MAESTRO and CASTRO - Petascale AMR Codes for Astrophysical Applications

Andy Nonaka
Lawrence Berkeley National Laboratory

SIAM CSE11 – March 3, 2001 – Reno, NV

Outline

- Overview of MAESTRO and CASTRO
- Numerical Implementation
- Parallel Implementation
- Scaling and Scientific Results

- Collaborators
  - LBNL Center for Computational Sciences and Engineering
    - Ann Almgren, Mike Lijewski, Candace Gilet, John Bell
  - Stony Brook University Dept. of Physics and Astronomy
    - Mike Zingale, Chris Malone

MAESTRO and CASTRO

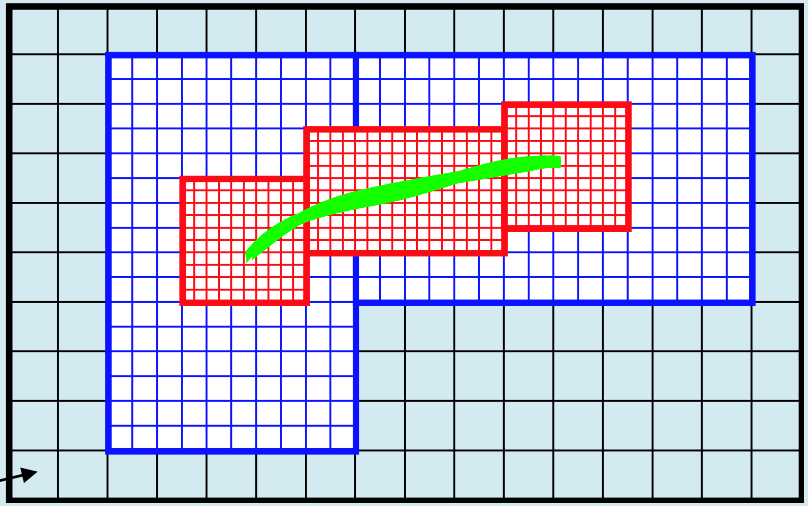
- Finite volume, block structured adaptive mesh refinement (AMR) codes for astrophysical phenomena
  - System of advection-reaction-diffusion equations
  - Modular equation of state
  - Modular reaction network
  - Massively parallel
    - MAESTRO scales to 100K cores
    - CASTRO scales to 200K+ cores
- CASTRO is a general compressible code
- MAESTRO is a low Mach number code, designed for efficient simulation of low speed flows (relative to the sound speed).

MAESTRO and CASTRO

- Both codes are mature and are being actively used in scientific investigations.
- MAESTRO
  - Type Ia supernovae; pre-ignition (Zingale, Stony Brook)
  - Type I X-Ray Bursts (Malone, Stony Brook)
  - Classical novae (Krueger, Calder, Stony Brook)
  - Convection in massive stars (Gilet, LBNL)
- CASTRO
  - Type Ia supernovae; post-ignition (Ma, Woosley, UCSC)
  - Core-collapse supernovae (Nordhaus, Burrows, Princeton)
  - Pair instability supernovae (Chen, Heger, U. of Minnesota; Joggerst, LANL)

General Framework

- Finite Volume: solution in each Cartesian cell represents the average over the cell

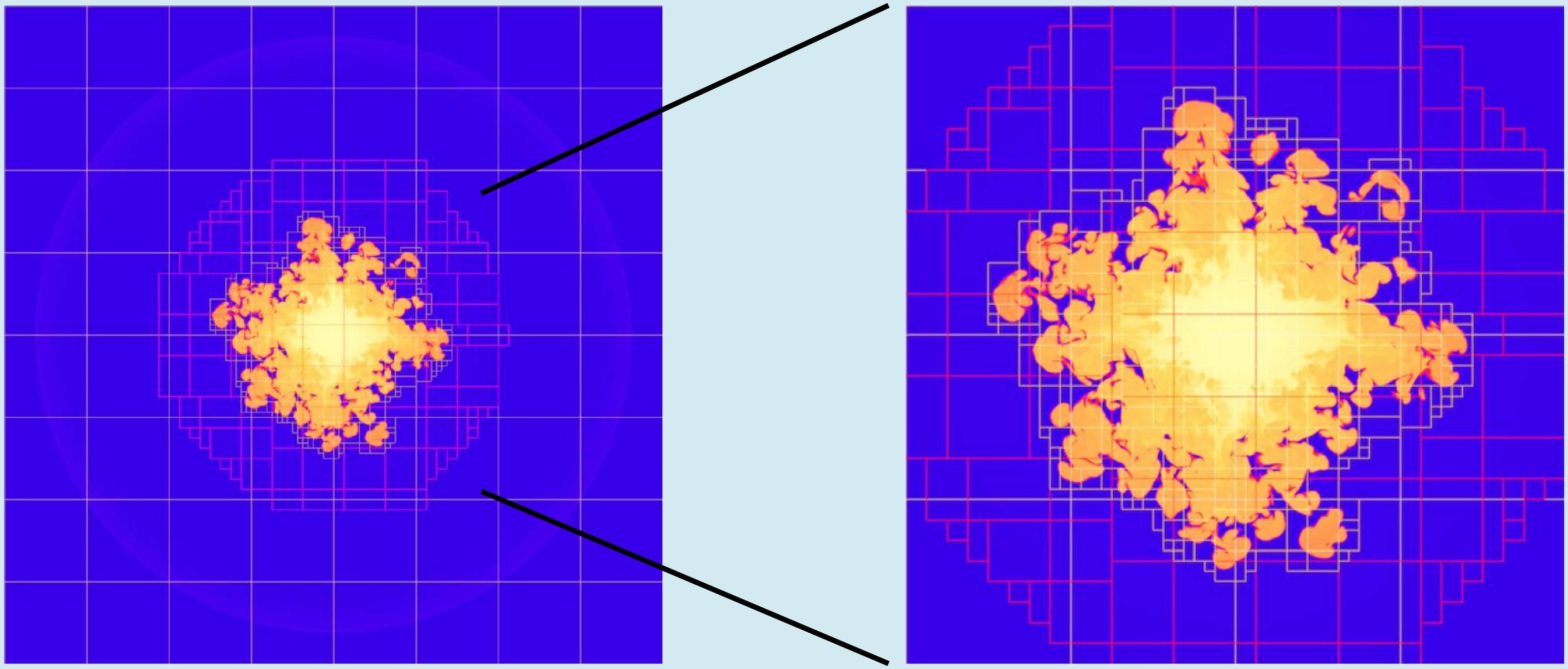


$$\rho, \mathbf{u}, p, T, \text{ etc.}$$

- AMR: Block-structured approach with logically rectangular grids

Example: Type Ia Supernova Explosion

- Temperature plot and zoom-in of a 3D Type Ia supernova explosion in CASTRO (Ma, Woosley, UCSC)



- Our software can handle many levels of AMR and scales well for problems with datasets that are 100GB – 1TB.

Software Overview

- BoxLib software framework provides set of tools for finite-volume block-structured AMR applications
  - C++ / Fortran90
  - Subcycling in time (CASTRO only)
- Parallel I/O
  - Peak I/O at NERSC (approx 13 GB/s) is comparable with NERSC benchmarks
- Hierarchical programming model
  - Hybrid MPI/OpenMP approach.

MAESTRO Overview

- MAESTRO is a low Mach number hydrodynamics solver for astrophysical flows.
  - Mach number: $M = U/c$
  - In the low Mach number regime, $M = O(10^{-2})$, acoustic waves carry little energy so we derive an equation set which excludes them.
  - Time steps are constrained by advective CFL, not acoustic CFL.

$$\Delta t_{\text{compressible}} < \max\left\{\frac{\Delta x}{|u|+c}\right\} \qquad \Delta t_{\text{lowMach}} < \max\left\{\frac{\Delta x}{|u|}\right\}$$

$$\Delta t_{\text{lowMach}} \sim \frac{1}{M}\Delta t_{\text{compressible}}$$

  - Low Mach time steps are a factor of $1/M$ larger!

