

Virtual Machine Management Technology for Operating Packet Switching System in a Virtualization Environment

Makoto Hamada, Eriko Iwasa, and Keiko Kuriu

Abstract

NTT Network Service Systems Laboratories is developing infrastructure technology for operating virtualized packet switching system service functions with the aim of realizing a network control infrastructure providing safe and secure services. In this article, we introduce virtual machine management technology for sharing hardware resources in a virtualization environment and onboarding and operating packet switching system service functions in conjunction with OpenStack.

Keywords: virtual machine management, packet switching, network functions virtualization

1. MAGONIA rollout

The NetroSphere concept [1] was announced by NTT in February 2015 as a new research and development concept aimed at revolutionizing the way to form a carrier network infrastructure. As part of this initiative, NTT Network Service Systems Laboratories is researching and developing a new server architecture called MAGONIA to enable the short-term creation and deployment of services and to achieve drastic reductions in development and operating costs.

MAGONIA provides a distributed processing base (DPB) [2, 3], Soft Patch Panel (SPP) technology [4], and virtual machine (VM) management technology as components that can be used by individual applications. In this article, we introduce MAGONIA VM management technology.

Studies on network functions virtualization (NFV) are moving forward in the European Telecommunications Standards Institute Industry Specification Group for NFV (ETSI NFV ISG). NFV is an initiative to run VMs on general-purpose servers and to provide diverse services in a flexible and prompt

manner by virtualizing (converting to software) network functions traditionally provided by dedicated hardware appliances. Virtualization makes it possible to run more than one virtually created machine on a single piece of hardware, which means that VMs can be dynamically created as needed, with each one sized most appropriately for the conditions at hand.

The ETSI NFV architecture is divided into three main components, as shown in **Fig. 1**: network functions virtualization infrastructure (NFVI), virtual network functions (VNF), and management and orchestration (MANO). The NFVI area consists of physical resources and virtualization functions for running VNF. It achieves a function called a hypervisor for generating multiple virtual servers on a physical server. The kernel-based VM (KVM) is a typical hypervisor. The VNF area, meanwhile, consists of network functions implemented as software. Finally, the MANO area, whose central role is operation automation, consists of functional blocks called the virtualized infrastructure manager (VIM), VNF manager (VNFM), and NFV orchestrator, each with separate roles.

The VIM manages NFVI and allocates VMs, the

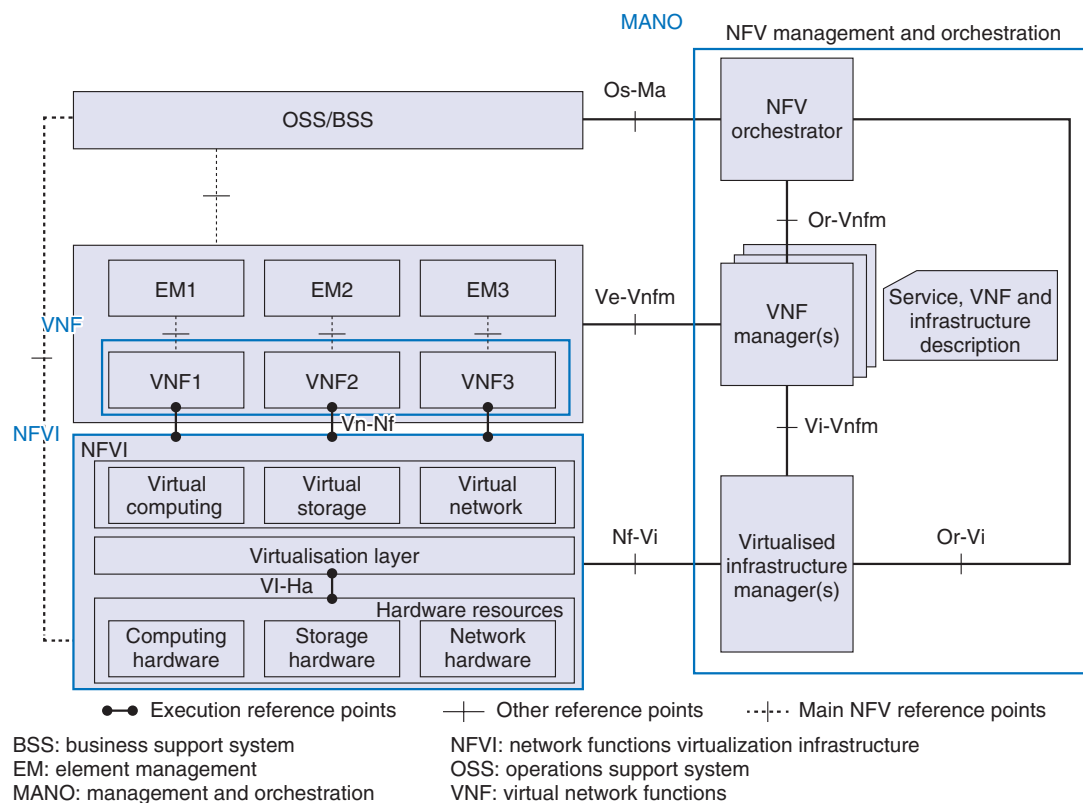


Fig. 1. ETSI NFV architecture.

