Multigrid methods for structured matrices on large scale supercomputers

Matthias Bolten

Bergische Universität Wuppertal

2012/09/03

Bergische Universität Wuppertal

- Medium-sized university in Northrine-Westfalia
- More than 16 000 students
- 250 professors
- In total more than 1 500 employees
- 7 departments and the school of education
- Close collaboration with Research Centre Jülich (supercomputing, atmospheric research)

Outline

Multigrid

Parallel multigrid

Scalability of `pmg`

Application

Scalability of application

Conclusion

Outline

Multigrid

Parallel multigrid

Scalability of `pmg`

Application

Scalability of application

Conclusion

The model problem

- For $\Omega = [0,1]^2$ consider the following Poisson equation

$$-\Delta u(x) = f(x), \text{ for } x \in \Omega \text{ and}$$
$$u(x) = 0 \text{ for } x \in \partial\Omega.$$

- Discretization on grid with $N = (n+1)\cdot(n+1)$ grid points using 5-point scheme yields linear system $Lu = f$, where

  $$\frac{1}{h^2}(4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}) = f_{i,j}, \text{ for } i,j = 1,\ldots,n_k,$$

  with $h = 1/n$ and $u_{i,j} = 0$ for $i \in \{0, n+1\}$ or $j \in \{0, n+1\}$.

- Eigenvalues and eigenvectors given by

$$\lambda_{l,m} = 4 - 2\cos(l\pi h) - 2\cos(m\pi h), \tag{1}$$
$$(\varphi_{l,m})_{i,j} = \sin(l\pi ih)\sin(m\pi jh), \tag{2}$$

The damped Jacobi method

Definition

Let $L \in \mathbb{R}^{N\times N}$, let $f \in \mathbb{R}^N$ and let the solution $u \in \mathbb{R}^N$ of the linear system $Lu = f$ be sought for.Let $D \in \mathbb{R}^{N\times N}$ contain the main diagonal of L. Then the *damped Jacobi method (JOR)* is defined as

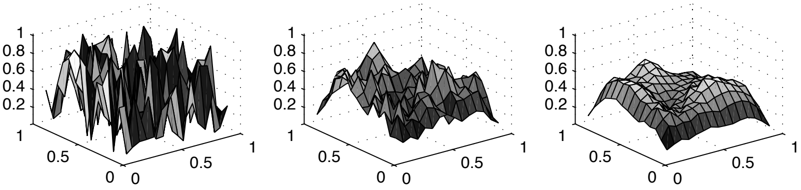
$$\begin{aligned} \phi_{\mathsf{JOR}} : \mathbb{R}^N \times \mathbb{R}^N &\to \mathbb{R}^N \\ (u^{(k)}, f) &\mapsto \phi_{\mathsf{JOR}}(u^{(k)}, f) = u^{(k+1)}, \end{aligned}$$

where $u^{(0)}$ is an initial guess and

$$u^{(k+1)} = u^{(k)} - \omega D^{-1}(Lu^{(k)} + f), k = 1, 2, \ldots .$$

The iteration matrix is given by $M_{\mathsf{JOR},\omega} = -\omega D^{-1}(D - L)$.

Behavior of the error



(a) Before

(b) After 1 step

(c) After 3 steps

Figure : Error of an arbitrarily chosen initial approximation and right hand side of the Laplacian discretized on the unit square using 15^2 grid points before and after application of one and three iterations of a damped Jacobi method with $\omega = 4/5$.

Smoother

- Due to the above observation, iterative methods like the Jacobi method are called *smoothers* in the multigrid setting.
- We distinguish *low* and *high* frequencies.

Definition

Let L be given by the model problem. The eigenvector $\varphi_{\ell;l,m}$ as given by (2) is called

$$\begin{aligned} &\text{low frequency,} && \text{if } \max(l,m) < (n_\ell + 1)/2, \\ &\text{high frequency,} && \text{if } (n_\ell + 1)/2 \leq \max(l,m). \end{aligned}$$

- *Smoothing factor* of the JOR method is defined as the worst factor by which a high frequency is damped.

