Auto Scaling of Containerized ACSs for CPE Management

Tse-Han Wang
Network Management Laboratory
Chunghwa Telecom Laboratories
Taoyuan, Taiwan
wangth@cht.com.tw

Yen-Cheng Chen
Department of Information Management
National Chi Nan University
Puli, Nantou 545, Taiwan
ycchen@ncnu.edu.tw

Chen-Min Hsu, Kai-Sheng Hsu, Hey-Chyi Young
Network Management Laboratory
Chunghwa Telecom Laboratories
Taoyuan, Taiwan
{paulhsu, kshsu, heychyi}@cht.com.tw

Abstract*—The increased popularity of over-the-top (OTT) services such as voice calls and text messaging popular has a great impact on the business of traditional telecom operations. It is important for a telecommunications company to maximize resource utilization so as to decrease OPEX/CAPEX (Operating Expense/Capital Expenditure). On the other hand, the deployment of a variety of digital services for broadband Internet access may complicate telecom operations. When more digital appliances, such as set-top boxes and IP cameras, are deployed at home, more configurations are required in both CPEs and POPs. CWMP is a management protocol between CPEs and auto configuration servers (ACSs) for remote management of those home network devices. It simplifies the service activation and configuration, and provides functionalities of diagnostics and monitoring. The scalability of ACS is an important issue as more CPEs and more configurations are needed, especially when a lot of Internet of Things (IoT) are widely adopted. This paper focuses on ACS containerization to deliver services quickly and to scale well. To dynamically control and manage the containerize ACS, we propose a middleware approach to support container management and to improve scalability horizontally based on designated policies.

Keywords—Virtualization; Container; CWMP; TR-069

I. INTRODUCTION

With the prosperity of mobile devices and Internet applications such as video streaming, the traffic in Internet has become more diverse than before. The demand of bandwidth is increasing significantly, but doesn't increase the revenue of a telecom operator. As over-the-top (OTT) services such as voice calls and text messaging become popular, the revenue contributed from traditional telecom services thus decrease. Telecom operators then take more effort to maximize resource utilization and decrease OPEX/CAPEX (Operating Expense/ Capital Expenditure).

It becomes very common that customers enjoy a variety of digital services at home through digital appliances connected to their home networks, which are then connected to the Internet. Typical digital appliances are set-top boxes, IP cameras, and smart home devices. It implies that more settings are required in networking devices. For better management, many networking devices are configured automatically by auto configuration servers (ACSs). CWMP (CPE WAN Management Protocol) is a management protocol developed for the communication between Customer-Premises Equipments (CPEs) and ACSs to enable remote management of home network devices. Through the centralized control of CPEs under the administration of ACSs, many management tasks can be simplified, including service activation, configuration, diagnostics, and monitoring.

It is anticipated the widespread adoption of the Internet of Things (IoTs), as well as the deployment of small cell base stations, will result in the great growth of CPEs. As a result. CWMP will play an important role in CPE management and ACSs must scale well to afford the management of more CPEs. One may estimate the increase in CPE numbers and then installs more ACSs to provide more capacity. This paper addresses this issue and provides a more efficient approach by operating system (OS) virtualization and integration with Chunghwa Telecom's COSMOS (CPE Operation Support Management and Optimization System) architecture to enhance its scalability and flexibility in ACS provisioning. To control and manage the containerized ACS dynamically, we propose a middleware framework to support container management and to scale ACSs based on designated policies.

The remainder of this paper is structured as follows. Section II describes related works. Section III describes the proposed architecture. Then its use cases are presented in Section IV. Finally, conclusions are given in Section V.

II. RELATED WORKS

A. TR-069

Technical Report 069 (TR-069) [1], published by Broadband Forum and known as CPE WAN Management Protocol (CWMP), is an application layer protocol for remote management of end user devices. TR-069 provides the communication between Customer Premises Equipment (CPE) and an Auto Configuration Server (ACS), including auto configuration, dynamic service provisioning, and the control of CPE management functions within an integrated framework, as shown in Figure 1.

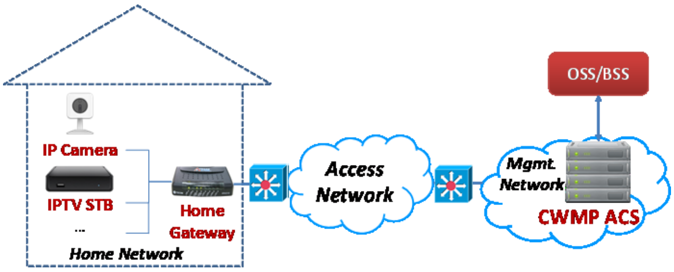

Fig. 1. TR-069 Management Architecture

B. LXC/Docker

Linux Containers (LXC) [2] is a lightweight Virtual Machine (VM), which performs virtualization in the OS level. LXC enables multiple isolated Linux instances running on a single host. It leverages cgroups (control groups) to limit and meter the use of resources, including memory, CPU, block I/O and network, and namespaces, to provide an isolated runtime environment.

