Research and Application of Energy Efficiency Optimization Algorithm for Spacecraft Simulation Platform

Zhou An(✉), Yi Yuan, Xun Zhou, Wenlong Song, Qi Miao, and Huifang Pan

Beijing Institute of Spacecraft System Engineering, Beijing 100094, China
anzhou163@163.com

Abstract. High performance computing plays an increasingly important role in the simulation and verification of the aerospace. The high-performance simulation and computing platform for spacecraft provides effective support for various professional fields such as spacecraft orbit design, mechanical structure analysis, electromagnetic simulation, etc. With the increasing scale and quantity of spacecraft, all kinds of simulation tasks tend to be complex and require high solving time, which puts forward higher requirements for the computing capability of the platform. While improving the computing power of the platform, its performance and scale are growing year by year, and its energy consumption is also growing synchronously. This paper combines the characteristics of various professional computing tasks in spacecraft simulation, experimental verifies with the actual operation data, analyzes the platform energy consumption model, and compares the optimized scheduling algorithm with the conventional scheduling algorithm. On the premise of not affecting the platform throughput and operation time, the overall energy consumption of the platform is reduced by an average of 25%, effectively improving the energy efficiency of the platform and saving the operation cost.

Keywords: Spacecraft Design · Simulation Calculation · Energy Efficiency Optimization · Job Scheduling

1 Introduction

High performance computing technology has been widely used in meteorology, energy, medical, industrial simulation and other fields. It is an important symbol of a country's scientific-technical development level and comprehensive national strength. Among them, high-performance computing plays an increasingly important role in the simulation and verification of the aerospace field. The high-performance simulation and computing platform for spacecraft (hereinafter referred to as the platform) provides effective support for various professional fields, such as spacecraft orbit design, mechanical structure analysis, electromagnetic simulation, etc. In the aerodynamic prediction and verification of Tianwen-1, the specific experiments of aerodynamics and aerothermodynamics

R. Li et al. (Eds.): MOBILWARE 2022, LNICST 507, pp. 165–176, 2023.
https://doi.org/10.1007/978-3-031-34497-8_14

were carried out in the Martian atmosphere during the Mars Entry, Descent and Landing (EDL) mission [1]. In the Chang'e-5 mission, an expanded mission orbit scheme was designed for the orbiter, and the trajectory design dynamics model simulation was carried out to maximize the use of mission resources [2]. The landing simulation of the return capsule of a new generation manned spacecraft, the airbag finite element model is established to simulate the landing buffer process of the return capsule, effectively reducing the possibility of hard landing [3].

With the increasing scale and quantity of spacecraft, all kinds of simulation tasks tend to be complex and require high solving time, which puts forward higher requirements for the computing capability of the platform. While improving the computing power of the platform, its performance and scale are growing year by year, and its energy consumption is also growing synchronously. The power consumption of China's data centers has increased at a rate of more than 12% for eight consecutive years. In 2018, the total power consumption of China's data centers accounted for 2.35% of the total social power consumption [4]. IBM and the US Energy Administration have identified the top ten challenges in building exascale supercomputer system as energy efficiency [5, 6]. In May 2022, the Frontier Supercomputer located in the Oak Ridge National Laboratory (ORNL) of the United States achieved 1.102Exaflop/s of actual test computing capacity in the Top 500 ranking, becoming the world's first exascale supercomputer. At the same time, the Frontier Supercomputer ranked first in the Green 500 with an extremely high energy efficiency ratio of 52.23 gigaflops per watt [7].

To improve the energy efficiency ratio of the data center is to balance the energy consumption and performance of the platform. The biggest challenge is to reduce the energy consumption of the platform on the premise of ensuring the overall throughput and quality of service of the platform. In the high-performance computing data center, the energy consumption of information technology (IT) equipment mainly comes from the operation of jobs on computing nodes. Different scheduling strategies make the distribution of jobs on computing nodes different. For servers without computing tasks in idle state, a large amount of energy consumption is wasted. A large part of the platform's energy consumption comes from the vacancy rate of computing nodes. How to optimize the job scheduling strategy on the premise of ensuring the platform throughput and computing efficiency is of great significance to improve the energy efficiency of the platform. In this paper, by analyzing the computing characteristics of spacecraft simulation tasks, modeling the platform operation distribution, resource utilization and energy consumption, then proposing a scheduling algorithm based on energy efficiency optimization, which effectively reduces the platform energy consumption and carbon emissions.

