Levels of Detail in Performance Analysis

December 18th, 2025

Jesús Labarta (jesus.labarta@bsc.es) , BSC

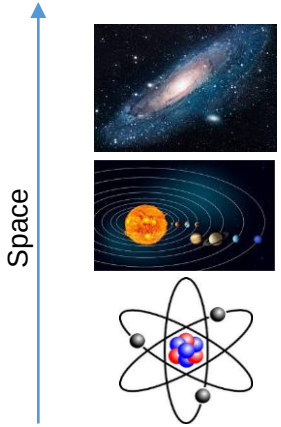
HORIZON-EUROHPC-JU-2023-COE

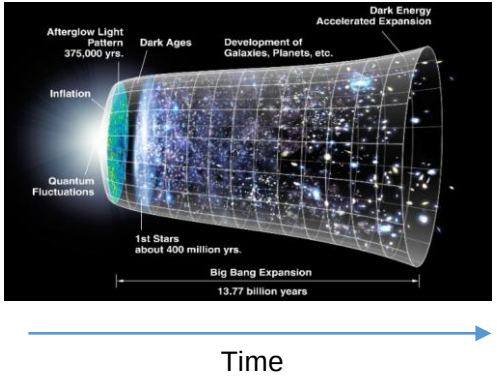
EuroHPC Joint Undertaking

1 January 2024– 31 December 2026

Grant Agreement No 101143931

Towards insight



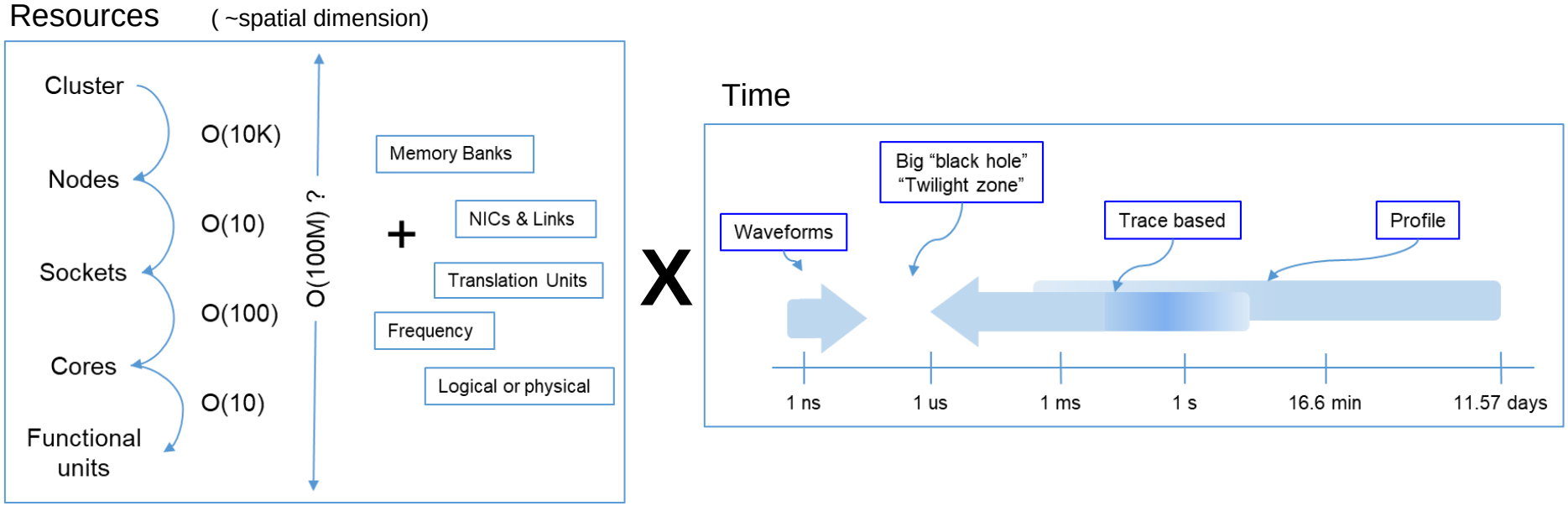


Microscopic "structure" determines macroscopic behavior

Scalability : not about size, about **dynamic range**!
Seamlessly sweep across scales

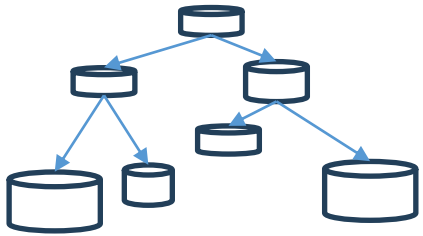
"As above, so below"
Similar concepts/mechanisms/tools at **all levels**

Dimensions in Performance Analysis

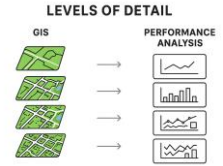


Levels of Detail (LoD)

- Navigation
- Data:
  - May be captured/structured at different levels of detail



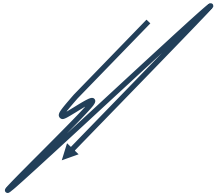
The analogy seen by ChatGPT



Surfing the Performance Data

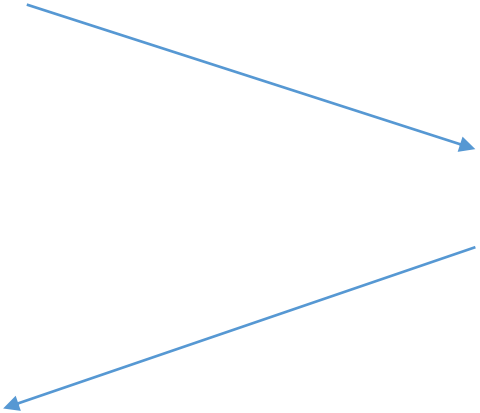
- Dynamic range
  - In scale
    - Zoom out, aggregate
    - Zoom in, detail

  - In semantics !
    - Flexibility in metric definition
    - Integrating know-how at many levels (system to microarchitecture)



Message

- Tool developers
  - Support
  - Enable productivity

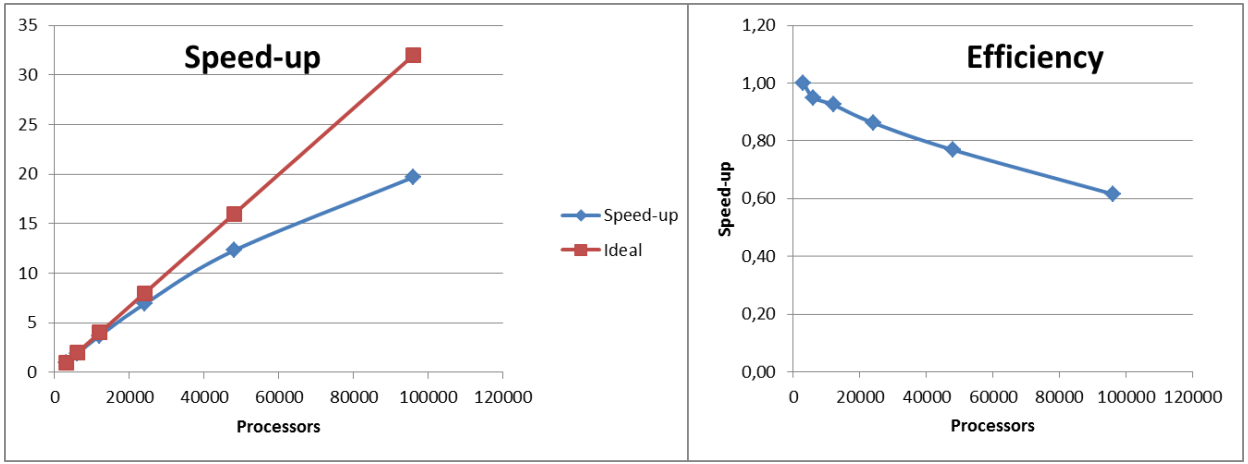


Navigation on a high dynamic range of scales

- Analysts
  - Exploit
  - Search for insight
    - Why, how
      - There is always a next why
    - How much
    - Towards “who to blame”, how to “counteract”, improve

