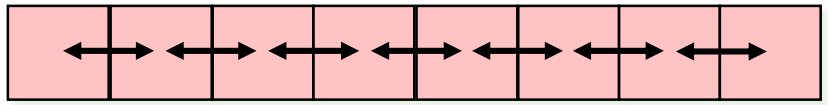
Parallelization

SAGE application

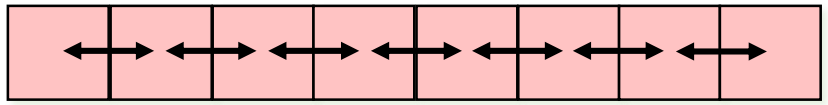
Mar 19, 2021

Compute vs. Communicate



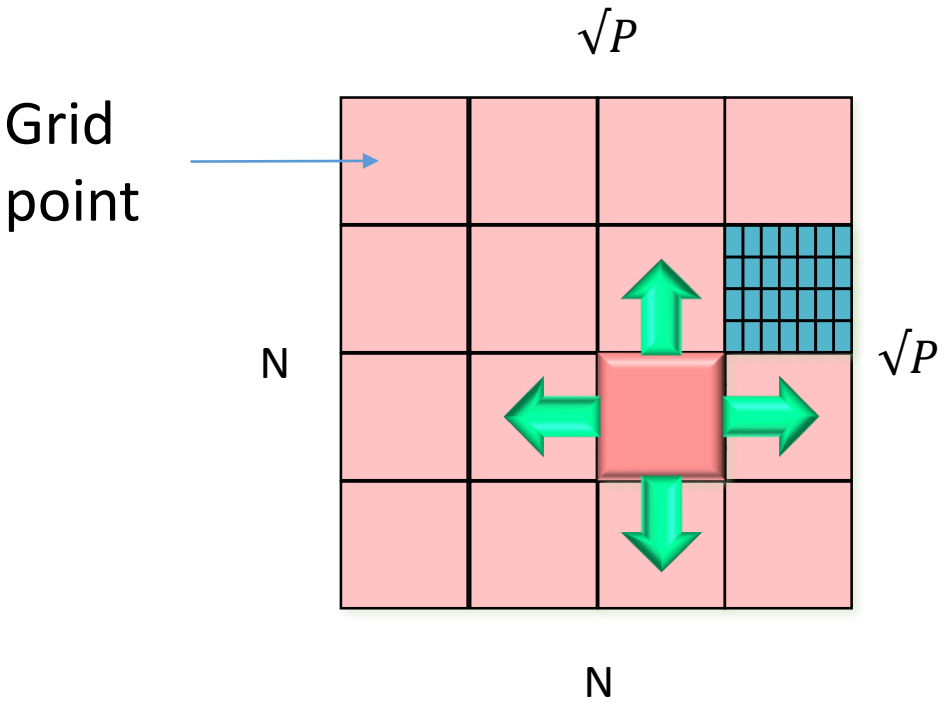
```
class for size in 200 2000 20000 200000 2000000 20000000 ; do for iter in `seq 1 3` ; do mpirun -np 20 -hosts csews11,csews20,csews42 ./nbsendrecv ${size} 200 ; done ; echo ; done
0.039241
0.057331
0.041966

0.220274
0.025506
0.036639

0.031834
0.033379
0.028718

0.072557
0.085644
1.068593

0.562996
1.174738
2.189748

4.797187
4.360296
4.546603
```

Compute vs. Communicate



```
class for commsize in 2 20 200 2000 20000 ; do for iter in `seq 1 3` ; do mpirun -ppn 4 -np 20 -hosts csews1,csews2,csews3,csews4,csews5 ./nbsendrecv 2000000 ${commsize} ; done ; echo ; done
0.325137
0.295568
0.267692

0.254105
0.321769
0.295561

0.254874
0.279193
0.302050

0.306233
0.307061
0.275115

0.339506
0.301106
0.302485
```

Domain Decompositions



2D domain

Halo exchange

- Each cell has some ghost regions
- Communication with neighbors

Communication to computation ratio

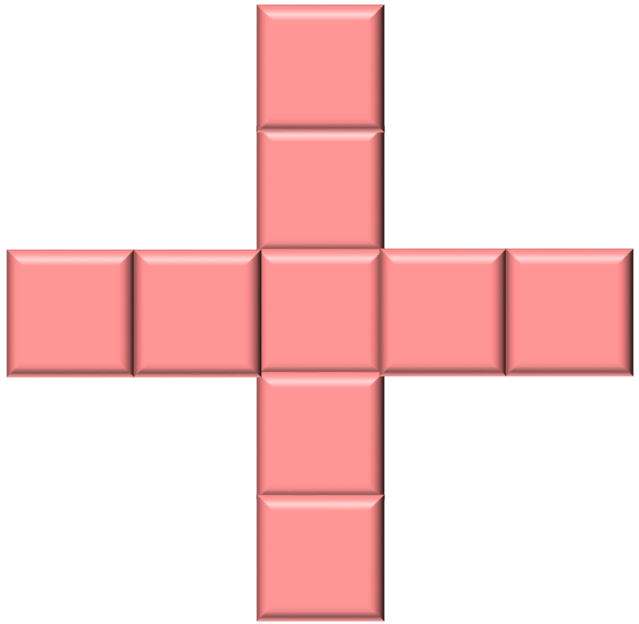
1D decomposition = $\frac{2N}{N^2/P}$

2D decomposition = $\frac{4N/\sqrt{P}}{N^2/P}$

Which is better?

ccr should be small for better performance

9-point stencil

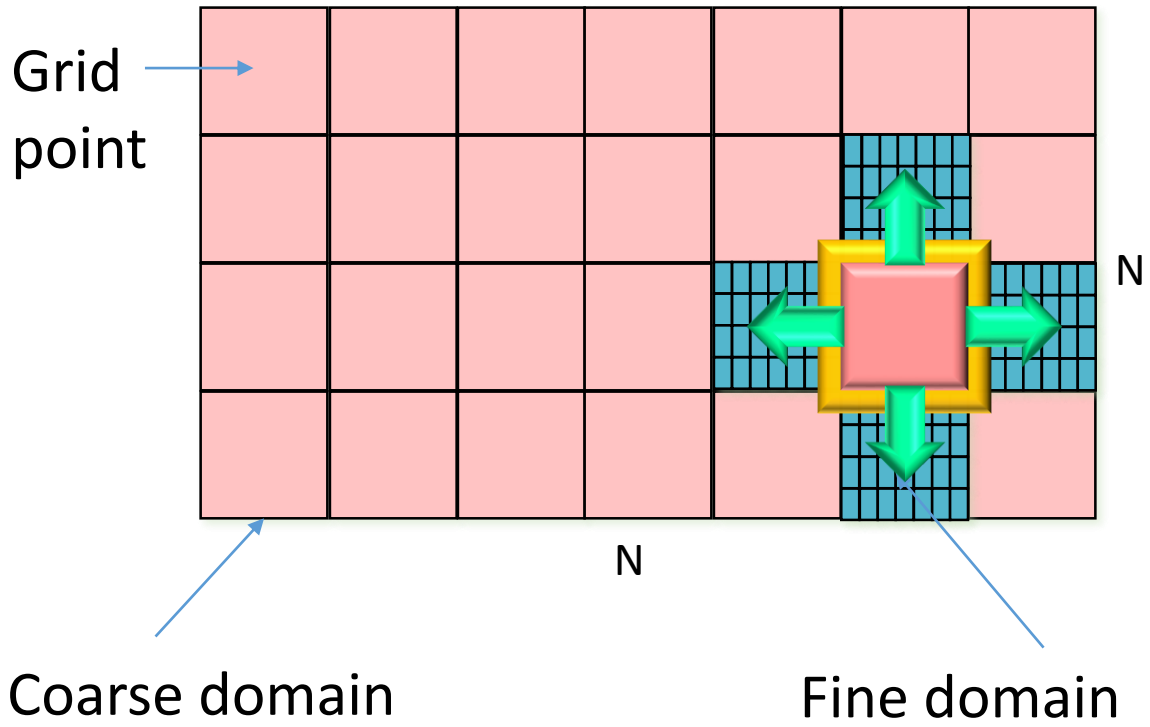


communication to computation ratio for

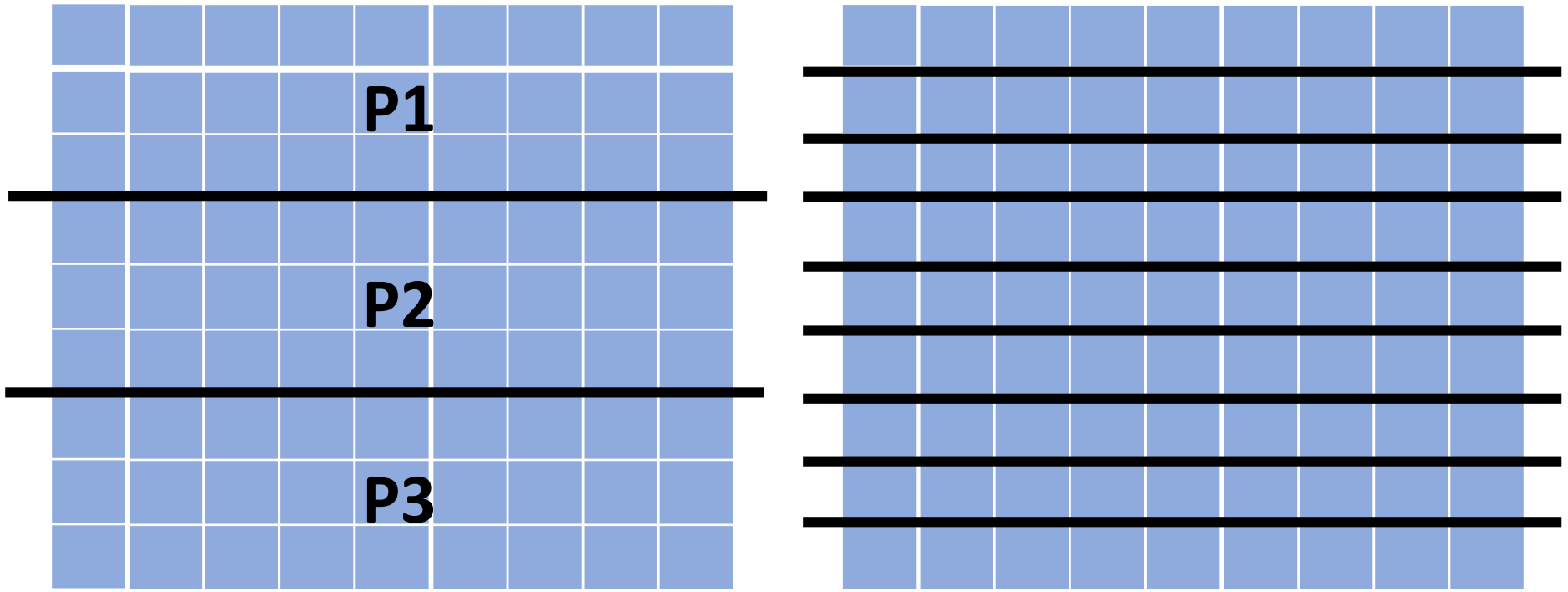
1D decomposition = $\frac{4N}{N^2/P}$

2D decomposition = $\frac{8N/\sqrt{P}}{N^2/P}$

Domain Refinement

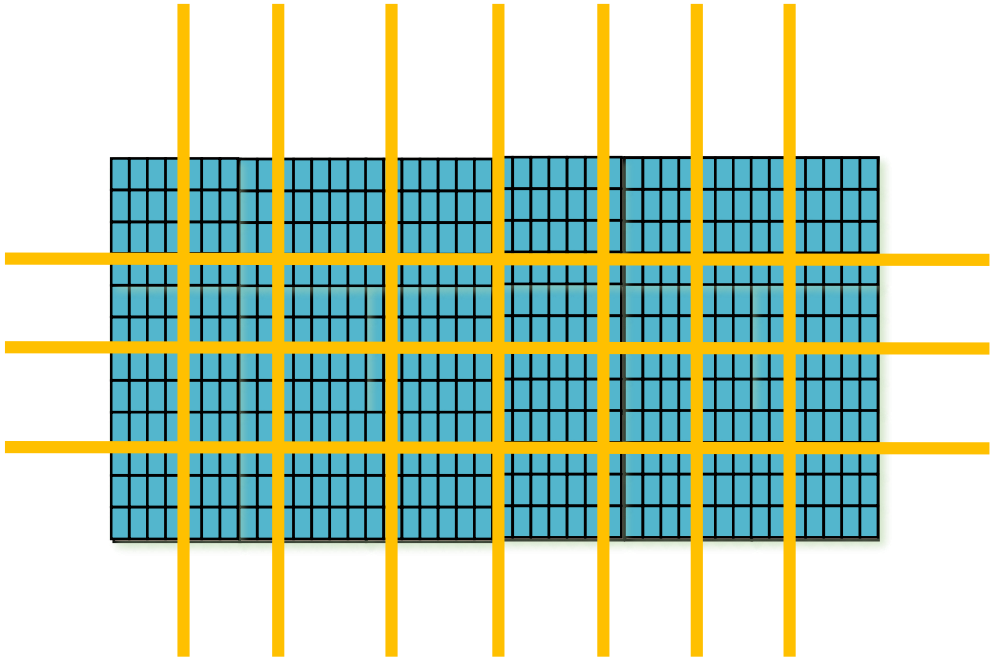


Granularity and Concurrency



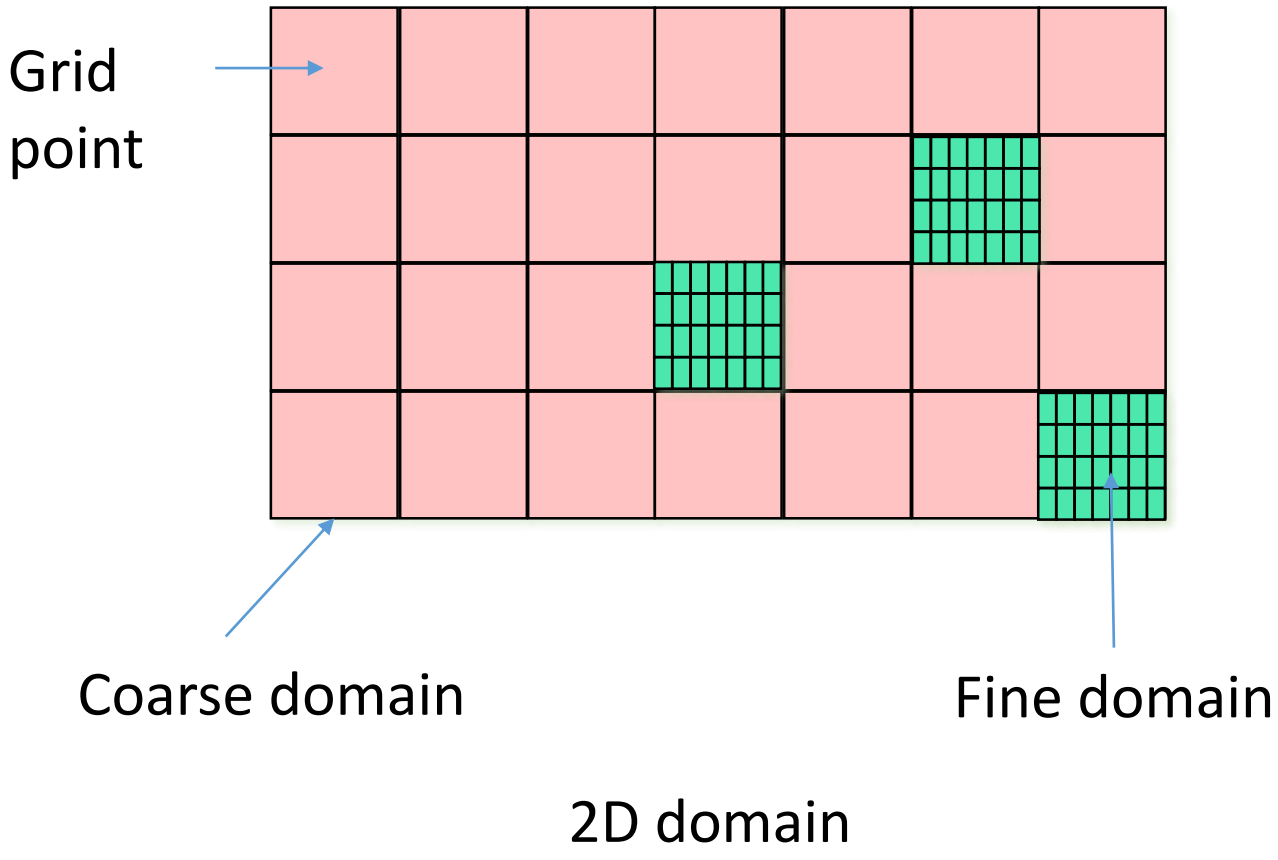
Decomposition granularity	Coarse-grained	Fine-grained
Degree of concurrency	Small	High

Over-decomposition



- There must always be sufficient work for a process!
- Balance between reducing communication volume and reducing computations/process such that processes do not idle
- Communication to computation ratio

Adaptive Mesh Refinement



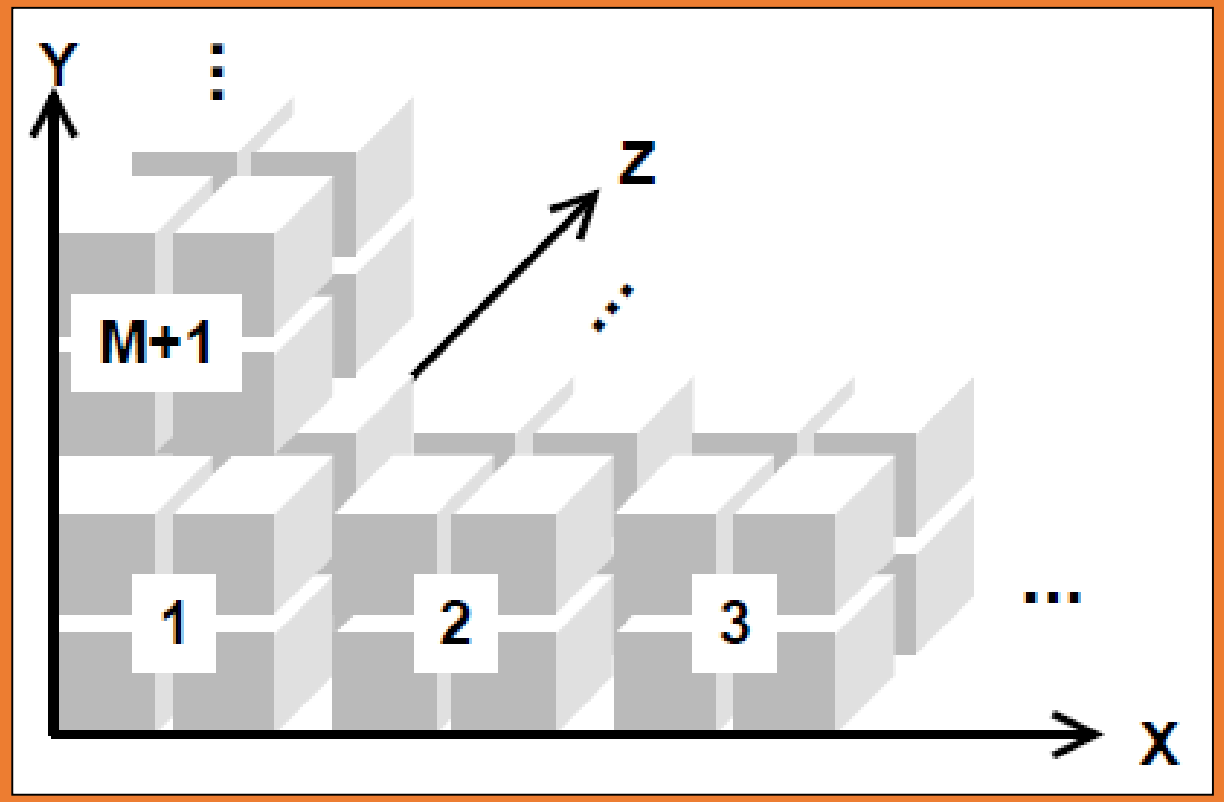
Q: Issue?
Load-imbalance

SAGE – An Example Application

- SAIC's Adaptive Grid Eulerian hydrocode
  - Multidimensional, multi-material hydrodynamics code
  - Uses AMR
  - Used for stockpile stewardship
- Parallel code, written in Fortran 90 and using MPI
  - 100,000+ LOC
- Spatial discretization of domain based on Cartesian grids
- X,Y,Z order assignment of grids/cells

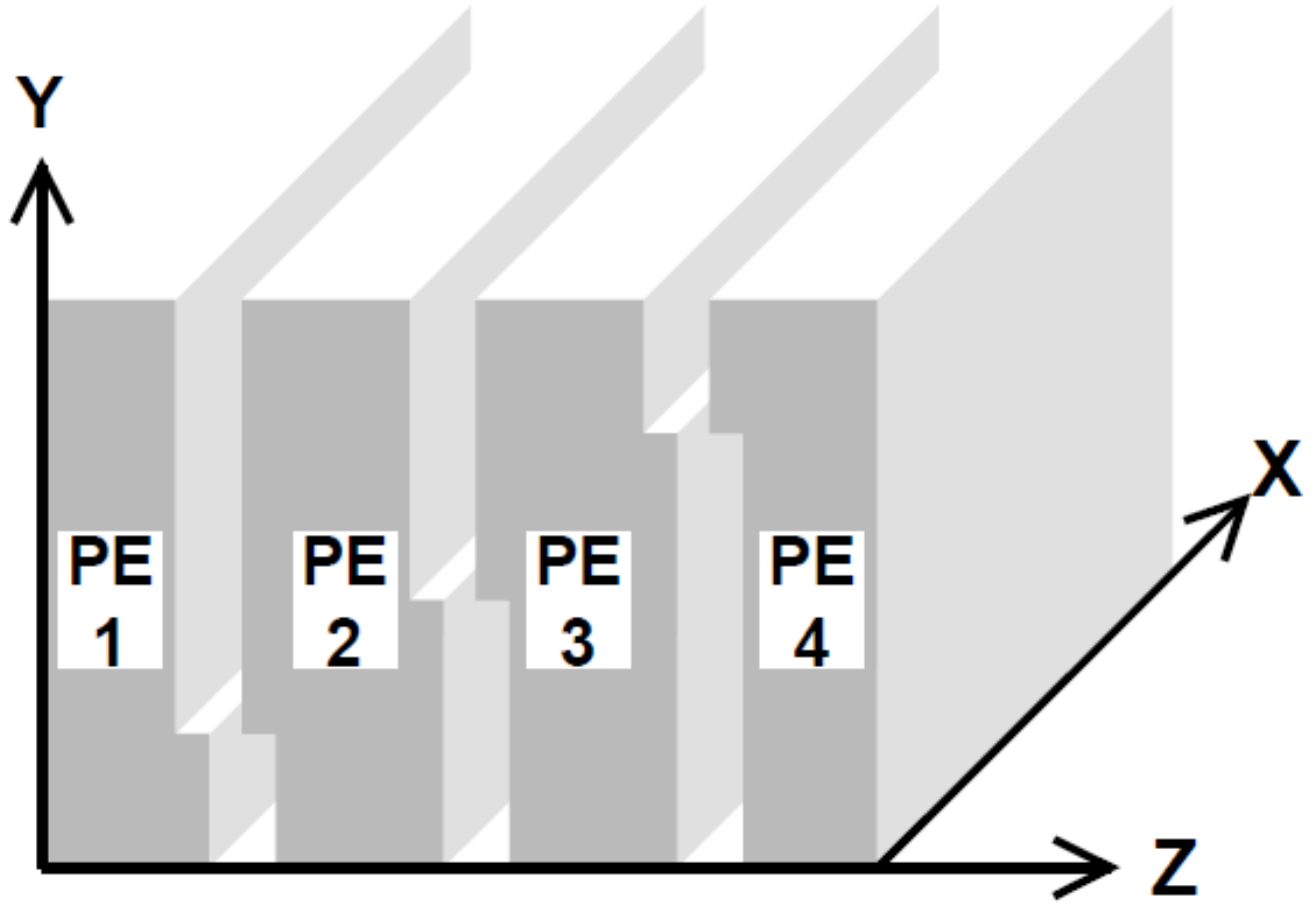
Kerbyson et al., "PREDICTIVE PERFORMANCE AND SCALABILITY MODELING OF A LARGE-SCALE APPLICATION", SC 2001

Domain Decomposition

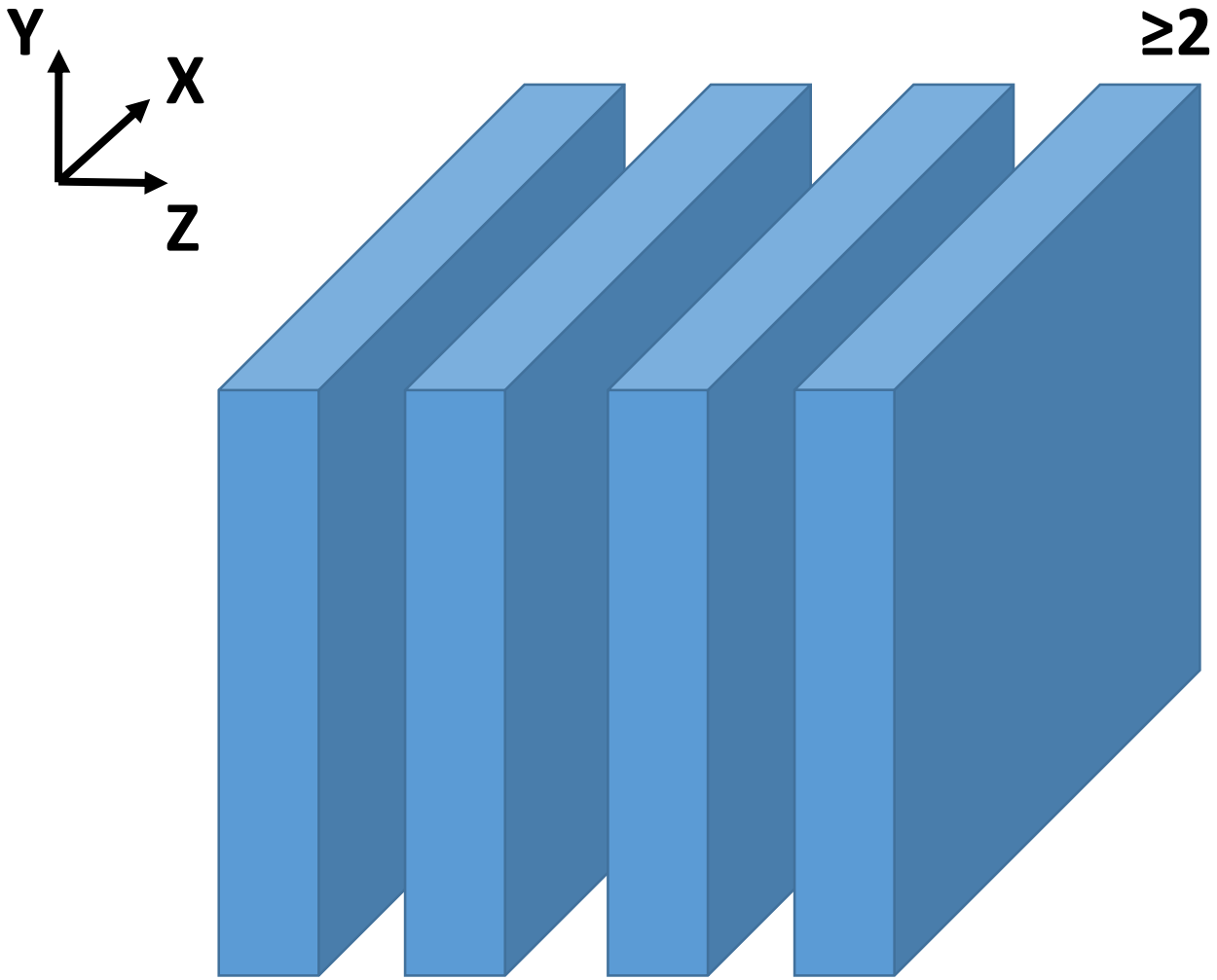


2x2x2

Process Assignment



Slab Decomposition



Example:

16 x 16 x 16 3D domain on 4 PEs
➔ 512 2 x 2 x 2 blocks
➔ 128 2 x 2 x 2 blocks / PE
➔ 16 X 16 surface area

CCR: Surface to volume ratio

Decomposition

- E cells per process, P processes
- Volume (#cells) total = E.P
- Volume of each sub-grid = E = w x L^2 (Assume square grid in XY)
- Total data communicated between PEs $\propto L^2$

Estimation

- E cells per process, P processes
- Volume (#cells) of entire grid = L^3 = E . P = (w x L^2) . P
- #Foils of width 2 per PE f = (L/2)/P

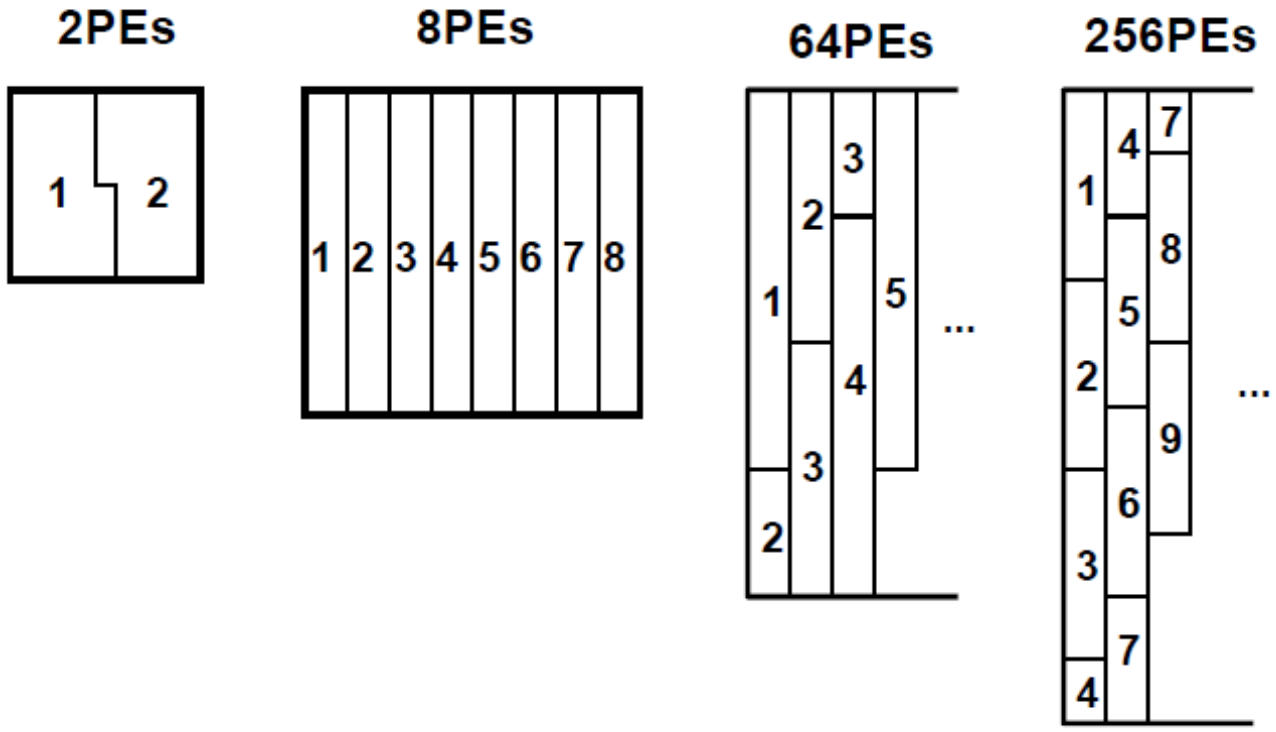
$$= \sqrt[3]{E.P}/2\text{P}$$

$$= \sqrt[3]{E/8P^2}$$

$$= \sqrt[3]{(E/8)/P^2}$$

➔ $\text{P} > \sqrt{\frac{E}{8}} \Rightarrow f < 1 \Rightarrow$ Communication with more #processors in Z