2 Related Work

With the rapid growth of energy consumption in data centers, the problem of high energy consumption has aroused widespread concern. The data center energy efficiency standard commonly used in the industry is mainly Power Usage Effectiveness (PUE) [8]. The ratio of the total energy consumption of the data center to the total energy consumption of IT equipment, it is used to measure the effective energy consumption of the data center.

Technologies for energy consumption optimization of IT equipment include dynamic voltage and frequency scaling (DVFS) and dynamic power management (DPM) [9]. DVFS reduces the power consumption and performance of the processor by reducing the frequency and voltage of the processor. DPM technology dynamically configures components according to the running status of processors, memory and other components, and provides the minimum number of moving components to reduce energy consumption.

Etinski et al. [10] based on the integer linear equation of power configuration, proposed a scheduling strategy using DVFS to dynamically adjust the power consumption of computing nodes to control the power consumption of the entire platform. Although the overall power consumption is effectively reduced, the job execution time is also increased.

Yang et al. [11] proposed a cloud infrastructure that monitors the status of the OpenStack platform and the real-time status of the virtual machine on it. The monitoring indicators include CPU utilization, memory load, energy consumption, and so on. Energy consumption can be saved through online migration of virtual machines. The cloud service of the data center strengthens the centralized management of resources, it through the integrated management of various resources and the distributed self-organization mechanism based on game theory, making decisions on dynamic resource calls with less delay, so as to reduce the operation power consumption [12].

The energy consumption of the data center mainly comes from the operating load of IT equipment and the operating consumption of refrigeration equipment [13]. Among them, 15% - 30% of servers in IT equipment are idle, and the energy consumption in idle state accounts for 50% - 60% of the full load energy consumption [14]. The server runs with the highest energy efficiency under full load and the energy efficiency drops sharply when the load decreases. For the common server utilization rate of 20% - 30%, its energy efficiency is less than 50%.

3 Energy Consumption Model Analysis

In the spacecraft simulation computing platform, computing nodes account for more than 90% of IT equipment. This paper mainly focuses on how to improve the energy efficiency ratio of computing nodes, thereby reducing the overall energy consumption of the platform.

In the process of spacecraft design simulation, due to the large scale of calculation model, the large number of calculation conditions and iterations, and the frequent parallel of multiple models, it is necessary to quickly respond to simulation tasks and modify design parameters. A satellite finite element model can reach the scale of 250000 nodes and 200000 units. The satellite static analysis condition and dynamic analysis condition need to carry out multiple iterative calculations. Deep space exploration, manned spaceflight, communication and navigation satellites and research projects are need to be analyzed and designed in parallel. The computing capacity of the platform is required to be high. At the same time, the model task has certain periodicity, which leads to the platform resources cannot be fully utilized in real time. When the platform resources cannot be used 100%, some computing nodes are idle, resulting in waste of energy consumption.

Assume that the number of platform nodes is N, the number of CPU cores per node is K, and the platform CPU resource is N × K. To simplify the model, set the node status as: idle, busy. The idle means that the node has no computing job running. The busy means that there is more than one calculation job running on this node. The system resources are divided by time dimension. The total time of platform resources is:

$$\mathrm{T} = \sum_{i=1}^{N} t_i^{idle} + t_i^{busy} \tag{1}$$

The t_i^{idle} represents the time when the node i is idle and the t_i^{busy} represents the time when the node i is busy.

The platform resource utilization rate (UR) is the ratio of the sum of nodes running calculation jobs to all resources of the platform.

$$\mathrm{UR} = \frac{\sum_{i=1}^{N} t_i^{busy}}{\sum_{i=1}^{N} t_i^{idle} + t_i^{busy}} \tag{2}$$

If no energy saving measures are taken, all nodes in idle status are in running status. The platform power consumption is:

$$\mathrm{E}_{IT} = \sum_{i=1}^{N} t_i^{idle} \times e_i^{idle} + t_i^{busy} \times e_i^{busy} \tag{3}$$

The e_i^{idle} represents the power consumption when the node i is idle, and the e_i^{busy} represents the power consumption when the node i is busy. For node power consumption in idle state, it is assumed that the power consumption is the same when the node types are the same.