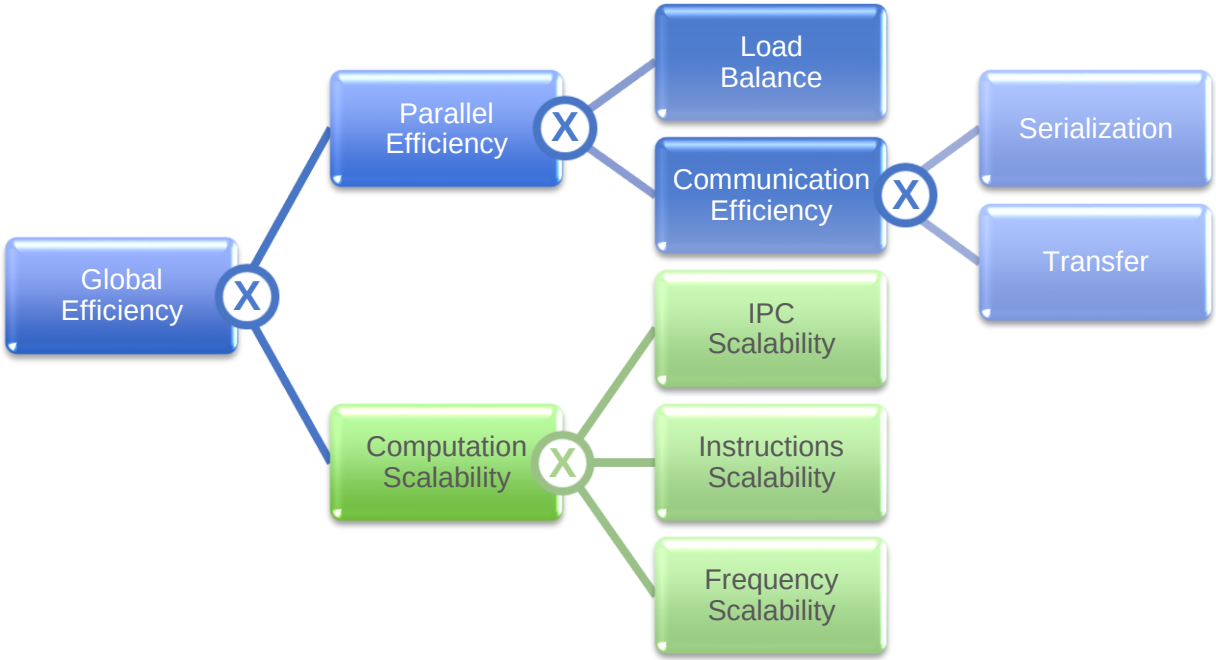
10 Km high view
(actually much further away)

Elapsed time



T(1)

Efficiency Model

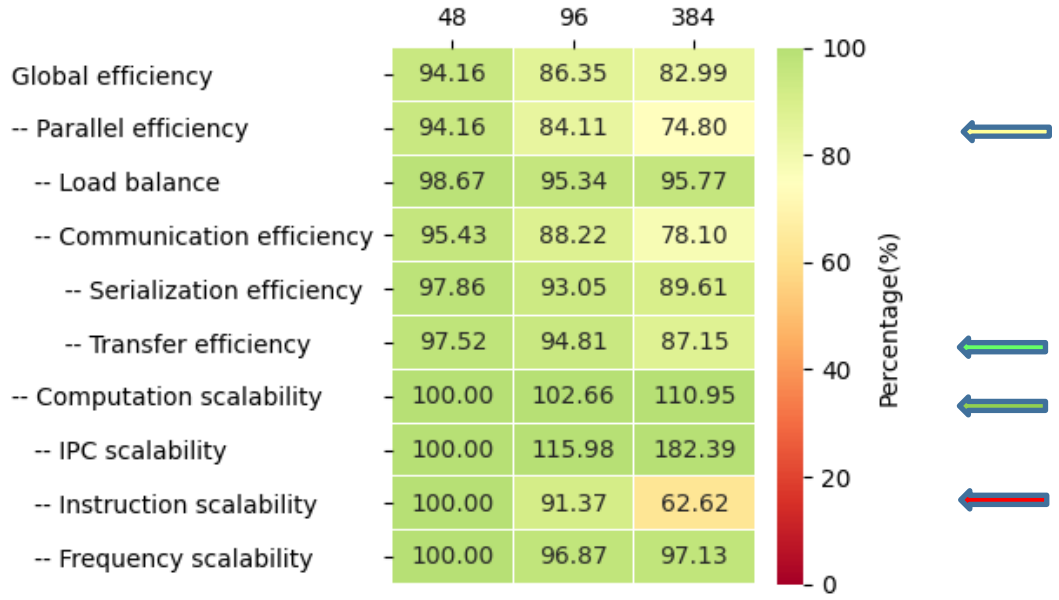


Semantics: Fundamentals of (parallel) computing !!

M. Casas et al, “Automatic analysis of speedup of MPI applications”. ICS 2008.

J. Giménez, et al, "Analyzing the Efficiency of Hybrid Codes" 2020 19th International Symposium on Parallel and Distributed Computing (ISPDC), 2020

Efficiency model



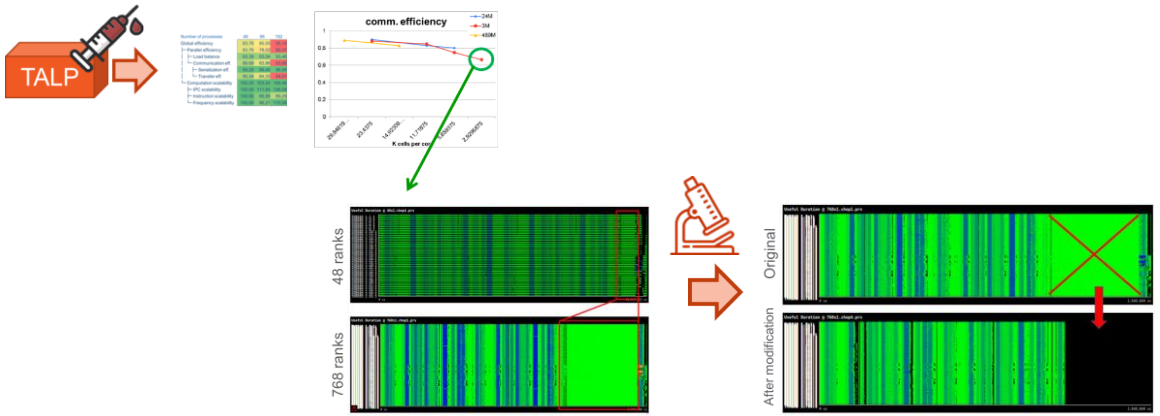
Avg. Useful IPC(48) =0.67

Avg. Useful Frequency(48) =2.061 GHz

Continuous monitoring

- Monitoring of system/application behavior
  - → Steer deeper data acquisitions and analysis capabilities

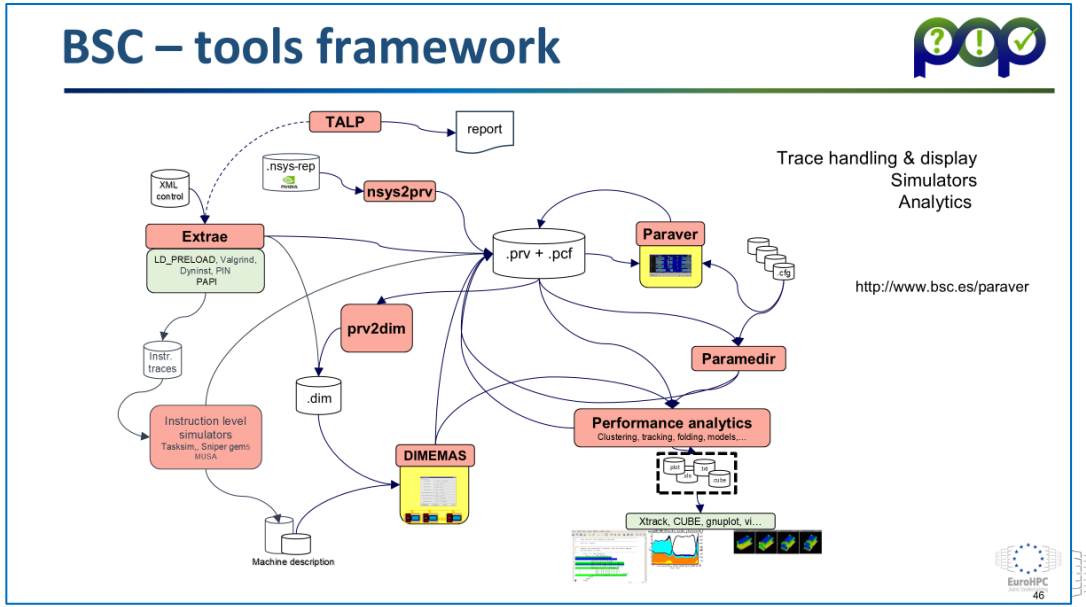
- online
  - **TALP**
  - EAR
  - OPT-CPT



J. Pocurull & M. Garcia @ the ExaFOAM project

Some examples

- Examples
  - Large scale CPU only
  - Small scale MPI + CUDA
  - Medium scale AI training
  - Fine grain RISC-V vector codesign

