



Chris Goodyer* Justin Luitjens Curtis Hamman
Martin Berzins, Steve Parker Tom Henderson Mike Kirby

Possible and Impossible (?) Applications Scalability

Examples from CFD
EHL code , NS Solver
Adaptive Mesh Refinement
Theory Uintah code

*CPDE Unit Leeds
School of Computing
University of Leeds



Acknowledge DOE ASC and EPSRC (UK)

IDC HPC User Forum 03



MPI Scaling - Never-ending Research



There are a number of tough important problems that are the constant topic of research on clustered systems using MPI message passing paradigms.

THE QUESTION IS:

ITS BEEN DECADES, WHY HAVEN'T THEY BEEN SOLVED?

ANSWER:

MAYBE THE ARCHITECTURE or PARADIGM IS SIMPLY WRONG

Dynamic load balance

Rezoning

Changing work per step

Shared Memory

simple

simple

Clusters

Impossible?

Impossible?

The Transpose Problem

Moderate 3D FFT

Climate Modeling

simple

simple

Impossible?

Impossible?

THE SIMPLE FACT IS:

SHARED MEMORY PROGRAMMING WORKS EASILY

CLUSTERS ARE OFTEN ILL SUITED FOR HPC



Prospects for CFD on Petaflops* [Keyes Kaushik , Smith 97]

Algorithms developers must:

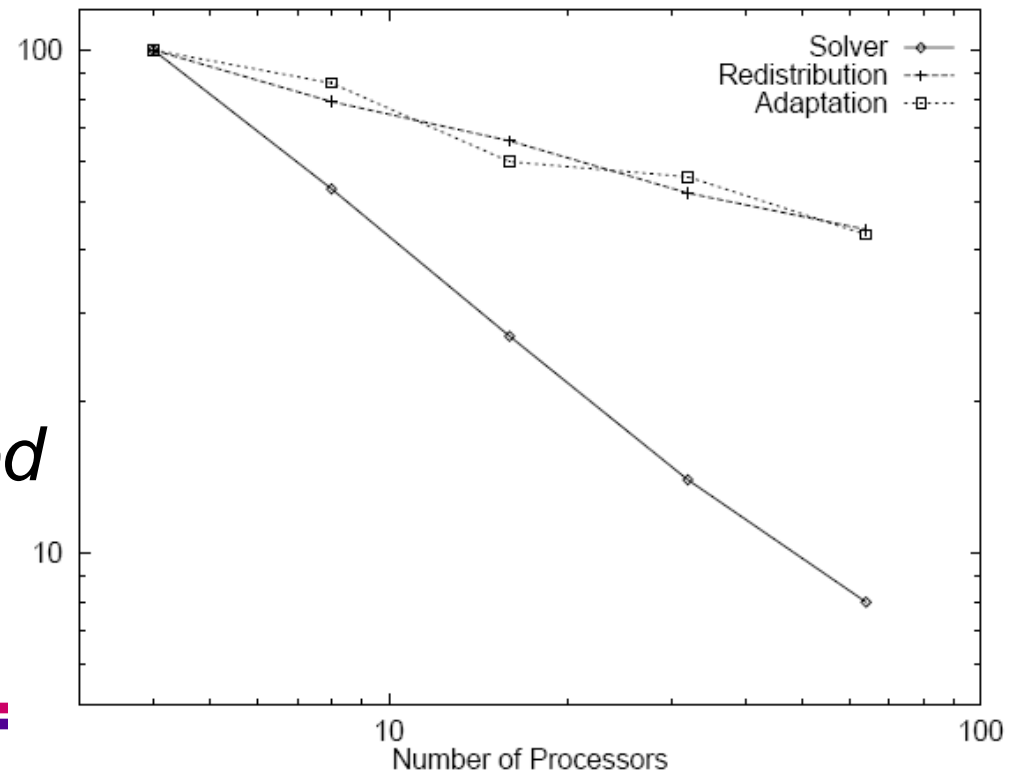
- learn to think natively in parallel
- Make choices informed by architecture
- understand that portable codes will be self tuning

Impossible Problems ?

Dynamic Rezoning

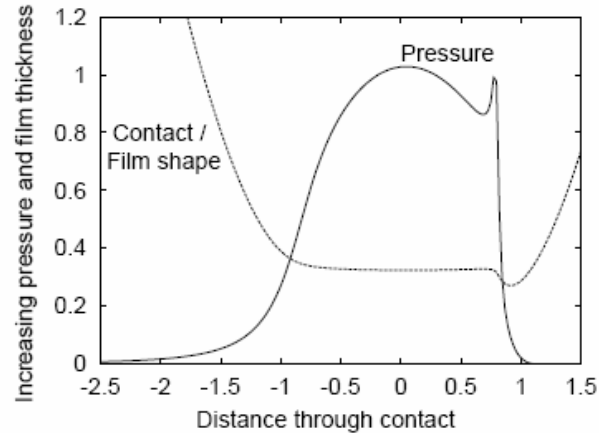
[Selwood , Berzins]

*Wissink et al. improved
on this in 2005*

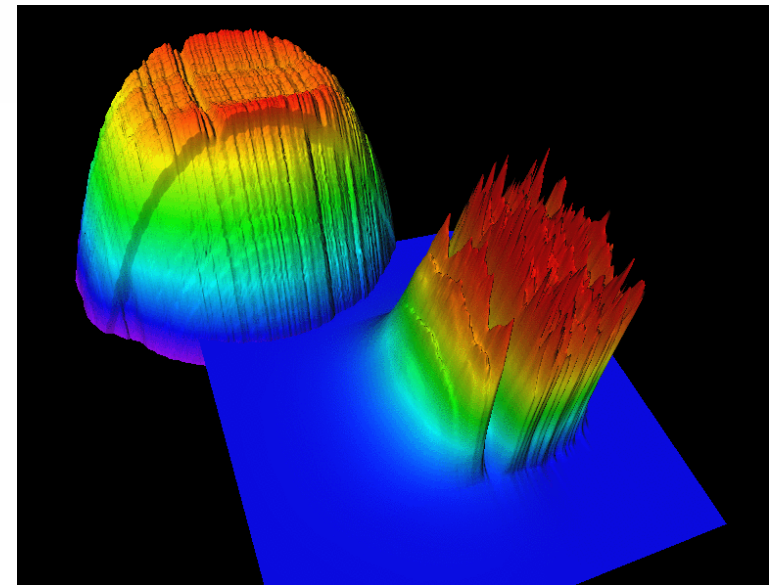
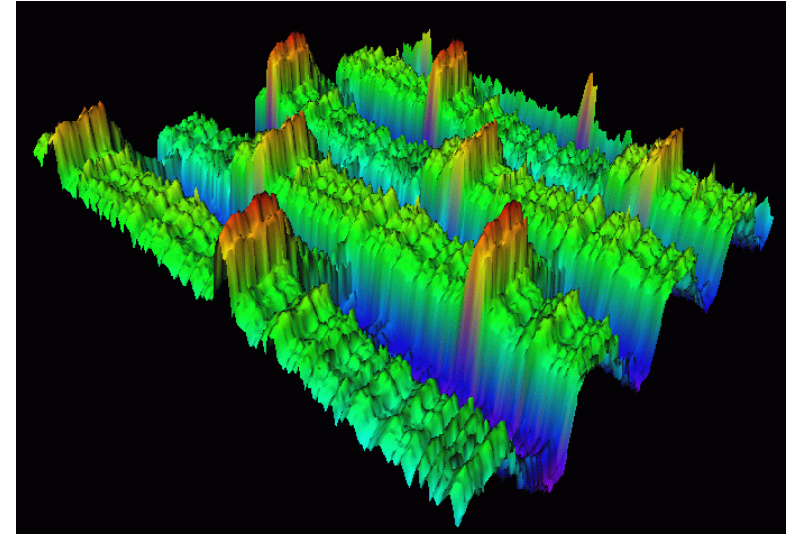


Measured Surface Roughness

Geometry and Pressure plots across an Elasto-hydrodynamic Lubrication contact



Undeformed contact is semi-circular

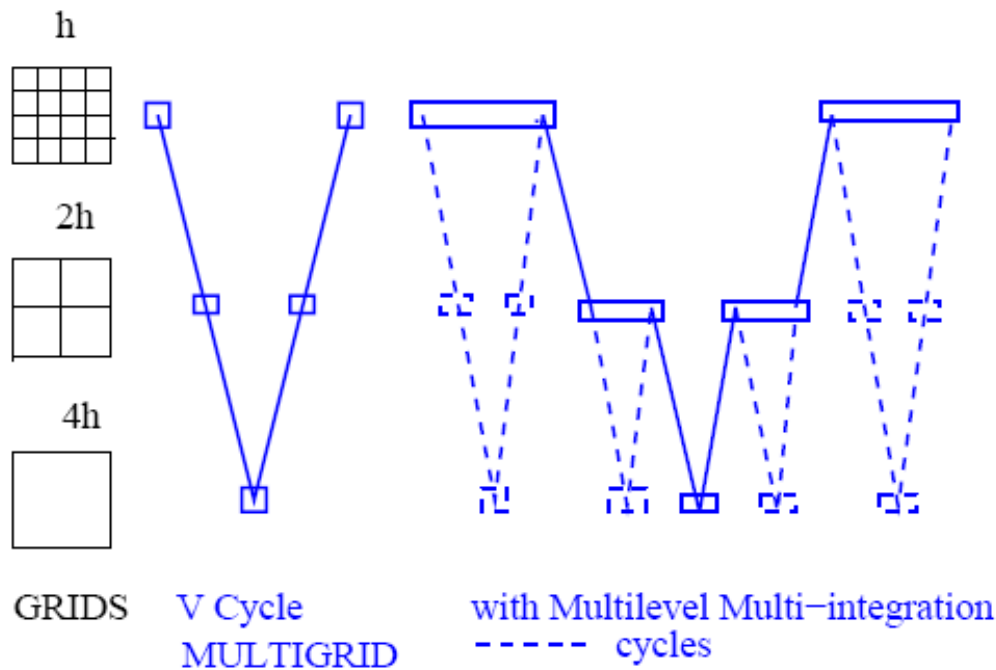


Computed Contact Shape

Computed
pressure profile
256M dense non-linear
equations time-dependent

ELASTO-HYDRODYNAMIC LUBRICATION

Multi-Level Multi-Integration Two Coarse Grids



*Reynolds Equation
2D Elliptic-hyperbolic
PDE in P and H*

*Point value of
film thickness, H
depends on all
pressures P*

$$H_{i,j} = H_{00} + \frac{X_i^2}{2} + \frac{Y_j^2}{2} - \frac{2}{\pi^2} \sum_{k=1}^{N_X} \sum_{l=1}^{N_Y} K_{i,j,k,l} P_{k,l}$$

