Low rank approximation and write avoiding algorithms

Laura Grigori

Alpines

Inria Paris - LJLL, UPMC

with A. Ayala, S. Cayrols, J. Demmel
Motivation - the communication wall

Time to move data >> time per flop
• Gap steadily and exponentially growing over time

Annual improvements
• Time / flop 59% (1995-2004) 34% (2006-2016)
• Interprocessor bandwidth 26%
• Interprocessor latency 15%
• DRAM latency 5.5%

DRAM latency:
• DDR2 (2007) ~ 120 ns 1x
• DDR4 (2014) ~ 45 ns 2.6x in 7 years
• Stacked memory ~ similar to DDR4

Time/flop
• 2006 Intel Yonah ~ 2GHz x 2 cores (32 GFlops/chip) 1x
• 2015 Intel Haswell ~2.3GHz x 16 cores (588 GFlops/chip) 18x in 9 years

Source: J. Shalf, LBNL
2D Parallel algorithms and communication bounds

- Memory per processor = $n^2 / P$, the lower bounds on communication are
  \[ \text{#words}_\text{moved} \geq \Omega \left( \frac{n^2}{P^{1/2}} \right), \quad \text{#messages} \geq \Omega \left( P^{1/2} \right) \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Minimizing #words (not #messages)</th>
<th>Minimizing #words and #messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cholesky</td>
<td>ScaLAPACK</td>
<td>ScaLAPACK</td>
</tr>
<tr>
<td>LU</td>
<td>ScaLAPACK uses partial pivoting</td>
<td>[LG, Demmel, Xiang, 08]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Khabou, Demmel, LG, Gu, 12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uses tournament pivoting</td>
</tr>
<tr>
<td>QR</td>
<td>ScaLAPACK</td>
<td>[Demmel, LG, Hoemmen, Langou, 08]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uses different representation of Q</td>
</tr>
<tr>
<td>RRQR</td>
<td>ScaLAPACK</td>
<td>[Demmel, LG, Gu, Xiang 13]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uses tournament pivoting, 3x flops</td>
</tr>
</tbody>
</table>

- Only several references shown, block algorithms (ScaLAPACK) and communication avoiding algorithms
- CA algorithms exist also for SVD and eigenvalue computation
Parallel write avoiding algorithms

Need to avoid writing suggested by emerging memory technologies, as NVMs:
- Writes more expensive (in time and energy) than reads
- Writes are less reliable than reads

Some examples:
- Phase Change Memory: Reads 25 us latency
  Writes: 15x slower than reads (latency and bandwidth)
  consume 10x more energy
- Conductive Bridging RAM - CBRAM
  Writes: use more energy (1pJ) than reads (50 fJ)
- Gap improving by new technologies such as XPoint and other FLASH alternatives, but not eliminated
Parallel write-avoiding algorithms

- Matrix A does not fit in DRAM (of size $M$), need to use NVM (of size $\frac{n^2}{P}$)

- Two lower bounds on volume of communication
  - Interprocessor communication: $\Omega \left( \frac{n^2}{P^{1/2}} \right)$
  - Writes to NVM: $\frac{n^2}{P}$

- Result: any three-nested loop algorithm (matrix multiplication, LU,..), must asymptotically exceed at least one of these lower bounds
  - If $\Omega \left( \frac{n^2}{P^{1/2}} \right)$ words are transferred over the network, then $\Omega \left( \frac{n^2}{P^{2/3}} \right)$ words must be written to NVM!

- Parallel LU: choice of best algorithm depends on hardware parameters

<table>
<thead>
<tr>
<th></th>
<th>#words interprocessor comm.</th>
<th>#writes NVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left-looking</td>
<td>$O((n^3 \log^2 P) / (P M^{1/2}))$</td>
<td>$O(n^2 / P)$</td>
</tr>
<tr>
<td>Right-looking</td>
<td>$O((n^2 \log P) / P^{1/2})$</td>
<td>$O((n^2 \log^2 P) / P^{1/2})$</td>
</tr>
</tbody>
</table>
Low rank matrix approximation

- Problem: given $m \times n$ matrix $A$, compute rank-$k$ approximation $ZW^T$, where $Z$ is $m \times k$ and $W^T$ is $k \times n$.

