Latest Trend of OpenStack, Open Source Software for Infrastructure as a Service, and NTT DATA's Activities

Masayuki Hanadate

Abstract

NTT DATA is researching and developing cloud infrastructure technology utilizing OpenStack. This article explains how OpenStack works and introduces some of NTT DATA's recent projects including OpenStack component development, OpenFlow collaboration technology development, cloud security component development, and the use of Swift.

1. OpenStack

1.1 Overview

OpenStack [1] is open source software (OSS) for centrally and efficiently operating and managing physical servers and devices comprising the cloud. It is under collaborative development by more than 160 companies around the world (as of April 26, 2012). The latest version (version 6, Folsom) was released in September 2012. Moreover, OpenStack is included in Ubuntu 12.04LTS, a Linux distribution.

NTT DATA has been participating in the OpenStack project since 2010 as one of the project startup members. It is conducting research and development (R&D), as well as promoting cloud services using OpenStack together with NTT's R&D laboratories.

The service model offered by OpenStack is infrastructure as a service (IaaS), which is a cloud service model suggested by the National Institute of Standards and Technology (NIST) of the USA [2].

OpenStack provides users with virtual machines that run on a hypervisor such as kernel-based virtual machines (KVMs) and XenServers. Any OpenStack users can access their virtual machines through networks and operate their computing resources (e.g., central processing units (CPUs), memory, hard disk drives, and IP (Internet protocol) addresses) allocated to their virtual machines.

The features of OpenStack are as follows:

(1) Multiple tenants

A single physical machine can host multiple virtual machines belonging to different cloud computing users. This reduces redundant computing resources leading to lower physical machine costs.

(2) On-demand self-service

Cloud computing users can manage their virtual machine operations (e.g., starting and stopping virtual machines) via a web-based graphical user interface (WebGUI), Amazon EC2-compatible application programming interface (API), and OpenStack API. Using these interfaces, users can agilely start their services without having a cloud computing administrator (provider). Furthermore, because cloud computing users actually carry out some operation management tasks that used to be conducted by the cloud provider, the cloud provider's management costs can be lower.

(3) Live migration

Live migration is a function that transfers values stored in a physical memory to another physical memory without interrupting the physical machines containing these memories. This interruption-free replacement of physical machines improves maintainability.

(4) Security

OpenStack has the following basic security functions: authentication of cloud computing providers or users, the hash-based message authentication code (HMAC) specified in the Amazon EC2-compable