VNFM manages VNF, and the NFV orchestrator manages, controls, and optimizes network services and resources in an integrated manner. While OpenStack has become the de facto standard for achieving a VIM, NFV products used throughout the world consist of original vendor developments based on OpenStack. Various telecommunications carriers are focusing on the application of NFV architecture to large-scale communications networks, but it is important that issues surrounding the development and implementation of packet switching service functions are solved.

2. Issues in virtualization of packet switching system service functions

The role of network packet switching system service functions such as firewall and network address translation (NAT) functions is to check and process the contents of all received packets. It is therefore necessary to process a large volume of packets with as short a delay as possible. Such packet switching system service functions have traditionally been

implemented using dedicated equipment to satisfy this strict performance requirement. In MAGONIA, these functions are converted into software (packet switching software) and operated on a common platform based on virtualization technology with the aim of reducing capital expenditure and operating expenses and invigorating service development.

However, a number of issues arise in virtualizing packet switching system service functions. Since virtualization technology achieves multiple VMs on a single piece of hardware, these VMs can interfere with each other and cause processing delay and unstable throughput. For a server system supporting web services, which is also targeted by virtualization technology, latency requirements are not strict, and it has been possible to configure a system that increases the number of hardware units whenever processing power becomes insufficient, even if throughput is unstable.

Packet switching system service functions, however, must process a large volume of packets with no allowance given for processing delay. In addition, many existing communications networks have been

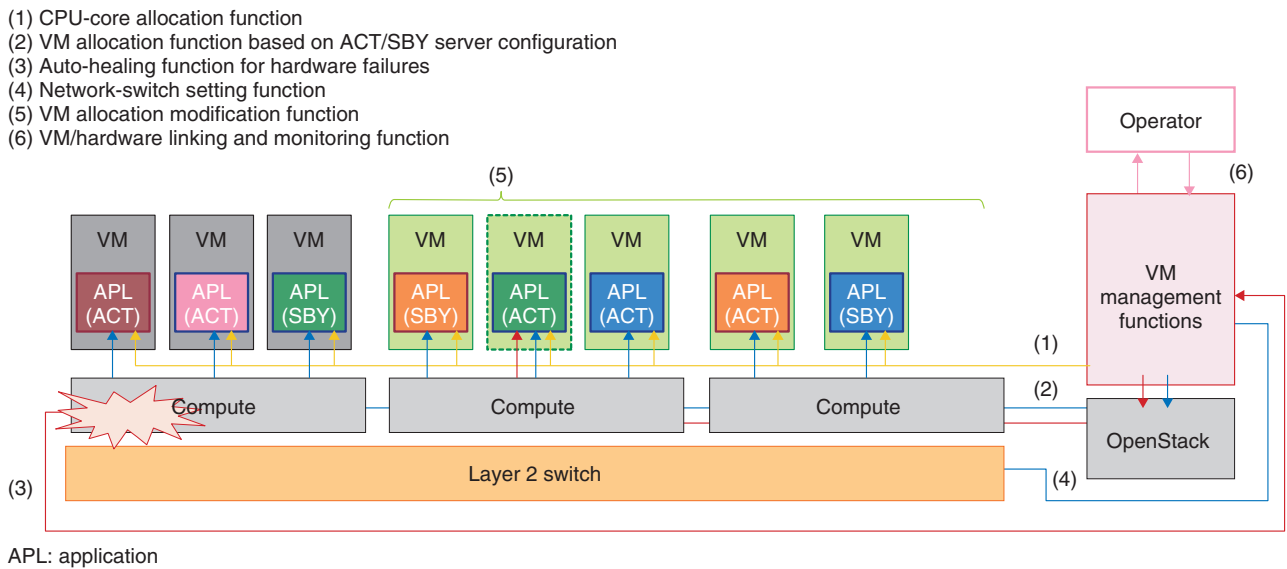


Fig. 2. Functions achieved by VM management technology.

constructed with a specific type of architecture that sets beforehand the number of users that can be processed by each piece of equipment, so for this and the above reason, performance cannot be maintained by simply increasing the number of hardware units in the case of web servers. Thus, when implementing packet switching system software using NFV architecture, the performance of virtualized service functions must be guaranteed assuming that type of architecture.

