Parallel matrix-matrix multiplication

HPC for numerical methods and data analysis

Laura Grigori

EPFL and PSI

October 8, 2024

Plan

Motivation for reducing communication

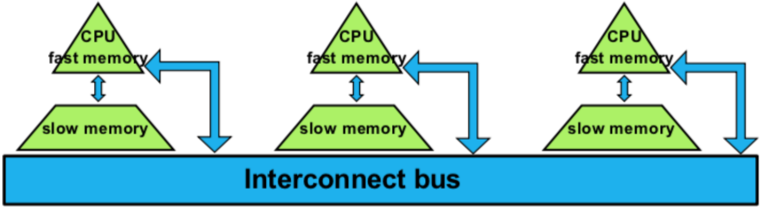
Minimizing communication in dense linear algebra

Communication lower bounds with tight constants

- Rectangular matrix multiplication

Motivation: the communication wall

- Runtime of an algorithm is the sum of:
  - #flops x time_per_flop
  - # words_moved / bandwidth
  - # messages x latency
- Time to move data $\gg$ time per flop
  - Gap steadily and exponentially growing over time



The communication wall: compelling numbers

Time/flop 59% annual improvement up to 2004[1]

2008 Intel Nehalem 3.2GHz$\times$4 cores (51.2 GFlops/socket) 1x

2020 A64FX 2.2GHz$\times$48 cores (3.37 TFlops/socket DP)[2] 66x in 12 years

DRAM latency: 5.5% annual improvement up to 2004[1]

DDR2 (2007) 120 ns 1x

DDR4 (2014) 45 ns 2.6x in 7 years

Stacked memory similar to DDR4

Network latency: 15% annual improvement up to 2004[1]

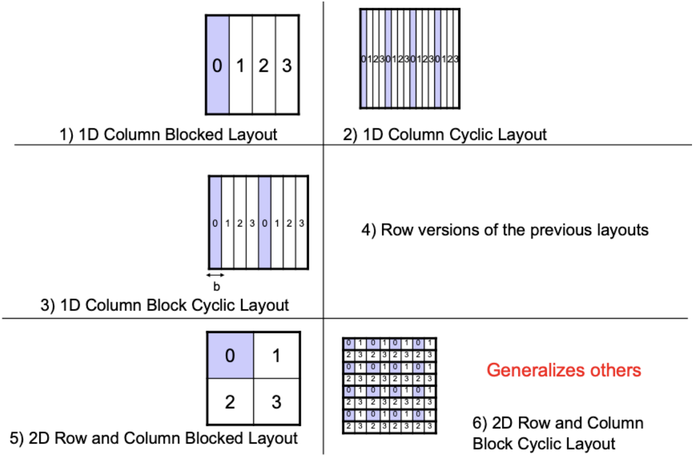
Interconnect: a few μs MPI latency

[1] "Getting up to speed, The future of supercomputing" 2004, data from 1995-2004

[2] Fugaku supercomputer `https://www.top500.org/system/179807/`

Matrix distributions

Suppose each processor has enough memory to store $1/P$-th of the data, $M = \Theta(n^2/P)$



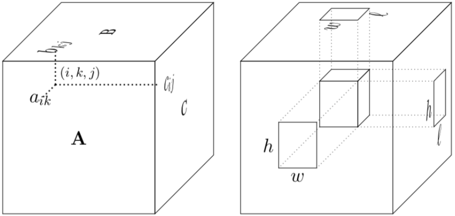
Communication lower bounds

Problem:

- Given a machine with two levels of memory, a fast memory of size M and a slow memory of infinite size
- Compute $\mathbf{C} = \mathbf{C} + \mathbf{A}\mathbf{B}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times r}$, for each element,

$$\mathbf{C}(i,j) = \mathbf{C}(i,j) + \sum_{k=1}^{n} \mathbf{A}(i,k) \cdot \mathbf{B}(k,j), \tag{1}$$

- What is the lower bound on the number of transfers between the slow and fast memory (number of reads and writes) ?



