White Paper TE-524263

Empowered by Innovation NEC

NEC's Carrier-Grade Cloud Platform

Deploying Virtualized Network Functions in Cloud

INDEX

1. Paving the way to Telecom Network Function Virtualization P. 3

2. Open Carrier-grade Hypervisor P. 3
   - Latency and jitter P. 4
   - Logging and tracing P. 6
   - Open eco system P. 7

3. Automation and Management P. 7

4. High degree of visibility P. 9

5. Conclusion P. 9

1. Paving the way to Telecom Network Function Virtualization

In the recent past, the IT industry has experienced a revolution with the advent of virtualization technology and cloud-based datacenters. Massive scale, reduced infrastructure cost, optimized operation and maintenance, agility in application delivery are some of the key factors that drove the widespread adoption. In contrast, telecom network functions and services are yet to leverage this trend and remain locked into a vendor proprietary hardware appliance model. Given the massive traffic growth foreseen in telecom carrier albeit a flattening ARPU, telecom operators are forced to consider technologies that achieves radical reduction in total cost of operation. To that end, following the trend set by IT cloud, telecom operators are now embracing a path leading to network functions virtualization (NFV) delivered through an operator cloud.

However, technologies and design principles that underlie a typical IT cloud applications cannot be seamlessly applied to telecom network services. There exist carrier-grade requirements that are specific to telecom network services. For example, packet processing latency and jitter can be critical to applications requiring service quality assurance such as voice call service. Service availability is another critical requirement for example in emergency call service. There is a service availability level difference between a mobile operator and an over-the-top (OTT) service provider. Therefore, the key question asked by the operators is: if virtualized network functions deployed in a cloud can provide the same level of performance, availability and scale as provided by the existing carrier-grade dedicated hardware based appliances? Furthermore, from an operation standpoint, the question is also about the complexity in provisioning and maintaining a cloud based network service. In other words, is network function virtualization ready for deployment?

NEC with a long and deep background in telecom networks as well as IT cloud brings the experience and innovation to deliver a carrier-grade cloud environment that addresses the above challenges. The two key parts of NEC carrier-grade cloud solutions are the carrier-grade hypervisor and the SDN service controller. The carrier-grade hypervisor provides a foundation for deploying network functions on a virtualized environment meeting the carrier-grade performance requirements. The SDN service controller provides the service management and operation platform that simplifies operation through automation and adds flexibility and agility to deliver on-demand network infrastructure as a service in a carrier-grade environment.

2. Open Carrier-grade Hypervisor

In supporting openness, NEC carrier-grade hypervisor is based on open-sourced KVM. Choice of KVM provides multiple advantages due to its openness. The use of KVM achieves customization and tuning flexibilities and in-depth visibility while providing common API to applications extensions running on the hypervisor. The above KVM features allowed NEC to enhance the KVM to fill the gaps that are critical to carrier-grade requirements. In the following, we discuss some of these key gaps and enhancements.

Latency and jitter

In any virtualization environment, hypervisor layer adds a significant amount of processing overhead. Furthermore, application/virtual machines running on common hardware have to share hardware resources such as CPU, Memory, Disk, I/O etc without any guarantee in terms of time-sensitive availability of resources. For example, if an application fails and restart process is started, most of capacity of CPU will be instantaneously occupied by the restart process and performance of another application running on the same hardware will be affected. Such type of resource allocation can lead to unpredictable performance of any given application. Unpredictability can lead to sudden increase in delay and associated jitter thereby significantly impacting the quality-of-service requirements.

To resolve such problems, NEC extended the KVM with three focal points of enhancements as shown in the figure below.