Coarse grid correction

Observation

- Iterative solvers produce geometrically smooth errors

Idea

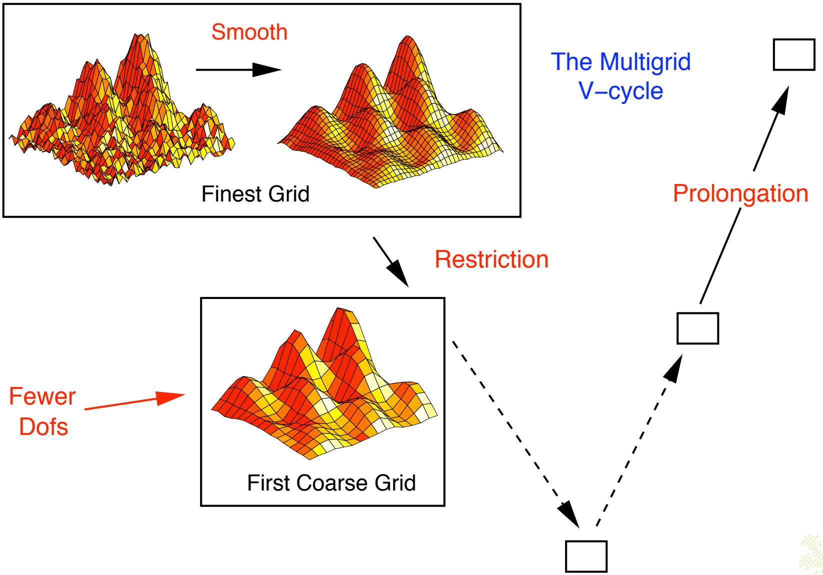
- Geometrically smooth error well-represented on a coarser grid
- System can be solved easier on coarser grid
- *Defect correction* using coarse grid approximation of error

$$u^{\mathsf{new}} = u^{\mathsf{old}} + \tilde{e}, \quad \tilde{e} \ = I_H^h L_H^{-1} I_h^H (f - L_h u^{\mathsf{old}})$$

- Additional components needed:
  - Restriction operator I_h^H (e.g. *injection*)
  - Prolongation operator I_H^h (e.g. *linear interpolation*)

Resulting procedure: *Coarse grid correction*

Multigrid



Smooth

Finest Grid

The Multigrid V–cycle

Restriction

Prolongation

Fewer Dofs

First Coarse Grid

Multigrid cycle

Multigrid cycle $x_{n_i} = \mathcal{MG}_i(x_{n_i}, b_{n_i})$

```
$x_{n_i} \leftarrow \mathcal{S}_i^{\nu_1}(x_{n_i}, b_{n_i})$
$r_{n_i} \leftarrow b_{n_i} - A_i x_{n_i}$
$r_{n_{i+1}} \leftarrow R_i r_{n_i}$
$e_{n_{i+1}} \leftarrow 0$
**if** $i + 1 = l_{\mathsf{max}}$ **then**
    $e_{n_{l_{\mathsf{max}}}} \leftarrow A_{l_{\mathsf{max}}}^{-1} r_{n_{l_{\mathsf{max}}}}$
**else**
    **for** $j = 1, \ldots, \gamma$ **do**
        $\mathbf{e}_{i+1} \leftarrow \mathcal{MG}_{i+1}(e_{n_{i+1}}, e_{n_{i+1}})$
    **end for**
**end if**
$e_{n_i} \leftarrow P_i e_{n_{i+1}}$
$x_{n_i} \leftarrow x_{n_i} + e_{n_i}$
$x_{n_i} \leftarrow \tilde{\mathcal{S}}_i^{\nu_2}(x_{n_i}, b_{n_i})$
```

