DOI 10.5162/ETTC2024/A8.4

Design and Implementation of a New Generation of Space Ground TT&C Software Architecture Based on Container Cloud

Zhong WeiHong[1]*, Feng Wei*[1]
[1] *BeiJing Research Institute of Telemetry, BeiJing China*
zhongwhong@qq.com

Abstract:
With the continuous development of space technology, in view of the increasing types and number of in-orbit spacecraft led to the increasing demand for ground measurement and control equipment. To address this issue, this paper puts forward a new generation of space ground TT&C software architecture based on container cloud. Based on the virtualization container cloud technology, we divide the space ground TT&C software, and dynamically reorganize the resources required for different system types of tasks. In practice, the architecture effectively improves the flexibility, reusability, availability and resource utilization of complex space ground TT&C software.

Key words:TT&C Software, Resource Reorganization, Virtualization, Container Cloud, Kubernetes.

Introduction

Today is the era of space, the space launch missions are becoming increasingly dense in the world. Multi-band, new systems and new model task are emerging, and the number of spacecraft in orbit has been significantly increased. The TT&C mode has changed from a few traditional systems to multiple types of systems, the number of tracking targets has increased from single target to multiple targets, and the whole system is also developing towards the trend of digitalization. For the multi-task TT&C system, the space ground TT&C software needs to quickly adapt to the requirements of task resource reorganization.

In the traditional modular space TT&C software architecture, the coupling degree between modules is high, and an exception in a small module may cause the entire system to crash, which greatly reduces the stability and fault tolerance of the system. Due to the high correlation between the functions of the system, the development of one functional module also depends on another functional module, which reduces the efficiency of software development and increases the development cycle. Meanwhile, the traditional TT&C software is directly run on a real physical machine with one machine running one software. In order to improve the fault tolerance of the system, a software often runs both primary and backup simultaneously. In this case, a TT&C software needs to occupy two machines at the same time, resulting in low resource utilization. Different software runs on different environment with different configuration, and the maintenance of configuration is quite difficult. Moreover, there are many dependency issues and low deployment flexibility.

To solve these problems, this paper introduces the container cloud technology, which runs the TT&C software in the containers and manage them uniformly through the container cloud technology. At the same time, a new software architecture is designed, which is based on the multi-task planning to divide the space ground TT&C software. Each software runs independently in a different container, and a machine can run multiple container software, which improves the utilization rate of internal resources of the system. In the event of software abnormalities or machine outage, there is no need to manually restart the software or replace the machine. The system can quickly restart the software automatically or rerun the outage service on another available machine, improving the availability of the system without reducing the resource utilization. The container cloud platform could manage the software configuration centrally. Different software has their own configuration in its own independent container operating environment, which is not affected by the deployment environment, and greatly enhances the flexibility.

Docker

In this paper, the space ground TT&C software is divided and the resources are reorganized, which requires the virtualization technology to isolate the computer resources. Mainstream virtualization technologies include two kinds: virtual machine (VM) and container technology. As we use virtual machine software such as VirtualBox need to simulate the whole machine, including virtualized hardware and its independent operating system, which is relatively heavyweight.

Docker is an open-source container engine and a way to implement container virtualization. As shown in Figure 1, unlike virtual machines, Docker containers and hosts share the same operating systems. It is more lightweight with extremely low performance. The container contains applications and all dependency packages by sharing kernels with other containers, and runs in user space with separate processes in the host operating system.

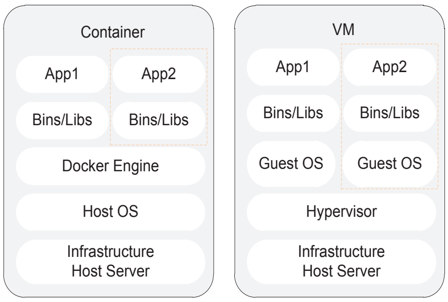

Fig. 1. Difference between container and VM.

Through Docker container technology, we can easily package the code, configuration, and dependencies of applications, turning them into easy-to-use building blocks, enabling applications to more efficiently utilize system resources, deploy faster and more efficiently, and achieving environmental consistency and high application reliability.

