

Artificial intelligence becomes a new kinetic energy to promote supercomputing

Fei Mo

Chengdu Foreign School, Chengdu, China

feimomf@gmail.com

Abstract. Big data not only provides an increasingly rich set of training data for artificial intelligence, but also puts higher demands on the computing power of computer systems. In recent years, China's supercomputers have been at the forefront of the world, providing strong computing platform support for the large-scale application of big data and artificial intelligence. At present, high-performance computing platforms represented by supercomputers mostly use heterogeneous parallel computing systems composed of CPU+accelerators, with a large number of computing cores that can provide powerful computing power for artificial intelligence and big data applications.

Keywords: supercomputing; Efficient computing; Artificial intelligence; Heterogeneous systems.

1. Introduction

As is well known, supercomputing (also known as high-performance computing, abbreviated as "supercomputing") is a manifestation of a country's comprehensive national strength and one of the key technologies supporting the sustainable development of national strength. It plays an important strategic role in national defense and security, high-tech development, and national economic construction.

In recent years, with the vigorous development of artificial intelligence, big data, and cloud computing technology, the application scope of supercomputing has been continuously expanding. For example, human perception of the world is increasing, and activities in different fields have left behind data trajectories, creating a demand for multidimensional and multi-disciplinary data comprehensive correlation analysis. Cloud computing has made computing facilities readily available, big data has driven artificial intelligence, and GPUs have made computing power a huge leap forward, making it possible to make a leap in analytical capabilities.

At the same time as the artificial intelligence era has given rise to new computing needs, massive data analysis and communication are also challenging traditional supercomputing system architecture and design. It can be said that the information processing needs of the artificial intelligence era have also brought new opportunities for innovation and transformation to the design of the intelligent supercomputing platform architecture.

According to data, 40% of global AI algorithm papers in 2018 were written by Chinese people. On July 20, 2017, the State Council released China's artificial intelligence strategy, which has become a national strategy. And China is also known for having a huge artificial intelligence market. Against this backdrop, what changes will the combination of supercomputing and artificial intelligence, known as the "national treasure", bring to our economy and society?

2. Early Development of Supercomputers

At the beginning of the birth of electronic computers, due to their high cost, only large scientific research institutions or commercial companies had the conditions for use. The first electronic computer, ENIAC, in its usual sense, was a vacuum tube based computing system designed with funding from the US military's ballistic research laboratory, which could automatically perform multiplication and accumulation. Although the original purpose of ENIAC was to calculate the ballistic table of artillery systems, J. von Neumann and others used this system for groundbreaking scientific calculations such as hydrogen bomb development and numerical weather simulation.

With the gradual development of integrated circuit technology in the 1950s and 1960s, the design and manufacturing of processors gradually became possible. Supercomputers also saw significant development during this period. Due to the embryonic development of processor hardware and computing software at that time, without many rules and compatibility constraints, S. Cray, known as the "father of supercomputers", proposed many genius ideas in the early stages, such as the collaborative design of the central processing unit (CPU) and auxiliary processing unit (PPU) in the supercomputer CDC6600 system, And the design of a vector processor specifically designed for large-scale array operations in scientific computing in Cray-1 computers. These design concepts are still being used, integrated, and redeveloped in different computing systems to this day.

From the birth of the CDC6600 in 1964 to almost the entire 1970s, supercomputers continued to use similar structures, where the entire system was supported by only one or several powerful processors, each with complex unit designs and functions inside.

It was not until the 1980s that another path for the development of supercomputers began to emerge. At the same time as the development of complex vector processors in supercomputers, microprocessors such as Intel's early 4004 and 8008 and personal computers are also undergoing rapid changes thanks to integrated circuit technology. Compared to high cost and low universality vector processors, although the performance of a single microprocessor cannot be compared, integrating a large number of low-cost, standardized microprocessors through the network can achieve higher computational performance. In 1981, researchers at the California Institute of Technology used 64 Intel 8086 processors and built the Cosmic Cube system through six dimensional hypercube interconnection. Afterwards, large-scale parallel processors integrating hundreds or thousands of processors, such as Intel's iPSC system, the Connection Machine system developed by MIT, and the wind tunnel simulator developed by Fujitsu in Japan, began to emerge, providing higher computational performance than the original vector processors.