In addition, existing communications networks consist of active and standby (ACT/SBY) servers and employ a failover mechanism in which a SBY server takes over the processing of an ACT server that fails. Under a virtualization environment, this means that VM allocation control must take into account a system having a redundant-configuration mechanism. The VM management technology in MAGONIA achieves functions for solving those issues that arise in applying network services to a virtualization environment.

3. Functions achieved by VM management technology

The functions achieved by VM management technology in MAGONIA are shown in **Fig. 2**. The central processing unit (CPU)-core allocation function (1), which is one of the KVM functions, allocates VM processing to CPU cores. This function achieves low processing delay and stable throughput by exclu-

sively allocating processes requiring high-speed processing to CPU cores. Next, the VM allocation function based on an ACT/SBY server configuration (2) can allocate VMs to optimal hardware in a system having a redundant configuration. In addition, the auto-healing function for hardware failures (3), the network-switch setting function (4), the VM allocation modification function (5), and the VM/hardware linking and monitoring function (6) support failover processing at the time of a hardware failure or fault occurrence and lifecycle management tasks.

4. CPU-core allocation function in MAGONIA VM management technology

The mechanism of the CPU-core allocation function is shown in **Fig. 3**. This function provided by MAGONIA VM management technology has three separate functions.

The first function allocates the processing of one virtual CPU within a VM to one physical CPU. In this way, it exclusively allocates a physical CPU core to a virtual CPU executing high-load processing, thereby maintaining the processing performance of that virtual CPU.

The second function exclusively allocates a group of virtual CPUs to a single physical CPU core. Applying this function to multiple virtual CPUs performing processing that consists of a light load on the machine can achieve effective use of physical CPU resources

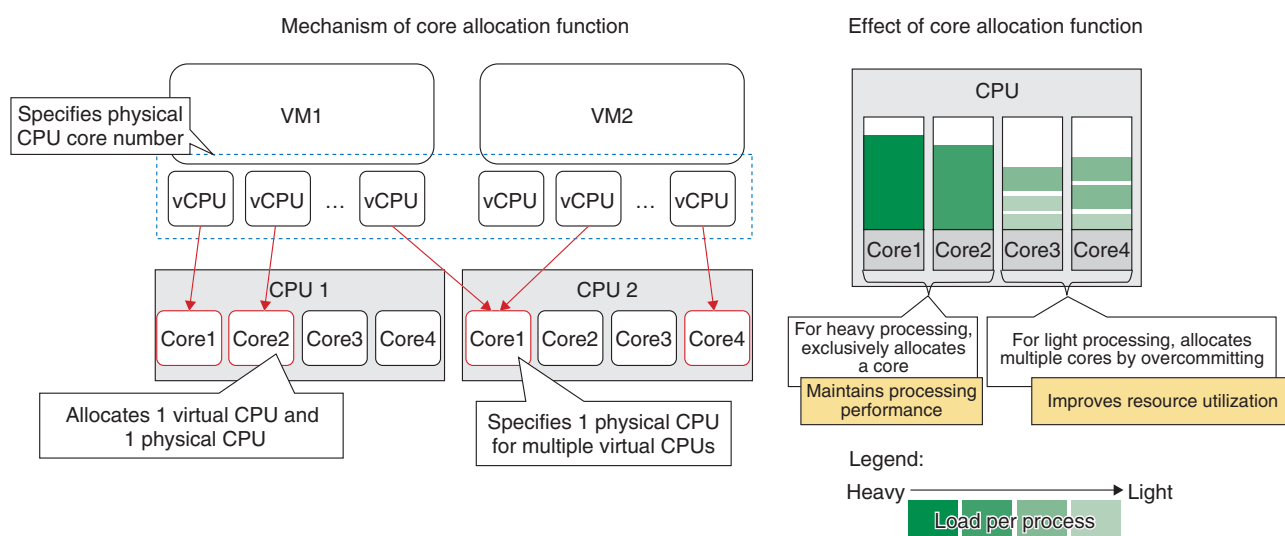


Fig. 3. CPU-core allocation function.

and ensure adequate processing time.

The third function specifies a certain core number in the physical CPU allocated to a virtual CPU. Performance can differ between CPU cores according to the network interface card or socket connected to each core, so specifying a core number enables core allocation that takes performance requirements into account. This function therefore makes it possible to maintain the processing performance of an allocated virtual CPU or avoid a drop in performance due to interruption of operating system processing.