For a computing node in idle state, it maintains the same level of power consumption as the normal operation state. That result a waste of energy. How to reduce the use of computing nodes and improve the utilization of a single computing node, without job scheduling and running time be affected. It is a breakthrough to improve the overall energy efficiency of the platform.

4 Algorithm Design and Implementation

Resource management and scheduling are the core components of high-performance computing platforms. A scheduling algorithm based on energy efficiency optimization is designed. Through reasonable resource scheduling and allocation, the power consumption of computing nodes is reduced while avoiding frequent node state transitions, ensuring platform throughput and efficient operation of jobs. If the node sleeps and wakes up frequently, it may lead to hardware failure, which will affect the stability of the platform. And it is not conducive to improving the energy efficiency of the platform.

The computing nodes Node $= \{N_1, N_2, \ldots, N_i\}$. The logical partitions Partition $= \{P_1, P_2, \ldots, P_j\}$, it is used to logically isolate computing nodes. The partition also provides targeted computing services such as node limit, CPU memory resource limit, and job number limit. When submitting a calculation job, it needs to be specified a partition. The job can be submitted to the partition according to the scheduling strategy. First, the platform match resources required by the job to available resources of the partition, and then select node to run the job (see Fig. 1).

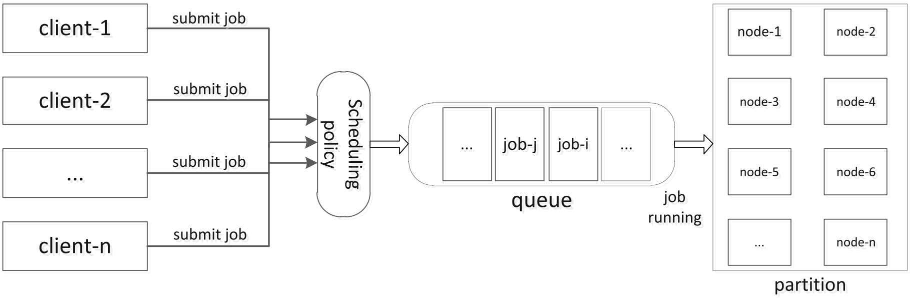

Fig. 1. Operation process of job submission

The scheduling algorithm based on energy efficiency optimization mainly focuses on the resource scheduling level of partition computing nodes. First, the computing nodes in the partition are numbered. In the process of job scheduling, the computing nodes with the highest number are preferred. Then initialize the computing nodes in the partition. All nodes are set to the running state and can receive jobs.

```
Algorithm 1: Job scheduling algorithm
queue=list(job-1,job-2,...job-n);
partition=list(node-1,node-2,...node-n)
allocateNode=list();
isSatisfy=false;
i=1;
while i<n
  job=queueList(i);Get the current task
  nodeNum=job.getNode();Get the number of nodes required for the job
  cpuNum=job.getCpu();Get the number of CPU cores required by the job
  for(j=1;j<n;j++)
    node=partitionList(j);Get the computing node j in the partition
    if node.status == busy || node.status == idle
        spareCpu=node.getSpareCpu();Get the number of available CPU cores of the computing node j
        if cpuNum<spareCpu
            allocateNode.add(job,node);
        if allocateNode.getJobNode>nodeNum
          isSatisfy=true;
          break;
  if isSatisfy
    job.run(allocateNode);Run job i on the acquired node
    else
    for(j=1;j<n;j++)
    node=partitionList(j);Get the computing node j in the partition
    if node.status==sleep
        node.start();
        node.status=idle;
        spareCpu=node.getSpareCpu();Get the number of available CPU cores of the computing node j
        if cpuNum<spareCpu
            allocateNode.add(job,node);
```

The strategy can be used combination with traditional scheduling strategies. First, the platform can receive submitted jobs in real time. Second according to certain strategies, it selects the first node in the partition that meets resource requirements to the selected job. This can effectively improve the utilization of computing nodes and optimize the energy efficiency of every single computing node.

```
Algorithm 2: Partition node energy consumption optimization algorithm
partition=list(node-1,node-2,...node-n);
i=1;
while i<n
  node=partitionlist(i);Get partition computing node
  if node.status==busy
    continue;
  if now-node.jobendtime>24
    node.status=sleep;
node.sleep();
```

In the process of resource scheduling, it is necessary to uniformly manage resource state switching, dynamically configure switching opportunities and conditions. It combines maintenance with scheduling strategies and energy consumption management requirements (see Fig. 2).