Kubernetes

Kubernetes is the most popular container management platform. It can realize a series of basic functions such as application deployment, scheduling and monitoring, and also supports service discovery, automatic restart, automatic scaling and many other functions.

Kubernetes is Google open source container cluster management platform, which provides a complete set of easy-to-use RESTful API (Representational State Transfer Application Programing Interface) for external service interfaces. The core idea of the Kubernetes management platform is to provide a highly available set of cluster self-healing mechanisms for container applications running on it, leaving the applications running on it in the expected state of users.

The architecture of Kubernetes is shown in Figure 2, which can be divided into two parts: control plane and working node. The control plane mainly includes components such as API Server, Controller Manager, Scheduler, and etcd key value database, which realizes the control functions of making global decisions on clusters (such as scheduling), detecting and responding to cluster events; while the working node is the physical machine actually running by the application service, each node runs multiple Pod, each Pod can run multiple containers, and multiple Pod of the same type jointly form one application service. Working nodes mainly include the kubelet and kube-proxy components, which can maintain the Pod running on the node and provide a running environment for each node.

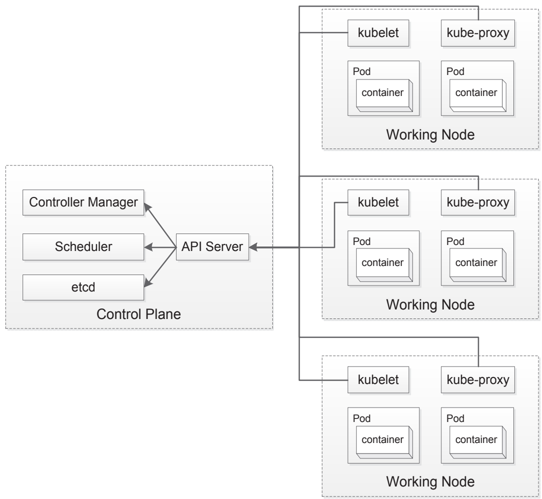

Fig. 2. Kubernetes architecture diagram.

Architecture Design

The architecture of general application software is usually categorized into two types: traditional physical machine deployment architecture and virtual machine deployment architecture. Traditional physical machine deployment architectures generally deploy applications on physical machines, but the failure to define resource boundaries for applications in a physical machine may lead to resource preemption between applications. One solution is to deploy different applications on different physical machines, running only one application per physical machine, but this makes the resource utilization too low per computer. The virtual machine deployment architecture allows multiple virtual machines running on a single physical machine, allowing applications to run between virtual machines and perform resource isolation and process isolation, enabling better use of resources on the physical machine. Container technology is similar to virtual machines, which also provides a certain degree of isolation, with its own file system, CPU, memory, process space, etc. Whereas a virtual machine is like virtualizing a complete computer running all its components (including its own operating system) on virtualized hardware, while a container is lighter and only shares the operating system between applications.

Compared to the first two general architecture deployment way, due to the lightweight characteristics of container, it creates more simple and quick, start fast, build simple, can support the application of rapid construction, deployment and update. So this paper based on container cloud technology, divide the space ground TT&C software into different application services, and dynamically restructure the resources.

The overall architecture of the new generation of space ground TT&C software based on container cloud is shown in Figure 1, which includes three layers: basic support, operations management and application service. The bottom level is the basic support layer, including the CPU, GPU, memory, network, storage and other resources required by the system, as well as the container cloud platform and various service images, which is the basis of the system operation. The second layer is the operations management, which mainly manages the operation and maintenance of the system, including image management, resource management, task management, resource scheduling, storage management and log management. The upper layer is the application service layer, in which the space TT&C software is divided into storage and forwarding software, data interaction software, system monitoring software, signal acquisition software, signal processing software and data processing software.