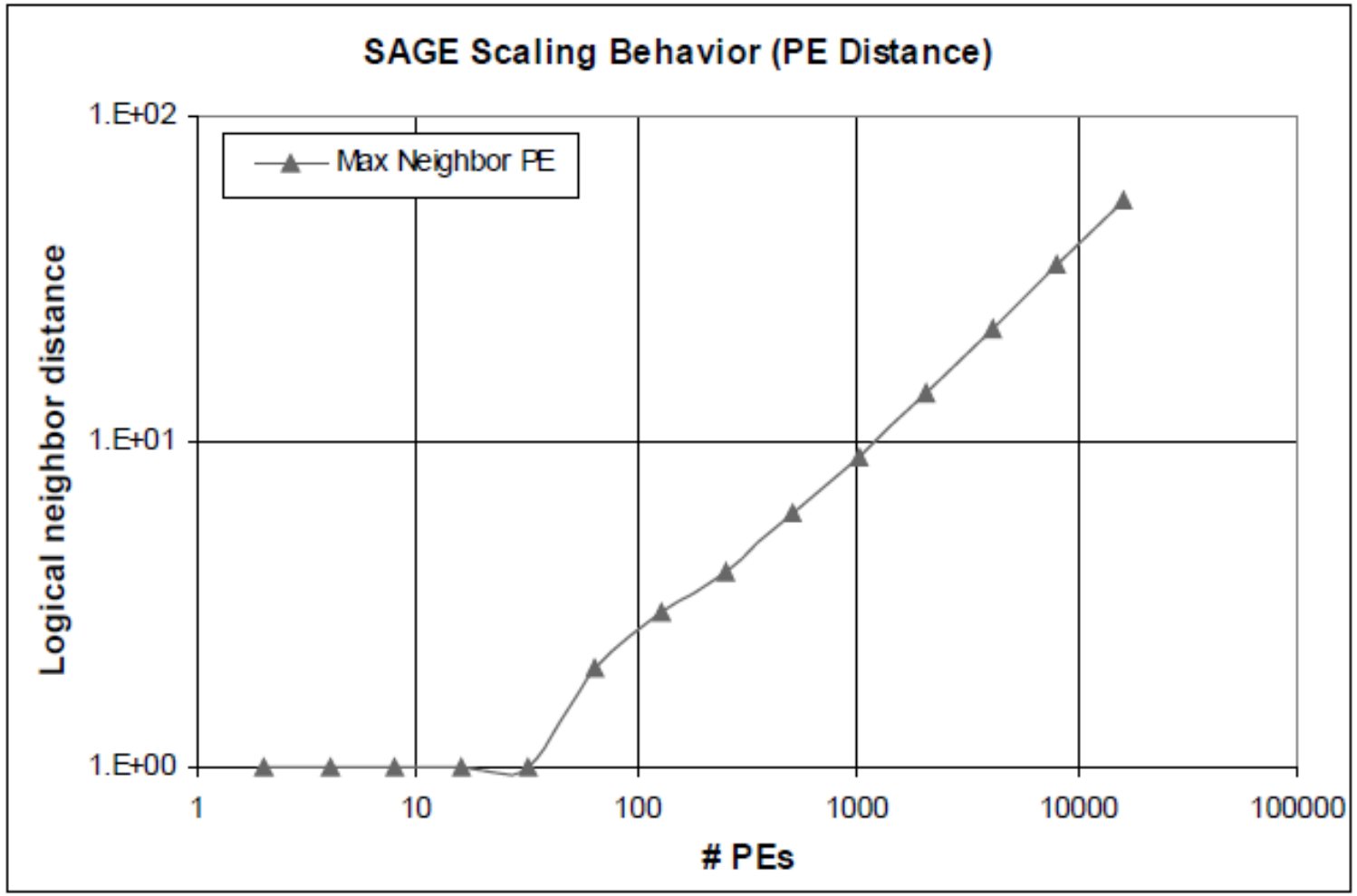
Assignment



Issue?

Max. logical PE distance = 1,1,2,4

Logical Neighbor Distance



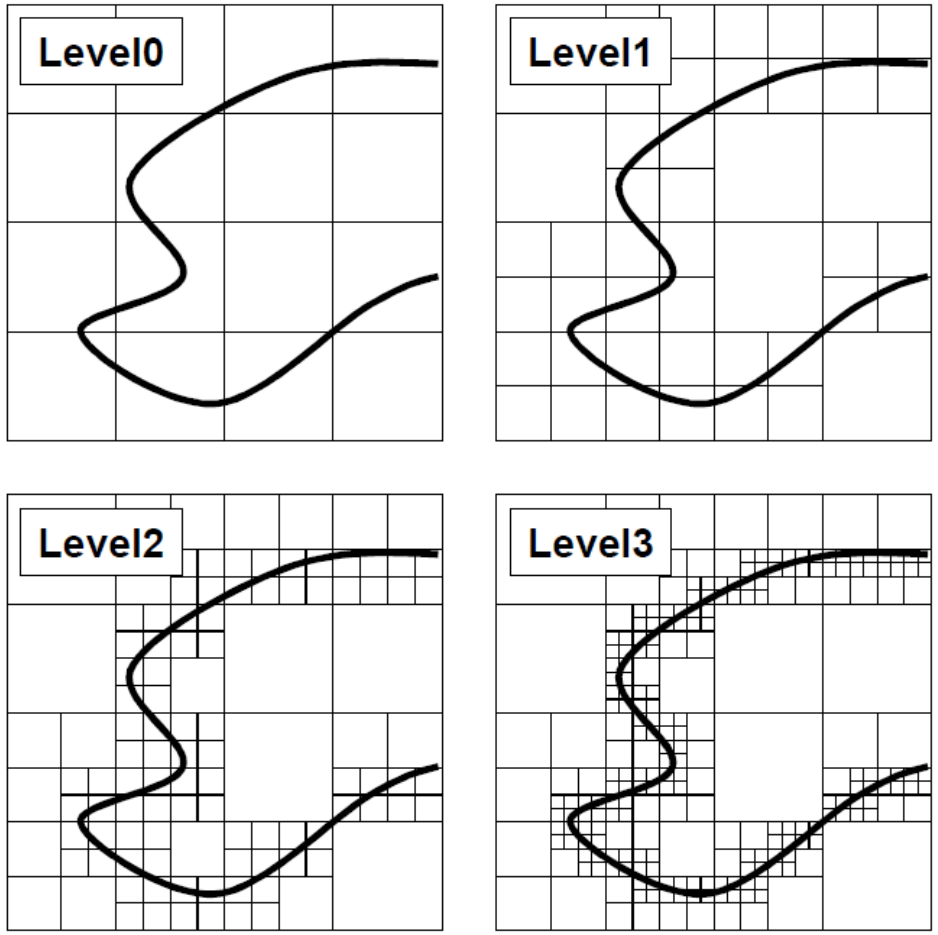
Communications in SAGE

Different types: gather, scatter, allreduce, bcast

- Few bcast during the initialization phase
- 120 allreduce (4 bytes) per time step
- Gather and scatter: require inter-processor boundary data
- 1 time step involves:
  - Gather (in all dimensions)
  - Compute
  - Scatter (in all dimensions)
  - 160 real and 17 integer in GS

