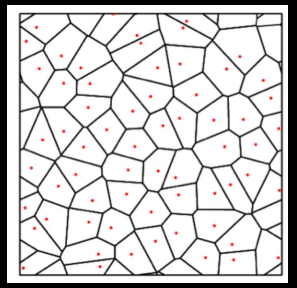
Accelerating radiative transfer (RT) in AREPO using GPUs

Oliver Zier (CfA)
(together with Rahul Kannan, Aaron Smith and Mark Vogelsberger)

oliver.zier@cfa.harvard.edu

The moving mesh method in AREPO

Combining finite volume method with moving mesh



Voronoi mesh

Springel 2010

Moment-based RT in AREPO (Kannan et al 2019)

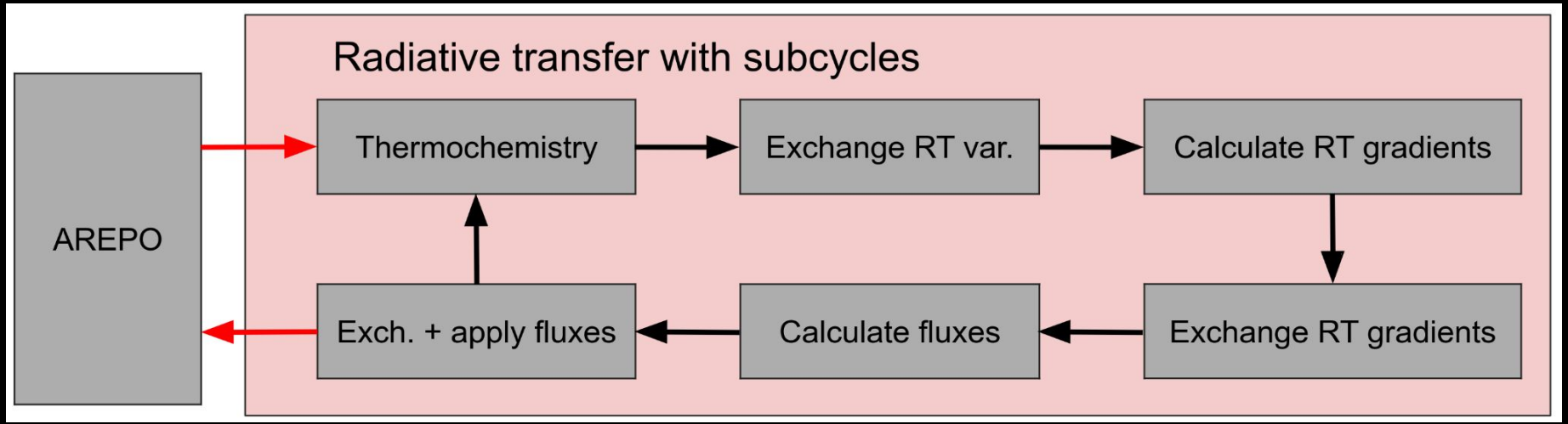
$$\frac{\partial E_r}{\partial t} + \nabla \cdot \boldsymbol{F}_r = S - \kappa_{\mathrm{E}}\, \rho\, \tilde{c}\, E_r \,,$$

M1 closure => standard FV

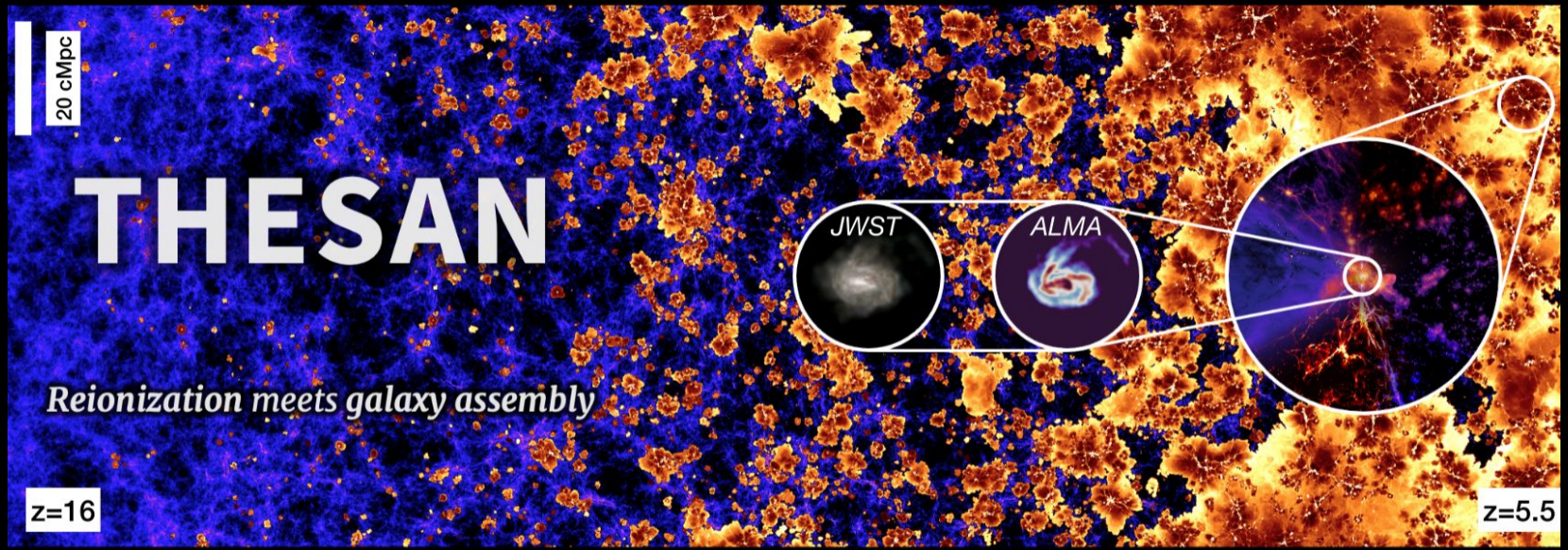
$$\frac{\partial \boldsymbol{F}_r}{\partial t} + \tilde{c}^2 \nabla \cdot \mathbb{P}_r = -\kappa_{\mathrm{F}}\, \rho\, \tilde{c}\, \boldsymbol{F}_r \,,$$

time step: $\Delta t_{\mathrm{RT}} \leq \eta \frac{\Delta x}{\tilde{c} + |\boldsymbol{v}_c|}$,

=> subcycling



Application: Thesan simulation

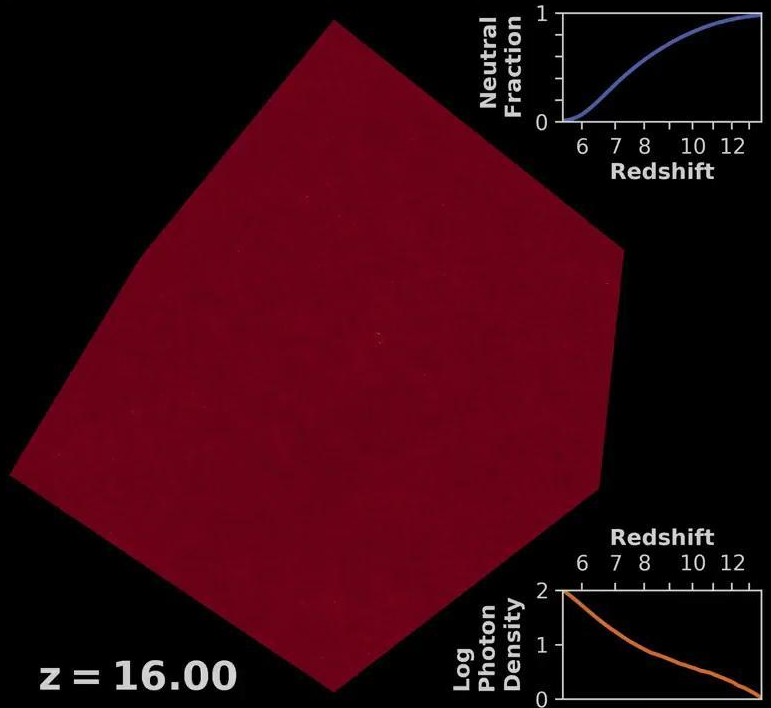


Large volume — high resolution — coupled reionization and galaxy formation simulations

