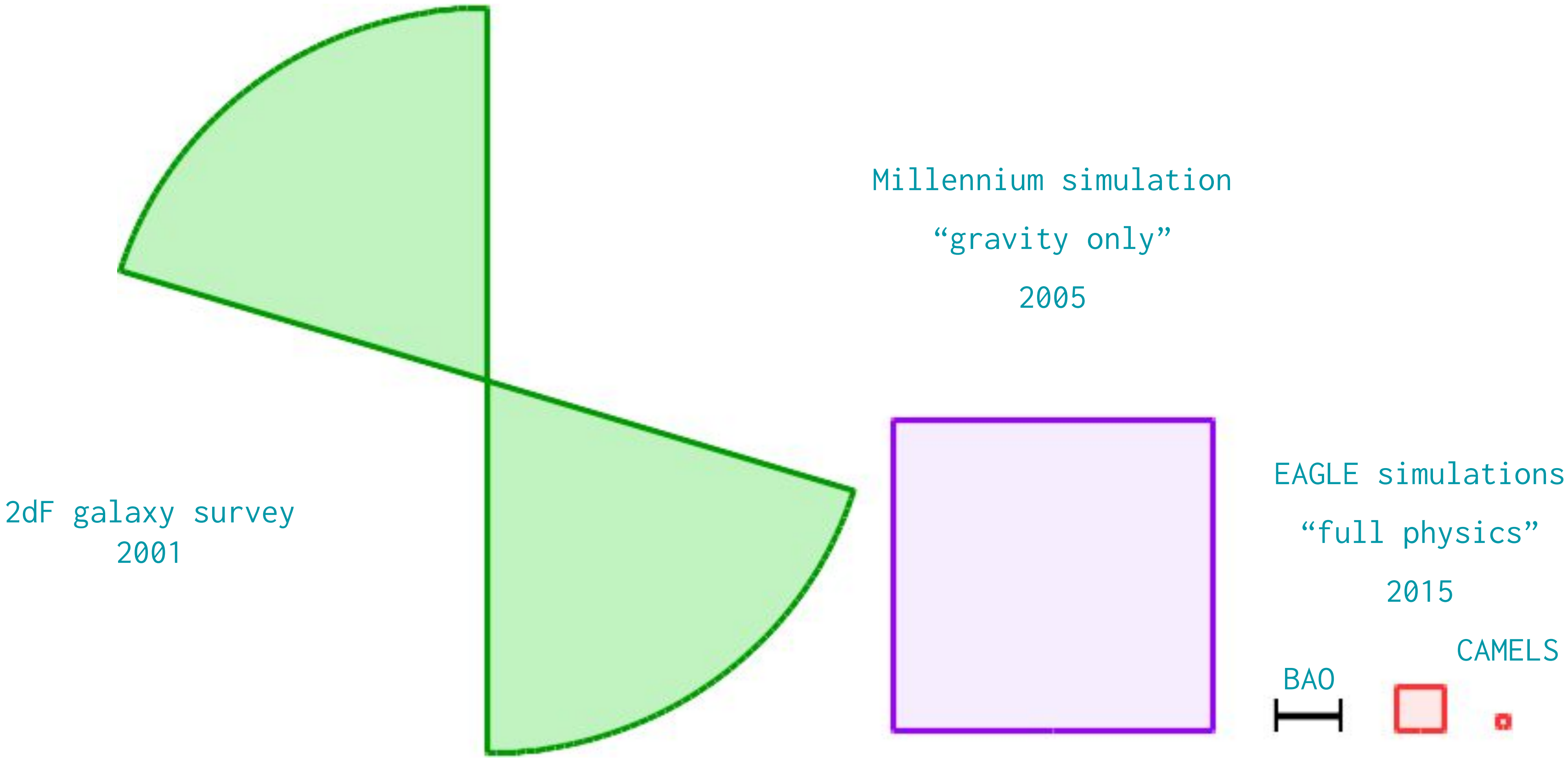
SPH With Interleaved Fine-grained Tasking

Matthieu Schaller

Lorentz Institute & Leiden Observatory

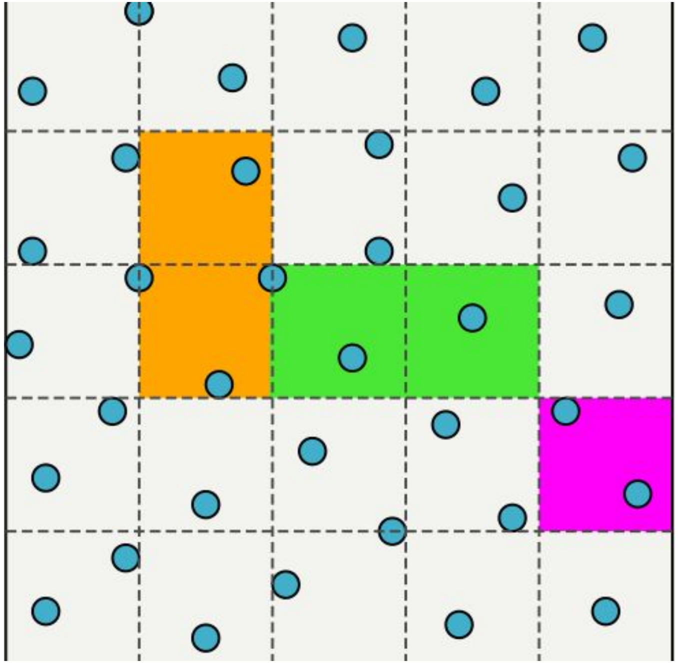
Cosmological scales in hydro runs?