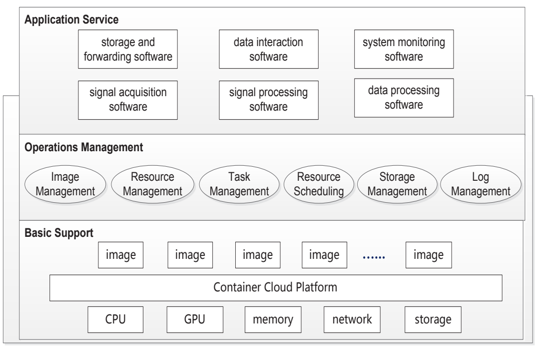

Fig. 3. System Architecture.

The entire system takes the task as the core, resolving the task when it comes. And the system resources are allocated and scheduled to meet the needs of the task operation. Select the relevant application service image to start the task software, the log and data will be collected and stored uniformly during the task, and the relevant software is withdrawn at the end of the task with the resources released.

Container Cloud Platform

The container cloud platform is the basis of the entire system. It consists of four components: Keepalived, Haproxy, Docker and Kubernetes. Keepalived is a LVS service high availability scheme based on the VRRP protocol, it can check the status of the server, if a server problems, Keepalived will remove it from the system, always keep the system has a server available to provide service. Haproxy is a high-performance load balancing software, which is responsible for balancing the access to Kubernetes to each control plane. Keepalived and Haproxy are to ensure the high availability of the system, only running on the control plane, while Docker and Kubernetes are the basis of container operation and management, running on each node.

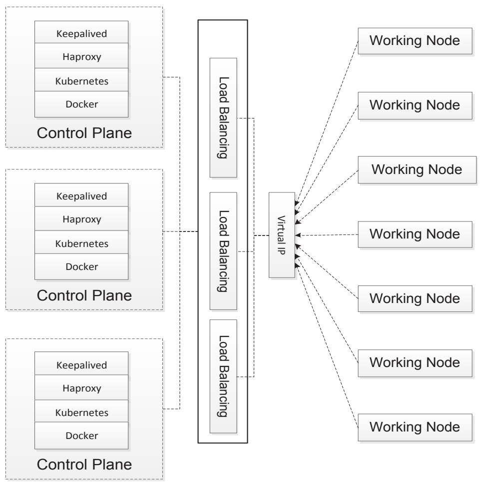

Fig. 4. Container Cloud Platform Topology.

The container cloud platform topology is shown in Figure 4. Keepalived component runs on each control plane, which virtually creates a unified IP, and the working node accesses the control plane through the virtual IP. On each control plane, there is also a Haproxy load balancer, which is responsible for diverting the requests of the working nodes and assigning them to the API servers of different control planes for processing.

Image Management

In the container cloud platform, each application software runs in the container in the form of services. The container encapsulates the computing resources and packages the CPU, GPU, memory, network, storage and other resources for the service use. The container where the service is located uses Linux's container technology to realize process isolation. The service seems to run in a separate operating system, with its own independent namespace such as file system, network, etc. Like the directory of ordinary applications, the container has its own "directory"---image. Image is the basis of the container operation, which includes the file system and program data needed for the container to run. As shown in Figure 5, different service containers can be packaged into different service images and stored in the image repository for unified management. When the task comes, it can dynamically download the task-related service image files from the image repository, and then start the container and run the related application services.

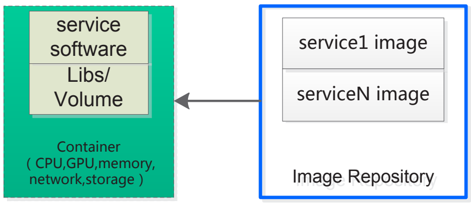

Fig. 5. Container and Image Repository.

The image repository is implemented using harbor technology. In order to ensure the high availability of the image repository, this paper has deployed harbor image repository on multiple servers. The image repository on different servers is configured with automatic synchronization. The image in any repository changes, and other repository can be automatically synchronized. The image management module visits the harbor image repository through the unique IP virtual out of the Keepalived. Even if the server of one or more of the repository in the cluster is down, as long as there is a image repository, it can ensure the normal operation of the image management function and ensure the high availability of the system.