Kannan+22a,b, Smith+22, Garaldi+22

The THESAN project

size: $(95.5\text{cMpc})^3$
end: z = 5.5
cells: $(2100)^3$
Illustris TNG + AREPO-RT

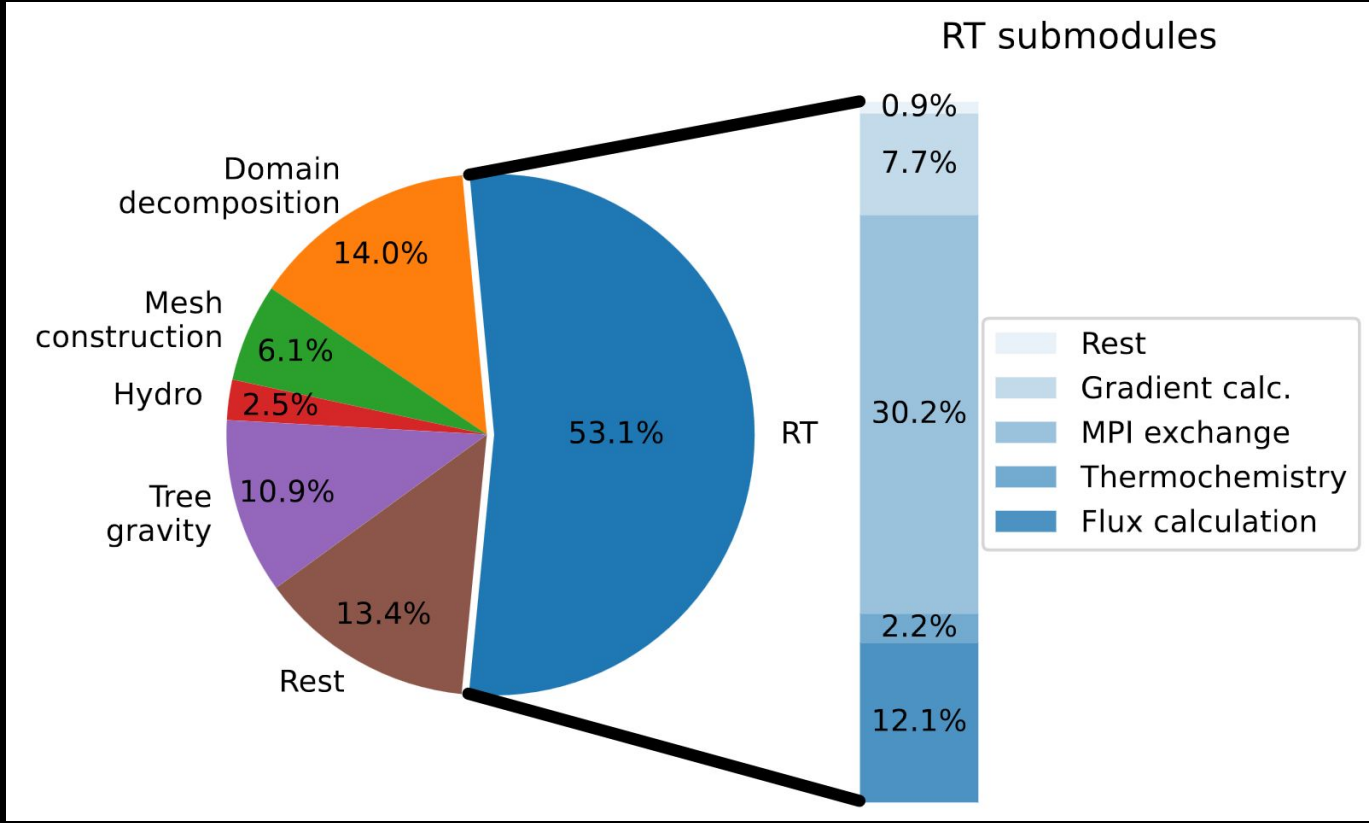


THESAN

Run time analysis - THESAN-1

- SuperMUC-NG
- 57,600 cores
- 28,000,000 CPU hours
- 3 frequency bins
- 64 subcycles

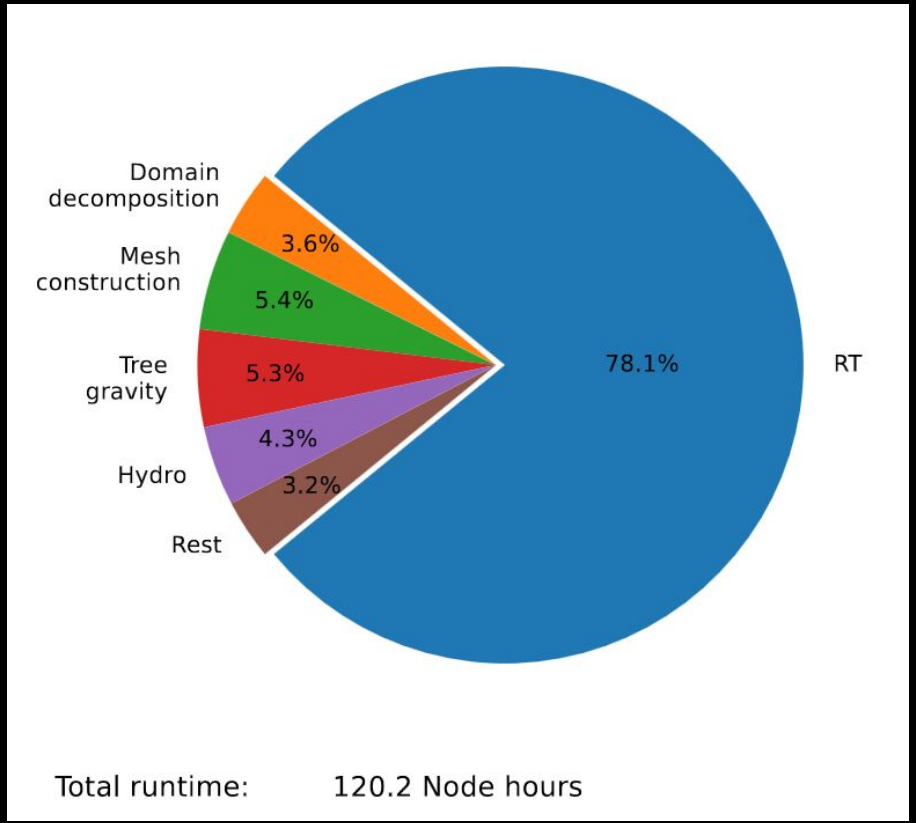
RT dominates costs!



