PAPI Support for Specialized AI Architectures

Tokey Tahmid
Heike Jagode
ttahmid@icl.utk.edu
jagode@icl.utk.edu
University of Tennessee Knoxville
Knoxville, Tennessee, USA

The Performance Application Programming Interface (PAPI) [4] offers a universal interface for monitoring performance counters from diverse hardware and software components, including I/O and network traffic [1], memory activity, and computational throughput. One of PAPI's current main goals is to develop unique performance, data movement, and power monitoring capabilities for AI chips that are primarily designed for machine-learning training and AI workloads. The aim is to help researchers improve AI architecture efficiency and identify hardware-specific bottlenecks to achieve optimal system configuration.

The AI chips initially investigated in this work include Intel Habana Gaudi 2/3, Cerebras Wafer-Scale Engine (WSE-3), and SambaNova Reconfigurable Dataflow Unit (RDU). One of the unique challenges for performance monitoring on these architectures is the lack of traditional hardware counters and the availability of open-source performance APIs. Our initial investigations show:

- Intel Gaudi handles profiling by tracing hardware events (and internal counters) into a trace buffer during runtime. This buffer is later decoded and analyzed to extract performance insights. Because the event and counter data are stored in a trace buffer, current work involves exploring the Intel Gaudi APIs to determine how to access this data—and, more importantly, how frequently and efficiently it can be accessed. If low-latency access proves possible, it could provide a viable pathway for integrating Gaudi support into PAPI.
- Cerebras is optimized for data-parallel scaling and relies heavily on proprietary internal performance metrics and software-managed scheduling mechanisms. These findings suggest that, beyond pursuing close collaboration with Cerebras engineers to access internal statistics or traces, monitoring support for Cerebras in PAPI would need to follow a non-traditional path; likely involving high-level software hooks or API-based telemetry rather than direct hardware register access.
- SambaNova uses a dataflow execution model that requires rethinking performance monitoring from the ground up. Although RDUs lack traditional counters, SambaFlow's graph analysis and compiler passes offer profiling opportunities (e.g., execution time, memory use, resource utilization) through graph annotations or pipeline tracking. Future work could wrap SambaFlow's profiling APIs or graph hooks into PAPI, enabling model-level performance insights even without low-level counters.

These findings highlight the need to rethink performance monitoring for emerging AI architectures, where traditional hardware introspection is not feasible. Current work focuses on developing alternative approaches to hardware performance counters, which are often unavailable on many AI systems. One promising direction is the use of PAPI Software Defined Events (SDEs) [3], which provide a flexible way to capture and expose software-level metrics from within applications. This allows users to track events and gain meaningful performance insights—including data movement and I/O-related metrics—regardless of the hardware or architecture used by the application.

A proof-of-concept implementation provides a platform for engagement with AI hardware vendors. The goals are two-fold: (1) to demonstrate the value of having PAPI support available on their platforms, and (2) to encourage vendors to expose at least a small set of essential hardware counters for low-level performance monitoring. Achieving these goals is critical, as performance insights enable optimization of applications, improved efficiency, and a deeper understanding of complex workloads on modern AI systems. This effort lays the groundwork for making performance tools such as PAPI broadly accessible across AI architectures that currently lack traditional hardware counter interfaces.

Our proof-of-concept implementation uses the vendor-agnostic HPL-MxP [2] benchmark together with PAPI SDE to register custom events—such as `sde_io_read_bytes`, `sde_io_write_bytes`, `sde_float32` and `sde_float16`— to track memory and network I/O, as well as the use of different precisions of floating-point operations (FLOPs) throughout the workload. Relevant sections of HPL-MxP are instrumented with `PAPI_start()` and `PAPI_stop()`, and the application is executed across various input sizes to monitor data movement and changes in FLOP precision usage.

Validation was performed on three platforms: ARM Neoverse-V2, Intel Sapphire Rapids, and AMD MI300. The results showed close agreement between SDE and hardware counts for both data traffic and FLOPs, with a *mean ± standard deviation* difference of *0.310% ± 5.315%*. This provides confidence in applying the SDE approach to systems without accessible hardware performance counters. The method is portable across AI chips and offers a practical path for broadening profiling support for performance and I/O in modern AI systems.

Our ongoing work involves running the instrumented HPL-MxP on the Gaudi architecture using the SDE-based method, with future plans to extend coverage to additional AI architectures. By enabling PAPI support on these platforms, the objective is to provide the high-performance computing and AI communities with portable, reliable monitoring tools that facilitate system optimization, enable I/O and compute efficiency analysis, and strengthen reproducibility in emerging AI workloads.

References

[1] Daniel Barry, Heike Jagode, Anthony Danalis, and Jack Dongarra. 2023. Memory Traffic and Complete Application Profiling with PAPI Multi-Component Measurements. In *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 393–402. doi:10.1109/IPDPSW59300.2023.00070

[2] Azzam Haidar, Stanimire Tomov, Jack Dongarra, and Nicholas J. Higham. 2019. Harnessing GPU tensor cores for fast FP16 arithmetic to speed up mixed-precision iterative refinement solvers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC '18)*. IEEE Press, Dallas, Texas, Article 47, 11 pages. doi:10.1109/SC.2018.00050

[3] Heike Jagode, Anthony Danalis, Hartwig Anzt, and Jack Dongarra. 2019. PAPI software-defined events for in-depth performance analysis. *The International Journal of High Performance Computing Applications* 33, 6 (2019), 1113–1127. arXiv:https://doi.org/10.1177/1094342019846287 doi:10.1177/1094342019846287

[4] Heike Jagode, Anthony Danalis, Giuseppe Congiu, Daniel Barry, Anthony Castaldo, and Jack Dongarra. 2025. Advancements of PAPI for the exascale generation. *The International Journal of High Performance Computing Applications* 39, 2 (2025), 251–268. arXiv:https://doi.org/10.1177/10943420241303884 doi:10.1177/10943420241303884