montblanc-project.eu | @MontBlanc_EU

Simulation Tools Mont-Blanc 3

Gabor Dozsa (Arm Research)

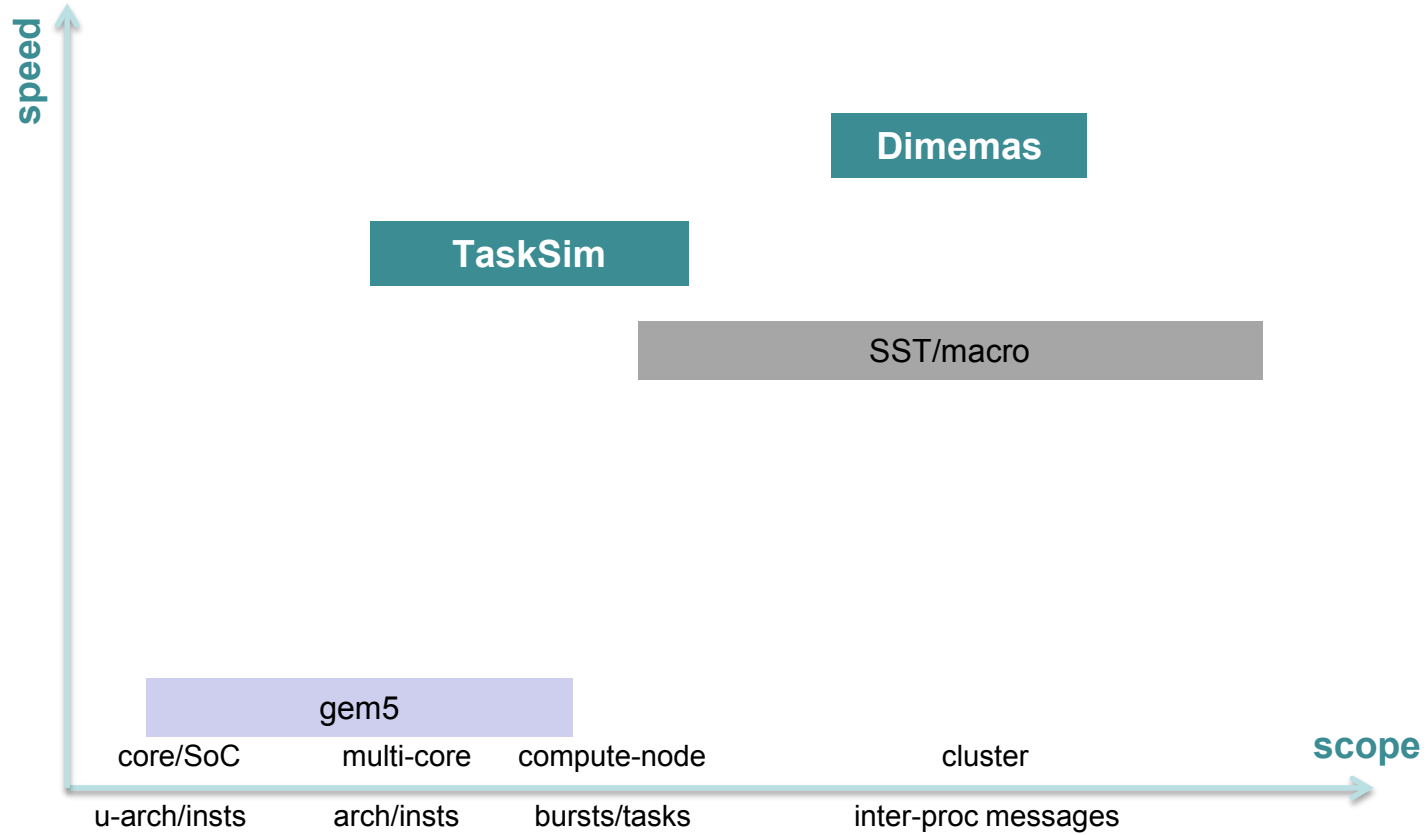
This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 671697

