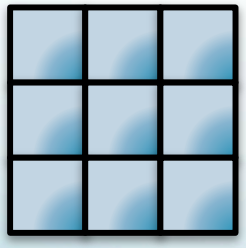
# “Challenges of Scaling Algebraic Multigrid Across Modern Multicore Architectures.”

Allison H. Baker, Todd Gamblin, Martin Schulz, and Ulrike Meier Yang

# Multigrid Solvers

- Method of solving linear equation systems
  - Transforms linear equation system into matrix equation of the form $A\mathbf{u} = \mathbf{f}$

$$\nabla^2\Phi = 4\pi G\rho \quad \Rightarrow \quad A\Phi = 4\pi G\rho$$



$$\frac{1}{\Delta x^2}\begin{bmatrix}
-4 & 1 & & 1 & & & & & \\
1 & -4 & 1 & & 1 & & & & \\
 & 1 & -4 & & & 1 & & & \\
1 & & & -4 & 1 & & 1 & & \\
 & 1 & & 1 & -4 & 1 & & 1 & \\
 & & 1 & & 1 & -4 & & & 1 \\
 & & & 1 & & & -4 & 1 & \\
 & & & & 1 & & 1 & -4 & 1 \\
 & & & & & 1 & & 1 & -4
\end{bmatrix}
\begin{bmatrix}
\phi_{1,1} \\ \phi_{1,2} \\ \phi_{1,3} \\ \phi_{2,1} \\ \phi_{2,2} \\ \phi_{2,3} \\ \phi_{3,1} \\ \phi_{3,2} \\ \phi_{3,3}
\end{bmatrix}
= 4\pi G
\begin{bmatrix}
\rho_{1,1} \\ \rho_{1,2} \\ \rho_{1,3} \\ \rho_{2,1} \\ \rho_{2,2} \\ \rho_{2,3} \\ \rho_{3,1} \\ \rho_{3,2} \\ \rho_{3,3}
\end{bmatrix}$$

# Multigrid Solution Process [1,2]

1. Run iterative smoother (e.g. Gauss-Seidel) on full-resolution matrix to remove high-frequency errors from initial guess
2. Coarsen problem domain, producing a lower-resolution grid with a smaller matrix
3. Run smoother again on coarsened equation, removing lower-frequency error terms
4. Replace initial guess with interpolated coarse solution.
5. Repeat Steps 1-4 until solution converges.

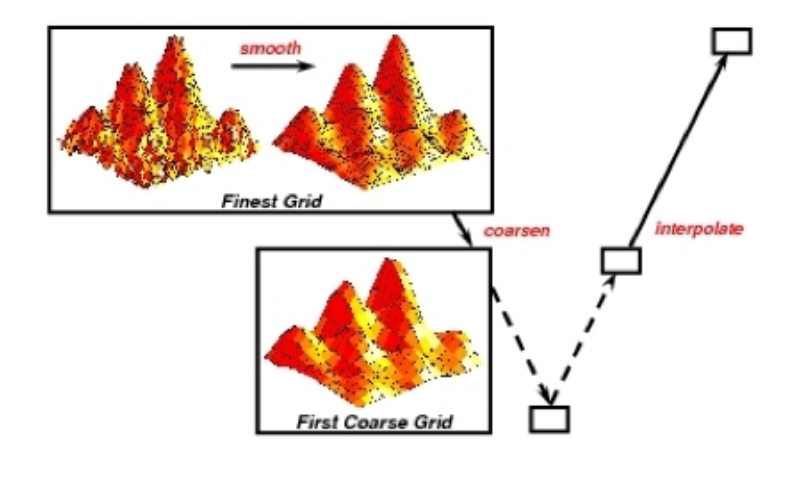
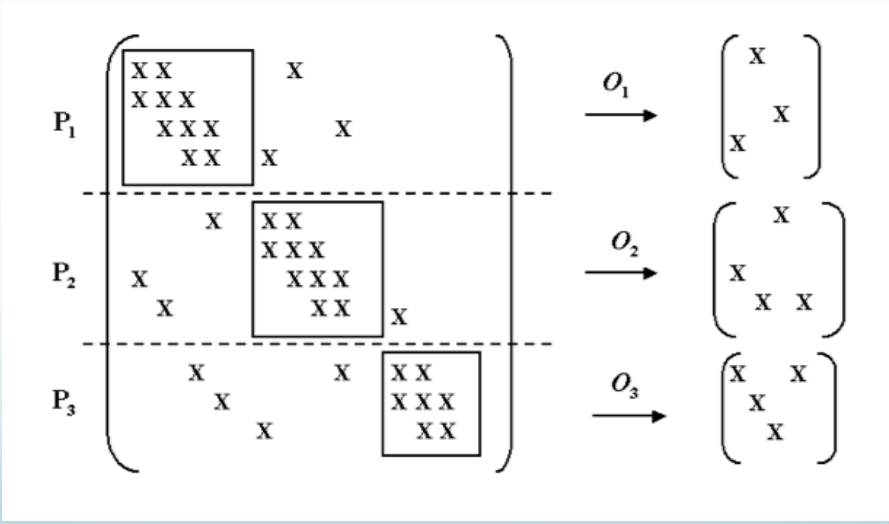