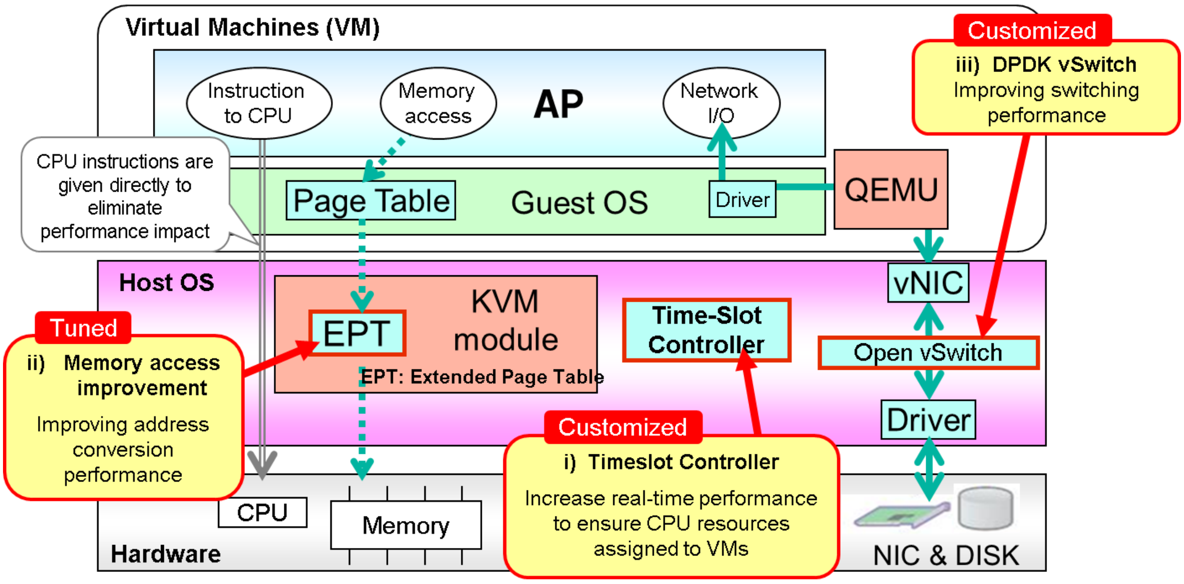

Fig.1 Customization and tuning to KVM

The first important extension is in introducing Timeslot controller which enables time-sensitive resource allocation for CPU along with priority control with capping bandwidth for network and disk I/O. Timeslot controller, as shown in figure below, ensures resources are committed in a timely manner to match the functional requirements of the network functions being virtualized. Further, it also prevents any sudden over-commit of resources which can take the system into an unstable state. The predictable bounds achieved through Timeslot controller meets the delay/jitter at packet level as required in any carrier-grade environment.

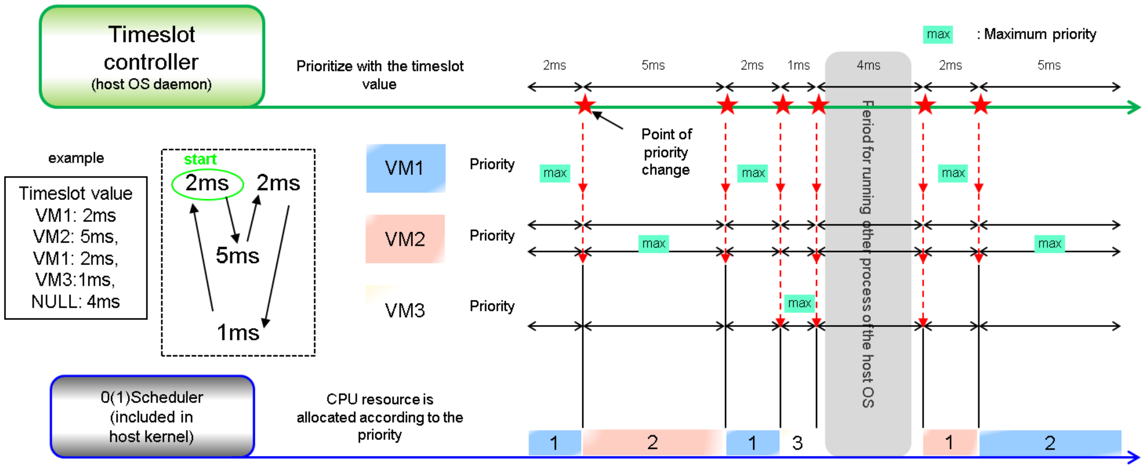

Fig.2 Time slot controller concept

Another factor which may cause latency and jitter due to the additional layer of hypervisor is I/O memory management unit (IOMMU), which is used in data packet processing in virtualized environment. It is improved by NEC's tuning for memory acceleration by right sizing of page specific to network functions requirements. As shown in figure below, increasing the page size from standard 4KB to 2MB significant improves performance by higher hit rate and lesser conversion steps.