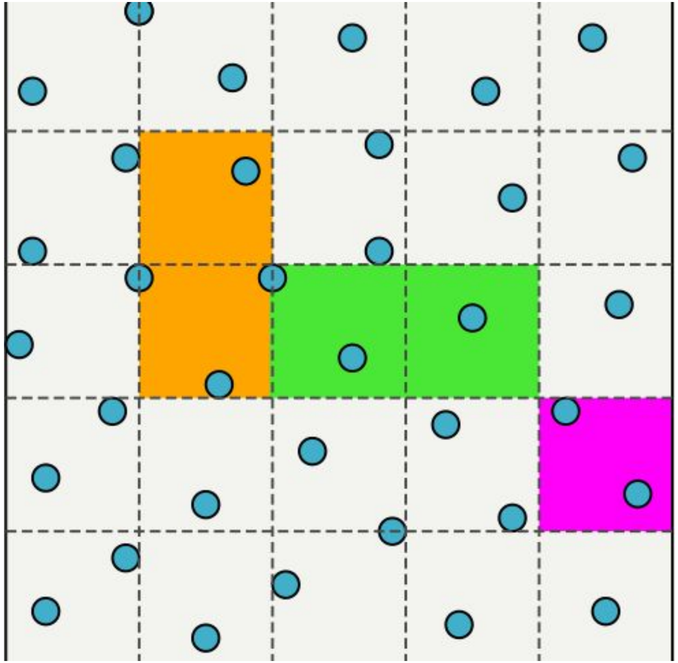
Design principles

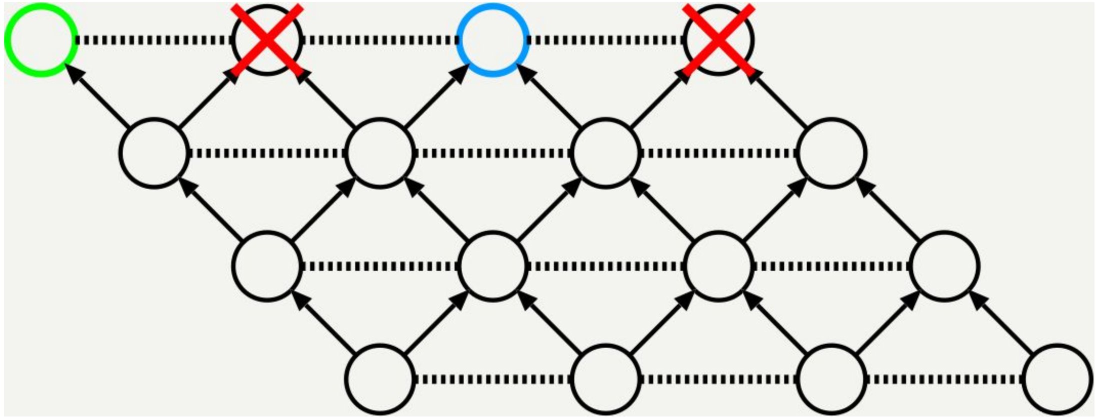
- Write a hydro solver (ngb finding) first, then gravity.
- Attempt to exploit all three levels of parallelism of modern clusters.
- Use dynamic scheduling of operations to reduce imbalances.
- All open-source, including all the detailed models (subgrid, ...).

Design principles



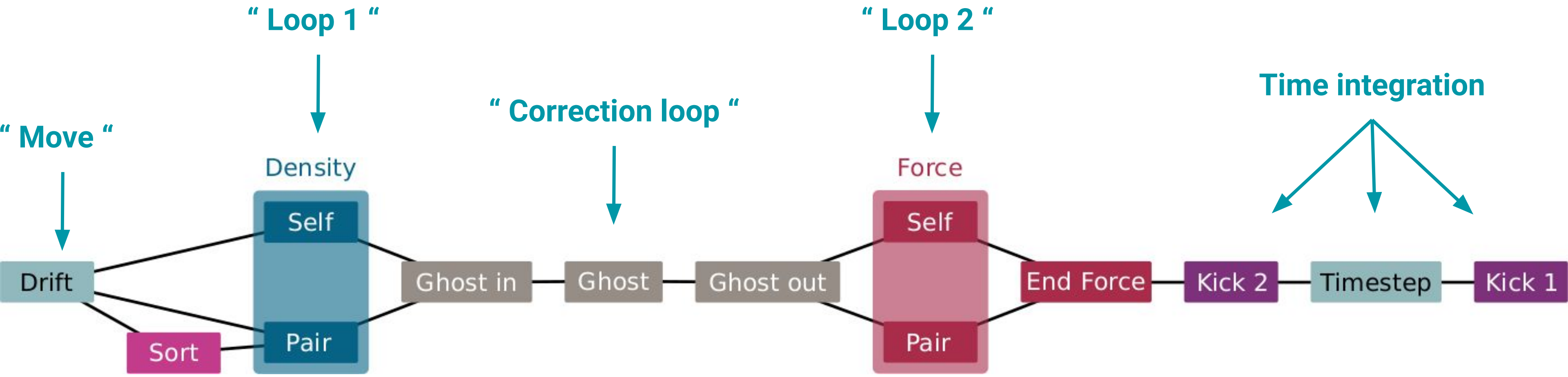
Design principles





Task-based parallelism for SPH

What happens to one cell “bundle” of particles during one time-step:

