RESEARCH PAPER PRESENTATION

A three-dimensional approach to parallel matrix multiplication

by R.C. Agarwal, S.M. Balle et al.

Presenter: Damin Xia

VANDERBILT UNIVERSITY

INTRODUCTION

- This paper is trying to develop a new 3D matrix multiplication algorithm for massively parallel processing.

- The performance of the algorithm is assessed by running the _GEMM routine: $\mathbf{C} = \beta\mathbf{C} + \alpha\mathrm{op}(\mathbf{A})\mathrm{op}(\mathbf{B})$ where $\mathrm{op}(\mathbf{X})$ stands for $\mathbf{X}$, $\mathbf{X}^{\mathrm{T}}$ or $\mathbf{X}^{\mathrm{C}}$ (conjugate transpose).

- The "3D" is constructed from: $C_{ij} = A_{il}B_{lj}$, and processor (i,j,l) computes $A_{il}B_{lj.}$

VANDERBILT UNIVERSITY

LITERATURE REVIEW

- Most parallel matrix multiplication algorithms used as building blocks are 2D algorithms.
- Comparing with 2D algorithms, the primary issue is that the 3D algorithm moves a factor of $P^{1/6}$ less data than the known 2D algorithms.
- Some researchers show that 3D-type algorithm is optimal for an LPRAM (see e.g. [1]). Some researchers discuss 3D and other types of algorithms for Boolean cubes and hypercubes (see e.g. [2]).The discussion of scalability of 3D algorithm can be found in [3].

[1]J. W. Demmel, M.T. Heath, and H. A. van der Vorst, "Parallel Numerical Linear Algebra," ***Acta Numerica 1993,*** Cambridge University Press, 1993, pp. 111-197.
[2] ***S*.** L. Johnsson and C.-T. Ho, "Algorithms for Multiplying Matrices of Arbitrary Shapes Using Shared Memory Primitives on Boolean Cubes," ***Technical Report TR-569,*** Yale University, New Haven, CT, 1987.
[3] A. Gupta and V. Kumar, "Scalability of Parallel Algorithms for Matrix Multiplication," Technical Report, Department **of** Computer Science, University of Minnesota, 1991; revised April 1994.

VANDERBILT UNIVERSITY

INNOVATION

- For distributed memory message-passing computers, the algorithm presented in this work has the least amount of communication of all the 3D algorithms cited.

- This paper also provides a solution which minimizes communication for matrices of arbitrary shapes.

- The 3D algorithm in this paper can be combined in a straightforward manner with the $O(n^{2.81})$Strassen algorithm for matrix multiplication, thereby allowing it to take full advantage of the latter's high efficiency.