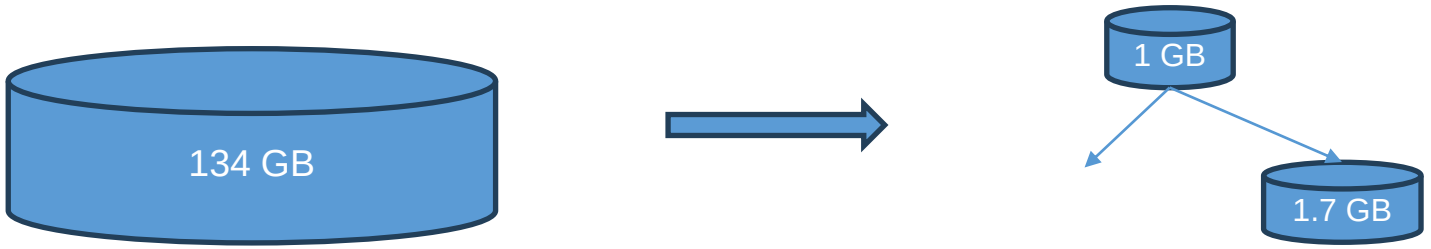


Example 1: large scale CPU only parallel code

From POP assessment POP_AR_021 and scaling experiments by Valentin Seitz (BSC)

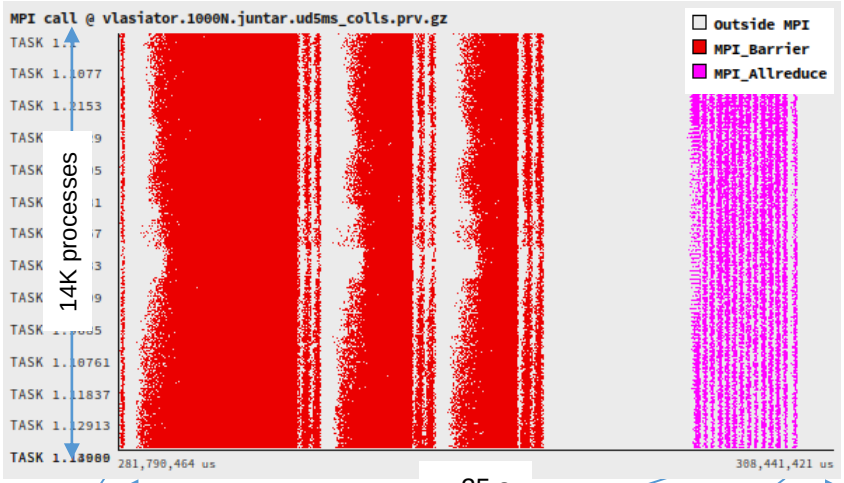
Large scale trace

- Stressing a real code to a very large core count even if not huge problem size
- CPU only
  - MPI + OpenMP
  - 14K processes, 112K threads
- Full detail trace size for one high level algorithmic step : 134 GB



Large scale

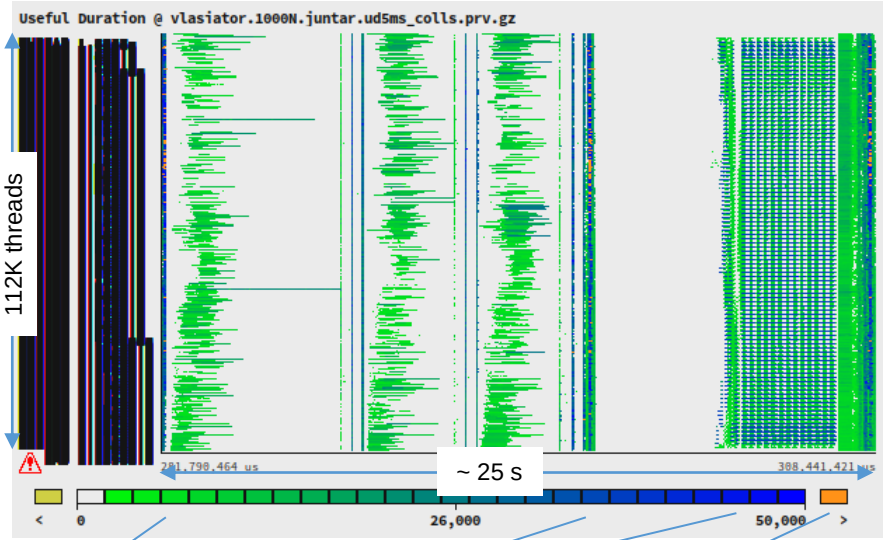
Filtering trace: **collectives, useful > 5ms** size: 1 GB



Par. Eff: 0.299
LB: 0.369
Comm eff: 0.811

Par. Eff: 0.408
LB: 0.409
Comm eff: 0.999

Process level metrics*



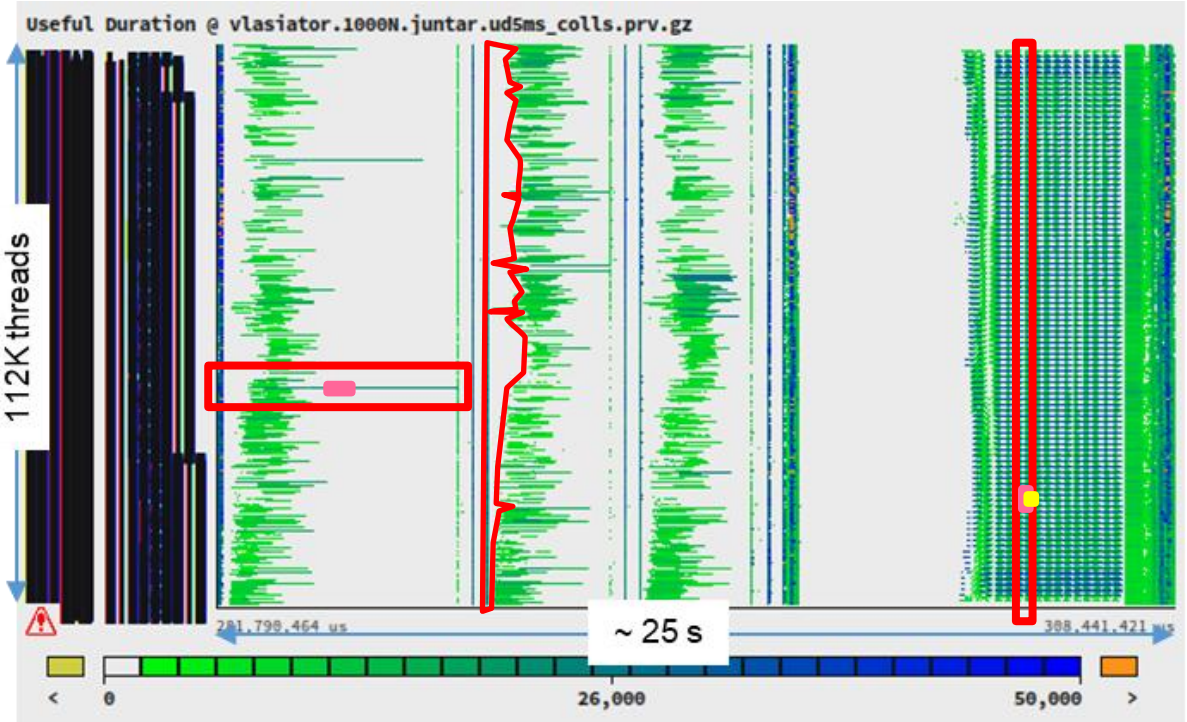
Par. Eff: 0.001
LB: 0.014
Comm eff: 0.362

