Parallel algorithm research of graph search and depth learning based on data mining

Liping Wang

Pingxiang University, Pingxiang Jiangxi, 337055

Keywords: Data mining; graph search; width-first search; depth learning; depth belief network

Abstract. At present, information technology presents exponential growth characteristics, it have entered the era of large data. Data is a strategic resource as important as self-heating resources and human resources, which implied huge economic value. How to effectively organize and deal with large data will play a huge role in the socio-economic development. The graph search and depth learning algorithms play a more and more important role in the processing of large data because of their strong ability of network analysis and feature recognition and classification. In this paper, we propose a parallel optimization method based on locality principle, synchronization cost reduction and load balancing to solve the problem of width-first search. Finally, this paper combines all the methods together, and proposes a width-first search method using heuristic search. The experimental results showed that the width-first search algorithm with parallel optimization has good acceleration effect.

Introduction

Large data refers to a collection of data that can not be managed, acquired, processed, and serviced in a short time by traditional hardware and software tools and computer technology [1-3]. Large data types included unstructured data, structured data and semi-structured data. Unstructured data presents a significant increase in the characteristics of the end of 2015, Internet applications in the proportion of unstructured data accounts for more than 75% of the entire data [4]. At the same time, because of the data explicit or implicit network exists, making the complex relationship between data.

Large data age, massive data is rich in information, but the observation of a variety of complex systems to large data, directly reflects the isolated data and scattered connections, but these reflect the relationship between the link is integrated by network [5, 6]. Therefore, the scientific research of large data is essentially the problem of network science. For example, genetic data constitute a network of genes, and Web data reflect social networks, brain science experimental data form neural networks. Many parameters and properties in the network, such as the number of medians, the number of cores, the average path length, the aggregation coefficient and the degree distribution, all have the potential to characterize the power behind the large data network [7].

With the increase of data size, how to improve the execution performance of program and reduce the power consumption has been a hotspot in the research of the background of large data and depth study. In this paper, based on the analysis of the width-first search algorithm, the parallel optimization of the program is carried out on the CPU and FPGA platforms. In addition, for the deep learning algorithm, the hardware development tool chain of the depth learning algorithm is used, the key code is designed in parallel using assembly language, and the corresponding simulator is designed for performance analysis and correctness verification.

1. Graph search parallel optimization

The breadth first search algorithm, also known as breadth first search algorithm, referred to as BFS, is the basis of graph search algorithm [8]. In short, BFS starts at the root node, traverses the tree nodes along the tree width, and terminates if all nodes reachable from the root node are found. This algorithm is also the prototype of a number of important graph algorithms. Dijkstra single source shortest path and Prim minimum spanning tree algorithm are used and width-first search algorithm similar to the idea.

As shown in Figure 1, the root node number is three. From the root node, it traverses all the neighboring nodes 0, 2, and 4 to obtain the hierarchical relationship (the father node of each node) of each node, and then expand the new nodes 0, 2, 4 respectively , The width-first search algorithm terminates until all nodes reachable from the root node.