*Form pressure sum on coarse grid and
add correction for “nearby” points*



*Reduces summation from $N*N$ to $N \log N$*

Multigrid/MLMI Computation and Comms Costs

Coarsest grid MLMI sum M^4 + broadcasts $M \times M$

Multigrid fine mesh $N \times N$ stripwise decomposed

MG/MLMI serial cost $N^2 + (N \log N)^2 + M^4$

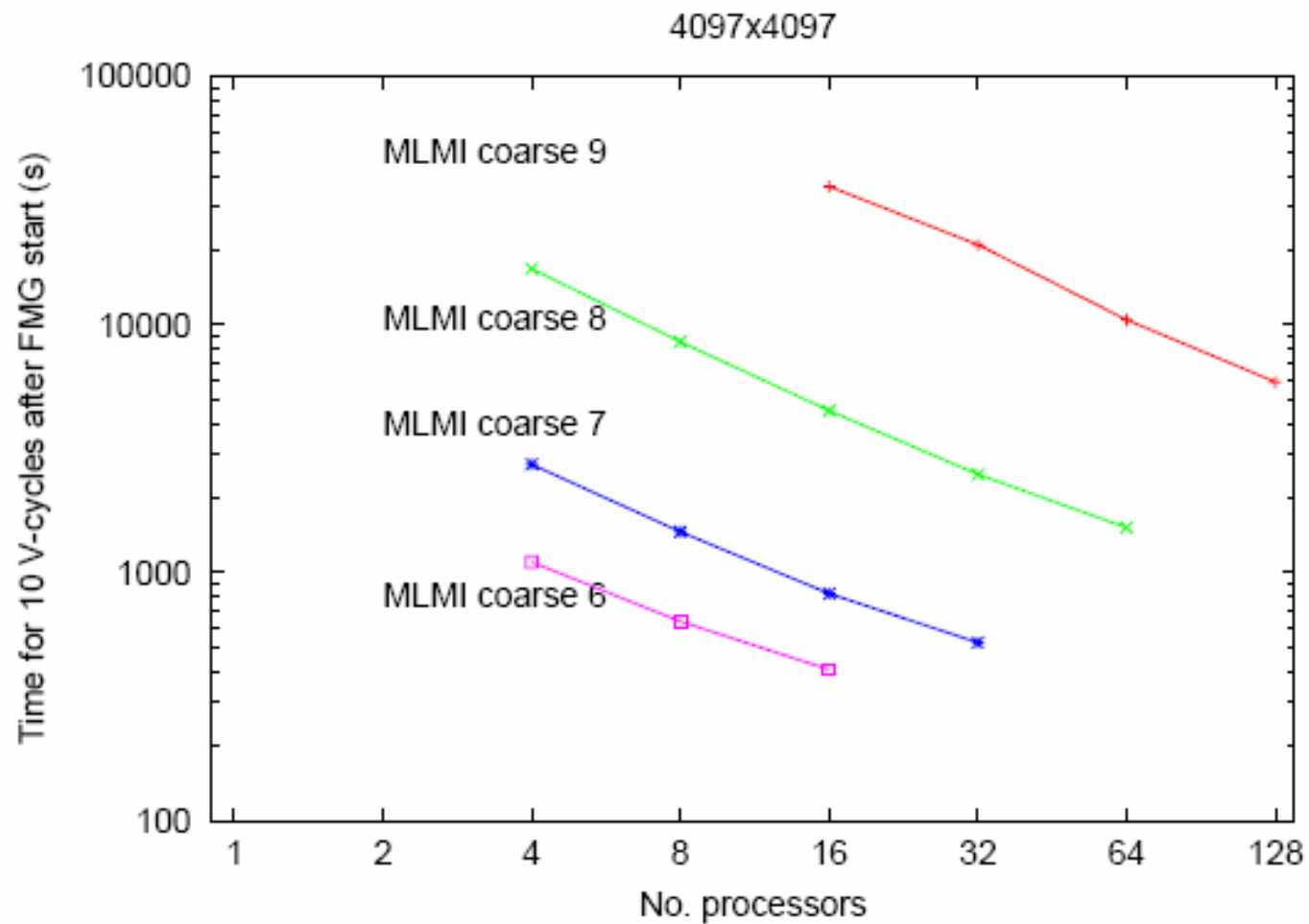
MG/MLMI communications costs

$N \log p + N + M \log p + M^2 p \log p$

With a multiplicity of constants and less significant terms.....



Parallel Timings - Vary MLMI on Grid 11



STRONG SCALABILITY

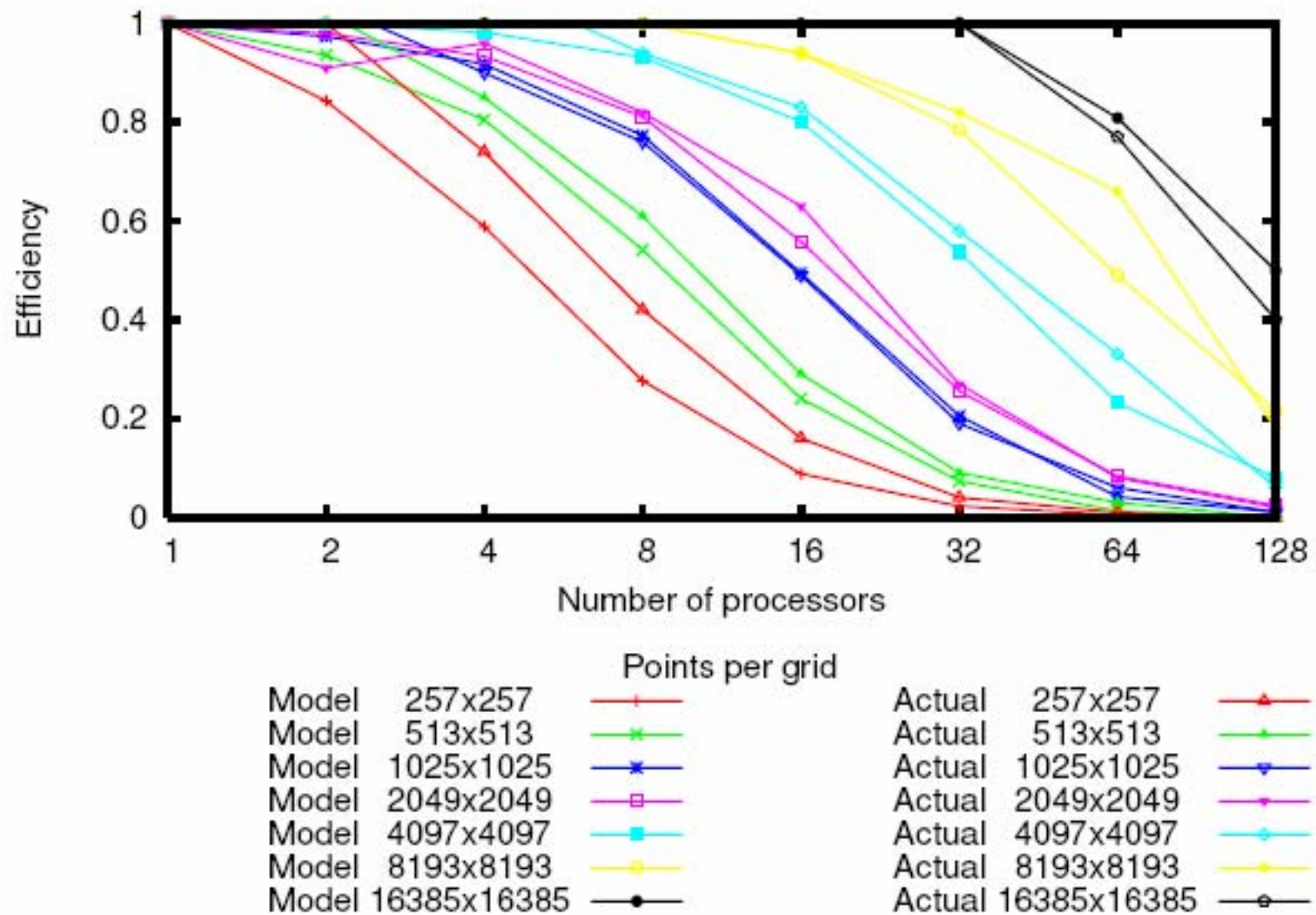


Figure 1: Comparison of the performance model to the actual parallel experiments

How far can we go with this algorithm?

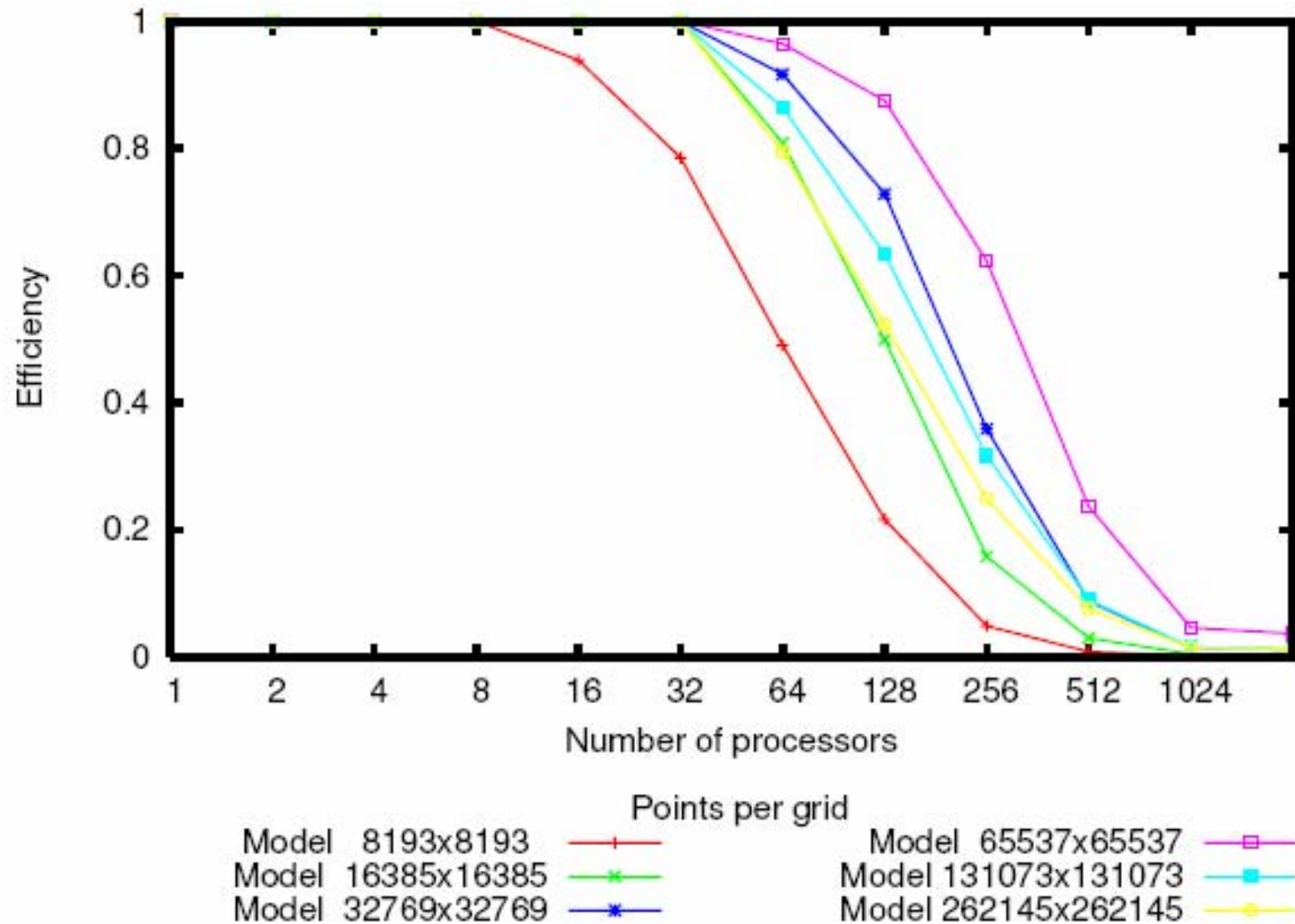
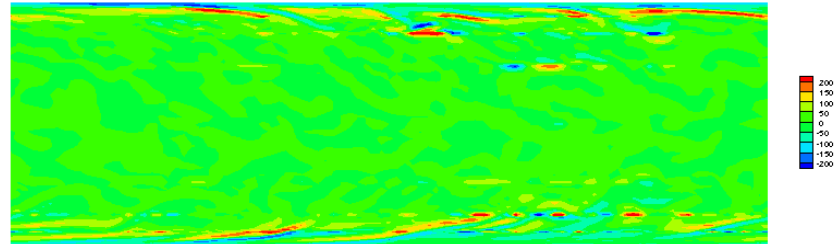


