Contention of Communications in Switched Networks with Applications to Parallel Sorting

UMBC REU Site: Interdisciplinary Program in High Performance Computing

Nil Mistry[1], Jordan Ramsey[2], Benjamin Wiley[3], and Jackie Yanchuck[4]

Graduate assistant: Xuan Huang[2], Faculty mentor: Matthias K. Gobbert[2]

Clients: Christopher Mineo[5] and David Mountain[5]

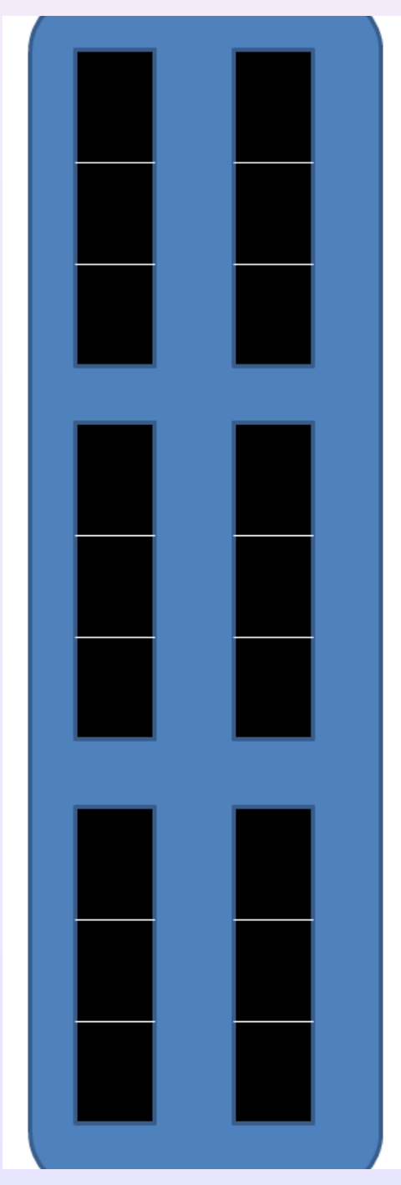
[1]University of Connecticut, [2]UMBC, [3]University of New Mexico, [4]Seton Hill University, [5]Advanced Computing Systems Research Program

Problem

Contention of communications across a switched network that connects multiple compute nodes in a distributed-memory cluster may seriously degrade the performance of parallel code. This contention is maximized when communicating large blocks of data among all parallel processes simultaneously. This communication pattern arises in many important algorithms such as parallel sorting. InfiniBand interconnects are the most popular high-performance networks in computing clusters presently. We use the cluster tara in the UMBC High Performance Computing Facility (HPCF) with a quad-data rate InfiniBand network to provide a test case if the capacity of a switched network can be a limiting factor in algorithmic performance.

