Hiding global communication latency in Krylov methods for systems of linear equations

P. Ghysels[1,2], T. Ashby[2,3], K. Meerbergen[4], **W. Vanroose**[1]

[1] University of Antwerp - Dept Math & CS, Belgium (pieter.ghysels@ua.ac.be)

[2] Intel ExaScience Lab, Leuven, Belgium,

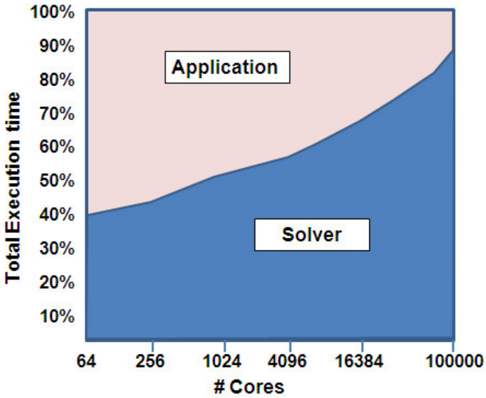
[3] Imec vzw, Leuven, Belgium,

[4] KU Leuven, - Dept CS, Leuven, Belgium

Thursday 28 june, 2012

Outline

Multiphysics application Charon on Cray XE6
GMRES with multi-level preconditioner



Report on the Workshop on Extreme-Scale Solvers: Transition to Future Architectures

Introduction

Increasing gap between computation and communication performance

- Floating point performance steadily increases
- Network latencies only go down marginally
  - Same for memory latency/BW (the “memory wall”)
- Avoid communication
  - Much work done at Berkeley (especially for dense algebra)
  - *s*-step Krylov methods, CA-GMRES, CA-CG, . . .
- Hide latency of communications
  - Also hide synchronization cost!

Generalized Minimal Residual (GMRES) method

- Dot-products are latency dominated
- Dot-products cannot be overlapped by other work in standard GMRES
- Other operations (stencil/AXPY) scale well

GMRES, classical Gram-Schmidt, solves $Ax = b$

$r_0 \leftarrow b - Ax_0$
$v_0 \leftarrow r_0/||r_0||_2$
for $i = 0, \ldots, m-1$ **do**
  $w \leftarrow Av_i$
  for $j = 0, \ldots, i$ **do**
    $h_{j,i} \leftarrow \langle w, v_j \rangle$
  end for
  $\tilde{v}_{i+1} \leftarrow w - \sum_{j=1}^{i} h_{j,i} v_j$
  $h_{i+1,i} \leftarrow ||\tilde{v}_{i+1}||_2$
  $v_{i+1} \leftarrow \tilde{v}_{i+1}/h_{i+1,i}$
  {apply Givens rotations to $h_{:,i}$ }
end for
$y_m \leftarrow \text{argmin}||H_{m+1,m} y_m - ||r_0||_2 e_1||_2$
$x \leftarrow x_0 + V_m y_m$

Y. Saad and M. H. Schultz (1986)

Sparse Matrix-Vector product

- Assume stencil computations
- Only communication with neighbors
- Good scaling

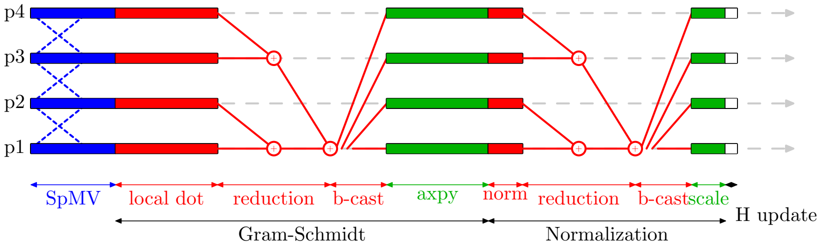
Dot-product

- Global communication
- Scales as $\log(P)$

Scalar vector multiplication, vector-vector addition

- No communication

GMRES iteration on 4 nodes

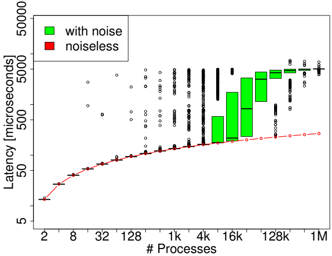


- Communication for SpMV can be overlapped with stencil calculations
- Typically a binomial tree reduction with height $\lceil \log_2(P) \rceil$
- Processor idling during all_reduce latency (not to scale)
- For modified Gram-Schmidt, (i+1) global reductions