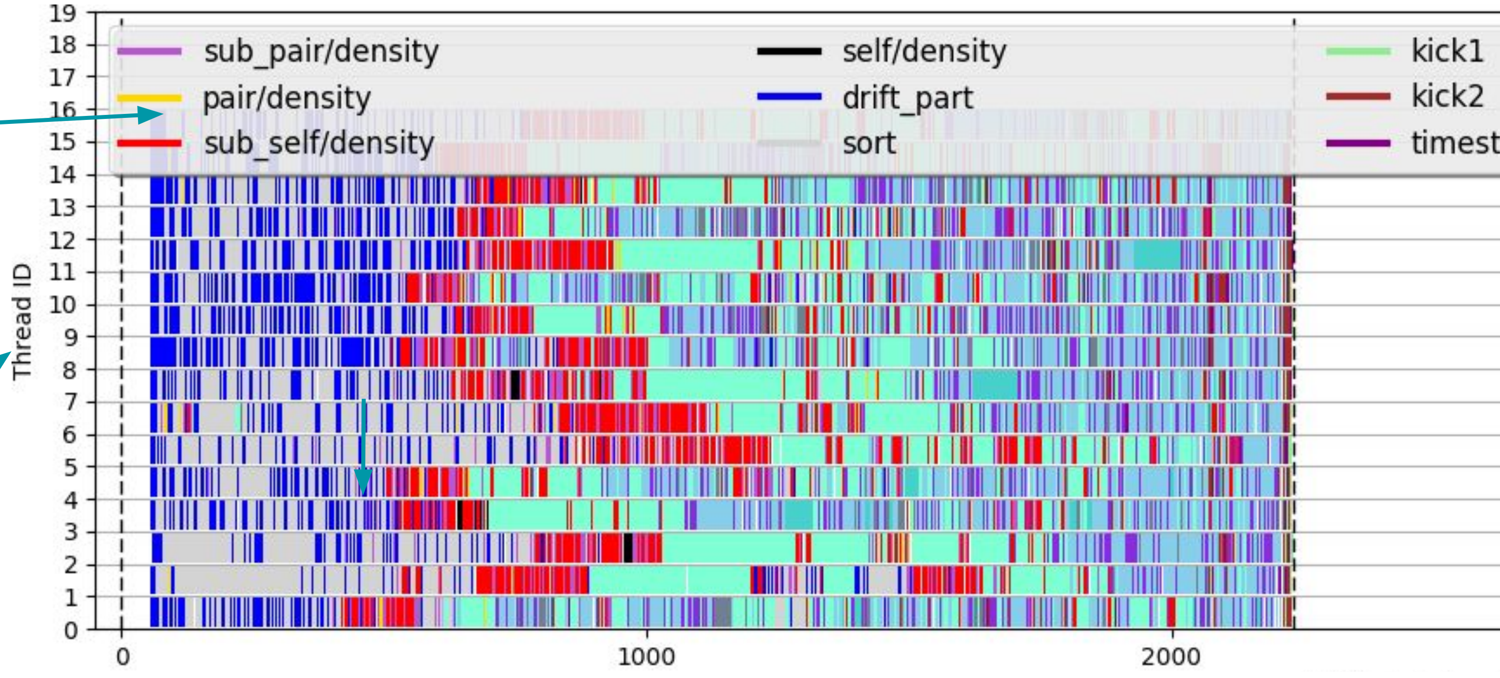


All the code within a task is very simple. No need for deep C knowledge
—> Easy to extend the code

Task-based scheduling

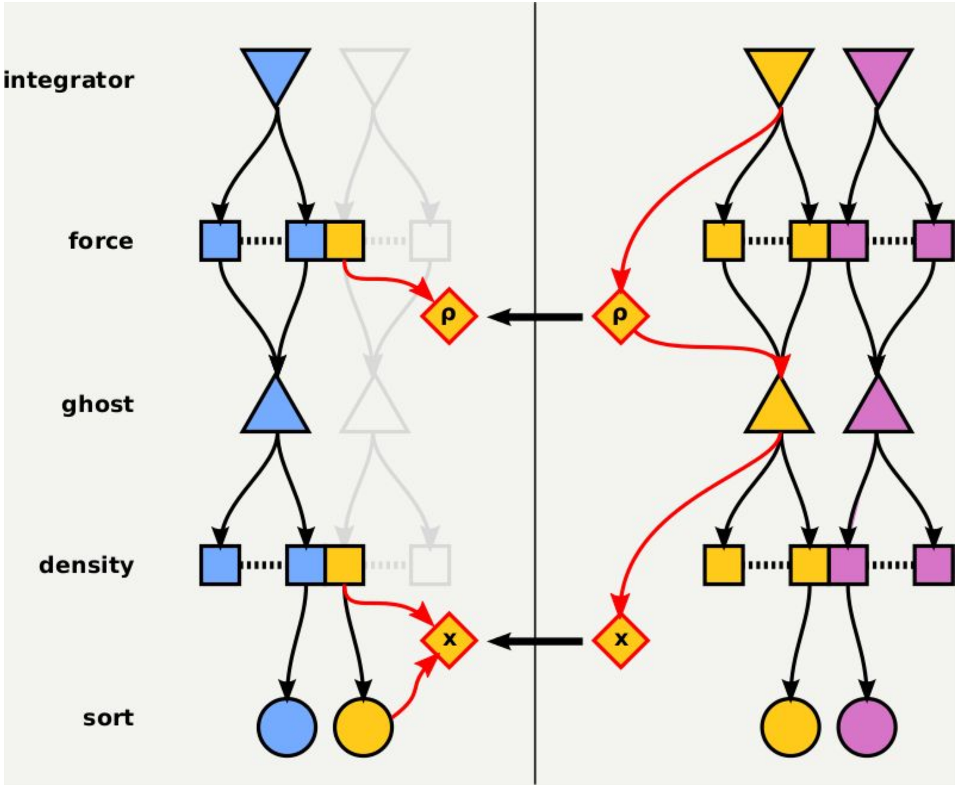
Physics loops ("tasks")

