

# SPH-EXA: A Framework for Scalable, Flexible, and Extensible Astrophysical and Cosmological Simulations

35<sup>th</sup> Workshop on Sustained (& Sustainable)  
Simulation Performance

Stuttgart, Germany  
April 14, 2023

Florina M. Ciorba

Florina Ciorba (PI) [florina.ciorba@unibas.ch](mailto:florina.ciorba@unibas.ch)

Ruben Cabezon (Co-PI) [ruben.cabezon@unibas.ch](mailto:ruben.cabezon@unibas.ch)

**Osman Seckin Simsek** [osman.simsek@unibas.ch](mailto:osman.simsek@unibas.ch)

Ahmed Eleliemy [ahmed.eleliemy@unibas.ch](mailto:ahmed.eleliemy@unibas.ch)

Lukas Schmidt [luke.schmidt@unibas.ch](mailto:luke.schmidt@unibas.ch)

José Escartin [jose.escartin@unibas.ch](mailto:jose.escartin@unibas.ch)

Lucio Mayer (Co-PI) [lmayer@physik.uzh.ch](mailto:lmayer@physik.uzh.ch)

Noah Kubli [noah.kubli@uzh.ch](mailto:noah.kubli@uzh.ch)

Darren Reed [darren.reed@uzh.ch](mailto:darren.reed@uzh.ch)

Sebastian Keller [sebastian.keller@cscs.ch](mailto:sebastian.keller@cscs.ch)

Jean-Guillaume Piccinali [jgp@cscs.ch](mailto:jgp@cscs.ch)

Jean Favre [jean.favre@cscs.ch](mailto:jean.favre@cscs.ch)

John Biddiscombe [john.biddiscombe@cscs.ch](mailto:john.biddiscombe@cscs.ch)

collaborators

Axel Sanz (UPC)

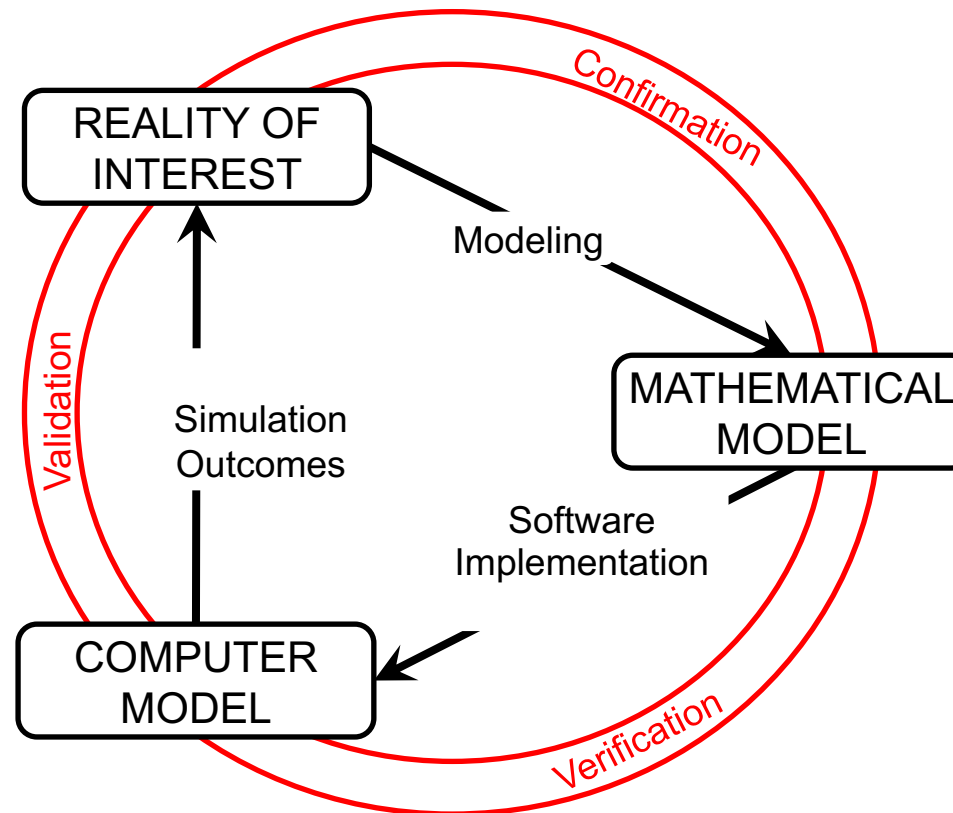
Joseph Touzet (Univ. of Paris-Saclay)





is a *scalable* and *fault tolerant*  
smoothed particle hydrodynamics simulation framework  
**interdisciplinarily co-designed** by computational physicists and computer  
scientists to exploit **Exascale** supercomputers.

# Classical Modeling and Simulation Philosophy



— Modeling and Simulation Activities

== Iterative Assessment Activities

[Adapted from: Schlesinger, S., "Terminology for Model Credibility," Simulation, Vol. 32, No. 3, 1979.]

# SPH-EXA Philosophy: Modeling, Simulation, and Optimization through Interdisciplinary Co-Design

## Parallelization

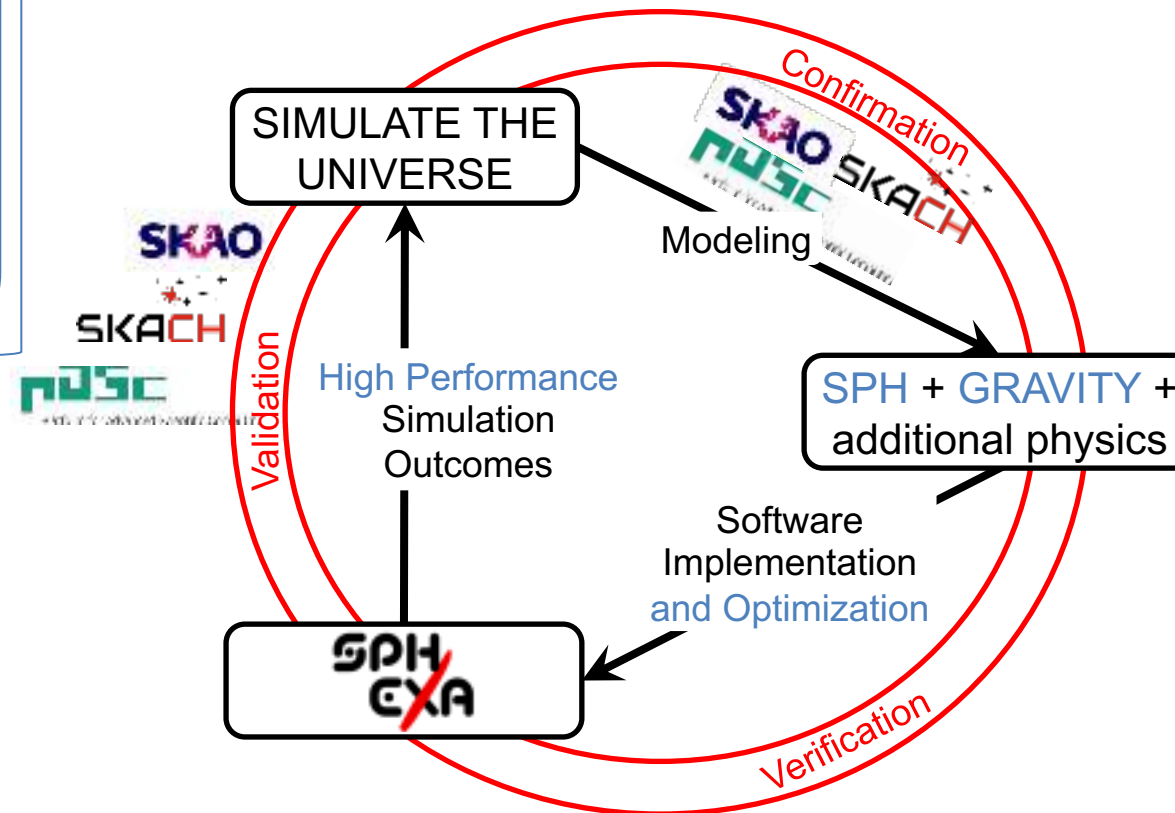
- Domain decomposition
- Multiprocess
- Multithreaded
- Acceleration
- Vectorization
- Overlapping comm. & comp.
- Scheduling and load balancing
- Parallel I/O
- ...

## Portability + Scalability

- Mixed precision
- Multiple accelerator models
- Minimal SW dependencies
- Cross-platform compilation, testing, benchmarking
- Fault tolerance
- Reproducibility
- ...

## Analysis + Visualization

- Performance profiling, tracing, visualization
- Scientific visualization via snapshots and in-situ
- ...



— Modeling, Simulation, and Optimization Activities  
= Iterative Assessment Activities

[Adapted from: Schlesinger, S., "Terminology for Model Credibility," Simulation, Vol. 32, No. 3, 1979.]