Global reductions and variability

Load imbalance, OS jitter, hardware variability . . .
Global communication → expensive global **synchronization**

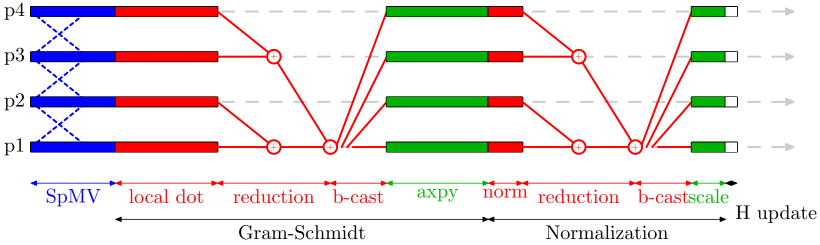
Influence of OS noise on time for a global reduction on Jaguar XT4



T. Hoefler, T. Schneider and A. Lumsdaine (2010)

Avoiding reduction for normalization

Can the extra reduction for the normalization be avoided?



Avoiding reduction for normalization

Given an orthonormal basis for the Krylov space $\mathcal{K}_{i+1}(A, v_0)$

$$V_{i+1} := [v_0, v_1, \ldots, v_{i-1}, v_i] \ ,$$

which satisfies the Arnoldi relation $AV_i = V_{i+1}H_{i+1,i}$

Avoiding reduction for normalization

Given an orthonormal basis for the Krylov space $\mathcal{K}_{i+1}(A, v_0)$

$$V_{i+1} := [v_0, v_1, \ldots, v_{i-1}, v_i] \ ,$$

which satisfies the Arnoldi relation $AV_i = V_{i+1}H_{i+1,i}$, then the steps

1: $z_{i+1} = Av_i$
2: Compute $\langle z_{i+1}, v_j \rangle$, for $j = 0, \ldots, i$ and $\|z_{i+1}\|$
3: $h_{j,i} = \langle z_{i+1}, v_j \rangle$, for $j = 0, \ldots, i$
4: $h_{i+1,i} = \sqrt{\|z_{i+1}\|^2 - \sum_{j=0}^{i} \langle z_{i+1}, v_j \rangle^2}$
5: $v_{i+1} = \left(z_{i+1} - \sum_{j=0}^{i} \langle z_{i+1}, v_j \rangle v_j \right) / h_{i+1,i}$

expand the basis to V_{i+2} and the Hessenberg matrix to $H_{i+2,i+1}$.

Avoiding reduction for normalization

Given an orthonormal basis for the Krylov space $\mathcal{K}_{i+1}(A, v_0)$

$$V_{i+1} := [v_0, v_1, \ldots, v_{i-1}, v_i] \ ,$$

which satisfies the Arnoldi relation $AV_i = V_{i+1}H_{i+1,i}$, then the steps

1: $z_{i+1} = Av_i$
2: Compute $\langle z_{i+1}, v_j \rangle$, for $j = 0, \ldots, i$ and $\|z_{i+1}\|$
3: $h_{j,i} = \langle z_{i+1}, v_j \rangle$, for $j = 0, \ldots, i$
4: $h_{i+1,i} = \sqrt{\|z_{i+1}\|^2 - \sum_{j=0}^{i} \langle z_{i+1}, v_j \rangle^2}$
5: $v_{i+1} = \left(z_{i+1} - \sum_{j=0}^{i} \langle z_{i+1}, v_j \rangle v_j \right) / h_{i+1,i}$

expand the basis to V_{i+2} and the Hessenberg matrix to $H_{i+2,i+1}$.

