUTING LABORATORY

NSF Awards #2311707 — #2311708 — #2311709

SPADE: Scalable Performance and Accuracy analysis for Distributed and Extreme-scale systems

PIs:
Heike Jagode (Lead) — UNIVERSITY OF TENNESSEE
Shirley Moore — UNIVERSITY OF TEXAS, EL PASO
Vince Weaver — UNIVERSITY OF MAINE

Co-PIs:
Anthony Danalis — UNIVERSITY OF TENNESSEE
Christoph Lauter — UNIVERSITY OF TEXAS, EL PASO

ABSTRACT

The SPADE project focuses on advancing monitoring, optimization, evaluation, and decision-making capabilities for extreme-scale systems. In Year 2, the team has focused on advancing several monitoring capabilities, with one key goal being the finalization of support for AMD's new RocProfiler SDK to enable hardware performance counter analysis on AMD APUs like the MI300A (used in El Capitan) and the upcoming MI350 series.

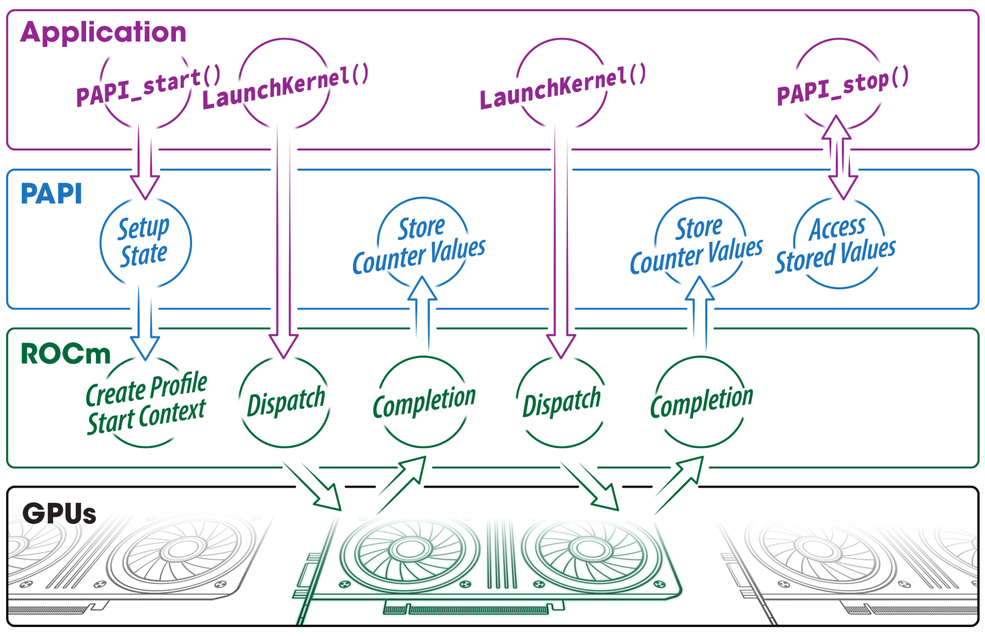
The SPADE team also continued extending the PAPI library to support heterogeneous CPUs. In Year 2, this work focused mainly on implementing and providing access to Intel's topdown Metrics---a structured interpretation of raw hardware performance counters---on supported heterogeneous Intel CPUs like Raptor Lake.

The SPADE team has also designed a compiler-based approach for detecting floating-point exceptions in GPU code. In addition, the team has investigated how using alternative floating-point formats and reduced-precision floating-point arithmetic affects the accuracy and performance of applications.

NEW APU/GPU MONITORING SUPPORT

PAPI SUPPORT FOR AMD APUs:

- Developed and finalized PAPI support for AMD's RocProfiler-SDK to enable performance counter analysis on AMD APUs, including MI300A (used in El Capitan) and the upcoming MI350 series.
- Delivered new PAPI component "**rocp_sdk**" through **three PAPI releases:**
  - **7.2.0 Beta1 (**August 2024)
  - **7.2.0 Beta2** (March 2025)
  - **7.2.0 Stable Release** (June 2025)
- PAPI "**rocp_sdk**" component supports **two monitoring modes**:
  - **Dispatch Mode:** Precise per-kernel counters, but no concurrency
  - **Sampling Mode:** Realistic execution, but shared counter values



PAPI PRESET SUPPORT FOR NVIDIA GPUs:

- Finalized development and integration of predefined events ("PAPI Presets") for GPU components, starting with NVIDIA GPUs (**introduced in PAPI 7.2.0**)
- **Goal of PAPI Presets for GPUs** is to address the challenge of up to 200,000 exposed native events per GPU, as seen on architectures like NVIDIA Hopper.
  → Simplifies access to these events via a more user-friendly, functional format.

