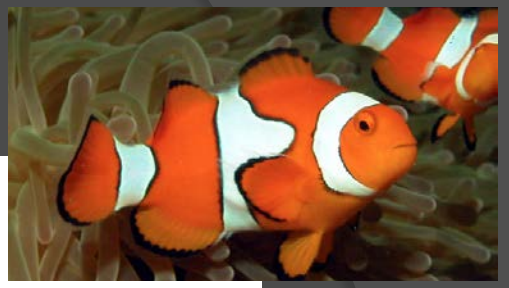
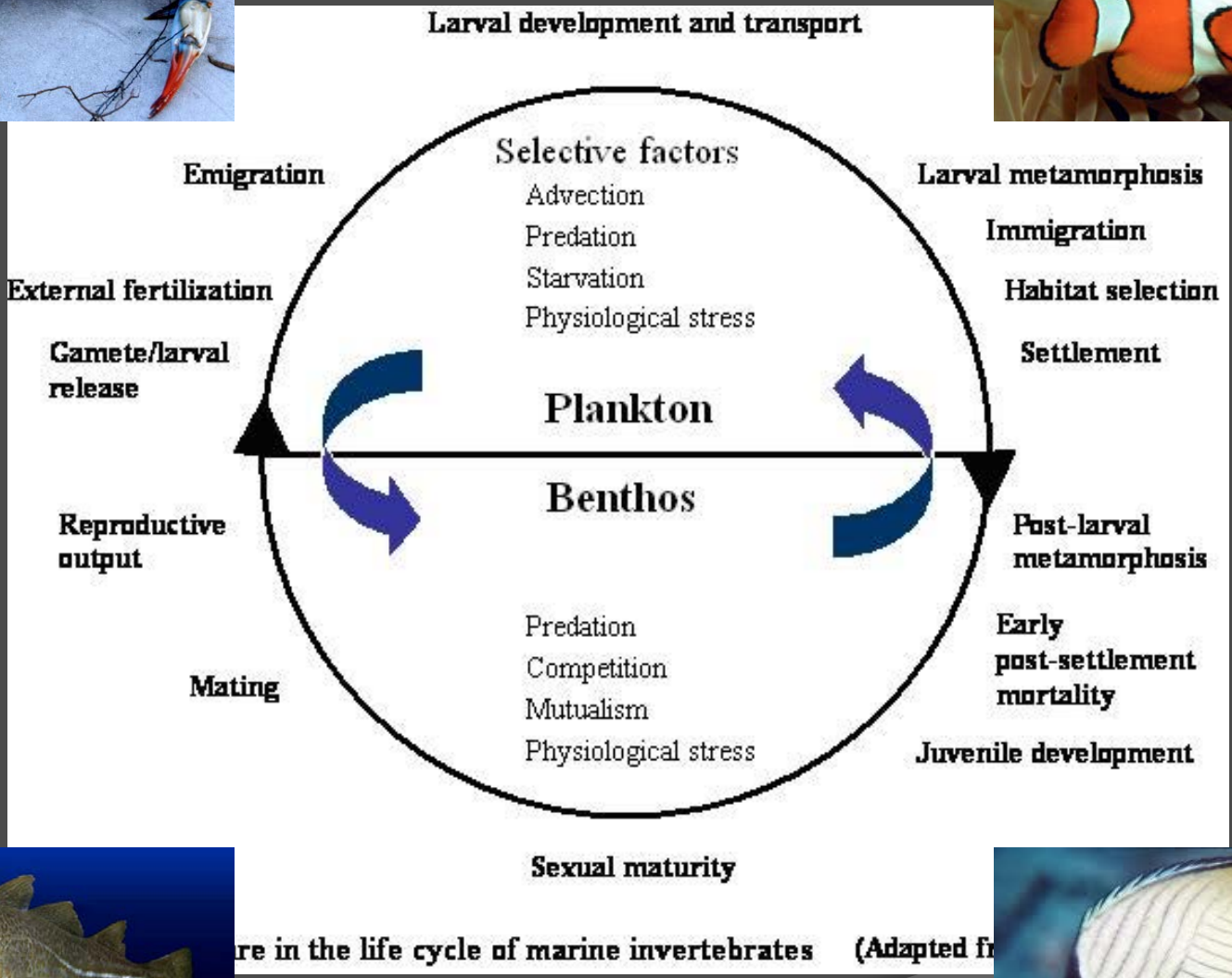
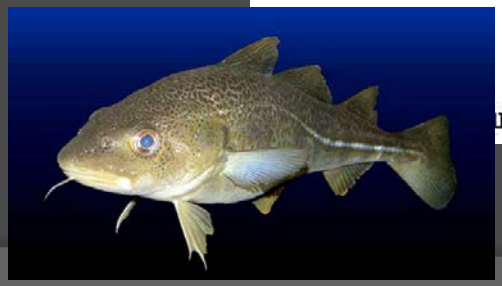
Benjamin Jones

