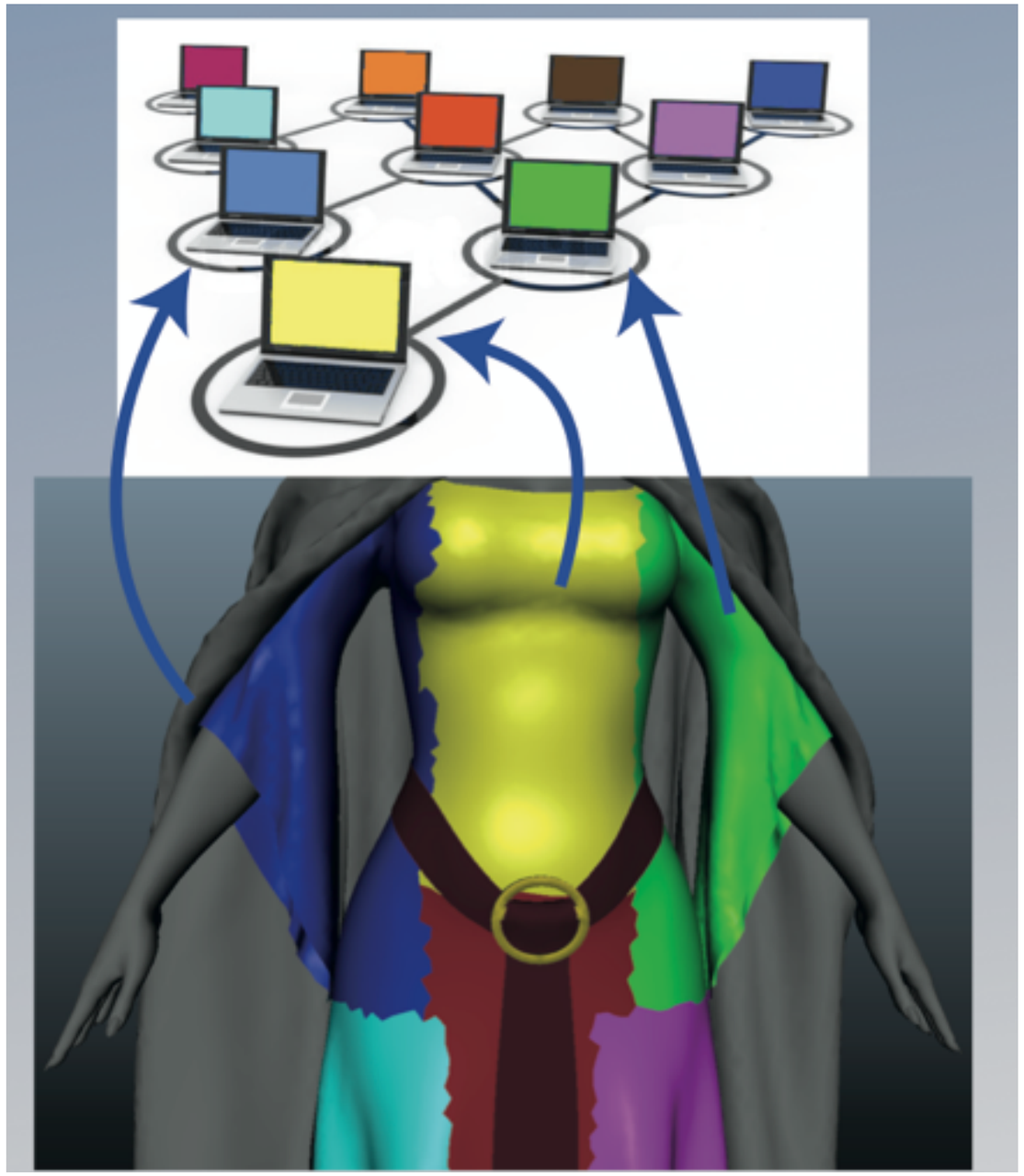
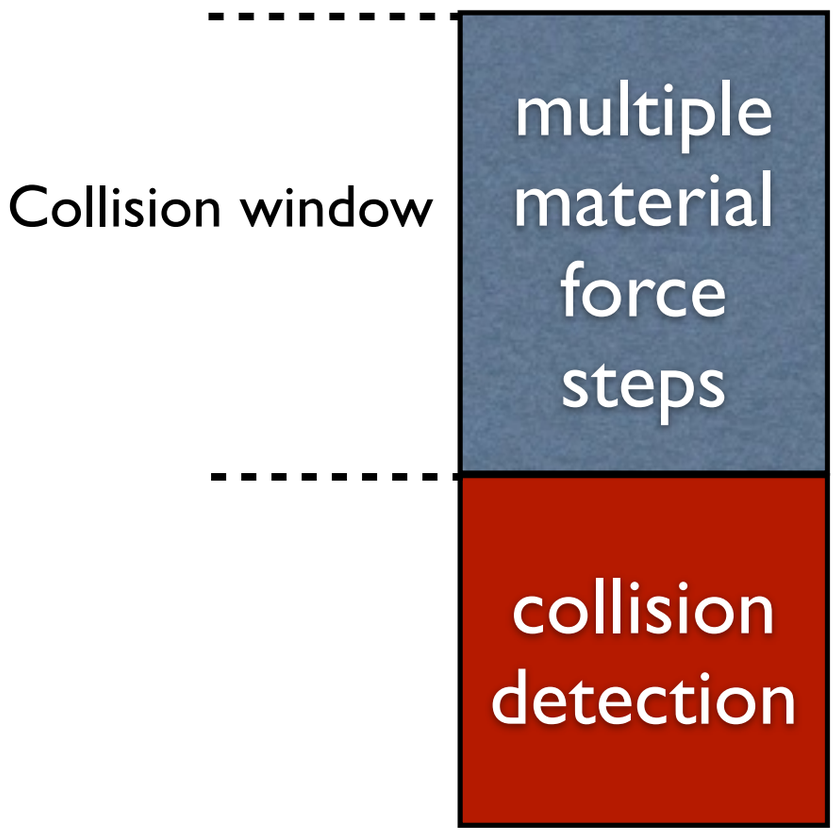
Decomposition

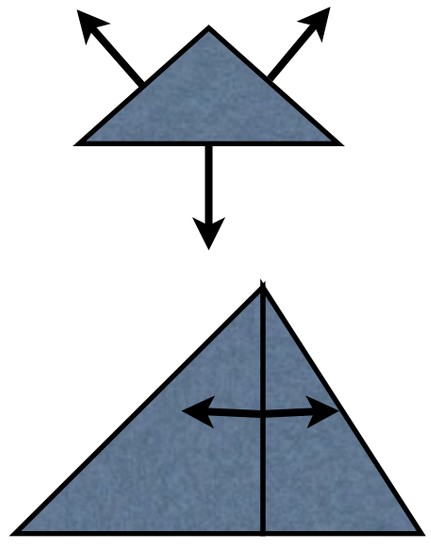
Cloth is decomposed into 2D Triangle mesh

The mesh is partitioned using METIS among work units (a 1D chare array)