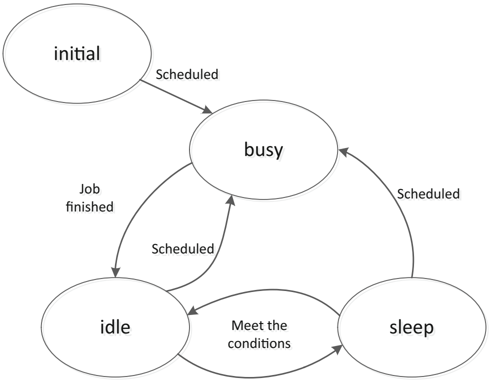

Fig. 2. Computing node resource state switch

1) When the partition computing node is idle for more than a certain period of time (such as 24 h), switch the node to the sleep state. That reduce the overall power consumption of the platform and improve the energy efficiency ratio.
2) When the node switches to the sleep state, configure the number of simultaneous switching nodes. That can avoid a large number of computing node simultaneous be sleep.
3) When the number of queued jobs in the partition exceeds a certain number (such as 5), the dormant node will be switched to the running state, and the number of switched nodes will be controlled to avoid the impact of energy consumption caused by a large number of node state switching.

In the process of simulation job scheduling on the platform, it is necessary to ensure that the job can obtain enough computing resources. Through saving the running time of a job, it can reduce energy consumption required by the job; On the other hand, by sleeping the idle nodes to reduce the number of running computing nodes, and then it will reduce the overall energy consumption of the platform.

When scheduling computing node resources, the running computing nodes should have high priority. So that the least recently used computing nodes are finally allocated, while reducing the frequent sleep and wake-up of resources. It also can improve the reliability of the platform.

Input: Job resource requirements (number of nodes, number of CPU cores), node sets.

Output: assigning node sets.

1) The initialization allocation node set is empty;

2) Select the running nodes from the node set in order;
3) Judge whether the number of available CPU cores of the selected node is greater than the number of CPU cores required by the job. If it is greater than the number, it will be placed in the allocation node set;
4) Judge whether the number of nodes in the allocation node set is greater than the number of job demand nodes, and return to the allocation node set if greater; Otherwise, repeat steps 3) and 4).

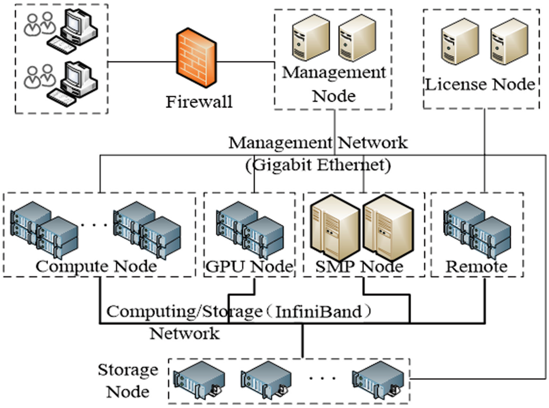

Fig. 3. The platform architecture

5 Simulation Experiment and Analysis

On the spacecraft simulation computing platform, we submit job for experiment verification. The platform architecture is shown in the Fig. 3.

5.1 Running Environment

The platform uses Slurm as a resource management and scheduling tool. Scheduling strategy uses first-come-first-service (FCFS). Computing nodes are composed of 20 blade servers. Computing networks are connected by IB switches. Key indicators of nodes are shown in the Table 1.

Table 1. Key indicators of nodes.

Name	Indicators
Type	Blade server
OS	CentOS7.6
CPU	2*24 Intel Xeon Gold 6240R 2.4GHz
Memory	12*16GB DDR4 2933MHz
IB Network	1*2 EDR 100Gb/s InfiniBand HCA Card

5.2 Data Acquisition

Data collection uses out-of-band monitoring and in-band monitoring. The out-of-band monitoring uses Intelligent Platform Management Interface (IMPI), and the in-band monitoring is based on an open source monitoring tool Nagios.