AMR

- The decision on whether to split or combine cells is determined by the current cell values in the calculation being performed.
- AMR enables more refined calculations to take place in those areas of the spatial grid characterized by more intense physical phenomena.
- A cell at level n is 8^n times smaller



Load Balance

- The adaptive refinement of cells can result in load imbalance
  - A large degree of activity in the refined region
- Load balance at the end of a cycle when the maximum number of cells on any processor is greater than 10% above the average
- Partition the domain (total #cells) into equal sized segments
- Requires communication of data from owned processes to the new processes

Performance of SAGE

$$T_{cycle}(P, E) = T_{comp}(E) + T_{memcon}(P, E) + T_{GScomm}(P, E) + T_{allreduce}(P)$$

$$T_{GScomm}(P,E) = C(P,E)\begin{pmatrix} 160.T_{comm}(Surface_Z.MPI_{Real8},P)+ \\ 17.T_{comm}(Surface_Z.MPI_{INT},P)+ \\ 160.T_{comm}(Surface_Y.MPI_{Real8},P)+ \\ 17.T_{comm}(Surface_Y.MPI_{INT},P)+ \\ 160.T_{comm}(Surface_X.MPI_{Real8},P)+ \\ 17.T_{comm}(Surface_X.MPI_{INT},P) \end{pmatrix}$$

contention on the processor network when using *P* processors due to distant processor neighbor communications

Performance of SAGE

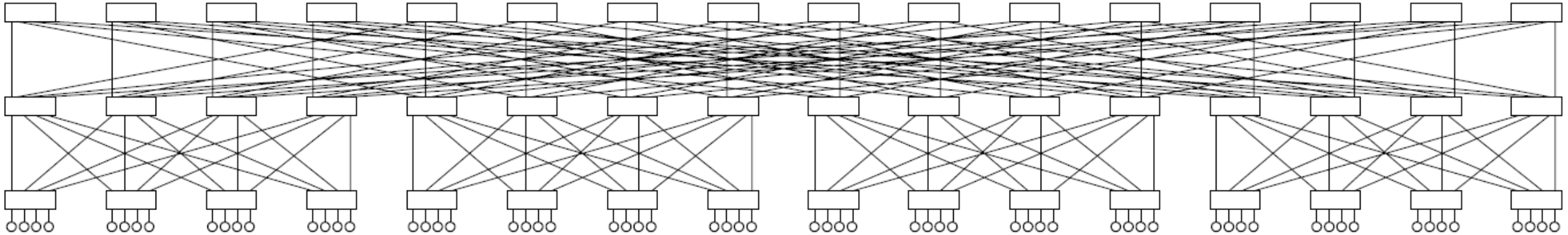
$T_{cycle}(P, E) = T_{comp}(E) + T_{memcon}(P, E) + T_{GScomm}(P, E) + T_{allreduce}(P)$

$$T_{GScomm}(P,E) = C(P,E)\begin{pmatrix} 160.T_{comm}(Surface_Z.MPI_{Real8},P)+ \\ 17.T_{comm}(Surface_Z.MPI_{INT},P)+ \\ 160.T_{comm}(Surface_Y.MPI_{Real8},P)+ \\ 17.T_{comm}(Surface_Y.MPI_{INT},P)+ \\ 160.T_{comm}(Surface_X.MPI_{Real8},P)+ \\ 17.T_{comm}(Surface_X.MPI_{INT},P) \end{pmatrix}$$

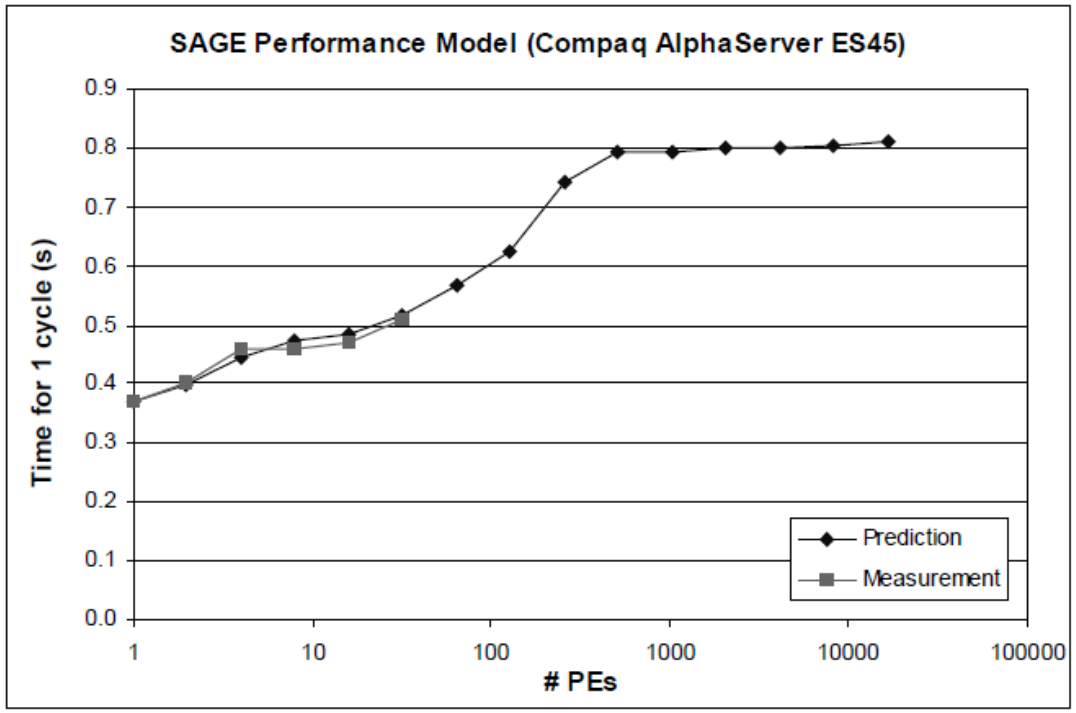
contention on the processor network when using *P* processors due to distant processor neighbor communications