During this process, there are still deep doubts in the field of scientific computing about whether it can handle such a large-scale parallel machine. According to the analysis of Amdal's law, if 10% of a program cannot be parallelized, even with 1000 processors, the remaining 90% of the time can be optimized to be negligible, resulting in a performance acceleration ratio (the ratio of time consumed by the same task running in a single processor system and a parallel processor system) of up to 10 times. It is questionable whether hundreds or even thousands of processors can achieve corresponding performance benefits. At the end of 1985, A. Karp challenged in an email that if anyone could achieve more than 200 times the acceleration of a scientific computing software on a parallel machine, he would privately pay a prize of \$100. In 1987, renowned computer researcher Gordon Bell officially established the "Gordon Bell Award" to reward and track the top application achievements on supercomputers. The winner of that year achieved a performance improvement of 400 to 600 times on the nCube system composed of 1024 nodes.

Over the past 30 years, the evaluation and awarding of over 20 editions of the "Gordon Bell Award" have witnessed the rapid development of supercomputers and high-performance computing applications relying on supercomputers, and have also become the highest award in the field of international supercomputing applications. From the application performance of about one billion floating-point operations per second in the past few years to the application performance of quantum circuit simulation exceeding 40 billion floating-point operations per second in 2021, significant progress has been made in both the computing power provided by supercomputer hardware and the related technologies of high-performance computing software.

3. Research direction of efficient computing for artificial intelligence

With the rapid increase in data volume, the drawbacks of long training time for big data mining and machine learning have constrained the application of large-scale artificial intelligence algorithms. Meanwhile, in recent years, high-performance computing or supercomputing has shown an accelerating trend, and parallel algorithms and platforms for big data and artificial intelligence have

quickly become hot topics in international scientific research and industry. The overall framework structure of an efficient computing platform for big data and artificial intelligence is shown in Figure 1.

