CENTRE OF EXCELLENCE FOR HPC
ASTROPHYSICAL APPLICATIONS

Optimization of the RAMSES code

Tine Colman

SF2A 2026

Co-funded by the European Union

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant agreement No 101093441.

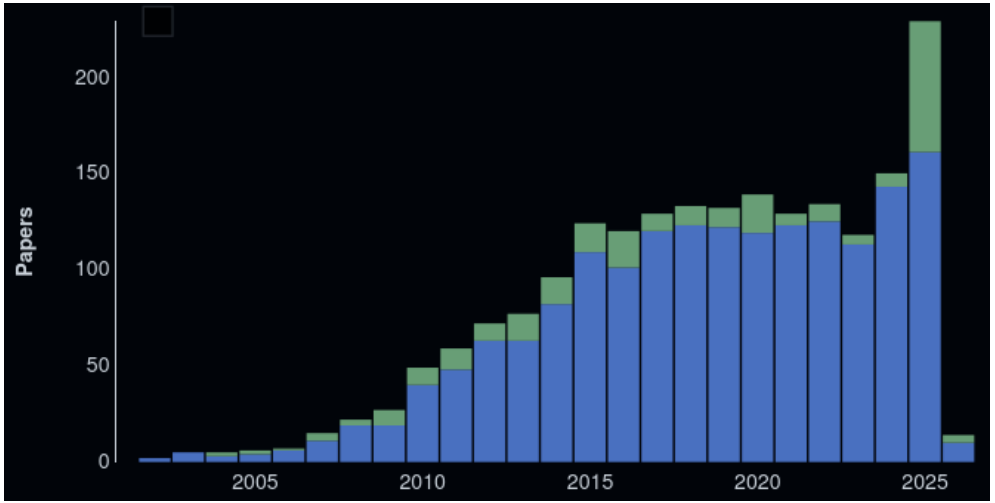
EuroHPC Joint Undertaking

The RAMSES code

Principle developer: Romain Teyssier

24 years of community developments

https://github.com/ramses-organisation/ramses



Evolution of the number of citations of the RAMSES first paper (Teyssier 2002)

Characteristics

- Fortran + MPI
- Grid-based with AMR & adaptive timestepping
- Collision-less particles (dark matter, tracers, …)

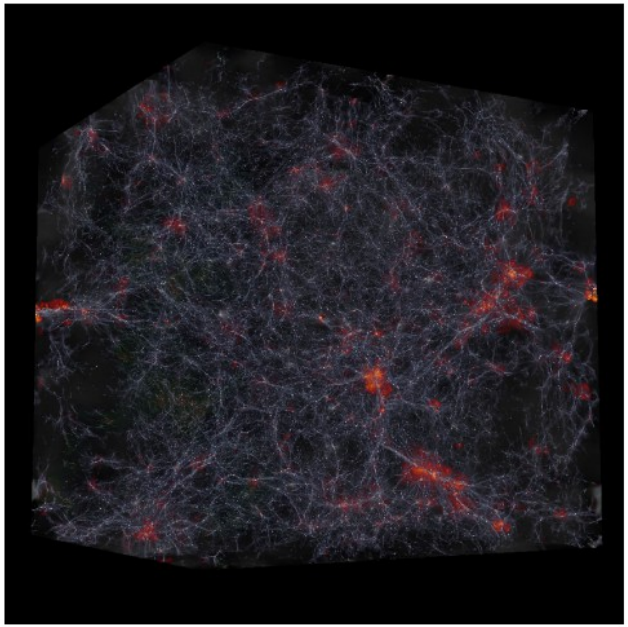
Physics

- Hydrodynamics / MHD
- (Self)-gravity
- Radiative transfer
- Star formation & feedback
- …

+ more being ported from private branches

Versatility of RAMSES

Mega Parsec ≅ 10^{24} cm



Joakim Rosdahl

Astrophysical applications

Cosmology

Galaxy formation and evolution

Interstellar medium

Star formation

Protostars and protostellar disks

Planet formation

Compressible turbulence

Stellar radius ≅ 10^{10} cm

Ahmad et al. 2023

The SPACE project

- Scalable Parallel Astrophysical Codes for Exascale
- 4 year European Center of Excellence project (01/2023 - 12/2026)
- Goal: prepare state-of-the-art astro codes to efficiently use exascale computation resources
- 7 codes: OpenGadget, ChaNGa, PLUTO, iPIC3D, RAMSES, FIL/GRACE, BHAC
- Collaboration: research institutes, computing centers, vendors

Types of optimizations

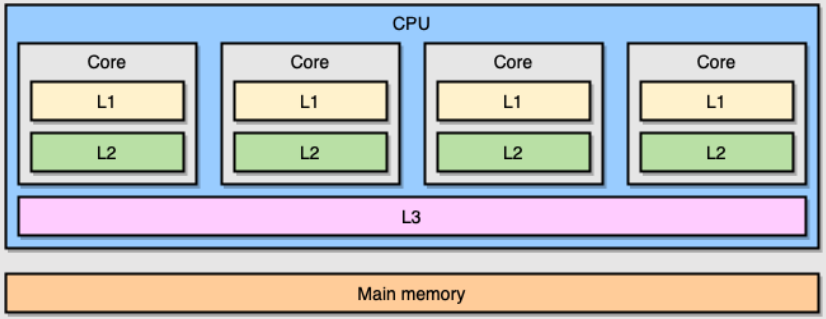
Algorithmic

"Same answer with less computation"

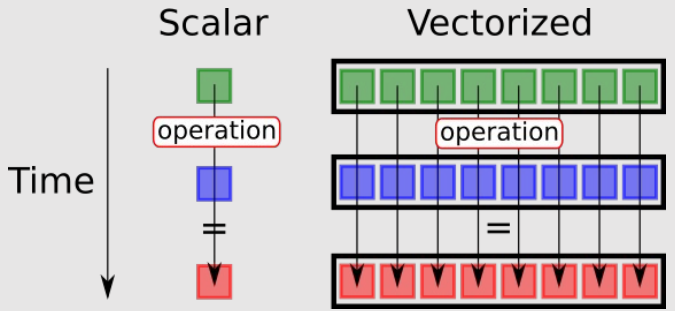
Parallelism

"Throw more hardware at the problem"

Memory



Vectorization (SIMD)



Compiler options

Port to GPU

Gprof: basic profiler from GNU

Information on:

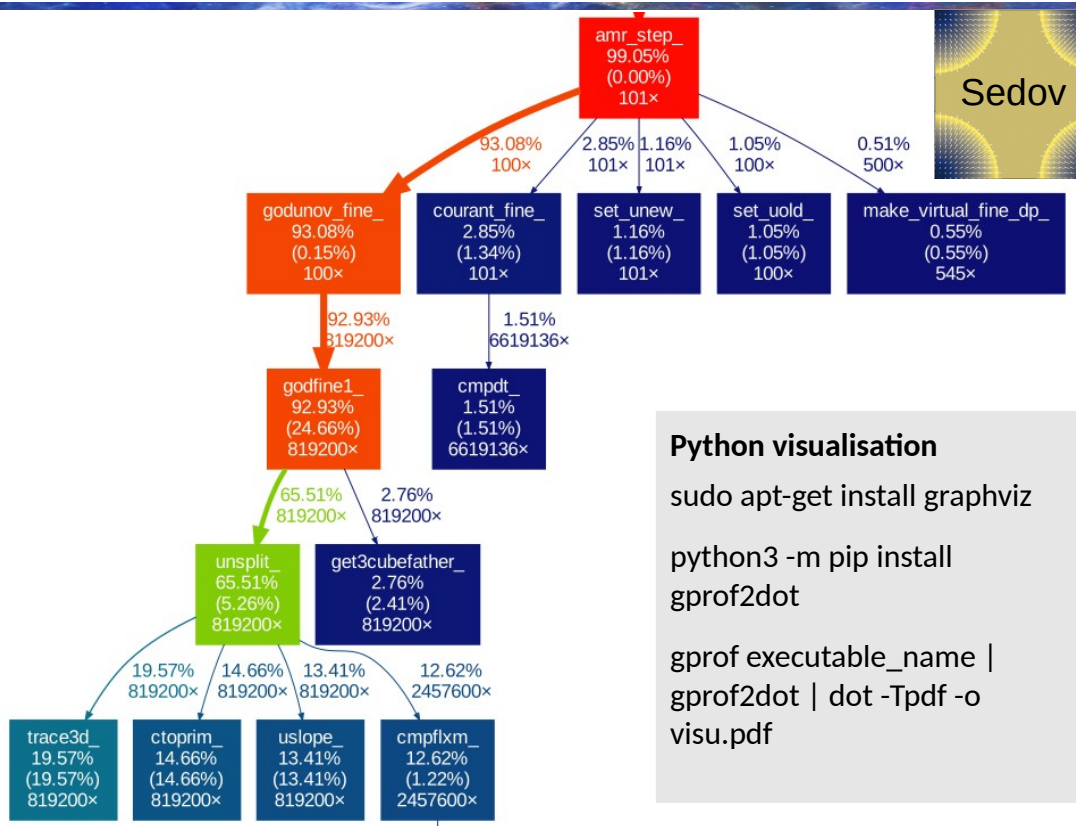
- Execution time per function → hot spots
- Number of function calls
- Call graph

Usage:

- Compile with -pg, then run as usual
- generates gmon.out file at the end of execution

Display results:

- gprof --flat-profile executable_name
- gprof --graph executable_name



Python visualisation

sudo apt-get install graphviz

python3 -m pip install gprof2dot

gprof executable_name | gprof2dot | dot -Tpdf -o visu.pdf

Advanced profiling with MAQAO

Loop id	Source Location	Source Function	Exclusive Coverage run_0 (%)	Vectorization Ratio (%)
1313	ramses_dev_gnu3d - umuscl.f90:530-666 [...]	trace3d	19.79	100
1190	ramses_dev_gnu3d - godunov_fine.f90:594-599	godfine1	19.52	0
1354	ramses_dev_gnu3d - umuscl.f90:894-941 [...]	ctoprim	7.19	100
1344	ramses_dev_gnu3d - umuscl.f90:1271-1279	uslope	3.19	30.51
1346	ramses_dev_gnu3d - umuscl.f90:1247-1255	uslope	3.13	30.51
1256	ramses_dev_gnu3d - godunov_fine.f90:62-63	set_unew	3.09	0
1345	ramses_dev_gnu3d - umuscl.f90:1259-1267	uslope	3.02	30.51
1263	ramses_dev_gnu3d - godunov_fine.f90:183-184	set_uold	2.98	0
1232	ramses_dev_gnu3d - godunov_fine.f90:750-754	godfine1	1.56	0
1237	ramses_dev_gnu3d - godunov_fine.f90:711-713	godfine1	1.39	0
1465	ramses_dev_gnu3d - godunov_utils.f90:694-696	riemann_llf	1.27	88.89

Modular Assembly Quality Analyzer and Optimizer

Gives information on issues/bottlenecks for each loop

Optimizing the hydro solver

Loop id	Source Location	Source Function	Exclusive Coverage run_0 (%)	Vectorization Ratio (%)
1313	ramses_dev_gnu3d - umuscl.f90:530-666 [...]	trace3d	19.79	100
1190	ramses_dev_gnu3d - godunov_fine.f90:594-599	godfine1	19.52	0
1354	ramses_dev_gnu3d - umuscl.f90:894-941 [...]	ctoprim	7.19	100
1344	ramses_dev_gnu3d - umuscl.f90:1271-1279	uslope	3.19	30.51
1346	ramses_dev_gnu3d - umuscl.f90:1247-1255	uslope	3.13	30.51

gain | potential | hint | expert

Vectorization

Your loop is fully vectorized, using full register length.

Details

All SSE/AVX instructions are used in vector version (process two or more data elements in vector registers).

Execution units bottlenecks

Performance is limited by execution of FP multiply or FMA (fused multiply-add) operations (the FP multiply/FMA unit is a bottleneck). By removing all these bottlenecks, you can lower the cost of an iteration from 51.00 to 47.17 cycles (1.08x speedup).

Workaround

Reduce the number of FP multiply/FMA instructions

Optimizing the hydro solver

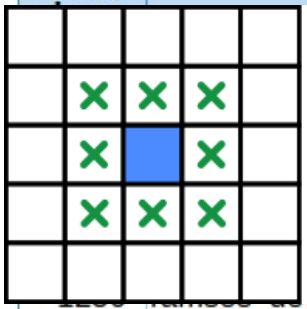
Loop id	Source Location	Source Function	Exclusive Coverage run_0 (%)	Vectorization Ratio (%)
1313	ramses_dev_gnu3d - umuscl.f90:530-666 [...]	trace3d	19.79	100
1190	ramses_dev_gnu3d - godunov_fine.f90:594-599	godfine1	19.52	0
1354	ramses_dev_gnu3d - umuscl.f90:894-941 [...]	ctoprim	7.19	100
1344	ramses_dev_gnu3d - umuscl.f90:1271-1279	uslope	3.19	30.51
1346	ramses_dev_gnu3d - umuscl.f90:1247-1255	uslope	3.13	30.51
1256	ramses_dev_gnu3d - godunov_fine.f90:62-63	set_unew	3.09	0
1345	ramses_dev_gnu3d - umuscl.f90:1259-1267	uslope	3.02	30.51
1263	ramses_dev_gnu3d - godunov_fine.f90:183-184	set_uold	2.98	0
1232	ramses_dev_gnu3d - godunov_fine.f90:750-754	godfine1	1.56	0
1237	ramses_dev_gnu3d - godunov_fine.f90:711-713	godfine1	1.39	0
1465	ramses_dev_gnu3d - godunov_utils.f90:694-696	riemann_llf	1.27	88.89

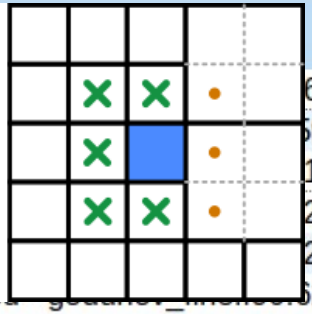
Execution units bottlenecks

Performance is limited by execution of divide and square root operations (the divide/square root unit is a bottleneck). By removing all these bottlenecks, you can lower the cost of an iteration from 14.50 to 9.50 cycles (1.53x speedup).

Sound speed computed but never used, sqrt expensive
→ remove it

Optimizing the hydro solver





	Source Function	Exclusive Coverage run_0 (%)	Vectorization Ratio (%)
6 [...]	trace3d	19.79	100
594-599	godfine1	19.52	0
1 [...]	ctoprim	7.19	100
279	uslope	3.19	30.51
255	uslope	3.13	30.51
62-63	set_unew	3.09	0
			30.51
			0
			0
			0
			88.89

Source Code

/home/tcolman/Codes/ramses_github_tine/bin/../hydro/godunov_fine.f90: 594 - 599

```
594:          do ivar=1,nvar
595:             do i=1,nexist
596:                uloc(ind_exist(i),i3,j3,k3,ivar)=uold(ind_cell(i),ivar)
597:             end do
598:             do i=1,nbuffer
599:                uloc(ind_nexist(i),i3,j3,k3,ivar)=u2(i,ind_son,ivar)
```

Circumvent check when grid is uniform
→ **~10% speedup**

Optimizing the hydro solver

Loop id	Source Location	Source Function	Exclusive Coverage run_0 (%)	Vectorization Ratio (%)
1313	ramses_dev_gnu3d - umuscl.f90:530-666 [...]	trace3d	19.79	100
1190	ramses_dev_gnu3d - godunov_fine.f90:594-599	godfine1	19.52	0
1354	ramses_dev_gnu3d - umuscl.f90:894-941 [...]	ctoprim	7.19	100
1344	ramses_dev_gnu3d - umuscl.f90:1271-1279	uslope	3.19	30.51
1346	ramses_dev_gnu3d - umuscl.f90:1247-1255	uslope	3.13	30.51
1256	ramses_dev_gnu3d - godunov_fine.f90:62-63	set_unew	3.09	0
1345	ramses_dev_gnu3d - umuscl.f90:1259-1267	uslope	3.02	30.51
1263	ramses_dev_gnu3d - godunov_fine.f90:183-184	set_uold	2.98	0
1232	ramses_dev_gnu3d - godunov_fine.f90:750-754	godfine1	1.56	0
1237	ramses_dev_gnu3d - godunov_fine.f90:711-713	godfine1	1.39	0
1465	ramses_dev_gnu3d - godunov_utils.f90:694-696	riemann_llf	1.27	88.89

Optimizing the hydro solver

Before: slopes calculated in advance and stored in large temporary arrays

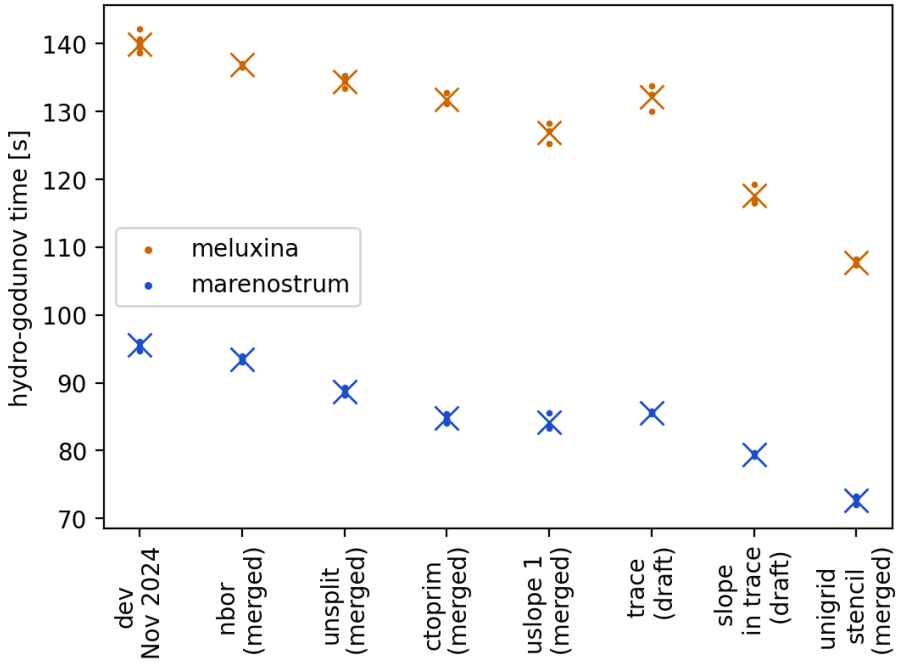
Now: call uslope inside loop in trace, to improve memory locality

→ **~10% speedup**

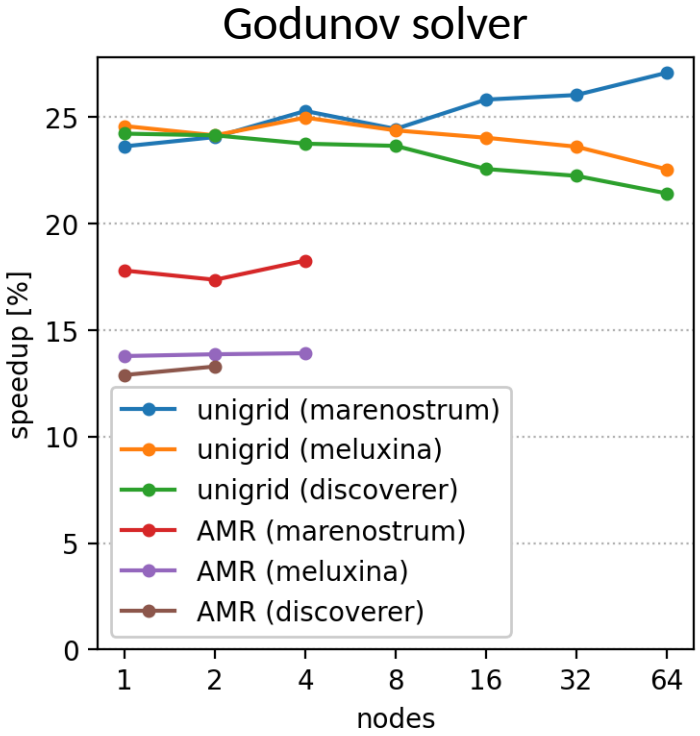
```
-    ! Compute TVD slopes
-    call uslope(qin,dq,dx,dt,ngrid)
-
     ! Compute 3D traced-states in all three directions
     if(scheme=='muscl')then
  #if NDIM==1
-       call trace1d(qin,dq,qm,qp,dx      ,dt,ngrid)
+       call trace1d(qin,qm,qp,dx      ,dt,ngrid)
  #endif
```

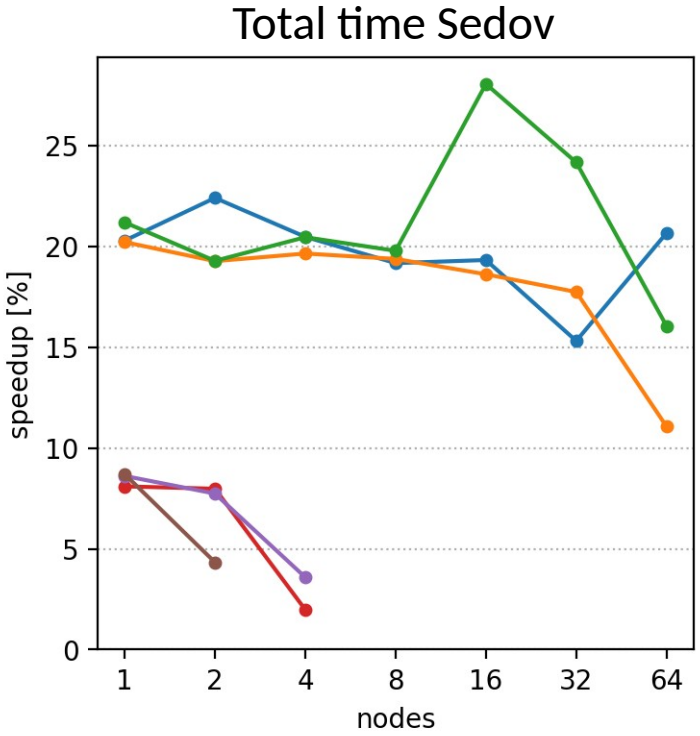
```
     real(dp),dimension(1:nvector,iu1:iu2,ju1:ju2,ku1:ku2,1:nvar)::q
-    real(dp),dimension(1:nvector,iu1:iu2,ju1:ju2,ku1:ku2,1:nvar,1:ndim)::dq
+    real(dp),dimension(1:nvector,1:nvar,1:ndim)::dq
     real(dp),dimension(1:nvector,iu1:iu2,ju1:ju2,ku1:ku2,1:nvar,1:ndim)::qm
     real(dp),dimension(1:nvector,iu1:iu2,ju1:ju2,ku1:ku2,1:nvar,1:ndim)::qp


  @@ -163,6 +157,9 @@ subroutine trace1d(q,dq,qm,qp,dx,dt,ngrid)

     do k = klo, khi
        do j = jlo, jhi
           do i = ilo, ihi
+             ! Cell-centered slopes
+             call uslope(q,dq,dtdx,i,j,k,ngrid)
+
              do l = 1, ngrid
```

Optimizing the hydro solver



Optimizing the hydro solver





MPI + OpenMP implementation

Shared memory parallelism inside nodes

=> reduce number of MPI domain

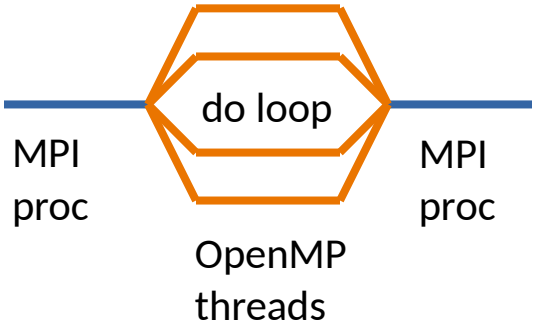
=> less time spent communicating

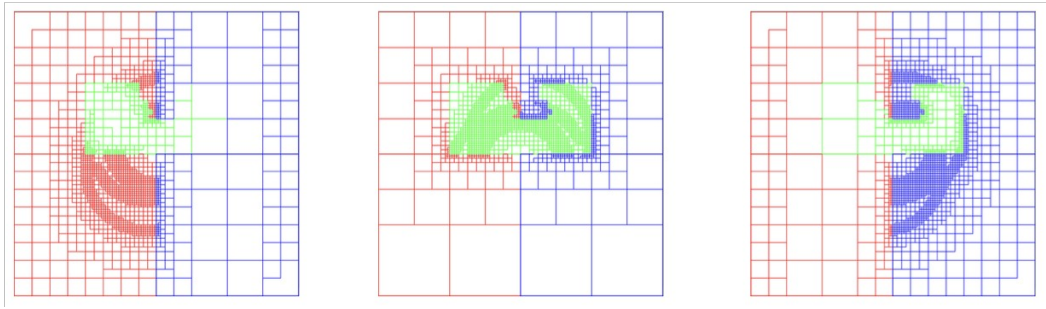
=> **improved scalability**

=> reduce number of MPI domain

=> reduced memory imprint ghost zones

=> **reduced memory usage**

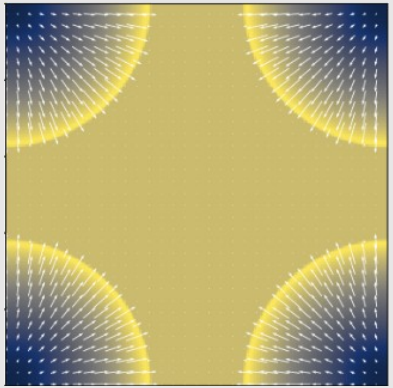




Caveat: need to add OpenMP everywhere

Starting point: RAMSES-yOMP

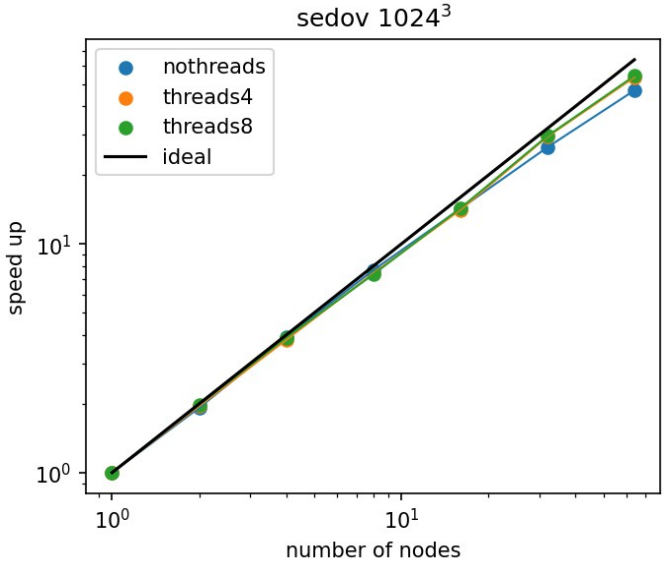
OpenMP for use-case 1: SEDOV ✅

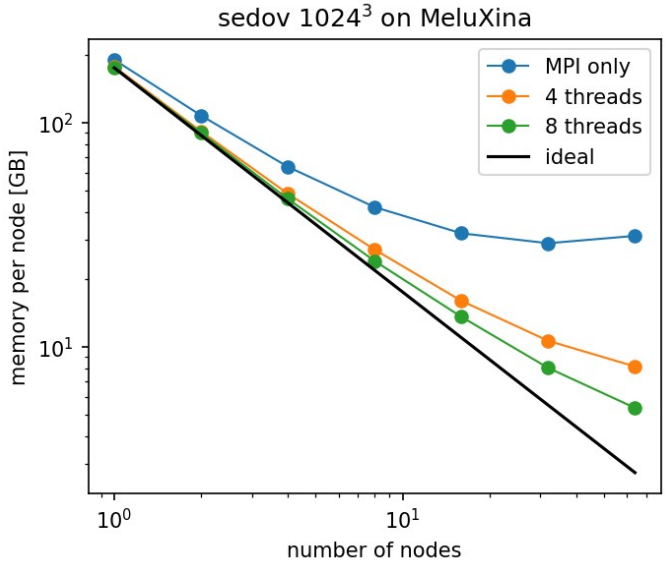


Setup: blastwave (unigrid)

Module: hydro

- Godunov solver
- communication

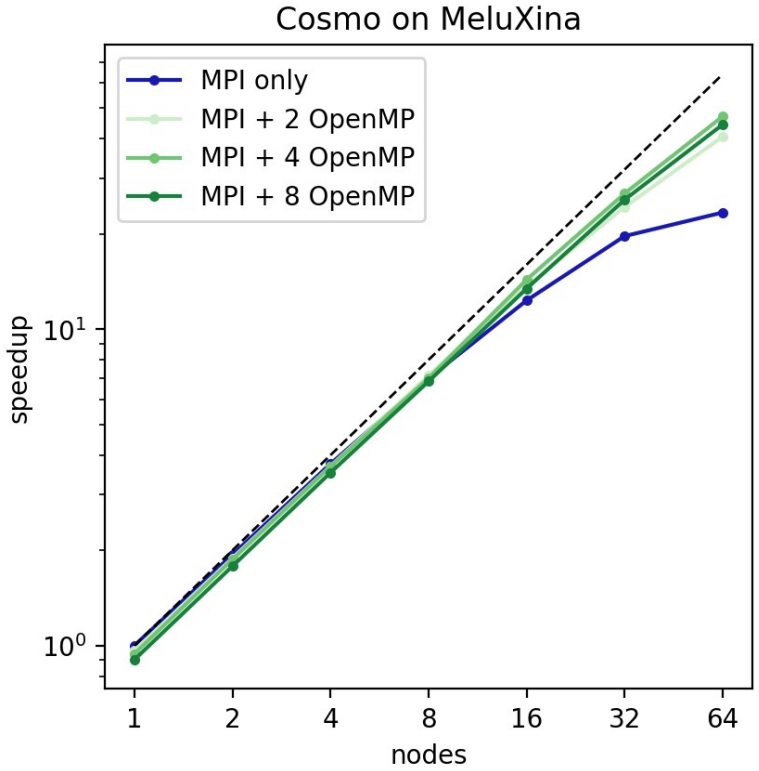




Results: small improvement in strong scaling

factor x4 decrease in memory on 64 nodes

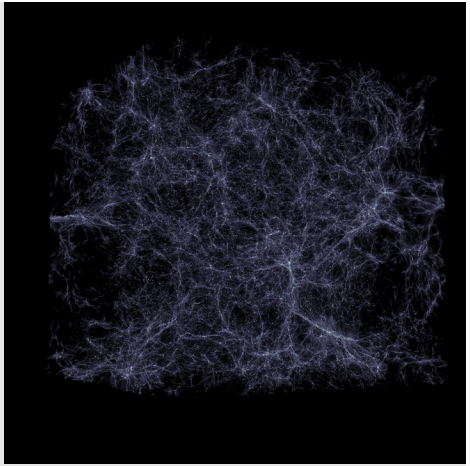
OpenMP for use-case 2: COSMO ✅



Setup: DM-only (unigrid)

Module: gravity

- Poisson multigrid solver
- Force calculation
- Particle-mesh (Cloud-in-cell)
- Moving of particles

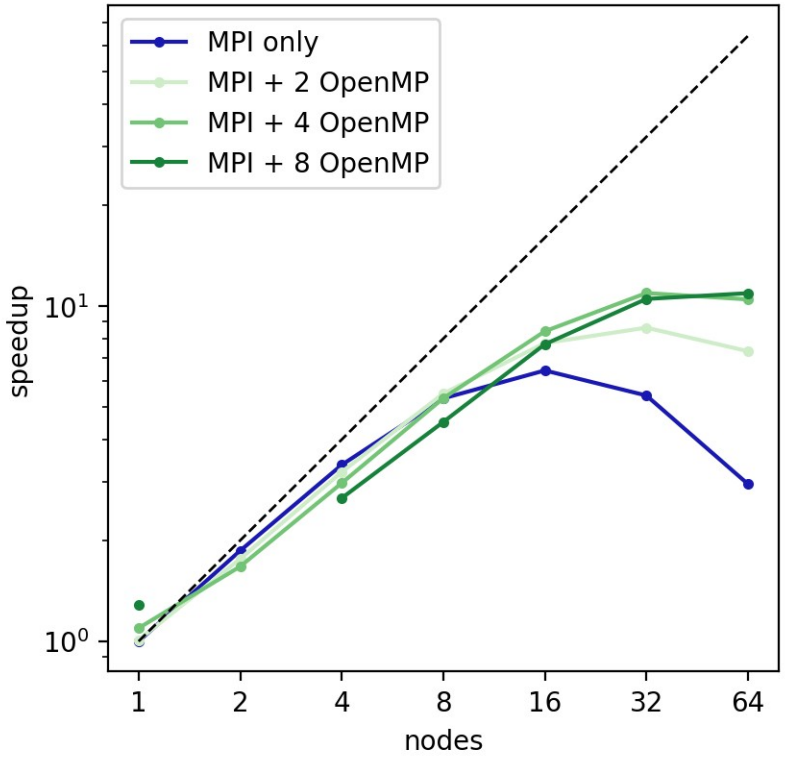


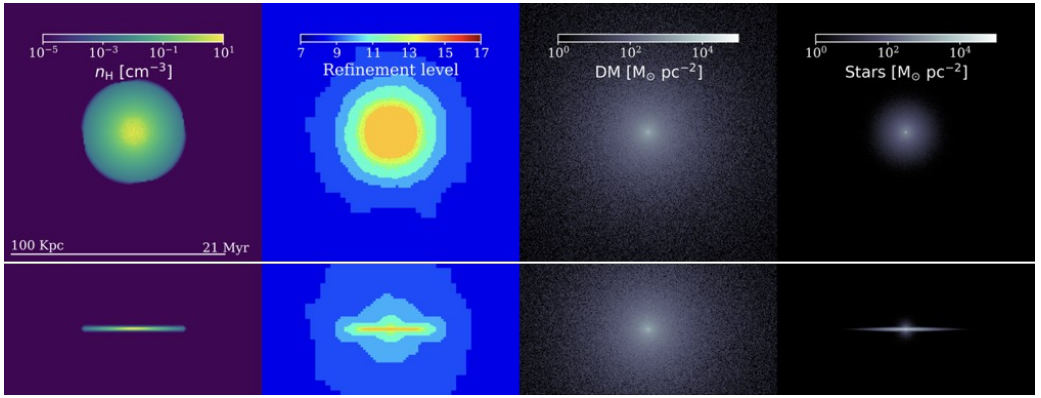
Results on 64 nodes:

x2 faster

scaling 37% → 74%

OpenMP for use-case 3: GALAXY ✅





Setup: gas, DM & stars particles

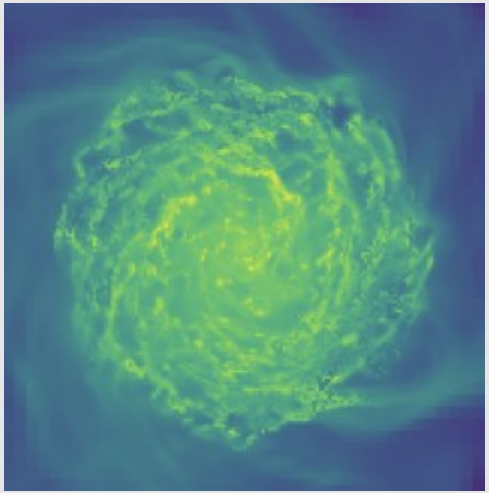
Modules: hydro + gravity **+ AMR**

Results: scaling from awful to less bad

OpenMP in realistic setups

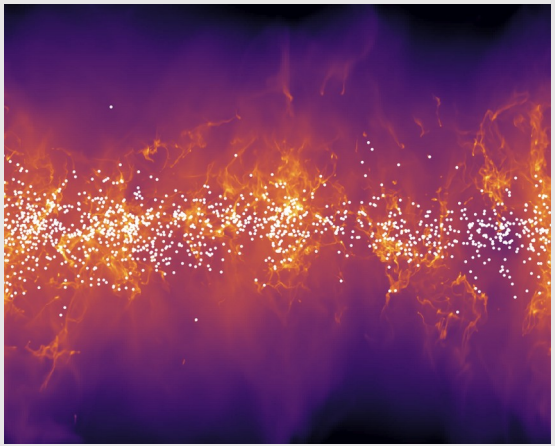
Star forming galaxy

- Star formation (in test)
- Stellar feedback (WIP)



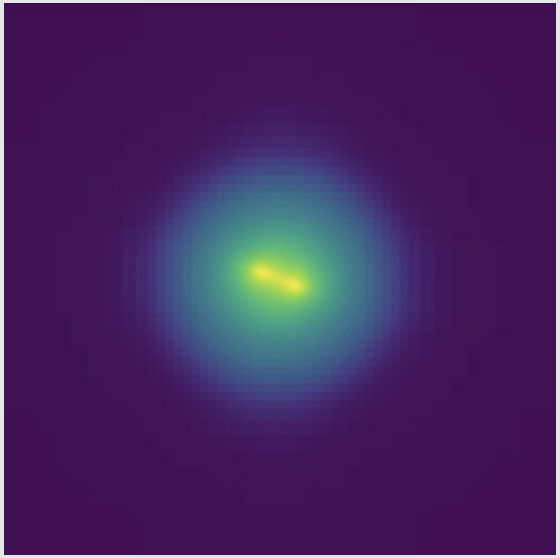
ISM in kpc box

- Sink formation (to do)
- Sink feedback (to do)
- RT (to test)

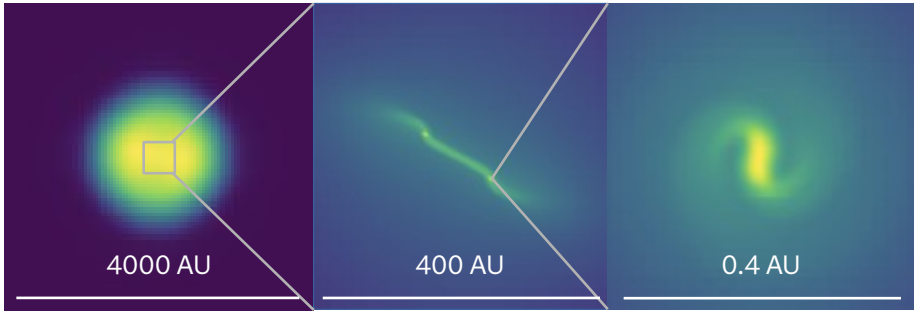


Collapsing core

- MHD (done)



OpenMP for collapse simulations



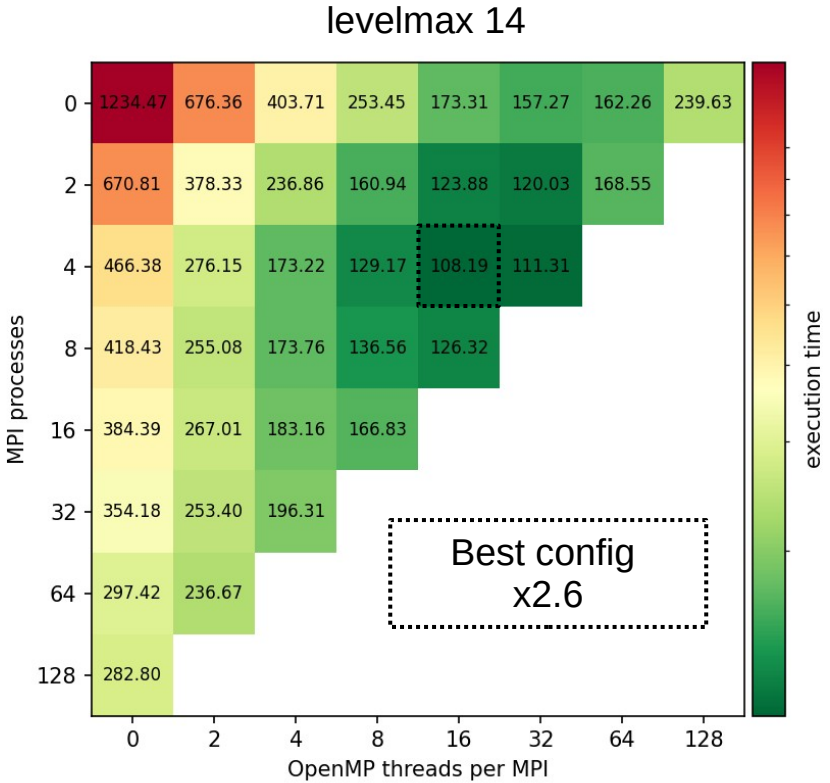
Physics: MHD, gravity, EOS

Challenge: deep AMR

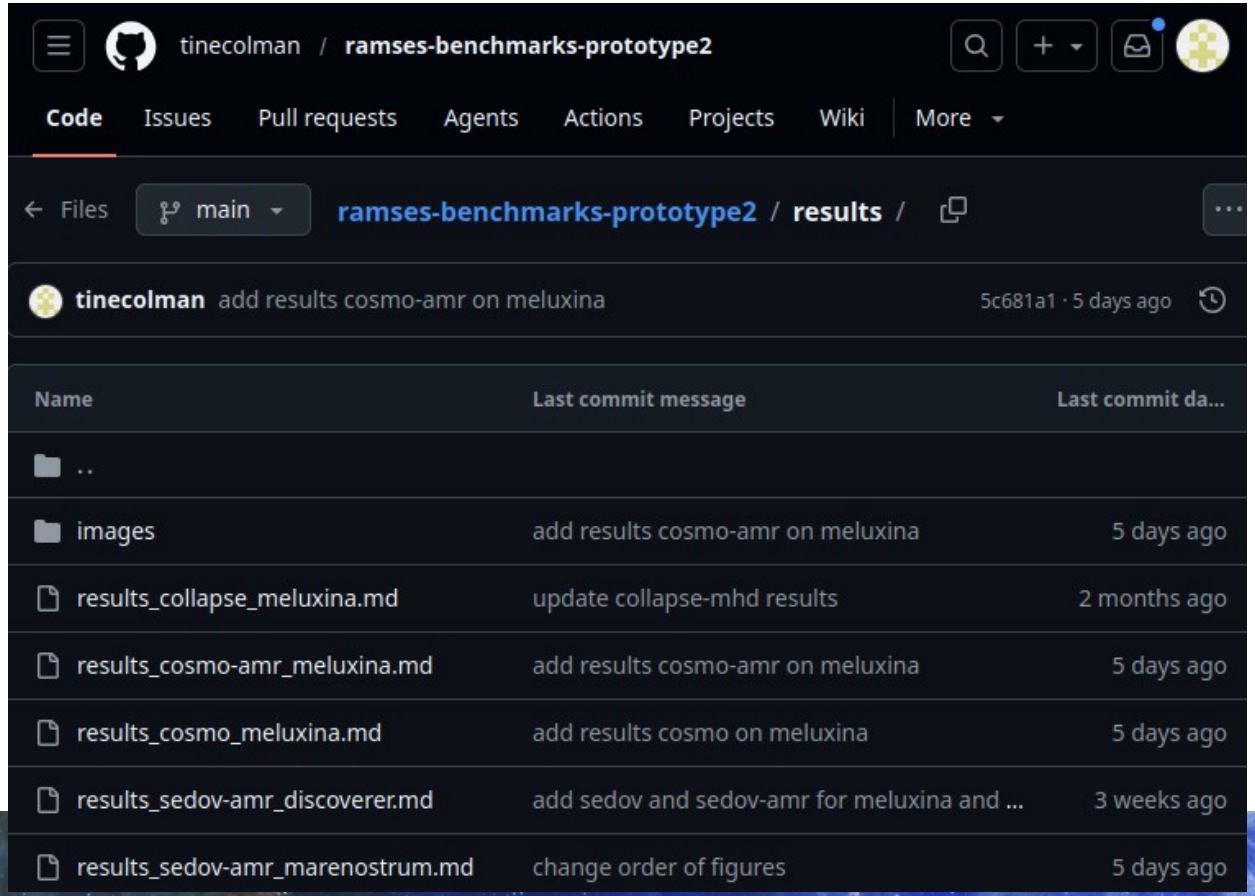
Results

- levelmax 14: speedup x2.6
- levelmax 20: speedup x4-5

  factor 10 - 20 less memory needed



Benchmark database



Benchmark database

Benchmark: sedov on meluxina

Benchmark description: sedov

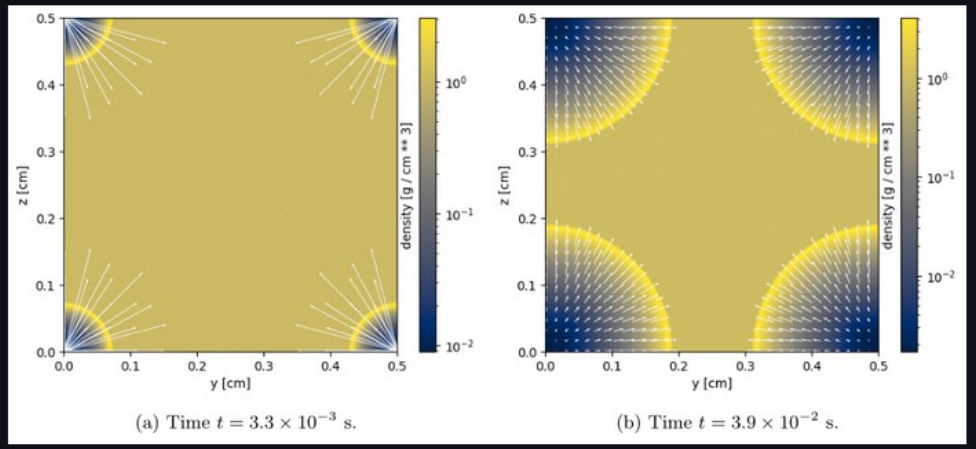
Cluster info: meluxina

On this page:

- Evolution of execution time with code version (figure &
- Strong scaling evolution (figure & table with efficiency)
- [if unigrid] Weak scaling evolution (figure & table with e
- [if unigrid] Combination of strong and weak scaling of t
- Optimal MPI-OpenMP configuration (figures)
- Memory usage (figure & table)

Sedov blast wave on a uniform grid benchmark

Benchmark type: Classical, Sedov



Description of the setup

This test is a standard hydrodynamical test used for verification and benchmarking of gas dynamic codes. We run this test in 3D. The physical setup consists of a uniform density ($\rho =$

Benchmark database

Benchmark: sedov on mel

Benchmark description: sedov

Cluster info: meluxina

On this page:

- Evolution of execution time with code version (figur
- Strong scaling evolution (figure & table with efficien
- [if unigrid] Weak scaling evolution (figure & table wi
- [if unigrid] Combination of strong and weak scaling
- Optimal MPI-OpenMP configuration (figures)
- Memory usage (figure & table)

Info

Hosted by LuxProvide in Luxemburg.

| Docs | User portal | System status | News & Events |

Support email: servicedesk@lxp.lu (create ticket on user portal)

Characteristics

TODO which cpu etc

Access

Applying for time

Time on the MeluXina cluster can be obtained through EuroHPC.

