Contention Bounds for Combinations of Computation Graphs and Network Topologies

Grey Ballard
Sandia National Laboratories
gmballa@sandia.gov

James Demmel
UC Berkeley
demmel@cs.berkeley.edu

Andrew Gearhart
UC Berkeley
agearh@cs.berkeley.edu

Benjamin Lipshitz*
UC Berkeley
lipshitz@cs.berkeley.edu

Oded Schwartz
Hebrew University†
odedsc@cs.huji.ac.il

Sivan Toledo
Tel-Aviv University
stoledo@tau.ac.il

Good connectivity of the inter-processor network is necessary for efficient parallel algorithms. Insufficient graph-expansion of the network provably slows down specific parallel algorithms that are communication intensive. While parallel algorithms that ignore network topology can suffer from contention along network links, for particular combinations of computations and network topologies, costly network contention may be inevitable, even for optimally designed algorithms. In this paper we obtain novel lower bounds on this *contention cost*.

Most previous communication cost lower bounds for parallel algorithms utilize *per-processor* analysis. That is, the lower bounds establish that some processor must communicate a given amount of data. These include classical matrix multiply, direct and iterative linear algebra algorithms, FFT, Strassen and Strassen-like fast algorithms, graph related algorithms, N-body, sorting, and others (cf. [1, 14, 12, 18, 15, 5, 3, 8, 11, 2, 16, 20, 10, 19]). By considering the network graphs, we introduce communication lower bounds for certain computations and networks that are tighter than those previously known. We translate per-processor bandwidth cost lower bounds to contention cost lower bounds by bounding the communication needs between a subset of processors and the rest of the processors for a given parallel algorithm (defined by a computation graph and work assignment to the processors), and divide by the available bandwidth, namely the words that the network allows to communicate simultaneously between the subset and the rest of the graph.

Contention Lower Bound. Consider a parallel algorithm run on a distributed-memory machine with P processors and connected via network graph G_{Net}. The *per-processor bandwidth cost* W_{proc} is the maximum over processors $1 \le p \le P$ of the number of words sent or received by processor p. Further, the *contention cost* W_{link} is the maximum over edges e of G_{Net} of the number of words communicated along e.

We prove the lower bound using graph expansion analysis. Recall that the small set expansion $h_s(G)$ of a graph $G = (V, E)$ is the minimum normalized number of edges leaving a set of vertices of size at most s. For $s \le |V(G)|/2$, we have

$$h_s(G) = \min_{S \subseteq V(G), |S| \le s} \frac{|E(S, V \setminus S)|}{|E(S)|}$$

where $E(S)$ is the set of edges that have at least one endpoint in vertex subset S and $E(S, V \setminus S)$ is the set of edges with only one endpoint in S. In this note, we provide the contention cost lower bound for regular networks:

THEOREM 1. *Consider a distributed-memory machine with P processors, each with local memory of size M, and a d-regular inter-processor network graph G_{Net}. Given a computation with input and output data size N, and lower bound on the per-processor bandwidth cost $W_{proc} = W_{proc}(P, M, N)$, for all algorithms that distribute the workload so that every processor performs $\Omega(1/P)$ of the computation, and distributing the input and output data such that every processor stores $O(1/P)$ of the data, the contention cost $W_{link} = W_{link}(P, M, N)$ is bounded below by*

$$W_{link}(P, M, N) \ge \max_{t \in T} \frac{W_{proc}(P/t, M \cdot t, N)}{d \cdot t \cdot h_t(G_{Net})}, \text{ where}$$

$$T = \{t : 1 \le t \le P/2, \exists S \subseteq V \text{ s.t. } |S| = t \text{ and } |E(S, V \setminus S)| = \Theta(h_t(G_{Net}) \cdot |E(S)|)\}.$$

