2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP) | 979-8-3503-0245-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSP58490.2023.10248580

A Job-Aware Decision Method for Hybrid HPC Cluster Scenarios

Yang Bai
Qilu University of Technology (Shandong Academy of Sciences)
Shandong Computer Science Center(National Supercomputer Center in Jinan)
Shandong Provincial Key Laboratory of Computer Networks
Jinan, China
10431200546@stu.qlu.edu.cn

Ying Guo*
Qilu University of Technology (Shandong Academy of Sciences)
Shandong Computer Science Center(National Supercomputer Center in Jinan)
Shandong Provincial Key Laboratory of Computer Networks
Jinan, China
guoying@sdas.org

Hu Zhang
Qilu University of Technology (Shandong Academy of Sciences)
Shandong Computer Science Center(National Supercomputer Center in Jinan)
Shandong Provincial Key Laboratory of Computer Networks
Jinan, China
zhanghu@sdas.org

Jibin Wang
Qilu University of Technology (Shandong Academy of Sciences)
Shandong Computer Science Center(National Supercomputer Center in Jinan) , Jinan, China
Shandong Provincial Key Laboratory of Computer Networks
Jinan, China
wangjb@sdas.org

Jing Chen
Qilu University of Technology (Shandong Academy of Sciences)
Shandong Computer Science Center(National Supercomputer Center in Jinan)
Shandong Provincial Key Laboratory of Computer Networks
Jinan, China
jingc@sdas.org

***Abstract*—As an emerging science and technology industry, the development direction and strategic height of high-performance computing has become one of the important symbols of scientific and technological progress nowadays, and also a centralized embodiment of a country's comprehensive strength in science and technology. To cope with the development demand of high performance computing, meeting the resource utilization conditions under multi-dimensional constraints and forming reasonable scheduling are always the most urgent issues. Based on the current development status of hybrid HPC clusters and the problem of resource wastage due to unreasonable scheduling allocation, a job task-aware migration policy method applicable to hybrid HPC cluster environment is proposed in this paper: ECP - Perception-aware decision method, ECP - Perception-aware decision method can target HPC job tasks submitted by users The ECP - Perception-aware decision method can make reasonable load prediction for HPC job tasks submitted by users, and migrate the job tasks to a cluster environment with relatively lower resource consumption or monetary cost based on the prediction result and the perception policy. After comparing several different job task migration models in simulated production environments, the results show that the ECP - Perception-aware decision approach enables HPC job tasks to obtain lower resource consumption or lower monetary cost in a hybrid HPC cluster environment than when running in a single physical HPC cluster or Kubernetes cluster, and also enables the hybrid HPC cluster to resources as well as users' demands are better balanced before, which better solves the multi-objective optimization problem of HPC job task-aware scheduling.**

Keywords- high performance computing；hybrid HPC clusters；multivariate resource demand forecasting；perceptual decision making；multi-objective optimization

I. INTRODUCTION

As the level of computer and network technology continues to soar, high-performance computing (HPC) has become the third scientific research paradigm after theoretical and experimental science. In recent years, researchers and HPC users have started to focus more on resource and time requirements based on the objective status of the rapid development and wide application of HPC. For example, if an HPC user submits an HPC application task at a certain scale, the current computing cluster does not provide enough resources for the task to run, and the task will be in a queue for a long time. When the current computing cluster does not provide enough resources to run the task, the task will be in a queue for a long time and cannot be run in time [1]. Driven by the above background, cloud computing platforms have the advantages of lightweight virtualization, controllable resources, and elastic provisioning compared to physical HPC clusters or supercomputers, and therefore, many researchers and HPC users have started to try to use cloud computing platforms as an economic alternative to physical HPC clusters or supercomputers from the economic aspect. Some researchers say that the virtualization technology in the cloud computing platform has some additional performance trade-offs and overheads in terms of network or storage, resulting in HPC case tasks running in the cloud computing technology platform cannot obtain comparable performance to that in supercomputers or HPC clusters, i.e., it cannot be a replacement for supercomputers or HPC clusters, but the cloud computing platform can be considered platforms as a complement to supercomputers or HPC clusters [2]. First, from the perspective of HPC users, considering that there exists a part of users with low computational performance requirements for job tasks and more focus on computational cost performance, they are willing to migrate their HPC applications to run on a cloud computing platform to obtain shorter queuing time and cheaper rental cost for machine hours. Second, from the perspective of the platform operator, transferring appropriate HPC tasks to the cloud platform can reduce the generation of resource fragments in the HPC cluster, thus improving the overall resource load balancing and utilization of the platform, as well as effectively shortening the user response time [3].