Low Mach Number Equation Set

- Derived from fully compressible equation set

$$\frac{\partial(\rho X_k)}{\partial t} = -\nabla\cdot(\rho X_k \mathbf{u}) + \rho\dot{\omega}_k$$ conservation of mass

$$\frac{\partial(\rho \mathbf{u})}{\partial t} = -\nabla\cdot(\rho \mathbf{u}\mathbf{u}) - \nabla\pi + \rho\mathbf{g}$$ conservation of momentum

$$\frac{\partial(\rho h)}{\partial t} = -\nabla\cdot(\rho h \mathbf{u}) + \nabla\cdot\kappa\nabla T + \rho H$$ conservation of energy

ρ	density	h	specific enthalpy
$\mathbf{u}$	velocity	κ	thermal diffusion coefficient
X_k	mass fraction of species “k”	H	reaction heating
$\dot{\omega}_k$	reaction rate of species “k”	π	deviation from ambient pressure
$\mathbf{g}$	gravity	T	temperature

Low Mach Number Equation Set

- Our system is closed with an equation of state, which keeps system in thermodynamic equilibrium.
  - Differentiate equation of state along particle paths to represent as a divergence constraint:

$$\nabla \cdot (\beta_0 \mathbf{u}) = \beta_0 S$$

$\beta_0 \rightarrow$ captures expansion/contraction of fluid due to changes in altitude

$S \rightarrow$ captures local compressibility effects due to reactions and thermal diffusion

- Numerical enforcement of divergence constraint analogous to solution methodology for incompressible flow
  - Pressure-projection method involving a variable-coefficient Poisson solve

Numerical Methodology

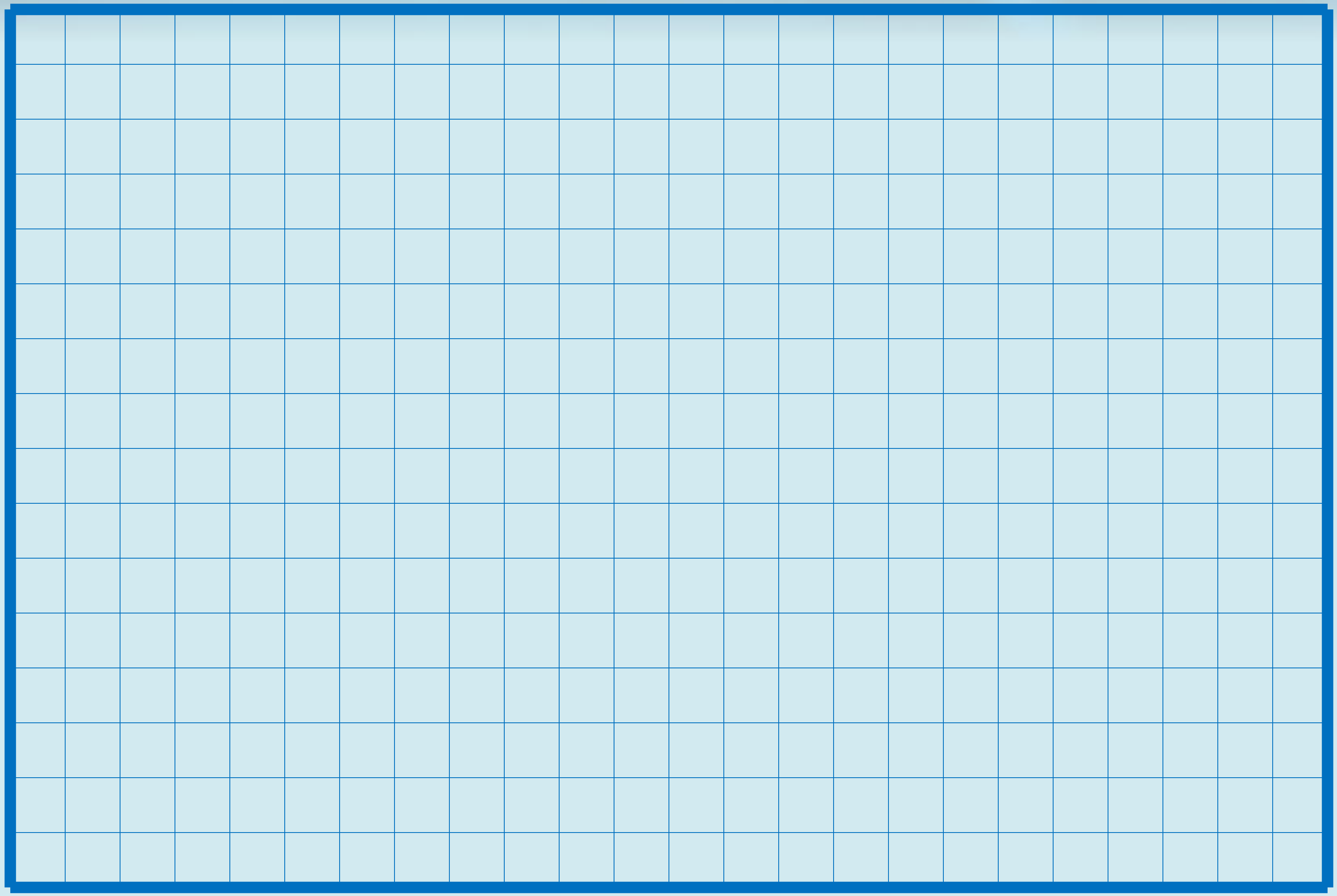
- Strang splitting couples advection/reaction/diffusion
  - Advection using Godunov approach
  - Reactions using stiff ODE solver
  - Diffusion semi-implicit (multigrid)
  - Divergence-constraint requires elliptic solve (multigrid)

$$
\begin{aligned}
\frac{\partial(\rho X_k)}{\partial t} &= -\underline{\nabla \cdot (\rho X_k \mathbf{u})} + \underline{\rho \dot{\omega}_k} \\
\frac{\partial(\rho \mathbf{u})}{\partial t} &= -\underline{\nabla \cdot (\rho \mathbf{u}\mathbf{u})} - \underline{\nabla \pi} + \underline{\rho \mathbf{g}} \\
\frac{\partial(\rho h)}{\partial t} &= -\underline{\nabla \cdot (\rho h \mathbf{u})} + \underline{\nabla \cdot \kappa \nabla T} + \underline{\rho H}
\end{aligned}
$$

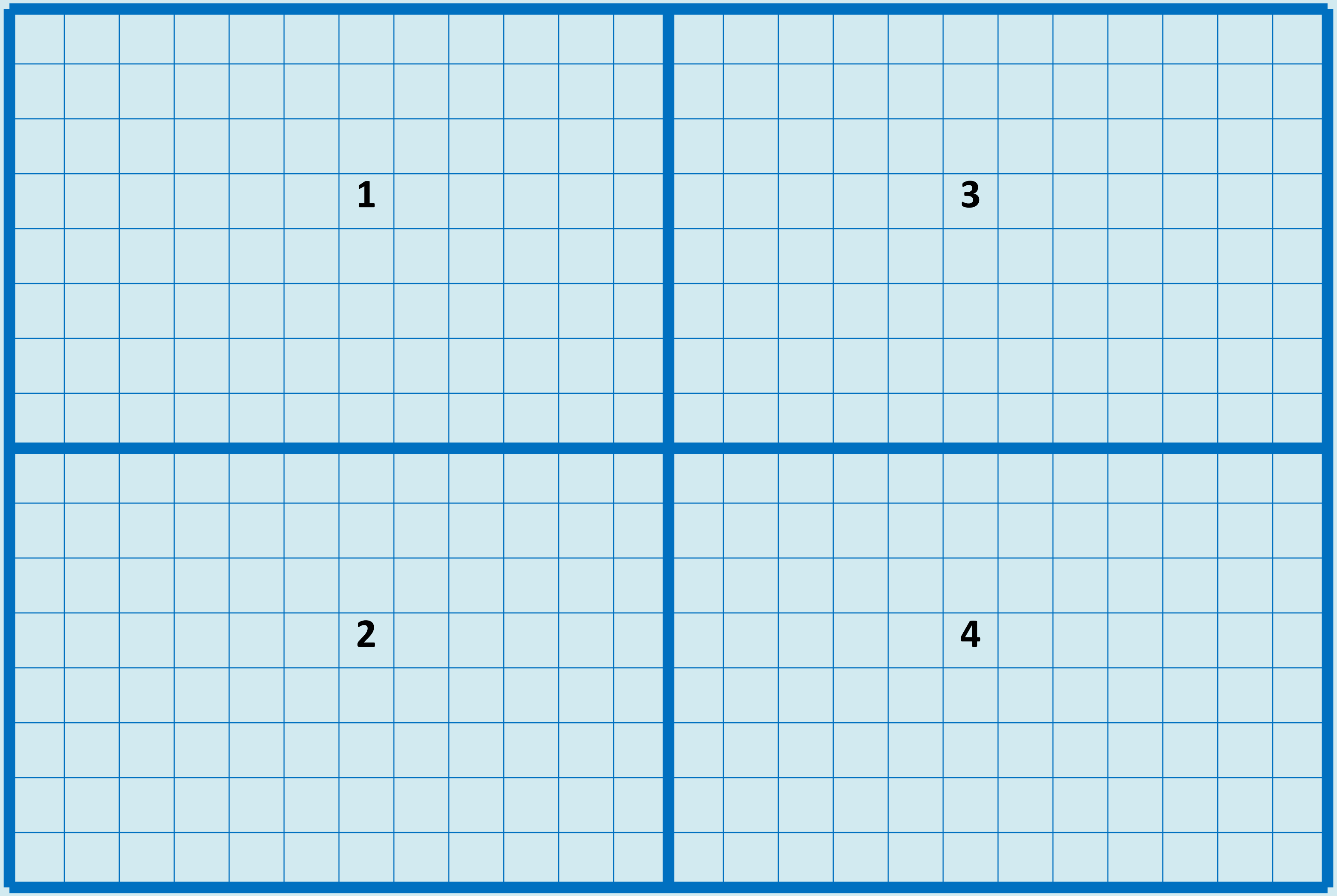
$$
\underline{\nabla \cdot (\beta_0 \mathbf{u}) = \beta_0 S}
$$