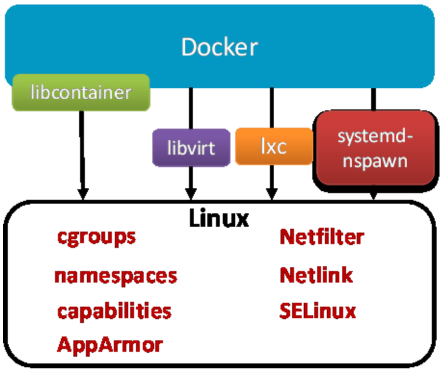

Fig. 2. Docker support different interfaces to leverage virtualization features of the Linux kernel

Docker [3] is an open source software that simplifies the deployment of applications inside containers. As shown in Figure 2, built on top of virtualization functionalities provided by the Linux kernel, Docker provides high-level APIs and supported different interfaces to create lightweight containers which run processes in isolation.

C. Enhancement in CWMP Management Capacity

CWMP has been widely used by telecom service providers. With the rapid growth of managed devices, an ACS for CWMP may reach the limit in resource consumption. Scalability is usually not considered as the top issue in the initial stage of a system deployment. As the number of services increases, capacity expansion becomes the primary issue in ACSs. In [4], a dynamic grouping and sub-ACS structure is proposed to add additional components to the original ACS management architecture. With the new model, the ACS management capacity can be expanded. However, the CWMP client function in a CPE should be equipped with a corresponding sub-ACS agent. In [5], a dynamic distribution technique is proposed to adaptively adopt an appropriate session management strategy to distribute each session to a selected CWMP cluster node. The study shows that session management is crucial to achieve better system performance and resource utilization.

III. CPE Operation Support Management and Optimization System

CPE Operation Support Management and Optimization System (COSMOS) is a CWMP-based Operations Support System (OSS) deployed on the management network of Chunghwa Telecom (CHT). COSMOS provides an integrated, multi-function, and easy-to-use operation environment. COSMOS supports management of multi-vendor and multi-user CPEs such as home gateways, IP cameras, video phones [6].

A. Design Concept

In the past, the deployment of ACS management nodes is laborious. Telecom service operators should firstly estimate the maximum number of devices to be managed, purchases enough servers, and then configure each server to manage a certain portion of these managed devices. Obviously, the scalability issue is the major concern in the setup of ACSs, especially when a great number of CPEs are deployed by users. Therefore, a more agile and flexible method is required to facilitate the installation of ACSs.

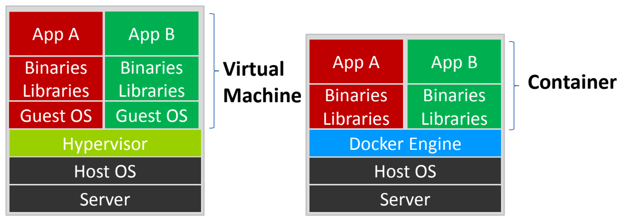

Fig. 3. Difference between Virtual Machine and Container

This paper adopts a virtualization technology to provide a more flexible way to deploy ACSs. Virtualization, a well-known technology in cloud computing, can improve server utilization, reduce power consumption, and simplify server provisioning without the need of physical hardware installation. The major concept of the system design is to construct a vendor-neutral resource pool for better resource utilization. Thus, the efficiency and flexibility of network operations can be improved. Recently, Docker, a new solution in OS-level virtualization, receives much attention for its application container framework. This paper introduces Docker into the proposed COSMOS system to deliver applications with the application-centric OS virtualization technology, different from other applications running under fully-isolated VMs. A VM usually runs with a virtual copy of all the supported hardware. This leads to a lot of CPU and memory cycles. Unlike VMs, a container only requires an OS, supporting binaries, libraries, and system resources to run a specific program. Figure 3 shows the difference between VMs and containers. A significant benefit of Docker is that it can package application runtime environment into an image and distributes this image on a different hardware.

In order to dynamically control and manage the containerized ACS, the proposed COSMOS system should support container management and be able to dynamically scale horizontally based on designated policies. Thus, functional components can be easily scaled horizontally to handle more requests simultaneously.

