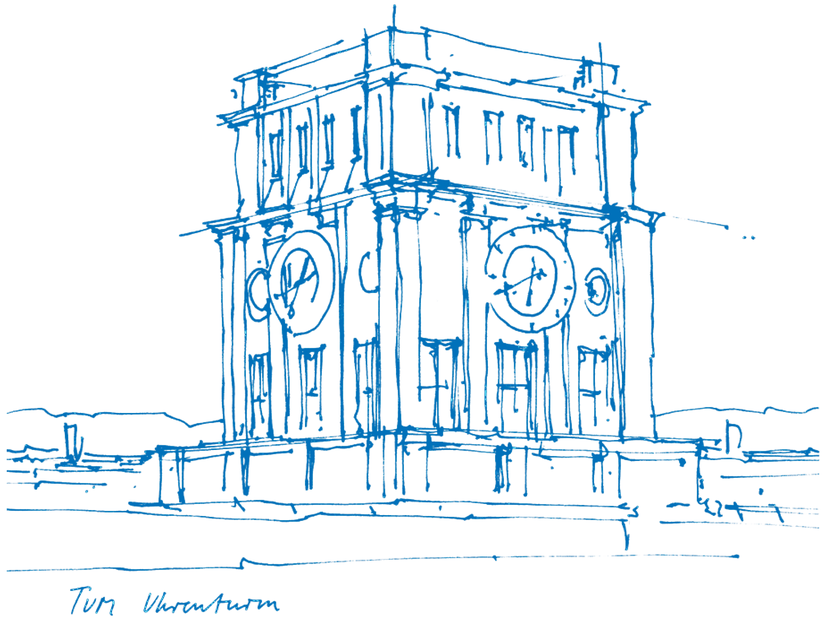
Understanding Heterogeneous Systems Through Topology-Aware Tools

Stepan Vanecek
stepan.vanecek@tum.de

Chair of Computer Architecture and Parallel Systems
Technical University of Munich

HPCSE 2026, 21st May 2026



Motivation

Motivation

1. Systems are increasingly complex
   - Heterogeneous
   - Dynamic, configurable

Motivation

1 Systems are increasingly complex

- Heterogeneous
- Dynamic, configurable

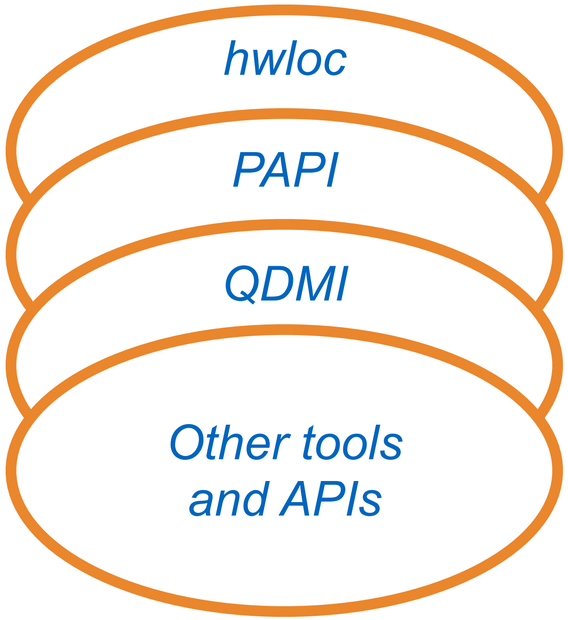
2 Memories play a crucial role

- More parallelism in compute
  - ➡ Higher demand for data
- More complex memory hierarchies
  - ➡ More difficult to understand and tune performance
- More heterogeneous systems, chips, memories
  - ➡ Portability issues