CPU cores



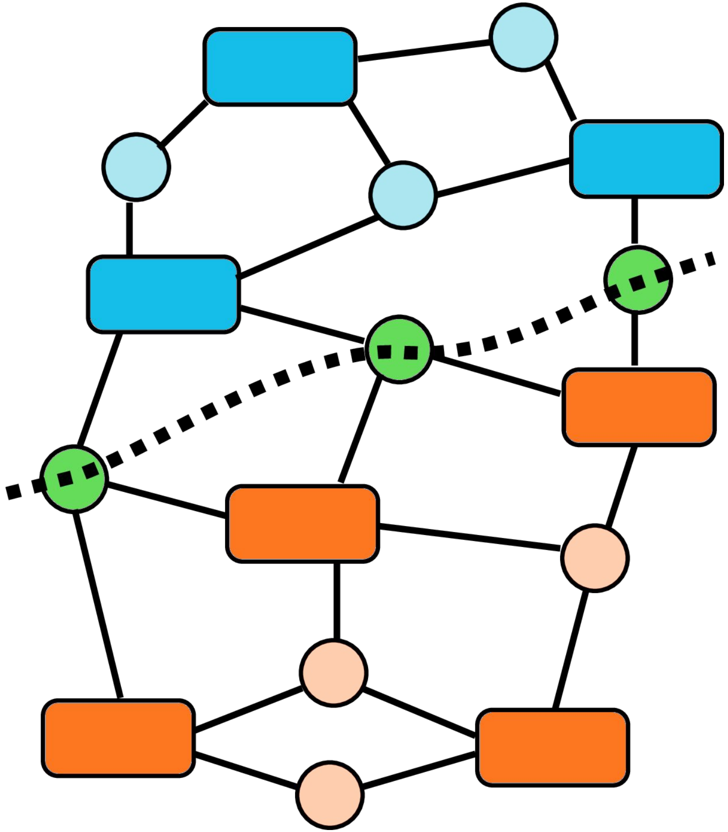
Wall-clock time [ms]

How about multiple nodes?



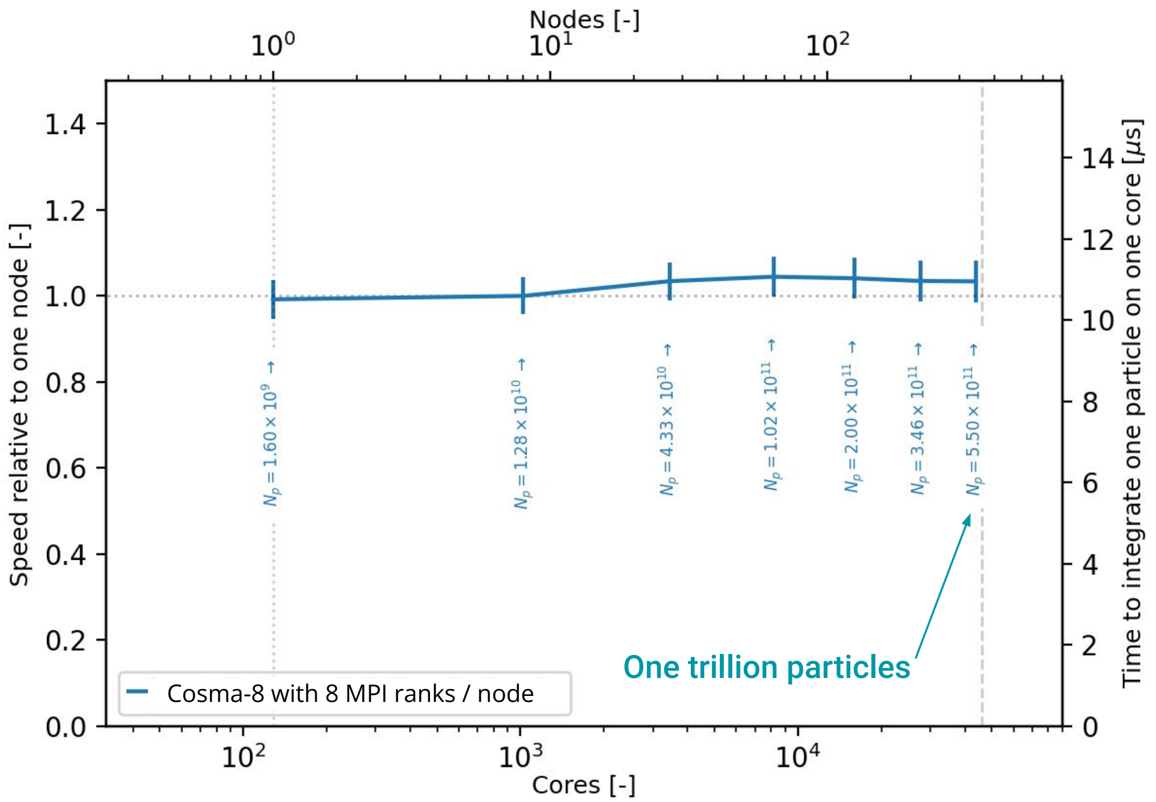
- Instead of sending all the particles and *then* compute, do it at the same time.
- Sending/receiving data is just another task type, and can be executed in parallel with the rest of the computation.
- Once the data has arrived, the scheduler unlocks the tasks that needed the data.

