

Integrated Management System for Distributed Micro-Datacenters

Byeongsik Kim*, Bhumchul Lee*

*Smart Network Research Department, Electronics and Telecommunications Research Institute (ETRI),

218 Gajeong-no, Yuseong-gu, Daejeon, KOREA

bskim25@etri.re.kr, bclee@etri.re.kr

Abstract— In this paper, we introduce a Micro-Datacenter (MDC) and Integrated Management System (IMS) that controls and manages the MDCs for supporting several services to users. The MDC is a small and rack-based datacentre system that designed to solve different problems or to handle different workloads that cannot be handled by traditional datacenters. The IMS controls and manages a multitude of MDCs to provide users with several services. And also, we introduce a case study applied to the Korea Advanced Research Network (KOREN).

Keywords— Virtualization, Datacenter, Cloud operating system, OpenStack, Virtual machine, KOREN

I. INTRODUCTION

A Datacenter (DC) is a physical or virtual infrastructure used by organizations or enterprises to centralize computing, storage, and networking devices. It generally requires extensive redundant or backup power supply systems, redundant networking connections, environmental control systems (e.g., cooling, fire suppression), and policy-based security systems for the enterprise's core applications. Datacenter management involves ensuring the reliability of both the connections to the datacenter as well as the mission-critical information contained within the datacenter's storage. It also entails efficiently placing application workloads on the most cost-effective compute resource available.

The Micro-Datacenter (MDC) mentioned in this paper is a smaller, rack (or box)-based datacenter system that designed to solve different problems or to handle different workloads that cannot be handled by traditional datacenters. Whereas an ordinary container-based datacenter hosts dozens of servers and thousands of virtual machines (VMs) within a shipping container, an MDC includes fewer than 10 servers and less than 100 VMs in a single rack. Therefore, MDCs don't require excessive power supply and environmental control systems.

Generally, MDCs are deployed into the geographically distributed areas. The MDC provide the users with several services by itself and also several MDCs can cooperate to provide specific services. For example, the services provided by MDCs are Contents Delivery Network (CDN) and Virtual Desktop Infrastructure (VDI) services. To provide users with these services, Integrated Management System (IMS) should control and manage the several MDCs distributed geographically. One of the most important functions of IMS is

to provide the service users with the several services from the MDCs that is located in the nearest to the users.

MDCs are constructed by using OpenStack that is a kind of cloud operating systems. OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds and controls large pools of compute, storage, and networking resources throughout an MDC. All of the resources are managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

In this paper, we introduce the MDC developed based on OpenStack and Integrated Management System that controls and manages a multitude of MDCs. And also, we introduce a case study applied to the Korea Advanced Research Network (KOREN).

II. RELATED WORKS

In this section, we briefly introduce the previous works related to this paper. These works include OpenStack, Content Delivery Network service, Virtual Desktop Infrastructure service and Korea Advanced Research Network.

A. OpenStack

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface [1]. OpenStack is a set of software tools for building and controlling cloud computing platforms for public and private clouds and is managed by the OpenStack Foundation, a non-profit which oversees both development and community-building around the project.

The OpenStack project is an open source cloud computing platform that supports all types of cloud environments. The project aims for simple implementation, massive scalability, and a rich set of features. Cloud computing experts from around the world contribute to the project. OpenStack provides an Infrastructure-as-a-Service (IaaS) solution through a variety of complementary services. Each service offers an application programming interface (API) that facilitates this integration. The following table provides a list of OpenStack services.

TABLE 1. OPENSTACK SERVICES

Service	Project	Description
Dashboard	Horizon	Provides a web-based portal to provide system administrators a look at what is going on in the cloud, and to manage it as needed.
Compute	Nova	Deploys and manages large numbers of virtual machines and other instances to handle computing tasks.
Networking	Neutron	Provides the networking capability for OpenStack and helps to ensure that each of the components of an OpenStack deployment can communicate with one another quickly and efficiently
Object Storage	Swift	Stores and retrieves arbitrary unstructured data objects via a RESTful, HTTP based API.
Block Storage	Cinder	Provides a block storage component, which is more analogous to the traditional notion of a computer being able to access specific locations on a disk drive.
Identity Service	Keystone	Provides identity services for other OpenStack services and multiple means of access, meaning developers can easily map their existing user access methods against Keystone.
Image Service	Glance	Provides image services to OpenStack and allows these images to be used as templates when deploying new virtual machine instances
Telemetry	Ceilometer	Allows the cloud to provide billing services to individual users of the cloud and keeps a verifiable count of each user's system usage of each of the various components of an OpenStack cloud.
Orchestration	Heat	Allows developers to store the requirements of a cloud application in a file that defines what resources are necessary for that application.

