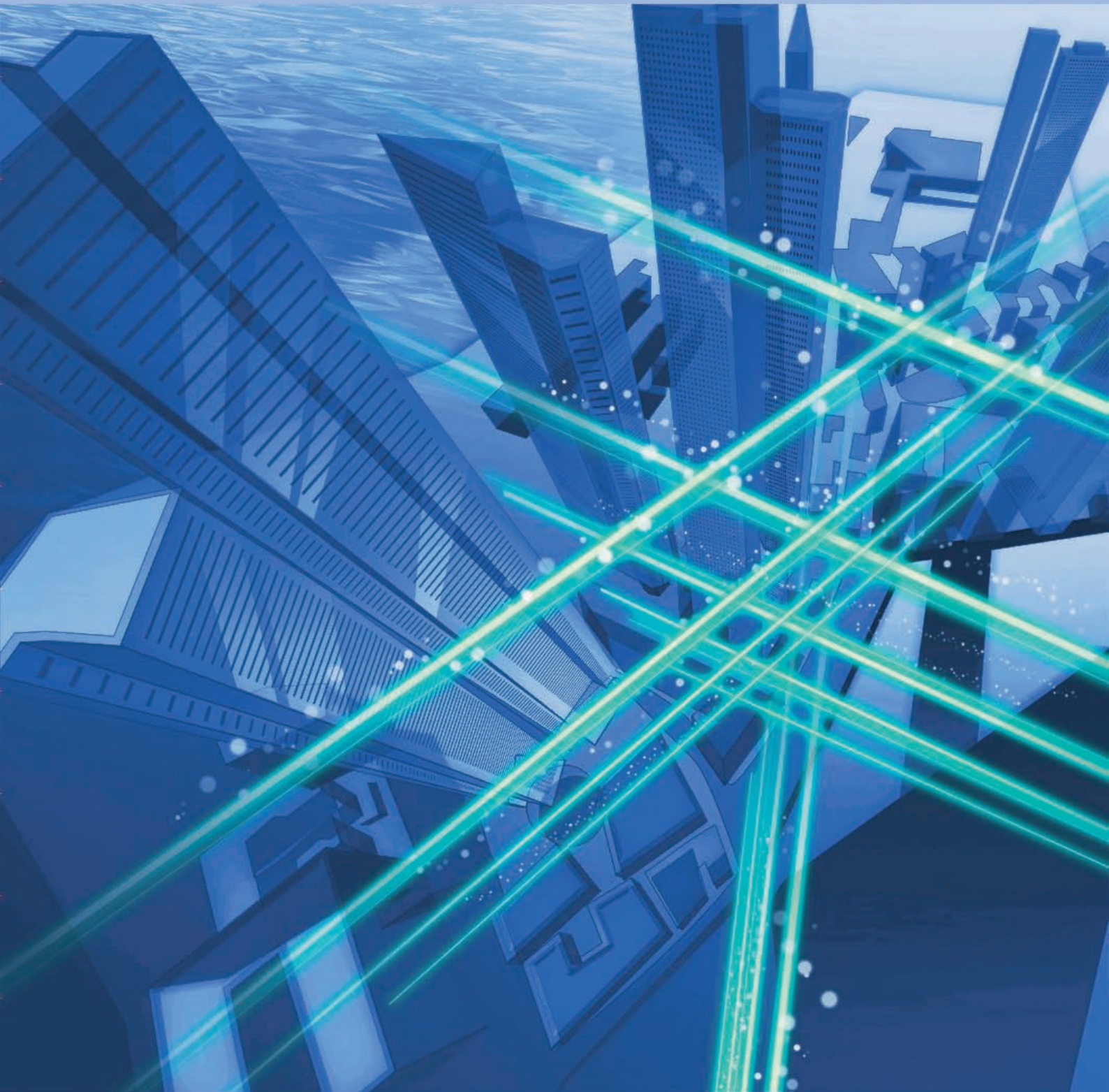


NTT Technical Review

6

2014



June 2014 Vol. 12 No. 6

NTT Technical Review

June 2014 Vol. 12 No. 6



Front-line Researchers

Dr. Shingo Tsukada, Senior Distinguished Researcher,
NTT Basic Research Laboratories

Feature Articles: Technologies for Promoting Use of Open Source Software that Contribute to Reducing TCO of IT Platform

OSSMA: Facilitating Migration to OSS

HeapStats: Your Dependable Helper for Java Applications, from Development to Operation

TUBAME: A Tool for Studying Java Application Migration

Faster System Building with No Reworking!
OSSVERT Smart Builder

Pacemaker: Increasing System Reliability

PostgreSQL: Leader in Cost Reduction

Linux and KVM Activities Providing Foundational Support for Use of OSS

Dramatic Changes in Java Troubleshooting—Pursuing Uncompromising Low-overhead Implementations

Regular Articles

Inspection Prioritizing Technique for Early Discovery of Defective Facilities—Manhole Facility Inspection Prioritization

Global Standardization Activities

Standardization Trends Relating to Smart TV

Practical Field Information about Telecommunication Technologies

Fault Cases and Countermeasures for Optical Fiber Cables in Optical Network Facilities

New NTT Colleagues

We welcome our newcomers to the NTT Group

Papers Published in Technical Journals and Conference Proceedings

Papers Published in Technical Journals and Conference Proceedings

Advance with the Conviction that All Paths—Even Detours—Lead to Success

Dr. Shingo Tsukada

Senior Distinguished Researcher, NTT Basic Research Laboratories



The Molecular and Bio Science Research Group has fabricated material for use as biomedical electrodes to continuously measure heartbeat and obtain electrocardiograms by simply wearing an item of clothing. Dr. Shingo Tsukada, a medical doctor-turned-researcher and an NTT Senior Distinguished Researcher in this group, expects *hitoe*, a textile developed jointly with Toray Industries, Inc. that incorporates these electrodes, to be useful in detecting latent diseases early. We asked him about the mindset that modern researchers should adopt and how the development of a healthy body can promote creative thinking in research.

Keywords: biomedical electrodes, nano-bio interface, PEDOT-PSS

Biomedical electrodes conceived from a doctor's clinical experience

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

Biotechnology is a field that has been researched at NTT laboratories since the 1980s. Today, the Molecular and Bio Science Research Group here at NTT Basic Research Laboratories is conducting research on nano-bio devices aimed at developing bio-compatible interfaces with the human brain. The wearable electrodes that we have developed for obtaining electrocardiograms originate in material that we originally used as ultra-small electrodes for recording fine signals generated by nerve cells in the brain. We now apply this material to ordinary-sized biomedical electrodes that can be attached to a person's skin.

Upon entering NTT, I became aware of a conductive polymer called PEDOT-PSS (Poly(3,4-ethylenedioxythiophene) polystyrene sulfonate), which I thought had great potential as a material for biomedical electrodes thanks to its flexible, biocompatible, and hydrophilic characteristics (**Fig. 1**). PEDOT-PSS, however, is quite fragile when wet, and in our initial experiments, prototypes easily broke down, resulting in a string of failures.

—What ideas led to the development of the wearable electrode?

It all began with research to find ways of supplementing damaged nerves (neural prosthesis) by using a type of ultra-small electrode that can be inserted into the brain. The traditional approach in regenerative medicine, which makes use of genes and stem

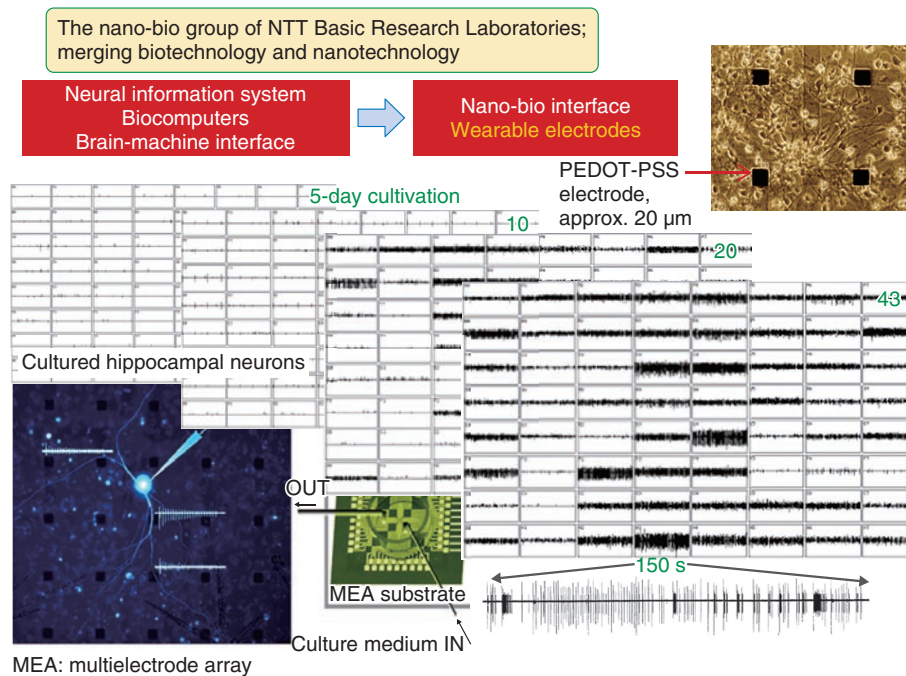


Fig. 1. Long-term measurement of neural activity using a conductive polymer.

cells, is to stretch the axonal fibers of a severed central nerve to reconstruct the nerve circuit. From a technical viewpoint, this approach can be very difficult, so the idea came to me that perhaps the severed location could be functionally repaired by bridging and connecting the gap electronically. This method requires the passing of electrical signals by piercing the brain with an electrode in the form of a thin needle. However, metallic needles that had been used up to that time lacked flexibility and caused inflammation and a gradual breakdown of brain tissue in the area where the electrode was inserted.

