On the shortest path to visibility

Development of a highly scalable shortest-path program for the Graph500 benchmark

***Yuji Shinano, Daniel Rehfeldt, Katsuki Fujisawa, Thorsten Koch**, Zuse Institute Berlin, Kyushu University, TU Berlin*

In Short

- The goal of this project is to develop a highly scalable shortest-path program for the Graph500 benchmark. In his way, the project should also lay the foundation for further applications on very large-scale graphs by using HPC systems.

The Graph500 http://graph500.org is an international benchmark for HPC systems regarding graph based algorithms started at the SC2010 conference:

> Data intensive supercomputer applications are increasingly important for HPC workloads, but are ill-suited for platforms designed for 3D physics simulations. Current benchmarks and performance metrics do not provide useful information on Data-intensive supercomputer applications are an increasingly important workload, but are ill-suited for platforms designed for 3D physics simulations. Application performance cannot be improved without a meaningful benchmark. Graphs are a core part of most analytics workloads. Backed by a steering committee of over 30 international HPC experts from academia, industry, and national laboratories, this specification establishes a large-scale benchmark for these applications. It will offer a forum for the community and provide a rallying point for data-intensive supercomputing problems. This is the first serious approach to augment the Top 500 with data-intensive applications.
>
> The intent of benchmark problems ("Search" and "Shortest-Path") is to develop a compact application that has multiple analysis techniques (multiple kernels) accessing a single data structure representing a weighted, undirected graph. In addition to a kernel to construct the graph from the input tuple list, there are two additional computational kernels to operate on the graph.
>
> This benchmark includes a scalable data generator which produces edge tuples containing the start vertex and end vertex for each edge. The first kernel constructs an undirected graph in a format usable by all subsequent kernels. No subsequent modifications are permitted to benefit specific kernels. The second kernel performs a breadth-first search of the graph. The third kernel performs multiple single-source shortest-path computations on the graph. All three kernels are timed.

A graph consists of a node and an edge set, where each edge connects exactly two nodes. For example, a node may correspond to an intersection in a road network, and an edge may correspond to a road between two intersections. In the analysis of social networks such as Twitter, a node corresponds to a user, and an edge corresponds to the Twitter follower relationship between two users. Extremely large-scale graphs have recently emerged in various application fields, such as transportation, social networks, cybersecurity, and bioinformatics. The number of nodes in such graphs has grown from billions to trillions, and the number of edges from hundreds of billions to tens of trillions. Therefore, the large-scale graph analysis has attracted significant attention as a new application of the next-generation supercomputer [4]. For such an analysis one needs to design algorithms that can run efficiently in a distributed-memory environment on such huge graphs. The foundation for such algorithms is laid by core techniques such as graph traversal (as in breadth-first search), or finding shortest-paths. While a huge amount of research has been done on such fundamental algorithms in sequential or even shared-memory context, a scalable distributed-memory adaptation is still highly challenging.

The *Applied Algorithmic Intelligence Methods* department at ZIB has decades of experience with the development and implementation of high-performance graph algorithms (see, e.g., [1–3]). Katsuki Fujisawa is the head of the team which brought the K-Computer and now the Fugaku system to the top position in the Graph500 benchmark. As part of the collaboration of ZIB with Katsuki Fujisawa at Kyushu University, our team was involved and helped to achieve the results that now show the Fugaku system far ahead of its competitors. The goal of this project is to develop a highly scalable, distributed-memory software for the Graph500 Benchmark 2, which is concerned with shortest-paths. Designing an efficient distributed-memory shortest-path algorithm is certainly more involved than a breadth-first search algorithm, but we still believe that it is possible to reach the top of the Graph500 Benchmark 2 within the next year.