The image management module consists of five sub-functions according to requirements: building image function, uploading image function, querying image function, downloading image function and deleting image function. Different from the virtual machine image, one of the unique characteristics of container image is image layer. Different image layers are joined together through union mount technology, and different images can share the same image layer. Thus, the same underlying part of different image in one machine only need to download and storage once, it can greatly reduce the image data transmission and storage, for the image stored in the shared image repository created convenient conditions.

Image is the basis of container operation, different tasks use different application service software, so you need to build different images according to the different application services and the environment on which they rely. When the task requirements change, you only need to delete and rebuild the corresponding image in the image repository to adapt to the changing requirements. After building the image, the image can only be used locally. In order to allow the rest of the container cloud platform machines to use the built image, the image needs to be uploaded to the harbor image repository. Other servers can query what images are in the image repository, and they can download the required images to the local in advance. When the task comes, there is no additional time and network bandwidth to download the images from the image repository.

Resource Management

The resource management function is to manage the resources that need to be scheduled in the system, including resource monitoring and the increase, deletion, isolation of resources. The resources here include two meanings: business resources and basic resources. Business resources refer to the antenna, channel, signal processing and other resources that implement business functions, which are combined into various TT&C task systems by dispatching the business resources. Basic resources refer to CPU, GPU, memory, network, storage and other operation support resources, which are the lowest level management and scheduling unit to realize business resources.

We can choose to monitor the use of resources in the system from three different perspectives. First, treating the system as a whole to view the number of used resources and the total number of resources in the system, and graphically shows the proportion of the current system used resources to easily analyze and monitor the overall resource use of the whole system. Secondly, the resource usage can be viewed according to the different resource types in the system. For example, it can list all the computers in the system and its real-time resource usage separately to analyze whether a computer in the system is overloaded with operation. Finally, the resources can be monitored from the perspective of the application service container, lists all the service containers managed in the system, display the real-time resource occupation and the proportion of allocated resources, analyze whether the service is resource-limited.

When the system needs to expand and shrink, it needs to increase and delete resources. When the resources in the system are insufficient to meet the daily task requirements, the insufficient resources can be manually added for the use of the tasks. When some resources in the system are damaged, the fault resources will be automatically isolated to prevent the task execution failure caused by selecting the wrong fault resources during resource scheduling. If the resource cannot be repaired, the failed resource can be removed manually.

For space TT&C software, there are requirements of continuous operation and online upgrade. The applications developed based on traditional platforms must apply for downtime in advance when upgrading, and conduct the outage update verification in the gap of the tasks, which increases the complexity of the plan and the inadequacy of verification for the busy space mission. The software in this system is running in the container in the form of service. The containers are isolated from each other. As long as the resources in the system are sufficient, some resources in the system can be isolated for non-stop test and verification of the software without affecting the task operation.

Task Management

In this paper, the TT&C system software is task-oriented, and the relevant TT&C software that needs to be used in the task is divided and resource reorganized. The input of the task can be resolved by remote plans or created locally. The service containers running in the system are created and deleted dynamically according to the start and end of the task.

As shown in Figure 6, the task management module includes the configuration management, plan resolution, manual management and real-time monitoring.