Snippet from "`papi_avail`" Utility showing NVIDIA GPU Presets:

```
NAME                    DESCRIPTION
PAPI_CUDA_FP16_FMA      CUDA Half precision (FP16) FMA instructions
PAPI_CUDA_BF16_FMA      CUDA Half precision (BF16) FMA instructions
PAPI_CUDA_FP32_FMA      CUDA Single precision (FP32) FMA instructions
PAPI_CUDA_FP64_FMA      CUDA Double precision (FP64) FMA instructions
PAPI_CUDA_FP_FMA        CUDA floating-point FMA instructions
PAPI_CUDA_FP8_OPS       CUDA 8-bit precision floating-point operations
    :stat=sum
        Mandatory stat qualifier [avg, max, min, sum]
    :device=0
        Mandatory device qualifier [0]
```

NEW CPU MONITORING SUPPORT

SUPPORT FOR INTEL 'Top-Down' METRICs:

- Implemented Intel's top-down metrics via a new PAPI component **"topdown"**, which has been released as part of **PAPI 7.2.0 Stable** (June 2025).
- **Challenge:** Traditionally, using Intel's top-down metrics requires managing a large number of performance counters.
  - Although newer Intel processors (Ice Lake, Sapphire Rapids, and newer) can read up to 8 top-down metrics at once via the PERF_METRICS MSR, **configuring and interpreting these MSRs is complex** and requires deep knowledge of Intel's performance monitoring architecture.

Frontend Bound	Fetch Latency	iTLB Miss iCache Miss Branch Recovery Length Change Prefix Decoded iCache Microcode Sequencer
	Fetch Bandwidth	Legacy MITE pipeline Decode Stream Buffer Loop Stream Detector
Backend Bound	Core Bound	Divider Port Utilization
	Memory Bound	Stores L1 cache L2 cache L3 cache DRAM

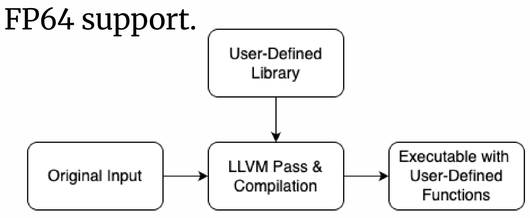
Retiring	Light Operations	Floating Point Integer Ops Memory Ops Fused Instructions Non-Fused Branches Nops
	Heavy Operations	Few Uop Instructions Microcode Sequencer
Bad Speculation	Branch Mispredicts	
	Machine Clears	

→ The **new PAPI "topdown" component** significantly simplifies this process by making these **metrics more accessible in a user-friendly format.**

Backend Bound	Frontend Bound	Bad Speculation	Retiring
Memory Bound	Fetch Latency	Branch Mispredicts	Heavy Operations

PERFORMANCE AND ACCURACY OF FLOATING-POINT COMPUTATIONS

- Implemented **compiler-based floating-point exception detection for GPUs** with the goal of integration with PAPI software-defined events (SDEs).
- Implemented **compiler-based transformation of FP64 to double-single representations** for platforms without FP64 support.
- Developed a tool to replace low-level functions with user-defined implementations in Libtorch applications.



- Investigated the impact of automatic mixed precision on performance and accuracy in GNNs for scientific machine learning.

Results from running 250 epochs on Pt Cluster dataset

	fp32 time	fp32 mem	fp32 MAE	AMP time	AMP mem	AMP MAE
CGCNN	467	215	0.17	496	212	0.18
MPNN	732	2923	0.14	698	1552	0.14
SchNet	509	210	0.18	551	209	0.18
MEGNet	902	336	0.18	961	217	0.16

- Publications
  - K. Ruiz-Rohena, C. Lauter, and S. Moore. Semi-Automatic Compile-Time Math Intrinsic Optimization Using LLVM. ASILOMAR 2025 Conference.
  - B. Cervantes and S. Moore. The Effect of Using Mixed Precision on Performance and Accuracy of GNN Workloads in Scientific ML. ASILOMAR 2025 Conference.

INNOVATIVE COMPUTING LABORATORY · THE UNIVERSITY OF TENNESSEE KNOXVILLE · NSF · UTEP · 1865 THE UNIVERSITY OF MAINE