# SPH-EXA Philosophy: Modeling, Simulation, and Optimization through Interdisciplinary Co-Design

## Parallelization

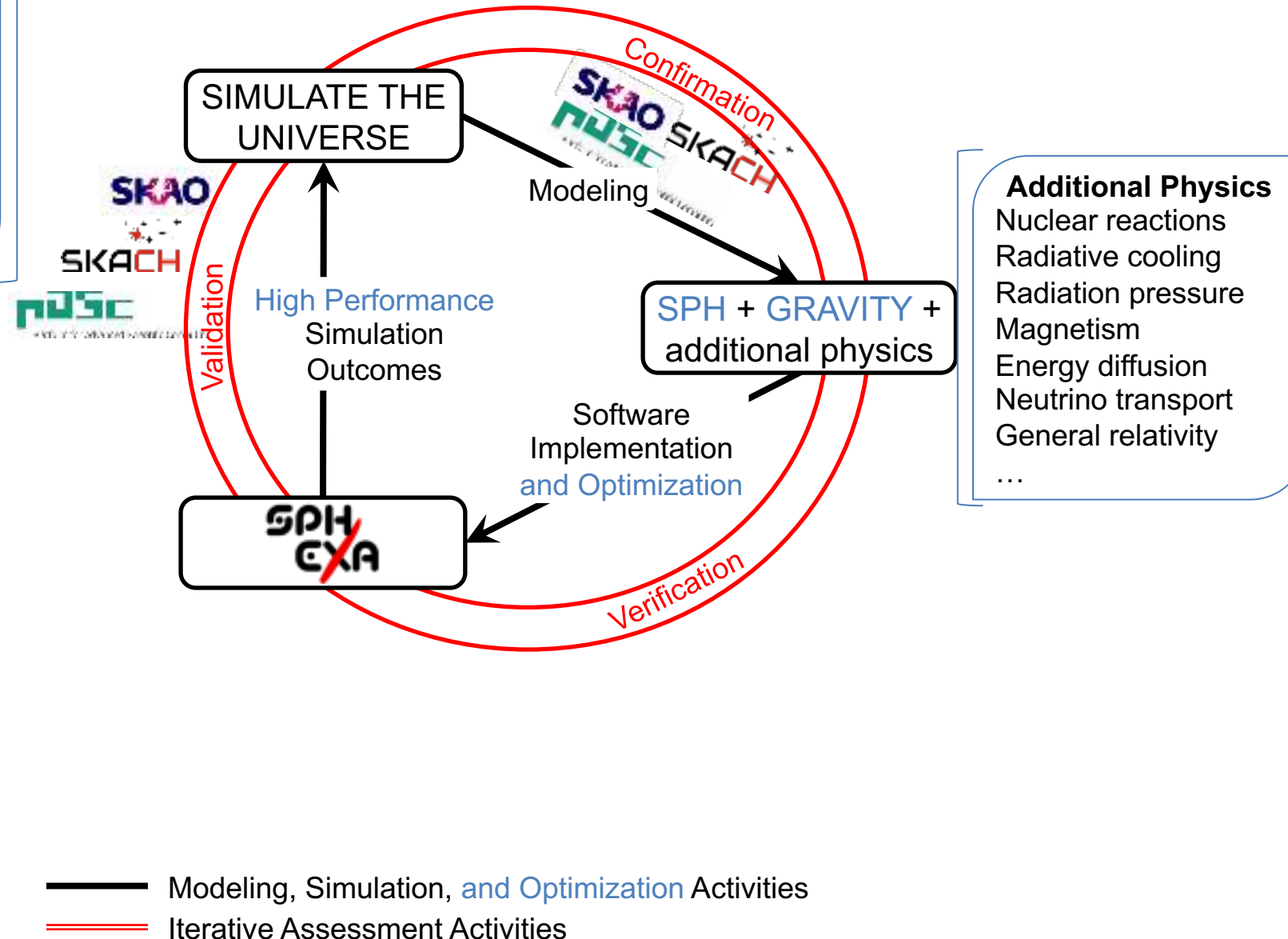
- Domain decomposition
- Multiprocess
- Multithreaded
- Acceleration
- Vectorization
- Overlapping comm. & comp.
- Scheduling and load balancing
- Parallel I/O
- ...

## Portability + Scalability

- Mixed precision
- Multiple accelerator models
- Minimal SW dependencies
- Cross-platform compilation, testing, benchmarking
- Fault tolerance
- Reproducibility
- ...

## Analysis + Visualization

- Performance profiling, tracing, visualization
- Scientific visualization via snapshots and in-situ
- ...



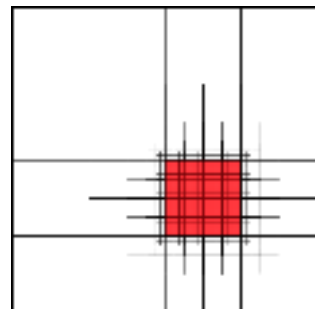
[Adapted from: Schlesinger, S., "Terminology for Model Credibility," Simulation, Vol. 32, No. 3, 1979.]

# SPH-EXA Framework Components



## SPH-EXA application front-end

*Parallel I/O and test case setup (ICs)*



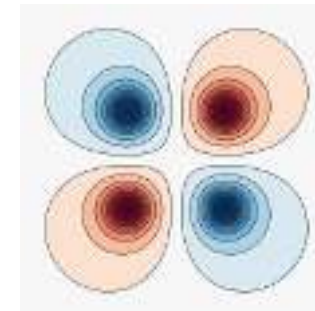
### Cornerstone

*Octree and domain  
decomposition framework*

SPHYNXC

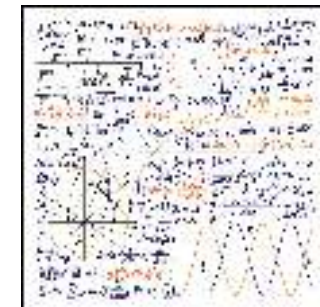
### SPH

*Hydrodynamics  
solver*



### Ryoanji

*N-body gravity  
solver*



### Physics modules

*Radiative cooling  
(GRACKLE)  
Nuclear reactions  
Star formation  
Stellar feedback  
(in development)*

...

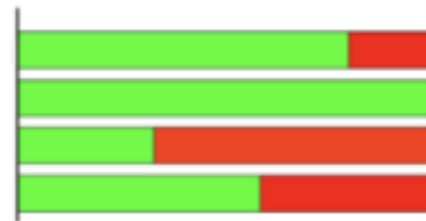


# SPH-EXA Performance Optimization Strategy



## Scalability

*weak, strong*



## Scheduling & Load-balancing

*dynamic, adaptive*  
*asynchronous*  
*execution*



## Heterogeneity

*portability on*  
*various*  
*CPU and GPU*  
*architectures*



## Fault-tolerance

*detection of and*  
*recovery from*  
*silent errors*

# SPH-EXA: Modular Software Design

- Each simulation has
  - an embedded initial conditions generator (if needed) and
  - its own propagator.
- Propagator may have components added / removed as needed.
- **Separation of concerns** between **domain decomposition**, **communication of particle data** (halo exchange) and **implementation of local and long-ranged forces** (gravity).

SPH-EXA  
framework

hydrodynamics

other physics

gravity

initial conditions setup

time-step loop

propagator

domain decomposition

halo exchange

SPH

gravity

additional physics

+ ...

