One-Sided Matrix Factorizations

• Basic dense linear algebra factorizations: LU, QR, Cholesky
• We now have complete set of communication–optimal algorithms for QR implementations

“Communication” means

• Parallel: Data movement between processes
• Sequential: Data movement between levels of memory hierarchy
• Words (processor bandwidth) and # messages (latency)

Proofing Communication Lower Bounds

Matrix Multiplication Lower Bounds

• Results of [TT86] and [W94] imply that multiplying $n \times n$ matrices via a conventional $n^3$ algorithm requires communicating $O(n^2)$ words and $O(n^2)$ messages.

on $p$ processors machines, where $M$ is the memory size for each processor. These bounds hold for the sequential case, setting $p=1$.

• For distributed storage, $M = \ell n$. In this case the lower bounds are $O(p \ell n)$ words and $O(p \ell n)$ messages.

Reducing Matrix Multiplication to LU Factorization

• We can easily extend the matrix multiplication lower bounds to LU factorization via the following reduction

$$
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix} = 
\begin{bmatrix}
L & I \\
C & D
\end{bmatrix} 
\begin{bmatrix}
I & L^{-1} B \\
0 & I
\end{bmatrix},
$$

• Given matrices $A, B, C,$ and $D$, we can compute the product by forming the matrix above, performing LU factorization, and extracting $C$. 

Reducing Matrix Multiplication to Cholesky Factorization

• A first attempt at extending the matrix multiplication lower bounds to Cholesky factorization is given by the following reduction

$$
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix} = 
\begin{bmatrix}
L & I \\
C & D
\end{bmatrix} 
\begin{bmatrix}
I & L^{-1} B \\
0 & I
\end{bmatrix},
$$

• This is not a complete reduction: forming the matrix includes computing $C$, which requires considerable communication.

• We introduce “masking” factors $I$ and $0$ which mask communication that does not affect the matrix product.

• Using these masking numbers, we create the following reduction

$$
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix} = 
\begin{bmatrix}
L & I \\
C & D
\end{bmatrix} 
\begin{bmatrix}
I & L^{-1} B \\
0 & I
\end{bmatrix},
$$

• and $L$ is the lower triangular Cholesky factor of $C$.

• With this reduction, we no longer have to compute $C$, but we still compute $L$ accordingly. For details, see [BCH94].

• Proving that the matrix multiplication lower bounds apply to QR factorization is a bit trickier, see [DG94-95].

TSQR

• Communication bottleneck of one-sided factorizations is the column panel factorization (full column submatrix).

• Existing implementations require $\frac{2}{3} n^2 \log V$ messages to factor panel of width $V$.

• New TSQR algorithm requires only $\frac{2}{3} n^2 \log n$ messages.

• Reduction tree can be chosen to fit sequential, parallel, or heterogeneous architectures

• TSQR is as numerically stable as Householder QR, and it meets the communication lower bounds

TSLU

• Proving required to maintain numerical stability in LU factorization

• New approach: only pivoting within block (in computer for large matrices and small blocks)

• Equivalent to parallel pivoting when there are as many processors as rows

• Instead, use TSLU to select best pivot rows

• Each node factors to find the best pivot rows from its children’s rows

• Final pivot rows are promoted to parent node

Table 1: Comparison of parallel CALU and ScaLAPACK’s parallel LU factorization

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CALU</th>
<th>ScaLAPACK</th>
</tr>
</thead>
<tbody>
<tr>
<td># words</td>
<td>$O(n^2 \log n)$</td>
<td>$O(n^2 \log n)$</td>
</tr>
<tr>
<td># messages</td>
<td>$O(n^2 \log n)$</td>
<td>$O(n^2 \log n)$</td>
</tr>
</tbody>
</table>

Table 4: Communication-Avoiding QR

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Communication-Avoiding QR</th>
</tr>
</thead>
<tbody>
<tr>
<td># words</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td># messages</td>
<td>$O(n^2)$</td>
</tr>
</tbody>
</table>

Table 6: Communication-Avoiding LU

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Communication-Avoiding LU</th>
</tr>
</thead>
<tbody>
<tr>
<td># words</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td># messages</td>
<td>$O(n^2)$</td>
</tr>
</tbody>
</table>

Communicating QR

• For parallel cases, existing ScaLAPACK routine meets bandwidth limit bound, and with the right block size it meets latency lower bound

• For sequential cases, existing LAPACK routine meets bandwidth lower bound but not latency lower bound

• Block-cyclic storage required for blocked algorithm to meet latency, lower bound

Future Work

• Explore lower bounds for asymptotically faster algorithms (e.g., Strassen)

• Explore lower bounds for tree-based factorizations and eigensystem problems

• Implement sequential Cholesky algorithm with correct data structure to minimize latency

• Implement these algorithms on manycore architectures

References