- Dot-products in line 2 can be combined in a single reduction
- Line 4 can lead to numerical instability
- Line 4 can lead to $\sqrt{}$-breakdown

Shifted matrix-vector product

improving stability

Given an orthonormal basis for the Krylov space $\mathcal{K}_{i+1}(A, v_0)$

$$V_{i+1} := [v_0, v_1, \ldots, v_{i-1}, v_i] \;, \quad \text{with} \quad AV_i = V_{i+1} H_{i+1,i}$$

and the shifts $\sigma_i \in \mathbb{C}$, then the steps

1: $z_{i+1} = (A - \sigma_i I)\, v_i$
2: Compute $\langle z_{i+1}, v_j \rangle$, for $j = 0, \ldots, i$ and $\|z_{i+1}\|$
3: $h_{j,i} = \langle z_{i+1}, v_j \rangle$, for $j = 0, \ldots, i-1$
4: $h_{i,i} = \langle z_{i+1}, v_i \rangle + \sigma_i$
5: $h_{i+1,i} = \sqrt{\|z_{i+1}\|^2 - \sum_{j=0}^{i} \langle z_{i+1}, v_j \rangle^2}$
6: $v_{i+1} = \left(z_{i+1} - \sum_{j=0}^{i} \langle z_{i+1}, v_j \rangle v_j \right) / h_{i+1,i}$

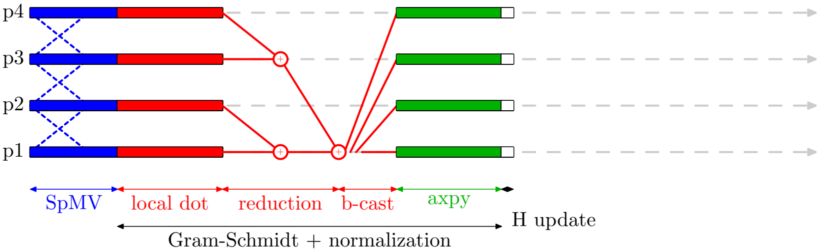
expand the basis to V_{i+2} and the Hessenberg matrix to $H_{i+2,i+1}$.

- If still $\surd$-breakdown occurs: re-orthogonalize or restart

Avoiding reduction for normalization

Only a single reduction per iteration

- Both reductions combined into 1
- Now reduces $i + 1$ floating point numbers i.s.o i



Overlapping dot-products and SpMV

Given the bases

$$V_i := [v_0, v_1, \ldots, v_{i-1}]$$
$$Z_{i+1} := [z_0, z_1, \ldots, z_{i-1}, z_i] \ ,$$

related by $z_0 = v_0$ and

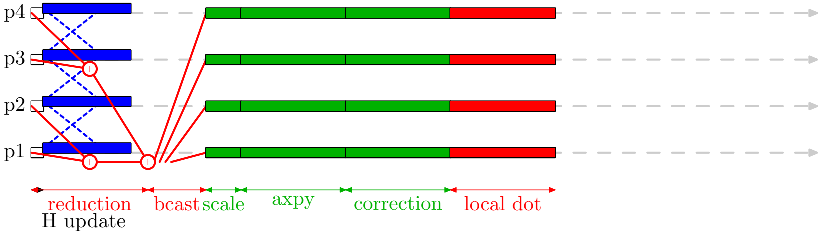
$$z_j = (A - \sigma I)\, v_{j-1} \ , \qquad (\sigma \in \mathbb{C})$$

then the following steps

1: Compute $\langle z_i, v_j \rangle$, for $j = 0, \ldots, i-1$ and $\|z_i\|$
2: $w = Az_i$
3: $h_{j,i-1} = \langle z_i, v_j \rangle$, for $j = 0, \ldots, i-2$
4: $h_{i-1,i-1} = \langle z_i, v_{i-1} \rangle + \sigma$
5: $h_{i,i-1} = \sqrt{\|z_i\|^2 - \sum_{j=0}^{i-1} \langle z_i, v_j \rangle^2}$
6: $v_i = \left(z_i - \sum_{j=0}^{i-1} \langle z_i, v_j \rangle v_j \right) / h_{i,i-1}$
7: $z_{i+1} = \left(w - \sum_{j=0}^{i-1} h_{j,i-1} z_{j+1} \right) / h_{i,i-1}$

expand the bases to V_{i+1} and Z_{i+1} and $H_{i,i-1}$ to $H_{i+1,i}$.