Par. Eff: 0.019
LB: 0.053
Comm eff: 0.364

Thread level metrics

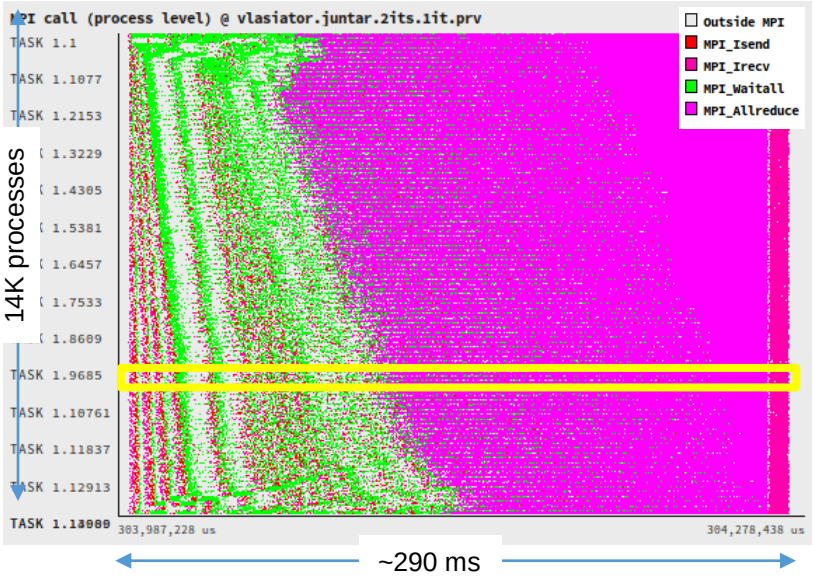
* based only on MPI collectives

Opportunity to focus



Zooming in time …

1 iteration, **all traced events**: 1.7 GB



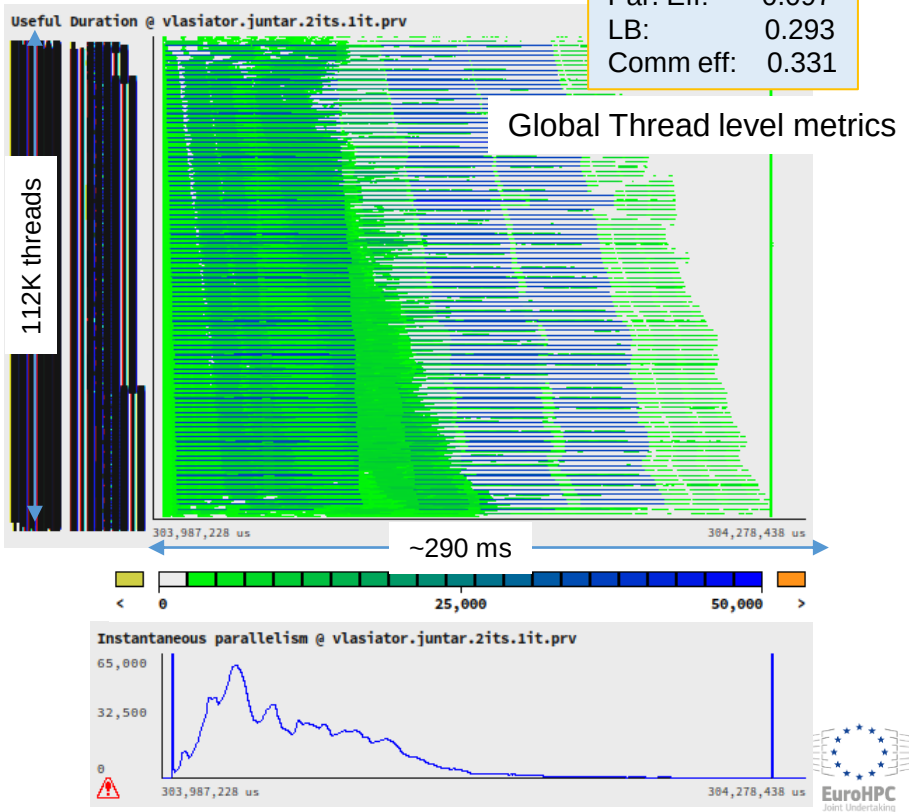
Par. Eff: 0.157
LB: 0.213
Comm eff: 0.738

Process level metrics*

* based on ALL MPI activity by main thread

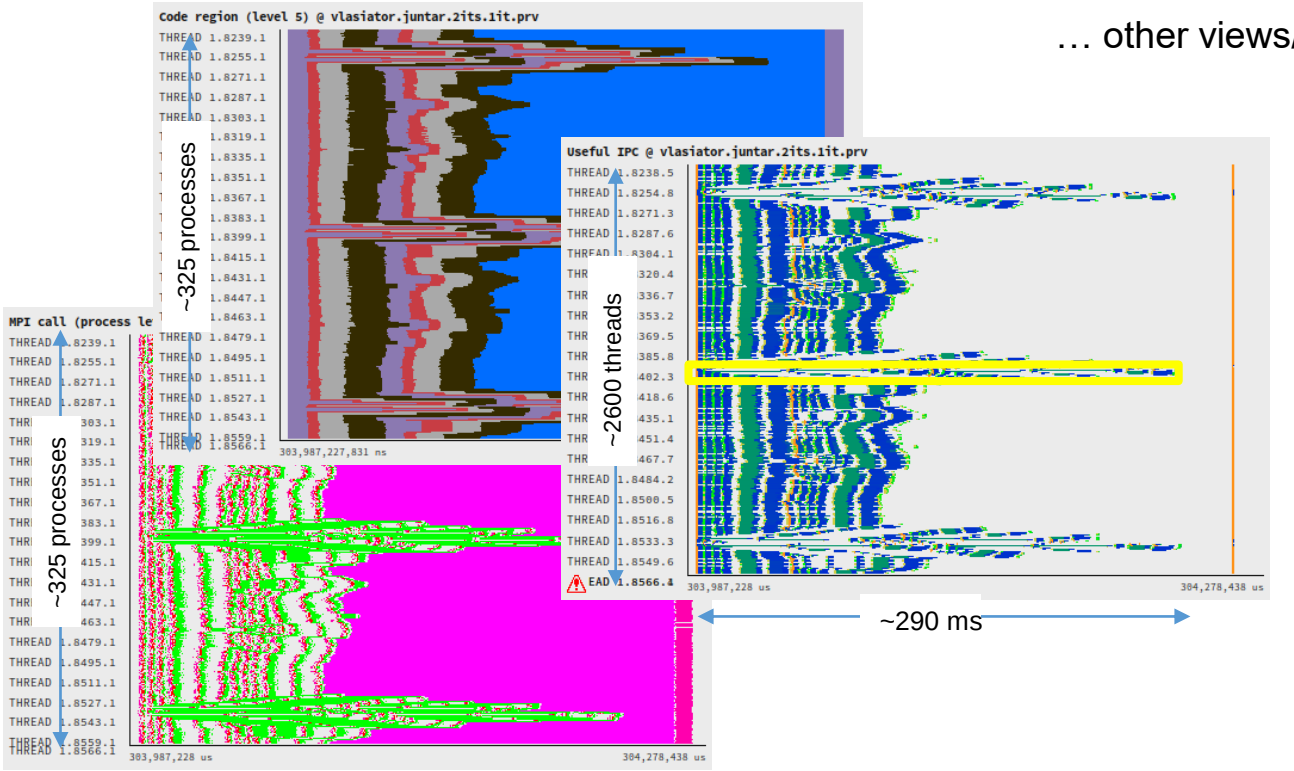
Par. Eff: 0.097
LB: 0.293
Comm eff: 0.331

