Minimizing Arithmetic & Communication Costs for Faster Matrix Computations

Oded Schwartz
The Hebrew University

ACA'17, July 17-21

Based on joint papers with

Grey Ballard, James Demmel, Andrew Gearhart, Olga Holtz, Elaye Karstadt, Ben Lipshitz, Yishai Oltchik, and Sivan Toledo.

Research supported by: ISF, BSF, Intel, Ministry of Science, Minerva, Einstein Foundation

Supercomputing resources by: PRACE, LinkSCEEM, ALCF, ORNL

Model & Motivation

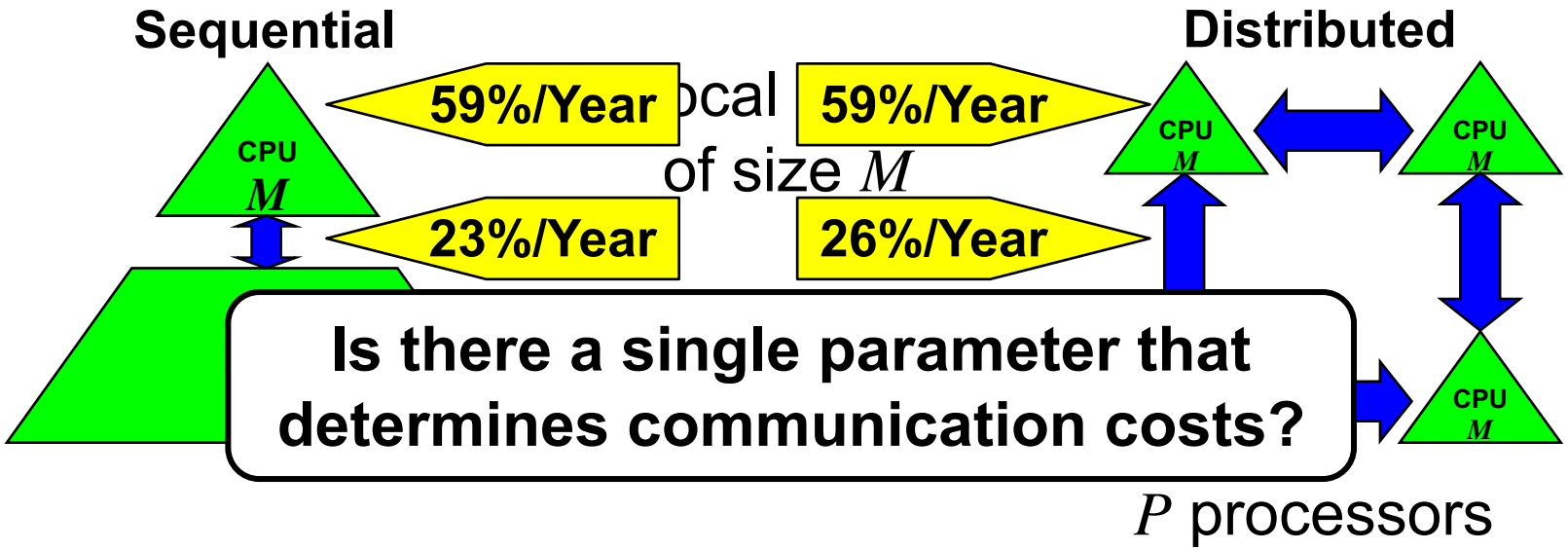
Two kinds of costs:

- Arithmetic (FLOPs)
- Communication: moving data

$$Running\ time = \gamma \cdot \#FLOPs + \beta \cdot \#Words \quad (+ \alpha \cdot \#Messages)$$

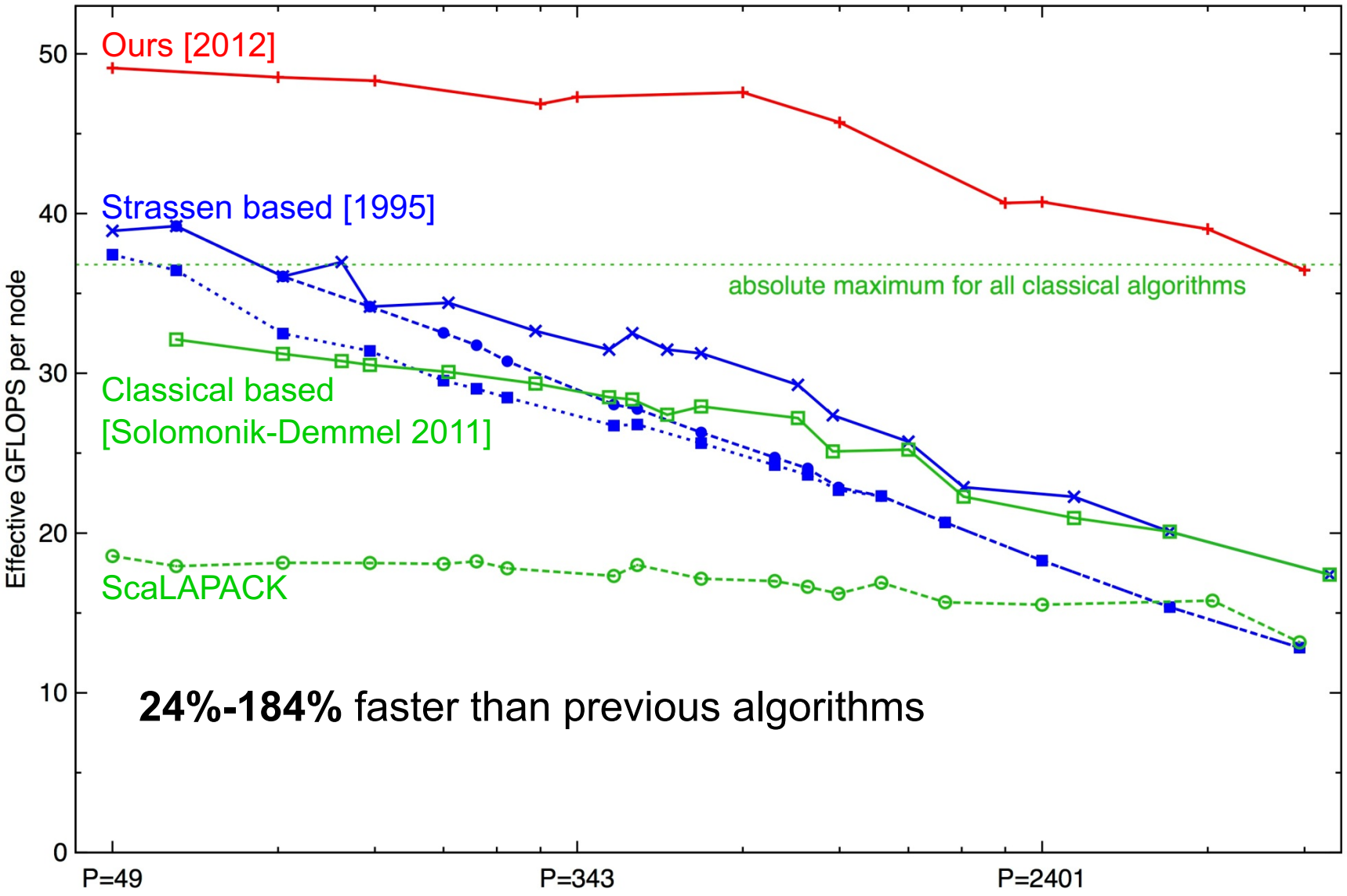
- (1) proving bounds on these communication costs
- (2) developing faster algorithms by minimizing communication

