Towards Automated Parallelization of Recursive Algorithms in SEJITS:

Beating MKL's Matrix Multiplication Using the BFS/DFS Approach

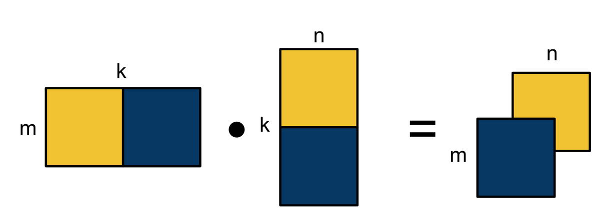
Jim Demmel, David Eliahu, Armando Fox, Ben Lipshitz, Oded Schwartz, Omer Spillinger

Summary

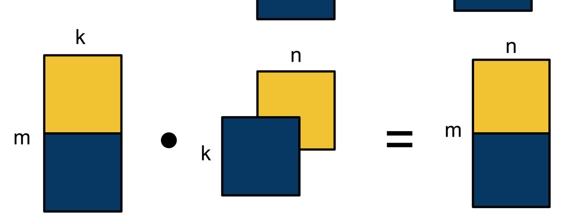
- We implement a Communication-Avoiding Recursive Matrix Multiplication algorithm (CARMA)
- First communication-optimal parallel rectangular matrix multiplication implementation (that we are aware of)
- Much simpler than the rectangular version of 2.5D [6] (~60 LOC)
- Faster than MKL in practice:
  - Faster for skinny matrices in which k is the largest dimension, up to 10x speedup
  - Faster for large square matrices, up to 20% speedup
  - Comparable performance for other matrix dimensions

Algorithm

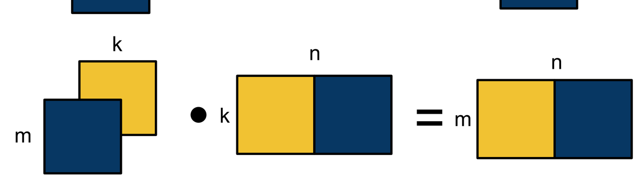
- Reduces communication by:
  - Using the BFS/DFS technique [2,3]
  - Always splitting along the longest dimension (m, k, or n)



