Design of Aerospace Cloud Computing Server Based on Docker Cluster

Zhenhui Dong[1(✉)], Luyuan Wang[1], Bowen Cheng[1], Zhihong Xu[1], and Chaoji Chen[1,2]

[1] Beijing Institute of Spacecraft System Engineering, Beijing 100094, China
564760683@qq.com

[2] School of Mechanical Engineering, Tongji University, Shanghai 201804, China

Abstract. To solve the problems of different space-based application platforms, lack of unified specifications and poor real-time performance, this paper proposes a design scheme of space-based cloud server based on Docker cluster, drawing on the container technology and cloud native architecture used by ground cloud computing. The computing module of the space-based cloud server realizes network communication through the route switching module, and integrates a high-capacity storage module to complete data access. The real-time operating system running integrated container services on the computing module and the lightweight container cluster management framework realize the dynamic deployment and orchestration of space-based container applications. The design framework has been verified in the space-based cloud server prototype, and the test results show that the framework achieves universal high-performance computing, virtual resource integration scheduling, and autonomous task migration. In the future, it can be deployed to space-based satellite clusters to provide users with space-based cloud information services, including universal high-performance computing, information intelligent processing, etc.

Keywords: Docker · Cluster · Cloud Computing · Server

1 Introduction

After years of construction and development, although the ground-based information acquisition, transmission, processing and application systems with professional applications as the main goal have been initially built, these systems are distributed in different regions, belong to different management departments, and have different development platforms and lack of unified specifications, forming a pattern of "chimney development", resulting in low efficiency of resource use, difficulties in information sharing, poor timeliness of task response, etc. It restricts the full play of the resource efficiency of the entire space-based information system.

With the continuous deepening of the space-based network demonstration, the space-based network information system with the ability of "super computing, fast storage, massive transmission and flexible reconstruction" is the core system that needs technical

R. Li et al. (Eds.): MOBILWARE 2022, LNICST 507, pp. 125–131, 2023.
https://doi.org/10.1007/978-3-031-34497-8_11

breakthrough. At the same time, in recent years, research institutions in various countries have carried out research on space-based high-performance computing technology, gradually exploring subversive computing methods, and steadily improving the computing efficiency of spacecraft. Under the background that satellite network has become a global hotspot, new technical requirements such as integrated satellite-terrestrial, satellite Internet, artificial intelligence applications, and on-board information processing pose serious challenges to satellite electronic information systems, which are facing a profound technological transformation. The future space-based network satellite information service system needs to break through the traditional architecture, and realize the leapfrog development of system capabilities and the fundamental transformation of product model through a new architecture design.

Cloud computing system, with its good universality, scalability and high reliability, provides a strong technical support for improving the service capability of spatial information service system [1]. In cloud computing, software and hardware are virtualized into logical resources. No matter what kind of device the terminal is, it can access services on demand through the network without paying attention to its internal structure and operation mode. The emergence of cloud computing technology provides a new idea for the implementation of spatial information services. The "space-based cloud" can directly provide service guarantee for users, and can also be used as a supplement and enhancement to the "ground-based cloud" to make up for the shortcomings of the "ground-based cloud" in terms of coverage, emergency support and mobile support capabilities.

2 Container Technology and Cloud Native Architecture

Virtualization refers to simulating physical hardware resources into multiple logical devices through software, and programs can run on independent logical devices without affecting each other. In virtualization technology, Docker plays an important role as an open-source container engine. Compared with traditional virtual machines, Docker containers are relatively lightweight. Docker daemons can directly communicate with the main control system, allocate resources to containers, and isolate containers from the main system and containers from each other [2]. Customized and integrated Docker operating system has the following advantages: 1) small size, fast startup; 2) The CPU does not need to support virtualization, and the hardware requirements are low; 3) Easy migration: Docker's image includes a complete environment that supports application operation, ensuring the consistency of the operating environment and facilitating migration; 4) Isolation: Docker applications are independent of the underlying infrastructure.

