Research on the Application of Artificial Intelligence Algorithms in GPU Resource Optimization and Scheduling Control

Zhen Li

Information Engineering Department, Jiangsu Maritime Institute, Nanjing, 211100, China

Keywords: Artificial intelligence algorithms; GPU; Resource optimization and scheduling; Control; Application

Abstract: GPU is a new commercial computing model and an extension and expansion of technologies such as distributed parallel processing and grid computing. It represents a new stage in the development of parallel computing technology and belongs to a brand new product. With the maturity of this field and the continuous growth of the internet and global enterprises, it is expected that more cloud service providers will emerge. This article focuses on the research and development scenarios of AI(Artificial Intelligence) algorithms and conducts a series of scheduling strategy studies based on GPU to improve the task execution efficiency and cluster resource utilization of the AI algorithm development platform. Finally, the research results indicate that when the historical time slot changes from 50 to 590, the accuracy of prediction becomes more and more accurate as the size of the time slot increases, and the prediction curve gradually becomes flat at 410 to 460. It can be inferred that there may be an extreme value around the time slot size of 460, and exceeding this extreme value has little impact on predicting low utilization time results. During the process of increasing average execution power, resource correlation algorithms are used to schedule tasks, reducing the overhead of the cloud system data center network and improving the system's resource utilization.

1. Introduction

In recent years, GPU has made breakthrough achievements in the fields of floating-point computing and parallel computing, and has great potential for improvement. GPU is a brand-new commercial computing mode, and it is also the extension and expansion of distributed parallel processing and grid computing. It represents a brand-new stage of the development of parallel computing technology and belongs to a brand-new product. There are many problems in GPU that need to be solved effectively, such as resource scheduling. The heterogeneous computing system based on GPU can provide computing power to meet the needs of industrial manufacturing, AI, medical care, aerospace and other fields under reasonable algorithm scheduling, and can provide core technologies for China's economic development and provide strong support for the strategy of "strengthening the country through science and technology" [1]. Resource scheduling is the main component of GPU, and its efficiency has a direct impact on the performance of GPU environment. Heuristic intelligent algorithm is the main research direction in this field. With the maturity of this field and the continuous growth of Internet and enterprises in the world, it is expected that more cloud service providers will emerge, provide a more diversified choice, more resources and services, and increase energy consumption and negative pressure on the environment [2-3]. However, the increased network circulation load and energy consumption brought by the network are gradually increasing, and the energy consumption of GPU is rarely taken into account in the existing resource scheduling algorithms [4]. In resource scheduling control, the energy consumption of GPU is still rarely involved. Although GPU can provide computing and storage services to save energy by using large shared servers and storage units, this effect will be especially obvious when end users tend to use a computer or terminal with low capacity and reduce energy consumption migration [5]. How to improve the utilization of GPU resources, and then improve the task throughput of AI algorithms, has gradually attracted more and more attention. In this paper, a series of scheduling strategies are studied based on GPU to improve the task execution efficiency and cluster resource utilization of

the AI algorithm research and development platform.

2. Research on GPU Resource Optimization and Scheduling Control Based on AI Algorithms

2.1. System control algorithm

The productization research and development of AI from academic research to production applications usually involves various steps and tools, especially in environmental deployment, testing, tuning, and other work, which often consumes a lot of time and expenses. Usually, when optimizing resource scheduling, each task is matched with the best computing unit, and in task replication based scheduling algorithms, the same computing task can be assigned to multiple computing units. The algorithm defaults to no communication overhead between tasks located in the same computing unit, By copying the predecessor task of a task to the idle time slice of its computing unit, the interaction cost between tasks and the idle time of the computing unit are reduced [6-7]. This algorithm uses the computational cost of computing units to exchange for communication costs between tasks, and is suitable for computing tasks with frequent communication between nodes. In order to make the controller adaptive and achieve automatic adjustment of controller parameters, AI algorithms can be used. Due to the parallel mechanism, learning ability, and memory function of AI algorithms, utilizing the self-learning characteristics of AI algorithms and combining traditional ideas, a GPU resource optimization scheduling controller is constructed, achieving automatic adjustment of conventional controller parameters [8]. The principle of the control system is shown in Figure 1.