Application performance tests have been conducted for different characteristics of HPC applications on various supercomputers and cloud platforms to analyze the application runtime bottlenecks and the correlation between the application characteristics and the observed performance to determine the suitable HPC job tasks for migration to the cloud platform [2]; some researchers have also conducted the prediction of the HPC job task runtime for different HPC applications on different supercomputers and cloud platforms to optimize the HPC job tasks by predicting the resource usage or runtime. The prediction of HPC job task runtime for occupied resources or runtime has been carried out to optimize by the prediction of resource occupation or runtime [4]. With continuous research, the problem of resource occupation for HPC job tasks have been derived from a single-objective optimization problem to a multi-objective optimization problem of HPC performance and resources, and even a balance between the financial cost associated with energy consumption by HPC administrators and the need of users to obtain results in a timely manner [5].

In response to the development of cloud computing and high-performance computing, the concept of hybrid clusters has emerged. The hybrid cluster environment can make full use of local resources as well as cloud platform resources, and a portion of HPC tasks that are more suitable for running in the cloud platform will be reasonably perceived and migrated to a suitable cluster or cloud platform for each HPC task submitted by users, so as to reduce the resource fragmentation generated in the physical environment and improve the overall HPC cluster environment. This reduces resource fragmentation in the physical environment and improves the resource utilization and load balancing of the entire hybrid HPC cluster environment. An increasing number of HPC cloud researchers are also targeting their research efforts to study the cost-effectiveness of migrating resource-intensive HPC applications from local cluster environments to public cloud platforms [6]. High-performance computing application scenarios are numerous and each application requires different environment dependencies, while container technology, as a more lightweight virtualization technology, which has higher resource utilization as well as more efficient execution performance, is introduced into the field of high performance computing to solve the problem of high performance application software with large system dependencies and complex environment configurations through mirror management mechanism. The application configurations are isolated, packaged into images packaged, and shared to achieve the goal of running HPC applications across clusters as well as handling large amounts of data [7]. We consider a hybrid cluster-based HPC job-aware decision-making approach for HPC applications of different sizes, considering their characteristics in different cluster environments, and combining container clouds as a complement to traditional HPC clusters with supercomputers effectively to provide a hybrid cluster-based HPC job-aware decision-making approach to support decision making by providing HPC users with a perceptual migration function. This includes: predicting the time required for the HPC job to run the results in different cluster environments in the hybrid HPC cluster environment and the occupancy and consumption load for resources based on the HPC job requirements submitted by the user, predicting the resource consumption of the HPC job in various cluster environments based on the resource consumption characteristics of the HPC job in different platforms; and combining the prediction results to Cluster-awareness.

Our contributions include the following: firstly, we propose a perceptual decision-making method, ECP (Energy Cost (Product) - Perception), in terms of resource consumption and monetary cost, by combining the multivariate feature prediction model MRDP (Multivariate Resource Load Forecasting) obtained from previous studies for HPC scenarios. Product)-Perception, which can evaluate and perceive the HPC job tasks submitted by users in a cluster environment with relatively less resource consumption and monetary cost generated by job operation; Secondly, we simulate the actual production environment of the data center by building our own physical HPC cluster and Kubernetes cluster environment to study the impact of our ECP-aware decision-making method on the data center, and finally verify that our method is feasible and effective.

II. RELATED WORK

As mentioned above, the core of our work is to combine the respective advantages of the traditional supercomputing environment and the cloud computing environment, to respond to the new demands of the current stage of HPC development, and to explore a new technical route, i.e., to form a job-aware decision-making method applied to hybrid HPC clusters, to migrate HPC job tasks to a more suitable environment for execution, as an attempt to use the cloud computing platform as a supplement to supercomputers or HPC clusters, to better complement and improve the HPC industrial ecology in China.

Task allocation for HPC jobs has been defined by system developers as managing and assigning rigid jobs to resources [8], that is, directly to clusters, platforms, or nodes. Based on the job requirements submitted by the user, the system first estimates the resource requirements as well as the running time of the job. For job scheduling policies, the most widely used HPC job scheduling policies are first-come, first-served (FCFS) and job backfill [9], where FCFS sorts the jobs in the queue based on their arrival time, and backfill is usually used with reservation to improve system utilization. Backfilling allows subsequent jobs in the queue to be advanced without delaying existing reservations. However, as the resource and performance requirements of HPC providers and HPC users have increased, researchers have begun to focus more on whether scheduling algorithms can better serve to save costs and reduce runtime. The literature [10] proposes a local critical path algorithm (SC-PCP) to allocate and schedule large-scale workflows in SaaS environments. This algorithm uses several approaches to reduce costs, such as recursion and non-fastest speed, which lead to necessary losses in time performance and are also less suitable for more complex task scheduling environments. The literature [11] proposes a hybrid cloud-based budget-constrained package task scheduling algorithm (BaTS) that requires scheduling as well as assigning jobs among different cloud resources based on the cost requirements proposed by the user to achieve a combined cost and completion time optimization goal. The algorithm dynamically adjusts the machines within the cluster according to time and cost during task scheduling, and this approach adds additional performance overhead and network transmission; moreover, there is still a need for further optimization of the algorithm in terms of minimizing completion time and improving flexibility.

