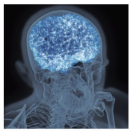
Performance Evaluation of Supercomputer Fugaku using Breadth-First Search Benchmark in Graph500

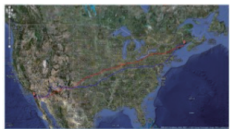
Masahiro Nakao†, Koji Ueno‡, Katsuki Fujisawa*, Yuetsu Kodama†, Mitsuhisa Sato†
(†RIKEN Center for Computational Science, ‡Fixstars Corporation, * Kyusyu University)

What is Graph500 ?

Background

- Many emerging large-scale data science applications require performance-scaling graph algorithms


Neural Network


Roadway Network

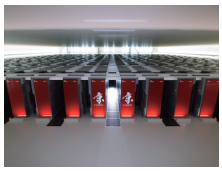
Cyber Security


SNS

- **Graph500** was started in 2010 as a competition for evaluating the performance of large-scale graph processing

Our previous research

- We tuned the performance of BFS, which is one of the Graph500 kernels, on the K computer
- It took the top spot on Graph500 a total of 10 times from 2014



- The K computer stopped operating in August 2019
- **The supercomputer Fugaku**, the successor to the K computer, took the top spot on Graph500 in June 2020

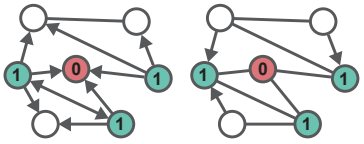
Overview of BFS algorithm

Distribution with 2D process grid[1]

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,C} \\ \vdots & \ddots & \vdots \\ A_{R,1} & \cdots & A_{R,C} \end{pmatrix}$$

Divide the adjacency matrix into a 2D process grid to reduce communication partners. Although omitted in the figure, Yoo' s distribution technique is applied to delete a part of the communication[2].

Hybrid-BFS with optimized search direction[3]



Top-down Bottom-up

In the middle of BFS, the number of vertices being searched increases explosively. To reduce it, Hybrid-BFS switches from top-down at the beginning and end of BFS and bottom-up at the middle of BFS.

Proposed techniques[4]

Bitmap-based CSR(BCSR) for adjacency matrix

We have developed a BCSR that uses bitmaps that can retrieve edge information more efficiently and with less memory than general CSR.

Edge List

source	0 0 6 7
destination	4 5 3 1

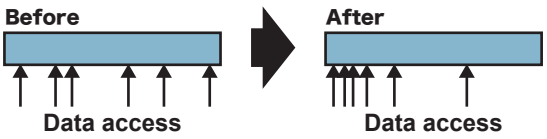
BCSR

row-starts	0 2 3 4
bitmap	1 0 0 0 0 0 1 1
offset	0 1 3
destination	4 5 3 1

- row-starts: skip vertices that has no edges
- bitmap: one bit for each vertex: represents the vertex has at least one edge or not
- offset: represents cumulative # of set bits from the beginning of bitmap to the corresponding word boarder

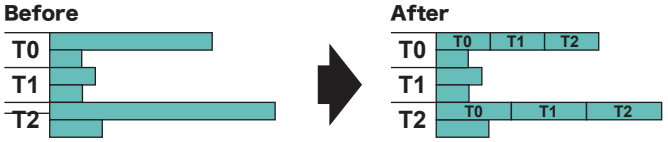
Vertex reordering

BFS requires heavy random memory accesses. To increase memory access locality, renumbering vertex ID in order of vertex degree.



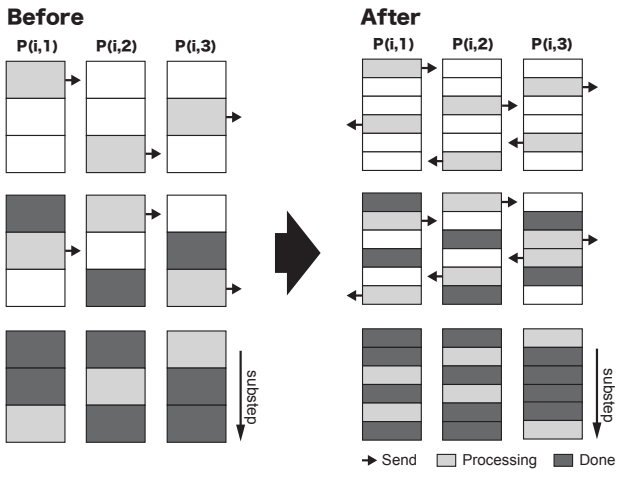
Top-Down load balancing

Since length of edge list varies for each vertex, it cause load imbalance among threads. To equalize the load, 2-step thread division is performed.



Overlapping communication and calculation

- To promote the overlap between communication and calculation, the calculation process is divided so that multiple communications can be executed at the same time.
- Furthermore, to effectively use a torus-topology network used in Supercomputer Fugaku, Blue Gene/Q, and so on, communication is performed in two directions.



Reference

[1] S. Beamer et al. ``Distributed memory breadth-first search revisited: Enabling bottom-up search.'' In 2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum, pp. 1618–1627, 2013.

[2] A. Yoo et al. ``A scalable distributed parallel breadth-first search algorithm on bluegene/l.'' In SC05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, pp. 25–25, Nov 2005.

[3] S. Beamer et al. ``Direction-optimizing breadth-first search.'' In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC12, pp. 12:1–12:10, 2012.

[4] Ueno et al, ``Efficient Breadth-First Search on Massively Parallel and Distributed-Memory Machines,'' DOI 10.1007/s41019-016-0024-y, 2017