Toolchain Overview

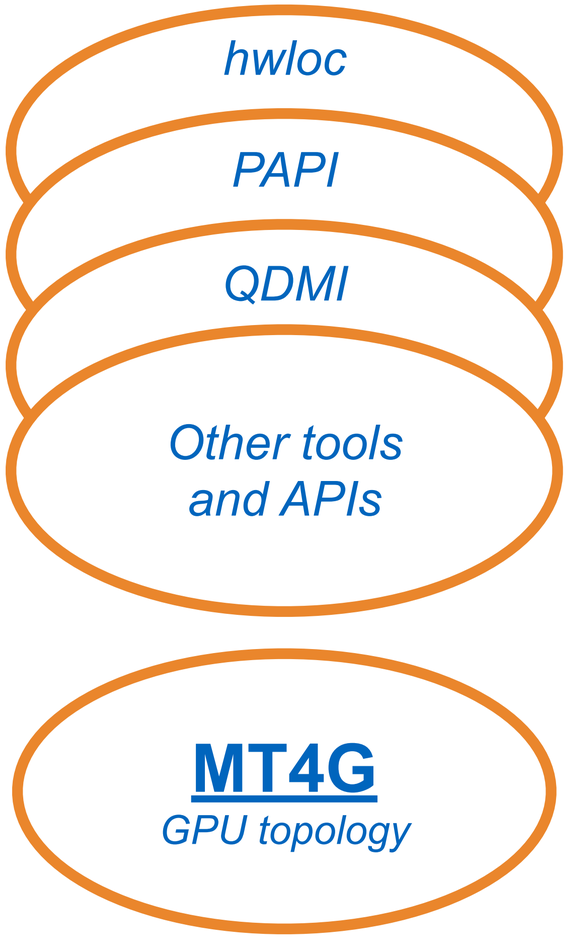
Toolchain Overview

Providing
topology
information



Toolchain Overview

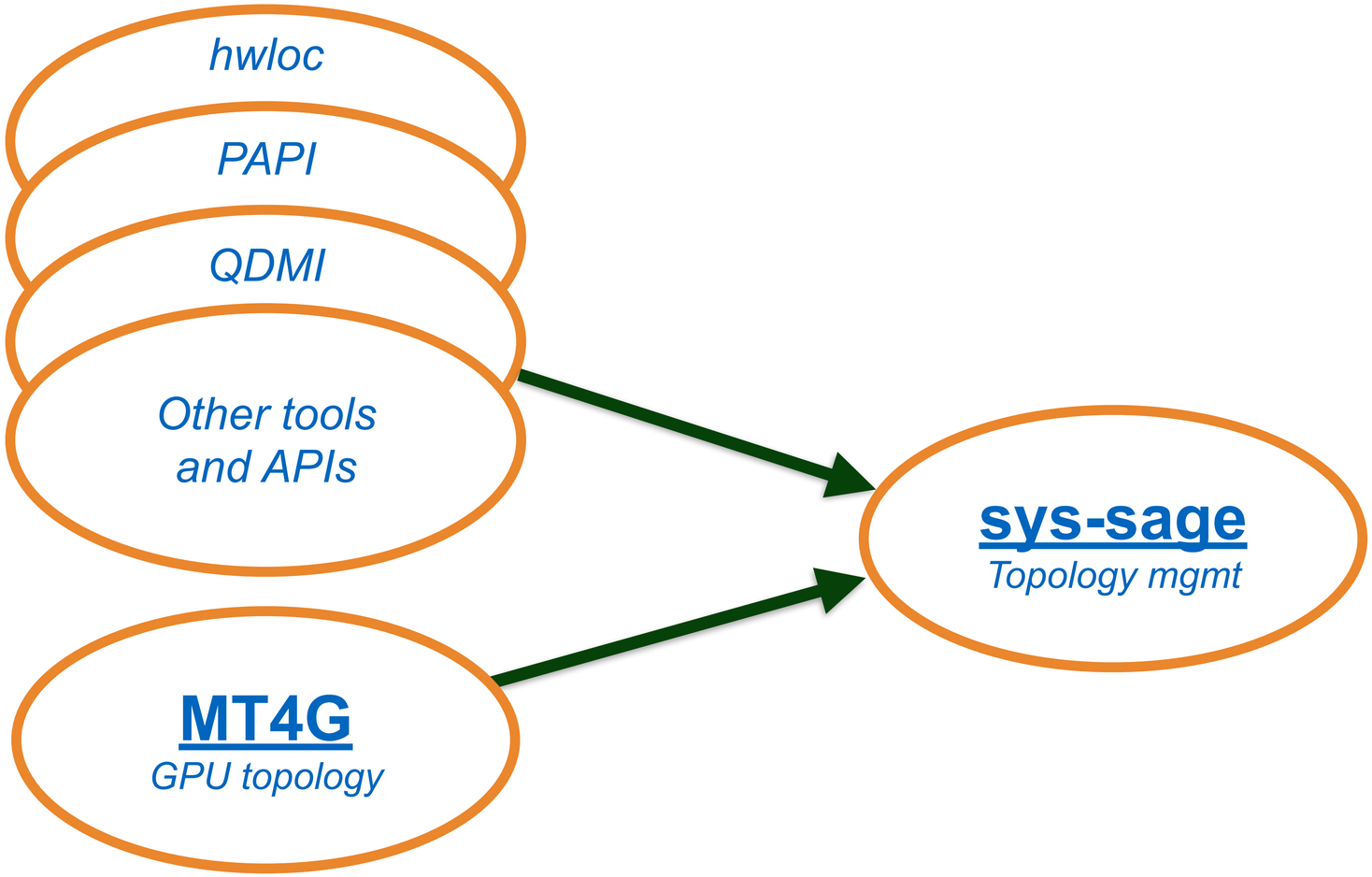
Providing
topology
information



Toolchain Overview

Providing
topology
information

Managing
topology
information

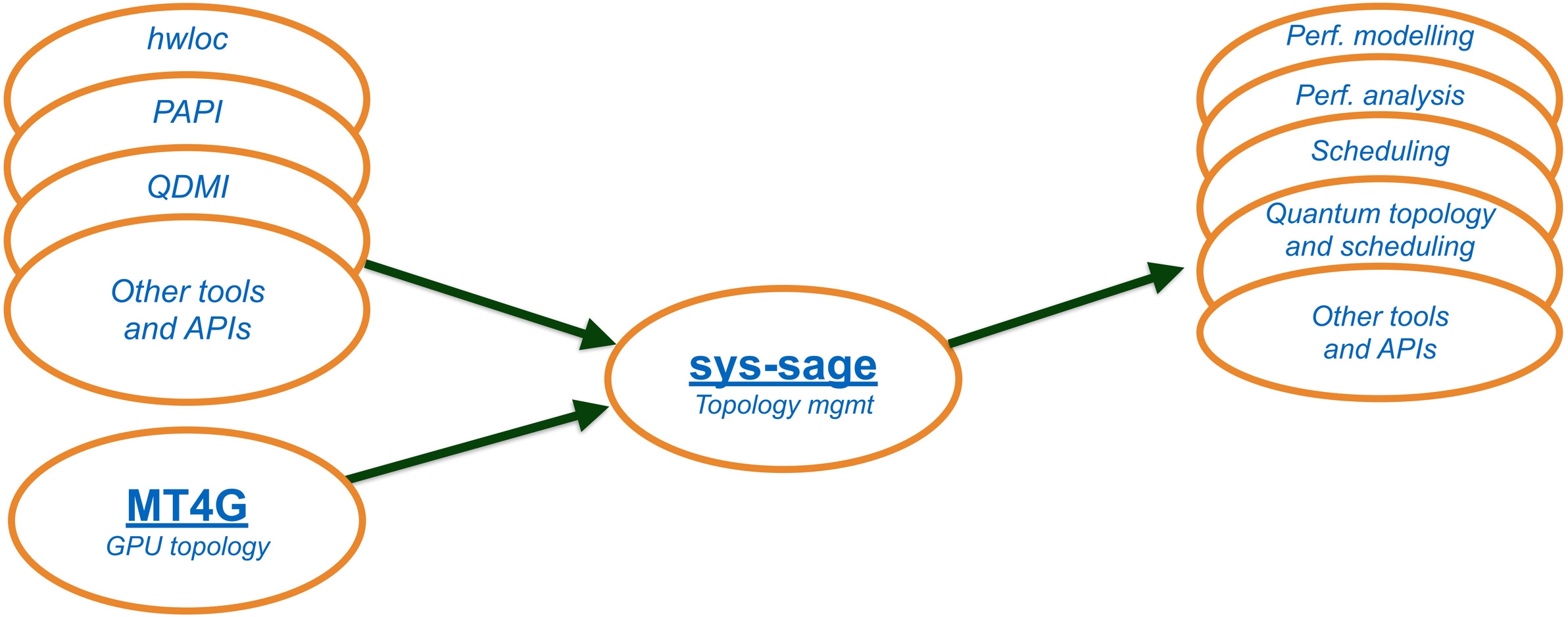


Toolchain Overview

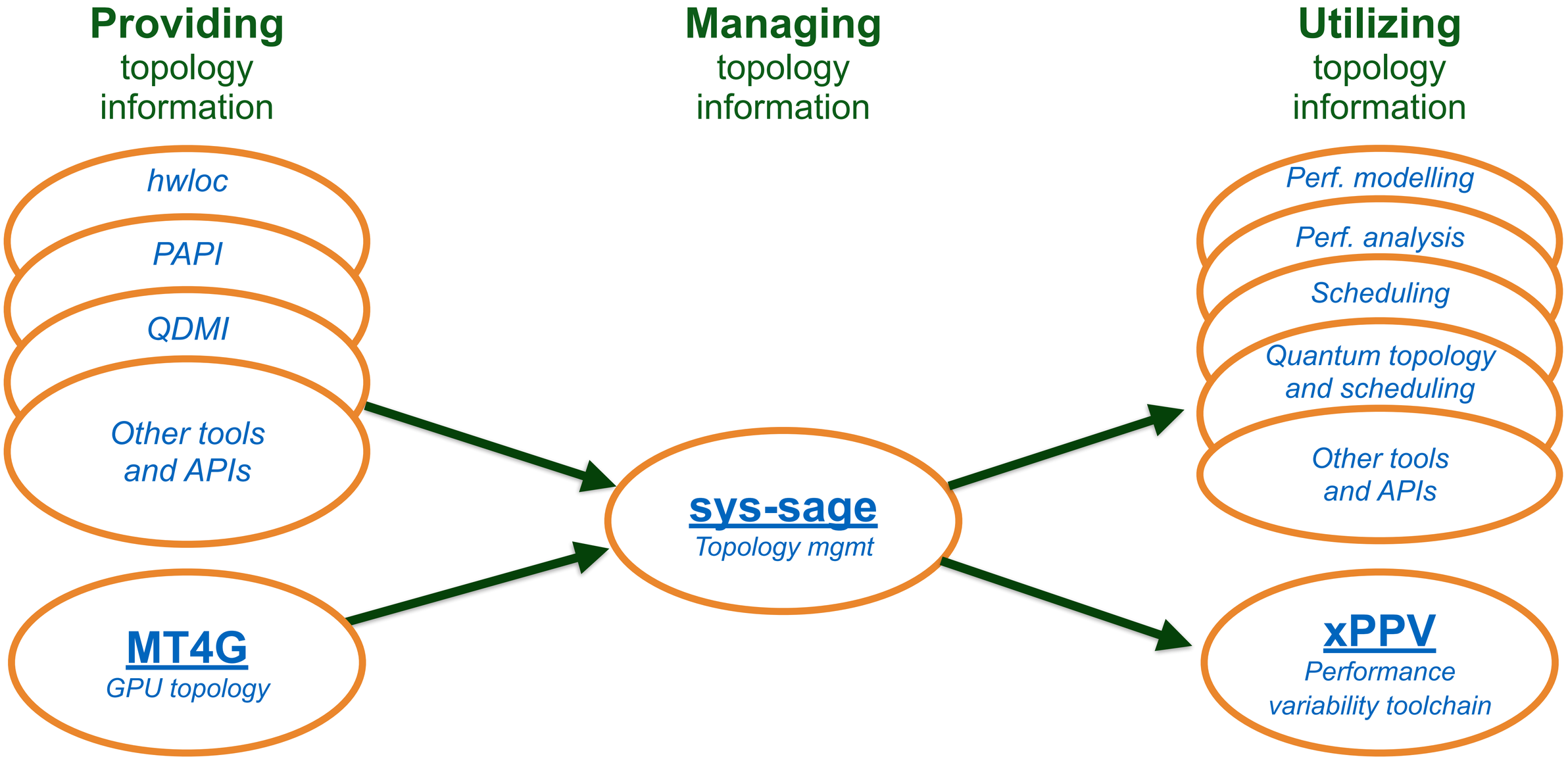
Providing
topology
information

Managing
topology
information

Utilizing
topology
information



Toolchain Overview



MT4G — Motivation

Vanecek, Mußbacher, Größler, Saroliya, Schulz. 2025. MT4G: A Tool for Reliable Auto-Discovery of NVIDIA and AMD GPU Compute and Memory Topologies. SC ProTools '25 https://doi.org/10.1145/3731599.3767518