PROOF. Partition the P processors into P/t subsets of size $t \in T$ (w.l.o.g., P is divisible by t), where at least one of the subsets s_t is connected to the rest of the graph with at most $d \cdot t \cdot h_t(G_{Net})$ edges. The existence of such a set s_t is guaranteed by the definition of $h_s(G_{Net})$ and T. Then s_t has a total of $M \cdot t$ local memory. By the workload distribution assumption, the processors in s_t perform a fraction $\Omega(t/P)$ of the flops, and by the data distribution assumption, s_t has local access to fraction $O(t/P)$ of the input/output. Hence we can emulate this computation by a parallel machine with P/t processors, each with $M \cdot t$ local memory, and apply the corresponding per-processor lower bound deducing that the processors in s_t require at least $W_{\text{proc}}(P/t, M \cdot t, N)$ words to be sent/received to the processors outside s_t throughout the running of the algorithm. At most $O(d \cdot t \cdot h_t(G_{Net}))$ edges connect s_t to the rest of the graph. Hence at least one edge communicates at least $\Omega\left(\frac{W_{\text{proc}}(P/t, M \cdot t, N)}{d \cdot t \cdot h_t(G_{Net})}\right)$ words. As t is a free parameter, we can pick it to maximize $W_{\text{link}}(P, M, N)$, and the theorem follows. □

Note that the memory-independent contention lower bound, $W_{\text{link}} = W_{\text{link}}(P, N)$, follows.

*Current affiliation: Google Inc.

†This work was done while at UC Berkeley.

Applications. We next demonstrate our bounds for direct dense linear algebra algorithms (including classical matrix multiplication) and fast matrix multiplication algorithms (such as Strassen's algorithm) on D-dimensional tori networks. Table 1 summarizes the contention bounds obtained by plugging in memory-dependent and memory-independent lower bounds for matrix multiplication and other linear algebra computations from [15, 6, 3] into Theorem 1 and using the properties of D-dimensional tori. The D-dimensional torus graph G_{Net} has degree $d = 2D$ and small set expansion guarantee of $h_s(G_{Net}) = \Theta\left(s^{-1/D}\right)$, see [9]. We treat D here as a constant. Table 1 summarizes the bounds.

		Mem. Dep.	Mem. Indep.
Direct Linear Algebra	W_{proc}	$\Omega\left(\frac{n^3}{PM^{1/2}}\right)$	$\Omega\left(\frac{n^2}{P^{2/3}}\right)$
	W_{link}	$\Omega\left(\frac{n^3}{P^{3/2-1/D}M^{1/2}}\right)$	$\Omega\left(\frac{n^2}{P^{1-1/D}}\right)$
Strassen and Strassen -like	W_{proc}	$\Omega\left(\frac{n^{\omega_0}}{PM^{\omega_0/2-1}}\right)$	$\Omega\left(\frac{n^2}{P^{2/\omega_0}}\right)$
	W_{link}	$\Omega\left(\frac{n^{\omega_0}}{P^{\omega_0/2-1/D}M^{\omega_0/2-1}}\right)$	$\Omega\left(\frac{n^2}{P^{1-1/D}}\right)$

Table 1: Per-processor bounds (W_{proc}) ([15, 5, 3, 6]) vs. the new contention bounds (W_{link}) on a D-dimensional torus for classical linear algebra and fast matrix multiplication (where ω_0 is the exponent of the computational cost).

Note that of the two contention bounds, the memory-independent one always dominates in these cases:

$$W = \Omega\left(\frac{n^{\omega_0}}{P^{\omega_0/2-1/D}M^{\omega_0/2-1}} + \frac{n^2}{P^{1-1/D}}\right) = \Omega\left(\frac{n^2}{P^{1-1/D}}\right),$$

by the fact that $P \geq P_{min} \geq n^2/M$, where ω_0 is the exponent of the computational cost.

Depending on the dimension of the torus D and number of processors, the tightest bound may be one of the previously known per-processor bounds or the memory-independent contention bound. See Figure 1 for the case of Strassen bounds on torus networks of various dimensions. For example, $D = 3$ is enough for perfect strong scaling of classical matmul but Strassen may need $D = 4$. Recall that perfect strong scaling is when, for a constant problem size, doubling the number of processors halves the runtime. Note that (see Figure 1) a contention-dominated range has a smaller region of perfect strong scaling.