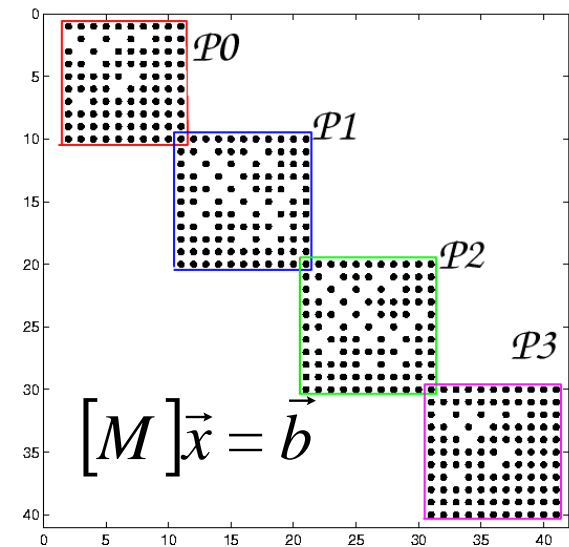
Figure 2: Predictions from performance model

Direct Numerical Simulation

Solve Navier-Stokes without turbulence models. Allows for detailed analysis of the physics



- Periodic in xz-plane FFT
- One-D spectral elements in y
 - Strongly local mass matrix
- Third-order stiffly stable in t



$P_y = \#$ procs in y-dir, $P_{xz} = \#$ procs in xz-dir $P_{total} = P_y * P_{xz}$

FFT is expensive while PCG is very efficient due to matrix structure

THIS KEY POINT DOES NOT SEEM TO HAVE BEEN EXPLOITED IN THE PAPERS

Parallel Algorithm

Two communicators

“Slab” comms for transforms
in xz-plane

“Pencil” comms for PCG
solver in y-dir

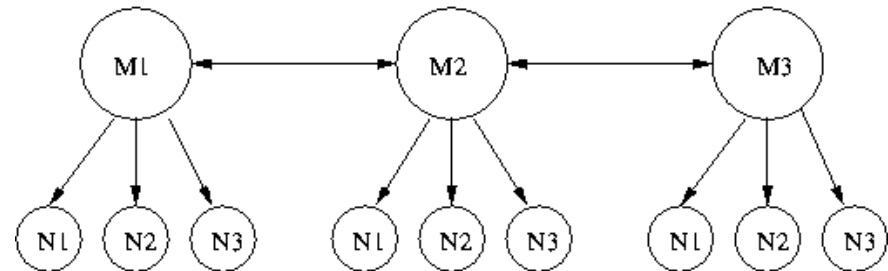
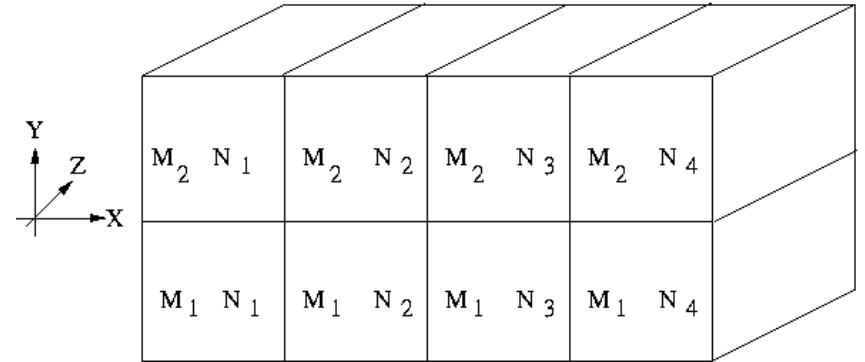
FFT (“Slab”) uses `MPI_Alltoall`

PCG (“Pencil”)

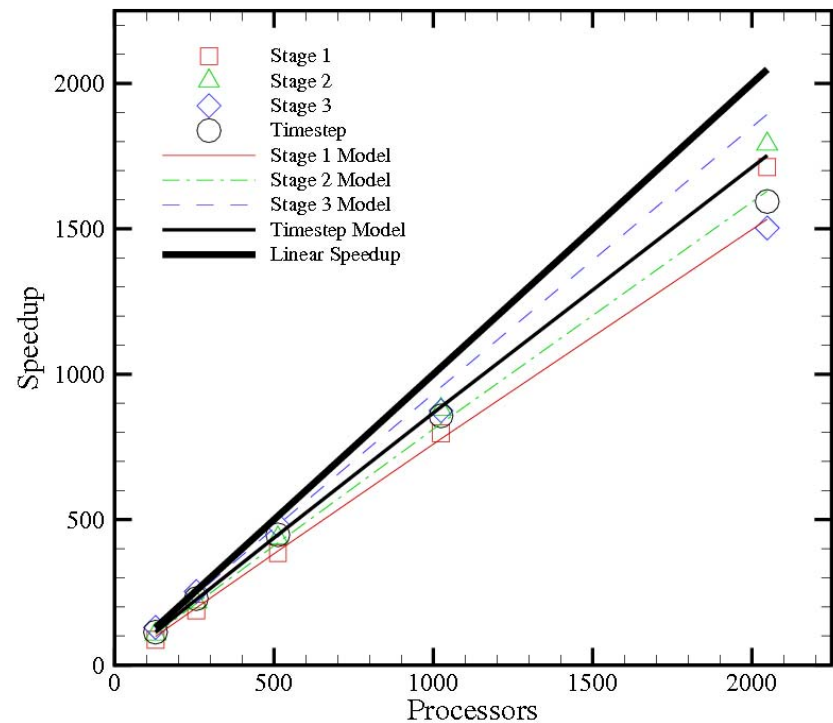
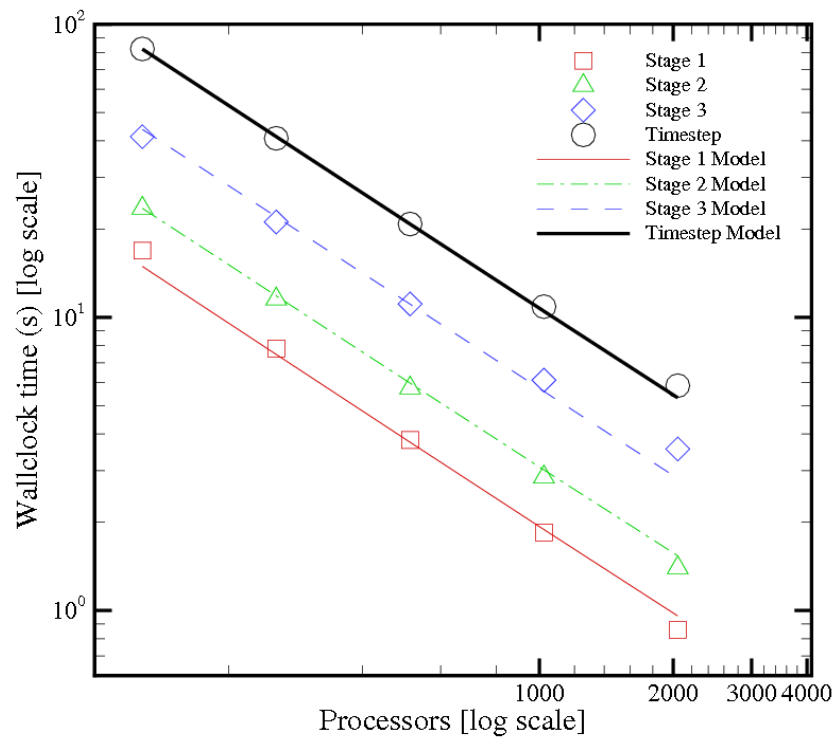
Pairwise comms +

