Christian Dominic Angus

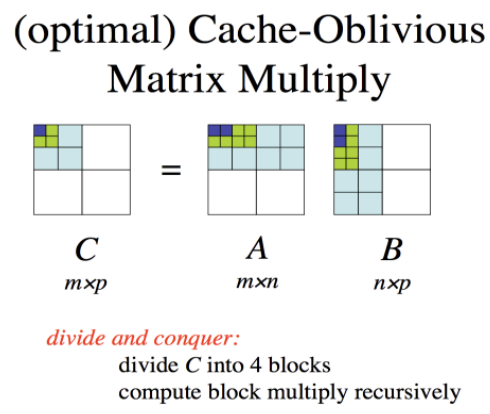
6197873

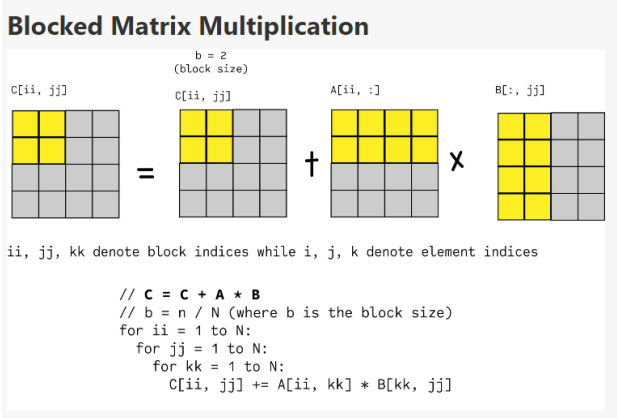
1. What three components can be used for determining the runtime of a parallel matrix multiply algorithm running on memory-distributed machines? Why is communication avoidance so important when dense linear algebra computations are executed?

Local computational tasks, the layout of the initial data and the communication schedule are the three components used to determine the runtime of a parallel matrix multiply algorithm running on a shared-memory machine. Communication avoidance improves scalability by reducing the transferred data volume at the cost of memory used, which is called “2.5D” algorithms, which is important while executing processes for dense linear algebra. The compromise of using 2.5D algorithms is to use less memory but you also lessen the use of bandwidth.

2. What is the difference between blocked vs cache-oblivious matrix multiplication algorithms? Illustrate with an example.

Cache-oblivious algorithms are not parameterized by fast memory cache or memory groups of contiguous data. It is made to take advantage of the cache the processor has without the parameters. The blocked matrix multiplication algorithm on the other hand operates on subarrays of data, instead of referring to an individual element. It is advantageous to take advantage of this algorithm because in local memory, small blocks of data can move fast within it.





3. What is graph partitioning, and why is it important in the context of parallel algorithms?

Graph partitioning is an algorithm technique used to decrease the size of the graph into smaller graphs and distribute the tasks between processors. In the context of parallel algorithms, it takes advantage of a significant amount of memory in parallel systems which is important. But one of the tradeoffs in graph partitioning is the limited number of ways you can represent data, making it one of the niche choices for partitioning in modern computing.

4. What techniques are used for partitioning irregular graph structures (e.g., Finite Element Mesh)? Can you outline how you would distribute such a graph on 4 processors?

Parallel Graphing Partitioning, Multilevel Partitioning Methods, Spectral Partitioning Methods, Kernighan-Lin Heursistic and Parallel Sparse Matrix Algorithm are some examples of techniques used to partition irregular graph structures. Because of the irregular structure of the graph, distributing the workload equally to four processors would need a recursive bisection, a divide and conquer technique, where the graph is divided between the four processors, but is not an optimal way of partitioning.

5. What is the "big idea" that is used for solving the N-Body big data problems? What is the difference between Barnes-Hut Algorithm and Fast Multipole Method? Illustrated with examples and demonstrated the complexity of each method.

If we were to assume that the answer at each point depends on data at all other points, then the solution we currently have would need to be at least $O(n^2)$ of work, because the point where the answer is must communicate or depend on at least one other point. The big idea then is if the dependence of data that is "distant" is compressed, we were to make the communication between points simpler, then by compressing data of groups of nearby points, we can cut the cost of communication at distant points. If we apply this idea recursively, the cost drops to O(n log n) or even O(n).

The Barnes-Hut algorithm has a time complexity of O(n log n); is good for low accuracy calculations, uses fixed information in every box but the number of boxes increases with

accuracy, and computes force. The Fast Multipole Method has a time complexity of O(n); computes every potential point, not just force. It uses a fixed number of boxes, but the amount of information in each box increases with accuracy. It is also difficult to implement.

Citation

Bader, Michael. “Cache Oblivious Matrix Multiplication Using an Element Ordering Based on a Peano Curve.” Linear Algebra and Its Applications, North-Holland, 2 May 2006, https://www.sciencedirect.com/science/article/pii/S0024379506001595.

Blelloch, Guy. “Cache-Oblivious Algorithms.” Algorithms in the Real World: Readings, Notes and Slides, 2018, https://www.cs.cmu.edu/afs/cs/project/pscico-guyb/realworld/www/slidesS18.html.

Chamberlain, Bradford L. “Graph Partitioning Algorithms for Distributing Workloads of Parallel Computations.” University of Washington, 1998, https://homes.cs.washington.edu/~bradc/.

Georganas, Evangelos, et al. “Communication Avoiding and Overlapping for Numerical Linear Algebra.” Grupo Arquitectura De Computadores - Universdade Da Coruna, https://gac.udc.es/~jorgeg/publications/SC12.pdf.

Jayaweera, Malith. “Blocked Matrix Multiplication.” Malith Jayaweera, 3 July 2020, https://malithjayaweera.com/2020/07/blocked-matrix-multiplication/.

Karypis, George, and Vipin Kumar. “Parallel Multilevel K-Way Partitioning Scheme for Irregular Graphs.” Association for Computing Machinery - Digital Library, https://dl.acm.org/doi/pdf/10.1145/369028.369103.

Santos, Eunice E. “Parallel Complexity of Matrix MULTIPLICATION1 - The Journal of Supercomputing.” SpringerLink, Kluwer Academic Publishers, June 2003, https://link.springer.com/article/10.1023/A:1023996628662.

Toma, Laura. “Algorithms in GIS (CSCI 3225).” Algorithms in GIS, https://tildesites.bowdoin.edu/~ltoma/teaching/cs3225-GIS/fall16/.

Yan, Yonghon. “CSCE 513 Computer Architecture, Fall 2018.” Index, https://passlab.github.io/CSCE513/.