

Open Source Software and Community Activities Supporting Development of Cloud Services at NTT Communications

Mahito Ogura, Xiaojing Zhang, Masaki Matsushita, Takeaki Matsumoto, Ankit Purohit, and Kazuaki Harada

Abstract

At NTT Communications, the opportunities for using open source software (OSS) in system development and operations have been increasing. In the case of cloud services, we have been using OpenStack as the OSS of choice while proactively engaging with the OpenStack community by submitting contributions, joining related organizations, and making presentations at OpenStack events. We have also been holding OSS-related study groups within the NTT Group as well as technology-exchange events with organizations outside the NTT Group.

Keywords: OpenStack, OSS, technology-exchange events

1. Initiatives toward OpenStack

NTT Communications (NTT Com) provides cloud services through its Enterprise Cloud [1] solution and uses OpenStack open source software (OSS) as a platform supporting these services. At NTT Com, we survey, test, and use OpenStack and other OSS products and hold study groups, conferences, and other events to share OSS-related knowledge and know-how.

1.1 Overview of OpenStack

OpenStack consists of various functions called components for managing virtual machines, controlling a network, and performing other tasks. The OpenStack user can build a cloud service that meets objectives by combining necessary components in accordance with the service to be provided.

Numerous companies and organizations can be cited as OpenStack users including CERN, Walmart, and China Mobile, and case studies of using and

operating OpenStack throughout the world have been reported. Developers from all over the world participate in the development of OpenStack OSS by developing new functions and enhancing existing functions on an almost daily basis.

1.2 Use of OpenStack in Enterprise Cloud

At NTT Com, we began to provide Cloudⁿ [2], the first public cloud service in Japan using OpenStack, in October 2013. Then, in March 2016, we introduced OpenStack into Enterprise Cloud 2.0, a new cloud service targeting enterprise core systems. After the launch of this service, we went on to release new functions in a stepwise manner to meet user needs, and in May 2017, we released Deployment Manager [3], a function based on the OpenStack component Heat^{*1}, that makes it easy to build a system through lump creation and deletion of virtual servers, storage

^{*1} Heat: An OpenStack component for performing template-based orchestration.

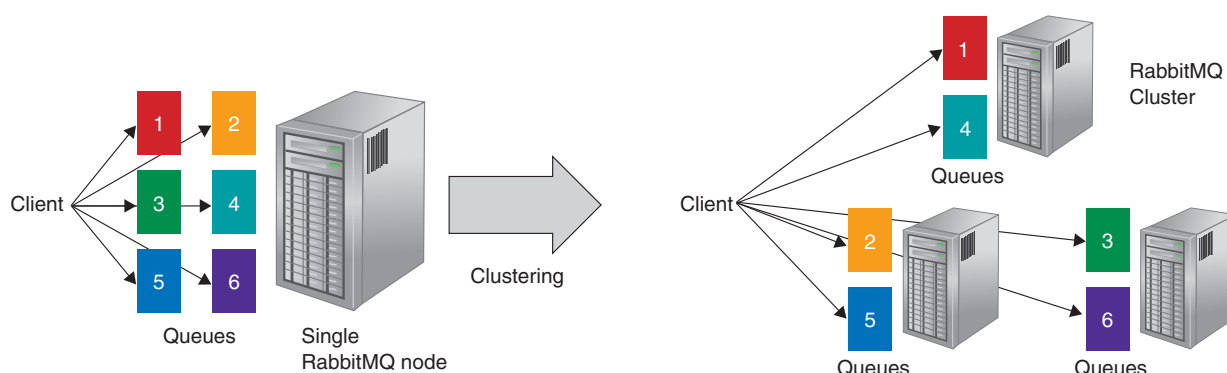


Fig. 1. Queue distribution in RabbitMQ cluster.

units, networks, and other resources. Furthermore, looking ahead to the spring of 2018, we plan to release a service based on the OpenStack component Trove^{*2} to facilitate the building of relational databases such as MySQL and PostgreSQL.