Simulation Tools Work Package

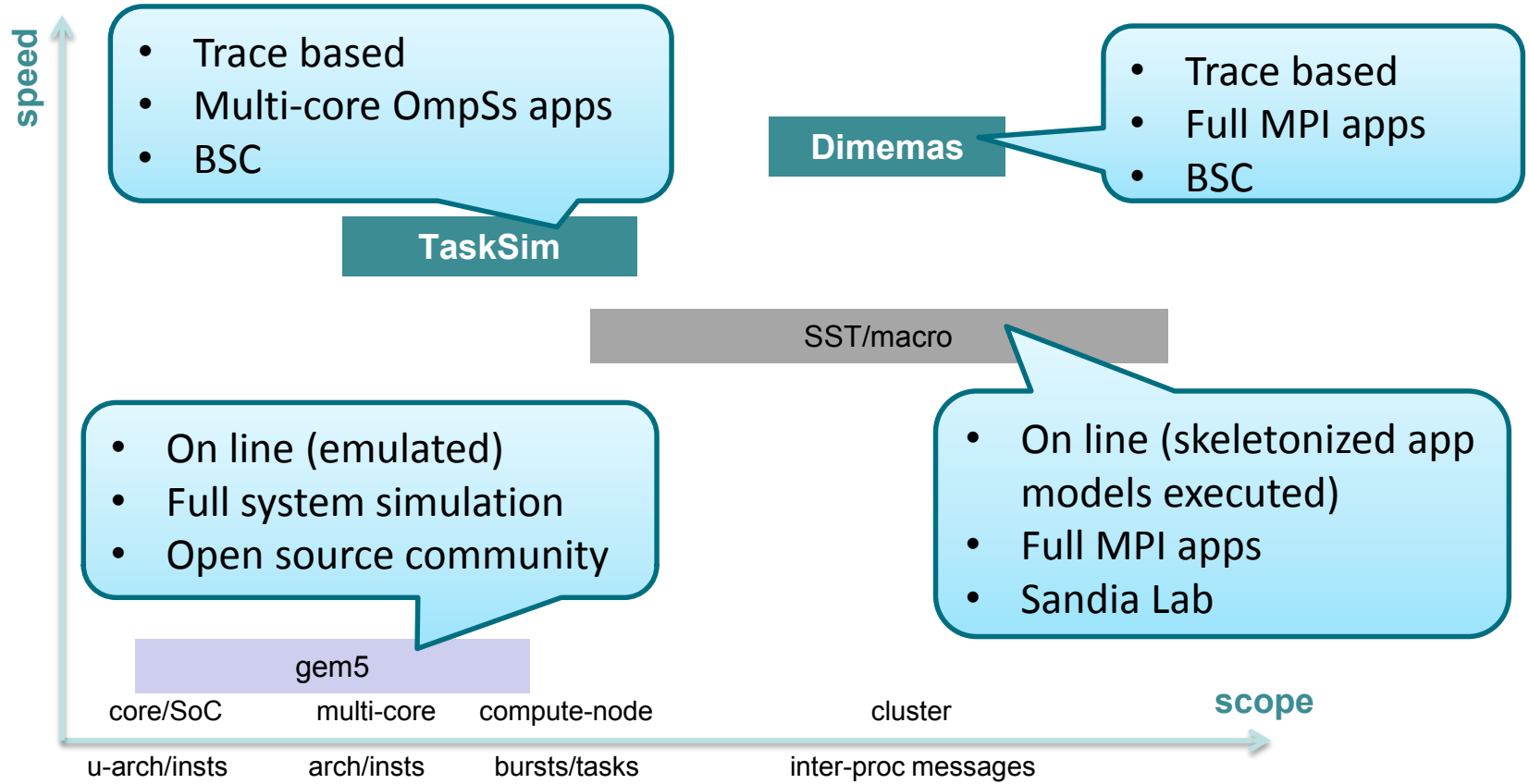
Main objectives

- "The project will develop a **new multi-scale simulation methodology** to enable efficient and accurate design space exploration for a balanced system architecture."
- "The multi-scale simulation infrastructure will be **built on top of existing tools** that are already in use at some of the partners and third parties."
- "The project aims to provide an integrated yet flexible and scalable workflow that can address efficiently the **design space exploration problem for complex heterogeneous high performance systems**."

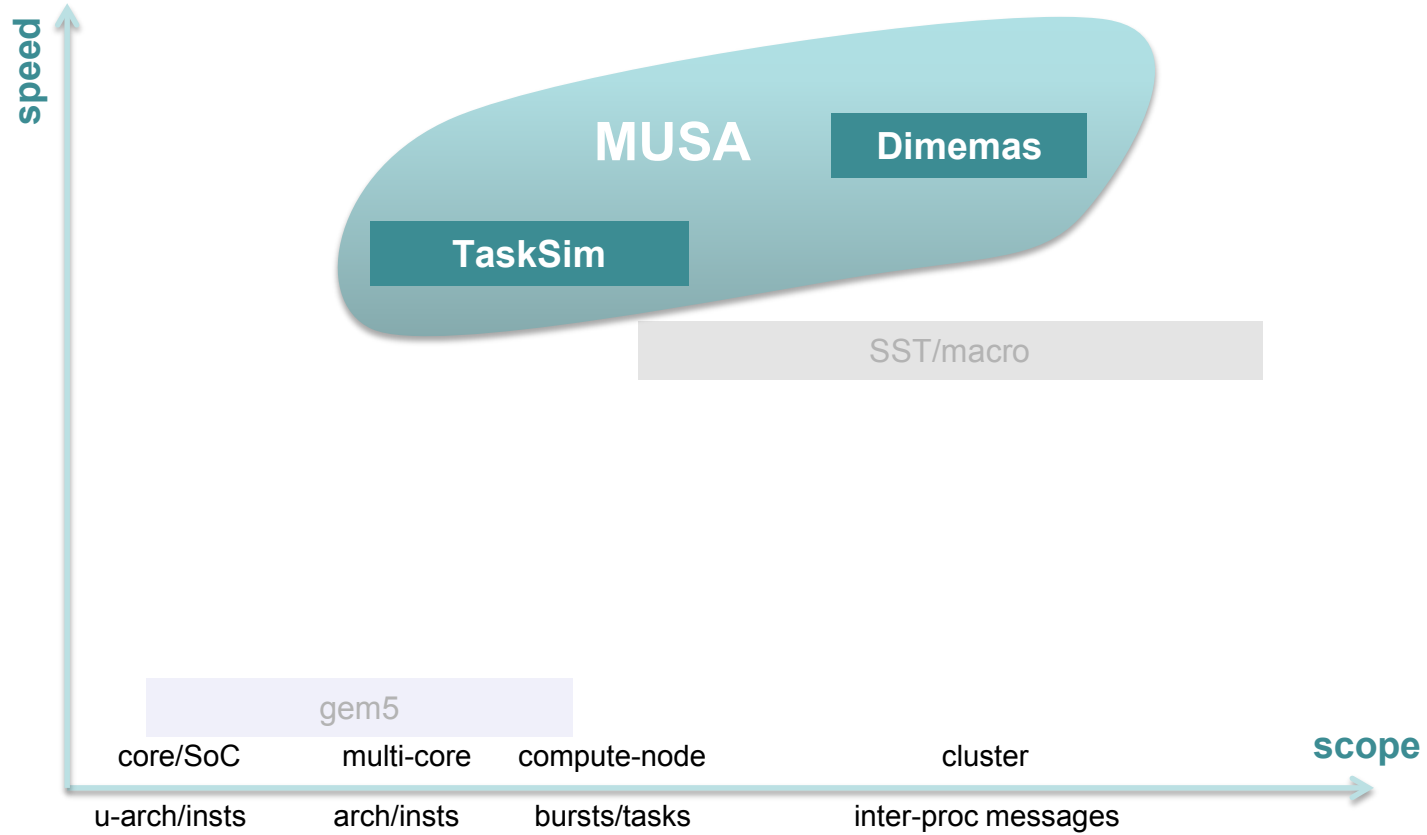
Existing simulation tools at project start



Existing simulation tools at project start



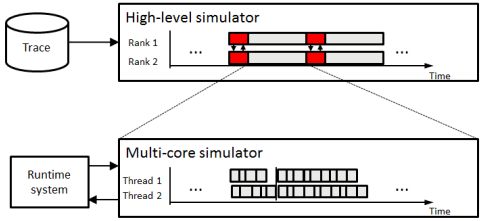
Multi-scale simulation : MUSA



MUlti-level Simulation Approach (MUSA)

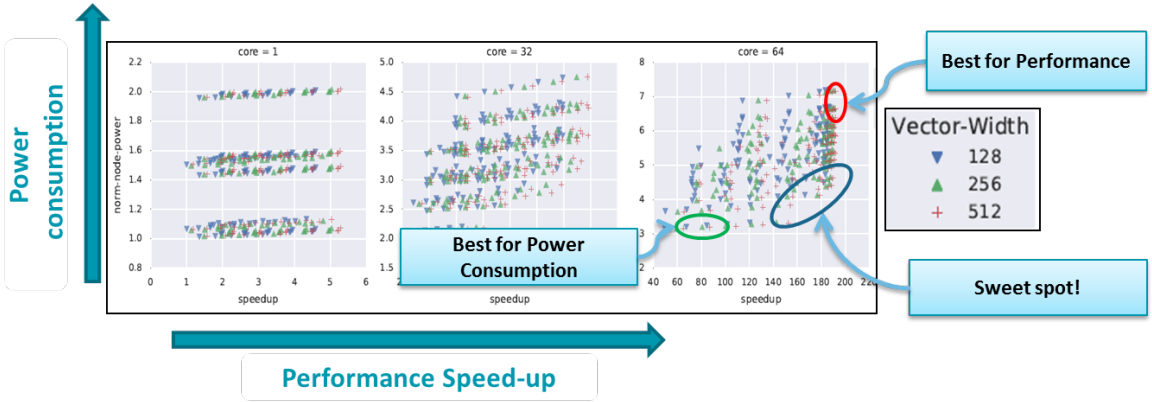
- **MUSA** is a **multi-level** simulation infrastructure that allow us to do **performance predictions** for **large scale systems**.