Future Research. In this work, we exclusively address link contention bounds for a subset of direct network topologies (the analysis of tori extends to meshes, and can be extended to hypercubes). We believe results for certain indirect network topologies (e.g. fat trees) should follow, though this requires integrating router nodes into the model.

We focus here on a subset of linear algebraic computations. Our results extend to further computations such as the $O(n^2)$ n-body problem, FFT/sorting and programs that access arrays with affine expressions.

A network may have expansion sufficiently large to preclude the use of our contention bound on a given computation, yet the contention may still dominate the communication cost. This calls for further study on how well computations and networks match each other. Similar questions have been addressed by Leiserson and others [7, 13, 17], and had a large impact on the design of supercomputer networks.

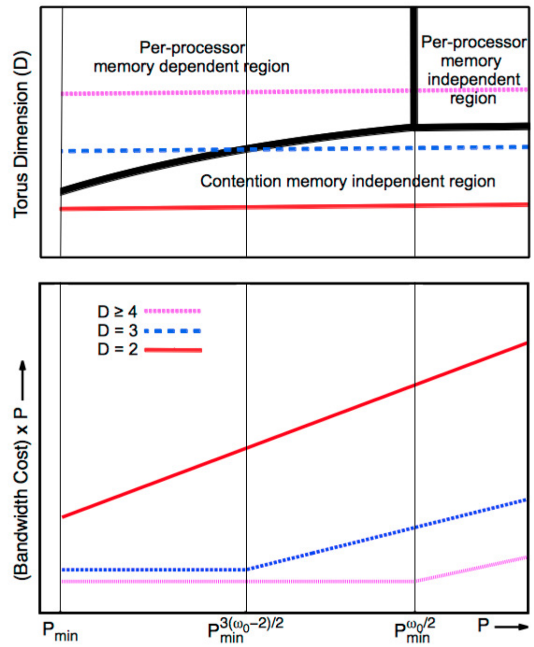

Figure 1: Communication bounds for Strassen's algorithm on D-dim. tori. Both plots share a log-scale x-axis in P. The upper plot illustrates the dominating bound, and is linear on the y-axis. The y-axis of the lower plot is log-scale, and horizontal lines represent perfect strong scaling.

Some parallel algorithms are network aware, and attain the per-processor communication lower bounds, when network graphs allow it (cf. [21] for classical matrix multiplication on 3D torus). Many algorithms are communication optimal when all-to-all connectivity is assumed, but their performance on other topologies has not yet been studied. Are there algorithms that attain the communication lower bounds for any realistic network graph (either by auto tuning, or by network-topology-oblivious tools)?

Acknowledgments. We thank Guy Kindler for pointing us to [9]. Research partially funded by DARPA Award Number HR0011-12-2-0016, the Center for Future Architecture Research, a member of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and ASPIRE Lab industrial sponsors and affiliates Intel, Google, Nokia, NVIDIA, Oracle, MathWorks and Samsung. Research is also supported by DOE grants DE-SC0004938, DE-SC0005136, DE-SC0003959, DE-SC0008700, AC02-05CH11231, and DE-SC0010200. Research is supported by grant 1045/09 from the Israel Science Foundation (founded by the Israel Academy of Sciences and Humanities), and grant 2010231 from the US-Israel Bi-National Science Foundation. This research is supported by grant 3-10891 from the Ministry of Science and Technology, Israel. This research was supported in part by an appointment to the Sandia National Laboratories Truman Fellowship in National Security Science and Engineering, sponsored by Sandia Corporation (a wholly owned subsidiary of Lockheed Martin Corporation) as Operator of Sandia National Laboratories under its U.S. Department of Energy Contract No. DE-AC04-94AL85000. Any opinions, findings, conclusions, or recommendations in this paper are solely those of the authors and does not necessarily reflect the position or the policy of the sponsors.

1. REFERENCES

[1] A. Aggarwal, A. K. Chandra, and M. Snir. Communication complexity of PRAMs. *Theor. Comput. Sci.*, 71:3–28, March 1990.

[2] G. Ballard, A. Buluç, J. Demmel, L. Grigori, B. Lipshitz, O. Schwartz, and S. Toledo. Communication optimal parallel multiplication of sparse random matrices. In *SPAA'13: Proceedings of the 25rd ACM Symposium on Parallelism in Algorithms and Architectures*, 2013.