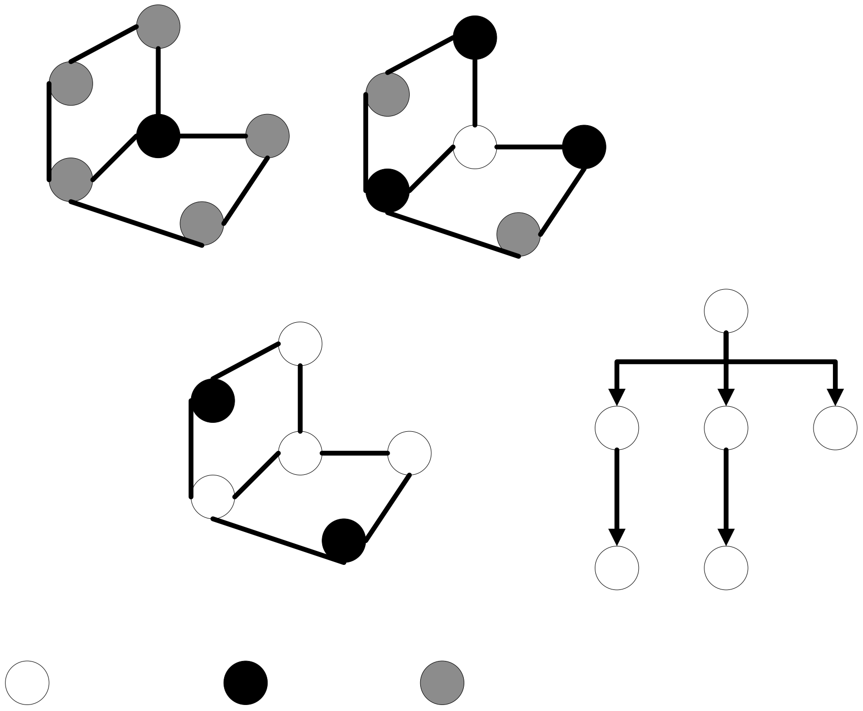

Figure 1. Width-first search process and search tree

The core idea of the width-first search algorithm can be summarized as follows: According to the current layer node, the next layer node is expanded. After the current layer node expands, the current layer and the next node queue are interchanged, and the process is iterated until all the nodes are expanded.

2. Design of width-first search based on FPGA

As shown in Figure 2 for the four PE algorithm accelerator structure, algorithm accelerator contains memory, interface unit, processing unit array and interconnect network structure. The structure of the graph to be processed is stored in memory and is used to hold the results of the width-first search algorithm. The graph structure information is stored in an adjacent data structure CSR.

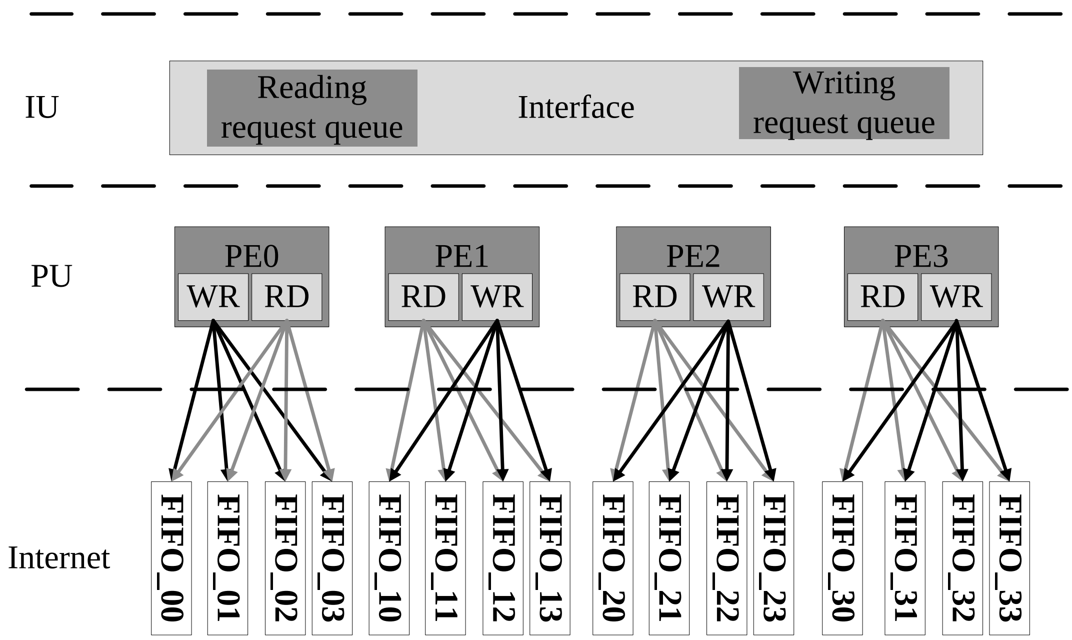

Figure 2. Width-first search algorithm accelerator

The other memory access request signal continues to be pulled high. When the previous access request is processed, the interface unit module polls the other PE's access request and releases the access right of the access channel. Here each access channel port width is 128bit, that is, each access memory can get 128bit per clock cycle of valid data.

