A Low Communication Overhead Breadth-First Search based on Global bitmap

Ziwei Peng†[1, 2], Yunfei Du[2, 3], Zhiguang Cheng[1, 2, 3]
([1] College of Computer, National University of Defense Technology, Changsha 410073, China)
([1] National Supercomputer Center in Guangzhou, Guangzhou 510006, China)
([1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China)
†E-mail: peng_ziwei@foxmail.com

Abstract

Breadth-First Search (BFS) is the underlying kernel algorithm for many graph applications such as social networks, medical informatics, transport systems, etc. Therefore, it has been absorbed as a core of Graph500, used to evaluate the capability of supercomputers in terms of big data processing. In this paper, we introduce into a global bitmap which is used to accelerate two approaches: the top-down and bottom-up. Specifically, the new top-down approach uses the global bitmap to indicate whether the vertices are visited or not, while the new bottom-up approach changes the frontier queue to the global bitmap to indicate whether the vertices are on the frontier. With the help of the global bitmap, the total number of communication messages produced by the BFS will be reduced significantly, and consequentially the BFS is accelerated. Meanwhile, our algorithm is optimized for storage on Knights Landing (KNL). We evaluate our proposal on both the KNL platform and the Tianhe-2 supercomputer. Experimental results demonstrate that the communication was time reduced to roughly 1/4 of the original. We obtain speedups of 2.2-3.1 compared to the top-down approach.

Keywords

Graph500, Breadth-First Search, Global Bitmap, Hybrid Approach.

Introduction

As a basic graph algorithm, breadth-first search (BFS) is a core component of m-any algorithms and has been widely used in many fields. Now the BFS attracted more and more attention, and a large number of literatures involving the optimization of BFS have been published. Researchers have explored varied methods to accelerate the BFS on different architectures. Beamber et al.[1] proposed a novel optimization on BFS which combines the top-down approach with the bottom-up approach. Agarwal et al.[2] introduced into the bitmap data structure to represent the vertices accessed in BFS, increasing the locality of the data. In this poster, we combine two methods and present a hybrid BFS algorithm based on a global bitmap which is used to indicate whether a vertex has been visited and whether a vertex is on the frontier. The bitmap helps to optimize both the top-down and bottom-up approaches by reducing the amount of communication messages significantly. We evaluate our proposal on both the KNL platform and the Tianhe-2 supercomputer.

Methods

- **Global visited bitmap in Top-down approach:** For the bottom-up approach, we can then utilize the low communication overhead top-down approach with global visited bitmap to cut down the communication time. Through this method, we can judge vertices based on whether they have been visited in the local node and reduce the communication overhead. When the process scans the vertices on the boundaries, it locally determines if the neighbors of those vertices have been accessed. If the neighbor of the vertex has been visited, the process doesn't need to send a message to the owner of the neighbor. Otherwise, the processor need to send a message containing the vertex's id number and the neighbor node's id number to the owner of the neighbor.
- **Global frontier bitmap in Bottom-up approach:** For the bottom-up approach, the distributed BFS judges whether the vertex is accessed locally, then sends its edge to its neighbors' processors. The processor will judge whether the neighbor vertex is on the frontier. If we can calculate whether the vertex's neighbor is in local processor rather than the target processor, we don't need to send the message to the target processor. In the same manner, the target processor will not need to send back the message to the source processor. So, we present one kind of bottom-up BFS based on global frontier bitmap to reduce the communication overhead. Before all processors synchronize their frontier bitmap globally, the bottom-up approach needs to convert the frontier queue to the frontier bitmap. This means that it will increase the computing overhead by a certain amount. However, this increase is negligible relative to the communication overhead reduction
- **Switch algorithms:** The algorithm switch is based on the current size of the frontier queue, and it is used to determining whether it is needed to switch algorithms in the current level of BFS, as Fig. 1 shows.