Smaller, single node simulation

- 1 Node (72 cores)
- THESAN model
- 256^3 cells

RT fully dominates costs



Adapting AREPO-RT for Exascale Computing: GPU Acceleration and Efficient Communication

Oliver Zier,[1]★ Rahul Kannan,[2] Aaron Smith,[3] Mark Vogelsberger[1] and Erkin Verbeek[4]
[1]*Department of Physics, Kavli Institute for Astrophysics and Space Research, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[2]*Department of Physics and Astronomy, York University, 4700 Keele Street, Toronto, ON M3J 1P3, Canada*
[3]*Department of Physics, The University of Texas at Dallas, Richardson, TX 75080, USA*
[4]*Center for Computational Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

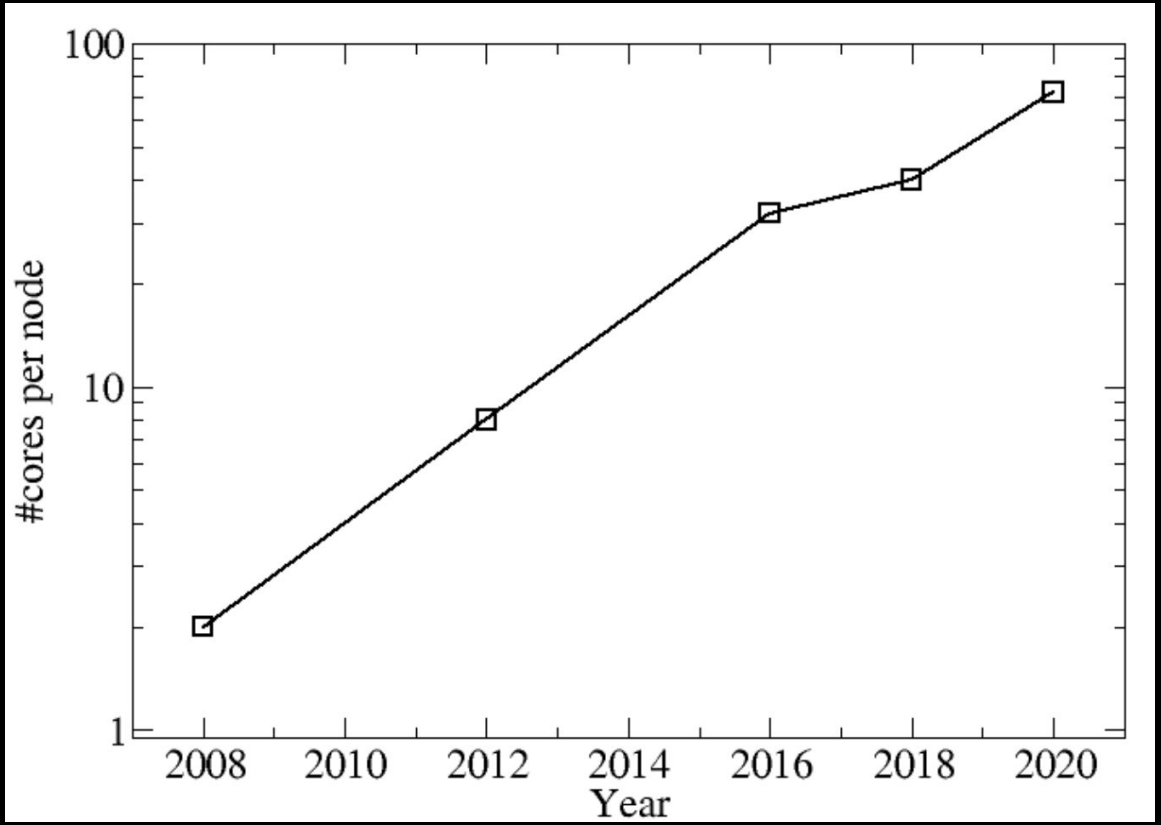
Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

Radiative transfer (RT) is a crucial ingredient for self-consistent modelling of numerous astrophysical phenomena across cosmic history. However, on-the-fly integration into radiation-hydrodynamics (RHD) simulations is computationally demanding, particularly due to the stringent time-stepping conditions and increased dimensionality inherent in multi-frequency collisionless Boltzmann physics. The emergence of exascale supercomputers, equipped with extensive CPU cores and GPU accelerators, offers new opportunities for enhancing RHD simulations. We present a novel optimization of AREPO-RT explicitly tailored for such high-performance computing environments. We implement a novel node-to-node communication strategy that utilizes shared memory to substitute intra-node communication with direct memory access. Furthermore, combining multiple inter-node messages into a single message substantially enhances network bandwidth utilization and performance for large-scale simulations on modern supercomputers. The single-message node-to-node approach also improves performance on smaller-scale machines with less optimized networks. Furthermore, by transitioning all RT-related calculations to GPUs, we achieve a significant computational speedup of around 15 for standard benchmarks compared to the original CPU implementation. As a case study, we perform cosmological RHD simulations of the Epoch of Reionization, employing a similar setup as the THESAN project. In this context, RT becomes sub-dominant such that even without modifying the core AREPO codebase, there is an overall threefold improvement in efficiency. The advancements presented here have broad implications, potentially transforming the complexity and scalability of future simulations for a wide variety of astrophysical studies. Our work serves as a blueprint for porting similar simulation codes based on unstructured resolution elements to GPU-centric architectures.