- **Features:**
  - Trace based simulator.
  - Captures **instruction**, OpenMP **runtime** and **network** level events.
  - Leverages **sampling techniques** for fast simulation of applications running in large scale systems.
  - **Tracing** using **DynamoRIO**, ARMv8 and x86_64 compatible.
  - Power measurements obtained using **McPAT** and **DRAMPower**
  - Simulations with #X nodes require a system with #X number of nodes.
    - As many cores as you want inside each node.
  - 2-3 MIPS in 'detail' mode, 500-1000 MIPS in burst mode.



A Design Space Exploration (DSE) with MUSA

- **We studied ~850 architectural simulations per application**



MPI+OPENMP APPS SIMULATED

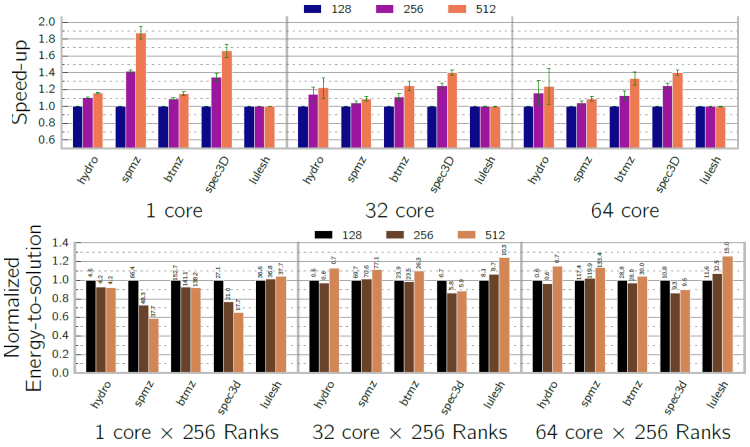
- HYDRO
- NAS-BT
- NAS-SP
- SPECFEM3D
- LULESH
- HPCG (WIP)

- **It allowed us to explore several architectural features**
  - Memory BW, number of cores, issue, cache and vector size.
- **We were are able to understand and quantify performance and power tradeoffs**
  - That is key insight to co-design new systems and apps.

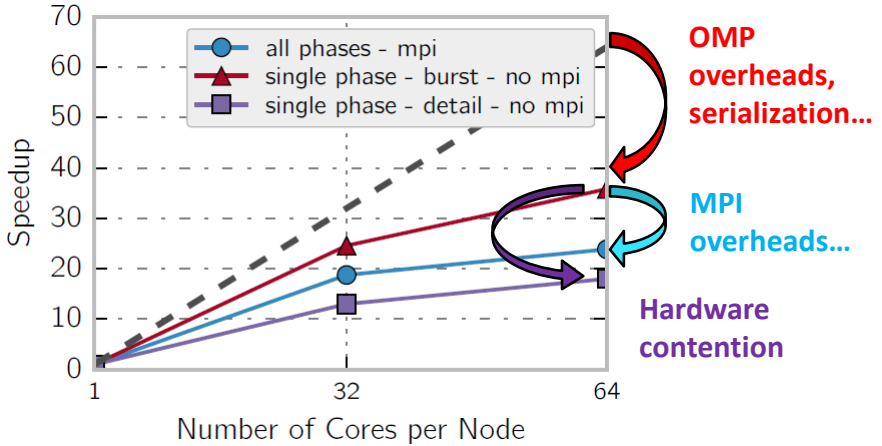
MONT-BLANC

Co-Design Insight (e.g. simulating 256 Nodes 1 to 64cores)

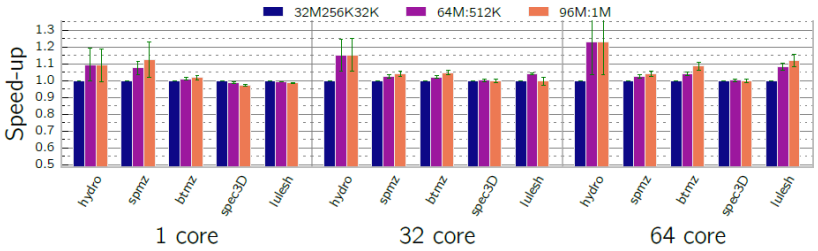
➔ Test wider vector FP units.

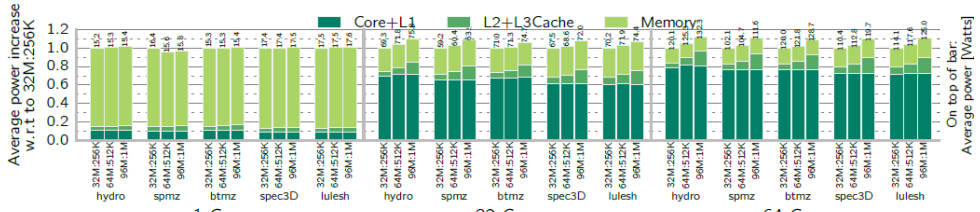


➔ Isolate scheduling, hardware resource contention and MPI overheads.