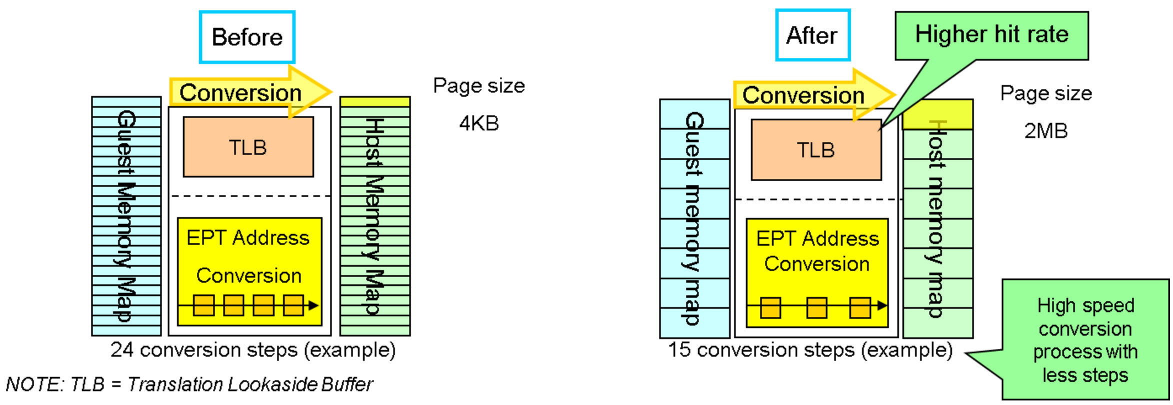

Fig.3 Memory access enhancement by Huge Page

To further improve the packet level performance from delay and jitter standpoint, NEC has collaborated with Intel in adding DPDK support in KVM. NEC contribution specifically targets network functions in supporting high packet I/O and high bandwidth inter-VM communication as shown in figure below.

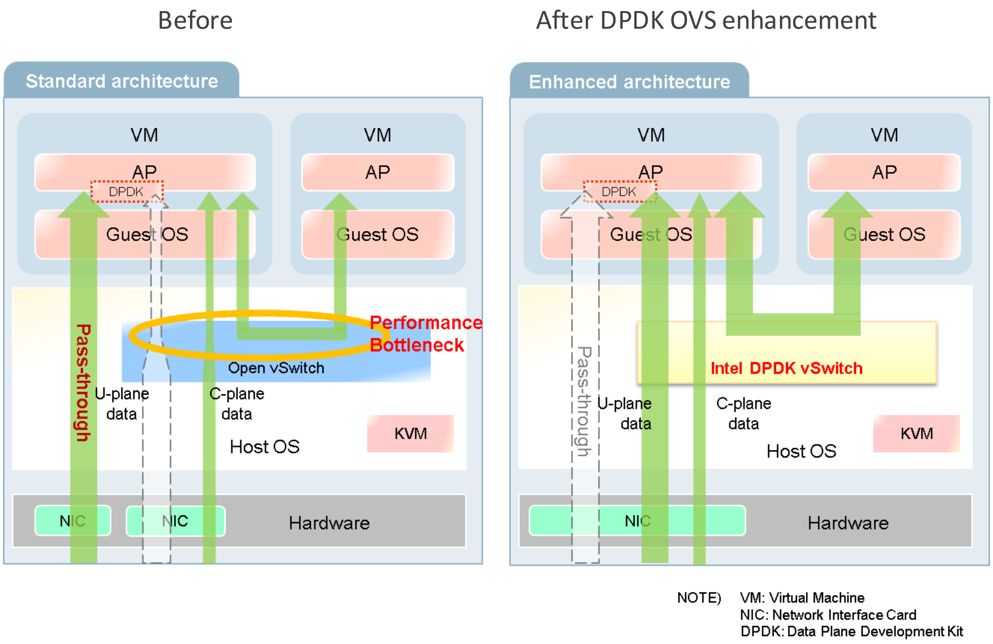

Fig.4 DPDK OVS enhancement

Logging and tracing

In a typical virtualization environment, there is a limited visibility into the underlying environment in terms of assessing failure conditions. Generally, Hypervisors do not provide sufficient logging data for analyzing virtualization layer and hardware resources assigned to each VM. Limited visibility makes hard to correlate failure condition and perform any root cause analysis with application level logging alone. Without the ability of assess the situation it becomes difficult to develop autonomous healing features required in a carrier-grade environment for delivering high availability. Furthermore, support and maintenance will become complex and time consuming making virtualization a difficult proposition from an operation standpoint.

To that end, NEC added and enhanced logging and tracing functions to co-relate across multiple layers: application, hypervisor and hardware. Such functions provides flight-recorder-like function execution histories in QEMU-KVM process and usage status and history of CPU, IO resource, etc. used for a VM (as shown in figure below), which largely improves analysis depth and realizes failure identification agility. The enriched logging and tracing data co-relation gives an insight into what occurred to physical to application layers, and enables operators to shorten time to deeply analyze even inside virtualization layer and physical layer of virtualization environment.