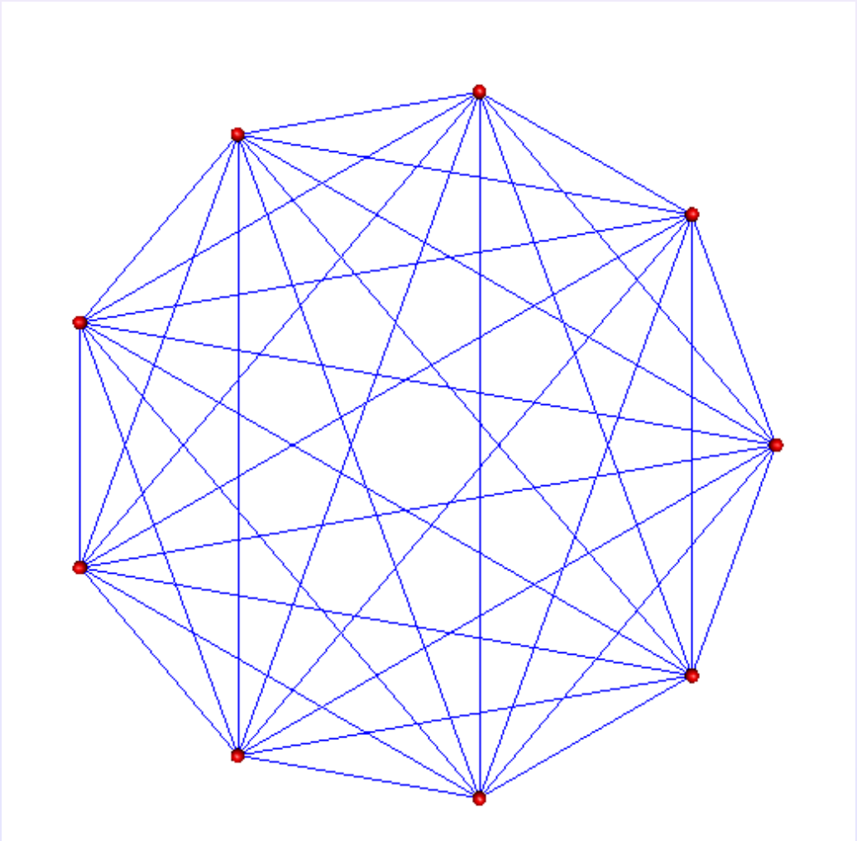
The Cluster tara at UMBC

- The distributed-memory cluster tara contains 82 compute nodes (arranged in two stacks of 'pizza boxes' in the racks), each with two quad-core Intel Nehalem X5550 processors (2.66 GHz, 8192 kB cache) and 24 GB of local memory.
- All nodes on tara are connected by a high-performance quad-data rate InfiniBand network. Each node connects by cable (the red fiber-optic cables in the photos) to one port in an 18-port InfiniBand leaf module. The central InfiniBand switch in turn provides connections between the leaf modules through its back plane.