MT4G — Motivation

CPUs

Tools & Interfaces for discovery:

- HWloc, LIKWID, lscpu, …

Providing valuable Information:

- Number of Cores, hyperthreading
- NUMAs, sockets, ..
- Cache hierarchy & properties (size, $lines, ..)
- Main memory (incl. sizes)

Vanecek, Mußbacher, Größler, Saroliya, Schulz. 2025. MT4G: A Tool for Reliable Auto-Discovery of NVIDIA and AMD GPU Compute and Memory Topologies. SC ProTools '25 https://doi.org/10.1145/3731599.3767518

MT4G — Motivation

CPUs

Tools & Interfaces for discovery:

- HWloc, LIKWID, lscpu, …

Providing valuable Information:

- Number of Cores, hyperthreading
- NUMAs, sockets, ..
- Cache hierarchy & properties (size, $lines, ..)
- Main memory (incl. sizes)

GPUs

Tools & Interfaces for discovery:

- —lack ot standardized ones—
- nvidia-smi, rocm-smi, hipdeviceproperties, ..

Scattered information:

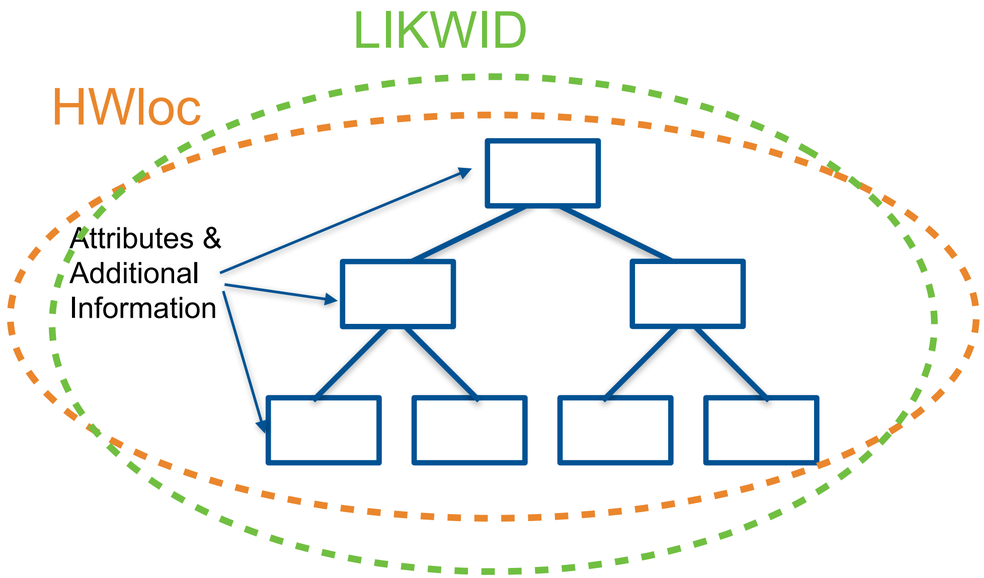
- nvidia-smi: GPU name, memory size
- cuda/hipdeviceprop: L2 cache size
- nvml: GPU configuration

Vanecek, Mußbacher, Größler, Saroliya, Schulz. 2025. MT4G: A Tool for Reliable Auto-Discovery of NVIDIA and AMD GPU Compute and Memory Topologies. SC ProTools '25 https://doi.org/10.1145/3731599.3767518

MT4G — Motivation

CPUs

Complete, cross-vendor



GPUs

Tools & Interfaces for discovery:

- —lack ot standardized ones—
- nvidia-smi, rocm-smi, hipdeviceproperties, ..

Scattered information:

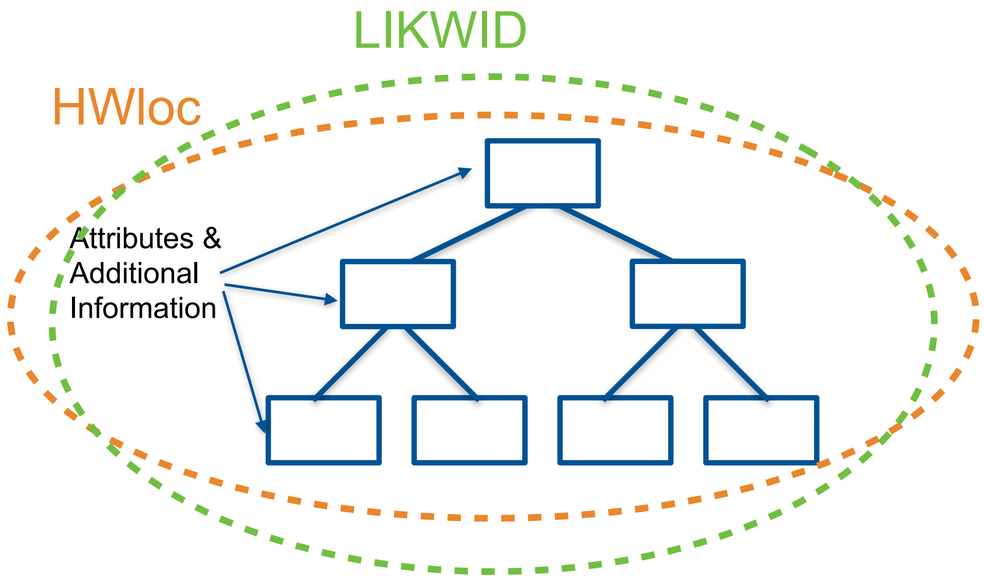
- nvidia-smi: GPU name, memory size
- cuda/hipdeviceprop: L2 cache size
- nvml: GPU configuration

Vanecek, Mußbacher, Größler, Saroliya, Schulz. 2025. MT4G: A Tool for Reliable Auto-Discovery of NVIDIA and AMD GPU Compute and Memory Topologies. SC ProTools '25 https://doi.org/10.1145/3731599.3767518

