

© 2002 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Citation:

L. Wittie, G. Sazaklis, Yaping Zhou and D. Zinoviev, "High throughput networks for petaflops computing," Proceedings Seventeenth IEEE Symposium on Reliable Distributed Systems (Cat. No.98CB36281), 1998, pp. 312-317, doi: 10.1109/RELDIS.1998.740515.

Doi:

<https://doi.org/10.1109/RELDIS.1998.740515>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

High Thruput Nets for Petaflops Computing

L. Wittie G. Sazaklis Y. Zhou D. Zinoviev *

May 1, 1998

1 Motivation

This work was undertaken to estimate the complexity of networks that can be used to connect several thousand processing elements and memory interfaces in a Petaflops cryocomputer [3]. Two different network architectures have been studied in detail: multistage banyan networks and truncated ("pruned") multidimensional meshes. For each architecture, we simulated many network shapes to determine the maximal aggregate throughput T and average latency λ_{avg} . We have learned many tricks for increasing network throughput without increasing the number of switching nodes.

2 RSFQ Superconductor Logic

Rapid Single Flux Quantum (RSFQ) logic [5, 6] is being developed for computing in 2004. Its switches and gates use superconductor Josephson junctions (JJ), which need only 1/100,000th the power of semiconductor circuitry. Although our prototype planning assumes 10 picosecond (ps) per integer instruction cycle, RSFQ logic potentially needs less than 2 ps. Ultra-fast, tiny supercomputers can be built without overheating. RSFQ computing is not yet easy: circuits work only near absolute zero (below 5° Kelvin) and they need many more active devices (JJs) per gate than semiconductors do, roughly 7 JJs per memory bit or binary input gate.

The prototype cryocomputer envisioned in this study uses 4,096 processors ([2]) each producing eight floating results per 30 ps cycle. The target sustained computing power is 0.1 Petaflops (0.1×10^{15} FLOPS). The system needs about 128 TeraBytes of main DRAM memory to keep the processors busy, that is roughly one million 128 MB DRAMs filling a 2 meter cube. Memory access latency is roughly 60 ns (6,000 integer instruction times): 36 ns for signal flight at 1/2 light speed, 2 ns for thermal-shielding memory interfaces, 20 ns for DRAMs, and 2 ns for switching network delays. DRAM speed and physical size are the main factors in memory latency, which must be hidden.

In RSFQ computing networks, bits are signaled by the absence or presence of single quanta of magnetic flux in tiny superconducting current loops. Josephson junctions become resistive when

*Depts of Computer Science and Physics, SUNY at Stony Brook, lw@cs.sunysb.edu, 516-632-8456 (M,F) and 516-271-5162 (S,TWT,S) This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the National Security Agency (NSA) through an agreement with the National Aeronautics and Space Administration.

too much current passes through them. For circuits built from 0.8 micrometer (μm) wide niobium strips, JJs current-biased to be just sub-critical can switch from superconductive to resistive and back in 1 ps, allowing 10 ps computer cycle times. Gates operate by the convergence of signal bits overloading superconductive junctions to force output pulses elsewhere.

Transmission times between adjacent switching nodes on the same chip are 6 ps. Inter-node signals between chips need 30 ps to charge 2.0 μm microstrip lines. Recovery time between successive packets on a link is less than 30 ps. Our simulations use 30 ps per hop as the network cycle time.

3 Summary of the Network Topologies

Each data path is 100 bits wide to pass a whole word in one cycle. Since signal selection and merging gates involve many JJs in RSFQ circuits, it is critical to consider network switches with only 2 one-way inputs and 2 one-way outputs. Both Banyan multistage networks and pruned meshes can use these minimally complex switches.

Banyan networks allow 2^K sources to communicate with 2^K destinations using K stages of 2×2 switching nodes. There is a unique path for any source to destination pair.

Pruned meshes are based on work presented first at the 1980 Purdue Interconnection workshop [7, 8] and later used to interconnect modules in Tera computers [1]. They are D-cubes (symmetric networks with nearest neighbor connections in each of D dimensions) systematically pruned to limit the number of links at each node. Mesh dimensionality can grow without increasing switch complexity.

“Game rules” were established for comparison of the two network families:

- 4,096 processing elements (PE) and 4,096 memory interfaces (MI) are connected to the network; however, the traffic for each PE and MI may be spread over several switching nodes to avoid local congestion.
- during a clock cycle, every PE issues at most 3 memory requests and receives at most 2 replies from memory, for a total maximum network traffic of 5 packets; this figure is 2.3 times greater than the memory traffic needed if PEs have on-chip instruction caches.
- consistent with 5 packets per PE per cycle, acceptable networks must transmit at least $T_0 = 20,000$ packets per cycle, or consist of two one-way networks accepting 12,000 and 8,000 packets per cycle.
- given the minimal acceptable throughput T_0 , we must choose a network architecture and shape that provides the optimal trade-off between hardware complexity and average delay.

