

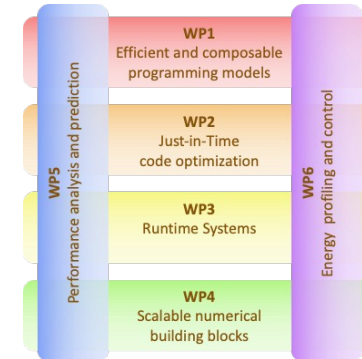
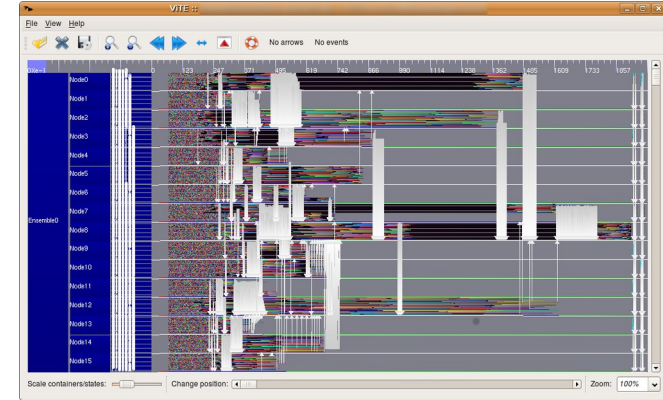
WP5 – Performance analysis and prediction

François Trahay
Télécom SudParis

Performance analysis at scale

Challenges

- Analyze performance of large-scale applications
 - Execution traces become huge (100+ GB)
 - Analysis tools must deal with these traces
 - > Need for suitable trace formats
- Visualize an execution trace
 - At a large scale there is too much information for visualization
 - Need to synthesize information
 - Performance analysis is like an investigation
 - Need for composable tools
- Integration in Exa-Soft
 - Collaboration with all the other building blocks of the software stack



Original workplan

- T5.1 Scalable tracing tool [**TSP**+Polaris]
 - Goal: design a new trace format for large-scale trace analysis
 - 1 PhD + 1 year engineer
- T5.2 System-wide post-mortem trace analysis [**Polaris**+Storm+Tadaam]
 - Collaboration with WP3
 - Goal: exploit the new trace format to devise new trace analysis techniques
 - 1 PhD + 1 year engineer
- T5.3 Fine-grained energy measurements [**TSP**+Storm]
 - Collaboration with WP5
 - Goal: collect energy consumption data at a μ s-scale granularity
 - 1 PhD
- T5.4 On-the-fly performance analysis that guides the runtime system [**Storm**+TSP+Polaris]
 - Goal: run performance analysis at during the execution, and give hints to the runtime system
 - 1 PhD

Need to reorganise the workplan

- Inria Polaris left the Exa-Soft project in 2024
- T5.1 Scalable tracing tool [TSP+Polaris]
 - Goal: design a new trace format for large-scale trace analysis
 - 1 PhD + 1 year engineer
- T5.2 System-wide post-mortem trace analysis [TSP+Polaris+Storm+Tadaam]
 - Goal: Develop new trace visualization techniques for large scale trace analysis
 - Collaboration with L. Schnorr (from Brasil)
 - 1 PhD + 1 year engineer
- T5.3 Fine-grained energy measurements [TSP+Storm]
 - Collaboration with WP5
 - Goal: collect energy consumption data at a μ s-scale granularity
 - 1 PhD
- T5.4 On-the-fly performance analysis that guides the runtime system [Tadaam+Topal+TSP+Polaris]
 - Collaboration with WP3
 - Goal: run performance analysis at during the execution, and give hints to the runtime system
 - 1 PhD