Key words: radiative transfer – methods: numerical – cosmology: reionization

Accelerated communication: Shared memory



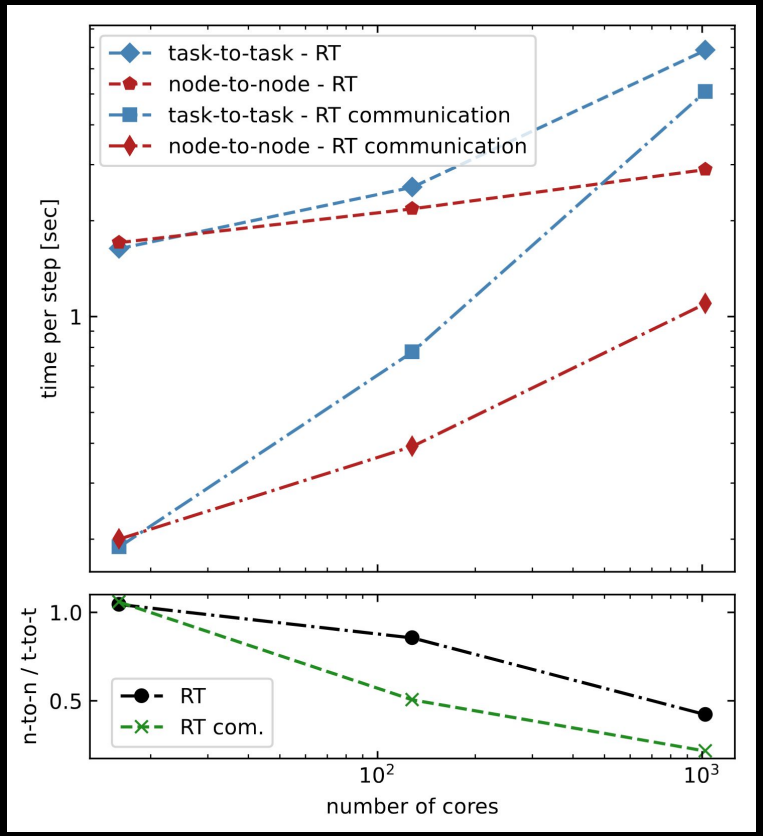
Max Planck Clusters,
future > 100

Cores share main memory
=> exploit in communication

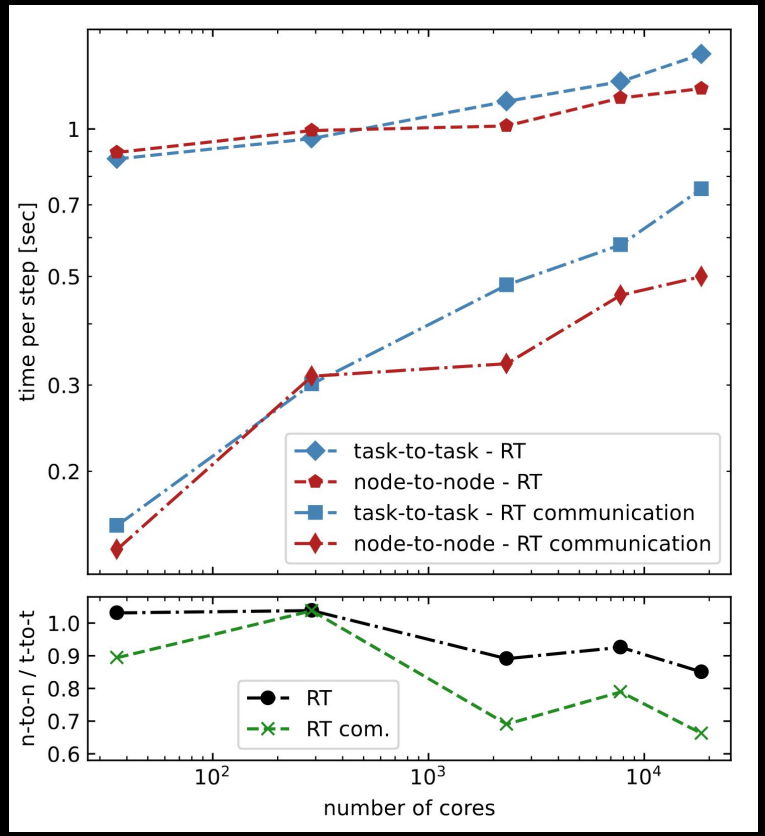
1. use direct memory access within node
2. bundle messages between nodes

But: synchronization dominates

Improved scalability (HII region test, weak scaling)

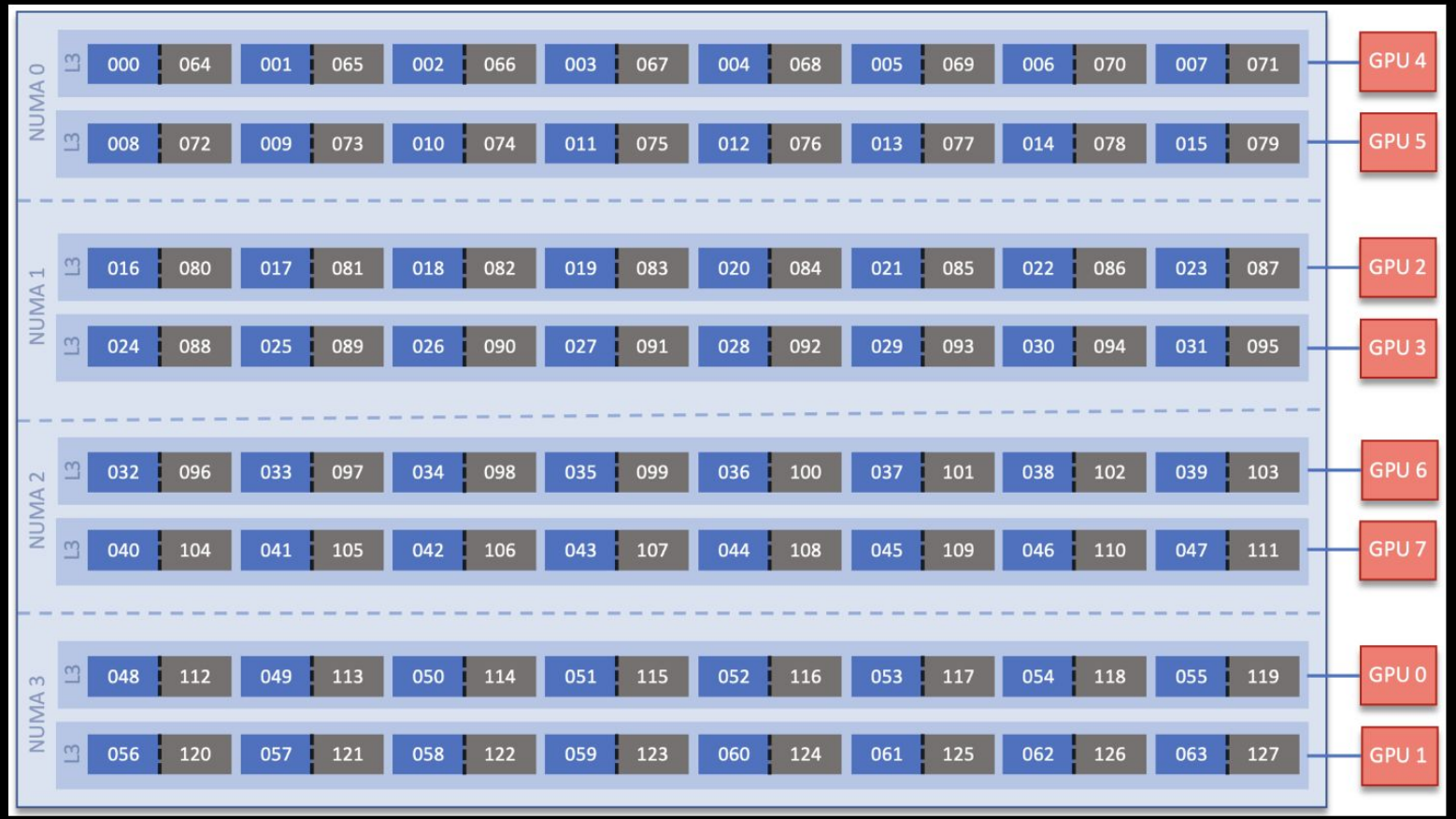


Engaging (MIT)