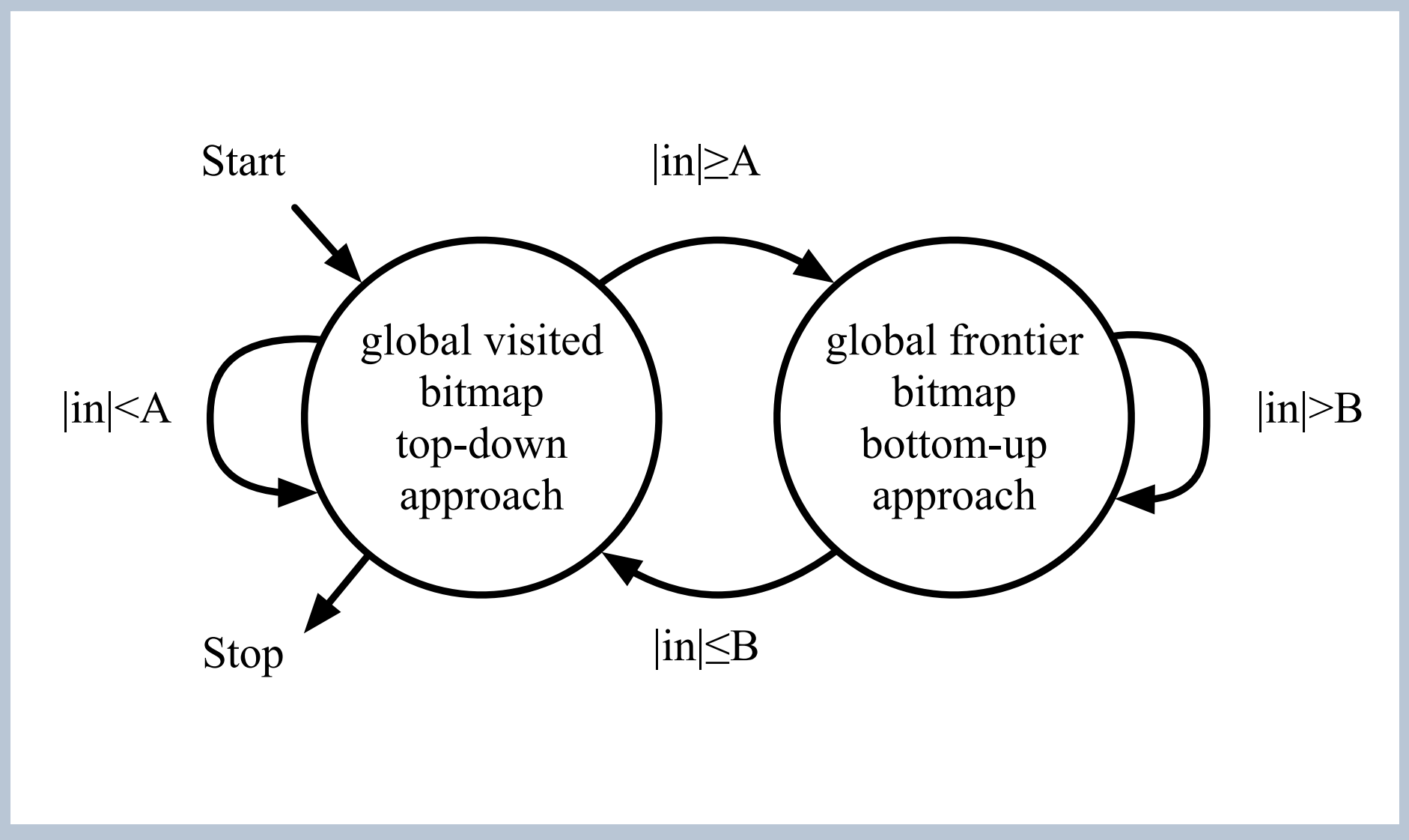
Start
|in|≥A
global visited bitmap top-down approach
global frontier bitmap bottom-up approach
|in|<A
|in|>B
Stop
|in|≤B

Fig. 1. switch algorithms

Experimental platform

We collected performance results on the Tianhe-2 system and KNL platform.