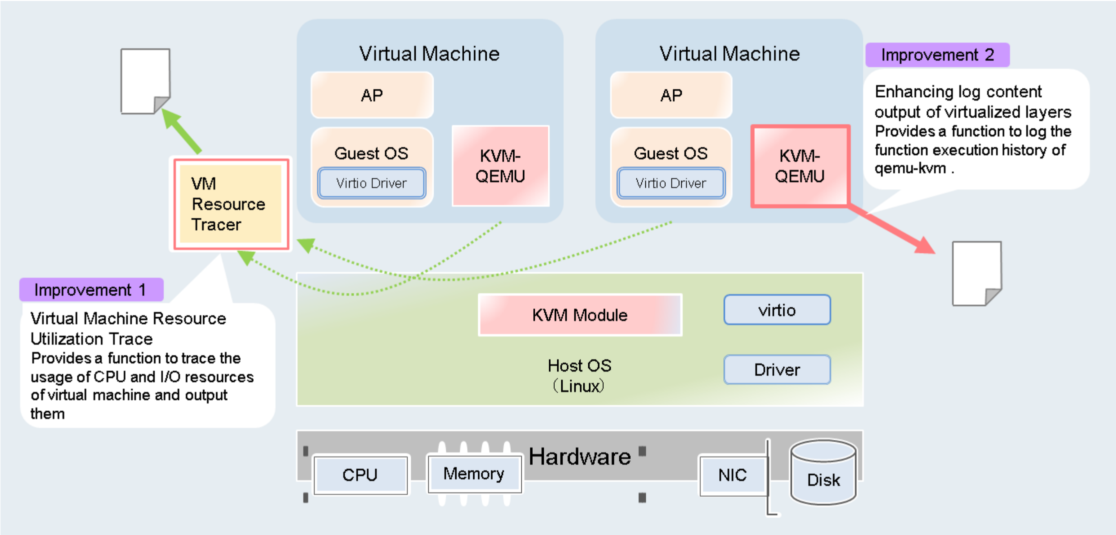

Fig.5 Logging and tracing enhancement

Open eco system

The enhanced KVM works on any commercial-off-the-shelf (COTS) server having Intel Architecture. It opens the platform to any third party vendor's virtualized network functions. The enhanced KVM has been used for NEC's virtualized mobile packet core (vEPC) and has been proven to meet the stringent requirements of packet core functionalities. Third party's NFV applications can easily leverage the benefits offered by the enhanced hypervisor. The carrier-grade hypervisor from NEC builds the foundation for an open common virtualization infrastructure where multiple multi-vendor network functions can be deployed and delivered. Such an open platform will also support the new paradigm of service chaining network functions in creating a rich, flexible and dynamic telecom network services.

3. Automation and Management

A necessary part of any service delivery platform (virtualized or non-virtualized) is the automation and management layer. The SDN Service Controller from NEC provides the key features that enable automation and management of network functions in a virtualized environment. For example, SDN Service Controller manages and controls virtual appliances based on monitoring both physical and virtual resources usage status. SDN Service Controller allocates virtualized service appliances to appropriate physical resources considering capacity demand and processing/traffic behavior. SDN Service Controller also manages reallocation of virtualized services in case of hardware failure. The key object of SDN Service Controller is to maximize the resource efficiency while meeting the performance and availability requirements of the network services.

The second important goal of SDN Service Controller is to minimize the operational complexity related to provisioning, maintenance and support. Telecom services operators traditionally need complex configurations and planning related to meeting future traffic trend and capacity expansion. SDN Service Controller streamlines and automates the operation process through "Service Description" template definitions as shown in figure below.