time-step integration

I/O and visualization



# SPH-EXA: Propagators for Specific Test Cases

## Turbulence Propagator\*

## Evrard Collapse Propagator\*

## Turbulence + Gravity Propagator\*

Domain Decomposition & Sync

Find Neighbors

XMass

Normalization & Gradh

Equation of State

IADVelocityDivCurl

AV Switches

MomentumAndEnergy

Update Time-step

Turbulence stirring

Update Quantities

UpdateSmoothingLength

Domain Decomposition & Sync

Find Neighbors

XMass

Normalization & Gradh

Equation of State

IADVelocityDivCurl

AV Switches

MomentumAndEnergy

Tree Upsweep

Gravity

Update Time-step

Update Quantities

UpdateSmoothingLength

Domain Decomposition & Sync

Find Neighbors

XMass

Normalization & Gradh

Equation of State

IADVelocityDivCurl

AV Switches

MomentumAndEnergy

Tree Upsweep

Gravity

Update Time-step

Turbulence stirring

Update Quantities

UpdateSmoothingLength

syncHalos

syncHalos

syncHalos

syncHalos

\*CPU, CPU+GPU

SPH-EXA  
framework

hydrodynamics

other physics

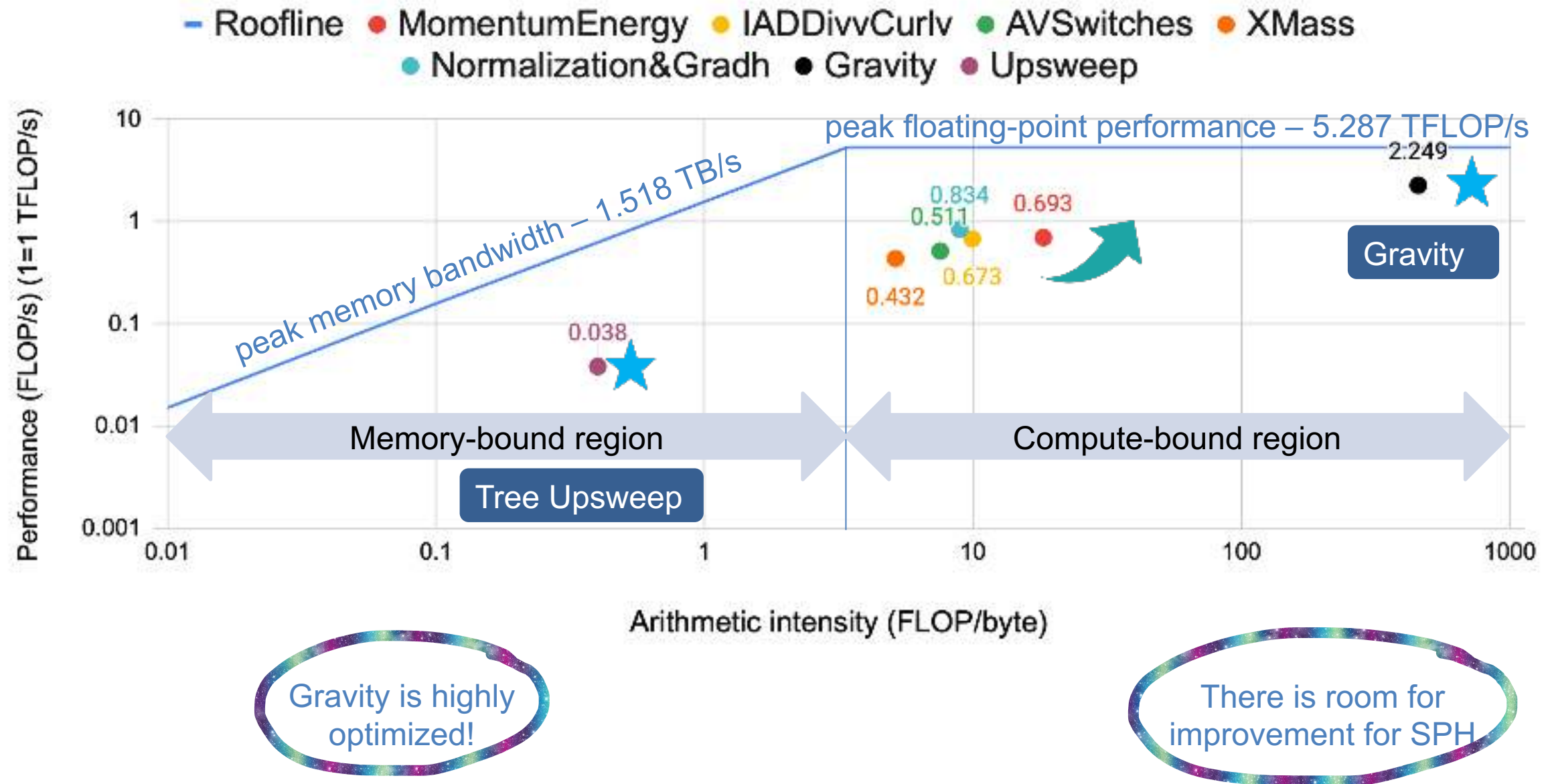
gravity



**Performance**

# Performance: Roofline Analysis for SPH-EXA

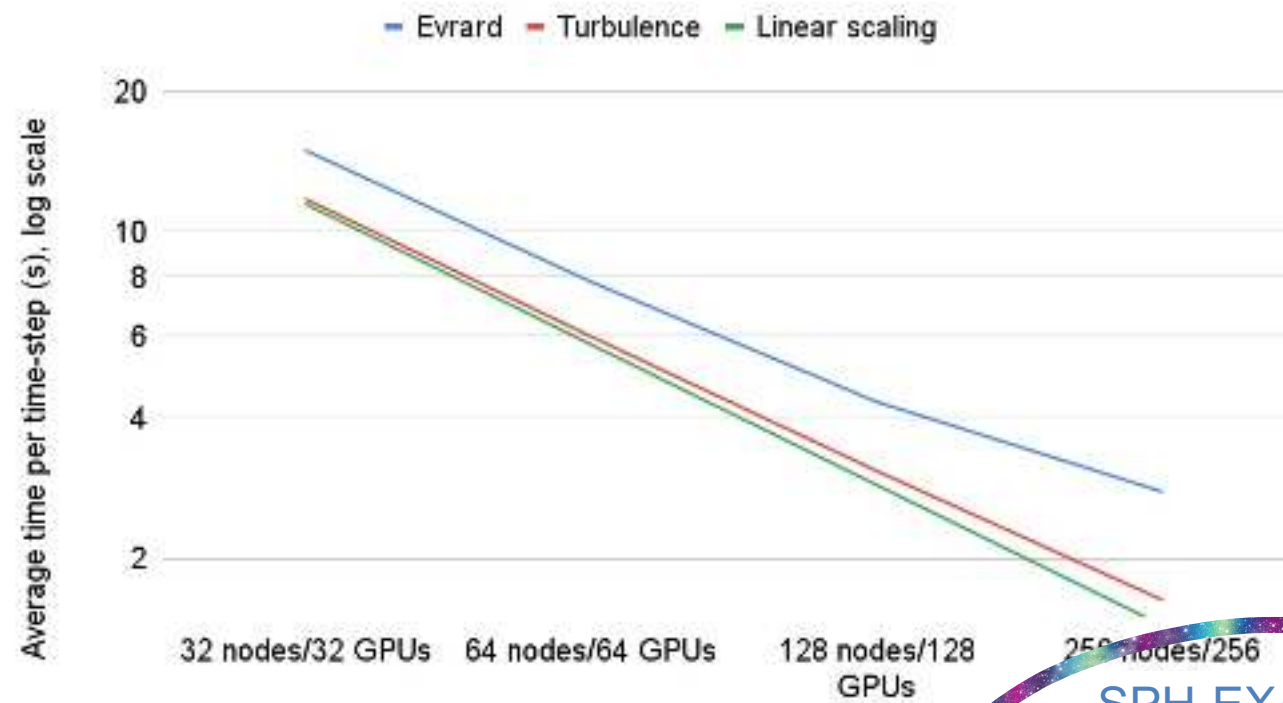
SPH-EXA Evrard collapse with 64M particles executed on a single Nvidia A100 GPU (*miniHPC*)