A Graph-based strategy



- For each task, we compute the amount of work (=runtime) required.
- We build a graph where the data are nodes and tasks are hyper-edges.
- Extra cost added for communication tasks to minimise them.
- `METIS` is used to split the graph such that the work (not the data!) is balanced.

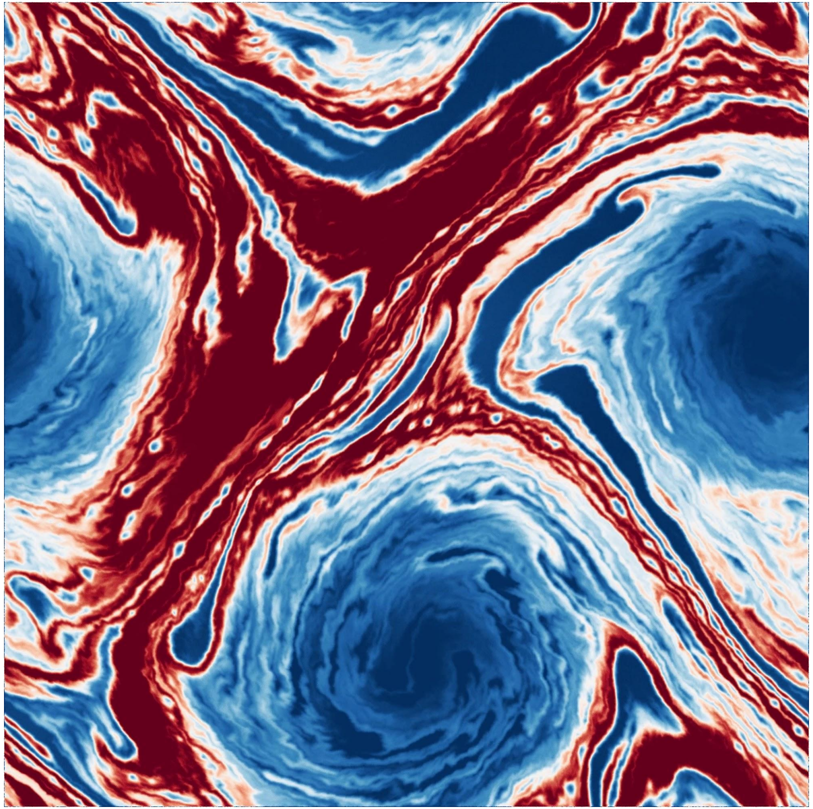
Weak-scaling to large systems



`DiRAC Cosma-8` **system @ Durham.**

360 nodes with
- **2x AMD 7H12**
- **1 TB of RAM**
- **HDR Inter-connect**

“SPHENIX” SPH flavour

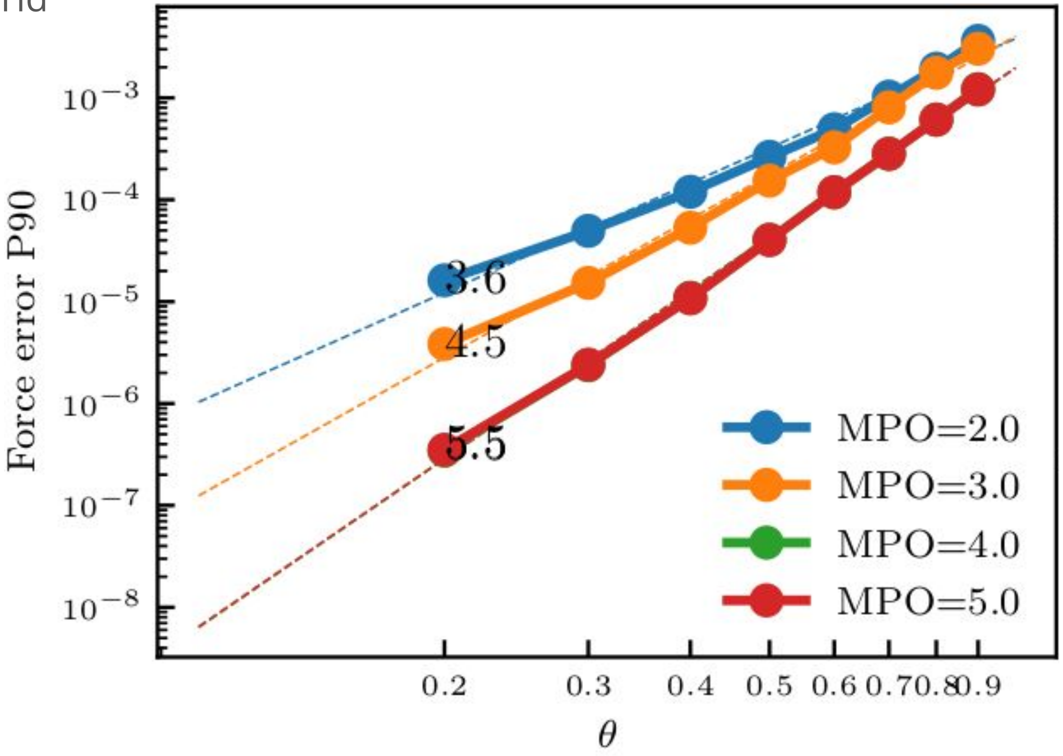


- Based on a density-energy formulation.
- Spatially-varying viscosity and diffusion (conduction) terms.
- Switches taylored for the needs of galaxy formation simulations (e.g. large feedback dumps).