VANDERBILT UNIVERSITY

IMPLEMENTATIONS

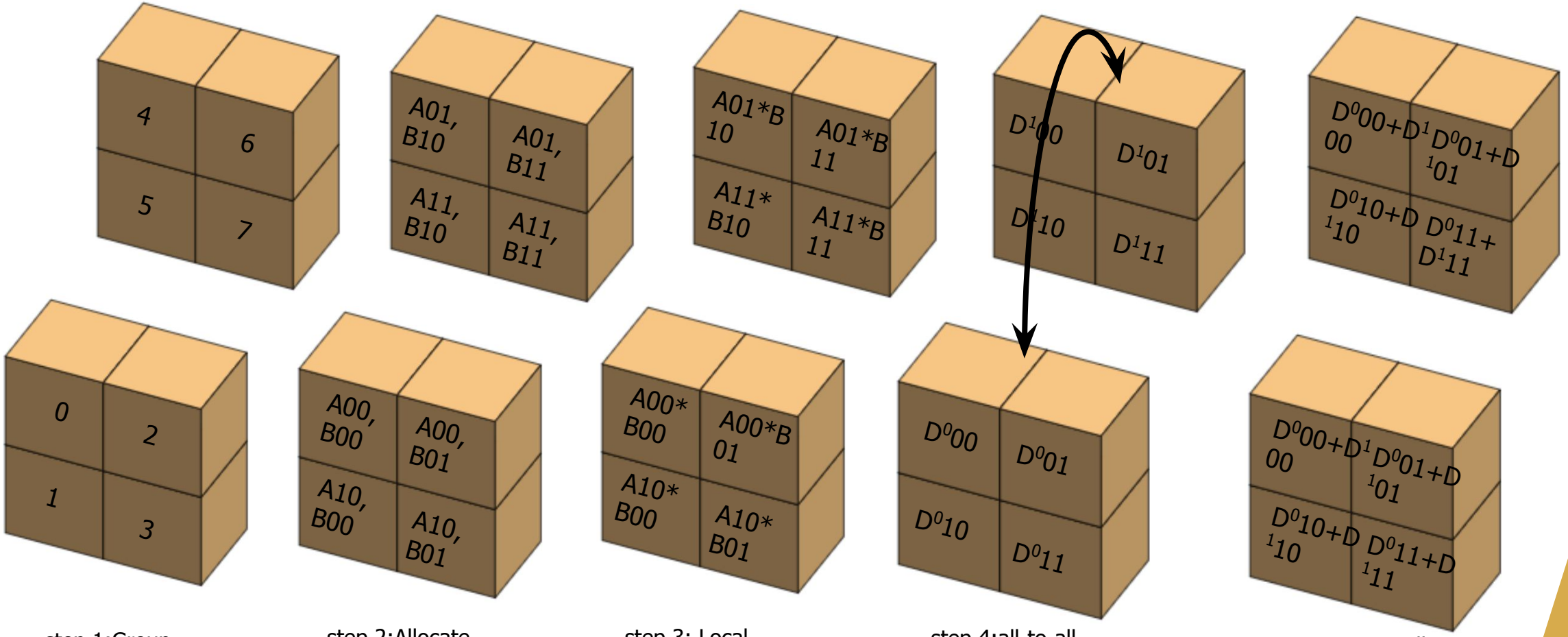
1. Grouping processes in a 3D fashion
2. Allocating spaces for submatrices of A, B and resulting matrix C: matrix A is laid out in d1-d3 plane, with d2 being the orthogonal dimension; matrix B is laid out in d2-d3 plane, with d3 being the orthogonal dimension; matrix A is laid out in d1-d2 plane, with d3 being the orthogonal dimension;
3. Perform a single local matrix product $D_{ij}^{l} = A_{il}B_{lj}$ on all processes.
4. All-to-all broadcast of matrix D_{ij}^{l} along d3 direction.
5. On every process, compute:

$$C_{ij}(l) = \beta C_{ij}(l) + \alpha \sum_{r} D_{ij}^{r}(l)$$

See next slide for illustration: a two-dimension matrix multiplication is used as example