p(1)-GMRES



- Overlap dot-product global communication latency with SpMV
- Some additional flops compared to standard GMRES
  $\rightarrow$ **correction** on the figure
- Still waiting if reduction takes longer than SpMV

Deeper pipelining

What if a global all-reduce takes longer than a matrix-vector product?

$$V_{i-\ell+1} = [v_0, v_1, \ldots, v_{i-\ell}]$$
$$Z_{i+1} = [z_0, z_1, \ldots, z_{i-\ell}, \underbrace{z_{i-\ell+1}, \ldots, z_i}_{\ell}]$$

$$z_j = \begin{cases} v_0\,, & j = 0\,, \\ P_j(A)v_0\,, & 0 < j < \ell\,, \\ P_l(A)v_{j-\ell}\,, & j \geq \ell \end{cases} \quad \text{with} \quad P_i(t) = \prod_{j=0}^{i} (t - \sigma_j)$$

- Sliding window now ℓ iterations long
- All shifts zero, $z_j = A^\ell v_{j-\ell}$, called the monomial basis
  like power method, convergence to dominant eigenvector

The change of basis matrix

The Z_i basis satisfies a similar Arnoldi relation

$$AZ_i = Z_{i+1}B_{i+1,i}$$

with the upper Hessenberg *change of basis matrix* $B_{i+1,i}$

$$B_{i+1,i} = \begin{bmatrix} \sigma_0 & & & & & \\ 1 & \ddots & & & & \\ & \ddots & \sigma_{\ell-1} & & & \\ & & 1 & h_{0,0} & \dots & h_{0,i-\ell} \\ & & & h_{1,0} & & \\ & & & & \ddots & \vdots \\ & & & & & h_{i+1-\ell,i-l} \end{bmatrix}$$

See also

M. Hoemmen (2010)

E. Carson, N. Knight and J. Demmel (2011)

The Z_k basis can be made orthonormal by a QR factorization

$$Z_k = V_k G_k$$

Add new column to G_k using the dot-products

$$\begin{cases} \langle z_{k-1}, v_j \rangle, & j < k - \ell \\ \langle z_{k-1}, z_j \rangle, & k - \ell \leq j < k \end{cases}$$

$$V_{k-\ell} = [\overbrace{v_0, v_1, \ldots, v_{k-\ell-1}}]$$
$$Z_k = [z_0, z_1, \ldots, z_{k-\ell-1}, \underbrace{z_{k-\ell}, \ldots, z_{k-2}}_{\ell}, z_{k-1}]$$

$$g_{j,k-1} = \langle z_{k-1}, v_j \rangle = \left(\langle z_{k-1}, z_j \rangle - \sum_{m=0}^{j-1} g_{m,j} g_{m,i} \right) / g_{j,j} \,,$$
$$g_{j,j} = \sqrt{\langle z_j, z_j \rangle - \sum_{m=0}^{j-1} g_{m,j}^2}$$

Extending the Arnoldi recurrence

Using the Arnoldi recurrence relation

$$AV_k = V_{k+1}H_{k+1,k}$$

and the QR factorization

$$Z_k = V_k G_k$$

do an incremental update of Arnoldi's Hessenberg matrix

$$\begin{aligned} H_{k+1,k} &= V_{k+1}^T A V_k = V_{k+1}^T A Z_k G_k^{-1} = V_{k+1}^T Z_{k+1} B_{k+1,k} G_k^{-1} \\ &= G_{k+1} B_{k+1,k} G_k^{-1} \\ &= \ldots \\ &= \begin{bmatrix} H_{k,k-1} & (G_k b_{:,k} + g_{:,k+1} b_{k+1,k} - H_{k,k-1} g_{:,k}) g_{k,k}^{-1} \\ 0 & g_{k+1,k+1} b_{k+1,k} g_{k,k}^{-1} \end{bmatrix} \end{aligned}$$

p(ℓ)-GMRES

$r_0 \leftarrow b - Ax_0\,; \quad v_0 \leftarrow r_0/||r_0||\,; \quad z_0 \leftarrow v_0$

for $i = 0, \ldots, m+\ell$ **do**

$\quad w \leftarrow \begin{cases} (A - \sigma_i I) z_i, & i < \ell \\ A z_i, & i \geq \ell \end{cases}$

