PAPI Support for Specialized AI Architectures

Tokey Tahmid, Heike Jagode

pdsw 2025:
10th International
Parallel Data Systems Workshop

PAPI for Memory and Network I/O

PAPI Performance Application Programming Interface

appio, io, infiniband, lustre, mx

Goal: Develop PAPI support for AI chips designed for AI/ML workloads

SC25 St. Louis, MO hpc ignites.
pdsw 2025

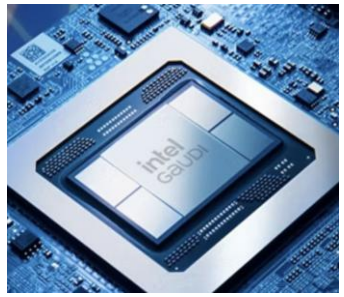
PAPI

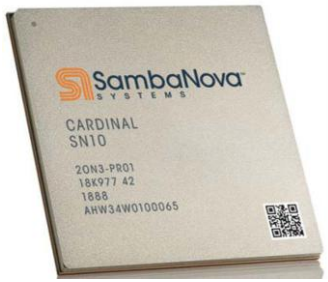
ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Challenges with Specialized AI Chips

- **Intel Gaudi:** Uses a trace-buffer–based profiling flow; exploring Gaudi Profiler APIs for accessing hardware counters
- **Cerebras:** Relies on proprietary metrics and software-managed scheduling; PAPI support via high-level software hooks or API-based telemetry
- **SambaNova:** Wrap SambaFlow's graph-level profiling APIs to expose execution, memory, and utilization metrics
- **Groq:** GroqView Profiler provides compile-time performance modeling





Images Source: intel.com, cerebras.ai, sambanova.ai, groq.com

SC25 St.Louis, MO hpc ignites.

pdsw 2025

PAPI

ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Alternative Approach with PAPI SDE

PAPI Software Defined Events (SDEs)

- Flexible way to capture and expose software-level metrics (data movement, I/O, FLOPs, etc.)
- Allows users to track events and gain meaningful performance insights
- Instrumented HPL-MxP with sde_io_read/write_bytes, sde_float32/float16
- PAPI_start()/PAPI_stop() to track SDEs
- Demonstrates how performance monitoring can be supported on AI hardware with PAPI

```
papi_sde_register_counter(sde_handle,"SDE_IO_READ_BYTES",
    PAPI_SDE_RO|PAPI_SDE_DELTA,PAPI_SDE_long_long,&sde_io_rb);
papi_sde_register_counter(sde_handle,"SDE_IO_WRITE_BYTES",
    PAPI_SDE_RO|PAPI_SDE_DELTA,PAPI_SDE_long_long,&sde_io_wb);

papi_sde_register_counter(sde_handle,"SDE_FLOAT16",
    PAPI_SDE_RO|PAPI_SDE_DELTA,PAPI_SDE_long_long,&sde_fp16);
papi_sde_register_counter(sde_handle,"SDE_FLOAT32",
    PAPI_SDE_RO|PAPI_SDE_DELTA,PAPI_SDE_long_long,&sde_fp32);
```

```
int EV = PAPI_NULL;
long long vals[4];

PAPI_add_named_event(EV, "sde:::HPL-MxP::SDE_IO_READ_BYTES");
PAPI_add_named_event(EV, "sde:::HPL-MxP::SDE_IO_WRITE_BYTES");
PAPI_add_named_event(EV, "sde:::HPL-MxP::SDE_FLOAT16");
PAPI_add_named_event(EV, "sde:::HPL-MxP::SDE_FLOAT32");

PAPI_start(EV);
io_test(io_bytes, "io_test.bin");
PAPI_stop(EV, vals);
```

SC25 St. Louis, MO hpc ignites.

pdsw 2025

PAPI

ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Validation Results

Architectures

- ARM Neoverse V2
- Sapphire Rapids (SPR)
- AMD MI300
- Intel Habana Gaudi

SC25 St. Louis, MO hpc ignites.
pdsw 2025

PAPI

ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Ongoing and Future Work

- Ongoing work includes exploring low-level hardware performance counters on Intel Gaudi to integrate into a PAPI gaudi component
- For future work we plan to extend coverage to additional AI architectures
- Objective is to provide the HPC and AI communities with portable, reliable monitoring tools in emerging AI workloads


ICL
INNOVATIVE
COMPUTING LABORATORY
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE