HiPACE++: Run efficiently

Transverse resolution

Longitudinal resolution

Number of plasma particles

Number of ranks

Number of beam particles

Profiling

Alexander Sinn

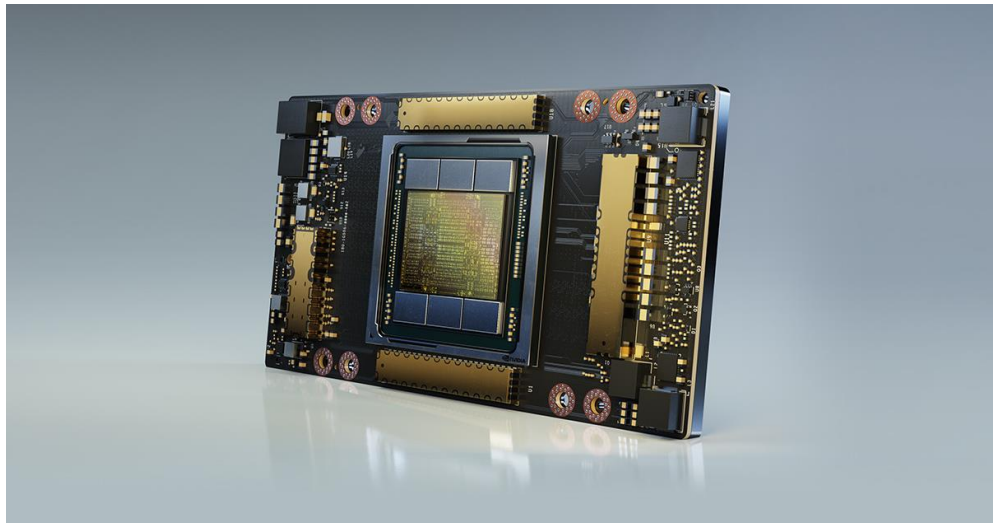
Hamburg, 11.07.2023

HELMHOLTZ

DESY.

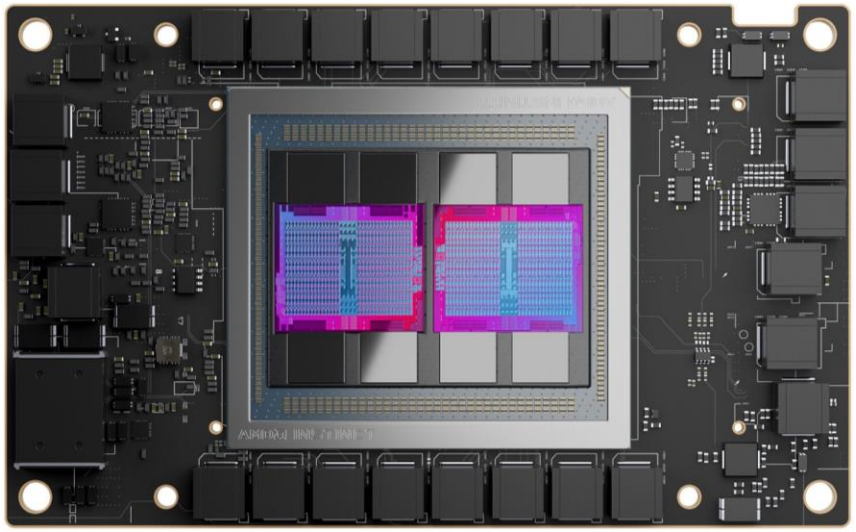
Common HPC platforms today:

NVIDIA A100 (40 GB/80 GB)



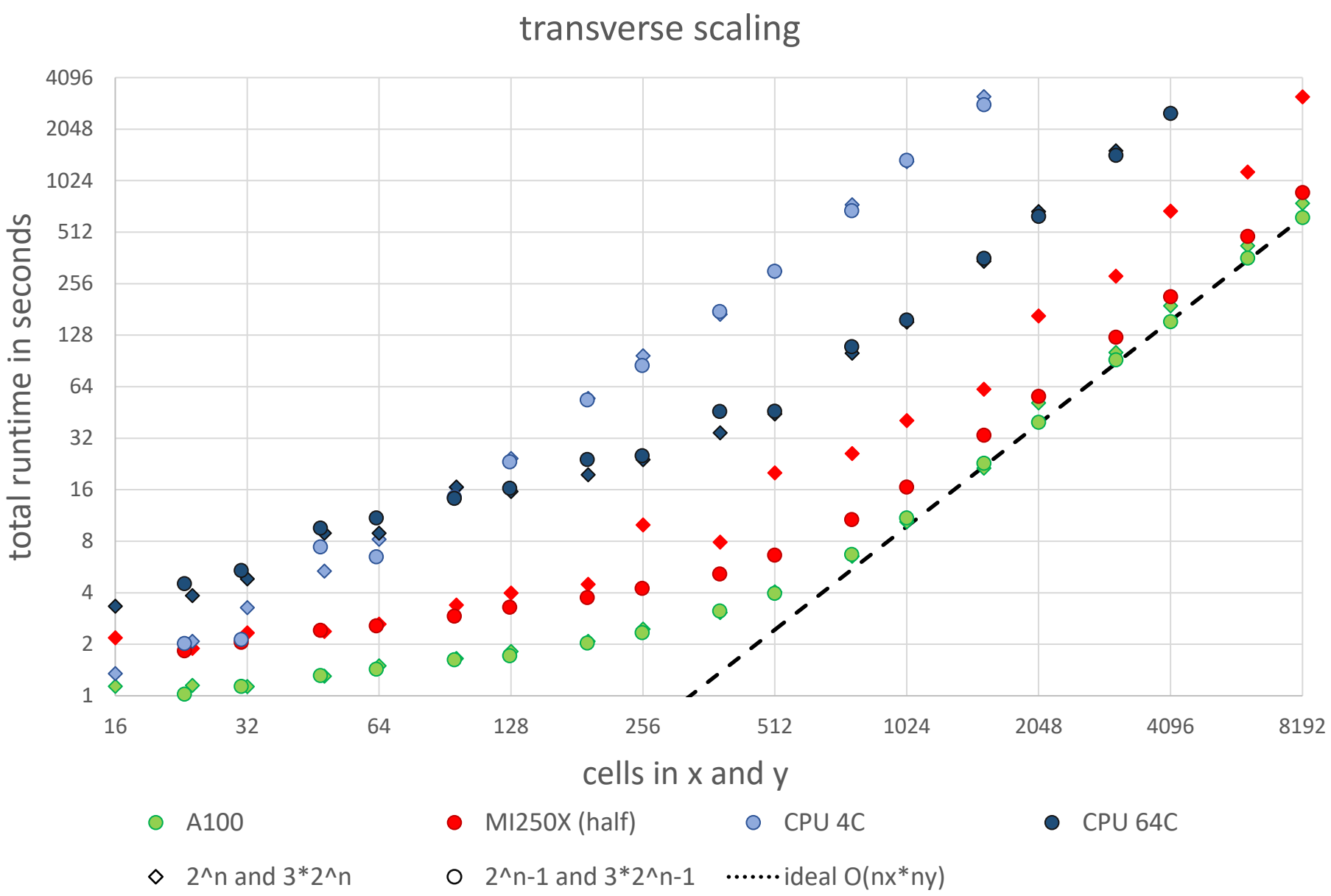
- **NERSC Perlmutter:**
  - 1536 × 4 × A100 40 GB
  - 256 × 4 × A100 80 GB
- **JUWELS Booster Module:**
  - 936 × 4 × A100 40 GB
- **DESY Maxwell (mpa):**
  - 8 × 4 × A100 40 GB
  - 1 × 4 × A100 80 GB

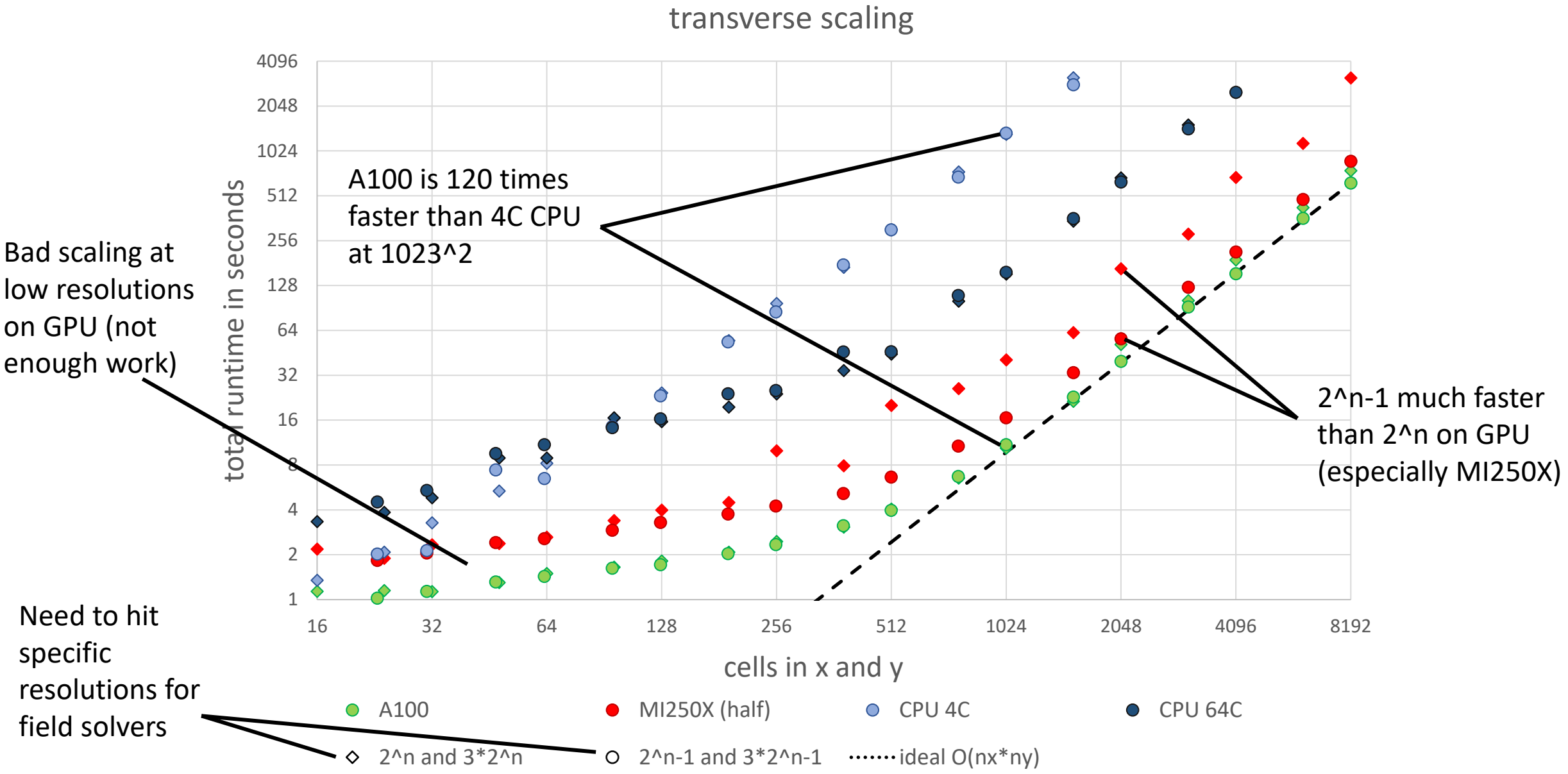
AMD MI250X (2 × 64 GB)



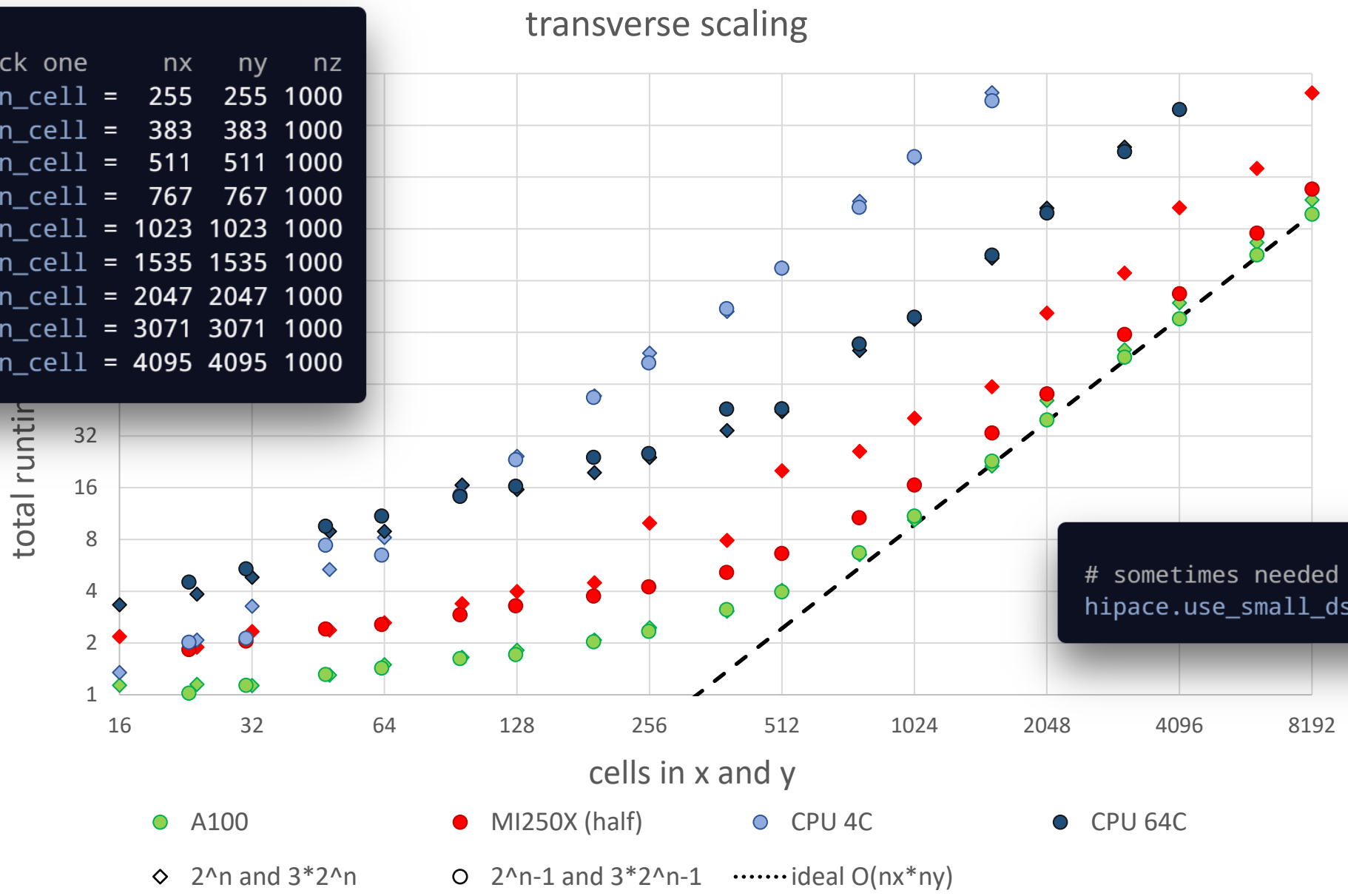
- **OLCF Frontier:**
  - 9472 × 4 × MI250X
- **CSC LUMI:**
  - 2560 × 4 × MI250X
- **CINES Adastra:**
  - 338 × 4 × MI250X

HiPACE++ supports AMD/APPLE/INTEL CPUs and AMD/NVIDIA GPUs

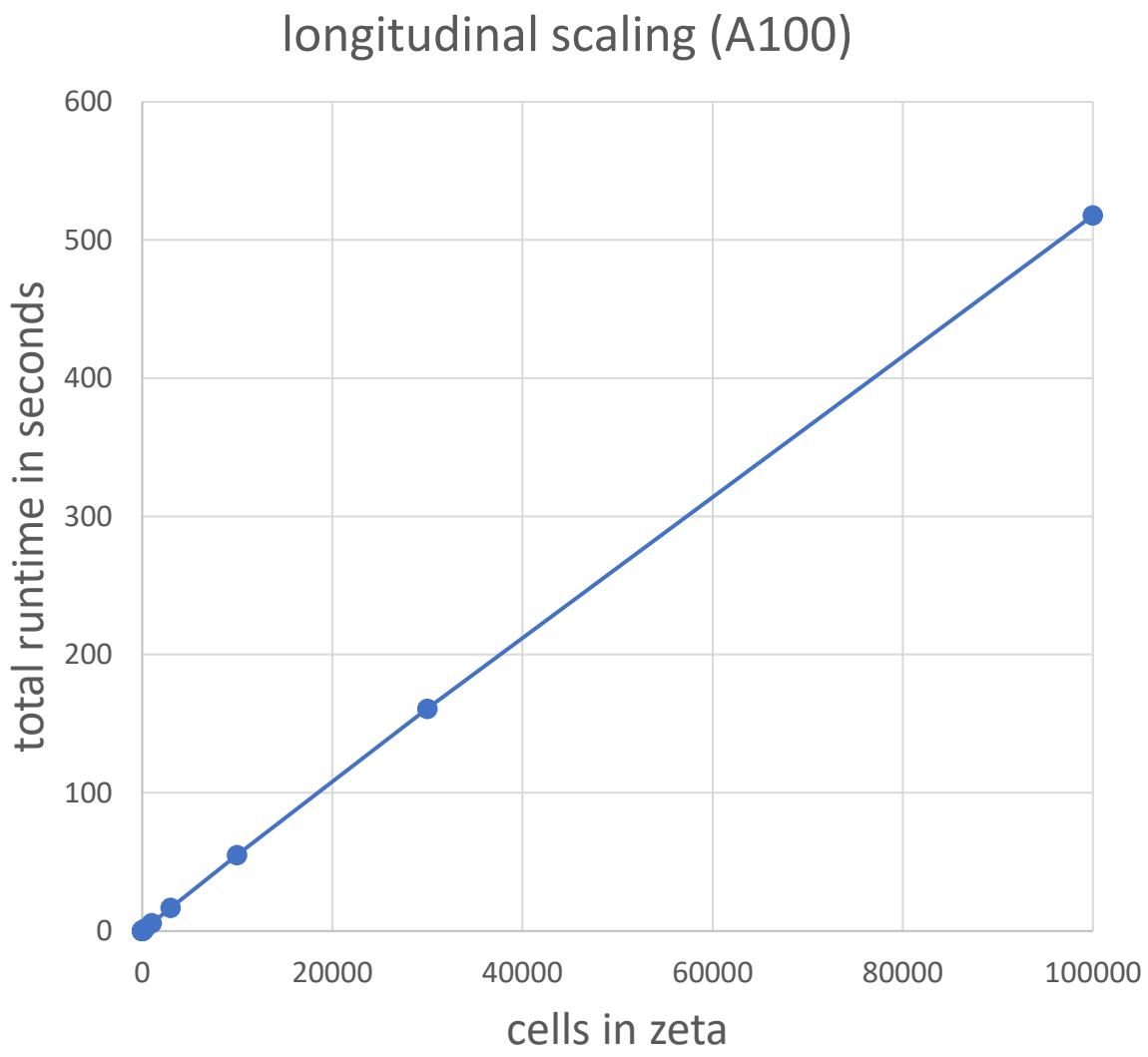

transverse scaling
total runtime in seconds
4096
2048
1024
512
256
128
64
32
16
8
4
2
1
16
32
64
128
256
512
1024
2048
4096
8192
cells in x and y
A100
MI250X (half)
CPU 4C
CPU 64C
2^n and 3*2^n
2^n-1 and 3*2^n-1
ideal O(nx*ny)

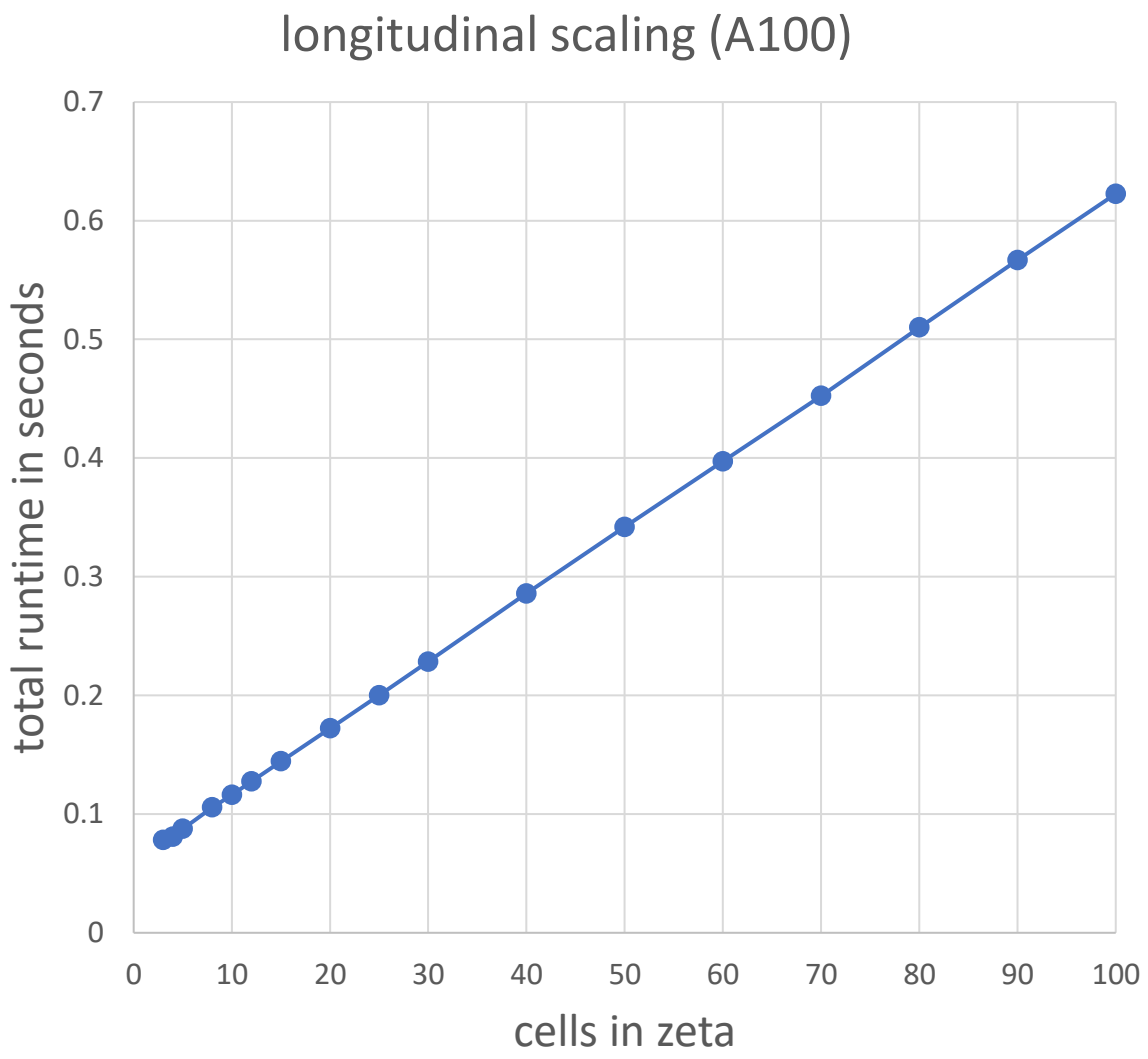

transverse scaling
total runtime in seconds
4096
2048
1024
512
256
128
64
32
16
8
4
2
1
16
32
64
128
256
512
1024
2048
4096
8192
cells in x and y
A100 is 120 times faster than 4C CPU at 1023^2
Bad scaling at low resolutions on GPU (not enough work)
2^n-1 much faster than 2^n on GPU (especially MI250X)
Need to hit specific resolutions for field solvers
A100
MI250X (half)
CPU 4C
CPU 64C
2^n and 3*2^n
2^n-1 and 3*2^n-1
ideal O(nx*ny)

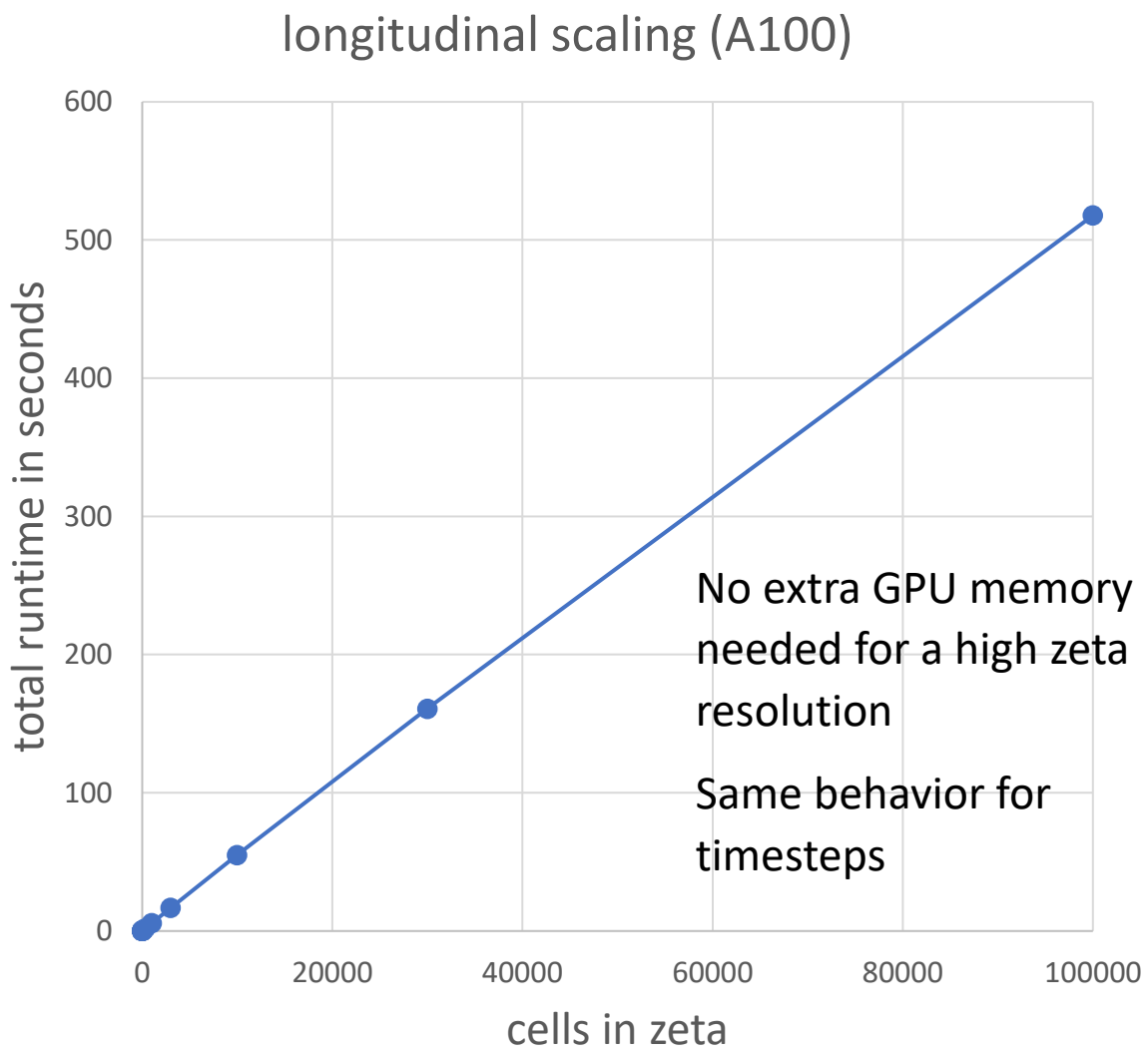
```
# pick one      nx   ny   nz
amr.n_cell =  255  255 1000
amr.n_cell =  383  383 1000
amr.n_cell =  511  511 1000
amr.n_cell =  767  767 1000
amr.n_cell = 1023 1023 1000
amr.n_cell = 1535 1535 1000
amr.n_cell = 2047 2047 1000
amr.n_cell = 3071 3071 1000
amr.n_cell = 4095 4095 1000
```

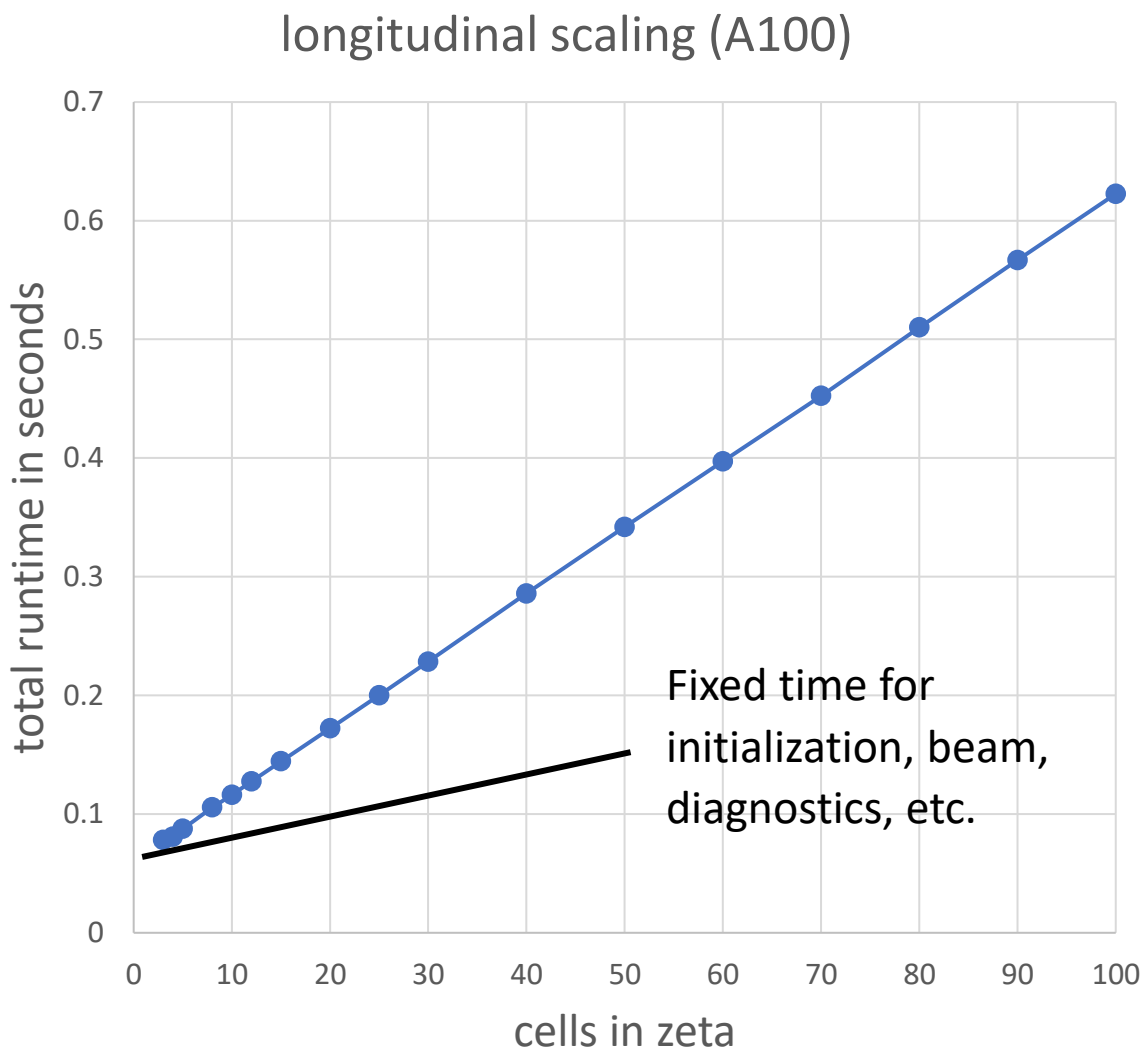


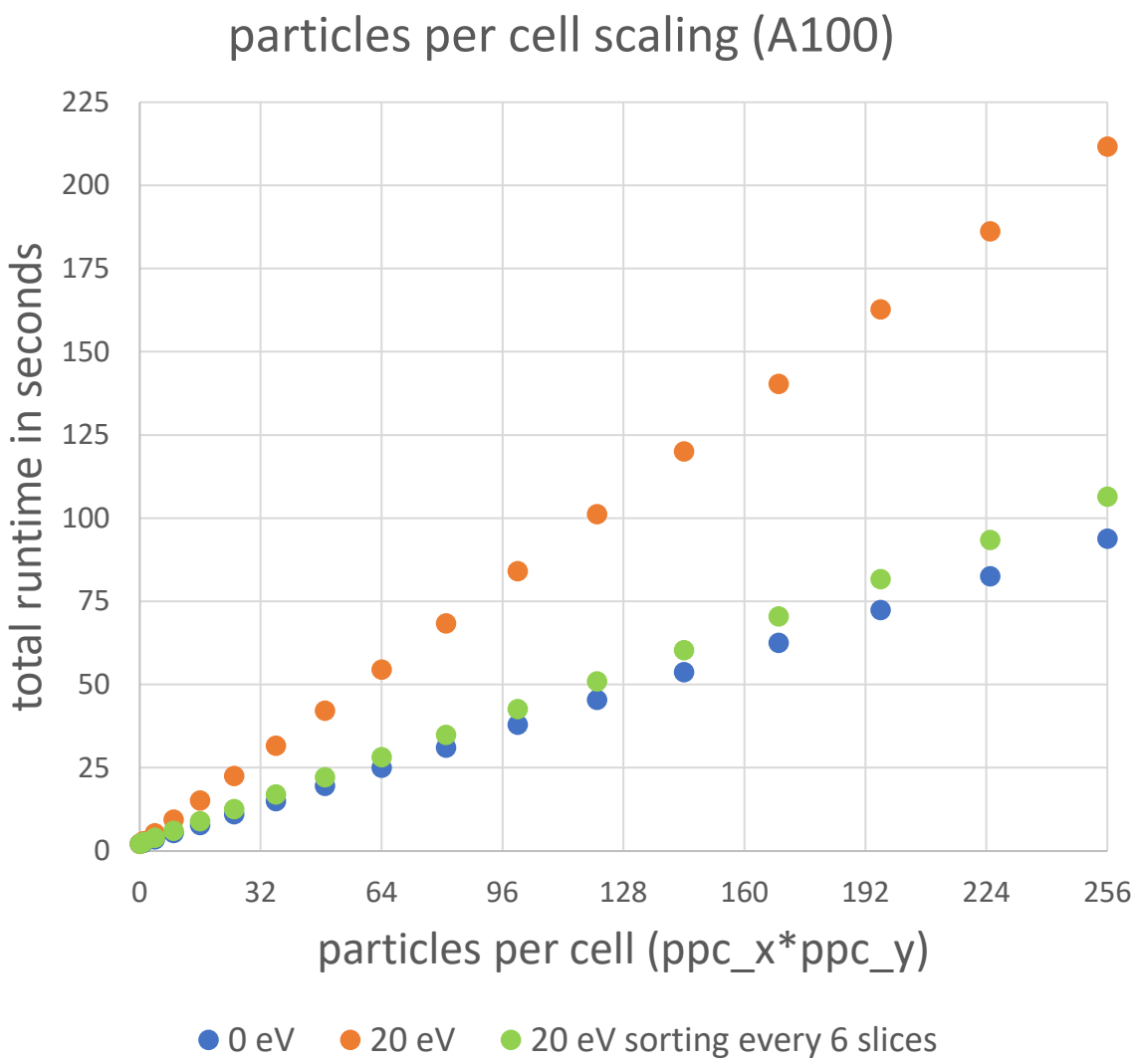

```
# sometimes needed on MI250X
hipace.use_small_dst = true
```

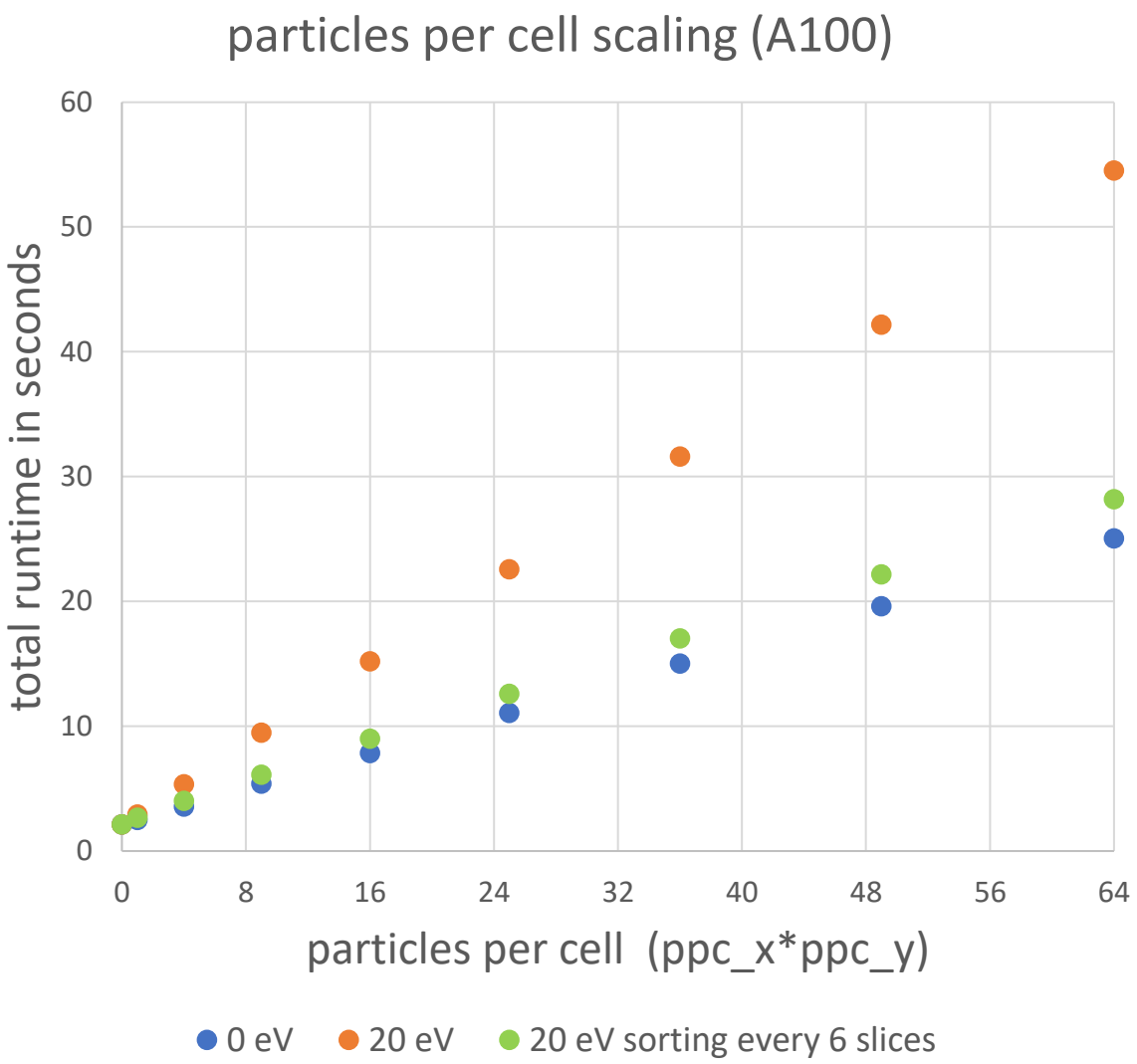

longitudinal scaling (A100)
total runtime in seconds
cells in zeta
0
100
200
300
400
500
600
0
20000
40000
60000
80000
100000

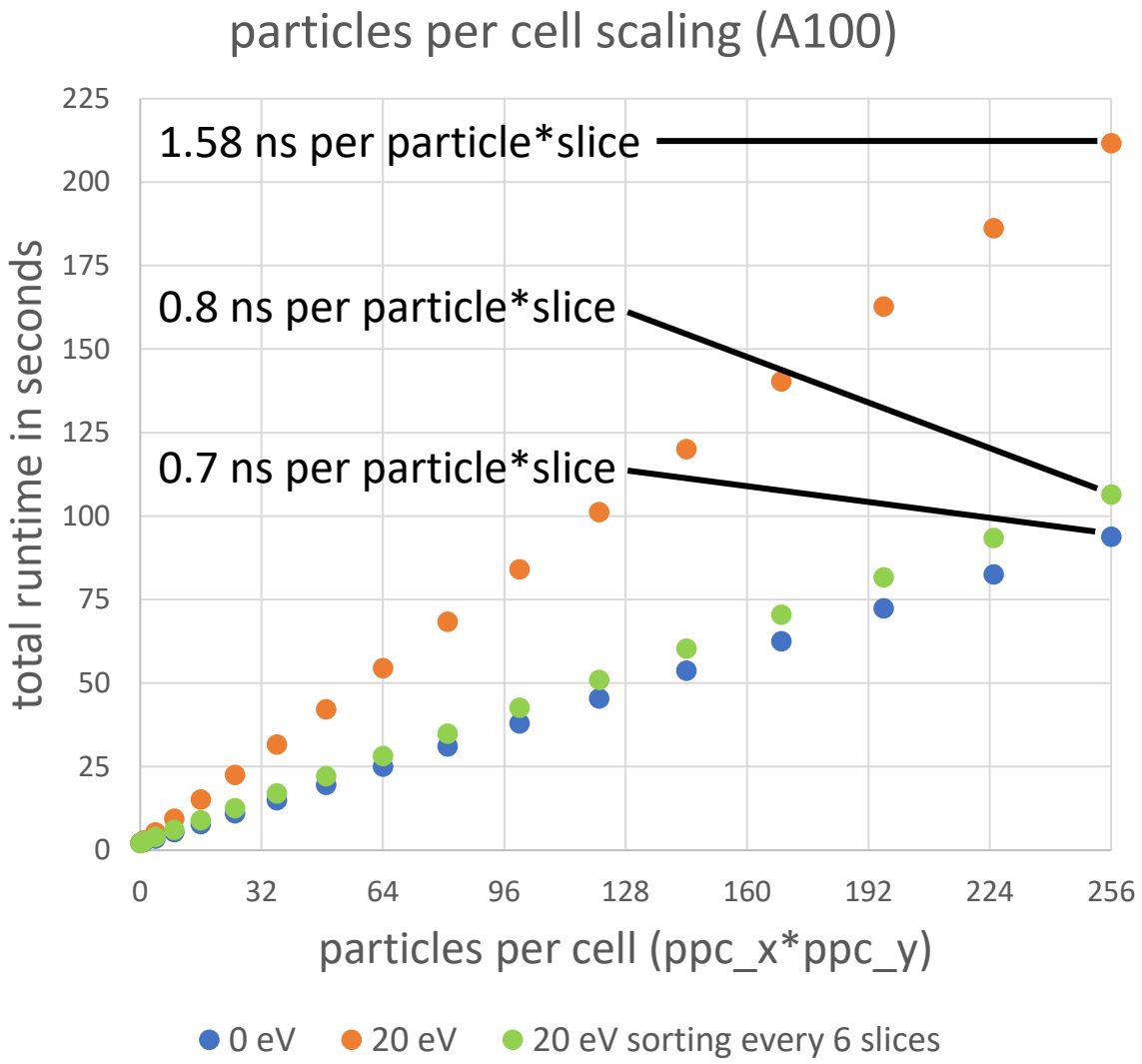

longitudinal scaling (A100)
total runtime in seconds
cells in zeta
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50
60
70
80
90
100

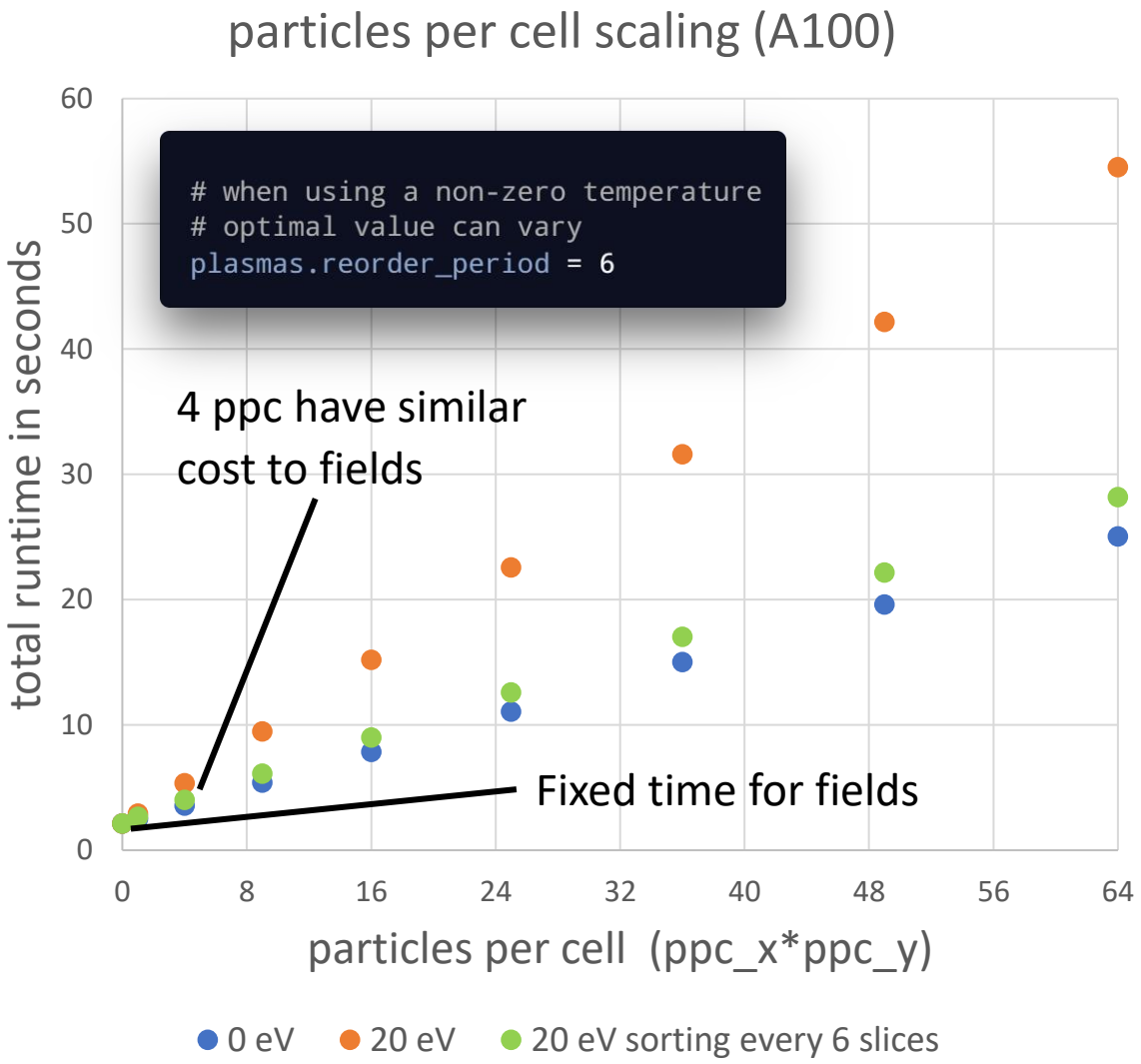

longitudinal scaling (A100)
total runtime in seconds
cells in zeta
0
100
200
300
400
500
600
0
20000
40000
60000
80000
100000
No extra GPU memory needed for a high zeta resolution
Same behavior for timesteps

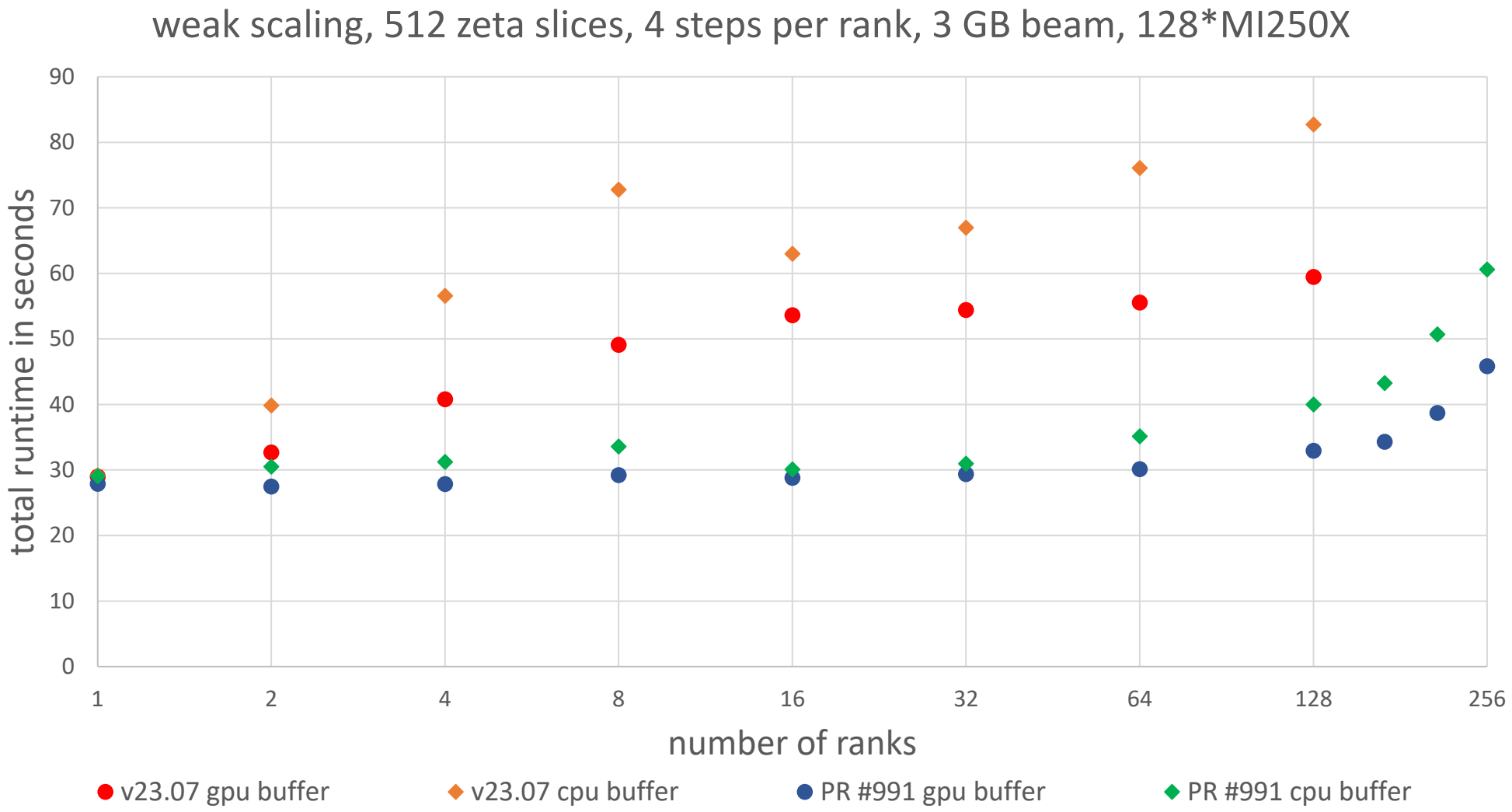

longitudinal scaling (A100)
total runtime in seconds
cells in zeta
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50
60
70
80
90
100
Fixed time for initialization, beam, diagnostics, etc.

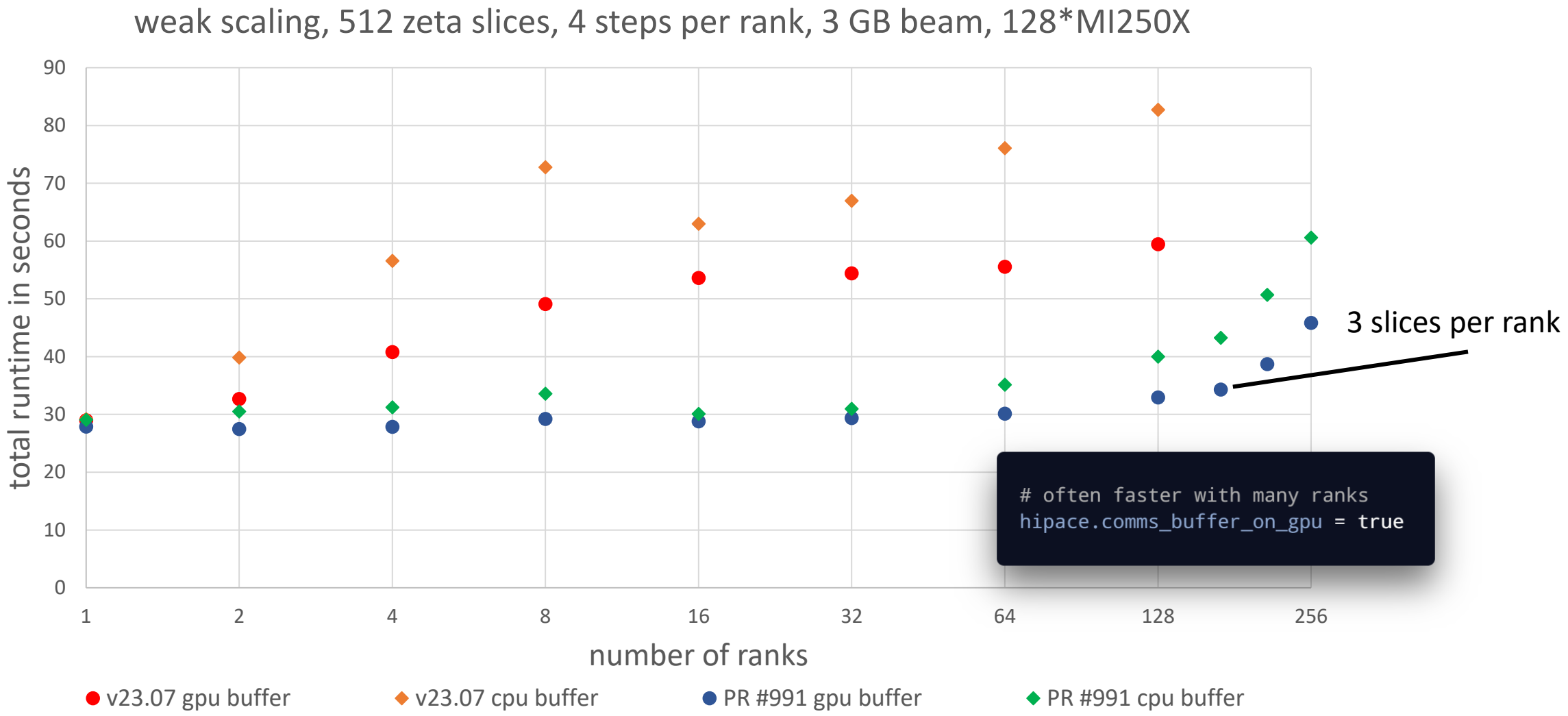

particles per cell scaling (A100)
total runtime in seconds
225
200
175
150
125
100
75
50
25
0
0
32
64
96
128
160
192
224
256
particles per cell (ppc_x*ppc_y)
0 eV
20 eV
20 eV sorting every 6 slices

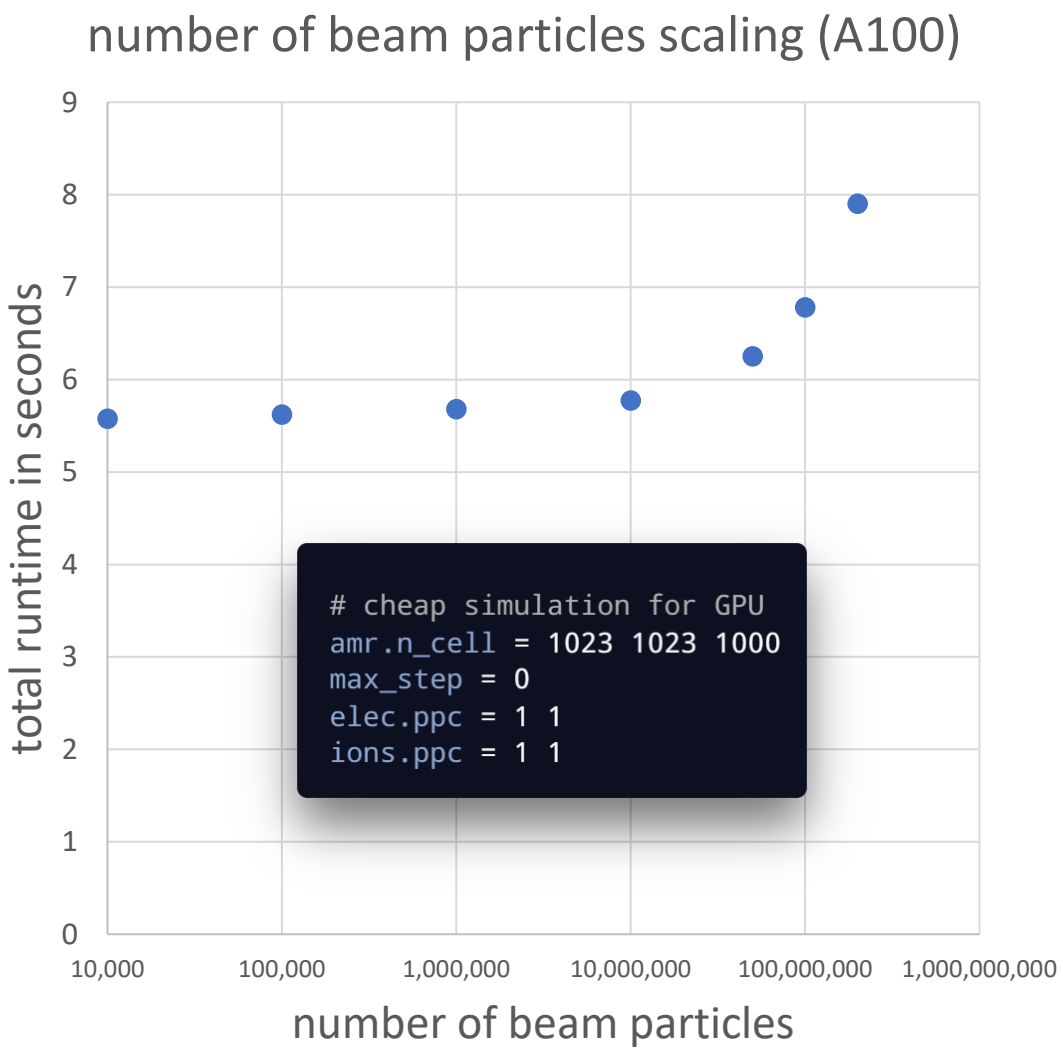

particles per cell scaling (A100)
total runtime in seconds
60
50
40
30
20
10
0
0
8
16
24
32
40
48
56
64
particles per cell (ppc_x*ppc_y)
0 eV
20 eV
20 eV sorting every 6 slices


particles per cell scaling (A100)
1.58 ns per particle*slice
0.8 ns per particle*slice
0.7 ns per particle*slice
total runtime in seconds
225
200
175
150
125
100
75
50
25
0
0
32
64
96
128
160
192
224
256
particles per cell (ppc_x*ppc_y)
0 eV
20 eV
20 eV sorting every 6 slices

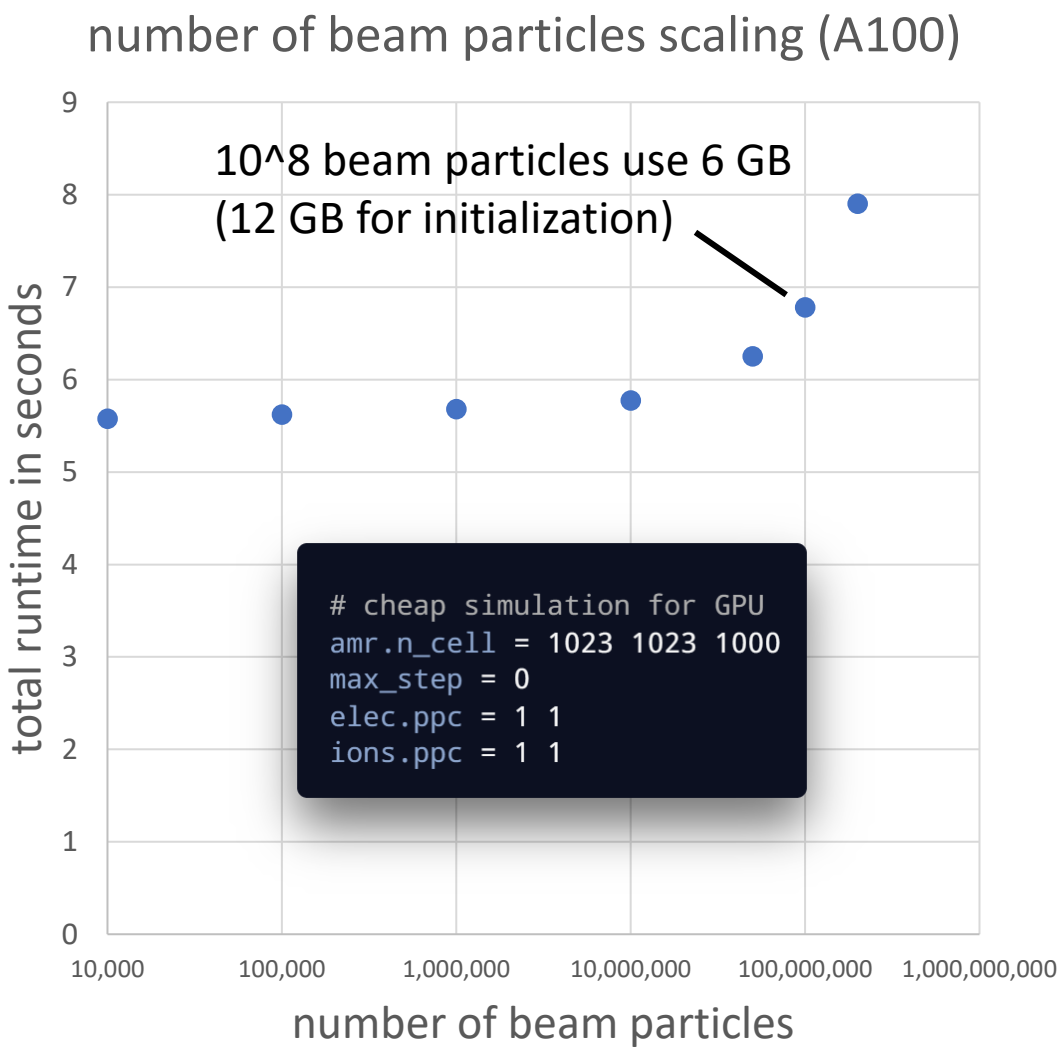

particles per cell scaling (A100)
# when using a non-zero temperature
# optimal value can vary
plasmas.reorder_period = 6
4 ppc have similar
cost to fields
Fixed time for fields
total runtime in seconds
60
50
40
30
20
10
0
0
8
16
24
32
40
48
56
64
particles per cell (ppc_x*ppc_y)
0 eV
20 eV
20 eV sorting every 6 slices


weak scaling, 512 zeta slices, 4 steps per rank, 3 GB beam, 128*MI250X
total runtime in seconds
0
10
20
30
40
50
60
70
80
90
1
2
4
8
16
32
64
128
256
number of ranks
v23.07 gpu buffer
v23.07 cpu buffer
PR #991 gpu buffer
PR #991 cpu buffer


weak scaling, 512 zeta slices, 4 steps per rank, 3 GB beam, 128*MI250X
total runtime in seconds
90
80
70
60
50
40
30
20
10
0
1
2
4
8
16
32
64
128
256
number of ranks
3 slices per rank
# often faster with many ranks
hipace.comms_buffer_on_gpu = true
v23.07 gpu buffer
v23.07 cpu buffer
PR #991 gpu buffer
PR #991 cpu buffer




```
# cheap simulation for GPU
amr.n_cell = 1023 1023 1000
max_step = 0
elec.ppc = 1 1
ions.ppc = 1 1
```




```
# cheap simulation for GPU
amr.n_cell = 1023 1023 1000
max_step = 0
elec.ppc = 1 1
ions.ppc = 1 1
```

```
# prefer fixed_weight over fixed_ppc beams
beam.injection_type = fixed_weight
beam.profile = gaussian

# use 10^6 beam particles minimum on GPU,
# consider 10^7 or 10^8
beam.num_particles = 1000000

# disable full beam diagnostic
diagnostic.beam_output_period = 0
# use in-situ diagnostic instead
beam.insitu_period = 1

# sometimes increasing both dt and the number
# of subcycles gives the same result faster
beam.n_subcycles = 100
```

The number of beam particles is not limited by computational performance.

Instead, it is limited by GPU memory, max int, multi-rank communication performance and diagnostic output size.

Profiling

```
# Needed for accurate pre-function profiling on GPU
hipace.do_device_synchronize = true
```

Total time, exclusive and inclusive per-function timings and per-function memory usage (GPU only)

```
TinyProfiler total time across processes [min...avg...max]: 53.13 ... 53.13 ... 53.13

-----------------------------------------------------------------------------------------------
Name                                                NCalls  Excl. Min  Excl. Avg  Excl. Max   Max %
-----------------------------------------------------------------------------------------------
hpmg::MultiGrid::solve1()                             2000      20.41      20.41      20.41  38.42%
ExplicitDeposition()                                  4000      9.266      9.266      9.266  17.44%
AdvancePlasmaParticles()                              4000      7.141      7.141      7.141  13.44%
DepositCurrent_PlasmaParticleContainer()              4000      6.766      6.766      6.766  12.74%
AnyDST::Execute()                                    12000       5.01       5.01       5.01   9.43%
FFTPoissonSolverDirichlet::SolvePoissonEquation()     6000     0.8344     0.8344     0.8344   1.57%
Fields::InitializeSlices()                            2000     0.5925     0.5925     0.5925   1.12%
AdvanceBeamParticlesSlice()                           2000     0.5879     0.5879     0.5879   1.11%


-----------------------------------------------------------------------------------------------
Name                                                NCalls  Incl. Min  Incl. Avg  Incl. Max   Max %
-----------------------------------------------------------------------------------------------
main()                                                   1      53.13      53.13      53.13 100.00%
Hipace::Evolve()                                         1      53.06      53.06      53.06  99.87%
Hipace::SolveOneSlice()                               2000      52.85      52.85      52.85  99.48%
Hipace::ExplicitMGSolveBxBy()                         2000      20.43      20.43      20.43  38.45%
hpmg::MultiGrid::solve1()                             2000      20.41      20.41      20.41  38.42%
ExplicitDeposition()                                  4000      9.266      9.266      9.266  17.44%
AdvancePlasmaParticles()                              4000      7.141      7.141      7.141  13.44%
Fields::SolvePoissonPsiExmByEypBxEzBz()               2000      6.887      6.887      6.887  12.96%
DepositCurrent_PlasmaParticleContainer()              4000      6.766      6.766      6.766  12.74%
FFTPoissonSolverDirichlet::SolvePoissonEquation()     6000      6.155      6.155      6.155  11.58%
AnyDST::Execute()                                    12000      5.321      5.321      5.321  10.01%


Device Memory Usage:
-------------------------------------------------------------------------------
Name                                        Nalloc  Nfree    AvgMem    MaxMem
-------------------------------------------------------------------------------
The_Arena::Initialize()                          1      1    12 MiB    29 GiB
sortBeamParticlesByBox()                        26     26  3090 MiB  6388 MiB
BeamParticleContainer::InitParticles()           9      9  7154 KiB  3097 MiB
PlasmaParticleContainer::InitParticles          42     42  1665 MiB  1703 MiB
Fields::AllocData()                              1      1   641 MiB   641 MiB
DenseBins<T>::buildGPU                          12     12   578 MiB   580 MiB
Hipace::ExplicitMGSolveBxBy()                   41     41   340 MiB   341 MiB
```

Profiling

```
# Needed for accurate pre-function profiling on GPU
hipace.do_device_synchronize = true
```

Total time, exclusive and inclusive per-function timings and per-function memory usage (GPU only)

Field solvers:

- hpmg::MultiGrid::solve1()
- AnyDST::Execute()

Plasma:

- ExplicitDeposition()
- DepositCurrent_PlasmaParticleContainer()
- AdvancePlasmaParticles()

Beam:

- AdvanceBeamParticlesSlice()
- DepositCurrentSlice_BeamParticleContainer()

```
TinyProfiler total time across processes [min...avg...max]: 53.13 ... 53.13 .. 53.13

-----------------------------------------------------------------------------------------------
Name                                                  NCalls  Excl. Min  Excl. Avg  Excl. Max   Max %
-----------------------------------------------------------------------------------------------
hpmg::MultiGrid::solve1()                               2000      20.41      20.41      20.41  38.42%
ExplicitDeposition()                                    4000      9.266      9.266      9.266  17.44%
AdvancePlasmaParticles()                                4000      7.141      7.141      7.141  13.44%
DepositCurrent_PlasmaParticleContainer()                4000      6.766      6.766      6.766  12.74%
AnyDST::Execute()                                      12000       5.01       5.01       5.01   9.43%
FFTPoissonSolverDirichlet::SolvePoissonEquation()       6000     0.8344     0.8344     0.8344   1.57%
Fields::InitializeSlices()                              2000     0.5925     0.5925     0.5925   1.12%
AdvanceBeamParticlesSlice()                             2000     0.5879     0.5879     0.5879   1.11%


-----------------------------------------------------------------------------------------------
Name                                                  NCalls  Incl. Min  Incl. Avg  Incl. Max   Max %
-----------------------------------------------------------------------------------------------
main()                                                     1      53.13      53.13      53.13 100.00%
Hipace::Evolve()                                           1      53.06      53.06      53.06  99.87%
Hipace::SolveOneSlice()                                 2000      52.85      52.85      52.85  99.48%
Hipace::ExplicitMGSolveBxBy()                           2000      20.43      20.43      20.43  38.45%
hpmg::MultiGrid::solve1()                               2000      20.41      20.41      20.41  38.42%
ExplicitDeposition()                                    4000      9.266      9.266      9.266  17.44%
AdvancePlasmaParticles()                                4000      7.141      7.141      7.141  13.44%
Fields::SolvePoissonPsiExmByEypBxEzBz()                 2000      6.887      6.887      6.887  12.96%
DepositCurrent_PlasmaParticleContainer()                4000      6.766      6.766      6.766  12.74%
FFTPoissonSolverDirichlet::SolvePoissonEquation()       6000      6.155      6.155      6.155  11.58%
AnyDST::Execute()                                      12000      5.321      5.321      5.321  10.01%


Device Memory Usage:
-----------------------------------------------------------------------------
Name                                        Nalloc  Nfree    AvgMem    MaxMem
-----------------------------------------------------------------------------
The_Arena::Initialize()                          1      1    12 MiB    29 GiB
sortBeamParticlesByBox()                        26     26  3090 MiB  6388 MiB
BeamParticleContainer::InitParticles()           9      9  7154 KiB  3097 MiB
PlasmaParticleContainer::InitParticles          42     42  1665 MiB  1703 MiB
Fields::AllocData()                              1      1   641 MiB   641 MiB
DenseBins<T>::buildGPU                          12     12   578 MiB   580 MiB
Hipace::ExplicitMGSolveBxBy()                   41     41   340 MiB   341 MiB
```

Questions?

```
# xy scaling
amr.n_cell = nxy nxy 2000

hipace.normalized_units = 1

amr.max_level = 0

max_step = 0
hipace.dt = 0
hipace.verbose = 3

hipace.depos_order_xy = 2

geometry.is_periodic = false  false  false  # Is periodic
geometry.prob_lo     = -8.   -8.   -6     # physical domain
geometry.prob_hi     =  8.    8.    6

beams.names = beam
beam.injection_type = fixed_weight
beam.profile = gaussian
beam.zmin = -5.9
beam.zmax = 5.9
beam.density = 3.
beam.u_mean = 0. 0. 2000
beam.u_std = 0. 0. 0.
beam.position_mean = 0 0 0
beam.position_std = 0.3 0.3 1.41
beam.num_particles = 10*1000

plasmas.names = elec ions
plasmas.neutralize_background = false

elec.density(x,y,z) = 1.
elec.ppc = 1 1
elec.element = electron

ions.density(x,y,z) = 1.
ions.ppc = 1 1
ions.element = proton

diagnostic.output_period = 0
diagnostic.diag_type = xz

my_constants.nxy = 1

hipace.do_device_synchronize = false
```

```
# zeta scaling
amr.n_cell = 1023 1023 nzeta

hipace.normalized_units = 1

amr.max_level = 0

max_step = 0
hipace.dt = 0
hipace.verbose = 3

hipace.depos_order_xy = 2

geometry.is_periodic = false  false  false  # Is periodic
geometry.prob_lo     = -8.   -8.   -6     # physical domain
geometry.prob_hi     =  8.    8.    6

beams.names = beam
beam.injection_type = fixed_weight
beam.profile = gaussian
beam.zmin = -5.9
beam.zmax = 5.9
beam.density = 3.
beam.u_mean = 0. 0. 2000
beam.u_std = 0. 0. 0.
beam.position_mean = 0 0 0
beam.position_std = 0.3 0.3 1.41
beam.num_particles = 1*1000*1000

plasmas.names = elec ions
plasmas.neutralize_background = false

elec.density(x,y,z) = 1.
elec.ppc = 1 1
elec.element = electron

ions.density(x,y,z) = 1.
ions.ppc = 1 1
ions.element = proton

diagnostic.output_period = 0
diagnostic.diag_type = xz

my_constants.nzeta = 1

hipace.do_device_synchronize = false
```

```
# ppc scaling
amr.n_cell = 1023 1023 500

hipace.normalized_units = 1

amr.max_level = 0

max_step = 0
hipace.dt = 0
hipace.verbose = 3

hipace.depos_order_xy = 2

geometry.is_periodic = false  false  false  # Is periodic
geometry.prob_lo     = -8.   -8.   -6     # physical domain
geometry.prob_hi     =  8.    8.    6

beams.names = beam
beam.injection_type = fixed_weight
beam.profile = gaussian
beam.zmin = -5.9
beam.zmax = 5.9
beam.density = 3.
beam.u_mean = 0. 0. 2000
beam.u_std = 0. 0. 0.
beam.position_mean = 0 0 0
beam.position_std = 0.3 0.3 1.41
beam.num_particles = 100*1000

plasmas.names = elec
plasmas.neutralize_background = true

elec.density(x,y,z) = 1.
elec.ppc = sqrt(nppc) sqrt(nppc)
elec.element = electron
elec.temperature_in_ev = 20

diagnostic.output_period = 0
diagnostic.diag_type = xz

my_constants.nppc = 0

hipace.do_device_synchronize = false

amrex.vector_growth_factor = 1.05

plasmas.reorder_period = 6
```

```
# rank scaling
amr.n_cell = 1023 1023 512

hipace.normalized_units = 1

amr.max_level = 0

max_step = nrank*4-1
hipace.dt = adaptive
hipace.verbose = 2

hipace.depos_order_xy = 2
hipace.use_small_dst = true
hipace.comms_buffer_on_gpu = true

geometry.is_periodic = false  false  false  # Is periodic
geometry.prob_lo     = -8.   -8.   -6     # physical domain
geometry.prob_hi     =  8.    8.    6

beams.names = beam
beam.injection_type = fixed_weight
beam.profile = gaussian
beam.zmin = -5.9
beam.zmax = 5.9
beam.density = 3.
beam.u_mean = 0. 0. 20000
beam.u_std = 0. 0. 0.
beam.position_mean = 0 0 0
beam.position_std = 0.3 0.3 1.41
beam.num_particles = 50*1000*1000

plasmas.names = elec
plasmas.neutralize_background = true

elec.density(x,y,z) = 1.
elec.ppc = 2 2
elec.element = electron

diagnostic.output_period = 0
diagnostic.diag_type = xz

hipace.do_device_synchronize = false

amrex.vector_growth_factor = 1.05

my_constants.nrank = 0
```

```
# beam scaling
amr.n_cell = 1023 1023 1000

hipace.normalized_units = 1

amr.max_level = 0

max_step = 0
hipace.dt = 0
hipace.verbose = 3

hipace.depos_order_xy = 2

geometry.is_periodic = false  false  false  # Is periodic
geometry.prob_lo     = -8.   -8.   -6     # physical domain
geometry.prob_hi     =  8.    8.    6

beams.names = beam
beam.injection_type = fixed_weight
beam.profile = gaussian
beam.zmin = -5.9
beam.zmax = 5.9
beam.density = 3.
beam.u_mean = 0. 0. 2000
beam.u_std = 0. 0. 0.
beam.position_mean = 0 0 0
beam.position_std = 0.3 0.3 1.41
beam.num_particles = nparbeam

plasmas.names = elec ions
plasmas.neutralize_background = false

elec.density(x,y,z) = 1.
elec.ppc = 1 1
elec.element = electron

ions.density(x,y,z) = 1.
ions.ppc = 1 1
ions.element = proton

diagnostic.output_period = 0
diagnostic.diag_type = xz

my_constants.nparbeam = 0

hipace.do_device_synchronize = false

amrex.vector_growth_factor = 1.05
```

```
# profile
amr.n_cell = 2047 2047 2000

hipace.normalized_units = 1

amr.max_level = 0

max_step = 0
hipace.dt = 0
hipace.verbose = 3

hipace.depos_order_xy = 2

geometry.is_periodic = false  false  false  # Is periodic
geometry.prob_lo     = -8.   -8.   -6     # physical domain
geometry.prob_hi     =  8.    8.    6

beams.names = beam
beam.injection_type = fixed_weight
beam.profile = gaussian
beam.zmin = -5.9
beam.zmax = 5.9
beam.density = 3.
beam.u_mean = 0. 0. 2000
beam.u_std = 0. 0. 0.
beam.position_mean = 0 0 0
beam.position_std = 0.3 0.3 1.41
beam.num_particles = 50000000

plasmas.names = elec ions
plasmas.neutralize_background = false

elec.density(x,y,z) = 1.
elec.ppc = 2 1
elec.element = electron

ions.density(x,y,z) = 1.
ions.ppc = 2 1
ions.element = proton

diagnostic.output_period = 0
diagnostic.diag_type = xz

hipace.do_device_synchronize = true

amrex.vector_growth_factor = 1.05
```