Parallelization of a particle-tracking model

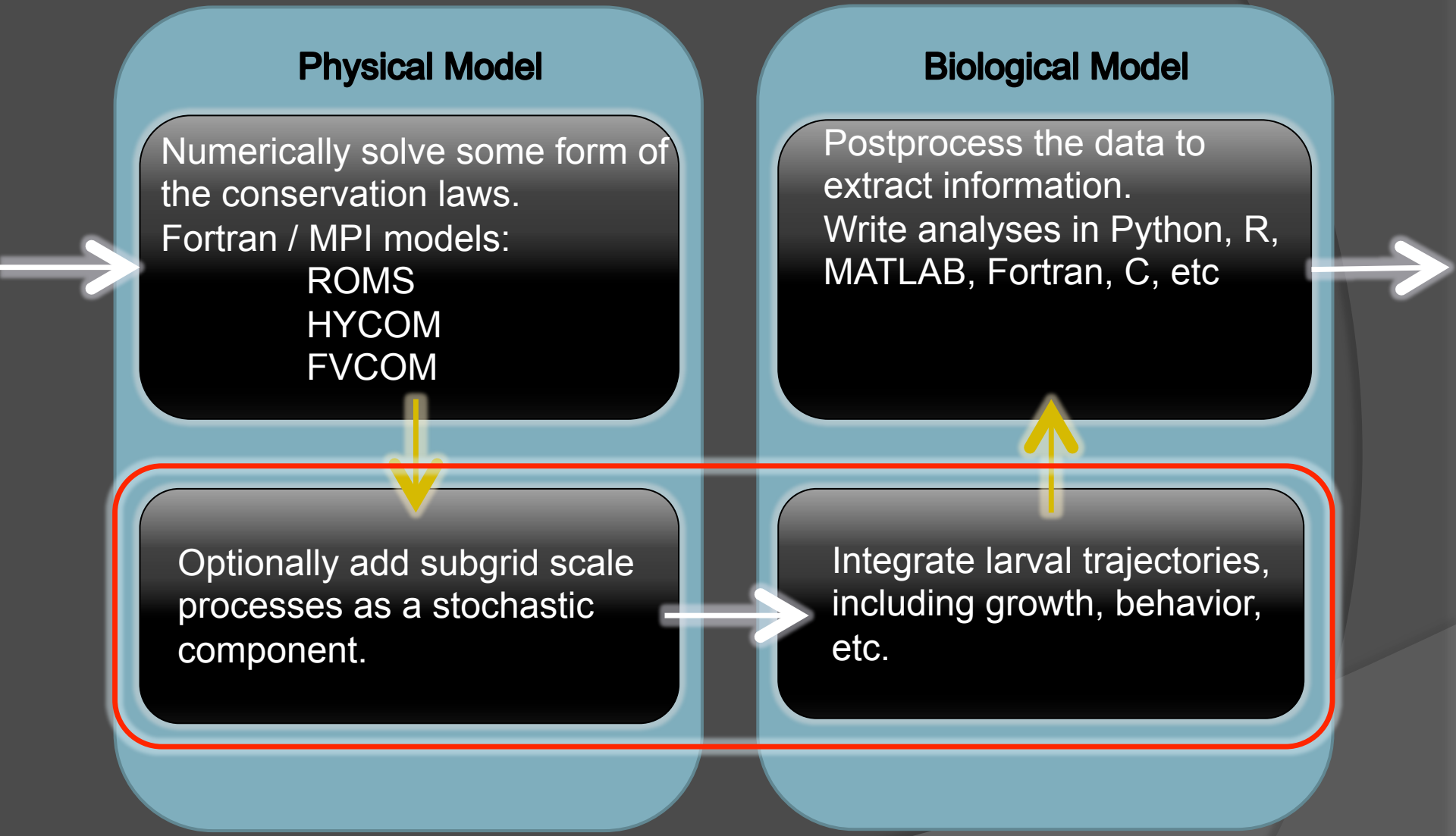


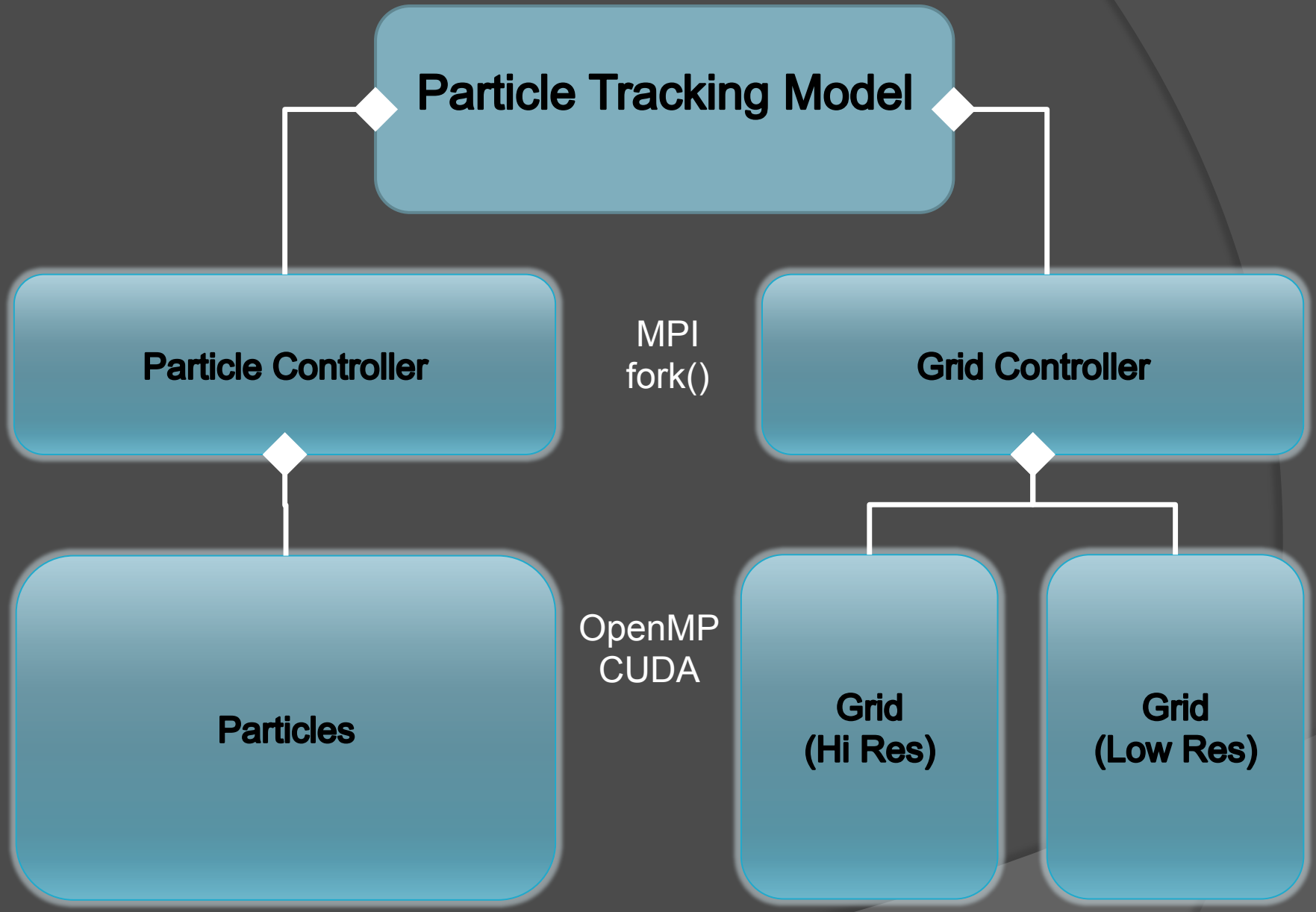


re in the life cycle of marine invertebrates (Adapted fr

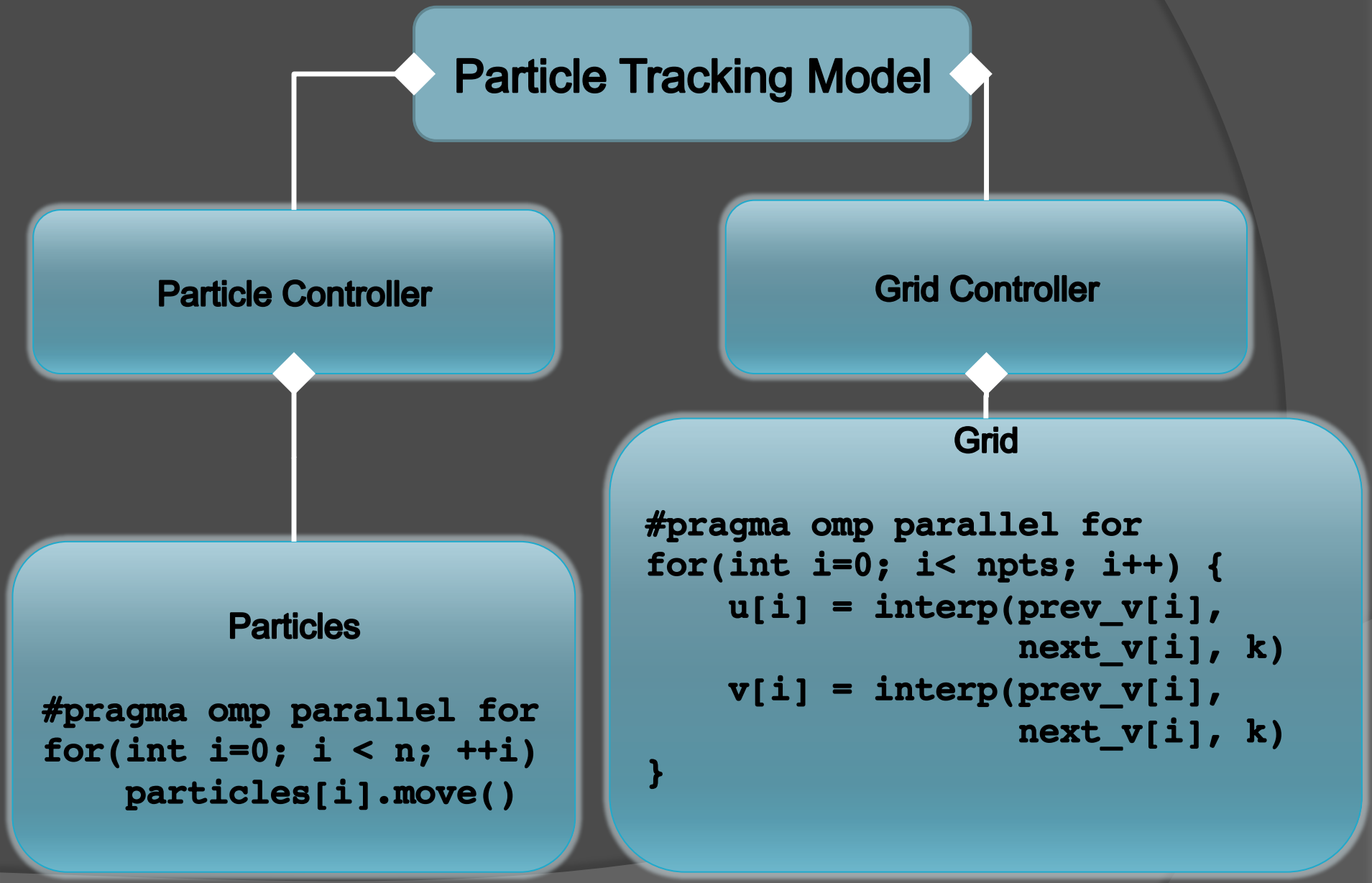