(a) One lattice point and its projections (b) Rectangular prism and its projections

$$\sqrt{wh \cdot w\ell \cdot h\ell} = wh\ell$$

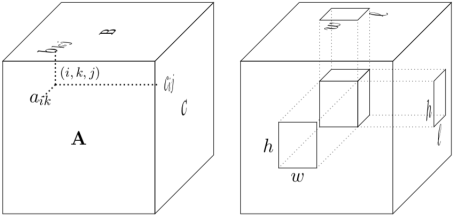
→ Rectangular prism most efficient shape for maximizing volume

Lemma 1 ([Loomis and Whitney, 1949])

V a finite set of lattice points (i, j, k) in $\mathbb{R}^3$,
V_x projection of V in x-direction: points (y, z) s.t. $\exists x'$ and $(x', y, z) \in V$
ditto for V_y and V_z. Then:

$$|V| \leq \sqrt{|V_x||V_y||V_z|}.$$

Lower bounds for matrix multiplication $\mathbf{C} = \mathbf{AB}$



(a) One lattice point and its projections (b) Rectangular prism and its projections

- $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{B} \in \mathbb{R}^{n\times r}, \mathbf{C} \in \mathbb{R}^{m\times r}$
- Instruction stream broken into segments
- Each segment contains x loads and stores
- M fast memory size

Using Lemma 1 and AM-GM inequality, bound total scalar multiplications per segment:

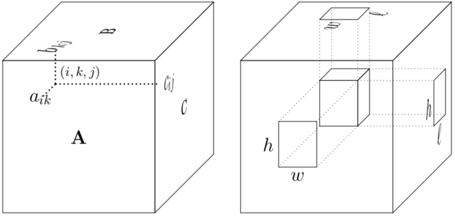
$$|V| \leq \sqrt{|V_{\mathbf{A}}||V_{\mathbf{B}}||V_{\mathbf{C}}|} \leq \left(\frac{|V_{\mathbf{A}}| + |V_{\mathbf{B}}| + |V_{\mathbf{C}}|}{3}\right)^{3/2} \leq \left(\frac{M+x}{3}\right)^{3/2}$$

$$\#segments \geq \left\lfloor \frac{mnr}{\left(\frac{M+x}{3}\right)^{3/2}} \right\rfloor \implies \#loads/stores \geq x \left\lfloor \frac{mnr}{\left(\frac{M+x}{3}\right)^{3/2}} \right\rfloor$$

or

$$\#loads/stores \geq 2M \left\lfloor \frac{mnr}{M^{3/2}} \right\rfloor \geq \frac{2mnr}{\sqrt{M}} - 2M$$

Lower bounds for matrix multiplication $\mathbf{C} = \mathbf{AB}$



(a) One lattice point and its projections (b) Rectangular prism and its projections

Lower bound for volume of communication:

$$\#loads/stores \geq 2M \left\lfloor \frac{mnr}{M^{3/2}} \right\rfloor \geq \frac{2mnr}{\sqrt{M}} - 2M$$

- Bound attained by a block algorithm if $\min\{m, n, k\} \geq \sqrt{M+1} - 1$
- For square matrices, algorithm uses $b \times 1$ blocks of A, $1 \times b$ blocks of B and $b \times b$ blocks of C, with $b \leq \sqrt{M+1} - 1$

Lower bounds for parallel algorithms

Memory dependent lower bounds

Compute $C = AB$ on P processors, assume $m = n = k$:

$$W = \#words\_moved \geq \frac{2n^3}{P\sqrt{M}} - M$$

- 2D algorithms: $M = \Theta(n^2/P) \implies W = \Omega(n^2/P^{1/2})$
- 3D algorithms: $M = \Theta(n^2/P^{2/3}) \implies W = \Omega(n^2/P^{2/3})$
- 2.5D algorithms: $M = \Theta(cn^2/P) \implies W = \Omega(n^2/(cP)^{1/2})$

Lower bounds for parallel algorithms

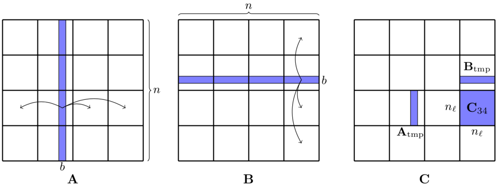
Memory dependent lower bounds

Compute $C = AB$ on P processors, assume $m = n = k$:

$$W = \#words\_moved \geq \frac{2n^3}{P\sqrt{M}} - M$$

- 2D algorithms: $M = \Theta(n^2/P) \implies W = \Omega(n^2/P^{1/2})$
- 3D algorithms: $M = \Theta(n^2/P^{2/3}) \implies W = \Omega(n^2/P^{2/3})$
- 2.5D algorithms: $M = \Theta(cn^2/P) \implies W = \Omega(n^2/(cP)^{1/2})$

SUMMA matrix multiplication



Cost of communication:

$$\beta \cdot O\left(\frac{n^2}{\sqrt{P}}\right) + \alpha \cdot O\left(\frac{n}{b}\log P\right)$$