`MPI_Allreduce` used
in Hemholtz solves

for pressure and viscous terms

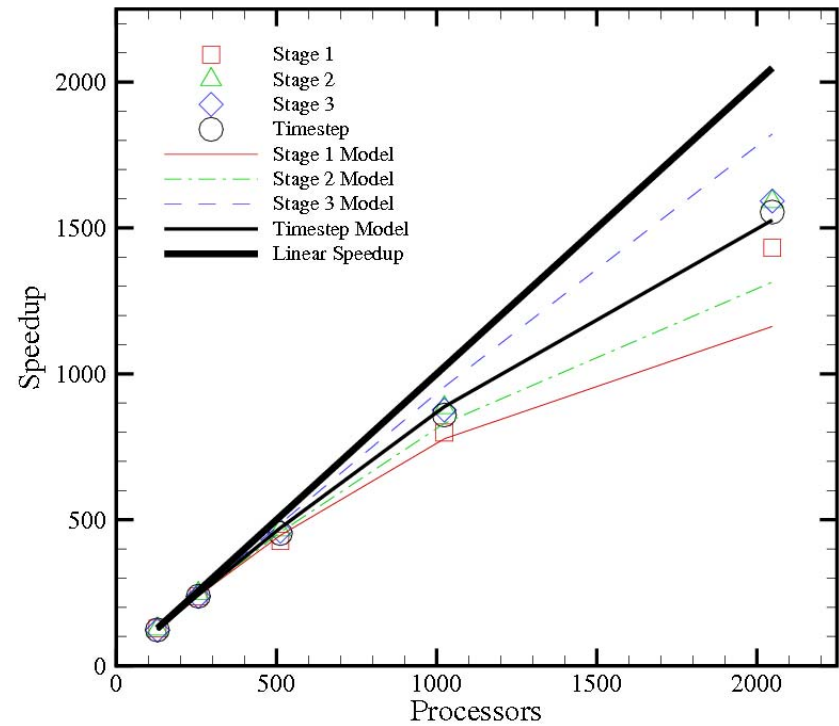
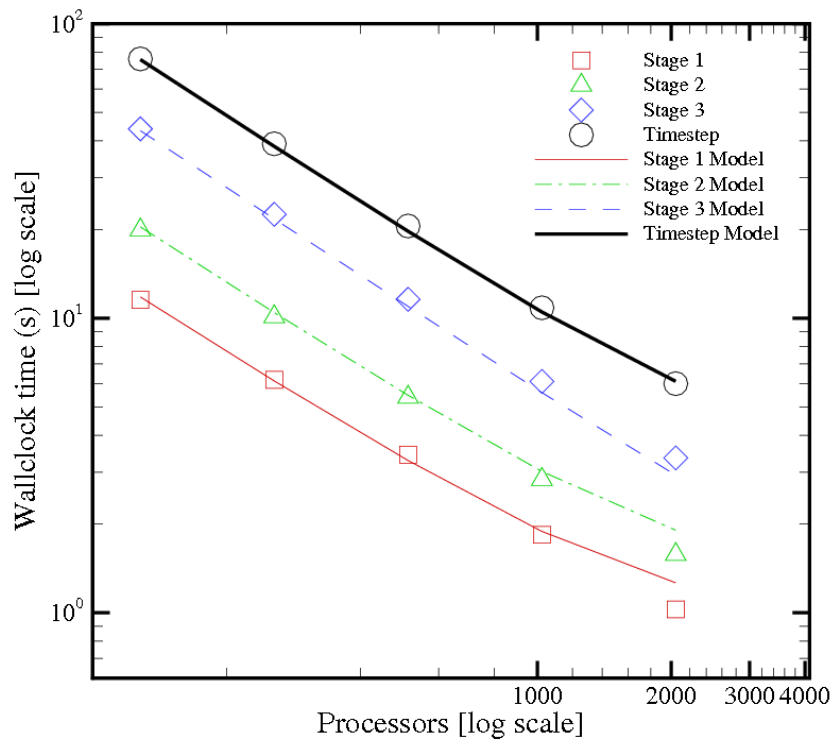


Strong scaling – Constant P_{xz}



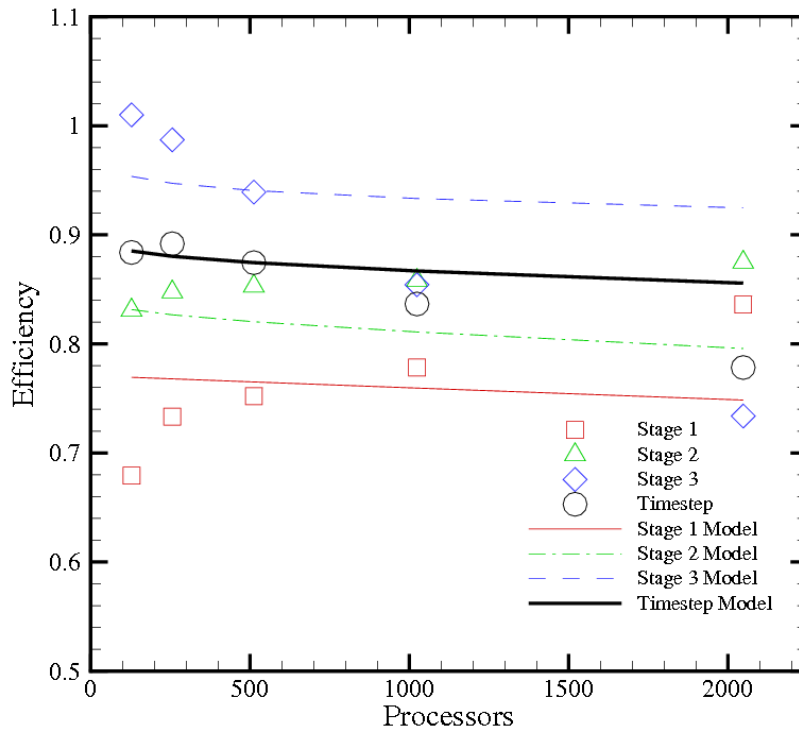
$$P_{xz} = 32, N_x = N_z = 512, N_e = 64, P = 8, \tau_w = 4.0 \text{ ns}, \tau_s = 4.0 \text{ } \mu\text{s}$$

Strong Scaling – Constant P_y

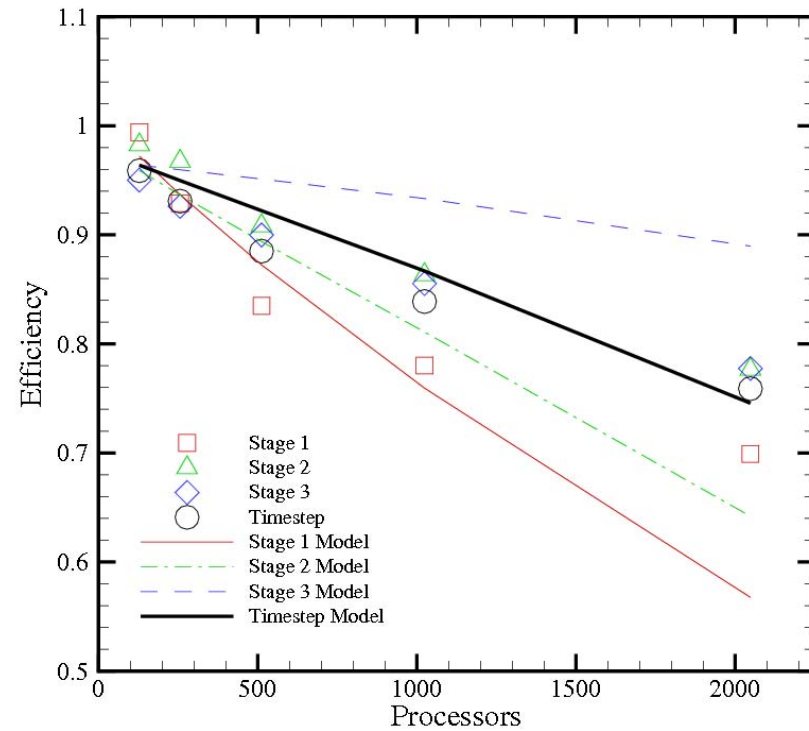


$P_y = 32, N_x = N_z = 512, N_e = 64, P = 8, \tau_w = 4.0 \text{ ns}, \tau_s = 4.0 \text{ } \mu\text{s}$

Strong Scaling – Efficiency



Constant P_{xz}

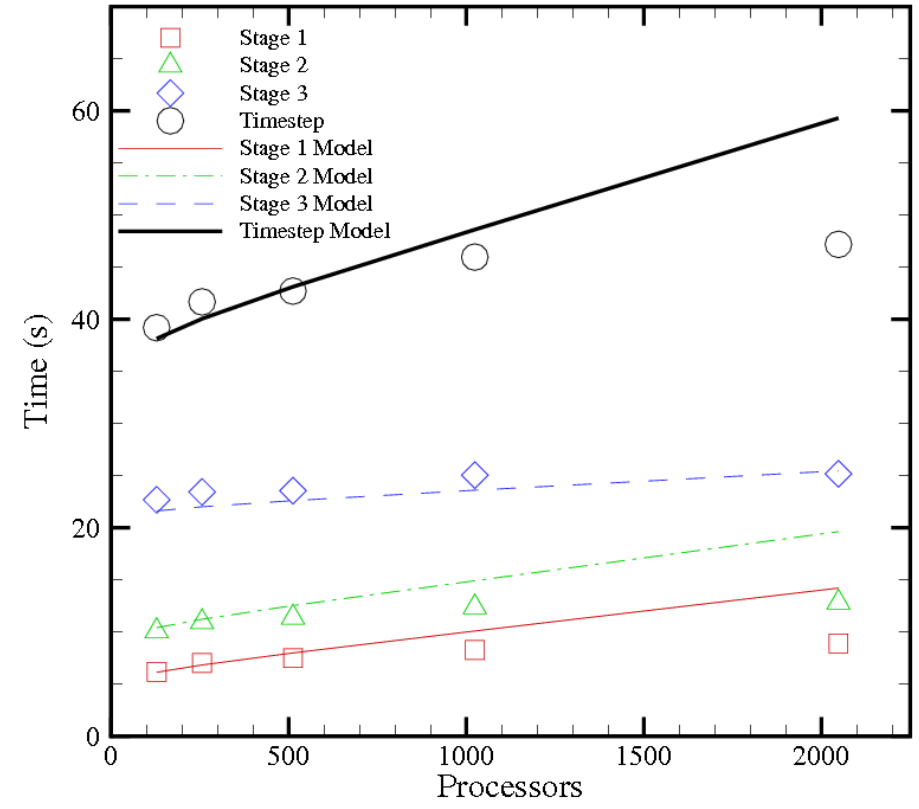
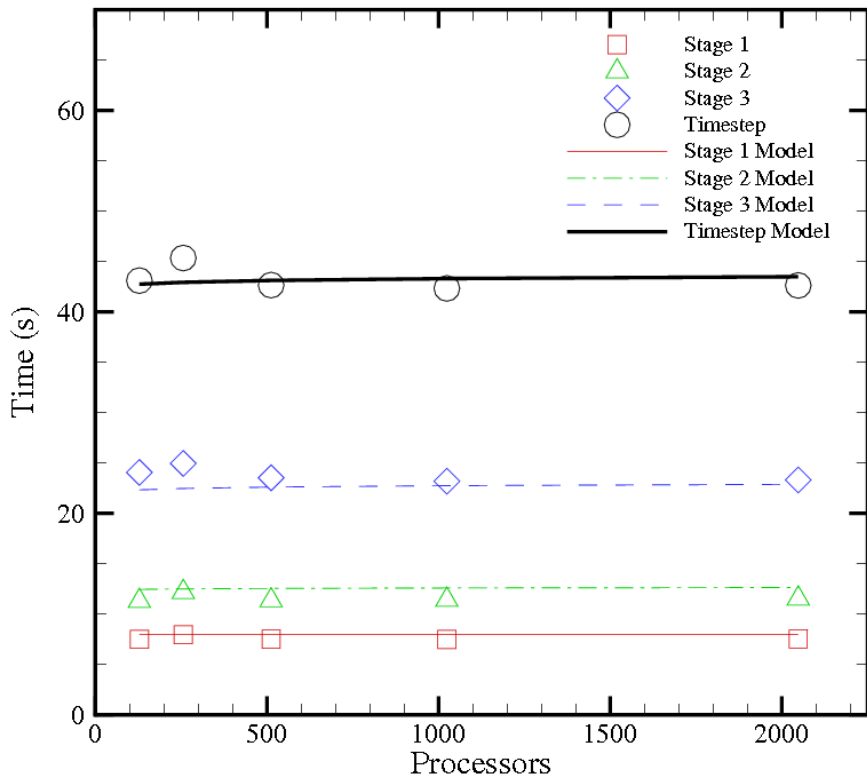