To achieve this function, it must be set in KVM instead of VIM. However, in OpenStack, the VIM de facto standard, there is no CPU-core allocation function with a granularity that can be set by KVM. For this reason, VM management functions use OpenStack as the VIM and implement a function for allocating the CPU cores needed for operating packet switching system service functions.

5. Effect of VM management technology

The results of increasing the number of VMs generated by VM management technology and measuring the number of packets that can be normally processed while supplying large volumes of packets to those

VMs are shown in **Fig. 4**. The plot points show that throughput performance increases linearly as the number of VMs becomes larger when applying VM management technology. It can therefore be seen that application of this CPU-core allocation function generates VMs that ensure required performance.

This VM management technology can also allocate CPU cores at a quantity needed for an ACT/SBY configuration as well as for a VM used for auto-healing at the time of a hardware failure to ensure stable provision of services immediately after system recovery.

6. Future development

The VM management technology of MAGONIA is expected to be useful for virtualizing functions having severe latency requirements and requiring stable throughput as in the case of packet switching system service functions. We aim to develop these functions into common functions for supporting the future server infrastructure; therefore, our plan is to expand associated applications and engage in proposal activities with OpenStack and other open source software communities.

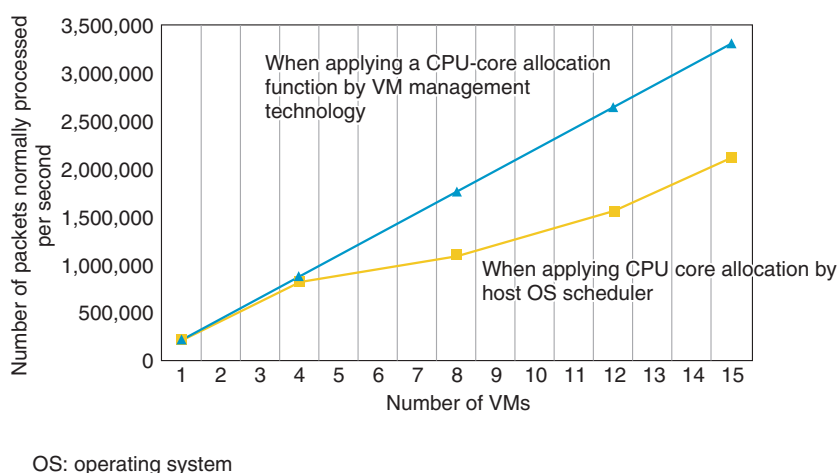


Fig. 4. Sample throughput by VM management technology.

References

- [1] T. Ito, "Accelerating the Demonstration and Adoption of NetroSphere," NTT Technical Review, Vol. 14, No. 10, 2016.
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201610fa1.html>
- [2] H. Kobayashi, T. Kitano, M. Okamoto, and T. Fukumoto, "MAGONIA (DPB: Distributed Processing Base) Applied to a Traffic Congestion Prediction and Signal Control System," NTT Technical Review, Vol. 14, No. 10, 2016.
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201610fa7.html>
- [3] NTT, "Sophisticated Network-node Integration Architecture," NTT R&D Forum, E-1, Feb. 2012.
<https://www.ntt-review.jp/library/ntttechnical.php?contents=ti1202rdfpa25e.pdf>
- [4] Soft Patch Panel - DPDK resources management framework,
<http://www.dpdk.org/browse/apps/spp/>



Makoto Hamada

Research Engineer, Server Network Innovation Project, NTT Network Service Systems Laboratories.

He received his B.S. and M.S. in mechanical engineering from Tokyo Institute of Technology in 1991 and 1993. He joined NTT Telecommunication Networks Laboratory in 1993, where he worked on the research and development of communication node systems, including Next Generation Network (NGN) session control servers. He is currently studying distributed systems and network functions virtualization. He is a member of the Institute of Electronics, Information and Communication Engineers.



Keiko Kuriu

Research Engineer, Server Network Innovation Project, NTT Network Service Systems Laboratories.

She received an M.S. in information and computer sciences from Nara Women's University in 2012 and joined NTT the same year. She began her career promoting distributed computing technology. She is currently studying network functions virtualization.



Eriko Iwasa

Research Engineer, Server Network Innovation Project, NTT Network Service Systems Laboratories.

She received an M.E. in electrical engineering from Waseda University, Tokyo, in 2009 and joined NTT the same year, where she was engaged in researching distributed computing technology. She is currently studying network functions virtualization.