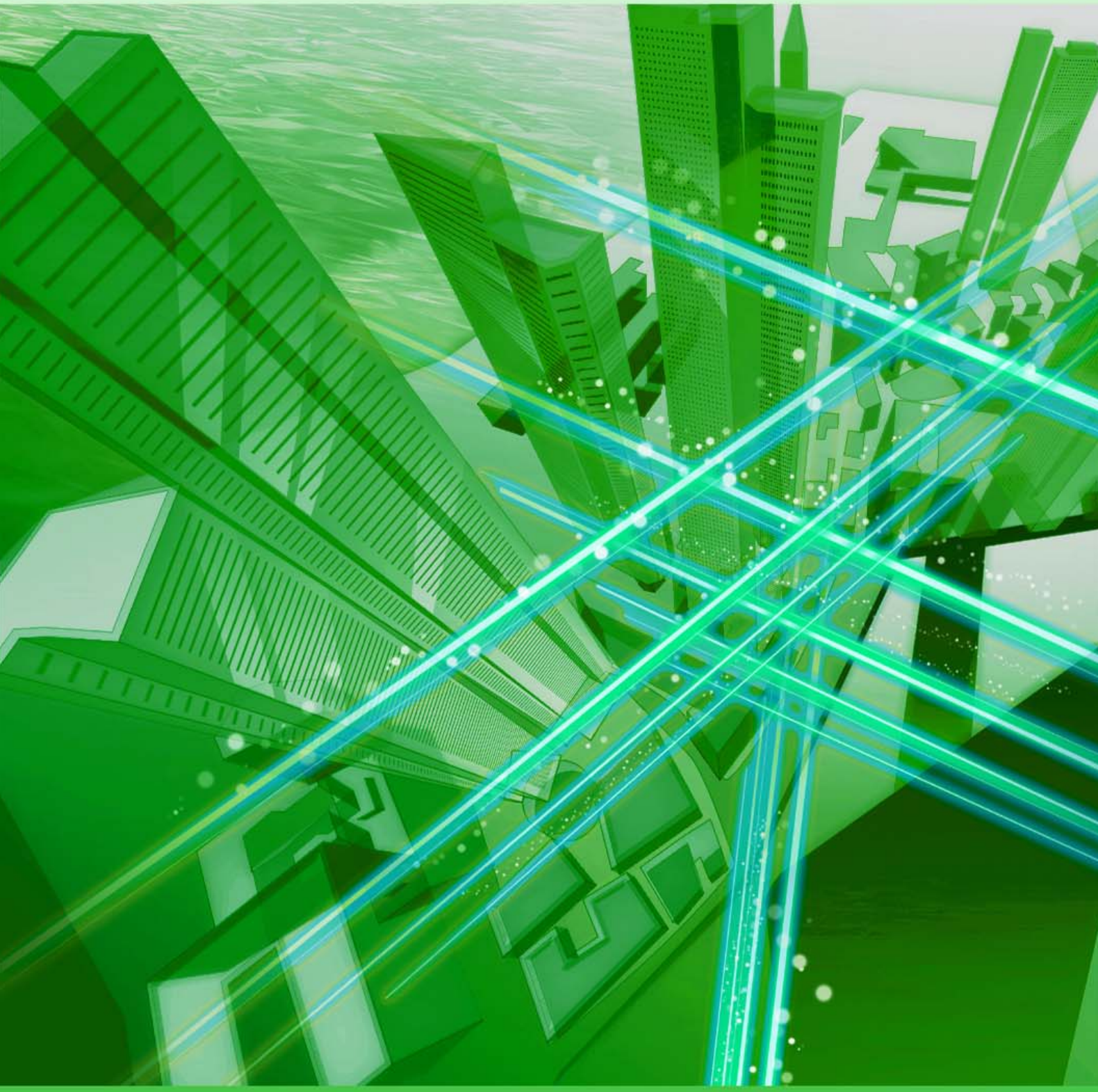
NTT Technical Review

2
2018



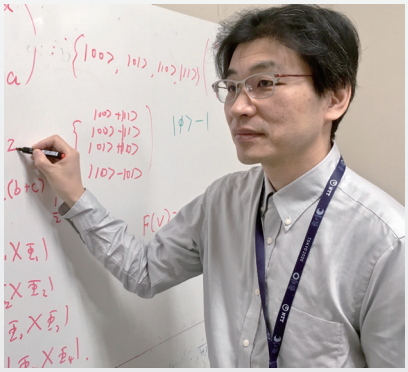
February 2018 Vol. 16 No. 2

NTT Technical Review

February 2018 Vol. 16 No. 2

A Period without Direction Is a Time for Improving One's Basic Research Abilities—Better to Change Your Perspective than Follow the Crowd

Seiichiro Tani
Senior Distinguished Researcher, NTT Communication Science Laboratories



Overview

Research toward the development of a practical quantum computer began more than 30 years ago in the 1980s, and it is now entering a new phase. The Ministry of Education, Culture, Sports, Science and Technology in Japan recognizes the significance of this development and has included 3.2 billion yen in its budget request as a development fund for photonics-quantum technology. Senior Distinguished Researcher Seiichiro Tani of NTT Communication Science Laboratories has had many world-first achievements in this field. The quantum computer is attracting attention as a driver of future trends, so we asked him to tell us about the current state of quantum computer research and some research achievements. We also asked him his views on how a researcher should approach the work of research.

Keywords: quantum computer, quantum algorithm, leader election problem

Achieving innovative research results by taking pioneering approach and attracting worldwide attention in the leader election problem

—Dr. Tani, please tell us about your current area of research.

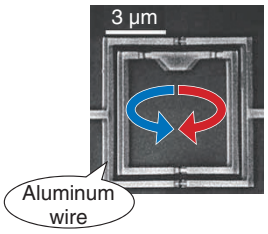
I'm involved in the research of quantum computers (**Fig. 1**). Similar to conventional computers (called "classical computers" in the world of quantum computers), quantum computers consist of hardware and will not run without placing something equivalent to software on that hardware. In my work, I theoretically consider how to create that something corresponding to software (quantum algorithms), that is, how to bring out the performance of that hardware. I started out on this research path about 15 years ago.

Let me point out here that a key premise in this research is that using a quantum computer to do computations will not necessarily be faster. There are many cases in which the speed of processing by a quantum computer will be about the same as that of a classical computer, depending on the type of computation. In fact, a lack of skill in creating a good quantum algorithm to run on quantum computers could result in even slower processing than classical computers.

With this issue in mind, we have established several

	Classical computer	Quantum computer
Unit of information	Bit	Quantum bit (qubit)
Mathematical expression	Boolean value (true/false)	Complex vectors
Elementary operations	Boolean operations (AND, OR, NOT)	Linear operators (unitary operators)
Physical representation	Semiconductors	Semiconductors, linear photonic devices, superconductors, etc.

Superconducting qubit [1]



Current direction	Information
Clockwise	0
Counterclockwise	1

Currents flowing in opposite directions coexist through a quantum effect (superposition).

Fig. 1. Concept of a quantum computer.

key research themes and objectives. To begin with, we must search for problems conducive to high-speed processing on quantum computers. We must then estimate the extent to which computation can be speeded up in such cases, find quantum computer functions that are truly needed for achieving such high-speed processing, and determine how such functions should be merged with computations by classical computers.

The concept of the quantum computer was first proposed in the 1980s, but it took until 1990 or so for research to really take off. During this 30-year period, the research of quantum computers evolved significantly in terms of both hardware and algorithms. Our group as well contributed to this history through various research achievements. One of these that received considerable recognition was distributed quantum algorithms for solving the leader election problem (**Fig. 2**).

The research on quantum algorithms can be broadly divided into two flows. These originate respectively in Peter W. Shor's quantum algorithm for factoring integers exponentially faster than any existing algorithm and Lov K. Grover's quantum algorithm for searching for a desired item of data from a huge amount of data (**Fig. 3**). Research on the application and generalization of each of these algorithms has spanned a period of about 20 years. We developed our distributed quantum algorithm for solving the leader election problem by taking an approach different from these two flows and received praise for it as a pioneering algorithm as a result.

In the leader election problem, a number of computers connected in a network must autonomously elect one of those computers as a single leader. This problem can be used to determine which computer among many is to perform a particular process. For cases in which the uniqueness of computer addresses is not guaranteed due to faults or other reasons, it is known that this problem cannot be solved in a finite amount of time in a network consisting of classical computers. In contrast, we devised an algorithm for efficiently solving the leader election problem in a network consisting of quantum computers and received worldwide recognition for it.

In this algorithm, multiple computers play rock-paper-scissors using the observed values of qubits (0 or 1), and the computer that survives multiple elimination rounds is determined to be the leader. Here, in the computation process, we have developed a technique to make the probability of a draw in rock-paper-scissors zero, with the result that a single leader can be elected in a finite number of rounds (in a finite amount of computation time) (**Fig. 4**).

This result was praised for theoretically showing that quantum computers are substantially different

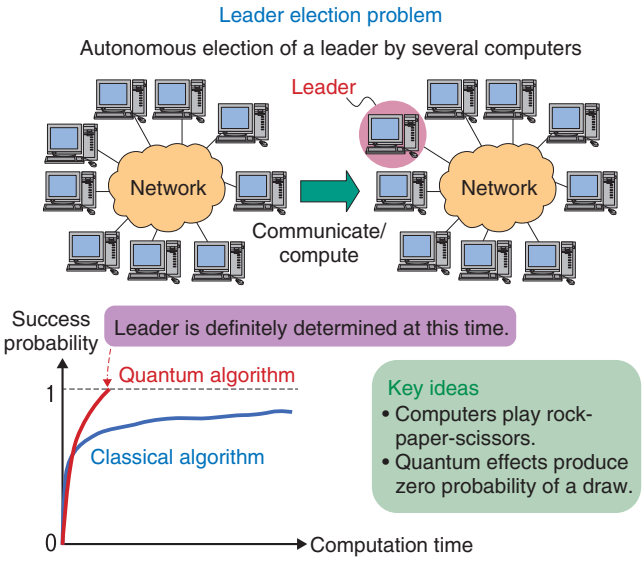


Fig. 2. Overview of distributed quantum algorithm for solving the leader election problem.

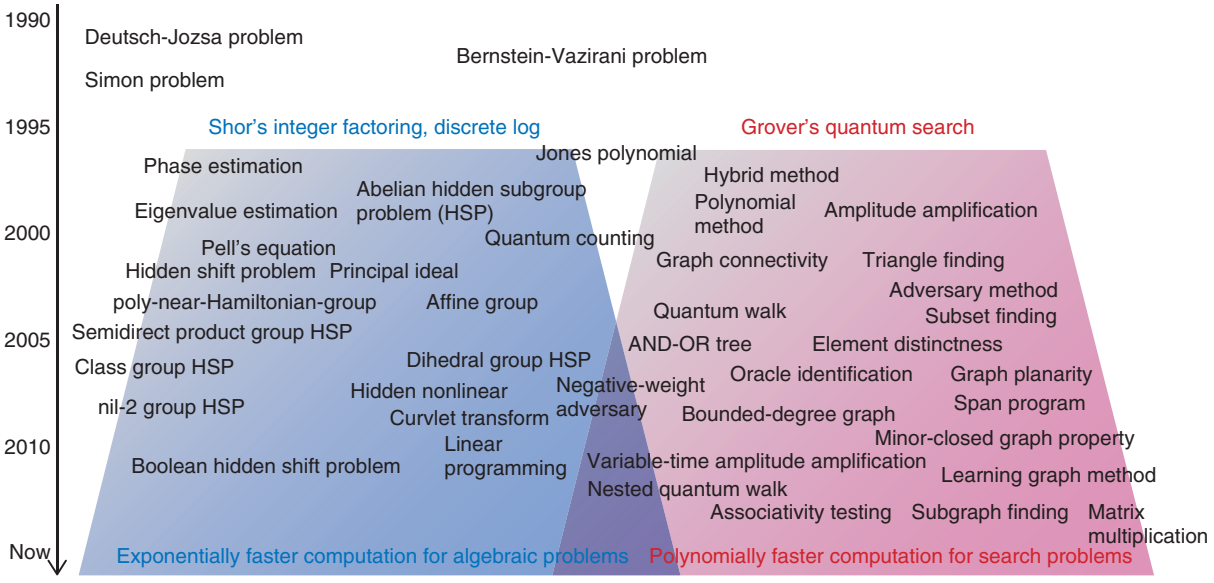


Fig. 3. Flow of quantum algorithm research.

from classical computers in terms of ability, but it also has merit from the viewpoint of practical application since it can be used in the automatic setup of a quantum network consisting of quantum computers. Here, automatic setup means that multiple quantum computers need only be physically interconnected to automatically assign node addresses, automatically configure routing paths, and automatically and randomly assign management authority. This result also means that computations on input data distributed throughout the network that were not previously possible with classical computers can now be done.

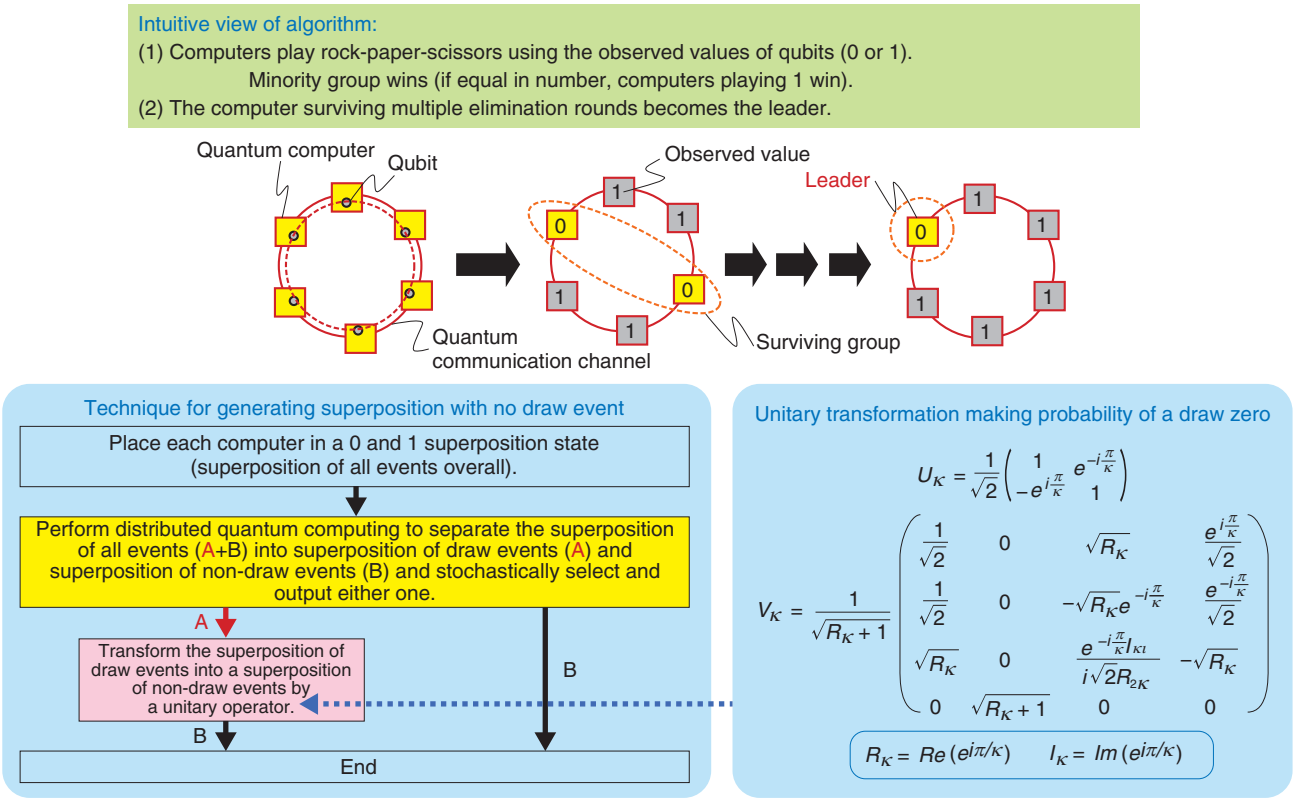


Fig. 4. Idea behind solving the leader election problem.

—You have obtained innovative results during the long history of quantum computer research.

All theoretical researchers aim to shed light on the unknown and determine the direction of future research. I am one of those researchers. With regard to the leader election problem, I feel that I may have done my best work in that research. Thanks to those research results, I was able to give an oral presentation at a top-level international conference on quantum information science soon after the start of serious quantum computer research in Japan and to have papers published in ACM Transactions, a journal of the Association for Computing Machinery (ACM) [2]. In this sense, I believe I accomplished my initial objectives. I was also very pleased to see my research results included in ACM's SIGACT News and in a textbook (authored by Rodney Van Meter).

Incidentally, in the research field of quantum algorithms, there are not many researchers from the preceding generation. Since quantum algorithm research commenced in the 1990s when researchers of our generation came of age, I think we can safely say that researchers of our generation and beyond have been virtual pioneers.

On stepping onto a train station platform, a solution suddenly struck me like lightning; you can't obtain results that make an impact without pushing yourself to the max

—How did you derive the algorithm for solving the leader election problem?

When I moved to NTT Communication Science Laboratories in 2003 after working on the research and development (R&D) of network protocols at the NTT Yokosuka R&D Center, I was already of the so-called mid-level generation. Consequently, as I was not an entry-level researcher, I could not imagine myself producing zero research results in my first year or two in my new position. I knew I had to produce some results, and in doing so, I had to find something that no one else in the world was working on and that I could apply my background to. Therefore, I began to look for a theme that I could devote

all my energy to. However, there were already a lot of researchers in the field of quantum computer research, and I didn't want to just follow the crowd as a newcomer. I therefore thought about approaching this research from a completely different perspective, so I started to voraciously read papers in this field.

Moreover, thinking that I wanted to produce major, unprecedented results that would make an impact, I also reviewed textbooks covering the subjects of classical computing as well as quantum computing. As a result, after looking at major problems taken up in textbooks as opposed to minor problems appearing in obscure corners of journal papers, I came across the leader election problem. This is an essential problem that is invariably discussed in textbooks on distributed computing, but at that time, it was a network-related issue that was not being researched much in the field of quantum computers.

My research started to bear fruit in only the fourth month after deciding to take on the leader election problem. I felt very fortunate in this regard. I didn't have any experience in the field of quantum computers, so it was like filling in the pieces of a puzzle time and time again. Moreover, there was no guarantee that that puzzle would be completed. However, I believed that I could complete the puzzle, so I would perform test calculations intensely for days on end while looking at stacks of notes on my desk. This was an interesting world in which I was doing calculations without the use of a computer to develop a theory on calculations to be performed by a computer!

In the fourth month of doing such calculations and trying to solve this puzzle without knowing when I might complete it, I finally completed the proof of a core lemma. At that moment, I could feel my heart pounding, and I only half believed what I had done. Given my desire to enter the field of quantum computer research and working hard with my whole body and soul to produce some kind of result, I was very happy about this, but on the other hand, it was all too sudden and unbelievable. On that day, I was commuting to work by car, but I cannot remember how in the world I got home.

Several months passed as I continued with my research. I had been thinking about another lemma around that time, and one day on returning from a business trip, I walked down a staircase at a train station and stepped onto a platform. At that very moment, the key to proving that lemma suddenly hit me, as if I had been struck by lightning. How can I explain this? Well, when I'm riding (getting on) a train, my objective of "moving" from one place to another is being fulfilled, so I feel a sense of ease in that I'm not wasting time, which enables me to relax. In fact, whenever I'm moving or about to move by a train or other means, things often pop into my head, and stepping onto the platform at that time was exactly such a situation.

—This theme that you found under pressure led to world-first results! This seems to say that some amount of pressure is good.

Yes, a reasonable amount of pressure can be beneficial! When starting out in their careers, researchers gain experience by solving small problems, but in the future, I think that researchers will aim for major achievements. Lots of pressure can arise in tackling a risky problem, but I think it's something that a researcher should face even if it takes time.

Although this may sound somewhat negative, theoretical research is strenuous. Rather than progressing little by little over time, research tends to suddenly leap forward at certain points in time. When that step will be climbed is unknown, and whether it will be climbed at all is also unknown. Until that opportunity arrives, a researcher must persevere in a state in which nothing is moving. This is mentally difficult beyond what can be conveyed in words. Of course, I know from experience that other types of difficulties exist in other fields. That is to say, if there is a field in which a researcher can demonstrate his or her abilities, it should be easy to convert that pressure to ability. In my case, the joy of research supersedes any difficulties, so I was able to withstand that pressure.

Once one has produced several research achievements, I feel that one can be optimistic and say without substantiation: "Such a time will come again." When you are young, however, it's easy to be pessimistic in the manner of "This is really turning out bad!" In the end, you can only believe in yourself. It sometimes happens that you are getting nowhere after thinking for half a year or even a whole year, so it's crucial that you set your own deadline without obsessing over results.

—By the way, why did you choose research as a career path?

There was a time before graduating from university when I had three months to submit my graduation thesis but I still had nothing to show in terms of results. I therefore worked frantically to squeeze out something, but on presenting my theory in our

research laboratory, a senior of mine pointed out a serious error, so I then had to work like crazy for the remaining month and a half. This was quite an experience in getting out of a difficult situation. At that time, I didn't know that the "time for climbing the stairs" comes suddenly without knowing when, so that uncertainty made for a very difficult time.

On the other hand, this experience only increased my enjoyment of theoretical research, and it occurred to me that the life of a researcher might not be bad at all. Later, on completing my master's degree, I arrived at a crossroads where I had to decide between looking for work and continuing my education, and given my frame of mind at that time, I thought that finding work in research would be quite cool. Many of my seniors at that time were involved in research activities while working at companies, so no one thought that employment meant being separated from research. I chose NTT because I had no doubt that I would be given the opportunity to pursue basic research. In actuality, though, a major organizational restructuring took place after I entered the company, causing a major transition in the research field that I was to be involved in. However, looking back at that time in an objective manner, I see that I was initially involved in theoretical research at university, that I then gained experience in application fields including the research of integrated circuits and network protocols after entering NTT, and that I was finally given the opportunity of returning to theoretical research. In hindsight, I think that all of these experiences have helped in some way to produce my current research results.

Find pleasure in your daily research and pursue research that will still be around 100 years from now

—What advice would you give to young researchers?

Although "usefulness" should be considered when finding a research theme even in basic research, I think it is difficult to lead a research life motivated solely on what might be useful. Shouldn't it also be important to take joy in the act of research itself? Basic research involves basic academic disciplines, and in our field, mathematics and physics in particular. Though we are not doing pure mathematics here, we do use mathematics as a technique. In fact, there are not a few researchers who enjoy the application of mathematics itself. Such pleasure is an extra benefit of research. I believe that feeling such joy in daily research can be a source of happiness for a researcher.

In addition, while going through a period without a clear-cut direction can be painful for a researcher, it's exactly a time for devoting energy to improving your basic research abilities. It's important to soundly accumulate those basic elements that will become your strengths in the future. As I mentioned earlier about our field being based on mathematics (in a broad sense), and with the difficulties that I experienced, I would recommend that researchers be thoroughly familiar with undergraduate mathematics at a minimum. I came to understand today that raising the level of my basic abilities is of utmost importance after going through an ongoing learning process throughout my research life. By constantly improving your basic abilities, I believe that something that was quite tough five years earlier will become something that you can do in the present!

It's also important to occasionally get away from research. I myself enjoy sports (soccer in particular) and driving. Pursuing research at the cost of everything else can lead you to a dead end and leave you mentally exhausted. It's rewarding to be involved in something completely different!

Also, if you have the opportunity to travel abroad, take it by all means. At the least, I would recommend that you stay for several months by yourself, and I would recommend in particular that you go to a place where postdoctoral researchers gather. I think that seeing the sparkle in their eyes will help to change your attitude toward research. I myself had the opportunity to study at the University of Waterloo in Ontario, Canada for one year starting in 2010. It was a place where top-class researchers in the field of quantum computers would gather, so being able to feel that atmosphere turned out to be a very valuable experience for me. It's advantageous to be in an environment where you can do research in a relaxed way, but it's also good to get out to international conferences and receive some stimuli from time to time.

—Dr. Tani, what is your outlook for the future?

The research on quantum computer hardware is advancing if only a little at a time. In these last two or three years, large amounts of research funds have been allocated to quantum computer research throughout the world, so I feel that progress in this area is speeding up. That being said, it will still take some time before an ideal quantum computer can be achieved. Till then, we can expect the creation of quantum computers with limited functions, but even

at that level, the solving of important problems will be most welcome. With this level of quantum computers, which we can expect to be implemented sometime in the near future, the question as to what types of problems in particular should be targeted is becoming increasingly significant. Actually, I am now involved in searching for such problems.

In addition, in consideration of the massive amount of technology already existing for classical computers, it would certainly be better if problems that could be solved on classical computers were actually solved on classical computers. This, however, leads to another question: Where in particular should we depend on quantum computers? We are still lacking in the ability to make such decisions. This question is also of interest to me and I am working on finding an answer.

Work that is current with the times is fine, but as a researcher, I want to pursue work that will still be around in the future. In this regard, I would like to take a sweeping view of both quantum and classical computations. Considering work that will still have value not just 5 or 10 years from now but even 100 years down the road, I would like to pursue long-range themes in parallel with those that can be tackled in the short term.

References

[1] H. Yamaguchi, "Research Frontier of Quantum Computers," NTT Technical Review, Vol. 10, No. 9, 2012.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201209fa1.html

[2] S. Tani, H. Kobayashi, and K. Matsumoto, "Exact Quantum Algorithms for the Leader Election Problem," ACM Transactions on Computation Theory, Vol. 4, No. 1, Article 1, 2012.

■ **Interviewee profile**
Seiichiro Tani

Distinguished Scientist (Senior Distinguished Researcher), Media Information Laboratory, NTT Communication Science Laboratories.

