International Civil Aviation Organization

Fifth Meeting of the Asia/Pacific Air Traffic Management Automation System Task Force (APAC ATMAS TF/5)

Chengdu, China, 5 – 7 June 2024

Agenda Item 4: ATM Automation System Implementation Experience by States

4.5 Development of New Technology

THE CONTAINER APPLICATION AND MICROSERVICES EXPLORATION OF AIR TRAFFIC MANAGEMENT AUTOMATION SYSTEM SERVERS

(Presented by China)

SUMMARY

This paper outlines the endeavors and challenges addressed in the exploration of container applications and microservices for traditional ATMAS servers, conducted by a team from ATMB of CAAC. Furthermore, it highlights several crucial issues that merit attention in future developments of ATMAS microservices.

1. INTRODUCTION

1.1 The Air Traffic Management Automation System (ATMAS) is a typical distributed real-time application system. At present, the production-level systems are generally deployed using traditional physical servers. During operation and maintenance, problems such as incompatibility between software and hardware upgrades, unbalanced resource utilization, long system restart and recovery time will occur.

1.2 In the paper “Architecture Design to Address System Upgrade and Interoperability Challenge” at the ATMAS TF/4 conference in 2023, Singapore discussed the issue of using open architecture and modern technology to improve the upgrade and interoperability of ATMAS. It put forward the concept that the open architecture of ATMAS can be realized based on microservices architecture. However, if microservices are used, cloud-native solutions and an entire new system architecture need to be considered. It is a long way.

1.3 Since 2016, a team from ATMB of CAAC has explored virtualization, containerization, and microservices in stages based on the traditional distributed ATMAS. At the same time, the team has put forward some vital issues of the ATMAS microservices. Considering the close relationship between containerization and microservices, the efforts could be helpful for the microservices of the next-generation ATMAS.

2. DISCUSSION

Container application of traditional ATMAS

2.1 At present, the team has completed the transformation of the ATMAS deployment environment from physical machines to virtualization and then to containerization of the cloud-native platform.

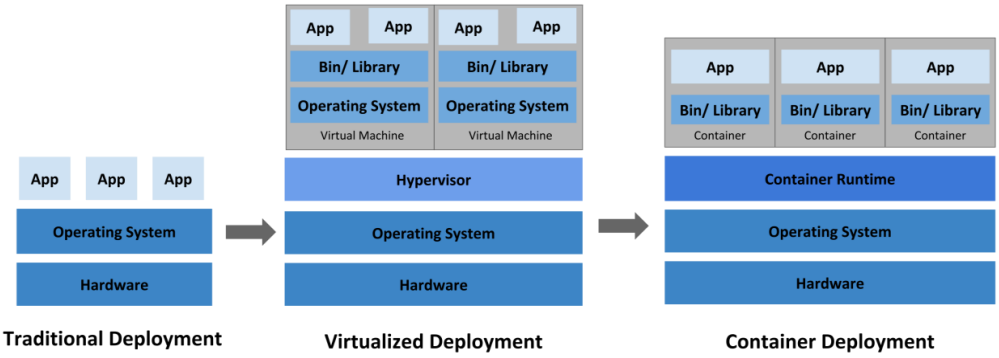

Fig.1 Environment transformation

2.2 From 2016 to 2019, a virtualization platform with shared storage was built, and ATMAS servers were migrated. Virtualization solutions simplified server management and maintenance. It saved resources, facilitated functional testing and brought high availability advantages such as load balancing, dynamic migration and rapid replication.

2.3 Since 2019, the container environment has been deployed. First, use the Dockerfile to build Docker images of the ATMAS servers, and then use Kubernetes to deploy and manage the containers uniformly. The transformation of the ATMAS servers from virtualization to a cloud-native platform is realized.

2.4 As is known, containers are process-oriented. To realize the microservices of ATMAS, an important feature is that all processes should run independently in containers and communicate normally. However, limited by the existing system architecture, the team only completed the whole server migration instead of process migration from the virtual machine to the container. At present, the on-site cloud-native platform is as follows:

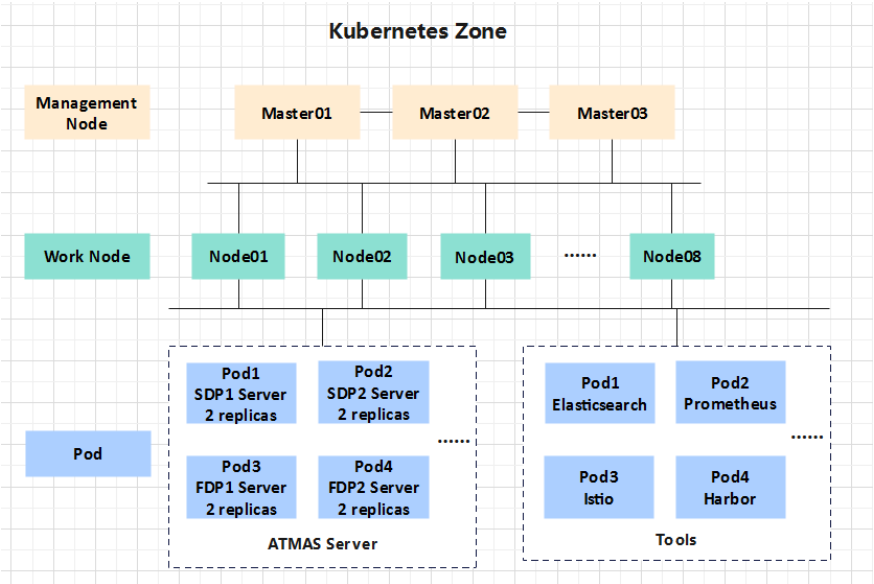

Fig.2 On-site container deployment platform

2.5 Comparing the three implementations, the current cloud-native containerization platform has inherent resource utilization, cold boot time, and resource scheduling advantages.

