Title	マルチノード水冷GPUクラスタにおけるファシリティ資源を考慮した計算制御手法の検討
Author(s)	高橋, 亮真
Citation	
Issue Date	2025-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/19830
Rights	
Description	Supervisor: 篠田 陽一, 先端科学技術研究科, 修士 (情報科学)

Study of computational control method considering facility resources in multi-node water-cooled GPU clusters

2210104 Ryoma Takahashi

In recent years, GPGPUs, which use GPUs to accelerate parallel computing, have become widespread, and the demand for large-scale GPU clusters in data centers is expanding in response to the AI boom. However, in computer clusters using GPUs, due to the nature of the workload, a small number of jobs occupy the majority of nodes in tasks such as machine learning and model training, making it difficult to design facilities based on traditional power design using statistical multiplexing. Therefore, the current options are to prepare expensive, large-capacity facilities to match the maximum power consumption of the GPU cluster, or to reduce the scale of the cluster, which is a major obstacle to the spread of GPU clusters. In this study, we propose control using scheduling as a method to efficiently operate GPU servers even in environments where such large-capacity facilities such as power supplies cannot be secured. Evaluation was performed using simulations. The reason for limiting the title of this study to water cooling is to simplify the simulation, as it would be necessary to delve into heat exhaust and airflow issues that are unrelated to the main topic if such large-scale facilities were used. This scheduler uses two approaches to improve operating efficiency with limited facilities resources in order to utilize as many computing resources as possible even in situations where such large-scale facilities resources cannot be secured. The first is power-based scheduling based on estimated power. Generally, most of the power consumption in a server is semiconductors such as CPU, GPU, and memory, and it is usually the device driver that controls these semiconductors to operate in a power-saving manner, and the content that the scheduler can do is limited. Therefore, the policy was to maximize the consumption of the given power budget rather than to operate with reduced power. Specifically, jobs are scheduled to reach the maximum power limit based on the estimated power consumption based on pod requests. However, even if the power is maximized, there is not enough power to fully operate another server in the situation, so we thought that there is a possibility that the computation nodes will be insufficient and task execution will be congested. Here, as the second approach, we decided to improve the operating efficiency and prevent job starvation due to blocking, referring to the "Tiresias" scheduler and the scheduler of K and Fugaku. Specifically, we adopted backfill and gang scheduling to improve the operating efficiency of the cluster and prevent deterioration of turnaround time and execution time.

The evaluation was performed assuming an NVIDIA DGX H100 with a maximum power consumption of 10.2kw and a kubernetes environment. The

reason for selecting the DGX H100 server is that it is an NVIDIA product popular for AI HPC due to its convenience and ease of development, and that it has a common configuration for high-end GPU servers. The reason for selecting k8s software is that it is widely adopted in GPU clusters of companies in Japan and overseas due to its high fault tolerance and convenience, and is suitable as software for general GPU clusters. In addition, the specification was set to evacuate preemption caused by Gang to the main data cache SSD because evacuation consumes a large amount of capacity, including the contents of the working memory.

For the evaluation, trace data from cluster-trace-gpu-v2023 of the Alibaba Cluster Trace Program published by Alibaba in China was used. This data is trace data of a Kubernetes GPU cluster by Alibaba, and was adopted because it is considered to be close to the actual workload environment of the Kubernetes cluster previously determined. In the evaluation results using Cluster Trace data, the power constraints of the data center were limited to 45kw and 60kw, and the proposed scheduling method was applied to the GPU cluster. For comparison, a conventional method was used with a scheduler that uses only computational resources. Simulation results showed that the proposed method had a 92% faster turnaround time and 5.8% faster execution completion time under 45kW conditions. In addition, a comparison under 60kW with the power of one DGX H100 added showed that the turnaround time was 38.2% faster and the execution completion time was 6.6% faster. This shows that the proposed method can reduce job blocking while maintaining a higher power usage rate, and can perform scheduling that results in faster turnaround times and execution times. In performing gang scheduling, we assumed that the backup destination was the cache SSD of the DGX main unit, but we also investigated the impact of the performance difference of this drive. As a result, the faster drive had a higher number of preemption attempts, resulting in slower results. Similarly, an experiment was conducted in which the preemption interval was changed, and it was confirmed that the longer the preemption interval, the slower the execution completion time. In this way, the proposed method was able to demonstrate advantages in both turnaround time and execution time compared to the conventional method. However, this may not be an accurate value due to the rough moderation of the simulation (including time penalties). In the future, we hope to improve the fidelity and discover weak workloads by using more accurate moderation and trace data with different workload characteristics. This work provides a new approach to improve the operating efficiency of GPU servers in power-constrained environments and promotes the deployment of servers in such environments.