MT4G — Motivation

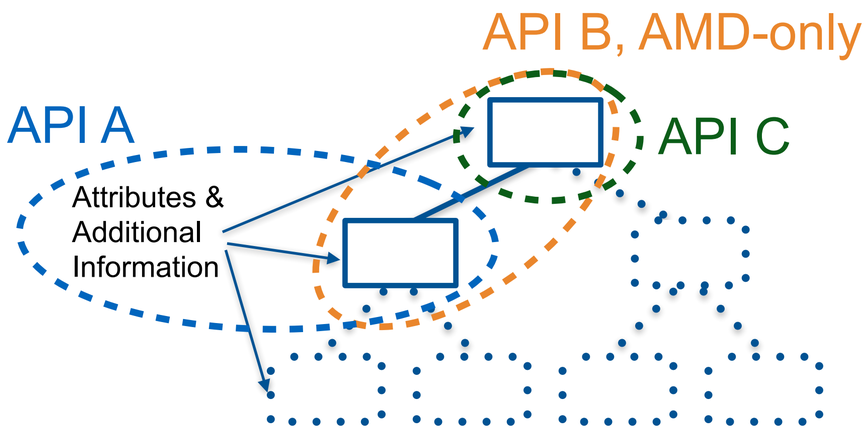
CPUs

Complete, cross-vendor



GPUs

Incomplete, scattered, vendor-specific

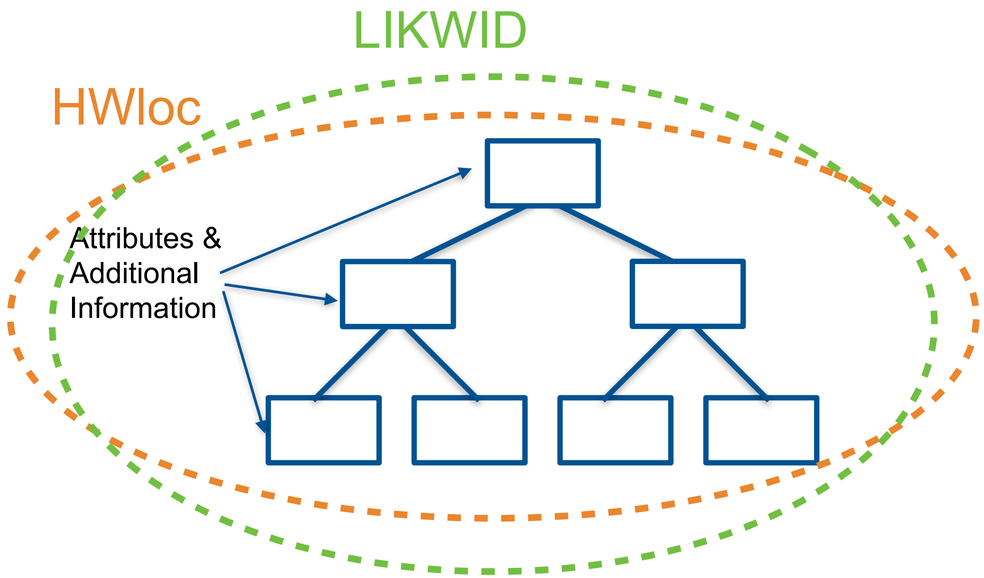


Vanecek, Mußbacher, Größler, Saroliya, Schulz. 2025. MT4G: A Tool for Reliable Auto-Discovery of NVIDIA and AMD GPU Compute and Memory Topologies. SC ProTools '25 https://doi.org/10.1145/3731599.3767518

MT4G — Motivation

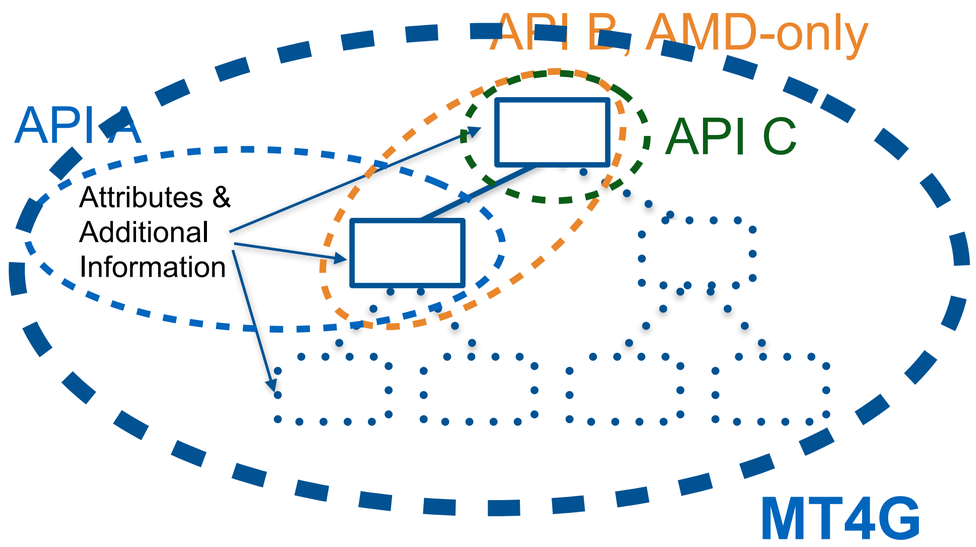
CPUs

Complete, cross-vendor



GPUs

Incomplete, scattered, vendor-specific



Vanecek, Mußbacher, Größler, Saroliya, Schulz. 2025. MT4G: A Tool for Reliable Auto-Discovery of NVIDIA and AMD GPU Compute and Memory Topologies. SC ProTools '25 https://doi.org/10.1145/3731599.3767518

MT4G — Reverse Engineering GPU Topologies

MT4G — Reverse Engineering GPU Topologies

What does MT4G provide?

Report on GPU building blocks and their attributes and capabilities

- NVIDIA (Pascal and newer)
- AMD (CDNA)

MT4G — Reverse Engineering GPU Topologies

What does MT4G provide?

Report on GPU building blocks and their attributes and capabilities

- NVIDIA (Pascal and newer)
- AMD (CDNA)

Target Memory Elements

NVIDIA

- L1 cache
- L2 cache
- Texture & Readonly cache
- Constant L1, L1.5 cache
- Shared memory
- Device memory

AMD

- L1 cache
- L1s cache
- L2 cache
- L3 cache (partially)
- Local Data Share
- Device memory

MT4G — Reverse Engineering GPU Topologies

What does MT4G provide?

Report on GPU building blocks and their attributes and capabilities

- NVIDIA (Pascal and newer)
- AMD (CDNA)

Target Memory Elements

NVIDIA

- L1 cache
- L2 cache
- Texture & Readonly cache
- Constant L1, L1.5 cache
- Shared memory
- Device memory

AMD

- L1 cache
- L1s cache
- L2 cache
- L3 cache (partially)
- Local Data Share
- Device memory

Queried properties

- Size
- Cache Line Size ($-only)
- Fetch Granularity ($-only)
- Physical Layout
- Load Latency
- Read & Write Bandwidth (L2 cache and above)

\+ Compute Resource Topological Information

MT4G — Attribute Coverage

Memory Element	Size		LD lat		R&W BW		cache line size		fetch granularity		Amount per SM/GPU		Physical hardware sharing	
Vendor	*NV*	*AMD*	*NV*	*AMD*	*NV*	*AMD*	*NV*	*AMD*	*NV*	*AMD*	*NV*	*AMD*	*NV*	*AMD*
(v) L1 cache	✅	✅	✅	✅	SOON	SOON	✅	✅	✅	✅	✅	✅	✅	
sL1 cache		✅		✅		SOON		✅		✅				✅
L2 cache	☑	☑	✅	✅	✅	✅	✅	☑	✅	✅	✅	☑		
L3 cache		☑		SOON		✅		☑		SOON		☑		
Texture cache	✅		✅		SOON		✅		✅		✅		✅	
Readonly cache	✅		✅		SOON		✅		✅		✅		✅	
Constant L1 cache	✅		✅		SOON		✅		✅		✅		✅	
Constant L1.5 cache	✅		✅		SOON		❌		✅		❌			
Shared Mem. / LDS	☑	☑	✅	✅	SOON	SOON								
Device Memory	☑	☑	✅	✅	✅	✅								

✅ = Available via microbenchmark; ☑ = Available via API; SOON = Work-in-progress; ❌ = not available

sys-sage — Motivation

Vanecek and Schulz. 2024. sys-sage: A Unified Representation of Dynamic Topologies & Attributes on HPC Systems. ICS'24
https://doi.org/10.1145/3650200.3656627

sys-sage — Motivation

? Current landscape of topological information

- Multiple tools (such as hwloc) provide static topological snapshot
- Increased need for dynamic/runtime data (interfaces also exist)
- Need to integrate multiple static and dynamic contexts ➡ manual integration