$\quad a \leftarrow i - \ell$

$\quad$ **if** $a \geq 0$ **then**

$\quad\quad g_{j,a+1} \leftarrow \left(g_{j,a+1} - \sum_{k=0}^{j-1} g_{k,j} g_{k,a+1}\right)/g_{j,j}, \; j = a - \ell + 2, \ldots, a$

$\quad\quad g_{a+1,a+1} \leftarrow \sqrt{g_{a+1,a+1} - \sum_{k=0}^{a} g_{k,a+1}^2}$

$\quad\quad$ # Check for breakdown and restart or re-orthogonalize if necessary

$\quad\quad$ **if** $a < \ell$ **then**

$\quad\quad\quad h_{j,a} \leftarrow \left(g_{j,a+1} + \sigma_a g_{j,a} - \sum_{k=0}^{a-1} h_{j,k} g_{k,a}\right)/g_{a,a}, \; j = 0, \ldots, a$

$\quad\quad\quad h_{a+1,a} \leftarrow g_{a+1,a+1}/g_{a,a}$

$\quad\quad$ **else**

$\quad\quad\quad h_{j,a} \leftarrow \left(\sum_{k=0}^{a+1-\ell} g_{j,k+\ell} h_{k,a-\ell} - \sum_{k=j-1}^{a-1} h_{j,k} g_{k,a}\right)/g_{a,a}, \; j = 0, \ldots, a$

$\quad\quad\quad h_{a+1,a} \leftarrow g_{a+1,a+1} h_{a+1-\ell,a-\ell}/g_{a,a}$

$\quad\quad$ **end if**

$\quad\quad v_a \leftarrow \left(z_a - \sum_{j=0}^{a-1} g_{j,a} v_j\right)/g_{a,a}$

$\quad\quad z_{i+1} \leftarrow \left(w - \sum_{j=0}^{a-1} h_{j,a-1} z_{j+\ell}\right)/h_{a,a-1}$

$\quad$ **end if**

$\quad g_{j,i+1} \leftarrow \begin{cases} \langle z_{i+1}, v_j \rangle, & j = 0, \ldots, a \\ \langle z_{i+1}, z_j \rangle, & j = a+1, \ldots, i+1 \end{cases}$

end for

$y_m \leftarrow \text{argmin} ||H_{m+1,m} y_m - ||r_0|| e_1||_2$

$x \leftarrow x_0 + V_m y_m$

Variation: p^1-GMRES

To avoid $\sqrt{}$-breakdown, compute norm explicitly

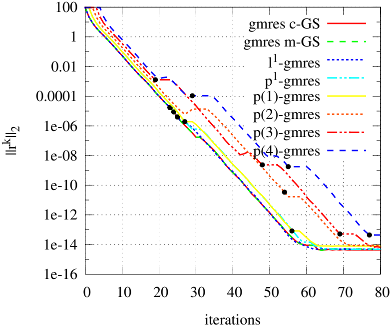
- Global reduction overlapped by 1 SpMV
- Delay of 2 iterations (one for orthogonalization, one for normalization)
- 1 reduction per iteration combining
  - orthogonalization of one step
  - normalization of previous step
- Implemented in PETSc 3.3 by Jed Brown
- Not necessary to store all z_i vectors