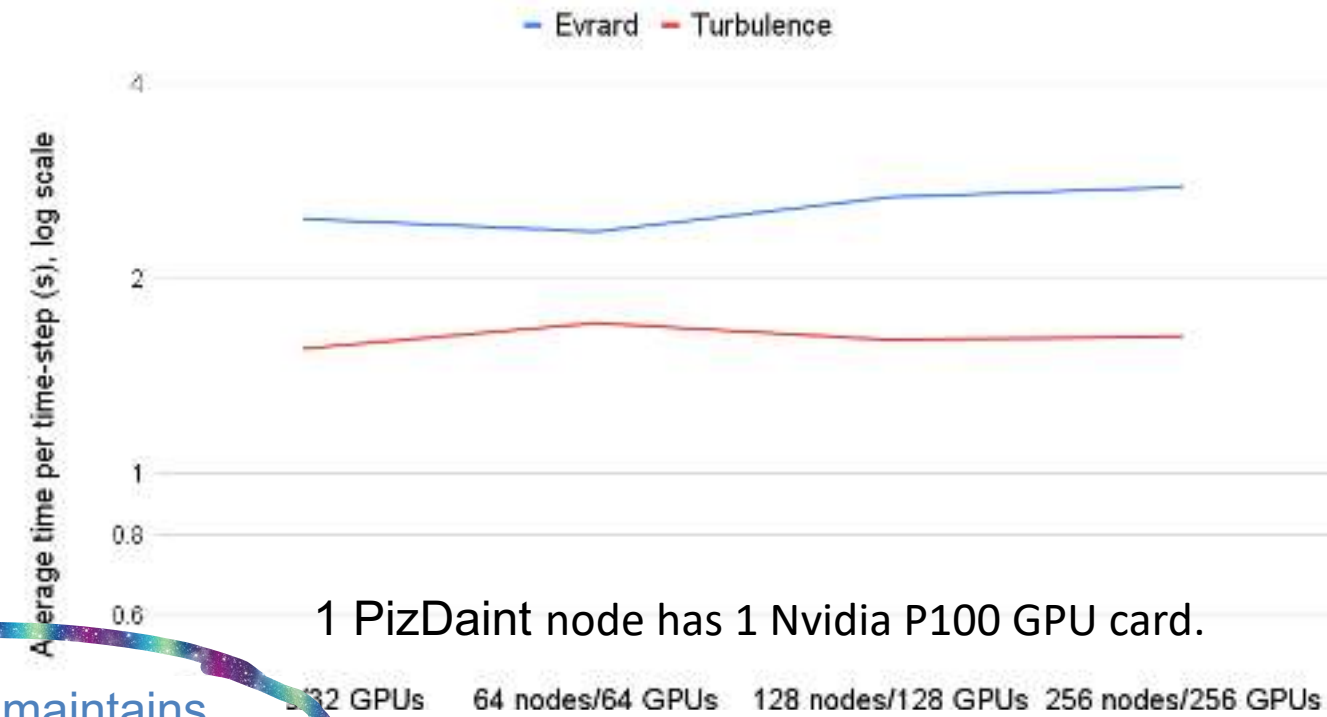


# Performance: Scaling of SPH-EXA Test Cases

Strong Scaling of SPH-EXA Tests on Piz Daint with 1B particles

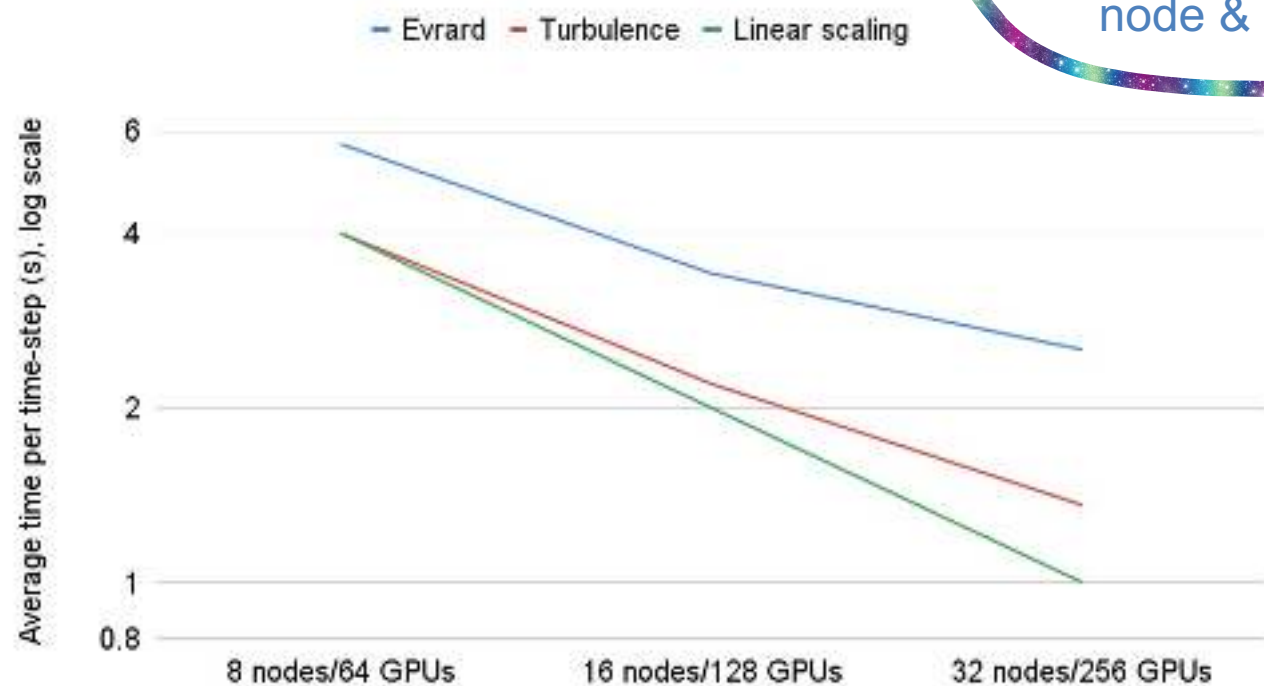


Weak Scaling of SPH-EXA Tests on Piz Daint with 4M particles per node

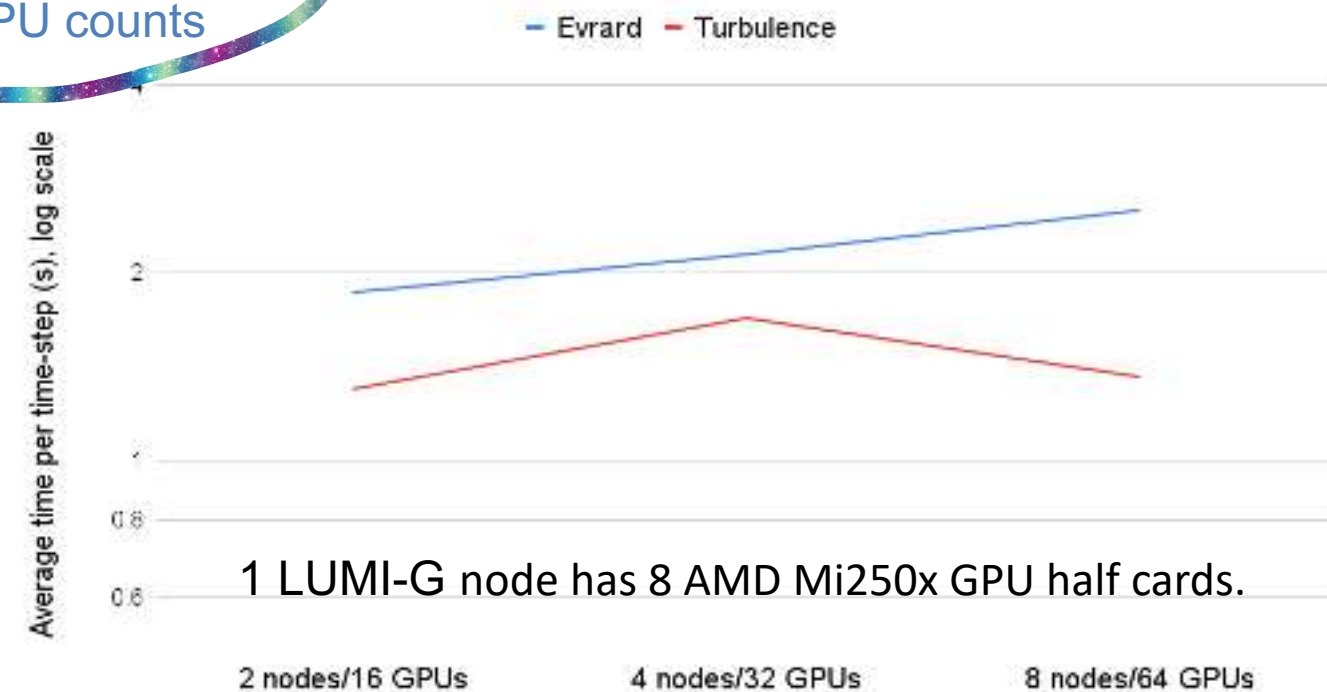


1 PizDaint node has 1 Nvidia P100 GPU card.

Strong Scaling of SPH-EXA Tests on LUMI with 1B particles



Weak Scaling of SPH-EXA Tests on LUMI with 32M particles per node/4M particles per GPU



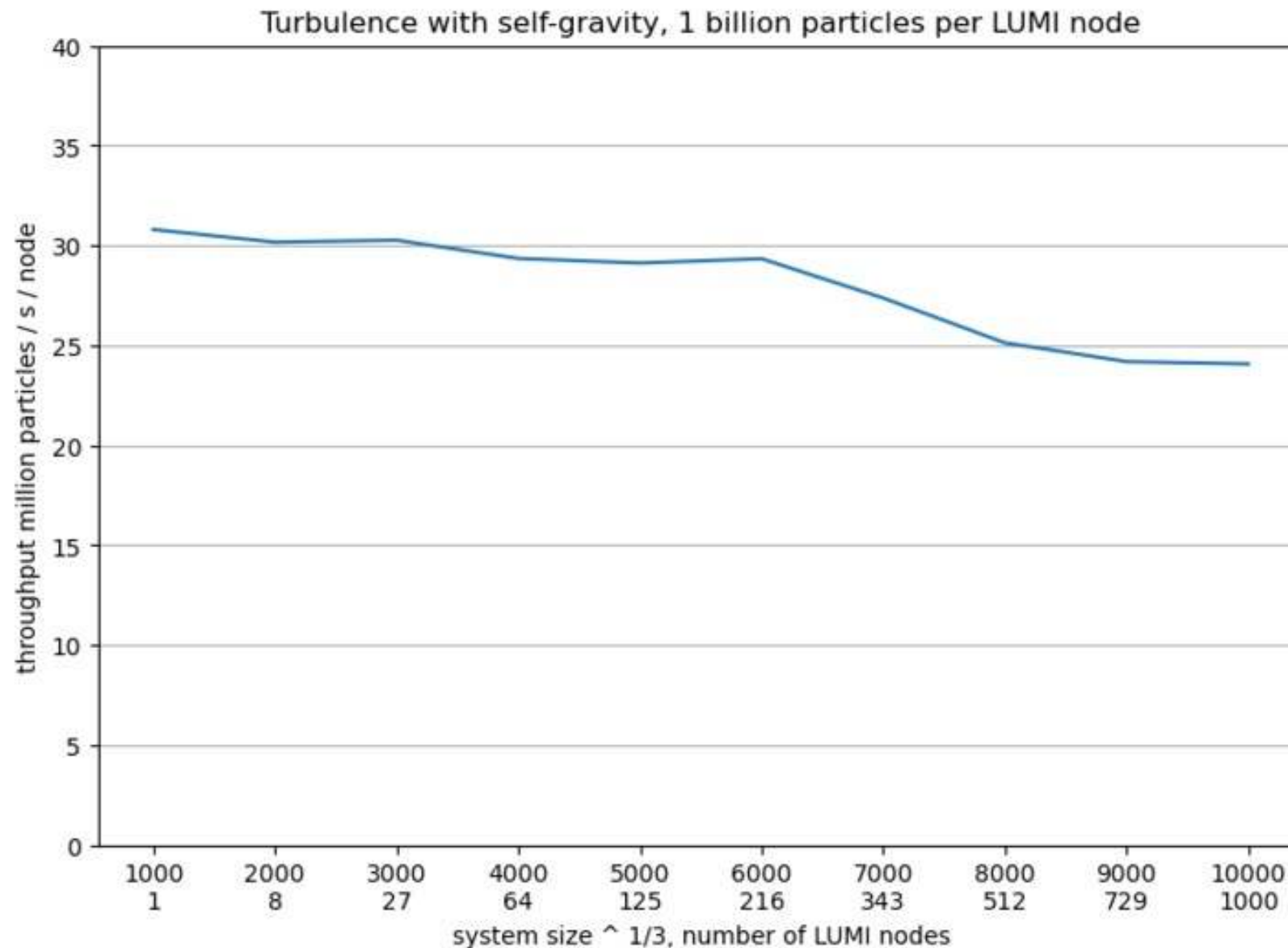
1 LUMI-G node has 8 AMD Mi250x GPU half cards.