Constant P_y



Weak Scaling – Constant P_{xz}

Weak Scaling – Constant P_y



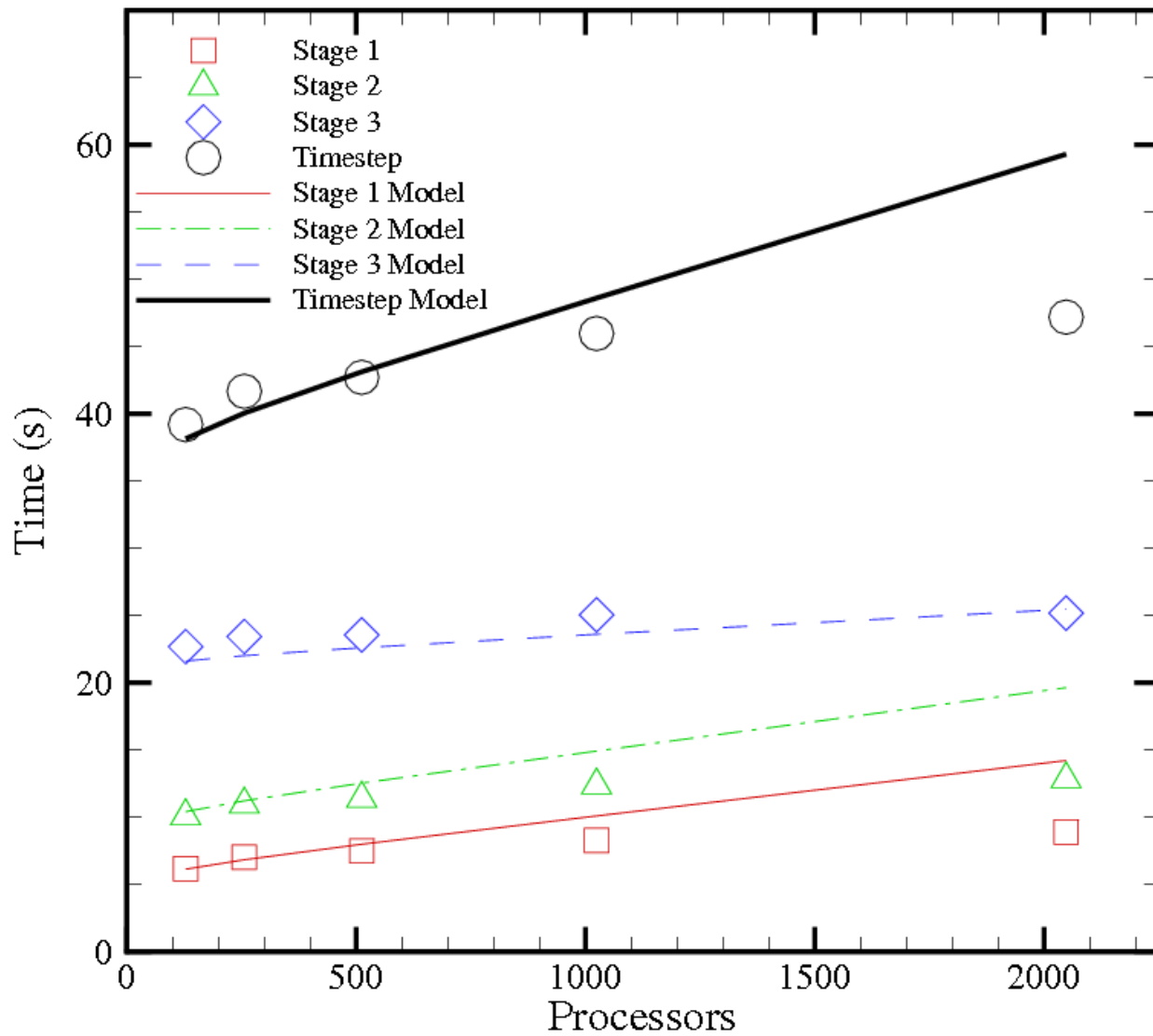
$P_{xz} = 32, N_x = N_z = 1024, N_{e_{128}} = 8,$

$P_y = 16, N_{x_{128}} = N_{z_{128}} = 512, N_e = 32, P = 8,$

$P = 8, \tau_w = 4.0 \text{ ns}, \tau_s = 4.0 \text{ } \mu\text{s}$

$\tau_w = 4.0 \text{ ns}, \tau_s = 4.0 \text{ } \mu\text{s}$

Weak Scaling – Constant P_y



$P_y = 16$, $N_{x_{128}} = N_{z_{128}} = 512$, $N_e = 32$, $P = 8$, $\tau_w = 4.0$ ns, $\tau_s = 4.0$ μ s

Dynamic Load Balancing Goals

Distribute work evenly.

Keep communication minimal.

Assumption: Maintain Spatial Locality

Fast: $O(N \log N)$?

Parallel for scalability

Incremental for adaptive meshes

SPACE FILLING CURVES – widely used for AMR

Hilbert Curve maintains locality well

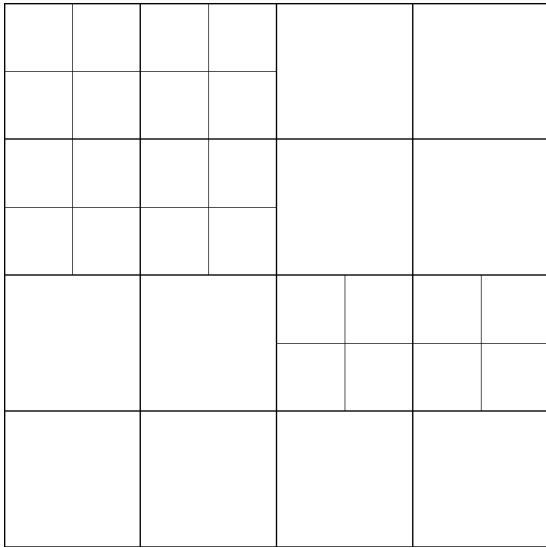
Recursive Binning Sort $O(N \log N)$



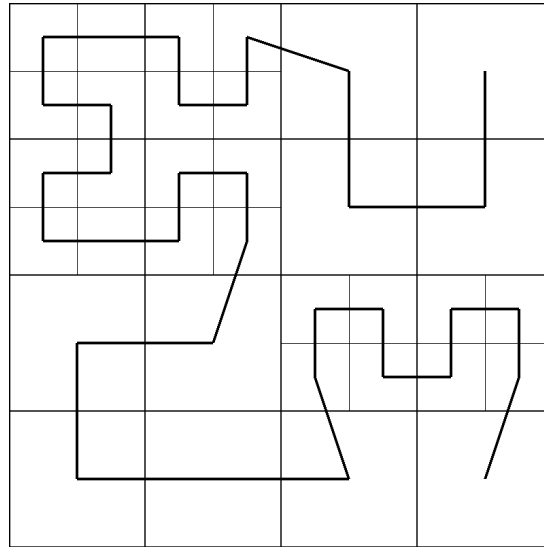
Space-Filling Curves: Example

Recursive local load balancing

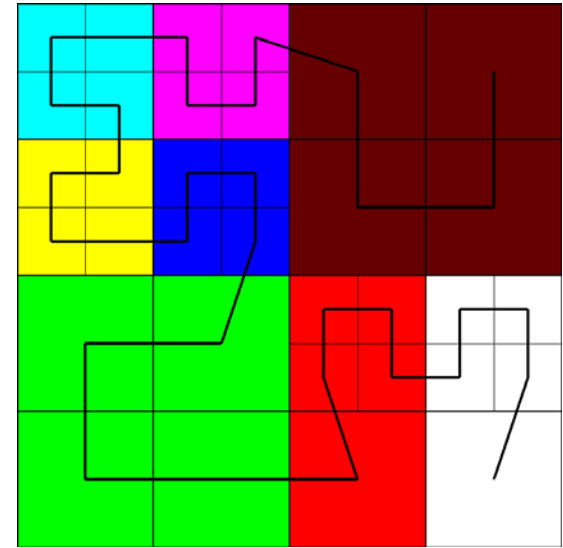
*Adaptive
mesh*



*Generate space
filling curve*



*Define processor
partition using mesh
cells on curve*



Provides mapping from multi-D to 1-D space.