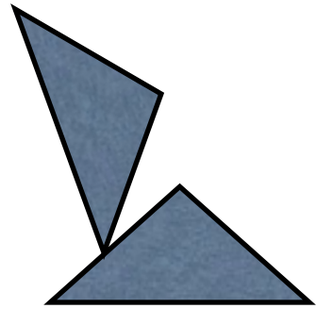
Overall Flow



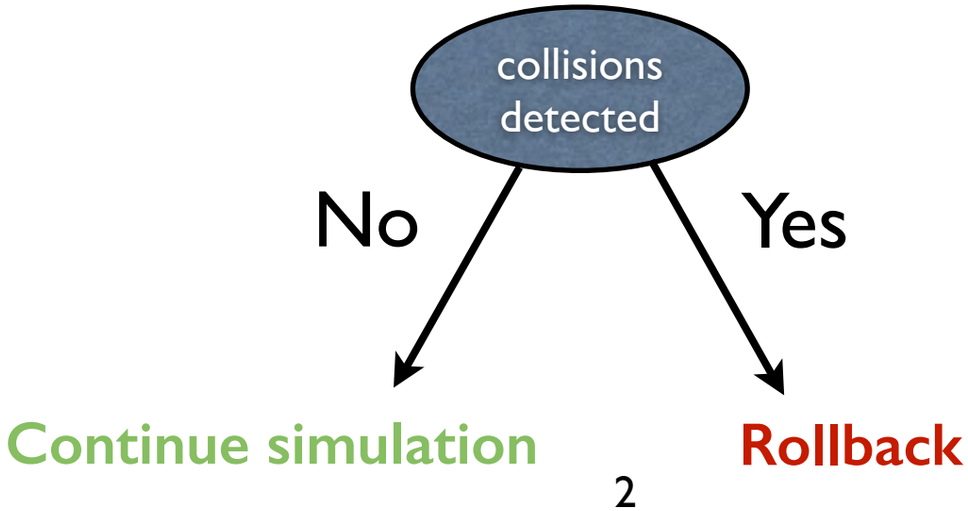


stretch force

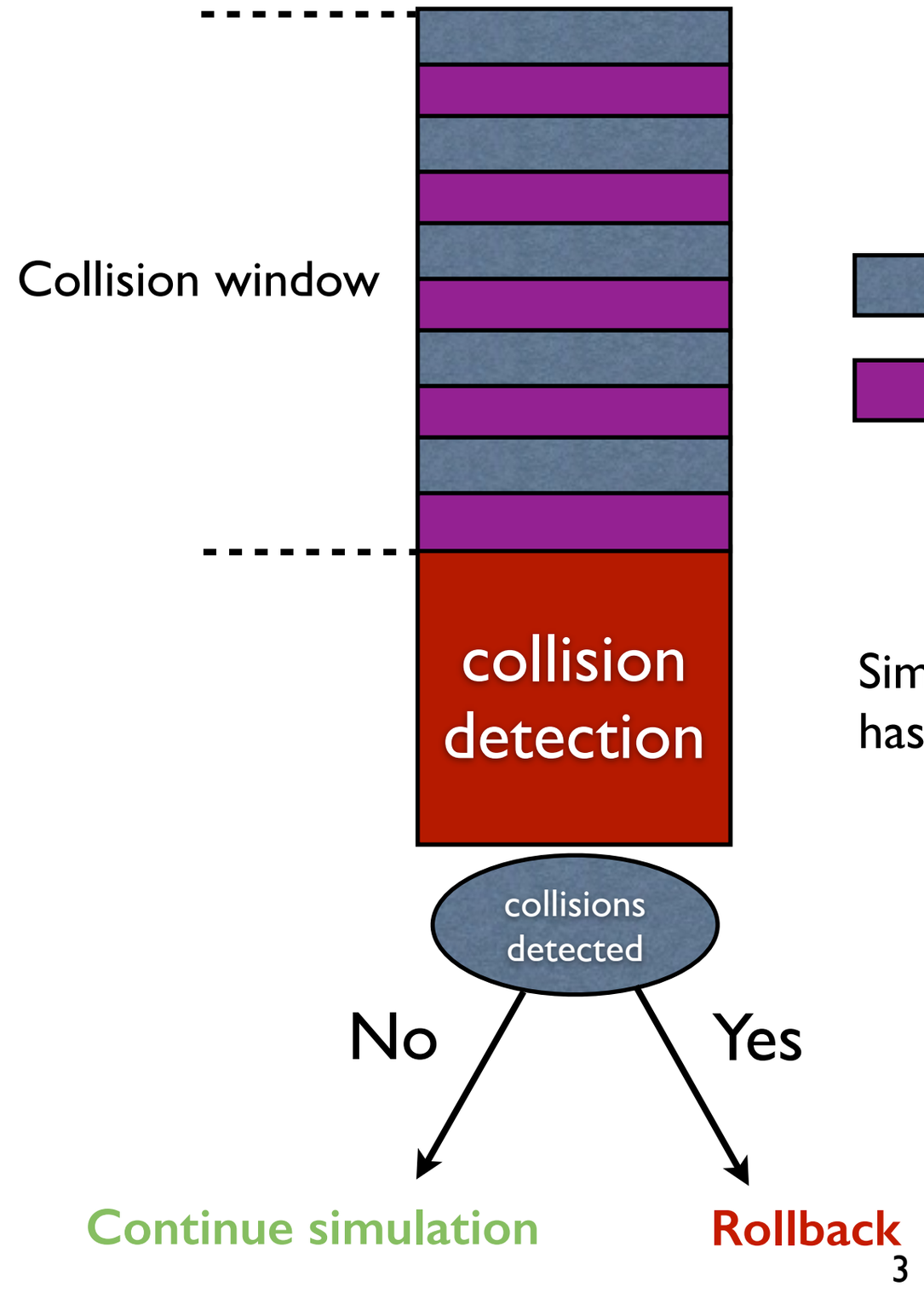
bending force



cloth pieces penetrate into each other



Rollback



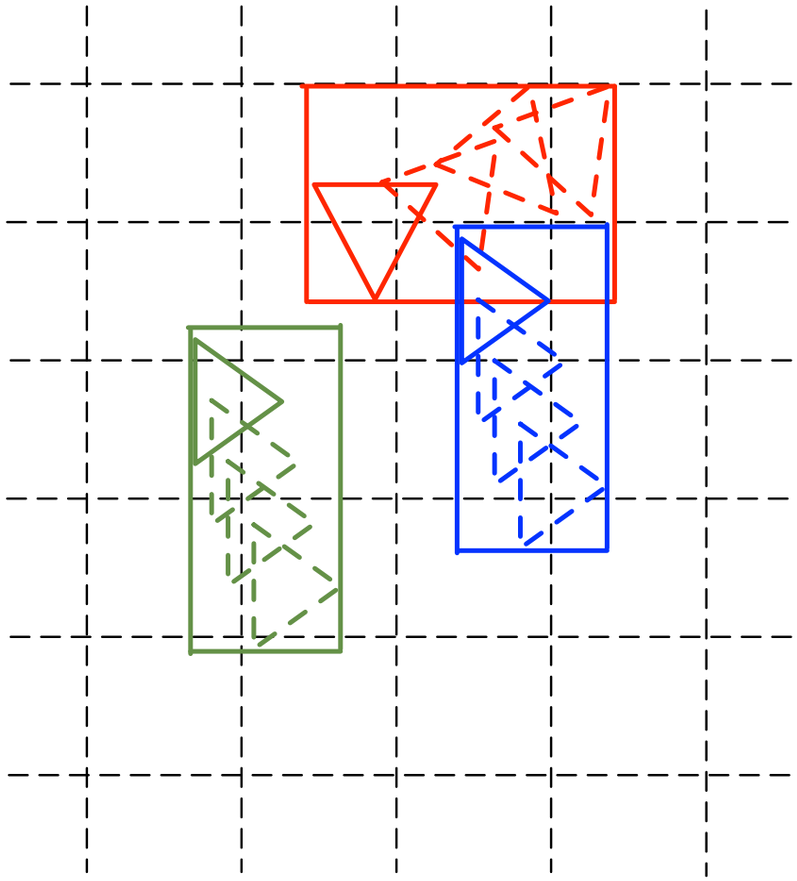
Material force

Penalty force prevents cloth piece from penetrating

Simulation proceeds only when no collision has been found.