Image Source: Baker at al. "Challenges of Scaling Algebraic Multigrid across Modern Multicore Architectures". IPDPS, 2011.

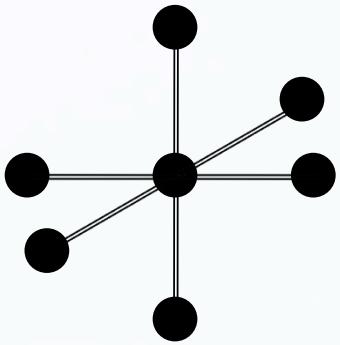
# Parallelism

- Matrix is stored in a parallel version of the compressed-sparse-row (CSR) format [3]
- Each processor gets a set of matrix rows; row-space further subdivided into local and remote referencing matrices.
- Local range must be communicated to receive processors during setup phase.

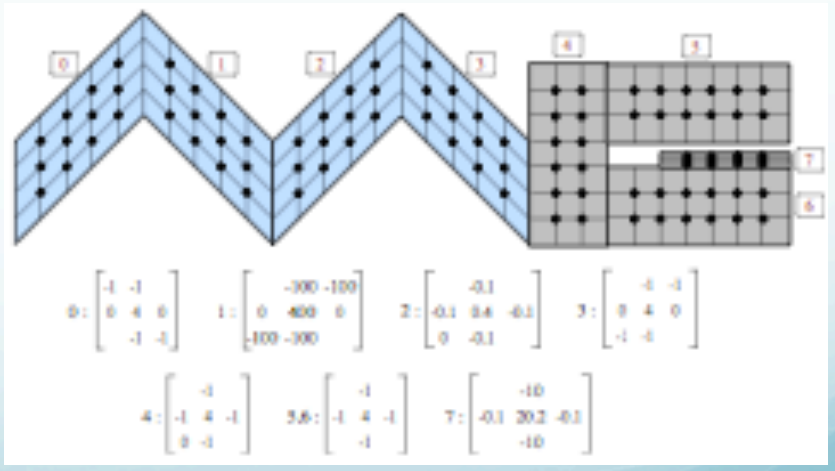


# Test Problems

- Laplace problem on 3D structured grid
  - Simple structure, seven point stencil.