SPH-EXA maintains performance at increasing node & GPU counts



# Performance: Weak Scaling Throughput of SPH-EXA Turbulence + Self-Gravity (*towards Stellar Formation*)

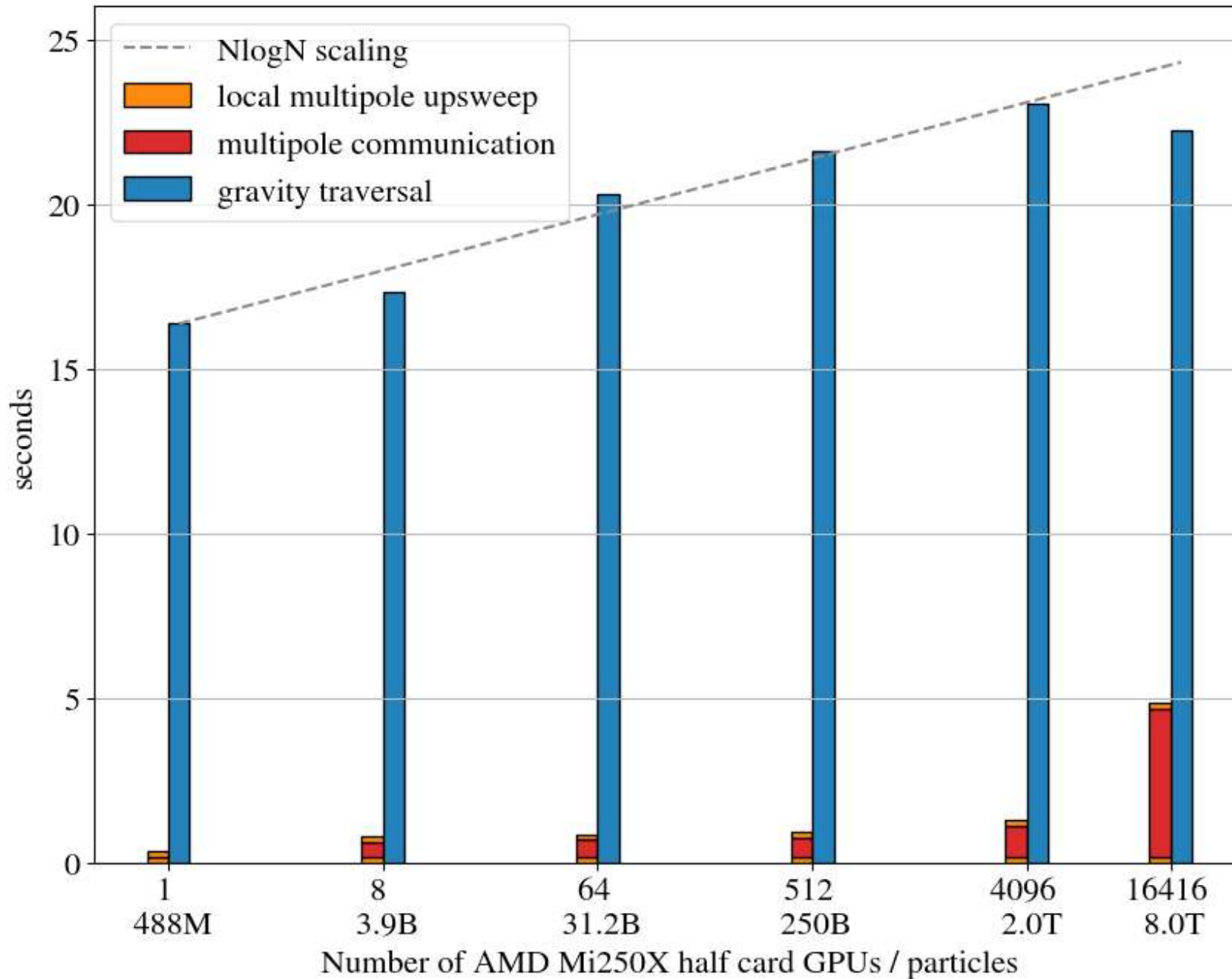
1B particles / node, on 1..1'000 **LUMI-G** nodes (each node has 8 AMD Mi250x GPU half cards),  
for a total of 1T particles ( $1'000^3$ ).



Note: Time per time-step (seconds) =  
1000 (million particles / node) /  
throughput (million particles / second / node).

# Performance: Weak Scaling of SPH-EXA Tree Code

**Barnes-Hut tree traversal (gravity)**, **local particle-particle (multipole upsweep)**, and **multipole communication** on LUMI-G with 488M particles/GPU and **8T particles in total** on 2'052 computing nodes (16'416 GPU half cards).

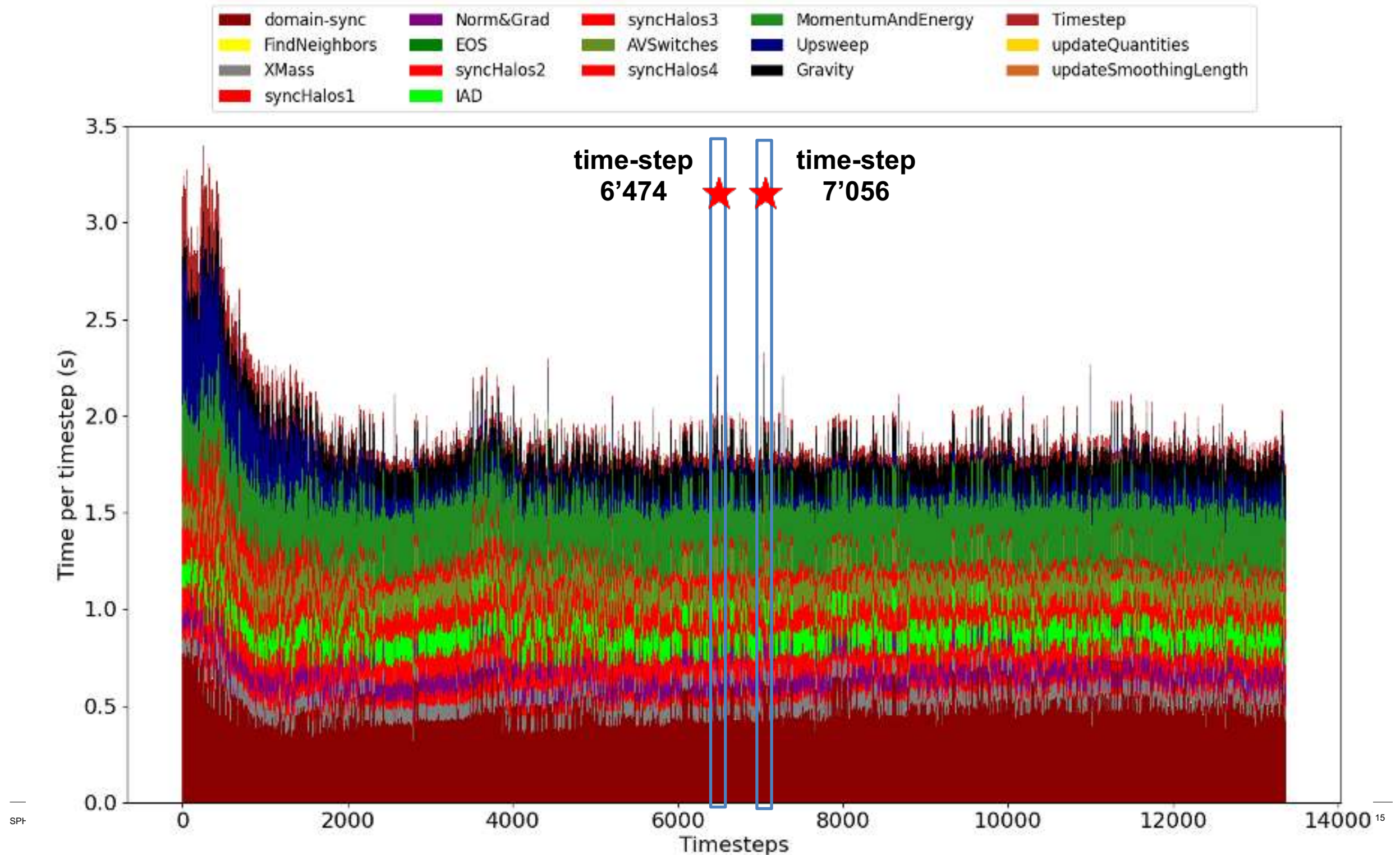


Approaching  
logarithmic  
weak scaling!

S. Keller, A. Cavelan, R. Cabezon,  
L. Mayer, and F. M. Ciorba,  
**Cornerstone: Octree  
Construction Algorithms for  
Scalable Particle Simulations.**  
PASC 2023 (to appear)

# Performance: Profile of Propagator Components Evrard Collapse

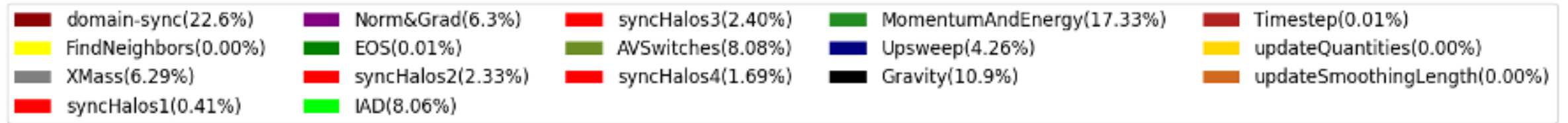
SPH-EXA Evrard collapse 1B particles 32 LUMI-G nodes (256 GPU half cards), 13'362 time-steps, rank 0/256