Solving this problem would require a highly flexible and biocompatible material that would be a good fit for the flexible and highly fragile tissue making up the brain. It was then that I came up with a method of coating silk threads used in surgical procedures with conductive PEDOT-PSS. Specifically, we were able to fabricate a strong, string-like composite material using a fiber such as silk as a base material. We then performed experiments with animals using this material as an electrode, and found that it could be used to make stable, long-term recordings of nerve signals (Fig. 2).

Through my experience as a doctor in clinical medicine, I became acutely aware of the need for a biomedical electrode that could be continuously con-

nected with minimal burden on the patient. We used this fiber material to fabricate electrodes that can be attached to the surface of the skin, thereby bringing about *hitoe* (single-layer) wearable-electrode material. This electrode material does not attach to the skin in an air-tight manner, which makes it comfortable to wear with little chance of skin rashes. *hitoe* can therefore be used to obtain long-term biomedical recordings such as electrocardiograms (Fig. 3).

The present stage is only one part of a process toward accomplishing our final objectives. In the next step, I would like to conduct a long-term investigation into nerve functions using this conductive fiber. Recent research has shown that abnormalities in nerve functions for regulating blood vessels and the heart can lead to hypertension and heart diseases. An ideal approach to clarify the mechanisms of such conditions would be to use the *hitoe* material to research everyday behavior in people's lives while conducting parallel experiments on nerve activity using implanted electrodes in animals. To date, no electrodes have been applied to long-term use, so research in this area has not been progressing as I would like, but *hitoe* should make such long-term research a reality.

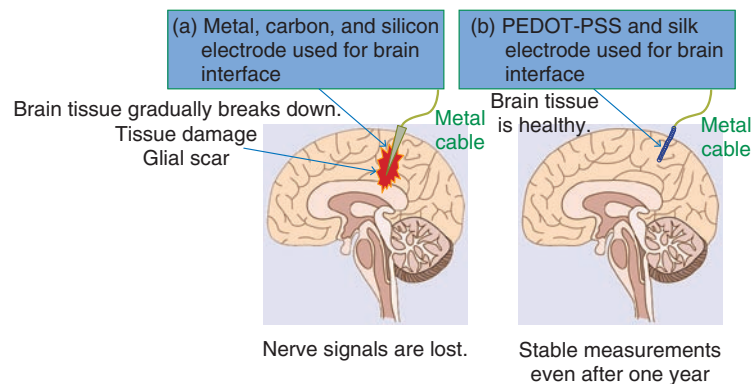


Fig. 2. Use of buried electrodes to achieve a brain-machine interface.

Sections of fabric coated with PEDOT-PSS are arranged on a shirt as electrodes. Heartbeat can be measured and electrocardiograms obtained simply by wearing clothing. Continuous long-term monitoring is possible with minimal burden on the wearer.

hitoe: human health, intelligent interface inner wear, textile technology, organic, electrode (“hitoe” means “one-layer textile” in Japanese)
 Uses ultrafine fiber (nano-fiber) developed jointly with Toray Industries to achieve wearable electrodes.

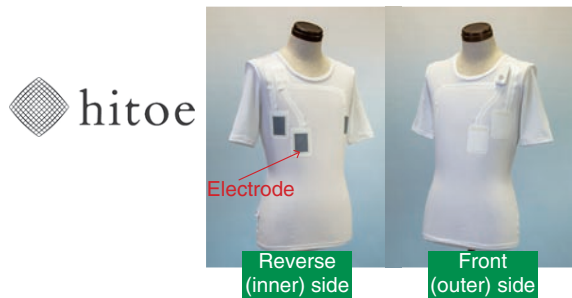


Fig. 3. Application involving wearable electrodes.

—How do you expect this research achievement to affect our everyday life in the future?

I would like to expand the use of this electrode material to medical care. For example, I can envision a system that uses *hitoe* to record a person’s biological signals in everyday life and then checks for unhealthy signs that can be used to detect a potential disease or reduce the risk of a heart attack, for example.

I am also investigating the application of *hitoe* to joint research on human cognition and behavior between NTT laboratories and universities. For example, regular exercise is an important element in maintaining a person’s health, but there are growing concerns that people—especially the young—are

exercising less. One reason why people may be reluctant to exercise or engage in sports is a feeling that they are not especially good at it. Perhaps *hitoe* can be used to overcome this negative attitude. We have found that an athlete’s biological signals recorded using *hitoe* contain interesting information that cannot be obtained from video images. It might therefore be possible to use the biological signals collected while a skilled sportsman is exercising as guidelines or teaching material in individual instruction. If *hitoe* can be used to help people quickly develop exercise skills, I think there is a good chance that these people will develop a love of exercising and will naturally increase the amount of exercise they do.

The need to “reset” by talking with friends in other fields and by sleeping and exercising

—*When exactly do you have such original ideas?*

When conversing with friends not only in my field but in completely different fields, such as theater arts and video production, I am always reminded how other people can have ideas or a sense of life completely different from my own, and that can be very stimulating for me.

Ideas also come to me when I am involved in outside activities such as listening to music or exercising. In particular, a very positive and forward-looking feeling arises within me when I am exercising. Although my career has had many twists and turns, I have been able to surmount any hardships I have had thanks to the presence of my friends and my love of exercise. During my research activities in the United States, I came to feel that Japanese people are somewhat pessimistic on the whole. Resetting your body by taking on a positive and bright attitude, sleeping well, and exercising should help you come up with novel ideas and original points of view.

—*That sounds like good advice from a doctor. By the way, you have a very unique background. Why did you decide to make the switch from a medical doctor to a researcher?*

I attribute that to my father, who was an employee of Nippon Telegraph and Telephone Public Corporation. As a child, I made many visits to Kanto Teishin Hospital in Gotanda, Tokyo, which was originally founded for NTT employees. It was there that I developed an interest in the work of doctors. I also had an interest in machines and electronic components when I was small, and I became adept at building electronic kits and fiddling with mechanical assemblies using a soldering iron, thanks to my father’s guidance.

Then, when I was in high school, I began to worry over which career path to take: doctor or researcher in a technical field. After much wavering between the two, I finally chose to become a doctor, and since I had been somewhat dexterous with my fingers since childhood, I decided to make use of that skill by becoming a surgeon. I chose, in particular, microneurovascular surgery in which I would connect nerves or blood vessels while peering through a microscope. This would involve, for example, the sewing together of blood vessels about one millimeter in diameter

using a needle with a special type of thread so that blood could circulate smoothly again. It would also mean the reconstruction of sensory nerves and motor nerves by connecting neural branches. It takes at least three years to acquire the skills to make such surgical connections, and during my initial training, I would look for spare time in between consultations with patients to practice those skills using small animals or other means. However, despite the skills that I finally acquired after much effort, I developed severe allergy symptoms on my hands and fingers due to the gloves and antiseptic solutions that are needed for surgery, and in the end, I had no choice but to abandon my plans to become a surgeon. Instead, I turned to basic medicine and began basic research in physiology and nerve reconstruction.

Today, in the field of medicine, there is no theme more important than medical treatment related to the heart and brain. These two organs are difficult to regenerate, so if they fall into a serious state or are affected by disease, treatment becomes difficult. With this in mind, I decided that I wanted to pursue research into ways of curing such complex diseases.

Twenty years ago, however, when I began my research, the research laboratory at the medical college that I attended was not fully equipped with the equipment and devices that I needed to carry out my experiments, and if a researcher cannot experiment, he cannot write papers. As a result, I followed the advice of my academic supervisor to “create whatever you need yourself” and succeeded in constructing research apparatus on my own. My research topic involved measuring the distribution of and change in the concentration of intracellular calcium within neurite extensions of neurons to be reconstructed in a special environment. I began constructing my equipment after purchasing the components that I needed in Akihabara, Tokyo’s electronic-supply district. For the expensive microscope and sensors, I gathered together the bare minimum of components and somehow managed to construct a device for measuring the concentration of calcium in neurons. In this way, I learned through my own experience that “no one but you can make it happen.”

Incidentally, NTT Basic Research Laboratories, with which we were doing joint research at that time, was equipped with state-of-the-art equipment for the same application as the device that I constructed, so I thought, “Some day, I want to do research at an institution like that.”