Outline

Multigrid

Parallel multigrid

Scalability of `pmg`

Application

Scalability of application

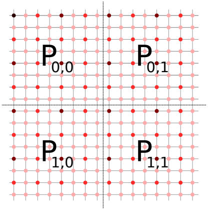
Conclusion

Parallelization of multigrid

- Variables distributed to processors
- Usually *domain splitting* approach
- Variable stays on assigned processor
- Yields idle processors on coarse levels
- Parallel time:

$$T_{\mathcal{MG}}(N,P) = \mathcal{O}(N/P + \log P)$$

- Result of global information exchange inherently necessary to solve the problem
- Still much better than `all-to-all`
- **But:** Becomes relevant on exascale machines



Parallel architectures

Forschungszentrum Jülich GmbH

Blue Gene/P
Massivley parallel system with modified PowerPC processors and 3D torus network for point-to-point communication plus tree network for global communication.

QPACE
Massively parallel system based on PowerXCell 8i with 3D torus network, developed by SFB/TRR 55.

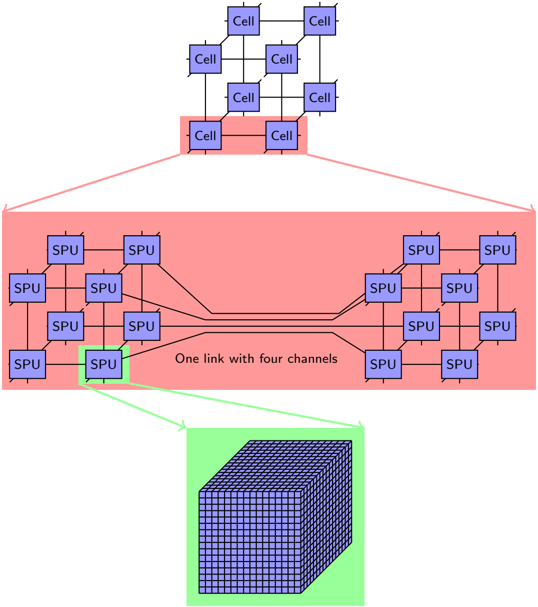
Baier et al., arXiv:0810.1559v1

Parallel multigrid solvers

- `pmg`
  - Multigrid solver for Toeplitz and circulant matrices
  - Includes algebraic optimizations based on given matrix
  - Optimal solver for subclass of Toeplitz and circulant matrices
  - Applicable to a broad range of applications
  - Uses torus archictecture extensively
  - Fits well to modern supercomputers like Blue Gene (Q included...)
- Multigrid for QPACE
  - Evaluation of QPACE for another architecture than QCD
  - Highly structured system suitable for solution of highly structured problems
  - Similar approach to `pmg` taken
  - Imlementation suitable for model-problem, only

Multigrid for QPACE

- Cell-based discretization instead of node-based (no communication necessary for restriction and prolongation)
- ω-Jacobi smoother with optimal smoothing parameter $\omega = \frac{2}{3}$
- System sizes $(n^3) \times (n^3)$ with $n = 2^k, k \in \mathbb{N}$
- $l_{\mathsf{max}} = k - 1$, so coarsest system has dimension $(2^3) \times (2^3)$
- Accelerator-centric programming model
- Local Storage used, only
- Domain-splitting, i.e. each SPU handles part of the domain
- Limited LS results in limitation of local domain to 16^3
- Measurements carried out on a 4^3 partition, resulting in global grid size 128^3
- Time for one V(2,2)-cycle: 1050 μs



16^3 grid on every SPU

Other parallel multigrid solvers

- **hypre** (`http://computation.llnl.gov/casc/hypre/`)
  - Solver package
  - Includes structured, semi-structured and algebraic multigrid methods
  - Developed at Lawrence Livermore National Laboratory
- **Trilinos** (`http://trilinos.sandia.gov/`)
  - Multi-physics simulation environment
  - Includes smoothed aggregation-based multigrid method
  - Available from Sandia National Laboratory
- . . .