He received a B.E. in information science from Kyoto University in 1993, and an M.S. and Ph.D. in computer science from the University of Tokyo in 1995 and 2006. He joined NTT in 1995 and studied algorithms for large-scale integrated circuit testing, multi-point communication, and distributed computing problems. He is currently working on algorithmic theory and complexity of quantum computing. From 2004 to 2009, he was also a researcher on the ERATO Quantum Computation and Information Project of Japan Science and Technology Agency (JST). From 2010 to 2011, he was a visiting researcher at the Institute for Quantum Computing, University of Waterloo, Ontario, Canada. He received the 2000 IEICE ISS Best Paper Award from the Institute of Electronics, Information and Communication Engineers (IEICE).

NTT's Increased Focus on Open Source Software

Masahisa Kawashima

Abstract

The emergence of new information technology (IT) applications such as artificial intelligence, the Internet of Things, and software-defined infrastructures is stimulating more use of open source software (OSS), and consequently, the skills needed for using OSS are broadening. While legacy IT applications would require skills in using only a few standard OSS packages, these new application areas require the ability to select software modules from many choices and integrate them into a system tailored to a company's or customer's requirements. Such integration activities often require collaboration with OSS communities. Additionally, because these activities incur costs, it is also important to develop business strategies to leverage OSS and gain a rewarding return.

Keywords: open innovation, open source software, software innovation

1. Introduction

In April 2006, the NTT laboratories founded the NTT Open Source Software Center (NTT OSS Center) in cooperation with NTT DATA, NTT Comware, NTT Software (now NTT TechnoCross), and NTT Advanced Technology, and in the decade since then, the adoption of open source software (OSS) in the NTT Group has steadily progressed. While the importance of OSS has remained the same or become greater in the information technology (IT) industry, its application domain has expanded substantially over the years. This article outlines the expansion of OSS application fields and new issues in using OSS, and introduces the Feature Articles in this issue.

2. Industrial trends

Several trends have become evident as the use of OSS has increased. These trends are briefly described in this section.

2.1 Diversification of IT systems

Traditionally, IT systems have been used for the efficient management of business information such as received orders, receipts/disbursements, and inventory that inherently exists in a company's business operations. This kind of legacy IT system is called a system of record (SoR). More recently, however, as reflected by smartphone applications, companies have started developing IT systems to maintain relations with consumers or keep track of shipped products. In contrast to SoR, this type of IT system is called a system of engagement (SoE). Companies are also showing a strong appetite for artificial intelligence (AI) and Internet of Things (IoT) applications. As described above, the role of IT systems is becoming increasingly diverse.

2.2 Virtualization of IT/NW infrastructure

A conventional IT and network (IT/NW) infrastructure[*1] requires time-consuming and error-prone manual operations for its configuration, which leads to the requirement of a long lead time, for example, a few weeks, even for small applications that can be developed in a week.

Today, however, virtualization and software-defined technologies have made it possible to use

*1 IT/NW infrastructure: In this article, IT/NW infrastructure is defined as an infrastructure that provides IT resources such as computers, storage, and networks as needed by IT applications.

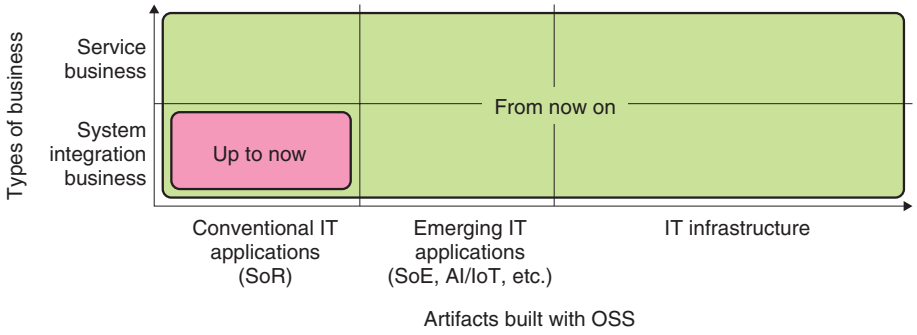


Fig. 1. Expansion of OSS application field.

software to take over many of the configuration tasks. This approach has been aggressively pursued to shorten the application release cycle. Moreover, as part of this trend, many customers have started using infrastructure-as-a-service (IaaS) services on the Internet instead of owning and managing their own infrastructures.

2.3 OTT-driven evolution of IT/NW infrastructure

Traditionally, the evolution of the IT/NW infrastructure used to rely on dominant global product vendors. The reason for this is that there have been no system integrators (SIers) or network providers that have a market share comparable to that of global product vendors.

This state of affairs, however, has been overturned by the emergence of over the top (OTT)[*2] operators that are expanding their business operations on a huge scale surpassing that of global product vendors. To release services quickly and make operations more efficient, OTT operators are proactively deploying software-based operation of the IT/NW infrastructure using virtualization and software-defined technology, and by doing so, are becoming the dominant force in infrastructure evolution.

3. Expansion of OSS application fields

The above industrial trends have created the following OSS application areas, while the use of OSS for SoR-type IT systems is still actively pursued (**Fig. 1**).

3.1 Application of OSS to SoE and AI/IoT systems

With the extremely rapid evolution of technologies in SoE and AI/IoT systems, it is essential to adopt the latest software and data-analysis technologies in developing these systems. Innovators in these technology fields are addressing such demands by proactively releasing software as OSS to gain users at an early stage, and as a result, system development using OSS has become mainstream.

3.2 Application of OSS to IT/NW infrastructure

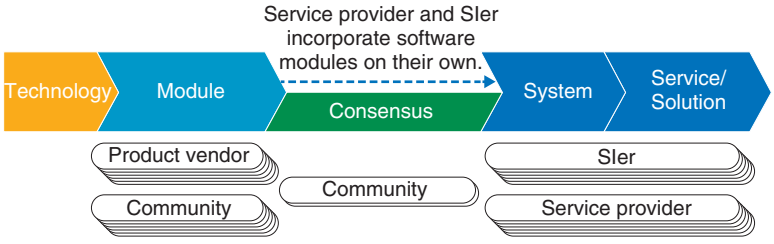
Some OTT operators are actively forming open communities and sharing their software products as OSS, which would increase the number of organizations or people that support their technical ideas or architectures. Such OSS communities make opportunities for SIers and service providers[*3] to pursue the evolution of their IT/NW infrastructures without having to rely on dominant global product vendors. This trend suggests that OSS will become far more important for SIers and service providers to remain competitive in infrastructure operation.

4. Change in IT value chain and issues in using OSS

In line with the above trend, the role of service providers and SIers is changing. A service provider or SIer constructs and operates a system to provide customers with solutions and services along the IT value chain shown in **Fig. 2**. In the past, a service provider or SIer would build and operate such a system while combining a few platform products[*4] (Fig. 2(a)).

*2 OTT: Refers to operators such as Amazon, Google, Facebook, and Microsoft that provide dominant services on the Internet.
*3 Service providers: In this article, refers to operators that provide network services, cloud services, web services, etc.
*4 Platform product: In this article, refers to a software product that provides many functions in an integrated manner so that they can be used in diverse applications. Examples of such platform products are Apache Web Servers, Oracle DB, JBOSS, and VMware.

(a) Platform-centric model



(b) Platform-less model

Fig. 2. Change in IT value chain.

However, because platform products are designed to satisfy a variety of requirements, their version update cycle is relatively long. Consequently, in the areas of AI/IoT and software-defined infrastructure, relying on platform products often becomes a hindrance to the timely delivery of new technologies.

This state of affairs has made it necessary for service providers or SIers to build a system by integrating small software pieces, that is, modules, by themselves instead of relying on platform products. In open communities, industry consensus-forming projects[*5] that facilitate such system building are gaining momentum (Fig. 2(b)).

This change in the IT value chain is shown in Fig. 2. The conventional IT value chain and the IT value chain in new areas can be expressed as a *platform-centric model* and *platform-less model*, respectively.

To promote the use of OSS amid this change in the IT value chain, the following issues must be addressed.

4.1 Developing product integration capabilities

As described above, companies can no longer rely on platform products supplied from the upstream of the IT value chain. Instead, they must integrate small software modules into systems tailored to their requirements. This requires product development skills such as choosing appropriate modules from among many choices and designing an easy-to-modify architecture, in addition to traditional system integration skills. Since it is difficult for a company to maintain a large army of skilled software engineers, companies should also be adept at narrowing down requirements and deriving a *just-in-time* product roadmap with respect to the company's service model and customer needs. Mastering agile software development practices is important in supporting a just-in-time product roadmap.

4.2 Building technical support capabilities to address the breadth of OSS

In the traditional platform-centric IT value chain, OSS that should be supported can be narrowed down to a small number of platform products. However, in the platform-less IT value chain, service providers or SIers rely on a variety of OSS products. Building a technical support capability to address the variety of OSS products is a new challenge.

4.3 Gaining trust in OSS communities

Typically, building a system with new OSS requires that more than a few bug patches be applied to the OSS. In addition, service providers or SIers sometimes come across the situation where functional extensions must be made to the OSS. However, getting such bug patches or functional extensions

*5 Industry consensus-forming projects: Examples include OPNFV (https://www.opnfv.org/) and OpenFog Consortium (https://www.openfogconsortium.org/).

accepted by the maintainer[*6] of that OSS is not straightforward. Companies must first establish a presence that facilitates the acceptance of proposals. For this purpose, it is necessary to participate regularly in technology summits and contribute actively to OSS development and maintenance activities. Making such efforts on an ongoing basis also helps a company to become knowledgeable with the design philosophy of that OSS and to make proposals that are more likely to be accepted.

4.4 Creating a company's unique value with OSS

The three types of activities above incur a significant cost, so companies would not be able to sustain those activities without having business strategies to leverage OSS and gain a rewarding return. One promising approach is to use OSS to develop unique products/services that leverage the company's inherent strengths. Not only do such efforts help a company gain the return, they will also strengthen the company's influence in OSS communities.

5. NTT Group activities

The NTT laboratories and NTT operating companies are actively collaborating to address the above issues. The Feature Articles in this issue introduce some of these efforts. To begin with, the article entitled "Open Source Software Efforts to Transform IoT/AI Services" [1] introduces AI and container-virtualization technologies, while the article "Open Source Software behind NTT Network Services" [2] introduces advances in the network, especially those related to the IT/NW infrastructure.

The article entitled "NTT's Contributions to OSS Upstream First Development" [3] introduces efforts at NTT laboratories toward establishing competence in community collaboration, while the article "Open Source Software Initiatives Supporting NTT Group Software Development and Operations" [4] introduces the NTT OSS Center and the spread of the Macchinetta software development framework within the NTT Group. Additionally, examples of OSS efforts at NTT operating companies are reported in the article "Open Source Software and Community Activities Supporting Development of Cloud Services at NTT Communications" [5], focusing specifically on OSS application and enhanced technical capabilities in cloud services at NTT Communications.

The article "Global Expansion of Apache Hadoop/Apache Spark Activities at NTT DATA" [6] introduces OSS for storing and processing massive amounts of data in the AI and IoT fields at NTT DATA, while the use of OSS in IT system operations at NTT Comware is explained in "Achieving Greater Work Efficiency in Systems Failure Analysis Using Elastic Stack" [7]. In particular, in the article on the expansion of Apache Hadoop/Apache Spark activities at NTT DATA, use cases in Japan and abroad involving Apache Hadoop/Apache Spark are described as examples of creating a company's unique value with OSS as discussed above, while the article on Elastic Stack discusses the topic of knowledge expansion achieved by combining Elastic Stack and OSS products for processing massive amounts of data as an example of developing product integration capabilities.

6. Future development

In the future, the capability to adopt OSS will greatly affect the competitiveness of service providers and SIers. The NTT Group is committed to accelerating the adoption of OSS through close collaboration between the NTT laboratories and NTT operating companies.

References

[1] J. Kato, H. Mitake, A. Suda, H. Yamada, K. Okitsu, and S. Kanai, "Open Source Software Efforts to Transform IoT/AI Services," NTT Technical Review, Vol. 16, No. 2, 2018.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201802fa2.html

[2] H. Shina, M. Muroi, H. Kobayashi, H. Takahashi, T. Hibi, T. Kinoshita, J. Kato, T. Fujita, and W. Ishida, "Open Source Software behind NTT Network Services," NTT Technical Review, Vol. 16, No. 2, 2018.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201802fa3.html

[3] K. Tsuyuzaki and T. Yamamuro, "NTT's Contributions to OSS Upstream First Development," NTT Technical Review, Vol. 16, No. 2, 2018.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201802fa4.html

[4] T. Abe, Y. Suga, G. Suzuki, A. Kanamaru, and T. Iwatsuka, "Open Source Software Initiatives Supporting NTT Group Software Development and Operations," NTT Technical Review, Vol. 16, No. 2, 2018.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201802fa5.html

[5] M. Ogura, X. Zhang, M. Matsushita, T. Matsumoto, A. Purohit, and K. Harada, "Open Source Software and Community Activities Supporting Development of Cloud Services at NTT Communications," NTT Technical Review, Vol. 16, No. 2, 2018.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201802fa6.html

[6] R. S. Ranaweera and A. Ajisaka, "Global Expansion of Apache

*6 Maintainer: A member or members of an open community responsible for updating or adding source code (may be called by a different name depending on the community).

Hadoop/Apache Spark Activities at NTT DATA," NTT Technical Review, Vol. 16, No. 2, 2018.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201802fa7.html
[7] N. Maeda, N. Agetsuma, K. Kamimura, Y. Suenaga, S. Takebayashi, and K. Yamashita, "Achieving Greater Work Efficiency in Systems Failure Analysis Using Elastic Stack," NTT Technical Review, Vol. 16, No. 2, 2018.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201802fa8.html

Masahisa Kawashima
Vice President, NTT Software Innovation Center.

He received a Ph.D. in electrical engineering from Waseda University, Tokyo, in 1994 and an M.Sc. in technology management from MIT's Sloan School of Management, USA, in 2002. Dr. Kawashima is the head of NTT Software Innovation Center. He has been engaged in technology and business development at NTT since joining the organization in 1994. With his enthusiasm for technology strategies, he has accomplished several initiatives to update NTT's business operations in line with new technology trends. He played a leading role in the organization of the NTT Open Source Software Center in 2006, the strategic Wi-Fi service renewal for NTT WEST in 2011, and the release of the NFV-enabled software-defined wide area network service platform called CLOUDWAN for NTT's global businesses in 2017.

Open Source Software Efforts to Transform IoT/AI Services

Jun-ya Kato, Hitoshi Mitake, Akihiro Suda, Hideki Yamada, Kengo Okitsu, and Sekitoshi Kanai

Abstract

The NTT laboratories have been focusing on expanding the use and development of open source software (OSS), primarily through the NTT Software Innovation Center (NTT SIC). NTT's active contributions to OSS communities such as OpenStack and Apache Hadoop have yielded productive results with refinement of the company's own technologies and training of personnel. Meanwhile, the wave of OSS is steadily spreading to new fields, including virtualization and artificial intelligence. NTT SIC is also engaged in these efforts, and its contributions are beginning to bear fruit. In this article, NTT SIC's work in OSS is introduced from a technical perspective and its community activities are also described.

Keywords: container, deep learning, parallel distributed processing

1. Introduction

Container-based virtualization technologies (hereinafter, container technologies) have gained attention in recent years, and the opportunities for their use have been increasing. Container technologies stand in contrast to virtual machine technologies from the standpoint of virtualization of the software execution environment. This set of technologies has lower overhead than hypervisor-based virtualization technologies, which places guest operating systems (OSs) in an intermediate layer, so launching and stopping containers is extremely fast.

Improvements to these technologies such as the addition of functions are also happening at a vigorous pace. Because container technologies generate a large quantity of containers within the same environment for an actual use case, and these containers are created and deleted frequently, container management is complex. As a result, open source software (OSS) for container management and execution has been proposed. Docker, Kubernetes, and etcd are introduced in this article as examples of such OSS platforms.

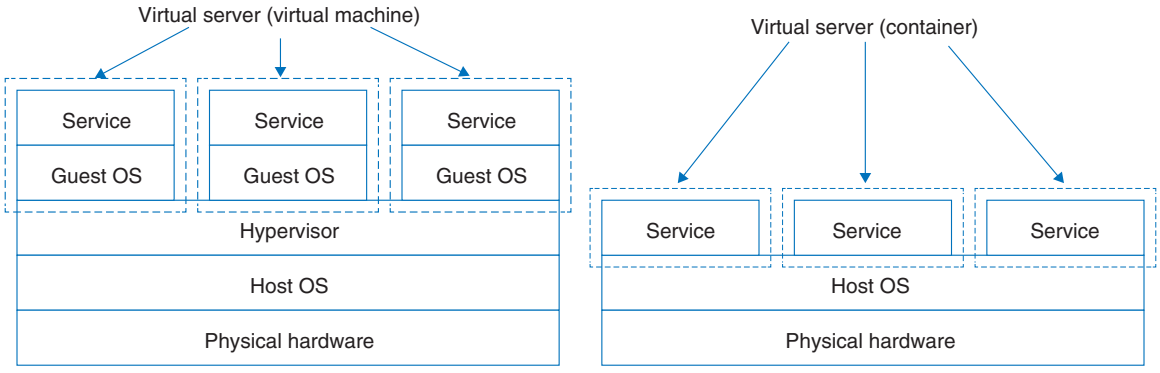
Next, we describe OSS in the field of artificial intelligence (AI). Here, we introduce deep learning frameworks, which are execution environments for deep learning techniques that have rapidly gained widespread use in recent years. Efforts to advance frameworks that support parallel distributed processing have also increased recently due to the growth of large-scale data models and the need to improve processing speeds. In this article, we present an overview of representative deep learning frameworks and examine technologies supporting their use, for example, graphics processing units (GPUs) for parallel distributed deep learning and high-speed interconnects.

2. Container technologies

In this section, we present an overview and describe examples of container technologies.

2.1 Overview of container-based virtualization

A set of technologies called container-based virtualization has been gaining attention in recent years. With container-based virtualization, only applications needed for services are retained between multiple virtual servers, and the virtual servers share an OS kernel (**Fig. 1**). In contrast to hypervisor-centered hardware emulation-based virtualization, container-based virtualization has little overhead that comes



(a) Hypervisor-based virtualization (b) Container-based virtualization

Fig. 1. Difference between virtual machines and containers.

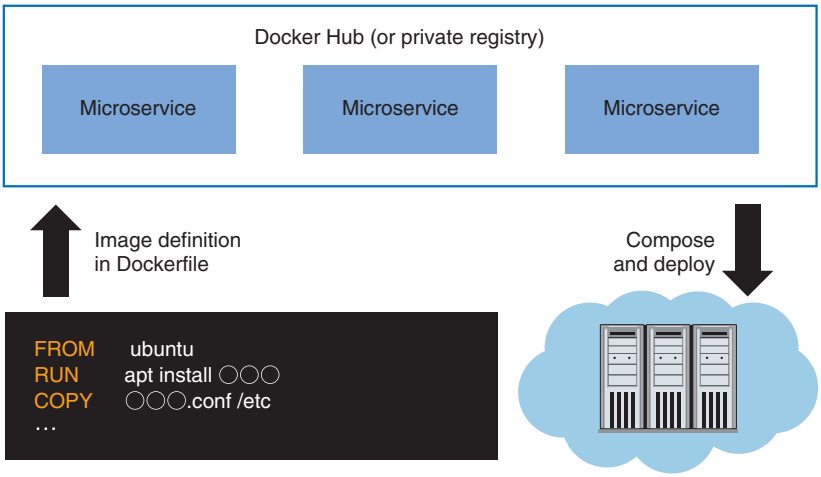


Fig. 2. Structure of Docker.

from the interposition of guest OSs. As a result, starting and stopping containers is extremely fast.

Many kinds of container platform software can build container images by combining the software needed to execute the images. We introduce well-known examples of such software here.

2.2 Docker-related efforts

Docker is a well-known container-management and execution platform. It uses a configuration file called Dockerfile that defines the content of the container to build a container image. The image can be distributed on a sharing platform called Docker Hub (**Fig. 2**). With this feature, users can compose images created by themselves or other users to easily build a service with the needed functions. Docker also comes standard with an orchestrator function called Swarm-mode, which enables distributed execution of containers on multiple nodes based on the workload. In addition, container images built with Dockerfile are highly portable. They can also be executed on other orchestrators such as Kubernetes and Mesos.

The only Docker/Moby* maintainer from Japan (as of October 2017) was selected from the ranks of the NTT Software Innovation Center (NTT SIC) in November 2016. Maintainers are developers who

* Docker/Moby: The transition of the Docker project to the Moby project began in April 2017. Product development by Docker, Inc. is based on the results of Moby's open development.

have the privilege of reviewing proposed changes to the source code such as bug patches and new functional additions, and deciding on whether to adopt the proposals. In other OSS communities, a committer is a role equivalent to that of a maintainer.

Maintainers are selected from developers who have regularly contributed to the community by the agreement of current maintainers. The maintainer from NTT SIC has been highly praised for his sustained activities of slightly less than a year, including analyzing and fixing issues related to Docker filesystem drivers and proposing container metadata management functions. Since assuming the role of a maintainer, he has taken on the responsibility of reviewing proposals from other developers and deciding on their adoption, as well as contributing a new functional proposal that shortens the container image generation time. This idea was included in the BuildKit project, which comprehensively rewrites the container image builder functions, in July 2017.

2.3 Kubernetes and etcd-related efforts

Kubernetes is a container scheduler for managing clusters on a large scale of several thousand units. Its main role is to deploy containers on resource-appropriate machines in response to task creation requests. Its strengths are its isolation of resources for each task by using container technologies such as Docker, smooth distribution of applications, and flexible scheduling based on the priority of each task.

The frequency of generating events such as notifications of user requests and task completions depends on the purpose and size of the cluster. In an enormous cluster, events are generated at high frequency. Storing the states of tasks in high-reliability and high-performance databases is therefore essential for stably operating a large-scale cluster.

A well known and leading OSS for storing task states is etcd. It is a distributed key value store that uses the Raft distributed consensus protocol. Its features are high availability and strong consistency. Kubernetes uses etcd for saving internal management information.

NTT SIC has been recognized for its sustained efforts to improve etcd, including improving product quality by fixing bugs related to etcd's Raft component and improving its authentication function at the design level. As a result, a maintainer of the etcd project was selected from NTT SIC in June 2016. There are only seven maintainers of the etcd project worldwide (as of October 2017).

In addition, Namazu, a testing tool for distributed systems developed by NTT SIC, is being used for etcd product improvement. At present, it is being incorporated into the official CI (continuous integration: automatic test environment) of the etcd project.

NTT SIC is also making efforts to expand the applicability of Kubernetes by working on the development of a scheduler that takes into account network topology and to provide support for remote direct memory access (RDMA), a low-latency and central processing unit (CPU)-efficient network protocol. These technologies are critical for deep learning frameworks, described below.

3. Deep learning frameworks

Here, we describe the current situation regarding deep learning frameworks and also touch on expected future trends.

3.1 Existing frameworks

The growing popularity of deep learning in recent years has led to the release of deep learning frameworks as OSS (**Table 1**). Deep learning methodology learns from data the parameters of a defined model (network structure used for deep learning) in response to a task. With this trained model, data such as images and speech can be recognized with high accuracy. Through the learning process, a great number of derivatives within the model are calculated and used. However, implementing the computational process manually each time a task or model is modified becomes extremely cumbersome. A framework has functions for automatically carrying out the computational process necessary for learning. Users can use deep learning simply by defining the model to be used and the learning method.

Companies have led the development of deep learning frameworks. For example, outside Japan, TensorFlow has been developed by Google, Microsoft Cognitive Toolkit (CNTK) by Microsoft, and MXNet by Amazon. Facebook has been developing Caffe2 as a framework for commercial development and released PyTorch as a research-oriented framework. TensorFlow's main benefit is that it has the most users and the largest community of all the frameworks. CNTK's strength is considered to be its fast performance and high scalability, and for MXNet, its flexibility in implementation. In Japan, Preferred Networks (PFN) has released Chainer, which provides support in Japanese. Chainer also uses a flexible scheme in which the model structure is automatically determined as data are processed. This scheme has spread to other

Table 1. List of deep learning frameworks.

Framework	Platform	Interface	Features	Main developer
TensorFlow	Linux, macOS, Windows, Android, iOS	Python, Java, C++, Go, Julia, C#, R, Haskell	Largest community; Can visualize with TensorBoard	Google
MXNet	Linux, macOS, Windows, Android, iOS	Python, R, Julia, Scala, Go, (Matlab, Javascript)	Can be flexibly implemented to suit the task; Numerous platforms	Amazon, Baidu, Carnegie Mellon Univ.
CNTK	Linux, Windows	Python, C++	Fast (especially RNN); High scalability	Microsoft
Chainer	Linux	Python	Effective implementation schemes for natural language processing	PFN
PyTorch	Linux, macOS	Python	Effective for natural language processing derived from Torch and Chainer	Facebook, Twitter, Salesforce
Caffe2	Linux, macOS, Windows, Android, iOS	C++, Python	Numerous platforms; High-speed, memory-efficient, high scalability	Facebook

frameworks.

Support for parallel distributed processing by frameworks is advancing. In Japan, Chainer is making headway in providing similar support. PFN has released an additional package for Chainer called ChainerMN, which supports distributed learning. The company claims that the product is six times as fast as TensorFlow (as of February 2017).

None of these frameworks has emerged as the decisive winner in terms of the work environment or processing capabilities. NTT SIC is therefore continuing to test each framework in accordance with a variety of use purposes.

3.2 Technological trends and challenges going forward

(1) Parallel distributed processing using high-performance computing resources

To improve the accuracy of deep learning, it is said that in general, the model must have deep layers. Consequently, however, the number of parameters to learn increases and the computational complexity explodes. The problem arises in which computations require several days to several weeks. The use of high-performance computing resources, which boast superior computational capabilities, and parallel distributed processing, which executes multiple computations in parallel, is being investigated in order to solve this problem. Frameworks are being extended to support this trend.

The use of GPUs as high-performance computing resources is becoming widespread. GPUs are especially strong when it comes to matrix operations, which make up almost all deep learning computations. At present, products from NVIDIA, which have become the de-facto standard, feature computational performance that is several times to several tens of times greater than that of CPUs. Several frameworks use cuDNN, a library provided by NVIDIA, to describe computation for their learning component. Thus, we see an example of the use of high-performance GPUs.