3. Experiments and results

3.1 Load balancing

Know the current level of the node queue CQ, find the next level of the node queue NQ, then the next level of the node queue in the father node in the CQ. This method leads to a lot of computation (memory access operation) are used in the next layer to determine whether the node has a parent node. We know that the graph with N nodes has at least N-1 edges, so reducing the computational complexity has a great effect on the improvement of program performance.

Table 1. The Performance report On BFS from Top to Bottom

Layer Count	Current Layer Counts	Edge inspections	Success Counts	Success Rate
0	1	32	32	1
1	32	996	880	0.968885
2	880	30587	30442	0.967532
3	30442	981533	925881	0.941054
4	925881	30105780	13540041	0.452224
5	13540041	429288441	2817845	0.006477
6	2817845	87795505	0	0

As shown in Table 1, the test set is Random undirected graph, factor = 16, scale = 24. For the fifth-level node, it is necessary to determine whether the node connected to it has been extended (the father node has been recorded). A lot of computation and access operations are used to determine whether the point adjacent to the layer node has been expanded. At the end of the test, only 0.6% of these operations succeeded. That is, 99.4% of the neighboring nodes have been extended. These operations are invalid, but increase the cost of program execution.

3.2 Performance comparison

In this paper, the random undirected graph on the FPGA was test, the test results were shown in Figure 3, the figure in the average node degree of 16. We can see that with the increase of the scale of the graph, the performance of the system is basically the same. By comparison, it can be seen that the 8PE design achieves good performance relative to the 4PE design because the higher the number of PEs, the higher the parallelism of the algorithm and the higher the utilization rate of the bandwidth.