Additionally, my experience working in a research laboratory at the University of California San Diego

in the United States continues to influence my research style. In that laboratory guided by a medical doctor, I was able to expand upon my clinical medicine oriented-basic research in neuroscience and generate a certain amount of results.

I was engaged, in particular, in basic research on the treatment of Alzheimer's disease and spinal cord injuries through a combination of gene therapy and stem cell transplants. It was here that I performed a successful experiment in stretching and reconnecting a severed central nerve. In this research, it took about five years from the time of this successful experiment to completing and submitting my paper.

From this experience, I came to realize that it is difficult to treat the neural and cardiac diseases of patients, which is the objective of clinical medicine, solely by using techniques from molecular and cellular biology. I therefore resolved to enter NTT Basic Research Laboratories to search for a new method that could restore neural and cardiac functions and cure those diseases.

NTT Basic Research Laboratories is a site where specialists from a wide variety of fields come together to perform research. The *hitoe* functional material was developed by successfully resolving key issues associated with water resistance, durability, and other characteristics through the collaboration of specialists in the chemistry and engineering fields and also by gaining technological cooperation from the textile manufacturer.

The “Tsukada Way” born of ideas rooted in clinical practice

—Much of your research seems to be based on your diverse experience and to be motivated by a feeling of necessity.

Perhaps I am an unrefined researcher. I think that many researchers employ commercially available equipment and devices and carry out experiments efficiently according to established protocol. I, on the other hand, am the type to prepare the tools that I need in a trial-and-error manner and to carry out experiments in a somewhat probing manner while visualizing what would take place in an actual clinic. For example, I strive to uncover even minor issues that occur with the electrode material and to eliminate each and every problem. At first glance, this may seem to be a roundabout approach, but I consider it to be an important process. Of course, this takes time, and failures can occur frequently, and there are many

tasks that I myself find troublesome, but it only takes one breakthrough to open up great possibilities toward discovering something new.

It is also important that I get as many people as possible to understand what my research is all about. When I make a presentation, I try to prepare exhibits that portray something of what I actually do in my research. Perhaps I do this simply out of necessity, but I think it's very important to “show and tell.”

On the occasion of a recent NTT press release, I was called upon to prepare a communication terminal marked with the *hitoe* logo and to give a measurement demonstration in which a model was asked to wear the terminal. For the people attending this demonstration, seeing a real functioning device in this way stimulates their interest and makes them think about future possibilities. It's important to arouse everyone's imagination!

Acting with a sense of flexibility

—Dr. Tsukada, what kind of mindset should the modern researcher adopt? Can you leave us with some encouraging words for young researchers?

Where do differences between a researcher and a doctor lie? First of all, a doctor works in proximity to patients, so not a day does go by without a doctor sensing what is needed in actual clinical practice. In contrast, a researcher has little opportunity to come into contact with actual users. Based on my past experience, I strive to apply the results I obtain from basic research to as many needs as possible in the real world. In addition, a doctor is obligated to explain specialized concepts in an easy-to-understand manner in order to gain consent for medical treatment. Now that I have become a researcher, I also make it a point to explain specialized technology in a manner that is as easy to understand as possible in terms familiar to the listener. The *hitoe* project has been progressing over a very short period of time, but I am conscious of the need to speed up decision making and actions even further. It is important that possibilities and problem areas be uncovered promptly and that appropriate measures be taken without delay.

Basic research in the 20th century involved clear-cut goals, such as solving major problems or submitting one's paper to high impact journals, and researchers were expected to produce results to achieve those goals.

Now that we have entered the 21st century, major changes appear to be taking place in this regard.

Today, young researchers are being asked to respond flexibly to the demands of the times, such as by taking up the challenge of solving latent problems in society. Structural problems in society lead to situations in which changes are difficult to deal with and things seem to go bad no matter what is done. Nevertheless, even under these conditions, I believe that we can learn something and grow.

Current scientific research and development is becoming highly advanced and compartmentalized. From the user's viewpoint, this is a time in which understanding what front-line researchers do is becoming difficult. Accordingly, while asking young researchers to continue their efforts in front-line, specialized research, I would at the same time ask them to be sensitive to the environment surrounding research today and to respond to major needs in society.

Interviewee profile

■ Career highlights

Shingo Tsukada received the M.D. Diploma and License from Toyama University in 1990, and the Ph.D. degree from the University of Tsukuba in 2003. He joined NTT Basic Research Laboratories as a Research Specialist in 2010 and has been in his present position since 2013. He is engaged in work on human-machine interfaces, including wearable electrodes and their systems, both in vivo and in vitro, especially in the neuroscience field. He is a member of the Japan Neuroscience Society (JNS), the Physiological Society of Japan, the Japanese Circulation Society, and the Society for Neuroscience.

OSSMA: Facilitating Migration to OSS

Tetsuya Harano, Tatsuya Muramoto, and Satoshi Iwasaki

Abstract

The NTT Open Source Software Center promotes the use of open source software (OSS) in order to reduce the total cost of ownership of systems within the NTT Group of companies. It also provides total support related to OSS migration in the form of its OSS Migration Assist (OSSMA) service. This article introduces OSS migration technologies, describes the OSSMA service and its benefits, and gives some application examples.

Keywords: OSS, migration, system development

1. Introduction

At NTT Group companies, efforts to promote the use of open source software (OSS) for in-house operation systems and middleware are taking root in order to reduce the total cost of ownership (TCO) of systems. These Feature Articles introduce some of the achievements of NTT Open Source Software (OSS) Center's technological developments that are aimed at accelerating the use of OSS at NTT Group companies [1]–[7].

2. Background to this initiative

The functionality and quality of OSS such as Linux have increased in recent years. This has been advantageous for companies using OSS for business systems, as it has resulted in reduced system development costs and lower system operation costs by avoiding vendor lock-in.

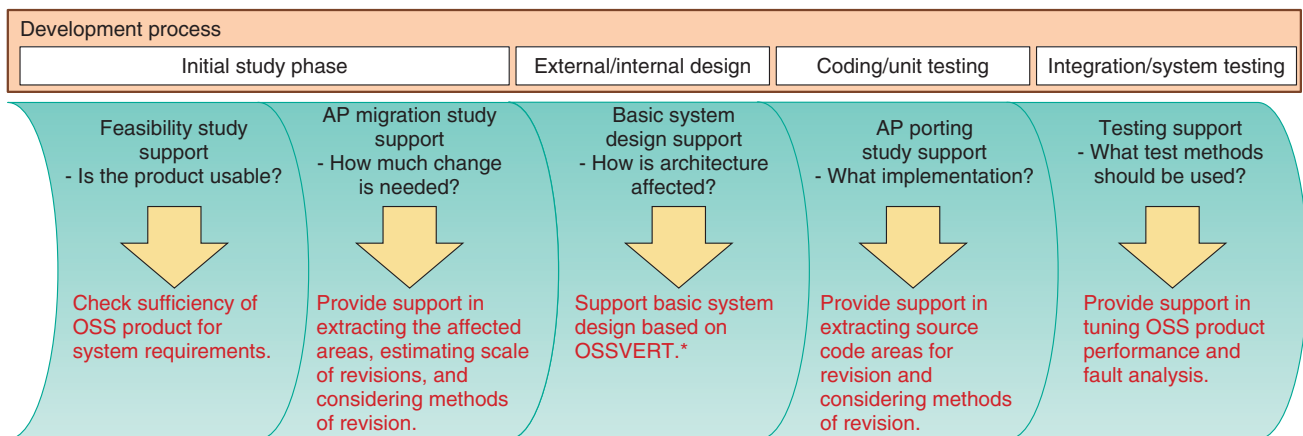
For system development in the NTT Group, we are actively introducing OSS for system upgrades as well as for new systems. Most system development proposals are for updates (over 80% of development proposals the NTT OSS Center is involved in), but most of these cases use particular functions of commercial middleware, or have application program (AP) specifications that depend heavily on such middleware. As a result, a development cost, referred

to as a migration cost, is incurred to eliminate these dependencies. There have been cases where OSS migration had to be abandoned because the migration cost was estimated to be higher than necessary, or the cost of the OSS migration work was higher than expected, and consequently, the TCO was not reduced. Thus, it is important to improve the accuracy of estimating the migration cost and to reduce the actual cost of migration.

To resolve these issues, the NTT OSS Center is accumulating know-how for checking whether OSS functionality and performance are sufficient for system requirements and what the extent of the effect will be on APs when using OSS. It is also providing the OSS Migration Assist (OSSMA) service to provide total support for the system development process.

3. Importance of determining migration feasibility in the initial study phase

The support provided by OSSMA in each development process is shown in **Fig. 1**. The initial study phase is particularly important here. In this phase, the scope of the effect of migrating from a commercial operating system (OS) and middleware to OSS is ascertained, a decision is made as to whether or not OSS can be applied, and if it can, an estimate of the cost of migration work is calculated. Two important



* OSS Suites Verified Technically (OSSVERT): A collection of know-how documents on OSS use, provided by the NTT OSS Center

Fig. 1. Support provided for each development process.

points to ascertain regarding the scope of the effect are described below.