Biophysical Model Overview

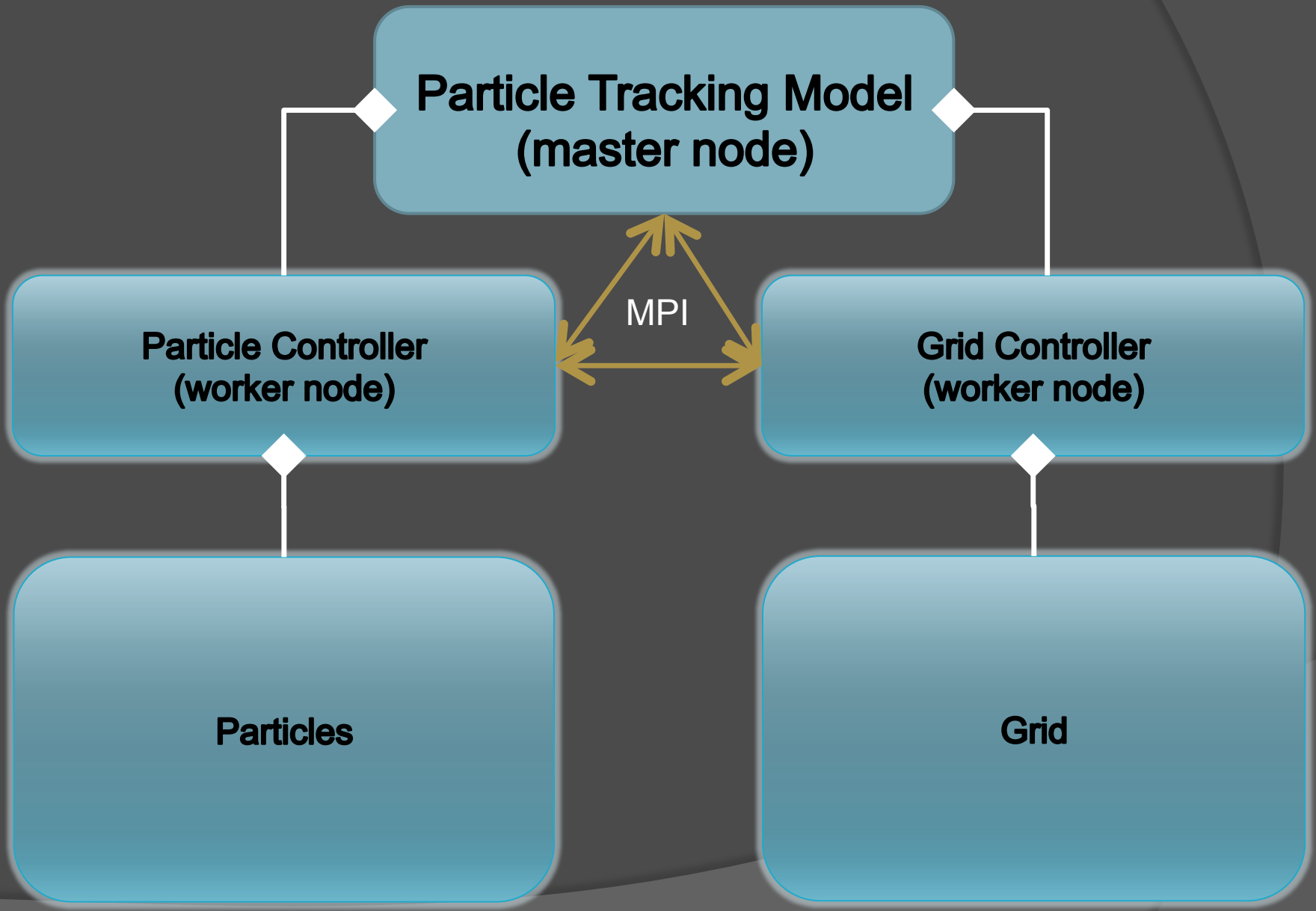



Particle Tracking Model
Particle Controller
MPI
fork()
Grid Controller
Particles
OpenMP
CUDA
Grid
(Hi Res)
Grid
(Low Res)