➔ Adjust Cache size per core ratio

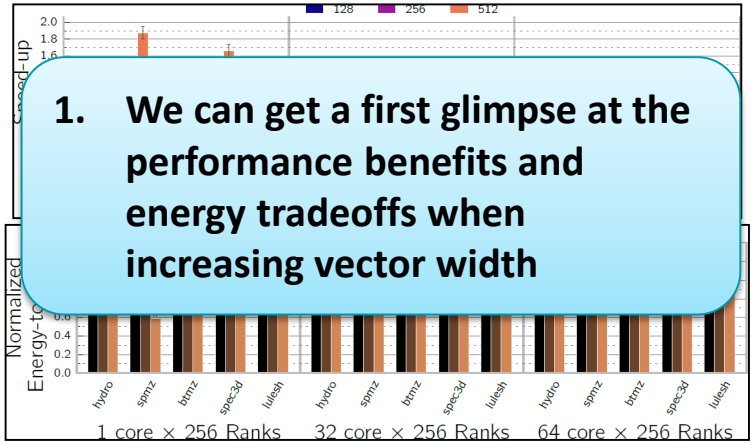




MONT-BLANC

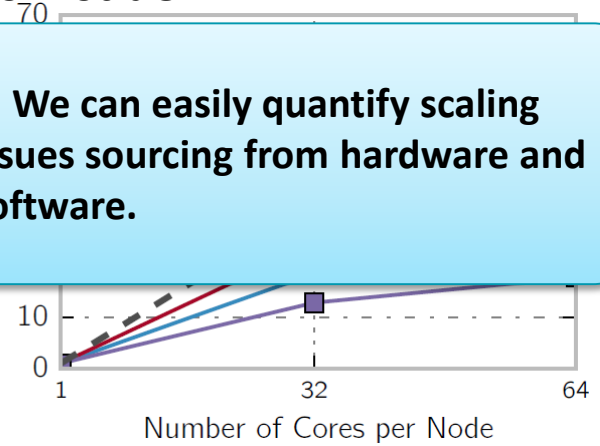
Co-Design Insight (e.g. simulating 256 Nodes 1 to 64cores)

➔ Test wider vector FP units.



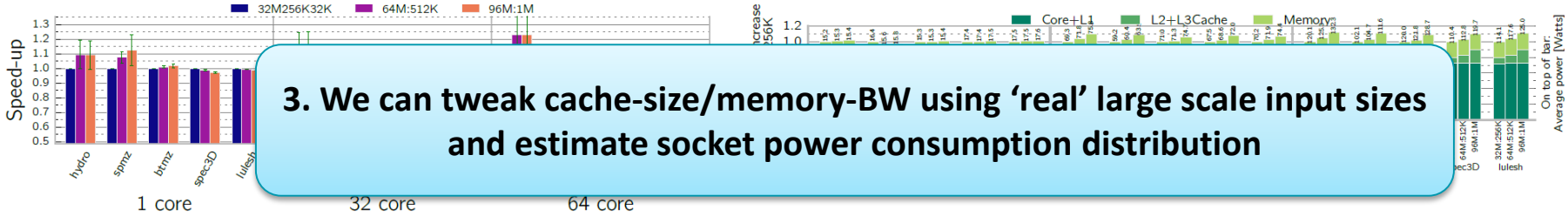
1. We can get a first glimpse at the performance benefits and energy tradeoffs when increasing vector width

➔ Isolate scheduling, hardware resource contention and MPI overheads.



2. We can easily quantify scaling issues sourcing from hardware and software.

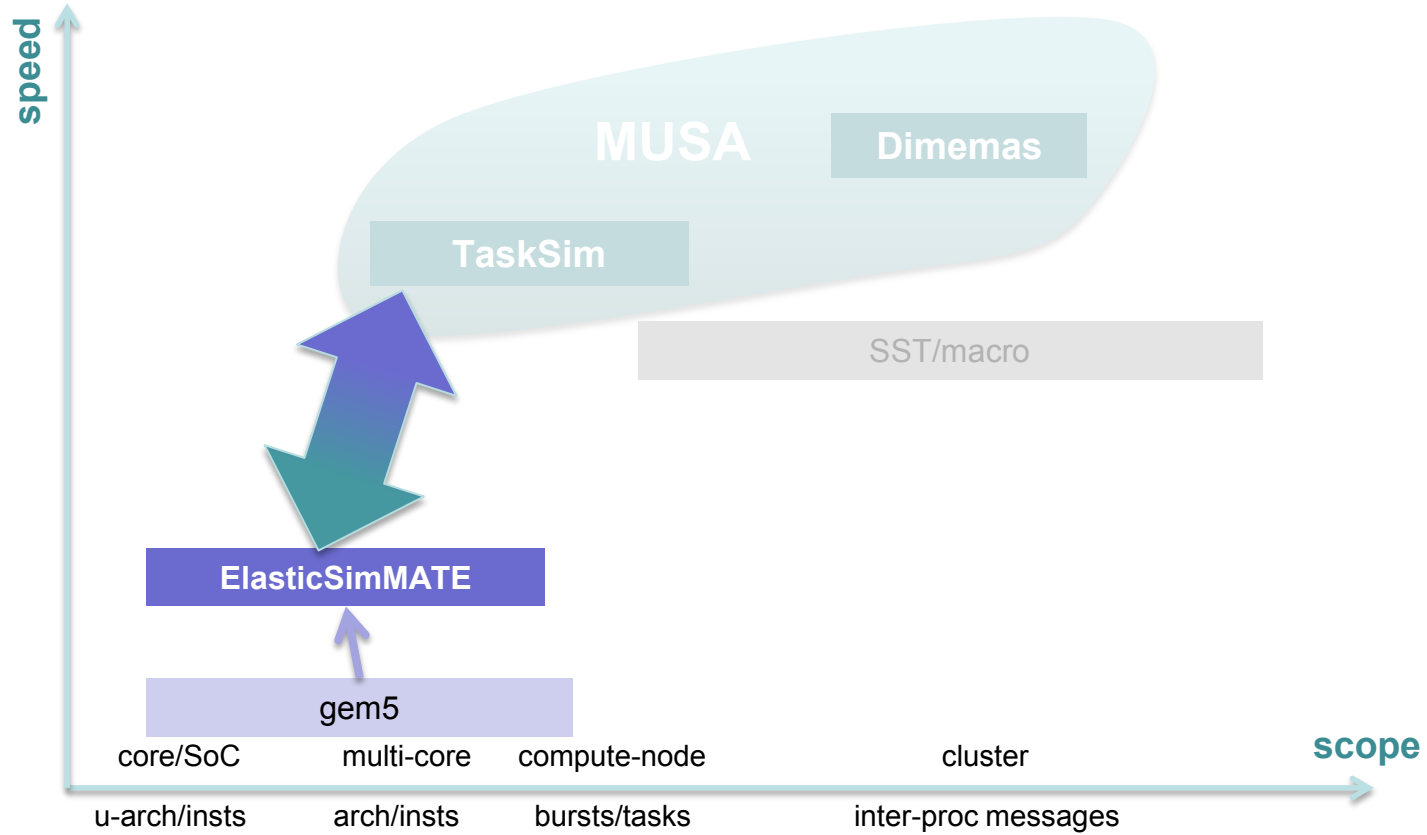
➔ Adjust Cache size per core ratio



3. We can tweak cache-size/memory-BW using 'real' large scale input sizes and estimate socket power consumption distribution