3.1 Checking adaptability to system requirements

Here, we check whether OSS can satisfy the functional and performance requirements of the system. If there will be difficulty satisfying these requirements, alternate solutions for the insufficiencies are considered, and the cost of such solutions is estimated. If these issues cannot be resolved, it will be necessary to forgo the use of OSS.

3.2 Checking the scale of migration

Here, we identify areas of the AP that will be affected by using OSS, study how revisions will be made, and estimate the cost of the development phase of the AP migration (from design through testing). If the AP migration cost is too high or the revisions are too difficult, it will be necessary to forgo the use of OSS.

The method of deciding whether TCO reductions can be gained through OSS migration is shown in **Fig. 2**. As shown in the figure, the TCO for a simple upgrade (upgrading the existing OS and middleware) is compared with that for migration to OSS; if the TCO of the OSS migration is lower, then the OSS migration should be selected. In order to compare TCOs appropriately, it is important to clarify the level of difficulty and the scale of the effect of the AP migration, and to estimate the migration cost in detail.

4. Overview of OSS migration technology and benefits of OSSMA

Generally, the part of the migration cost that is most difficult to estimate is the cost of porting APs, which requires clarifying problems in porting, comprehensively extracting the corresponding sections in the source code, and considering suitable ways to revise them. For example, one area affected by OS migration is *computations involving unions*^{*1}. However, not all unions require revisions; the effects depend on the type of data stored in the union and the method of computation. Revisions are only necessary when APs are affected, and the revisions should be made in such a way that the source code does not depend on differences in the central processing unit (CPU) architecture.

With OSSMA, tools implementing the accumulated know-how of the NTT OSS Center are used to extract areas affected by migration to four products: Linux, JBoss^{*2}, PostgreSQL^{*3}, and Crane^{*4}. Technical issues when porting applications to each of these products are explained below.

4.1 OS

- For shell programs and APs written in the C

*1 Union: A data structure that allows different types of data to be stored in the same memory area.

*2 JBoss: An OSSJava 2 Enterprise Edition (J2EE) application server.

*3 PostgreSQL: An OSS relational database management system (RDBMS).

*4 Crane: Integrated operations management software supporting OSS and developed by NTT laboratories.

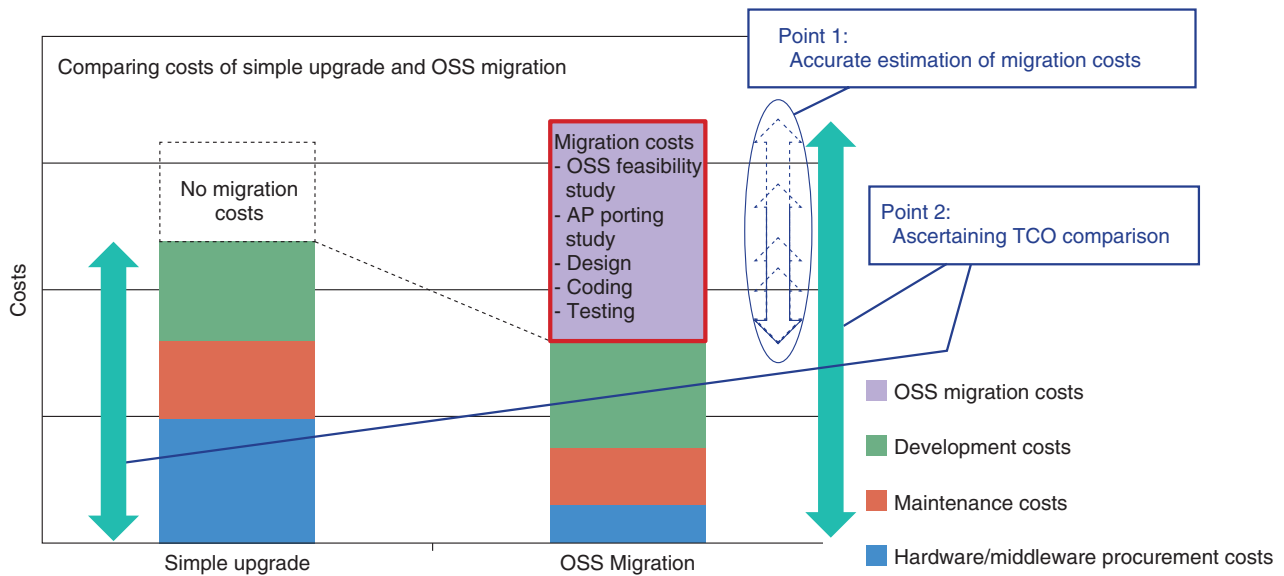


Fig. 2. Determining TCO reduction due to OSS migration.

programming language, differences in OS commands, system call specifications, and memory storage formats for different CPU architectures (big-endian, little-endian, etc.)^{*5} must be considered.

- OSSMA tools conduct over 2000 logic checks in analyzing AP syntax.

4.2 Application server

- Even though the effects of changing products are relatively small because Java is a platform independent language, APs must be revised with respect to upgrading Java itself and differences between any commercial middleware being used and OSS functionality.
- Upgrading Java itself must also be considered when upgrading commercial middleware.
- The TUBAME^{*6} OSS tool can be used to identify the extent of these effects.

4.3 DBMS

- Structured Query Language (SQL) used in C, Java APs and shell scripts, as well as database definitions, must be revised where they use proprietary features (non-standard SQL functions, preprocessors, etc.) of a commercial database management system (DBMS).
- The db_syntax_diff^{*7} OSS tool can be used to identify the extent of these effects; the tool checks over 1400 points in analyzing SQL

logic.

- The business data stored in the database must also be migrated.

4.4 Operations management

- Parameter definitions used in a jobnet^{*8} of commercial operations management software need to be revised in order to use the jobnet functions of Crane.
- Revising the jobnet means that operational procedures change, so maintenance workflows and procedures must be reviewed as well.

Using OSSMA is beneficial in that the parts of APs affected can be checked quickly during the initial study phase, so the upgrade plan can be decided quickly. The areas affected and how to revise them can also be clarified in the initial study phase, which

^{*5} x86/x86-64 is little-endian (data are written to memory starting with the least-significant byte), SPARC is big-endian (data are written to memory starting with the most significant byte), and PA-RISC and PowerPC are bi-endian (able to do either). Since there are differences in how data can be stored in memory, programs that perform memory operations must behave differently for big-endian and little-endian architectures.

^{*6} TUBAME: An OSS tool available at the following URL: <https://github.com/TUBAME/migration-tool>

^{*7} db_syntax_diff: An OSS tool available at the following URL: https://github.com/db-syntax-diff/db_syntax_diff

^{*8} jobnet: Defines how and in what order the jobs should be executed so that multiple jobs (processes) can be run in sequence.

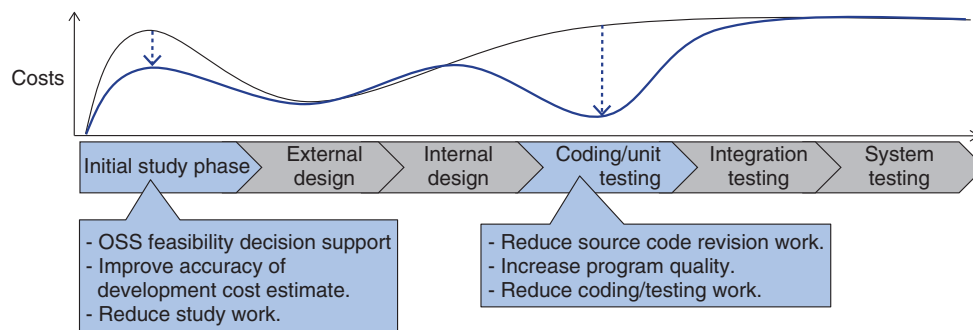


Fig. 3. Benefits of using OSSMA in the development process.

provides the added benefit of reducing the cost of coding and unit testing during the production and testing phases. The benefits of using OSSMA over the entire development process are shown in Fig. 3.

5. OSSMA application example

Here, we give an example of using OSSMA when updating a business support system (System A) in an NTT Group company. System A was developed during the period from May to December in 2012 and has been provided commercially since January 2013. The software architecture of System A before and after updating is shown in Fig. 4. The OS and main software applications were switched from commercial products to OSS. Two points of concern were identified while studying the system's applicability during the initial study phase:

- (1) PostgreSQL may not meet the performance requirements.
- (2) The AP revision cost for PostgreSQL is high, so the benefit of reducing TCO may not be attainable.

The concerns raised in the initial study phase had many significant effects on the basic system design, so at the NTT OSS Center, we placed particular emphasis on them in our consultations. For this project, we handled these two issues as follows.