Global Thread level metrics

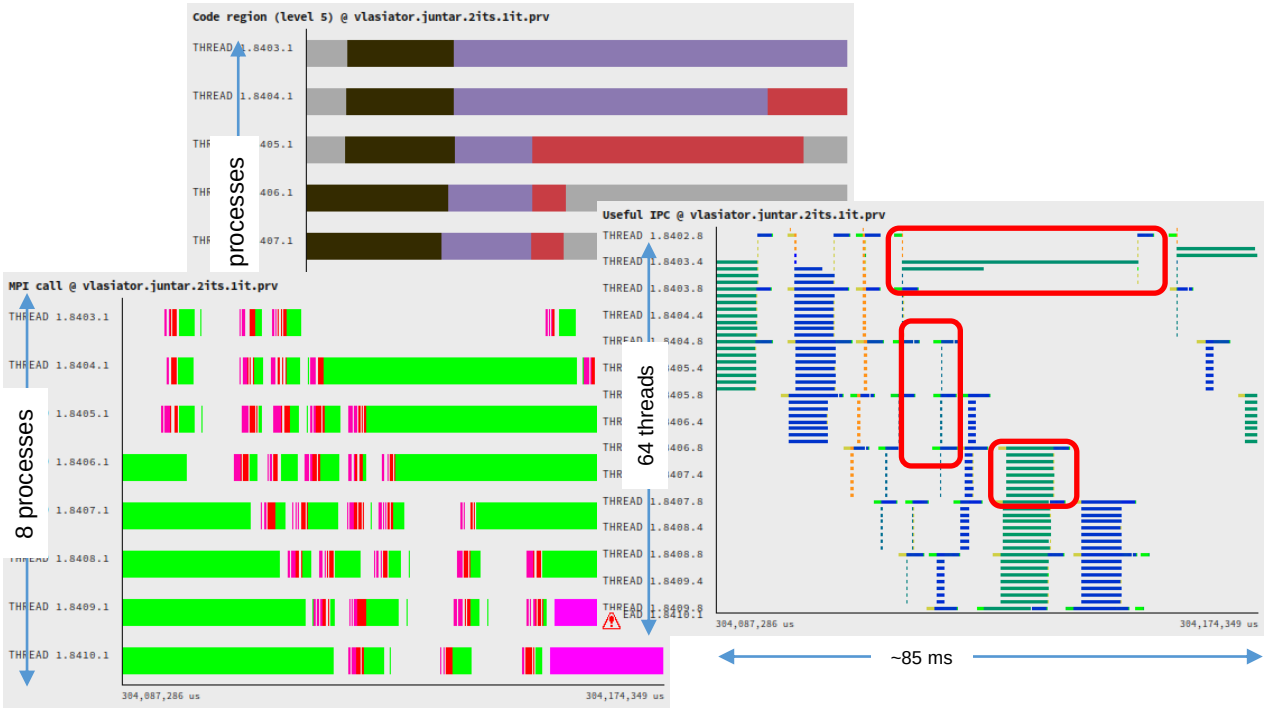


… and space …

… other views/metrics …



… till “microscopic” scale



Example 2: Small scale MPI+CUDA code

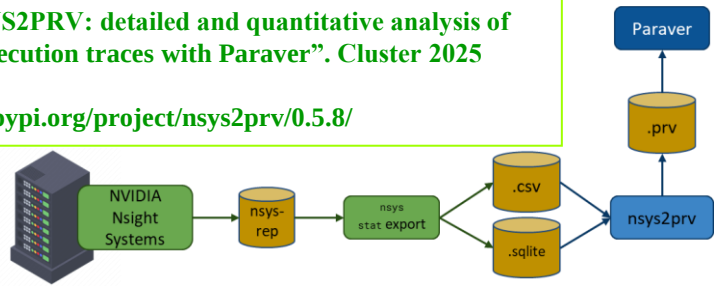
MPI + CUDA

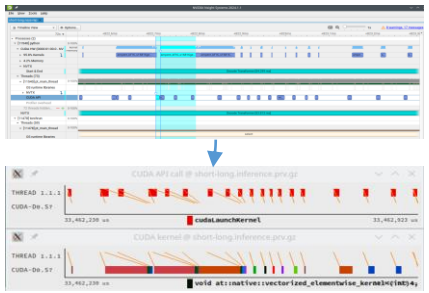
- GPU centric parallel program
- 32 GPUs

- nsys2prv

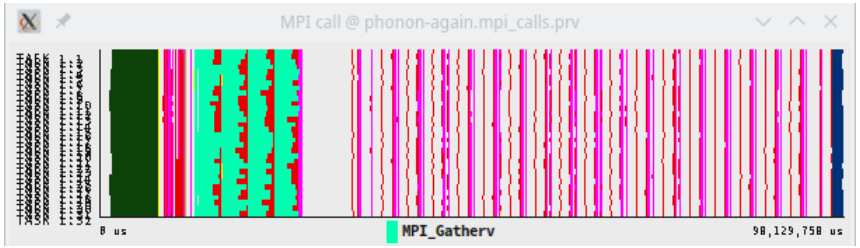
M. Clascá et al. “NSYS2PRV: detailed and quantitative analysis of large-scale GPU execution traces with Paraver”. Cluster 2025

https://pypi.org/project/nsys2prv/0.5.8/



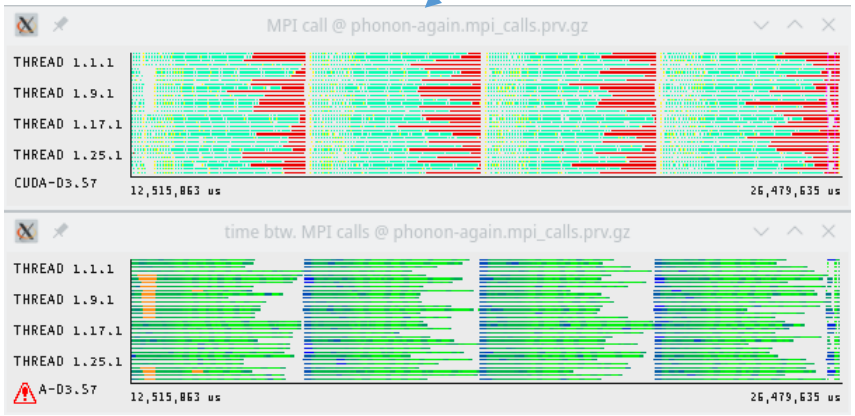


Structure

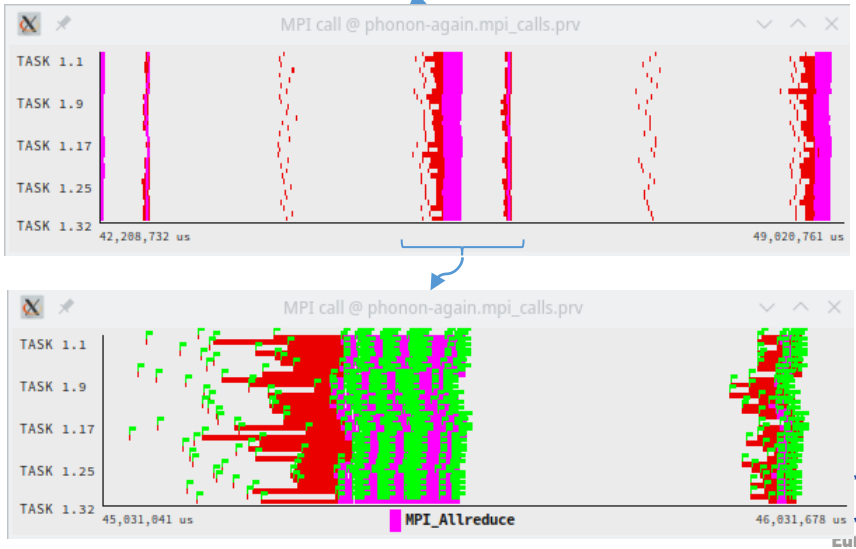


Full trace: ~ 23 GB
Filter MPI calls: 18 MB
Phase 2: 1.2 GB

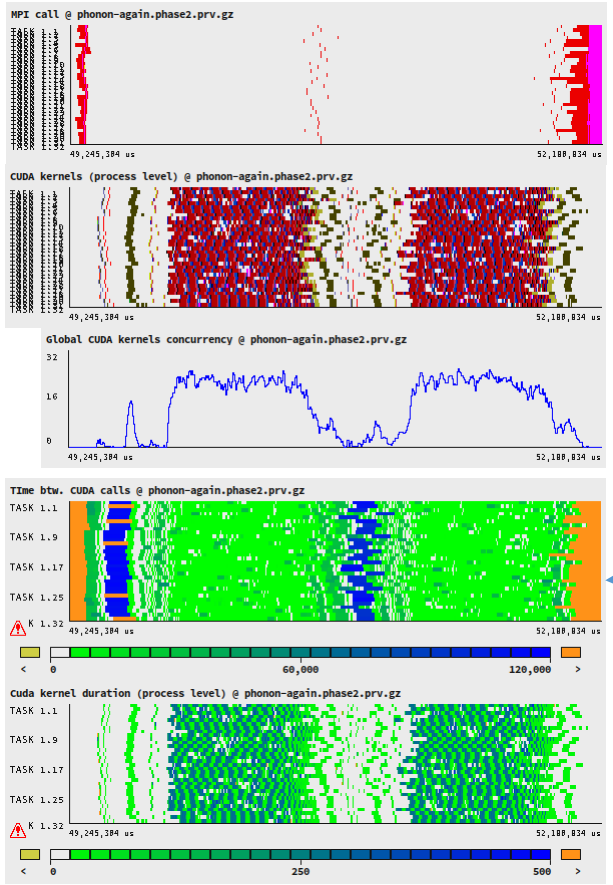
Phase 1

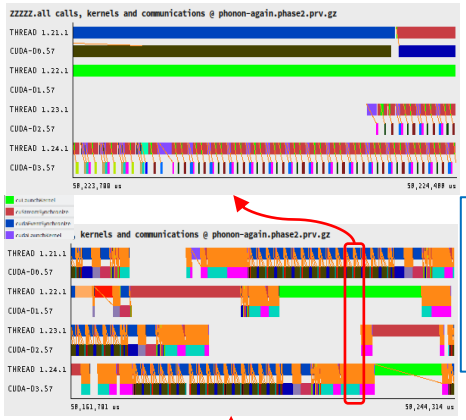


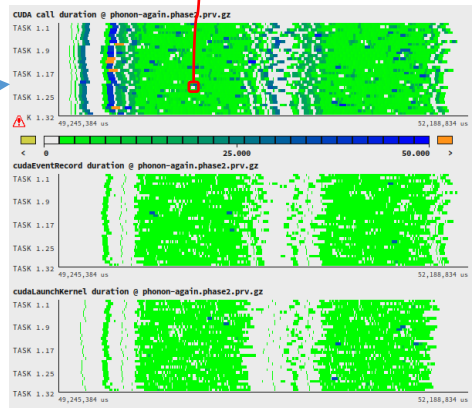
Phase 2



Phase 2







Full trace: ~ 23 GB
Filter MPI calls: 18 MB
1 it. phase 2: 1.2 GB

Single view with kernel launches, executes, …

Similar to other browsers

Not scalable, useful if in context

A sparse/anisotropic space
Resources x Time

There is also “noise” in GPU programming

Example 3: Medium scale AI training

From analyses with M. Clasca, A. Kumar, G. Kestor (BSC)