MONT-BLANC

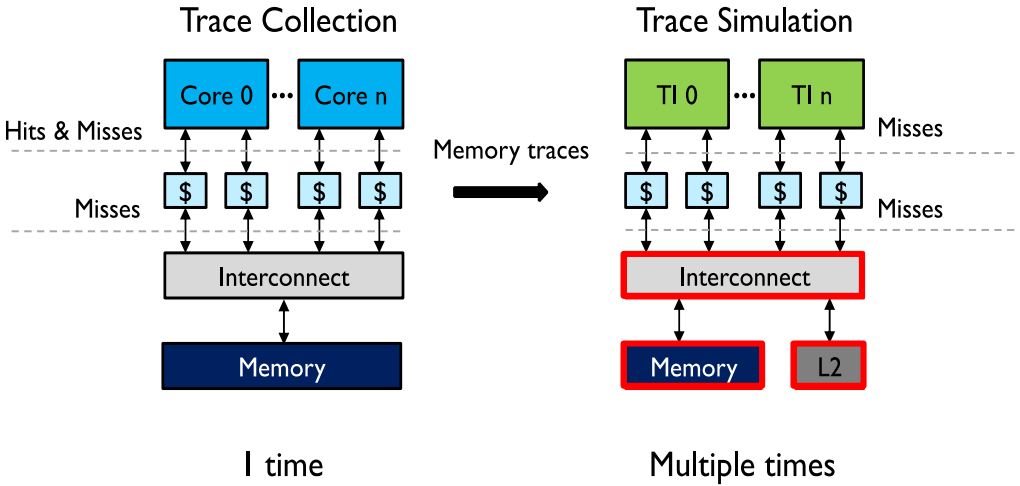
ElasticSimMATE



MONT-BLANC

ElasticSimMATE: Trace driven simulation in gem5

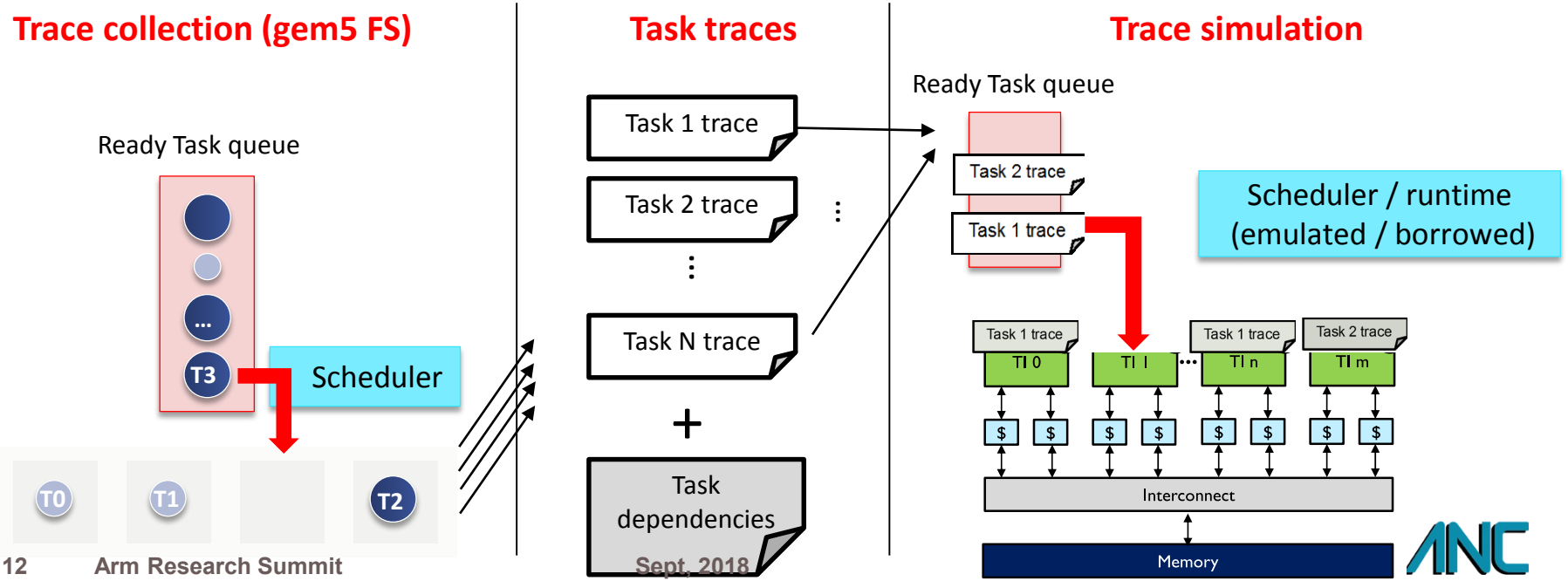
- **Trace-driven simulation, in gem5**
  - Simulation speedup: cores → Trace Injectors (TI)
  - Traces (gem5 FS) are application & CPU core specific
  - Interconnect, L2+ caches, memory can be explored @ replay time



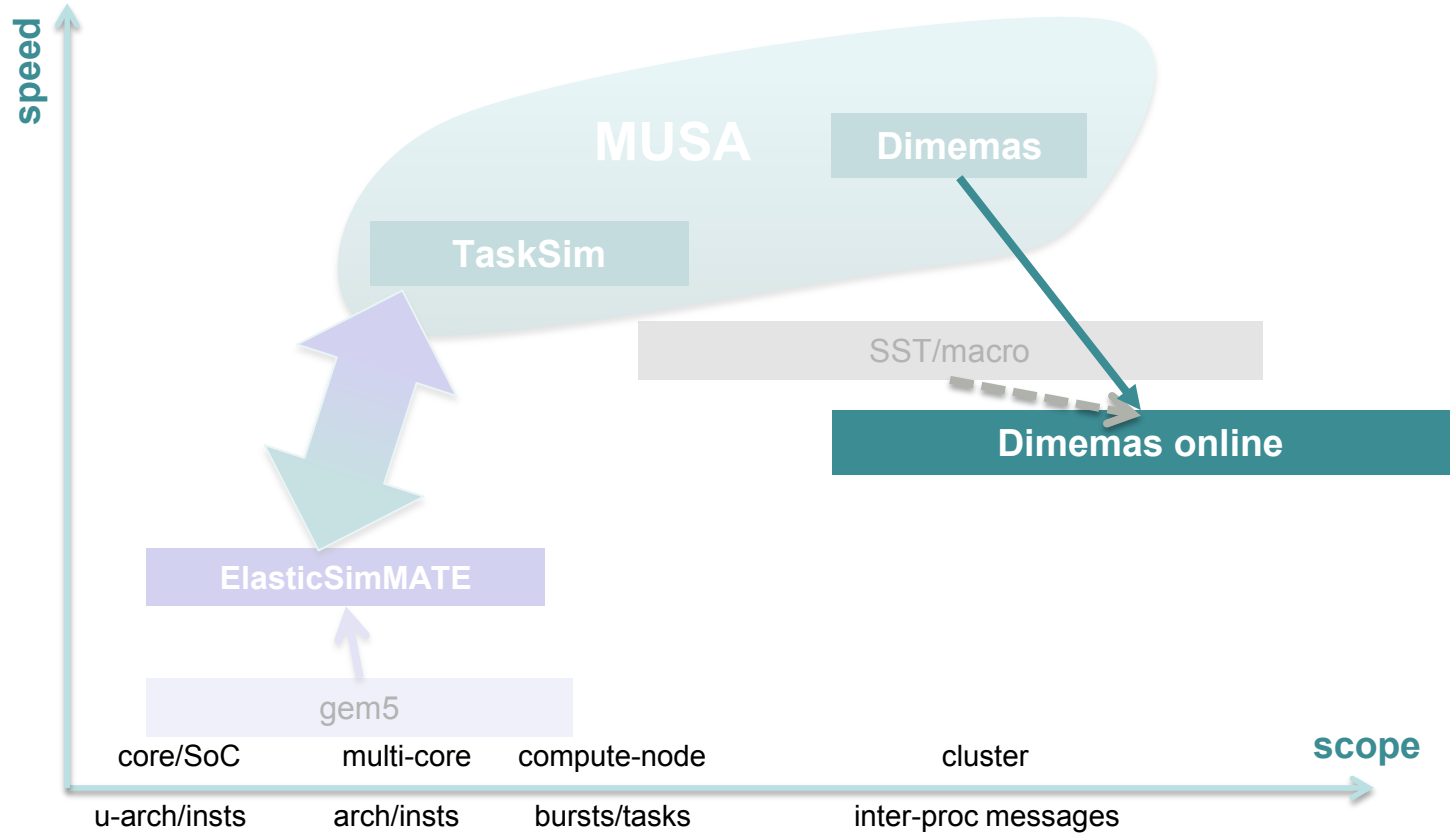
- **ESM Modes**:
  - **Synchronization Mode**: Trace replication mechanism (weak scaling)
  - **Task Mode**: Task traces are assigned to the different number of cores (strong scaling)