Require: A, B, C are $n \times n$ matrices in identical 2D block distribution across processors
Require: Processors arranged in $\sqrt{P} \times \sqrt{P}$ grid where $n_\ell = n/\sqrt{P}$ is an integer
Require: Proc (I, J) owns $n_\ell \times n_\ell$ submatrix $M_{IJ} = M((I-1)n_\ell+1 : In_\ell, (J-1)n_\ell+1 : Jn_\ell)$

```
function C = SUMMA(C, A, B, b)
    (I, J) = MyProcID()
    for K = 1 to √P do
        for k = 1 to n_ℓ/b do
            Proc (I, K) broadcasts A_IK(:, (k−1)b+1:kb) to proc(I, :), store in A_tmp
            Proc (K, J) broadcasts B_KJ((k−1)b+1:kb, :) to proc(:, J), store in B_tmp
            C_IJ = C_IJ + A_tmp · B_tmp
        end for
    end for
end function
```

Presentation from van de Geijn and Watts '96

3D Parallel Matrix Multiplication

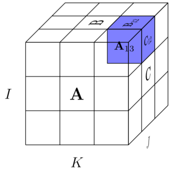


Figure: Proc(1,2,3)

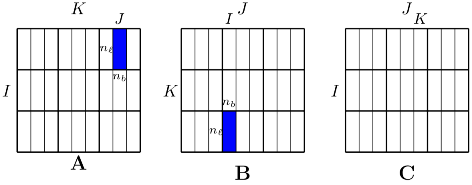


Figure: Initial distribution of A and B

Processors arranged in $\sqrt[3]{P} \times \sqrt[3]{P} \times \sqrt[3]{P}$ grid
A, B are $n \times n$ matrices in 2D block distribution across $\sqrt[3]{P} \times (\sqrt[3]{P})^2$ processor grid where $n_\ell = n/\sqrt[3]{P}$ and $n_b = n/(\sqrt[3]{P})^2$ are integers
Processor (I, J, K) owns $n_\ell \times n_b$ submatrices

$$A_{IKJ} = A_{IK}(:, (J-1)n_b+1 : Jn_b)$$

$$B_{KJI} = B_{KJ}(:, (I-1)n_b+1 : In_b)$$

$$C_{IJK} = C_{IJ}(:, (K-1)n_b+1 : Kn_b)$$

3D Parallel Matrix Multiplication

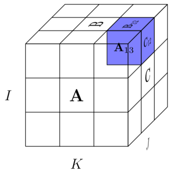


Figure: Proc(1,2,3)

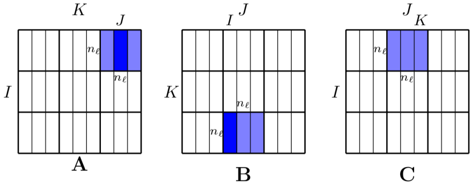


Figure: All-gathers and local computation

Assert: $C = C + AB$ is $n \times n$ matrix in 2D block distribution across processors so that processor (I, J, K) owns C_{IJK}

function $C = \text{3D-M}\textsc{atmul}(C, A, B)$
  $(I, J, K) = MyProcID()$
  All-gather A_{IKJ} across $Proc(I, :, K)$, store in A_{IK}
  All-gather B_{KJI} across $Proc(:, J, K)$, store in B_{KJ}
  $\overline{C}_{IJ} = A_{IK} \cdot B_{KJ}$
  Reduce-scatter $\overline{C}_{IJ}$ across $Proc(I, J, :)$, combine result with C_{IJK}
end function

3D Parallel Matrix Multiplication

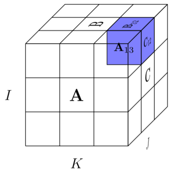


Figure: Proc(1,2,3)

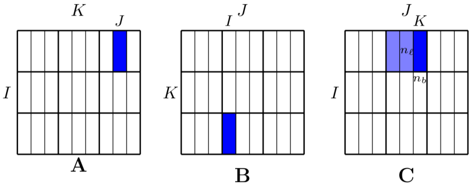


Figure: Reduce-scatter to obtain final distribution of C

Assert: $C = C + AB$ is $n \times n$ matrix in 2D block distribution across processors so that processor (I, J, K) owns C_{IJK}

function $C = \text{3D-}\textsc{Matmul}(C, A, B)$
    $(I, J, K) = MyProcID()$
    All-gather A_{IKJ} across $Proc(I, :, K)$, store in A_{IK}
    All-gather B_{KJI} across $Proc(:, J, K)$, store in B_{KJ}
    $\overline{C}_{IJ} = A_{IK} \cdot B_{KJ}$
    Reduce-scatter $\overline{C}_{IJ}$ across $Proc(I, J, :)$, combine result with C_{IJK}
end function