Figure 1 shows the conceptual architecture of a typical OpenStack environment.

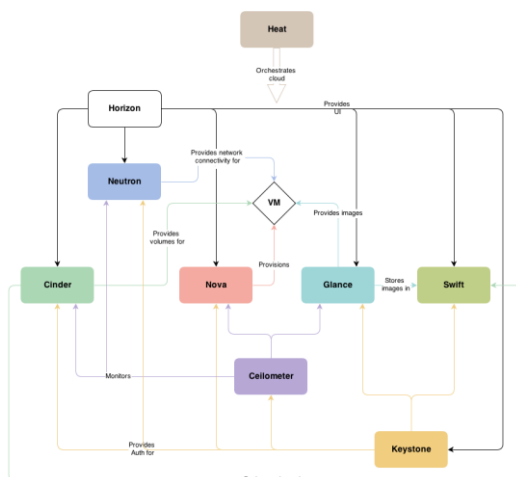


Figure 1. OpenStack Conceptual Architecture

B. CDN and VDI Services

Content Delivery Network (CDN) is a network of distributed servers that store content and deliver it to a user based on the geographic locations of the user, the origin of webpage and a content delivery server [2]. The concept of CDN is to keep content on the CDN server closet to the end user to speed up the page loading time and avoid global network bottlenecks.

Virtual Desktop Infrastructure (VDI) refers to the process of running a user desktop inside a virtual machine that lives on a server in the datacentre [3]. It's a powerful form of desktop virtualization because it enables fully personalized desktops for each user with all the security and simplicity of centralized management. VDI enables customers to streamline management and costs by consolidating and centralizing the desktops while delivering end-users mobility and the freedom to access virtual desktops anytime, from anywhere, on any device. It's important to understand, however, that VDI is only one form of desktop virtualization.

C. KOREN

Korea Advanced Research Network (KOREN) is a non-profit test-bed network infrastructure established for facilitating research and development and international joint research cooperation [4]. It provides quality broadband network test-bed for domestic and international research activities to the industry, academia, and research institutions, enabling testing of future network technologies and supporting R&D on advanced applications.

KOREN enables internetworking and field testing, which cannot be performed over commercial networks and validation research on high-capacity application services. KOREN reduces R&D cost for researchers including provision of free Gbps range network connection service and enables international research cooperation through internetworking with international research networks and is the only research test-bed network that allows high-capacity traffic solely for testing purposes.

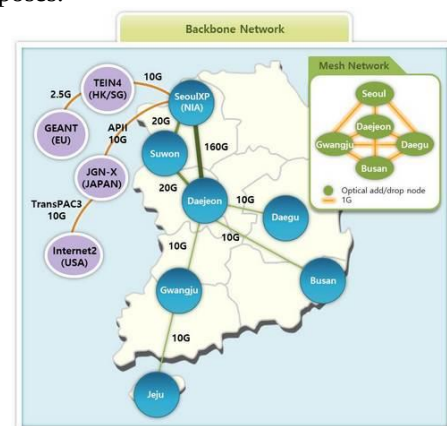


Figure 2. KOREN's Backbone Network

KOREN's backbone network is connecting 6 large cities throughout Korea (Seoul, Suwon, Daejeon, Gwangju, Daegu, Busan) at the speed of 10Gbps ~ 160Gbps (Figure 2). In order

to provide customized services for each different user environment such as research, test, and pilot service KOREN is efficiently operated in terms of R&D network, test-bed network, and service network. It is also connected with other networks such as International Research Network (APII/TEIN), Broadband Convergence Network (BcN) pilot network.

III. MICRO-DATACENTERS AND INTEGRATED MANAGEMENT SYSTEM

A. Micro-Datacenters (MDCs)

MDCs are constructed by physical servers and switches. OpenStack is used to provide an Infrastructure-as-a-Service (IssS) solution through the physical devices.

As OpenStack is highly configurable to meet different needs with various computing, networking, and storage options, we can choose our own OpenStack adventure using a combination of core and optional services. Figure 3 shows the rack-based MDC system. The control node runs the Identity service, Image Service, management portions of Compute and Networking, Networking plug-in, and the dashboard. It also includes supporting services such as a SQL database, message queue, and Network Time Protocol (NTP). Optionally, the control node runs portions of Block Storage, Object Storage, Orchestration, Telemetry, Database, and Data Processing services. These components provide additional features for this environment.

The network node runs the Networking plug-in and several agents that provision tenant networks and provide switching, routing, NAT, and DHCP services. This node also handles external (Internet) connectivity for tenant virtual machine instances.