The real-time power of each computing node is obtained based on the IMPITOOL tool, and the power consumption data are summarized to the management node. Key services of Nagios monitoring tools include icinga and nrpe. The nrpe services deployed on the computing nodes is to monitor the usage of CPU and memory resources. It also can monitor the operation of job process services. The nrpe services transfer monitoring indicator data to the icinga services deployed on management nodes. The management node configures the relational database MariaDB and the sequential database InfluxDB, which annotates the collection time for all collected data and facilitates the association analysis.

Set collection frequency to 1m for the monitoring data. The collection index and data style are shown in the Table 2.

Table 2. Sample of Energy Consumption Data.

Name	IP	Power(W)	CPU(%)	Memory(%)	Time
node1	30.10.10.1	405	68.2	32	2022–03-20 12:28:10
node2	30.10.10.2	468	93.3	23.2	2022–03-20 12:28:10
node3	30.10.10.3	474	88.7	19.7	2022–03-20 12:28:10
node4	30.10.10.4	436	82.7	15.3	2022–03-20 12:28:10
…	…	…	…	…	…
node20	30.10.10.20	243	1	4.8	2022–03-20 12:28:10

5.3 Experimental Settings

Job selection: From the structure, fluid, heat, electromagnetic simulation calculation, select typical application examples to build a real simulation job sequence.

Platform Load Settings: In order to comprehensively analyze the energy consumption comparison of the optimization algorithm in the platform simulation environment, different platform loads are controlled by the number of jobs. On this basis, the CPU usage and power consumption of nodes are counted according to the time distribution, and the CPU usage and energy consumption distribution under different load conditions are analyzed.

Job Scheduling Strategy: A comparative test of FCFS scheduling strategy and scheduling strategy based on energy efficiency optimization.

5.4 Result Analysis

In the actual simulation calculation job, three jobs are selected from the specialties of structure, fluid, heat and electromagnetism to form 12 different size and professional operation sequences, which cover the routine calculation tasks. During the testing process, control the number of running jobs and the parallel core of each job to obtain different CPU utilization of nodes.

Statistical analysis is conducted for the empty load and the load of 30%, 60% and 90% respectively, lasting for 13 h. The CPU utilization changes with time are shown in the figure. Under the corresponding load, the power consumption of the computing node is monitored in real time for 13 h. The power consumption changes with time as shown in the figure. Through comparative analysis of CPU utilization and power consumption over time, it is found that CPU utilization is positively related to power consumption, and power consumption increases with the increase of CPU utilization. When the computing node is empty, the power consumption is about 240W. When the computing node is 90% loaded, the power consumption is about 470W. The empty load power consumption of computing nodes is half of the full load power consumption. When the overall load of the platform is low, the proportion of idle node power consumption will be greatly increased. It is very important to reduce the empty load ratio of computing nodes to improve the energy efficiency ratio of the platform (Fig. 4).

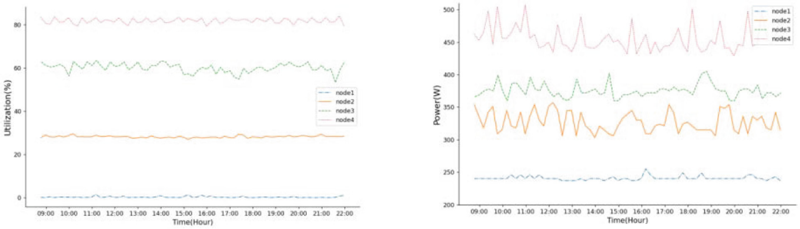

Fig. 4. Changes in CPU and memory utilization of computing nodes under different loads

Under the condition that the running jobs are consistent, the scheduling algorithm based on energy efficiency optimization is compared with the FCFS scheduling algorithm. The total power consumption is counted under different platform loads. The load and power consumption changes of computing nodes are shown in the Fig. 5.