We point out here that we do not simply incorporate OpenStack components such as Heat and Trove in Enterprise Cloud 2.0 as a base in new services. We also actively engage with the OpenStack community such as by proposing functional enhancements for improved security and reporting and fixing bugs. Enterprise Cloud has been deployed in seven countries/regions (Japan, United States, United Kingdom, Germany, Singapore, Hong Kong, and Australia) as of September 2017. Each of these hubs connects to NTT Com's high quality and secure network infrastructure. NTT Com plans to continue its use of OSS in furthering the evolution of global and seamless cloud services as a provider of a carrier cloud.

2. Presentations at OpenStack Summit Boston 2017

OpenStack Summit [4] is held twice a year, coinciding with the release of new versions of OpenStack. For each summit, there is an open call for contributions, and the chairperson of each session track decides which of the collected contributions to accept after holding a community vote. About 20% of around 1000 submittals are generally accepted. We introduce here two presentations made by the NTT Com Technology Development Division at the OpenStack Summit held in Boston in May 2017:

- (1) Scale-out RabbitMQ Cluster Can Improve Performance While Keeping High Availability
- (2) Container as a Service on GPU Cloud: Our

Decision among K8s (Kubernetes), Mesos, Docker Swarm, and OpenStack Zun

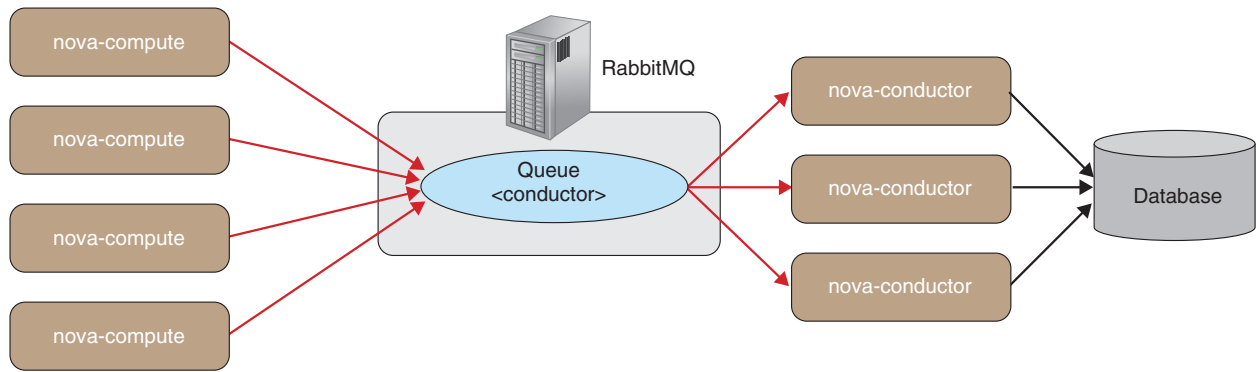
2.1 Scale-out RabbitMQ Cluster Can Improve Performance While Keeping High Availability

When OpenStack has been deployed on a large scale, a bottleneck has been found to occur in the message queue (MQ) at the time of asynchronous processing inside and outside the components. Methods known by OpenStack operators for solving this problem include MQ tuning and load distribution by dividing processing among multiple MQ clusters. However, it is not a simple task for operators to tune or operate multiple MQ clusters.