B. *Architecture*

The current COSMOS architecture follows a two-layer hierarchical design and consists of two entities, *Global ACS* and *Core ACS*. Similar to a *Master-Worker* pattern, *Global ACS* (Master) has a global view of managed devices and can perform processes simultaneously across all *Core ACSs* (Workers). A message queue is used for communication between the *Global ACS* and the other *Core ACSs*. COSMOS provides five operation management functions: alarm, management, firmware management, event management, report management, and device management, as shown in Figure 4. Alarm management allows operators to specify the destination of an alarm notification according to the severity and type of an event. Notifications can be sent via emails or Short Message Service (SMS). Firmware management enables the scheduling of firmware image upgrades of devices by different image versions and PoPs. Event management lists event messages by categories or severity levels of events. Events come from the system itself, devices, and audit logs. Report management provides various statistics reports for operators to understand the overall status of managed devices. Device management shows information of managed devices such as device configuration, and provides diagnostics and control functions, including real-time device status monitoring, device rebooting, configuration restoring, and so on. Moreover, device management provides device templates which define provisioning parameters for designated devices. In addition to the five management functions, an authentication and authorization module is included for the control of operation privileges among multiple users. To support multi-vendor CWMP-enabled devices, COSMOS unifies the view of manage devices via resource adaptors, which provide an intermediary abstraction of devices. The application plugins in *Global ACS* are responsible for collecting data from the resource adaptors of *Core ACSs*.

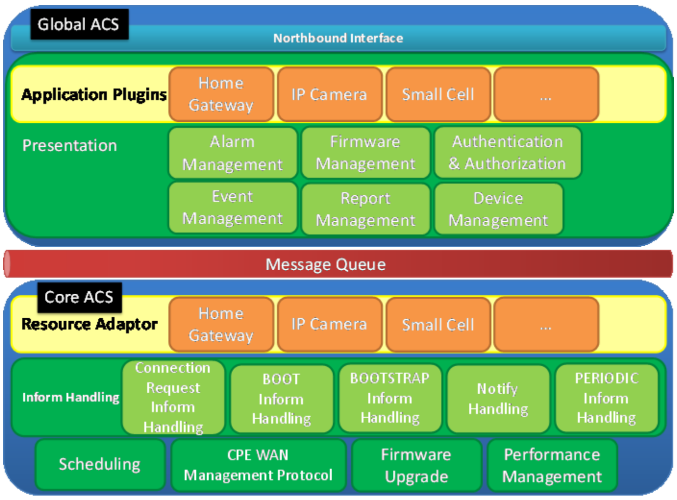

Fig. 4. COSMOS Software Architecture

In COSMOS, each *Core ACS* directly interacts with a certain number of managed devices. A *Core ACS* is a server which implements the ACS methods defined in [1]. It also supports a scheduling mechanism that manages jobs automatically by pre-defined statements. When more management tasks should be done simultaneously, more *Core ACSs* are required. *Core ACSs* are deployed in a container and built as a container image. Figure 5 illustrates an extended architecture of COSMOS. A middleware is introduced in the control layer the management of container resources and the load balancer. The middleware has three major components. The *container resource manager* gets resource usage statistics, monitors health, and manages the lifecycle of containers via southbound APIs. The *load balancer manager* checks the status of the load balancer and adds a new mapping rule between the virtual IP address and the real IP address of a *Core ACS*. *CLI/REST/Docker Client* provides southbound interfaces for use in the container resource manager and the load balancer manager. In this proposed design, a load balancer is required in the infrastructure layer for dispatching workloads across multiple *Core ACSs*.

C. *ACS Auto Scaling*

An important feature proposed in the extended COSMOS architecture is the auto scaling of ACSs. This can be realized via the policy engine module in the middleware. The policy engine allows the *Global ACS* to enforce the rules for determining when *Core ACSs* should be scaled out. A northbound interface is provided to allow the auto scaling of *Core ACSs* via the *Global ACS* dashboard manually. Auto-scaling is important for better availability and processing capability of applications.