5.1 Issue (1)

The pre-update system used a partitioning function^{*9} of the commercial DBMS in order to increase processing performance. There was a risk that performance requirements would not be met when porting this function to PostgreSQL (with partitioning), so we prepared a test environment with the same specifications as System A and tested the performance

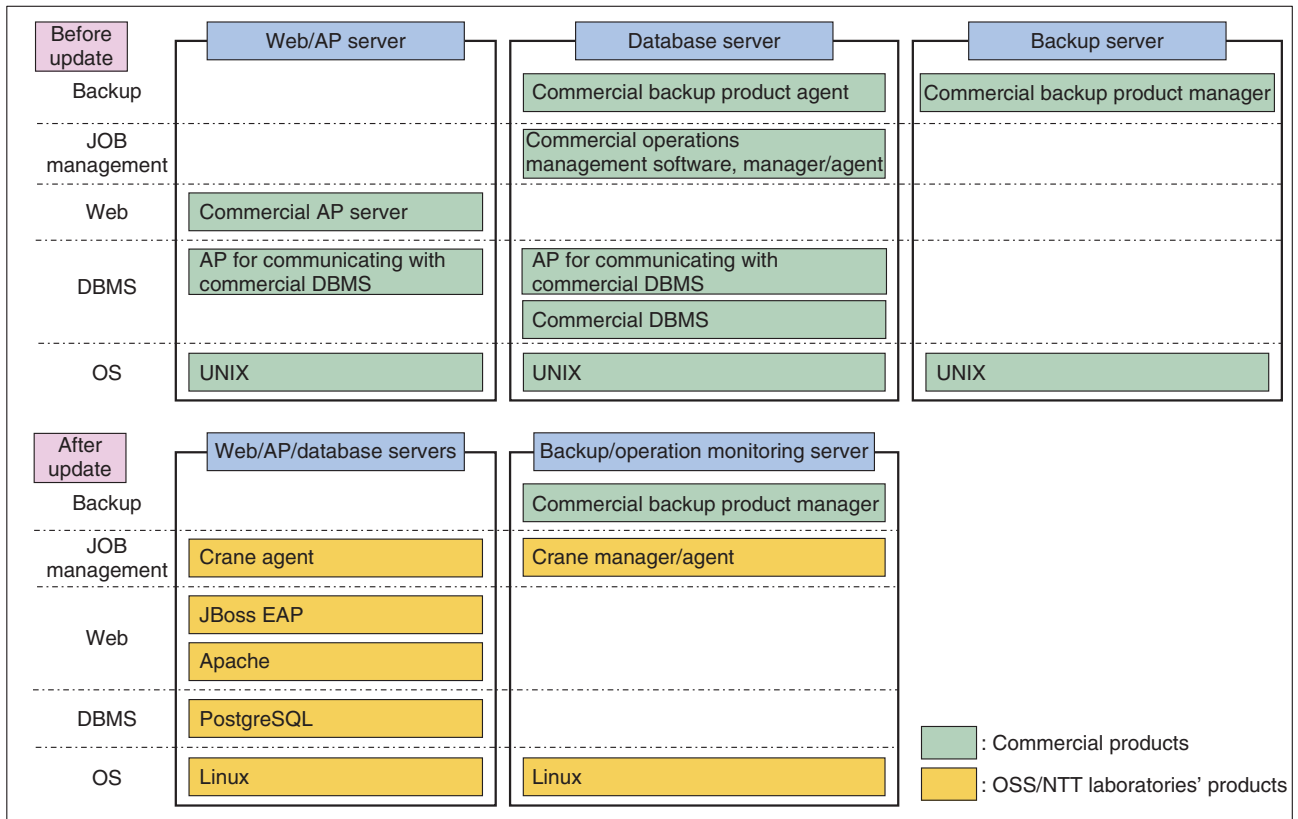
with respect to partitioning. The results of these tests are shown in Fig. 5. The figure shows that System A performance requirements will be met with PostgreSQL, eliminating the concern regarding the development project. It also shows that performance requirements will be met with PostgreSQL without partitioning, which contributes to development efficiency by simplifying the system design.

5.2 Issue (2)

We obtained the source code for the existing system and used `db_syntax_diff` to extract the affected sections of the APs and estimate the scale of revisions required. The results of studying the portability are shown in Figs. 6(a) and (b). The scale of the revisions are shown in Fig. 6(a), which indicates that the source code has a scale of 250 kL (kilo-lines) of code, and the required revisions are on a scale of 9 kL (just under 4% of the source code). The results of evaluating the level of difficulty of the revisions are in Fig. 6(b). Out of the 9 kL to be revised, more than 99% have a difficulty level of Low 1 or Low 2. (After the development was completed, we received feedback from the project team members who said that these results were very thorough and extremely useful.)

The migration cost of the project was estimated based on the above results. The project team recognized that TCO would be reduced sufficiently, so they decided to use PostgreSQL. We continued to provide technical support throughout the entire development process; we supported tasks such as updating files, which reduced any uncertainty the team members had about the development project.

*9 Partitioning function: A function that makes it easier to handle huge amounts of data by partitioning it into multiple tables in a way that is transparent to the AP.



EAP: enterprise application platform

Fig. 4. System A software architecture before and after update.

No.	Task type	Task	Performance target value	Result (with partitioning)	Result (without partitioning)
1	Online	Task A	4.9 s or less	○ 4.69 s	○ 2.98 s
2	Online	Task B	0.1 s or less	○ 0.095 s	○ 0.08 s
3	Online	Task C	3.8 s or less	○ 3.20 s	○ 0.86 s
4	Batch	Task D	736 s or less	○ 336 s	○ 284 s
5	Batch	Task E	850 s or less	○ 787 s	○ 767 s

Test conditions

- Hardware
 - CPU Intel Xeon 2.27 GHz × 4core
 - MEM 24 GB
 - Storage 146 GB × 10 (VRAID10)
- Software
 - RHEL 5.4
 - PostgreSQL9.2 (16 partitions)

RHEL: Red Had Enterprise Linux

Fig. 5. Results of System A SQL task PostgreSQL performance test.

Area studied	No. of source files	Lines of source code	No. of changes	Lines requiring visual check	Unique lines to be changed
SQL/DDDL	Approx. 740	Approx. 250 kL	Approx. 11 kL	Approx. 5 kL	Approx. 9 kL
Java					
Pro*C					

(a) System A revision scope report

No.	Degree of difficulty	SQL/DDDL	Java	Pro*C
1	Low 1	4186	1	11
2	Low 2	5361	798	94
3	Medium	20	0	0
4	High	0	0	0
Total		9567	799	105

Degree of difficulty levels:
 - Low 1: Can be fixed automatically
 - Low 2: Very easy, but cannot be fixed automatically
 - Medium: Cannot be fixed automatically, and surrounding processes must be checked
 - High: Cannot be fixed automatically, and PostgreSQL has no substitute function, so study is required.

(b) Degree-of-difficulty report for System A

DDL: data definition language

Fig. 6. Results of migration study.

6. Future development

We will continue to work to expand the scope of migration products and the accuracy in identifying areas in APs that will be affected so that OSSMA can be used more widely. We are actively making improvements to create a service that is easier to use based on feedback from actual projects and customer requests. We will continue to work from a customer perspective with OSSMA and other services, so that NTT OSS Center activities contribute to further increasing the competitive strength of NTT.

References

- [1] S. Takao, Y. Suenaga, Y. Kubota, H. Waki, and M. Nagafusa, "Heap-Stats: Your Dependable Helper for Java Applications, from Development to Operation," NTT Technical Review, Vol. 12, No. 6, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201406fa2.html>
- [2] M. Urano, K. Nakamura, and M. Nagafusa, "TUBAME: A Tool for Studying Java Application Migration," NTT Technical Review, Vol. 12, No. 6, 2014.
- [3] S. Suzuki, H. Toyoda, R. Sugahara, H. Yamada, and H. Tezuka, "Faster System Building with No Reworking! OSSVERT Smart Builder," NTT Technical Review, Vol. 12, No. 6, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201406fa4.html>
- [4] K. Fujioka, Y. Takeshita, K. Mori, T. Tanaka, K. Higashi, and K. Mii, "Pacemaker: Increasing System Reliability," NTT Technical Review, Vol. 12, No. 6, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201406fa5.html>
- [5] A. Kawada, G. Nishimura, J. Teramoto, E. Fujita, T. Yamamuro, and T. Wakamori, "PostgreSQL: Leader in Cost Reduction," NTT Technical Review, Vol. 12, No. 6, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201406fa6.html>
- [6] K. Yoshida, K. Goto, T. Makita, T. Nakazawa, and S. Kihara, "Linux and KVM Activities Providing Foundational Support for Use of OSS," NTT Technical Review, Vol. 12, No. 6, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201406fa7.html>
- [7] Y. Suenaga, "Dramatic Changes in Java Troubleshooting! — Pursuing Uncompromising Low-overhead Implementations," NTT Technical Review, Vol. 12, No. 6, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201406fa8.html>



Tetsuya Harano

Manager, OSS Promotion Unit, NTT Open Source Software Center.