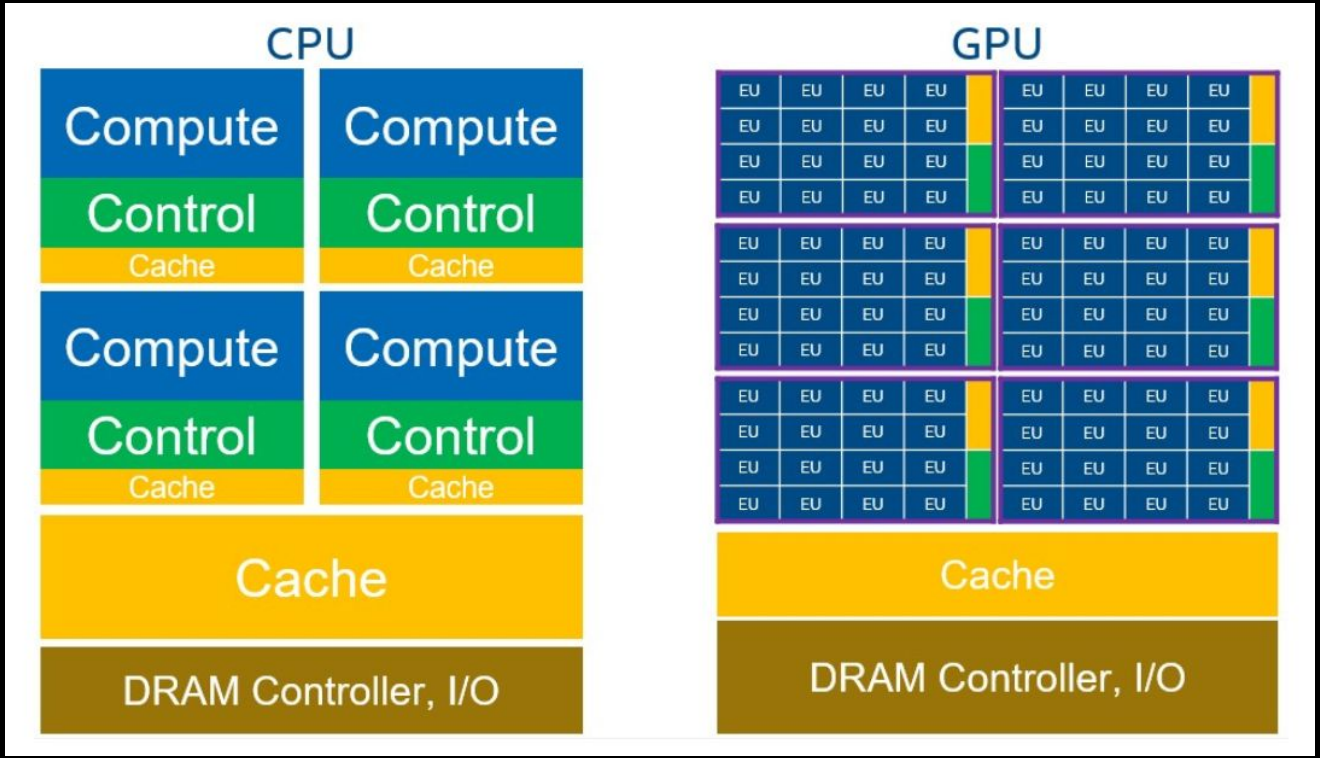
Raven MPCDF

A modern supercomputer node: Frontier



> 99% FLOPS from GPUs

GPU accelerators

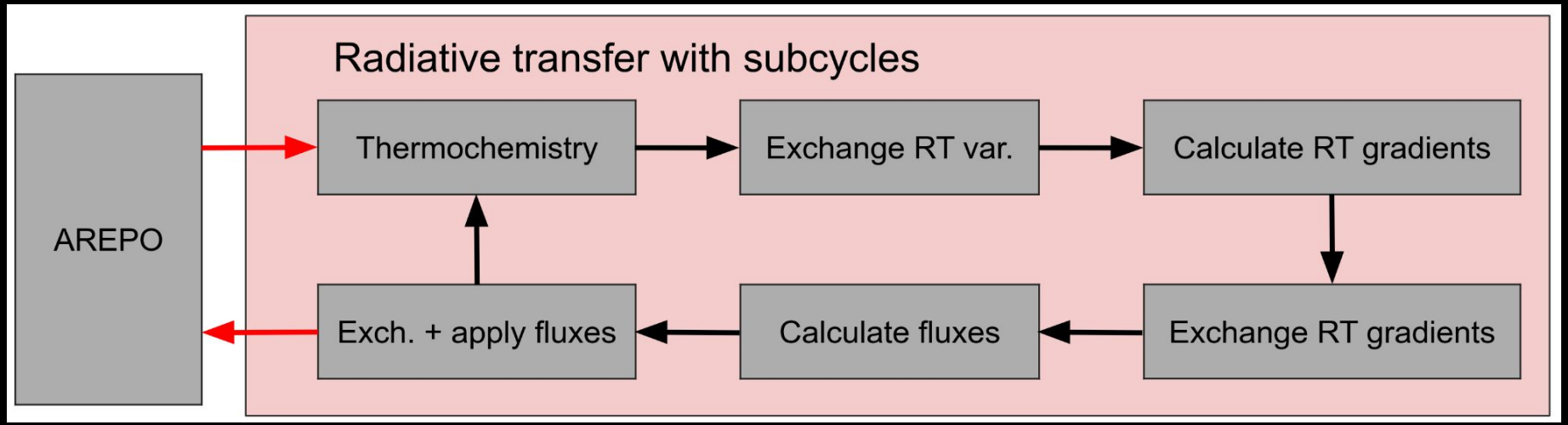


> 10^5 threads

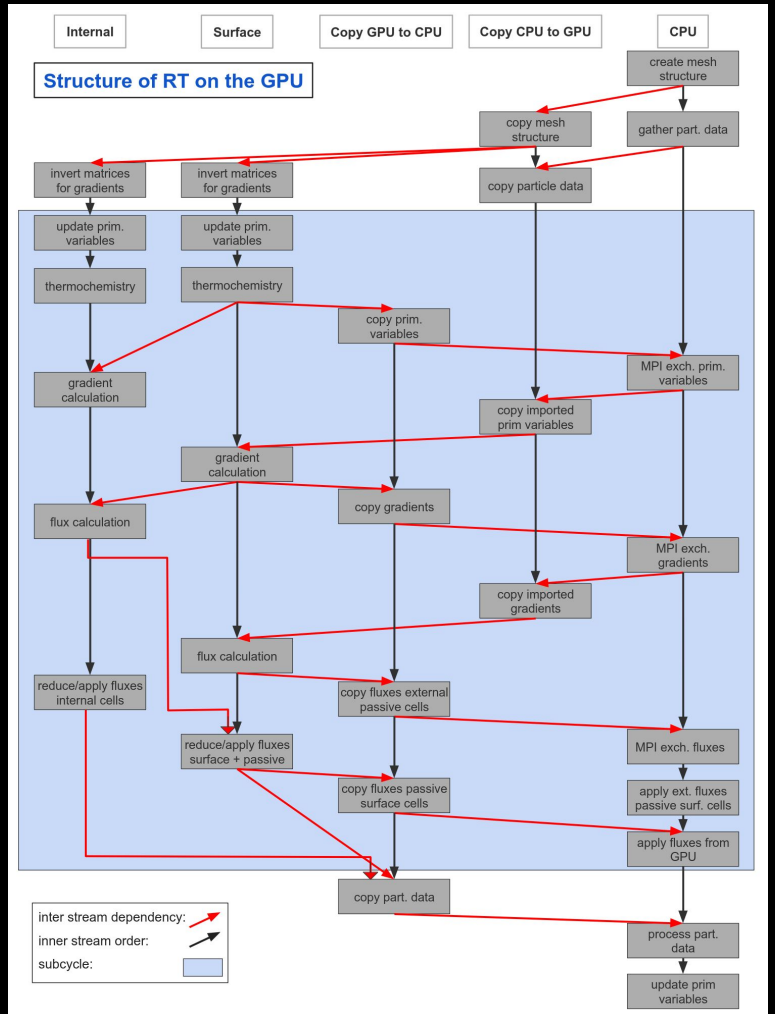
GPU requires special programming (e.g. CUDA)

Porting radiative transfer to GPU

- rewrite to independent loops over interfaces + cells
- optimize memory layout (structs of arrays)
- algorithmic changes (not all libraries exist on GPU)
- explicitly copy between CPU and GPU
- use own abstraction layer to support CUDA / HIP / CPU



Structure of RT on GPU



Memory Layout on the GPU

Particle based data

Cell properties (e.g. primitive variables/gradients)

active internal cells	active surface cells	passive internal cells	passive surface cells

Arrays used to connect cells and interfaces

#interfaces per cells	…	N[i-1]	N[i]	N[i+1]	…
offset per cell	…	O[i-1]	O[i]	O[i+1]	…

indices of all interfaces for each cell:

…	I[O[i-1] + N[i-1]-1]	I[O[i]]	…	I[O[i] + N[i]-1]	I[O[i+1]]	…

Interface based data

Interface properties (e.g. size or normal vector)