Parallel distributed processing techniques mainly use a method called data parallelism, which makes use of multiple GPUs. Data parallelism is a method in which the same model is copied to multiple GPUs, and different parameter learning computations are executed on the learning data in parallel. The parameters are then updated based on communications from each GPU. Vast improvements in speed can be expected by parallelizing parameter learning, for which until now learning data had been input into one GPU sequentially. As a result, active efforts are underway to support each framework. Work is also being carried out to provide multi-GPU support in one machine and multi-node support using GPUs in multiple machines.

We can thus expect to further improve speed by increasing the number of machines and GPUs. However, it is known that as parallelism increases, communication to update parameters becomes the performance bottleneck. Research on how to skillfully arrange parameter data and apply high-performance communication techniques has begun for each framework. The discussions and implementations are expected to be energetic from here on.

Going forward, NTT SIC plans to study which architectures are the most optimal for actual problems.

(2) Communication processing technologies to support parallel distributed processing

The use of high-speed interconnects such as InfiniBand, widely used in the field of high performance computing (HPC), is considered effective as a high-speed communication approach. RDMA is used instead of general Transmission Control Protocol/Internet Protocol (TCP/IP) to achieve low-latency communications when using high-speed interconnects to their maximal limits. RDMA is capable of reading/writing the memory of remote machines without CPUs as the intermediary. General communication programs are described using socket application programming interfaces (APIs). However, RDMA communication programs use low-level descriptive APIs called verbs and Message Passing Interface (MPI), a parallel computing interface widely used for HPC.

RDMA support for TensorFlow is steadily moving forward. RDMA verb communication was implemented in April 2017 [1], and GPU Direct RDMA was implemented in August 2017 [2]. Additionally, distributed processing using MPI for inter-node communication is being developed for ChainerMN [3]. Partial implementation and discussions in communities have also begun for MXNet and CNTK [4]. In the near future, RDMA-related activities will become even more critical.

Because RDMA communication procedures are more complex in actual use than general TCP/IP, NTT SIC is presently studying ways to simplify implementation by abstracting APIs in order to popularize RDMA communication.

4. Future development

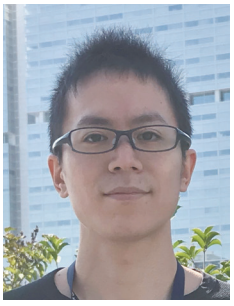
In this article, we introduced new OSS efforts being carried out by NTT SIC in the areas of container-based virtualization technologies and deep learning frameworks. Because virtualization is a required technology for system design and architecture, the application of lightweight virtualization container technologies will continue to expand going forward. Also, deep learning frameworks will advance in their ability to handle parallel distributed processing.

The NTT laboratories are continuing to advance research and development related to these technologies. At the same time, the labs are focusing on using their own approaches. For example, the NTT laboratories are building parallel distributed deep learning frameworks on container clusters and evaluating the frameworks during actual use. These efforts not only contribute to AI research, but are also expected to be applied to a wide range of other problems.

References

[1] ibverbs-based RDMA path, https://github.com/tensorflow/tensorflow/pull/8943
[2] GPU Direct RDMA Out-of-Band Tensor Transport, https://github.com/tensorflow/tensorflow/pull/11392
[3] ChainerMN distributed deep learning frameworks (in Japanese), https://research.preferred.jp/2017/05/chainermn-beta-release/
[4] CNTK v.2.0 RC 2 Release Notes, https://github.com/Microsoft/CNTK/wiki/CNTK_2_0_RC_2_Release_Notes

Jun-ya Kato
Senior Research Engineer, Distributed Computing Technology Project, NTT Software Innovation Center.
He joined NTT Information Sharing Platform Laboratories in 2002, where he studied next-generation Internet protocol IPv6. He joined the architecture design project for IPv6 Internet service provider (ISP) support on NGN (Next Generation Network) in 2008. He also technically supported NTT operating companies participating in World IPv6 Day and World IPv6 Launch, which were global IPv6 deployment events in 2011 and 2012, respectively. He then moved to the Technology Development Department, NTT Communications, where he developed network security services such as DDoS (distributed denial of service) protection in the ISP backbone. His current research is focused on software-defined networking and distributed computing platforms.



Hitoshi Mitake
Researcher, Distributed Computing Technology Project, NTT Software Innovation Center.
He received a B.E. and M.E. from Waseda University, Tokyo, in 2010 and 2012. He joined NTT in 2012 and has been working on distributed storage systems and cluster management systems.

Akihiro Suda
Researcher, Distributed Computing Technology Project, NTT Software Innovation Center.
He received a B.E. and a Master of Informatics in computer science from Kyoto University in 2012 and 2014. He joined NTT in 2014 and has been a maintainer of several kinds of container-related OSS such as Moby (formerly Docker), Moby BuildKit, and CNCF containerd.

Hideki Yamada
Research Engineer, Distributed Computing Technology Project, NTT Software Innovation Center.
He received a B.E. and M.E. from Tokyo Institute of Technology in 2005 and 2007. He joined NTT Information Sharing Platform Laboratories in 2007. His recent research area is distributed storage.

Kengo Okitsu
Researcher, Distributed Computing Technology Project, NTT Software Innovation Center.
He received a B.S. and M.S. in computer science from the Tokyo Institute of Technology in 2008 and 2010. He joined NTT in 2010 and has been studying cloud resource management and scheduling.

Sekitoshi Kanai
Researcher, Distributed Computing Technology Project, NTT Software Innovation Center.
He received a B.E. and M.E. in physics and physico-informatics engineering from Keio University in 2013 and 2015. He joined NTT in 2015 and has been studying deep learning algorithms.

Open Source Software behind NTT Network Services

Hideki Shina, Masahito Muroi, Hiroaki Kobayashi, Hirokazu Takahashi, Tomoya Hibi, Takeshi Kinoshita, Jun-ya Kato, Tomonori Fujita, and Wataru Ishida

Abstract

Open source software (OSS) has been gaining importance in many industries, including network services. Members of the NTT laboratories have been involved in numerous activities in OSS communities both as users and as active contributors by doing coding and by providing requirements for telecom operators. These efforts are expected to give us a significant advantage over our competitors. This article introduces the involvement of the NTT laboratories in OSS community activities related to OpenStack Blazar, Lagopus, and GoBGP.

Keywords: Blazar, Lagopus, GoBGP

1. Transformation for the development of network services

The use of open source technologies has been increasing in various fields, including the telecommunications field, which traditionally relied on the use of well-integrated proprietary software and hardware from specific vendors. Open source software (OSS) has been adapted in network infrastructures just as it was adapted in information technology infrastructures. Current trends in network infrastructures are shifting to utilizing white box (non-branded) systems that are composed of standardized hardware, OSS, and sometimes self-developed technologies. In line with these trends, NTT has been very active in OSS communities, which should contribute to improving the speed and flexibility of development of our network infrastructures.

The NTT laboratories develop various open source products and contribute to many OSS communities, including OpenStack—a major OSS for cloud infrastructure, Open Networking Foundation (ONF)—an organization for standardizing software-defined networking (SDN), and Open Platform for Network Functions Virtualization (OPNFV)—an organization that facilitates the development and evolution of network functions virtualization (NFV) components across various open source ecosystems. These efforts help us to tailor OSS for our services. The NTT laboratories' activities in OSS communities are introduced in the following section.

2. Blazar

Blazar [1] is a service for reserving OpenStack resources such as Compute. It can be used for efficiently managing ownership of limited resources and guaranteeing resource availability for highly prioritized services.

The NTT Software Innovation Center (NTT SIC) and NTT Network Service Systems Laboratories (NTT NS Labs) are jointly promoting the Blazar project. NTT SIC has a lot of experience in developing OSS such as OpenStack. NTT NS Labs, on the other hand, has developed many network services in production. The synergies gained by the collaboration of these two organizations have been accelerating the development of Blazar.

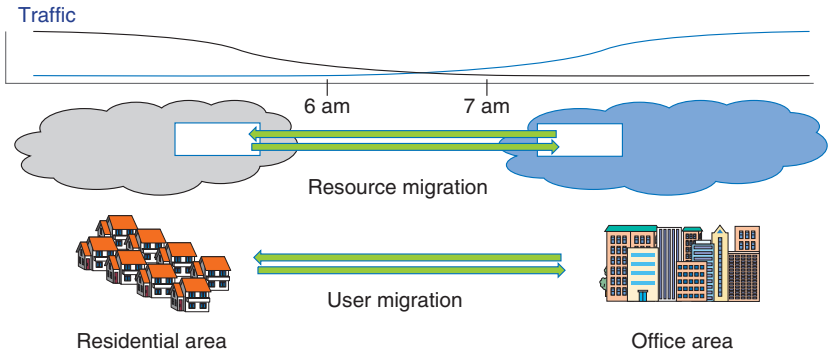


Fig. 1. Blazar use case: Resource migration.

2.1 NTT activities

The Blazar project was initiated in 2013. The project was named Climate at that time and renamed Blazar later. The first version of Blazar was released with OpenStack Icehouse in 2014. Around that time, the OPNFV Promise project [2] was organized for defining the requirements of resource reservation in NFV and accelerating implementation of related open source projects. The NTT laboratories and NTT DOCOMO have been participating in it. Members of the Promise project had been looking for an open source project that could potentially satisfy Promise requirements for years to come. They eventually approached the Blazar project members, and we began contributing to it. We have since made numerous contributions, which has led to our leading the Blazar project as a Project Team Lead and as core reviewers in the latest release cycle.

2.2 Recent Blazar releases

In March 2017, a new version of Blazar was released for the first time in three years with OpenStack Ocata. This version of Blazar allows operators to reserve computing resources in a host unit. New features such as instance-based reservation and a dashboard interface have since been developed and included in the latest version of Blazar, which was released with OpenStack Pike in August 2017.

We must pay close attention to maintaining consistency between Blazar and other core components such as Nova. In May 2017, we had discussions with Nova project members at the OpenStack Summit Boston, and we agreed that Nova would focus on the current use of resources, while Blazar would focus on the future use of resources. Such decisions are very important for keeping OpenStack simple and well organized.

2.3 Use cases in NFV

The OPNFV Promise project has decided to use Blazar. The following use cases are assumed based on actual operations of telecom operators:

(1) Resource migration: Resources can be reserved based on the prediction of daily traffic. For example, traffic increases in office areas in the daytime and in residential areas at night. An operator can use Blazar to reserve a sufficient amount of resources in office areas in the daytime and in residential areas at night (**Fig. 1**).

(2) Maintenance work: Resources can be reserved that will be used during planned maintenance work such as software updates.

(3) Guaranteed provisioning: Services that need a large amount of resources can fail because of a lack of resources while starting up. Sufficient resources can be reserved beforehand to prevent such failures.

2.4 Future roadmap

Blazar was approved as an official OpenStack project in September 2017. This will accelerate its development. To satisfy the needs of users—especially research institutes and telecom operators—we have been developing new features and plan to release a new version with OpenStack Queens in February 2018. In this release, we are focusing on improving reliability by adding new features such as monitoring

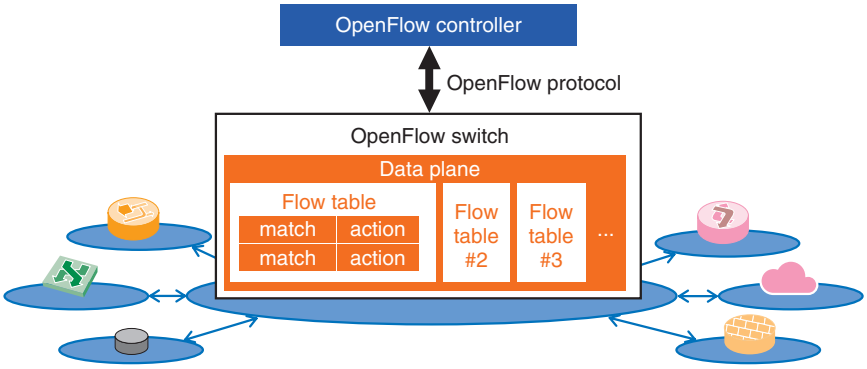


Fig. 2. OpenFlow switch.

of reserved resources and availability zone support [3]. We also plan to support new resources such as network and storage resources in the future.

3. Lagopus

Lagopus [4] is a software switch that enables fast packet forwarding on a general-purpose server while having the highest compatibility with OpenFlow Switch Specification 1.3. Lagopus and other OpenFlow switches operate in accordance with instructions coming from OpenFlow controllers, which specify match conditions and actions (i.e., what actions to take against which types of packets) (**Fig. 2**). Lagopus supports Ethernet, IPv4 (Internet protocol version 4), MPLS (multiprotocol label switching), PBB (provider backbone bridging), and other types of protocols and is also equipped with quality of service functionalities.

The development of Lagopus was initiated by NTT in 2013 with the aim of expanding the applicability of SDN and NFV, both of which have since become widely used technologies in use cases ranging from datacenters to wide area networks, and in 2014 it became open source. Lagopus employed Data Plane Development Kit (DPDK), an input/output library for fast packet handling that became available as OSS just before the development started. The open source Lagopus performed over-10-Gbit/s packet forwarding and processing with 1 million flow table entries, an achievement that was thought to be difficult at that time and thus attracted lots of attention. It now provides 40-Gbit/s packet forwarding and supports fast interconnections with virtual machines as well as a wide variety of tunneling protocols.

3.1 Development of Lagopus

Open-sourcing Lagopus made it possible to achieve cooperation from various organizations and individuals in and outside NTT. It has been used in research and development aimed at commercial uses as well as in field tests exploring novel networks that make the most of OpenFlow capabilities. In general, one type of network equipment can be connected with many other types in a wide variety of uses, which makes it very difficult to carry out interoperability testing exhaustively. By conducting a number of field tests in cooperation with developers, however, we have continuously proved Lagopus's interoperability with other equipment in many practical use cases.

For example, Interop Tokyo, an annual exhibition on information and communication technologies, was a good opportunity to feature Lagopus for that purpose. In the three consecutive years from 2015, it was used in ShowNet, a dedicated network showcasing cutting-edge technologies while providing Internet access to both exhibitors and participants. In each year, Lagopus successfully proved interoperability in different use cases and contributed to the operation of novel networks.

In 2015, Lagopus served as one of the switches that constituted SDN-based Internet exchange (SDN-IX) in ShowNet. We collaborated with members of a Japan-Europe joint research project called NECOMA (Nippon-European Cyberdefense-Oriented Multilayer threat Analysis) and achieved not only layer-2 switching but also interconnection of virtual local area networks and segregation of malicious traffic (**Fig. 3**), winning a special award for ShowNet in the category of Software Defined Infrastructure.

In 2016, we operated Lagopus so as to provide interconnections of virtual networking functions.

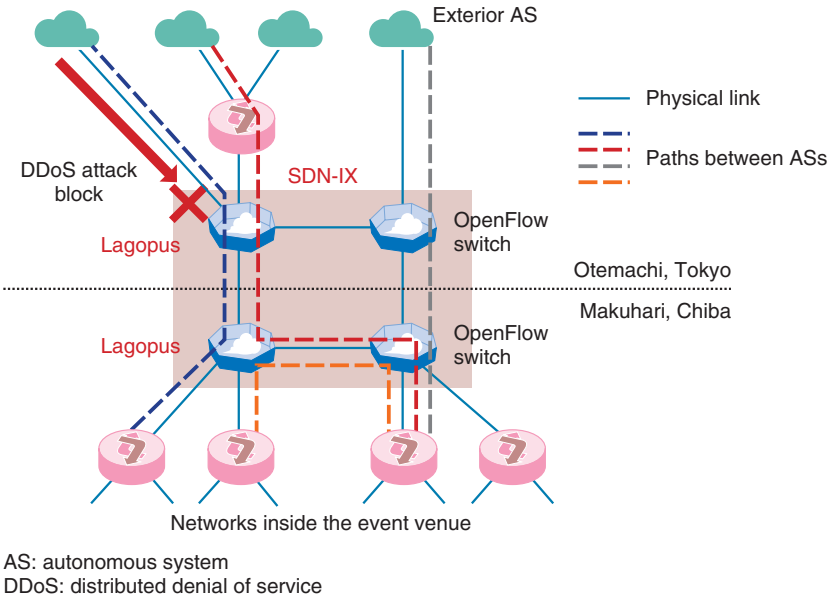


Fig. 3. SDN-IX.

Lagopus used virtual network interface cards (NICs) rather than physical NICs to achieve flexible and fast traffic control. This functionality, with DPDK employed, had been developed for Lagopus and was later contributed to the DPDK community.

In 2017, Lagopus demonstrated tunneling functionality and its interoperability. OpenFlow 1.3 does not support tunneling, which makes it difficult to achieve flexible flow control in an overlay-type network. The tunneling functionality that had been added to Lagopus as an enhancement of OpenFlow 1.3 successfully proved to expand the applicability of the specification.

All of these efforts have helped to enhance Lagopus's functionalities and improve its interoperability and stability.

3.2 Evolving Lagopus

We are now working to equip Lagopus with routing functionalities. Although we have confirmed through the experiments and discussions with outside users that flexible flow control supported by OpenFlow was highly useful, we have also come to conclude that interconnection with existing routers is necessary for Lagopus to be utilized in a commercial network. Against this background, we began developing routing functionalities with the help of the Lagopus community. The initial version was released as OSS in August 2017. The functionalities currently available for Lagopus Router are limited to some simple ones. However, we are making progress in enhancing them for the purpose of achieving a router that enables flexible traffic control using both existing routing protocols and OpenFlow-like match-action instructions.

4. GoBGP

Border Gateway Protocol (BGP) is one type of protocol for performing routing exchange. It has become widely used for routing exchange on the Internet between organizations such as Internet service providers and corporate enterprises known as autonomous systems. BGP has been applied in a variety of use cases, so it is not limited to the Internet. For example, some operators use BGP to exchange label information on a carrier backbone to provide IP virtual private network (VPN) connections, or they apply BGP to network control in datacenters to achieve a distributed arrangement of computing resources such as servers while maintaining scalability. These use cases demonstrate the expansion of BGP application fields.

GoBGP [5] is a BGP implementation written in the Go language, a programming language developed by Google. Its language specifications include standard functions for describing parallel processing, and an extensive library of functions has been prepared. The

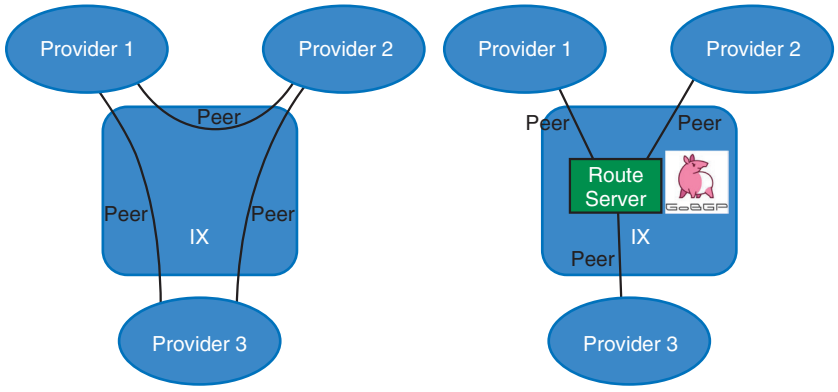


Fig. 4. Role of route server in Internet exchange (IX).

Go language is especially popular in the system programming field. Additionally, because today's server hardware is mounted with multicore central processing units, GoBGP has been designed for high performance by leveraging the features of the Go language and abundant hardware resources.

The main functions currently supported by GoBGP are listed below:

- Provides both application programming interface (API) (gRPC) and command line interface as control interfaces
- Supports a vendor-independent configuration model (OpenConfig)
- Monitoring (MRT: Multi-Threaded Routing Toolkit; BMP: BGP Monitoring Protocol)
- Route reflector (includes Add-Path function)
- Route server
- Policy control
- Resource public key infrastructure (RPKI)
- Flowspec
- VPN functions (Ethernet VPN, layer-3 VPN, virtual routing and forwarding)
- Graceful restart
- Operating system (OS) data plane control (zebra* linking)

In short, GoBGP already supports a variety of functions, but given that BGP use cases are much broader, we are working to add more useful functions for many users while promoting the participation of outside developers by developing GoBGP as OSS.

4.1 GoBGP activities

(1) API for smooth linking with external systems

GoBGP has been commercially deployed as a route server function in the RouteFEED service of the Japan Network Access Point (JPNAP) provided by INTERNET MULTIFEED CO. RouteFEED is a service that mediates routing exchange between various types of providers connected to JPNAP. Without RouteFEED, such providers have to perform routing exchange between themselves, but in the new system, a provider only has to establish a BGP connection (peer) with the route server to perform routing exchange with another provider (**Fig. 4**). The route server must set policy information for each peer, so scalability with respect to the number of peers is required. We have applied high-performance GoBGP to meet this requirement. Additionally, a GoBGP API function for linking with an external operation system is being used to automate [6] complicated setting operations performed manually in the past.

(2) Integration for white box switch control

A white box switch is hardware equipped with general-purpose forwarding-type application-specific integrated circuits (ASICs). The network OS (software) for controlling this hardware can be selected according to the application. An open network OS is often based on Linux. A BGP router can be configured by combining GoBGP with a white box switch

* Zebra: A function included in routing implementations such as Quagga and FRR (Free Range Router). It includes a function for reflecting received routing information in the routing table of the OS kernel in charge of forwarding.

mounted with a Linux-based network OS.

To transfer packets, forwarding equipment must write forwarding rules in a routing table called a forwarding information base (FIB). In GoBGP, there is an FIB operation mechanism linked to the zebra function that enables routes received by BGP to be reflected in the FIB as forwarding rules. For network OS control that targets forwarding ASICs, setting rules in the FIB of that hardware enables the configuration of a BGP router capable of hardware forwarding.

(3) Application to container-oriented virtual network control

Calico [7] is a powerful OSS package for configuring a virtual network required for distributed execution of multiple containers. It supports major container orchestrators such as Kubernetes, Docker, and Mesos and can control connections between various types of systems. BGP has been used for IP connection control between containers, and while BIRD has been traditionally used as a BGP function, GoBGP, which excels in scalability, can also be applied to large-scale use cases.

4.2 GoBGP future outlook

Use cases applying BGP will continue to expand, and the emergence of new protocol extensions can be expected. We aim to apply GoBGP to a variety of use cases through the participation of outside developers in an open-development environment. We also aim to promote the growth of the business market using OSS and to achieve technical advances and invigorate business toward creation of a software-based network infrastructure.

References

[1] Blazar, https://docs.openstack.org/blazar/latest/
[2] Promise, https://wiki.opnfv.org/display/promise/Promise
[3] Blueprints for Blazar, https://blueprints.launchpad.net/blazar
[4] Lagopus, http://www.lagopus.org/
[5] GoBGP, https://osrg.github.io/gobgp
[6] Press release issue by NTT on September 30, 2016. http://www.ntt.co.jp/news2016/1609e/160930b.html
[7] Calico, https://www.projectcalico.org/

Trademark notes

Hideki Shina
Senior Research Engineer, Supervisor, NTT Network Service Systems Laboratories.

He received a B.E. and M.E. in information and computer engineering from Kanazawa Institute of Technology in 1994 and 1996. He joined NTT Network Service Systems Laboratories in 1996 and studied a distributed and high-availability platform for switching systems. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

Masahito Muroi
Researcher, NTT Software Innovation Center.

He received an M.E. in computer science from the University of Electro-Communications, Tokyo, in 2011. He joined NTT the same year. He is now working as a cloud architect for NTT's public/private cloud. He is also a Project Team Lead in the OpenStack Blazar project and one of the core reviewers in the OpenStack Congress project. He has been working on the development of NTT's cloud with OpenStack since the Diablo release in 2011.

Hiroaki Kobayashi
Researcher, NTT Network Service Systems Laboratories.

He received a B.E. and M.E. in computer and information sciences from Tokyo University of Agriculture and Technology in 2011 and 2013. He joined NTT Network Service Systems Laboratories in 2013. He specializes in the area of distributed computing, telecom networks, and NFV. He is an active contributor to the OpenStack and OPNFV communities. He is a core developer of the OpenStack Blazar project and a contributor to the OPNFV Promise project. He is a member of IEICE.

Hirokazu Takahashi
Senior Research Engineer, NTT Network Innovation Laboratories.

He received a B.E. and M.E. in electrical engineering from Nagaoka University of Technology, Niigata, in 2000 and 2002. He is currently researching software architectures for network systems and high-performance packet processing techniques.

Tomoya Hibi
Researcher, NTT Network Innovation Laboratories.

He received a B.E. and M.E. in computer science and engineering from Toyohashi University of Technology, Aichi, in 2010 and 2012. His current research is focused on software networking and packet processing techniques.

Takeshi Kinoshita
Research Engineer, NTT Network Innovation Laboratories.

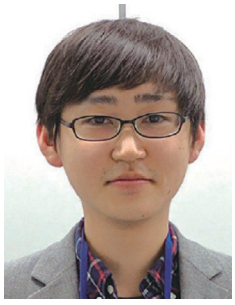
He joined NTT Optical Network Systems Laboratories in 1996, where he studied optical fiber communication systems in access networks and their management. He then moved to the research and development center of NTT WEST, where he was responsible for developing and introducing commercial services, including broadband internet access, IPv4/v6 VPN, and wide area Ethernet. He also worked on reducing the energy consumption of various components in datacenters such as servers, network equipment, and air conditioning systems. His current research interests include SDN, NFV, and other network softwarization related technologies. He is also involved in standardization efforts for various standards organizations.

Jun-ya Kato
Senior Research Engineer, NTT Software Innovation Center.

He joined NTT Information Sharing Platform Laboratories in 2002, where he studied next-generation Internet protocol IPv6. He joined the architecture design project for IPv6 Internet service provider (ISP) support on NGN (Next Generation Network) in 2008. He also technically supported NTT operating companies participating in World IPv6 Day and World IPv6 Launch, which were global IPv6 deployment events in 2011 and 2012, respectively. He then moved to the Technology Development Department, NTT Communications, where he developed network security services such as DDoS (distributed denial of service) protection in the ISP backbone. His current research is focused on SDN and distributed computing platforms.

Tomonori Fujita
Senior Research Engineer, NTT Software Innovation Center.