Outline

Multigrid

Parallel multigrid

Scalability of `pmg`

Application

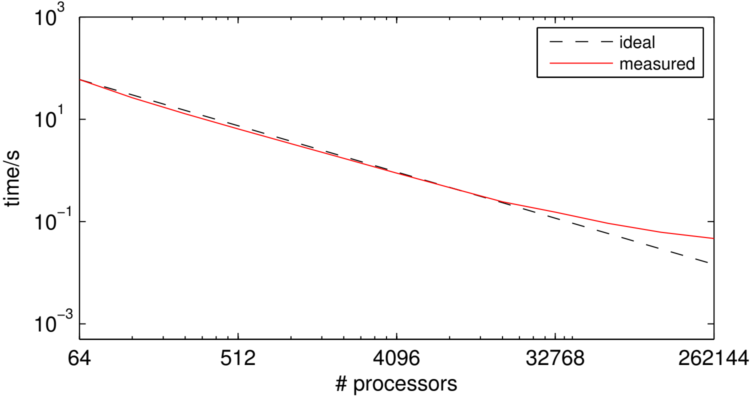
Scalability of application

Conclusion

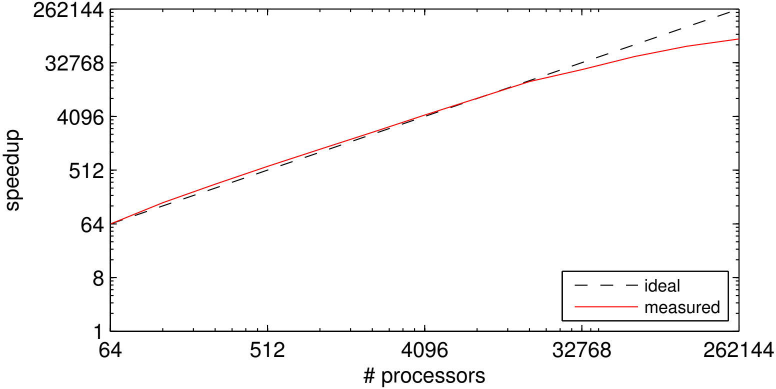
Scalability of the solver

- Strong scaling results on 64 – 262144 processors of Blue Gene/P (Jugene at Jülich Supercomputing Centre)
- 3-level circulant matrices
- $512 \times 512 \times 512$ unknowns
  ($\rightsquigarrow$ small problem, only 512 unknowns/core on 64 racks)
- System reduced to 1 unknown on coarsest level
- Toeplitz results do not vary much

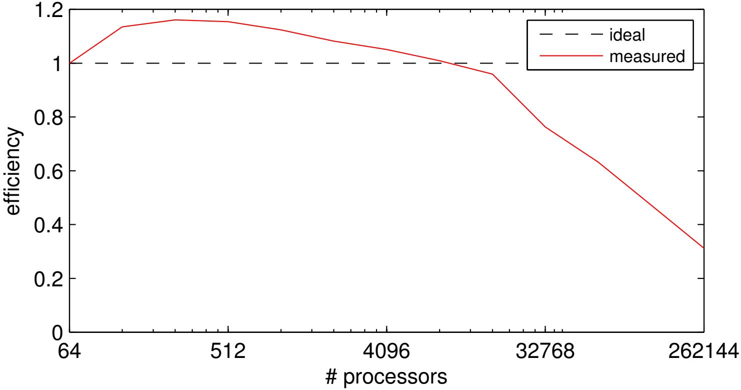
Strong scaling: $512 \times 512 \times 512$ circulant (time)



Strong scaling: $512 \times 512 \times 512$ circulant (speedup)



Strong scaling: $512 \times 512 \times 512$ circulant (efficiency)



Outline

Multigrid

Parallel multigrid

Scalability of `pmg`

Application

Scalability of application

Conclusion

Particle simulation

- Interaction modeled via pairwise or multi-particle potentials

$$U(\mathbf{x}_i, \mathbf{x}_j) \text{ or } U(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_m})$$

- Pairwise potentials often depend on distance, i.e.

$$U(\mathbf{x}_i, \mathbf{x}_j) = U(r_{ij}), \text{ where } r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