- Problem with diverse applications
  - from scientific computing: fast solvers for integral equations, H-matrices
  - to data analytics: principal component analysis, image processing, …

- Used in iterative process by multiplication with a set of vectors
  
  $Ax \rightarrow ZW^T x$

  Flops: $2mn \rightarrow 2(m + n)k$
Low rank matrix approximation

• Problem: given m x n matrix A, compute rank-k approximation ZW^T, where Z is m x k and W^T is k x n.

• Best rank-k approximation \( A_k = U_k \Sigma_k V_k^T \) is the rank-k truncated SVD of A

\[
\min_{\text{rank}(\tilde{A}_k) \leq k} \| A - \tilde{A}_k \|_2 = \| A - A_k \|_2 = \sigma_{k+1}(A)
\]

Original image, 707x256

Rank-75 approximation, SVD

Rank-38 approximation, SVD

Image source: https://upload.wikimedia.org/wikipedia/commons/a/a1/Alan_Turing_Aged_16.jpg
Low rank matrix approximation: trade-offs

- Truncated CA-SVD
- Truncated SVD
- Lanczos Algorithm
  - CA rank revealing QR
  - (strong) QRCP
  - LU with column/row tournament pivoting
  - LU with column, rook pivoting

Flops ➔ Accuracy ➔ Communication

Randomized algorithms ??
Select k cols using tournament pivoting

Partition \( A = (A_1, A_2, A_3, A_4) \).
Select \( k \) cols from each column block,
by using QR with column pivoting

At each level \( i \) of the tree
At each node \( j \) do in parallel
Let \( A_{v,i-1}, A_{w,i-1} \) be the cols selected by
the children of node \( j \)
Select \( b \) cols from \( (A_{v,i-1}, A_{w,i-1}) \),
by using QR with column pivoting
Return columns in \( A_{ji} \)
LU_CRTP: LU with column/row tournament pivoting

- Given $A$ of size $m \times n$, compute a factorization

  $$P_r A P_c = \begin{pmatrix} \overline{A}_{11} & \overline{A}_{12} \\ \overline{A}_{21} & \overline{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \overline{A}_{11}^{-1} I \\ \overline{A}_{21} \overline{A}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \overline{A}_{11} & \overline{A}_{12} \\ \overline{A}_{21} & \overline{A}_{22} \end{pmatrix},$$

  $$S(\overline{A}_{11}) = \overline{A}_{22} - \overline{A}_{21} \overline{A}_{11}^{-1} \overline{A}_{12},$$

  where $\overline{A}_{11}$ is $k \times k$, $P_r$ and $P_c$ are chosen by using tournament pivoting

- LU_CRTP factorization satisfies

  \[
  1 \leq \frac{\sigma_i(A)}{\sigma_i(\overline{A}_{11})}, \quad \frac{\sigma_j(S(\overline{A}_{11}))}{\sigma_{k+j}(A)} \leq \sqrt{(1 + F^2(n - k))(1 + F^2(m - k))},
  \]

  \[
  \|S(\overline{A}_{11})\|_{\text{max}} \leq \min\left((1 + F\sqrt{k})\|A\|_{\text{max}}, F\sqrt{1 + F^2(m - k)\sigma_k(A)}\right)
  \]

  for any $1 \leq i \leq k$ and $1 \leq j \leq \min(m,n) - k$, $F \leq \frac{1}{\sqrt{2k}}(n/k)^{\log_2(2\sqrt{2k})}$
• Given LU_CRTP factorization

\[ P_r A P_c = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \bar{A}_{11} \\ \bar{A}_{21}\bar{A}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{11} & \bar{A}_{12} \end{pmatrix}, \]

the rank - k CUR approximation is

\[ \tilde{A}_k = \begin{pmatrix} I & \bar{A}_{11} \\ \bar{A}_{21}\bar{A}_{11}^{-1} & \bar{A}_{11} \end{pmatrix} = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} \]

• \( \bar{A}_{11}^{-1} \) is never formed, its factorization is used when \( \tilde{A}_k \) is applied to a vector

• In randomized algorithms, \( U = C^+ A R^+ \), where \( C^+ \), \( R^+ \) are Moore-Penrose generalized inverses
Results for image of size 256x707

Original image, 707x256

LU_CRTP: Rank-38 approx.