Collision Detection

- Charm++ collide library for **broad** phase collision detection*



Create bounding box for the trajectory of each triangle

Result:

Red collides with Blue

Red collides with Green **False Positive**

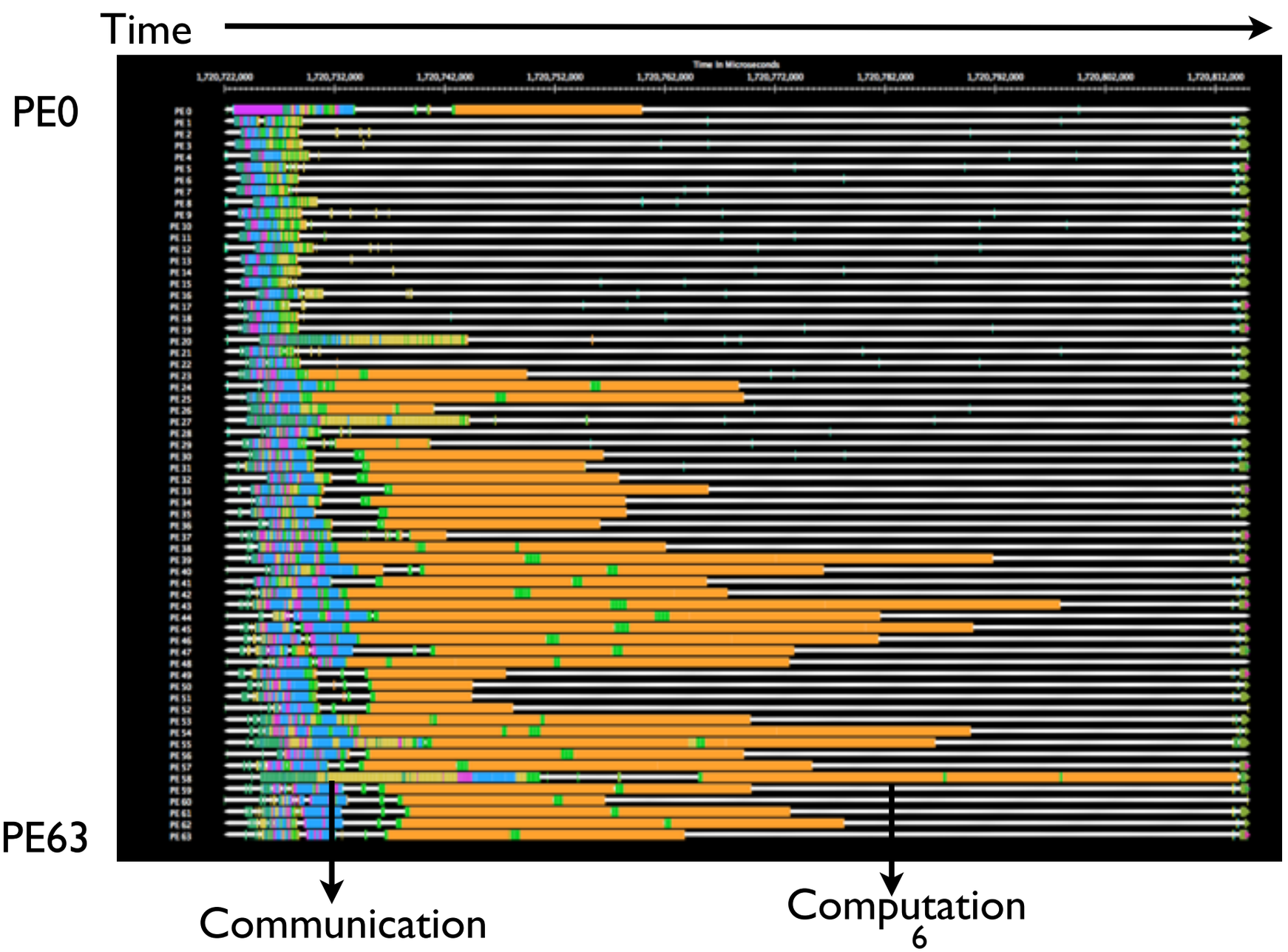
* *A Voxel-Based Parallel Collision Detection Algorithm*. Orion Lawlor, Laxmikant Kale. ICS02

Narrow Phase Collision Detection

- Filter out the **false positive** results
- A closer look at the position of each triangle at each time step.
- First, we distribute the potential collision results evenly among all the processors.