VANDERBILT UNIVERSITY

IMPLEMENTATION



step 1:Group processes

step 2:Allocate spaces for submatrices

step 3: Local matrix multiplication

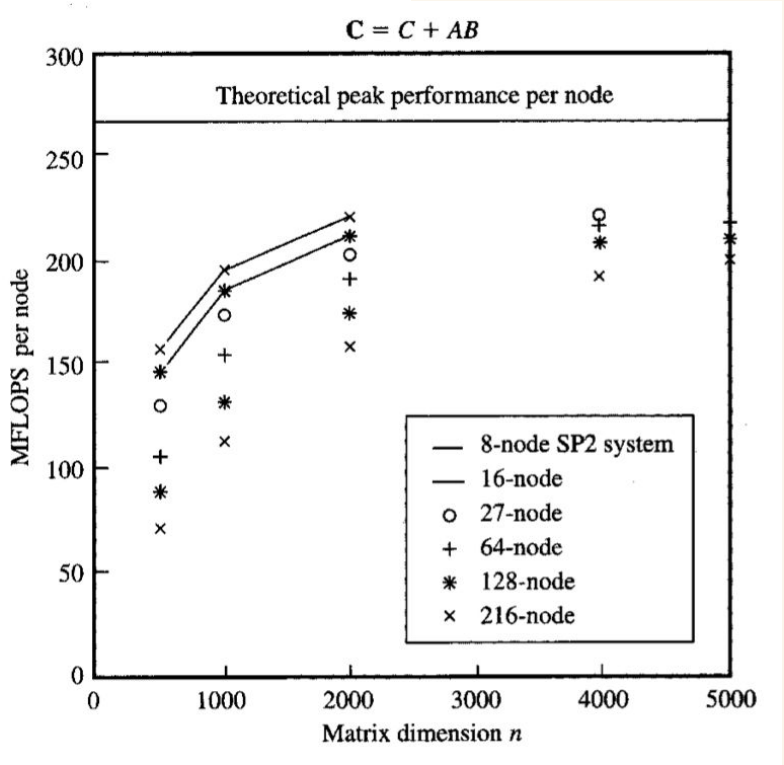
step 4:all-to-all broadcast along d3

step 5:Sum up all the matrices

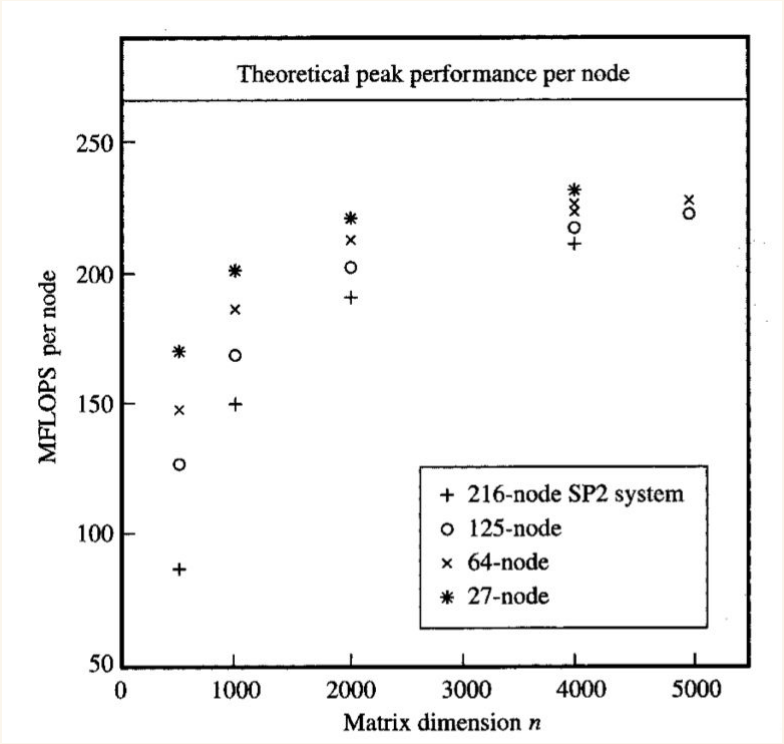
VANDERBILT UNIVERSITY

PERFORMANCE RESULTS

- The algorithm can yield very high performance even for relatively small matrices (good node performance).



Performance results for the 3D parallel double-precision PDGEMM when using DGEMM for the local call. The global input matrices are square.



Performance results for the 3D parallel double-precision complex PZGEMM when using ZGEMM for the local call. The global input matrices are square.

VANDERBILT UNIVERSITY

PERFORMANCE RESULTS

- For different matrix operations(see $\mathrm{op}(\mathbf{X})$ on the first slide), the algorithm yields the same performance;
- Strassen's algorithm can be easily applied to the algorithm and improve the performance;
- The comparison between the presented 3D algorithm and 2D ScaLAPACK PDGEMM algorithm shows that the 3D algorithm performs better in both total computation time and utilization of the CPUs.

CONCLUSIONS AND COMMENTS

- Conclusion:
  - The presented 3D approach is perfectly load-balanced for both communication and computation.
  - The choice of data distribution reduces the amount of communication from that required by the other 3D algorithms by a factor of 5/3.
  - The algorithm not only results in less communication but also produces better node performance.

- Comment:
  - 3D algorithm is a more efficient way comparing with 2D ones, and with the presented algorithm, we can even further improve its performance by including Strassen's algorithm. However, for the same matrix size, 3D algorithms require more processes, which may not be practical sometimes.

VANDERBILT UNIVERSITY