Save time, Save energy



The Fastest Matrix Multiplication in the West

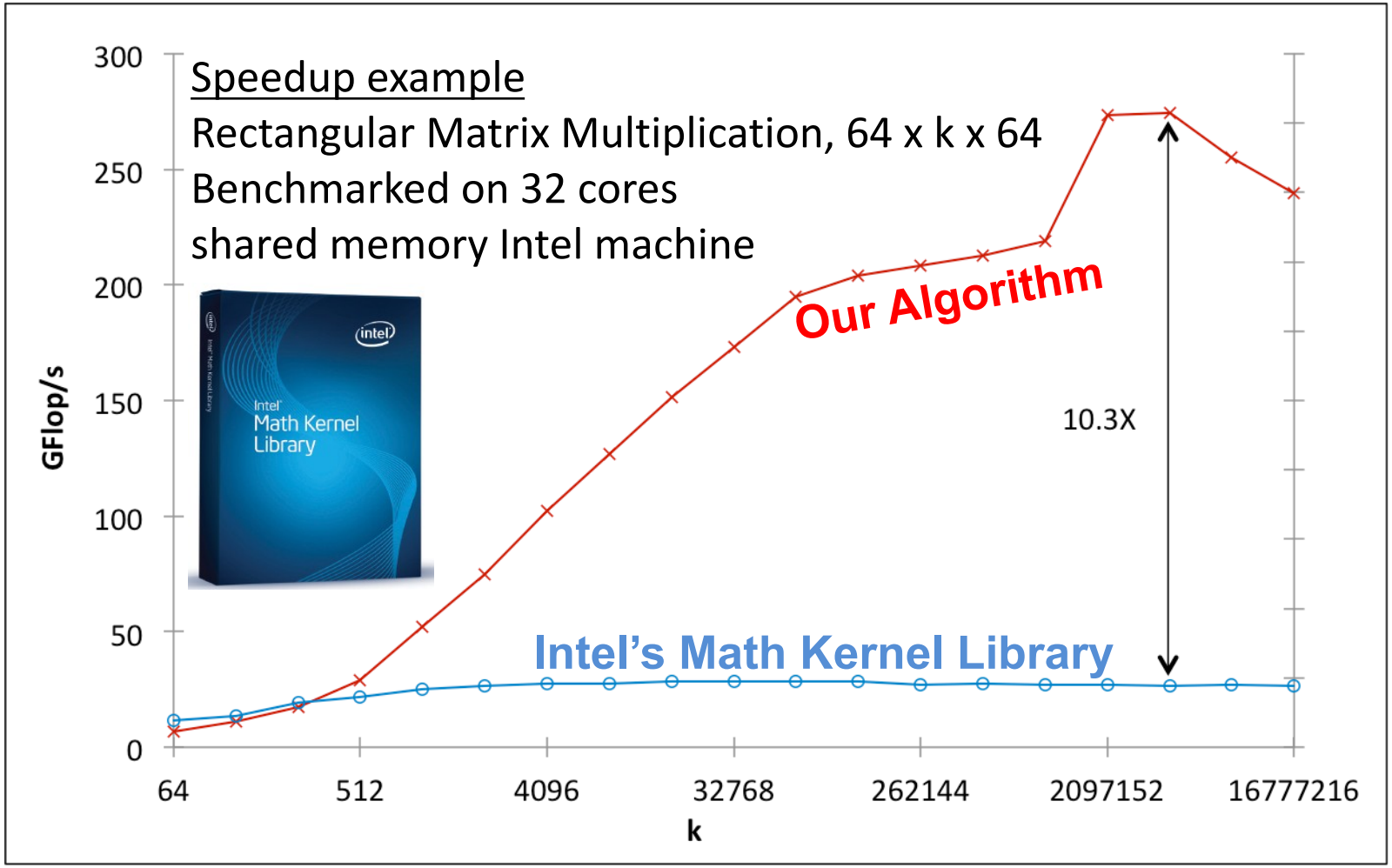
Franklin (Cray XT4), Strong Scaling, n = 94080 [BDHLS SPAA'12]



The Fastest Matrix Multiplication in the West [DEFKLSS, SuperComputing'12]

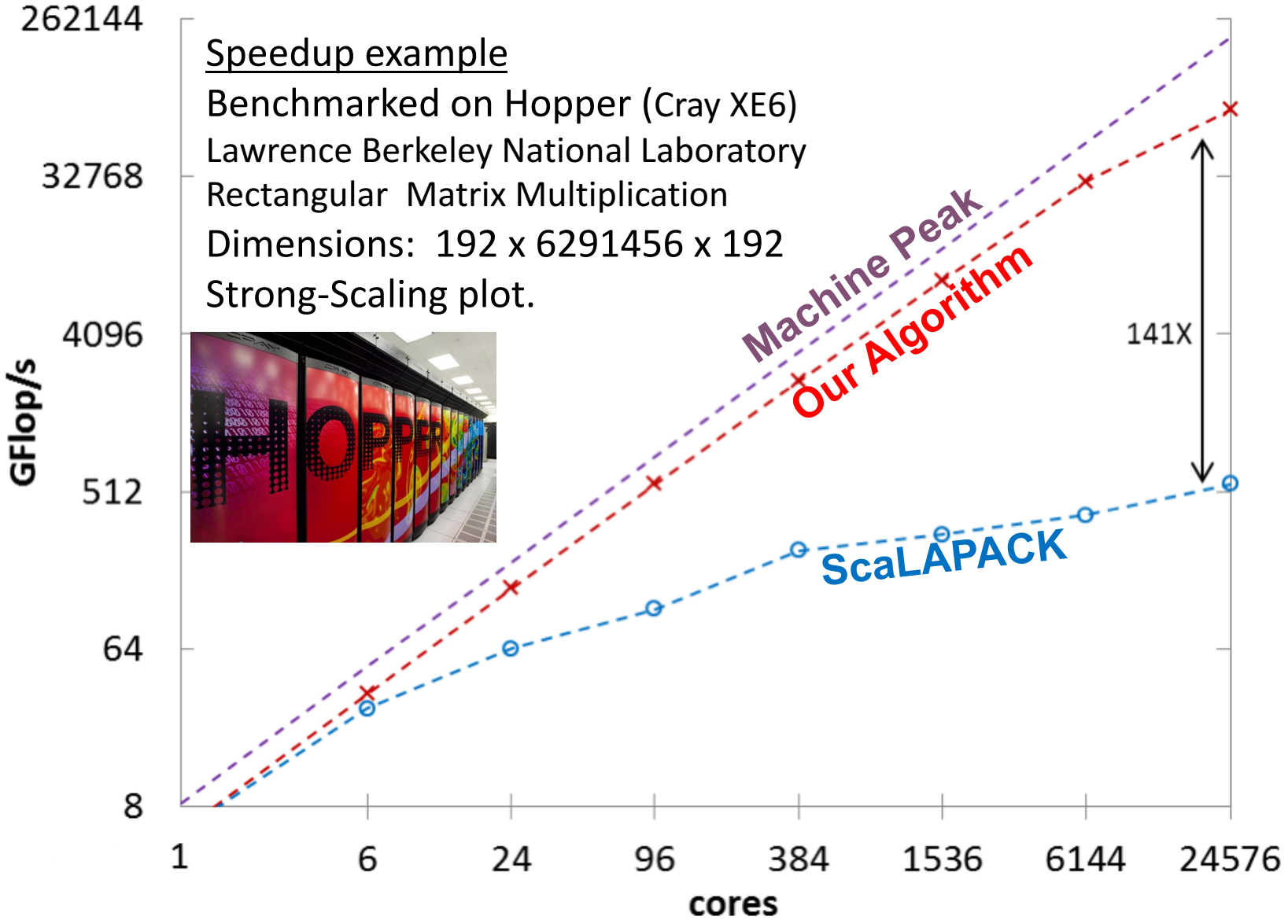
"…Highest Performance and Scalability across Past, Present & Future Processors…"