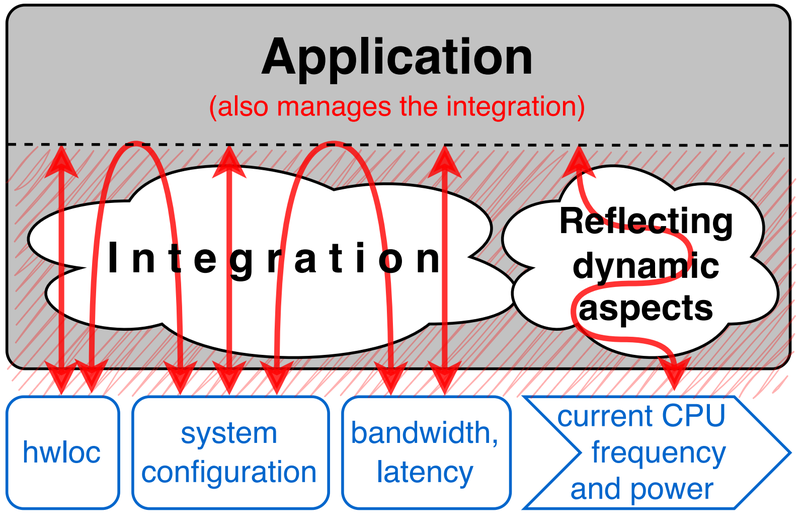
Vanecek and Schulz. 2024. sys-sage: A Unified Representation of Dynamic Topologies & Attributes on HPC Systems. ICS'24
https://doi.org/10.1145/3650200.3656627

sys-sage — Motivation

? Current landscape of topological information

- Multiple tools (such as hwloc) provide static topological snapshot
- Increased need for dynamic/runtime data (interfaces also exist)
- Need to integrate multiple static and dynamic contexts ➡ manual integration

Traditional approach

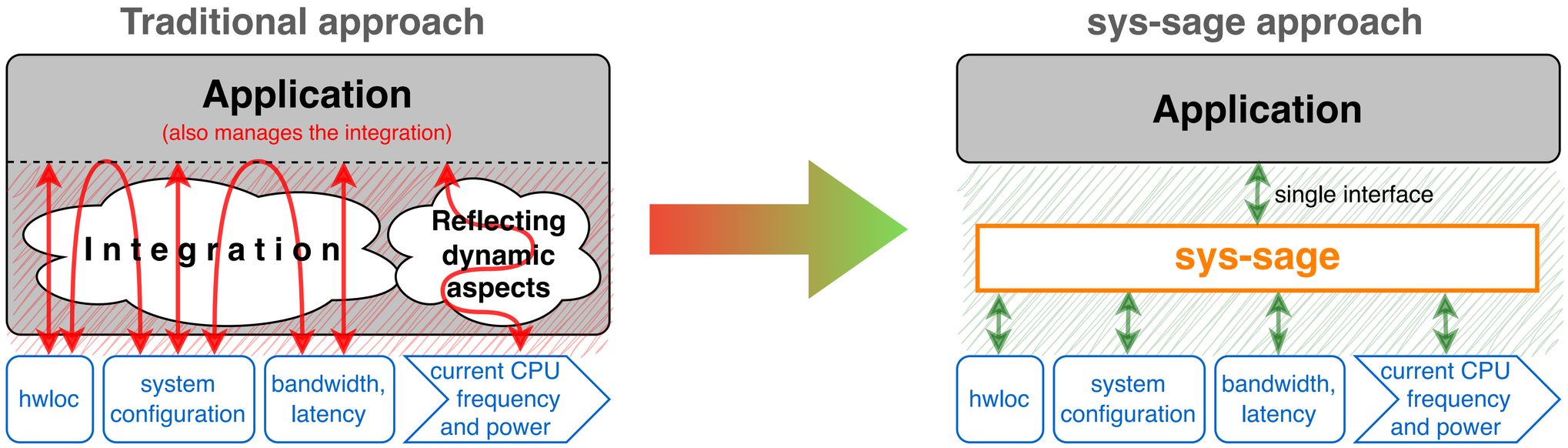


Vanecek and Schulz. 2024. sys-sage: A Unified Representation of Dynamic Topologies & Attributes on HPC Systems. ICS'24 https://doi.org/10.1145/3650200.3656627

sys-sage — Motivation

? Current landscape of topological information

- Multiple tools (such as hwloc) provide static topological snapshot
- Increased need for dynamic/runtime data (interfaces also exist)
- Need to integrate multiple static and dynamic contexts ➡ manual integration



Vanecek and Schulz. 2024. sys-sage: A Unified Representation of Dynamic Topologies & Attributes on HPC Systems. ICS'24
https://doi.org/10.1145/3650200.3656627

sys-sage

sys-sage

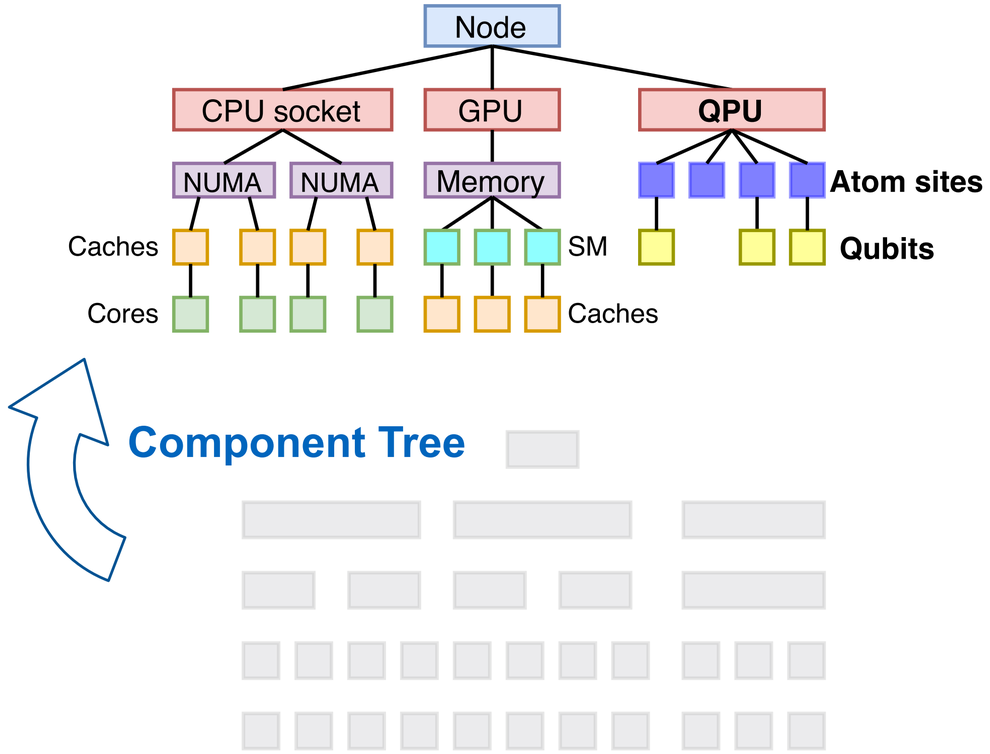
Internal Data Representation

sys-sage

Internal Data Representation

1. Components

- Hierarchical representation (hwloc-like)
- Simple to understand, Mandatory
- Examples: Static CPU/GPU/QPU HW topologies