He received the B.S. in chemistry from Hiroshima University in 1994. He joined NTT in 1994. He moved to NTT Open Source Software Center in 2011, where he has been working as a system engineer. He provides technical support in the system development phase and carries out troubleshooting in the system operation phase to achieve a smooth introduction of OSS systems.



Satoshi Iwasaki

Unit Leader, OSS Promotion Unit, NTT Open Source Software Center.

He received the B.E. and M.E. in radio communication engineering from the University of Electro-Communications, Tokyo, in 1986 and 1988, respectively. He joined NTT in 1988 and engaged in R&D of facsimile communication network services. From 2001 to 2009, he worked in the IT Operation System and Service Department at NTT Communications. He moved to NTT Open Source Software Center in 2009. He is a member of the Institute of Electronics, Information and Communication Engineers.



Tatsuya Muramoto

Leader, OSS Promotion Unit, NTT Open Source Software Center.

He received the B.E. and M.E. in social system engineering from the University of Tsukuba, Ibaraki, in 1994 and 1996, respectively. He joined NTT Information and Communication Systems Laboratories in 1996 and moved to NTT Open Source Software Center in 2007. He is interested in the cost reduction effect of using OSS and is currently engaged in promoting OSS products.

HeapStats: Your Dependable Helper for Java Applications, from Development to Operation

Shinji Takao, Yasumasa Suenaga, Yuji Kubota, Hiroaki Waki, and Masahiro Nagafusa

Abstract

HeapStats is an open source software monitoring and analysis tool developed by the NTT Open Source Software Center. HeapStats enables people to monitor detailed runtime information about the heap memory (an area of memory) managed by the Java Virtual Machine with minimal effect on the performance of the Java application. HeapStats helps to resolve problems rapidly when debugging during development, testing, and operation. This article introduces HeapStats and its strengths, application scenarios, and examples of analysis using it.

Keywords: Java, fault analysis, system monitoring

1. Introduction

Part of our technical support related to Java^{*1} at the NTT Open Source Software (OSS) Center involves handling fault analysis requests from customers. However, for some of these requests, we cannot obtain enough data when the fault occurs to immediately investigate or analyze it. In such cases, we ask the client to recapture the data, although sometimes, such as when the fault occurs infrequently, it can take a very long time to reproduce the fault and obtain the data. In conventional analysis, data such as a heap dump^{*2} are retrieved. However, this can also place a heavy load on the system, and the output files can be very large, so it is often difficult to obtain a heap dump from an operational system. When it cannot be obtained, a text file called a class histogram must be obtained instead. In such cases, even more manual work is required for analysis, and the information that can be obtained is limited (**Fig. 1(a)**).

For these reasons, at the NTT OSS Center, we have developed a Java Virtual Machine (JVM) monitoring and analysis tool called *HeapStats*, with the main goal of enabling rapid analysis when a problem in a

Java application occurs that causes a fault such as insufficient heap memory.

HeapStats operates with low overhead, so it has minimal effect on the performance of the Java application. Moreover, it can continuously obtain detailed information on the internal heap memory state such as heap memory usage by class and reference relationships among classes, and it can analyze faults and display the results graphically. These capabilities enable it to continuously obtain sufficient information to analyze faults and immediately respond when problems occur suddenly, even on an operational system (**Fig. 1(b)**).

We have published HeapStats as open source software (OSS) in a community called IcedTea. Our goal is to have it used in many projects, obtain feedback from users, and make further improvements [1]. IcedTea is a development community for the OpenJDK package used by many Linux distributors, with

*1 Java is a registered trademark of Oracle Corporation and/or its affiliates in the USA and other countries.

*2 Heap dump: An output file of the contents of heap memory managed by the Java Virtual Machine.

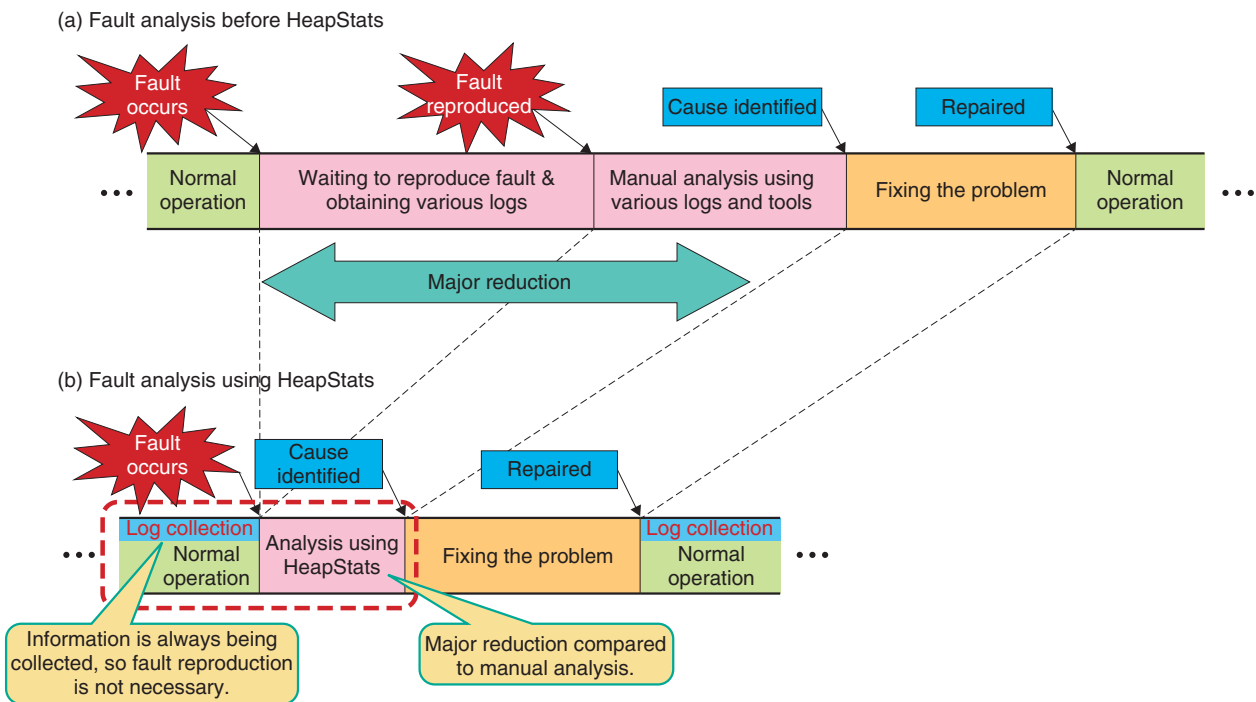


Fig. 1. Fault analysis using HeapStats.

participation by many developers and users. Currently, the HeapStats project on IcedTea is operated mainly by three committers^{*3}, who are from the NTT OSS Center.

2. HeapStats overview

HeapStats is composed of two programs: the JVM monitoring agent (agent) and the analyzer (Fig. 2). The agent gathers the information needed to analyze faults. It can be invoked easily by adding a start-up option when starting a Java process, and it continuously monitors factors such as heap memory usage and occurrence of deadlocks. It can also alert other operation monitoring tools using Simple Network Management Protocol (SNMP) traps, so it can also be used as a simple monitoring tool. The analyzer is an application that provides a GUI (graphical user interface) displaying the various kinds of JVM information obtained by the agent, and also provides analysis support.

2.1 Agent

The agent collects information related to heap memory. It was built conforming to the JVM garbage collection (GC) implementation as described below

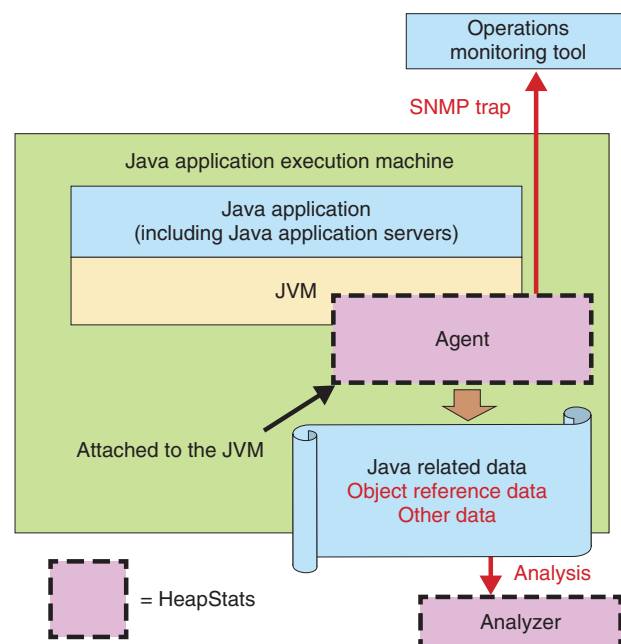


Fig. 2. HeapStats system architecture.

*3 Committer: A manager of an OSS project with the right to update the source code repository.

