Graph 500 Performance on a Distributed-Memory Cluster

UMBC REU Site: Interdisciplinary Program in High Performance Computing

Team members: Jordan B. Angel[1], Amy M. Flores[2], Justine S. Heritage[3], Nathaniel C. Wardrip[4]

Graduate assistant: Andrew M. Raim[5], Faculty mentor: Dr. Matthias K. Gobbert[5]

Client: Dr. Richard C. Murphy[6], Dr. David J. Mountain[7]

[1]East Tennessee State University [2]Grinnell College [3]Dickinson College [4]Saint Cloud State University [5]UMBC

[6]Sandia National Laboratories [7]Advanced Computing Systems Research Program

Problem

The Graph 500 benchmark evaluates a computer's performance in accessing data efficiently by measuring the number of traversed edges per second (TEPS) of large random graphs.

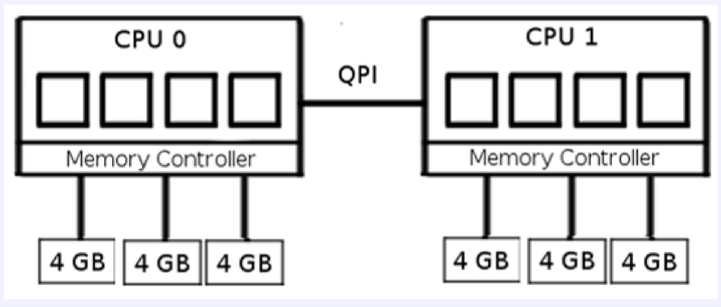
Efficient data access is important for many modern applications, more so than floating-point operations per second (FLOPS), measured by the celebrated LINPACK benchmark (TOP500).

Social networking provides examples of large graphs. Facebook, for example, reports approximately 1 billion active users, each with an average of 150 friends. Each user is represented by a vertex, and "friendships" are represented by edges per vertex.

Distributed-Memory Cluster Tara

The cluster tara at UMBC High Performance Computing Facility (HPCF) has 82 compute nodes with two quad-core Intel Nehalem X5550 CPUs (2.66 GHz, 8 MB cache) and 24 GB memory each.

All nodes are connected by a high performance InfiniBand (quad data rate) interconnect network to each other and the 160 TB central storage.



The Intel Nehalem CPUs on a compute node have 3 memory channels, each with one DIMM of 4 GB. The two CPUs are connected to each other via Intel's ultra-fast QuickPath Interconnect (QPI).

Graph 500 Specifications

The problem complexity of the graph in the benchmark is characterized by a scale S. This graph has 2^S vertices, and 16 edges per vertex.

A scale 31 graph has approximately 2.1 billion vertices — twice the size of Facebook — and requires 1171 GB of memory. This requires 64 nodes and is the largest possible problem on tara.

The Graph 500 ranking is based on the harmonic mean of 64 breadth-first searches through the benchmark graph starting at different vertices.

Results for Scale 31

We study the behavior of running increasingly more processes on the two quad-core CPUs across the 64 compute nodes, thus reducing the memory usage per process.

Processes per node	Memory per process (GB)	Memory used
1	17.36	72%
2	8.95	37%
4	4.75	20%
8	2.71	11%

The following table reports the overall run time in HH:MM:SS and the rate of edge traversal in GTEPS (billions of traversed edges per second).

Processes per node	Run time HH:MM:SS	GTEPS
1	01:06:39	0.605
2	00:45:04	0.911
4	00:42:55	0.946
8	01:03:48	0.621

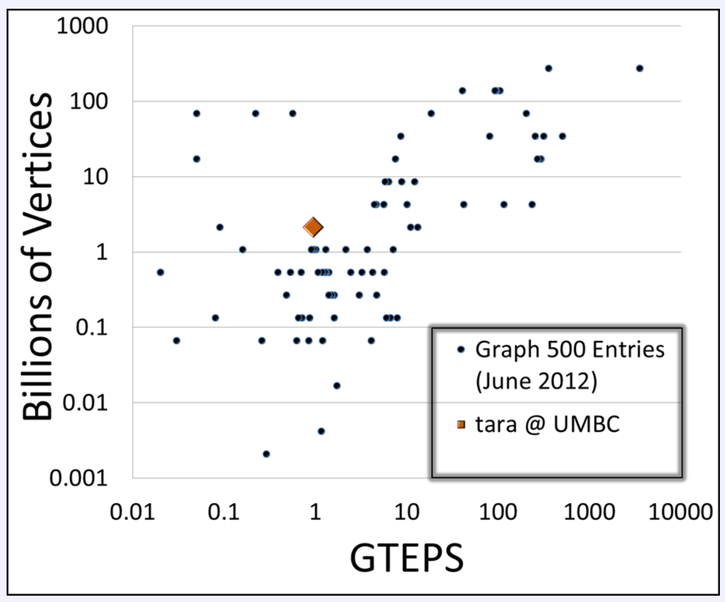
The results bring out that two and four processes per node are the best compromise between CPU usage and memory access, as demonstrated by the GTEPS being optimal for four processes.

Conclusions

For one process per node, the problem requires more than half of each node's memory; therefore, the process has to access some of the memory through the other CPU on the node, resulting in nonuniform memory access.

For eight processes per node, the memory channels have become saturated with traffic, resulting in the processes competing to read memory.

The result of the implementation of the Graph 500 reference code on tara would place UMBC HPCF at rank 58, based on the 0.946 GTEPS, in the June 2012 list.



We will officially submit our results to the November 2012 Graph 500 ranking.

References

- Background on the Graph 500 benchmark: `www.graph500.org`
- For results of all studies, see the technical report HPCF–2012–11 at `www.umbc.edu/hpcf` > Publications.

Acknowledgments

- REU Site (funded by NSF and NSA): `www.umbc.edu/hpcreu`
- UMBC, HPCF, CIRC