- Potentials either short- or long-ranged, where U long-ranged, if $U(r)$ decreases slower than $1/r^d$ for d dimensions
- Efficient scalable methods exist for short-ranged potentials, e.g. SPaSM (Gordon Bell prizes 1993, 1995, finalist 2005)
- Treatment of long-ranged potentials more involved
- Important potentials long-ranged (e.g. Coulomb potential)

Coulomb potential

- Particle interactions due to electrostatics described by Coulomb potential

$$U(r) = \frac{\epsilon_0}{4\pi}\frac{1}{r}$$

- Electrostatic energy V of system consisting of N particles with charges q_i and pairwise distances r_{ij} in free space given by

$$V = \frac{1}{2}\sum_{i=1}^{N} q_i \sum_{j=1, j\neq i}^{N} q_j U(r_{ij})$$

- Analogously for periodic systems with N particles per unit cell

$$V = \frac{1}{2}\left[\sum_{i=1}^{N} q_i \sum_{j=1, j\neq i}^{N} q_j U(r_{ij}) + \sum_{\mathbf{z}\in\mathbb{Z}^3\setminus\{\mathbf{0}\}} \sum_{i=1}^{N} q_i \sum_{j=1}^{N} q_j U(r_{ij}+\mathbf{z})\right]$$

Common methods

name	type		complexity	description
	free	periodic		
Ewald		✓	$\mathcal{O}(N^{3/2})$	Splits sum into short- and long-ranged part, solved in real and Fourier space, respectively
P3M		✓	$\mathcal{O}(N \log N)$	Introduces a mesh in Ewald summation to use FFT
Barnes-Hut	✓		$\mathcal{O}(N \log N)$	Organizes particles in clusters and computes particle-cluster interactions
FMM	✓	✓	$\mathcal{O}(N)$	Computes cluster-cluster interactions and thus reduces runtime compared to Barnes-Hut

Here: Alternative mesh-based method using multigrid (related to P3M)

MCM	(✓)	✓	$\mathcal{O}(N)$	Reformulation of problem as PDE, splitting into short- and long-ranged part

The need for supercomputing

Approaches in particle simulation:

1. **Monte Carlo**
   Stochastic minimization of system energy
2. **Molecular dynamics**
   Time evolution of systems by computing forces

Supercomputing needed for both approaches, because

- Energy/forces computed many times (many $\gg 1000$)
- Large number of particles desireable

(Parts of Monte Carlo embarrassingly parallel)

ScaFaCoS

Parallel Coulomb solvers currently developed in ScaFaCoS project (Uni Bonn, TU Chemnitz, FZ Jülich, Uni Stuttgart, Uni Wuppertal, Cognis, BASF), funded by BMBF.

Consistent mathematical formulation

- Electrostatic potential of particle i given by solution of following Poisson equation evaluated at $\mathbf{x}_i$ multplied by q_i

$$-\Delta\Phi_i(\mathbf{x}) = \rho_i := \frac{1}{\varepsilon_0} \sum_{j=1, j\neq i}^{N} q_j \delta(\|\mathbf{x} - \mathbf{x}_j\|_2)$$

- Replacement of δ-distributions by ρ_g, where
  - $\rho_g(\mathbf{x}) := g(|\mathbf{x}|)$
  - g sufficiently smooth and s.t. $\int_{\mathbb{R}^3} \rho_g(\mathbf{x}) d\mathbf{x} = 1$
  - Φ_g known analytically
- Only solution of the following Poisson equation needed

$$-\Delta\Phi(\mathbf{x}) = \rho := \frac{1}{\varepsilon_0} \sum_{j=1}^{N} q_j \rho_g(\|\mathbf{x} - \mathbf{x}_j\|_2)$$