Intel's Math Kernel Library. From: http://software.intel.com/en-us/intel-mkl

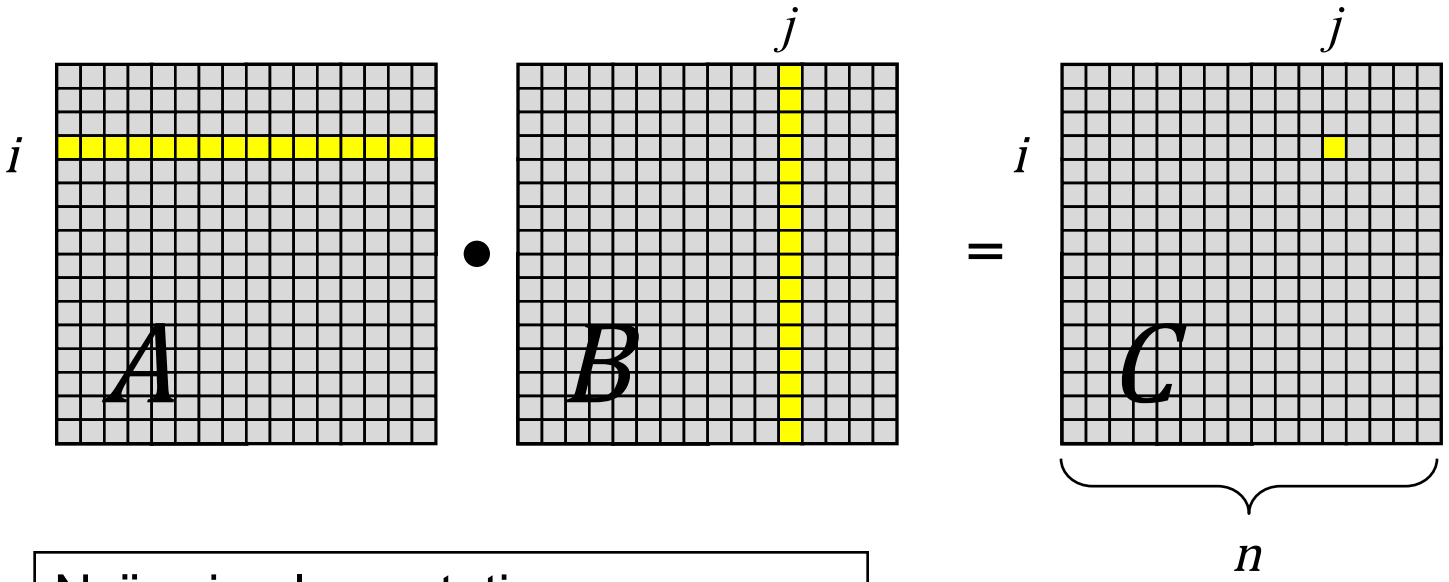


The Fastest Matrix Multiplication in the West

[DEFKLSS, SuperComputing'12]



Example: Classical Matrix Multiplication



Naïve implementation

For $i = 1$ to n

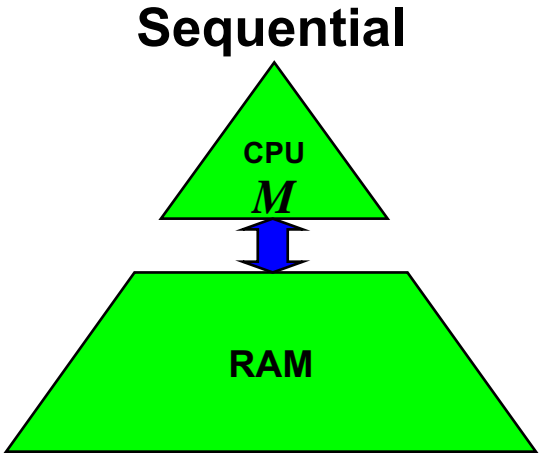
  For $j = 1$ to n

    For $k = 1$ to n

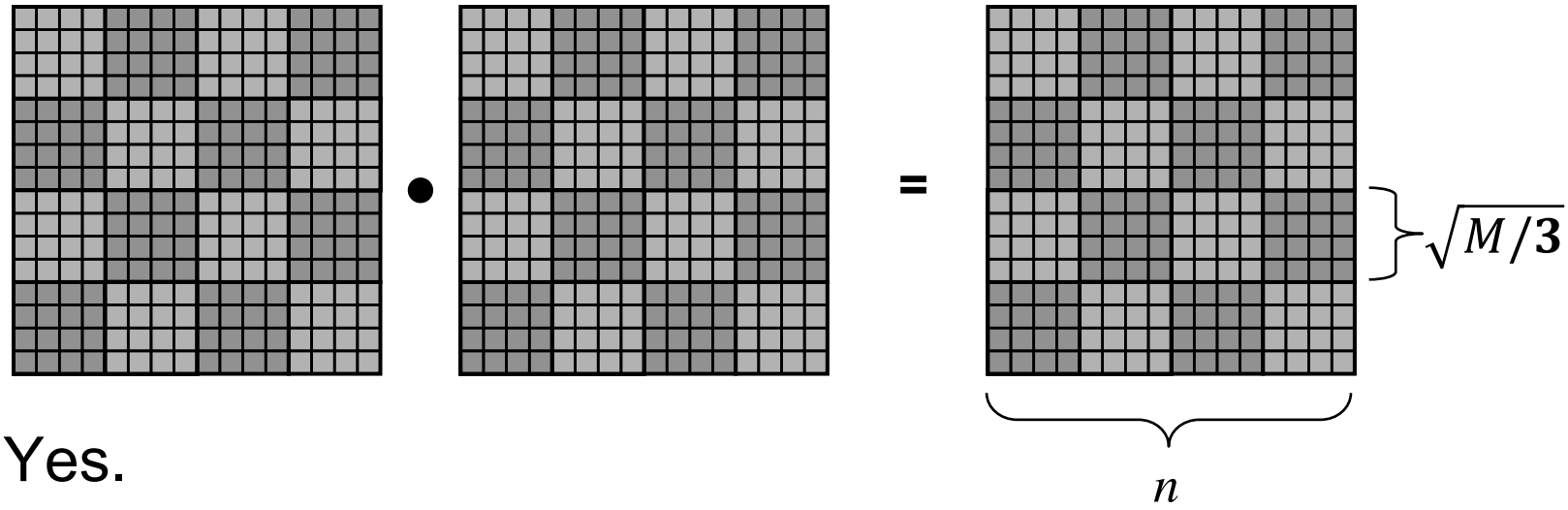
      $C(i, j) = C(i, j) + A(i, k)B(k, j)$

Bandwidth cost: $BW = \Theta\ (n^3)$

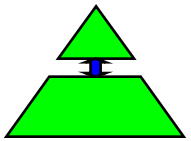
Can we do better?



Example: Classical Matrix Multiplication



Yes.

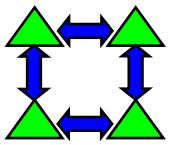


Compute block-wise [BLAS]

$$BW = O\left(\left(\frac{n}{\sqrt{M}}\right)^3 \cdot M\right) = O\left(\frac{n^3}{\sqrt{M}}\right)$$

M is fast/local memory size

$M < n^2$



[Cannon 69] $M = \Theta(n^2/P)$, [McColl Tiskin 99, Solomonik Demmel 11] any M

$$BW = O\left(\left(\frac{n}{\sqrt{M}}\right)^3 \cdot \frac{M}{P}\right)$$

Can we do better?

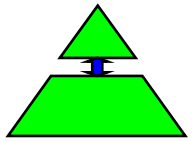
Lower bounds: "3-nested loops"

No!

[Hong & Kung 81]

- Sequential Mat-Mul

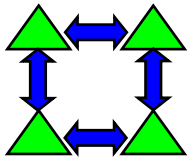
$$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^3 \cdot M\right)$$



[Irony,Toledo,Tiskin 04]

- Sequential and parallel Mat-Mul

$$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^3 \cdot \frac{M}{P}\right)$$

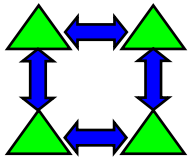


[Ballard,Demmel,Holtz, S 11a]

- All direct numerical linear algebra

  BLAS, LU, Cholesky, LDL^T,
  QR, eigenvalues, SVD,…
  Dense or sparse matrices,
  Sequential, parallel distributed, shared…

$$\Omega\left(\frac{\#FLOPs}{\left(\sqrt{M}\right)^3} \cdot \frac{M}{P}\right)$$



Impact: Many new algorithms (ours and others) that are communication optimal (match our lower bound)

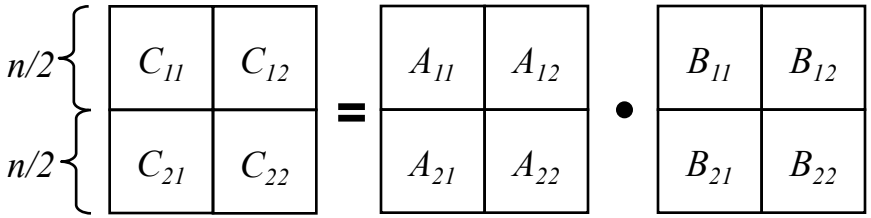
Symmetric and non-symmetric Eigenvalues, SVD,…

How about the communication costs of algorithms that have a more complex structure?