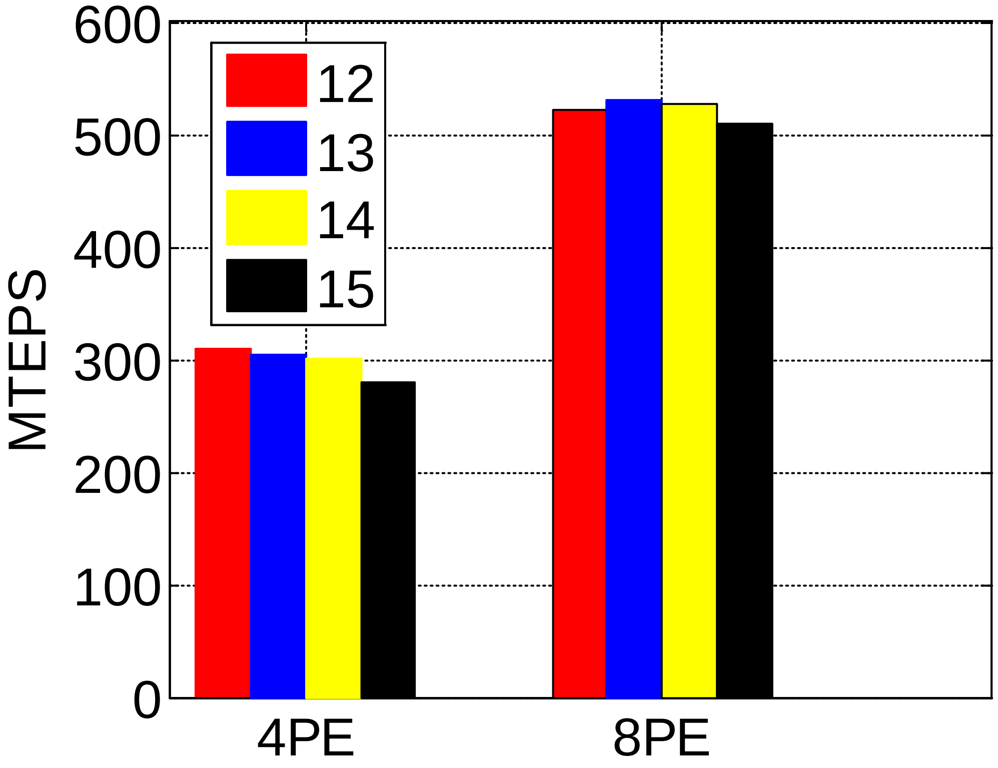

Figure 3. Relationship between test performance and PE scale

The width-first search algorithm is a hotspot of graph algorithm. In this chapter, the software algorithm is improved to adapt to the hardware design, and the whole computing system is realized based on FPGA. The realization of the process, the use of the message transmission method to achieve the search process layer synchronization, and proposed a variety of optimization methods such as fine-grained parallel, functional segment merging, access to the queue and access and memory and so on.

Summary

In this paper, based on the analysis of the top-down width-first search algorithm based on layer synchronization, a multi-PE is set up in the FPGA to implement the parallel optimization based on the characteristics of the parallel execution thread. In this paper, several optimization methods such as fine-grained parallelism, merging of function segments, memory queuing and access integration are proposed. The experimental results show that the proposed scheme has good scalability and good acceleration effect. In this paper, based on the analysis of the width-first search algorithm, the parallel optimization of the program is carried out on the CPU and FPGA platforms. In addition, for the deep learning algorithm, the hardware development tool chain of the depth learning algorithm is used, the key code is designed in parallel using assembly language, and the corresponding simulator is designed for performance analysis and correctness verification.

Acknowledgement

This work was supported by Science and Technology Research Project in Department of Education, Jiangxi Province, 2015 (GJJ151274), Jiangxi Provincial Intellectual Property Soft Science Research Project (ZR201610), Humanities and Social Sciences Research Project of Jiangxi Province, 2015 (TQ1516) , Art Science Planning Project of Jiangxi Province (YG2014255, YG2015189) and Pingxiang Science and Technology Support Program, 2015: Research and Development of Animation Rendering Service Platform in the Central Region Based on Cloud Computing.

.

References

[1] Rajani, Nazneen, Kate McArdle, and Inderjit S. Dhillon. "Parallel k nearest neighbor graph construction using tree-based data structures." 1st High Performance Graph Mining workshop, Sydney, 10 August 2015. 2015.

[2] Holder, Lawrence B., et al. "Current and Future Challenges in Mining Large Networks: Report on the Second SDM Workshop on Mining Networks and Graphs." ACM SIGKDD Explorations Newsletter 18.1 (2016): 39-45.

[3] M. Sadek, A. Tarighat, A. H. Sayed, “A Leakage-based Precoding Scheme for Downlink multi-user MIMO Channels” , IEEE Transactions on Wireless Communications, vol. 26, no.8, pp.1505-1515, 2008.

[4] Leskovec, Jure, and Rok Sosič. "SNAP: A General-Purpose Network Analysis and Graph-Mining Library." ACM Transactions on Intelligent Systems and Technology (TIST) 8.1 (2016): 1.

[5] Bandyopadhyay, Sanghamitra, et al. "Guest Editorial: special issue on advances in multiobjective evolutionary algorithms for data mining." IEEE Transactions on Evolutionary Computation 18.1 (2014): 1-3.

[6] Alcala-Fdez, Jesus, et al. "Comparison of KEEL versus open source Data Mining tools: Knime and Weka software." (2016).

[7] M. Sadek, A. Tarighat, A. H. Sayed, “Active Antenna Selection in multi-user MIMO Communications,” IEEE Transactions on Signal Processing, vol. 55, no. 4, pp. 1498-1510, 2007.

[8] Akoglu, Leman, Hanghang Tong, and Danai Koutra. "Graph based anomaly detection and description: a survey." Data Mining and Knowledge Discovery29.3 (2015): 626-688.