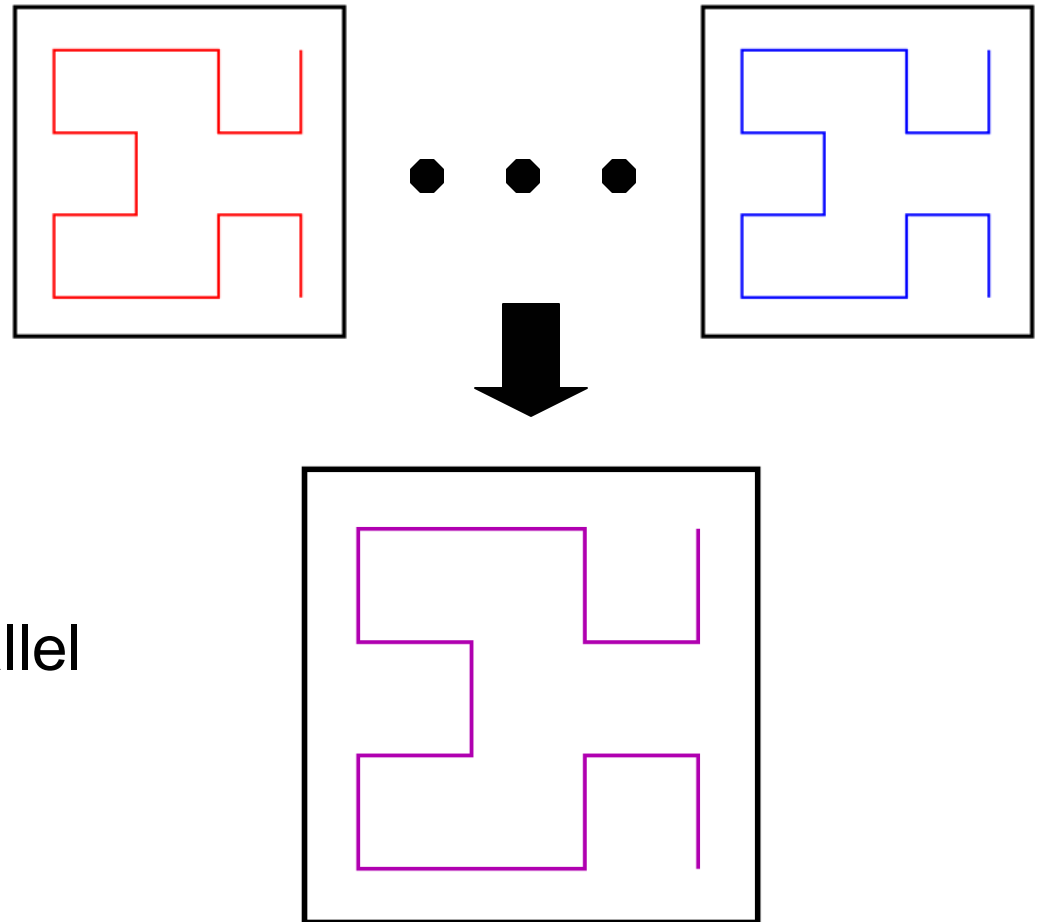
Parallel Generation

- Goals

- Fast
- Scalable
- Incremental

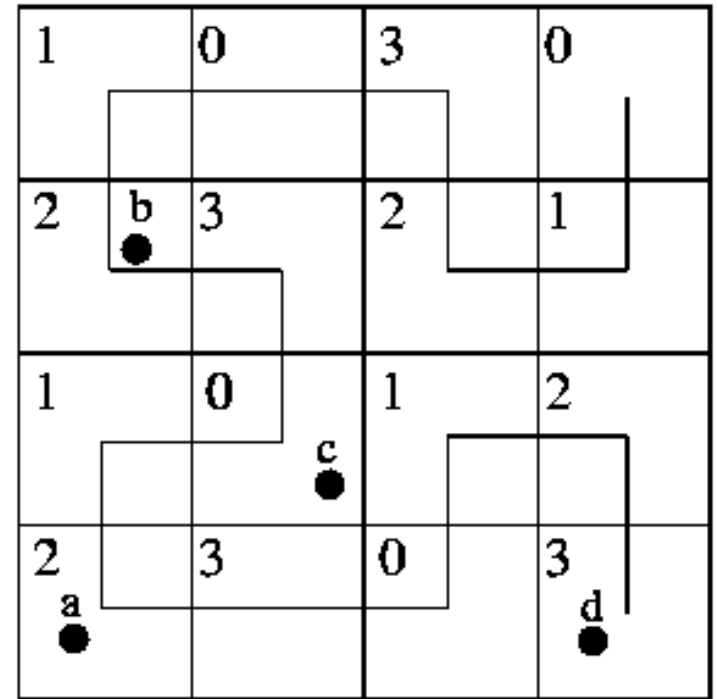
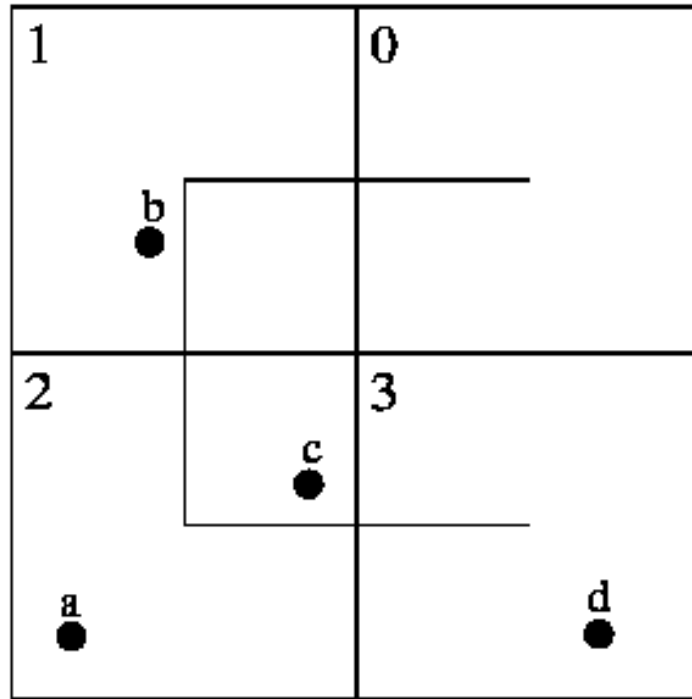
- Strategy

- Partition points
- Form curves in parallel
- Merge Curves



Relative Visitation Order

$O(n \log n)$



Digit Histories: a:22 b:12 c:20 d:33 Hence order is b c a d

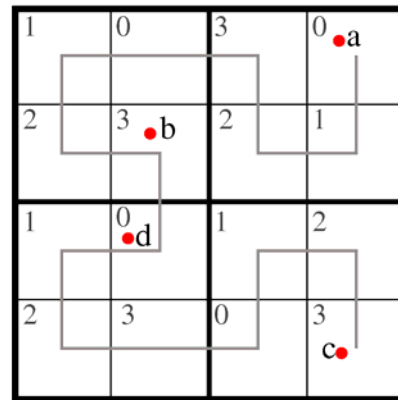
Create once and save -unique traversal order index

Send history during merging- cheaper than sending locations and recreating index -integer Comparison

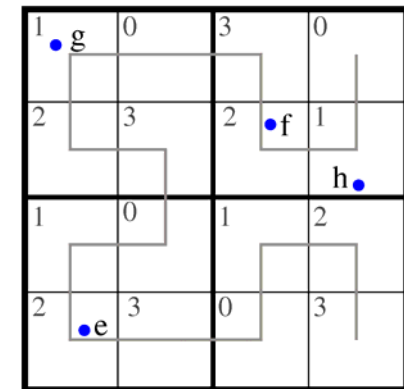


Parallel Merging

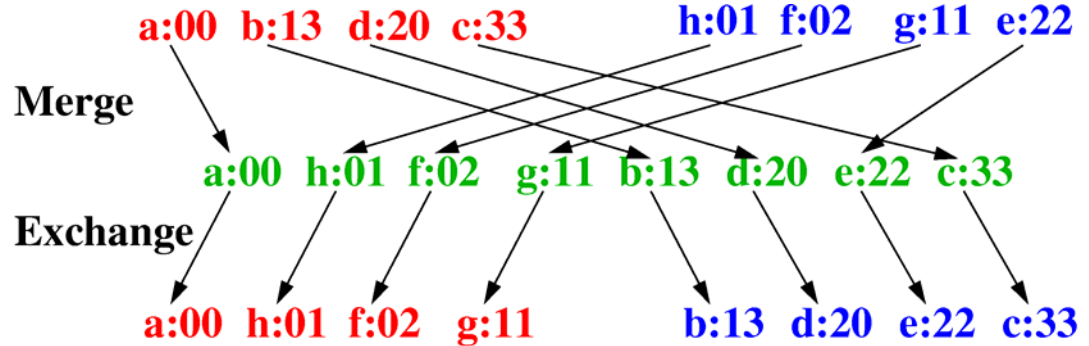
- Merge-Exchange
 - $O(N/P)$
 - Use digit histories
 - Asynchronous
- Two Pass Merging
 - Primary
 - Mostly sort quickly
 - $O(N/P \log P)$
 - Cleanup
 - Finalize sort
 - $O(N/P \log^2 P)$



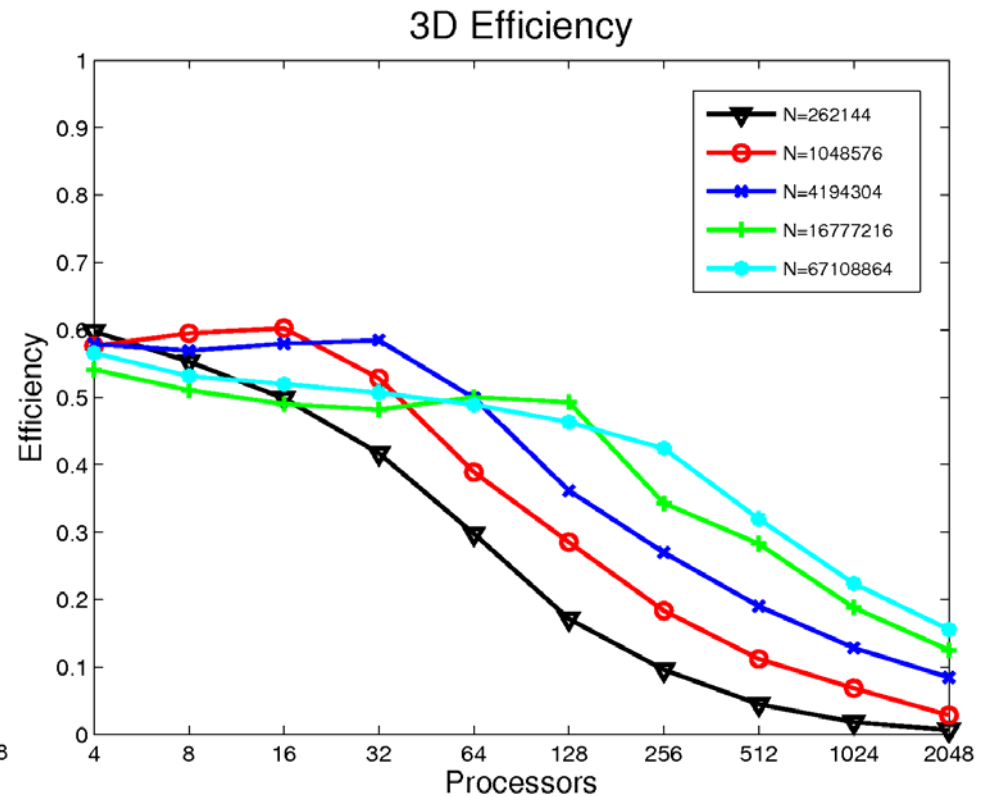
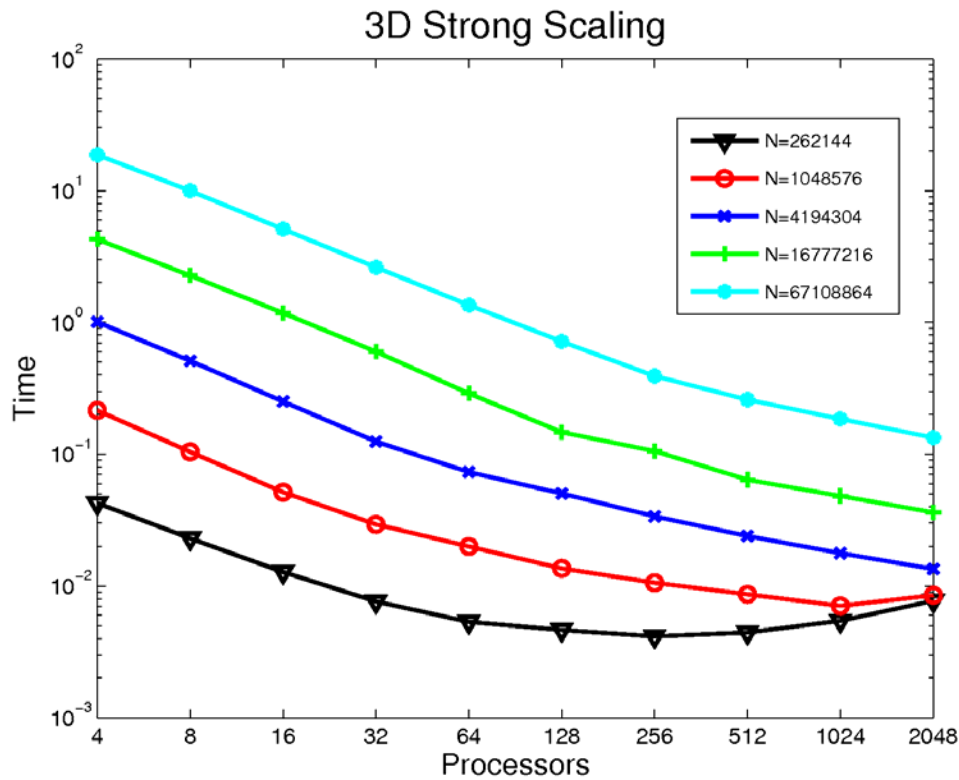
a:00 b:13 d:20 c:33