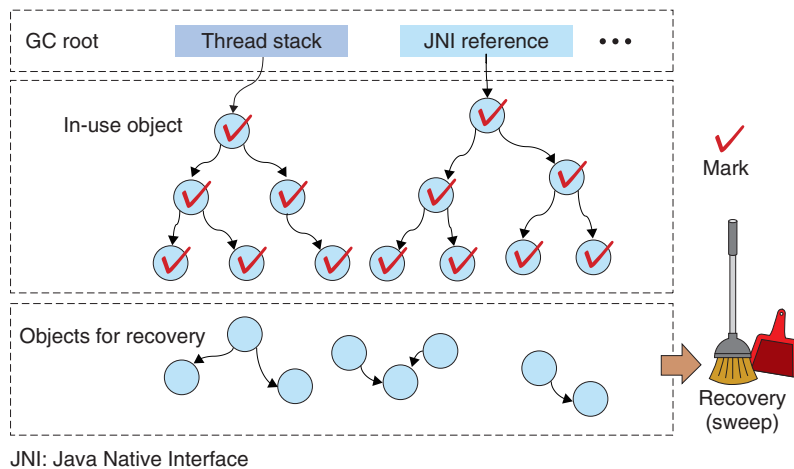


Fig. 3. Applying GC marks.

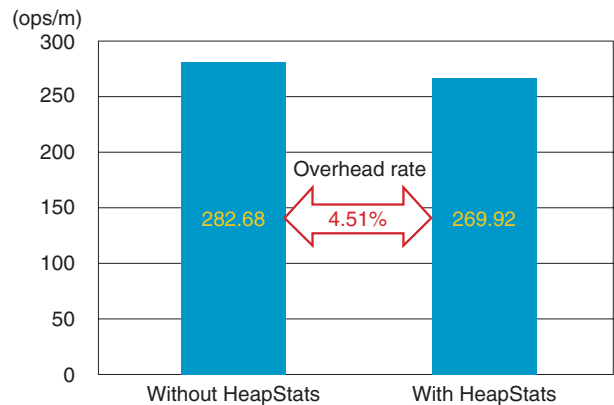
in order to minimize the overhead for the application being monitored.

When the GC is recovering heap memory from Java objects that are no longer in use, it looks for and marks objects still in use in order to prevent recovery of memory still in use (Fig. 3). The agent interposes a process into the function performing this marking process to obtain a variety of information, so the agent and the GC marking process operate together. To eliminate application programming interface (API) call overhead in doing so, it obtains data by directly referencing the heap memory data addresses needed.

The information gathered by the agent is narrowed down to what is needed for analysis, which keeps the volume of data small compared to that of a heap dump. The information obtained for each GC is output to a log (a snapshot file) immediately, so there is no impact on memory.

The agent uses these techniques to obtain the data with low overhead. We used SPECjvm2008 [2], a standard benchmark for measuring the performance of Java runtime environments, to compare scores when using and not using HeapStats. It showed that the overhead was kept to 4.51%*⁴ (Fig. 4). This allows information to be collected continuously, even on an operating system, which is not possible when using heap dumps.

The agent also gathers a variety of information regarding aspects other than heap memory, for example, server resources. In addition to the information normally gathered (Fig. 5(a)), the agent gathers even more information when an out-of-memory error



* Measurement environment
 - Benchmark tool: SPECjvm2008 1.01
 - Machine: DELL PowerEdge R810 (Xeon X7542, 32 GB memory)
 - OS: Red Hat Enterprise Linux Server release 6.3 x86_64
 - Java: java-1.7.0-openjdk-1.7.0.25-2.3.10.4.el6_4.x86_64
 - Java execution options: -Xms4500m -Xmx4500m -XX:+UseG1GC
 -agentpath: <agent library>

Fig. 4. HeapStats overhead rate (SPECjvm2008 Composite Result).

(OOM) or deadlock occurs in the JVM (Fig. 5(b)). The agent runs on x86 and x86_64 architectures running Linux and Java SE6 or greater, and installation packages are available for Red Hat Enterprise Linux and others.

2.2 Analyzer

The analyzer displays information related to the

*⁴ Differs depending on application and machine environment.

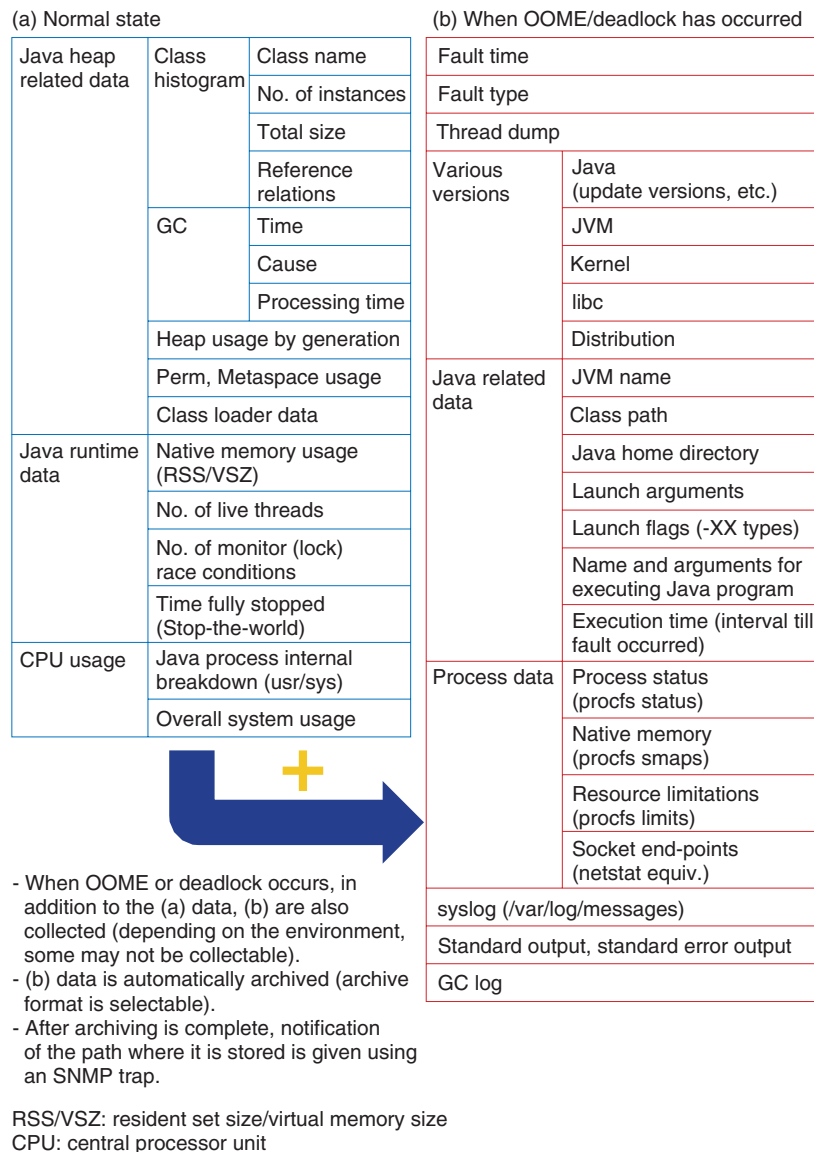


Fig. 5. HeapStats data collection items.

snapshot files and server resources collected by the agent. In particular, the information recorded in snapshot files is displayed graphically to aid analysis. A screen with the time sequence of heap memory usage information and the per-class number of objects and heap memory usage are shown in **Fig. 6(a)**. A screen graphically showing the referential relationships for a class object (what classes reference an object of a given class)^{*5} is indicated in **Fig. 6(b)**. This is often a large amount of information and can be difficult to analyze using conventional heap-dump analysis tools. To simplify the analysis, the analyzer provides

functions to narrow the results by a particular class name or to sort them by the number of objects or the heap memory usage per class.

A concrete example of analysis using the analyzer is described below.

3. HeapStats application scenario

Testing and debugging related to heap memory can

*5 A display of class reference relationships is available in HeapStats 1.1.0 or later.

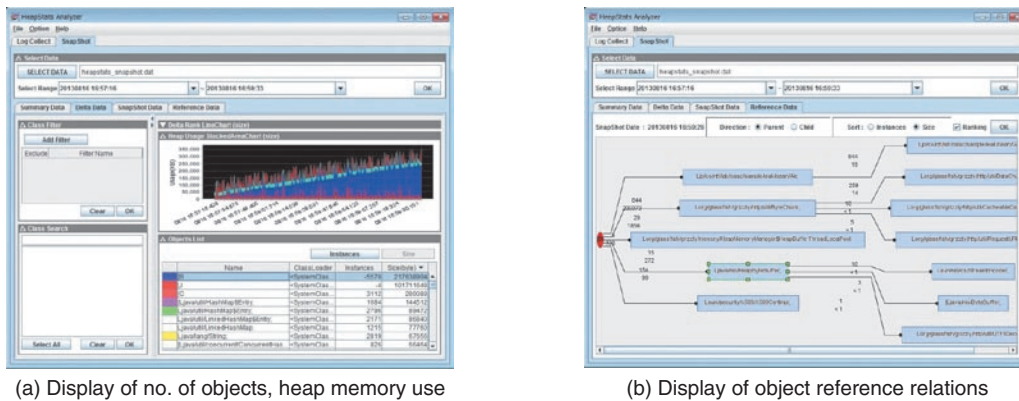


Fig. 6. HeapStats analysis screen.