Narrow Phase

Timeline view from projections

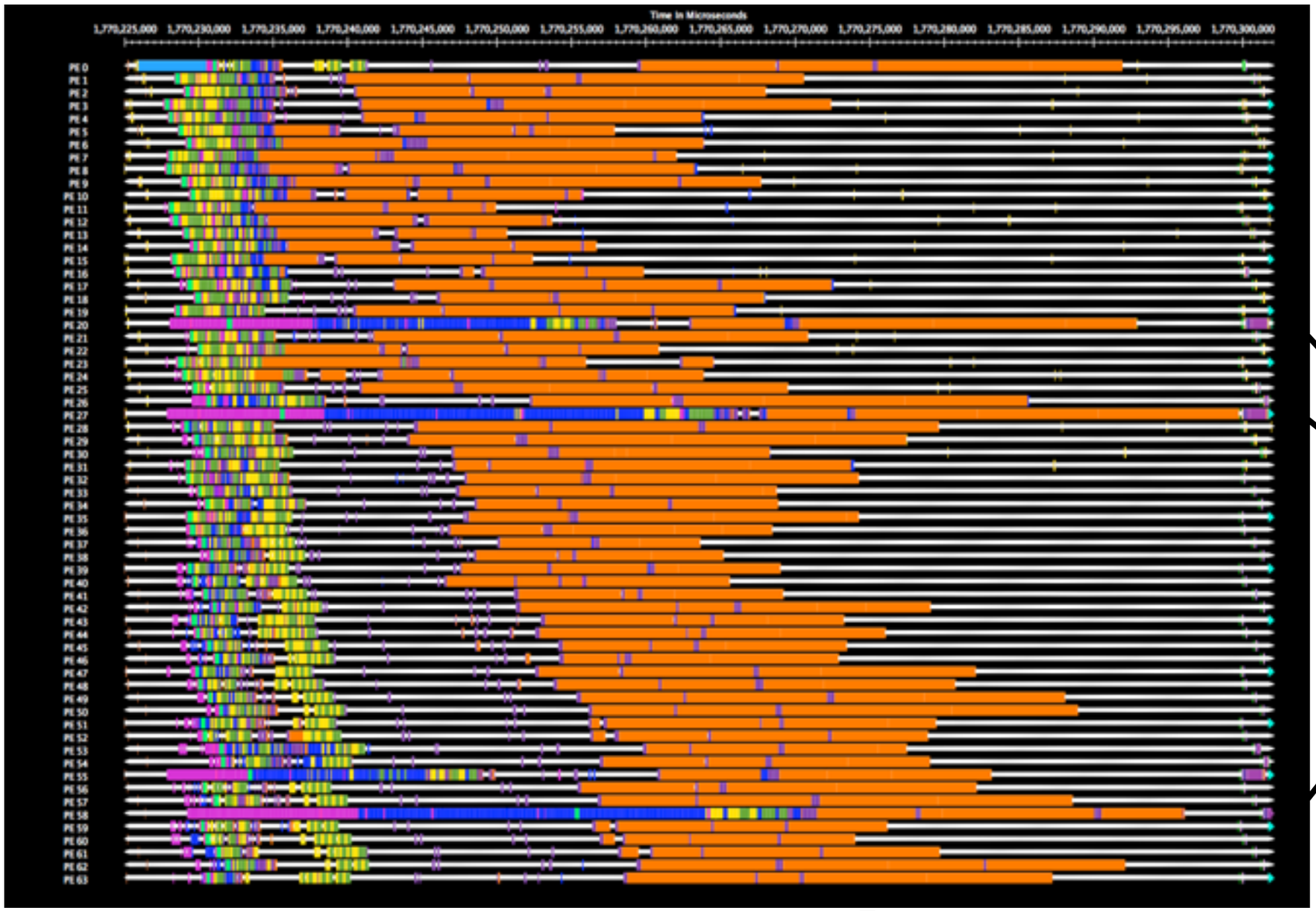


Computation Imbalance?

- Time spent on each collision pair is different
- Different trajectory length
- Collision happens earlier or later?

Narrow Phase: Principle of Persistence

Take the variation into consideration by profiling

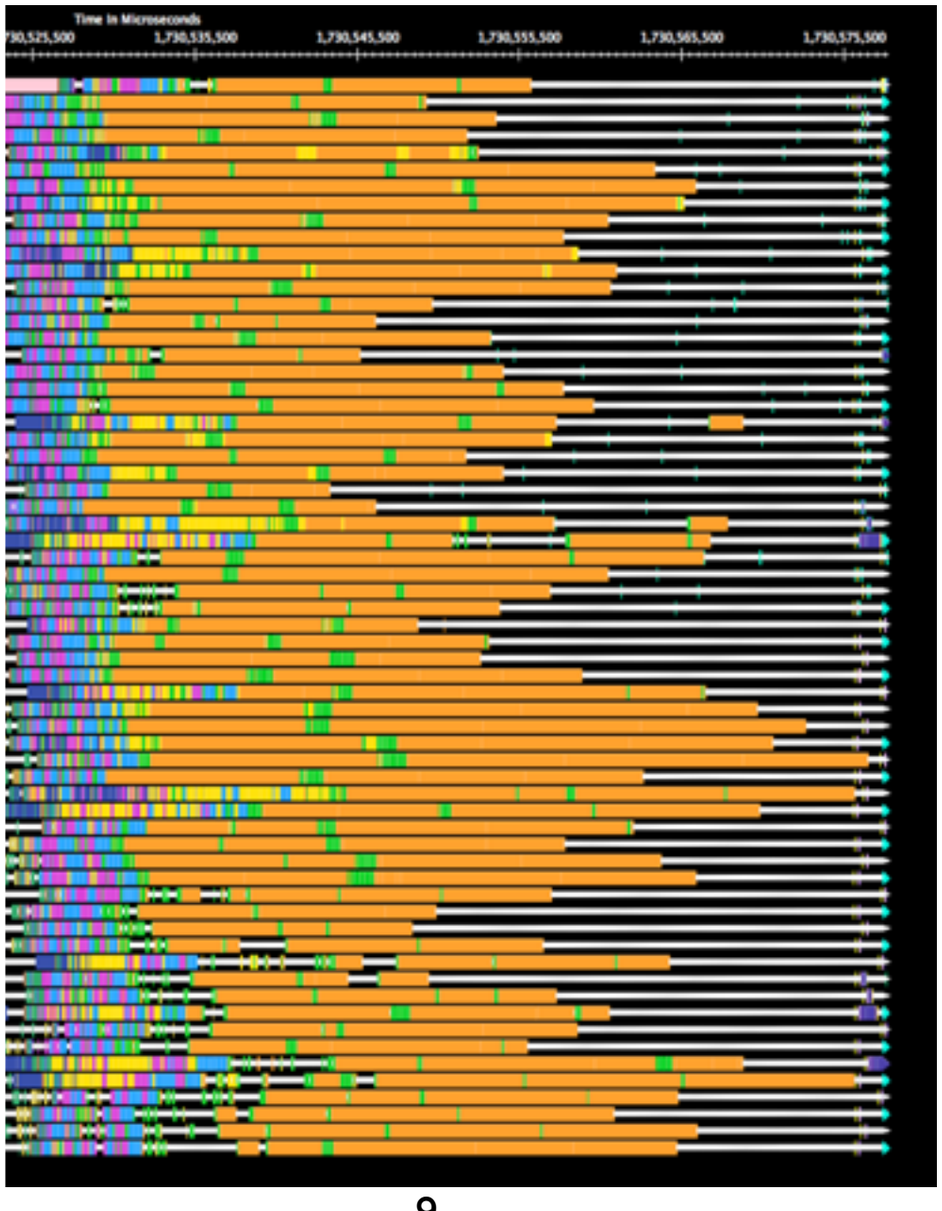


more communication

Narrow Phase

- Communication Imbalance
- Replicate the data on the heavily loaded processors to a few other processors

Narrow Phase: Effect of Replication



Why does replication take long?

- **Large message size**
- **Less data reuse**

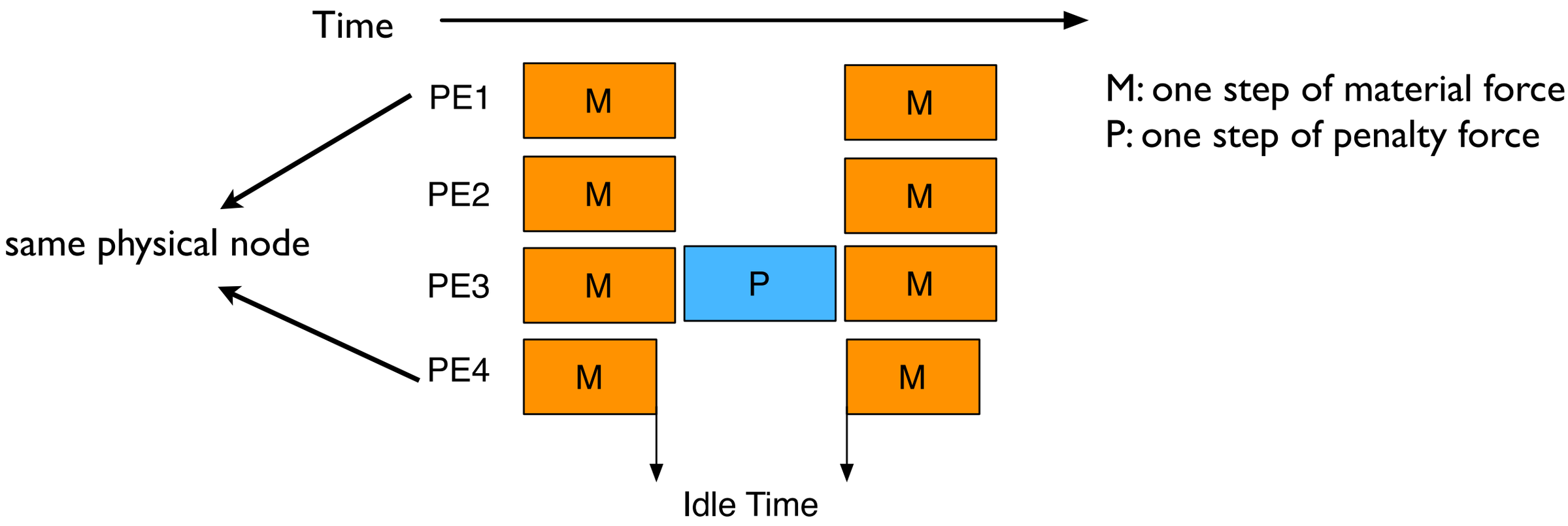
Narrow Phase: Communication Imbalance

- Let's consider the physical node as an execution unit
- In the same node, processors with **less** communication can naturally offload the computation work from the processors with **more** communication.
  - Enabled by **asynchronous message driven execution**
- **Priority** based execution is important
  - Communication request gets processed as soon as possible