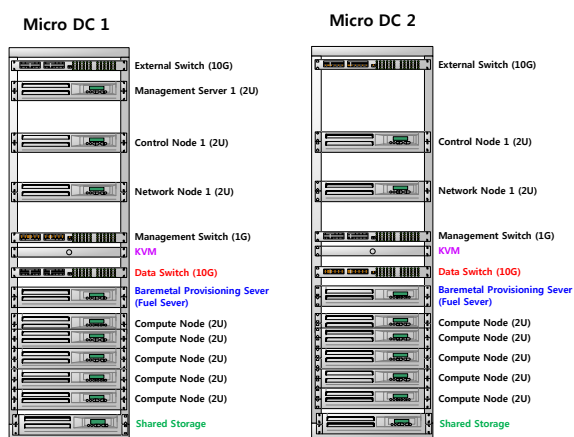


Figure 3. Micro-Datacenters' Rack Architecture

The compute nodes run the hypervisor portion of Compute that operates tenant virtual machines or instances. By default, Compute uses KVM as the hypervisor. The compute nodes also run the Networking plug-in and an agent that connect tenant networks to instances and provide firewalling (security groups) services. Optionally, the compute nodes run a Telemetry agent to collect metrics. Also, they can contain a

third network interface on a separate storage network to improve performance of storage services.

The shared storage nodes contain the disks that the Object Storage service uses for storing accounts, containers, and objects. More than two of these nodes can be used and contain a second network interface on a separate storage network to improve performance of storage services.

B. Integrated Management System (IMS)

IMS is a management server for policies, processes, and procedures used to ensure that the MDCs can fulfil all tasks required to achieve their objectives. The main functions of IMS are in Table 2. The functions are part of it, not the whole of it.

TABLE 2. IMS MAIN FUNCTIONS

Functions	Descriptions
Service Region Insert/Delete	Register/unregister geographic locations of service regions
MDC Insert/Delete	Insert/Delete MDC in(to) the service regions
Flavour Create/Delete	Create/Delete the flavour for virtual machine(server) creation
Service Create/Delete	Create/Delete the specific service (e.g., CDN or VDI)
Region Insert/Delete in Service	Insert/Delete the region in(to) a specific service
Virtual Network Create/Delete	Create/Delete a virtual (Tenant) network
Virtual Router Create/Delete	Create/Delete a virtual router in a tenant network
Firewall Create/Delete	Create/Delete a firewall in a tenant network
Virtual Server Create/Delete/Start/Stop/Restart	Create/Delete/Start/Stop/Restart a specific virtual server(machine)
Virtual Server Monitoring	Monitor a virtual server(machine) (e.g., CPU/Memory/Disk usage)
Software Auto-Installation	Install automatically specific application software into a virtual server(machine)
Virtual Server Scale-Out/Scale-In	Scale-Out/Scale-In automatically virtual server(machine)

IV. APPLICATION INTO KOREN

In this section, we describe the application scenario and result of MDCs and IMS into KOREN and also show the example of CDN and VDI services.

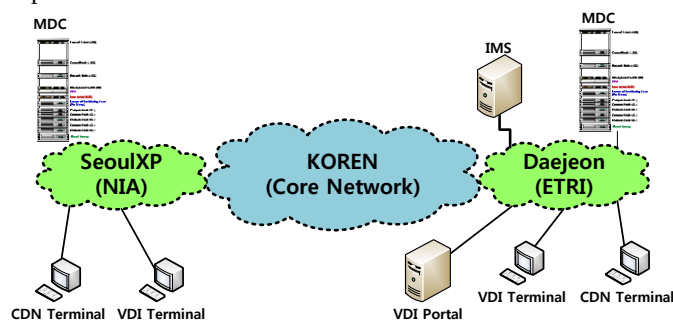


Figure 4. Test Network using KOREN

Figure 4 shows the test network that two MDCs are deployed and distributed geographically in Seoul and Daejeon, KOREA. The interconnection between Seoul and Daejeon is connecting to each other at the speed of 160Gbps. The IMS manages and controls two MDCs to provide service users with CDN and VDI services.