- 3D diffusion problem, complicated geometry
  - Complex grid jumps [1], anisotropic geometry



# Test Machines

- *Hera*:  864-node QDR-InfiniBand system
  - Four 2.3 GHz AMD quad-core processors per node
  - 32 GB RAM per node (NUMA)
- *Jaguar*: 18688-node Cray XT-5
  - Two AMD Operon Hex-core processors per node
  - 16 GB RAM per node (NUMA)
- *Intrepid:* 40960-node Quad-core Blue Gene/P
  - One quad-core 850 MHz Power 450 processor/node
  - 2 GB memory per node (UMA)

# Quad-Core Cluster (Hera)

- 864 nodes, QDR InfiniBand interconnects
- 4 sockets per node, quad-core AMD 8356 Opterons
- 2 MB L3 shared cache
- 32 GB of memory, divided between four sockets
- Memory outside of local partition can be accessed via HyperTransport

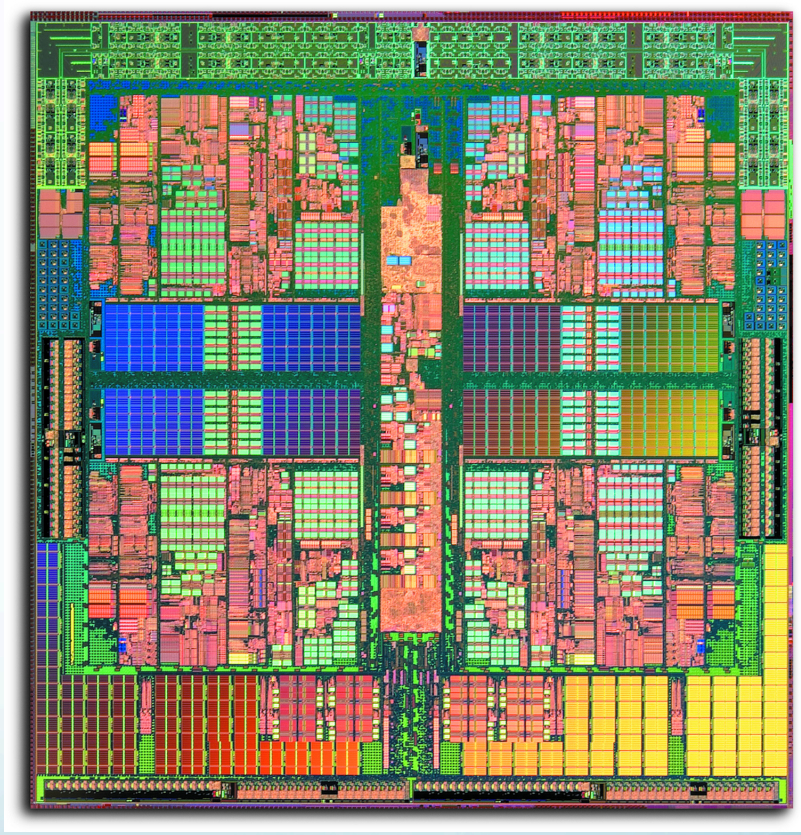


Image Source: Advanced Micro Devices, Inc. via Wikipedia

# Cray XT5 (Jaguar)

- 18868 Nodes nodes, SeaStar 2+ interconnects
- Two AMD Hex-Core Opterons per node
- 16 GB of memory, divided up between sockets
- 2D torus network topology