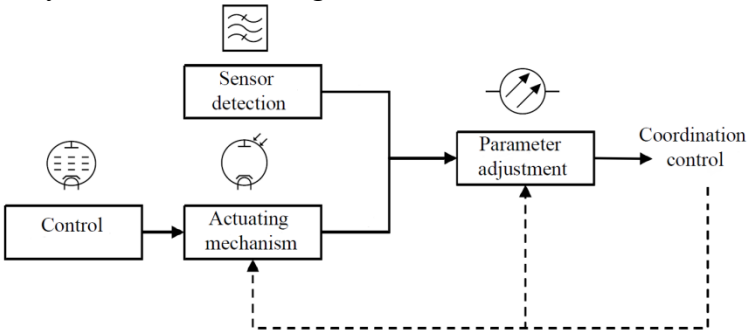

Figure 1 Control System Schematic Diagram

The core concept is to construct a critical path task sequence based on b-level and t-level values, and then copy the GPU and its direct predecessor tasks to the same computing unit, advancing the start time of the task. The calculation formula is as follows:

$$t1(t_i) = \max\left(c_{i,j} + w(t_j) + tl(t_i)\right) \quad (1)$$

$$b1(t_i) = w(t_i) + \max\left(c_{i,j} + bl(t_j)\right) \quad (2)$$

The core concept of the algorithm is to execute Join tasks as early as possible. The Fork Join graph dedicated algorithm addresses the issue of interactive channel conflicts in the actual GPU resource optimization scheduling process. The controller provides a flexible framework for the construction of control strategies. By adjusting parameters, conflicts between various performance indicators in the control system can be effectively reduced [9-10].

2.2. Resource optimization scheduling control strategy

Our GPU resource optimization and scheduling strategy targets scenarios where data center tasks arrive dynamically, with the goal of maximizing system throughput as much as possible. System throughput can be measured by the number of instructions per cycle (IPC) executed in the kernel. The expression for IPC is as follows:

$$IPC = \frac{Inst}{Cycles} \quad (3)$$

Among them, $Inst$ represents the total number of running instructions for a kernel, $Cycles$ represents the number of clock cycles required for the kernel to run alone, and the IPC of the kernel can be expressed as the ratio of the total number of running instructions for the kernel to the number of clock cycles required for the kernel to run. We use normalized IPC to measure system throughput. The normalized IPC of a kernel refers to the IPC that is normalized to the kernel when running separately. The specific expression for normalized IPC is as follows:

$$norm\_IPC = \frac{IPC^{MK}}{IPC^{SK}} \quad (4)$$

Among them, IPC^{MK} represents the IPC exhibited by the kernel when running together with other kernels, and IPC^{SK} represents the IPC exhibited by the kernel when running alone.

The ultimate goal of scheduling strategies is to maximize the overall system throughput in typical application scenarios where data center tasks arrive dynamically. The tasks submitted by the user will be imported into the task waiting queue, and during the scheduling process at each scheduling point, the selection of placement strategies and the calculation of waiting margin will be completed based on performance prediction models. At the same time, the tasks in the task queue will be reordered based on the waiting margin. The implementation of waiting queues is based on a heap structure, which can achieve automatic sorting of tasks based on event triggering mechanisms. When the number of task nodes in a parallel task graph far exceeds the number of computing units in a heterogeneous system, the communication overhead between subtasks during the execution process is extremely high, which increases the computational cost. Aggregating small-scale tasks together, reducing the number of tasks to be scheduled, and increasing the task size in exchange for a reduction in communication overhead and scheduling length. Our scheduling strategy mainly consists of two performance analysis modules: kernel collaborative operation performance analysis module and kernel analysis module, as well as research on AI algorithms based on the above two performance analysis modules in GPU resource optimization scheduling control.

3. Experimental results and analysis

3.1. Experimental setup

Due to the high utilization rate of GPU resource optimization, the task may fail due to too long waiting time, or because of too long execution time, although all submitted tasks have been completed, it wastes too much energy consumption of the cloud system. For such a large number of tasks coming in a certain period of time, this chapter puts forward the energy consumption optimization research of GPU resource scheduling in AI system based on task tolerance value. In this chapter, in order to test the performance and efficiency of the energy consumption optimization algorithm of cloud computing resource scheduling designed by time prediction in different environments, a cloud computing test platform with 130 nodes is built by using Cloud Sim simulation under the laboratory platform. In the experiment, in order to make the algorithm get better performance and not lose too much overhead of the system, the following settings and functions are verified. Firstly, the system resource utilization rate of the algorithm and the setting of historical time slots are verified. Secondly, the experimental scale of the proposed algorithm and the algorithm to be compared is set. When the cloud node is 130, the experiment of system utilization threshold y and total energy consumption of the system is carried out, and the average value of 95 results is selected as the experimental result. The analysis is shown in Figure 2.