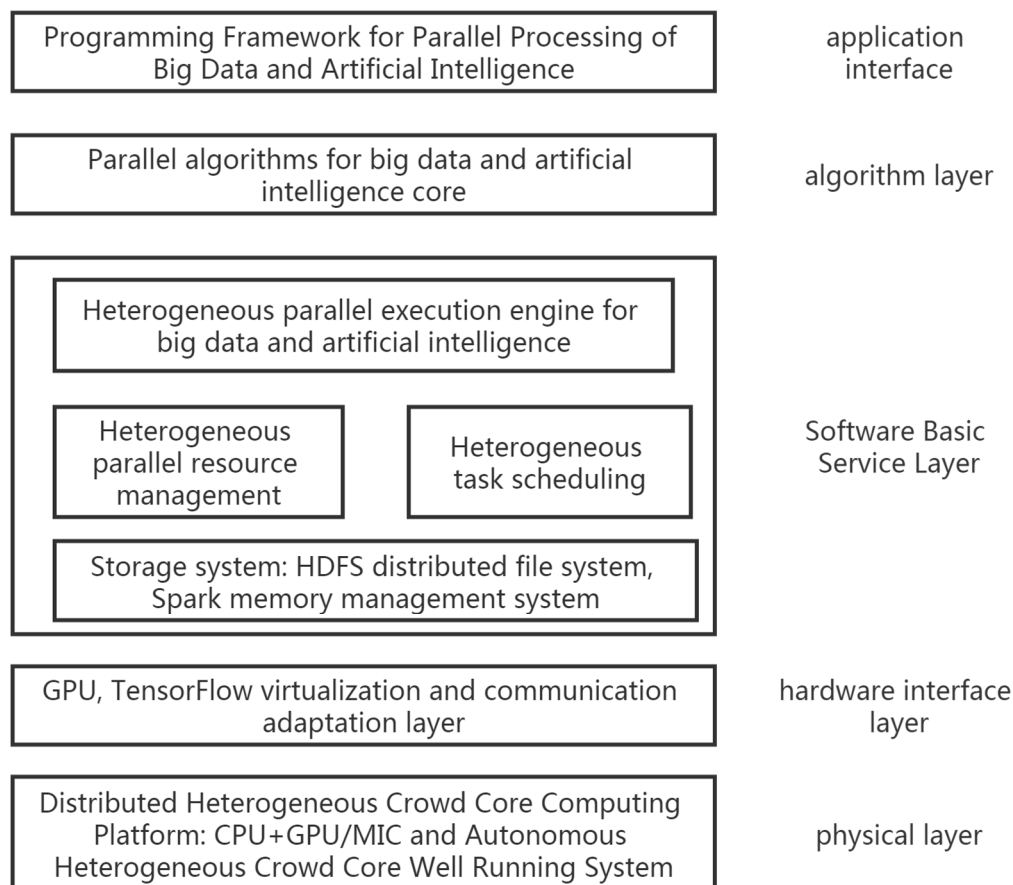


Figure 1. Overall framework structure of an efficient computing platform for big data and artificial intelligence

3.1 Parallel Processing Architecture for Big Data and Artificial Intelligence

3.1.1 Architecture of Heterogeneous Crowd Core Parallel Processing Platform for Big Data and Artificial Intelligence

The current computer architecture of heterogeneous computing has become the mainstream of high-performance computing and supercomputing. Its powerful computing power provides good support for big data processing and artificial intelligence applications. However, the design of heterogeneous parallel architecture needs to consider the requirements of efficient and easy-to-use parallel processing platform design, such as massive data input and intermediate data storage of multimodal machine learning, the mixing and collaboration of various machine learning algorithms, data correlation analysis between multimodal data and other feature fusion algorithms, as well as frequent communication and synchronization required for updating model parameters within a certain mode.

3.1.2 Heterogeneous Crowd Core Parallel Processing Framework for Big Data and Artificial Intelligence

Under the overall architecture, based on the modeling and analysis structure of multimodal machine learning algorithms, fully considering the universality, scalability, and improvement of parallel processing efficiency of the system, we plan to draw on the advantages of using parameter

servers to handle frequent communication on machine learning platforms such as TensorFlow and MXNet. In order to meet the needs of large-scale multimodal machine learning, we will improve it and design an associated network server. To solve the problem of algorithm collaboration in multimodal machine learning parallel processing, a parallel processing framework supporting CPU+GPU/MIC heterogeneous structures and autonomous multi core heterogeneous systems is designed and implemented on this basis.

3.1.3 Efficient resource management and task scheduling strategies

Model the computing power, storage, and network communication capabilities of general CPU+GPU/MIC and domestic autonomous heterogeneous multi-core systems, and design corresponding resource management and task mapping mechanisms. At the same time, based on the characteristics of different machine learning algorithms, efficient task mapping methods within and between computing nodes are studied to fully utilize the computing potential of various computing devices within and between nodes, and improve computing efficiency. At the same time, based on the formal modeling results in the multimodal machine learning process, we plan to explore the scheduling theory and algorithms of directed graph DAG and cyclic graph DFG for multimodal machine learning.

3.2 Design and Implementation of Scalable Heterogeneous Parallel Algorithms and Models for Artificial Intelligence

3.2.1 Heterogeneous parallel algorithm design for multimodal data processing and artificial intelligence

Based on the characteristics of large data volume, complex structure, and redundant information reflected in the formal modeling process of multimodal data, a deep analysis is conducted on the parallelism, parallel granularity, and scale of core algorithms involved in preprocessing, feature extraction, feature fusion, and decision-making processes (Tucker, CP, random gradient, convolutional computation and convolutional neural networks, recurrent neural networks, etc.), which are redesigned and involved in the process. Large scale multimodal data is used as input, Design parallel algorithms for universal heterogeneous multi core systems based on CPU+GPU/MIC and heterogeneous multi core systems based on domestic independent processors.