He received a B.E. and M.E. from Waseda University, Tokyo, in 1998 and 2000. He joined NTT in 2000 and has been researching operating systems. He is a member of the Information Processing Society of Japan and the Association for Computing Machinery.



Wataru Ishida
Researcher, NTT Software Innovation Center.

He received a B.S. in information science in 2011 and an M.E. in engineering from the University of Tokyo in 2013. He joined NTT Software Innovation Center in 2013 and has been working on the development of SDN software, in particular, Ryu SDN Framework and GoBGP.

NTT's Contributions to OSS Upstream First Development

Kota Tsuyuzaki and Takeshi Yamamuro

Abstract

Open source software (OSS) has been used in various information technology systems in recent years. OSS is developed by the collaborative OSS community, which consists of many companies and individuals. Additionally, several large cloud service providers such as Google and Facebook have released their own software used in their services as open source to continue to improve the software in the OSS community. We explain here the advantages of companies developing software as OSS and the significance of companies' contributions to OSS communities. In addition, we also describe two examples of the NTT laboratories' involvement in OSS communities.

Keywords: OSS, OpenStack, Apache Spark/Hivemall

1. Upstream first

Open source software (OSS) is computer software whose source code has been made available with a license in which the copyright holder gives anyone the right to freely change and distribute the software for any purpose. OSS is continually upgraded and enhanced as features are added, and its bugs are fixed through an ongoing collaborative effort by the OSS community. In recent years, we have seen a growing tendency for major companies such as Google, Facebook, and IBM to make their own software available as OSS.

Companies that adopt OSS can do so using two strategies: utilizing the software as-is or developing new features to better meet the companies' needs. In both cases, a significant new approach called *upstream first* [1] has emerged. The idea of upstream first is to enable the companies' own developers to work together with the upstream developers and build relationships with the upstream community to improve the OSS. When the first strategy of utilizing OSS as-is is applied, it is beneficial to follow the latest version of the software as upstream first, as it ensures the best quality of the software. With the second strategy of building additional functionalities into the OSS, the upstream first approach is effective in avoiding conflicts between source code lines added by company developers and those of the community version of the OSS. We explain these two strategies more closely using real examples.

1.1 First strategy: following the latest version

Most software packages developed recently must be updated periodically with the latest version to maintain the software quality. While these changes are mostly minor, they sometimes address serious issues such as security vulnerabilities. Most OSS today employs shared identifiers called Common Vulnerabilities and Exposures (CVE)[*1] [2] to manage the impact of vulnerabilities and how they can be corrected. Fixing vulnerabilities typically involves a three-step process: (1) the party having discovered the vulnerability issues a confidential report detailing the vulnerability to the security contact team; (2) core developers in the security contact team devise a fix for the vulnerability working in a closed environment; (3) and finally, the fix is made available, and the CVE is announced to the general public. Therefore, when a company lets its own developers join the

*1 CVE: The standard for information security vulnerability names for managing vulnerability fixes, identifying publicly known vulnerabilities with standard identifiers, and notifying users.

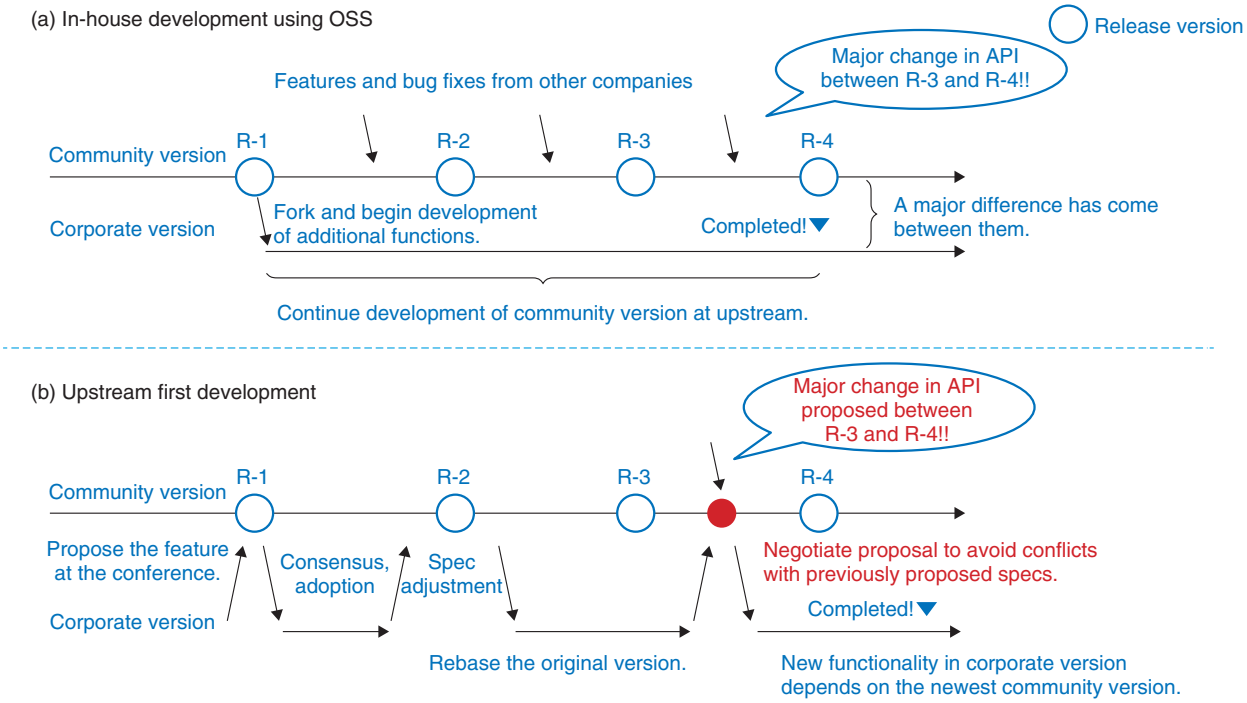


Fig. 1. Difference between in-house development using OSS and upstream first approach.

security contact team responsible for developing the OSS, it brings the company—as a stakeholder—on board so that members are clearly aware of the security vulnerability and can help come up with a solution that fixes the vulnerability when the issue is disclosed. Note that if there is an insufficient number of core developers working on a solution in step (2), the progress made during that process will be slow in addressing the vulnerability. Therefore, taking the lead in supporting the core developers of the OSS ultimately benefits the company, as it helps to safeguard the security of their own corporate systems.

1.2 Second strategy: avoiding conflict with the community version

There may be instances when a company would like to use OSS, but the software lacks some critical capability that the company requires. When this capability is truly indispensable, the quick and easy solution might be simply to develop the features in house. In this approach (**Fig. 1(a)**), the company forks the community software and then develops its own features for its product. However, because OSS is developed by an open community, the application programming interface (API) specifications can be changed in a specific time period. In order to manage the synch with the latest community version, the company has to constantly upgrade that portion of the software that it modified. This is not always easy, especially in the case of OSS communities that are very active, where the portion to upgrade could include tens of thousands or even hundreds of thousands of changed lines of code. Maintaining both the upstream changes and the in-house code would incur an enormous cost much greater than just keeping the firm's own in-house modified version of the software up and running.

The way to avoid this situation is to develop the features as OSS community code and give it back to the community. Software modification costs can be held down by approaching the requests early on at the initial idea stage. This approach may reduce the instances of the software conflicting with other features proposed by other companies in the community. These activities make it possible to minimize differences between a company's added features and the community version of the software that includes significant bug fixes, and it will make it far easier to integrate upstream work into a company's system. Moreover, developing engaging functional features as OSS will enable the feature to be enhanced by other companies. This kind of collaborative approach

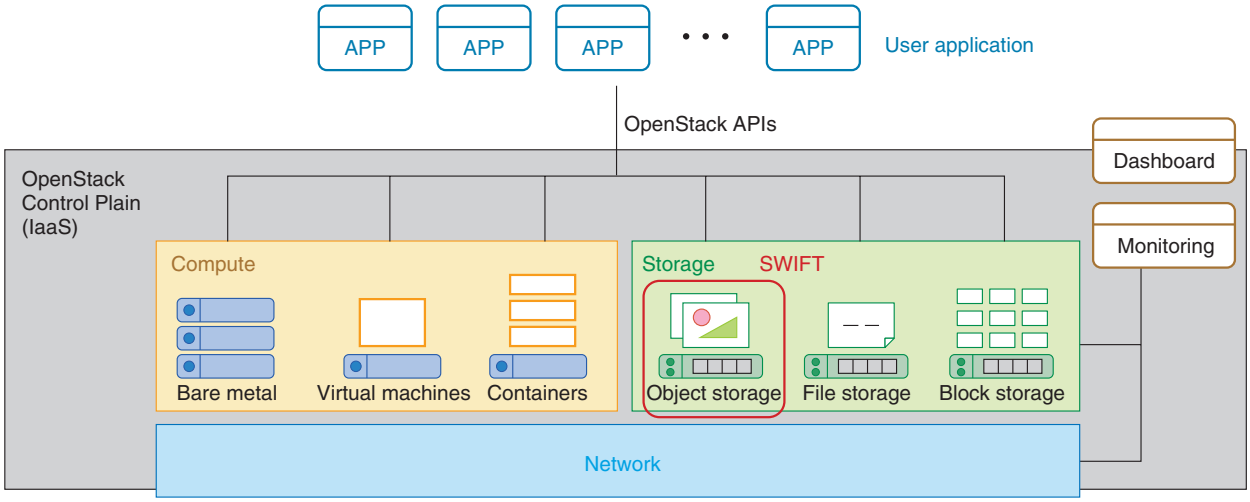


Fig. 2. OpenStack components and Swift.

to OSS development that minimizes conflicts among software features in advance is the way of upstream first, and it results in OSS products that benefit the entire community of OSS users (**Fig. 1(b)**).

This is why companies practice upstream first, develop new software features in collaboration with OSS communities, and increasingly enjoy the advantages of software quality assurance. The NTT Group is involved in the upstream first approach through active participation in several OSS communities including Linux, OpenStack, and Apache. In the next two sections, we describe how the NTT Software Innovation Center accomplishes an upstream first approach in dealing with two open source projects: OpenStack Swift and Apache Spark/Apache Hivemall.

2. OpenStack Swift

OpenStack [3] is an OSS platform maintained by the OpenStack Foundation for implementing infrastructure-as-a-service (IaaS)[*2] solutions. OpenStack consists of several components; an overview is shown in **Fig. 2**. Development policies for each component are discussed at two development conferences that are held biannually: OpenStack Summits, with current attendance in excess of 5000 people, and Project Team Gatherings (PTGs)[*3]. Day-to-day development work is conducted by developers in the globally distributed community via Internet Relay Chat (IRC) and mailing lists. OpenStack is released around six-month cycles, a significantly shorter cycle time than products released by other OSS communities.

OpenStack Swift [4] is an OpenStack component that implements storage technology called *object storage*. Swift object storage provides certain features to ensure data durability: data are stored redundantly; redundancy is monitored within the system; and Swift works to automatically replicate the data in case of a hard drive failure to avoid loss of data. OpenStack Swift has been robust and stable since its release in the first rollout of OpenStack in 2010. Consequently, Rackspace, one of the major players of OpenStack, initially employed the project as a commercial offering.

NTT Group companies have also conducted several case studies on OpenStack [5]. In case studies using OpenStack Swift, the NTT labs have proposed various functional improvements and bug fixes and have made other contributions to the community. Having gained the trust and respect of the community through these activities, the NTT labs have built a community action team that includes one of the core developers in the OpenStack Swift community, and are actively involved in OSS research and development (R&D).

Here, we highlight the NTT labs' involvement in the erasure coding scheme[*4] for OpenStack Swift, an area where NTT's input has been especially influential. OpenStack Swift released a powerful new data

*2 IaaS: Infrastructure as a service, a form of cloud computing that provides virtualized computing resources over the internet.

*3 PTG: The Project Teams Gathering is an event held every six months where OpenStack developers decide the content of the next development cycle.

storage scheme called *erasure coding* in 2015. This new scheme helped to significantly reduce the cost of deploying Swift at NTT Group companies.

The NTT labs have experimented with the upstream first strategy since the beginning of development, and they took the lead in proposing fixes and features to the OpenStack community. For example, we contributed a significant bug fix [6] to the erasure code library. Although the bug was not a security vulnerability, it was a library error that could have resulted in a catastrophic loss of stored data. The NTT labs discovered it during the verification process. Our community action team with a core developer analyzed the problem immediately, quickly verified the library bug responsible, and came up with a fix [7]. These efforts not only preserved the quality of our own products, but also helped save considerable data of many other users in the OpenStack Swift community.

In another case involving the development of features, NTT made a major contribution through its development of *globally distributed erasure coding*. When disaster recovery (DR) clusters[*5] are built with OpenStack Swift across multiple datacenters, global cluster functionality is used to ensure high performance. However, this presents a serious problem when this functionality is used together with an erasure coding scheme; it does not guarantee there will be sufficient data stored in a datacenter, and there is a risk that data might be unavailable if the network connection between datacenters is blocked or interrupted.

The NTT labs contributed the globally distributed erasure coding scheme to resolve this issue. A stable version of the scheme was released in August 2017 just two years after it was first proposed in August 2015, and it was highlighted as a major new functional capability in the new features in the OpenStack community [8]. During the two-year development period for this new scheme, roughly 180,000 lines of code were altered for OpenStack Swift overall, but OpenStack Swift was able to smoothly merge the 15,000 lines of code in the globally distributed erasure coding due to the application of the upstream first strategy in the initial idea proposal and the code development. Furthermore, as the result of upstream first development, the OpenStack Swift community is now considering further work on the feature to make it more convenient and efficient, and this will also benefit NTT Group companies.

Illustrating the use case, making it beneficial as the core technology, and achieving upstream first will enable other companies to join the development, and it will result in the growth of the software, the code, and even the community much like a living thing. Of course, this greatly benefits NTT as well.

It is apparent from these examples that it is important to maintain a healthy OSS community in order to develop new functional capabilities and achieve high quality. Needless to say, this is all vitally important to ensure further quality improvement of an individual company's own products.

3. Apache Spark/Apache Hivemall

Apache Spark [9] is an OSS distributed parallel processing framework developed by AMPLab, a research group at the University of California, Berkeley. Apache Spark includes not only a core component for distributed parallel execution, but also many useful libraries for SQL, machine learning, and streaming, as shown in **Fig. 3**. Apache Hivemall [10] is an OSS distributed machine learning library accepted as an incubator project by the Apache Software Foundation (ASF) in October 2016. Makoto Yui at Treasure Data, Inc. is an original author of Hivemall; NTT developers also became involved in the community activity at the beginning of 2016. Finally, the developers both at Treasure Data and NTT proposed Hivemall to the ASF. Apache Hivemall can be used on widely used distributed processing frameworks such as Spark. Hivemall incorporates state-of-the-art distributed machine learning algorithms, which is different from existing machine learning libraries.

The Spark community is growing rapidly, and over 3000 participants from all over the world attended the recent Spark Summit held in June 2017. NTT is also involved in the development and is committed to the community. For example, it has enhanced APIs for porting Hivemall functions into Spark and has made a number of bug fixes and improvements in Catalyst. Catalyst is an optimizer component that affects many applications running on Spark, and it will likely play a significant role in NTT Group's use cases in the future. Therefore, we have stepped up efforts to fix bugs and enhance the performance of Catalyst. The

*4 Erasure coding scheme: A reliable data protection scheme implemented in Swift that reduces the amount of data stored on physical disks. Original data are segmented into fragments and encoded with redundant data pieces and stored across a set of different locations or storage media.

*5 DR cluster: A disaster recovery cluster is a system configuration in which hardware is deployed at multiple geographically dispersed sites, so if one (or several) datacenters fail due to a natural disaster or other unforeseen event, the overall system remains up and running.

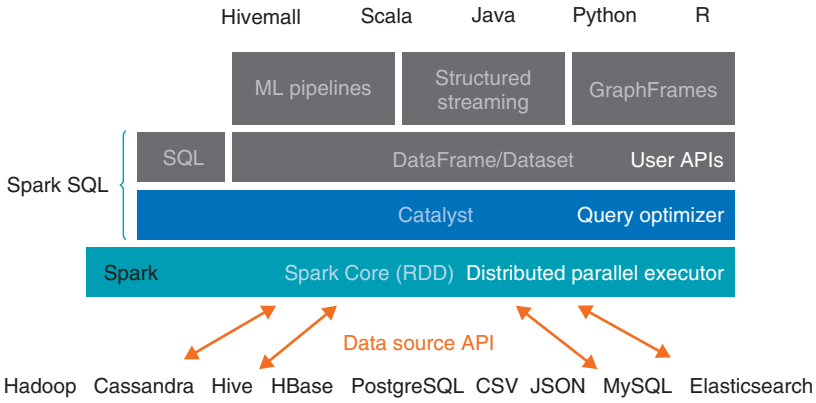


Fig. 3. Spark and Hivemall software stack.

Spark community recently took a step forward to support the use of graphics processing units (GPUs) in a resource scheduler. Spark currently only uses central processing unit (CPU) cores as resource scheduling units; a job scheduler splits a user query into tasks and assigns them to CPU cores. With the increasing popularity of deep learning and its need for computing resources, it is important to officially discuss GPU support in Spark native components.

The development of Apache Hivemall is progressing as Makoto Yui and other developers continue to implement a range of sophisticated machine learning algorithms and utility functions for feature engineering. In these activities, NTT is responsible for porting Hivemall functions into Spark and implementing useful and efficient functionalities that users actively request and Spark does not provide. Some of these functionalities are not implemented by Spark because they do not comply with the Spark development policy or are too complicated to implement as first-class Spark functionalities. With the valuable experience we have obtained through these OSS activities, we are currently trying to provide a scalable and highly efficient distributed parallel framework with Spark and Hivemall.

We believe that being involved in OSS communities such as Spark and competing with distinguished developers in the world will not only help us keep up with existing OSS products but will also open the way to discover innovative next-generation technologies that will emerge in future. We are now moving forward to apply Spark and Hivemall into realistic use cases with NTT Group companies. We plan to develop robust technologies that are widely used by NTT Group companies while building complementary relationships with OSS communities.

4. Future work

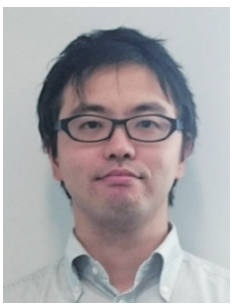
This article highlighted two major OSS projects that illustrate NTT's deep involvement in a range of worldwide efforts through its active participation in various OSS communities. OSS development is growing not simply because companies are eager to adopt free software for their own private purposes, but rather because involvement in such communities provides a way for companies to achieve success and grow with the communities. The NTT labs are leveraging open source through an upstream first approach in OpenStack, Apache Spark, Apache Hivemall, and other OSS projects. We are committed to ongoing R&D that contributes to NTT Group companies while benefiting the OSS communities to which we belong.

References

[1] R. Bryant, "OpenStack: Upstream First," Aug. 2016. https://www.slideshare.net/tesoracorp/openstack-upstream-first
[2] CVE, https://cve.mitre.org/
[3] OpenStack, https://www.openstack.org/
[4] Swift, https://www.openstack.org/software/releases/ocata/components/swift
[5] Press release issued by NTT DATA on January 15, 2015 (in Japanese). http://www.nttdata.com/jp/ja/news/release/2015/011500.html
[6] K. Tsuyuzaki, Bug report, OpenStack Object Storage (swift), Bug #1639691, Nov. 2016. https://bugs.launchpad.net/swift/+bug/1639691
[7] Corrupted fragment report, https://github.com/01org/isa-l/issues/10
[8] Press release issued by OpenStack on August 30, 2017, "OpenStack Pike Delivers Composable Infrastructure Services and Improved Lifecycle Management." https://www.openstack.org/news/view/340/openstack-pike-delivers-composable-infrastructure-services-and-improved-lifecycle-management

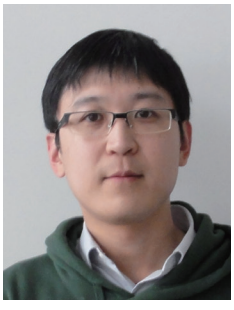
[9] Apache Spark, https://spark.apache.org
[10] Apache Hivemall, https://hivemall.incubator.apache.org

Trademark notes



Kota Tsuyuzaki
Software Engineer, Cloud Solution Project, NTT Software Innovation Center.

He received an M.E. in information engineering from Waseda University, Tokyo, in 2010. Since joining NTT in 2010, he has been developing distributed storage systems. He has been working on OpenStack Swift for approximately five years and has been a member of the Swift Core Team since June 2015.



Takeshi Yamamuro
Research Engineer, Distributed Computing Technology Project, NTT Software Innovation Center.

He received an M.E. from the Faculty of Science and Technology, Sophia University, Tokyo, in 2008 and joined NTT the same year. He has since been working on database management systems. His research interests include compression and hardware-aware algorithms (e.g., SIMD, NUMA and GPU). He received the IPSJ Yamashita SIG Research Award from the Information Processing Society of Japan (IPSJ) in 2015. He is a member of IPSJ and the Database Society of Japan.

Open Source Software Initiatives Supporting NTT Group Software Development and Operations

Takehito Abe, Yoshiharu Suga, Gengo Suzuki, Akira Kanamaru, and Takuya Iwatsuka

Abstract

The operation of system integrator and service provider businesses conducted by NTT Group companies requires particularly high performance and quality SoR (systems of record) information technology (IT) systems, and these systems must also be maintained and operated stably and efficiently over long periods of time. This article introduces some initiatives in community development to improve the quality, functionality, and performance of open source software (OSS) and to accumulate and develop know-how for promoting application of OSS at the NTT Open Source Software Center. It also reviews initiatives involving the Macchinetta framework to improve the productivity and quality of software development and to eliminate duplicated investment at the NTT Software Innovation Center, with the goal of realizing an environment in which OSS can be used with confidence throughout the life cycle of IT systems.

Keywords: OSS total support, Macchinetta framework, OSSVERT

1. OSS Technology Initiative: PostgreSQL

The NTT Open Source Software Center (NTT OSS Center) is collaborating with NTT Group companies and the open source software (OSS) community, conducting technical development, and accumulating know-how regarding various OSS, from operating systems (OS) to middleware, so that software development, maintenance, and operation of information technology (IT) systems centered on OSS can be performed stably and efficiently in the businesses being operated by the Group companies. Database management systems (DBMSs) hold a very important position in information and communication systems, and the NTT OSS Center has put great effort into expanding application of PostgreSQL* since its inception. The objectives have included enhancing performance and functionality as database sizes have increased, and facilitating migration from commercial DBMSs. There have also been initiatives to develop large-scale databases using external table functionality, to conduct research and development (R&D) on distributed databases, and to improve compatibility so that Structured Query Language (SQL) dialects defined by commercial DBMSs can run unmodified on PostgreSQL (**Fig. 1**).

The latest version, PostgreSQL 10, introduces declarative table partitioning, which was developed under the leadership of members of the NTT OSS Center. Partitioning is a function whereby performance or operability is improved by dividing and storing data in multiple tables, and it is particularly useful when handling very large datasets. In previous versions, PostgreSQL did not provide a dedicated partitioning function, and *pseudo-partitioning* had to be implemented by combining table inheritance functions, triggers, and CHECK constraints. With the introduction of a dedicated partitioning function,

* PostgreSQL: An OSS relational database management system.

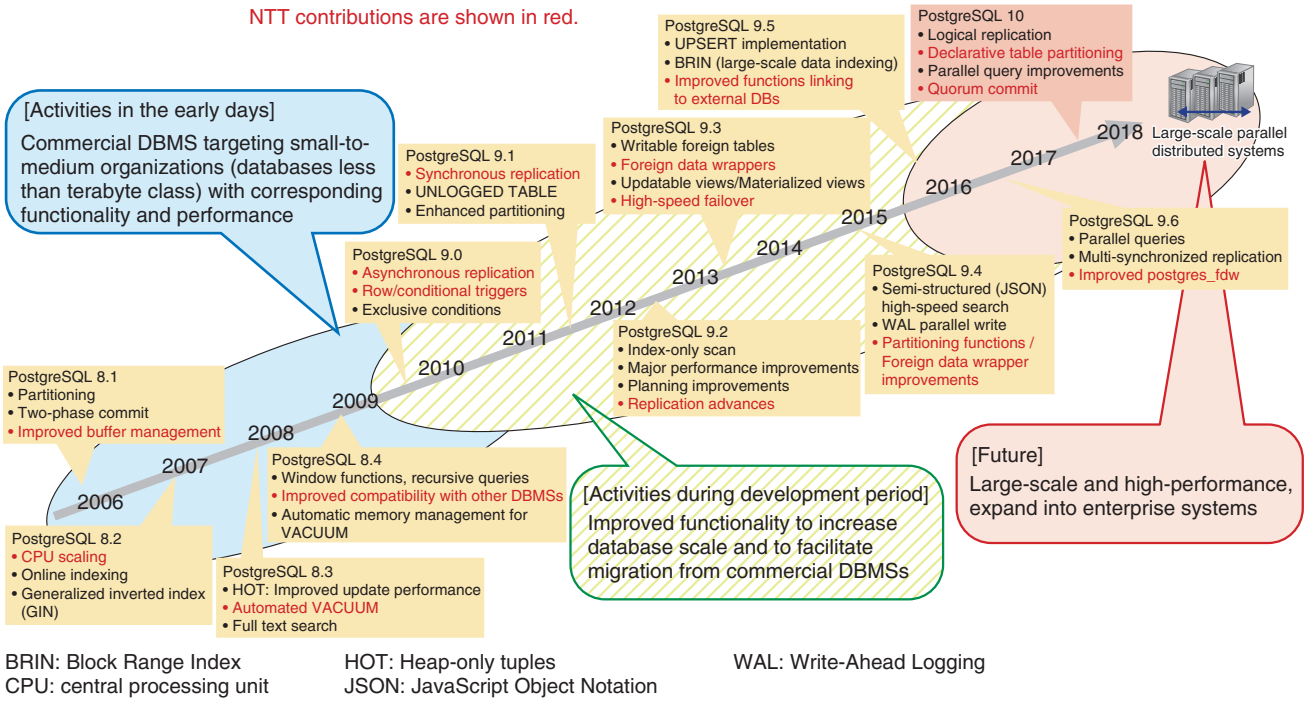


Fig. 1. Contributions to PostgreSQL development.

partitioning can be applied by simply declaring a partitioning structure in SQL when creating the table (CREATE TABLE statement). Data insertion is faster that it was with the earlier pseudo-partitioning practice.