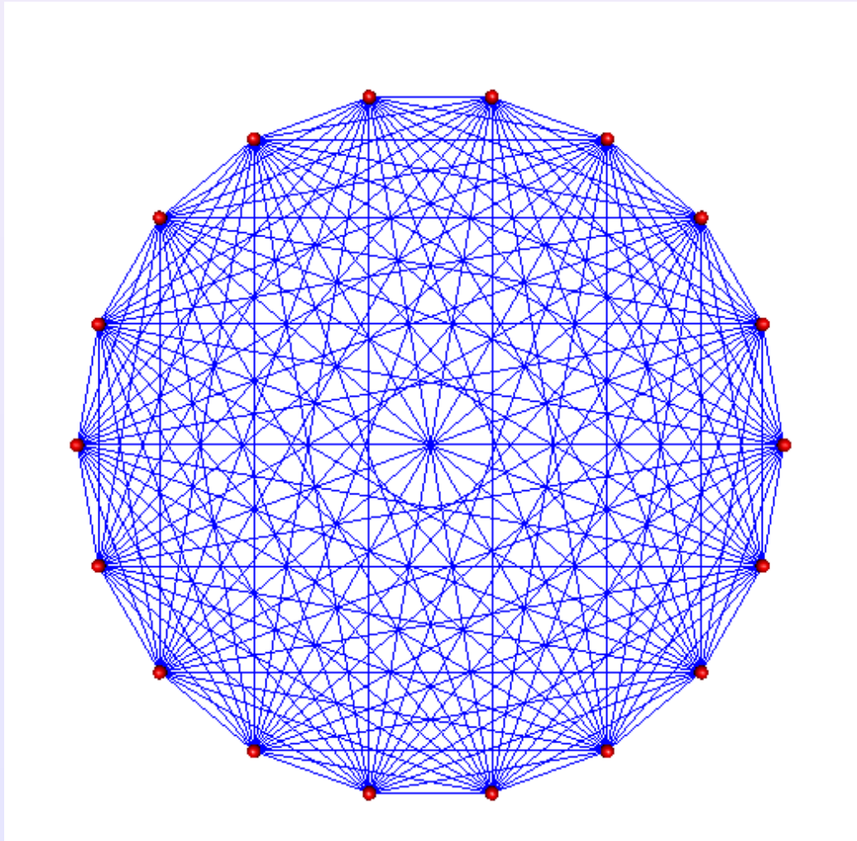


All-to-All Communications

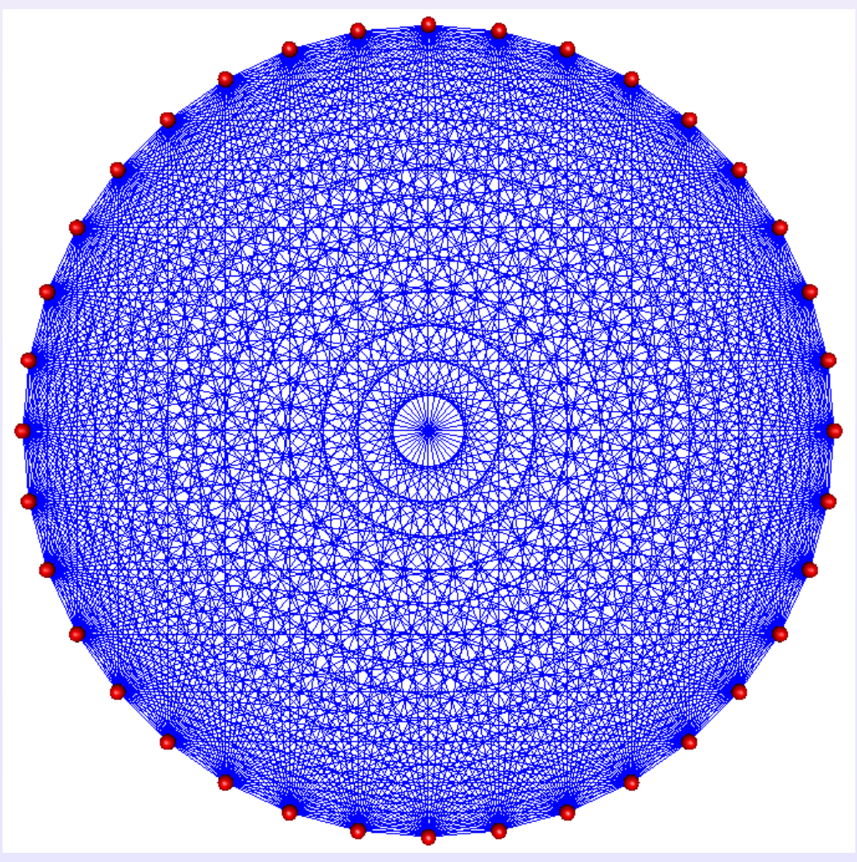
- An All-to-All communication simultaneously sends and receives data between all parallel processes in a code. Since it is eventually not possible to have physical cable connections between all possible pairs of ports in the InfiniBand switch and its leaf modules, All-to-All commands necessarily lead to contention between the required pairwise communications. The network schematics give an impression how many cables would be needed to connect $N = 9, 18, 36$ nodes, respectively.
- The MPI All-to-All communication command sends the j^{th} block of its input array from Process i to Process j and receives it into the i^{th} block of the output array on Process j. To test the InfiniBand network, we will maximize the contention by communicating the largest block sizes possible.



$N = 9$



$N = 18$



$N = 36$

Test Case with Maximum Local Memory

- A global array of n vectors, each comprising m double-precision numbers, is split onto the p parallel processes. Each local array is of length $l_n := n/p$.
  - All-to-All communication redistributes blocks of the local arrays among all process pairs, without ever assembling the global array. Contention arises due to the simultaneous nature of these pairwise communications.
  - We maximize the contention by designing the test case to have all blocks to be of same maximum size, l_n/p.
  - 8 parallel processes on each compute node maximizes (i) contention on each node for the All-to-All communications among its local processes and (ii) contention when all local processes access the InfiniBand cable at the same time.
- We consider $N = 1, 3, 9, 18, 36$ nodes.
  - $N = 3, 9$ are adjacent nodes in the leaf module, and $N = 18, 36$ are all nodes connected by one and two leaf modules, respectively.
  - Length of global array $n = 1{,}492{,}992$, so $l_n = n/p$ and $l_n/p = n/p^2$ are integers for all possible p under consideration.
  - $m\,(l_n/p) =$ number of double-precision numbers communicated between process pairs in the All-to-All communications.
  - In order to keep the block size in the All-to-All communications as large as possible, the vector length m is designed to increase with increasing p.

Nodes N	1	3	9	18	36
Processes p	8	24	72	144	288
$m = 512\,N$	512	1,536	4,608	9,216	18,432
Length n of global array of m vectors and their memory in GB:					
Length n	1,492,992	1,492,992	1,492,992	1,492,992	1,492,992
Memory	6 GB	17 GB	51 GB	103 GB	205 GB
Length $l_n = n/p$ of local arrays of m vectors and their memory in MB:					
Length l_n	186,624	62,208	20,736	10,368	5,184
Memory	729 MB	729 MB	729 MB	729 MB	729 MB
Length l_n/p of block size of m vectors in All-to-All and their memory in MB:					
Length l_n/p	23,328	2,592	288	72	18
Memory	91 MB	30 MB	10 MB	5 MB	3 MB

Results and Conclusions

Observed wall clock time in seconds using p parallel processes on N nodes for four formulas for array size m (ERR indicates error encountered):

Nodes N	1	3	9	18	36
Processes p	8	24	72	144	288
$m = 512\,N$	0.60	1.64	2.09	2.28	2.30
$m = 800\,N$	1.79	3.05	3.73	5.01	6.73
$m = 810\,N$	1.80	2.83	3.30	5.54	ERR
$m = 1024\,N$	85.00	170.62	ERR	ERR	ERR

- With local memory constant and contention on the network maximized, the run times grow with the number of processes. We can conclude that this test case creates stress on the InfiniBand network and that its performance will limit the scalability of parallel algorithms that use All-to-All communications.
- Furthermore, for cases with larger memory requirement, we encounter excessive run times and eventually memory errors.

References and Acknowledgments

- For complete details, see HPCF–2013–13 at `www.umbc.edu/hpcf` > Publications
- UMBC High Performance Computing Facility (HPCF): `www.umbc.edu/hpcf`
- REU Site: `www.umbc.edu/hpcreu`, NSF, NSA, UMBC, HPCF, CIRC