4 Banyan Nets

One design uses a flat Banyan network with internal buffers on all links. RSFQ logic is fundamentally asynchronous, so we use a request/acknowledgment mechanism based on credit (see [4]) for flow

control. Each of 4,096 PEs can deliver up to 8 packets per cycle to the network, each destined for some MI. The 4,096 MIs reply to PEs, but at $2/3$ the average PE rate.

In the smallest banyan design that can deliver 20,000 packets per cycle, there are 32,768 switching nodes in each of 16 layers. At the input end, each node has 2 input ports apiece; at the other end, each has two output ports. At both ends, each node connects to one MI and one PE. Each PE or MI is linked to 8 nodes at each end. Alternating less and more busy ports evens local traffic rates and keeps packets in a burst from one PE or MI from conflicting immediately for input ports.

There can be a single one-way banyan network for both traffic directions, PE-to-MI and MI-to-PE. Each network node is a 2-by-2 switch with two input queues. The number of slots per queue is the maximum credit value. The network forms a cylinder with the 4,096 PEs and the 4,096 MIs alternating in a layer that stitches together the input and output ends of the banyan.

5 Pruned Multidimensional Ring Meshes

Pruned multidimensional meshes have also been studied to connect processors to memories in a Petaflops computer. They consist of switching nodes regularly spaced along 2 to 6 axes. Each node is directly connected to four or more of its neighbors by links that form rings, each spanning one dimension. By using higher dimensions, like 5D or 6D, average path delays between randomly located nodes can be kept very low, roughly $DW/2$ round-trip for an $N = W^D$ mesh with two-way rings. By carefully eliminating internode rings, the number of links per switch can be made as low as 2×2 with little increase in average path lengths. Meshes have many more possible shapes than $H \log 2H$ networks such as the banyan, allowing much more precise tuning of sizes and traffic loads. As our results show, their asymptotic behavior is like that of multistage networks.

The key to pruning a mesh is to keep all rings that run in one “vertical” direction. Each node has links in one other direction. One-way rings need only 2×2 switches. A pruned 3D mesh is like an office building with many elevators but hallways that run in only one direction per floor. To reach an office on the same floor, one may have to take elevators between floors.

The basic routing protocol for a pruned mesh is to take the next free link that moves a packet closer to its destination. When a packet reaches a ring running in a “horizontal” (non-vertical) direction for which it has not reached its destination, it takes that horizontal ring if it is not too busy and is going in the proper “left” or “right” direction.

For high-throughput in meshes small enough to have high traffic per link:

- Give packets on vertical links priority over those on horizontal links to guarantee routing progress: once the destination displacement along one horizontal axis has been eliminated, a packet hops no more in that direction.
- Let packets on vertical links that need to eliminate a destination displacement prefer a horizontal ring that goes the shorter of the two possible ways; for rings with 10 positions, from 0 to 9, or 9 to 0, choose “right” from 7 to 8, not “left” from 7 to 6 ... to 9 to 8.
- The vertical ring width should be a multiple of the number of horizontal axes, so horizontal ring loads are the same, avoiding structural hotspots.

- For one-way rings, have two sets of horizontal rings per vertical ring: half go “right” and half go “left” in each dimension.
- Space active nodes (with PE or MI inputs) well apart so that no vertical or horizontal ring has too much local traffic from multiple active nodes.
- Spread inputs from each PE or MI over several nodes on different rings.
- So all links are equally saturated, make all horizontal rings of similar widths; for symmetric loads, ring traffic is proportional to ring width.
- Keep the mesh disproportionately short vertically, because empirically vertical link saturation rapidly leads to reduced throughput.

Meshes have a basic difference from banyans: a banyan has a unique path per source-destination pair; meshes have many different shortest routes for any pair. Consider all the equally short ways one can move diagonally across a large Manhattan street grid (if taxis and one-way streets do not get in the way). Short paths are important since average path length times maximum packet throughput is the roughly constant carrying capacity of any net.

6 General Results for Banyan and Mesh Nets

The use of credit-based flow control improves maximum throughput in both banyan and mesh networks by nearly a factor of three. In meshes with unbuffered one-way links, the smallest network that sustains a throughput of 20,000 packets per cycle has 1,331,000 nodes. With credit and careful use of adaptive routing, networks of 466,560 nodes and 4 buffers per link were adequate. Banyan networks gave similar improvements with credit flow control.

Careful placement of inputs from both PEs and MIs is critical to eliminate hotspots in both types of network. In general, input ports from the same PE or MI should be scattered over 4 to 8 nodes to avoid hotspot congestion during bursts of activity from any one node. For meshes, it is important to keep the maximum number of active switching nodes (ones with a PE or MI port) per ring as low and uniform as possible to avoid structural hotspots.