ElasticSimMATE: strong scaling analysis

- **ESM exploits OmpSs tasking API for strong scaling analysis**
  - Traces are bound to *Tasks* and not *Cores* anymore
  - Arbitrary # cores in replay, for strong scaling analysis



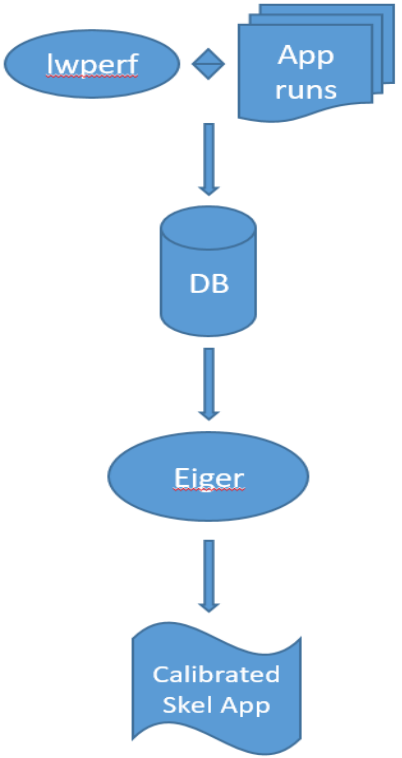
Dimemas online



MONT-BLANC

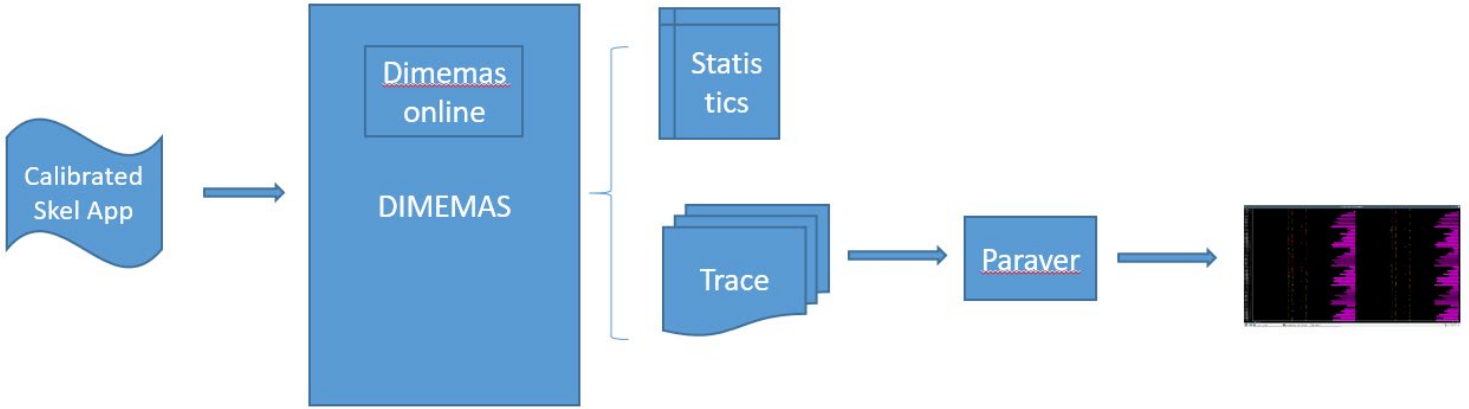
Skeleton models of MPI applications

- **Skeleton application**: replace compute blocks with parametrized *compute_time* expressions
  - *compute_time* expressions generated by the Eiger statistical model generator (Sandia Lab).
- Skeleton models has some **advantages** over using MPI traces:
  - Just one skeleton needed to work with different input parameters and number of processes.
  - No need of large storage for traces.
  - Allows for simulation of large number of processes.
- **SST/macro** (Sandia) simulates with **skeletons**.
- **Dimemas** (BSC) simulates using **MPI traces**.