sys-sage

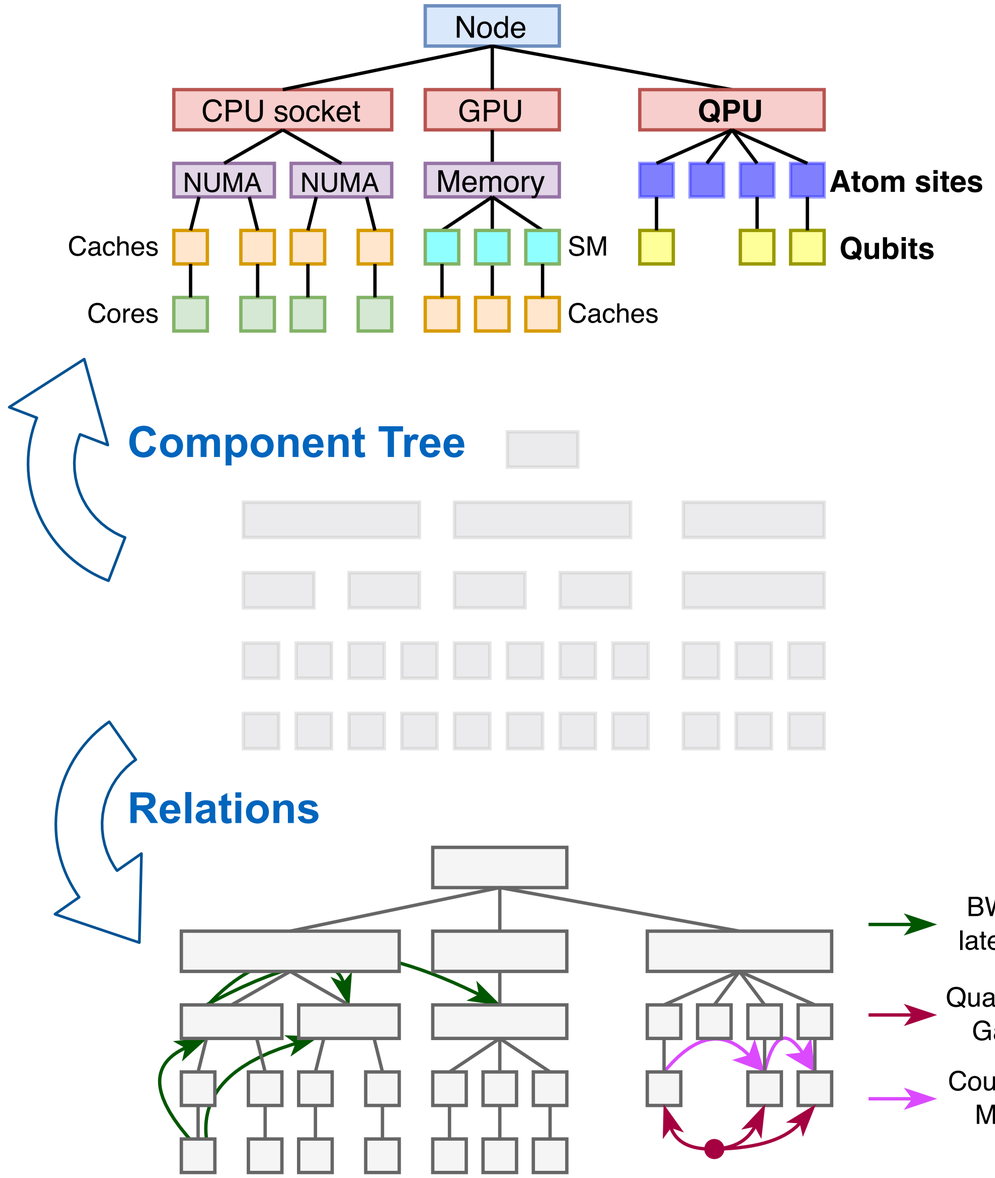
Internal Data Representation

1. Components

- Hierarchical representation (hwloc-like)
- Simple to understand, Mandatory
- Examples: Static CPU/GPU/QPU HW topologies

2. Relations

- Any Relation of 1 or more Components
- Orthogonal to the Component Tree
- More dynamic, complex information
- Examples: data exchange rates between components, dynamic settings, HW counter readings



sys-sage

Mishra, Vanecek, Echavarria, Deng, Mete, L.Schulz, M.Schulz. *Towards a Unified Architectural Representation in HPCQC: Extending Sys-Sage for Quantum Technologies*. ISC High Performance 2025
https://doi.org/10.23919/ISC.2025.11017506

sys-sage

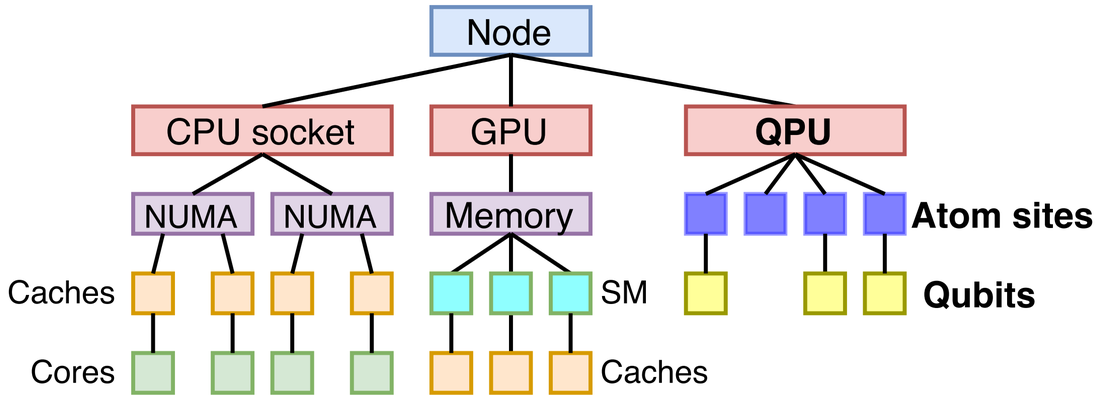
Quantum Extension

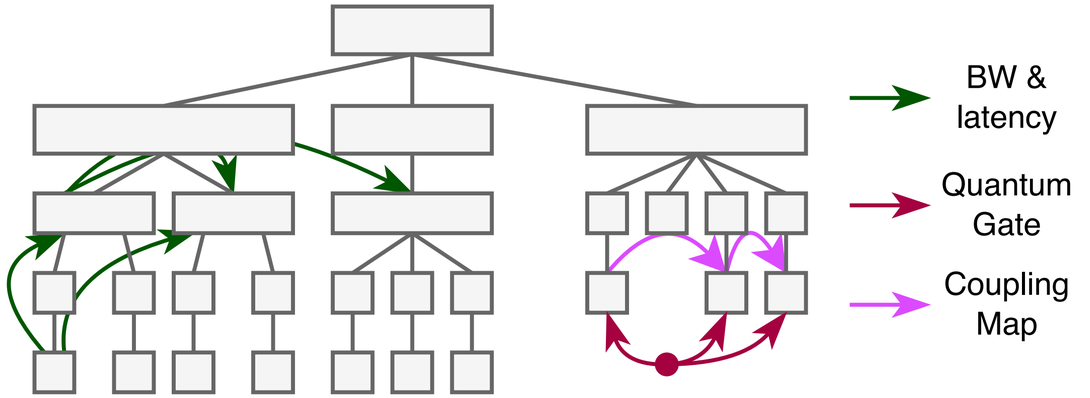
- Quantum Processing Units (QPUs) begin to accompany CPUs and GPUs in HPC systems
- sys-sage was extended to support QPUs alongside CPUs/GPUs
- Possible to retrieve information from existing APIs, such as QDMI or Qiskit runtime

Mishra, Vanecek, Echavarria, Deng, Mete, L.Schulz, M.Schulz. *Towards a Unified Architectural Representation in HPCQC: Extending Sys-Sage for Quantum Technologies*. ISC High Performance 2025 https://doi.org/10.23919/ISC.2025.11017506

Quantum Extension

- Quantum Processing Units (QPUs) begin to accompany CPUs and GPUs in HPC systems
- sys-sage was extended to support QPUs alongside CPUs/GPUs
- Possible to retrieve information from existing APIs, such as QDMI or Qiskit runtime