- **Tianhe-2:** Tianhe-2 is equipped with 17920 nodes, each containing two 12-core Xeon E5 CPU. The front-end system consisted of 4096 Galaxy FT-1500 CPUs. Tianhe-2 has a speed of 33.9PFlops and a peak performance of 54.9PFlops. Its abundant computing resources and fast computing speed make it the best accelerator for the research project.
- **KNL platform:** The 2nd generation Intel Xeon Phi™ processors (code-named "Knights Landing") are specialized computing platforms capable of delivering better performance for some applications than general-purpose CPUs such as Intel Xeon products.

Experimental Results

- **Time breakdown analysis:** For the five methods in the Fig 2, the barrier time is about the same. The barrier operation is mainly due to the unbalanced load between the processors. As you can see in, our approach reduces the amount of communication overhead. The global visited bitmap method reduces the communication overhead in the top-down approach and the global frontier bitmap method reduces the communication overhead in the bottom-up approach. We also notice that the "allreduce time" is very short. This implies that our method is very effective. The hybrid direction optimizing approach's communication time is about a ¼ of the direction-optimizing approach and one in five of the original version. So our approach is quite effective in reducing communication overhead.

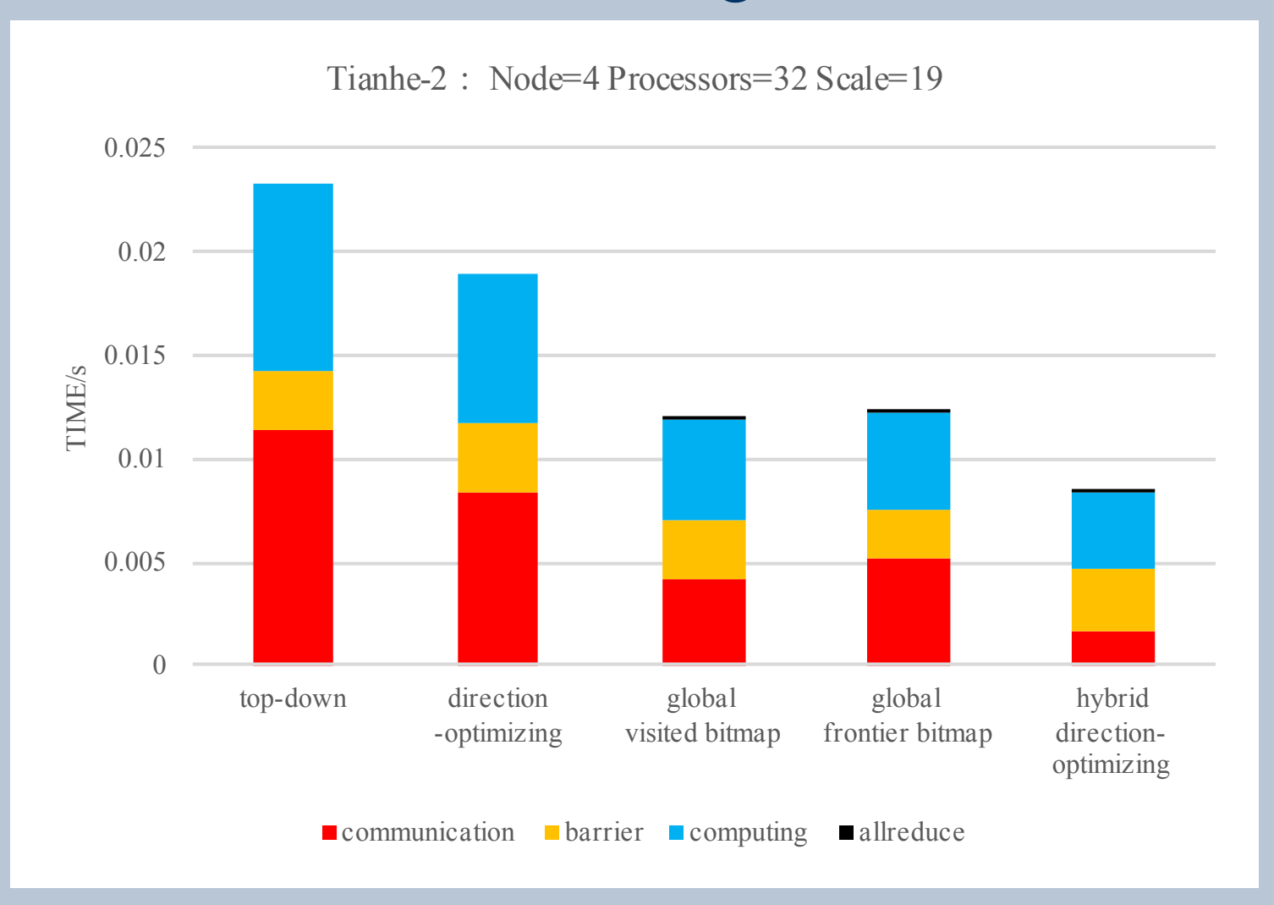

Fig. 2. time breakdown on Tianhe-2

Ps:
The barrier time is used to synchronize all processors during the end of a BFS level.
The allreduce time is used to allreduce the bitmap during all processors.

- **Scalability in Tianhe-2:** we measured the weak scalability of the proposed BFS algorithm on fixed problem size per node (each node has 2^{16} vertices) and present the results in Fig. 3. we see that the hybrid design is about 2.2-3.1 times faster than the official version. And we measured the strong scalability of the proposed BFS algorithm in Tianhe-2 system, presenting the results in Fig. 4. The hybrid method can reduce the communication overhead both in the top-down process and the bottom-up process, making 1.9-3.0 times faster than the official version.