Implementation Details

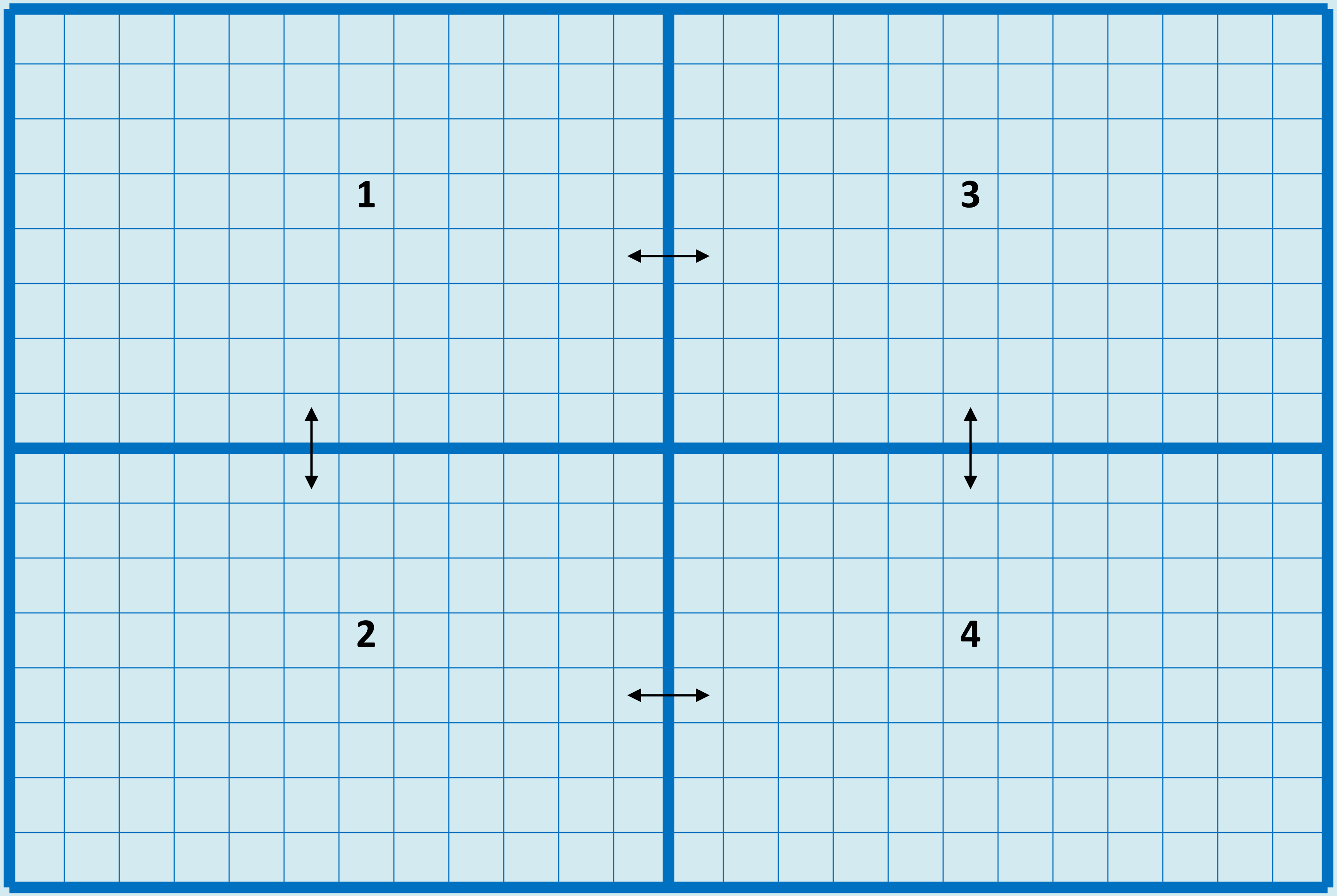
- Divide computational domain into cells.



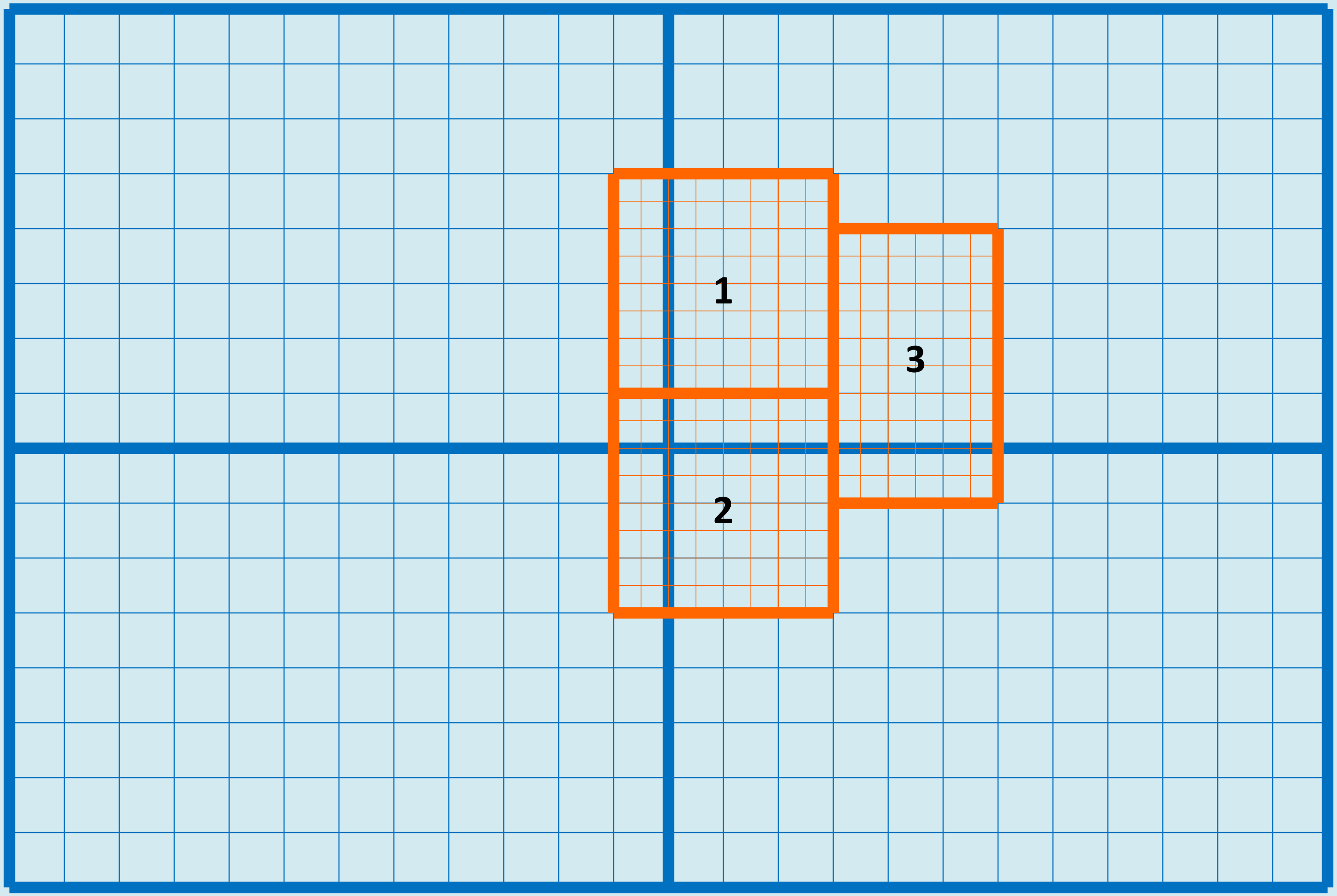
- Divide the domain into grids
- In a pure MPI instantiation, we assign each grid to a core.