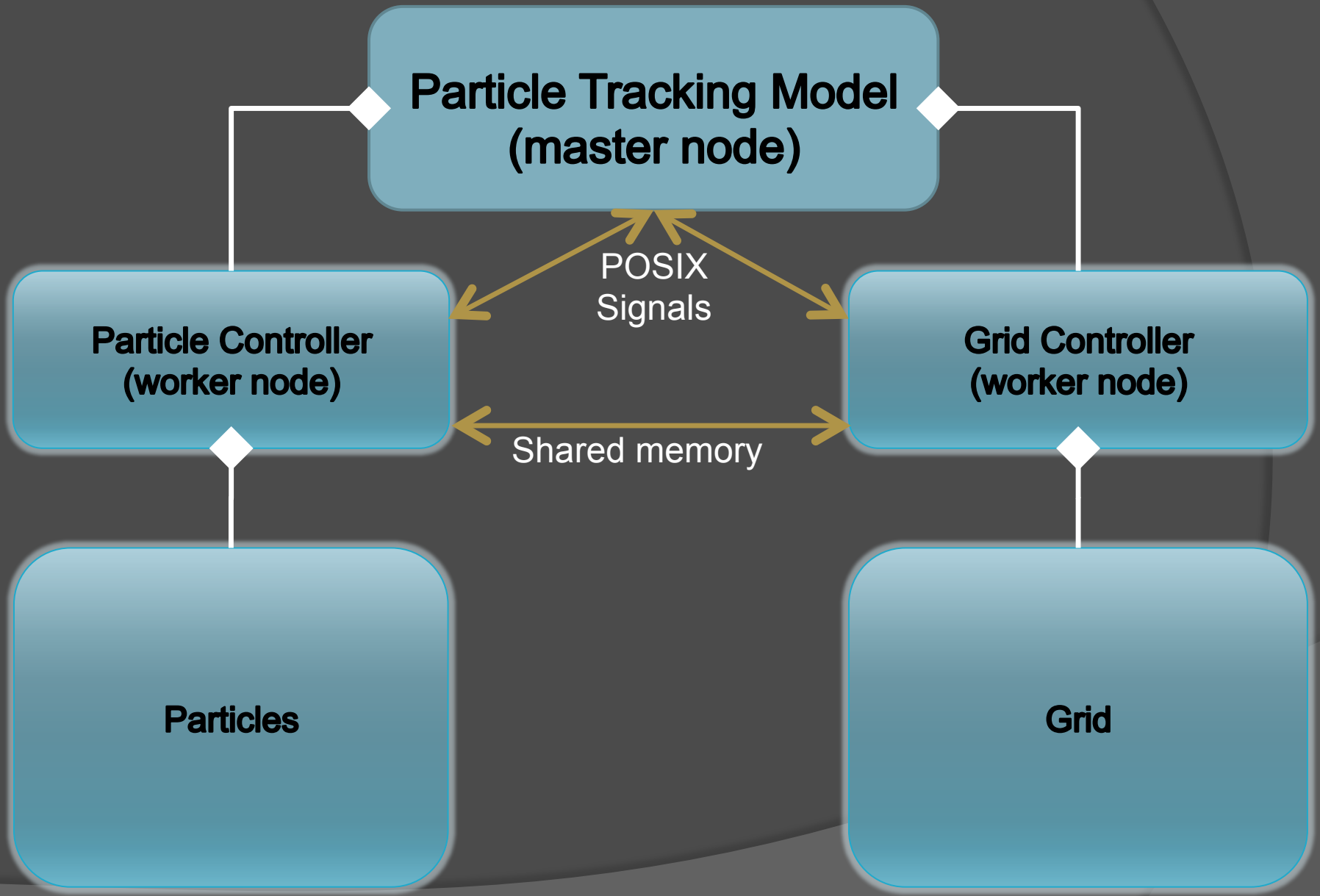
OpenMP Parallelization



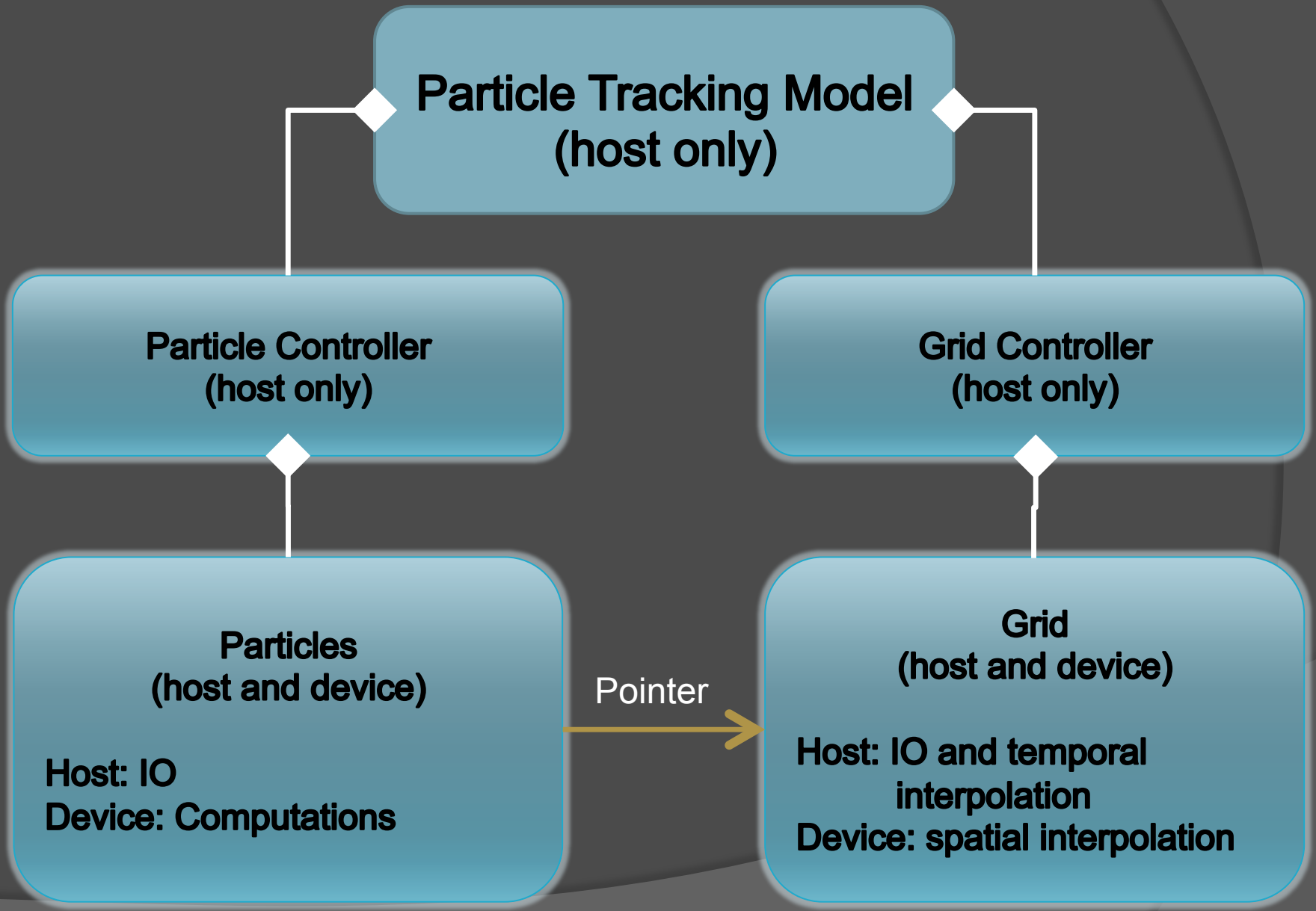
MPI Parallelization



Multi-process parallelization