Narrow Phase: Node-level Parallelization

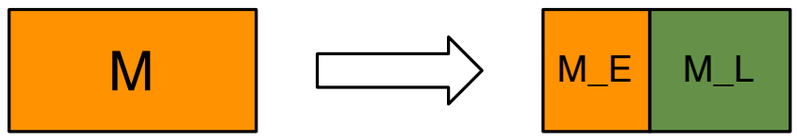
- Communication requests get processed first
  - **Higher** priority
- Different communication computation ratio, but work gets balanced overall
  - Processors poll for work as soon as they get idle

Force Computation

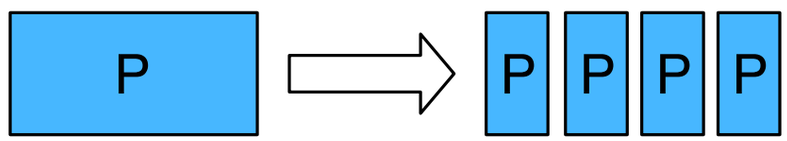


- Phase by phase simulation
- Imbalanced distribution of penalty force
  - Collisions only happen in a few regions

Decomposing Force Computation

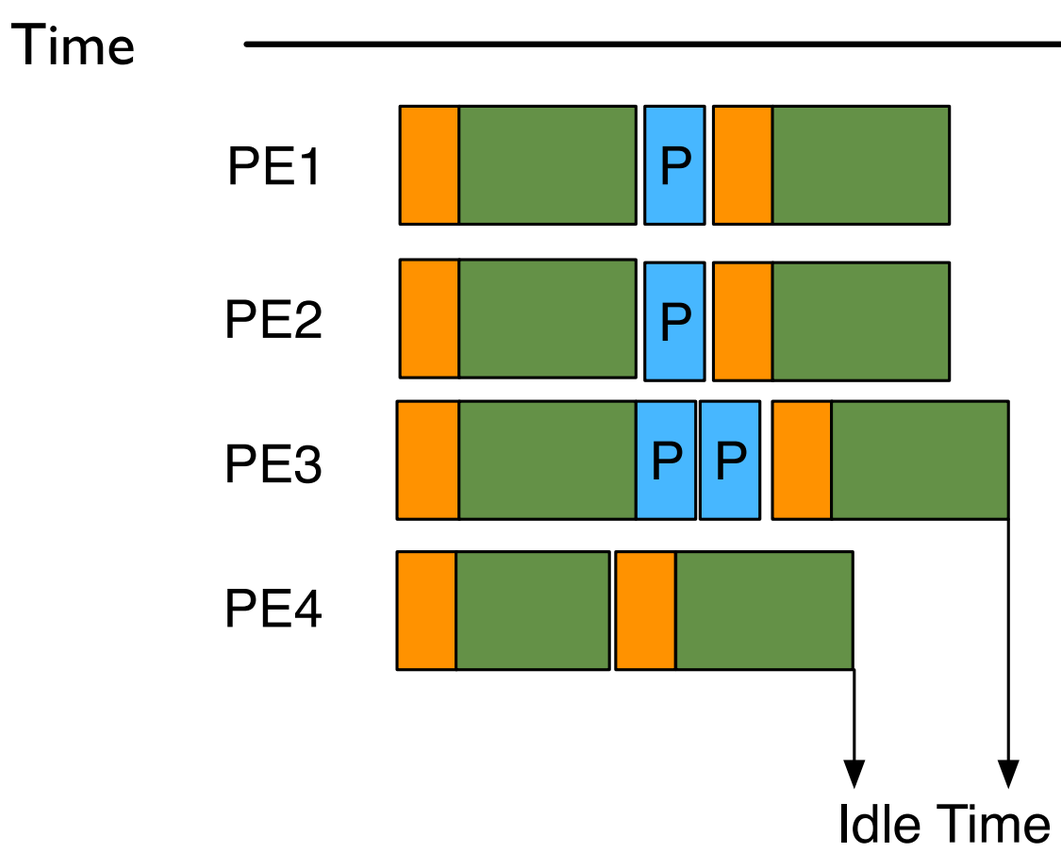


M_E: external computation
M_L: local computation



Penalty force decomposition

