Communication-avoiding linear algebra

Mark Hoemmen
University of California EECS ParLab
ParLab Retreat, Santa Cruz, CA
4-6 June 2008
mhoemmen@cs.berkeley.edu

Concepts

“Linear algebra” includes
- Dense: Direct methods (factorizations)
  - One-sided: LU, QR
  - Two-sided: Reduction to tridiagonal / Hessenberg form
- Eigenvalue decomposition, SVD
- Sparse: Iterative and direct methods
  - Krylov subspace iterations
  - One-sided factorizations (LU, QR)

“Communication” means
- Parallel: Data movement between processors
- Sequential: Data movement between levels of memory hierarchy
- # words (inverse bandwidth) and # messages (latency)

“Avoiding communication” means
- Given initial data layout, and
- Given type of algorithm (e.g., QR based on orthogonal linear transformations),
- Reduce # words and/or # messages asymptotically
- Without (seriously) affecting numerical accuracy

Parallel and sequential
- “Sequential” means: explicitly manage transfers between levels of memory hierarchy (a.k.a. “out-of-core”)

Motivation

Flops are free, bandwidth is money, latency is physics

Communication speeds improve exponentially slower than floating-point rate
- Flop rate: 59% / year (1988-2004, projected beyond)
  - Scales like # transistors (“free” from Moore’s Law)
- Memory bandwidth: only 23% / year (after 1995)
  - Limited by market forces: cost and power
  - Multicore means more processors sharing less bandwidth
- Memory latency: only 5.5% / year
  - Limited by physics (e.g., speed of light)
- Network & storage devices on similar curves

Factories

“Tall skinny” QR factorization
- \( m \times n \) matrix with \( m \gg n \), 1-D block row layout
- ScaLAPACK (existing code): \( O(n \log P) \) messages
- TSQR: \( O(\log P) \) messages (optional)

- Pick different tree shape to tune for architecture

Not just for tall skinny matrices
- Current parallel QR of matrix in 2-D block (cyclic) layout
- Factorization of block column ("panel") is bottleneck
- Instead, use TSQR for panel factorization in \( O(\log P) \) messages
- Example: \( n \times n \) matrix, 2-D block (cyclic) layout
- Current algorithm (ScaLAPACK’s PDGEQRF)
  - Parallel: \( O(n^2) \) more messages than lower bound
  - Sequential: \( O(n^2) \) more messages than lower bound
- Using TSQR for panel factorization gets within a log factor of lower bounds in parallel case, attains lower bound in sequential case

Same approach works for LU also
Unlike QR, need to pivot for correctness and accuracy.
- Naïve scheme: only within each processor’s block
  - No communication
  - But not stable (more processors means less accuracy)
  - However, this identifies a candidate pivot row for each block
- Permute these up to the top and redo the panel factorization without pivoting (fast!)
- Numerically stable in practice

\( k \)-step Krylov methods
- Current methods: Cyclic dependency between sparse matrix-vector product (SpMV) and orthogonalization / dot product

Algorithms

<table>
<thead>
<tr>
<th>Dense</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Cholesky already optimal)</td>
<td>Various Krylov methods (CG, GMRES)</td>
</tr>
<tr>
<td>Done</td>
<td>Future</td>
</tr>
<tr>
<td>QR, LU: 1-D, 2-D layouts</td>
<td>Two-sided factorizations (Direct) eigensolvers</td>
</tr>
<tr>
<td>LU: new pivoting scheme</td>
<td>Good preconditioners</td>
</tr>
<tr>
<td>Beyond linear algebra: PDEs</td>
<td></td>
</tr>
</tbody>
</table>

Results

- Significant modeled and benchmarked speedups
- Good for strong scaling (small problem on many processors)
- New algorithms appear to be stable in practice

Credits

- QR: James Demmel, Laura Grigori, Mark Hoemmen, Julien Langou (LAPACK Working Note # 204)
  - LU: James Demmel, Laura Grigori, Hua Xiang (INRIA tech report: http://hal.inria.fr/inria-00277901/fr/)
  - Matrix powers kernel for Krylov methods: James Demmel, Mark Hoemmen, Marghoob Mohiyuddin, Katherine Yelick (UC Berkeley tech report EECS-2007-123)
  - Krylov methods algebra and analysis: James Demmel, Mark Hoemmen (tech report in preparation)