- Time-step limiter.

Borrow+22

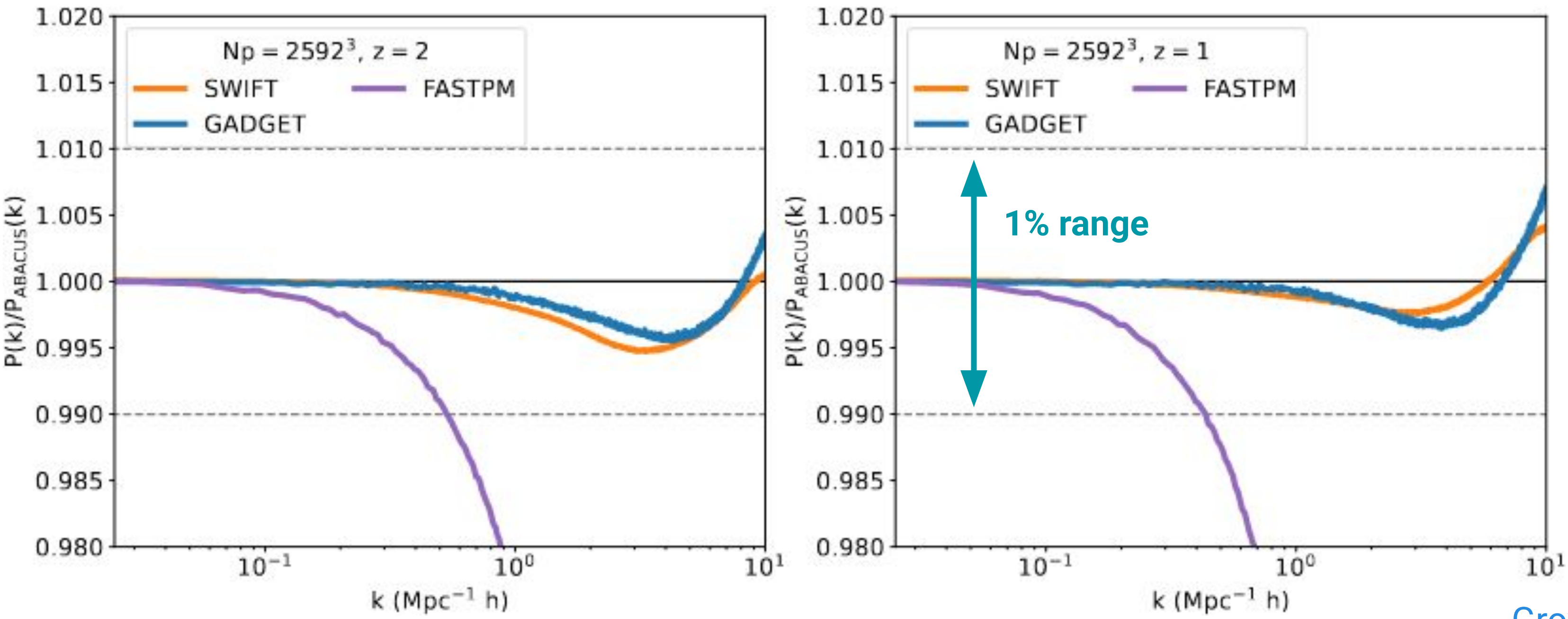
Accuracy checks - Basic gravity

- Gravity uses FMM coupled to a PM grid for periodic calculations.
- Adaptive “opening angle”.
- Mixed-precision arithmetic and exploits vector-instructions.
- Convergence properties agree with expectations.



Accuracy comparison - Cosmo runs

DESI code comparison effort — P(k) prediction — Codes compared to ABACUS.



Grove+21

Other components

- Particle-based " delta-f " neutrinos. (Elbers+21)
- SPH-based M1-closure RT solver. (Chan+21)
- Particle light-cones and healpix maps.
- On-the-fly FOF and power-spectra.
- Other SPH solvers (Anarchy, Gasoline2-like, PHANTOM-like) and Gizmo-MF[MV].
- Multiple networks of subgrid models (EAGLE, FLAMINGO, GEAR, AGN jets, ...) .