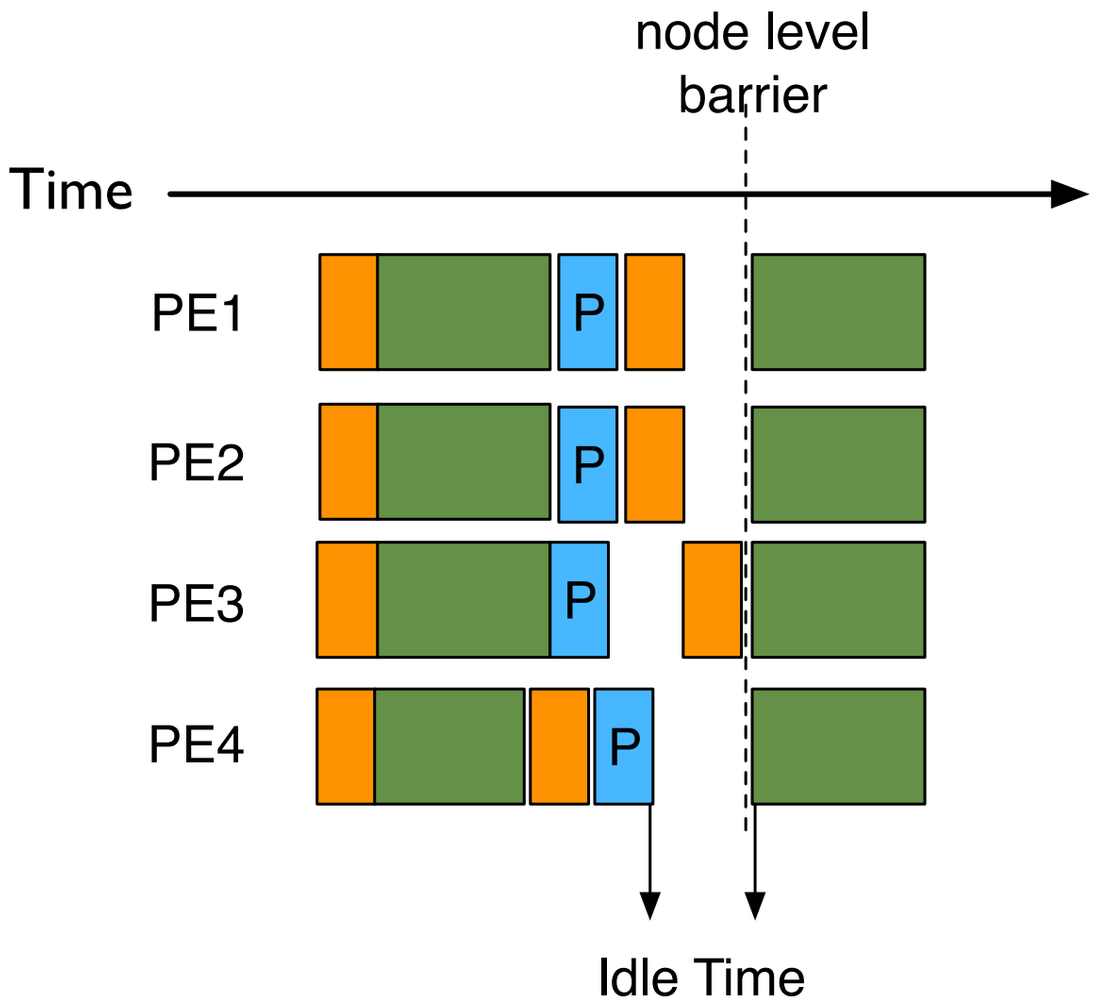
Overdecomposed Force Computation



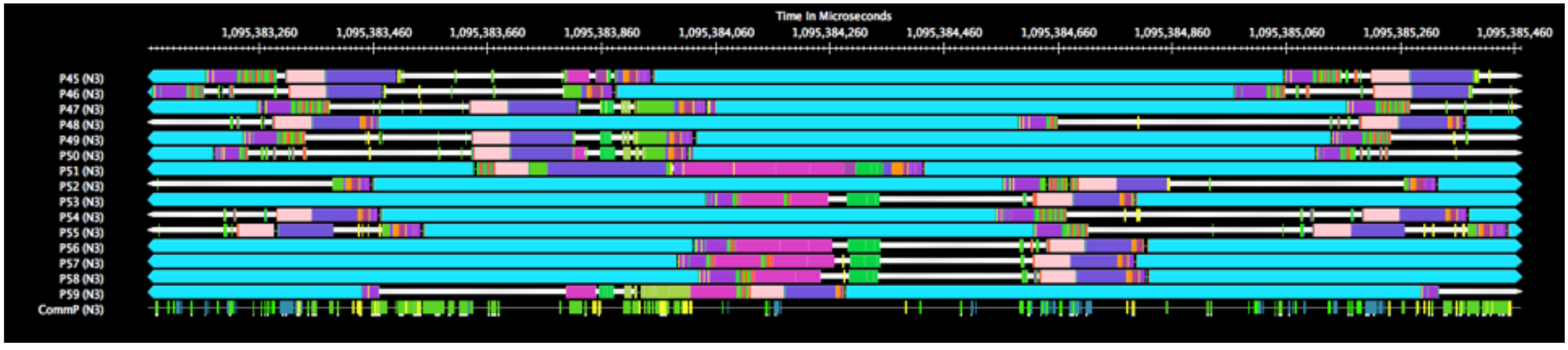
Other processors can help PE3 with penalty computation except **PE4**. Why?

The message which invokes the long material force calculation on PE4 comes early!

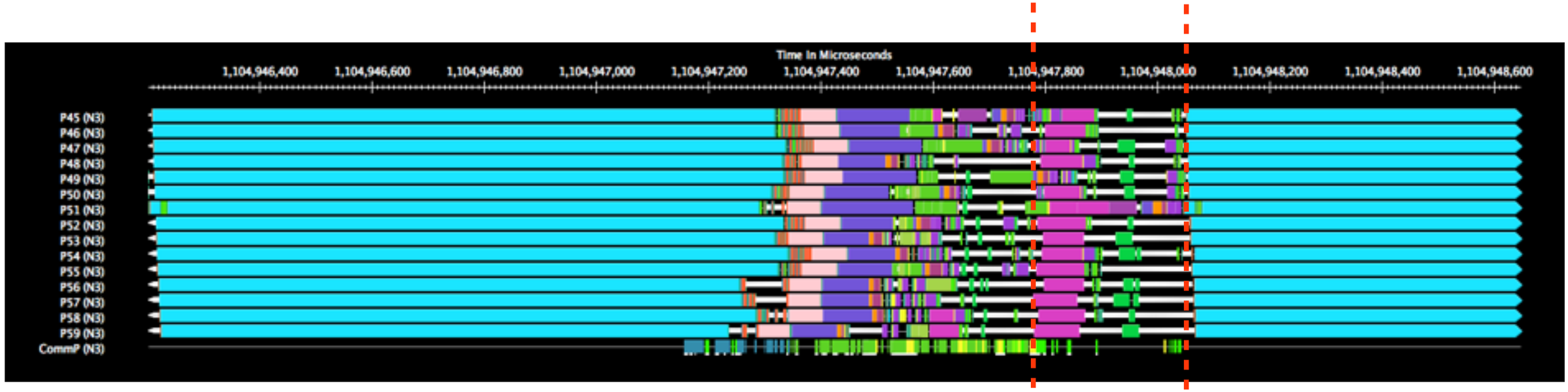
Force Computation



A low cost node level phase barrier to delay the long computation.
PE4 can participate to help PE3 with penalty force calculation.

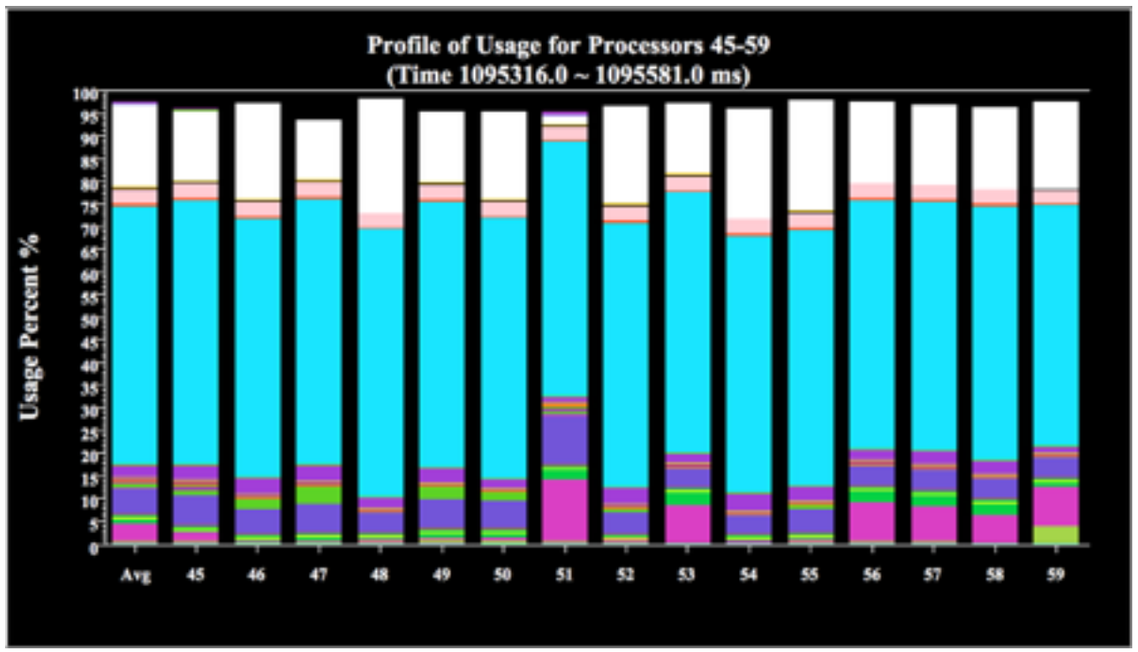


Only a few PEs (53,56,57,58,59) help with the Penalty force computation (from PE51).