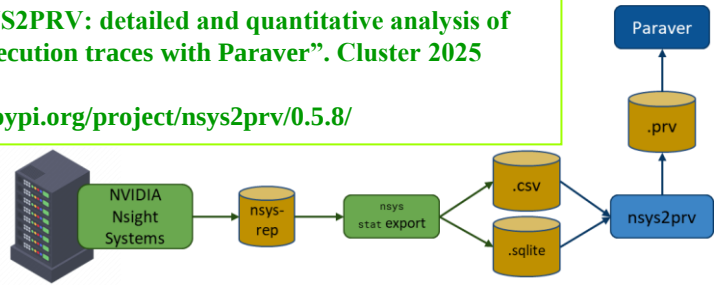
LLAMA fine tuning @ 128 GPUs

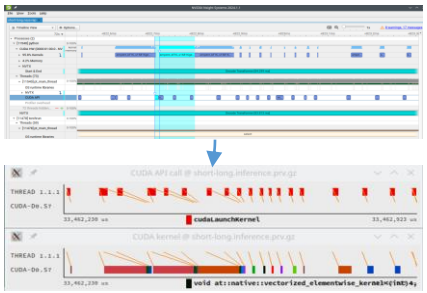
- Axolotl framework (DeepSpeed, ZeRO-2/3 data parallel)
- How it behaves?
  - Efficiency (% of peak, utilization of the resources,…) and causes
  - Does it have load imbalance ? Is communication a bottleneck ?

60 % of peak?
Good enough?

M. Clascá et al. “NSYS2PRV: detailed and quantitative analysis of large-scale GPU execution traces with Paraver”. Cluster 2025

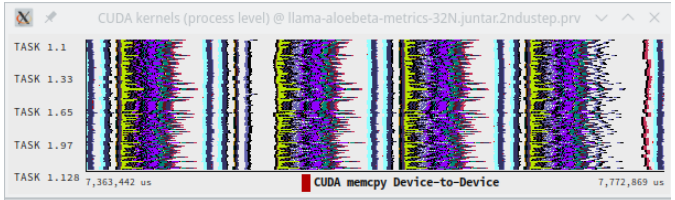
https://pypi.org/project/nsys2prv/0.5.8/



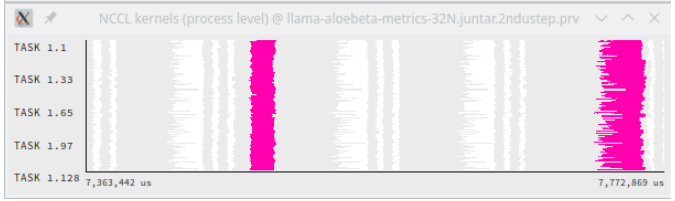


Sweeping across levels

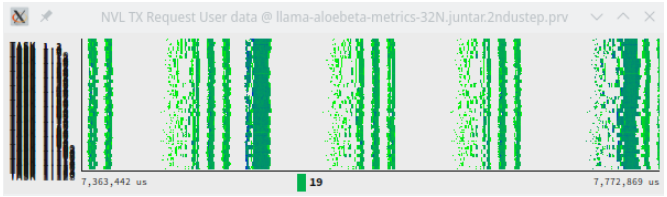
Compute kernels



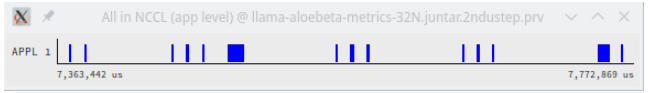
NCCL kernels



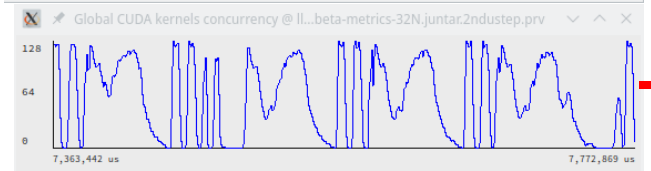
NVLINK utilization



All threads in collective



Aggregated # active compute kernels



Par. Eff:	0.476
LB:	0.728
Comm. Eff:	0.795
Orch. Eff::	0.824

Trace size: 2.3 GB

Aggregation
- “Basic”, predefined, …
- \+ semantics
- Combining different sources

The importance of flexibility in the browser

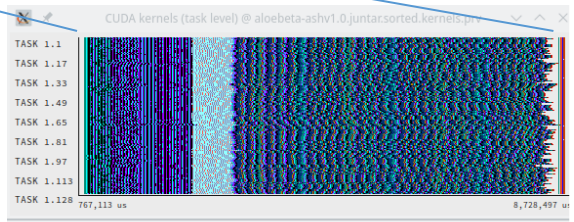
To overlap or not to overlap

Expected improvement … but similar time ☹

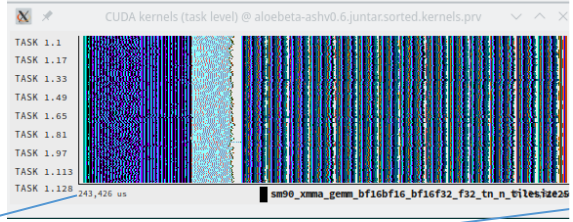
Full trace: ~ 3.2 GB
Filter kernels: 560 MB

GPU ‖ efficiency: 91.2 %
Load balance: 96 %
Comm. Eff.: 94.7 %
Comp. Eff.: 93.5 %

Overlapping



Non Overlapping

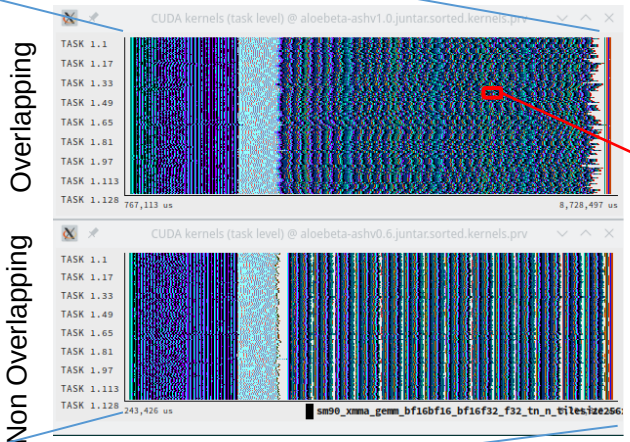


GPU ‖ efficiency: 85.3 %
Load balance: 94 %
Comm. Eff.: 90.3 %
Comp. Eff.: 100 %

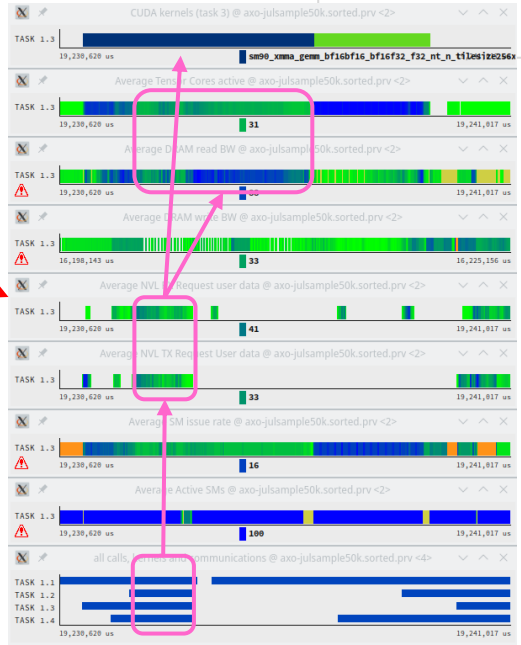
To overlap or not to overlap

Expected improvement … but similar time ☹

GPU ‖ efficiency: 91.2 %
Load balance: 96 %
Comm. Eff.: 94.7 %
Comp. Eff.: 93.5 %



GPU ‖ efficiency: 85.3 %
Load balance: 94 %
Comm. Eff.: 90.3 %
Comp. Eff.: 100 %



LOD data

Actually different run … same effects
only 4 GPUs
very high frequency sampled hwcs