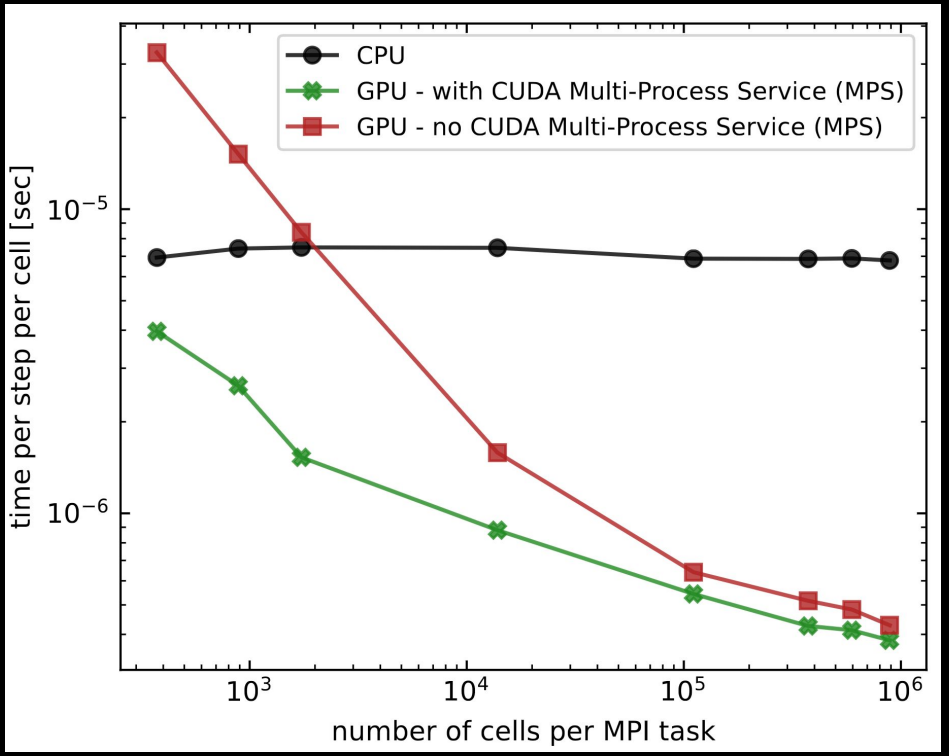
internal interfaces	surface interfaces with active neighbour	surface interfaces with passive neighbour