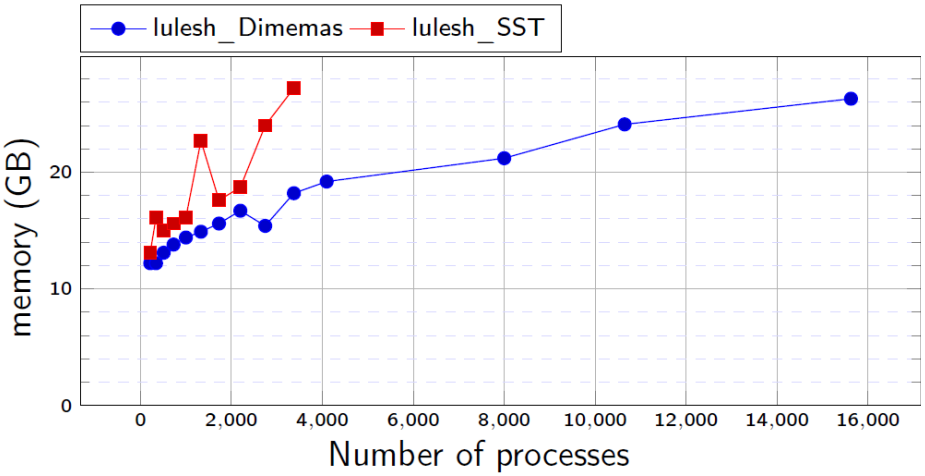


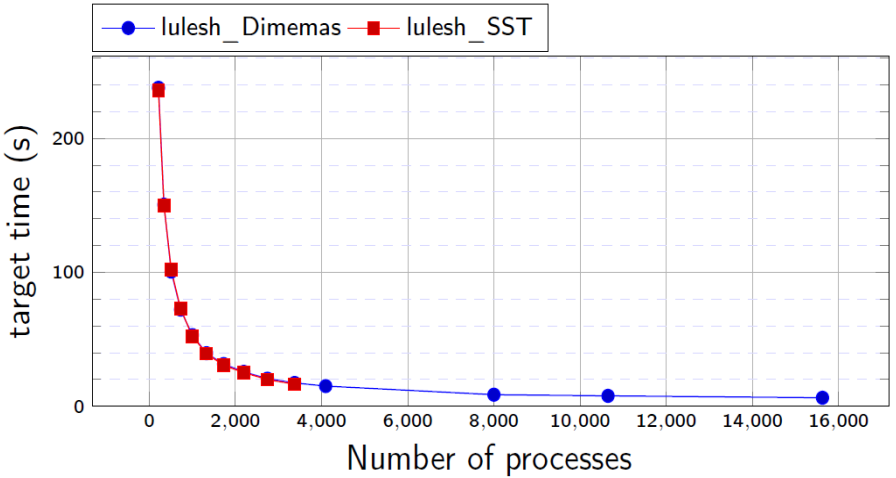
Dimemas online

- **Dimemas online**: extended Dimemas to work with skeletons instead of traces.
  - Simulated processes as threads in a single node.
  - Implemented part of MPI.
  - Dimemas can generate Paraver trace from the simulation of the skeleton.
    - Visualise the runtime behavior of the skeleton application.



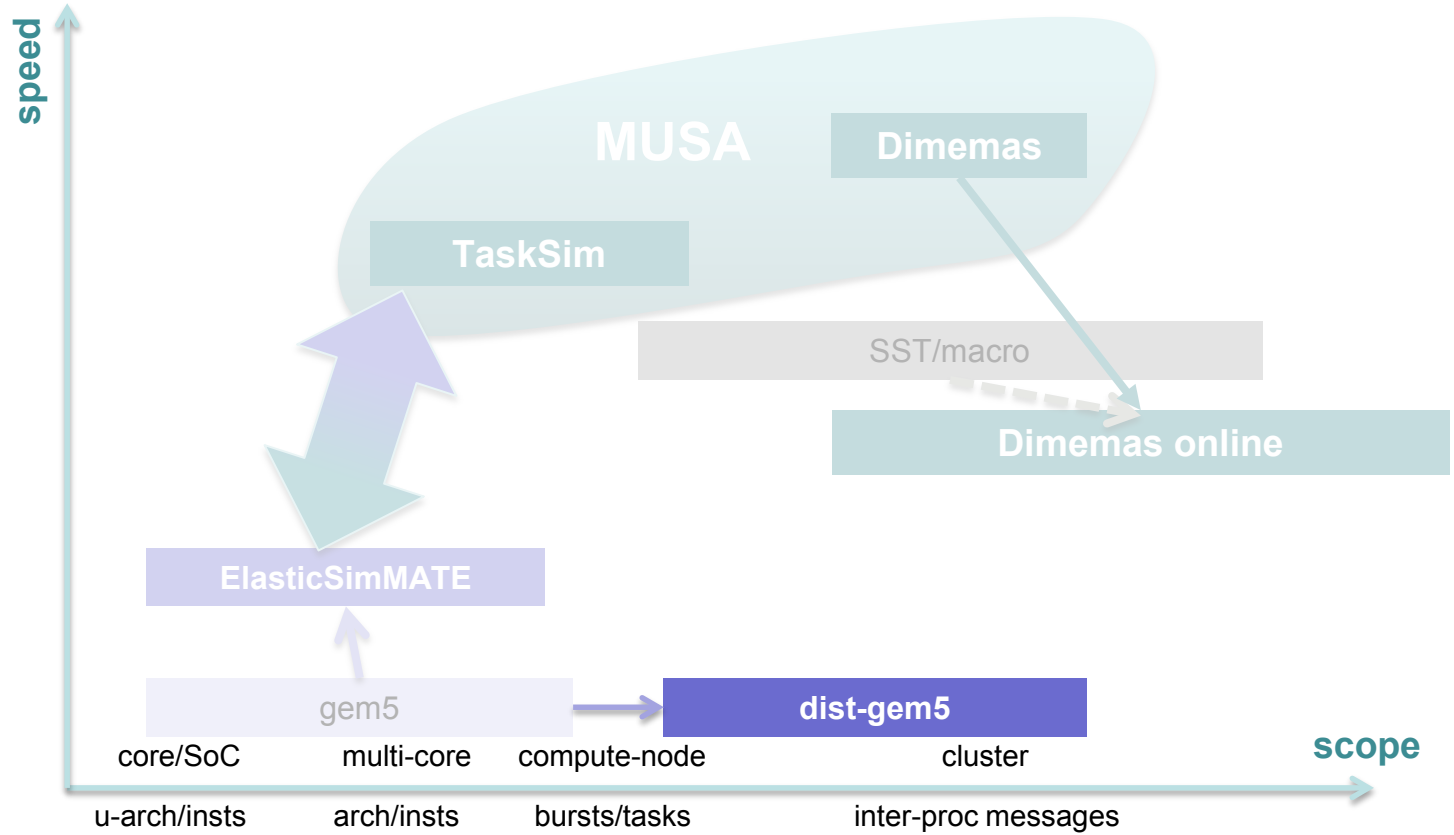
Dimemas online simulation results





- Simulated **lulesh** with 1050^3 elements, 5 iterations, varying the number of MPI procs
  - 15k procs, 45 min of simulation, 25.9 GB memory usage, trace of 1.6 GB.
  - 91k procs, 23 hours of simulation, 47.9 GB memory usage.
- Dimemas online is more scalable than SST/Macro
  - **Memory usage** is the limitation.
- Dimemas online predicts the same target times than SST/macro for lulesh
  - Accurate **CPU models** are more important than detailed network simulation