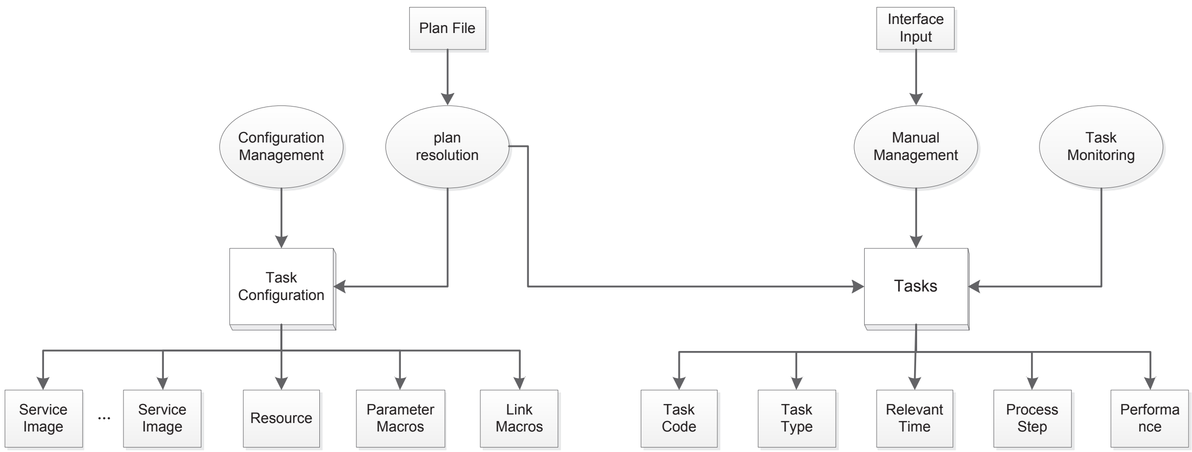

Fig.6. Task Management Functions.

The configuration management is mainly managing the relevant configuration of the tasks. The configuration of a task includes which service images it relies on, which resources it needs to occupy, parameter macros, link macros, etc. For example, for the TT&C task of a USB system, it is necessary to configure USB signal processing software image, USB data processing software image, USB system monitoring software image, etc. Then, the types and quantities of antenna resources, channel resources, computing resources and other resources used by it are also need to be configured in advance for resource scheduling. Each system has its own unique parameter list and related link configuration. Different tasks need to be configured separately. During the task execution process, the system monitor software reads the parameter macro and the link macro, and sets the macro to each stand-alone device in the system.

The plan resolution receives a remote task plan file, parses it according to the protocol format, and transforms it into an input recognized by the system. In the process of resolving the plan file, first judge whether the file format is correct, if error then reject and generate the format error rejection response. For task creation plans, the plan files are decomposed into independent tasks to read the task-related configuration from the configuration management. Then judge the legitimacy of tasks and the rationality of resource use, if failure then also produce the task-level plan failure response. After the plan is resolved, tasks are created and displayed through the real-time monitoring. For task cancellation planning, cancel the locally created tasks that did not start according to the task code.

The manual management is that the user manually CRUD tasks through interface. When creating a task, enter the task code, type, preparation time, start time, end time, resource scheduling policy, and then assign the resource scheduling module for resource conflict judgment. If the resource conflict, then the user is prompted to create the task failure. After creating the task, if parameters such as task execution time change, you can manually edit the task parameters on the interface.

The task monitoring is responsible for monitoring the running status of tasks, including real-time tasks and historical tasks. Real-time task monitoring can view whether the task has started, where the process has started, whether it is performed smoothly, and other task overview information. To view the task details, you can open the system monitoring software interface to view. Historical task monitoring can view the performance of historical tasks, including the actual start time, actual end time, execution success and other information.

Resource Scheduling

When creating a task, system need to schedule the resources according to the resource allocation information of the task. Resource scheduling is one of the core functions of the system, and it needs to consider the following points: *1)Resource utilization: this is the core indicator of resource scheduling. In resource allocation and scheduling, tasks need to comprehensively evaluate the use of all resources in the system, adopt appropriate scheduling strategies, and try the best to run the most tasks under the condition of limited resources; 2)Flexibility: Different tasks require different resources, and different times will run different tasks, so we need to support a variety of scheduling policies for flexible resource scheduling; 3)Real-time: In a large-scale system with complex environment, the number of tasks and resources is huge, it can be dispatched efficiently without affecting the normal operation of tasks.*

The flow of resource scheduling is shown in Figure 7. After creating the task, the resource scheduling module first obtains the resources required by the task, and classifies them according to the resource types such as CPU, GPU, memory, and so on. The second step is to obtain all the current resources and resource occupation in the system from the resource management module. The third step is to pre-select all the required equipment according to the resource demand and resource surplus. Finally, using the corresponding scheduling strategy according to the actual situation calculate the priority weight and select the best resources. The scheduling strategy can be manually specified by the user or automatically select the optimal policy.