Also, for hybrid cluster environments and the derivation of heterogeneous cloud platform environments, researchers have paid more attention to the load energy consumption as well as resource utilization of the platform. Literature [12] proposed a load balancing scheduling method (SALB) for cloud environments based on a simulated annealing algorithm, which performs dynamic migration of virtual machines based on CPU utilization of physical devices. This algorithm has better load scheduling performance than the traditional simulated annealing algorithm and rotation algorithm; however, other load metrics such as memory and bandwidth are not considered. The literature [13] proposes a resource scheduling algorithm (BFOHH) based on a hyper heuristic algorithm, based on bacterial foraging optimization techniques, which forms the workflow through a Lublin model and compares the execution time and cost changes according to the expected time calculation model. The algorithm compensates for the shortcomings of the related algorithms such as insufficient performance and poor adaptability when facing tasks in heterogeneous environments; however, it lacks optimal management of available nodes when performing resource scheduling, and the stability and reliability of resources still need further improvement. The literature [14] proposes a geographically distributed data center scheduling balancing strategy for the overall load and designs a MinBro-wn scheduling technique to reduce the power consumption by combining the power consumption of different features, and this scheduling scheme performs task scheduling by calculating the power consumption generated by the jobs on different data centers. Compared with other scheduling methods, this scheme reduces power consumption to a greater extent; however, it imposes certain requirements on the bandwidth and environment of the data center [15].

In summary, the current development of HPC job scheduling and application awareness under hybrid clusters in job migration, cloud computing, and task scheduling migration as well as current job migration technologies address the reasonable needs of some users today, but a macro view of the development of the field, with the continuous enrichment of high performance computing, most of the current scheduling schemes and application-aware systems have some imperfections. Examples include time and cost issues and the range of dimensions involved. It is easy to see that there is much room for improvement in application-aware and scheduling strategies for high-performance computing. In this thesis, we propose a new task-aware decision method for HPC jobs that applies to high-performance computing. In it, we perceive HPC application jobs submitted by users based on two cluster environments, the traditional physical cluster computing environment, and the container cloud cluster environment, to select the cluster environment that is more suitable for running the job.

III. PROBLEM DESCRIPTION AND ANALYSIS

The formulation of task placement or migration decisions for hybrid clusters or cloud environments has been addressed by researchers as a multi-objective optimization problem [16]. The three aspects involved in dealing with the multi-objective optimization problem are a service provider - IaaS; user or application - SaaS; and workload. In the case of the multi-environment task placement problem, the workload can also be understood as SaaS. where the IaaS service provider pursues the goal of minimizing energy consumption; SaaS wants to improve or at least maintain the performance of the workload; and the user wants to reduce the cost of resources and loads.

We describe the perceptual migration function as solving a multi-objective optimization (MOO) problem that consists of three nominal types of costs, namely energy consumption (Ec), user monetary cost (Uc), and workload performance cost (Wpc). These three cost attributes relate to the two different aspects mentioned above: service provider (IaaS); user or application (SaaS); taking into account the respective attributes and characteristics, we map each cost precisely to the specific objective given below: IaaS - reduction of resource energy consumed by job tasks during execution - Ec; work Load SaaS - to improve or at least maintain the possible performance level at a defined cost - Wpc; User SaaS - to reduce or at least maintain the cost as much as possible according to the platform consumption agreement - Uc.

In the above objective, workload performance can be understood as the antonym of workload execution time (R), and we aim to reduce or at least maintain R-Wpc; at the same time, the user cost incurred is related to the workload runtime submitted by the user and the resource power consumption incurred to maintain the performance, indicating that Uc is proportional to R. Therefore, if the objective to improve or at least maintain the possible at a determined cost level of performance can be achieved, then the goal to reduce or at least maintain the cost as much as possible can also be intuitively achieved, and the performance improvement in terms of runtime will achieve the user's goal. Therefore, in this study, the third objective is no longer considered and analyzed as a separate objective. Therefore, we transformed the multi-objective problem into a bi-objective problem. The two objectives of the transformed bi-objective optimization problem are: to reduce the consumption of resource energy (E), and to reduce the cost incurred by the operational task load (C). These objectives can be described by a mathematical formula as an objective function F, as shown in Equation (1).