[3] G. Ballard, J. Demmel, O. Holtz, B. Lipshitz, and O. Schwartz. Brief announcement: strong scaling of matrix multiplication algorithms and memory-independent communication lower bounds. In *Proceedings of the 24th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '12, pages 77–79, New York, NY, USA, 2012. ACM.

[4] G. Ballard, J. Demmel, O. Holtz, B. Lipshitz, and O. Schwartz. Graph expansion analysis for communication costs of fast rectangular matrix multiplication. In G. Even and D. Rawitz, editors, *Design and Analysis of Algorithms*, volume 7659 of *Lecture Notes in Computer Science*, pages 13–36. Springer Berlin Heidelberg, 2012.

[5] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. Minimizing communication in numerical linear algebra. *SIAM Journal on Matrix Analysis and Applications*, 32(3):866–901, 2011.

[6] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. Graph expansion and communication costs of fast matrix multiplication. *Journal of the ACM*, 59(6):32:1–32:23, Dec. 2012.

[7] P. Bay and G. Bilardi. Deterministic on-line routing on area-universal networks. In *Proceedings of the 31st Annual Symposium on the Foundations of Computer Science (FOCS)*, pages 297–306, 1990.

[8] G. Bilardi, M. Scquizzato, and F. Silvestri. A Lower Bound Technique for Communication on BSP with Application to the FFT. In *Euro-Par 2012 Parallel Processing*, pages 676–687. Springer, 2012.

[9] B. Bollobás and I. Leader. Edge-isoperimetric inequalities in the grid. *Combinatorica*, 11(4):299–314, 1991.

[10] M. Christ, J. Demmel, N. Knight, T. Scanlon, and K. Yelick. Communication lower bounds and optimal algorithms for programs that reference arrays - part 1. Technical Report UCB/EECS-2013-61, EECS Department, University of California, Berkeley, 2013.

[11] M. Driscoll, E. Georganas, P. Koanantakool, E. Solomonik, and K. Yelick. A communication-optimal n-body algorithm for direct interactions. In *proceedings of the IPDPS*, 2013.

[12] M. T. Goodrich. Communication-efficient parallel sorting. *SIAM J. Computing*, 29(2):416–432, 1999.

[13] R. I. Greenberg and C. E. Leiserson. Randomized routing on fat-tress. In *Proceedings of the 26th Annual Symposium on the Foundations of Computer Science (FOCS)*, pages 241–249, 1985.

[14] J. W. Hong and H. T. Kung. I/O complexity: The red-blue pebble game. In *Proc. 14th STOC*, pages 326–333, New York, NY, USA, 1981. ACM.

[15] D. Irony, S. Toledo, and A. Tiskin. Communication lower bounds for distributed-memory matrix multiplication. *J. Parallel Distrib. Comput.*, 64(9):1017–1026, 2004.

[16] N. Knight, E. Carson, and J. Demmel. Exploiting data sparsity in parallel matrix powers computations. In *Proceedings of PPAM '13*, Lecture Notes in Computer Science. Springer (to appear), 2013.

[17] C. E. Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers*, C-34(10):892–901, 1985.

[18] J. P. Michael, M. Penner, and V. K. Prasanna. Optimizing graph algorithms for improved cache performance. In *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS 2002), Fort Lauderdale, FL*, pages 769–782, 2002.

[19] M. Scquizzato and F. Silvestri. Communication lower bounds for distributed-memory computations. *arXiv preprint arXiv:1307.1805*, 2014. STACS'14.

[20] E. Solomonik, E. Carson, N. Knight, and J. Demmel. Tradeoffs between synchronization, communication, and work in parallel linear algebra computations. Technical Report (Submitted to SPAA'14), University of California, Berkeley, Department of Electrical Engineering and Computer Science, 2013.

[21] E. Solomonik and J. Demmel. Communication-optimal parallel 2.5D matrix multiplication and LU factorization algorithms. In *Euro-Par'11: Proceedings of the 17th International European Conference on Parallel and Distributed Computing*. Springer, 2011.