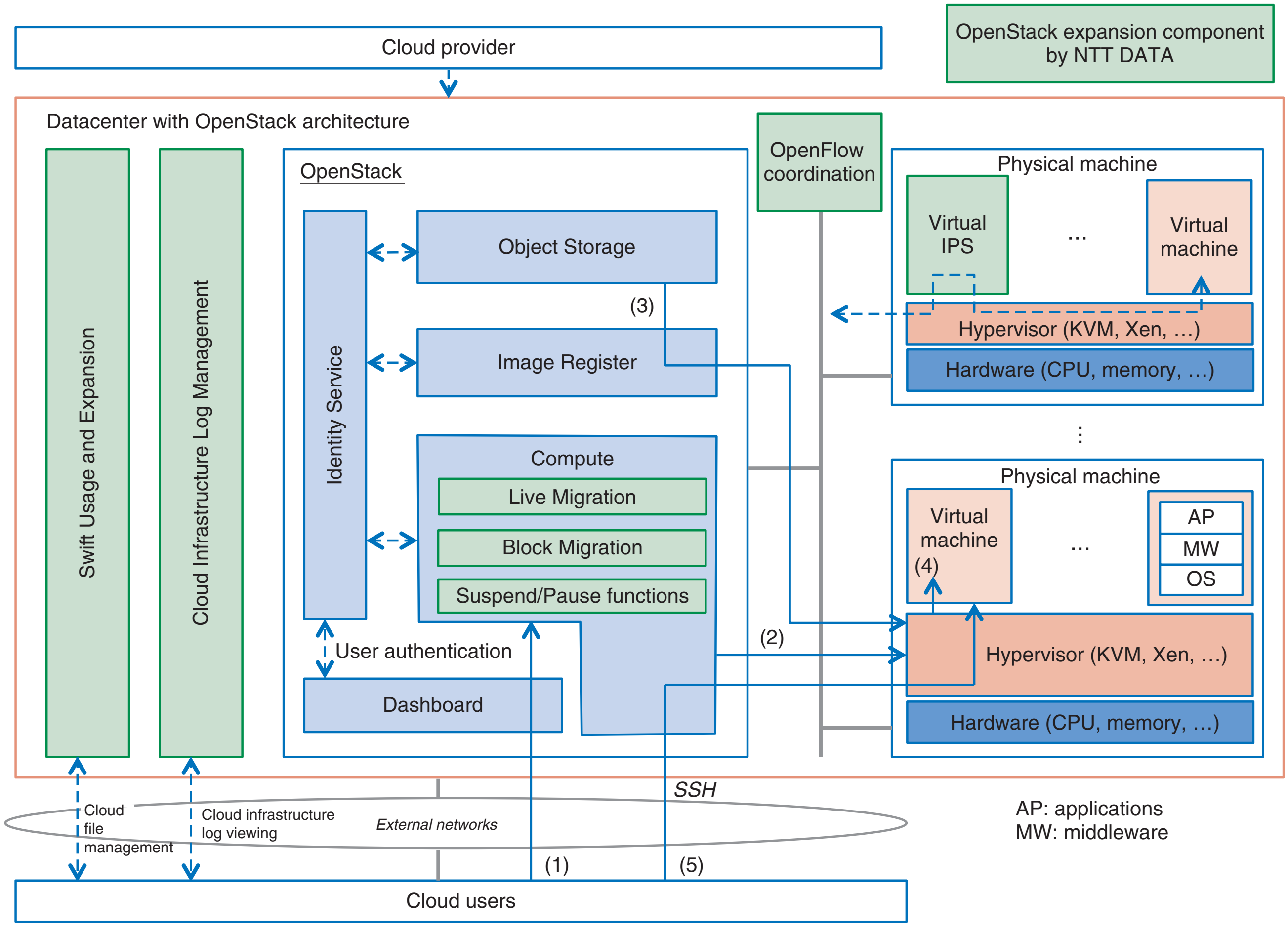


Fig. 1. Configuration of OpenStack.

API, an iptables*-based firewall to isolate virtual machines of different tenants, a virtual private network (VPN), and encryption of communications between a user terminal and a virtual machine by SSH (secure shell).

1.2 Operation

OpenStack comprises multiple components [3], [4] and provides IaaS functions by coordinating them (**Fig. 1**). These components (and their respective code names) are as follows:

The Compute (Nova) component provides computing resource management, computing resource allocation, and message transfer. In the computing resource management, the Compute component manages physical resources (e.g., CPUs and memory) in OpenStack. In the computing resource allocation, the Compute component determines which physical resources should be allocated to the cloud computing user; and in the message transfer, the cloud computing user sends and receives cloud control messages (e.g., to invoke, terminate, or pause the virtual machine).

The Object Storage (Swift) component stores the template information of some available virtual machines (as VM images).

The Image Registry (Glance) component reads the VM image selected by the Compute component from the Object Storage component and transmits it to the physical machine.

The Identity Service (Keystone) component centrally stores identities (IDs) and passwords of the cloud computing users and providers. It also provides user authentication and component authorization.

The Dashboard (Horizon) component provides the WebGUI to the cloud computing users.

* iptables is the name of an application program.

As a brief introduction to the OpenStack process, we describe an example of the procedure for invoking a virtual machine by using the abovementioned components.

(1) Receiving a message: A cloud computing user submits a request to the Compute component to start a virtual machine. At this point, the user selects the computing resources (CPUs, memory amount, disk capacity, etc.), virtual machine types (operating system (OS), etc.), and other options.

(2) Allocating computing resources: The Compute component decides the physical machine appropriate for the requested computing resources, as well as IP addresses to be used. It then notifies the hypervisor for the selected physical machine to start virtual machine operations.

(3) Loading the VM image: The hypervisor requests the Image Registry to transmit the VM image. The Image Registry identifies the VM image to invoke from among several VM images stored in the Object Storage and transmits this identified image to the hypervisor.

(4) Starting the virtual machine: The hypervisor invokes this transmitted VM image.

(5) Using the virtual machine: The cloud computing user accesses the started virtual machine through networks and uses the computing resources of the virtual machine.

2. NTT DATA's projects

NTT DATA's development efforts related to OpenStack involve the following functions (Fig. 1).

(1) We developed the Live Migration, Block Migration, KVM Pause/Suspend functions, as well as other functions, which we contributed to the OpenStack community.

(2) In response to security concerns about cloud computing, which is the top issue hindering cloud computing introduction into the Japanese market, we developed some security functions (the integrated log management system of the cloud computing infrastructure and the virtual intrusion prevention system (virtual IPS)), which complement the default security functions of OpenStack.

(3) We are working to coordinate NTT DATA's OpenFlow Controller with OpenStack.

(4) Using the Object Storage (Swift), we are working to establish technology (Swift Reference Architecture) for building a highly reliable peta-byte-class distributed storage system at a low cost.