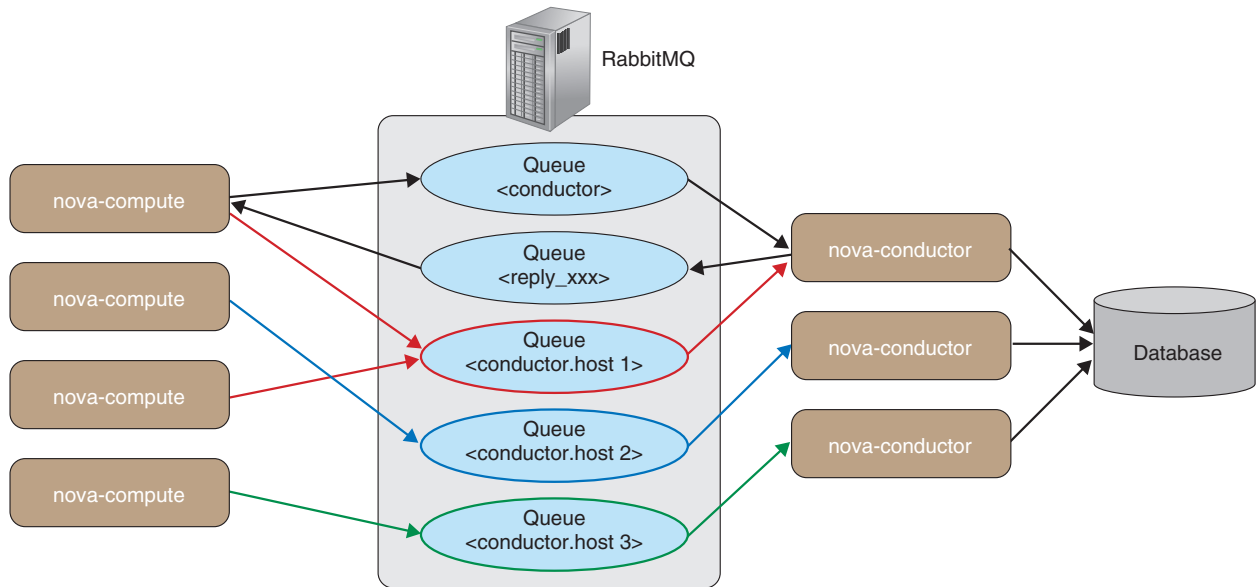
NTT Com, meanwhile, is planning to expand the scale of Enterprise Cloud 2.0, so it is therefore necessary to improve the performance and operation of MQ. In Boston, we teamed up with the NTT Software Innovation Center (NTT SIC) to present methods for improving MQ operation and OpenStack internal operation. To begin with, we presented the results of testing a method for improving performance by readjusting the settings of a RabbitMQ cluster, which is one type of MQ often used in OpenStack, and by scaling out a single RabbitMQ cluster while maintaining redundancy in internal data (Fig. 1).

Next, we found that the conductor function that adjusts processing for each nova component node managing a virtual machine could be a bottleneck, and we presented the results of testing a method in which the conductor function distributes the data flowing through a conductor among multiple conductor nodes (Fig. 2).

^{*2} Trove: An OpenStack component for providing DBaaS (database as a service) on OpenStack.



(a) Messaging in existing conductor



(b) Improved method

Fig. 2. Improved conductor.

2.2 Container as a Service on GPU Cloud: Our Decision among K8s, Mesos, Docker Swarm, and OpenStack Zun

Graphics processing unit (GPU) computing has been attracting attention in recent years as an efficient means of processing the workloads associated with artificial intelligence, big data analysis, and other large amounts of data. Cloud providers including Amazon, Microsoft, and Google have begun to provide GPU-equipped virtual machines. In this presentation, we introduced the results of testing and comparing a variety of OSS tools with the aim of finding the best method for building and operating a GPU

environment (Table 1).

It is important that the work of making configuration settings is simplified for cloud users who wish to concentrate their efforts on primary tasks such as machine learning, and that it is possible to fairly deploy GPU resources for cloud operators who wish to manage resources efficiently. Using container technologies such as Docker^{*3} and nvidia-docker^{*4} in

^{*3} Docker: A container engine provided by Docker Inc. Though originally OSS, it has since been migrated to the Moby project. Docker products are developed based on Moby deliverables.

^{*4} nvidia-docker: A container image and its utility group officially provided by NVIDIA for using NVIDIA GPUs from Docker.

Table 1. Results of comparison test.

	Lump management of GPU cluster	Assignment of multiple GPUs	GPU isolation	Docker support	Batch task execution
OpenStack Zun	Not supported	Not supported	Not supported	Supported	Not supported
Docker Swarm/ Swarm Mode	Not supported	Not supported	Not supported	Supported	Not supported
Apache Mesos	Supported	Supported	Supported	Not supported	Supported
Kubernetes	Supported	Supported	Supported	Supported	Supported