To further expand PostgreSQL into enterprise systems, the NTT OSS Center is working to implement *sharding*, which will enable table partitions to be distributed over multiple servers to distribute load and improve performance. The new partitioning and foreign data wrapper (FDW) function are necessary as basic functions of sharding. The NTT OSS Center has also led the enhancements of FDW, which enables tables on a remote server to be accessed as though they were local tables. Some functionality is still not adequate for implementing sharding, for example, support for distributed transactions, but the NTT OSS Center will continue development in collaboration with the community and will work to expand PostgreSQL into a database that can be scaled out to larger sizes.

Note that in addition to partitioning, PostgreSQL 10 incorporates various other new functionalities and improvements, including a new logical replication function that enables replication by table, enhancements to parallel queries so they can be used in more cases, and a new quorum commit functionality developed under the leadership of those at the NTT OSS Center.

2. Technically verified OSS suite: OSSVERT

Since its inception, the NTT OSS Center has been operating its OSS Suites Verified Technically (OSSVERT) program. In OSSVERT, the NTT OSS Center verifies the operation of systems composed of OSS that is recommended and supported by the Center and provides technical materials summarizing the results. Technical materials provided include model overviews, installation procedures, environment definitions, and verification reports. Use of the recommended middleware configurations and various settings verified and provided by OSSVERT facilitates rapid and efficient system design and construction. The main OSS in OSSVERT configurations is maintained and supported by the NTT OSS Center, so safe and secure operational support is available.

In the past, OSSVERT initiatives have dealt with verifying the construction of three-tier web systems on physical servers, but with recent trends to expand the use of OSS, the NTT OSS Center has begun using OSSVERT to verify other types of OSS that are

expected to be used more widely in the future. Specific initiatives include verifying the OpenStack (RDO Liberty) run-time environment, cloud environments utilizing Amazon Web Services (AWS), and three-tier web systems built on Docker.

Beyond simply verifying operation, the NTT OSS Center has also been using OSSVERT to test new combinations of middleware in order to evaluate their potential for service provision and to discover any constraints or issues as early as possible. Software is checked for any problems particularly from the perspective of availability and performance under high loads, so that any necessary cautions or considerations when designing and building systems can be clearly identified, and any defects are discovered, investigated, and analyzed. This enables solutions or workarounds to be found quickly.

A strength of OSSVERT is that it provides information that can be used immediately based on testing. A certain amount of information can be obtained from technology books and the Internet, but it is very difficult to find information on how to appropriately operate various enterprise functionalities using the latest configurations. OSSVERT provides reference information suitable for enterprise use based on practical testing.

3. Use of OSS for enterprise application development: Macchinetta framework

It is becoming very common to use platform software called an application framework (AP framework) when developing applications for various types of business operations (business APs). An AP framework is software that mediates between the business AP and the software operating environment composed of hardware, OS, and middleware. An AP framework can increase productivity by providing various common functions, and it has other benefits such as promoting more uniform quality since the framework has conventions that reduce differences among programmers developing the application, and more efficient maintenance and management, since common knowledge based on the framework makes systems easier to understand.

The NTT Software Innovation Center has developed the Macchinetta framework as a new AP framework that can be used throughout the NTT Group [1]. This will improve efficiency for personnel training, software development, and maintenance, and it also has the benefit of eliminating duplicate investment since it is used throughout the Group. The basic policies in developing this framework were to use OSS of a global standard, to ensure it was applicable on a wide range of IT systems, and not to be constrained by proprietary technology.

The Macchinetta framework is composed of a recommended OSS stack model, which is a stack of OSS products needed for common functionality, and technical documentation to facilitate development using the AP framework (**Fig. 2**). The recommended OSS stack model has been tested in multiple combinations of OSS products so that they can be stably used in such combinations. In addition to components for developing web systems with popular architectures, it also includes batch processing architectures necessary for business APs, and client-side components for improving operability. It has been adopted as the standard AP framework for NTT Group companies doing systems development, such as NTT DATA and NTT Comware. In principle, it will be used for all new projects and has a large and growing record of practical applications.

To provide even higher quality support and increase the reliability of the AP framework, we continue to participate actively in improving OSS and are conducting R&D in cooperation with the OSS community. To implement major functionalities such as web application server-side processing and batch processing in the Macchinetta framework, we are using an OSS set centered around the Spring Framework [2], which is developed by the Spring community. We are providing bug patches to improve the quality of this OSS and proposing new functionality based on needs within the NTT Group. In particular, we are contributing by working on vulnerabilities that could lead to security risks. These are investigated and analyzed in collaboration with NTT Group companies, and bug-fix proposals are created in communication with principal developers in the community.

An active OSS community is also essential to ensure that the OSS we are using has many users and continues to be developed and maintained for a long time. We are actively contributing to invigorating the community such as by making presentations explaining the latest technologies at events sponsored by the community in Japan and around the world, and by promoting technical and event information on the web.

4. Future plans

The NTT OSS Center and NTT Software Innovation Center endeavor to increase the usability of OSS

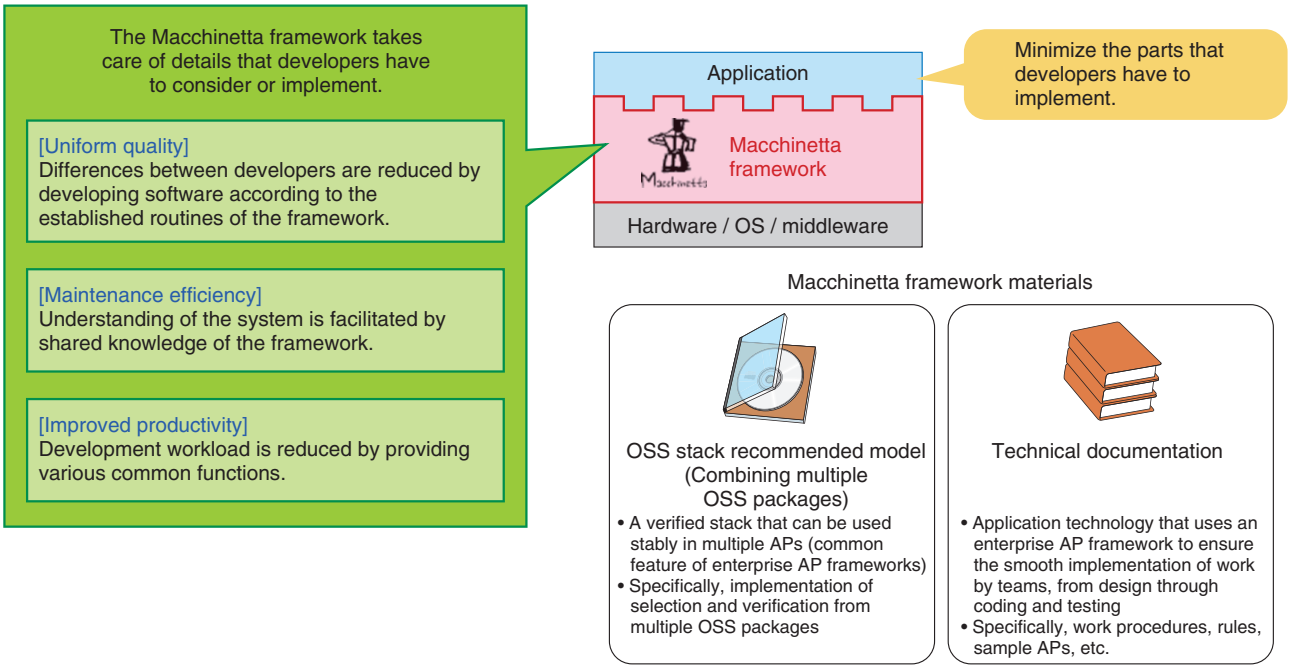


Fig. 2. Overview of Macchinetta framework.

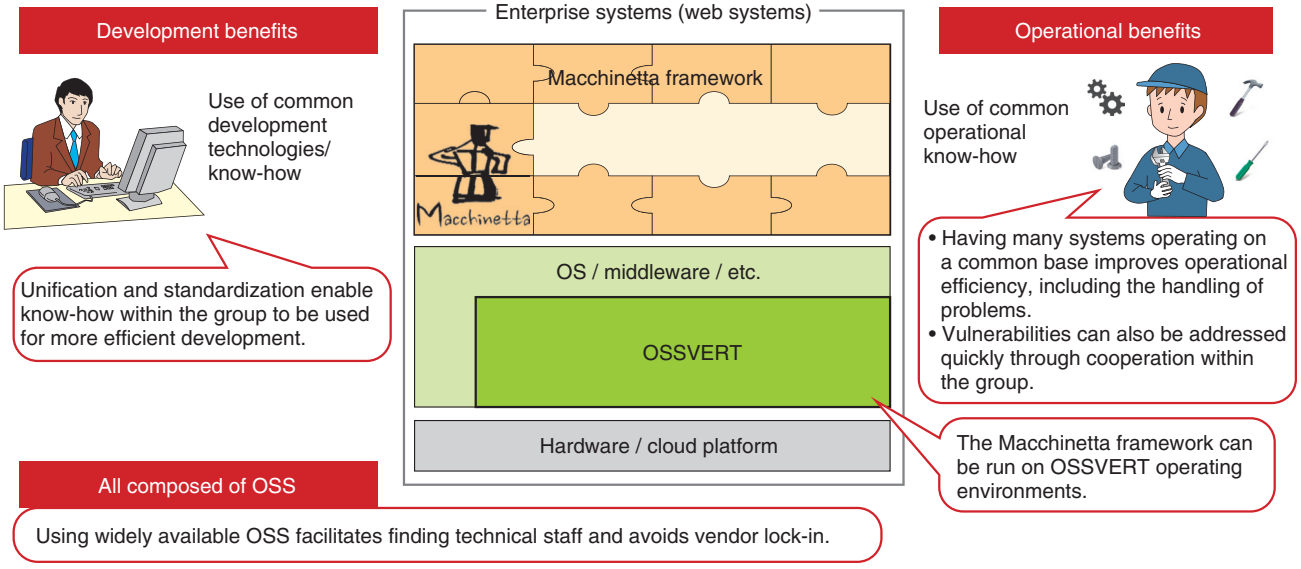


Fig. 3. Software development and operation with OSSVERT and the Macchinetta framework.

by testing coordinated operation of OSSVERT and the Macchinetta framework as described above (**Fig. 3**). We are also providing the necessary technologies and tools for software development and operation, supplying know-how, and supporting problem solving so that system updates for IT system security and modernization can be done and the OSS can be maintained and operated stably and efficiently over long periods of time. Through these endeavors, we are promoting the use of IT in NTT Group

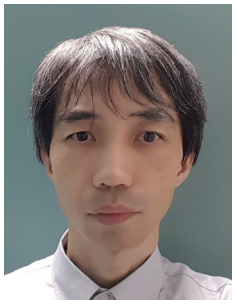
companies and helping to reduce their capital expenditure and operating expenses.

References

[1] G. Suzuki, A. Kanamaru, T. Iwatsuka, J. Katada, S. Okada, S. Mochida, K. Natsukawa, K. Motohashi, T. Hishiki, T. Kaneko, K. Tanabe, H. Izumoto, M. Sakai, K. Yamashita, and Y. Iwaki, "Improving the Efficiency of Application Development Based on the Macchinetta Framework," NTT Technical Review, Vol. 15, No. 2, 2017. https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201702fa2.html

[2] NTT DATA, "Complete Introduction to Spring: Java Application Development Using the Spring Framework," Shoeisha, 2016 (in Japanese).

Trademark notes



Takehito Abe
Senior Research Engineer, Open Source Software SE Project (NTT OSS Center), NTT Software Innovation Center.
He received a B.S. and M.E. in materials science and engineering from Waseda University, Tokyo, in 1993 and 1995, and a Ph.D. in informatics from the Graduate University for Advanced Studies, Kanagawa, in 2009. He joined NTT in 1995. His research interests include open source software, distributed computing, and media processing. He is a member of the Information Processing Society of Japan (IPSJ).

Yoshiharu Suga
Senior Research Engineer, Open Source Software SE Project (NTT OSS Center), NTT Software Innovation Center.
He received a B.S. and M.E. in computer science from Tokyo Institute of Technology in 1998 and 2000. He joined NTT in 2000 and studied multimedia information retrieval, an in-memory database system, and integration of distributed heterogeneous information sources. His research interests include database management systems and infrastructure software technology. He is the manager of NTT OSS Center's DBMS team and a member of the Institute of Electronics, Information and Communication Engineers (IEICE).

Gengo Suzuki
Senior Research Engineer, Software Engineering Project, NTT Software Innovation Center.
He received a B.S. and M.S. in mathematics from Tohoku University, Miyagi, in 1988 and 1990, and a Ph.D. in computer science from the University of Electro-Communications, Tokyo, in 2013. He joined NTT in 1990. His research interests include database design, multi-database systems, graph data management, and software engineering. He is a member of IPSJ and IEICE.

Akira Kanamaru
Senior Research Engineer, Supervisor, Software Engineering Project, NTT Software Innovation Center.
He received a B.E., M.E., and Ph.D. from Tohoku University, Miyagi, in 1996, 1998, and 2001. He joined NTT in 2001. His research interests include computer networks and software engineering.

Takuya Iwatsuka
Researcher, Software Engineering Project, NTT Software Innovation Center.
He received a B.E. and M.E. in computer science from Nagoya University, Aichi, in 2011 and 2013. He joined NTT in 2013. His interests include programming language theory, program verification, and software engineering.

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 Cloud^{n} [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.

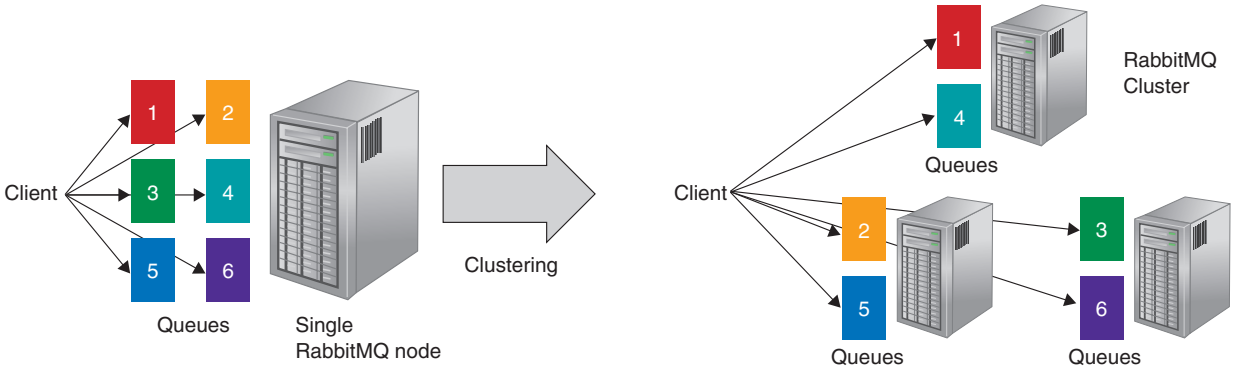


Fig. 1. Queue distribution in RabbitMQ cluster.

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

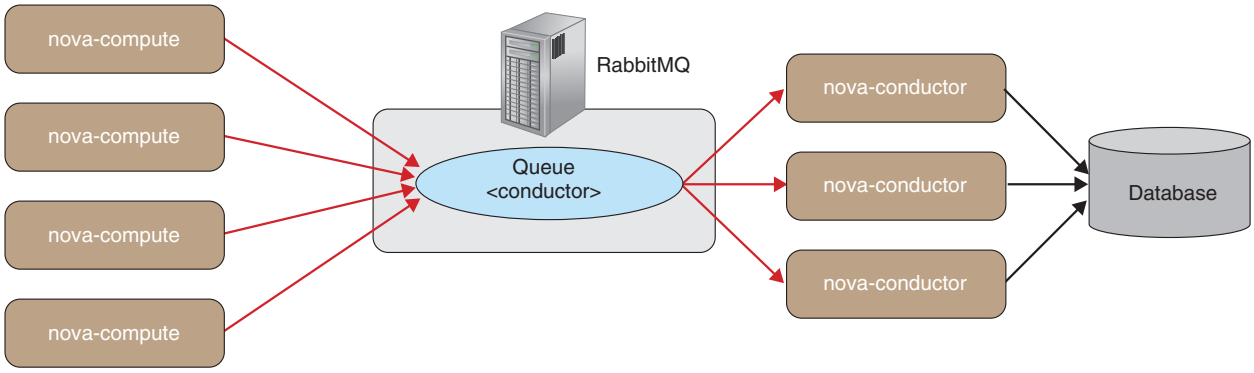
2.1 Scale-out RabbitMQ Cluster Can Improve Performance While Keeping High Availability

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.

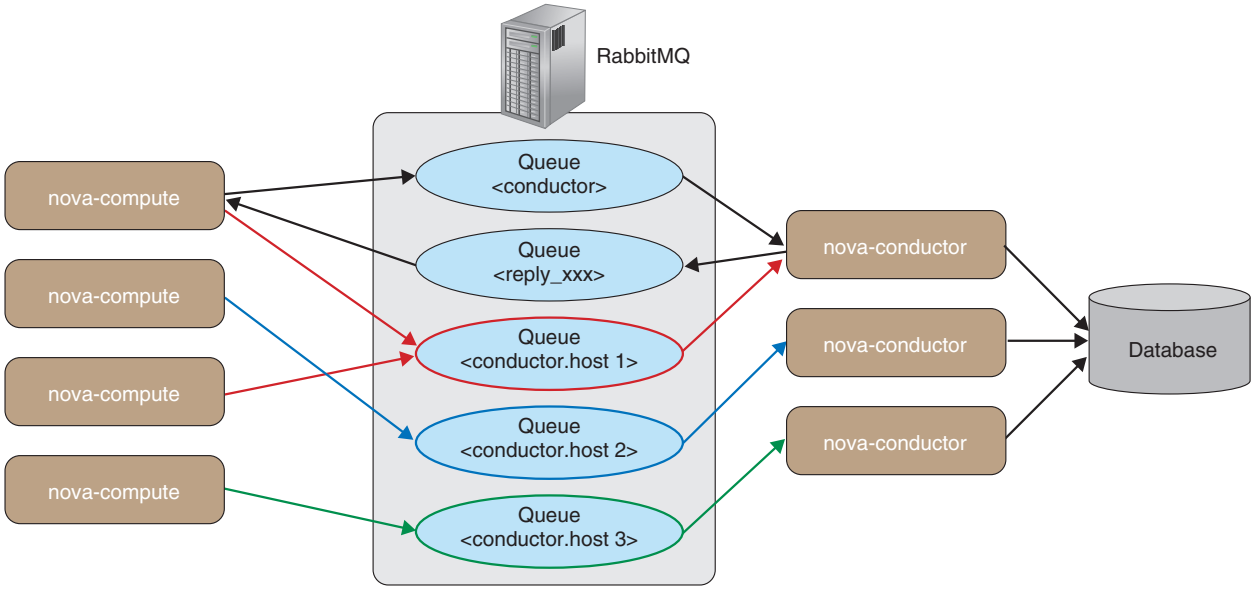
NTT Com, meanwhile, is planning to expand the scale of Enterprise Cloud 2.0, so it is therefore necessary to improve the performance and operation of MQ. In Boston, we teamed up with the NTT Software Innovation Center (NTT SIC) to present methods for improving MQ operation and OpenStack internal operation. 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

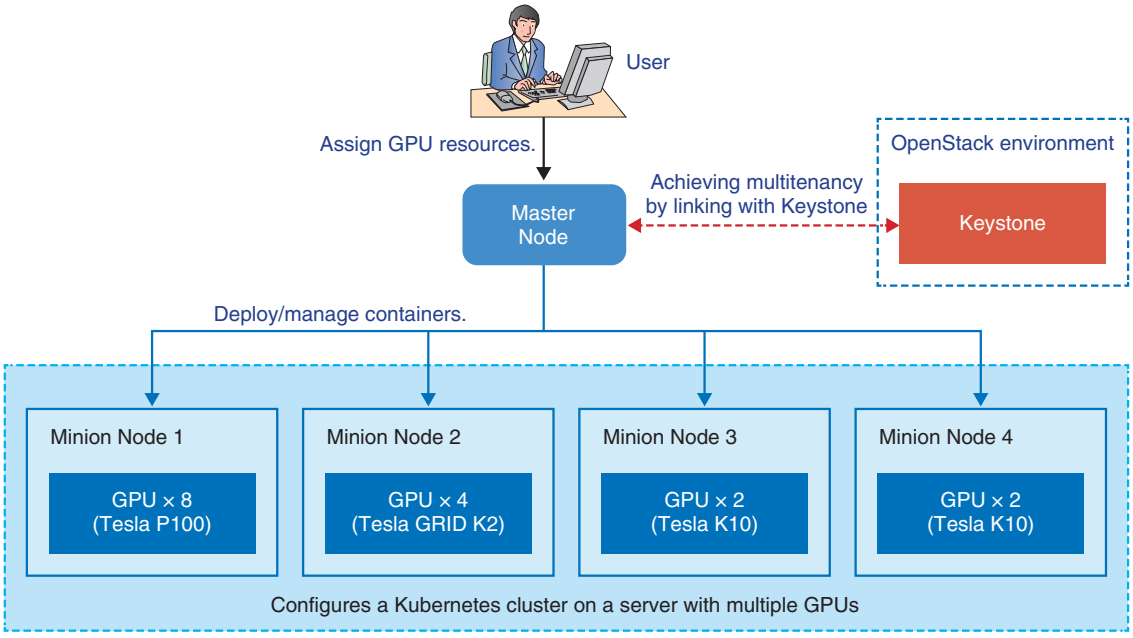


Fig. 3. Conceptual diagram of test environment.

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

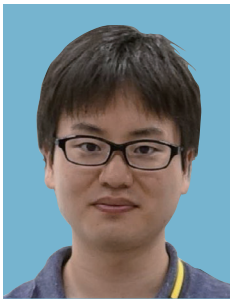
*5 Cloud-native: Systems and services designed assuming use on the cloud.

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.

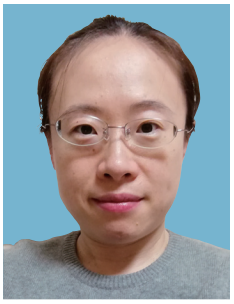
References

[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/eclc/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/

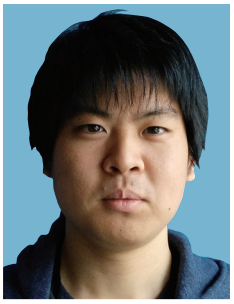
Trademark notes



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.



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).



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.

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.

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.

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.

Global Expansion of Apache Hadoop/ Apache Spark Activities at NTT DATA

Ravindra Sandaruwan Ranaweera and Akira Ajisaka

Abstract

Apache Hadoop and Apache Spark are open source software projects that have been attracting worldwide attention as platforms for big data applications. NTT DATA has been developing systems using Hadoop/Spark since Hadoop/Spark's early days and providing one-stop technical support for the entire NTT DATA Group. Furthermore, in addition to participating in community-based development to reflect the feedback gained from technical support service into Hadoop/Spark itself, NTT DATA has been presenting the know-how accumulated from these activities at numerous events to raise its global presence.

Keywords: big data, parallel distributed processing, global

1. Hadoop/Spark

An increasing number of companies are attempting to create new businesses or expand existing businesses based on the results of analyzing huge amounts of various types of data known as big data. Apache Hadoop was introduced in 2006 and enables massive amounts of diverse data to be stored, processed, and analyzed in a realistic period of time at a reasonable cost. Apache Spark, meanwhile, was developed by postgraduate students at the University of California, Berkeley to efficiently perform repetitive and complex processing of big data, which Hadoop is not good at doing.

At present, Spark is becoming increasingly popular thanks to a number of key features. For example, it supports programming languages such as Python and R widely used in the data analysis industry, incorporates a machine-learning library, and performs various types of optimization.

2. Hadoop/Spark activities at NTT DATA

NTT DATA began using Hadoop in 2008 before Hadoop began attracting attention worldwide. At that time, Hadoop was still immature, and there were no functions satisfying the strict requirements pertaining to availability and operability demanded by enterprise customers. The NTT DATA team working on Hadoop, however, felt that Hadoop could bring significant benefits to enterprise customers. With this in mind, the team conducted tests on applying NTT DATA's know-how in system integration in order to meet such strict availability/operability requirements, and published that know-how in the form of a test report released in 2010 [1].

This report helped to make NTT DATA's work on Hadoop well known, and since then, NTT DATA has gone on to construct and provide systems using Hadoop/Spark in all sorts of industries including telecommunications, real estate, public administration, finance, media, and manufacturing. NTT DATA's involvement is not limited to just a portion of system development, but rather, it provides a wide range of services covering the entire system development process including planning, design, development, and support to provide its customers with systems that can lead to new business opportunities (**Fig. 1**). Providing versatile services in this way enables NTT DATA to satisfy the genuine needs of its customers.

Furthermore, to enhance services and raise the level of customer satisfaction, NTT DATA proactively

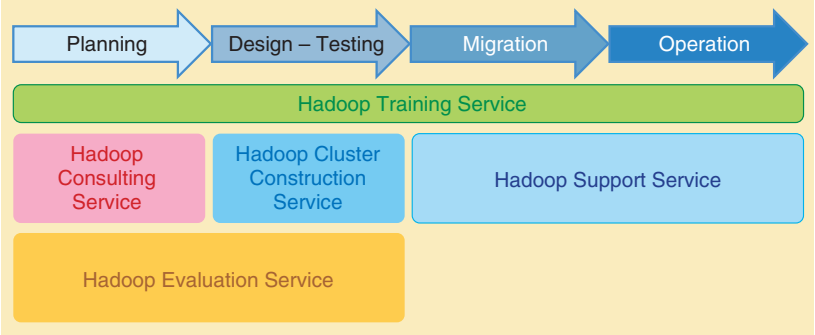


Fig. 1. NTT DATA Hadoop/Spark service menu.