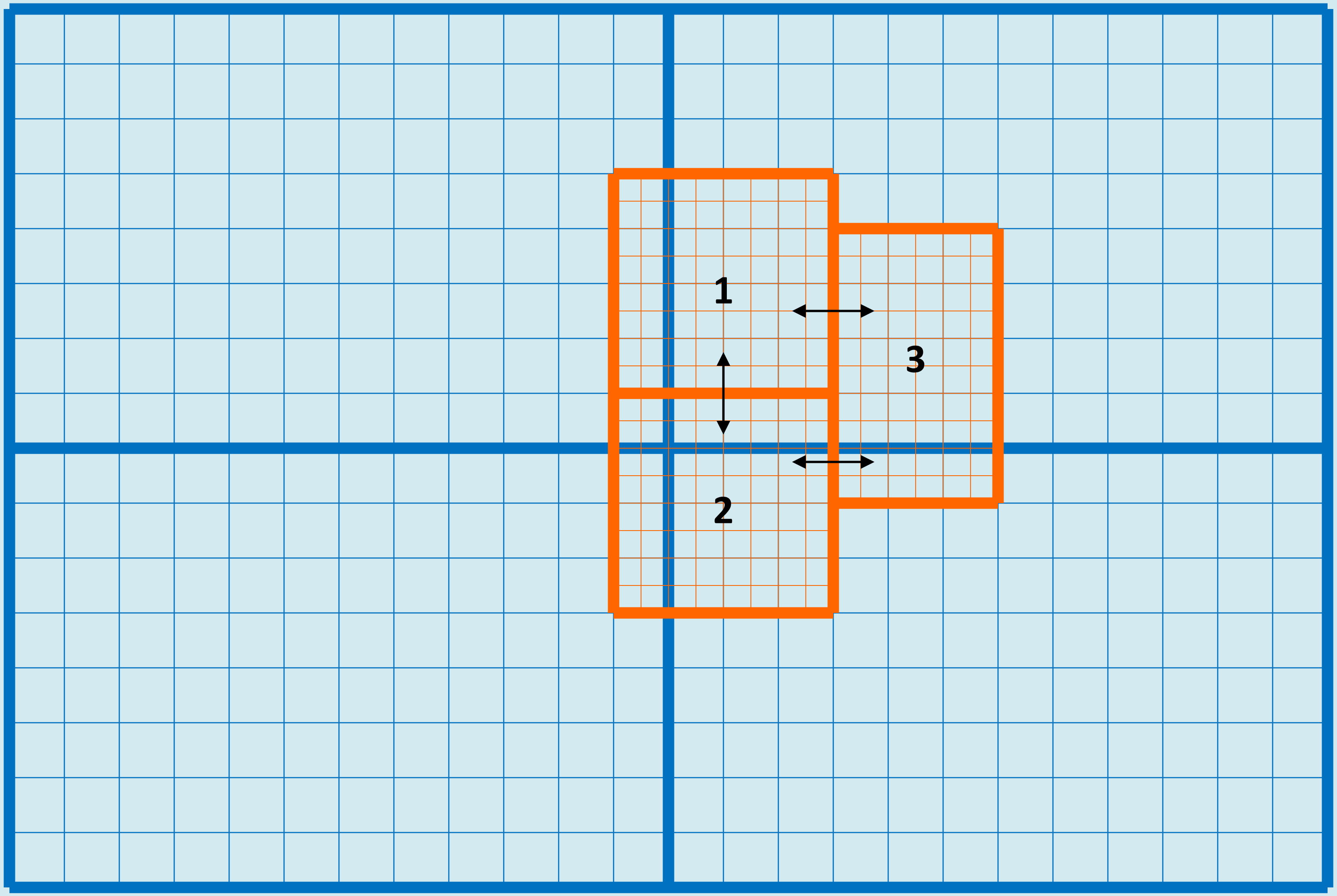
- Each core advances the solution on the grid(s) it owns
- Communication between grids to fill ghost cells



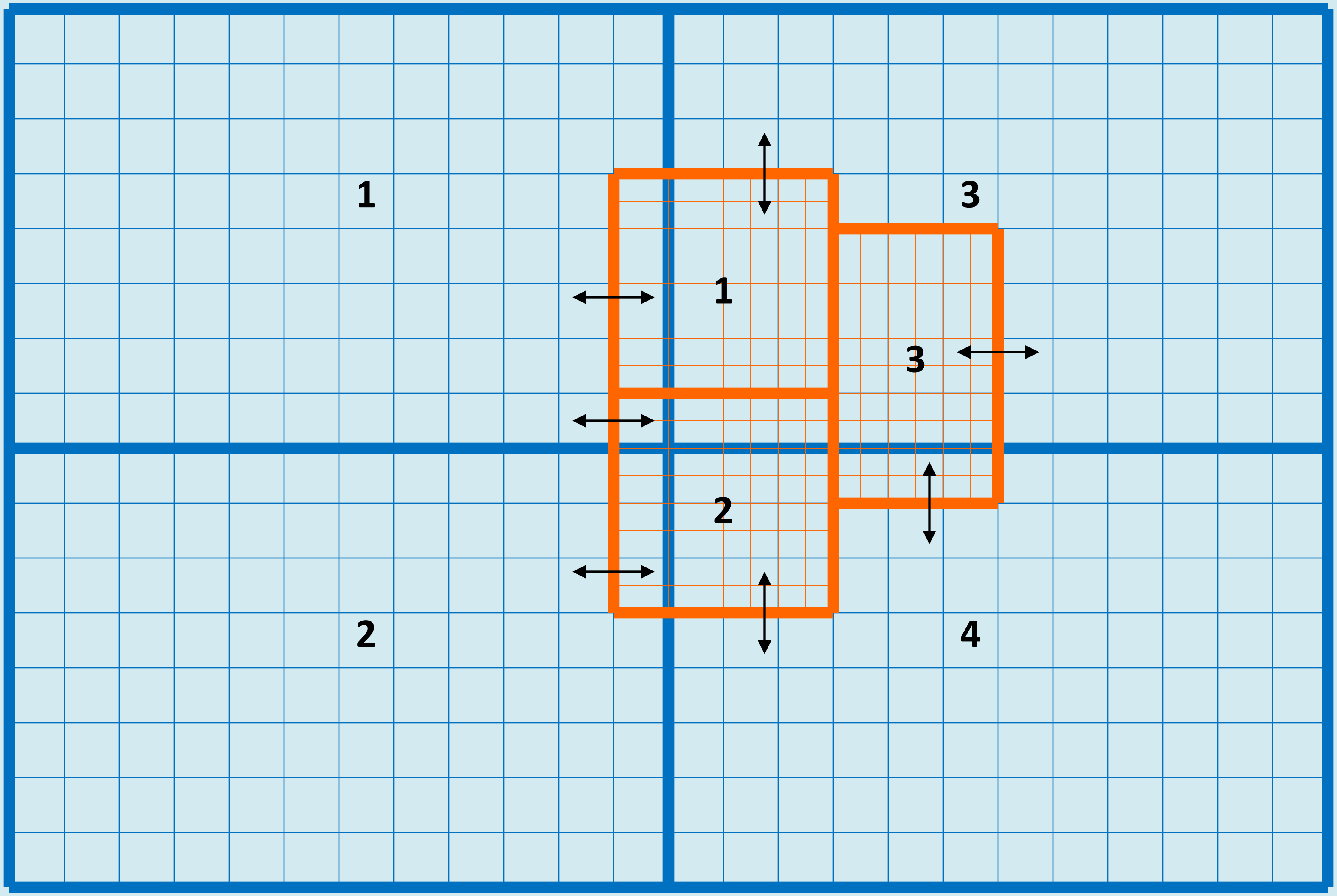
- We repeat this process for each level of refinement.
  - Integrate levels separately
  - Synchronize solution between levels



- Each core advances the solution on the grid(s) it owns
- Communication between grids to fill ghost cells