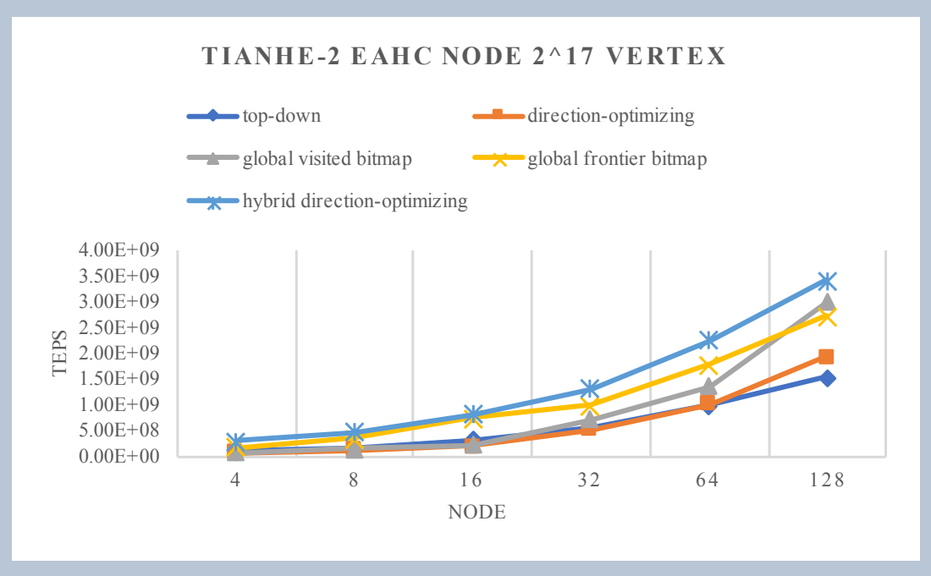

Fig. 3. scaling" results on Tianhe-2

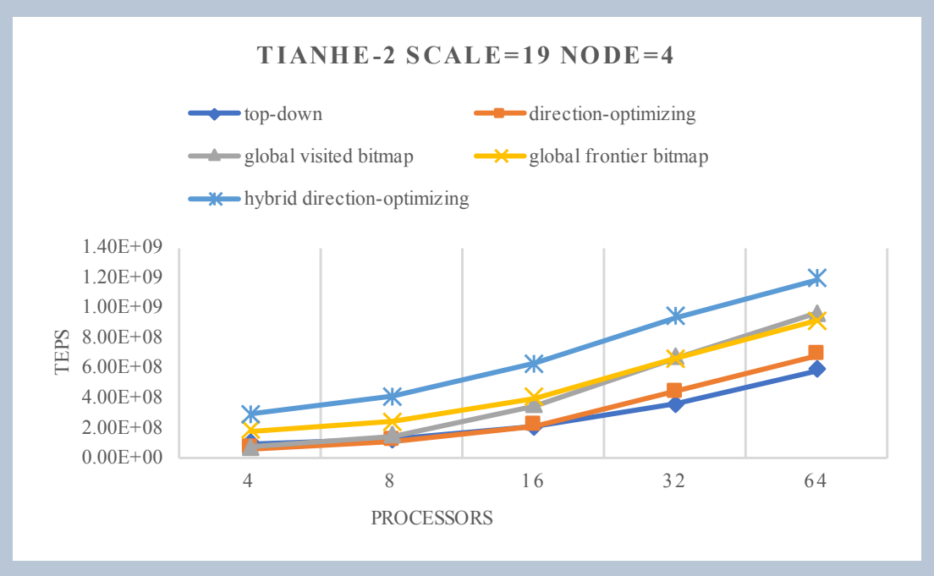

Fig. 4. "strong scaling" results on Tianhe-2

- **Scalability in KNL:** We measured the scalability of the proposed BFS algorithm in the KNL system. "Weak scaling" results on KNL are presented in Fig. 5. Meanwhile as showed in Fig. 6, our algorithm has a good processor scalability.