$$F = \begin{cases} minimise(E) & where\ E = \sum_{i=1}^{hosts} P_i \\ minimise(C) & where\ C = \sum_{j=1}^{w} cost_j \end{cases} \quad (1)$$

where E denotes the total energy consumed by the resources of all hosts in a particular cluster, and Pi is the resource consumption of a particular host i, which can be considered as a nonlinear function related to the CPU utilization level of the hosts [17]. In addition, $cost_j$ denotes the cost incurred by the consumption of a particular feature of a workload w in a particular cluster, such as execution time, number of nodes, and core time. The sum of the costs of all features in a job task is denoted by C. C is generally provided by each cluster platform in the data center, and the monetary cost incurred by the job is related to the response time of the job running load as well as the resources occupied. The lower the value of C, the lower the value of C, indicating that with the same demand the lower the cost required for job tasks to run on that cluster, and also implies higher performance [18].

To better solve the perceptual decision problem, we consider transforming the above dual-objective problem into a single-objective minimization problem, which requires us to combine the above two objectives in different ways. The literature [19] proposed an ERP metric, which is an energy-corresponding time product metric that responds to the tradeoff between energy, performance and cost. Therefore, ERP metrics are widely used in hybrid cloud environments as a tradeoff to select various cloud environments [20, 21]. ERP metrics aim to capture the tradeoff between energy and performance by comparing the advantages and disadvantages of different environments or policies through energy-corresponding time multipliers, also known as energy-delay products (EDP). Reducing ERP can be pursued to maximize the "performance-to-power ratio", i.e., to maximize the performance achieved by consuming one watt-hour of energy, where the performance of the workload is expressed as the inverse of the response time. the evaluation metric given by ERP is given by Equation (2):

$$ERP = E \ * \ R \quad (2)$$

In our equation, the performance of a workload is represented by the job task load generation cost C proportional to the running time R. Therefore, in conjunction with our equation definition, we reinterpret the specified attribute of the ERP metric as the product of resource consumption and load monetary cost (ECP, Energy Cost Product), and the ECP evaluation metric is given by Equation (3):

$$ECP = E \ * \ C \quad (3)$$

While the single objective of our bi-objective optimization problem is to pursue the reduction and study the behavior of evaluating computational resource planning in terms of scheduling or merging, therefore, combined with our evaluation metric ECP it can be concluded that the smaller the result of ECP, the more cost-effective it is to indicate the performance of the resources among themselves, as shown in the definition of equation (4):

$$\min\,(ECP) \quad (4)$$

For example, in equation (4), if job X has an ECP evaluation value of a in the physical HPC cluster environment and an ECP value of b in the Kubernetes cluster environment, and a > b, then the Kubernetes cluster is selected as the best load cluster for the job to run, but if a = b, indicating that the workload is equally good in both clusters, then a random assignment policy is used to allocate the HPC job tasks.

IV. ECP - PERCEPTION PERCEPTUAL DECISION METHOD

For the two evaluation criteria of resource consumption and monetary cost incurred, we adopt the ECP-aware tradeoff metric, which is the product of energy consumption and load monetary cost (ECP, Energy Cost Product), in pursuit of migrating jobs to cluster environments with relatively lower ECP. We build a multivariate resource demand forecasting model, MRDP (Multivariate Resource Load Forecasting), for a hybrid HPC cluster scenario to forecast the resource load demand of job tasks, and at the same time get the monetary cost required for job tasks to run according to the data center tariff; for job tasks running in a certain cluster, The process of implementing the ECP - Perception-aware decision module is shown in Figure 1.

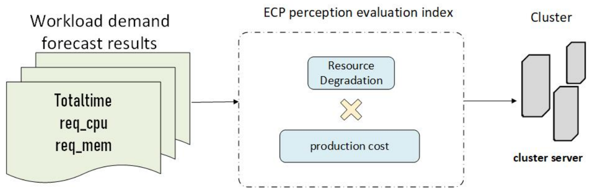

Figure 1. The realization process of perceptual decision function

Job Resource Demand Forecasting: In our previous research, we have constructed a forecasting model MRDP (Multivariate Resource Load Forecasting) for job task resource demand forecasting, which solves a Multiple Input - Multiple Output regression problem. The model gets the prediction results in two environments, the physical HPC cluster and the Kubernetes cluster, based on the job requirements submitted by users. We get the model that combines the neural network algorithm and integration learning in machine learning to build our model; meanwhile, the random forest algorithm as a way of integration learning is used as the second layer of the model, which is integrated and enhanced on the basis of the original integration. The random forest algorithm is also used as a second layer of the model, which is integrated and strengthened on top of the original integrated one so that the model has higher accuracy and stronger fitting ability. We obtained the historical job run logs from the physical HPC cluster and the container-based Kubernetes cluster, and after issuing cleaning and filtering, the physical HPC cluster has 10241 data and the container-based Kubernetes has 10369 data as the training and testing datasets of the model. We compared the model with the MRDP model after missing only CNN, missing only DNN, and missing only LSTM, SVR, and RF for algorithm evaluation to verify the reliability of the MRDP model. Also, we used three evaluation metrics, mean squared error (MSE), mean absolute error (MAE), and coefficient of determination (R^2), as a measure of effectiveness, to take out the final 95% confidence interval training to evaluate the effectiveness of the prediction model, which was shown to have achieved the predicted effect in previous studies and will not be elaborated much in this thesis, and the final evaluation results are shown in TABLE 1.