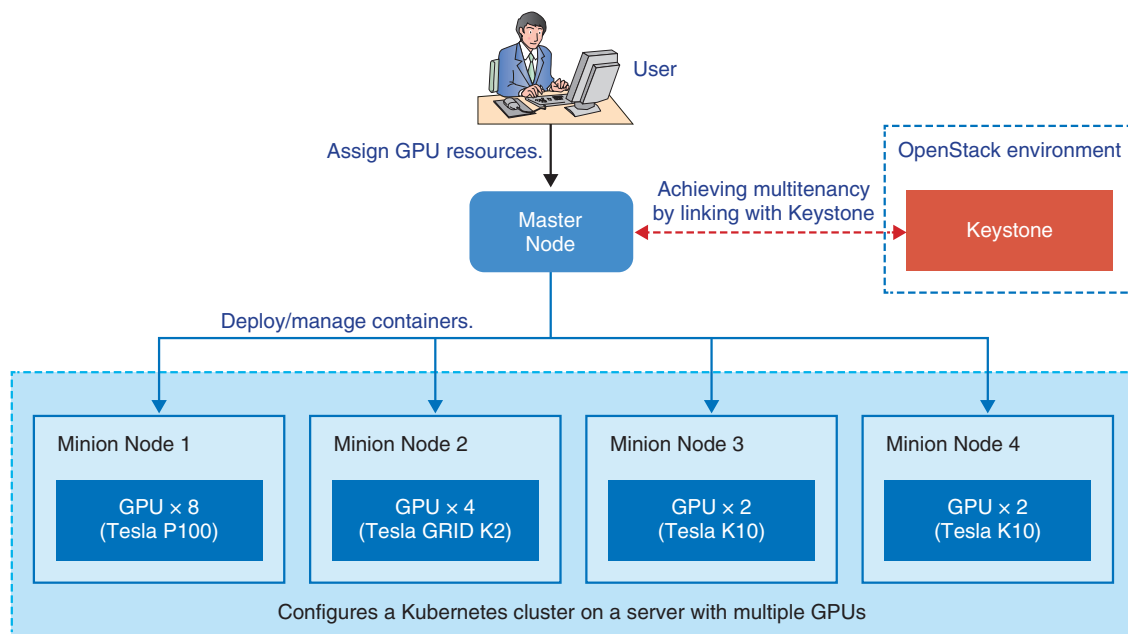


Fig. 3. Conceptual diagram of test environment.

place of virtual machines makes it possible to deploy applications rapidly and to minimize configuration settings without having to worry about the interdependencies and combinations of guest OSs (operating systems), libraries, and device drivers. However, a scheduling function for deciding which GPU container to deploy on which server has not yet matured, and a clear-cut method for providing GPU resources fairly has yet to be found.

Consequently, in the process of building a GPU container platform for in-house use, we surveyed, tested, and compared some common container management tools such as Kubernetes, Apache Mesos, Docker Swarm, and OpenStack Zun. As a result, we built a test environment (**Fig. 3**) using Kubernetes as the best container management tool based on criteria such as assignment of multiple GPUs, resource isolation

from GPUs in other containers, and Docker support. In the presentation, we introduced the know-how that we gained from building and operating this test environment together with actual use cases.

3. Study groups and events inside/outside NTT Group

Many groups are involved in sharing information on container management and other related topics, and NTT is also contributing. We report here on these groups and the issues they are working on.

3.1 Container and cloud-native study groups

In cooperation with NTT SIC and together with NTT Group engineers, we have been holding container study groups since October 2015 as a forum for



Photo 1. NTT Tech Conference scene.

studying container techniques and exchanging opinions. These study groups have been sharing knowledge and operational know-how on Docker, Kubernetes, Apache Mesos, and other tools released as OSS, as well as reports from participants of conferences such as DockerCon and KubeCon. However, a somewhat broader range of technologies and knowledge going beyond containers will be needed in the future, so the name of this group was changed to the cloud-native^{*5} study group as of the sixth meeting. The stated purpose of this study group is as follows: “The cloud-native study group will share information and hold problem consultations with NTT Group engineers involved in cloud computing and will accumulate cloud-related know-how. It will also discuss design techniques and study items for cloud-native systems based on use cases to accumulate know-how on designing such systems.” In this way, we will continue to share knowledge and hold discussions with engineers within the NTT Group.

3.2 NTT Tech Conference

NTT Com makes a strong effort to develop software personnel with the aim of enhancing internal skills. As a part of this effort, we have established the NTT Tech Conference [5] to enable NTT Group engineers to share what they have been learning in their software activities on a voluntary basis.