Arrays used to connect with cells

indices of left neighbour cells

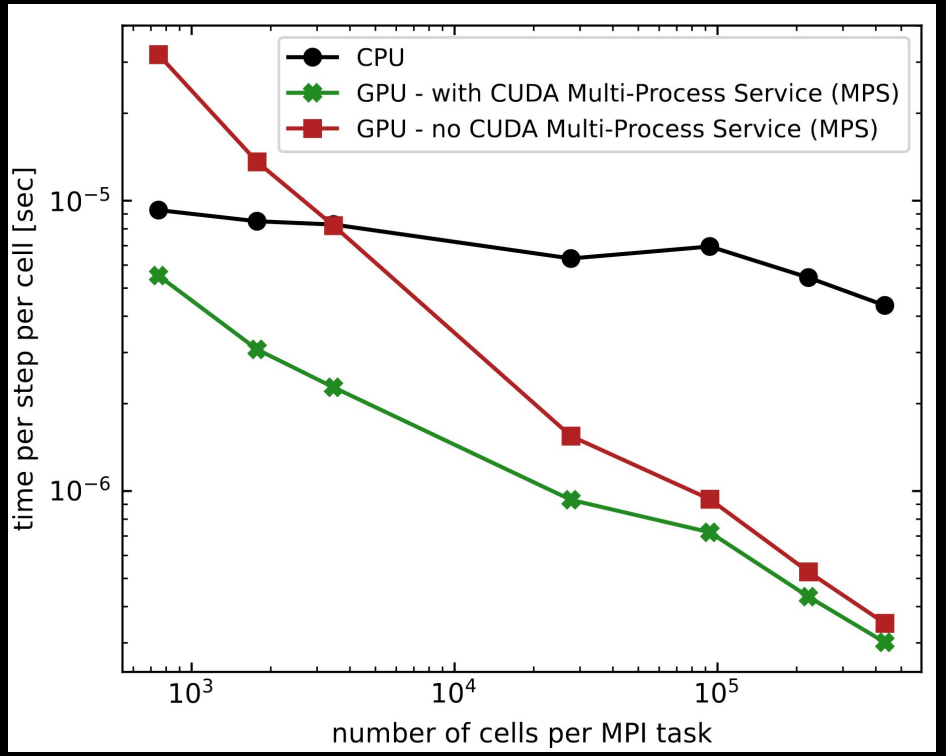
indices of right neighbour cells

Speedup Raven, 72 cores, 4 A100 GPUs, 1 Node



pure Photon transport

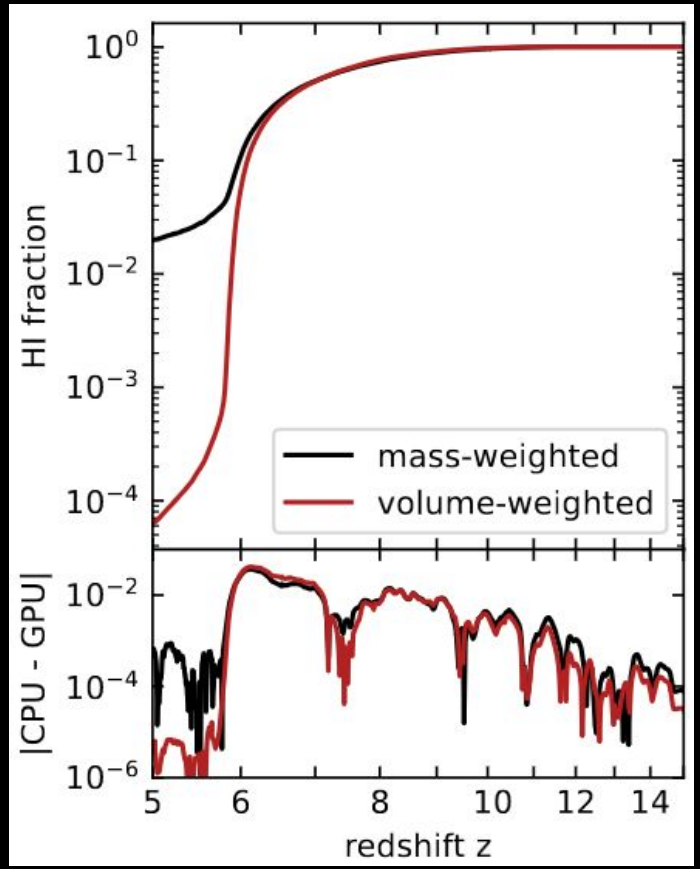
15x Speedup



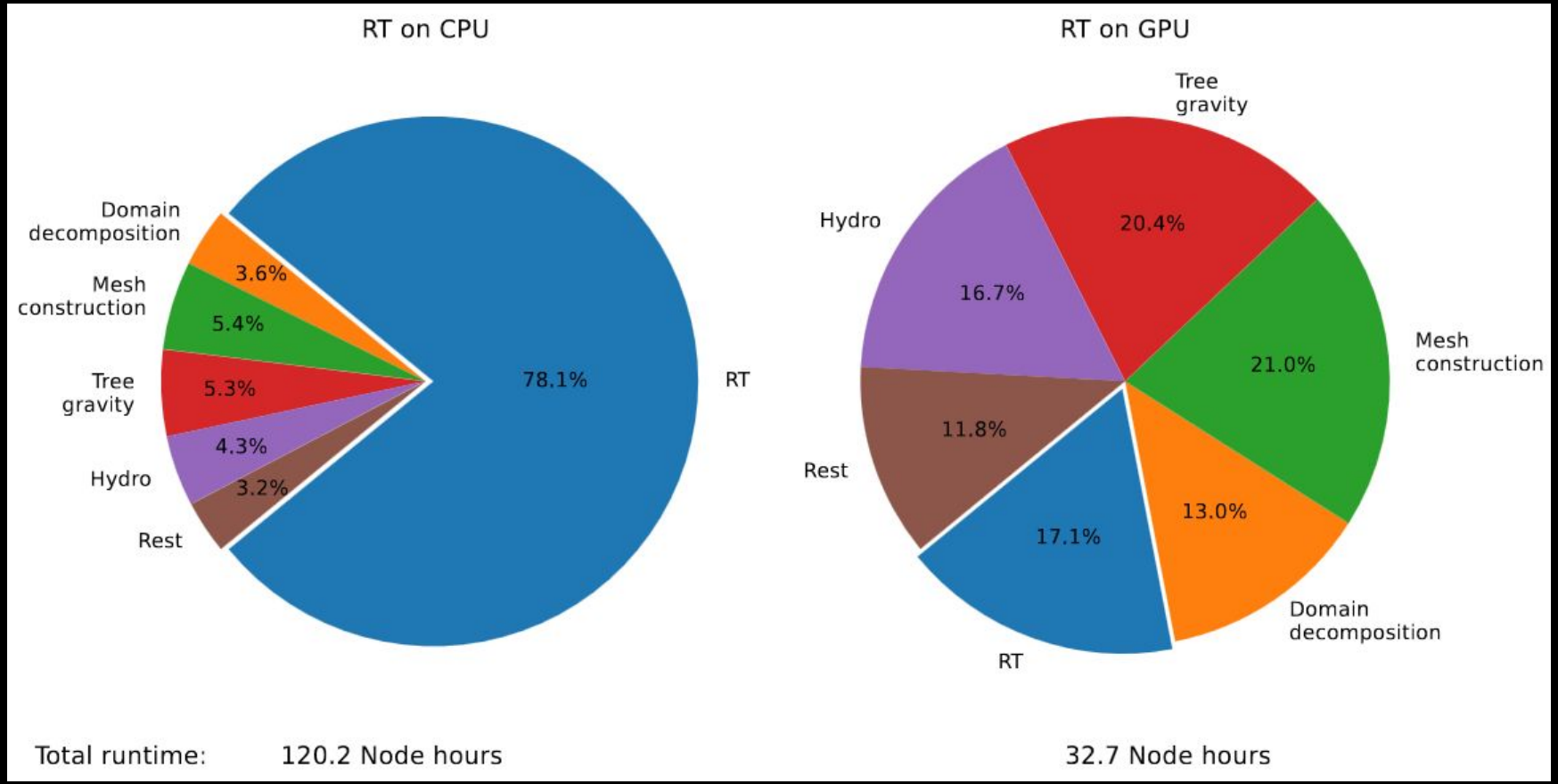
Expanding HII region

Cosmological simulation (THESAN model)

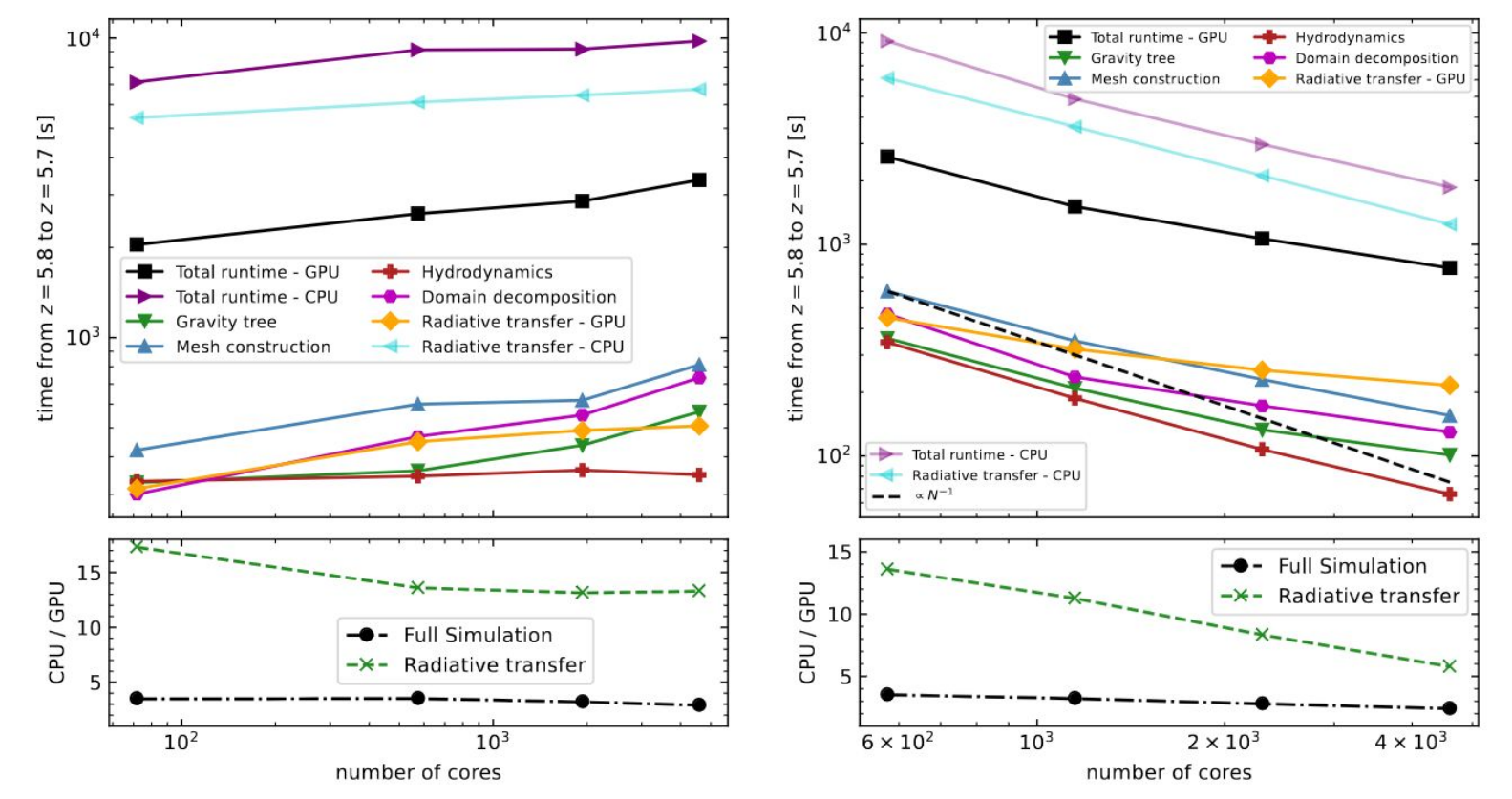
Test	$L_{\rm box}$ [cMpc]	$N_{\rm particles}$	$z_{\rm start}$	$z_{\rm end}$	$N_{\rm nodes}$
Evolution	23.3	2×256^3	49	5	1
Weak Scaling	23.3	2×256^3	5.8	5.7	1
	46.6	2×512^3	5.8	5.7	8
	69.9	2×768^3	5.8	5.7	27
	93.1	2×1024^3	5.8	5.7	64
Strong Scaling	46.6	2×512^3	5.8	5.7	4
	46.6	2×512^3	5.8	5.7	16
	46.6	2×512^3	5.8	5.7	32
	46.6	2×512^3	5.8	5.7	64



Cosmological simulation (THESAN model, 1 node)