3.2.2 Model clipping and compression

The large parameter scale is a hallmark feature of intelligent learning algorithms represented by deep learning, which requires a large amount of memory. In order to reduce the parameter size of deep neural networks, model pruning and compression are two commonly used effective methods.

Model pruning is a method of reducing the size of deep neural network parameters by removing small weight parameters or neural network structures. This method can be specifically divided into fine-grained pruning (also known as unstructured pruning) and coarse-grained pruning (also known as structured pruning), as shown in Figure 2. Training as many parameters as possible is considered a guarantee for the learning ability of neural networks. Therefore, removing some parameters will not cause significant damage to the performance of the model. In addition, the weight size of the model parameters represents the importance of the feature. Therefore, removing small weight parameters does not compromise the learning ability of the model. The method of model pruning is simple and effective, but it also has certain limitations. Firstly, Liu et al.'s work shows that parameter size and parameter weight cannot directly represent the learning ability and importance of the network. Secondly, the network parameter matrix obtained by the unstructured pruning method is sparse, and without dedicated hardware and software support, it cannot effectively achieve model calculation, compression, and acceleration. Secondly, structured pruning methods will result in permanent loss of certain capabilities of the network model and significantly reduce the performance of the network. Thirdly, structured pruning methods also require special hardware and software support. Finally, the importance of weights, neurons, network channels, and network layers is not static, so permanently deleting a certain parameter or structure in the network can affect the generalization ability of the

network. Therefore, the dynamic pruning network method based on the type of task or the characteristics of input data is more suitable for the application scenarios of heterogeneous devices and diverse needs in future intelligent environments.

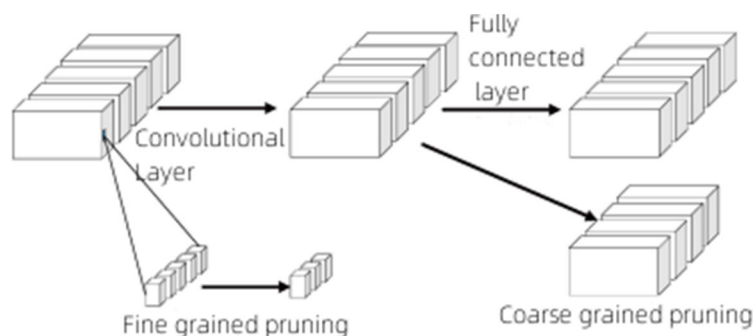


Figure 2. Example of Model Pruning

Model compression is a method of reducing network computing overhead and energy loss through techniques such as weight sharing or tensor decomposition. Unlike pruning methods, model compression focuses more on the operation of convolutional layers, as convolutional computation occupies the main computational cost in the training and inference process of deep neural networks. In order to meet the real-time requirements of network applications, it is necessary to accelerate the training and inference process of deep neural networks. Denil et al. proposed several CNN compression methods with the aim of reducing redundant networks. Mathieu et al. demonstrated that Fourier domain convolution can be twice as fast as regular convolution operations. Denton et al. compressed the weight matrix of the fully connected layer of the network using singular value decomposition (SVD), which improved the network computation speed without significantly reducing the prediction accuracy of the model. At present, methods based on quantization, Hash technology, cyclic projection, and tensor decomposition have shown good performance in network compression. However, the above work only compresses one or several layers in the network and cannot compress the overall network. But for complex tasks and networks like ImageNet, compressing the entire CNN is very important. Zhang et al. improved convolutional efficiency and overall network compression by considering nonlinear element acceleration, and this method can be applied to existing neural network frameworks such as Caffe, Torch, and Theano. However, the rank selection involved in this method still requires significant computational overhead. Due to the large scale and long training time of deep neural networks, the overall compression network remains extremely challenging. The current compression methods mostly focus on the computation or energy optimization of convolutional or fully connected layers, but to some extent, they affect the performance of the network. Therefore, in the future, it is necessary to further explore model compression methods that can effectively reduce network computing and energy consumption costs, and minimize the impact on network performance.