- split along the dimension k



- split along the dimension m



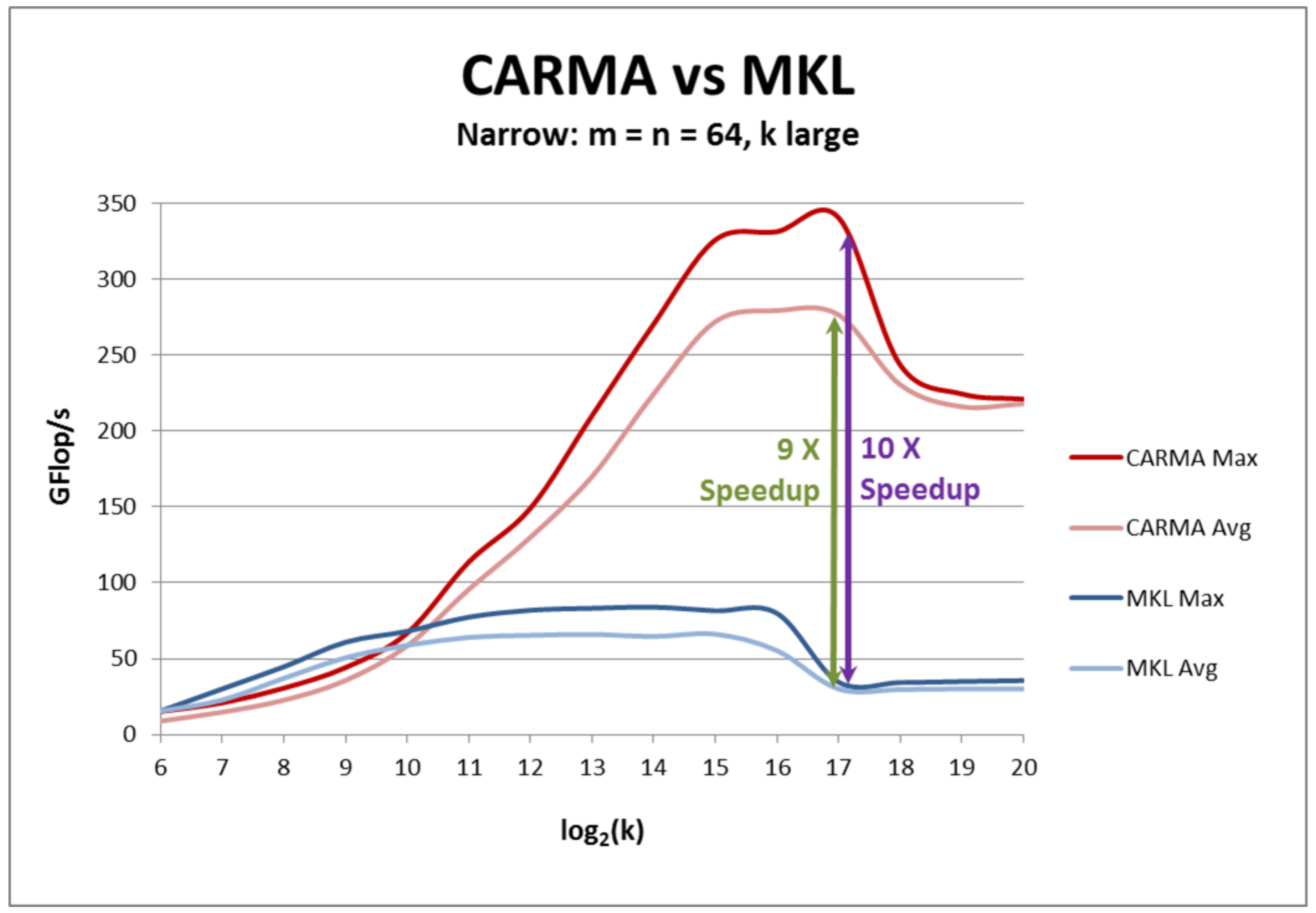
- split along the dimension n

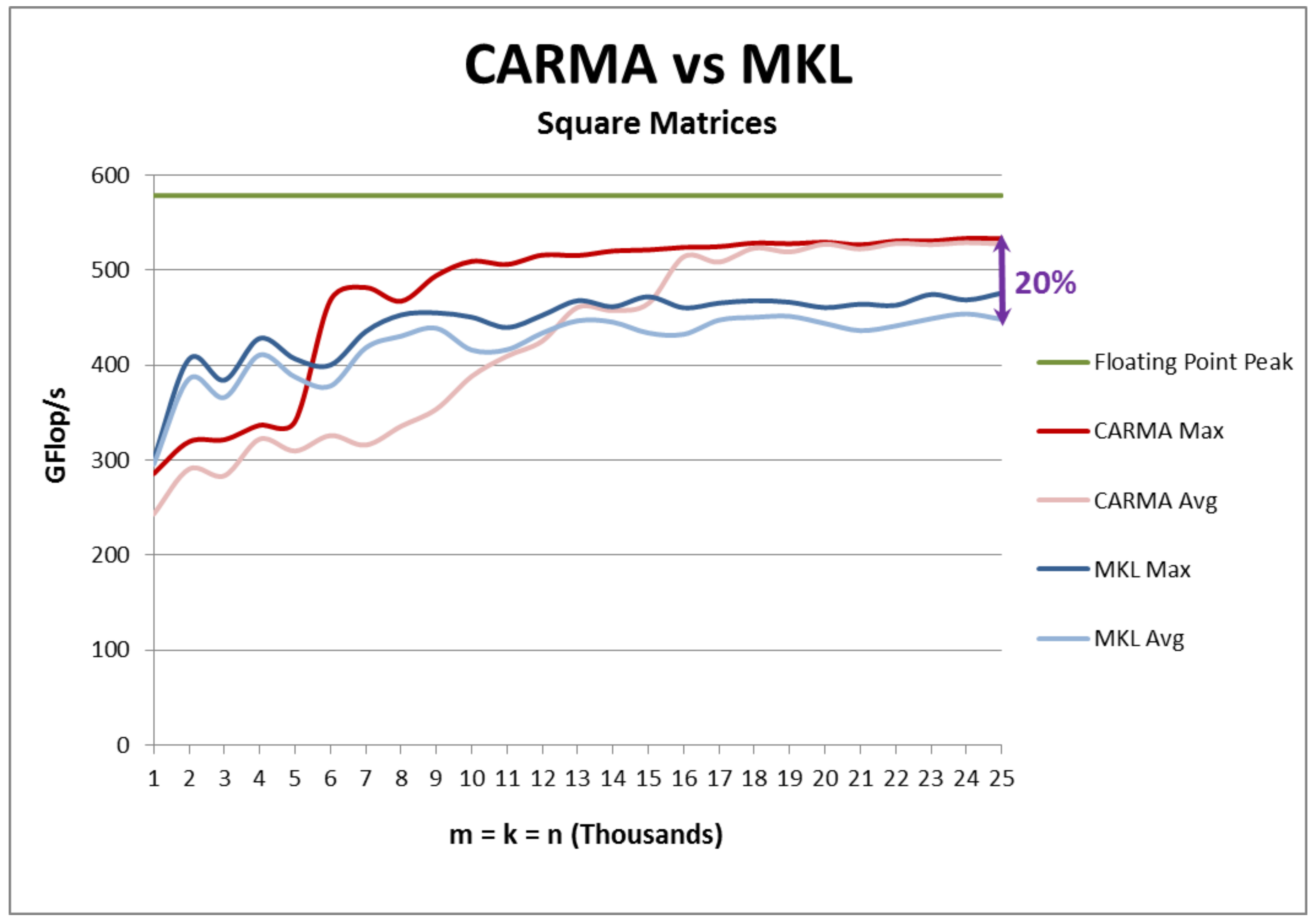
- Independent breadth first search (BFS) / depth first search (DFS) decisions made at each level of recursion tree, MKL used to solve subproblems
- BFS steps execute two subproblems independently on separate processors
  - Requires replication of the smallest matrix
- DFS steps execute two subproblems sequentially and decrease memory usage
  - Increases future communication costs
- Interleaving BFS and DFS steps yields an algorithm which remains in the bounds of RAM