- Two-way communication between levels required to synchronize solution

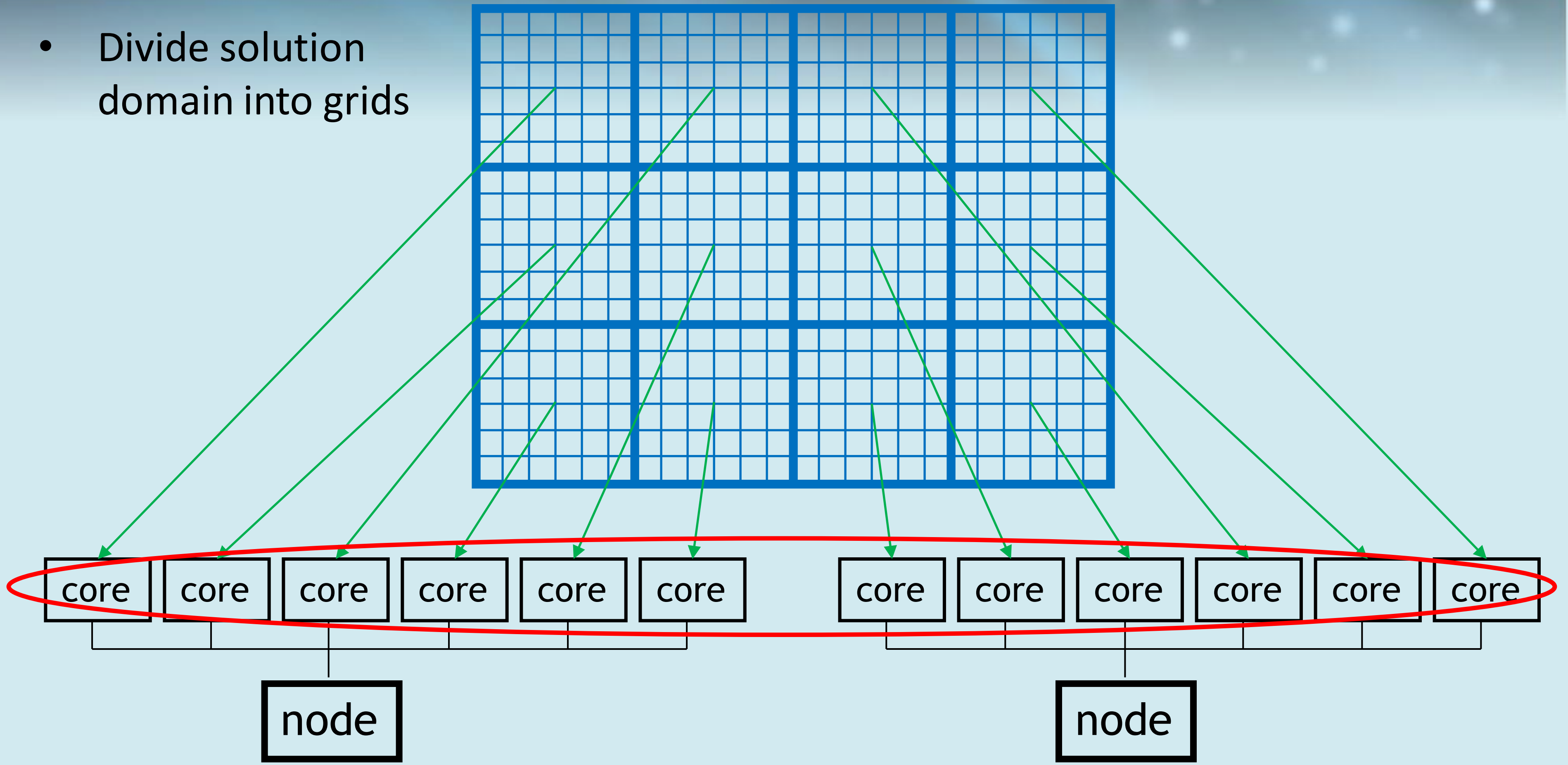


Communication Requirements

- Advection and reaction steps require minimal communication – transfer of ghost cell data.
- Multigrid requires intensive communication – often 1000+ calls to transfer ghost cell data per linear solve.
- Also, there is a non-trivial amount of time spent determining communication patterns between grids.
- Our approach is to reduce communication overhead by reducing the number of MPI processes.
  - Hybrid MPI/OpenMP approach.

Parallel Implementation – Pure MPI

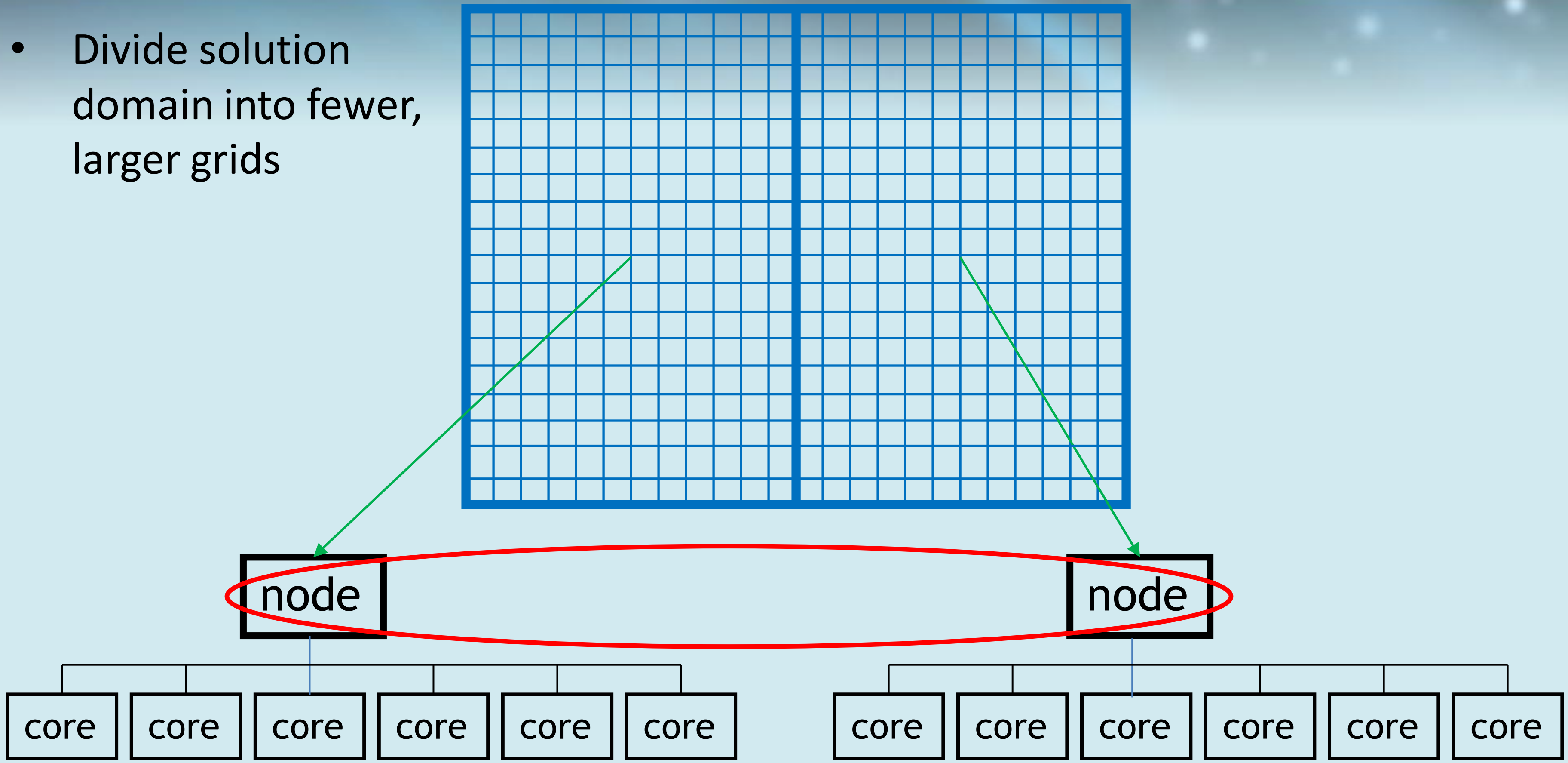
- Divide solution domain into grids



- Each grid is assigned to a core
- Cores communicate each other using MPI
  - In this example, we require 12 MPI processes.

Parallel Implementation – Hybrid MPI/OpenMP

- Divide solution domain into fewer, larger grids



- Each grid is assigned to a node
  - Spawn a thread on each core to work on the grids simultaneously
- Nodes communicate each other using MPI
  - In this example, we require 2 MPI processes.

Advantages of Hybrid MPI/OpenMP

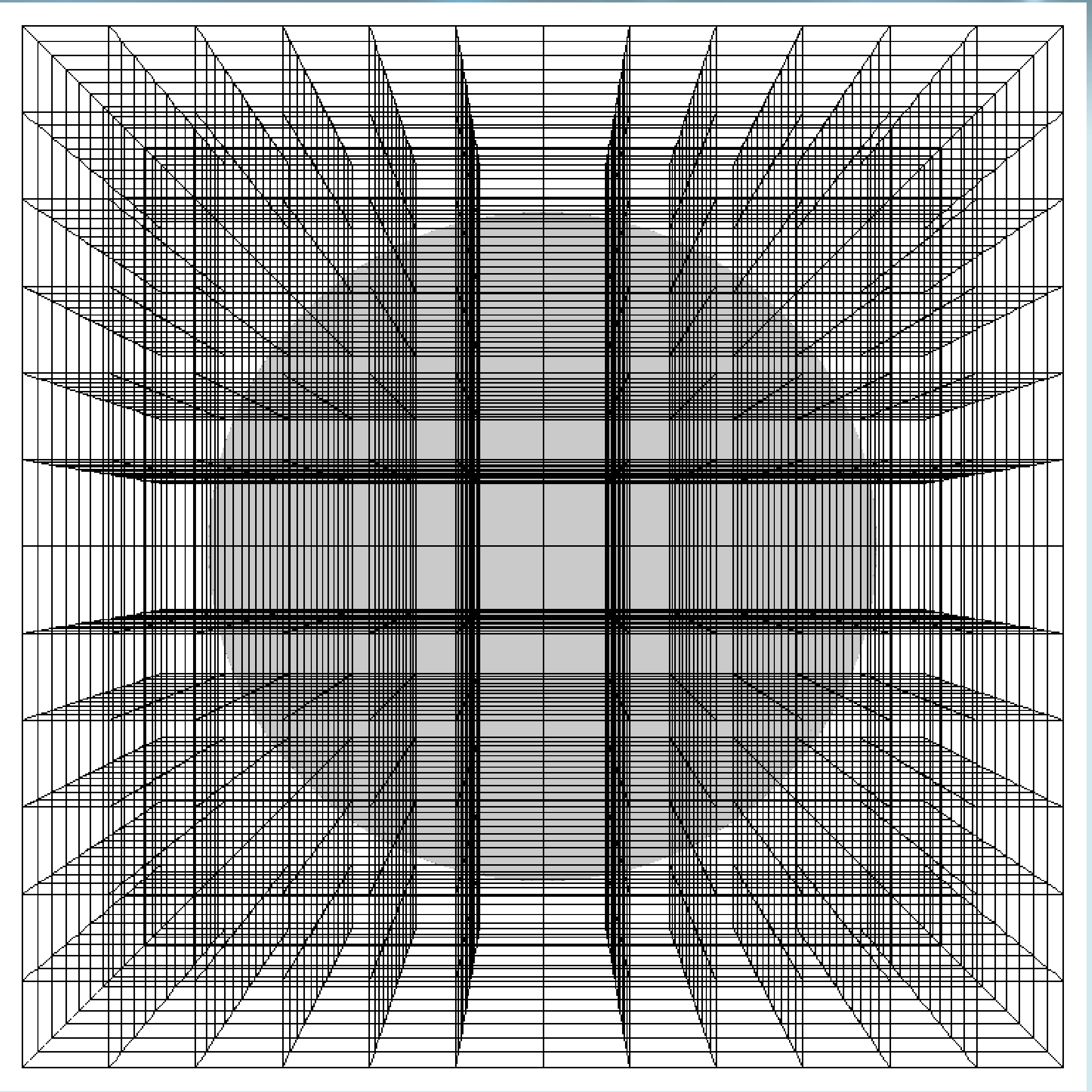
- Fewer MPI processes lead to reduced communication time
  - Especially important in communication-intensive multigrid
- Fewer grids leads to reduced memory overhead requirements
  - Metadata contains a mapping of grids and their associated MPI processes.