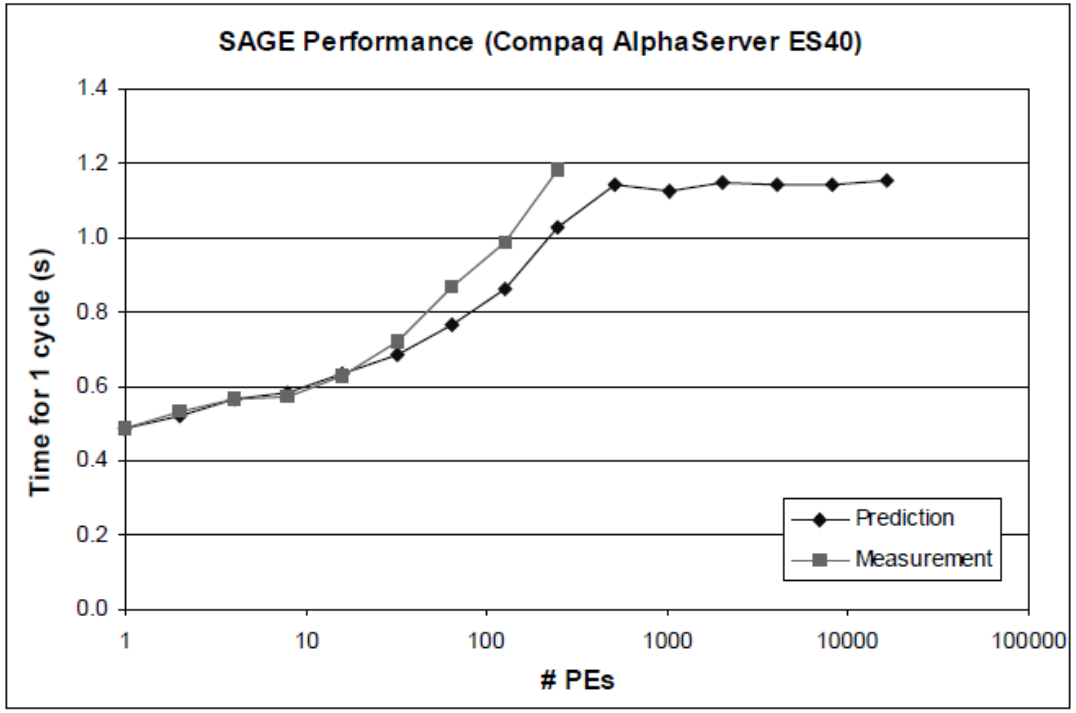
$$L_c(S,P)+S.\frac{1}{B_c(S,P)}$$

Fat-tree

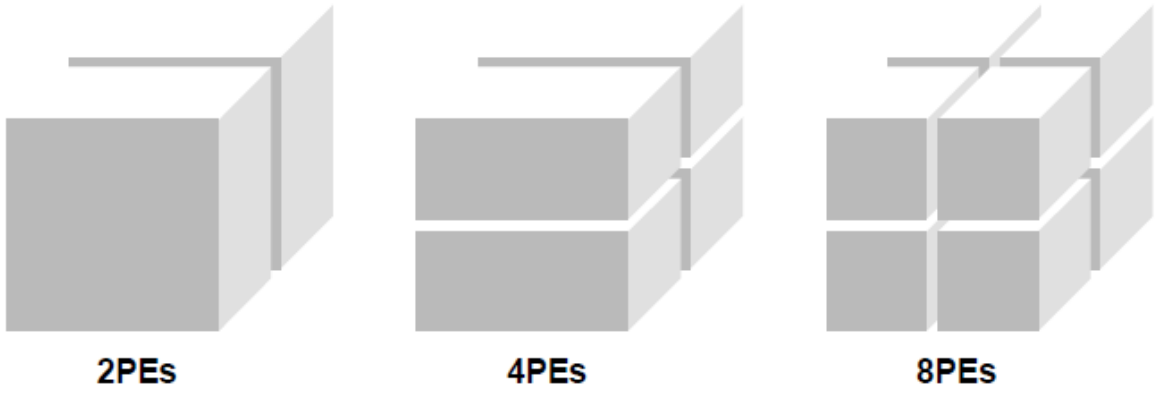


Prediction vs. Measurement

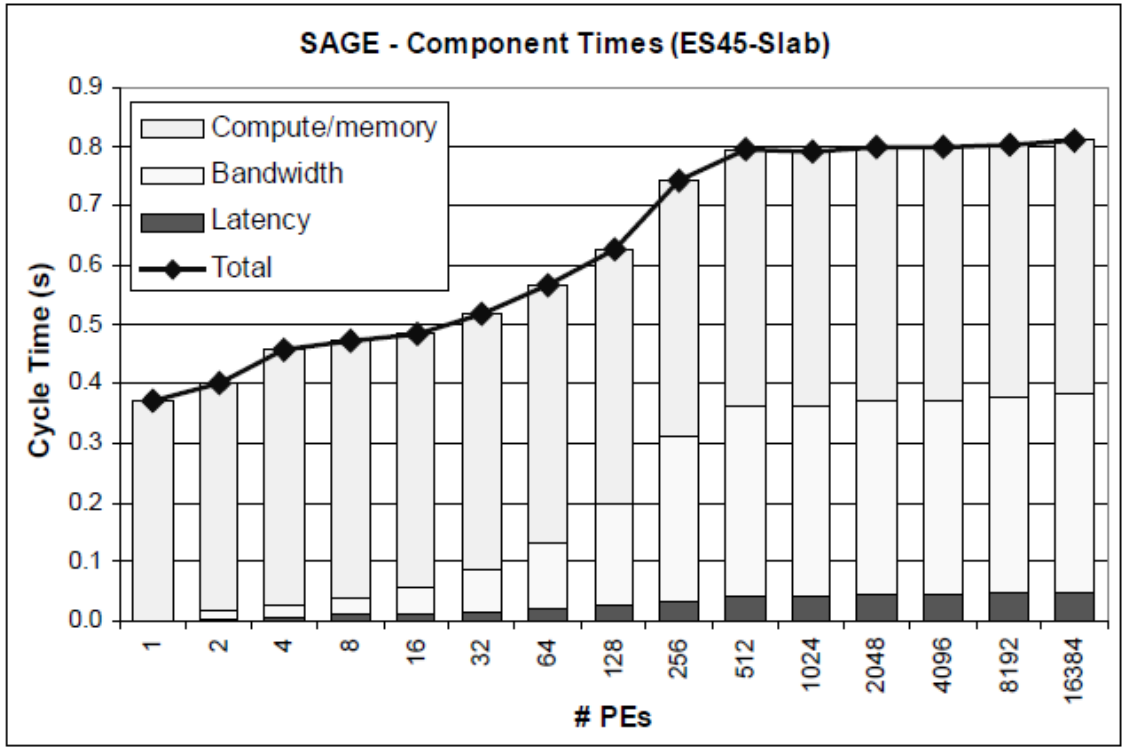