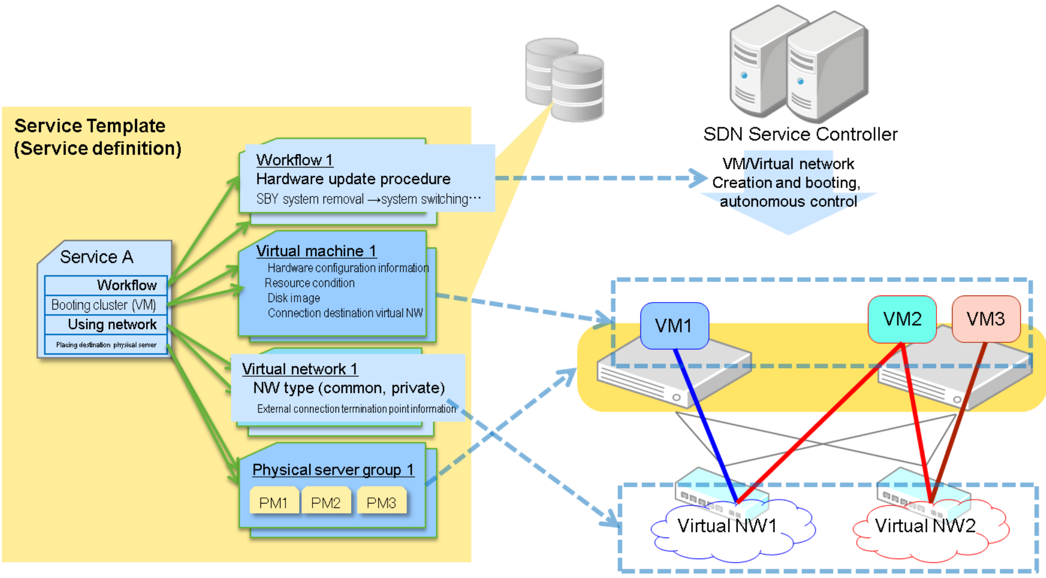

Fig.6 "Service Description" template

The service templates define automation related workflows. Various conditions and actions can be set in the workflow, for example, "new C-Plane function VM for S-GW is to be created to expand its capacity seeing CPU allocated to the VM exceeds 60% usage for 5 minutes". In executing such workflow, necessary VM representing specific network function can be created according to traffic volume or processing capacity demands. The example illustrates how SDN Service Controller can enable just-in-time automatic scaling to capacity demand in an optimized fashion as shown in figure below.

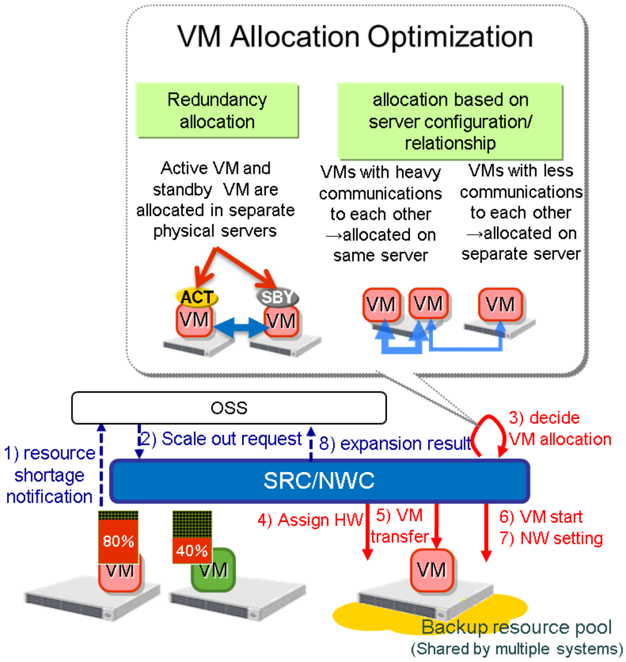

Fig.7 Optimized VM allocation

Automated capacity provisioning brings significant advantages. Operators now do not have to pre-configure capacity to handle peak time traffic or based on two-to-three year forecast on traffic growth. There is no requirement for lead time in arranging additional dedicated hardware or need for skilled engineers for initial setup and provisioning.

4. High degree of visibility

In order to fulfill the carrier-grade requirement, NEC SDN Service Controller plays an important role to visualize co-relation of application and hardware with real-time status monitoring of each application and hardware. Also SDN Service Controller in working with carrier-grade hypervisor detects real-time load status of hardware device, e.g. CPU, IO, memory, etc., understands each hardware resource's availability. SDN Service Controller provides intelligent resource allocation of VM to a physical machine which has sufficient capacity and meets the optimization constraints in terms of inter-VM traffic and redundancy types.

5. Conclusion

Most of operators in global have consensus that network function virtualization is a key driver in achieving openness and economy of scale. NEC's SDN Service Controller and the enhanced KVM address operators' existing pains in achieving required performance and availability in an open virtualized environment. Meanwhile, it opens up new revenue opportunities for telecom operators through dynamic and flexible chaining of services and quick and lean service launch based on assured performance and carrier-grade availability.

Abbreviations

AP	Application
API	Application Programming Interface
ARPU	Average Revenue Per User
CPU	Central Processing Unit
DPDK	Data Plane Development Kit
I/O	Input/Output
KVM	Kernel-based Virtual Machine
NIC	Network Interface Card
QEMU	Quick Emulator
SDN	Software-defined networking
VM	Virtual Machine