Other Techniques

- As we approach the coarser levels of multigrid, we transfer the problem to fewer, larger grids, thus reducing the number of MPI processes in the bottom solve.
- Hash sorting to efficiently compute communication patterns by more intelligently searching for neighboring grids.
- Cache commonly used communication patterns.

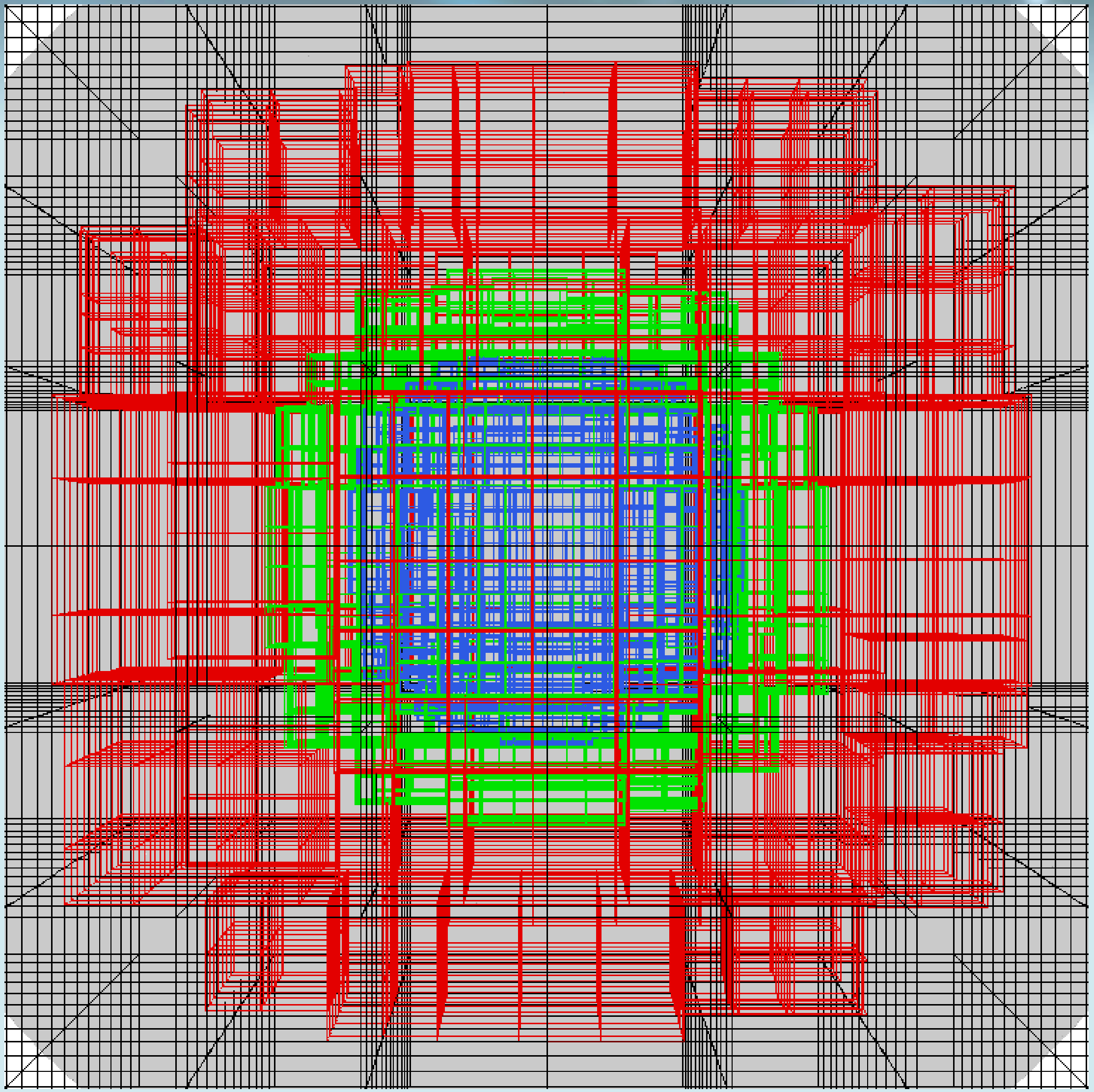
Type Ia Supernovae

- Full star dynamics
- 5000 km^3 domain
- 576^3 resolution
  - $1728 \cdot 48^3$ grids