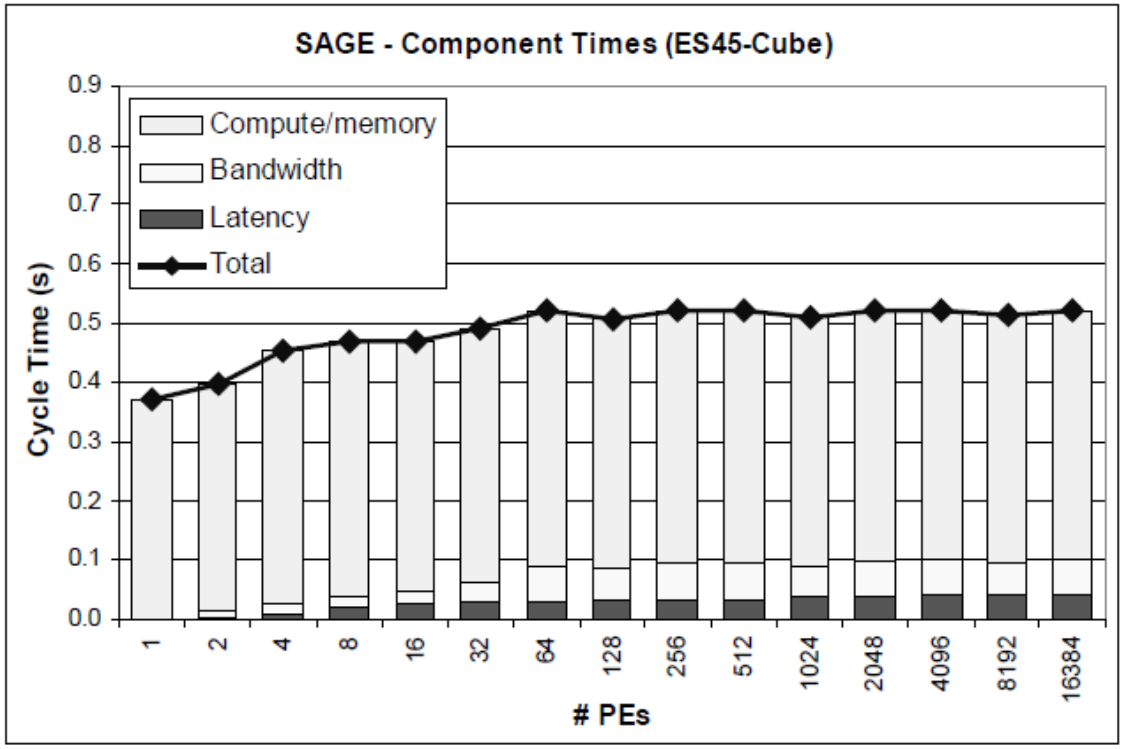


Cube Decompositions

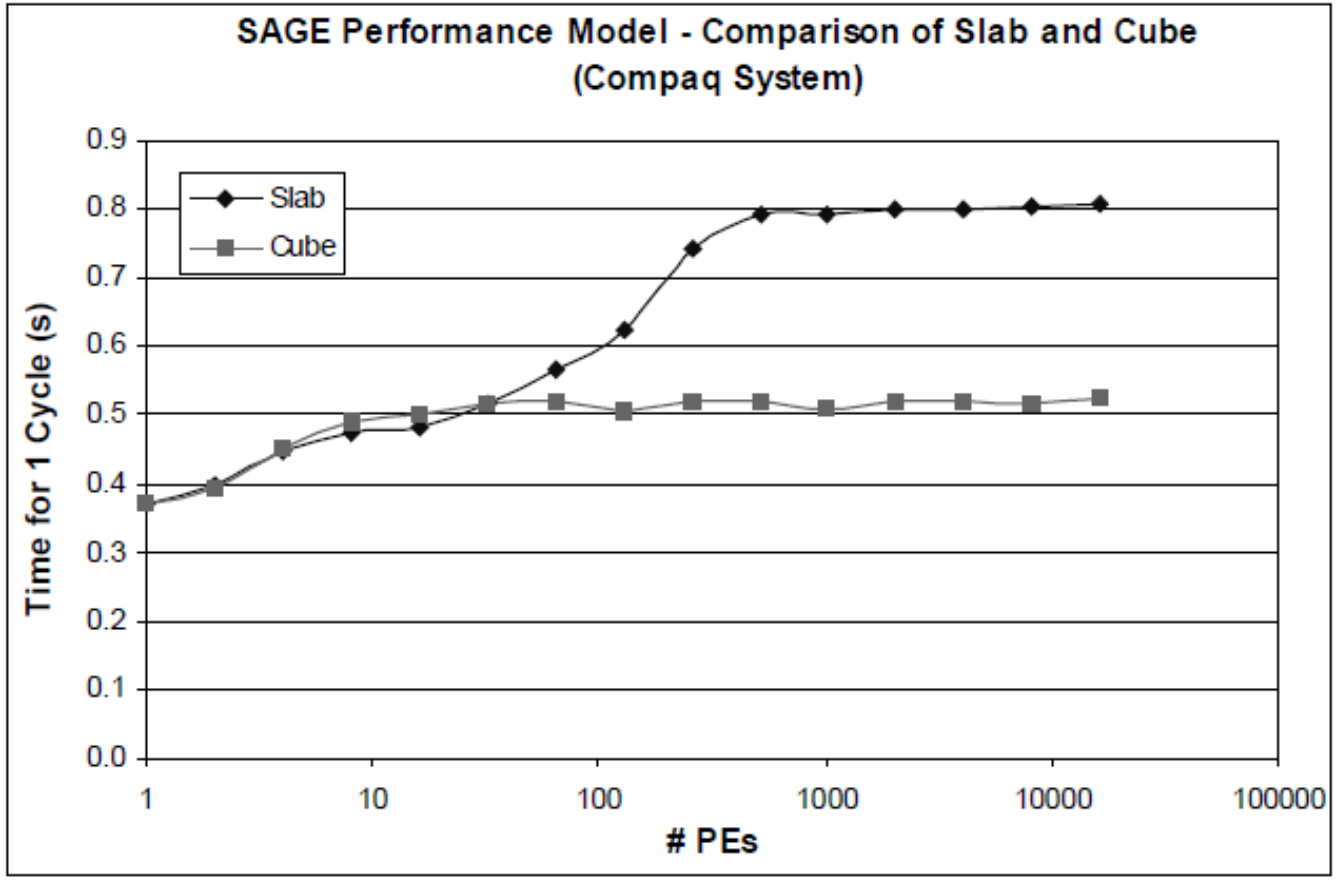


Slab vs. Cube Component Times

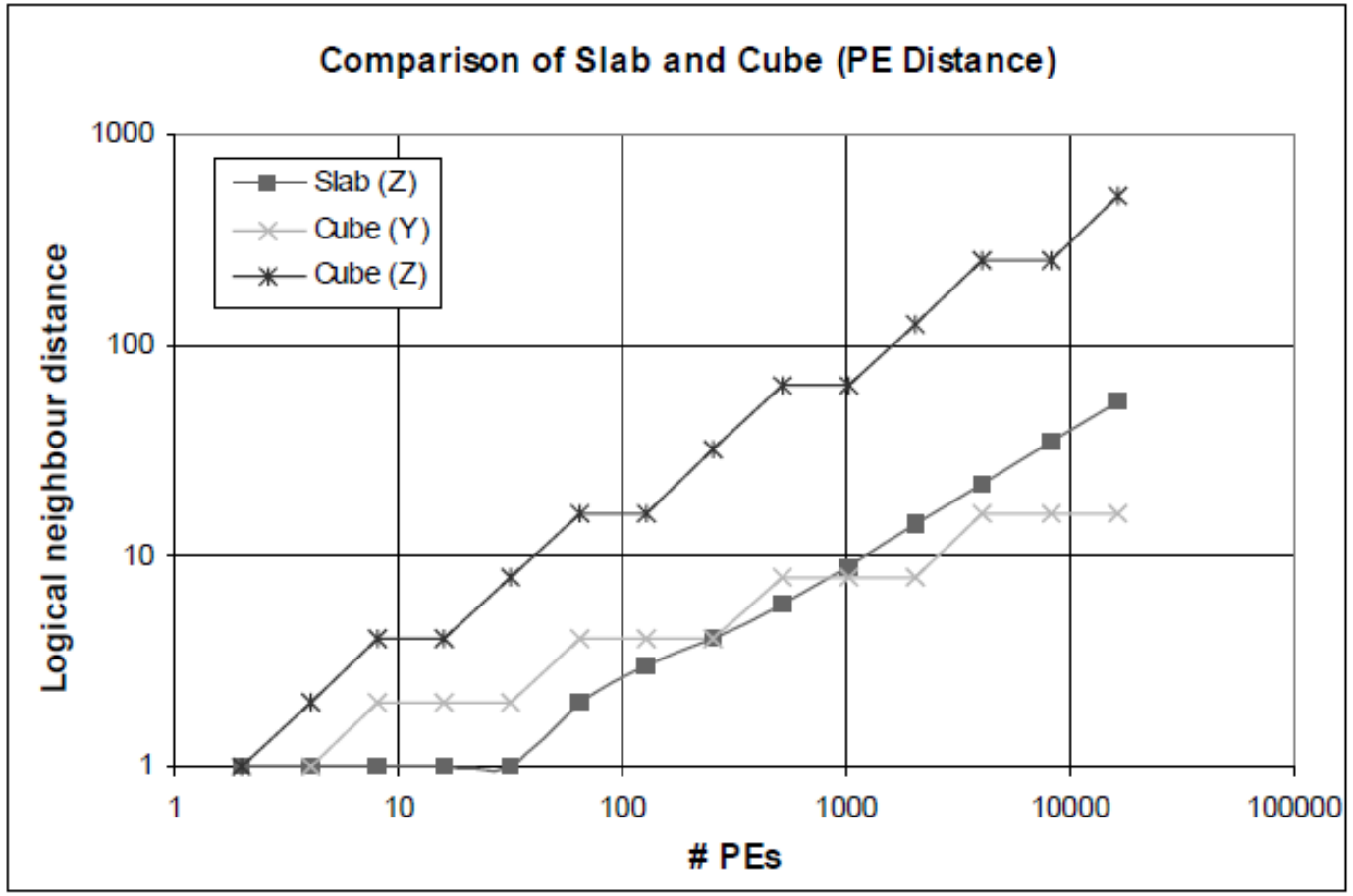




Prediction Results



Slab vs. Cube PED



THANKS