Cloud native technology is a collection of a series of technologies, including containers, container orchestration, microservices, DevOps and intelligent operation and maintenance technologies, among which containers are the core technologies of cloud native infrastructure [3, 4]. Cloud native is designed to deploy to the cloud. To ensure the successful operation of container applications, the operating system needs to support relevant services. Cloud Native widely adopts the Docker + Kubernetes technical architecture. Users can create application images based on needs and run them directly on the cloud.

The space-based cloud native technology adopts an open computing-storage-network architecture, and is based on the advantages of high-speed interconnection, space-time benchmark, and multi satellite global coverage of space-based networks to build a highly reliable and high-performance cloud native computing system architecture. Carry out system business analysis and function definition, have the ability of modularization, scalability, plug and play, flexible reconfiguration, realize the sharing and on-demand allocation of computing, storage, network resources, support the dynamic migration of tasks, the migration process is uninterrupted, users are not aware of it, the single machine in a single satellite processes the task load balance, and the satellite in a constellation processes the task load balance.

In view of the requirements for information security, fault isolation and recovery, reliability and real-time of the future space-based information service system, the physical hardware resources are simulated as multiple logical devices through container virtualization technology, and programs can run on mutually independent logical devices without mutual influence, thus providing a lightweight isolation environment for software development, deployment and operation. Based on the lightweight container management platform, the integration of container application development and operation and maintenance is realized, and the intelligent model is rapidly deployed to the space-based network environment. When a computing node in the system fails, the container instance will be automatically scheduled to other redundant nodes to ensure the continuous operation of the business.

3 Design of Space-Based Cloud Computing Server

The space-based cloud native high-performance processing platform system requires a high degree of intelligence, networking and systematization. The traditional space-based information system cannot meet the requirements in terms of computing capacity, storage capacity, communication speed and distributed collaboration. In order to achieve high-speed transmission, intelligent processing, mass storage and other space-based distributed cloud information services, the space-based cloud computing server is designed based on the SpaceVPX standard. Based on the SpaceVPX standard, the hardware modules with different functions are assembled to form a combined application product. The hardware modules are connected through backplane buses at different levels, so as to achieve a high degree of flexibility and scalability in the information interaction between the modules in the whole machine. The prototype of the space-based cloud server is internally integrated with high-performance computing module, high-capacity storage module and routing switching module, realizing the integration of computing, storage and network resources.

The layered design of the Docker based cloud server system is shown in Fig. 1. It consists of five elements: high-performance computing module, customized lightweight on-board operating system, container cluster management middleware, container application, and Rancher monitoring system. After integrating the virtualization container technology on the customized lightweight operating system and realizing the application service containerization, the container applications can be scheduled and deployed through the container cluster management middleware, so as to achieve the resource

virtualization and balanced scheduling of multiple physical computing units in the satellite, provide a unified standard virtual operating environment for various users and applications, and achieve on-demand resource allocation and elastic computing.