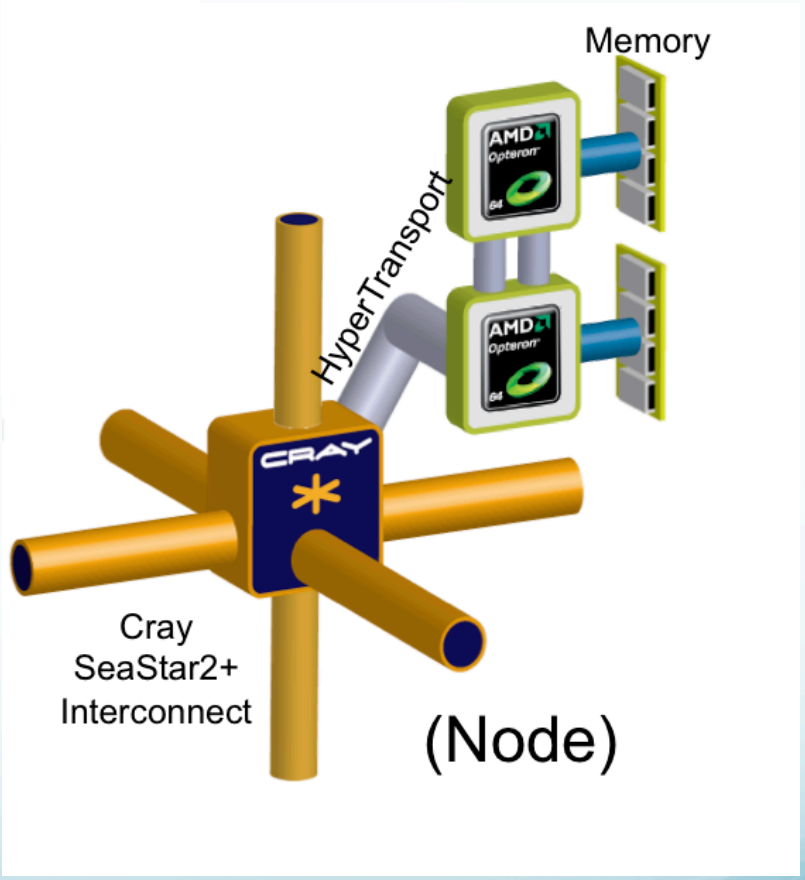


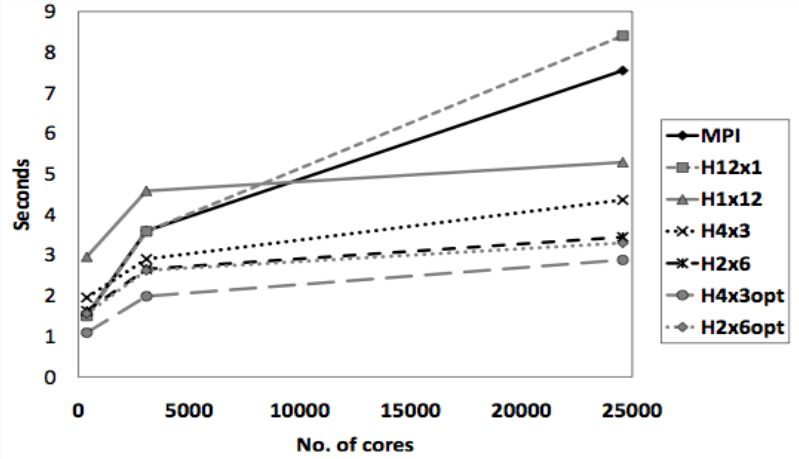

Image Source: NCCS
(http://www.nccs.gov/wp-content/uploads/2010/02/AMD-5.09.10.pdf)

# BlueGene/P Cluster (Intrepid)

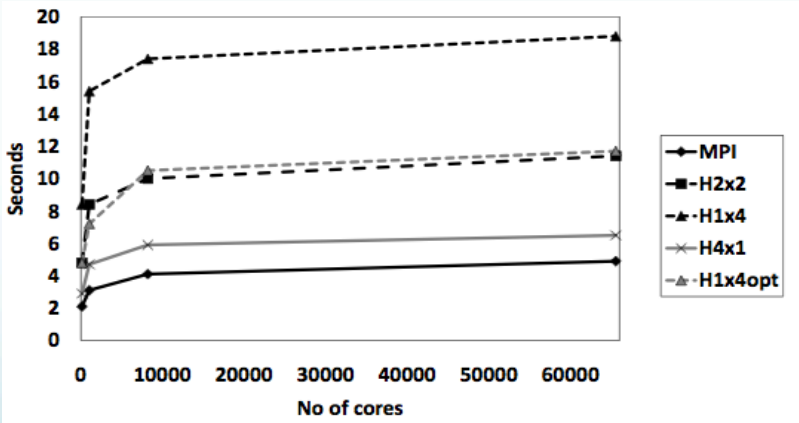
- 40960 nodes, one quad-core PowerPC 450 processor per node
- 2 GB memory per node, shared by all cores
  - Uniform Memory Access
- 3D torus network topology
  - Periodicity in all dimensions

# Result Summary

- Hera
  - Extremely poor MPI-only performance
  - 1-thread OpenMP run performs worst during solve
  - H4x4 best at low core counts, H2x8 overtakes it
- Jaguar
  - Slightly better MPI-only performance, but still poor
  - NUMA-related issues on the H1x12 MG-1 trial
  - H4x3, H2x6 perform best, esp. with optimization
- Intrepid
  - Fast node interconnects make MPI viable



(c) Setup times for AMG-GMRES(10) on the MG problem.



