SAND2014-0680C

Contention Bounds for Combinations of Computation Graphs and Network Topologies*

[Brief Announcement]

Grey Ballard
Sandia National Laboratories
gmballa@sandia.gov

James Demmel
UC Berkeley
demmel@cs.berkeley.edu

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

Oded Schwartz
UC Berkeley
odedsc@cs.berkeley.edu

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

ABSTRACT

Network topologies can have significant effect on inter-processor communication costs of algorithms. Parallel algorithms that ignore network topology can suffer from contention along network links. However, for particular combinations of computations and network topologies, costly network contention may be inevitable, even for optimally designed algorithms. We obtain a novel contention lower bound that is a function of the network and the computation graph parameters. To this end, we compare the communication bandwidth needs of subsets of processors and the available network capacity (as opposed to per-processor analysis in most previous studies). Applying this analysis we improve communication cost lower bounds for several combinations of fundamental computations on common network topologies.

Categories and Subject Descriptors

F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*Computations on matrices*

General Terms

Algorithms, Design, Performance.

Keywords

Network topology, Communication-avoiding algorithms, Strong scaling, Communication costs.

*We acknowledge funding from Microsoft (Award #024263) and Intel (Award #024894), and matching funding by U.C. Discovery (Award #DIG07-10227). Additional support comes from ParLab affiliates National Instruments, Nokia, NVIDIA, Oracle and Samsung, as well as MathWorks. Research is also supported by DOE grants DE-SC0004938, DE-SC0005136, DE-SC0003959, DE-SC0008700, DE-FC02-06-ER25786, AC02-05CH11231, and DARPA grant HR0011-12-2-0016. 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.
** Current affiliation: Google Inc.

SPAA'14, June 23 - 25, 2014 Prague, Czech Republic.

1. INTRODUCTION

Good connectivity of the inter-processor network is a necessary for fast execution of parallel algorithm. 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 costs*.

We use a variant of the distributed-memory communication model (cf, [10, 11, 6]), where the bandwidth-cost of an algorithm is proportional to the number of words communicated by one processor along the critical path (we omit the latency cost / message count discussion from this note). As in the distributed-memory communication model we have P processors and local memories of size M for each processor. However, here, we do not assume all-to-all connectivity, but rather some network graph with P vertices G_{Net}. In this note we assume all edges (network links) have same bandwidth. We leave out memory injection-rate issues from this model. For the sake of simplicity, we assume in this note: that the workload is perfectly balanced (so all processors perform the same number of flops); that no re-computation is performed (i.e., between processors); that input data is initially evenly distributed across all processors (and each item appears once); and that similarly for the output data at the end.

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, 16, 14, 21, 17, 6, 3, 9, 13, 2, 18, 24, 12, 23]).

By considering the network graphs, we introduce tighter communication lower bounds for certain computations and networks than previously known. We bound the communication needs between a subset of processors and the rest of the processors for a given parallel algorithm (computation graph and work assignment to the processors), and divide it by the number of words that the network allows to communicate simultaneously between the subset and the rest of the graph. We call this the *contention cost*. Applying the main theorem we improve (i.e., increase) communication cost lower bounds for several combinations of fundamental computations on common network topologies. These contention bounds are known to be attainable only for several combinations, thus motivating further algorithmic research. They may suggest directions for hardware/network design tailored for heavily used computation kernels and may assist when scheduling users' applications to a supercomputer.

2. CONTENTION LOWER BOUNDS

Small set expansion $h_s(G)$ of a d-regular graph $G = (V, E)$ is the minimum normalized number of edges leaving a set of vertices of size at most s. Formally,
$h_s(G) = \min_{S \subseteq V(G), |S| \leq s} \frac{|E(S,\bar{S})|}{d|S|}$, where $|E(S, \bar{S})|$ is the number of edges connecting S to the rest of the graph.

DEFINITION 2.1. *Let Alg be a parallel algorithm run on a distributed model with P processors, and a network graph G_{Net}. The* contention cost W_{Cont} *is the maximum over edges e of $E(G_{Net})$ of the number of words communicated on e.*

THEOREM 2.2. *Let Alg be a parallel algorithm run on a distributed model with P processors, each with local memory of size M, network graph G_{Net}. Assume that workload is perfectly balanced , that data is evenly distributed, both the input (of size N), and the output. Let $W(P, M, Alg, N)$ be the communication costs (by per-processor analysis). Then the contention costs is*

$$W_{Cont}(P, M, G_{Net}, Alg, N) \geq \max_{t \in [P]} \frac{W(P/t, M \cdot t, Alg, N)}{t \cdot h_t(G_{Net})}$$