Productivity support

Synchronized “multispectral” views

Interpretation

NVLINK activity …

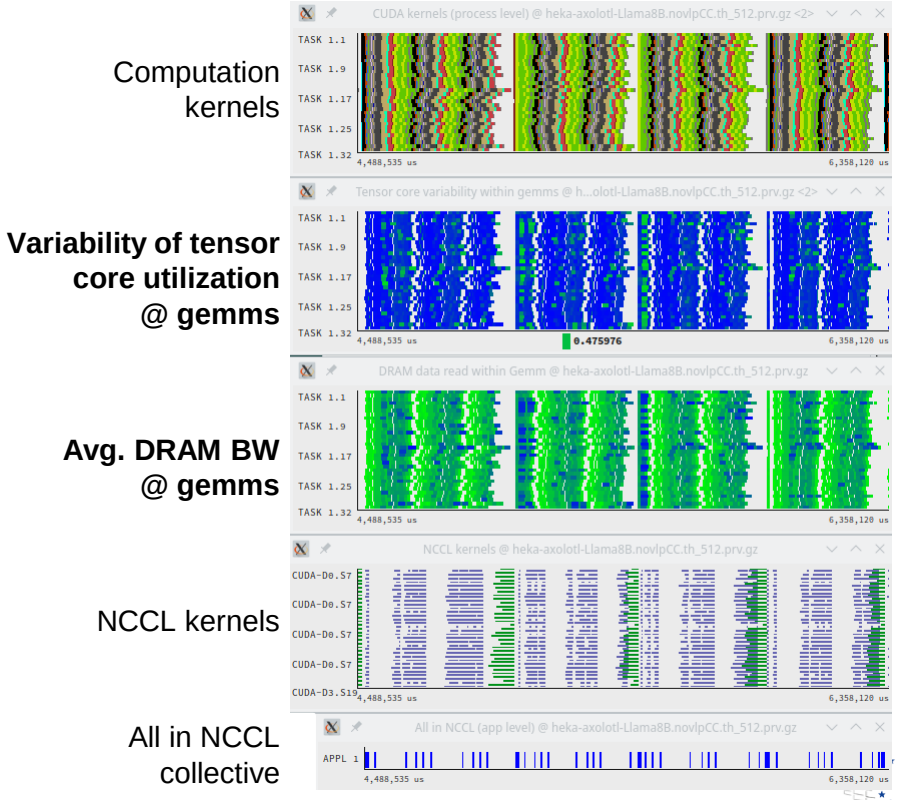
… causes reduction in tensor core activity …

… and rise in DRAM BW …

… till the kernel completes !!!!

Sweeping across levels

- Zooming out …
  - New metrics aggregating microscopic effect at selected granularity and with tailored semantics …
  - … to find further behavior at a more global scale …
- … and dig down again



" If I can get you close enough, can you track them?
Now that I know what to listen for, I will."
Hunt for Red October

Sweeping across levels

- Zooming out …
  - New metrics aggregating microscopic effect at selected granularity and with tailored semantics …
  - … to find further behavior at a more global scale …
- … and dig down again

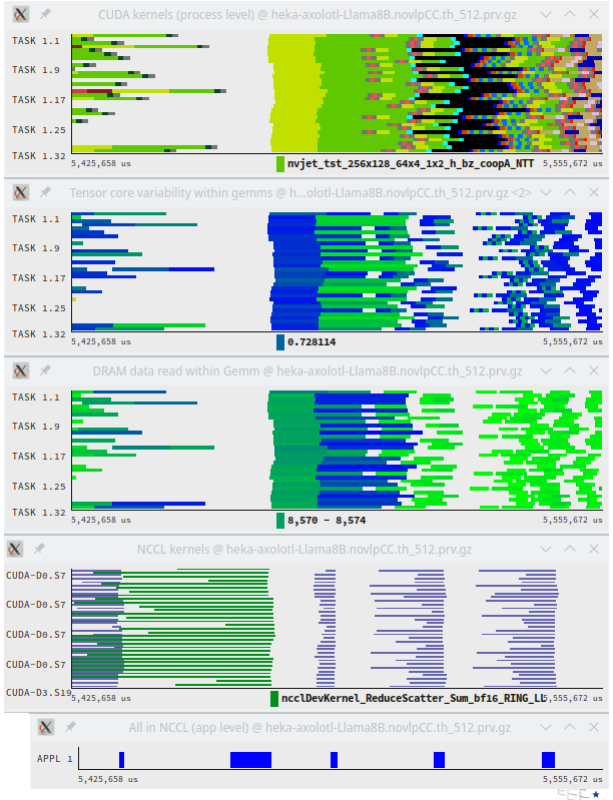
Computation kernels

Variability of tensor core utilization @ gemms

Avg. DRAM BW @ gemms

NCCL kernels

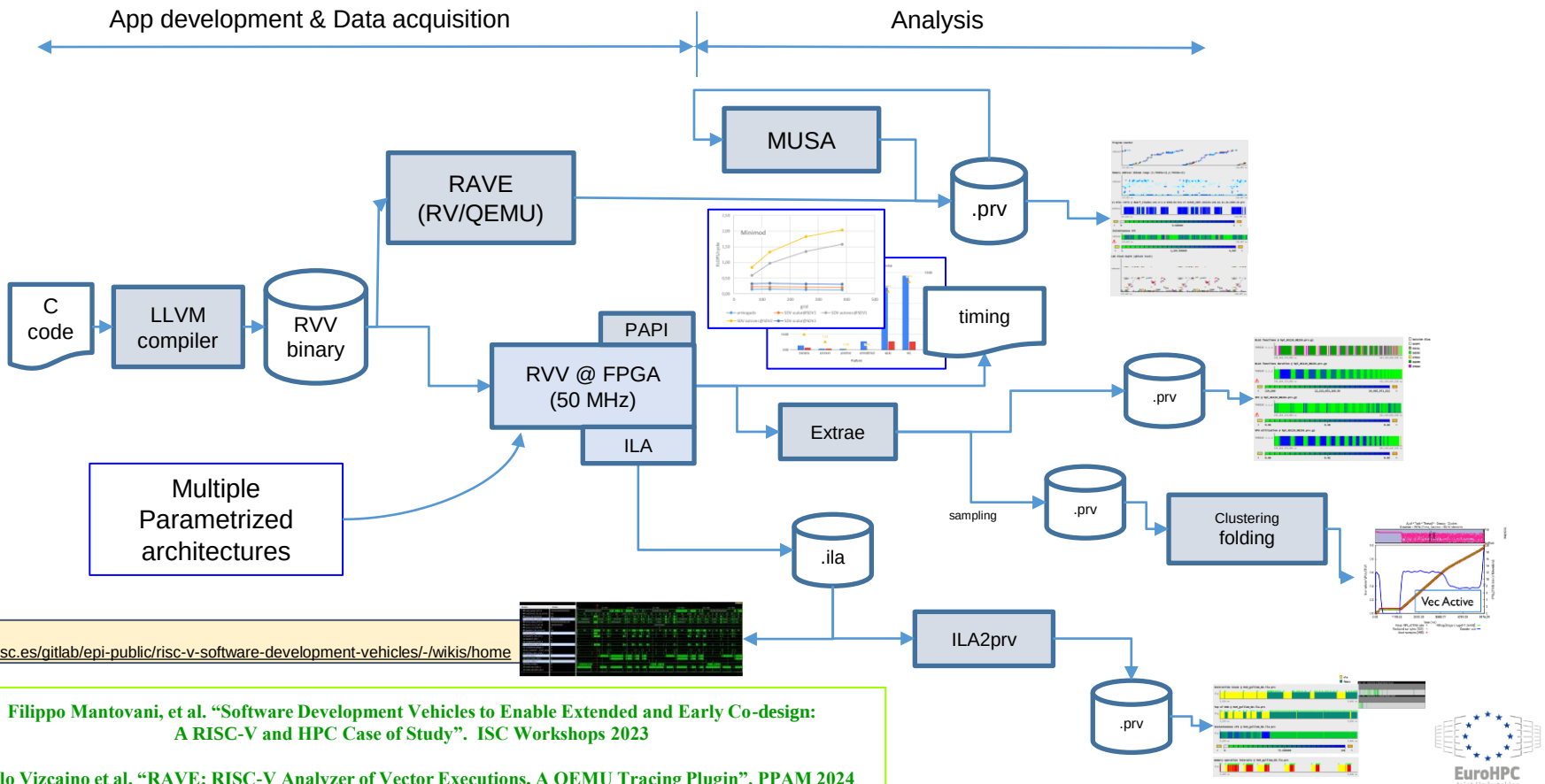
All in NCCL collective



> " If I can get you close enough, can you track them?
> Now that I know what to listen for, I will."
>
> Hunt for Red October

Example 4: Fine grain RISC-V vector co-design

LOD for “co-design”