The JJ complexity of switching nodes grows super-linearly with the number of links per node. Network carrying capacity per node grows only linearly with links/node. It is better to use more 2x2 switches than fewer 4x4 switches.

7 Simulation Results

The following table summarizes results for both banyan and mesh networks that allow at least 20,000 packets/cycle to be delivered. Each Banyan network has 524,288 switching nodes to handle inputs from 4,096 processors and 4,096 memory interfaces spread over 32,768 + 32,768 ports. The pruned mesh results use a wide variety of node counts, with inputs spread over 8,182 + 24,576 ports for larger networks and 32,768 + 32,768 ports for the smaller nets. The complexity of each node in both banyan and mesh networks for 2x2 switches, 4 links per node and K buffers per link is

($16+8K$) JJs. Increasing the number of buffers used by the credit system improves both throughput and acceptance probability for any given number of nodes. However, having more buffers requires more Josephson junctions per node.

Credits	Load (pkts/cyc)	Throughput (packets/cycle)	Avg. Latency	Accept. Prob.	Cost (nodes)	Cost (JJ)
Full Load						
1	27295.7	8774.01	33.00	32.14 %	524,288	12 M
2	54591.5	22002.00	28.94	40.29 %	524,288	16 M
3	54591.5	27774.70	33.04	50.87 %	524,288	20 M
4	54591.5	31380.10	37.68	57.48 %	524,288	25 M
Expected Load (42 %)						
1	11464.21	8623.40	27.13	75.22 %	524,288	12 M
2	22928.46	21179.60	22.30	92.39 %	524,288	16 M
3	22928.46	22907.14	20.63	99.91 %	524,288	20 M
4	22928.46	22925.20	20.55	99.99 %	524,288	25 M
Best Banyan Solution ($T > 20K$, min. latency, min. cost)						
2	21836.6	20831.40	21.45	95.40 %	524,288	16 M
Full Load for Meshes						
1 6D	23412	21696.3	33.74	92.67 %	4x4 295,245	28 M
1 6D	23914	20498.8	66.02	85.72 %	2x2 1,331,000	32 M
2 6D	20836	20352.3	32.39	97.68 %	2x2 900,000	29 M
3 6D	23119	20472.0	38.64	88.55 %	2x2 656,100	26 M
4 6D	24682	20268.9	47.13	82.12 %	2x2 524,880	25 M
4 6D	27374	20076.1	40.18	73.34 %	2x2 466,560	22 M
Best Mesh Solution ($T > 20K$, min. latency, min. cost)						
4 6D	27374	20076.1	40.18	73.34 %	2x2 466,560	22 M

Smallest Banyan and Mesh Networks with Throughput Above 20,000 Packets/30 ps

Because of their superior response with just two input buffers per link, banyan networks have lower JJ complexity than pruned meshes with comparable throughput, even though the best mesh network has fewer nodes and links than the best banyan. The shape flexibility of meshes is more important for smaller computing networks.

References

- [1] R. Alverson, D. Callahan, D. Cummings, B. Koblenz, A. Porterfield, and B. Smith. The Tera Computer System. In *Int. Conf. on Supercomputing*, pages 1–6, June 1990.
- [2] P. Bunyk, M. Dorojevets, K. Likharev and D. Zinoviev. RSFQ Subsystem for HTMT Petaflops Computing. *Tech. Report 03*, SUNY HTMT RSFQ Group, Dec. 1997
- [3] G. Gao, K. K. Likharev, P. C. Messina, and T. L. Sterling. Hybrid Technology Multithreaded Architecture. In *Proc. Frontiers'96*, pages 98–105, Annapolis, MD, 1996. Available electronically via anonymous FTP from <ftp://rsfq1.physics.sunysb.edu/pub/ieee.htm.ps>.

- [4] H. Kung and R. Morris. Credit-based flow control for ATM networks. *IEEE Network Magazine*, pages 40–48, March/April 1995.
- [5] K. Likharev. Rapid Single-Flux-Quantum Logic. In H. Weinstock and R. Ralston, editors, *The New Superconducting Electronics*, pages 423–452. Kluwer, Dordrecht, The Netherlands, 1993.
- [6] K. Likharev. Ultrafast Superconductor Digital Electronics: RSFQ Technology Roadmap. *Czech. J. Phys.*, 46:3331–8, Dec. 1996.
- [7] L. D. Wittie. Architectures for Large Networks of Microcomputers. In *Proc. Workshop on Interconnection Networks*, IEEE, pages 31–40. Purdue Univ., April 1980.
- [8] L. D. Wittie. Communication Structures for Large Networks of Microcomputers. *IEEE Trans. on Computers*, C-30(4):264–273, April 1981.