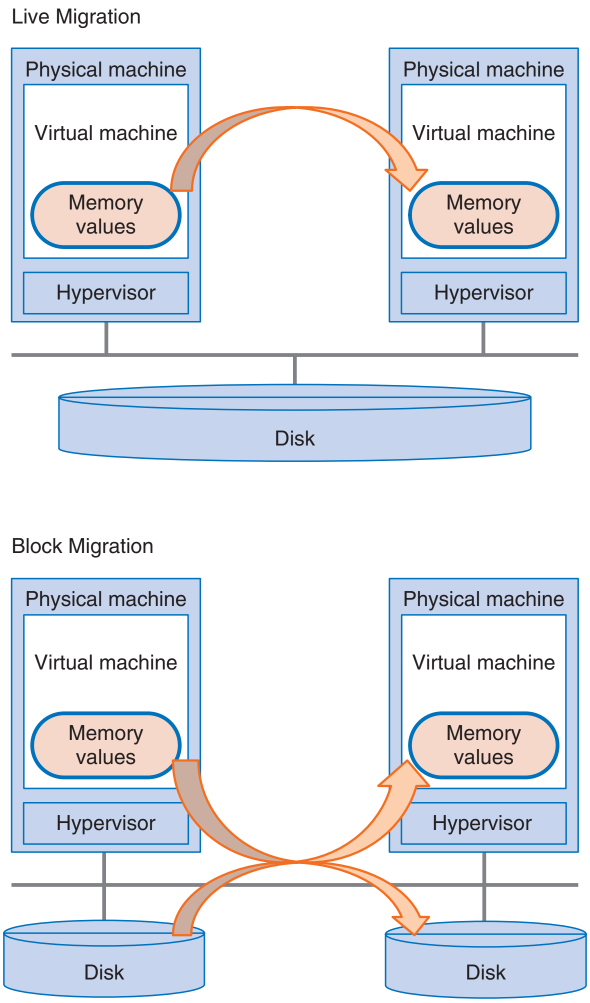


Fig. 2. Mechanisms of Live Migration and Block Migration.

2.1 Block Migration

Block Migration, released in OpenStack Version 4 (Diablo), is a function for moving the virtual machine including its VM image and memory contents, on one physical machine (the original physical machine) to another (the target physical machine) (**Fig. 2**).

Live Migration transfers memory contents but not disk contents, so both the original and target physical machines must access the same disk. Therefore, it is impossible to execute Live Migration if the original and target physical machines cannot access the same disk (e.g., if they are in different datacenters). By contrast, Block Migration transfers both memory and disk contents on one physical machine to another. This makes it possible to move the virtual machine between different datacenters. By combining Live Migration with Block Migration, we expect to improve the maintainability of virtual machines.

2.2 KVM Pause/Suspend support

This function was released in OpenStack Version 4 (Diablo). The procedure for pausing or suspending a virtual machine is as follows: (1) save the virtual machine, (2) stop the virtual machine, and (3) restart the virtual machine stored in step (1). However, up until OpenStack version 3 (Cactus), it was impossible to pause or suspend virtual machines because the status of a virtual machine was not stored on a disk or in memory. Therefore, NTT DATA developed the Pause function to store the virtual machine status in memory and the Suspend function to store the virtual machine status to disk.

2.3 Infrastructure Integrated Log Management for cloud computing

Because cloud computing users can directly access their own virtual machine, they can also read the virtual machine's log file (e.g., the OS, middleware, and application logs) by themselves. However, viewing this cloud computing infrastructure log file (e.g., virtual environment log and OpenStack log (such as that of the Compute component and Image Registry component)) is not an easy task because a single log file contains all of the logs for the cloud users and each log record requires independent access control.

In response, NTT DATA has developed a cloud computing infrastructure log management system that can extract the logs that are relevant to a specific user in real time from the single log file. This system enables cloud users to understand the status of their cloud computing infrastructure operation through the logs in a safe and speedy manner. This reassures them about the safety of the cloud computing infrastructure, which is not readily visible to users.

2.4 Virtual Intrusion Prevention System

Virtual IPS is the system that monitors communication packets in Layer 3 or higher, detects unauthorized or malicious packets, and prevents their transfer to virtual machines. When an existing IPS product is being installed in a cloud computing system accessed and shared by multiple users, it is necessary to set a different signature file for each user in the existing IPS product, and we must consider user requirements for system performance and security level when making these signature files.

Moreover, in the case of monitoring the communication of virtual machines residing on one physical server, we must monitor the hypervisor on the physical server. However, the existing IPS product cannot monitor the communication of virtual machines inside the hypervisor. Therefore, when this communication is being monitored using the existing IPS product, the virtual machines must communicate through the network outside the physical machine monitored by the existing IPS product.

To solve these inefficiencies, NTT DATA provides a virtual machine equipped with Suricata, an OSS IPS product, as a virtual IPS. For communication via this virtual IPS, we are currently developing technology for monitoring a variety of communication packets, which are exchanged among virtual machines and other devices, for every virtual machine.