h:01 f:02 g:11 e:22

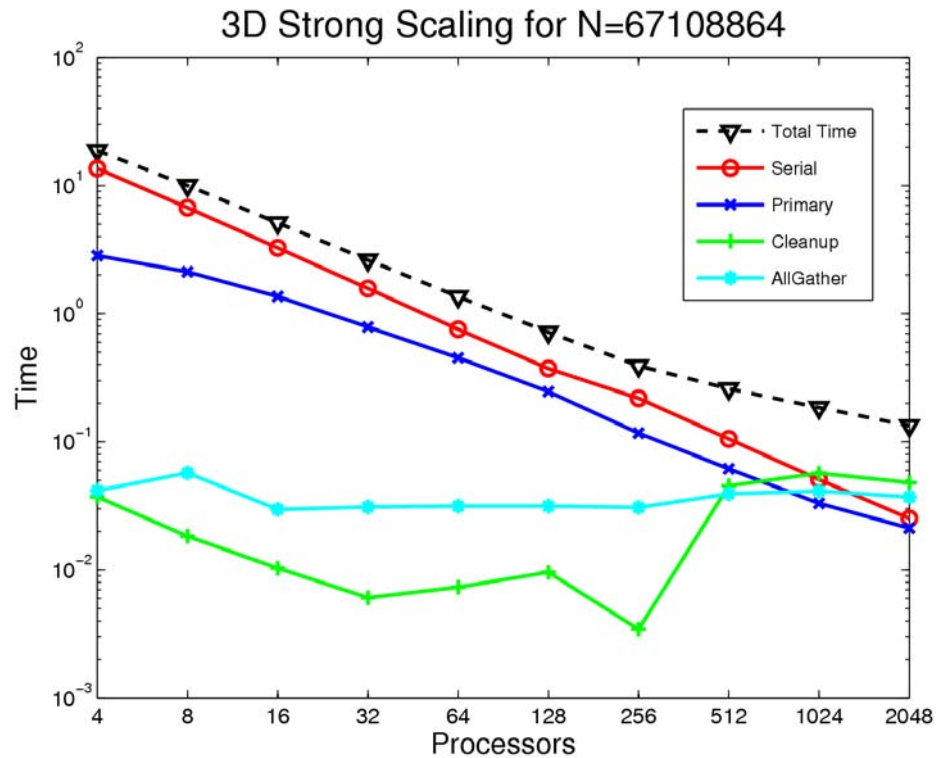
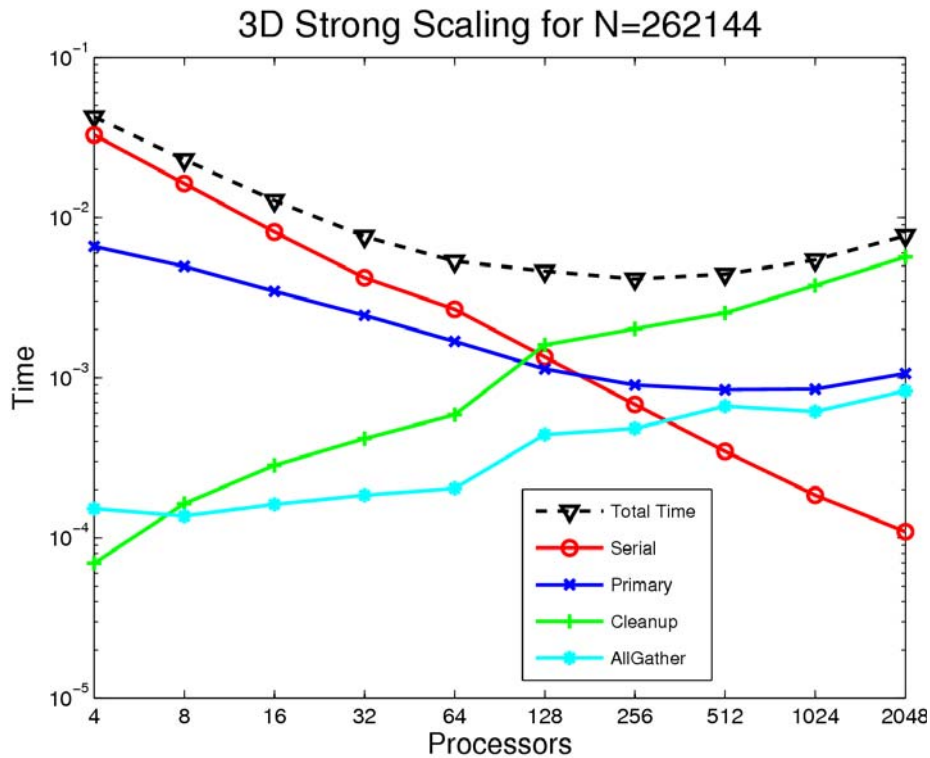


Strong Scaling



Thunder: 512 nodes with 4x 1.4 Ghz processors

Where Was Time Spent?



Thunder: 512 nodes with 4x 1.4 Ghz processors

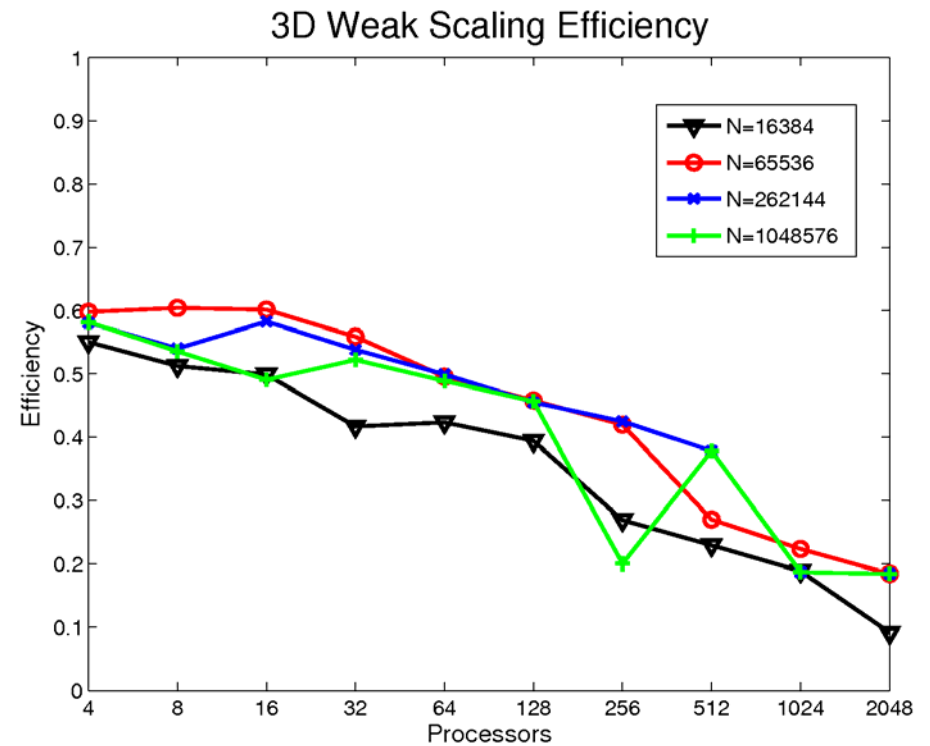
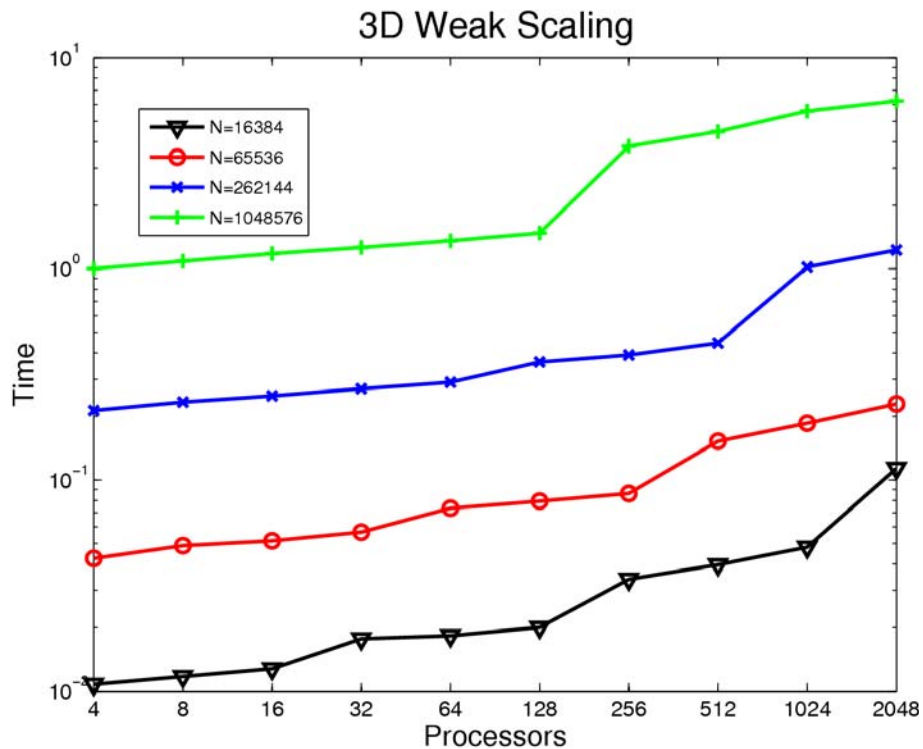
Conclusions

- Impossible problems may not be so.
- **Education is probably the key - let the students have controlled access to large machines....**
- Performance models help in all the cases shown
- Timing components is very revealing
- Algorithmic innovation is almost always needed
- Space-filling curves can be generated quickly in parallel
- Asynchronous communication is a necessity
- Weakest Link will prevent scalability
 - **Global Communication in at least two of the cases**