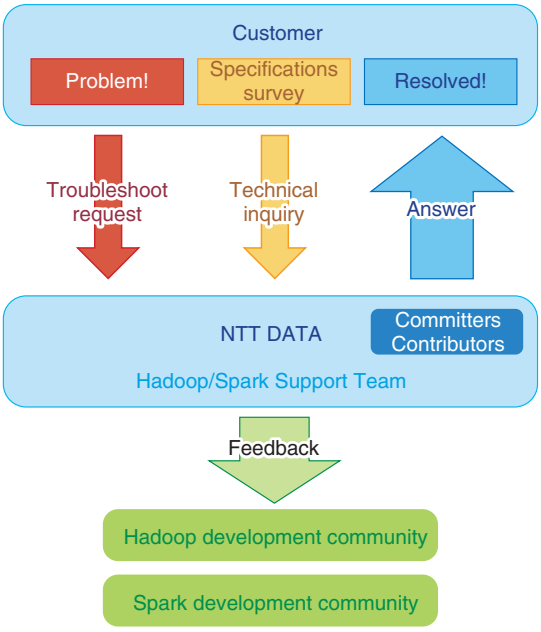


Fig. 2. Knowledge gained from Hadoop/Spark system development is fed back to the development community.

feeds back the knowledge it has gained from Hadoop/Spark system development to the development community (**Fig. 2**).

For example, if some type of problem should occur in system development due to a software bug, NTT DATA would interact with the community to fix that bug. In this way, there is no need for NTT DATA to install and manage its own software patches. For the customer, meanwhile, updating to a new version provides a fundamental solution to the problem while preventing its software from becoming detached from the software updates provided by the community.

NTT DATA has also developed a variety of Hadoop/Spark functions and merged them with the community. For example, in Spark, to simplify performance tuning and debugging, NTT DATA guided the development of the Timeline Viewer tool for visualizing which process has run or is running on which server in chronological order. A screenshot of the Timeline Viewer function whose development was headed by NTT DATA is shown in **Fig. 3**. This type of activity was carried out through discussions and collaborations with developers from around the world.

The Hadoop/Spark development community recognized the value of this work and elected several of these NTT DATA developers to the position of

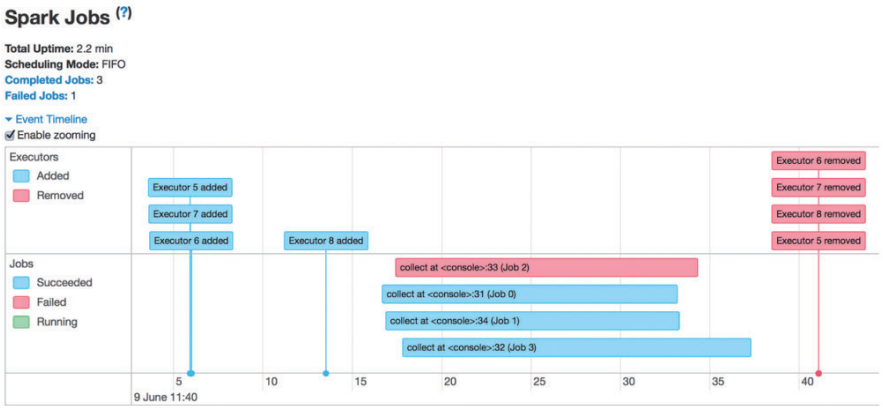


Fig. 3. Screenshot of Spark Timeline Viewer developed under NTT DATA leadership.

Photo 1. Hadoop/Spark Enterprise Solutions Seminar held by NTT DATA.

committer (a developer having the right to modify Hadoop/Spark source code), thereby making these developers the first Hadoop/Spark committers from a Japanese company. They continued with their Hadoop/Spark development activities even after becoming committers, resulting in their being promoted to members of the Project Management Committee that manages Hadoop projects.

In Japan, there are many companies that, while possessing large volumes of data, are not making use of that data. These companies who would like to analyze the large amounts of data in an efficient manner to generate new business value have come to NTT DATA asking for support. In response to these needs, NTT DATA holds the annual Hadoop/Spark Enterprise Solutions Seminar [2] to explain how enterprises can use Hadoop/Spark to create new value (**Photo 1**). This seminar introduces advanced use cases driven mostly by NTT DATA in a variety of industries within Japan as well as Hadoop/Spark use cases driven by NTT DATA Group companies outside Japan. It presents the benefits that Hadoop/Spark can bring to enterprises in an easy-to-understand manner and attracts many individuals from NTT DATA enterprise customers. In addition to the above, NTT DATA regularly holds a variety of seminars and meetups within Japan to introduce know-how of a more technical nature and to promote its technical expertise in Hadoop/Spark.

3. NTT DATA's Hadoop/Spark related global activities

NTT DATA does not share its knowledge and know-how only at seminars and events within Japan. Since it pursues system development using advanced technologies, it has been involved in a few world-first use cases applying those technologies. NTT DATA

Photo 2. Presentation by the NTT DATA Hadoop/Spark team in the USA.

actively participates in seminars outside Japan to introduce these cutting-edge use cases and to promote the company's technical expertise. To date, it has given presentations at a variety of conferences including ApacheCon, Apache: Big Data, Dataworks Summit (formerly Hadoop Summit), Strata Data Conference (formerly Strata + Hadoop World), Spark Summit, Kafka Summit, and Global Big Data Conference, and it plans to continue doing so [2]. Scenes from NTT DATA's Hadoop/Spark team giving presentations in the United States in 2017 are shown in **Photo 2**. No other Japanese company regularly presents such advanced results at global Hadoop/Spark-related events, and in this capacity, NTT DATA acts as a representative of Japan.

Although Hadoop/Spark-related technologies are gaining in popularity, the pool of Hadoop/Spark technical personnel around the world is not growing. The reasons given for this include differences in past technologies and in ways of thinking, and the need for knowledge covering a wide range of technologies involving hardware, operating systems, databases, networks, and distributed processing. It is an unfortunate fact that there are relatively few Hadoop/Spark developers within the NTT DATA Group despite being a company whose employee numbers have grown to 110,000 worldwide through strategic acquisition of firms around the world. Under these conditions, non-Japanese NTT DATA Group companies often request support from the Japan-side Hadoop/Spark team for system development involving big data analysis.

For example, requests are being received from the United States, Europe, and Asia for help in creating and reviewing proposals, designing system architectures, developing applications, and performance tuning. End users (customers) have also recognized the technical expertise of the Japan-side Hadoop/Spark team that responds to such requests. In particular, Hadoop/Spark technical personnel in Japan have received praise from global customers for their ability to support an entire system, in contrast to the trend outside Japan to focus only on one specific area.

4. Future development

At NTT DATA, we wish to expand the use of Hadoop/Spark even further to help our customers create new value. Furthermore, in addition to Hadoop/Spark, we plan to incorporate the latest research results in open source middleware for distributed processing to improve its performance and scalability. We are presently holding discussions with researchers from the NTT laboratories on the latest research trends in distributed processing, and we hope to continue these discussions at an even deeper level going forward.

References

[1] NTT DATA, "Project Report on 2009 Industry-Academia Collaboration Project on Software Engineering Practice," Mar. 2010 (in Japanese). http://www.meti.go.jp/policy/mono_info_service/joho/downloadfiles/2010software_research/clou_dist_software.pdf

[2] Website of NTT DATA Hadoop/Spark Enterprise Solutions Seminar (in Japanese), http://oss.nttdata.com/hadoop/event.html

Trademark notes

Ravindra Sandaruwan Ranaweera
Software Engineer, OSS Professional Services, System Engineering Headquarters, Platform Engineering Department, NTT DATA Corporation.

He received his B.E. and M.E. in network optimization from the University of Electro-Communications, Tokyo, in 2012 and 2014. He joined NTT DATA in 2014 and has since been working as a platform engineer/consultant on distributed systems using Apache Hadoop/Spark.

Akira Ajisaka
Software Engineer, OSS Professional Services, System Engineering Headquarters, Platform Engineering Department, NTT DATA Corporation.

He received his B.E. in engineering and his M.E. in applied mathematics and physics from Kyoto University in 2009 and 2011. He joined NTT DATA in 2011 and has since been working on distributed systems using Apache Hadoop. He is an Apache Hadoop committer and Project Management Committee member.

Achieving Greater Work Efficiency in Systems Failure Analysis Using Elastic Stack

Nao Maeda, Norito Agetsuma, Kenichi Kamimura, Yasumasa Suenaga, Shinya Takebayashi, and Katsuyuki Yamashita

Abstract

The keys in analyzing serious system failures are collecting information and logs in various formats, extracting the necessary information from the collected logs, and analyzing the correlated items; however, such procedures are time-consuming. NTT Comware is working on reducing the work time and simplifying the procedure by using Elastic Stack. This article introduces the details of the method and some future scenarios.

Keywords: log analysis, Elastic Stack, metrics analysis

1. Strategic objective for improving the efficiency of system failure log analysis

The procedure for analyzing logs in the event of a serious system failure should include carrying out various analysis tasks on the operating system (OS), applications, and database, rather than simply logging records of applications of the system concerned (which also must be done). Such log records contain various kinds of information in their own formats. It is also necessary to extract the necessary information from massive amounts of data—from several thousand megabytes (several thousand lines) to tens of thousands of gigabytes (several tens of thousands of lines) of data—and correlate them for analysis in the log visualization/analysis process. In the conventional method, as shown in **Fig. 1**, these operations are conducted manually, including tabulation using a spreadsheet program, which requires additional man-hours.

This log visualization/analysis work accounted for 42% of the total system failure log analysis in a certain system failure handled by NTT Comware, as shown in **Fig. 2**. Therefore, achieving higher efficiency in this task is a major challenge for reducing the time from the occurrence of the system failure to recovery. Under these circumstances, we are currently introducing initiatives to reduce the overall work hours by achieving higher efficiency of log visualization work and analysis in the system failure log analysis process.

2. Approach to achieve highly efficient log visualization/analysis

We have been working on a way to increase work efficiency by using Elastic Stack, which we describe in this section.

2.1 Elastic Stack

Elastic Stack is a combination of the following open source software (OSS): Beats[*1], Logstash[*2],

*1 Beats: A data shipper to transfer data to Elasticsearch/Logstash. It is stored in servers and obtains resource data or log files that it transfers on a real-time basis. Beats includes Filebeat, Metricbeat, Packetbeat, Winlogbeat, and Heartbeat.

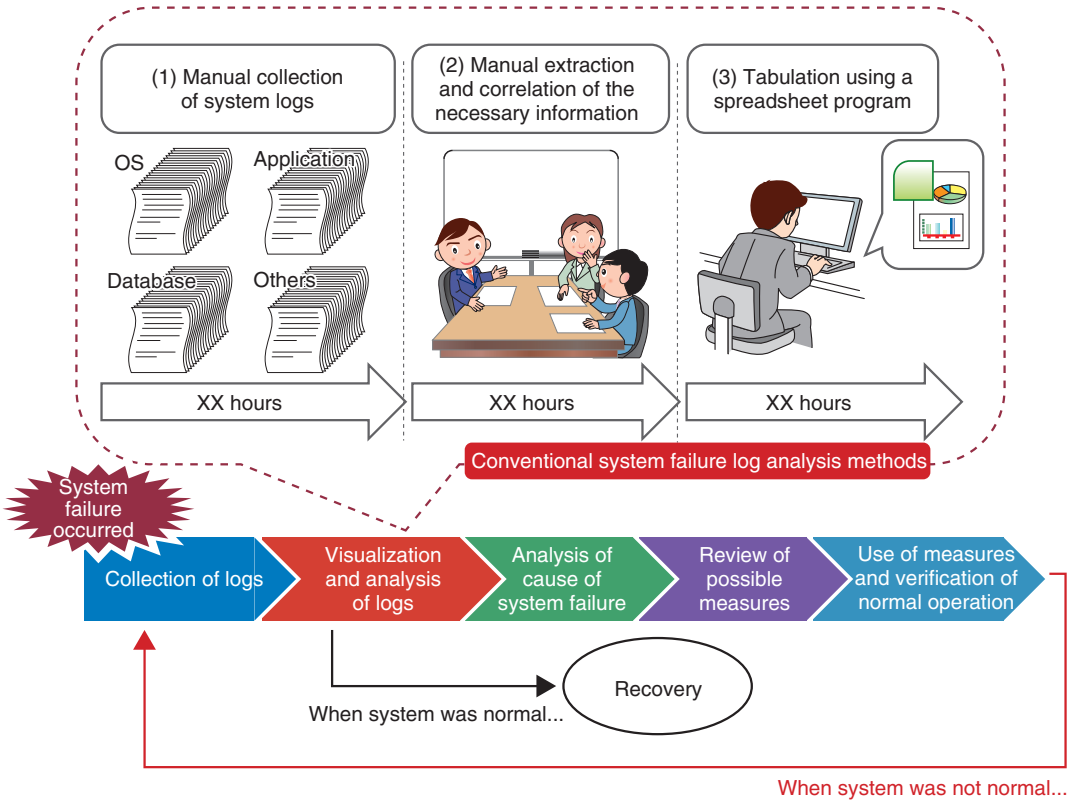


Fig. 1. Conventional procedure of system failure log analysis.

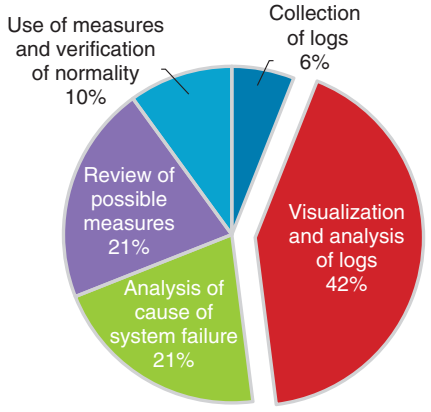


Fig. 2. Composition of recovery work in the event of a system failure.

Elasticsearch[*3], and Kibana[*4], the development of which is led by Elasticsearch Co. [1]. This is an example of a data analysis infrastructure that has been gaining attention recently for its use in the collection, accumulation, and/or visualization of big data. The general architecture of Elastic Stack is shown in **Fig. 3**.

Each tool in Elastic Stack has a function, for example, log collection (Beats), log analysis (Logstash), log accumulation (Elasticsearch), and visualization (Kibana). The combination of these tools enables advanced visualization of data in formats, including time series graphs, pie charts, and statistical information.

2.2 Advantages of Elastic Stack

Although many tools other than Elastic Stack have been released as infrastructure tools for collecting/accumulating/visualizing big data, we needed to

*2 Logstash: A log collection tool to obtain, analyze/convert, and output logs in an appropriate format.
*3 Elasticsearch: The full text search server in which the Apache Lucene search engine runs. This product is used to store/search/analyze massive amounts of data as well as to facilitate computing log records in a specific area.
*4 Kibana: A tool to visualize data input to Elasticsearch. This product facilitates advanced visualization of data such as time-lapse graphs, pie charts, and statistical information that was too complicated to achieve with existing graphics tools.

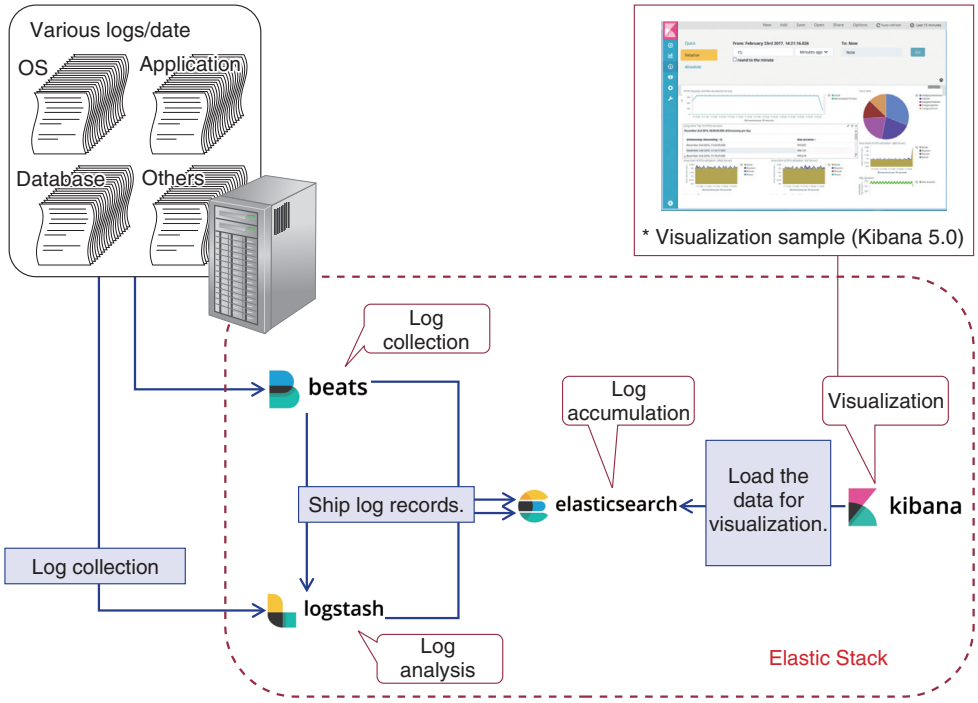


Fig. 3. Architecture of Elastic Stack.

select the optimum tool that would have the following required capabilities in this initiative:

(1) Collection and analysis of multiple logs in different formats
(2) Correlation of specific items among accumulated logs and simultaneous visualization of multiple logs
(3) Continuous log monitoring

Requirements (1) and (2) are essential since they are included in the necessary work to do in the process of log collection and analysis, as mentioned in section 1.

Requirement (3) is also defined as one of the requirements that need to be met in order to promptly cope with a system failure, since it is necessary to collect, analyze, and visualize the latest logs on a real-time basis.

As a result of a comparative study done on similar infrastructure tools, Elastic Stack was selected since it easily meets the above requirements in terms of functions. It also has an active development community. Furthermore, since Elastic Stack is OSS, there were no initial costs to use it such as for license fees. This was also one of the reasons it was selected.

2.3 Efficient log visualization/analysis method

NTT Comware has documented a system failure log analysis procedure that explains how to achieve efficient log visualization and analysis. Failure analysis engineers, application developers, and operations managers can use this document to visualize OS/application/database logs using Elastic Stack and to reduce the working time to determine a suspected cause of failure.

The document has three sections as follows.

(1) How to Install Beats, Logstash, Elasticsearch, and Kibana

It is necessary to construct a system environment that facilitates use of the method, such as by installing each tool in Elastic Stack. This section explains how to install each tool to simplify this work process.

(2) Collection and Visualization/Analysis Procedure for Various Logs

This section summarizes the procedure to collect OS/application/database logs using Beats and Logstash, store them in Elasticsearch, and visualize them using Kibana (**Fig. 4**). The settings described in this document are based on OSSVERT™ (OSS Suites VERified Technically)[*5] provided by the NTT OSS

*5 OSSVERT: Know-how on the use of OSS provided by the NTT OSS Center.

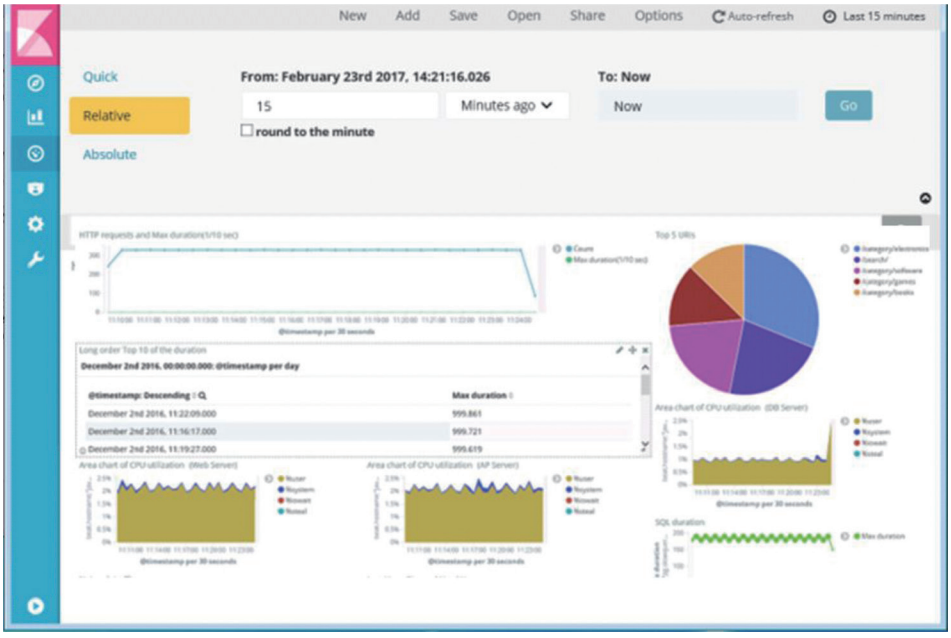


Fig. 4. Visualization example using Kibana.

Center. The settings can be used as they are in systems that have settings that conform to OSSVERT™.

(3) How to Back Up and Restore Data Stored in Elasticsearch

When the system environment is constructed based on the above two steps, and log visualization/analysis continues, it is important to avoid reducing the storage capacity with the accumulated logs and to prepare the procedure to restore the log records in case of a malfunction of the disk that stores the logs. This section describes how to back up and restore the log records stored in Elasticsearch.

2.4 Results of efficient system failure log analysis method using Elastic Stack

We are implementing the abovementioned system failure log analysis procedure at NTT Comware and promoting such analysis methods using Elastic Stack having the same architecture shown in Fig. 3.

The results of the implementation revealed that the cost of the complicated system failure log analysis was reduced to about one-twentieth (1/20) on average by using the method.

The following highlights a successful case in which the amount of time spent analyzing system failure logs was reduced. In this example, system failures were occurring that frequently showed high central processing unit (CPU) usage. The method was applied to analyze the system failure logs, and it was found that the surge in the CPU usage rate was due to unnecessary UNIX commands input by the operator. With conventional system failure log analysis methods, it would typically take ten working days to confirm normality because of the large size of the log records, often many gigabytes. In contrast, this method contributed to quick troubleshooting, since the time to complete the whole process was shortened to about four hours.

2.5 Additional knowledge

NTT Comware provides other knowledge necessary for applying Elastic Stack at the company, in addition to the abovementioned system failure log analysis procedure. The key points are as follows.

(1) Elastic Stack Sizing Test Report/Elastic Stack Sizing Guide

It is essential to carefully consider the size and design of the system in order to apply the system failure log analysis method using Elastic Stack with commercial systems handling massive amounts of log records or in systems that are significantly larger than usual. Therefore, NTT Comware tested the sizing of the system that uses Elastic Stack and documented the results in the Elastic Stack Sizing Test Report/Elastic Stack Sizing Guide. This document summarizes the guidelines of the hardware environment

to procure when designing the log analysis mechanism such as the basic performance and the scalability of Elastic Stack.

(2) Elastic Stack/Splunk Comparison Survey Report

When Elastic Stack is proposed for use, it may be a requirement in some cases to first conduct a comparative survey and obtain results on other similar tools, including commercial products. Since a comparative survey report on Splunk*6 [2] is often requested when making proposals (to our customers), we provide to them the Elastic Stack/Splunk Comparative Survey Report, which summarizes the comparative results in terms of functions (i.e., ease of learning, adaptability, and expandability) and performance of Splunk and Elastic Stack. The documentation makes it possible to further understand the characteristics of Elastic Stack and Splunk and to easily select appropriate products for customers.

3. Application support experience and future plans

Elastic Stack can be used for visualization and analysis of not only system failure logs but also regular task logs in commercial systems by applying the method/knowledge of this initiative. NTT Comware has been involved in 22 Elastic Stack deployment projects up to August 2017 and is promoting optimal system development utilizing advanced OSS-based technologies.

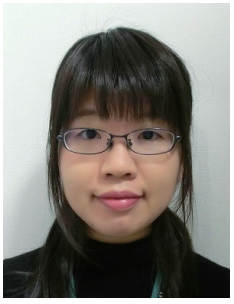
We are continuing to accumulate product knowledge on Elastic Stack since version upgrades and new function updates are essential for application of the efficient system failure log analysis method at our company. Furthermore, because we need more specific knowledge on Elastic Stack when used in combination with OSS such as Apache Kafka*7 [3] and Apache Spark*8 [4] to process large-scale data, we are also accumulating knowledge on these and other types of software. They have often been used in projects to create a log analysis infrastructure, a trend that has been increasing in recent years.

References

[1] Website of Elasticsearch Co., https://www.elastic.co/
[2] Splunk, https://www.splunk.com/
[3] Apache Kafka, https://kafka.apache.org/
[4] Apache Spark, https://spark.apache.org/

Trademark notes

*6 Splunk: A commercial tool provided by Splunk. This is used to collect and visualize log records and settings on a real-time basis.
*7 Apache Kafka: OSS to achieve a distributed computing message queue.
*8 Apache Spark: An open source distributed computing framework.

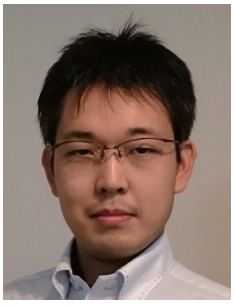


Nao Maeda
SE Section, Technology Planning Department, NTT Comware Corporation.
She received a B.E. and M.E. in informatics from Kwansei Gakuin University, Hyogo, in 2013 and 2015. She joined the NTT Group in 2015.

Norito Agetsuma
SE Section, Technology Planning Department, NTT Comware Corporation.
He received a Diploma in information technology from O-HARA Professional Training School of information and business, Tokyo, in 2008. He joined the NTT Group in 2008. He is a co-author of the book "Perfect Java EE (2016)" in Japanese.

Kenichi Kamimura
Engineer, SE Section, Technology Planning Department, NTT Comware Corporation.
He graduated from Kaseda High School, Kagoshima. He joined the NTT Group in 1984.

Yasumasa Suenaga
Engineer, SE Section, Technology Planning Department, NTT Comware Corporation.
He received a B.E. in information and communication engineering from Tokyo Denki University in 2006. He joined the NTT Group in 2006. He is a reviewer of OpenJDK and a committer of IcedTea (HeapStats).