$r_0 \leftarrow b - Ax_0$; $\beta \leftarrow ||r_0||_2$
$v_0 \leftarrow r_0/\beta$; $z_0 \leftarrow v_0$
for $i = 0, \ldots, m+1$ **do**
  $w \leftarrow Az_i$
  if $i > 1$ **then**
    $v_{i-1} \leftarrow v_{i-1}/h_{i-1,i-2}$
    $z_i \leftarrow z_i/h_{i-1,i-2}$
    $w \leftarrow w/h_{i-1,i-2}$
    $h_{j,i-1} \leftarrow h_{j,i-1}/h_{i-1,i-2}$, $j = 0, \ldots, i-2$
    $h_{i-1,i-1} \leftarrow h_{i-1,i-1}/h^2_{i-1,i-2}$
  end if
  $z_{i+1} \leftarrow w - \sum_{j=0}^{i-1} h_{j,i-1} z_{j+1}$
  if $i > 0$ **then**
    $v_i \leftarrow z_i - \sum_{j=0}^{i-1} h_{j,i-1} v_j$
    $h_{i,i-1} \leftarrow ||v_i||_2$
  end if
  $h_{j,i} \leftarrow \langle z_{i+1}, v_j \rangle$, $j = 0, \ldots, i$
end for
$y_m \leftarrow \text{argmin}||H_{m+1,m} y_m - \beta e_1||_2$
$x \leftarrow x_0 + V_m y_m$

Random lower bidiagonal matrix

- $A \in \mathbb{R}^{500\times 500}$ with $A_{i,i} = 1 + p_i$ and $A_{i+1,i} = q_i$
  - p_i and q_i uniformly distributed on $[0, 1]$
- $\kappa_2(A) \approx 5$
- unsymmetric, nonnegative, not normal, positive definite and diagonally dominant

Monomial basis, $\sigma_i = 0$



Changing the basis for stability

- Newton basis
  - Take ℓ Ritz values as shifts for p(ℓ)-GMRES
- Chebyshev basis
  - Chebyshev polynomial, minimal over an ellipse surrounding the spectrum
  - Three term recurrence

$$z_{i+1} = \frac{\sigma_i}{\sigma_{i+1}} \left[\frac{2}{c}(d - A)z_i - \frac{\sigma_{i-1}}{\sigma_i} z_{i-1} \right]$$

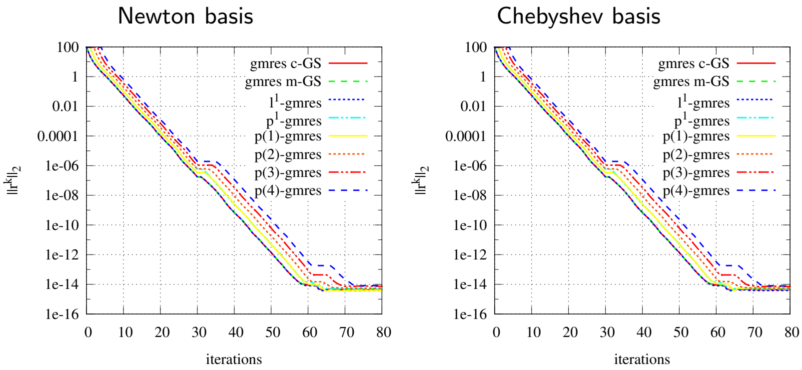
  - When all eigenvalues are real, zeros of Chebyshev polynomial are

$$\sigma_i = \frac{\lambda_{\min} + \lambda_{\max}}{2} + \left(\frac{\lambda_{\max} - \lambda_{\min}}{2} \right) \cos\left(\frac{(2i+1)\pi}{2l} \right), \quad \text{for } i = 0, \ldots, \ell - 1$$

- Complex shifts can be avoided for real matrices
- Put shifts in (modified) Leja ordering to maximize their internal distances