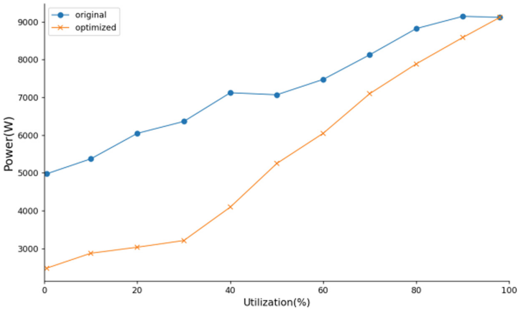

Fig. 5. The load and power consumption changes of computing nodes under different platform loads.

The scheduling algorithm based on energy efficiency optimization is compared with the FCFS algorithm, and the power consumption is reduced by 25% on average. When the overall load of the platform is low, the power consumption is reduced by 50%, and the energy-saving effect is obvious. As the overall load of the platform increases, the rate of power reduction continues to decline. Under full load, the power consumption of the two algorithms tends to be the same.

At the same time, in order to avoid the impact of the scheduling algorithm based on energy efficiency optimization on the platform throughput and job running time, the idle state computing node is reserved in the scheduling algorithm to ensure that the job response time is equal to the conventional scheduling algorithm.

6 Conclusion

This paper analysis the characteristics of spacecraft simulation based on the spacecraft simulation computing platform. Collecting the actual operation data and analyze the platform energy consumption model. On the premise of not affecting the platform throughput and operation time, compare the optimized scheduling algorithm with the conventional scheduling algorithm, the overall energy consumption of the platform is reduced by an average of 25%. The algorithm effective improving the energy efficiency and reducing the cost. Later, we can further analyze simulation computing tasks based on historical job operation data, develop timing analysis and prediction based on machine learning algorithms, to make it more energy efficiency.

References

1. Li, Q., Rao, W., Cheng, X., Wei, H., Wang, C., Dong, J.: Aerodynamic design, analysis, and validation techniques for the Tianwen-1 entry module. Astrodynamics **6**(1), 39–52 (2022). https://doi.org/10.1007/s42064-021-0123-z
2. Liu, L., Liu, Y., Chen, M., et al.: Trajectory schemes of chang'e-5 extended missions to liberation points. J. Astronaut. Astronaut. **43**(3), 293–300 (2022)
3. Li, J., Zhang, C., Xing, W.: Numerical simulation studies of the cushion process for the new generation of manned spacecraft re-entry capsule. Aerosp. Return Remote Sens. **42**(2), 12–19 (2021)

4. Lin, H., Peng, Y., Xia, Y., et al.: Summary of energy consumption evaluation method standard for China's data center. Energy China. **42**(8), 36–39 (2020)
5. Agerwala, T.: Challenges on the road to exascale computing. In: Proceedings of the 22nd Annual International Conference on Supercomputing, vol. 2, (2008)
6. Lucas, R., Ang, J., Bergman, K., et al.: Top ten exascale research challenges. DOE ASCAC subcommittee report, pp. 1–86 (2014)
7. TOP 500 list. https://www.top500.org/. Accessed June 2022
8. Jaureguialzo, E.: PUE: the green grid metric for evaluating the energy efficiency in DC (Data Center). Measurement method using the power demand. In: 2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC), pp. 1–8. IEEE (2011)
9. Chen, G., Huang, K., Knoll, A.: Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination. ACM Trans. Embed. Comput. Syst. (TECS) **13**(3s), 1–21 (2014)
10. Etinski, M., Corbalan, J., Labarta, J., et al.: Parallel job scheduling for power constrained HPC systems. Parallel Comput. **38**(12), 615–630 (2012)
11. Yang, C.-T., Wan, T.-Y.: Implementation of an energy saving cloud infrastructure with virtual machine power usage monitoring and live migration on OpenStack. Computing **102**(6), 1547–1566 (2020). https://doi.org/10.1007/s00607-020-00808-7
12. Khani, H., Yazdani, N., Mohammadi, S.: Power-aware game for cloud computing: a distributed mechanism based on game theory for minmizing power consumption in cloud scale datacenter. In: 6th International Symposium on Telecommunications (IST), pp. 598–601. IEEE (2012)
13. Samadiani, E., Joshi, Y.: Energy efficient thermal management of data centers via open multi-scale design: a review of research questions and approaches. J. Enhanc. Heat Transf. **18**(1), 15–30 (2011). https://doi.org/10.1615/JEnhHeatTransf.v18.i1.20
14. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. Computer **40**(12), 33–37 (2007)