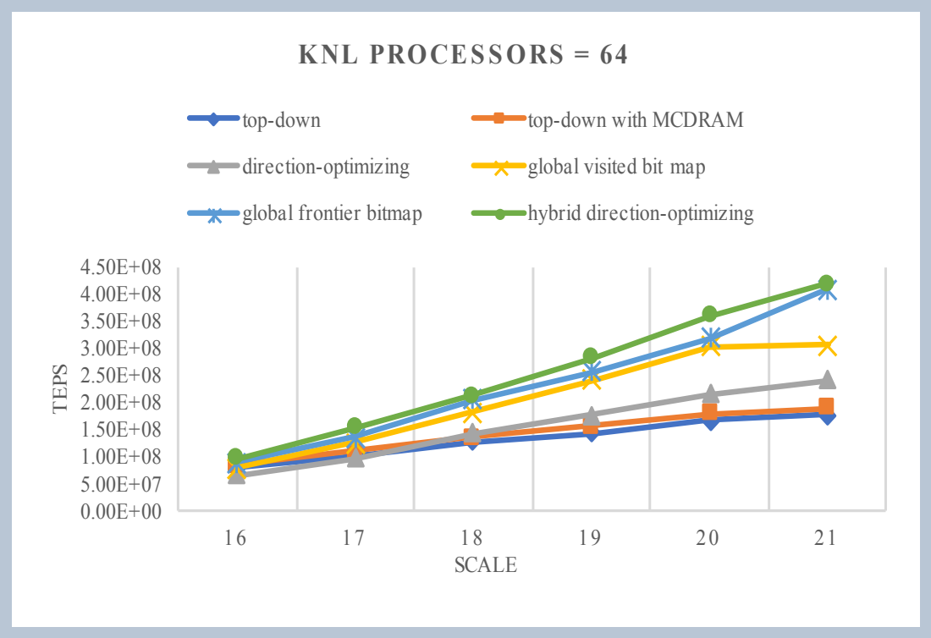

Fig. 5. scaling" results on KNL

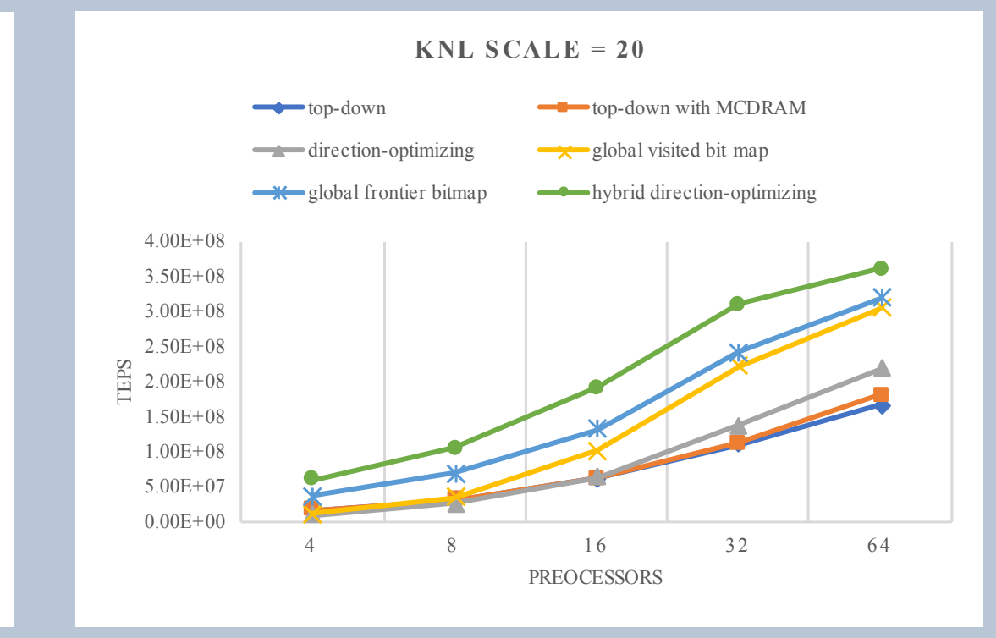

Fig. 6. "strong scaling" results on KNL

Conclusion And Future Work

In this poster, we have cut the communication and calculation times in the basis of the direction-optimizing BFS. We present two global bitmap approaches to accelerate the BFS: a top-down approach with global visited bitmap and a bottom-up approach with global frontier bitmap. We used a hybrid approach to combine the advantages of both. Additionally, we optimized the computation of the bottom-up approach as well as the storage of the KNL coprocessor. We performed experiments on the Tianhe-2 and KNL systems with good results.

Listed below are optimizations that we intend to explore in future work:

Distributed BFS with 2D partitioning. In future work, we will use the 2D decomposition to split the data among the nodes. Then analyze the communication optimization on the basis of the 2D version.

Exploiting Single Instruction Multiple Data (SIMD) in KNL. The basic idea of SIMD optimization is to scan the vertices' neighbors simultaneously. The problem to be solved is that there may be discontinuities in the visits of the vertices' neighbors. Optimization of the BFS in the heterogeneous system which uses the KNL system as a set of coprocessors rather than as a CPU.

References

[1] S. Beamer, K. Asanovic, and D. A. Patterson, "Searching for a Parent Instead of Fighting Over Children: A Fast Breadth-First Search Implementation for Graph500," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2011-117, Nov 2011.
[2] V. Agarwal, F. Petrini, D. Pasetto and D. A. Bader, "Scalable Graph Exploration on Multicore Processors," *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, New Orleans, LA, 2010, pp. 1-11.