Account creation

After the project leader creates an account for you, you will receive an email with your username and instructions to login to the user portal. The procedure is also detailed here.

Benchmark database

Benchmark: sedov on meluxina

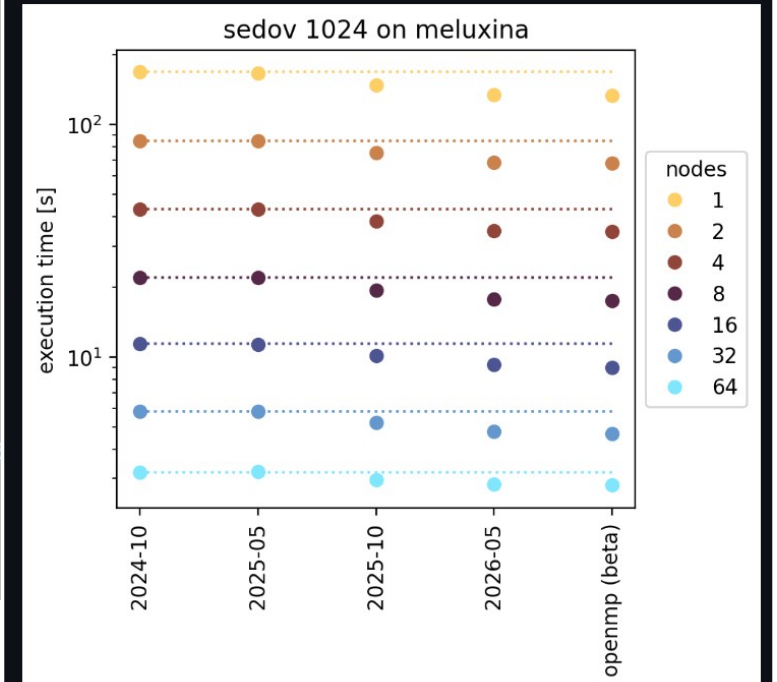
Benchmark description: sedov

Cluster info: meluxina

On this page:

- Evolution of execution time with code version (figure & table with speedup)
- Strong scaling evolution (figure & table with efficiency)
- [if unigrid] Weak scaling evolution (figure & table with efficiency)
- [if unigrid] Combination of strong and weak scaling of the latest code releas
- Optimal MPI-OpenMP configuration (figures)
- Memory usage (figure & table)

Evolution of execution time with code version



Benchmark database

Benchmark: sedov on meluxina

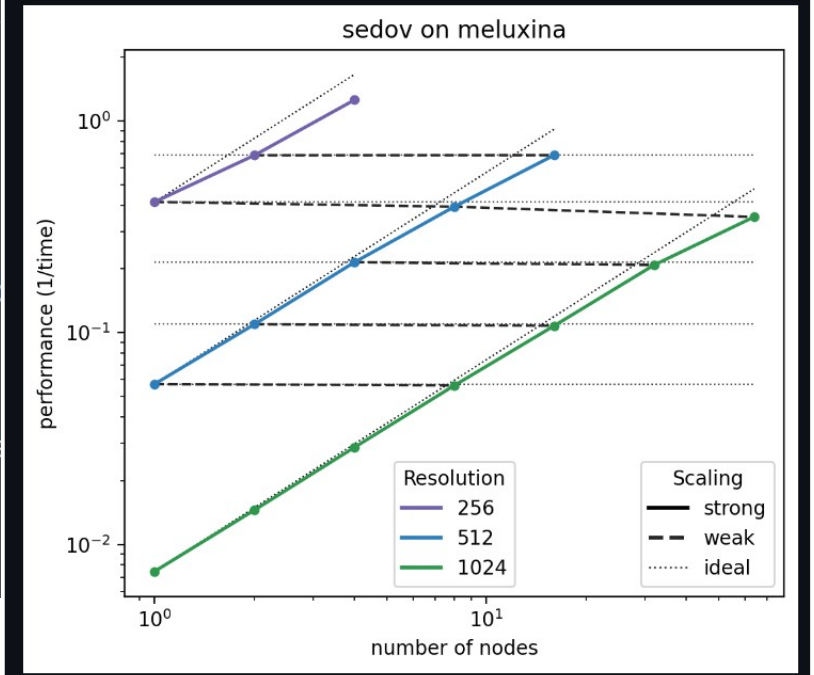
Benchmark description: sedov

Cluster info: meluxina

On this page:

- Evolution of execution time with code version (figure & table with speed
- Strong scaling evolution (figure & table with efficiency)
- [if unigrid] Weak scaling evolution (figure & table with efficiency)
- [if unigrid] Combination of strong and weak scaling of the latest code re
- Optimal MPI-OpenMP configuration (figures)
- Memory usage (figure & table)

Strong and weak scaling of the latest code release (2026-05)



Benchmark database

Benchmark: sedov on meluxina

Benchmark description: sedov

Cluster info: meluxina

On this page:

- Evolution of execution time with code version (figure & table w
- Strong scaling evolution (figure & table with efficiency)
- [if unigrid] Weak scaling evolution (figure & table with efficienc
- [if unigrid] Combination of strong and weak scaling of the late
- Optimal MPI-OpenMP configuration (figures)
- Memory usage (figure & table)

nodes	2024-10	2025-05	2025-10	2026-05	openmp (beta)
1	1.000	1.000	1.000	1.000	1.000
2	0.986	0.974	0.972	0.974	0.980
4	0.970	0.958	0.958	0.963	0.966
8	0.954	0.943	0.947	0.944	0.952
16	0.922	0.919	0.906	0.904	0.922 (MPI=64 OMP=2)
32	0.902	0.886	0.882	0.875	0.888
64	0.822	0.805	0.776	0.737	0.739 (MPI=64 OMP=2)

Benchmark database

Benchmark: sedov on meluxina

Benchmark description: sedov

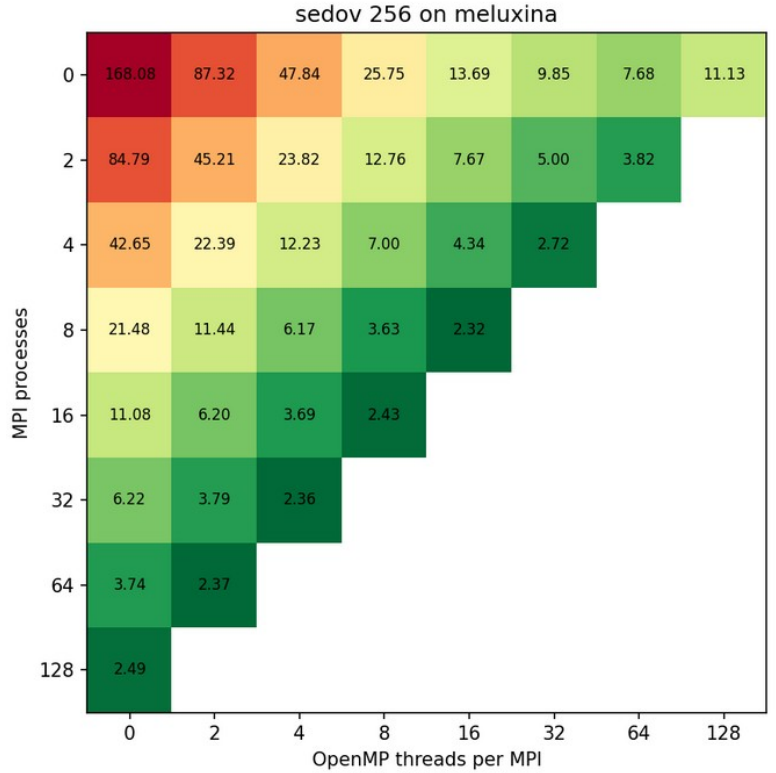
Cluster info: meluxina

On this page:

- Evolution of execution time with code version (figure & table with spee
- Strong scaling evolution (figure & table with efficiency)
- [if unigrid] Weak scaling evolution (figure & table with efficiency)
- [if unigrid] Combination of strong and weak scaling of the latest code r
- Optimal MPI-OpenMP configuration (figures)
- Memory usage (figure & table)

OpenMP configuration guidelines

The figures below give inside in the behaviour of OpenMP for different resolutions of this setup. Shows is which MPI - OpenMP configuration is most optimal in terms of execution time. The data is for the latest OpenMP version (commit e9846974 on branch openmp) , corresponding to the version used for the previous figures.