Shinya Takebayashi
Engineer, SE Section, Technology Planning Department, NTT Comware Corporation.
He received a B.E. in information and computer science from Kanagawa Institute of Technology in 2005. He joined the NTT Group in 2005. His main interest is parallel computing such as GPU acceleration and high performance computing clusters.

Katsuyuki Yamashita
Manager, SE Department, Core Technology Quality Management and Engineering Division, NTT Comware Corporation.
He received a B.E. and M.E. in communication engineering from Osaka University in 1997 and 1999. He joined NTT Comware in 1999. His main research interest is improving the efficiency of software development.

Electrical Current Generation by Sorting Thermal Noise

Kensaku Chida, Katsuhiko Nishiguchi, and Akira Fujiwara

Abstract

We have achieved the operation of Maxwell's demon with an electrical device, which is a feedback operation based on single-electron motion. Maxwell's demon is related to the lower bound of energy consumption in electrical devices and power generation efficiency in small systems. We therefore anticipate that this achievement will contribute to creating nanoscale energy-efficient electrical devices.

Keywords: thermal noise, silicon transistor, Maxwell's demon

1. Second law of thermodynamics and Maxwell's demon

The second law of thermodynamics states that every physical system eventually becomes random. This law prohibits us from creating ordered motion of electrons, or electrical current, from thermal noise unless we modify the electrons externally. Maxwell's demon is a thought experiment created by Scottish mathematician James Clerk Maxwell to contradict the second law of thermodynamics; it is an ideal entity that can perform a feedback operation by observing objects at the level of thermal noise and can create ordered motion of electrons using energy from thermal noise without our having to modify the electrons.

Maxwell's demon has been vigorously discussed among physicists for more than 150 years because it appears to violate the second law. The discussions have clarified that Maxwell's demon uses information about the thermal motion of electrons, which requires energy. This means that we need a certain amount of energy to obtain the information, and we can create the same amount of energy at maximum from it.

This idea leads to the concept of information thermodynamics, in which the role of information is on the same footing as energy. Information thermodynamics reveals the lower bounds of the thermodynamic cost of information processing [1] and provides us knowledge about small heat engines such as molecular motors whose motions are driven by thermal fluctuations. Maxwell's demon is the epitome of information thermodynamics, which is considered to be related to energy-efficient biological systems.

In this study, we succeeded in generating electrical current and power with Maxwell's demon in one of the most natural ways: observing electron thermal motion, sorting electrons with the obtained information, and outputting the sorted electrons that had large energy.

2. Silicon nanotransistors and operation of Maxwell's demon

In the experiment, we used a silicon nanodevice on a silicon-on-insulator wafer. In our device, a single-electron box electrostatically defined by two transistors provides doors through which electrons can enter and exit the box, and a capacitively coupled detector with single-electron sensitivity detects thermal fluctuation in the box at the single-electron level (**Fig. 1**) [2]. By switching the transistors on and off, we can open and close the entrance to the box and the exit from it individually. The number of electrons in the box was observed in real time at a frequency of about 14 Hz by measuring the current flowing through the detector. All the measurements were done at room

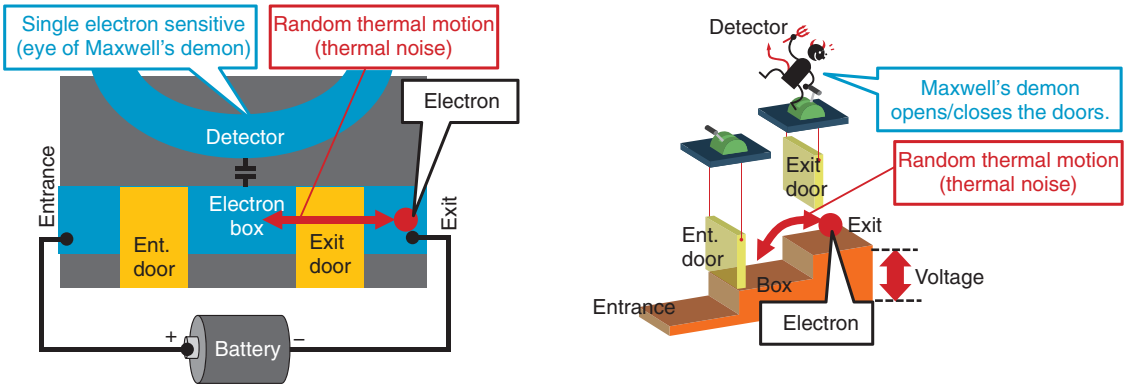


Fig. 1. Device structure. A schematic illustration of the structure of our device (left), and a corresponding schematic illustration of Maxwell's demon (right). The nanometer-scale silicon transistors serve as doors that control the motion of electrons. Electrons move in a direction determined by the voltage as an average flow. However, at the single-electron level, the motion is random because of thermal noise. The detector can observe the random motion.

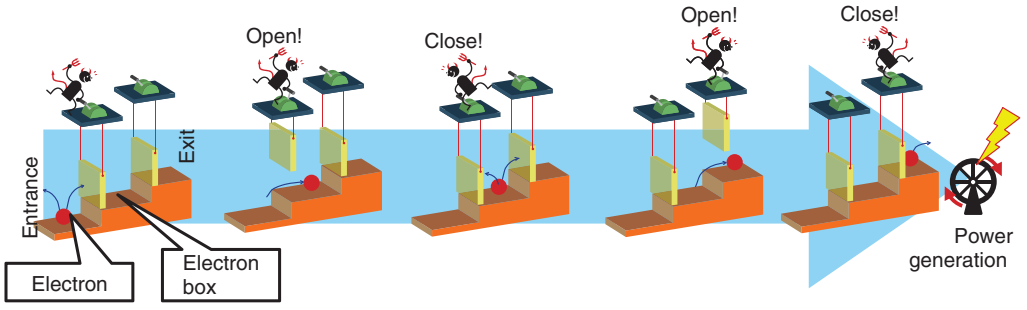


Fig. 2. Operation of Maxwell's demon. Maxwell's demon opens and closes the entrance and exit doors based on the number of electrons in the electron box, which thermally fluctuates.

temperature.

We performed feedback control based on the number of electrons in the box [3] as follows (**Fig. 2**). First, we open the entrance and observe random electron motion between the entrance and the box. After electrons have entered the box, we close the entrance. Then, we open the exit and observe random electron motion between the exit and the box. Finally, after the electrons have exited the box, the exit is closed. By repeating these procedures, we can move electrons from the entrance to the exit. We carefully calibrated the experimental setup to achieve the operation of Maxwell's demon without doing any external work on the electrons [4].

3. Rectification of thermal noise with Maxwell's demon

Carrying out the above procedures enables sorted electrons to flow as electrical current. They can even climb up the potential energy across the entrance and exit (**Fig. 3**); this climbing is the power generation. When the source-drain bias voltage V_{SD} is ~30 mV, the generated power shows the maximum value of 0.5 zW (10^{-21} W). The quantitatively estimated information-to-energy conversion efficiency is 18%, which is reasonably high and consistent with our theoretical simulation. This consistency indicates that silicon nanodevices are an ideal platform for studying Maxwell's demon and information thermodynamics. The simulation also demonstrated that the power output increases as the detector becomes faster and the box becomes smaller. Therefore, further advances in

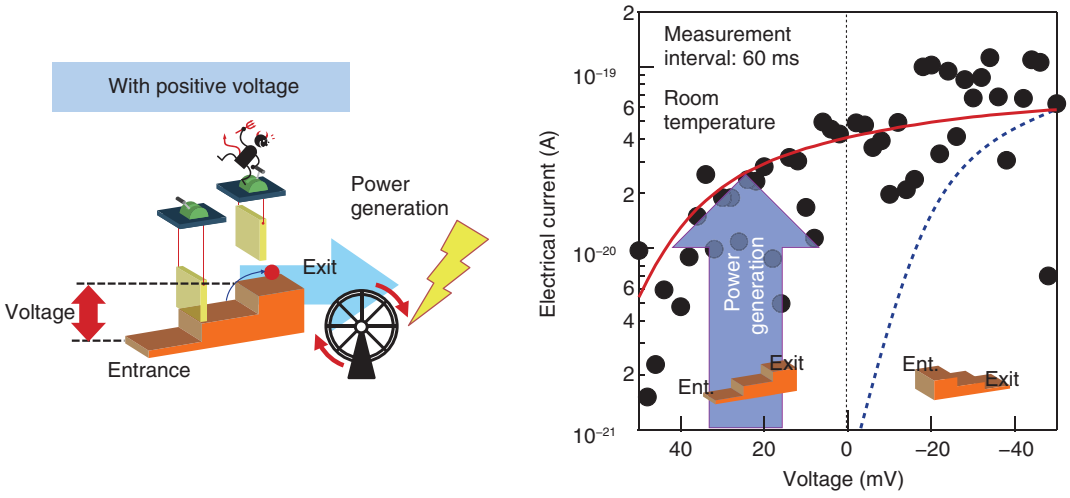


Fig. 3. Power generation with Maxwell's demon. On the left is a schematic illustration of power generation with Maxwell's demon. Maxwell's demon sorts electrons with large energy and outputs them. The energy can be utilized as electrical power as depicted. The right graph shows the electrical current (in the direction corresponding to electron motion from the entrance to the exit) as a function of the voltage. The voltage represents the height of the exit relative to that of the entrance (when the exit is higher than the entrance, the voltage is positive). The dotted blue line is the expected electrical current without Maxwell's demon. In this situation, electrons flow from the entrance to the exit only when the voltage is negative. In contrast, with Maxwell's demon, electrons are expected to flow from the entrance to the exit even when the voltage is positive, which is shown by the red line. The experimentally obtained electrical currents, plotted as black circles, are similar to the expected value with Maxwell's demon. When the voltage is positive, electrons climb up the voltage, and Maxwell's demon generates electrical power as illustrated on the right.

transistor technology will lead to an increase in the demon's power output.

4. Future work

The results in this work are closely related to the lower bound of energy consumption in electrical devices and the efficiency of small heat engines. To achieve high energy-to-power conversion efficiency, biomolecules, for example, molecular motors, are thought to use information about themselves to perform their operation at proper timings. Such an efficient process in biomolecules can be modeled and analyzed in the framework of Maxwell's demon. We will attempt in the future to deepen our understanding of the mechanism responsible for the high efficiency in biological systems and thereby make an electrical device that mimics their high efficiency.

References

[1] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," IBM J. Res. Dev., Vol. 5, No. 3, pp. 183–191, 1961.
[2] K. Nishiguchi, Y. Ono, and A. Fujiwara, "Single-electron Thermal Noise," Nanotechnology, Vol. 25, No. 27, 275201, 2014.
[3] K. Chida, K. Nishiguchi, G. Yamahata, H. Tanaka, and A. Fujiwara, "Thermal-noise Suppression in Nano-scale Si Field-effect Transistors by Feedback Control Based on Single-electron Detection," Appl. Phys. Lett., Vol. 107, 073110, 2015.
[4] K. Chida, S. Desai, K. Nishiguchi, and A. Fujiwara, "Power Generator Driven by Maxwell's Demon," Nat. Commun., Vol. 8, 15301, 2017.

Kensaku Chida

Research Scientist, Nanodevices Research Group, Physical Science Laboratory, NTT Basic Research Laboratories.

He received a B.S. in environmental science from Okayama University in 2008 and an M.S. and Ph.D. in chemistry from Kyoto University in 2010 and 2013. He joined NTT Basic Research Laboratories in 2013. Since then, his research interests have focused on feedback control of fluctuations of single electrons in nanometer-scale devices. He is a member of the Physical Society of Japan and the Japan Society of Applied Physics (JSAP).

Katsuhiko Nishiguchi

Senior Research Scientist, Distinguished Researcher, Nanodevices Research Group, Physical Science Laboratory, NTT Basic Research Laboratories.

He received a B.E., M.E., and Ph.D. in electrical engineering from the Tokyo Institute of Technology in 1998, 2000, and 2002. He joined NTT Basic Research Laboratories in 2002. Since then, he has been engaged in research on the physics and technology of silicon nanometer-scale devices for large-scale integrated circuit (LSI) applications with low power consumption and new functions. He was an invited researcher at the National Center for Scientific Research in France in September 2008 and also a guest researcher at Delft University of Technology, Delft, the Netherlands, during 2012–2013. He was appointed a Distinguished Researcher of NTT in 2011. He received the IUPAP Young Author Best Paper Award at the International Conference on Physics of Semiconductors in 2000, the Graduate Student Silver Award at the Materials Research Society 2000 Fall Meeting, the Young Scientist Award at the JSAP Spring Meeting in 2000, the JSAP Outstanding Paper Award in 2013, and the Commendation for Science and Technology from the Minister of Education, Culture, Sports, Science and Technology of Japan (the Young Scientists' Prize) in 2013. He is a member of JSAP and the Institute of Electrical and Electronics Engineers (IEEE).

Akira Fujiwara

Senior Distinguished Researcher, Senior Manager of Physical Science Laboratory, and Group Leader of Nanodevices Research Group, NTT Basic Research Laboratories.

He received a B.E., M.E., and Ph.D. in applied physics from the University of Tokyo in 1989, 1991, and 1994. He joined NTT LSI Laboratories in 1994 and moved to NTT Basic Research Laboratories in 1996. Since 1994, he has been researching silicon nanostructures and their application to single-electron devices. He was a guest researcher at the National Institute of Standards and Technology, Gaithersburg, MD, USA, during 2003–2004. He was a director of JSAP in 2010 and 2011 and a visiting professor at Hokkaido University in 2013. He was appointed a Distinguished Researcher of NTT in 2007 and Senior Distinguished Researcher of NTT in 2015. He received the SSDM Young Researcher Award in 1998, an SSDM Paper Award in 1999 at the International Conferences on Solid State Devices and Materials, and Japanese Journal of Applied Physics Paper Awards in 2003, 2006, and 2013. He was awarded the Prize for Science and Technology from the Minister of Education, Culture, Sports, Science and Technology in 2017. He was supported by the funding program for Next Generation World-Leading Researchers (NEXT Program), the Japan Society for the Promotion of Science, in 2011–2014. He is a member of JSAP and a senior member of IEEE.

Report on 16th CJK (China, Japan, and Korea) IT Standards Plenary Meeting

Hideyuki Iwata

Abstract

The 16th CJK IT (information technology) Standards Meeting was held in Jeju, Korea, on August 30 and 31, 2017. The meeting is attended by members of four domestic standards bodies from the participating countries of China, Japan, and Korea: TTA (Telecommunications Technology Association) of Korea, CCSA (China Communications Standards Association) of China, ARIB (Association of Radio Industries and Businesses) of Japan, and TTC (The Telecommunication Technology Committee) of Japan. Its purpose is to enable these bodies to share information about the standardization strategies of the respective countries. Attempts were made for the first time to shorten this and subsequent meeting periods by holding only a Plenary Meeting and a TACT (Group on Administrative Matters) meeting, which is responsible for defining rules regarding meeting administration.

Keywords: CJK IT Standards Meeting, IoT, 5G

1. Overview

The CJK IT Standards Meeting is held annually, with its venues rotating among the three countries. The previous meeting was held in Xining, China and the latest in Jeju, Korea (**Photo 1**). Besides TACT, the CJK Meeting currently has four working groups (WGs): NSA (Network and Service Architecture), IMT (International Mobile Telecommunications), IS (Information Security), and WPT (Wireless Power Transmission). Progress in the activities of these WGs over the last 12 months was reported at the Plenary Meeting. It was agreed in the previous Plenary not to hold WG meetings simultaneously with a Plenary Meeting and to limit the number of participants to the bare minimum (up to eight persons) in order to reduce the meeting cost and the preparation workload. As of this Plenary, the meeting period has been compressed from the previous three days to one-and-a-half days, and only Plenary and TACT meetings were held.

2. Results from Plenary Meeting

The main results of the Plenary Meeting are reported in this section.

2.1 Activity progress reports from each country's standards bodies

(1) TTA (Korea)

The Telecommunications Technology Association (TTA) reported that it had a membership of 204 organizations and employs 439 persons (64 in standardization, 42 in planning, 314 in certification, and 19 in other activities). It tackled two primary standardization issues for fiscal year (FY) 2017: a hyper-connected society platform (Internet of Things (IoT), fifth-generation mobile communications systems (5G), cloud/big data, artificial intelligence, and security) and smart convergence services (realistic media, smart devices, health-related ICT (information and communication technology), smart homes, and autonomous cars). It reformed its organization to address 5G and initiated a global certification service for oneM2M.

Photo 1. Participants of 16th CJK IT Standards Meeting.

(2) CCSA (China)

The China Communications Standards Association (CCSA) stated that it had 358 regular members, 13 supporting members, and 29 observers. Its main standardization issues for FY 2017 were targeted at the fields of 5G; industrial Internet and smart manufacturing; telematics and connected cars, smart cities, and cyberspace security. It also reported that as a result of a revision of the Standardization Law by the Chinese government, Consortium Standards were defined in addition to the existing National Standards and Industry Standards.

(3) ARIB (Japan)

The Association of Radio Industries and Businesses (ARIB) reported that it had 189 regular members and 13 supporting members. In FY 2017, its standardization activities focused on 5G, promotion of worldwide diffusion of UHDTV (ultra high-definition television) systems, and frequency spectrum adjustment for implementing ITS (intelligent transport systems).

(4) TTC (Japan)

With 90 regular members and 6 supporting members, the Telecommunication Technology Committee (TTC) directed its main standardization efforts at 5G, IoT, and all-IP (Internet protocol) telephone networks in FY 2017. It made a brief presentation on the IoT Innovation Promotion Committee, which was established in June 2017.

2.2 Discussions on standardization strategies

In light of the establishment by ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) of two new Focus Groups (FGs)—FG DLT (Distributed Ledger Technology) and FG DFC (Digital Fiat Currency)—and of the establishment of ISO/TC (International Organization for Standardization/Technical Committee) 307 Blockchain DLT, TTA proposed the formation of a related WG in CJK. After some discussion, it was agreed to study blockchains in an ad hoc group within the IS WG. ARIB invited Chinese and Korean companies to participate in a feasibility test driven by 5GMF (Fifth Generation Mobil Communications Promotion Forum) in Japan. It was decided to discuss this issue in the IMT WG.

3. Results from TACT meeting

TACT discussed adding to the CJK Guidelines a change in the way the Plenary Meeting is held, as practiced on this occasion. The Plenary Meeting approved this change.

4. Reports from individual WGs

The reports from WGs are summarized in this section.

(1) NSA WG

No WG meetings have been held in the last two

years. The Plenary Meeting discussed whether or not this WG should be continued, and it was decided to contact the Korean person in charge of this WG because there is a need to study non-wireless issues of 5G.

(2) IMT WG

The 47th meeting of this WG was held in Shanghai, China, the 48th in Busan, Korea, and the 49th in Fukuoka, Japan. The WG submitted five contributions to ITU-R (Radiocommunication Sector) Working Party 5D (IMT systems). The next meeting will be held in Suzhou, China.

(3) IS WG

The 12th meeting was held in Busan, Korea and the 13th in Guangzhou, China. At those meetings the issue of how ITU-T Study Group 17 should be organized in the new study period was discussed, and DLT was studied. The next meeting was scheduled for Japan in January 2018.

(4) WPT WG

The 13th meeting was held in Osaka, Japan and the 14th in Jeju, Korea. The WG gave its consent to the CJK Technical Report on WPT Edition 4 at the WG level and proposed it to AWG-21 (21st Meeting of the Asia-Pacific Telecommunity Wireless Group). The next meeting will be held in China.

5. Future plan

The 17th Plenary Meeting will be jointly hosted by ARIB and TTC in Matsue, Japan, in October 2018.

Hideyuki Iwata

Senior Research Engineer, Supervisor, Research and Development Planning Department, NTT.

He received a Ph.D. in electrical engineering from Yamagata University in 2011. From 1993 to 2000, he conducted research on high-density and aerial optical fiber cables at NTT Access Network Service Systems Laboratories. Since 2000, he has been responsible for standardization strategy planning for NTT research and development. He has been a delegate of International Electrotechnical Commission (IEC) Subcommittee 86A (optical fiber and cable) since 1998 and of the ITU-T Telecommunication Standardization Advisory Group since 2003. He is a vice-chair of the Expert Group on Bridging the Standardization Gap in the Asia-Pacific Telecommunity Standardization Program Forum. In 2004, he received an award from the IEC Activities Promotion Committee of Japan for his contributions to standardization work in IEC.

Gas Leak Search Techniques for Underground Metallic Cables

Abstract

Gas leaks that occur in underground metallic cables can potentially cause circuit problems that disrupt communications. This article describes the techniques used to identify gas leaks in underground cables. This is the forty-fourth article in a series on telecommunication technologies. This contribution is from the Access Engineering Group, Technical Assistance and Support Center, Maintenance and Service Operations Department, Network Business Headquarters, NTT EAST.

Keywords: gas leak, metallic cable, underground cable

1. Introduction

Underground cables that have been laid throughout Japan are found in underground conduits and manholes, and in such environments, countermeasures must be taken to prevent damage caused by rainwater and groundwater. In particular, because the core wires of metallic cables are made of copper, the penetration of water into the cable or a cable splice can cause circuit faults, and measures are therefore taken to prevent such water penetration.

With subscriber metallic cables, the interval from the NTT central office to the feeder point is called the feeder section and that from the feeder point to the customer premises is called the distribution section. In the feeder section interval, the cable interior is filled with dry air to create sufficient pressure to prevent the penetration of water (**Fig. 1**).

In gas-filled cables, the occurrence of a gas leak will cause the gas pressure within the cable to drop, thereby increasing the risk of water penetration. It is therefore important to monitor the gas pressure and take appropriate measures.

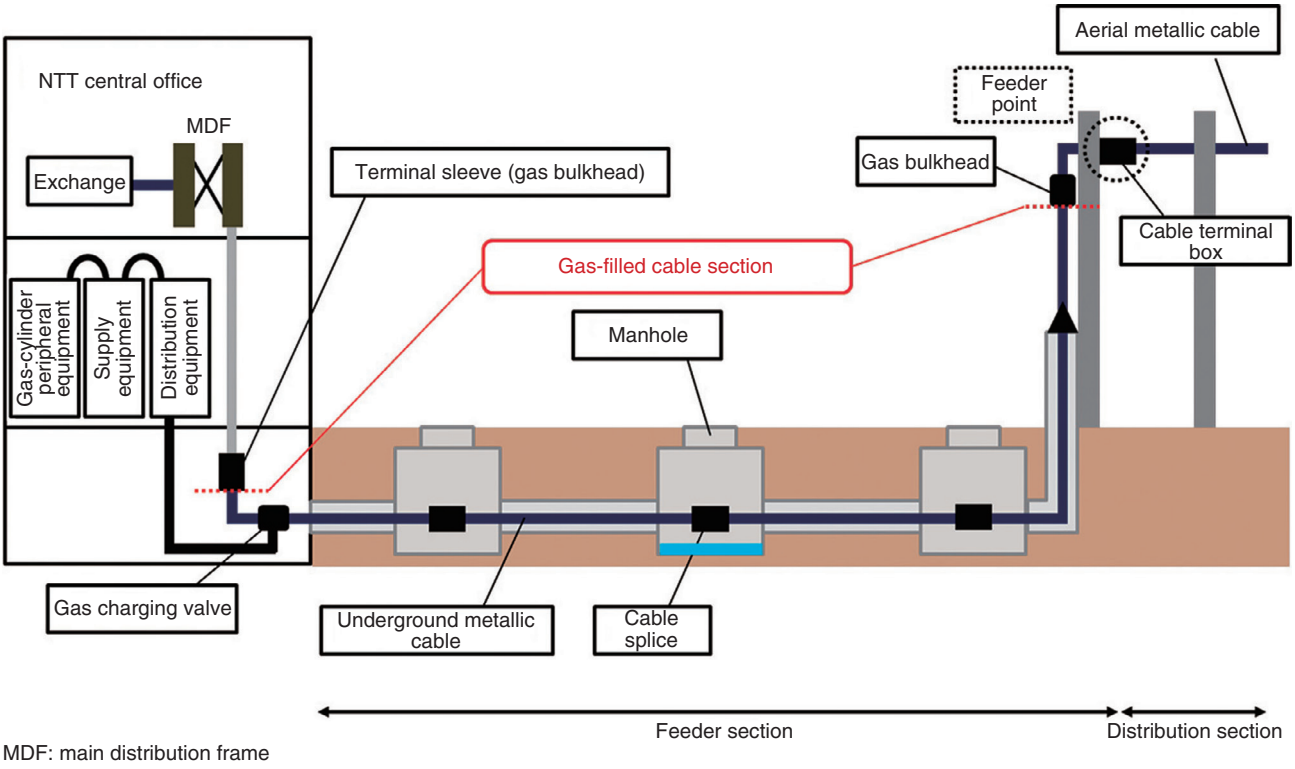
2. Gas leak factors

One factor in gas leaks is cable movement (creeping phenomenon) in which a cable moves from its original position due to temperature-related elongation/contraction, vibrations caused by passing vehicles, and other factors. A gas leak can occur if excessive force comes to be applied to the cable gripper at a cable splice due to cable movement (**Fig. 2**). Creeping can also damage the cable jacket and create pinholes that can also be a factor in gas leaks. Places with a soft foundation where heavy vehicles tend to pass and straight roads with a notably uneven surface constitute environments in which cable creeping can easily occur.

3. Inference of gas leak points

A system has been deployed that can remotely monitor the gas pressure within underground metallic cables. This remote gas monitoring system includes a pressure transmitter that converts gas pressure within cables to electrical signals, a remote monitoring unit (RMU) that receives data from a pressure transmitter, an alert display (ALD) that displays alerts, and a workstation (WS) or personal computer (PC) that displays data (**Fig. 3**).

In this system, a pressure transmitter capable of measuring gas pressure is installed at each cable connection point. This makes it possible to remotely measure and tabulate gas pressure within the cable and to use this data to prepare a pressure distribution chart. If there is no gas leak in the cable, the plot on the pressure distribution chart will be flat. However, if a gas leak is present, a drop in pressure can be seen originating at the leak location. Then, after a certain



MDF: main distribution frame

Fig. 1. Configuration of metallic facilities.