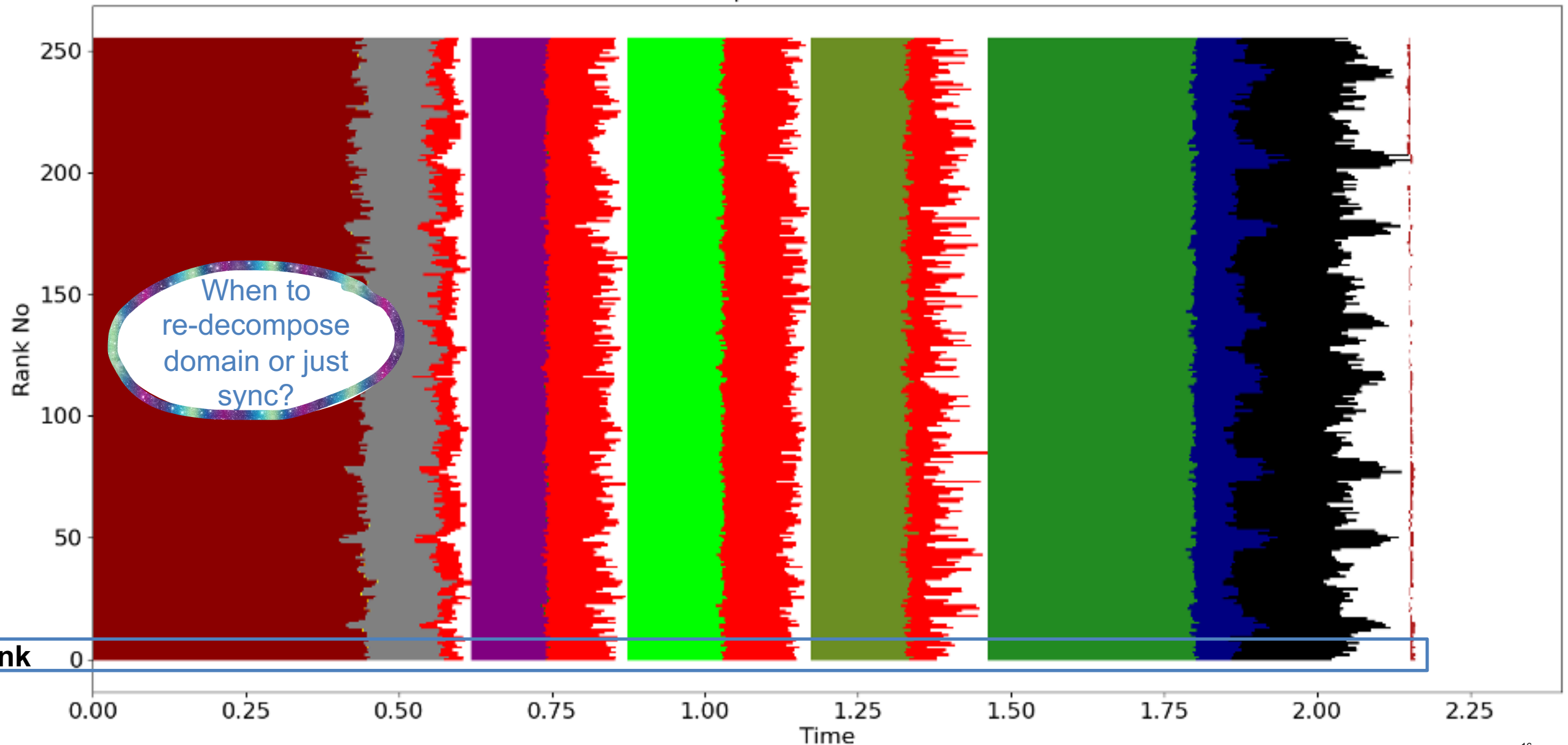


# Performance: Profile of Propagator Components Evrard Collapse

SPH-EXA Evrard collapse 1B particles 32 LUMI-G nodes (256 GPU half cards), zoom in on time-step 6'474

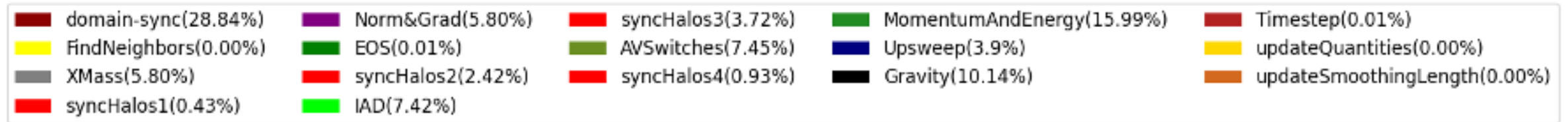


Timestep 6474 - Total idle time 9.36% \* 2.20 = 0.205 seconds

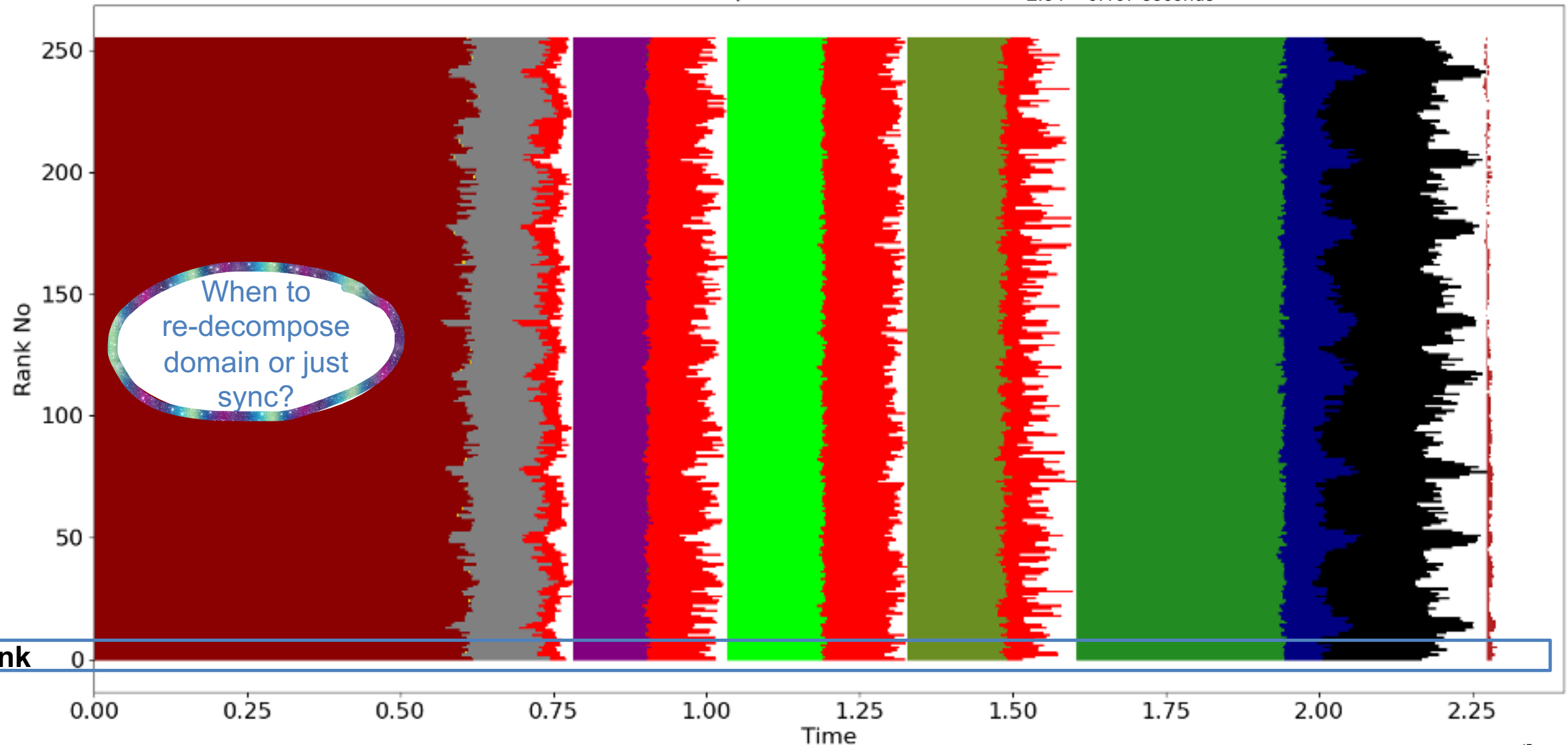


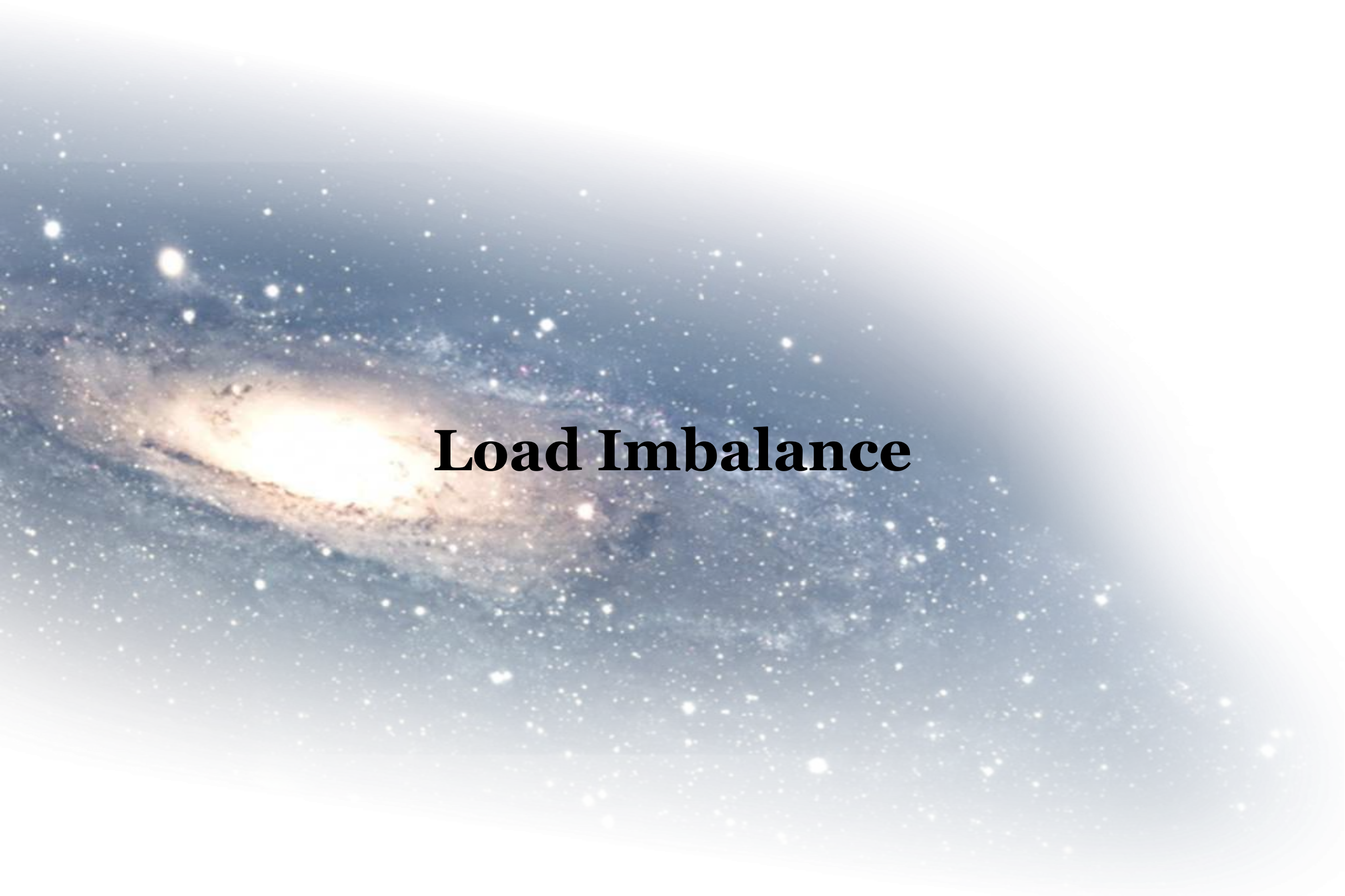
# Performance: Profile of Propagator Components Evrard Collapse

SPH-EXA Evrard collapse 1B particles 32 LUMI-G nodes (256 GPU half cards), zoom in on time-step 7'056



Timestep 7056 - Total idle time 7.15% \* 2.34 = 0.167 seconds





# **Load Imbalance**



# Performance: Load Imbalance

Which metric  
is better?