PROOF. Consider a partitioning of the P processors into P/t subsets of size 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 network graph with exactly $t \cdot h_t(G_{Net})$ edges (the existence of such set s_t is guaranteed by the definition of $h_s(G_{Net})$. s_t has a total of $M \cdot t$ local memory. By perfect workload distribution, the processors in s_t perform t/P fraction of the flops workload, and by the perfectly balanced data distribution assumption s_t has local access to t/P fraction 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 analysis deducing that the processors in s_t require at least $W(P/t, M \cdot t, Alg, N)$ words to be sent/received to the processors outside s_t throughout the running of the algorithm. Exactly $t \cdot h_t(G_{Net})$ edges connect s_t to the rest of the graph. Hence at lease one edge communicates $\frac{W(P/t, M \cdot t, N)}{t \cdot h_t(G_{Net})}$ words. Since t is a free parameter, we can pick it to maximize W_{Cont}, and the theorem follows. □

Observe that for W we can plug in both types of per-processor lower bounds: memory independent (cf. [3]) and memory dependent ones (cf. [17, 5, 7, 4]).

Applications.

In this note we demonstrate our bounds on a few combinations only, namely direct linear algebra algorithms, Strassen, and Strassen-like matrix multiplication on d-dimensional tori networks. Table 1 and Figure 1 summarize the contention bounds obtained by plugging in memory dependent and memory independent lower bounds from [17, 7, 3] into Theorem 2.2, and by the properties of d-dimensional tori: they have a degree $2d$ and small set expansion guarantee of $h_s(G_{Net}) = \Theta\left(s^{(d-1)/d}/s\right)$. The contention bounds for naive N-body problem on tori do not improve previously known memory dependent and independent bounds ($\Omega(n^2/PM)$, $\Omega(n/\sqrt{P})$, cf. [13] and their references).

The memory-dependent bounds for classical and Strassen's matrix multiplication are contention dominated when $d < d_{dep} = \frac{2}{\omega_0 - 2}$. The memory-dependent bound dominates for $\frac{n^2}{M} \leq P \leq \left(\frac{n^2}{M}\right)^{d(\omega_0-2)/2}$. When this happens, we have a perfect strong scaling range. That is, for a fixed problem size, increasing the number of processors by a constant factor reduces the communication costs (and running time) by the same constant factor (see [3] for further discussion). Note that this range is smaller than the perfect strong scaling range when per-processor bounds dominate. For Strassen's

		Per-processor	Contention
Direct Linear Algebra	Memory Dep.	$\Omega\left(\frac{n^3}{PM^{3/2-1}}\right)$	$\Omega\left(\frac{n^3}{P^{3/2-1/d}M^{3/2-1}}\right)$
	Memory Indep.	$\Omega\left(\frac{n^2}{P^{2/3}}\right)$	$\Omega\left(\frac{n^2}{P^{(d-1)/d}}\right)$
Strassen and Strassen -like	Memory Dep.	$\Omega\left(\frac{n^{\omega_0}}{PM^{\omega_0/2-1}}\right)$	$\Omega\left(\frac{n^{\omega_0}}{P^{\omega_0/2-1/d}M^{\omega_0/2-1}}\right)$
	Memory Indep.	$\Omega\left(\frac{n^2}{P^{2/\omega_0}}\right)$	$\Omega\left(\frac{n^2}{P^{(d-1)/d}}\right)$

Table 1: Per-processor bounds ([17, 3, 7]) vs. the new contention bounds on d dimensional torus for direct linear algebra cubic time algorithms (including classical matrix multiplication) and fast matrix multiplication ($\omega_0 = \log_2 7$ for Strassen's algorithm).

matrix multiplication this is a pretty significant effect. For a good enough network, the perfect strong scaling range is $P_0 < P < P_0^{\omega_0/2} \approx P_0^{1.40}$; for a 3d torus, the perfect strong scaling range is $P_0 < P < P_0^{3(\omega_0-2)/2} \approx P_0^{1.21}$.

When $d < d_{dep} = \frac{2}{\omega_0-2}$, both contention bounds apply, but the memory-independent one always dominates:

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

(using $P \geq n^2/M$). Table 2 summarizes the required dimension of the torus so the contention bound is not the bottleneck (for memory dependent and independent bounds).

Algorithm	ω_0	Contention free at torus dimension: dependent	independent
Classical	3	3	2
Strassen [26]	≈ 2.81	4	3
Schönhage[22]	≈ 2.55	5	4
Strassen [27]	≈ 2.48	6	5
Vassilevska [28]	≈ 2.3727	7	6

Table 2: Minimum torus dimension, so that communication-cost is not contention-bounded for a selection of algorithms. Memory dependant (perfect strong scaling range) and independent requirements are shown separately. Last three algorithms are under some technical assumption / conjecture, see details in [7].