SWIFT-EAGLE model

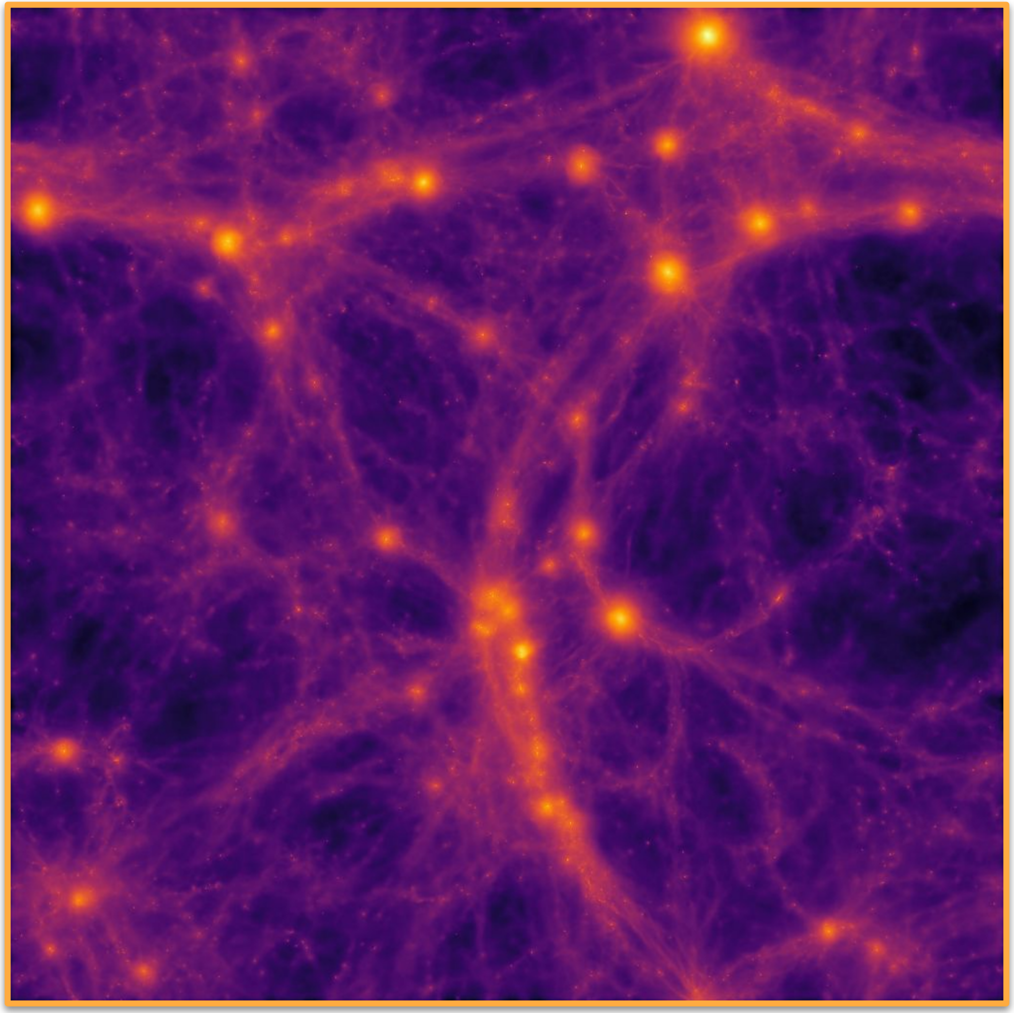
- Metal-line cooling using Ploeckinger+Schaye 2020 tables.
- Star formation threshold based on cold phase.
- Thermal (or kinetic) stochastic stellar feedback.
- Enrichment from SNII, SNIa, and AGB.
- AGN accretion + thermal feedback.
- Model parameters calibrated to GSMF + mass-size + BH masses at $m_{gas} = 10^6$

Key differences w.r.t to existing CAMELS:

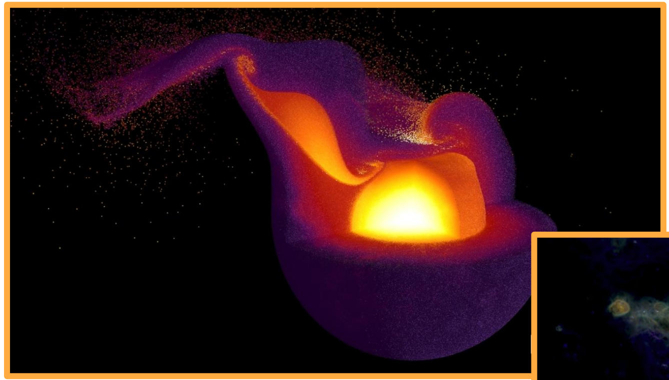
- No decoupled winds

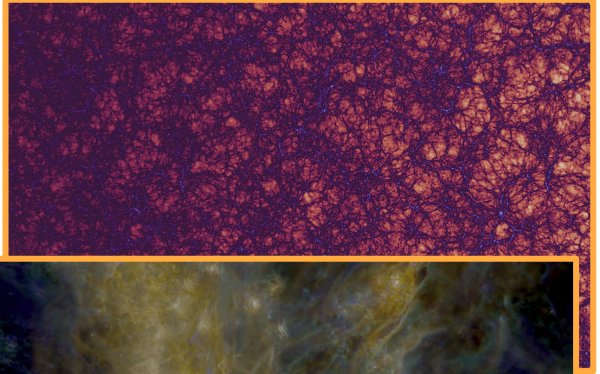
- Subgrid equations not linked to halo mass or redshift