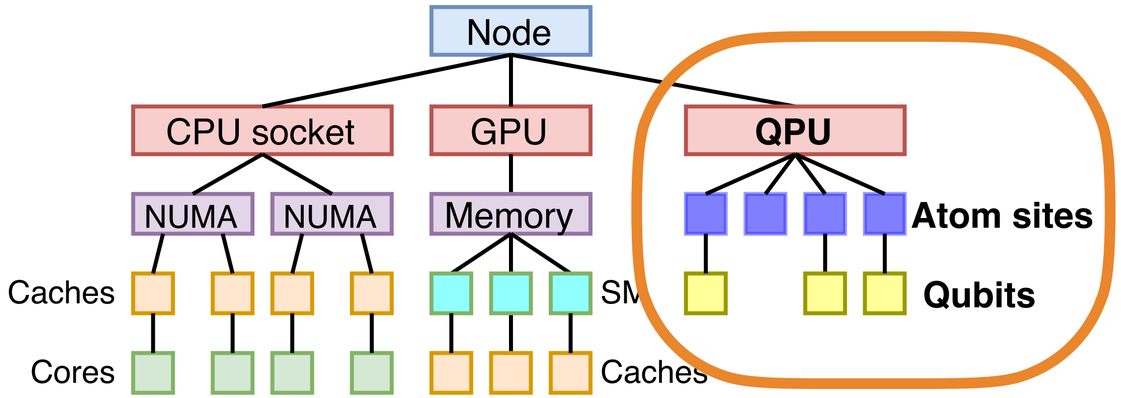


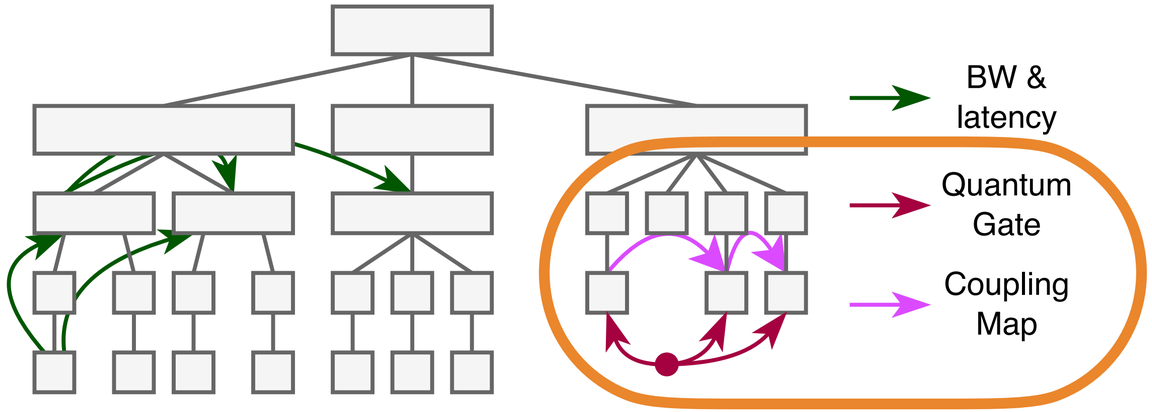
Mishra, Vanecek, Echavarria, Deng, Mete, L.Schulz, M.Schulz. *Towards a Unified Architectural Representation in HPCQC: Extending Sys-Sage for Quantum Technologies*. ISC High Performance 2025 https://doi.org/10.23919/ISC.2025.11017506

sys-sage

Quantum Extension

- Quantum Processing Units (QPUs) begin to accompany CPUs and GPUs in HPC systems
- sys-sage was extended to support QPUs alongside CPUs/GPUs
- Possible to retrieve information from existing APIs, such as QDMI or Qiskit runtime





Mishra, Vanecek, Echavarria, Deng, Mete, L.Schulz, M.Schulz. *Towards a Unified Architectural Representation in HPCQC: Extending Sys-Sage for Quantum Technologies*. ISC High Performance 2025
https://doi.org/10.23919/ISC.2025.11017506

sys-sage — Out-of-the-box integrations

Data Parsers

static information at startup

- hwloc (CPU topology)
- MT4G (GPU topology)
- IQM (quantum topology information)
- MUSA topology (simulator-based CPU topology)
- Performance benchmarks and microbenchmarks (through xPPV)
  - FIRESTARTER, BabelSTREAM, NAS Parallel Benchmarks, HPL, LULESH, cccbench

3rd Party API

dynamic information during runtime

- Intel PQOS (CPU resource isolation)
- NVIDIA MIG (GPU resource isolation)
- proc_cpuinfo (CPU frequency)
- PAPI (performance counters)
- QDMI (Quantum systems information)

xPPV — Understanding CPU Performance Variability

xPPV — Understanding CPU Performance Variability

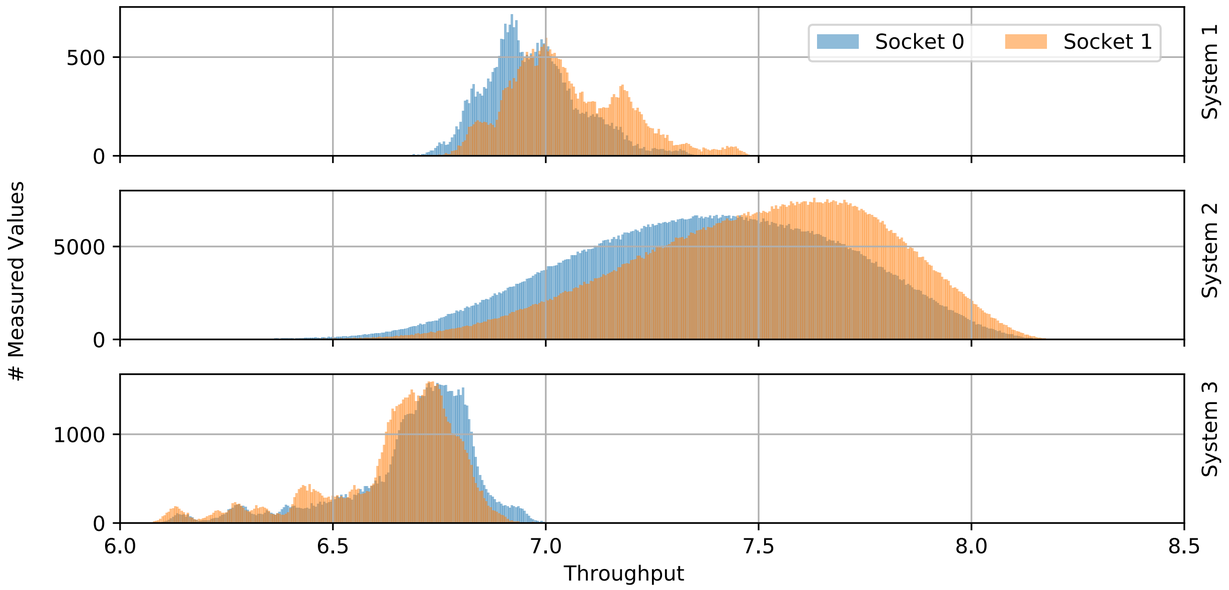
HPC systems' components exhibit different performance

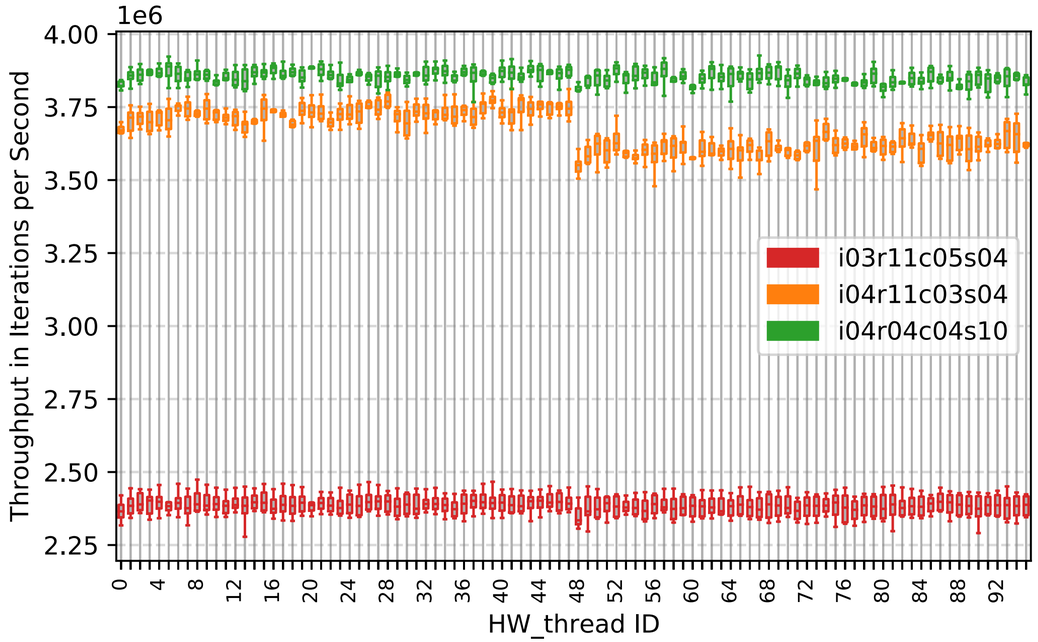
- *(random)* effects of manufacturing process
- *(random)* misconfiguration / HW failures
- *(systemic)* chip/node/system architecture artifacts

xPPV — Understanding CPU Performance Variability

HPC systems' components exhibit different performance

- *(random)* effects of manufacturing process
- *(random)* misconfiguration / HW failures
- *(systemic)* chip/node/system architecture artifacts





xPPV — End-to-end Toolchain

xPPV — End-to-end Toolchain

1 MPPV — Topology-aware workload placement

- Runs arbitrary benchmarks on specified HW resrouces
- sys-sage topology context (hwloc)

xPPV — End-to-end Toolchain

1 MPPV — Topology-aware workload placement

- Runs arbitrary benchmarks on specified HW resrouces
- sys-sage topology context (hwloc)