(c) Setup times for AMG-GMRES(10) on the MG-1 problem.

Image Source: Baker et al. "Challenges of Scaling Algebraic Multigrid across Modern Multicore Architectures." IPDPS, 2011.

# Results Discussion

- NUMA effects noticeable on Hera, Jaguar.
  - Use of MCSup to constrain threads to local memory partitions improves performance on NUMA machines
  - Process pinning required for memory locality constraints to be effective
- Poor interconnect speed on non-BlueGene machines makes MPI transactions expensive.
  - Expected to become a problem as number of cores on chip outstrips increases in interconnect speed.

# “Hierarchical Parallelization of Gene Differential Analysis”

Mark Needham, Rui Hu, Sandhya Dwarkadas, Xing Qiu

# Gene Differential Association Analysis

- Determine whether two genes have different correlation patterns under different conditions.
- Partition *n* subjects into *G* subgroups. Calculate correlation vectors and *N*-statistics (measures change in gene correlation over two conditions [4]).
- Shuffle groups *K* times and recalculate *N*-statistics using new groupings.
- Compute *p*-value using permuted *N*-statistics (low *p*-value indicates change in gene correlation across conditions [4]).

# Parallelized Algorithm

- Permutation tasks shared across processors using Python and MPI.
- *N*-statistics calculated using C++ and pthreads.
- *m x n* data array replicated across MPI processes
- Two *m x G* subgroup arrays, *m*-element *N*-statistic array on each MPI process (shared access for pthreads).

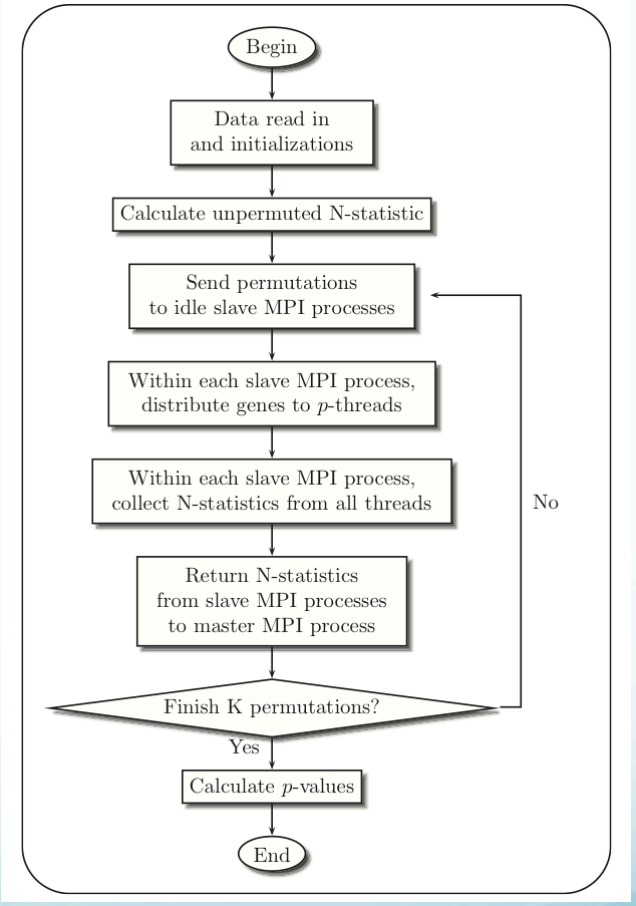



Image Source: Needham et al. "Hierarchical Parallelization of Gene Differential Association Analysis." BMC Bioinformatics, 2011.