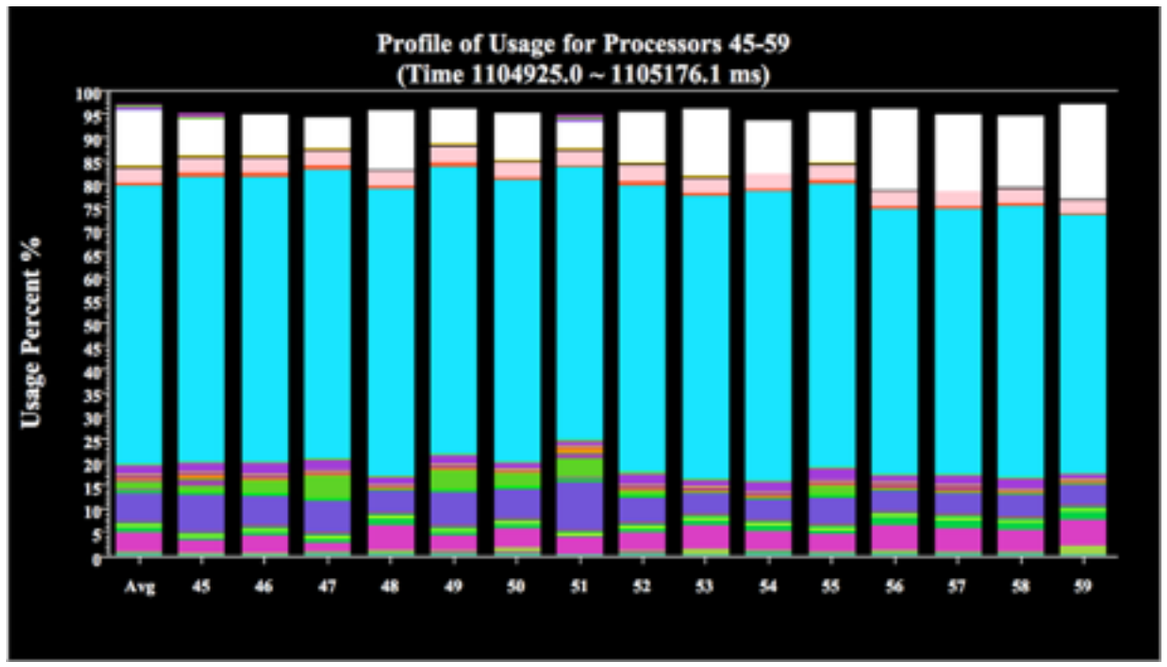


Penalty force computation (from PE51) is divided among all PEs on the node.