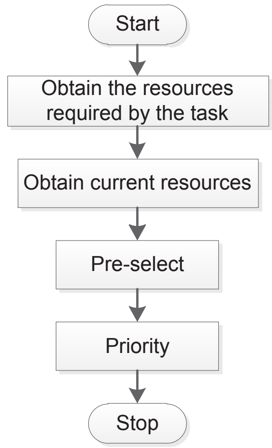

Fig. 7. Resource Scheduling Flowchart.

After deleting task, release the resources for other tasks.

Log Management

The system log mainly records the running status of the program and user's operation behavior, which plays a very important role for the post-hoc analysis of the system. Log records are categorized into three different levels, error, warning and normal. Error log refers to the log that affects the normal operation of the system and must be avoided; the warning log is a log with potential risks and should be avoided as far as possible; the normal log is some user-perceived events generated during the normal operation of the system.

By checking the process content of the program in the log, the software developers can quickly find out where the bugs appear, the defects of the system operation and quickly repair the loopholes. Meanwhile, with the development of big data technology, the log, as an important data source in the program, can improve the level of the system intelligence and monitor the system operation to improve the performance and detect faults through data mining technology.

In this paper, the program is distributed in the various containers of the cluster in the form of application services, so the log information generated by the system is also scattered in the isolated containers, which brings some difficulties to the collection of logs, and the collection of logs from different services becomes a particularly important function. At the same time, once the container is destroyed, the data in it is gone, and the host where the container is located will dynamically schedule according to the resource load of the system, so it brings challenges to the storage and unified query of logs.

To sum up, the log management in the system mainly includes three functions: log sending, log storage and log query. The log sending function can provide an interface for each application service in the system. Through this interface, the application services running in different containers of different nodes in the cluster can send the logs rather than directly stored in the container. Log management uses redis's message queue mechanism to receive log forwarding and store them uniformly in a database that can be accessed when needed to query all logs produced in the platform. Log query supports a variety of combinations, such as can separately query a certain time period, a certain task, a certain log level log and display, to improve the efficiency of log analysis.

The full flow chart of log management is shown in Figure 8, during the log-sending process, the application service first connects the redis node in the cluster. When log information is generated during the run, application send the log by calling the message sending interface of redis; The log storage function calls the message receiving interface to receive log information sent from different application services, then connect on the local MySql database, store the received logs in the database in an order; the log query function, according to the query conditions selected by the user, perform the corresponding sql statement to query the logs in the database. Finally, the log is displayed on the interface for user analysis.

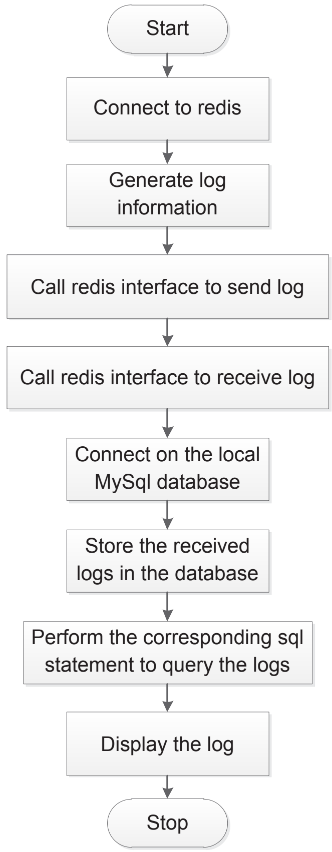

Fig. 8. Log Management Flowchart.

Future Prospects

This paper studies the container cloud technology, divides the space ground TT&C software into different application services and runs in different containers, so as to design a new generation of space ground TT&C software architecture. Using this architecture as a framework, the application services and basic resources required by different tasks are dynamically reorganized. The system has high flexibility, good maintainability, and has the advantages of high availability and high resource utilization. In the future, we can study how to divide application services into smaller microservices to reduce the coupling of software in the system, and a single service with smaller granularity requires less resources, which can effectively reduce scattered resources in the system and improve resource utilization. All the software in this system is running locally. In the future, we can study how to reduce the delay of remote communication, so that different types of software can be deployed in different places according to the needs to realize unmanned stations and ultra-remote TT&C. At the same time, other space application software similar to the TT&C software architecture can be studied to expand the business scope and improve the compatibility of the system.