Motivation

- Splitting along the largest dimension is communication optimal [4]
- BFS / DFS approach has already proved useful in the distributed model, both for the classical algorithm and for Strassen-based algorithms. See poster titled "Communication-Avoiding Parallel Strassen: Implementation and Performance"
- Reduces communication cost between local and shared memory
- BFS / DFS recursive approach guarantees optimal (up to a constant factor) performance for any hardware and input parameters

Results





Tests performed on emerald.millennium: 4 NUMA regions each with 8 cores (Intel® Xeon® Processor X7560)
using version 10.3.6 of Intel's Math Kernel Library (MKL)
average and Max are over 15 trials in the square case and 40 in the narrow case

Communication Lower bounds

- Bandwidth cost lower bound: [5]

$$BW = \Omega\left(\frac{mnk}{P\sqrt{M}}\right)$$

  - Attainable only if

$$M = O\left(\min\{m, n, k\}^2\right)$$

- Although these results are for distributed memory systems, they also apply to communication costs between NUMA regions

Future Work

- Explore additional performance gains by:
  - Using different recursive tree fanouts
  - Creating subproblems to exploit NUMA regions
  - Writing a multi-node version of CARMA using MPI
  - Using N-Morton and other recursive data layouts
  - Auto-tuning for optimal depth and pattern of BFS/DFS
- Extend the automatic parallelization of recursive sequential algorithms
  - Incorporate into SEJITS [3]
  - Generalize CARMA to similar algorithms (e.g. Floyd-Warshall, FFT)
  - Parallelize arbitrary sequential recursive algorithms

References

[1] G. Ballard, J. Demmel, B. Lipshitz, O. Schwartz. Communication-Avoiding Parallel Strassen: Implementation and Performance. Poster session presented at: Par Lab Summer Retreat 2012. Tecnical Report UCB/EECS-2012-90, 2012

[2] G. Ballard, J. Demmel, B. Lipshitz, O. Schwartz. Communication-Optimal Parallel Algorithm for Strassen's Matrix Multiplication. Symposium on Parallelism in Algorithms and Architectures, 2012.

[3] B. Catanzaro, S. A. Kamil, Y. Lee, K. Asanovic, J. Demmel, K. Keutzer, J. Shalf, K. A. Yelick, A. Fox. SEJITS: Getting Productivity and Performance With Selective Embedded JIT specialization. Technical Report UCB/EECS-2010-23, 2010.

[4] M. Frigo, C. Leiserson, H. Prokop, S. Ramachandran. Cache-Oblivious Algorithms. Proceedings of the 40th Annual Symposium on Foundations of Computer Science, 1999.

[5] D. Irony, S. Toledo, A. Tiskin. Communication Lower Bounds for Distributed-Memory Matrix Multiplication. Journal of Parallel and Distributed Computing, 2004.

[6] E. Solomonik, J. Demmel. Communication Optimal Parallel 2.5D Matrix Multiplication and LU Factorization Algorithms. Euro-Par 2011.

Research supported by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #DIG07-10227). Additional support comes from Par Lab affiliates National Instruments, Nokia, NVIDIA, Oracle, and Samsung.