Outline

Multigrid

Parallel multigrid

Scalability of `pmg`

Application

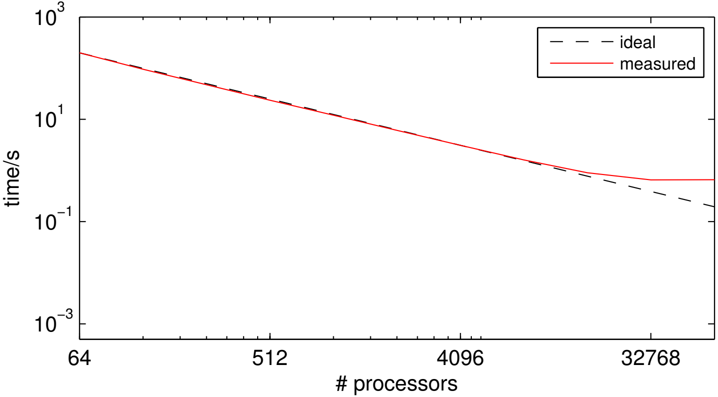
Scalability of application

Conclusion

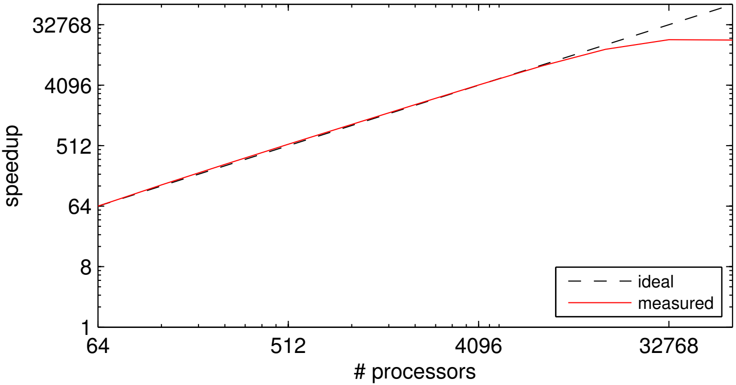
Scalability of the application

- Strong scaling results on 64 – 65536 processors of Blue Gene/P (Jugene at Jülich Supercomputing Centre)
- Artificial imhomogeneous example of a cloud wall with 9830400 particles
- Potential computed to a root means squared error of 10^{-3}
- Width of replacing charge distribution: 12 grid spacings
- Solving system with $512 \times 512 \times 512$ unknowns
- Share of multigrid solver: 7.1 %

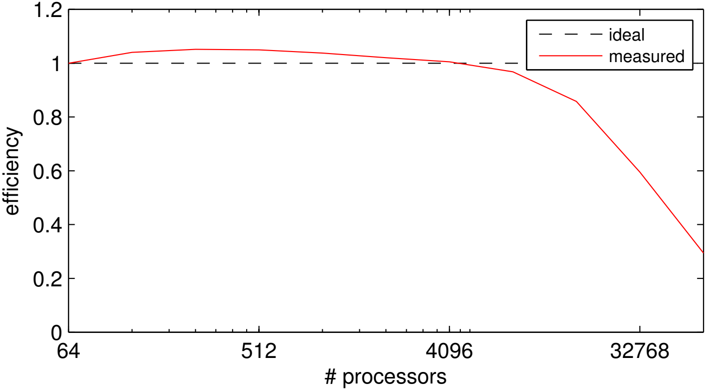
Strong scaling: Cloud wall with 9830400 particles (time)



Strong scaling: Cloud wall with 9830400 part. (speedup)



Strong scaling: Cloud wall with 9830400 part. (efficiency)



Outline

Multigrid

Parallel multigrid

Scalability of `pmg`

Application

Scalability of application

Conclusion

Conclusion and outlook

Conclusion

- Multigrid methods are optimal solvers for many systems
- Multigrid has excellent strong scaling behavior
- Torus architecture especially well-suited for many structured problems from physics
- Applications directly benefit from scalability
- Scalable application allows, e.g., for large scale molecular dynamics simulations with many time steps

Outlook

- Parallel multigrid still active area of research
- Further developments are necessary for exascale
- Approaches to reduce influence of $\mathcal{O}(\log P)$ factor include block-smoothers and aggressive coarsening

Multigrid methods for structured matrices on large scale supercomputers

Matthias Bolten

Bergische Universität Wuppertal

2012/09/03