Recall: Strassen's Fast Matrix Multiplication

[Strassen 69]

- Compute 2 x 2 matrix multiplication using only 7 multiplications (instead of 8).
- Apply recursively (block-wise)



$$M_1 = (A_{11} + A_{22}) \cdot (B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22}) \cdot B_{11}$$
$$M_3 = A_{11} \cdot (B_{12} - B_{22})$$
$$M_4 = A_{22} \cdot (B_{21} - B_{11})$$
$$M_5 = (A_{11} + A_{12}) \cdot B_{22}$$
$$M_6 = (A_{21} - A_{11}) \cdot (B_{11} + B_{12})$$
$$M_7 = (A_{12} - A_{22}) \cdot (B_{21} + B_{22})$$

$$T(n) = 7 \cdot T(n/2) + \Theta(n^2)$$

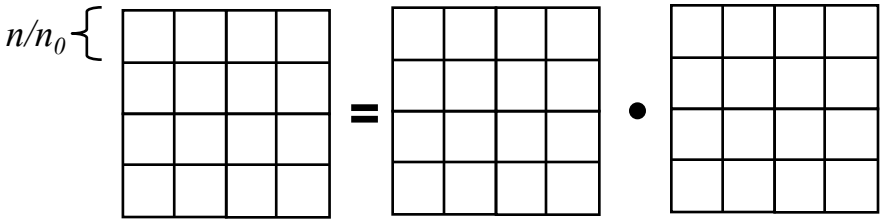
$$T(n) = \Theta(n^{\log_2 7})$$

$$C_{11} = M_1 + M_4 - M_5 + M_7$$
$$C_{12} = M_3 + M_5$$
$$C_{21} = M_2 + M_4$$
$$C_{22} = M_1 - M_2 + M_3 + M_6$$

Strassen-like algorithms

Subsequently…

- Compute n_0 x n_0 matrix multiplication using only $n_0^{\omega_0}$ multiplications (instead of n_0^3).
- Apply recursively (block-wise)



$\omega_0 \approx$

2.81 [Strassen 69],[Strassen-Winograd 71]

2.79 [Pan 78]

2.78 [Bini 79]

2.55 [Schönhage 81]

2.50 [Pan Romani,Coppersmith Winograd 84]

2.48 [Strassen 87]

2.38 [Coppersmith Winograd 90]

2.38 [Cohn Kleinberg Szegedy Umans 05] Group-theoretic approach

2.3730 [Stothers 10]

2.3728640 [Vassilevska Williams 12]

2.3728639 [Le Gall 14]

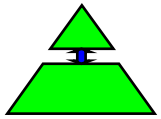
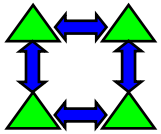
$$T(n) = n_0^{\omega_0} \cdot T(n/n_0) + \Theta(n^2)$$

$$T(n) = \Theta(n^{\omega_0})$$

Communication costs lower bounds for matrix multiplication

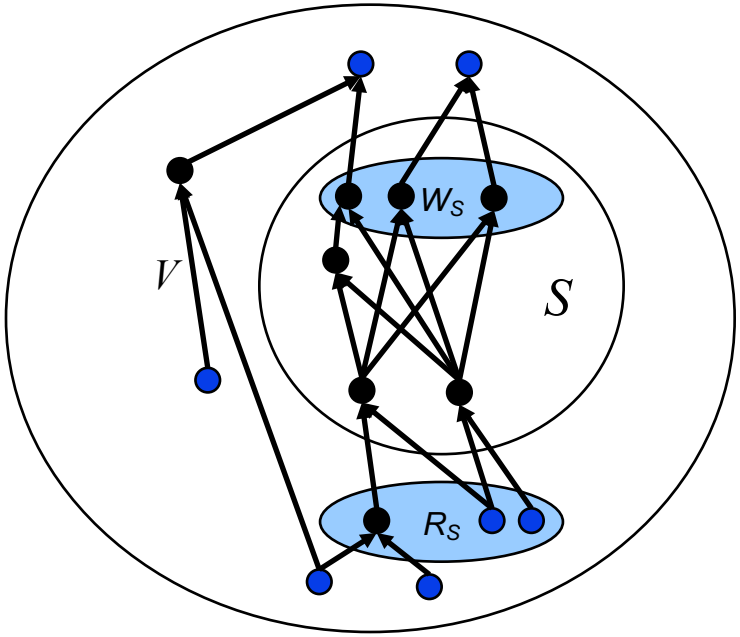
[Ballard, Demmel, Holtz, S. 2011b]:

- Sequential and parallel
- Novel graph expansion proof

	For Strassen's:	Strassen-like:	Classic (cubic):
 Sequential	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 7} M\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\omega_0} M\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 8} M\right)$
 Distributed	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 7} \frac{M}{P}\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\omega_0} \frac{M}{P}\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 8} \frac{M}{P}\right)$

The Computation Directed Acyclic Graph

- Input / Output
- Intermediate value
- Dependency



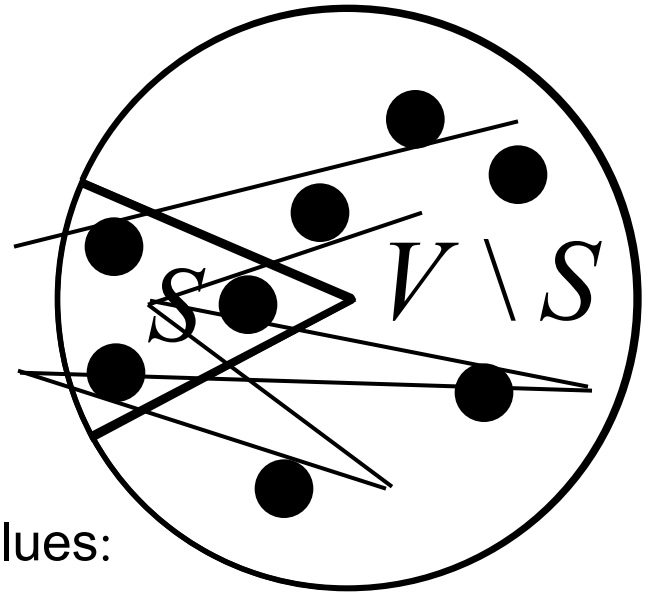
How can we estimate R_s and W_s?

By bounding the expansion of the graph!

Expansion

[Ballard, Demmel, Holtz, S. 2011b],
in the spirit of [Hong & Kung 81]

Let $G = (V,E)$ be a graph



$$h \equiv \min_{S,|S|\le\frac{|V|}{2}} \frac{\left|E\left(S,\overline{S}\right)\right|}{\left|E\left(S\right)\right|}$$

A is the normalized adjacency matrix
of a regular undirected graph, with eigenvalues:

$$1 = \lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$$

$$\gamma \equiv 1 - max\{\lambda_2, |\lambda_n|\}$$

Thm: [Alon-Milman84, Dodziuk84, Alon86]

$$\frac{1}{2}\gamma \le h \le \sqrt{2\gamma}$$

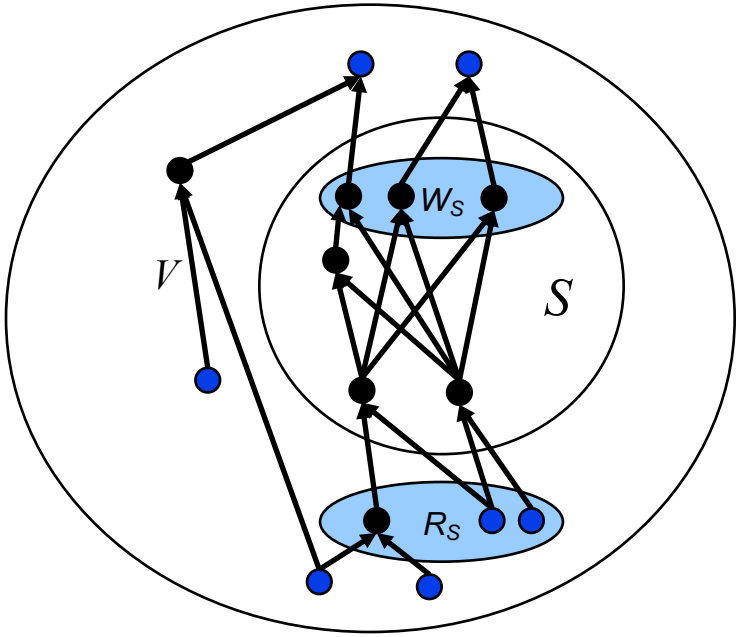
Small sets expansion:

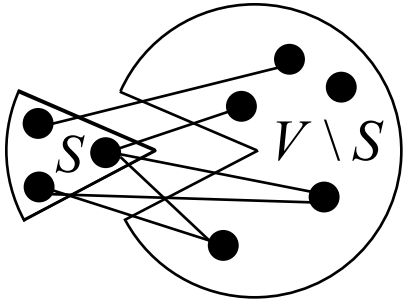
$$h_s \equiv \min_{S,|S|\le s} \frac{\left|E\left(S,\overline{S}\right)\right|}{\left|E\left(S\right)\right|}$$

Expansion

The Computation Directed Acyclic Graph

Input / Output
Intermediate value
Dependency





Communication-Cost is (Small-Sets) Graph-Expansion

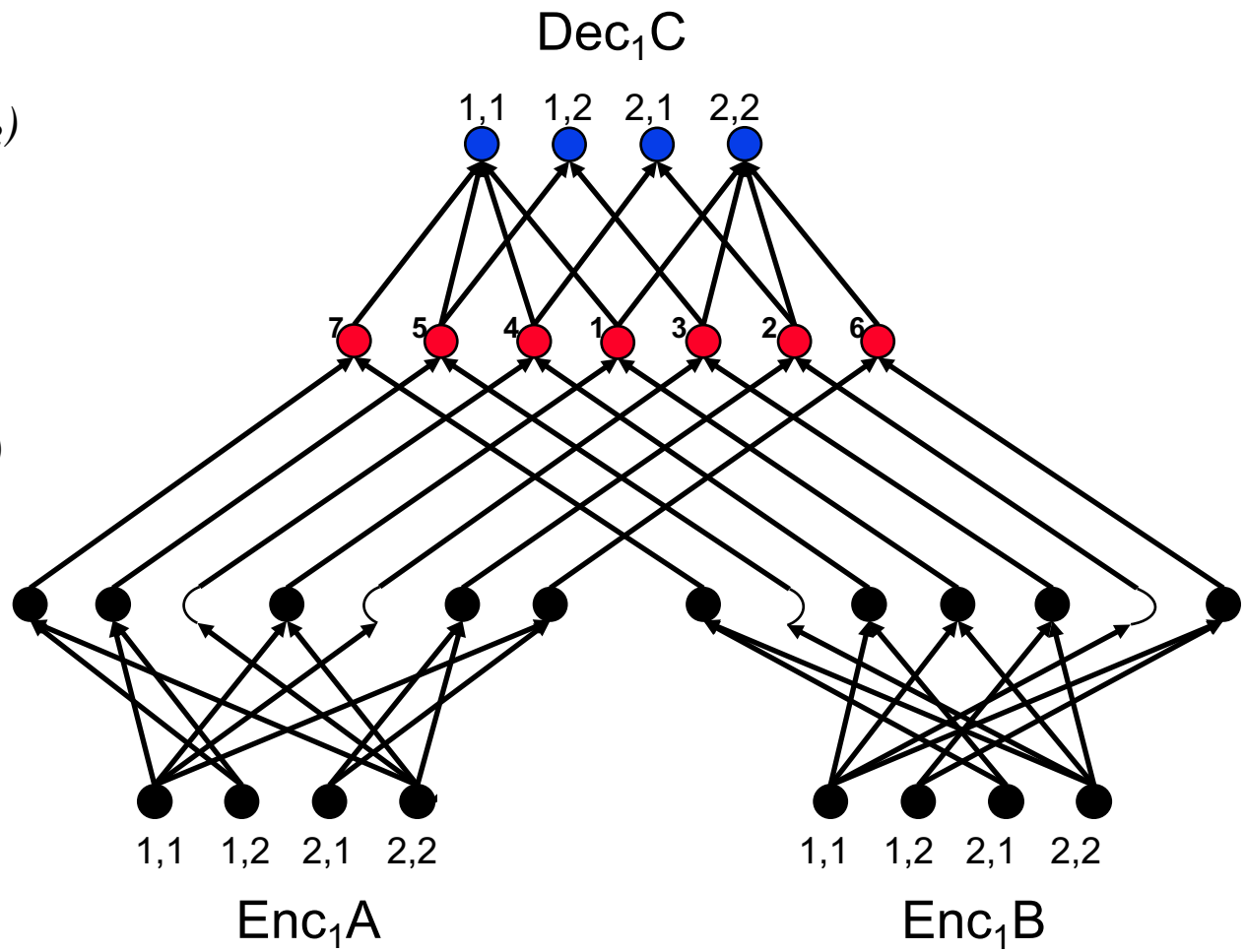
What is the Computation Graph of Strassen?

Can we Compute its Expansion?

The DAG of Strassen, n = 2

$M_1 = (A_{11} + A_{22}) \cdot (B_{11} + B_{22})$
$M_2 = (A_{21} + A_{22}) \cdot B_{11}$
$M_3 = A_{11} \cdot (B_{12} - B_{22})$
$M_4 = A_{22} \cdot (B_{21} - B_{11})$
$M_5 = (A_{11} + A_{12}) \cdot B_{22}$
$M_6 = (A_{21} - A_{11}) \cdot (B_{11} + B_{12})$
$M_7 = (A_{12} - A_{22}) \cdot (B_{21} + B_{22})$

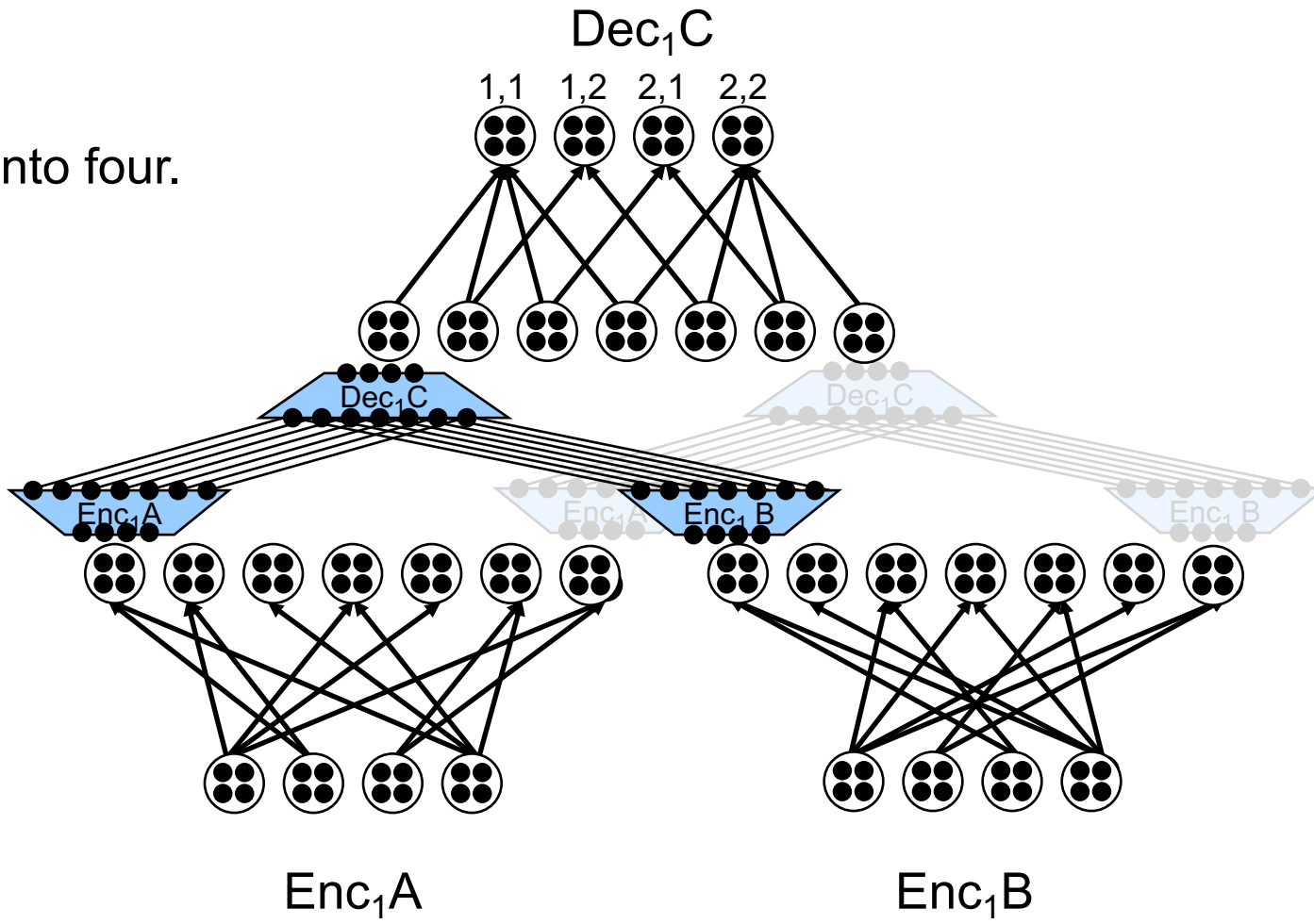
$C_{11} = M_1 + M_4 - M_5 + M_7$
$C_{12} = M_3 + M_5$
$C_{21} = M_2 + M_4$
$C_{22} = M_1 - M_2 + M_3 + M_6$