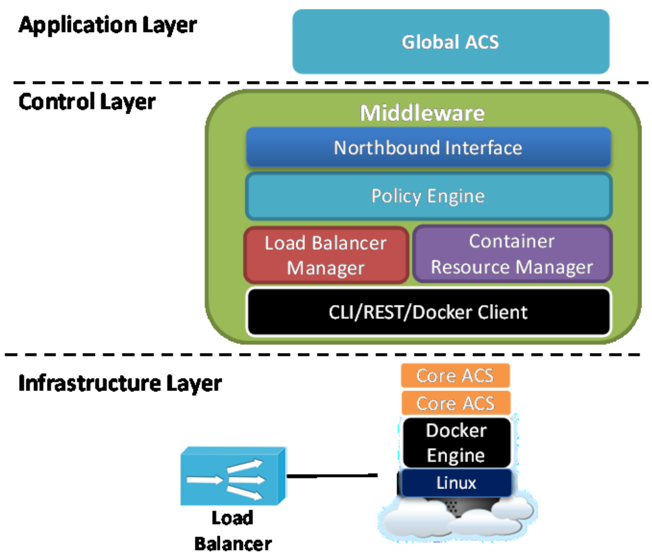

Fig. 5. COSMOS Architecture Extension

CPU and memory metrics are used to estimate the performance of COSMOS. The policy engine provides REST APIs to the *Global ACS* to set up parameters, including CPU utilization, memory usage threshold values, and minimum and maximum size of containers. The container resource manager monitors resource usage periodically. When the resource usage exceed the designated threshold values within a given time period, a new *Core ACS* container and the mapping rule of a virtual and a real IP address will be created.

IV. USE CASES

A. Firmware Image Files Upgrade/Downgrade

CWMP provides firmware image files management of CPEs. The firmware image files management functions provided by CWMP are firmware image version identification, files download initiation (firmware image upgrade/downgrade), and notification upon the success or failure of a file download [1].

If there is a need to upgrade the firmware of all managed home gateways, the operator can upload the firmware software image first and then schedules the firmware upgrade for all applicable home gateways. For better performance, the operator may create more *Core ACS* instances to perform firmware upgrade operations. Firmware upgrade will become more critical in the IoT era. As a variety of IoT CPEs are deployed, a number of firmware updates will follow.

B. Service Rate Test

Telecom service providers are usually responsible to provide actual service data rate for broadband access subscribers. There is service rate test mechanism in COSMOS [7]. Currently, COSMOS randomly selects managed home gateways of subscribers with the support of service rate test. Due to resource restriction, a *Core ACS* server cannot afford the service rate test of all managed home gateways. Through containerized COSMOS, more *Core ACSs* can be scaled out on demand to share the load of service rate test. As a result, more managed home gateways can be involved in service rate test. The above scheme can also be applied to QoS (Quality of Service) test [8] to verify whether the actual QoS metrics such as delay, jitter, and packet loss meet the service requirements of customers that have been implemented on COSMOS.

V. CONCLUSION

Resource allocation in an agile and flexible way is a challenging task for all enterprises. Recently, the increased popularity of cloud computing services and Software Defined Networking (SDN) affects the way enterprises purchase and use hardware. Nowadays, computing, networking, and storage infrastructure can be under the control of software without the physical intervention of human beings. Effective management and orchestration of software-defined infrastructure may facilitate the centralized control of the physical infrastructure and thus improve the resource utilization. In this paper, we have presented a CWMP-based OSS developed by Chunghwa Telecom. An extended OSS architecture is also proposed to enable *ACS* auto scaling within a container platform. The middleware has been implemented to prove the concept of resource provisioning through container virtualization. In the future, we will study the management of container clusters and the design of a *microservices* architecture for Telecom OSS.

REFERENCES

[1] TR-069 Amendment 5, "TR-069 CPE WAN Management Protocol", Broadband Home Technical Working Group, Broadband Forum, November 2013.

[2] "Linux Containers," https://linuxcontainers.org

[3] "Docker," https://www.docker.com

[4] Lee, K. M., Chu, H., Chu, J., Lin, Y., Hsiao, C. F., and Hou, T. W., "ACS management capacity enhancement mechanism in CWMP", *Electronics Letters*, January 2014.

[5] Lee, K. M., Teng, W. G., Wu, J. N., Chen, P. Y., Huang, M. K., & Hou, T. W., "Dynamic distribution technique for session management in CWMP server clusters", *Electronics Letters*, October 2014.

[6] TR-104 Issue2, "TR-104 Provisioning Parameters for VoIP CPE", Broadband Home Technical Working Group, Broadband Forum, March 2014.

[7] Yang, W. C., Jhan, J. D., Chen, D. Y., Lai, K. H., & Lee, R. R., "Service rate test mechanism and management of broadband access network", Asia-Pacific Network Operations and Management Symposium (APNOMS), September 2013.

[8] Yang, W. C., Jhan, J. D., Chen, D. Y., Lai, K. H., & Lee, R. R., "Quality of service test mechanism and management of broadband access network", Asia-Pacific Network Operations and Management Symposium (APNOMS), September 2014.