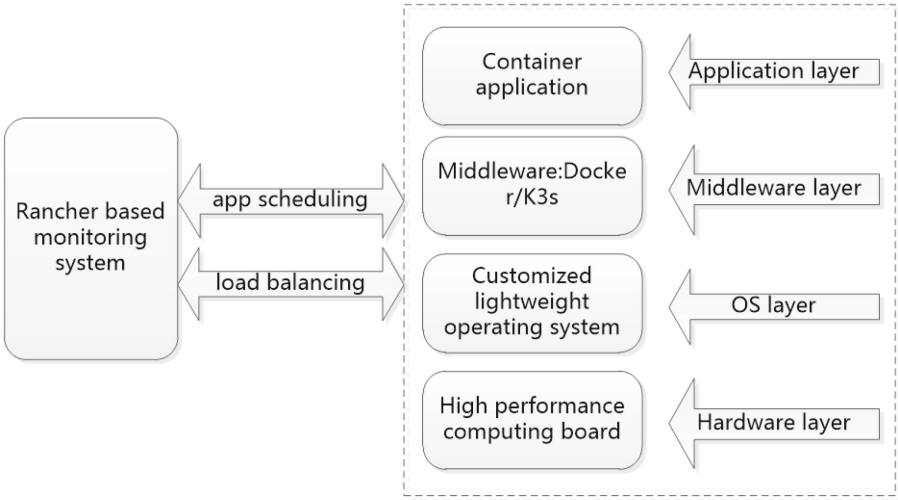

Fig. 1. Layered design of space-based cloud server system

3.1 Design of Real-Time Operating System for Integrated Container

The traditional embedded real-time operating systems such as VxWorks and RTEMS are not completely suitable for the application scenarios of space-based cloud services after being customized. The reason is that space-based cloud services require the operating system to support virtualization, cloud native, AI and other technologies in addition to the high real-time requirements of the operating system. However, the early embedded real-time operating systems in the aerospace field have limited functions and poor software ecology. At present, Linux operating system is widely used in cloud computing servers on the ground. Although it is not a hard real-time operating system, the real-time performance can be greatly improved to microseconds by inserting RT-patch into the Linux kernel, which can meet the requirements of real-time processing on the satellite.

Integrating Docker services into the Linux operating system depends on the Yocto project [5], which requires the support of multiple components such as meta-virtualization, meta-networking, meta-filesystems, and meta-python. The above components can be obtained from the Internet and added to bblayers.conf file through the bitbake-layers add-layer command. After that, the local.conf file is configured and the system is compiled during the system configuration. After the compilation is successful, the QEMU tool integrated by Yocto can be used for simulation testing to verify the availability of Docker services. It should be noted that the Linux kernel layer needs to be customized to support Docker to work in bridge mode. The Docker image created after the kernel configuration is completed can work normally.

3.2 K3s Based Container Cluster Management

The onboard operating system integrating Docker obtains the application image from the Docker's official public repository by default. Due to the special operating environment of the onboard computer, it cannot connect to the public repository, so you need to configure a private image source. Select a node in the cluster as the private repository of the container image. The user uploads the image of the satellite application to the private warehouse. When deploying K3s, specify each node in the cluster to obtain the image from the private repository.

K3s is a lightweight cloud native framework based on Kubernetes [6]. A single node can be selected as the single node architecture of the server side. The K3s cluster of the single node architecture can theoretically meet all the task requirements. The architecture is shown in Fig. 2. In a single node architecture, the cluster has only one K3s server node and embedded SQLite database. Each agent node is registered to the same server node. The administrator of the K3s cluster can directly allocate all node resources in the cluster through the cluster API functions on the K3s server node.

Deploying K3s in the on-board computer environment can facilitate the management of container applications on multiple computing cells to achieve load balancing and resource integration among computing cells, thus further realizing multi cell collaborative computing. In addition, K3s can realize the migration of cluster content application, so as to ensure the reliable operation of applications in the K3s cluster.

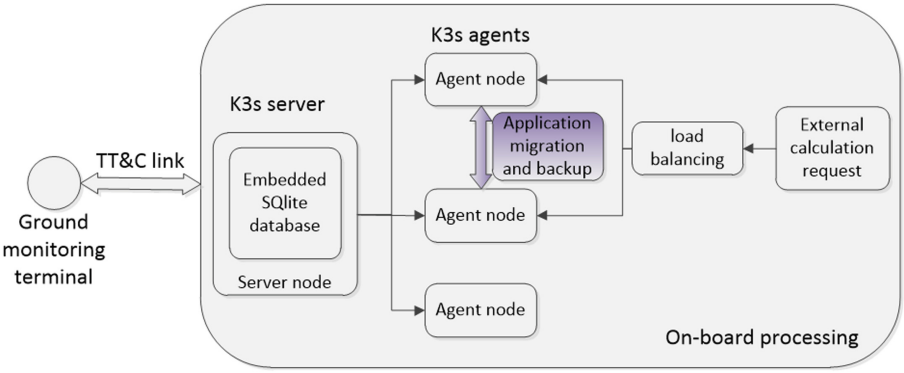

Fig. 2. K3s based container cluster management

4 Instance Verification

On the prototype of the space-based cloud server, the K3s based container cluster management is verified by an example. In order to facilitate the deployment and management of containers, the container management platform Rancher is installed on the space-based cloud servicer. The Rancher Server can be deployed in one of the high-performance computing modules. The ground-based monitoring terminal can intuitively monitor the health status and capacity of the container cluster through the browser, and perform elastic resource allocation.