At an NTT Tech Conference, NTT Group engineers present their own technology-related knowledge and activities with the aim of exchanging opinions with

other engineers from inside and outside the NTT Group. The second meeting held on August 10, 2017, brought together 221 participants from within and outside the NTT Group. In this meeting, under the title of “Invitation to Participate in OSS Development Communities—Examples of NTT Group OSS Activities,” OSS developers in the NTT Group held a session for discussing and exchanging opinions with participants on the development progress and development method of various OSS products. This session provided a forum for sharing information on participating in the development of various OSS communities and for exchanging opinions on development methods and community conditions (Photo 1).

NTT Com plans to hold more meetings of the NTT Tech Conference in the future as a forum for exchanging information on OSS and other software technologies.

4. Future development

At NTT Com, we will continue to develop services using OpenStack and other OSS products with the aim of responding rapidly to user needs. We seek to contribute to the growth of the OSS community not only by using OSS but also by proposing and improving functions. At the same time, we will hold study

^{*5} Cloud-native: Systems and services designed assuming use on the cloud.

groups and events to facilitate technology exchanges with engineers both inside and outside the NTT Group with the aim of strengthening the internal skills and technological competence of the entire NTT Group.

References

- [1] Enterprise Cloud, <http://www.ntt.com/en/services/cloud/enterprise-cloud.html>

- [2] Public Cloud, <http://www.ntt.com/en/services/cloud/cloudn.html>
 [3] Deployment Manager, <https://ecl.ntt.com/en/documents/tutorials/ecdc/rsts/dpm/index.html>
 [4] Website of OpenStack Summit Boston 2017, <https://www.openstack.org/summit/boston-2017/>
 [5] Website of NTT Tech Conference (in Japanese), <https://ntt-developers.github.io/ntt-tech-conference/>

Trademark notes

All brand names, product names, and company names that appear in this article are trademarks or registered trademarks of their respective owners.



Mahito Ogura

Software Engineer, NTT Communications Corporation.

He received a Bachelor of Information Science and Technology and a Master of Information Science and Technology from Osaka Institute of Technology University in 2007 and 2009. He joined NTT Comware in 2009 and moved to NTT Communications in 2014. His main interest is cloud and software engineering, especially software for cloud computing.



Takeaki Matsumoto

Researcher, NTT Communications Corporation.

He received a B.E. and M.E. in computer science from Chuo University, Tokyo, in 2013 and 2015. He joined NTT Communications in 2015. His main interest is cloud infrastructures.



Xiaojing Zhang

Researcher, NTT Communications Corporation.

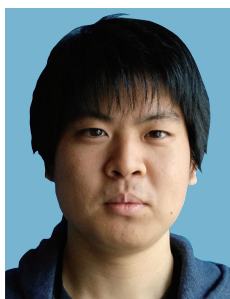
She received a B.E. and M.E. in computer science from Kyushu University, Fukuoka, in 2005 and 2007. She joined NTT in 2007. Her main interest is software engineering, especially model based development. She is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).



Ankit Purohit

Researcher, NTT Communications Corporation.

He received a B.E. in computer technology from Nagpur University in 2012 and an M.E. in mechatronics from Pune University in 2015. He joined NTT Communications in 2016 and is currently working in the OpenStack team. His main interests are cloud computing and high performance computing.



Masaki Matsushita

Software Engineer, NTT Communications Corporation.

He received a B.A. in economics from Keio University, Kanagawa, in 2014 and an M.S. in information science from Japan Advanced Institute of Science and Technology (JAIST) in 2016. He has been an OpenStack contributor since the Kilo release. He contributes to the OpenStack libraries known as oslo.log, oslo.config, and Trove. He proposed a fluent-formatter blueprint to output logs to Fluentd as a machine-readable format. It was released in the Newton release. He also contributes to the Ruby programming language and works on improving its performance.



Kazuaki Harada

Software Engineer, NTT Communications Corporation.

He received a Bachelor of Information Science and Technology and a Master of Information Science and Technology from Osaka Institute of Technology in 2015 and 2017. He joined NTT Communications in 2017. His main interests are field-programmable gate arrays and heterogeneous computing.