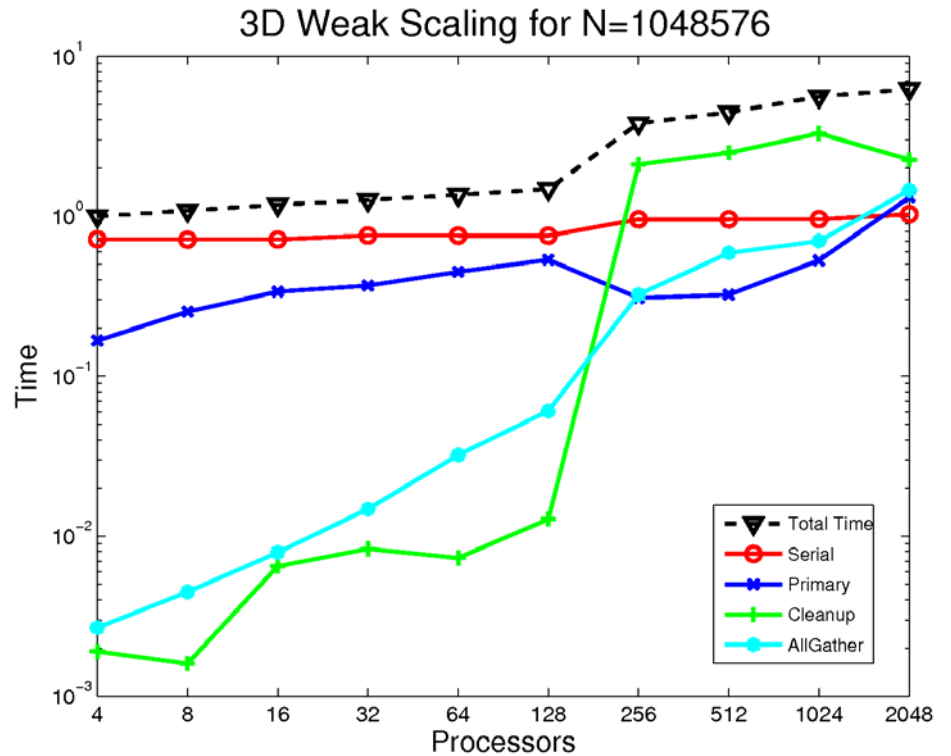
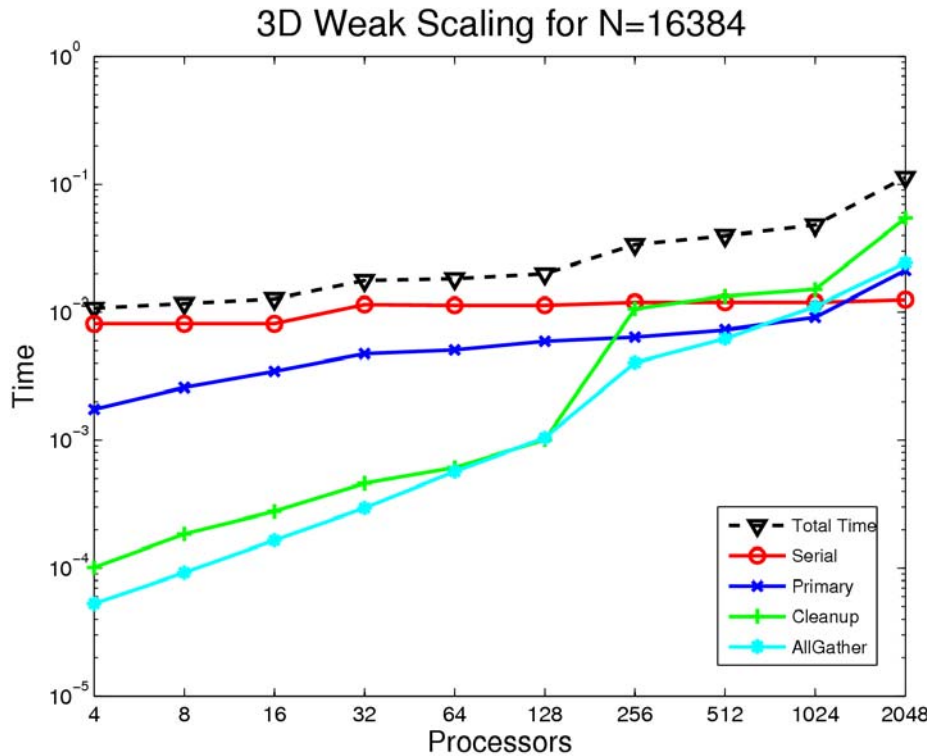
This work was supported by the University of Utah's Center for the Simulation of Accidental Fires and Explosions (C-SAFE) funded by the Department of Energy, under subcontract No. B524196.

Weak Scaling



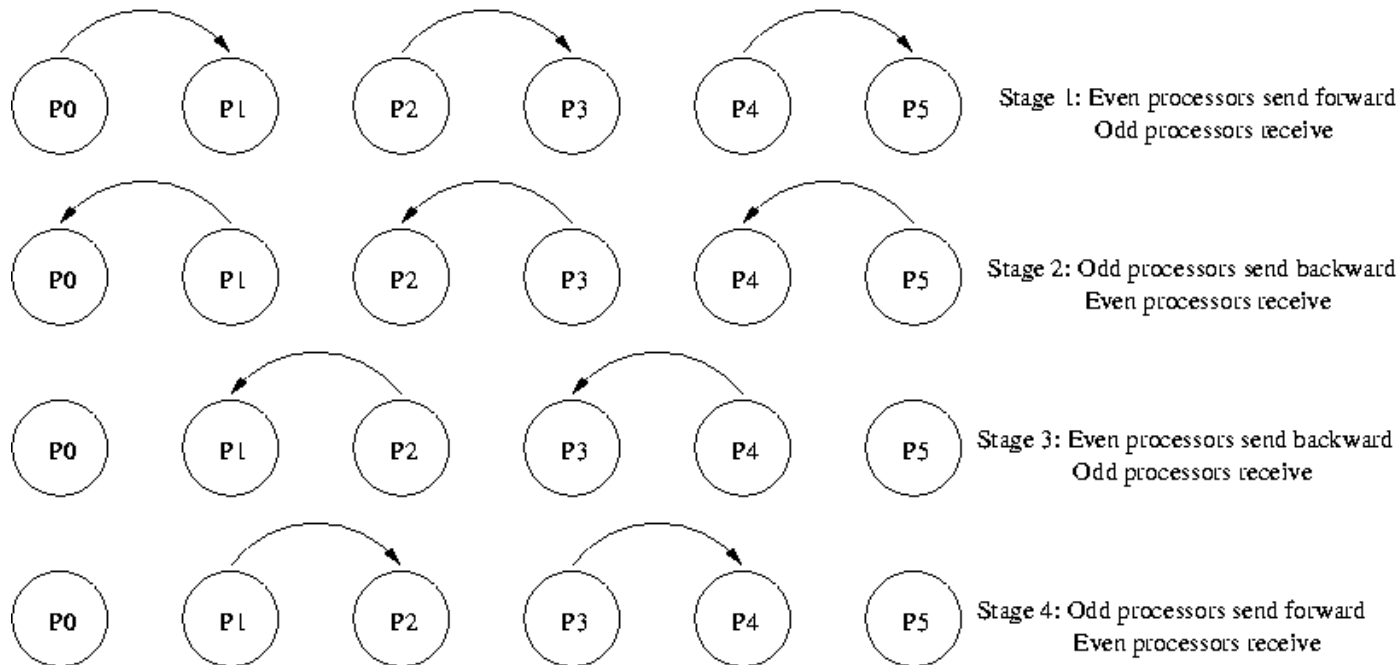
Thunder: 512 nodes with 4x 1.4 Ghz processors Z

Where Was Time Spent?



Thunder: 512 nodes with 4x 1.4 Ghz processors

Parallel Algorithm



- Stage 1 (FFT) -- Non-linear computations
- Stage 2 (FFT + Pairwise + Allreduce) -- Helmholtz pressure solver
- Stage 3 (FFT + Pairwise + Allreduce) -- Helmholtz viscous solver

Reynolds Equation and Film Thickness Equation

$$\frac{\partial}{\partial x} \left(\frac{\rho h^3}{\eta} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\rho h^3}{\eta} \frac{\partial p}{\partial y} \right) = 6 \left\{ \frac{\partial (u_s \rho h)}{\partial x} + \frac{\partial (v_s \rho h)}{\partial y} + 2 \frac{\partial (\rho h)}{\partial t} \right\}$$

η is viscosity $1 - 10^6$, u_s, v_s are surface velocities in x and y **2D circular point contact Film thickness, h :**

$$h(x,y) = h_{00} + \frac{x^2}{2R_x} + \frac{y^2}{2R_y} + \frac{2}{\pi E'} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{p(x',y') dx' dy'}{\sqrt{(x-x')^2 + (y-y')^2}},$$

Conservation law - applied load carried entirely by lubricant film:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x,y) dx dy = F$$

Surface Roughness replace $\frac{x^2}{2R_x} + \frac{y^2}{2R_y}$ with real surface profile.



Performance Model

• Computation

$$T_{S1} = \frac{A_1}{F} \left[72 + 15Q + 15J_e \frac{N_x N_z N_y}{P} + 24 + 2.5 N_x N_z N_y \log_2 N_x N_z \right]$$

$$T_{S2} = \frac{A_2}{F} \left[49 + 4Q + 6J_e \frac{N_x N_z N_y}{P} + 31 + 16P + 29 + 4PK \frac{N_x N_z N_e P}{P} + 23 + 2.5 N_x N_z N_y \log_2 N_x N_z \right]$$

$$T_{S3} = \frac{A_3}{F} \left[24 N_x N_z N_y + 3 + 31 + 16P + 29 + 4PK \frac{N_x N_z N_e P}{P} + 9 + 2.5 N_x N_z N_y \log_2 N_x N_z \right]$$

On Thunder

$$A_1 = 0.65$$

$$A_2 = 0.85$$

$$A_3 = 1.40$$

$$F = 100 \text{ Mflops}$$

$$\tau_w = 4.0 \text{ ns}$$

$$\tau_s = 4.0 \text{ } \mu\text{s}$$

$$K = 10 \text{ PCG iterations}$$

$$Q = P+2$$

$$J_e = 3 \text{ (third order)}$$

• Communication

$$T_{C1} = 7 \left[P_{xz} \frac{P_{xz} - 1}{s} + 3 \frac{P_{xz} - 1}{P_{xz} P_y} N_x N_z N_y \right] \tau_w$$

$$T_{C2} = 6 \left[P_{xz} \frac{P_{xz} - 1}{s} + 3 \frac{P_{xz} - 1}{P_{xz} P_y} N_x N_z N_y \right] \tau_w + 5 \left[P_{xz} \frac{P_{xz} - 1}{s} + \frac{P_{xz} - 1}{P_{xz} P_y} N_x N_z N_y \right] \tau_w$$

$$+ C K \frac{2}{s} \frac{N_x N_z}{P_{xz}} \tau_w + 2 K \log_2 P_y \frac{N_x N_z N_e P}{P_{xz} P_y} \tau_w$$

$$T_{C3} = 3 \left[P_{xz} \frac{P_{xz} - 1}{s} + 3 \frac{P_{xz} - 1}{P_{xz} P_y} N_x N_z N_y \right] \tau_w + C K \frac{2}{s} \frac{3 N_x N_z}{P_{xz}} \tau_w + 2 K \log_2 P_y \frac{3 N_x N_z N_e P}{P_{xz} P_y} \tau_w$$

$$S = \frac{T_{S1} + T_{S2} + T_{S3}}{\frac{T_{S1} + T_{S2} + T_{S3}}{P_{xz} P_y} + T_{C1} + T_{C2} + T_{C3}}$$