Progress of the workplan

- T5.1 Scalable tracing tool [TSP]
 - Goal: design a new trace format for large-scale trace analysis
 - 1 PhD (Catherine Guelque, 2023-2026)
 - 1 year engineer (to be hired in Dec. 2025)
- T5.2 System-wide post-mortem trace analysis [TSP+Storm+Tadaam]
 - Goal: Develop new trace visualization techniques for large scale trace analysis
 - Collaboration with L. Schnorr (from Brasil)
 - 1 PhD (to be hired in 2026)
 - 1 year engineer (to be hired in 2026)
- T5.3 Fine-grained energy measurements [TSP+Storm]
 - Collaboration with WP5
 - Goal: collect energy consumption data at a μ s-scale granularity
 - 1 PhD (Jules Risse, 2024-2027)
- T5.4 On-the-fly performance analysis that guides the runtime system [Tadaam+Topal]
 - Collaboration with WP3
 - Goal: run performance analysis at during the execution, and give hints to the runtime system
 - 1 PhD (to be hired in 2026)

T5.1 Pallas: a generic trace format for large HPC trace analysis

Catherine Guelque's PhD at Telecom SudParis



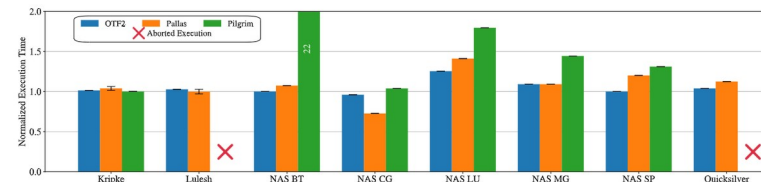
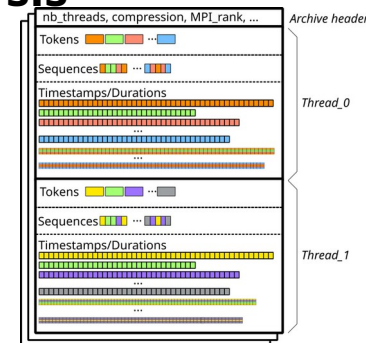
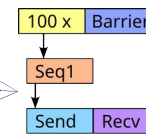
Pallas: structured trace format

- On the fly detection of sequences/loops
- Trace storage in a structured format
 - Separate metadata (eg sequences of events) from data (eg timestamp)
- Provide a generic way to store event
 - Implement the OTF2 API
 - Integration with existing tracing tools (EZTrace)
 - Able to trace the main parallel programming runtimes (MPI, OpenMP, CUDA, StarPU, Kokkos, ...)

Evaluation on Jean-Zay on 4096 cores

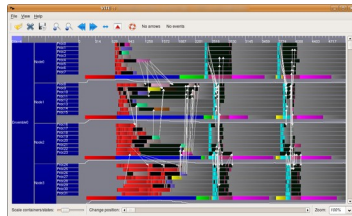
- EZTrace/OTF2 vs EZTrace/OTF2/Pallas
 - Low overhead for most applications
 - Almost instantaneous trace analysis

More details in Catherine's talk on Friday
Pallas: a generic trace format for large HPC trace analysis. IPDPS 2025



T5.2 Scalable trace visualization

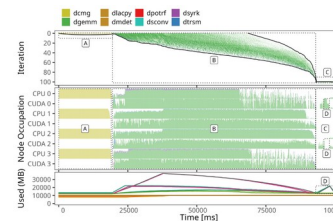
Several existing visualization tools



ViTE



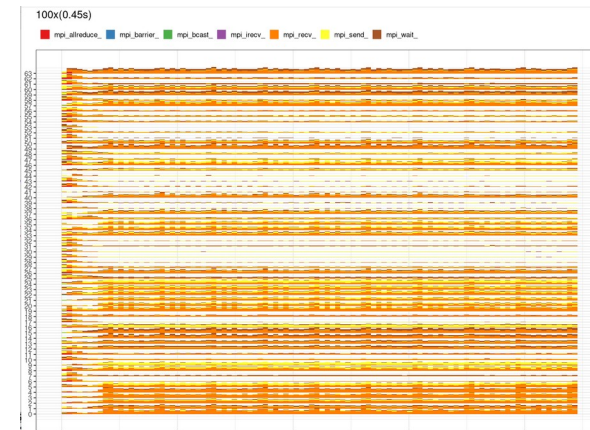
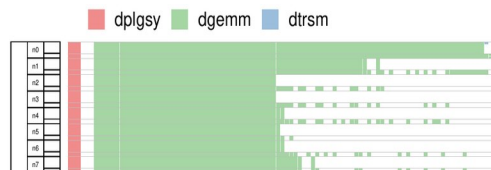
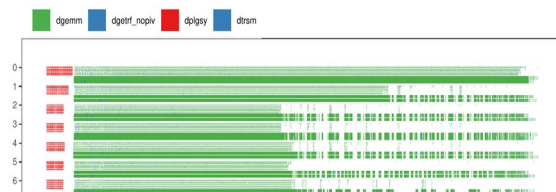
EasyPAP