Three high-performance computing modules are deployed in the prototype computer. One of them is selected as the server node of K3s, and the other two are selected as agent nodes (agent1 and agent2 respectively). The Rancher Server is deployed on the server node. Figure 3 shows the dynamic migration management of container applications in cloud computing services. In the case test process, after the initial deployment of 10 container applications, the three nodes ran multiple container applications according to the equilibrium strategy. The agent2 node was manually isolated from the system. The container applications originally running on the agent2 node were automatically migrated to the server node and agent1 node. After the agent2 node is rejoined to the system, after deleting multiple container applications from the server node and agent1 node, some of the deleted container applications will be automatically migrated and deployed to agent2 node according to the load balancing policy.

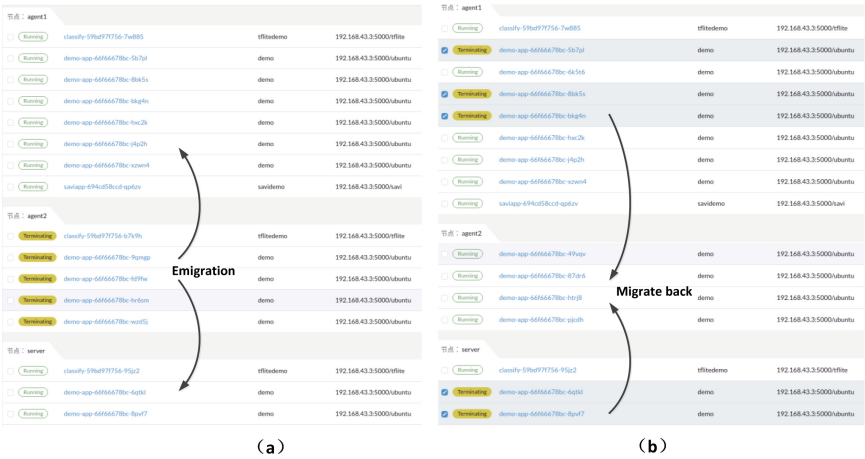

Fig. 3. Dynamic migration management of container applications

5 Conclusions

To improve the business carrying capacity of on-board computer, on the one hand, it depends on the improvement of hardware computing power, and on the other hand, it needs the support of on-board computer application software. The current on-board application software has some compatibility problems when running on cross platform systems. At the same time, the special software ecological environment of on-board computers adds inconvenience to software deployment. Running the application software on the on-board computer in the form of a container can not only solve the compatibility problem during software deployment, but also improve the security of on-board applications through resource isolation between containers. At the same time, the container cluster management method is used to deploy satellite applications on multiple computing cells. The container application can be deployed on multiple computing cells with

one click only once. After the application of container technology, the development and maintenance efficiency of on-board software will be greatly improved.

Acknowledgment. This research is sponsored by "National Key R&D Program of China" with its research number:2022YFB2902700.

References

1. Straub, J., Mohammad, A.:Above the cloud computing: applying cloud computing principles to create an orbital services model. In: The 6th Proceeding on Sensors and Systems for Space Applications. Washington D. C., pp. 879–879. SPIE (2013)
2. Bo, P., Peng, Y., Zhicheng, M., Jianguo, Y.: Performace measurement and analysis of ARM embedded platform using Docker container. J. Comput. Appl. **37**(S1), 325–330 (2017)
3. He, Z., Huang, D., Yan, L., Lin, Y., Yang, X.: Kubernetes based converged cloud native infrastructure solution and key technologies. Telecommun. Sci. **36**(12), 77–88 (2020)
4. Lu, G., Chen, C., Huang, Z., Huang, Z.: Research on intelligent cloud native architecture and key technologies for cloud and network integration. Telecommun. Sci. **36**(9), 67–74 (2020)
5. Du, D., Hu, A., Li, L., Zhang, Y.: Customizing Linux distribution based on the Yocto. Microcomput. Appl. **35**(14), 68–70 (2016)
6. Qiuxia, Y., Bin, Z., Lin, L., Yingying, W., Le, C.: Analysis of cloud native based edge computing products and projects. Inf. Commun. Technol. **15**(4), 71–78 (2021)