3D Parallel Matrix Multiplication

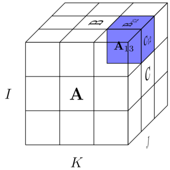


Figure: Proc(1,2,3)

Communication cost

$$\beta \cdot O\left(\frac{n^2}{P^{2/3}}\right) + \alpha \cdot O(\log P)$$

Assert: $C = C + AB$ is $n \times n$ matrix in 2D block distribution across processors so that processor (I, J, K) owns C_{IJK}

function $C = \text{3D-M}\textsc{atmul}(C, A, B)$
    $(I, J, K) = MyProcID()$
    All-gather A_{IKJ} across $Proc(I, :, K)$, store in A_{IK}
    All-gather B_{KJI} across $Proc(:, J, K)$, store in B_{KJ}
    $\overline{C}_{IJ} = A_{IK} \cdot B_{KJ}$
    Reduce-scatter $\overline{C}_{IJ}$ across $Proc(I, J, :)$, combine result with C_{IJK}
end function

Communication Complexity of Dense Linear Algebra

Matrix multiply, using $2n^3$ flops (sequential or parallel)

- Hong-Kung (1981), Irony/Tishkin/Toledo (2004)
- Lower bound on Bandwidth $= \Omega(\# flops / M^{1/2})$
- Lower bound on Latency $= \Omega(\# flops / M^{3/2})$

Same lower bounds apply to LU using reduction

- Demmel, LG, Hoemmen, Langou, tech report 2008, SISC 2012

$$\begin{pmatrix} I & & -B \\ A & I & \\ & & I \end{pmatrix} = \begin{pmatrix} I & & \\ A & I & \\ & & I \end{pmatrix} \begin{pmatrix} I & & -B \\ & I & AB \\ & & I \end{pmatrix}$$

And to almost all direct linear algebra

[Ballard, Demmel, Holtz, Schwartz, 09]

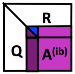
also extended to fast linear algebra

2D Parallel algorithms and communication bounds

Memory per processor $= \Theta(n^2/P)$, lower bounds on communication:

$$\#\text{words\_moved} \geq \Omega(n^2/\sqrt{P}), \quad \#\text{messages} \geq \Omega(\sqrt{P})$$

Most classical algorithms (ScaLAPACK) attain
lower bounds on #words_moved
but do not attain lower bounds on #messages



	ScaLAPACK	CA algorithms
LU	partial pivoting	tournament pivoting (TP) [LG, Demmel, Xiang, 08]
QR	column based Householder	reduction based Householder [Demmel, LG, Hoemmen, Langou, 08] [Ballard, Demmel, LG, Jacquelin, Nguyen, Solomonik, 14]
RRQR	column pivoting	tournament pivoting (TP) [Demmel, LG, Gu, Xiang 13] randomized QRCP +TP [Martinsson, Voronin 15], [Duersch, Gu 15]
Eig(A)	Hessenberg/QR alg	[Ballard, Demmel, Dumitriu 10]

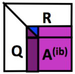
Only several references shown

2D Parallel algorithms and communication bounds

Memory per processor $= \Theta(n^2/P)$, lower bounds on communication:

$$\#\text{words\_moved} \geq \Omega(n^2/\sqrt{P}), \quad \#\text{messages} \geq \Omega(\sqrt{P})$$

Most classical algorithms (ScaLAPACK) attain
lower bounds on #words_moved
but do not attain lower bounds on #messages



	ScaLAPACK	CA algorithms
LU	partial pivoting	tournament pivoting (TP) [LG, Demmel, Xiang, 08]
QR	column based Householder	reduction based Householder [Demmel, LG, Hoemmen, Langou, 08] [Ballard, Demmel, LG, Jacquelin, Nguyen, Solomonik, 14]
RRQR	column pivoting	tournament pivoting (TP) [Demmel, LG, Gu, Xiang 13] randomized QRCP +TP [Martinsson, Voronin 15], [Duersch, Gu 15]
Eig(A)	Hessenberg/QR alg	[Ballard, Demmel, Dumitriu 10]

Only several references shown

Conclusions

- Some of the methods discussed available in libraries
  - LAPACK, ScaLAPACK, SLATE, Spark, GNU Scientific library, Cray scientific library
- Material based on upcoming book on *Communication-Avoiding Algorithms*, with G. Ballard, E. Carson, J. Demmel

References (1)

Loomis, L. H. and Whitney, H. (1949).
An inequality related to the isoperimetric inequality.
Bulletin of the American Mathematical Society, 55(10):961 – 962.