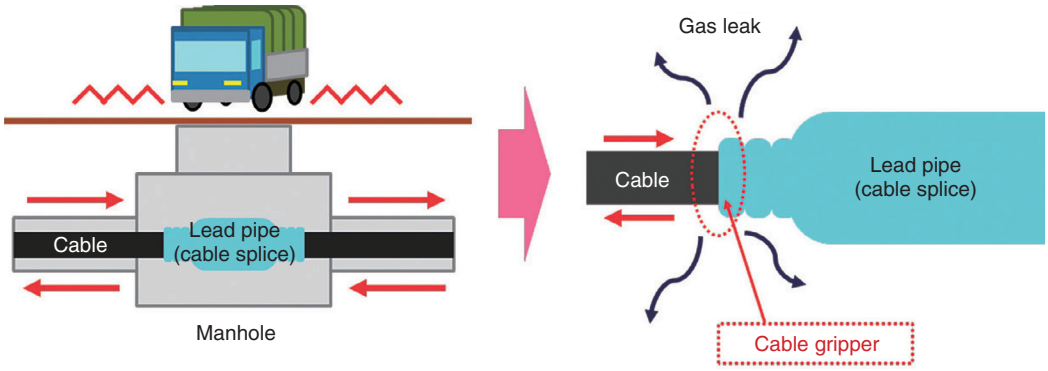


Fig. 2. Gas leak caused by cable movement.

amount of time, if there is no change in the size of the gas leak hole or in the gas feeding pressure, the result will be an equilibrium distribution in which the distribution of pressure applied to the terminal side from the leak point is invariable (**Fig. 4**).

The gas leak point is inferred by analyzing the pressure distribution chart using pressure measurement values at a minimum of four points. Here, given that the pressure distribution chart indicates a state of equilibrium distribution, the leak point is inferred to be the intersection of the straight line on the gas-feeding side and the straight line on the terminal side (**Fig. 5**).

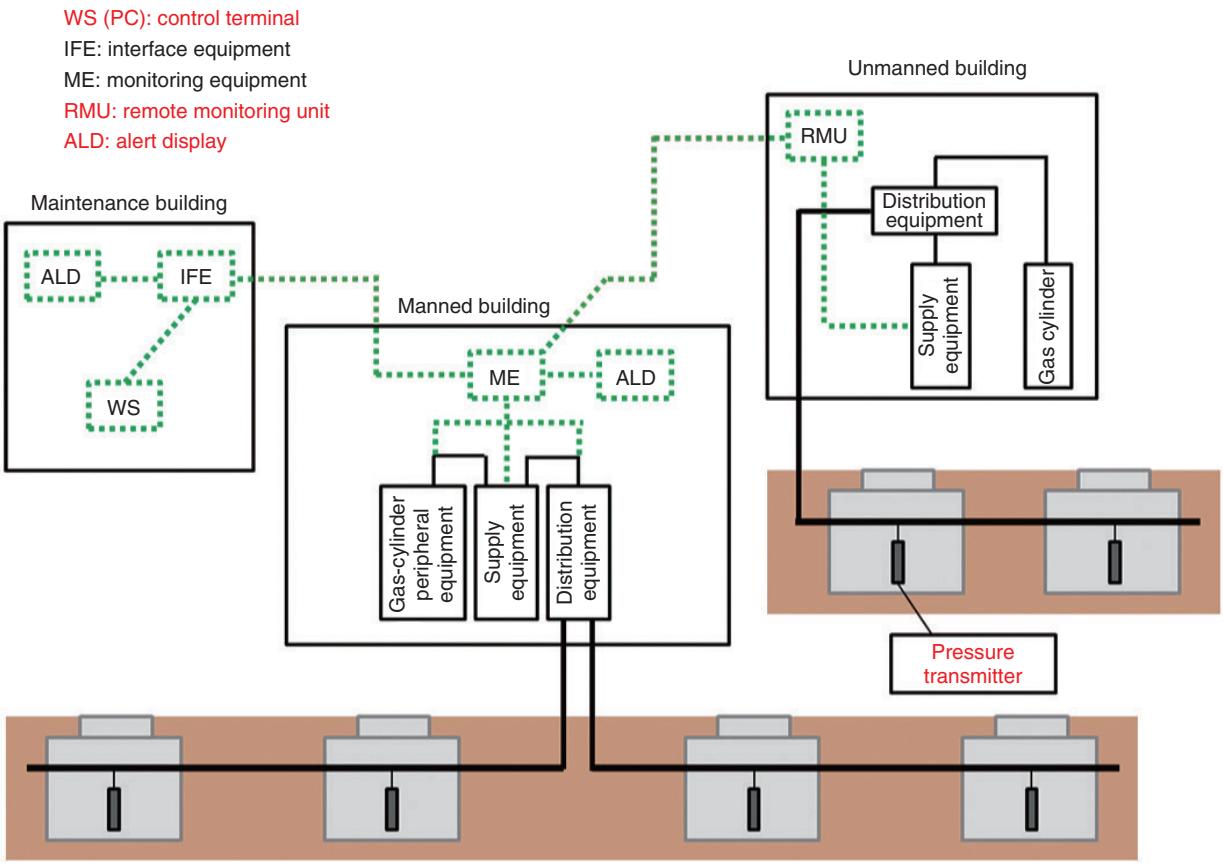


Fig. 3. Configuration of gas facilities.

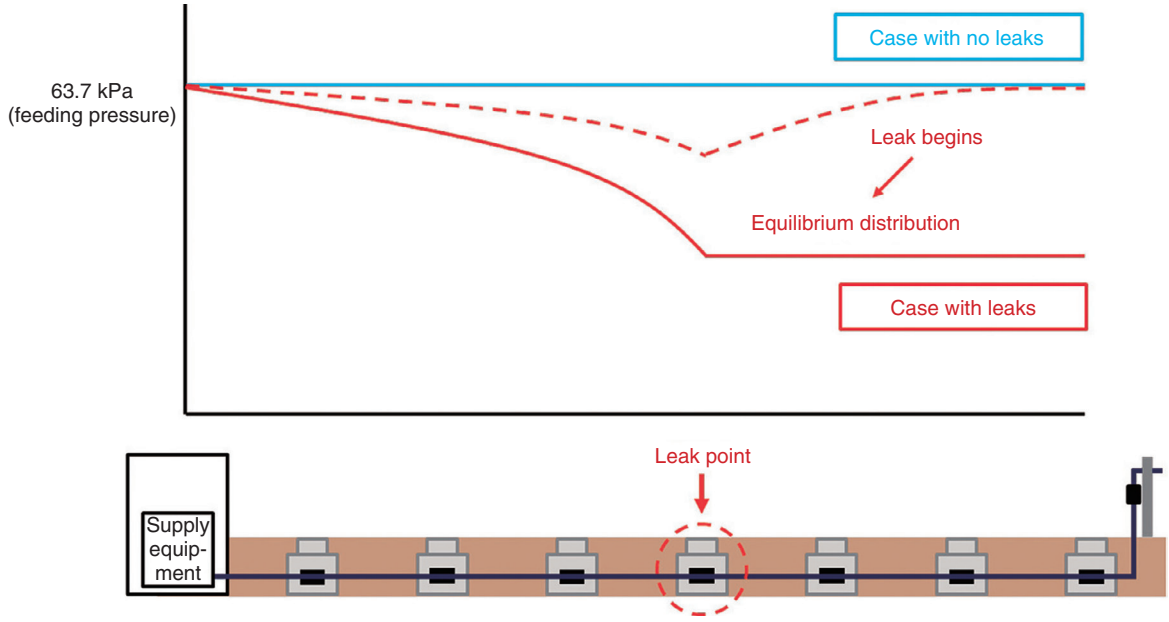


Fig. 4. Gas pressure distribution chart.

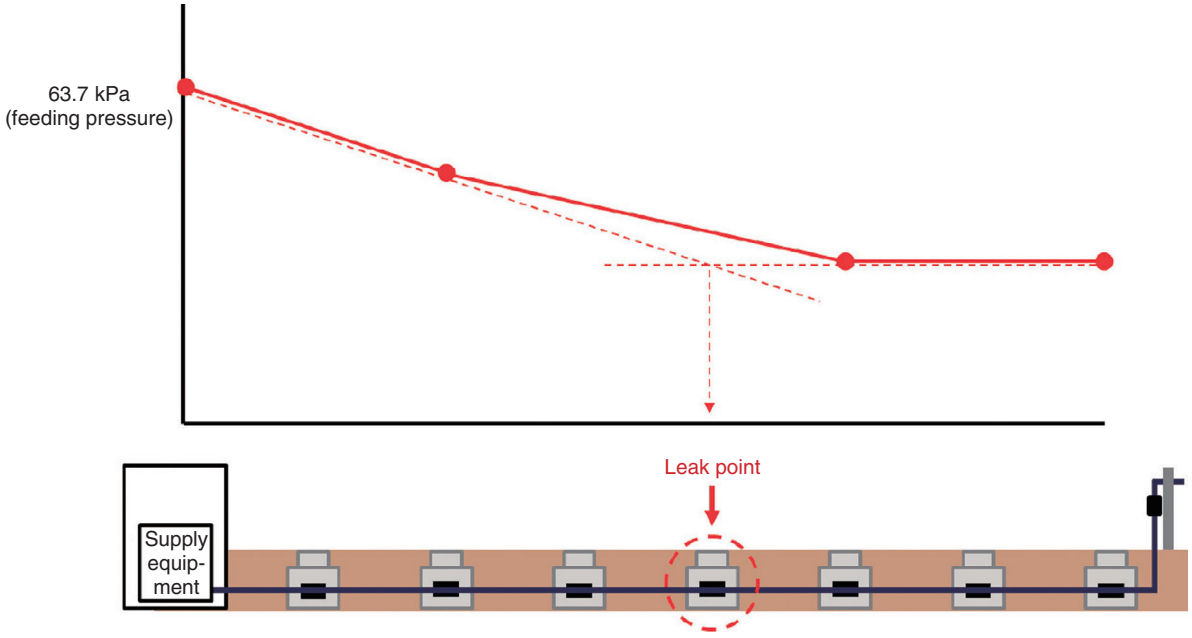


Fig. 5. Inference of gas leak point.

4. Introduction to gas leak search technology

(1) Gas leak search using soapy water

A leak hole can be promptly searched for if the result of the above gas leak point inference method indicates that a manhole is the suspected location of a leak. Once a maintenance personnel is inside the manhole, he/she coats the surface of the cable or connection point with soapy water or similar, and if a leak hole does in fact exist, it will be revealed by the formation of soapy-water bubbles as the gas leaks out (**Fig. 6**).

A search with soapy water is relatively easy to do, but variations of this technique and its results can easily occur between maintenance personnel, depending on the concentration of the soapy water and the ease of coating the target material, which raises concerns that some leaks may be missed.

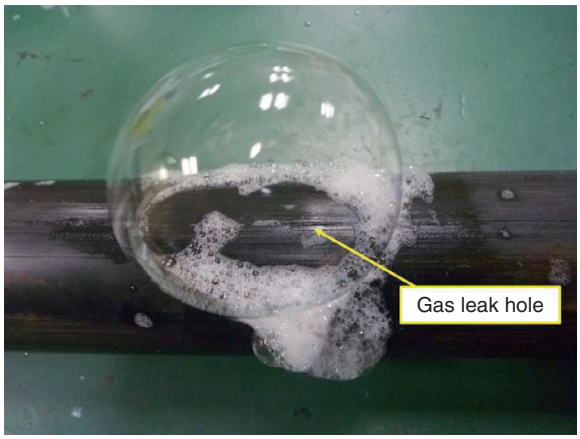


Fig. 6. Search for gas leak using soapy water.

(2) Gas leak search using helium gas

If a leak hole is not found with soapy water after conducting a leak search inside all manholes concerned by inferring gas leak points, or if a location other than a manhole or handhole (e.g., within a conduit interval) turns out to be the suspected location, a leak search can be conducted using helium gas.

Helium gas has a number of advantageous properties; it is non-combustible and odorless, has a high degree of dryness (does not degrade the electrical insulation of cable core wires), and does not react with other substances. In short, it is chemically stable while being much lighter than air (mass ratio: 1:0.17). Furthermore, its presence in the atmosphere is extremely low, which makes it useful as a tracer gas. Moreover, it is not environmentally destructive, nor is it harmful to the human body.

In a helium gas search, the dry air in a cable is replaced with helium gas. Then a search for a helium gas leak is carried out using helium gas detection

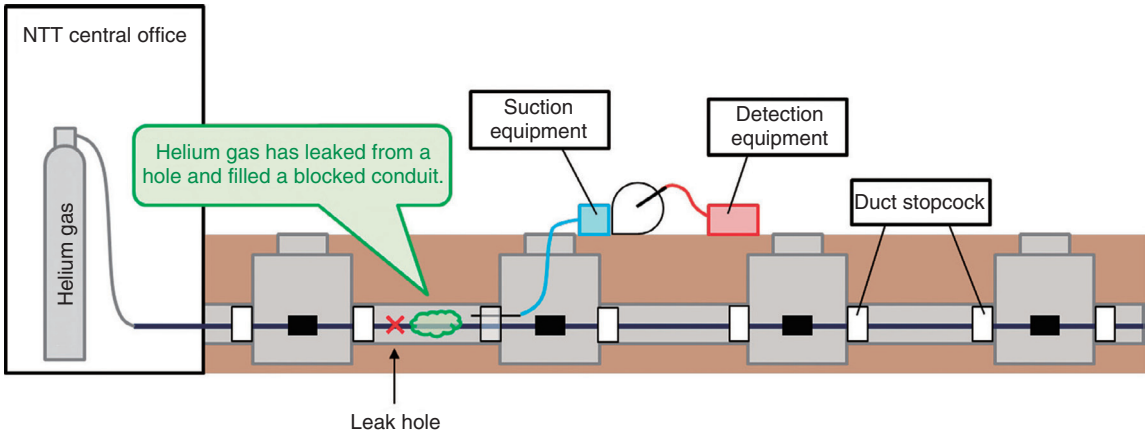


Fig. 7. Gas leak search using helium gas.

Fig. 8. Helium gas detection equipment.

equipment. In preparation for this search, a stopcock is used to close the duct outlet of the conduit suspected of being the leak location. If a gas leak hole exists within this conduit, the helium gas leaking from the hole will fill the closed conduit. The air in the conduit is sucked out using suction equipment and then examined by the detection equipment. The existence of a leak hole within the conduit can therefore be checked based on the presence or absence of helium gas (**Fig. 7**).

A gas leak search using helium gas requires replacing the dry air within the cable with helium gas and using specialized search equipment, making it more complex than the soapy water technique (**Fig. 8**). However, it is a more reliable method of finding leaks than the soapy water search method.

5. Process of gas leak search

We describe here an example process of searching for a gas leak in an underground metallic cable.

First, the gas leak point is inferred based on the gas pressure distribution chart obtained by the gas remote monitoring system. In this process, the suspected leak location is established based on the gradients of the gas pressure plots and the terminal pressure. If the inferred leak point is inside a manhole, the cable or connection point can be coated with soapy water to search for a location where soapy-water bubbles form. At this time, gas leaks at the cable gripper of a connection point will be carefully checked since such leaks have a high probability of occurring there. If a gas-pressure-drop alert should be displayed, the cable can be charged with clean gas (from a cylinder) from the terminal side to increase gas pressure as an emergency measure, but this may make it difficult to correctly grasp the gas pressure distribution. It is important to take note of the gas pressure distribution at the time of a gas pressure drop and compare it with the gas pressure distribution after charging with clean gas from the terminal side.

When a leak hole cannot be found using soapy water despite conducting a leak search in all manholes concerned based on the results of the gas leak point inference method, or if the suspected leak location is within a conduit interval, a leak search will then be performed using helium gas. Before the search is conducted, the duct outlet of the conduit enclosing the suspected leak location will be closed using a stopcock. To replace the dry air in the cable with helium gas, the cable will be charged with helium

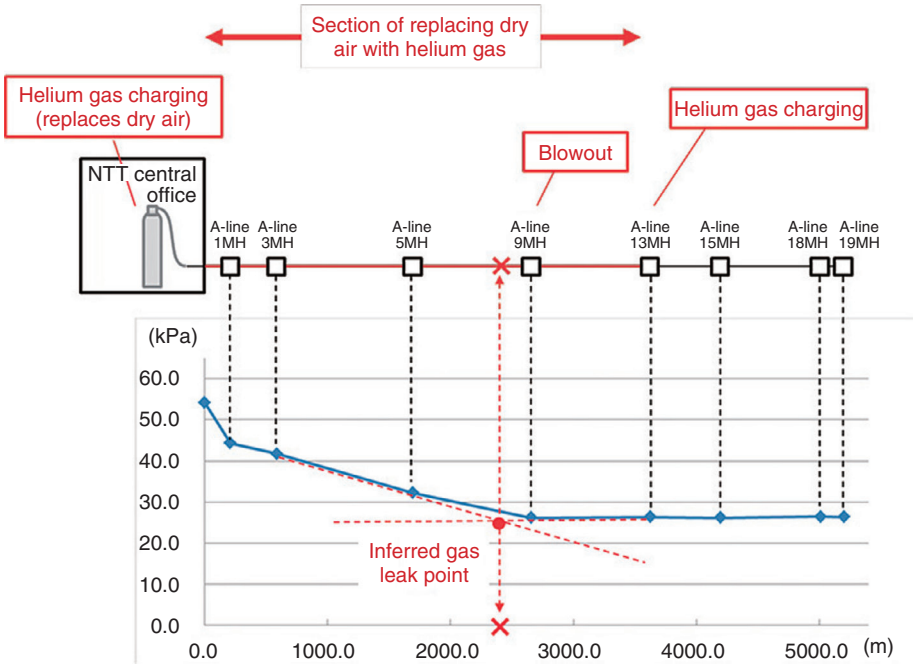


Fig. 9. Example of gas pressure distribution chart and replacement of air with helium gas.

gas from the NTT central office gas room via points preceding the suspected leak location. To facilitate the replacement of dry air with helium gas at this time, the dry air within the cable will be blown out via a gas valve at the connection point on the other side of the helium gas charging point corresponding to the suspected leak location. Once the dry air within the cable has been sufficiently replaced with helium gas, the air within the closed-off conduit is sucked out by suction equipment and passed to helium gas detection equipment to check for the presence of helium gas within the conduit. If the cable within this conduit does indeed have a leak hole, helium gas will be detected at this time (**Fig. 9**).

6. Conclusion

This article introduced techniques used to search for gas leak points in underground metallic cables as a preventive measure against circuit faults. The Technical Assistance and Support Center is committed to using the knowledge and experience it has gained in the maintenance and operation of diverse facilities to respond to requests for technology consultations and support from all concerned parties and to promote technology dissemination with the aim of improving the reliability of access facilities.

External Awards

CSS2017 Excellent Paper Award

Winner: Yuhei Kawakoya, Makoto Iwamura, Jun Miyoshi, NTT Secure Platform Laboratories
Date: October 24, 2017
Organization: The Computer Security Symposium (CSS) 2017, Information Processing Society of Japan Special Interest Group on Computer Security

For "Taint-assisted Forensics for IAT Reconstruction."
Published as: Y. Kawakoya, M. Iwamura, and J. Miyoshi, "Taint-assisted Forensics for IAT Reconstruction," Proc. of CSS2017, 2A3-3, Yamagata, Japan, Oct. 2017.

Nishina Memorial Prize

Winner: Hiroki Takesue, NTT Basic Research Laboratories
Date: November 10, 2017
Organization: Nishina Memorial Foundation

For the realization of large-scale coherent Ising machines.

IEEE Microwave Theory and Techniques Society Japan Young Engineer Award

Winner: Takuro Tajima, NTT Device Technology Laboratories
Date: November 30, 2017
Organization: IEEE (Institute of Electrical and Electronics Engineers) Microwave Theory and Techniques Society Japan Chapter

For "Compact THz LTCC Receiver Module for 300 GHz Wireless Communications."
Published as: T. Tajima, H. Song, and M. Yaita, "Compact THz LTCC Receiver Module for 300 GHz Wireless Communications," IEEE Microwave and Wireless Components Letters, Vol. 26, No. 4, pp. 291–293, Apr. 2016.

Best Paper

Winner: Bomin Mao, Zubair Fadlullah, Fengxiao Tang, and Nei Kato, Tohoku University; Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani, NTT Network Innovation Laboratories
Date: December 5, 2017
Organization: IEEE Global Communications Conference (GLOBECOM) 2017

For "A Tensor Based Deep Learning Technique for Intelligent Packet Routing."
Published as: B. Mao, Z. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "A Tensor Based Deep Learning Technique for Intelligent Packet Routing," Proc. of IEEE GLOBECOM 2017, Singapore, Dec. 2017.

Best Presentation Award

Winner: Yuko Ueno, Tetsuhiko Teshima, Calum Henderson, and Hiroshi Nakashima, NTT Basic Research Laboratories
Date: December 12, 2017
Organization: The 6th International Conference on BioSensors, BioElectronics, BioMedical Devices, BioMEMS/NEMS & Applications 2017 (Bio4Apps2017)

For "Three-Dimensional Protein Detection by Graphene Micro-roll Aptasensor."
Published as: Y. Ueno, T. Teshima, C. Henderson, and H. Nakashima, "Three-Dimensional Protein Detection by Graphene Micro-roll Aptasensor," Bio4Apps2017, Tokyo, Japan, Dec. 2017.

Papers Published in Technical Journals and Conference Proceedings

Behavioral Analysis of Kinetic Telepresence for Small Symmetric Group-to-group Meetings

K. Otsuka

IEEE Transactions on Multimedia, Vol. PP, No. 99, November 2017.

Nonverbal behavior analysis revealed the effect of MMSpace, a kinetic telepresence developed for social telepresence, on small symmetric group-to-group conversations. MMSpace consists of kinetic avatars, equipped with flat projection screen panels as faces, that can change their pose and position automatically to mirror the remote user's head motions. The advantage is the realistic kinetic expression of human head movements, which form gestures like nodding and indicate the focus of visual attention, through the use of four degree-of-freedom low-latency precision actuators. Another feature is the support of eye contact among remote participants, which is made possible by the avatars' kinetic pose changes and by adaptive camera selection for orienting the user's face toward the remote addressee. Its limitation is its room-scale infrastructure and restricted participant positions. The target was a symmetric 2×2 setting, and participants' nonverbal behaviors, including gaze directions and head gestures, were compared among three conditions: MMSpace with/without physical motions and face-to-face settings. There was a significant

difference between the conditions in terms of the duration of glance/mutual glances, total gaze transition time, amount of head gesturing, and co-occurrences of head gestures in the remote participants. The results indicate that the avatar's physical motion can elicit longer (mutual) glances with a shorter total transition time and more (co-) occurrences of head gestures, and it makes MMSpace-based conversations closer, in terms of these nonverbal statistics, to face-to-face ones compared with those of a static version of MMSpace without physical motion.

Generating Three-dimensional Shapes by Using Texture Synthesis

M. Sawayama, M. Okabe, S. Nishida, and Y. Dobashi

The Japanese Journal of Psychonomic Science, Vol. 36, No. 1, pp. 56–65, December 2017.

This research note reviews experimental methods to elucidate the visual processing underlying material perception, and considers how to generate experimental stimuli of three-dimensional shapes for the experiments. For generation of a computer graphics image of a three-dimensional object, it has been widely known that its shape features can affect the material appearance of the object. However, it has not been established how to systematically control the shape features to investigate the effect. Here we suggest utilizing texture synthesis algorithms. Specifically, we used a height map of a three-dimensional object as a source image and synthesized a novel height map by using a texture synthesis algorithm. We tested three algorithms to generate the height maps; i) synthesis based on image statistics, ii) example-based synthesis, and iii) synthesis using a convolutional neural network. We discuss how effective the texture synthesis algorithms are to investigate the effect of the shape features on the material perception.

32-core Erbium/Ytterbium-doped Multicore Fiber Amplifier for Next Generation Space-division Multiplexed Transmission System

S. Jain, C. Castro, Y. Jung, J. Hayes, R. Sandoghchi, T. Mizuno, Y. Sasaki, Y. Amma, Y. Miyamoto, M. Bohn, K. Pulverer, M. Nooruzzaman, T. Morioka, S. Alam, and D. J. Richardson

Optics Express, Vol. 25, No. 26, pp. 32887–32896, December 2017.

We present a high-core-count 32-core multicore erbium/ytterbium-doped fiber amplifier (32c-MC-EYDFA) in a cladding pumped configuration. A side pumping technique is employed for ease of pump coupling in this monolithic all-fiber amplifier. A minimum gain of >17 dB and an average noise figure (NF) of 6.5 dB is obtained over all cores in the wavelength range 1534 nm–1561 nm for −4 dBm input signal power. The core-to-core variation for both amplifier gain and NF is measured to be <2 dB. The 32c-MC-EYDFA was then tested in a repeatered multicore fiber loop system, and transmission over distances >1850 km was successfully demonstrated. We also compare the total power consumption of our MC-EYDFAs with that of 32 conventional single core erbium doped fiber amplifiers to illustrate the potential power saving benefits.

Modular Representation of Layered Neural Networks

C. Watanabe, K. Hiramatsu, and K. Kashino

Neural Networks, Vol. 97, pp. 62–73, January 2018.

Layered neural networks have greatly improved the performance of various applications including image processing, speech recognition, natural language processing, and bioinformatics. However, it is still difficult to discover or interpret knowledge from the inference provided by a layered neural network, since its internal representation has many nonlinear and complex parameters embedded in hierarchical layers. Therefore, it becomes important to establish a new methodology by which layered neural networks can be understood.

In this paper, we propose a new method for extracting a global and simplified structure from a layered neural network. Based on network analysis, the proposed method detects communities or clusters of units with similar connection patterns. We show its effectiveness by applying it to three use cases: (1) network decomposition: it can decompose a trained neural network into multiple small independent networks, thus dividing the problem and reducing the computation time; (2) training assessment: the appropriateness of a trained result with a given hyper parameter or randomly chosen initial parameters can be evaluated by using a modularity index; and (3) data analysis: in practical data, it reveals the community structure in the input, hidden, and output layers, which serves as a clue for discovering knowledge from a trained neural network.