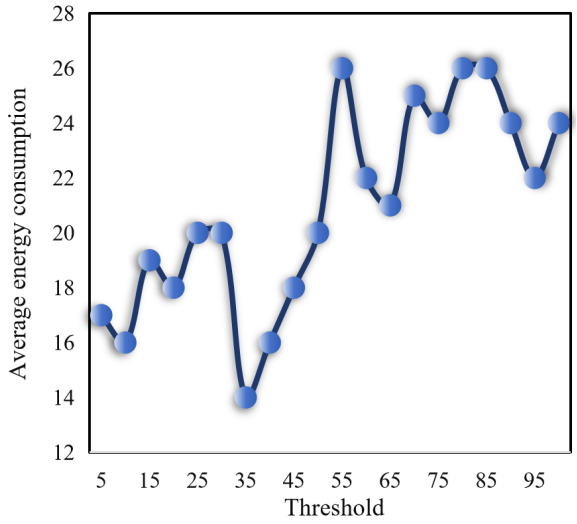

Figure 2 Schematic diagram of the relationship between system utilization threshold and energy consumption

As shown in Figure 2, it can be seen that when the system utilization threshold y is between 0-25%, the total energy consumption of the system does not change much, and frequent transitions to the system's state consume too much energy, which is offset by the energy saved by sleep. When the system utilization rate is between 35% -70%, the energy consumption of the system gradually increases. After multiple experiments and repeated testing, it has been found that when the system utilization threshold is about 75%, the system can achieve the best results through optimization algorithms.

130 nodes were selected for an experiment on system prediction accuracy with a random task of 80 and a cloud system scale. The experimental results were taken as the average energy consumption of 95 tests. The experimental analysis of historical time slots and accurate prediction is shown in Figure 3.

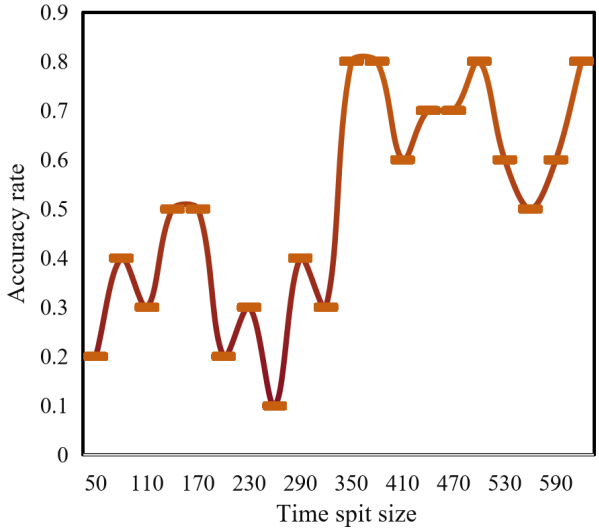

Figure 3 Schematic diagram of time slot and accuracy

From Figure 3, it can be seen that when the historical time slot changes from 50 to 590, the accuracy of the prediction becomes more and more accurate as the size of the time slot increases, and the prediction curve gradually becomes flat at 410 to 460. It can be inferred that there may be an extreme value around the time slot size of 460, and exceeding this extreme value has little impact on predicting low utilization time results.

3.2. Experimental Summary

When the node size of the cloud system is 0-128 nodes, the system utilization threshold y is 72%,