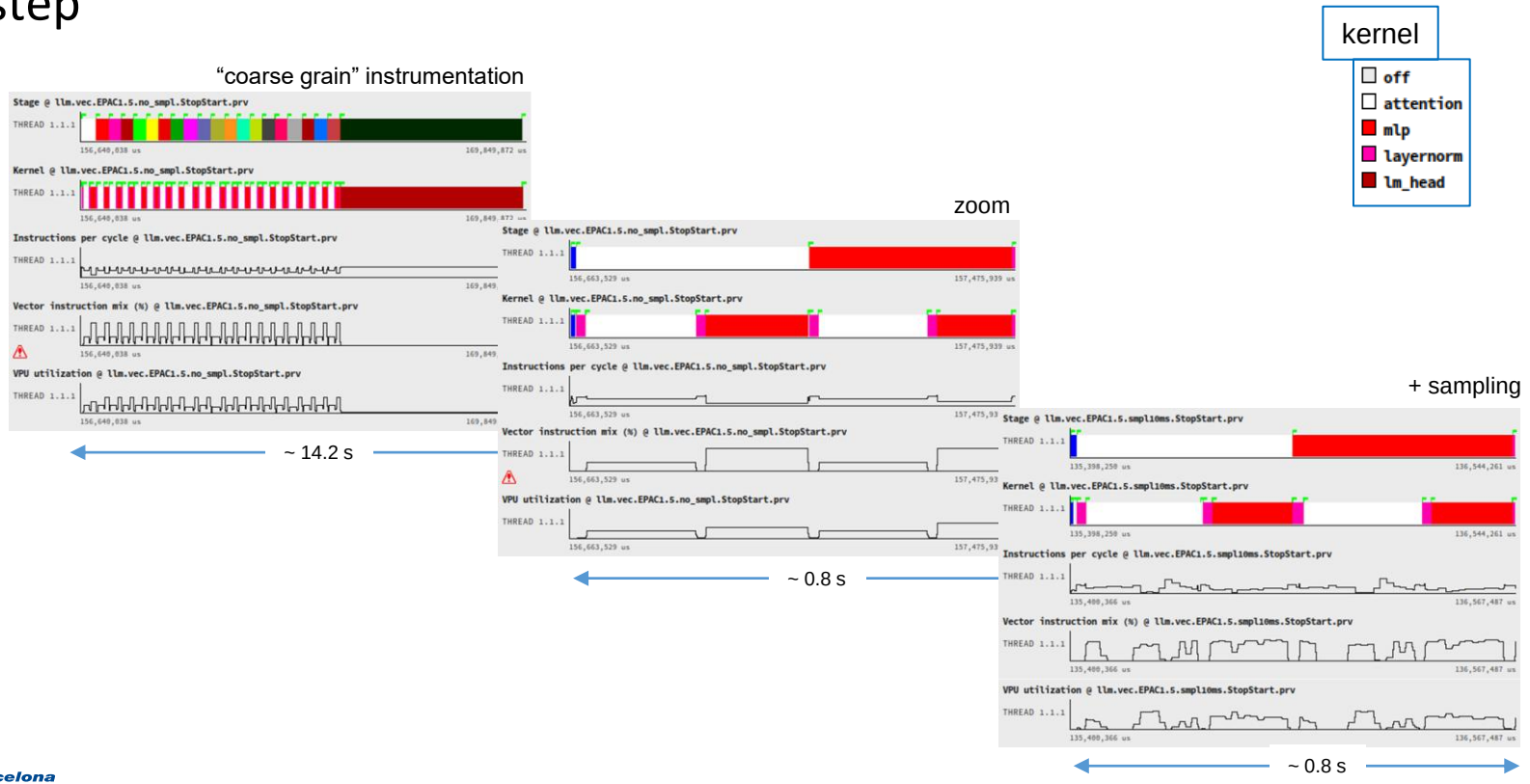


Filippo Mantovani, et al. “Software Development Vehicles to Enable Extended and Early Co-design: A RISC-V and HPC Case of Study”. ISC Workshops 2023

Pablo Vizcaino et al. “RAVE: RISC-V Analyzer of Vector Executions, A QEMU Tracing Plugin”. PPAM 2024

Llama @ EPAC1.5

- 1 step

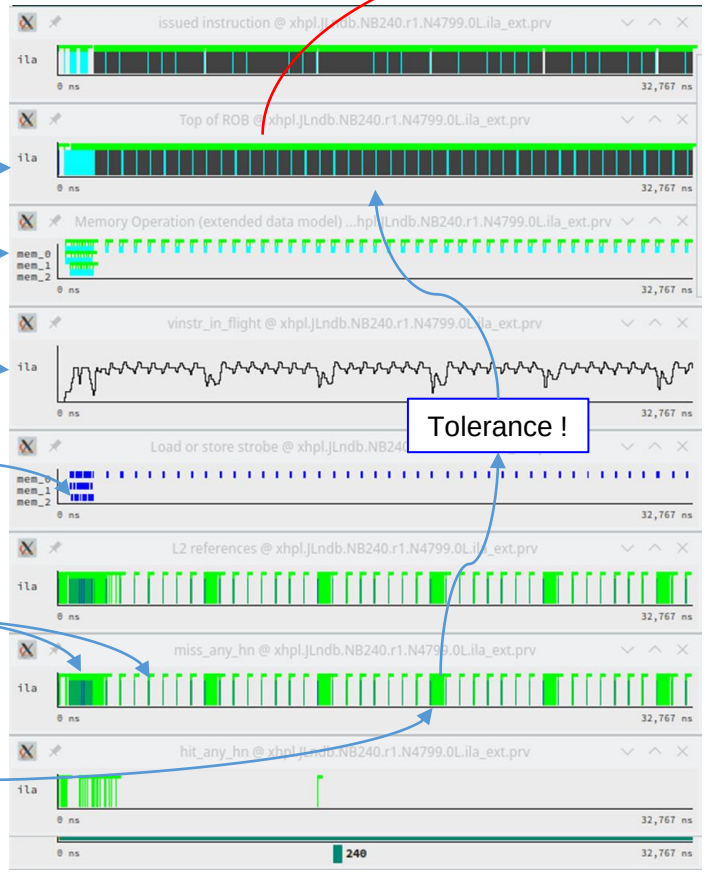


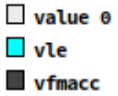
Cycle level look @ DGEMMs

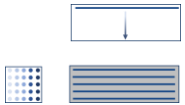
- Only 32K cycles

→ 98% of peak !!

All good?







Instruction at top of ROB

Execution of memory instr.

Instructions in flight @ VPU

Data to VPU
High BW ?(% of peak)

Vector loads: always miss

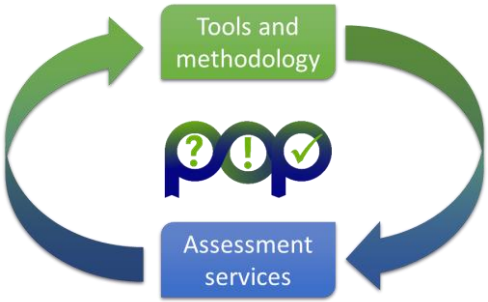
Periodic scalar misses

Tolerance !

To conclude …

POP

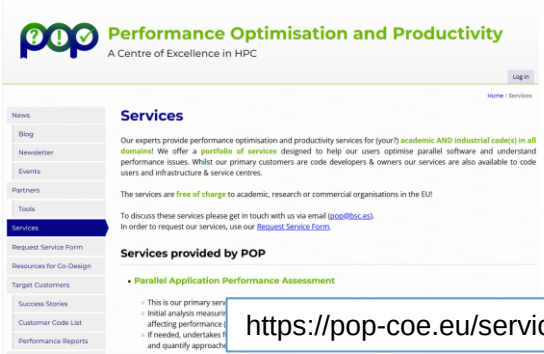
- Performance optimization and productivity COE



FREE Services provided by the CoE

- **Parallel Application Performance Assessment**
  - Primary service
  - Initial analysis measuring a range of performance metrics to assess quality of performance and identify the issues affecting performance (at customer site)
  - If needed, undertakes further performance evaluations to identify the root causes of the issues found and qualify and quantify approaches to address them (recommendations)
- **Second Level Services**
  - Second level services may follow after conclusion of an initial performance assessment:
    - **Proof-of-concept**: explore the potential benefit of proposed optimisations by applying them to selected regions of the applications
    - **Correctness-check**: evaluate the correctness of hybrid MPI + OpenMP applications
    - **Energy-efficiency study**: investigate improvements of energy consumption or efficiency
    - **Advisory study**: ongoing consultancy for customers that choose to implement proposed optimisations on their own
- Note: Effort shared between our experts and customer!

10



HORIZON-EUROHPC-JU-2023-COE

EuroHPC Joint Undertaking

Grant Agreement No 101143931

1 January 2024– 31 December 2026

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EuroHPC Joint Undertaking

Summary

- Because macroscopic behavior of nature depends on its microscopic structure/behavior …
- … scalable (**dynamic range**) analysis tools and methodologies …
- … able to integrate know-how at multiple levels …
- … are needed to really get the insight …
- … that can help us improve our systems and how we use them …
- … in the **HPC and AI worlds**

Keep dynamic range in mind

jesus.labarta@bsc.es

… and may be more

Performance Optimisation and Productivity 3

A Centre of Excellence in HPC

Contact:

- https://www.pop-coe.eu
- pop@bsc.es
- @POP_HPC
- youtube.com/POPHPC

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101143931. The JU receives support from the European Union's Horizon Europe research and innovation programme and Spain, Germany, France, Portugal and the Czech Republic.