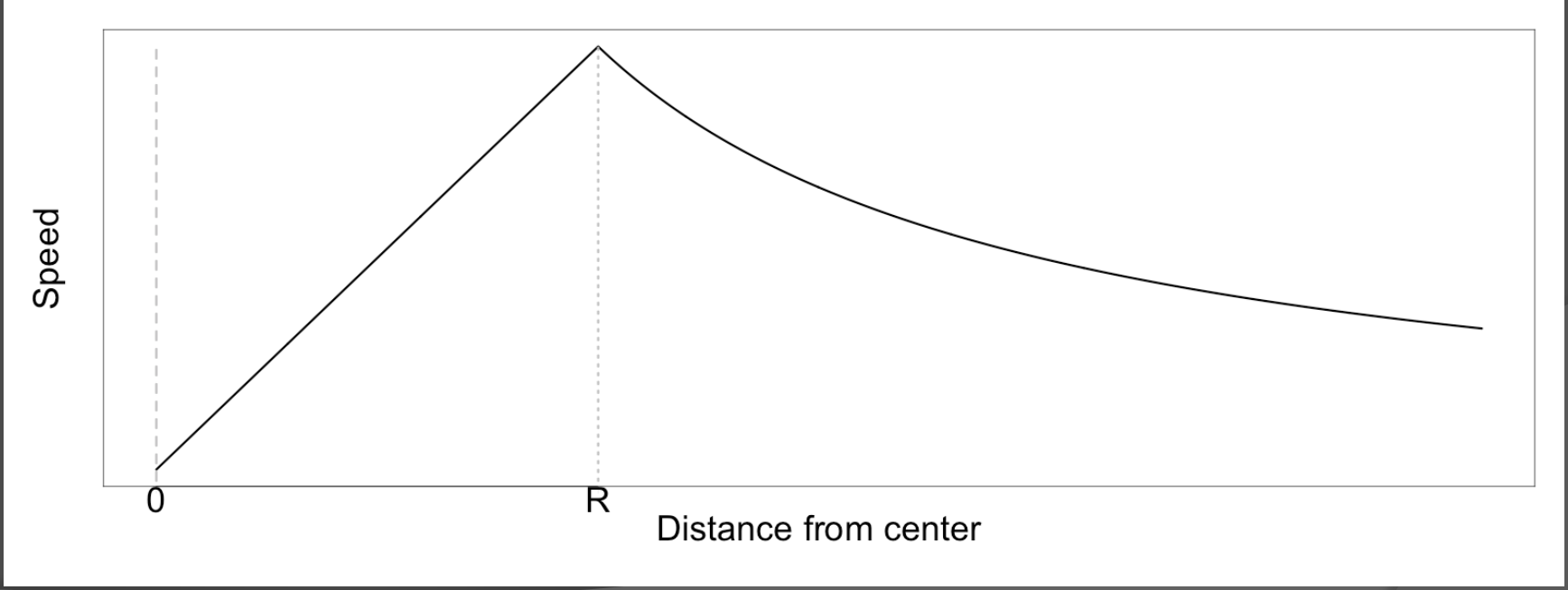


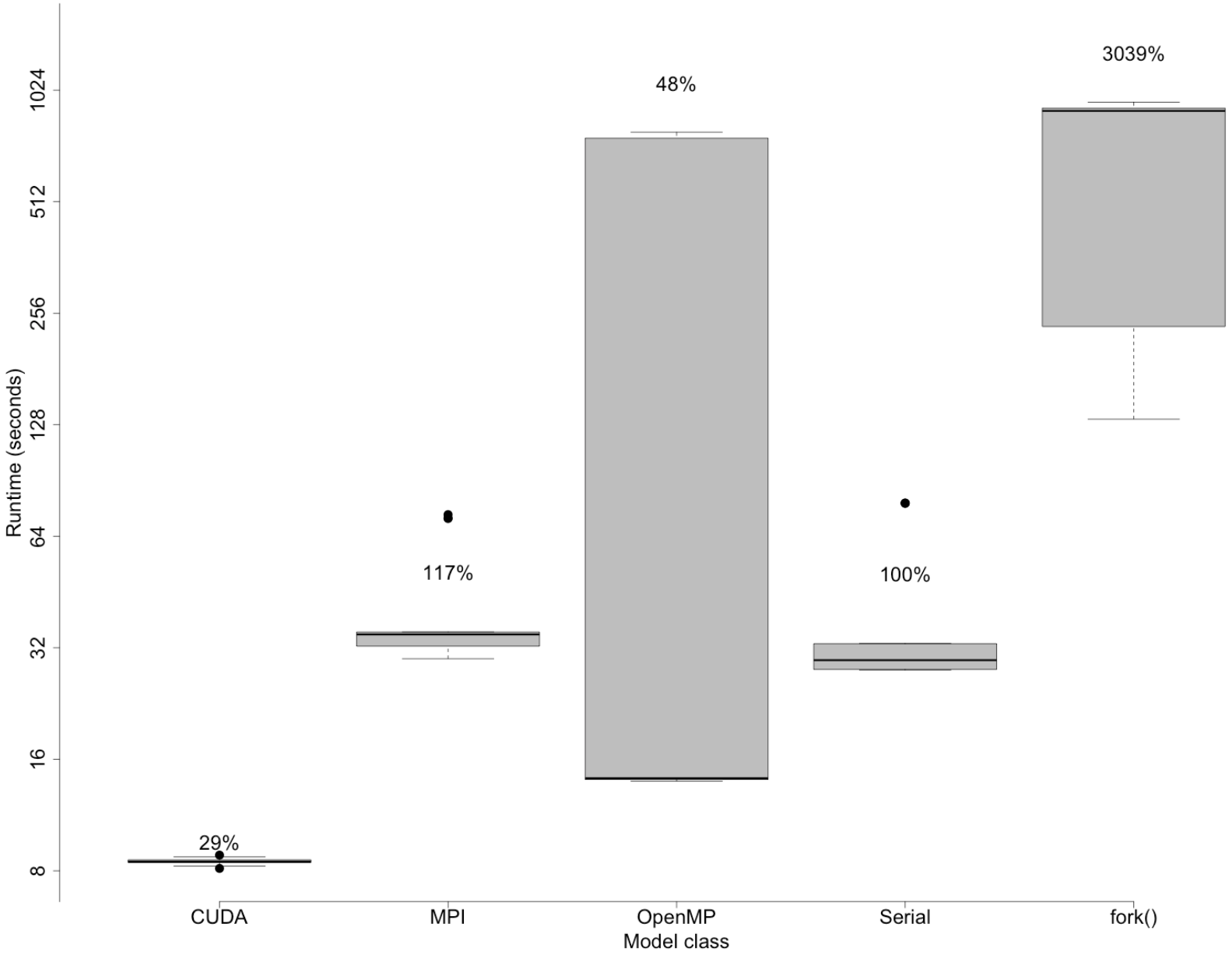
CUDA

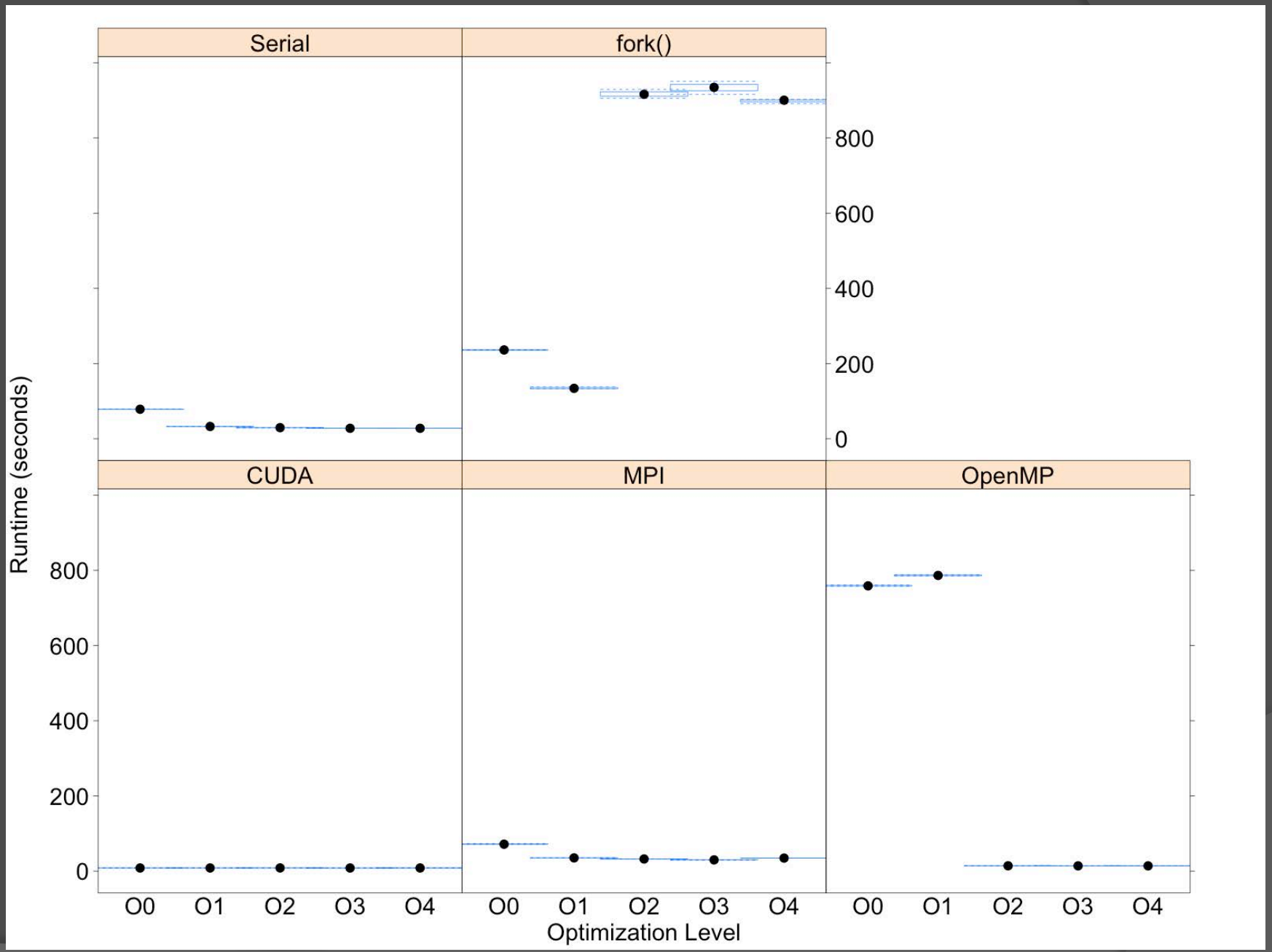

Particle Tracking Model
(host only)
Particle Controller
(host only)
Grid Controller
(host only)
Particles
(host and device)
Host: IO
Device: Computations
Pointer
Grid
(host and device)
Host: IO and temporal
interpolation
Device: spatial interpolation