Benchmark database

Benchmark: sedov on meluxina

Benchmark description: sedov

Cluster info: meluxina

On this page:

- Evolution of execution time with cod
- Strong scaling evolution (figure & tab
- [if unigrid] Weak scaling evolution (fi
- [if unigrid] Combination of strong an
- Optimal MPI-OpenMP configuration
- Memory usage (figure & table)

nodes	MPI-only	OMP=2	OMP=4	OMP=8	OMP=16	OMP=32	OMP=64	best
1	207.74 GB	198.98 GB	195.07 GB	192.91 GB	191.70 GB	191.00 GB	190.53 GB	91.7 %
2	116.62 GB	104.51 GB	99.71 GB	97.66 GB	96.85 GB	-	-	83.0 %
4	67.35 GB	58.27 GB	52.29 GB	50.02 GB	49.22 GB	-	-	73.1 %
8	43.55 GB	33.82 GB	29.07 GB	26.29 GB	25.02 GB	-	-	57.4 %
16	32.88 GB	21.89 GB	16.86 GB	14.63 GB	13.13 GB	-	-	39.9 %
32	29.59 GB	16.56 GB	10.94 GB	8.51 GB	7.34 GB	-	-	24.8 %
64	33.18 GB	14.89 GB	8.25 GB	5.50 GB	4.29 GB	-	-	12.9 %

Doing your own benchmarks

https://github.com/tinecolman/ramses-benchmarks-prototype2

```
./launch_benchmark_suite.sh -c meluxina -b "cosmo" -r "1024" -n "1 2 4" -m "0 4" -i 3
```

```
################################################
#    Launching RAMSES performance benchmark    #
################################################
Repository url: https://github.com/tinecolman/ramses
Branch: openmp
Commit hash: e9846974
Benchmark: cosmo
Compile job submitted with Job ID: 4728505
Waiting for compile job to finish...
Compile job completed successfully.
Launched cosmo 1024 on 1 nodes with 128 procs/node and 0 threads/proc [JOB ID 4728529]
Launched cosmo 1024 on 1 nodes with 128 procs/node and 0 threads/proc [JOB ID 4728530]
Launched cosmo 1024 on 1 nodes with 128 procs/node and 0 threads/proc [JOB ID 4728531]
Launched cosmo 1024 on 1 nodes with 32 procs/node and 4 threads/proc [JOB ID 4728532]
Launched cosmo 1024 on 1 nodes with 32 procs/node and 4 threads/proc [JOB ID 4728533]
Launched cosmo 1024 on 1 nodes with 32 procs/node and 4 threads/proc [JOB ID 4728534]
Launched cosmo 1024 on 2 nodes with 128 procs/node and 0 threads/proc [JOB ID 4728535]
Launched cosmo 1024 on 2 nodes with 128 procs/node and 0 threads/proc [JOB ID 4728536]
```

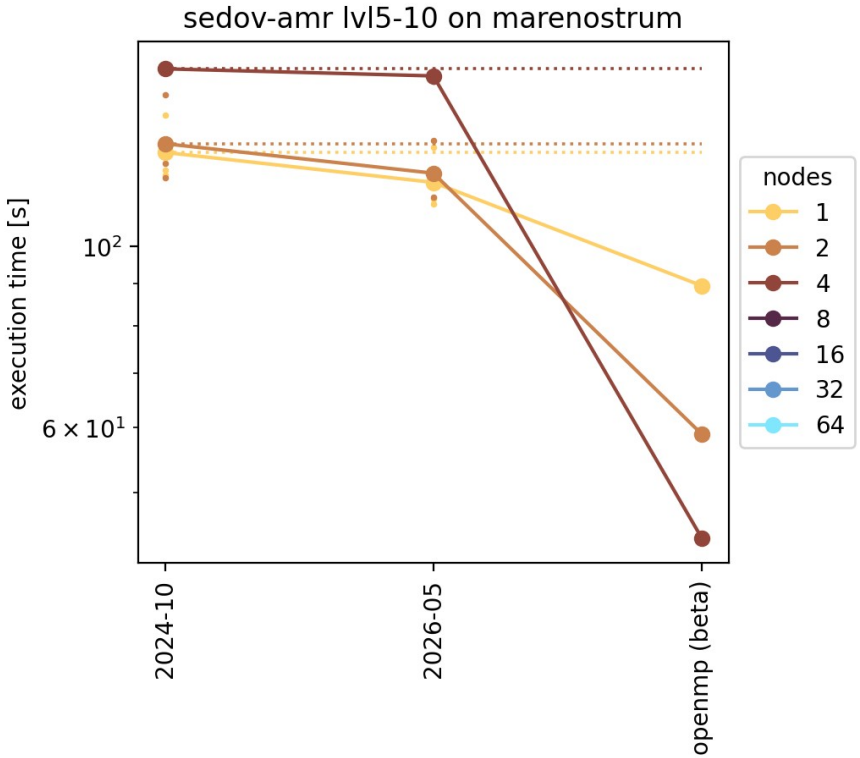
Summary

RAMSES release 2026.05,
including SPACE optimizations

OpenMP implementation

- DONE: all 3 SPACE use-cases
- TODO: stars & sinks

Benchmark database & scripts



Acknowledgement & Disclaimer

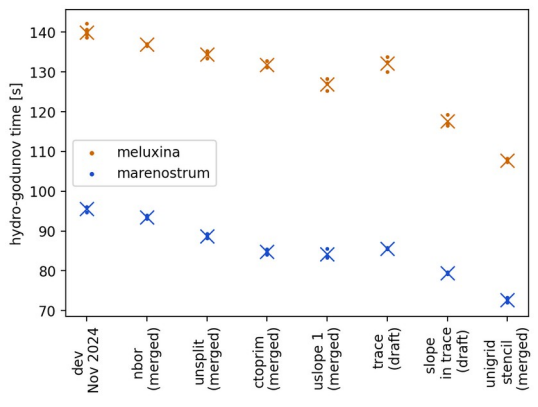
Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant agreement No 101093441.

Release 2026.05 highlights

Hydro solver optimization

Vectorization & memory access

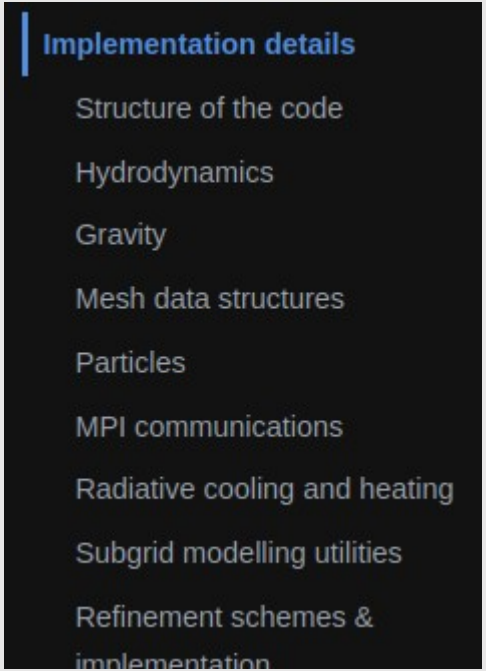
10 – 25% speedup



CI/CD improvements

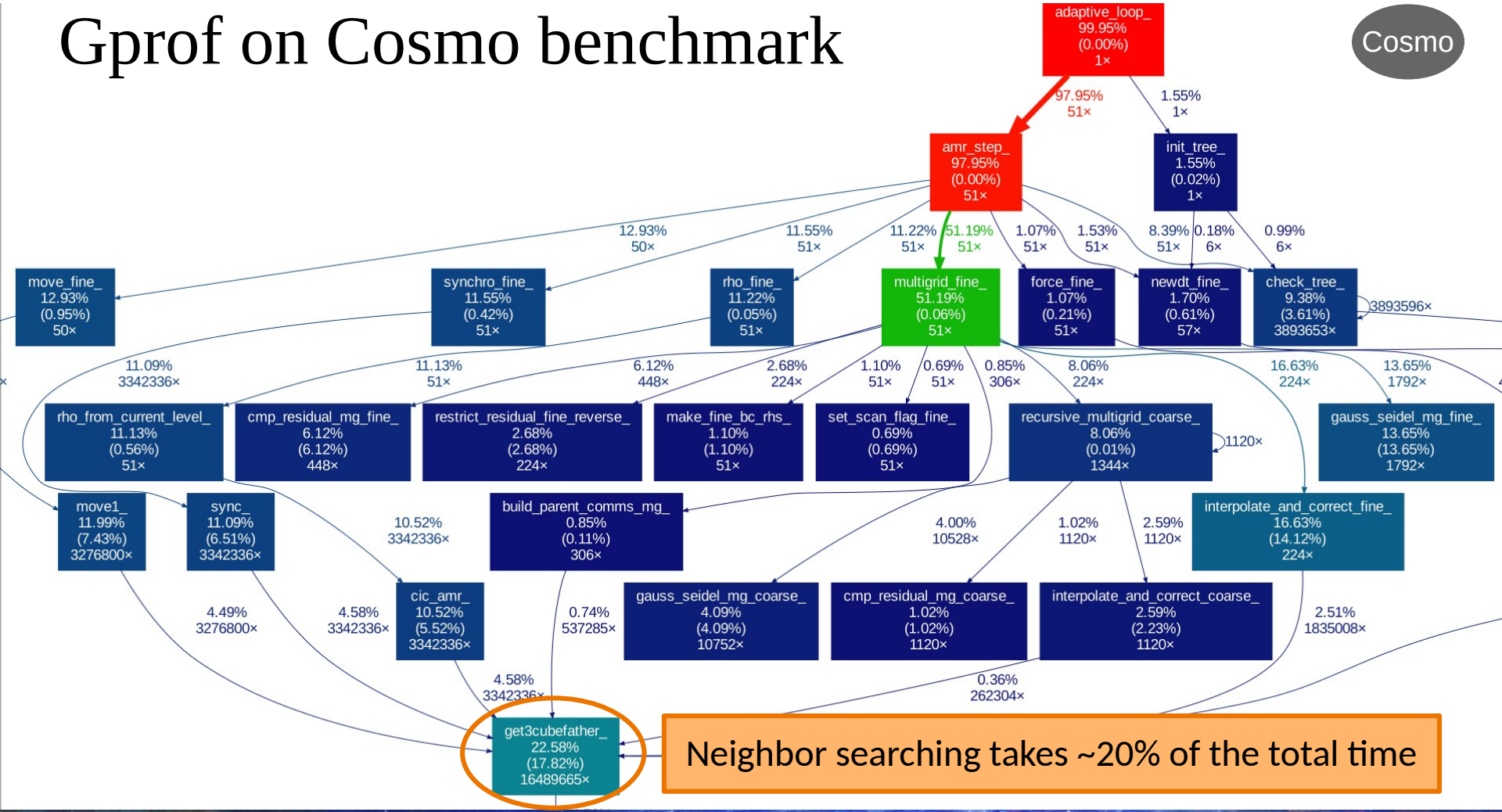
- Many tests added or updated
- Improvements to the infrastructure
- Coverage 47 → 64%

Developer documentation



Gprof on Cosmo benchmark

Cosmo



Neighbor searching takes ~20% of the total time

Neighbor searching

Elementary routine used in many parts of the code

Issue 1: routine determines two results (cells and grids), but only one is used

Issue 2: filtering to change to order in which cells are processed has non-contiguous memory access (and is applied trice!)