TABLE 1. EVALUATION RESULTS OF MRDP MODEL AND OTHER METHODS IN VARIOUS CLUSTER ENVIRONMENTS

Physical HPC cluster						
	MRDP	***Lack of LSTM MRDP***	***Lack of CNN MRDP***	***Missing DNN MRDP***	***SVR***	***RF***
MSE	0.0004	0.0264	0.0189	0.0179	0.43356	0.216
MAE	0.0087	0.1374	0.1128	0.1053	0.63781	0.4898
R^2	0.9896	0.8437	0.8655	0.8921	0.7928	0.8234

Kubernetes cluster						
	MRDP	***Lack of LSTM MRDP***	***Lack of CNN MRDP***	***Missing DNN MRDP***	***SVR***	***RF***
MSE	0.0003	0.0218	0.0192	0.0177	0.29281	0.2495
MAE	0.0099	0.1333	0.125	0.12855	0.55257	0.4414
R^2	0.9883	0.8529	0.8662	0.9133	0.8189	0.8658

Resource consumption prediction based on CPU level: our resource energy consumption is oriented to the cluster when the job tasks submitted by users are being executed, and the specific nodes or pods to which the jobs are scheduled when they are executed are determined by the scheduling algorithm of the cluster environment itself, so it is unrealistic to evaluate the energy consumption of a node or a pod in a single way; moreover, in terms of resource consumption of a cluster There is also no clear way to directly calculate the resource energy consumption of a server-based computer cluster, so we use a large number of mathematical models to obtain the estimated resource usage of a clustered environment. The magnitude of the energy used is proportional to the level of CPU consumption, and the literature [22] proposes an experimental evaluation on hundreds of servers, where the model is accurate up to 95%. The authors also proposed a nonlinear energy consumption estimation model given by the equation, as shown in Equation (5), and demonstrated that the accuracy of this nonlinear energy consumption estimation model is up to 99% [23] where P represents the energy consumption usage; U represents the level of CPU usage, indicating the current job task running at the current cluster CPU usage level; where the CPU usage level is given by U, P max and P max denotes the energy usage when the CPU usage is 100% and 0%. r represents the calibration parameter for the minimization of the squared error, which is calculated by previous researchers and yields a value of about 1.4.

$$P = P_{trivial} + (P_{maximum} - P_{trivial}) * \ (2U - U^r) \qquad (5)$$

V. FUNCTIONAL EVALUATION OF PERCEPTUAL DECISION MODULE

To validate the cost impact of our perceptual decision approach on the resources and workloads in the data center and the load incurred, our job migration is arranged as follows: (1) All submitted job tasks run on a physical HPC cluster; (2) All submitted job tasks run on a Kubernetes cluster; and (3) perceptual migration of user submitted job tasks in a hybrid HPC cluster using our ECP - Perception perceptual decision approach.

A. Evaluation Metrics

In the simulated production environment, we observe the operation of the job task load and the resource consumption of the resource data center by using the three job operation environment selection conditions mentioned above. We will evaluate the observation of our perceptual migration strategy from the following perspectives:(1) Resource energy consumed by job tasks during execution - using CPU utilization level as a measure;(2) The economic cost of the workload as it runs.

B. Experimental Environment and Configuration

In this subsection, we explain the experimental environment built to simulate the production environment and the HPC application and job submission model.

We selected and deployed a regular physical HPC cluster built with two physical machines and a Kubernetes cluster built with four virtual machines based on container technology as our reference system based on different interconnections, operating systems, and virtualization support conditions, and conducted experiments on the simulation system between the two clusters using our ECP - Perception-aware decision making approach. Each cluster was deployed with a resource monitor that can monitor the whole cluster. In the constructed Kubernetes cluster, the nodes refer to virtual machines and the number of cores refers to virtual cores, as shown in TABLE 2 for the specific environment description.

