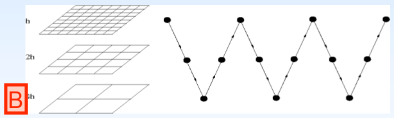
Scalable Algorithms for Distributed Computing Applications

Hans De Sterck, Applied Mathematics (cross-appointed to CS-SCICOM), with Jeff Butler, Lei Tang, Quoc Nguyen, James Pearson, Scott Rostrup, Paul Ullrich, Peter Stechlinski, Chen Zhang, Aleks Papo, Zhiyuan Wu

(1) Scalable multilevel linear algebra solvers on Blue Gene/L (128,000 processors)

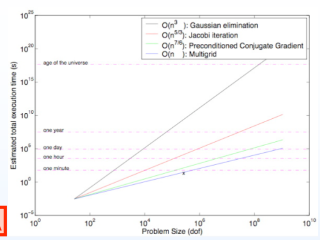
- solve $\mathbf{Au} = \mathbf{f}$ with billions of unknowns
- O(n) scalability of algorithms is essential A
- use recursive, multilevel approach (reduce error components at appropriate resolution, similar to FFT or quicksort ideas) B
- we have developed new approaches for multigrid coarsening and interpolation
- target: scalability on Blue Gene/L at Lawrence Livermore National Lab (US, California) (128,000 processors) C
- result: our new methods scale much better than existing methods D

B

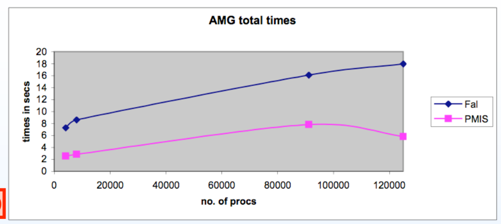


C

A



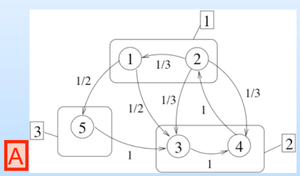
D



(2) Multilevel solvers for Google's PageRank

- determine web ranking based on link structure
- find stationary probability vector of Markov chain $B\,\mathbf{x} = \mathbf{x}$ $\quad \|\mathbf{x}\|_1 = 1$ A
- use agglomeration based on strength of connection
- use recursive, multilevel approach: MAA algorithm B C
- PageRank regularization of web graph (add all-to-all connections with weight $(\alpha = 0.15)$) D
- multilevel method is orders of magnitude more efficient than one-level methods, for Markov chains with local connections

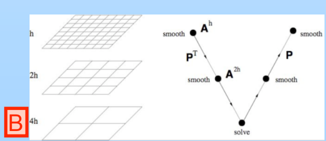
A



$$B\,\mathbf{x} = \mathbf{x} \qquad \|\mathbf{x}\|_1 = 1 \qquad B = \begin{bmatrix} 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/2 & 1/3 & 0 & 0 & 1 \\ 0 & 1/3 & 1 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

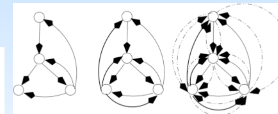
$$\mathbf{x}^T = [2/19\ 6/19\ 4/19\ 6/19\ 1/19]$$

B



D $B_{PR} = (1-\alpha)\,N(G + \mathbf{e}\,\mathbf{d}^T) + \alpha\,N(\mathbf{e}\mathbf{e}^T)$

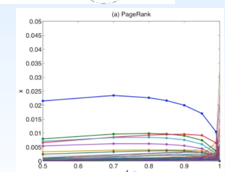
$(\alpha = 0.15)$



C

Algorithm: Multilevel Adaptive Aggregation method (V-cycle)

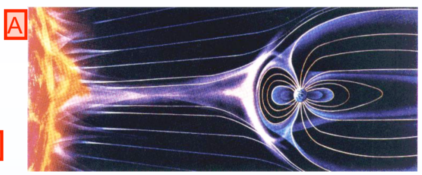
$\mathbf{x} = \text{MAA\_V}(A, \mathbf{x}, \nu_1, \nu_2)$
begin
$\quad \mathbf{x} \leftarrow N(\text{Relax}(A, \mathbf{x}))$ $\quad \nu_1$ times
$\quad$ build P based on $\mathbf{x}$ and A $\quad$ (P is rebuilt every level and cycle)
$\quad A_c = P^T A\, \text{diag}(\mathbf{x})\, P\, \text{diag}(P^T \mathbf{x})^{-1}$
$\quad \mathbf{x}_c = \text{MAA\_V}(A_c, N(P^T \mathbf{x}), \nu_1, \nu_2)$ $\quad$ (coarse-level solve)
$\quad \mathbf{x} = N(\text{diag}(P\, \text{diag}(P^T \mathbf{x})^{-1} \mathbf{x}_c)\, \mathbf{x})$ $\quad$ (coarse-level correction)
$\quad \mathbf{x} \leftarrow N(\text{Relax}(A, \mathbf{x}))$ $\quad \nu_2$ times
end



(3) Parallel simulations for Space Weather forecasting

- 'space weather': solar mass ejections reach earth's magnetic field, perturbing satellites, communication, power grids A
- solar plasma is modeled by 3D nonlinear PDE system (hyperbolic conservation law) B
- space weather prediction: scalable numerical solution of 3D PDE system on large parallel computers C
- project with Natural Resources Canada

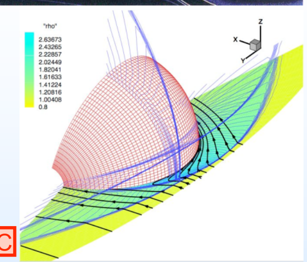
A



B

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho\,\vec{v} \\ \vec{B} \\ e \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho\,\vec{v} \\ \rho\,\vec{v}\vec{v} + I\left(p + \vec{B}\cdot\vec{B}/2\right) - \vec{B}\vec{B} \\ \vec{v}\vec{B} - \vec{B}\vec{v} \\ \left(e + p + \vec{B}\cdot\vec{B}/2\right)\vec{v} - \left(\vec{v}\cdot\vec{B}\right)\vec{B} \end{bmatrix} = 0$$

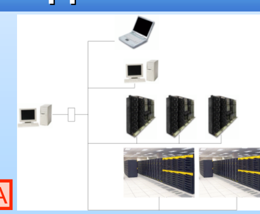
C



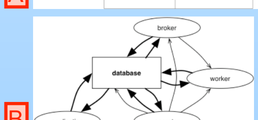
(4) GridBASE: scalable database-driven grid computing system applied to biomedical problems

- develop flexible and scalable grid computing system A
- data-base driven B
- deploy on heterogeneous network (SHARCNET) C
- taskfarming applications for biomedical problems

A



B



C