=> refactor for efficiency

```
  5  subroutine get3cubefather(ind_cell_father,nbors_father_cells,&
142    else    ! else, more complicated...
153      ! Loop over position
154      do ind=1,twotondim
155
156        ! Select father cells that sit at position ind
157        iok=0
158        do i=1,ncell
159          if(pos(i)==ind)then
160            iok=iok+1
161            ind_grid_ok(iok)=ind_grid_father(i)
162          end if
163        end do
164
165        if(iok>0)&
166        & call get3cubepos(ind_grid_ok,ind,nbors_father_ok,nbors_grids_ok,iok)
167
168        ! Store neighboring father cells for selected cells
169        do j=1,threetondim
170          iok=0
171          do i=1,ncell
172            if(pos(i)==ind)then
173              iok=iok+1
174              nbors_father_cells(i,j)=nbors_father_ok(iok,j)
175            end if
176          end do
177        end do
178
179        ! Store neighboring father grids for selected cells
180        do j=1,twotondim
181          iok=0
182          do i=1,ncell
183            if(pos(i)==ind)then
184              iok=iok+1
185              nbors_father_grids(i,j)=nbors_grids_ok(iok,j)
186            end if
187          end do
188        end do
```

Neighbor searching

Solution 1: split routine into two parts

Solution 2: rewrite to remove if-statements

cosmo profiling Name	Inclusive Time w.r.t. Wall Time(s) old	new
gauss_seidel_mg_fine	15.31	15.17
interpolate_and_correct_fine	12.08	12.15
move1	9.89	9.75
sync	9.32	9.60
mca_btl_vader_poll_handle_frag	9.32	7.83
get3cubefather	17.86	0.28
cic_amr	6.92	6.64
cmp_residual_mg_fine	6.23	6.15
gauss_seidel_mg_coarse	4.16	4.03
check_tree	4.19	3.97
get3cubepos	3.85	3.79

```
48   subroutine get3cubefather(ind_cell,nbor_cells,ncell,ilevel)
126    else    ! else, more complicated...
127
128      ! Get the cell's position in its grid, that is the
129      ! index ind, between 1 and twotondim.
130      do i=1,ncell      Tine Colman, 12 months ago • [SPACE] Optimize and docume
131        pos(i)=(ind_cell(i)-ncoarse-1)/ngridmax+1  !integer devision
132      end do
133
134      ! Convert the cell's index to the index of the grid to which the cell belo
135      do i=1,ncell
136        iskip=ncoarse+(pos(i)-1)*ngridmax
137        ind_grid_father(i)=ind_cell(i)-iskip
138      end do
139
140      ! Using the grid index and cell position, get the neighbor cells
141      call get3cubepos(ind_grid_father,pos,nbor_cells,ncell)
```

```
150  subroutine get3cubefather_grids(ind_cell,nbors_grids,ncell,ilevel)
223    else    ! else, more complicated...
225      ! Get the cell's position in its grid, that is the
226      ! index ind, between 1 and twotondim.
227      do i=1,ncell
228        pos(i)=(ind_cell(i)-ncoarse-1)/ngridmax+1  !integer devision
229      end do
230
231      ! Convert the cell's index to the index of the grid to which the cell belo
232      do i=1,ncell
233        iskip=ncoarse+(pos(i)-1)*ngridmax
234        ind_grid_father(i)=ind_cell(i)-iskip
235      end do
236
237      ! Using the grid index and cell position to get neighbor grids
238      call get_grids_of_nbor_cells(ind_grid_father,pos,nbors_grids,ncell)
```

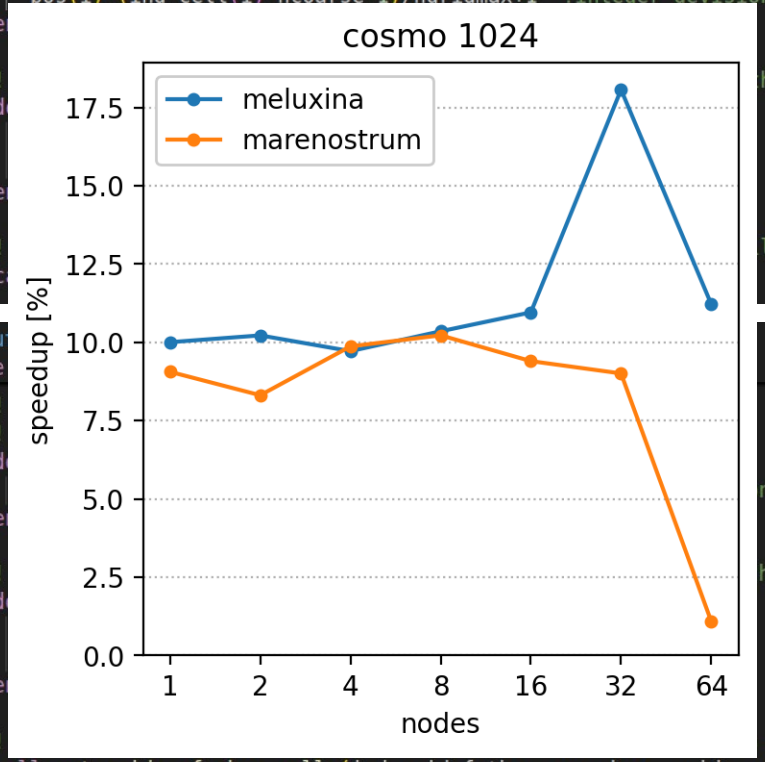
Neighbor searching

Solution 1: split routine into two parts

Solution 2: rewrite to remove if-statements

cosmo profiling Name	Inclusive Time w.r.t. Wall Time(s)	
	old	new
gauss_seidel_mg_fine	15.31	15.17
interpolate_and_correct_fine	12.08	12.15
move1	9.89	9.75
sync	9.32	9.60
mca_btl_vader_poll_handle_frag	9.32	7.83
get3cubefather	17.86	0.28
cic_amr	6.92	6.64
cmp_residual_mg_fine	6.23	6.15
gauss_seidel_mg_coarse	4.16	4.03
check_tree	4.19	3.97
get3cubepos	3.85	3.79

```
48   subroutine get3cubefather(ind_cell,nbor_cells,ncell,ilevel)
126    else    ! else, more complicated...
127
128      ! Get the cell's position in its grid, that is the
129      ! index ind, between 1 and twotondim.
130      do i=1,ncell      Tine Colman, 12 months ago • [SPACE] Optimize and docume
131        pos(i)=(ind_cell(i)-ncoarse-1)/ngridmax+1  !integer devision
132      e
133
134      !                                                    he cell belo
135      d
136
137
138      e
139
140      !                                                    ls
141      c
150  subrou
223    else
225      !
226      !
227      d
228
229      e                                                    n
230
231      !                                                    he cell belo
232      d
233
234
235      e
236
237      !
238      call get_grids_of_nbor_cells(ind_grid_father,pos,nbors_grids,ncell)
```



- **[OpenMP] dynamic scheduling for particle loops** ✓ openmp SPACE
  #251 by tinecolman Member was merged on Feb 24
- **Openmp misc** ✓ openmp SPACE
  #247 by tinecolman Member was merged on Feb 20
- **OpenMP for the galaxy benchmark** ✓ openmp SPACE
  #246 by tinecolman Member was merged on Feb 19
- **Openmp for MHD collapse test** ✓ openmp SPACE
  #244 by tinecolman Member was merged on Feb 18
- **OpenMP for RT (without refinement)** ✓ openmp SPACE
  #243 by tinecolman Member was merged on Feb 17
- **Openmp make tree fix** ✓ bug openmp SPACE
  #216 by tinecolman Member was merged on Nov 27, 2025
- **Openmp in flagging for refinement** ✓ openmp SPACE
  #215 by tinecolman Member was merged on Dec 11, 2025
- **Working version of OpenMP in particles for COSMO (part2)** ✓ openmp SPACE
  #213 by tinecolman Member was merged on Nov 20, 2025
- **adjust omp for set_uold/unew to allow for more than 8 threads** ✓ openmp
  #207 by tinecolman Member was merged on Oct 17, 2025
- **[OpenMP] Progress of openmp for cosmo box: poisson multigrid solver** ✓
  #200 by tinecolman Member was merged on Sep 30, 2025
- **Openmp virtual_boundaries** ✓ openmp SPACE
  #196 by tinecolman Member was merged on Sep 23, 2025

MPI + OpenMP

Shared memory parallelism inside nodes

=> reduce number of MPI domain

=> less time in MPI communications &

reduced memory imprint ghost zones

=> **improved scalability**

Starting from RAMSES-yOMP

Dedicated branch on RAMSES GitHub