	Resource utilization	**Redundancy**	**Cold boot time**	**Resource scheduling**	**Deployment method**
Physical	Low	Several physical machines	Minute	None	Manual
Virtualization	Medium	Several virtual machines + Fault tolerance	Ten seconds	Partial elasticity	Manual
Containerization	High	Several pods + Several replicas	Millisecond	High elasticity	Automatic

Tab.1 Comparison of deployment environment

2.6 An on-site typical application is seamless and rapid rolling updates and rollbacks of different ATMAS software versions by the following steps:

a) Package and image the server programs of different software versions.

b) Choose the desired release strategy through Kubernetes controllers.

c) Update the pod replicas to the new version one by one automatically without interrupting service provision.

2.7 To explore new applications for ATMAS, other open-source tools are applied in the platform to optimize the maintenance:

a) ELK can complete the centralized collection, storage, management, and analysis of logs, helping technicians control the overall operation better and avoid failure judgment due to omissions. At present, the critical point in the test is that manufacturers need to standardize the format of each log and establish keywords as the basis of log analysis.

b) Prometheus can provide more detailed monitoring of big data support and analysis. It is convenient for data maintenance and governance, which perfectly meets the preventive maintenance needs in ATMAS.

c) Istio has powerful network traffic management functions. It can only be used for TCP traffic monitoring in traditional systems. However, for next-generation systems using microservices, Istio can realize grayscale release of software testing through network traffic splitting of different versions, which fully meets the requirements of multi-scenario testing.

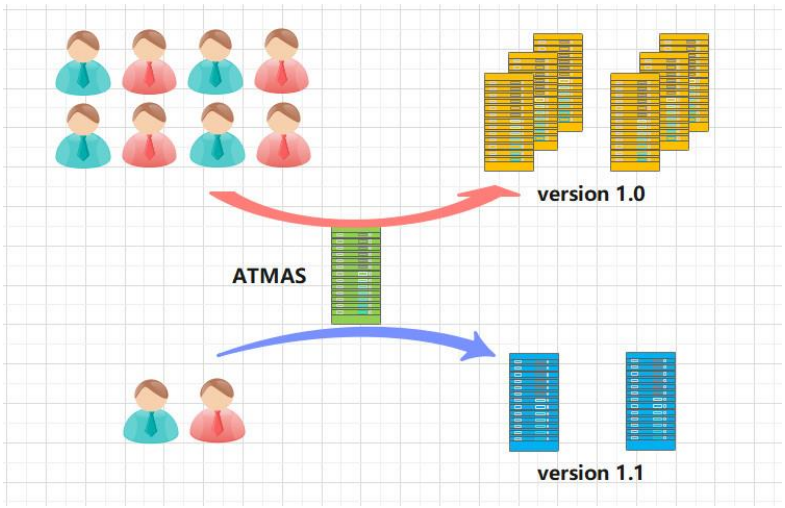

Fig.3 Grayscale release for ATMAS in the future

2.8 In the process of containerization, many problems arose from the incompatibility between traditional ATMAS architecture and the cloud-native platform. The problems and solutions are as follows：

a) The on-site ATMAS relies on multicast communication. However, the container network usually does not support multicast. To solve the problem, the team used Project Calico and static multicast router software to realize the multicast communications among different pods.

b) ATMAS usually use uppercase to identify the hostname in operation. However, the hostname in the pod can only be set to lowercase. To solve the problem, the team checked and adjusted some ATMAS configuration files to ensure the system programs ran normally.

Microservices exploration and restrictions of traditional ATMAS

2.9 After containerizing the system server in the cloud-native platform, the team tried to explore the system's microservices further. However, based on the traditional system, containerization is the limitation. It is difficult to run each server process in an independent container and ensure regular communication to realize microservices. The main reason is that the system structure and software design are based on a traditional distributed architecture, which does not match the architecture of microservices. The specific issues, which also need to be focused on in the design of next-generation system, are as follows:

a) Traditional systems usually use fixed IP addresses to communicate, but the IP address in the container is dynamic. Therefore, the systems cannot use the service registration and service discovery of the cloud-native platform.

b) The network communication of the traditional system mainly depends on multicast, while the granular characteristic of the cloud-native network traffic management requires TCP connections.

c) The processes of the system servers are mainly deployed in a tightly coupled way, relying on private middleware for communication. However, microservices require that all processes should run in independent containers loosely coupled

and use shared middleware to realize the communication between them.

d) Traditional systems only support IPV4 without IPV6.

Prospects

2.10 In several years of testing, the team conducted the container application and microservices exploration of traditional ATMAS servers. Considering the possible problems in the whole-module microservices of ATMAS, the partial-module microservices can be used as a prior option for testing and verification. We hope that more partners join in the testing and researching of system microservices and explore the subsequent management, operation, and maintenance.

2.11 In the future, the team will continue to explore the application of new technologies in ATMAS, such as cloud-native storage and Foundation Models.

3. ACTION BY THE MEETING

3.1 The meeting is invited to:

a) note the information contained in this paper; and

b) discuss any relevant matter as appropriate.

– – – – – – – – – – – – –