How to find out **where**  
load imbalance occurs?

SPH-EXA Evrard collapse 1B particles 32 LUMI-G nodes (256 GPU half cards), zoom in on time-step 7'056

| Function                 | <i>mu</i> [s] | <i>sigma</i> [s]   |
|--------------------------|---------------|--------------------|
|                          | mean          | standard deviation |
| Domain-sync              | 0.606389      | 0.013153           |
| FindNeighbors            | 0.000166      | 0.000176           |
| XMass                    | 0.121879      | 0.003699           |
| <b>synchronizeHalos1</b> | 0.032528      | 0.008045           |
| Normalization&Gradh      | 0.121920      | 0.003168           |
| EquationOfState          | 0.000208      | 0.000024           |
| synchronizeHalos2        | 0.098366      | 0.014876           |
| IadVelocityDivCurl       | 0.156083      | 0.003412           |
| synchronizeHalos         | 0.118142      | 0.009928           |
| AVswitches               | 0.156683      | 0.005781           |
| <b>synchronizeHalos3</b> | 0.058787      | 0.024282           |
| MomentumAndEnergy        | 0.336191      | 0.003839           |
| <b>Upsweep</b>           | 0.081960      | 0.022270           |
| <b>Gravity</b>           | 0.181756      | 0.074035           |
| Timestep                 | 0.031484      | 0.013568           |
| UpdateQuantities         | 0.000012      | 0.000002           |
| UpdateSmoothingLength    | 0.000007      | 0.000002           |

# Performance: Load Imbalance

Which metric  
is better?

How to find out **where**  
load imbalance occurs?

SPH-EXA Evrard collapse 1B particles 32 LUMI-G nodes (256 GPU half cards), zoom in on time-step 7'056

| Function                 | <i>mu</i> [s]<br>mean | <i>sigma</i> [s]<br>standard deviation | <b>lambda</b> [%]<br>percent load imbalance | <b>c.o.v.</b><br>coefficient of variation | <b>g1</b> [s]<br>skewness | <b>g2</b> [s]<br>kurtosis | <b>   _2</b><br>spatial load imbalance |
|--------------------------|-----------------------|----------------------------------------|---------------------------------------------|-------------------------------------------|---------------------------|---------------------------|----------------------------------------|
| Domain-sync              | 0.606389              | 0.013153                               | 4.02533                                     | 0.021690                                  | -0.651420                 | -0.030760                 | 0.347215                               |
| FindNeighbors            | 0.000166              | 0.000176                               | 1441.97                                     | 1.065210                                  | 10.447184                 | 133.10157                 | 17.04340                               |
| XMass                    | 0.121879              | 0.003699                               | 9.12331                                     | 0.030352                                  | -0.132935                 | -0.388142                 | 0.485597                               |
| <b>synchronizeHalos1</b> | 0.032528              | 0.008045                               | 84.0751                                     | 0.247323                                  | 0.770605                  | 0.351868                  | 3.957170                               |
| Normalization&Gradh      | 0.121920              | 0.003168                               | 7.15826                                     | 0.025981                                  | -0.158050                 | 0.095049                  | 0.415701                               |
| EquationOfState          | 0.000208              | 0.000024                               | 108.483                                     | 0.116774                                  | 6.291246                  | 49.594312                 | 1.868390                               |
| synchronizeHalos2        | 0.098366              | 0.014876                               | 29.2261                                     | 0.151226                                  | -0.362067                 | -0.037816                 | 2.419610                               |
| IadVelocityDivCurl       | 0.156083              | 0.003412                               | 5.2726                                      | 0.021861                                  | -0.264229                 | 0.271389                  | 0.349742                               |
| synchronizeHalos         | 0.118142              | 0.009928                               | 19.3858                                     | 0.084032                                  | -0.328625                 | -0.341294                 | 1.344530                               |
| AVswitches               | 0.156683              | 0.005781                               | 9.8906                                      | 0.036896                                  | -0.027327                 | -0.192141                 | 0.590239                               |
| <b>synchronizeHalos3</b> | 0.058787              | 0.024282                               | 124.847                                     | 0.413044                                  | 0.401849                  | -0.385256                 | 6.608710                               |
| MomentumAndEnergy        | 0.336191              | 0.003839                               | 2.92188                                     | 0.011419                                  | -0.613625                 | 0.081833                  | 0.182783                               |
| <b>Upsweep</b>           | 0.081960              | 0.022270                               | 71.4701                                     | 0.271711                                  | 0.533585                  | -0.767855                 | 4.347380                               |
| <b>Gravity</b>           | 0.181756              | 0.074035                               | 85.7485                                     | 0.407330                                  | -0.054193                 | -1.302345                 | 6.895250                               |
| Timestep                 | 0.031484              | 0.013568                               | 15.3133                                     | 0.077221                                  | -0.026883                 | -1.088475                 | 1.235540                               |
| UpdateQuantities         | 0.000012              | 0.000002                               | 70.1588                                     | 0.138918                                  | 1.680663                  | 3.873821                  | 2.222690                               |
| UpdateSmoothingLength    | 0.000007              | 0.000002                               | 103.561                                     | 0.257074                                  | 1.444387                  | 1.811151                  | 4.113190                               |

severity of load  
imbalance

extent of variability  
in relation to the  
mean  
>1: sigma > mu

lack of symmetry:  
++: above mean,  
--: below mean  
load

large: few but  
extreme outliers  
small: many but  
modest outliers

magnitude of  
load imbalance  
across the vector  
of processes

# Performance: Load Imbalance

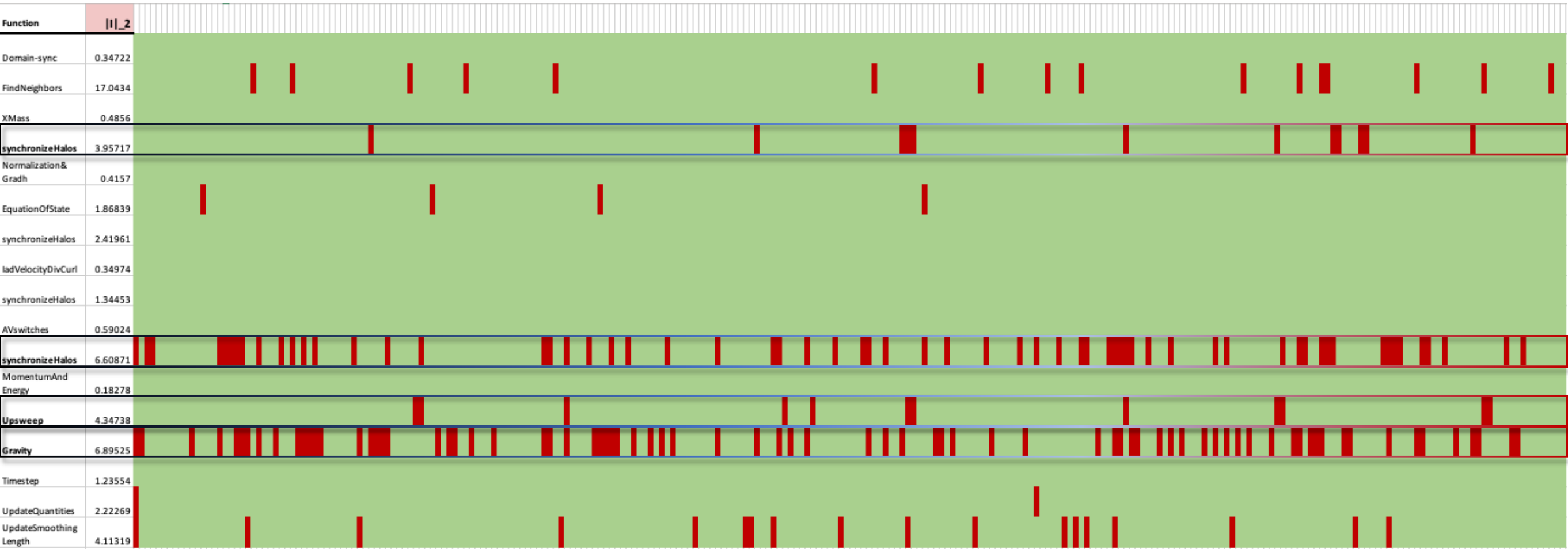


What causes  
load imbalance on  
arbitrary ranks?