3.3 Stability and Optimization Methods of Heterogeneous Crowd Core Parallel Processing Algorithms and Platforms

In order to improve the computational performance and stability of parallel algorithms and platforms, the following three aspects need to be studied:

3.3.1 Communication optimization methods for large-scale multimodal machine learning parallel processing platforms

In large-scale multimodal machine learning, frequent communication and synchronization are required for the collaboration between various modal algorithms and model updates within the algorithm. Based on its characteristics, communication optimization methods for model compression,

irregular communication optimization methods, and tensor compression based communication optimization methods need to be studied.

3.3.2 Pipeline processing technology within and between nodes

Research the pipeline technology between the CPU and GPU/MIC within parallel computing nodes or the main processor and multi-core coprocessors within domestic autonomous supercomputing nodes, as well as between nodes, to improve system performance. Design the optimal task overlap scheme within and between nodes, hiding as much as possible the number and time of data transmission between CPUs and co processors, as well as between nodes.

3.3.3 Runtime Fault Tolerance and Stability Technology

In response to the current situation of large-scale heterogeneous multi core parallel systems with large scale, complex software and hardware composition, and relatively low reliability, research is conducted on the system stabilization technology for ultra large scale parallel processing systems oriented towards multimodal machine learning. Corresponding fault prediction, recovery, and system reconstruction fault-tolerant mechanisms, as well as checkpoint based fault-tolerant mechanisms, are proposed to enhance the system's continuous operation ability. The fault-tolerant model and fault-tolerant scheduling algorithm of heterogeneous parallel systems are studied. By studying the local probability characteristics of each host node failure and interconnection network failure in heterogeneous computing resources, a dynamic low redundancy strategy with failure rate as the standard and a fault-tolerant model with host and network failure rate as the core are proposed. On this basis, a random fault-tolerant scheduling algorithm that compromises system efficiency and reliability objectives is designed for core algorithms or modules with high reliability requirements in multimodal machine learning, and whose task execution time is a general task with known random variables.

4. Summary

The use of high-performance computing systems, especially supercomputing systems, as computing platforms for big data and artificial intelligence is gradually becoming a trend. Various new types of processors have been applied, continuously increasing the computing power of the system, thereby promoting the large-scale and deep development of big data and artificial intelligence applications, which in turn puts higher demands on computing systems. Improving the computing performance of a single processor alone cannot keep up with the development needs of applications, and large-scale heterogeneous parallel computer systems based on multi-core and slave cores have become the mainstream of development. However, due to the complexity of heterogeneous systems and the high difficulty of parallelizing big data and artificial intelligence applications, there is an urgent need to improve the computational efficiency of executing big data and artificial intelligence applications on large-scale heterogeneous parallel computer systems. The improvement of computing efficiency is a systematic project, which needs to be studied from the bottom level of resource management, task scheduling, and basic algorithm design, communication optimization, to the upper level of model parallelization and parallel programming. While giving full play to the powerful parallel computing capabilities of many-core processor, it also needs to improve the utilization of computing resources. Additionally, optimizing algorithms and models can effectively reduce energy consumption costs.

References

- [1] Wu F, Lu C, Zhu M, et al. Towards a new generation of artificial intelligence in China[J]. *Nature Machine Intelligence*, 2020, 2(6):312-316.
- [2] Huang Y, Zhang Z, Wang N, et al. A new direction to promote the implementation of artificial intelligence in natural clinical settings[J]. 2019.

- [3] Wan M. The Challenge and Research of Artificial Intelligence to the Legal Problems of Intellectual Property[C]// International Conference on Contemporary Education, Social Sciences and Humanities. 2019.
- [4] Kang B, Guo J, Meng X. Rapid Implementation of COVID-19 AI Assisted Diagnosis System Based on Supercomputing Platform[C]// International Conference on Universal Village. IEEE, 2020.
- [5] Zheng X. Artificial Intelligence Development and Leading Force Cultivation in Shanghai[J]. Scientific Development, 2019.
- [6] Rajna G. Brain-Inspired Supercomputing. 2017.