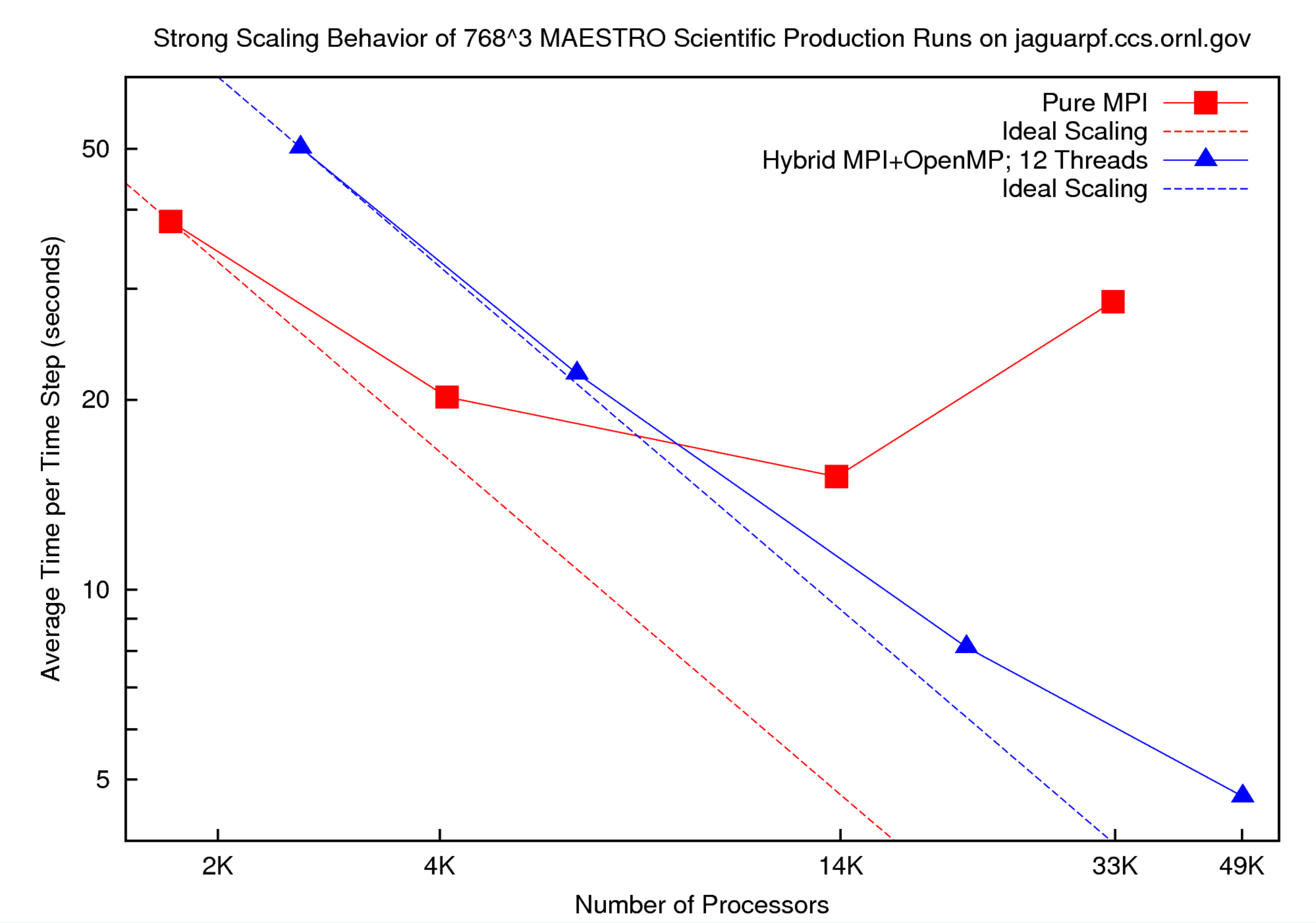


Type Ia Supernovae

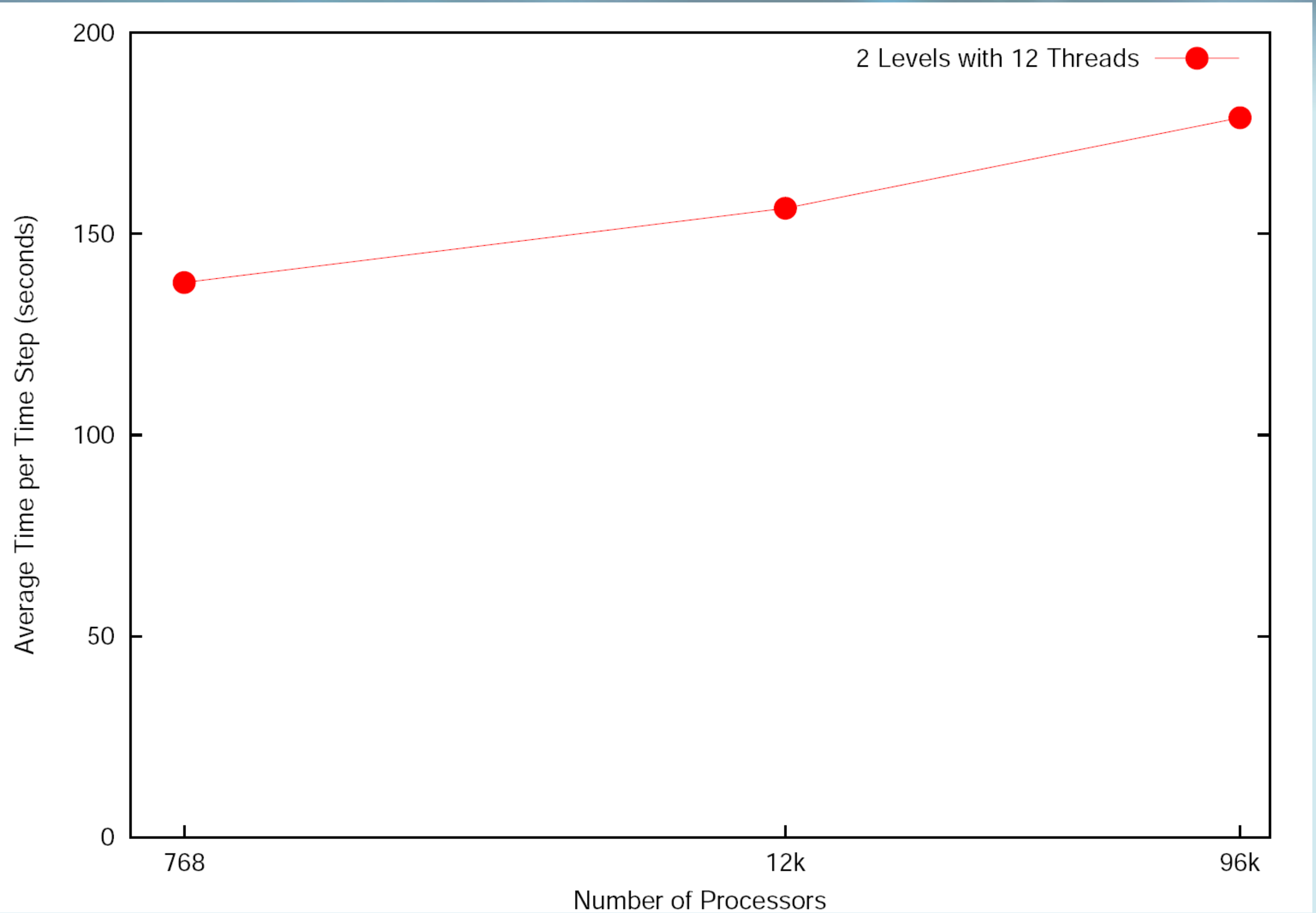
- Full star dynamics
- 5000 km^3 domain
- 576^3 resolution
  - 1728 · 48^3 grids
- 1152^3 resolution
  - 1831 grids
- 2304^3 resolution
  - 2449 grids
- 4608^3 resolution
  - 7072 grids



MAESTRO Strong Scaling

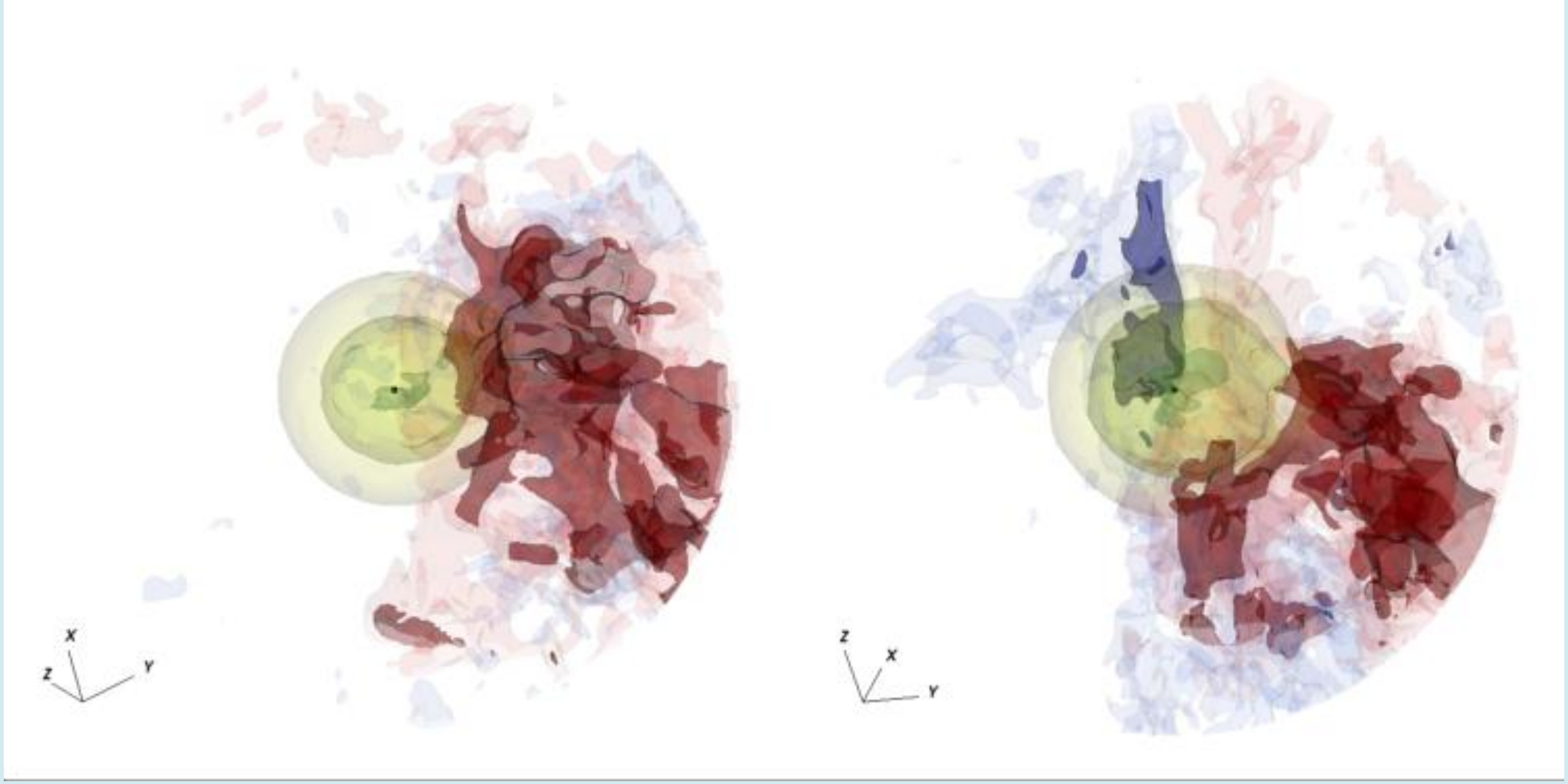


MAESTRO Weak Scaling



Type Ia Supernovae with MAESTRO

- We have performed simulations of convection in a white dwarf preceding a Type Ia supernova on the jaguarpf XT5 at OLCF
  - 10K cores, 7M CPU-hours per simulation, 1152^3 effective resolution with 2 AMR levels.
  - Will perform more studies at (up to) 4308^3 effective resolution with 4 AMR levels.