and the historical time slot size is 570, the average energy consumption of the algorithm based on time prediction is about 7.62% lower than that of the average prediction algorithm under different task sizes, while it is 20.52% lower than that of the management optimization strategy. With the gradual increase of tasks, the algorithm based on time prediction gradually approaches and equals to the idle energy consumption of management optimization algorithm. Achieve higher service availability through network redundancy. Although only one data center is designed, cloud service providers usually maintain redundant centers for efficient data transportation and load balancing among these data centers. The goal of virtualization is to keep tasks parallel, improve flexibility, scalability and utilization of hardware resources, and then reduce the hardware and operating costs of data centers. Random task sets are scheduled to cloud service resource nodes for execution through task waiting queues. However, the arrival time of random tasks is uncertain and the resource requirements are unequal. The research assumes that the task waiting queue is long enough, and the cloud resource service center is responsible for reasonably scheduling task execution. The biggest advantage of adopting AI technology is that it can flexibly map the performance strength of physical resources to virtual servers. By appropriately reducing the resources allocated to tasks, the system load rate is increased to optimize the resource utilization rate of cloud computing nodes. By increasing the parallelism of task execution, the task waiting time is reduced, and then the waiting energy consumption is optimized. Historical time slot algorithm can reduce the error rate of prediction and adjust it quickly when there is an error; The proposed algorithm saves more energy than the average energy consumption prediction algorithm and energy consumption management optimization algorithm by setting the system utilization threshold and selecting the time slot size, and also shows some advantages in some system performance.

4. Conclusions

In order to reduce the transmission energy consumption during the execution of cloud computing systems when the random task volume is normal and moderate, this article designs an optimized scheduling algorithm based on cloud computing resource correlation energy consumption from the perspective of resource correlation. This article conducts research on the application of AI algorithms in GPU resource optimization scheduling control. This article also proposes a resource optimization scheduling control strategy. When the number of task nodes in the parallel task graph far exceeds the number of computing units in heterogeneous systems, the communication overhead between subtasks during the calculation execution process is extremely high, which increases the computational cost. Aggregating small-scale tasks together, reducing the number of tasks to be scheduled, and increasing the task size in exchange for a reduction in communication overhead and scheduling length. The optimization of transmission energy consumption was achieved through task scheduling and allocation processes in AI algorithms, as well as network load levels. The research results indicate that when the historical time slot changes from 50 to 590, the accuracy of prediction becomes more and more accurate as the size of the time slot increases, and the prediction curve gradually becomes flat at 410 to 460. It can be inferred that there may be an extreme value around the time slot size of 460, and exceeding this extreme value has little impact on predicting low utilization time results. During the process of increasing average execution power, resource correlation algorithms are used to schedule tasks, reducing the overhead of the cloud system data center network and improving the system's resource utilization. This article also designs a resource optimization scheduling control strategy to achieve a balance between the allocation of memory resources and task execution performance.

References

[1] Nguyen T T, Vo D N.The application of an effective cuckoo search algorithm for optimal scheduling of hydrothermal system considering transmission constraints[J].Neural Computing & Applications, 2022, 18(9):1-22.

[2] Nguyen T T, Vo D N.The application of an effective cuckoo search algorithm for optimal scheduling of hydrothermal system considering transmission constraints[J].Springer London, 2019, 30(8):10-18.

[3] Kang L, Liu D, Wu Y,et al.An Improved DE Algorithm for Solving Multi-Furnace Optimal Scheduling of Single Crystal Silicon Production[J].International Journal of Pattern Recognition and AI, 2023, 37(02):10-22.

[4] Shi H, Cao G, Ma G,et al.Intelligent Service Scheduling Decision Technology under AI Technology[J].Journal of Physics: Conference Series, 2021, 1986(1):012078-012087.

[5] Patronas G, Vlassopoulos N, Bellos P,et al.Accelerating the scheduling of the network resources of the next-generation optical data centers[J].Parallel computing, 2023, 19(20):36-44.

[6] Meng R, Rao Y, Zheng Y,et al.Modelling and solving algorithm for two-stage scheduling of construction component manufacturing with machining and welding process[J].International Journal of Production Research, 2018, 56(19):6378-6390.

[7] Liu D, Yao Z, Chen L.Emergency Scheduling Optimization Simulation of Cloud Computing Platform Network Public Resources[J].Complexity, 2021, 2021, 20(11):10-18.

[8] Jordehi A R.Enhanced leader particle swarm optimisation (ELPSO): a new algorithm for optimal scheduling of home appliances in demand response programs[J].AI Review, 2020, 53(3):2043-2073.

[9] Wang Y, Qiao J, Lin S,et al.An Approximate Optimal Solution to GPU Workload Scheduling[J].Computing in Science & Engineering, 2018, PP(5):21-30.

[10] Liu S, Wang N.Collaborative Optimization Scheduling of Cloud Service Resources Based on Improved Genetic Algorithm[J].IEEE Access, 2020, 86(99):11-19.