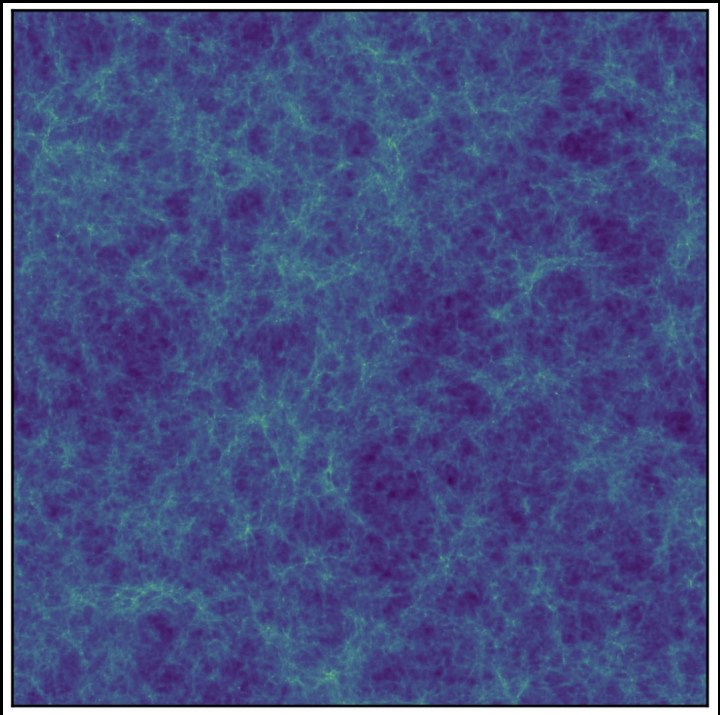
Scaling on Raven



weak scaling

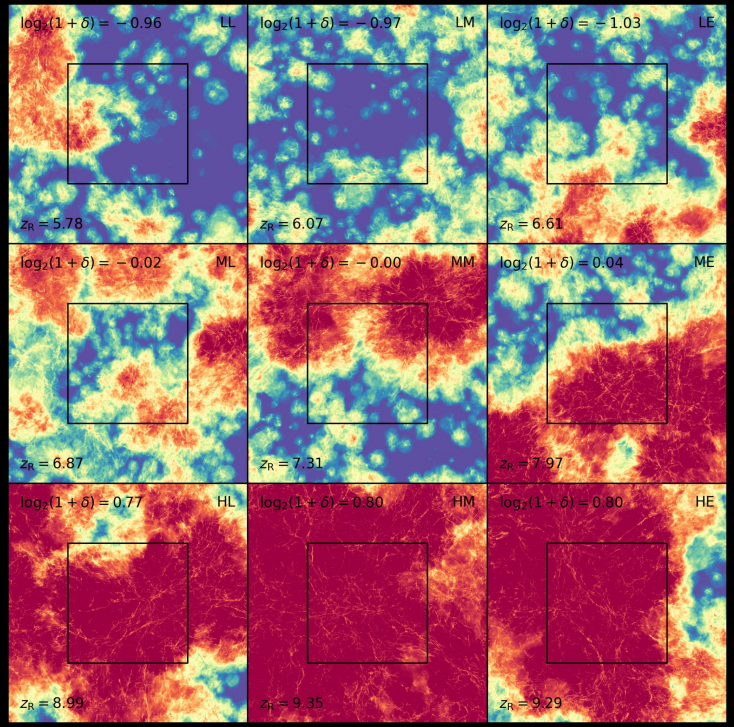
strong scaling

Applications



- larger boxes
- more frequency bins

=> reduced computing costs



- high resolution zoom in
- small boxes

=> reduced run time

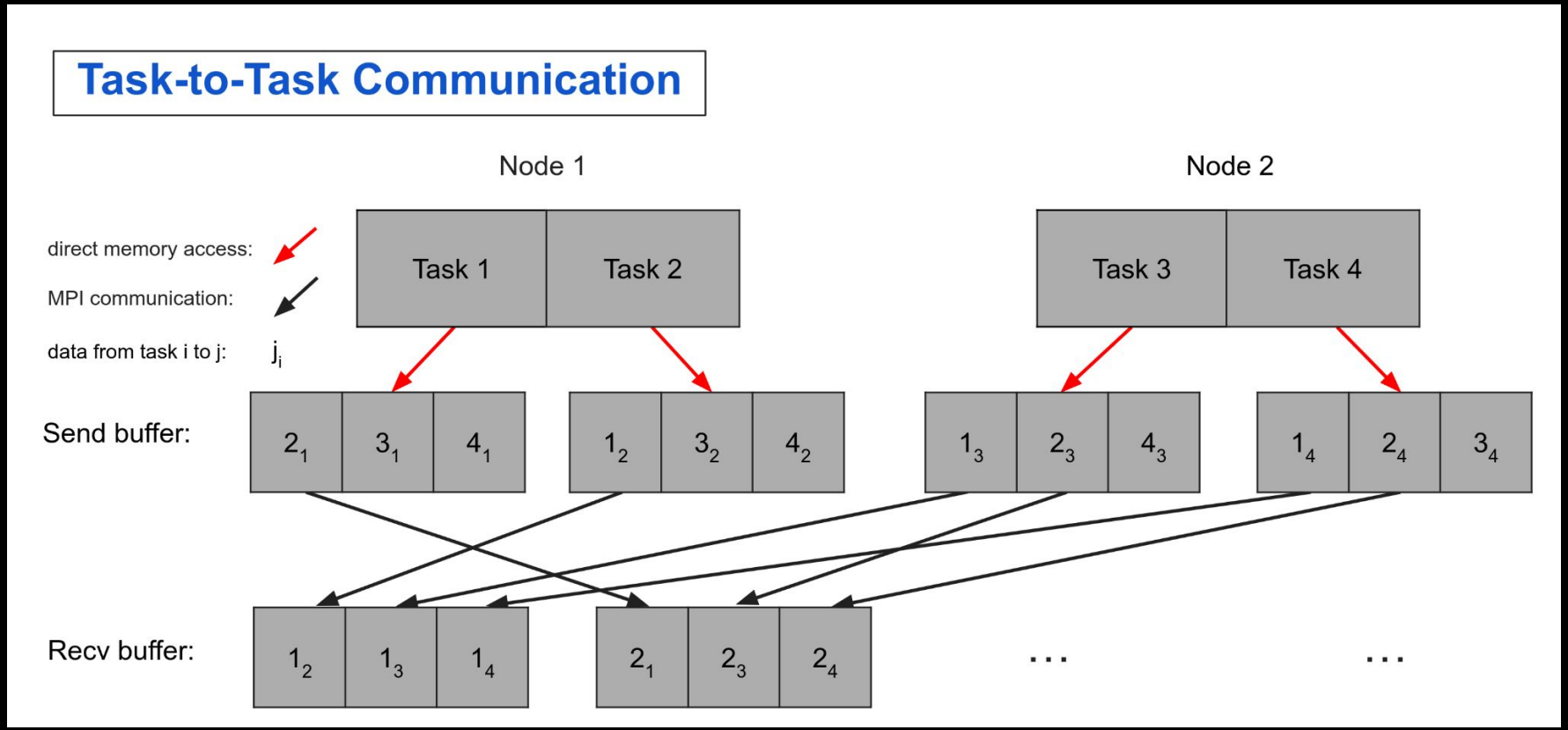
Summary

- RT very expensive (small time steps or subcycling)
- exploit shared memory in communication
- GPU acceleration (15x compared to old CPU version)
- abstraction layer for GPU in AREPO: finite volume + source terms

=>

- significantly reduced wall time for high-res simulation with small boxes
- reduced computing time in large simulations
- easier porting of expensive physics to GPU

Old communication scheme



New communication scheme