The Graph500 benchmark was designed to measure a computer system's performance for applications that require irregular memory and network access patterns, such as large-scale graph analysis. Following its announcement in June 2010, the Graph500 list was released in November 2010, and since it has been updated semiannually. The Graph500 benchmark measures the supercomputer's performance by executing a breadth-first search (BFS) for Benchmark 1 ("Search") and performing a single source shortest-paths (SSSP) for Benchmark 2 ("Shortest-Path") in terms of traversed edges per second (TEPS). The benchmark performs the following steps:

1. Generate the edge list.
2. Construct a graph from the edge list (**timed**, Kernel 1).
3. Randomly sample 64 unique search keys with degree at least one, not counting self-loops.
4. For each search key:
   a) Compute the parent array(**timed**, Kernel 2).
   b) Validate that the parent array is a correct BFS search tree for the given search tree.
5. For each search key:
   a) Compute the parent array and the distance array(**timed**, Kernel 3).
   b) Validate hat the parent array/distance vector is a correct SSSP search tree with shortest-paths for the give search tree.
6. Compute and output performance information.

Here, Kernel 1 (graph construction), Kernel 2 (BFS), and Kernel 3 (SSSP) can be provided by each participant. This gives the freedom to develop programs specialized to the Supercomputer used. Only the sections marked as **timed** are included in the performance information.

From 2014 to 2021, Katsuki Fujisawa's team held the top position at the Graph500 Benchmark 1 ("BFS") by utilizing systems such as K [5] and Fugaku [6,7]. We tuned the BFS package to Lisa and attended the Graph500 competition in June 2021. Lisa is now on the 7th position in the list. This fact shows that the benchmark is not only about hardware, but also that the employed Kernel programs are very important. Unfortunately, we currently do not have a Kernel 2 program (SSSP) available. However, we hope that this project will repeat the success story for the BFS Graph500 competition also for its SSSP counterpart.

WWW

https://www.zib.de/members/shinano

More Information

[1] T. Koch, D. Rehfeldt, Y. Shinano, *An exact high performance solver for Steiner tree problems in graphs and related problems* in Proceedings of HPSC 2018, Springer (2020), doi:978-3-030-55240-4_10.

[2] Y. Shinano, D. Rehfeldt, T. Koch, *Building Optimal Steiner Trees on Supercomputers by Using up to 43,000 Cores* Proceedings of CPAIOR 2019, Springer (2019), doi:10.1007/978-3-030-19212-9_35.

[3] G. Gamrath, T. Koch, S.J. Maher, D. Rehfeldt, Y. Shinano, *SCIP-Jack – A solver for STP and variants with parallelization extensions*, Mathematical Programming Computation, 9:2 (2017), doi:10.1007/s12532-016-0114-x.

[4] Katsuki Fujisawa, Toyotaro Suzumura, Hitoshi Sato, Koji Ueno, Satoshi Imamura, Ryo Mizote, Akira Tanaka, Nozomi Hata, Toshio Endo, Advanced Computing and Optimization Infrastructure for Extremely Large-Scale Graphs on Post-peta-scale Supercomputers, Advanced Software Technologies for Post-Peta Scale Computing, Springer, pp 207-226, 2019.

[5] Koji Ueno, Toyotaro Suzumura, Naoya Maruyama, Katsuki Fujisawa and Satoshi Matsuoka, Efficient Breadth-First Search on Massively Parallel and Distributed Memory Machines, Data Science and Engineering, Springer, Volume 2, Issue 1, pp. 22–35, 2017.

[6] Masahiro Nakao, Koji Ueno, Katsuki Fujisawa, Yuetsu Kodama and Mitsuhisa Sato, Performance Evaluation of Supercomputer Fugaku using Breadth-First Search Benchmark in Graph500, poster, the IEEE Cluster 2020, September 14-17, 2020, Kobe, Japan.

[7] Masahiro Nakao, Koji Ueno, Katsuki Fujisawa, Yuetsu Kodama and Mitsuhisa Sato, Performance of the Supercomputer Fugaku for Breadth-First Search in Graph500 Benchmark, high-performance Computing, Lecture Notes in Computer Science, vol 12728, Springer, pp.372–390, 2021.

Project Partners

Kyushu University, RIKEN-RCCS

Funding

BMBF Research Campus MODAL