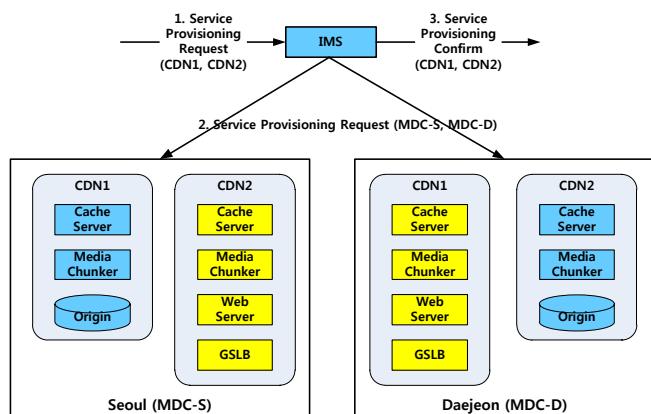


Figure 5. CDN Service Infrastructure

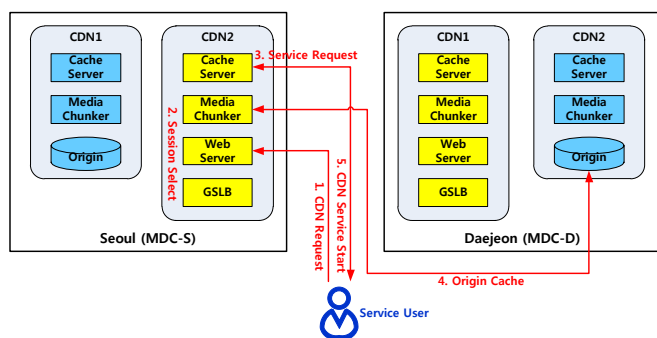


Figure 6. CDN Service Scenario

Figure 5 represents the infrastructure for CDN service. The infrastructure is composed of cache servers, media chunkers, Web servers, Global Server Load Balancers (GSLBs). Firstly, CDN service provider requests the provisioning of the service to the MDCs through the IMS. And then, each MDC initializes the components for the service and confirms the initialization setup. Finally, the IMS sends the conform message to the service provider's operator.

After the procedures in Figure 5, as shown in Figure 6 service user requests the CDN service to Web server. At this point of time, MDC-S checks if the contents requested from the user are in MDC-S itself. If MDC-S does not have the contents, MDC-S downloads the contents from the origin server in MDC-D. After that, MDC-S, the nearest MDC, provides the CDN service to the user.

Figure 7 shows the scenario of VDI service at the condition of Figure 4. VDI service provider creates the infrastructure throughout VDI portal (or UI) and allocates the virtual machine to the customer. The customers who want to use the VDI service check the virtual machine allocated to themselves. And then, the virtual machine is moved and executed to the nearest location of the customer.

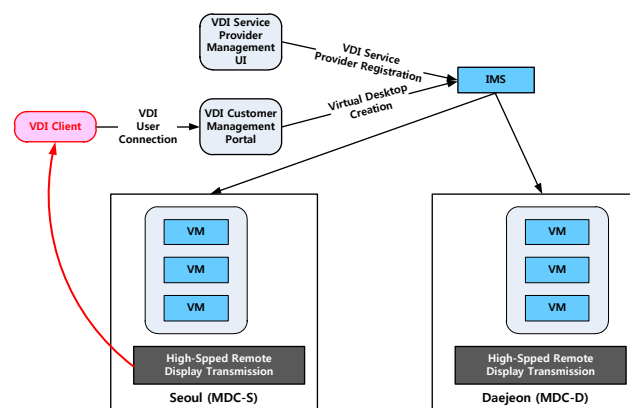


Figure 7. VDI Service Scenario

V. CONCLUSIONS

This paper describes the results of Micro-datacenters developed by using OpenStack and also the Integrated Management System managing and controlling the multitude of the Micro-datacenters to provide users with services.

In the near future, we have the plan to develop the more services, e.g., Content Centric Network (CCN) service and Network Function Virtualization (NFV) service.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0101-15-233, Smart Networking Core Technology Development)

REFERENCES

- [1] OpenStack Homepage, <http://www.openstack.org/>
- [2] Scot Hull, *Content Delivery Networks : Web Switching for Security, Availability, and Speed*, McGraw-Hill, 2002.
- [3] What is VDI ?, <https://www.citrix.com/glossary/vdi.html>
- [4] KOREN Introduction, <http://www.koren.kr>



Byeongsik Kim is a principle member of engineering staff in Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. He received his Ph.D., M.S., and B.S. degrees in computer engineering from Chungnam National University in 2001, 1997, and 1995, respectively. He was a visiting researcher at National Institute of Standards and Technology (NIST), USA in 1997. His recent research areas include network resource management, network QoS, cloud operating systems, and network virtualization.



Bhum-Cheol Lee received the B.S. degree from Kyung-Hee University, Seoul, Korea, in 1982, and the M.S. and Ph.D. degrees from Yonsei University, Seoul, in 1983 and 1997, respectively. From 1983 to 1995, he was with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, as an Interface, Switching, Link, and Network Synchronization Engineer. Currently he is the Project Leader of the Smart Node Platform Research Section, ETRI. His research interests are synchronization, line coding, analog and digital circuit design, and network virtualization.