TABLE 2. HPC OPERATING ENVIRONMENT DESCRIPTION

	Physical HPC cluster	**K8S cluster**
System Version	X86_64 GNU/Linux	Ubuntu18.04
Processors in a Node	2 × Intel® Xeon® Gold 5218 CPU@ 2.30GHz	4 × Intel® Xeon® Processor
CPU Cores	128	128
Memory	512GB	512GB
Disk	1200GiB	1200GiB
Scheduler	Slurm	Volcano
Monitors	Prometheus	Prometheus
$P_{trivial}$	24.3(KWh)	54.1（KWh）
$P_{maximum}$	258(KWh)	801(KWh)

In terms of HPC application selection, five application programs, LAMMPS, NAMD, WRF, ROMS, and QE, were selected from today's mainstream HPC applications based on the demand characteristics of HPC application scenarios such as computational performance requirements, data load capacity requirements, and communication energy requirements, involving atomic molecular dynamics, weather forecasting models, physical ocean models, electronic structure mechanics, and molecular The five HPC discipline application areas were selected. Each application arithmetic case is deployed on the physical HPC cluster, and the configuration environment and input files of the arithmetic cases are encapsulated into images and mounted in the image repository so that the Kubernetes cluster has the same arithmetic case images as the physical HPC cluster.

In the two cluster environments described above, we install and deploy the above five HPC application cases, and run the above application cases in the two cluster environments separately. Finally, all arithmetic cases are sensed to the target cluster in the hybrid HPC cluster according to the ECP - Perception-aware decision method proposed in this thesis. In order to simulate the working mode of the production environment, this experiment adopts the script submission method to submit the jobs to the cluster centrally, and at the same time, the data characteristics of the job tasks running are obtained through the workload coordinator and the cluster resource monitor, in which the resource monitor deployed in the cluster obtains the resource occupation of the cluster every 8 seconds, and for the data characteristics of the cluster resource situation obtained every 8 seconds interval Through the database matching method, the peak resource consumption during the window from the time of job submission to the time of job completion is taken out to express the resource load on the cluster when running the job; meanwhile, the monetary cost of executing the job is calculated by referring to the HPC cluster charge in Shanhe Supercomputing Platform and the charge of Aliyun Container Service ACK Platform. Costs.

After information filtering and data cleansing, there are 10,106 data under the condition that all job tasks are scheduled to run in a physical HPC cluster environment, 10,208 data under the condition that all job tasks are scheduled to run in a Kubernetes cluster environment, and 10,300 data under the condition that the job tasks are sensitively migrated to a more appropriate cluster environment in a hybrid cluster by the ECP - Perception-aware decision method. 10,300 data under the condition that all the tasks are scheduled to run in the Kubernetes cluster environment.

Some of the data characteristics obtained above resource consumption and monetary cost are characterized by means of data normalization in order to represent the level of resource consumption and monetary cost incurred by each case running at different scales in each cluster environment.

Table 3 shows the resource consumption and monetary cost of each HPC application case under different submission modes. Perception-aware decision method in hybrid clusters for HPC application job tasks, resource consumption is improved, especially for the Kubernetes cluster environment, the virtualization of the environment makes the Kubernetes cluster resource consumption a bit higher compared to the physical HPC cluster, most of the HPC application However, for strongly compute-intensive applications, the performance of running job tasks in a physical HPC cluster environment is better than that in a Kubernetes cluster environment, and as observed in Table 3, the performance of WRF and ROMS, which are two strong computationally intensive applications, have lower energy savings than LAMMPS, NAMD, and QE, which are three weak computationally intensive applications with relatively weak requirements on the computing power of the cluster environment, after perceptual scheduling by ECP - Perception-aware decision method, and the resource consumption savings are less than 5%. The degree of resource consumption saving is not obvious.

In order to verify the resource-saving effect of the ECP-Perception-aware decision method in a real production environment, two application cases with different performances, LAMMPS and WRF, QE, and ROMS, are submitted to run in three job submission modes at the same time, and the resource consumption of the strong computation-intensive application cases and the weak computation-intensive application cases running in the cluster at the same time are compared and observed. We further verify the resource-saving effect of the ECP - Perception-aware decision method when multiple types of application cases are run simultaneously. When application cases of different performances are submitted to run in the cluster at the same time, the energy consumption savings are higher than 10% when compared to running in a physical HPC cluster environment, and higher than 25% when compared to running in a Kubernetes cluster environment, proving that the ECP - Perception-aware decision method produces better resource savings in a real production environment.

Table 4 shows the monetary cost of the five HPC application cases under different submission conditions. For the submission mode of submitting all HPC application jobs to a physical HPC cluster, the monetary cost of perceptual migration of HPC application jobs in a hybrid cluster environment using the ECP - Perception-aware decision method is less However, the monetary cost of perceptually scheduling some application jobs under the ECP-Perception-aware decision method is lower than that of the Kubernetes cluster, but the monetary cost of perceptually scheduling some computationally intensive applications such as ROMS and WRF, which require more computational power, is higher after the ECP-Perception-aware decision method. In terms of economic cost, the monetary cost of these tasks in a hybrid cluster environment after the ECP - Perception-aware migration is less than the monetary cost of deploying them all on a Kubernetes cluster, but focusing only on the monetary cost factor may make ROMS, WRF which are strongly computationally intensive applications with higher computational power requirements, to yield lower operational performance on a Kubernetes cluster than on a physical HPC cluster.