# Hardware

- 40 cores across 5 machines
  - One processor used for running Python script; that machine is not used for computation
  - 32 cores available
- Dual quad-core 3 GHz Intel Xeon processors
- 16 GB memory, Gigabit Ethernet interconnects
- 6 MB L2 cache per core

# Result Summary

- Multithreaded simulations outperform MPI alone
- MPI carries significant memory overhead due to data replication
- Pinning processes to cores improves multithreaded performance
- Additional threading beyond 2 threads yields little advantage on pinned system

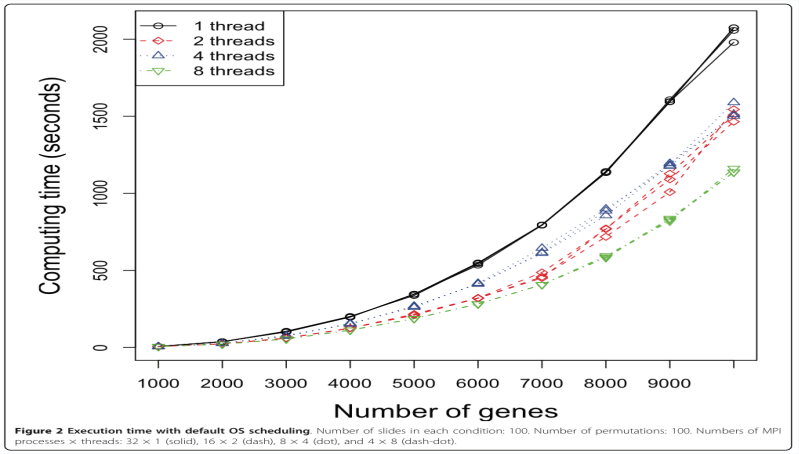



**Figure 2 Execution time with default OS scheduling.** Number of slides in each condition: 100. Number of permutations: 100. Numbers of MPI processes × threads: 32 × 1 (solid), 16 × 2 (dash), 8 × 4 (dot), and 4 × 8 (dash-dot).

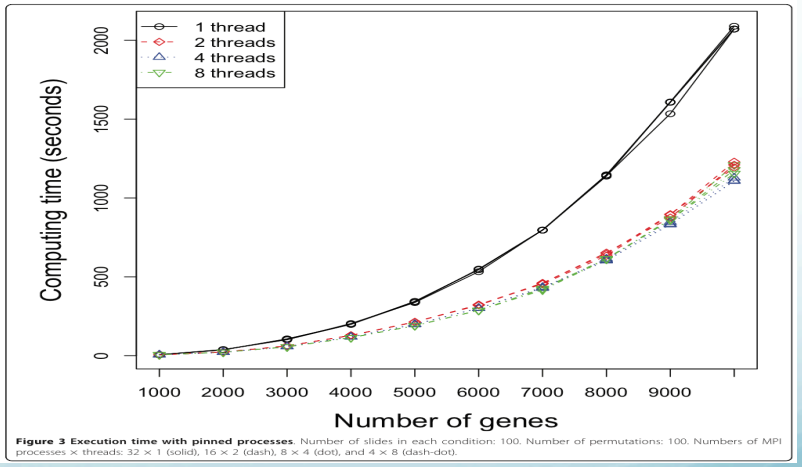



**Figure 3 Execution time with pinned processes.** Number of slides in each condition: 100. Number of permutations: 100. Numbers of MPI processes × threads: 32 × 1 (solid), 16 × 2 (dash), 8 × 4 (dot), and 4 × 8 (dash-dot).

Image Source: Needham et al. "Hierarchical Parallelization of Gene Differential Association Analysis." BMC Bioinformatics, 2011.

# Result Summary,ctd.

- Jagged speedup curve attributed to imperfect load balancing.
- Without process pinning, scheduling errors amplify the uneven quality of the speedup curve [4].
- Scheduling issues can also seriously degrade performance in unpinned threads [4].