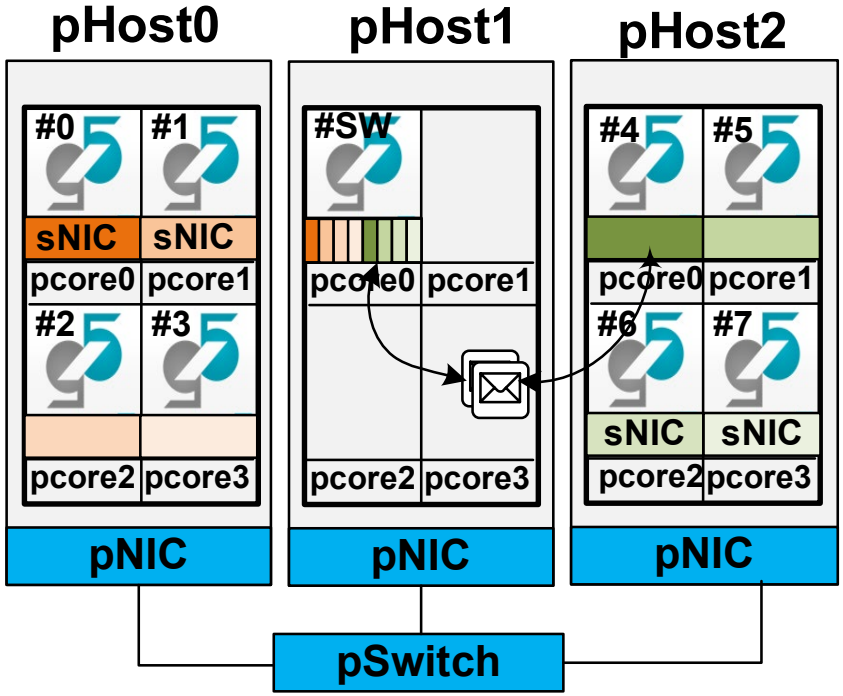
dist-gem5 : Extending gem5 to simulate a cluster



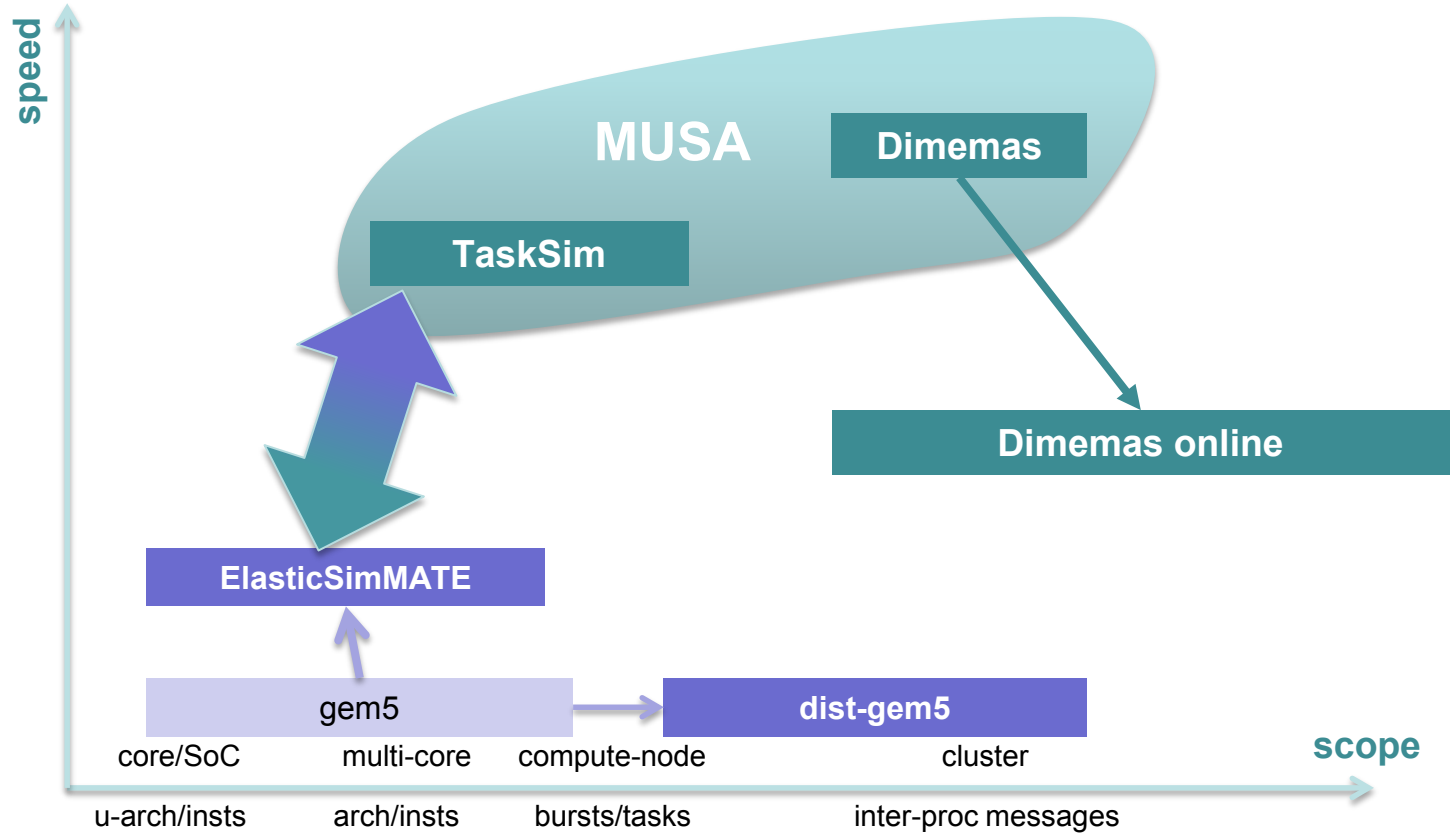
MONT-BLANC

dist-gem5

- **Simulating cluster of compute nodes on a real cluster**
  - Each gem5 simulates a compute node and a NIC
  - All communication links and interconnect is simulated by an additional gem5
  - gem5 instances run in parallel



Mont-Blanc 3 simulation tools



Conclusions

- **We have demonstrated the feasibility of the multi-scale simulation approach**
  - MUSA is used extensively for the design space exploration in the project
- **We have pushed the limits of simulating full HPC systems**
  - Scaling out
    - simulated up to 91K MPI processes with Dimemas online
  - Zoom into fine grain details when it matters
    - MUSA and ElasticSimMATE joint simulation workflow
    - dist-gem5 to simulate clusters in full system mode
- **Project work not covered in this presentation**
  - SVE support implemented for gem5
  - Reducing simulation time by utilizing CERE codelets and the Barrier point methodology