3. DISCUSSION AND FUTURE RESEARCH

Bisections. For matrix multiplication algorithms (and many other ones) on common network topologies, the contention lower bound is maximized for subsets of processors of size about $P/2$; that is, when the inter-processor network bisection is concerned. Is this always the case, or do we expect to have combinations of algorithms and networks where contention bounds dominate, but for cuts other than the bisection? Contrived example could be when $h_s(G_{Net})$ is not a decreasing function of s. For example, two networks of processors, a large and a small one, where each of them is well connected, but the connection between the large and the small one is narrow.

Applicability. Other immediate applications of the main theorem for combinations of networks (e.g., tori of various dimensions, meshes, hypercubes, fat-tree and dragonfly) and

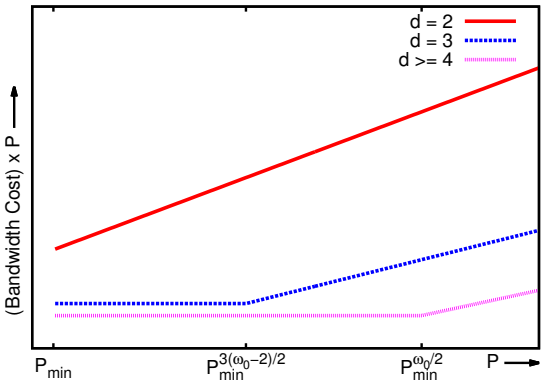

Figure 1: Communication bounds for Strassen's algorithm on d-dimensional tori (log-log scale). Horizontal lines correspond to perfect strong scaling. $P_{\min}$ is the minimum number of processors required to store the input.

classes of algorithms (e.g., algorithm that access arrays, see [12]) are excluded from this note. Note however that good enough network expansion is not always good enough. A network may have expansion sufficiently large to preclude the use of our contention bound for a given computation, yet the contention may be inevitable for any parallel algorithm realizing the computation on the network. This calls for further study on how well computations and networks match each other. Similar questions have been addressed in a series of elegant papers by Leiserson and others carrying optimistic message, and having a large impact on how networks for parallel supercomputers are designed. They showed that the equivalent of graph expansion in the physical world is essentially sufficient [8, 15, 19]. In particular, parallel computer that uses a fat tree communication network can simulate any other computer, at the cost of at most polylogarithmic slowdown.

Communication Efficient Algorithms. Some parallel algorithms are network aware, and attain the per-processor communication lower bounds, when network graphs allows it, (cf. [20, 25] 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 (e.g., for matrix multiplication) that attain the communication lower bounds for any network graph (either by auto tuning, or by network-topology-oblivious tools)? Can we benchmark the effect of contention lower bound in practice, on various combinations of computations and supercomputers?

4. 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 *Proceedings of the Mediterranean Conference on Algorithms*, 2012. To appear. Available as UC Berkeley Technical Report EECS-2012-194.

[5] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. Minimizing communication in numerical linear algebra. Submitted. Available from http://arxiv.org/abs/0905.2485, 2010.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] L. S. Blackford, J. Choi, A. Cleary, E. D?Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*. SIAM, Philadelphia, PA, USA, May 1997. Also available from http://www.netlib.org/scalapack/.

[11] E. Chan, M. Heimlich, A. Purkayastha, and R. Van De Geijn. Collective communication: theory, practice, and experience. *Concurrency and Computation: Practice and Experience*, 19(13):1749–1783, 2007.

[12] 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, May 2013.

[13] 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.

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

[15] 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.

[16] J. W. Hong and H. T. Kung. I/O complexity: The red-blue pebble game. In *STOC '81: Proceedings of the thirteenth annual ACM Symposium on Theory of Computing*, pages 326–333, New York, NY, USA, 1981. ACM.

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

[18] 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.

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

[20] W. F. McColl and A. Tiskin. Memory-efficient matrix multiplication in the BSP model. *Algorithmica*, 24:287–297, 1999. 10.1007/PL00008264.

[21] 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.

[22] A. Schönhage. Partial and total matrix multiplication. *SIAM Journal on Computing*, 10(3):434–455, 1981.

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

[24] 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.

[25] 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.

[26] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.

[27] V. Strassen. Relative bilinear complexity and matrix multiplication. *Journal fűr die reine und angewandte Mathematik (Crelles Journal)*, 1987(375–376):406–443, 1987.

[28] V. V. Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th Symposium on Theory of Computing*, STOC '12, pages 887–898, New York, NY, USA, 2012. ACM.