2 APPV — Analyse & Visualize Raw Results

- Data Preprocessing
- Automated Analysis and Plot Generation

xPPV — End-to-end Toolchain

1 MPPV — Topology-aware workload placement

- Runs arbitrary benchmarks on specified HW resrouces
- sys-sage topology context (hwloc)

2 APPV — Analyse & Visualize Raw Results

- Data Preprocessing
- Automated Analysis and Plot Generation

3 UPPV — Exposing Variability Information

- An API as an extension of sys-sage
- Variability information in the topology context

xPPV — End-to-end Toolchain

1 MPPV — Topology-aware workload placement

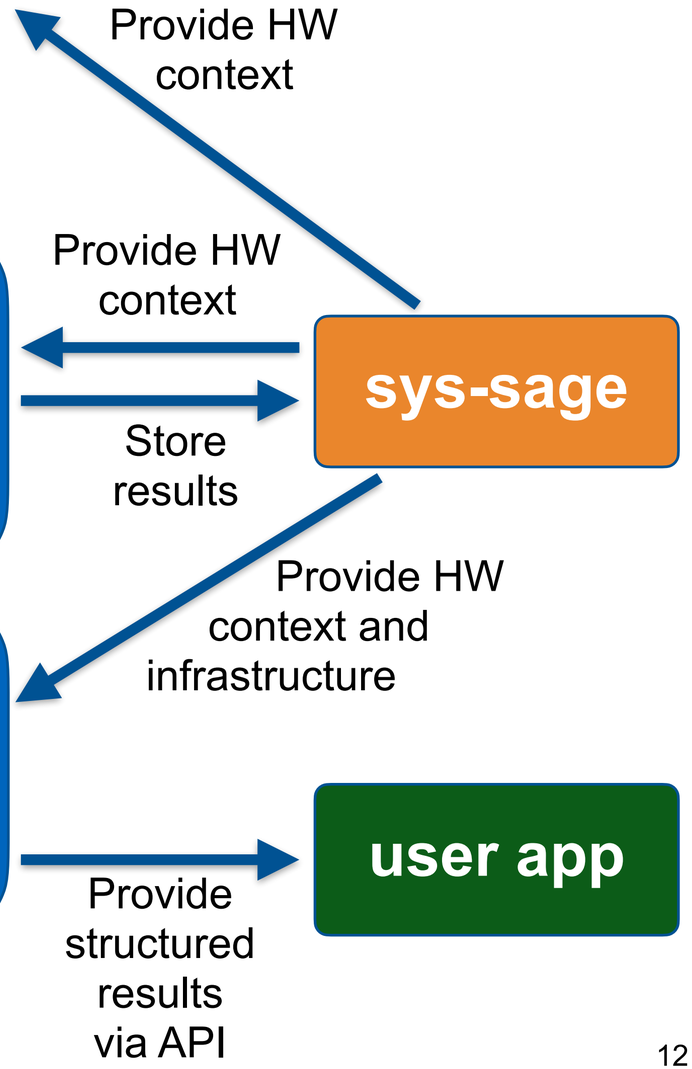
- Runs arbitrary benchmarks on specified HW resrouces
- sys-sage topology context (hwloc)

2 APPV — Analyse & Visualize Raw Results

- Data Preprocessing
- Automated Analysis and Plot Generation

3 UPPV — Exposing Variability Information

- An API as an extension of sys-sage
- Variability information in the topology context



Understanding Heterogeneous Systems Through Topology-Aware Tools

A toolkit for collecting, managing, and utilizing topological information about HPC systems

Tools

MT4G

GitHub: https://github.com/caps-tum/mt4g

Paper ➡

sys-sage

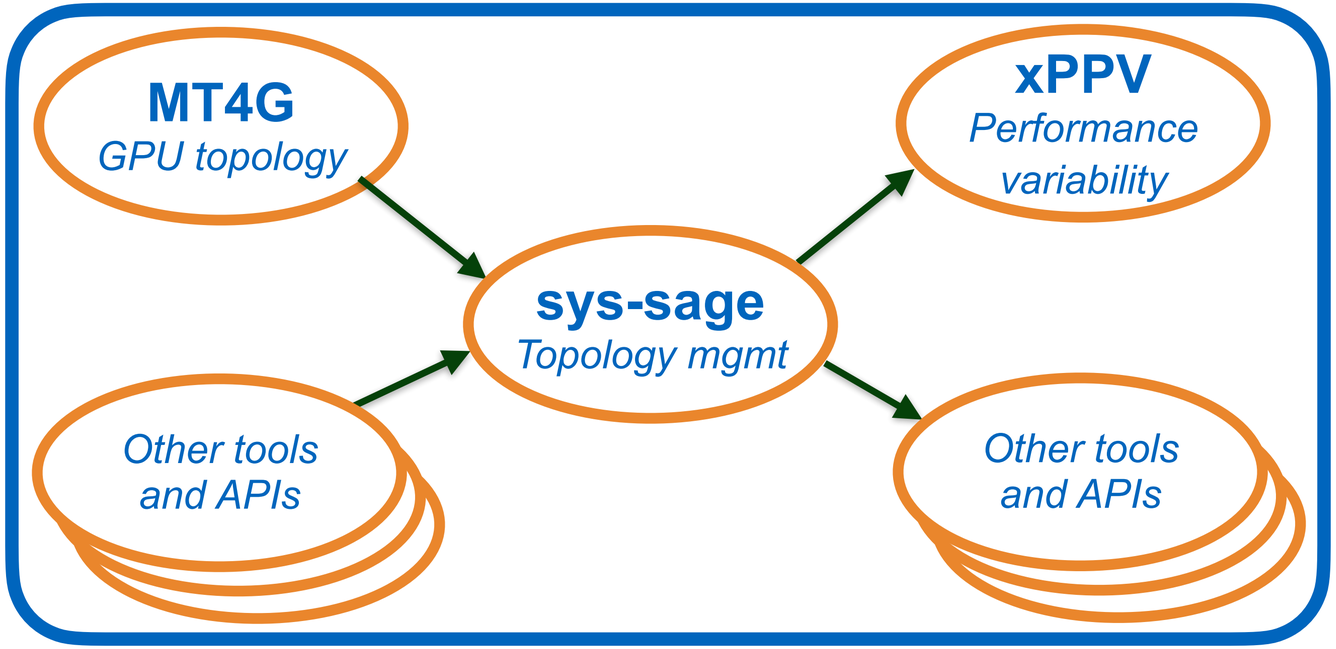
GitHub: https://github.com/caps-tum/sys-sage

⬅ Original Concept Paper

Quantum Extension Paper ➡

xPPV

Work-in-progress, not yet publicly available.
Reach out if you are interested.



Stepan Vanecek
stepan.vanecek@tum.de

Technical University of Munich
Chair of Computer Architecture and Parallel Systems (CAPS)
Prof. Martin Schulz

SPONSORED BY THE
Federal Ministry of Education and Research
PLASMA PEPSC
DEEP-SEA
SEANERGYS
ENERGY EFFICIENT EXASCALE