The DAG of Strassen, n=4

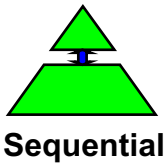
One recursive level:

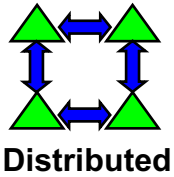
- Each vertex splits into four.
- Multiply blocks



Communication costs
lower bounds for matrix multiplication

Algorithms attaining these bounds?





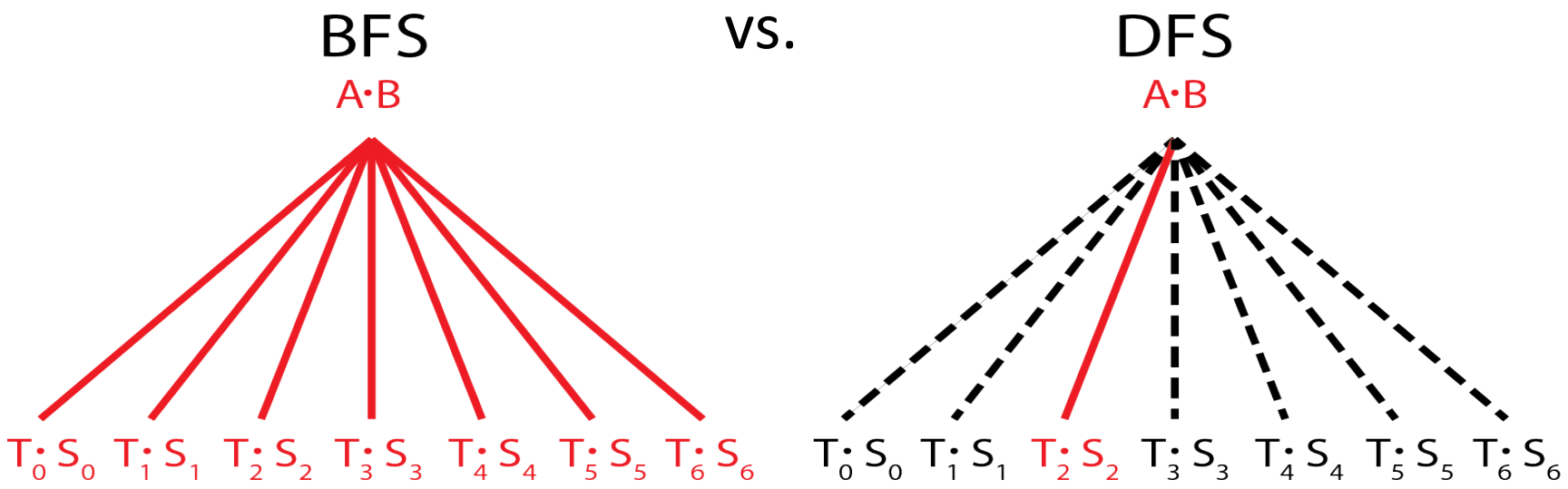
	For Strassen's:	Strassen-like:	Classic (cubic):
Sequential	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 7} M\right)$ ✓	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\omega_0} M\right)$ ✓	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 8} M\right)$ ✓
Distributed	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 7} \frac{M}{P}\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\omega_0} \frac{M}{P}\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 8} \frac{M}{P}\right)$ ✓

Next

[Cannon 69, McColl Tiskin 99, Solomonik Demmel '11]

Communication Avoiding Parallel Strassen

At each level of recursion tree, choose either breadth-first or depth-first traversal.

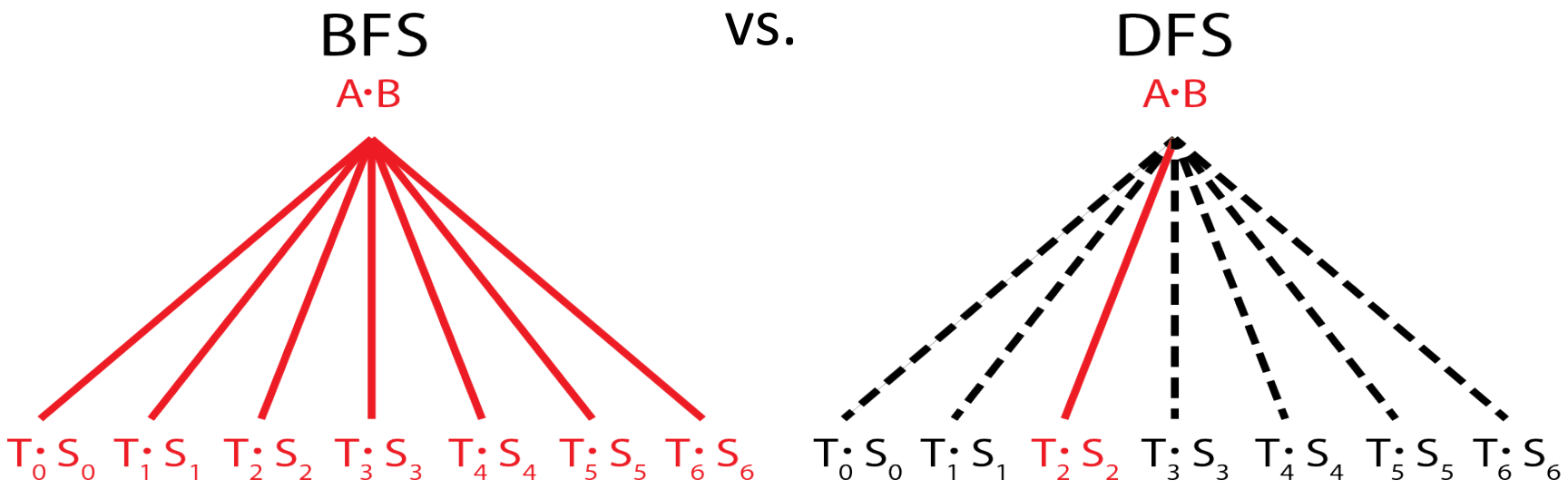


- Runs all 7 sub-problems in **parallel**
  - Each on P/7 processors
- Needs 7/4 as much extra memory
- Requires communication, but
- All-BFS (if possible) minimizes communication

- Runs all 7 sub-problems **sequentially**
  - Each on all P processors
- Needs 1/4 as much extra memory
- No immediate communication
- Increases bandwidth by factor of 7/4
- Increases latency by factor of 7

The Algorithm: Communication Avoiding Parallel Strassen *(CAPS)*

At each level of recursion tree, choose either breadth-first or depth-first traversal.