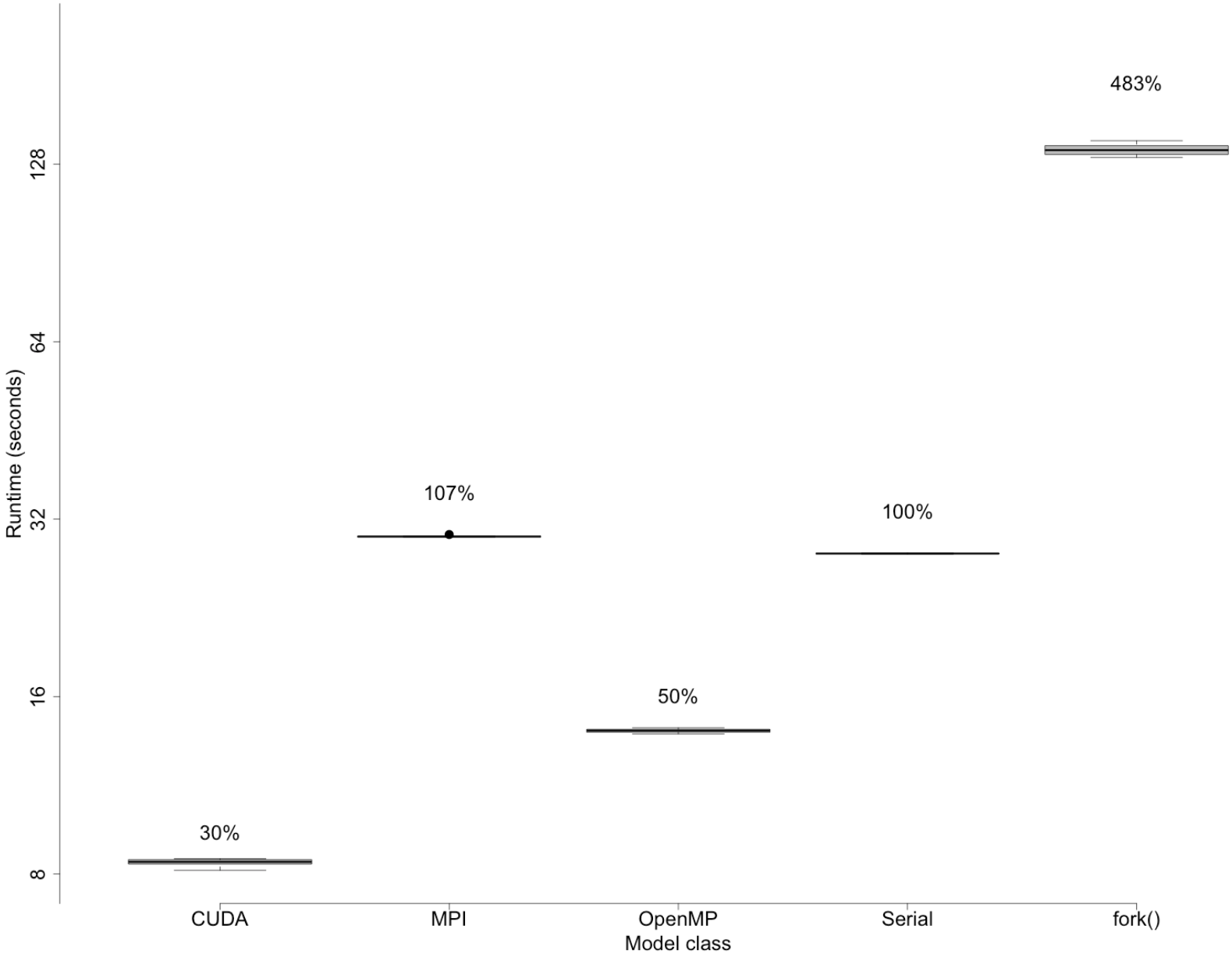
Flow field and parameterization

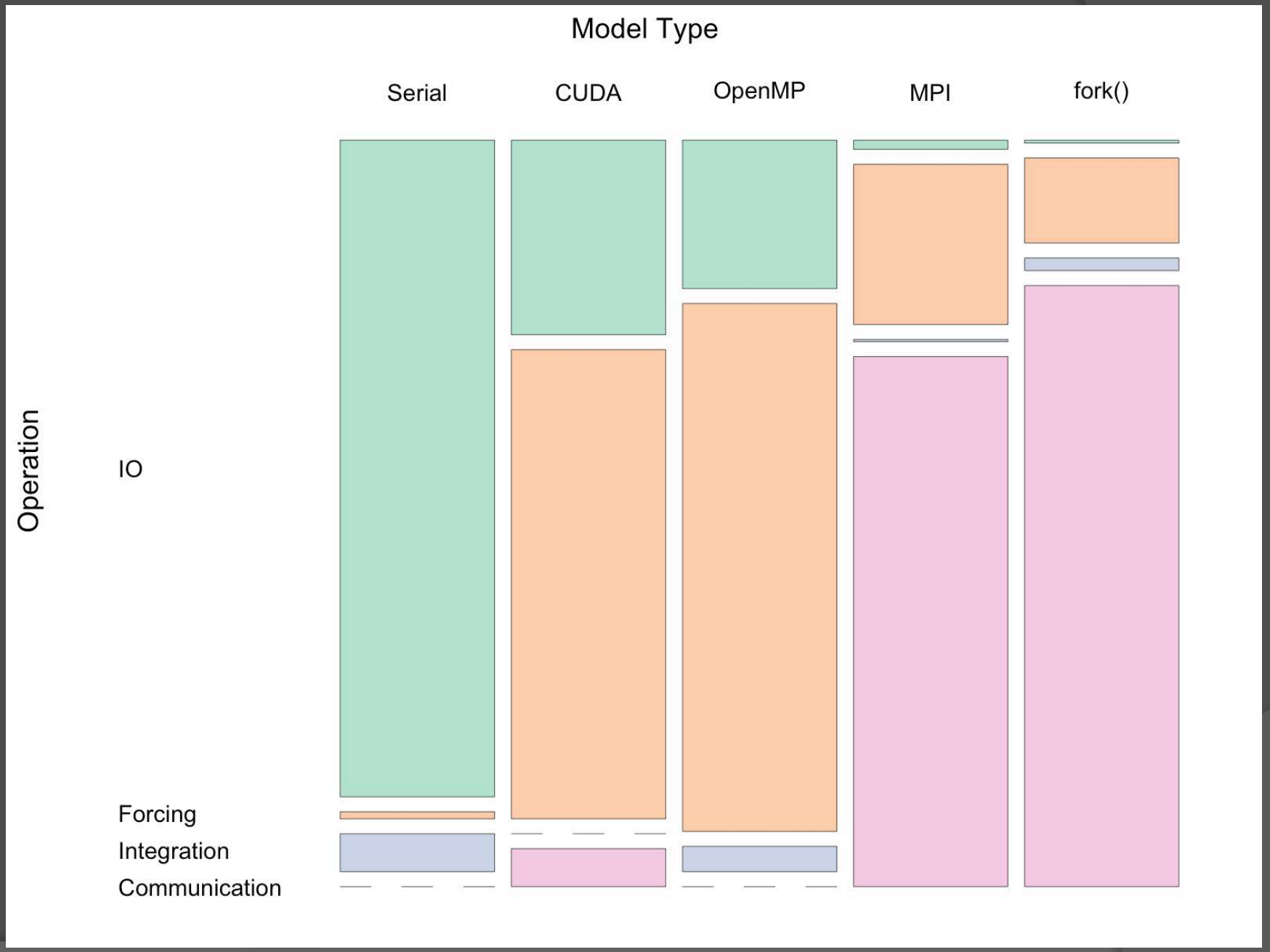
- 2D steadily translated Rankine vortex
- 500km X 1000km grid
- 1.5km and 4km lattice