StarVZ

Objective: leverage Pallas trace format to visualize large traces

- Composable visualization tool
 - Visualization of very large traces in a web browser
 - Python-based API to allow users to dig into performance data
- Spatio-temporal traces aggregation
 - Group similar behavior to synthesize information



Visualization of a Pallas trace

T5.3 Fine-grain energy measurement

Jules Risse's PhD at Inria/Telecom SudParis



- Energy measurement (eg RAPL, NVML) is coarse grain (typically 20 ms)
- StarPU tasks are fine-grain (typically 100 μ s)

Goal: model the energy consumption of tasks

- Log tasks execution on each measurement interval
- Build an overdetermined linear system
- Model the energy consumption of a task running on a device
- Will be used in an energy-aware scheduling policy in StarPU

More details in Jules' talk on Friday
Fine-grain energy consumption modeling of HPC task-based programs. Cluster 2025

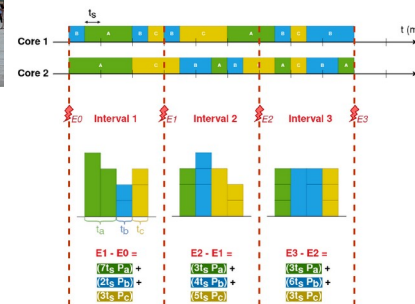
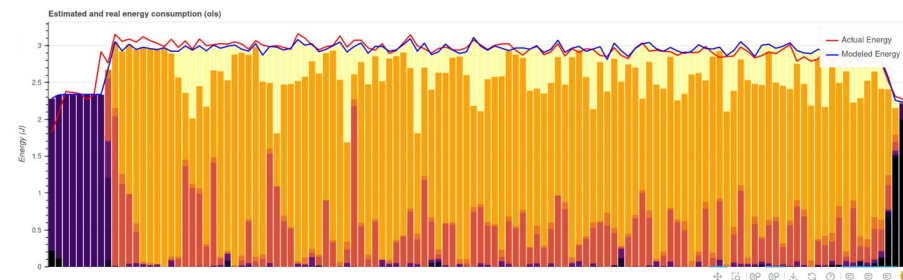


Figure – Linear system generation

$$\begin{pmatrix} 7t_s & 2t_s & 3t_s \\ 3t_s & 4t_s & 5t_s \\ 3t_s & 6t_s & 3t_s \end{pmatrix} \begin{pmatrix} P_a \\ P_b \\ P_c \end{pmatrix} = \begin{pmatrix} E_1 - E_0 \\ E_2 - E_1 \\ E_3 - E_2 \end{pmatrix} \quad (1)$$



Conclusion

- Progress of the workplan
 - T5.1 Scalable tracing tool: **on time**
 - T5.2 System-wide post-mortem trace analysis: **hiring delayed, but ongoing preliminary work**
 - T5.3 Fine-grained energy measurements: **on time**
 - T5.4 On-the-fly performance analysis that guides the runtime system: **hiring delayed**
- Status of the software stack
 - EZTrace/ ViTE: can be used in production
 - Already tested with StarPU (WP3), Chameleon (WP4), and energy-reader (WP6)
 - Pallas: prototype
 - Experiments on benchmarks with StarPU and Chameleon
- Next steps
 - Visualization of large traces
 - On-the-fly performance analysis
 - Collaboration with the other workpackages
 - Experiments on application demonstrators, and real-life applications (Gysela)