- Runs all 7 sub-problems in **parallel**
  - Each on P/7 processors
- Runs all 7 sub-problems **sequentially**
  - Each on all P processors

CAPS

If EnoughMemory **and** $P \geq 7$

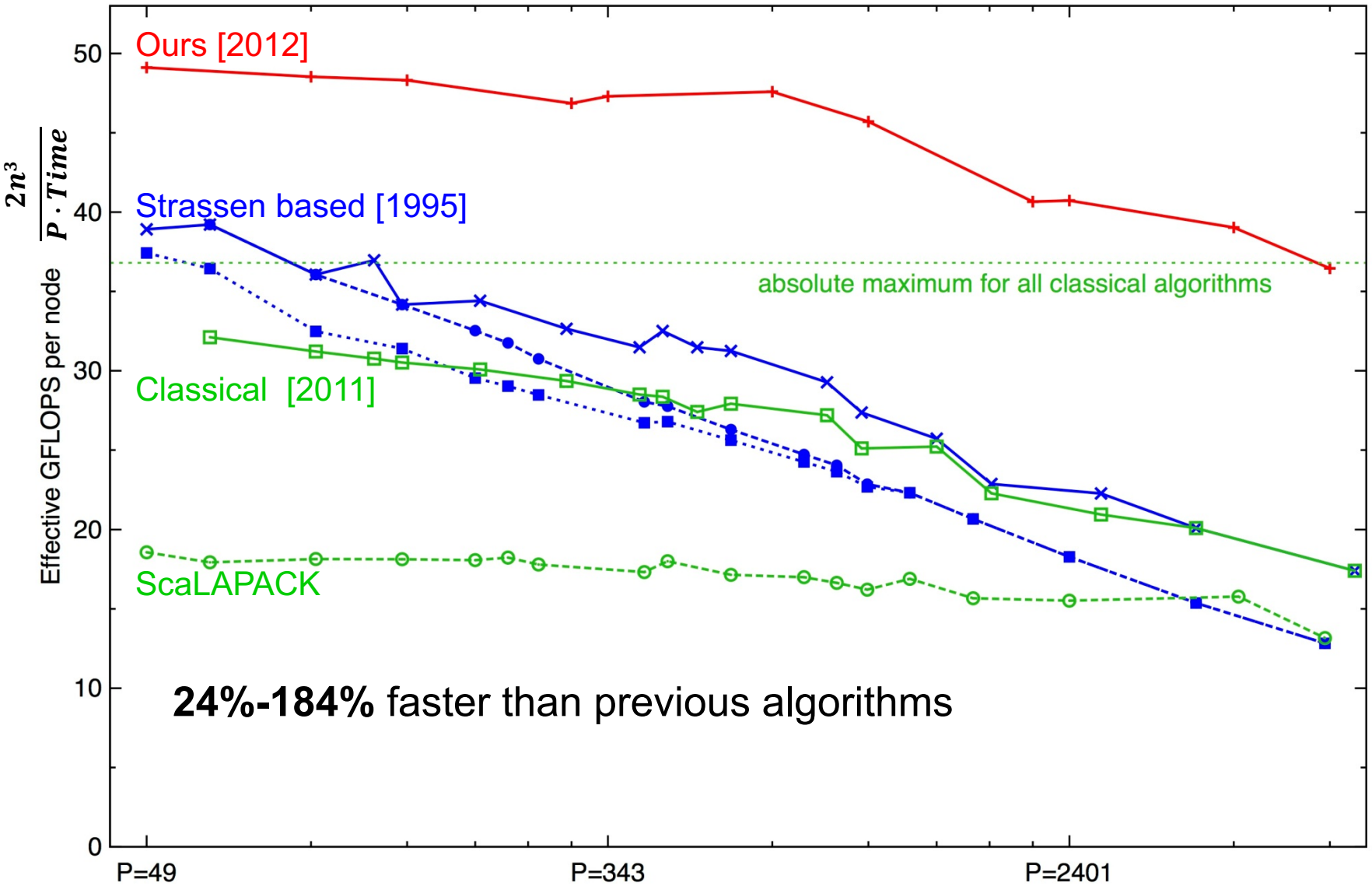
then BFS step

else DFS step

end if

The Fastest Matrix Multiplication in the West

Franklin (Cray XT4), Strong Scaling, n = 94080 [BDHLS SPAA'12]



Can we do better?

Model & Motivation

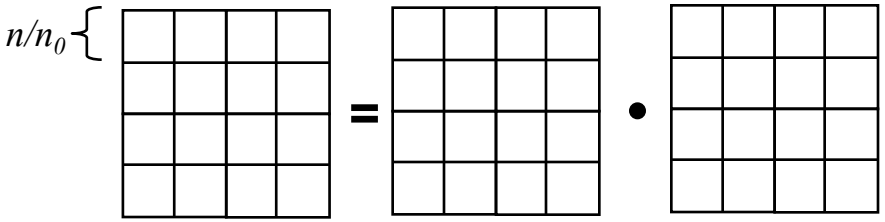
Two kinds of costs:

- Arithmetic (FLOPs)
- Communication: moving data

$$Running\ time = \gamma \cdot \#FLOPs + \beta \cdot \#Words \quad (+ \alpha \cdot \#Messages)$$

Strassen-like algorithms

Can we improve the hidden constants?



$\omega_0 \approx$

2.81 [Strassen 69],[Strassen-Winograd 71]

2.79 [Pan 78]

2.78 [Bini 79]

2.55 [Schönhage 81]

2.50 [Pan Romani,Coppersmith Winograd 84]

2.48 [Strassen 87]

2.38 [Coppersmith Winograd 90]

2.38 [Cohn Kleinberg Szegedy Umans 05] Group-theoretic approach

2.3730 [Stothers 10]

2.3728640 [Vassilevska Williams 12]

2.3728639 [Le Gall 14]

$$T(n) = n_0^{\omega_0} \cdot T(n/n_0) + \Theta(n^2)$$

$$T(n) = \Theta(n^{\omega_0})$$

<2,2,2;7>-algorithms

Algorithm	Additions	Arithmetic Count
Strassen [33]	18	$7n^{\log_2 7} - 6n^2$
Strassen-Winograd [36]	15	$6n^{\log_2 7} - 5n^2$
Ours	12	$5n^{\log_2 7} - 4n^2 + 3n^2 \log_2 n$

Can we do better?

No!

Thm [Probert 76]

- Any Strassen-like algorithm with 2x2 base case and with 7 multiplications requires at least 15 additions.

Yes!

Thm [Karstadt & S. 2017]

- There is a Strassen-like algorithm with 2x2 base case and with 7 multiplications that requires12 additions.

Faster matrix multiplication by base change

Adapting from [Bodrato 2010]

Elaye Karstadt

Algorithm 1 Alternative Basis Strassen-like Multiplication

Input: $A \in R^{n \times m}$, $B^{m \times k}$

Output: $n \times k$ matrix $C = A \cdot B$

1: **function** $ABS(A, B)$

2: $\tilde{A} = \psi(A)$ ▷ $R^{n \times m}$ basis transformation

3: $\tilde{B} = \phi(B)$ ▷ $R^{m \times k}$ basis transformation

4: $\tilde{C} = RBA(\tilde{A}, \tilde{B})$ ▷ $\langle n, m, k; t \rangle_{\phi,\psi,\upsilon}$-algorithm