M. Hoemmen, PhD thesis (2010)

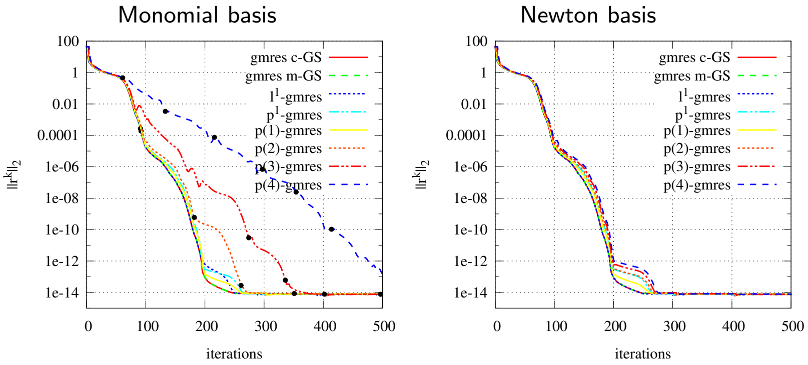
Changing the basis for stability



- Eigenvalues are randomly distributed between 1 and 2
  - Use ℓ Ritz values as shifts
  - Use zeros of the degree ℓ Chebyshev polynomial minimal over $[1, 2]$
- Residual for p(ℓ)-GMRES is delayed ℓ steps
- Restart after 30 iterations, re-fill pipeline, again ℓ steps delay

Finite difference stencil, PDE900

- real, unsymmetric, 900×900 matrix with $\kappa(A) \approx 2.9 \cdot 10^2$
- 5 point stencil for linear elliptic equation with Dirichlet boundary conditions on a regular 30×30 grid

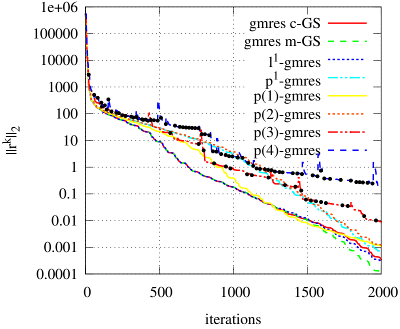


- Longer pipelining depth → more frequent restarts

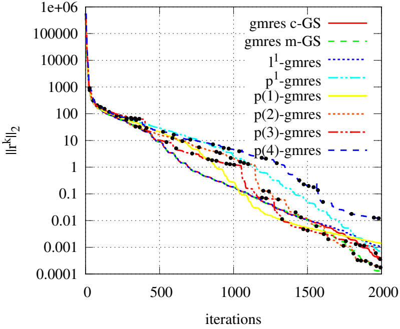
Oil reservoir simulation (ORSIRR1)

- real, unsymmetric, 1030×1030 matrix, 6858 non-zeros and $\kappa(A) \approx 100$
- Matrix Market, generated from oil reservoir simulation

Monomial basis

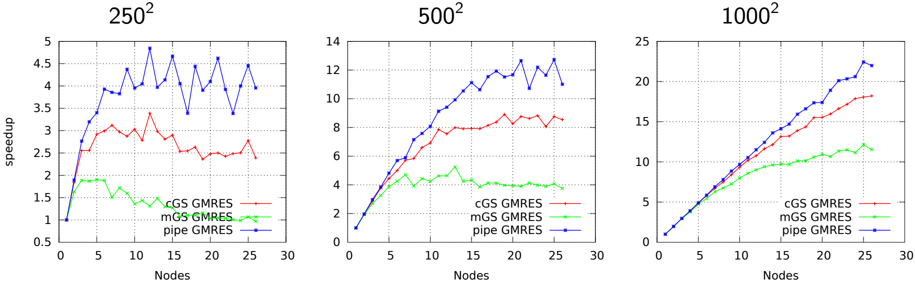


Newton basis



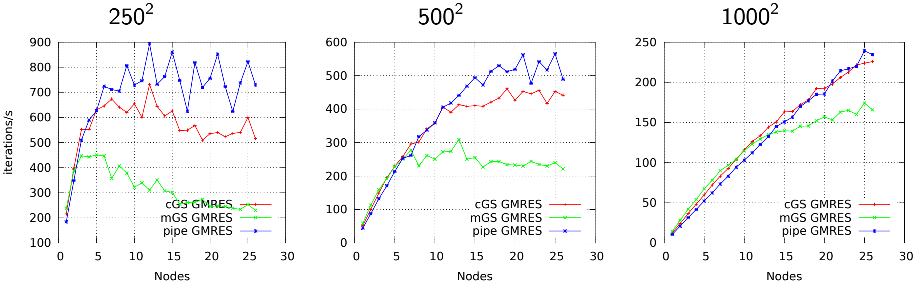
Benchmarks

- Implementation of p[1]-GMRES variation
- 2D Poisson problem on regular square grid
- Non-blocking all-reduce included in MPI-3 standard (In progress)
  - OpenMPI 1.5.5, using libNBC 1.0.1
  - MPICH2 1.5.2 → MPIX_Iallreduce