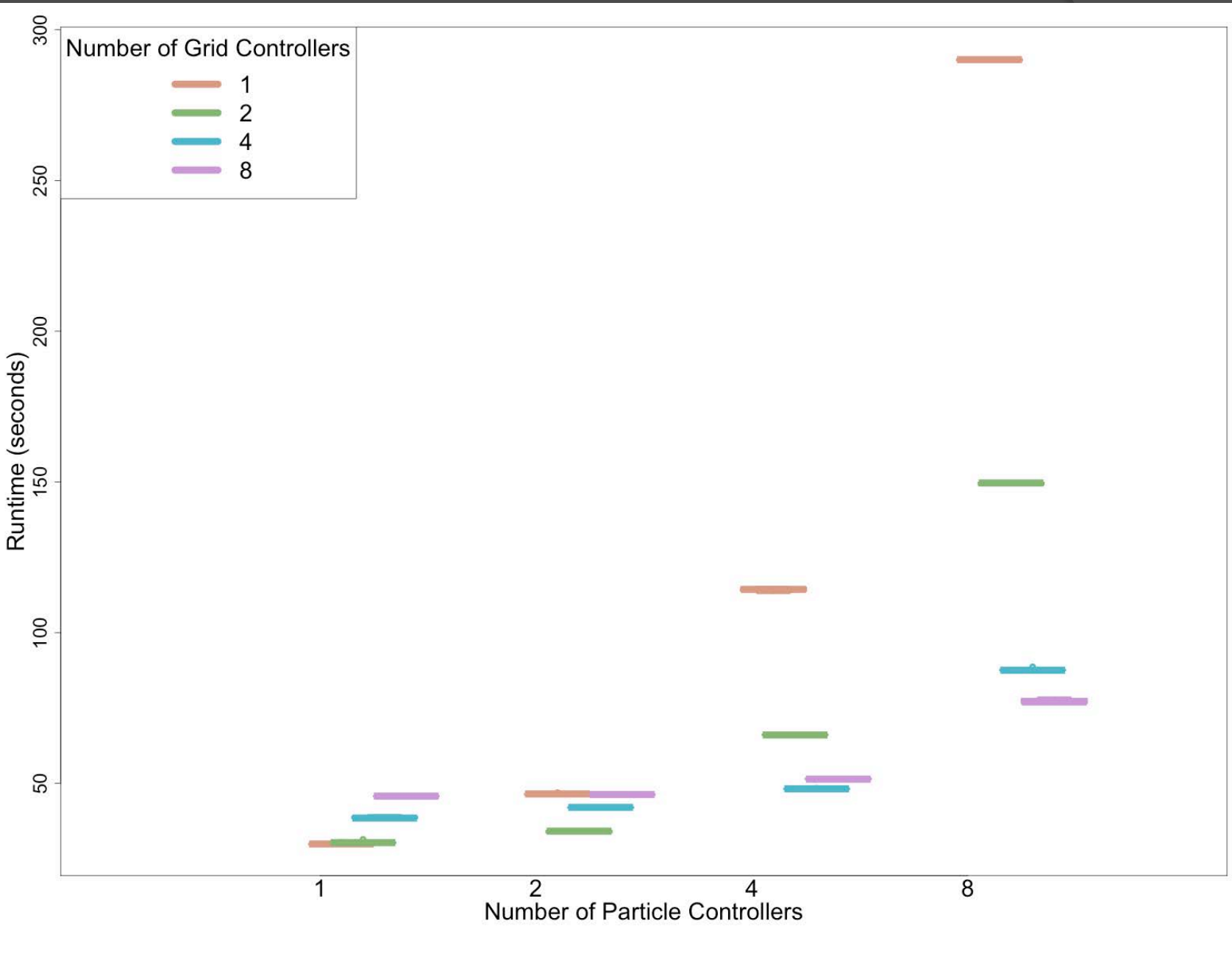



Runtime (seconds)
8
16
32
64
128
256
512
1024
29%
117%
48%
100%
3039%
CUDA
MPI
OpenMP
Serial
fork()
Model class


Serial
fork()
CUDA
MPI
OpenMP
Runtime (seconds)
0
200
400
600
800
O0
O1
O2
O3
O4
Optimization Level


Runtime (seconds)
8
16
32
64
128
30%
107%
50%
100%
483%
CUDA
MPI
OpenMP
Serial
fork()
Model class


Model Type
Serial
CUDA
OpenMP
MPI
fork()
Operation
IO
Forcing
Integration
Communication


Number of Grid Controllers
1
2
4
8
Runtime (seconds)
300
250
200
150
100
50
1
2
4
8
Number of Particle Controllers

Possible Extensions: Features

- Support for real ocean models
  - FVCOM *
  - HYCOM
  - ROMS
- Advection-diffusion module *
- Biological module
  - Density-dependence *
  - Growth
  - Movement

Possible Extensions: MPI

- Try MPICH2
- Minimize communication
- Optimize arrangement of nodes
- OpenMP within each node
- Dynamically balance number of grid controllers and particle controllers

Possible Extensions: CUDA

- Tune threads per block vs. number of blocks
- Reduce data movement
- Make used of shared or thread memory
- Move temporal interpolation to GPU
- Integrate analyses into the model

Conclusions

- Optimization level matters
- GPUs can offer cheap parallelism
- Beware of context switching
- Communication is expensive
- Lots of opportunity for improvement