In order to verify the resource-saving effect of the ECP - Perception-aware decision method in a real production environment, two application cases, LAMMPS and WRF, QE and ROMS, which also differ significantly in performance, are submitted to run in three job submission modes simultaneously, and the cost incurred by the strong compute-intensive application cases compared to the weak compute-intensive application cases running in the cluster at the same time is observed. We further verify the cost savings of the ECP - Perception-aware decision method when multiple types of application cases are run simultaneously. When application cases of different performances are submitted to run in the cluster at the same time, the cost savings are higher than 10% when compared to running in a physical HPC cluster environment, and the cost savings are positive when compared to running in a Kubernetes cluster environment, which proves that the ECP - Perception-aware decision method has limited cost savings in a real production environment. The results are positive. Therefore, considering the conditions of resource consumption and performance, the ECP - Perception-aware decision method proposed in this thesis is meaningful for it.

After experimental comparison and analysis, it is proved that the job submission mode of the hybrid cluster environment based on ECP - Perception-aware decision making method proposed in the paper is relatively less resource consuming or costly compared with submitting HPC jobs in a separate physical HPC cluster or a separate Kubernetes cluster. This approach of combining the physical and cloud resources as a supplement to the supercomputer resources brings relatively higher cost performance for HPC users, and also relatively improves the overall load balance and utilization of the HPC operating platform.

Translated with www.DeepL.com/Translator (free version)

TABLE 3 COMPARISON OF CLUSTER ENERGY CONSUMPTION OF HPC APPLICATIONS UNDER DIFFERENT SUBMISSION CONDITIONS

	Submission mode (1) Generates energy consumption	**Submission mode (2) Generates energy consumption**	**Submission mode (3) Generates energy consumption**	**Compare all the energy savings on the physical HPC clusters**	**Compare the energy savings of all Kubernetes clusters**
LAMMPS	456.951	577.002	408.255	10.66%	29.25%
NAMD	437.154	604.886	384.106	12.13%	36.42%
QE	483.602	582.107	400.734	17.14%	31.16%
WRF	560.06	887.32	535.02	4.47%	39.7%
ROMS	549.062	875.263	531.052	3.28%	39.33%
LAMMPS+WRF	679.365	840.329	602.936	11.25%	28.25%
QE+ROMS	642.825	823.207	573.528	10.78%	30.33%

TABLE 4 COMPARISON OF MONETARY COSTS OF HPC APPLICATIONS UNDER DIFFERENT SUBMISSION CONDITIONS

	Delivery mode (1) incurs costs	**Delivery mode (2) incurs costs**	**Delivery mode (3) incurs costs**	**Compare the cost savings on all HPC clusters**	**Compare the cost savings all over the Kubernetes cluster**
LAMMPS	427.772	393.742	381.1625	10.9%	3.25%
NAMD	454.960	431.315	417.833	8.16%	3.16%
QE	801.544	771.082	688.567	14.09%	10.7%
WRF	1073.2	743.541	949.463	11.53%	-27.69%
ROMS	1024.88	747.061	841.125	17.93%	-12.59%
LAMMPS+WRF	1659.76	1507.21	1480.834	10.78%	1.75%
QE+ROMS	1879.14	1660.31	1639.551	12.75%	1.25%

VI. CONCLUSION

In this paper, we investigate the problem of a perception migration strategy for HPC job tasks in hybrid cluster scenarios in the field of high performance computing, in order to facilitate the development of HPC applications in hybrid clusters by minimizing the overall completion time or cost of user tasks while satisfying user requirements, improving the efficiency of cloud resource usage and reducing energy consumption. We introduce a job task-aware migration strategy approach for hybrid HPC cluster environments: ECP - Perception-aware decision method, which can make reasonable resource demand prediction for jobs and migrate them to a cluster environment with relatively lower resource consumption or lower monetary cost to reduce the This reduces the resource consumption or monetary cost of running HPC job tasks. After an experimental comparison, it is confirmed that our approach enables HPC jobs to obtain relatively lower resource consumption or lower monetary cost in a hybrid HPC cluster environment than in a single physical HPC cluster or Kubernetes cluster. Considering the principle that the scale of different clusters in hybrid clusters requires the pursuit of control variables, the production environment is built and simulated by us independently. In the future, we expect that the hybrid HPC cluster environment can be developed more maturely so that our perceptual decision-making method can be applied to larger-scale hybrid cluster environments to provide decision-making for HPC job-aware scheduling solutions.