Benchmarks

- Implementation of p^1-GMRES variation
- 2D Poisson problem on regular square grid
- Non-blocking all-reduce included in MPI-3 standard (In progress)
  - OpenMPI 1.5.5, using libNBC 1.0.1
  - MPICH2 1.5.2 $\rightarrow$ MPIX_Iallreduce



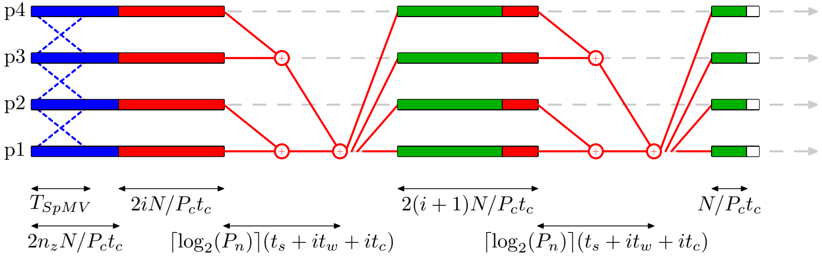
Performance model

Counting flops and messages

- A single message of size m takes $T(m) = t_s + mt_w$
- SpMV communication cost: send 6 faces of 3D cubic grid

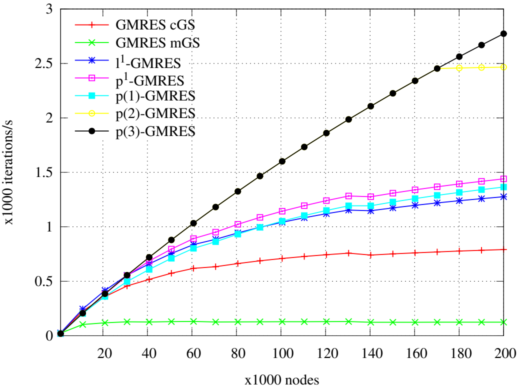
$$T_{\text{spmv}} = 6(t_s + (N/P_n)^{2/3} t_w)$$

- Matrix has n_z non-zeros $\rightarrow$ $2n_z$ flops per SpMV
- Single flop takes t_c



Predicted scaling on 200, 000 nodes

- Prediction of a strong scaling experiment on XT4 part of Cray Jaguar
- $N = 2000^3 = 8 \cdot 10^9$ unknowns



Conclusions & Outlook

Conclusions

- Presented GMRES variations where dot-product latency can be overlapped
- ...at the expense of some redundant flops
- Showed performance benefits on small and large clusters using benchmarks and performance model
- Pipelining can already help to hide expensive global synchronization costs
- Pipelined-GMRES in PETSc (thanks to Jed Brown) shows significant performance improvements compared to standard GMRES for strong scaling experiments

Outlook

- Efficient implementations, benchmarking
- Possible test scenarios
  - coarse solve in multigrid U-cycle
  - Schur complement solve
- Pipelining for short term recurrence methods
  - CG, BiCGStab, IDR, ...
- Hide communication between CPU and accelerator (like GPU or 'Xeon Phi')

References

- Y. Saad and M. H. Schultz (1986)
- E. Carson, N. Knight and J. Demmel (2011)
- T. Hoefler, T. Schneider and A. Lumsdaine (2010)
- M. Hoemmen, PhD thesis (2010)
- U.S. DoE, Report on the Workshop on Extreme-Scale Solvers: Transition to Future Architectures (2012)

- P. Ghysels, T. Ashby, K. Meerbergen and W. Vanroose, *Hiding global communication latency in the GMRES algorithm on massively parallel machines*, Submitted, Technical Report 04.2012.1
- P. Ghysels, B. De Fraine, P. Costanza, T. Ashby, K. Meerbergen and W. Vanroose, *Strong scaling improvements using communication hiding Krylov methods*, Submitted