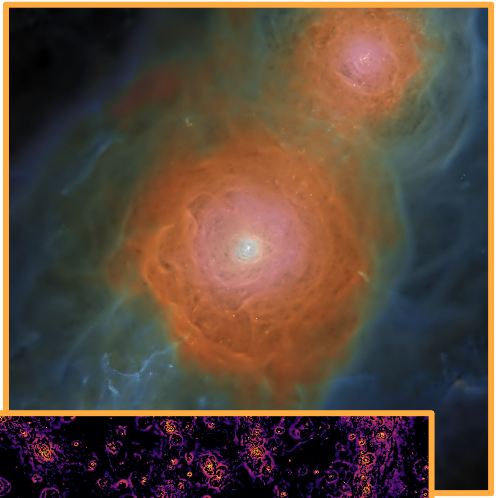
CAMELS plans

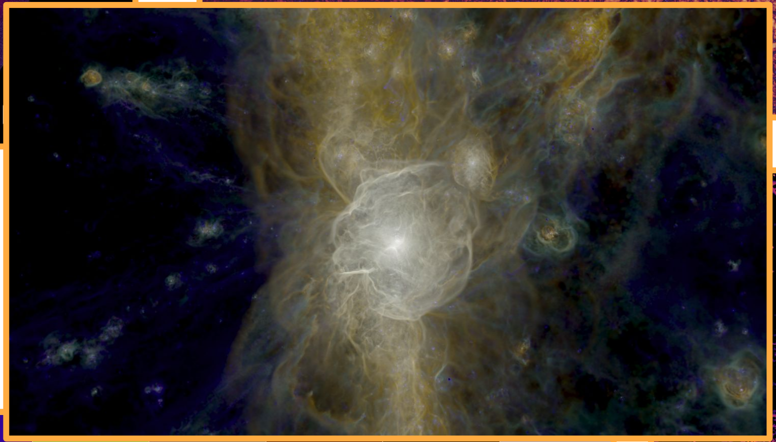


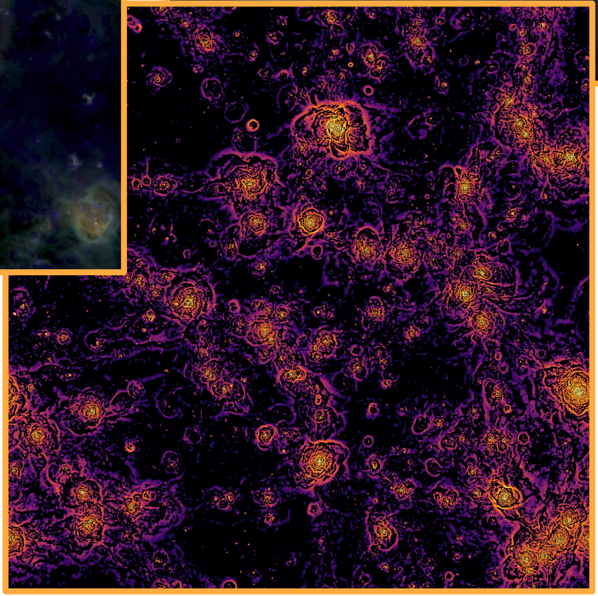
- monofonIC generator (using NGenIC phases)
- SWIFT code
- EAGLE-like model calibrated to $m_{gas} = 10^6$
- VELOCIraptor halo finder
- 600 CPU hours

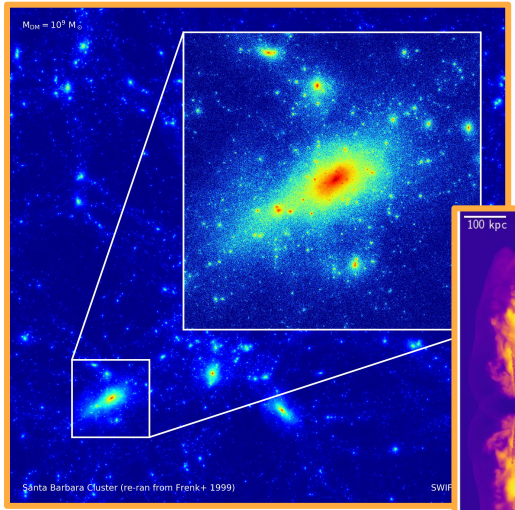


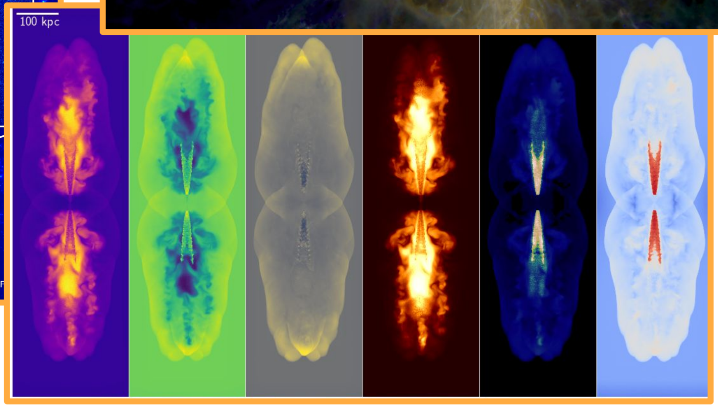












www.swiftsim.com