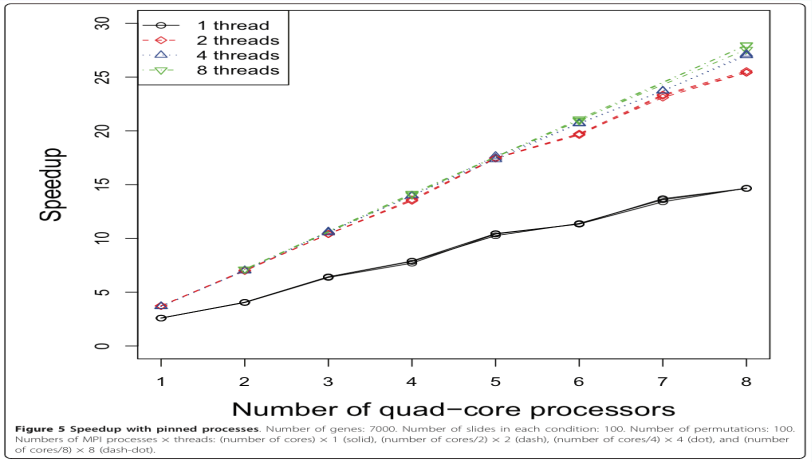



**Figure 5 Speedup with pinned processes**. Number of genes: 7000. Number of slides in each condition: 100. Number of permutations: 100. Numbers of MPI processes × threads: (number of cores) × 1 (solid), (number of cores/2) × 2 (dash), (number of cores/4) × 4 (dot), and (number of cores/8) × 8 (dash-dot).

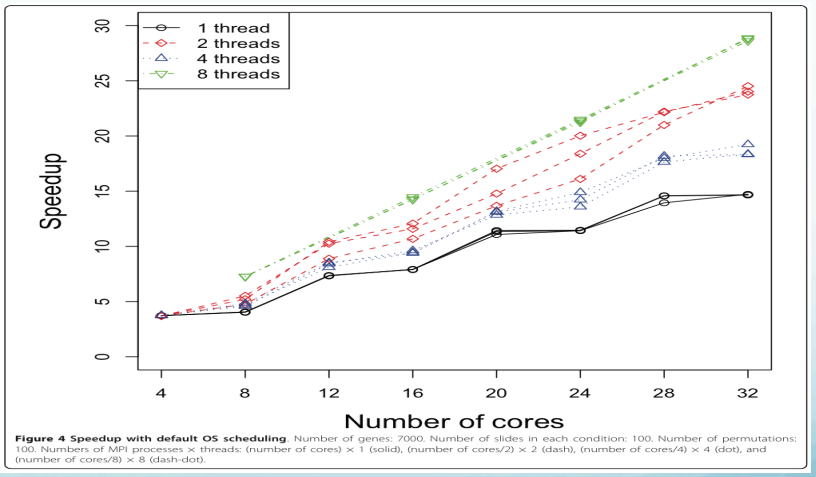



**Figure 4 Speedup with default OS scheduling**. Number of genes: 7000. Number of slides in each condition: 100. Number of permutations: 100. Numbers of MPI processes × threads: (number of cores) × 1 (solid), (number of cores/2) × 2 (dash), (number of cores/4) × 4 (dot), and (number of cores/8) × 8 (dash-dot).

Image Source: Needham et al. "Hierarchical Parallelization of Gene Differential Association Analysis." BMC Bioinformatics, 2011.

# References


1. Baker, A. et al. "Challenges of Scaling Algebraic Multigrid Across Modern Multicore Architectures." IPDPS, 2011.
2. Falgout, R. D. "An Introduction to Algebraic Multigrid", *Computing in Science And Engineering*, 2006.
3. Falgout, R. D., Jones, J. E., and Yang, U. M. "Pursuing Scalability for Hypre's Conceptual Interfaces", ACM Transactions in Mathematical Software, 2005.
4. Needham, M. Rui, H., Dwarkadas, S., Qiu, X. "Hierarchical Parallelization of Gene Differential Association Analysis.", BMC Bioinformatics, 2011.