Effect of Node-level Phase Barrier

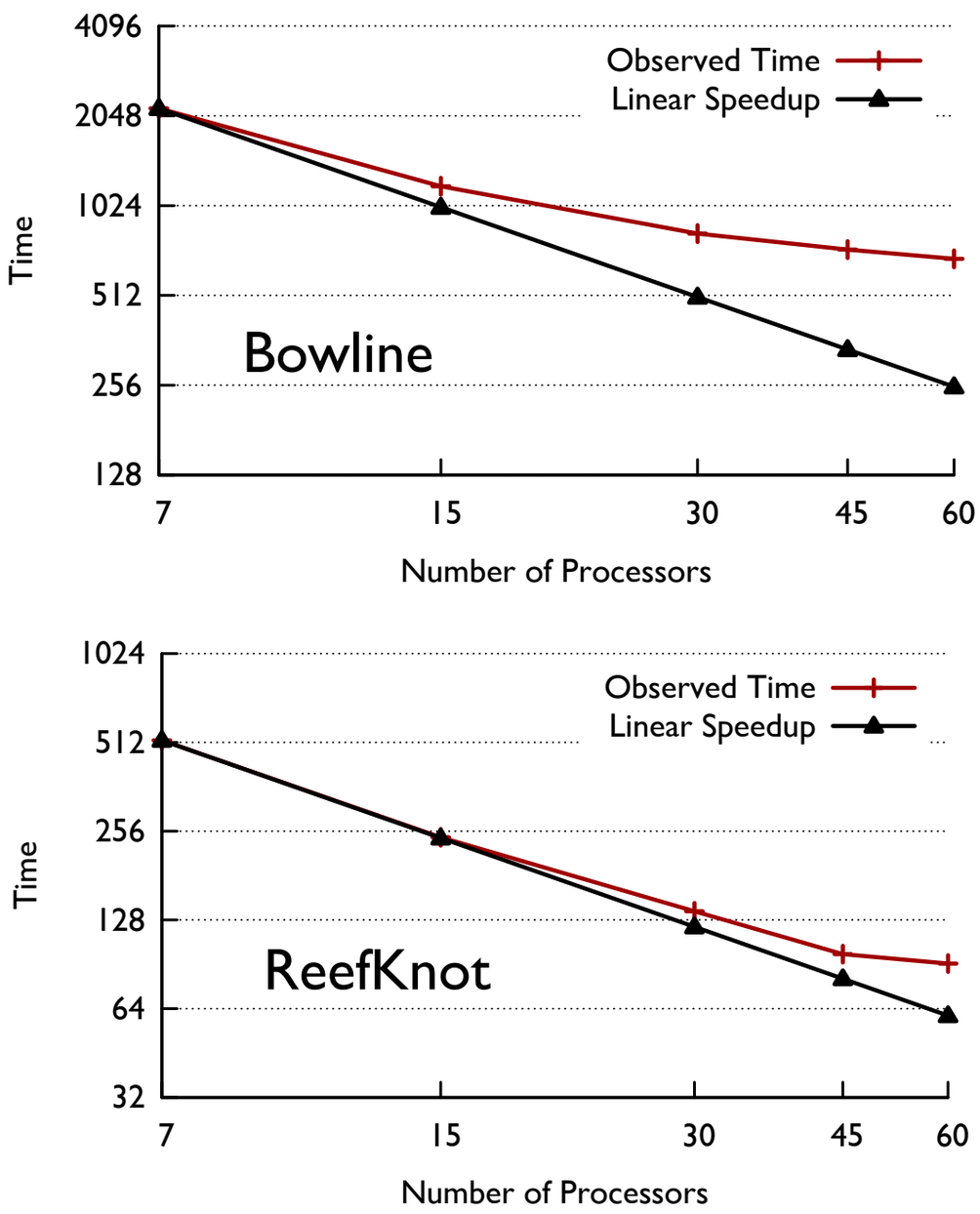


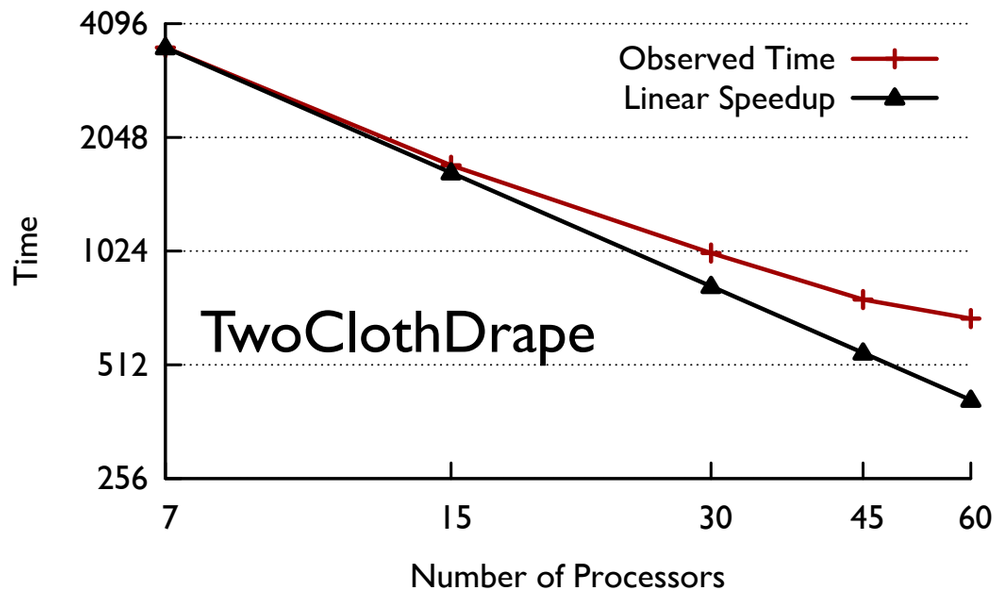
Without node-level phase barrier



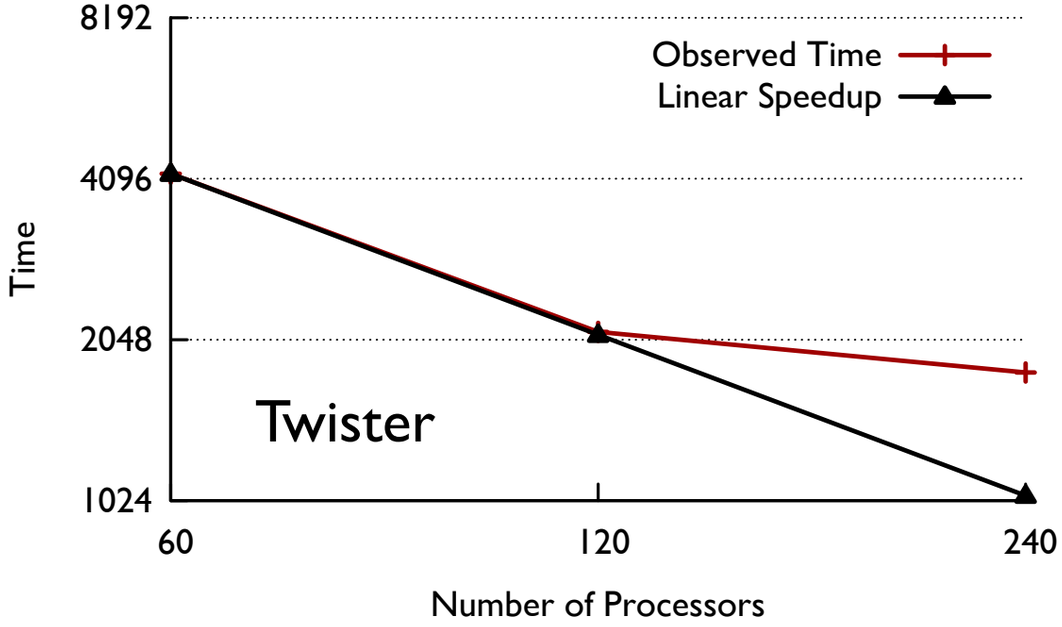
With node-level phase barrier

Results

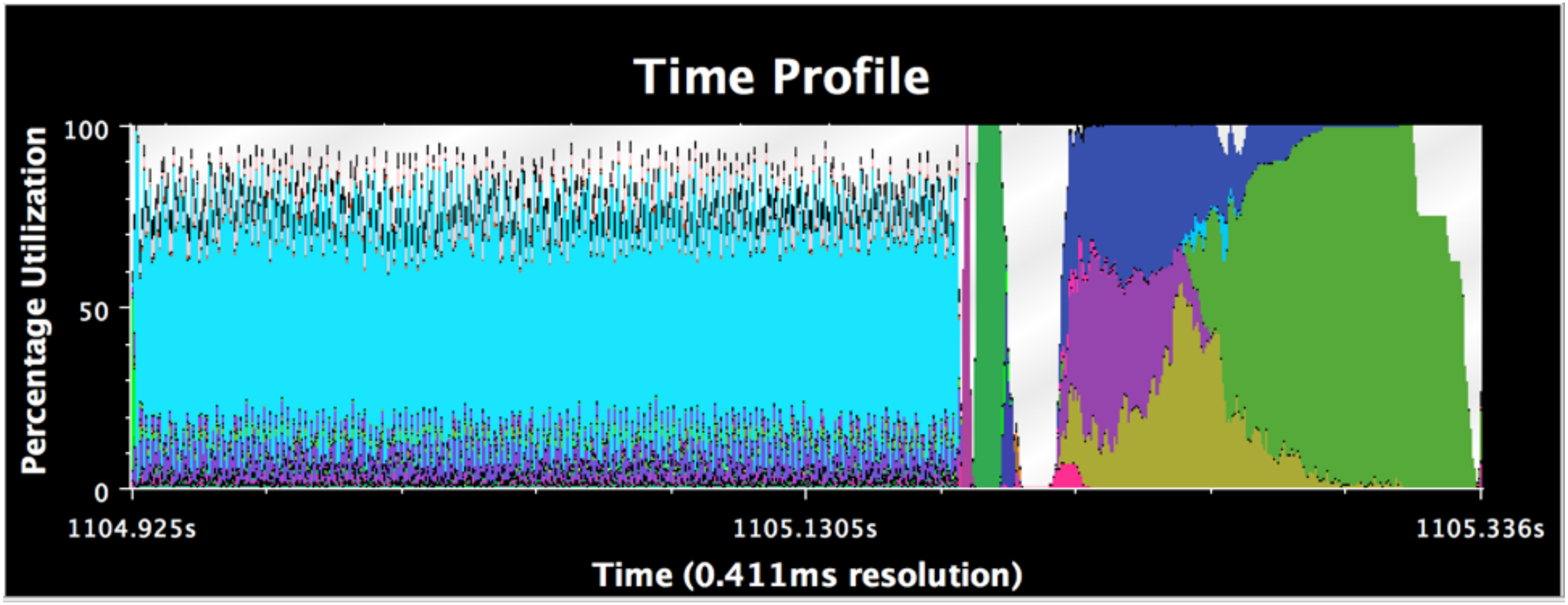

4096
2048
1024
512
256
128
Time
Observed Time
Linear Speedup
Bowline
7
15
30
45
60
Number of Processors


4096
2048
1024
512
256
Time
Observed Time
Linear Speedup
TwoClothDrape
7
15
30
45
60
Number of Processors

1024
512
256
128
64
32
Time
Observed Time
Linear Speedup
ReefKnot
7
15
30
45
60
Number of Processors


8192
4096
2048
1024
Time
Observed Time
Linear Speedup
Twister
60
120
240
Number of Processors

Future Work



Force Calculation

Broad Detection

Narrow Detection