ACKNOWLEDGMENT

Our research is supported by the Fund for Research, Development and Application of Key Technologies for Autonomous and Controllable Arithmetic Network (2022JBZ01-01) of Qilu University of Technology and the Fund for Research, Development and Application of Key Technologies for Supercomputing Internet (2022CXGC020106) of Shandong Province Key R&D Program.

REFERENCES

[1] Netto M A S, Cunha R L F, Sultanum N. Deciding when and how to move HPC jobs to the cloud[J]. Computer, 2015, 48(11): 86-89.

[2] Gupta A, Kale L V, Gioachin F, et al. The who, what, why, and how of high performance computing in the s1: 306-314.

[3] Ying Guo, Yang Bai. An HPC job cluster awareness method based on mixed cluster:CN202210487700.2[P]. 2022-07-12.

[4] Andresen D, Hsu W, Yang H, et al. Machine learning for predictive analytics of compute cluster jobs[J]. arXiv preprint arXiv:1806.01116, 2018.

[5] Balaprakash P, Tiwari A, Wild S M. Multi objective optimization of HPC kernels for performance, power, and energy[C]//International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems. Springer, Cham, 2014: 239-260.

[6] Netto M A S, Calheiros R N, Rodrigues E R, et al. HPC cloud for scientific and business applications: taxonomy, vision, and research challenges[J]. ACM Computing Surveys (CSUR), 2018, 51(1): 1-29.

[7] Pahl C, Brogi A, Soldani J, et al. Cloud container technologies: a state-of-the-art review[J]. IEEE Transactions on Cloud Computing, 2017, 7(3): 677-692.

[8] Fan Y, Lan Z, Rich P, et al. Hybrid Workload Scheduling on HPC Systems[C]//2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2022: 470-480.

[9] Mu'alem A W, Feitelson D G. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling[J]. IEEE transactions on parallel and distributed systems, 2001, 12(6): 529-543.

[10] ABRISHAMI S, NAGHIBZADEH M. Deadline- constrained work -flow scheduling in software as a service Cloud [J]. Scientia Irani -ca, 2012, 19 (3): 680 – 689.

[11] OPRESCU A M, KIELMANN T. Bag - of - Tasks Scheduling un -der Budget Constraints [C] // IEEE Second International Confe - rence on Cloud Computing Technology and Science. IEEE Com -puter Society, 2010: 351 - 359.

[12] FAN Z Q, SHEN H, WU Y B, et al. Simulated - Annealing Load Balancing for Resource Allocation in Cloud Environments[C] // International Conference on Parallel and Distributed Compu -ting , Applications and Technologies. IEEE ,2014 : 1 - 6.

[13] ARON R, CHANA I. Bacterial for aging based hyper – heuristic for resource scheduling in grid computing [J]. Future Generation Computer Systems , 2013 , 29 (3): 751 – 762.

[14] Nigro, Libero. "Parallel Theatre: An actor framework in Java for high performance computing." Simulation Modelling Practice and Theory 106 (2021): 102189.

[15] Arunarani, AR, Dhanabalachandran Manjula, and Vijayan Sugumaran. "Task scheduling techniques in cloud computing: A literature survey." Future Generation Computer Systems 91 (2019): 407-415.

[16] Khan A A, Zakarya M, Rahman I U, et al. HeporCloud: An energy and performance efficient resource orchestrator for hybrid heterogeneous cloud computing environments[J]. Journal of Network and Computer Applications, 2021, 173: 102869.

[17] Zakarya M. Energy and performance aware resource management in heterogeneous cloud datacenters[M]. University of Surrey (United Kingdom), 2017.

[18] Ferreto T C, Netto M A S, Calheiros R N, et al. Server consolidation with migration control for virtualized data centers[J]. Future Generation Computer Systems, 2011, 27(8): 1027-1034.

[19] Gupta V. Stochastic models and analysis for resource management in server farms[D]. Carnegie Mellon University, 2011.

[20] Gandhi A, Gupta V, Harchol-Balter M, et al. Energy-efficient dynamic capacity provisioning in server farms[J]. School of Computer Science, Carnegie Mellon University, Tech. Rep. CMU-CS-10-108, 2010.

[21] Zakarya M, Gillam L. Managing energy, performance and cost in large scale heterogeneous datacenters using migrations[J]. Future Generation Computer Systems, 2019, 93: 529-547.

[22] Fan X, Weber W D, Barroso L A. Power provisioning for a warehouse-sized computer[J]. ACM SIGARCH computer architecture news, 2007, 35(2): 13-23.

[23] Dayarathna M, Wen Y, Fan R. Data center energy consumption modeling: A survey[J]. IEEE Communications Surveys & Tutorials, 2015, 18(1): 732-794.