SPH-EXA Evrard collapse 1B particles 32 LUMI-G nodes (256 GPU half cards), zoom in on time-step 7'056

Spatial imbalance per rank, measured with  $|\vec{l}|_2$

Ranks 0-255







**Sustainability**

# Sustainability: Estimated Energy Needed for SPH-EXA on LUMI

“Hero” Runs on LUMI-G (Finland)

12 hours on Dec 19, 2022

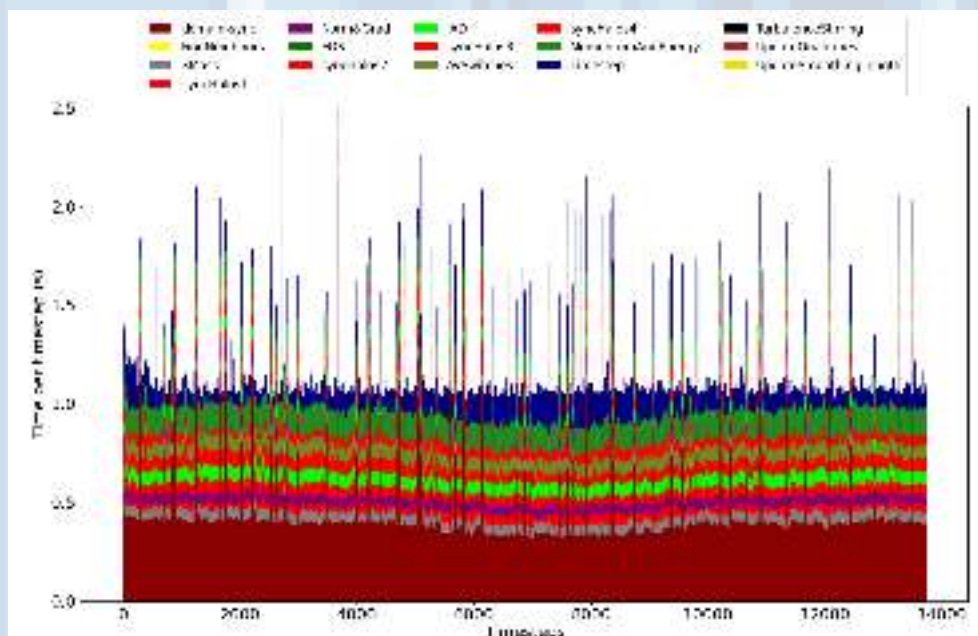
2'052 AMD EPYC 7A53 CPUs

16'416 AMD Instinct MI250X 64GB GPUs

PUE 1.03

SPH-EXA Tests

- **Turbulence simulation** (no gravity)
- **Pure gravity weak scaling** (tree code on earlier slide)



Calculator: <https://www.green-algorithms.org>

Details about your computation

To understand how eco your carbon footprint, check the details about the LUMI-G compute center!

Carbon-neutral compute center!

5.83e+02e Carbon footprint

61.11 MWh Energy needed

Runtime (HH:MM)

Type of cores: Both

CPUs

Number of cores: 131328

Model: Other

What is the Thermal Design Power (TDP) value per core of your CPU? This can easily be found online (usually 10-15W per core):

4375

GPUs

Number of GPUs: 16416

Model: Other

What is the Thermal Design Power (TDP) value per core of your GPU? This can easily be found online (usually around 200W):

200

Memory available (in GB): 2101346

Select the platform used for the computations:

Local server

Select location:

Europe

Finland

Do you know the real usage factor of your GPU?

Yes No

Do you know the Power Usage Efficiency (PUE) of your local data centre?

Yes No

529.55 tree-years Carbon sequestration

3.33e+04 km in a passenger car

2.5 flights NYC-Melbourne

Computing cores VS Memory

Donut chart showing GPU 80% and Memory 15.2%.

How the location impacts your footprint

Bar chart showing Emissions (gCO2e) for various locations: Switzerland, France, USA, Canada, United Kingdom, China, India, Australia.

Slide not for social media!



# Sustainability: Estimated Energy Needed for SPH-EXA on Frontier

“Hero” Runs on Frontier (US, Tennessee)

A **hypothetical** 12 hours run

2'052 AMD EPYC 7A53 CPUs

16'416 AMD Instinct MI250X 64GB GPUs

PUE 1.03

SPH-EXA Tests

- **Turbulence simulation** (no gravity)
- **Pure gravity weak scaling** (tree code on earlier slide)



# Sustainability: Estimated Energy Needed for SPH-EXA on Piz Daint

“Hero” Runs on Piz Daint (Switzerland)

A **hypothetical** 12 hours run

2’052 Intel® Xeon® E5-2690 v3 CPUs  
2’052 NVIDIA® Tesla® P100 16GB GPUs

PUE 1.20

SPH-EXA Tests (**smaller runs** than LUMI)

- **Turbulence simulation** (no gravity)
- **Pure gravity weak scaling**  
(tree code on earlier slide)

On other carbon-positive  
compute centers, parallelism  
adds up all the carbon footprint ☹️  
**Need to account for CO<sub>2</sub>e to  
sustain simulation  
performance!**





# What's Next for SPH-EXA?

~~SPH-EXA~~

Further Increase **scalability**  
(optimize SPH, decompose/sync,  
balance load execution, mix precision)  
and **fault tolerance**.

Account for **carbon footprint**  
in design of experiments!

**Green Algorithms**  
towards environmentally sustainable computational science

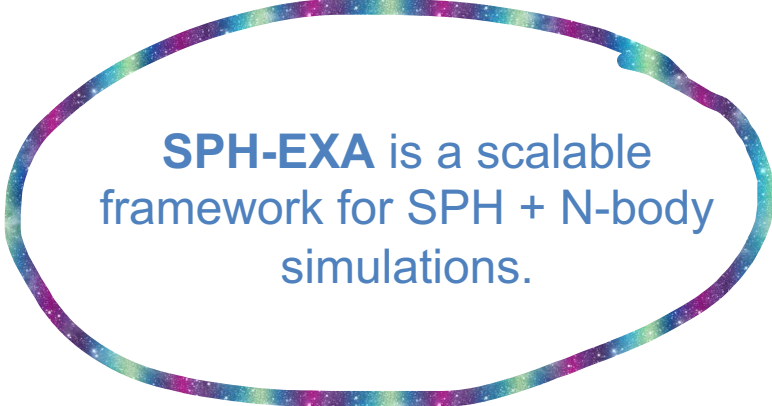
**SKACH** **POSC**  
Platform for Advanced Scientific Computing

Simulate **PHOEBOS**,  
**Supernovas**, etc. ...  
(additional physics).

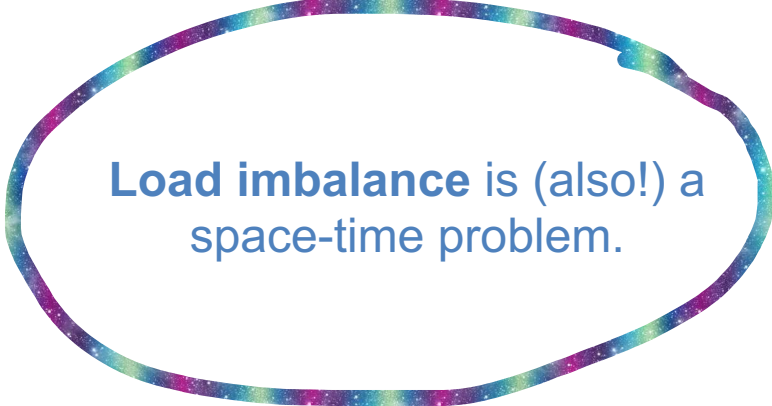
Combine visualization of  
**scientific & performance**  
**data**.



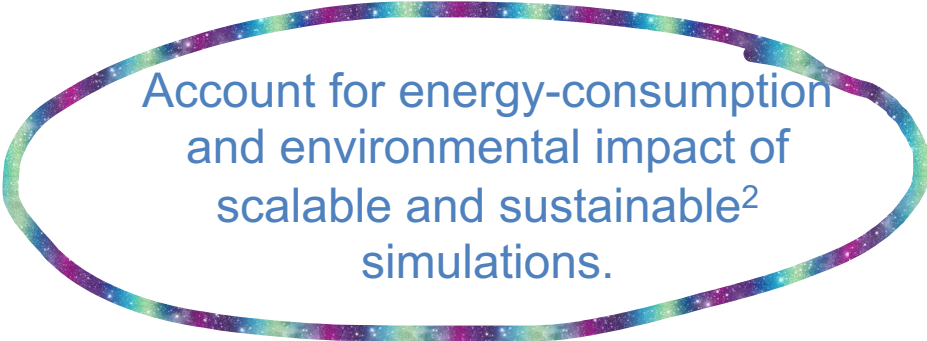
# Take aways



**SPH-EXA** is a scalable framework for SPH + N-body simulations.



**Load imbalance** is (also!) a space-time problem.



Account for energy-consumption and environmental impact of scalable and sustainable<sup>2</sup> simulations.

# SPH-EXA: A Framework for Scalable, Flexible, and Extensible Astrophysical and Cosmological Simulations

35<sup>th</sup> Workshop on Sustained (& Sustainable)  
Simulation Performance

Stuttgart, Germany  
April 14, 2023

Florina M. Ciorba

Florina Ciorba (PI) [florina.ciorba@unibas.ch](mailto:florina.ciorba@unibas.ch)

Ruben Cabezon (Co-PI) [ruben.cabezon@unibas.ch](mailto:ruben.cabezon@unibas.ch)

**Osman Seckin Simsek** [osman.simsek@unibas.ch](mailto:osman.simsek@unibas.ch)

Ahmed Eleliemy [ahmed.eleliemy@unibas.ch](mailto:ahmed.eleliemy@unibas.ch)

Lukas Schmidt [luke.schmidt@unibas.ch](mailto:luke.schmidt@unibas.ch)

José Escartin [jose.escartin@unibas.ch](mailto:jose.escartin@unibas.ch)

Lucio Mayer (Co-PI) [lmayer@physik.uzh.ch](mailto:lmayer@physik.uzh.ch)

Noah Kubli [noah.kubli@uzh.ch](mailto:noah.kubli@uzh.ch)

Darren Reed [darren.reed@uzh.ch](mailto:darren.reed@uzh.ch)

Sebastian Keller [sebastian.keller@cscs.ch](mailto:sebastian.keller@cscs.ch)

Jean-Guillaume Piccinali [jgp@cscs.ch](mailto:jgp@cscs.ch)

Jean Favre [jean.favre@cscs.ch](mailto:jean.favre@cscs.ch)

John Biddiscombe [john.biddiscombe@cscs.ch](mailto:john.biddiscombe@cscs.ch)

collaborators

Axel Sanz (UPC)

Joseph Touzet (Univ. of Paris-Saclay)