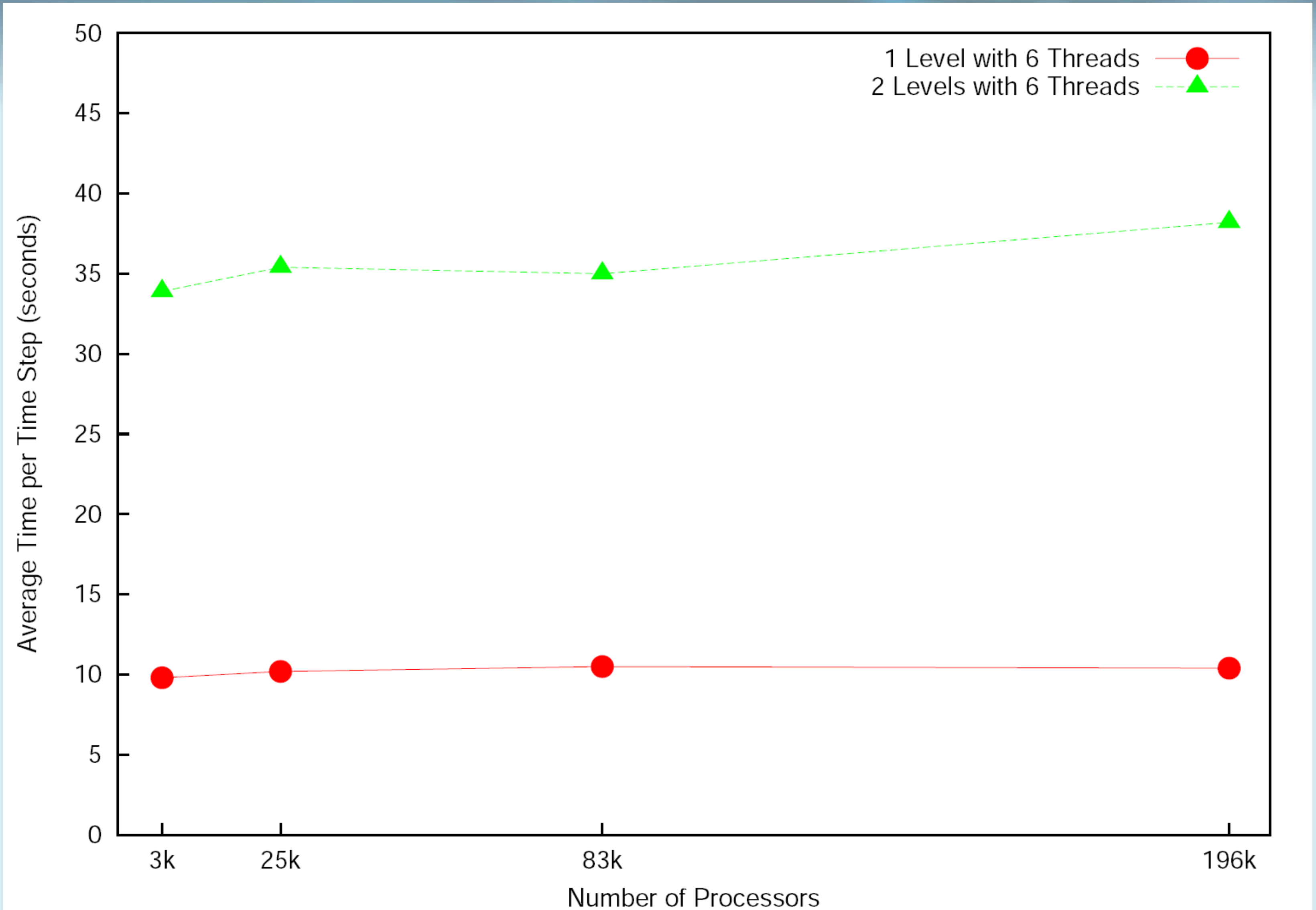
CASTRO Overview

- Standard compressible equations of motion

$$
\begin{aligned}
\frac{\partial(\rho X_k)}{\partial t} &= -\nabla\cdot(\rho X_k \mathbf{u}) + \rho\dot{\omega}_k \\
\frac{\partial(\rho \mathbf{u})}{\partial t} &= -\nabla\cdot(\rho \mathbf{u}\mathbf{u}) - \nabla p + \rho\mathbf{g} \\
\frac{\partial(\rho E)}{\partial t} &= -\nabla\cdot(\rho \mathbf{u} E + p\mathbf{u}) + \rho\mathbf{u}\cdot\mathbf{g} + \nabla\cdot k_{\mathrm{th}}\nabla T + \rho H
\end{aligned}
$$

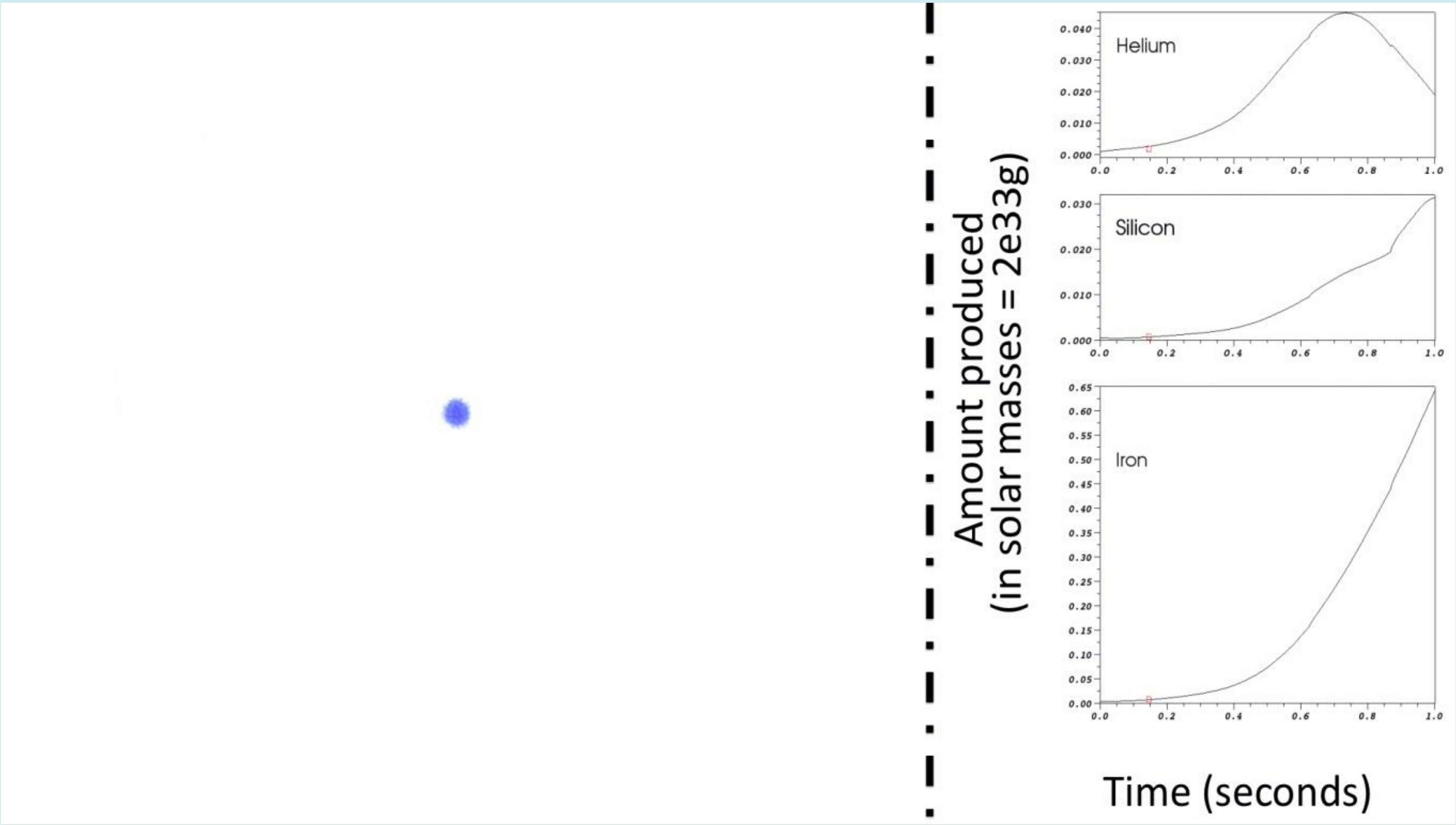
- Advection (Godunov method) and reactions (stiff ODE solver) require little communication.
- Semi-implicit thermal diffusion and self-gravity (Poisson equation) are optional.
  - Using a monopole gravity approximation and explicit thermal diffusion, CASTRO scales to 200K+ cores.

CASTRO Weak Scaling



Type Ia Supernovae with CASTRO

- CASTRO has been used to perform simulations of the explosion phase of a Type Ia Supernova on jaguarpf (Haitao Ma, UCSC)
  - 12K cores, 2.5M CPU-hours, 8192^3 effective resolution with 5 AMR levels.



Summary

- Low Mach number AMR code MAESTRO scales to 100K cores, performing science using O(10K) cores.
- Compressible AMR code CASTRO scales to 200K cores, performing science using O(10K) cores

- We are interested in testing the scalability of our codes on the next generation machines, which may have 24+ cores per node.