5: $C = \upsilon^{-1}(\tilde{C})$ ▷ $R^{n \times k}$ basis transformation

6: **return** C

Basis change in $O(n^2 lg\ n)$

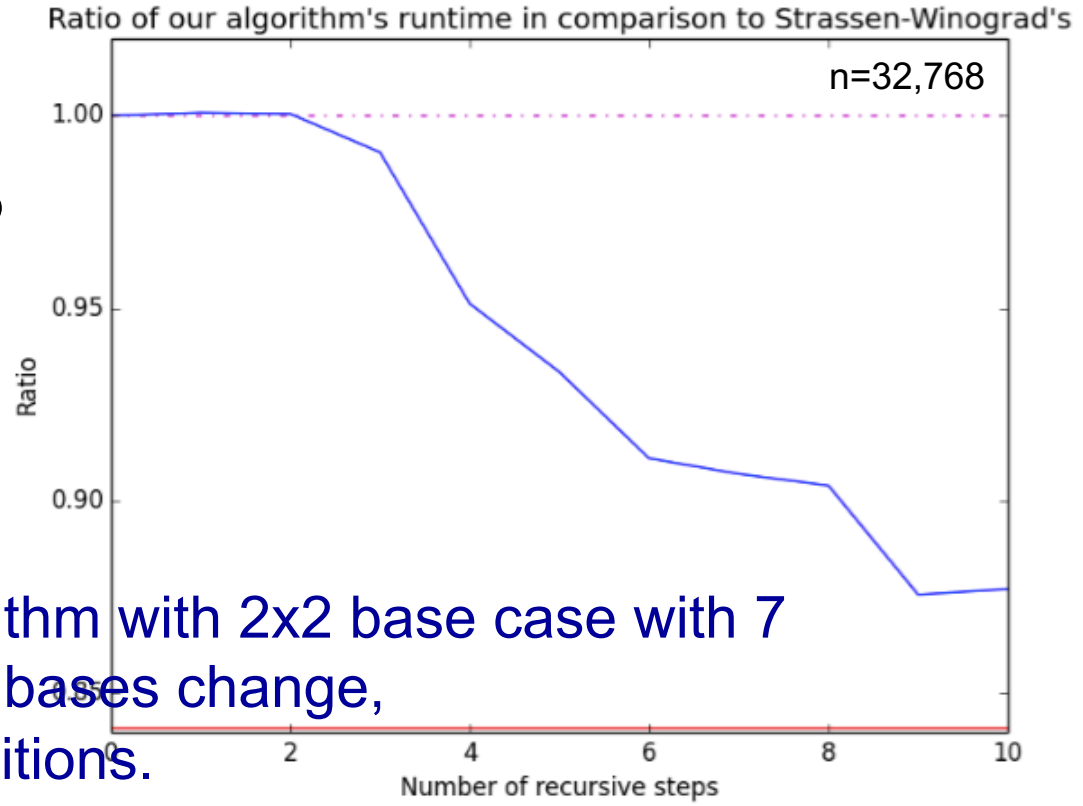
The new bases allow sparser bilinear operation.

$$U_{opt} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \quad V_{opt} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & 0 \end{pmatrix} \quad W_{opt} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$\psi_{opt} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad \psi_{opt}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 1 & 1 \end{pmatrix}$$

Testing our new <2,2,2;7>-algorithm

Algorithm	Additions	Arithmetic Count	I/O-Complexity
Strassen [33]	18	$7n^{\log_2 7} - 6n^2$	$6 \cdot \left(\frac{\sqrt{3} \cdot n}{\sqrt{M}}\right)^{\omega_0} \cdot M - 18n^2 + 3M$
Strassen-Winograd [36]	15	$6n^{\log_2 7} - 5n^2$	$5 \cdot \left(\frac{\sqrt{3} \cdot n}{\sqrt{M}}\right)^{\omega_0} \cdot M - 15n^2 + 3M$
Ours	12	$5n^{\log_2 7} - 4n^2 + 3n^2 \log_2 n$	$4 \cdot \left(\frac{\sqrt{3} \cdot n}{\sqrt{M}}\right)^{\omega_0} \cdot M - 12n^2 + 3n^2 \cdot \log_2 \left(\sqrt{2} \cdot \frac{n}{\sqrt{M}}\right) + 5M$



Can we do better?

No!

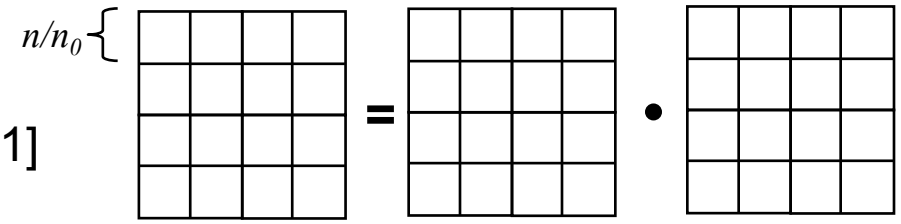
Thm [Karstadt & S. 2017]

Adapting [Probert 76]

- Any Strassen-like algorithm with 2x2 base case with 7 multiplications with any bases change, requires at least 12 additions.

Strassen-like algorithms

Can we improve the hidden constants?



$$T(n) = n_0^{\omega_0} \cdot T(n/n_0) + \Theta(n^2)$$

$$T(n) = \Theta(n^{\omega_0})$$

$\omega_0 \approx$

2.81 [Strassen 69],[Strassen-Winograd 71]
2.79 [Pan 78]
2.78 [Bini 79]
2.55 [Schönhage 81]
2.50 [Pan Romani,Coppersmith Winograd 84]
2.48 [Strassen 87]
2.38 [Coppersmith Winograd 90]
2.38 [Cohn Kleinberg Szegedy Umans 05] Group-theoretic approach
2.3730 [Stothers 10]
2.3728640 [Vassilevska Williams 12]
2.3728639 [Le Gall 14]

It may be hard. Sparsification is NP-hard [McCormick 83].

Even NP hard to approximate [Gottlieb and Neylon 2010].

Lucy to the rescue

Finding faster matrix multiplication algorithms, using our Lucy cluster:

Algorithm	Linear Operations	Leading Coefficient	Alternative-Basis Linear Operations	Improved leading coefficient	Reduction Factor
$\langle 2,2,2;7\rangle$[36]	15	6	12	5	16.6%
$\langle 3,2,3;15\rangle$[2]	64	15.06	52	7.94	47.3%
$\langle 2,3,4;20\rangle$[2]	78	9.96	58	7.46	25.6%
$\langle 3,3,3;23\rangle$[2]	87	8.91	75	6.57	26.3%
$\langle 6,3,3;40\rangle$[31]	1246	55.63	202	9.39	83.2%

Algorithm 1 Alternative Basis Strassen-like Multiplication

Input: $A \in R^{n\times m}$, $B^{m\times k}$

Output: $n \times k$ matrix $C = A \cdot B$

1: **function** $ABS(A, B)$
2: $\tilde{A} = \psi(A)$ ▷ $R^{n\times m}$ basis transformation
3: $\tilde{B} = \phi(B)$ ▷ $R^{m\times k}$ basis transformation
4: $\tilde{C} = RBA(\tilde{A}, \tilde{B})$ ▷ $\langle n, m, k; t\rangle_{\phi,\psi,\upsilon}$-algorithm
5: $C = \upsilon^{-1}(\tilde{C})$ ▷ $R^{n\times k}$ basis transformation
6: **return** C

Conclusions

Algorithmic tools and matching lower bounds for:

- Arithmetic (FLOPs)
- Communication: moving data

$$Running\ time = \gamma \cdot \#FLOPs + \beta \cdot \#Words \quad (+ \alpha \cdot \#Messages)$$

Minimizing Arithmetic & Communication Costs for Faster Matrix Computations

Oded Schwartz
The Hebrew University

ACA'17, July 17-21

Based on joint papers with

Grey Ballard, James Demmel, Andrew Gearhart, Olga Holtz, Elaye Karstadt, Ben Lipshitz, Yishai Oltchik, and Sivan Toledo.

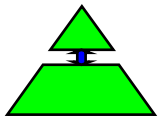
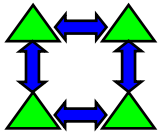
Thank you

Research supported by: ISF, BSF, Intel, Ministry of Science, Minerva, Einstein Foundation

Supercomputing resources by: PRACE, LinkSCEEM, ALCF, ORNL

Extra Slides

Communication costs for matrix multiplication

	For Strassen's:	Strassen-like:	Classic (cubic):
 Sequential	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 7} M\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\omega_0} M\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 8} M\right)$
 Distributed	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 7} \frac{M}{P}\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\omega_0} \frac{M}{P}\right)$	$\Omega\left(\left(\frac{n}{\sqrt{M}}\right)^{\log_2 8} \frac{M}{P}\right)$
Memory independent bounds [Ballard, Demmel, Holtz, Lipshitz, S. 2012b]:	$\Omega\left(\frac{n^2}{P^{2/\log_2 7}}\right)$	$\Omega\left(\frac{n^2}{P^{2/\omega_0}}\right)$	$\Omega\left(\frac{n^2}{P^{2/3}}\right)$ [McColl Tiskin 99, Solomonik & Demmel 11]

Minimizing Arithmetic & Communication Costs for Faster Matrix Computations

Oded Schwartz
The Hebrew University

ACA'17, July 17-21

Based on joint papers with

Grey Ballard, James Demmel, Andrew Gearhart, Olga Holtz, Elaye Karstadt, Ben Lipshitz, Yishai Oltchik, and Sivan Toledo.

Thank you

Research supported by: ISF, BSF, Intel, Ministry of Science, Minerva, Einstein Foundation

Supercomputing resources by: PRACE, LinkSCEEM, ALCF, ORNL