2.5 Swift Reference Architecture

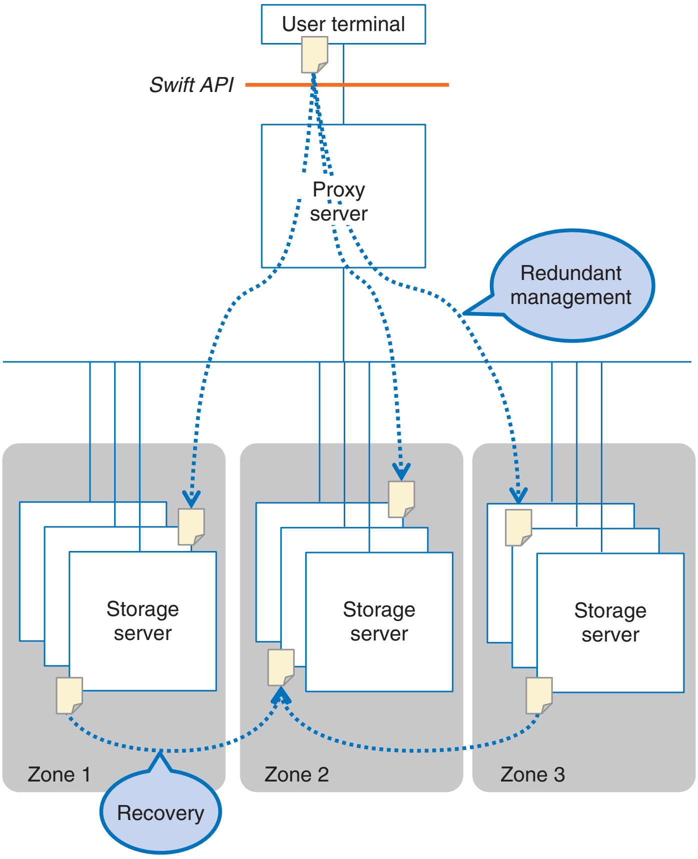
Swift has been developed by US Rackspace for its cloud file storage service. Its main components (**Fig. 3**) are a proxy server and a storage server. The storage server stores objects (which are text files), image files, movie files, and directory metadata files. The proxy server mediates the communication between a cloud computing user and the storage server using Swift API.

There is usually more than one storage server, and these servers are managed as a group called a zone. One object is stored on multiple storage servers, which are assigned to different zones for redundancy. All objects are always monitored by the monitoring processes. If an object is removed because of file or disk trouble, then this lost object is recovered automatically as soon as possible by copying other redundant objects.

To integrate a petabyte-class distributed storage system using Swift, we must consider the system architecture, network design, parameter setting, and system configurations, which must suit user requirements (e.g., the size of files to be stored, network bandwidth to be used, hardware processing capability, failure rates, permissible recovery time, and hardware costs). For this purpose, NTT DATA has standardized design and setting knowhow as the Swift Reference Architecture in order to offer fast and stable large-scale distributed object storage at a low cost.

3. NII projects

Through the dodai project conducted together with the National Institute of Informatics (NII), NTT



Each storage server comprises an account server that manages metadata, a container server, and an object server that manages files.

Fig. 3. Basic configuration of Swift.

DATA has built a mechanism for implementing a cloud API on a leased physical machine for clients who particularly value high machine performance for cloud computing or ones who are unable to access a virtual environment for licensing reasons. Building upon this dodai project, we created a prototype of the Academic Community Cloud System for research purposes for NII in FY2011. We are now working to put this physical machine cloud system into practical use after appropriate operational assessment.

References

[1] OpenStack. http://openstack.org

[2] T. Grance and P. Mell, "The NIST Definition of Cloud Computing," NIST Special Publication, No. 800-145, Sept. 2011.

[3] M. Noguchi, "Basics of Cloud Building, Learning with OpenStack, Currently Popular," Nikkei Linux, No. 151, pp. 120–124, 2012 (in Japanese).

[4] News report (in Japanese).
http://techtarget.itmedia.co.jp/tt/news/1101/13/news06.html

Masayuki Hanadate
Manager, NTT DATA Corporation.
He received the B.E. degree in electrical and communication engineering from Tohoku University, Miyagi, in 1997. Since joining NTT Information and Communication Systems Laboratories in 1997, he has been engaged in R&D of information security systems, such as ones for NTT's e-ticket/e-money, smartcard applications, security protocols, and so on. Since moving to NTT DATA in 2010, he has developed some cloud security solutions, such as the virtual IPS/IDS and the distributed object storage and its security option products. He is a member of the Information Processing Society of Japan.