LUPP: Rank-75 approximation

SVD: Rank-38 approx.

SVD: Rank-75 approximation

LU_CRTP: Rank-75 approx.
Tournament pivoting for sparse matrices

A has arbitrary sparsity structure\hspace{1cm} G(A^T A) is an \( n^{1/2} \) - separable graph

\[
\text{flops}(TP_{FT}) \leq 2\text{nnz}(A)k^2 \\
\text{flops}(TP_{BT}) \leq 8\frac{\text{nnz}(A)}{P}k^2 \log \frac{n}{k}
\]

\[
\text{flops}(TP_{FT}) \leq O\left(\text{nnz}(A)k^{3/2}\right) \\
\text{flops}(TP_{BT}) \leq O\left(\frac{\text{nnz}(A)}{P}k^{3/2} \log \frac{n}{k}\right)
\]

- Randomized algorithm by Clarkson and Woodruff, STOC’13

Given \( n \times n \) matrix \( A \), it computes \( LDW^T \), where \( D \) is \( k \times k \), such that

\[
\|A - LDW^T\|_F \leq (1 + \varepsilon)\|A - A_k\|_F, \hspace{1cm} A_k \text{ is the best rank - } k \text{ approximation.}
\]

\[
\text{flops} \leq O(\text{nnz}(A)) + n\varepsilon^{-4} \log^{O(1)}(n\varepsilon^{-4})
\]

- Tournament pivoting is faster if
  \[
  \varepsilon \leq \frac{1}{(\text{nnz}(A)/n)^{1/4}}
  \]
  or if \( \varepsilon = 0.1 \) and \( \text{nnz}(A)/n \leq 10^4 \)
## Performance results

Comparison of number of nonzeros in the factors L/U, Q/R.

<table>
<thead>
<tr>
<th>Name/size</th>
<th>Nnz A(:,1:K)</th>
<th>Rank K</th>
<th>Nnz QRCP/ LU_CRTP</th>
<th>Nnz LU_CRTP/LUPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rfdevice 74104</td>
<td>633</td>
<td>128</td>
<td>10.0</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td>2255</td>
<td>512</td>
<td>82.6</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>4681</td>
<td>1024</td>
<td>207.2</td>
<td>0.0</td>
</tr>
<tr>
<td>Parab_fem 525825</td>
<td>896</td>
<td>128</td>
<td>-</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>3584</td>
<td>512</td>
<td>-</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>7168</td>
<td>1024</td>
<td>-</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Performance results

Selection of 256 columns by tournament pivoting

Edison, Cray XC30 (NERSC) – 2x12-core Intel Ivy Bridge (2.4 GHz)

Tournament pivoting uses SPQR (T. Davis) + dGEQP3 (Lapack), time in secs

Matrices: \( n \times n \) \quad \text{vs} \quad \text{n x n/32}

- Parab_fem: 528825 x 528825          528825 x 16432
- Mac_econ:  206500 x 206500          206500 x 6453

<table>
<thead>
<tr>
<th>Matrices</th>
<th>Time ( n \times 2k )</th>
<th>Time ( n \times n/32 )</th>
<th>Number of MPI processes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPQR+GEQP3</td>
<td></td>
<td>16          32          64          128         256         512         1024</td>
</tr>
<tr>
<td>Parab_fem</td>
<td>0.26</td>
<td>0.26+1129</td>
<td>46.7        24.5        13.7        8.4         5.9         4.8         4.4</td>
</tr>
<tr>
<td>Mac_econ</td>
<td>0.46</td>
<td>25.4+510</td>
<td>132.7       86.3        111.4       59.6        27.2        -           -</td>
</tr>
</tbody>
</table>
Conclusions

• Deterministic low rank approximation algorithm
  • Accuracy close to rank revealing QR factorization
  • Complexity close to randomized algorithms

• Future work
  • Design algorithms that do not need explicitly the matrix
  • Do a thorough comparison with randomized algorithms

Thanks to: EC H2020 NLAFET
Further information:
http://www-rocq.inria.fr/who/Laura.Grigori/