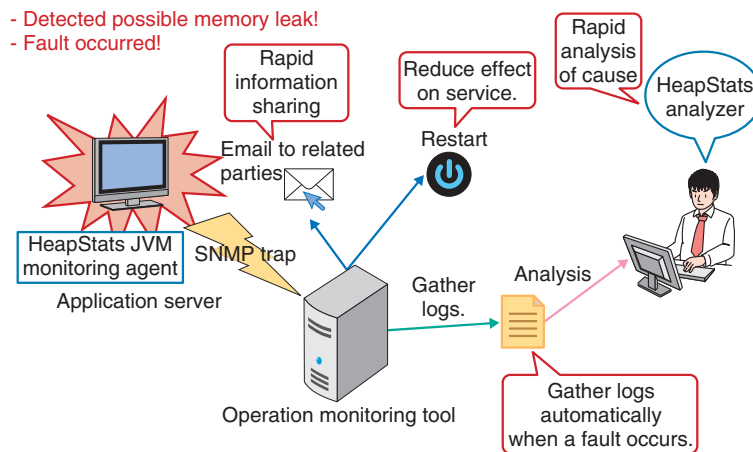


Fig. 7. Operation monitoring tool and how it works.

be done more efficiently by using HeapStats starting with the development stage. Java programs entrust the management of heap memory to the JVM, so they do not have to explicitly release memory. However, if a program unintentionally maintains references to some objects, the JVM cannot release the memory, and the heap memory usage increases, which can lead to running out of free space in heap memory (memory leaks). The heap memory analysis functions of HeapStats are useful for detecting memory leaks early.

Normally, memory leaks are discovered during development testing, but if they progress slowly over long periods of time or are triggered by a particular operation, they could manifest during operation as well. Insufficient heap memory can also occur even

without a memory leak, such as when the amount of processing exceeds that anticipated during design, or when processing requiring a large amount of heap memory occurs. These can be related to the number of users or an increase in accumulated data, so they can occur after operating for a certain amount of time. Thus, even programs with adequate debugging and testing before release must be monitored for problems that could arise during operation, so HeapStats can be effective in the operational stages as well.

A scenario using HeapStats linked to operations monitoring tools through SNMP is shown in **Fig. 7**. Notifications that predict faults related to the detailed state of heap memory—such as particular classes consuming large amounts of heap memory—are sent to related parties, which enables a rapid response

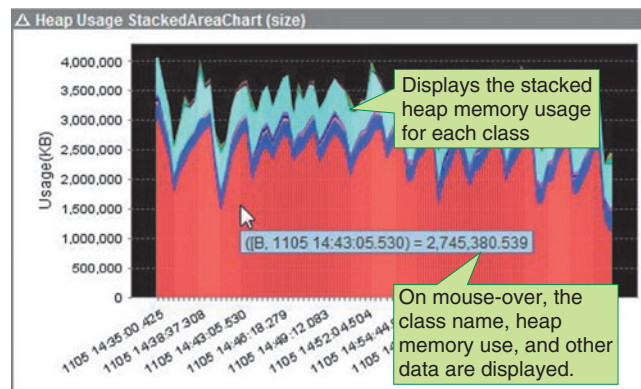


Fig. 8. Stacked area chart.

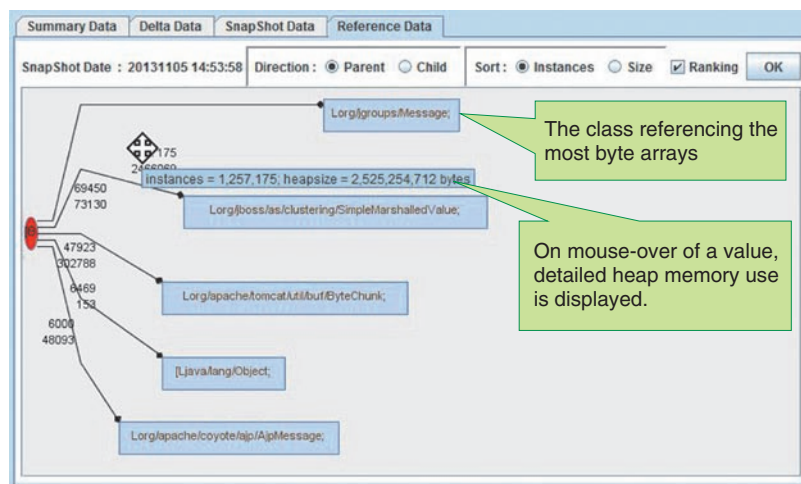


Fig. 9. Class reference relation graph.

when such faults occur.

4. Analysis example

We next introduce an example of a defect occurring during testing at the NTT OSS Center, which we were able to locate quickly. During testing of a system into which HeapStats had been installed, Major GC^{*6} events began occurring frequently, causing performance to drop. The heap memory usage stacked area chart (Fig. 8) showed that the program was consistently using quite a large amount of memory, and most of it was allocated to byte arrays. Next, we displayed the class reference relation diagram (Fig. 9) and identified that the class for messages exchanged between application servers for application server

redundancy (clustering) was using a large number of byte arrays. From that, we focused our investigation on tuning the settings related to clustering and were able to quickly identify the cause. If we had not used HeapStats, it would have been difficult to understand the reference relations between classes, and we would not have known why byte arrays were being used. Consequently, it would probably have taken much longer to identify the cause.

5. Future development

In the future, we will promote HeapStats as an

*6 Major GC: A process in which the JVM performs recovery over the entire heap memory.

effective analysis tool that can be used to solve problems during development, testing, and operation by introducing it into projects, and to contribute to reducing the total cost of operations. We also hope to add functionality based on user feedback, and to make further quality and performance improvements. Moreover, we would like to integrate it with other

OSS analysis tools.

References

- [1] HeapStats, <http://icedtea.classpath.org/wiki/HeapStats>
- [2] Standard Performance Evaluation Corporation, <http://www.spec.org/jvm2008/>



Shinji Takao

Senior Expert, Application Software Technology Unit, NTT Open Source Software Center.

He received the M.M.G. and the Ph.D. in engineering from Keio University, Tokyo, in 1996 and 2008, respectively. He joined the NTT Group in 1996. He is a committer of IcedTea (HeapStats).



Hiroaki Waki

Senior Manager, Application Software Technology Unit, NTT Open Source Software Center.

He received the M.E. in information engineering from Toyohashi University of Technology, Aichi, in 1991. He joined the NTT Group in 1991.



Yasumasa Suenaga

Senior Expert, OSS Promotion Unit, NTT Open Source Software Center.

He received the B.E. in information and communication engineering from Tokyo Denki University, in 2006. He joined the NTT Group in 2006. He is a committer of IcedTea (HeapStats), and an author of the JDK9 development community. In April 2014, he moved to NTT Comware.



Masahiro Nagafusa

Leader, Application Software Technology Unit, NTT Open Source Software Center.

He received the B.L.S. from the University of Library and Information Science (currently, the School of Informatics at the University of Tsukuba, Ibaraki), in 1989. He joined the NTT Group in 1989. He manages the support operations of more than 15 web/OSS products including JBoss.



Yuji Kubota

Expert, Application Software Technology Unit, NTT Open Source Software Center.

He received the M.S. in informatics, Kyoto University in 2009. He joined the NTT Group in 2009. He is a committer of IcedTea (HeapStats).

TUBAME: A Tool for Studying Java Application Migration

Masato Urano, Kyohei Nakamura, and Masahiro Nagafusa

Abstract

The NTT Open Source Software Center is developing a tool that automatically identifies sections of applications needing revision and computes the cost of making the revisions when migrating a Java application, and has made it available as open source software (OSS). This tool reduces the amount of manual work required to perform the migration and supports a quantitative study of the effects of migration on Java applications. This article gives an overview of the tool and its strengths and illustrates its effectiveness through an application example.

Keywords: OSS, migration, TUBAME

1. Introduction

The NTT Open Source Software (OSS) Center is actively promoting migration from commercial application servers (AP servers) such as WebLogic to open source software (OSS) AP servers such as JBoss^{*1} [1] in order to reduce total cost of ownership (TCO). Various processes are needed to perform a migration, but the first process, the preliminary investigation, is essential for the overall project to run smoothly (**Fig. 1**). This is because the existing Java application operating on the commercial AP server uses an application programming interface (API) specific to that AP server, and configuration files are defined in a way specific to that AP server, so these sections of applications must be revised during the migration. After investigating the number and location of such sections, the results are used in performing the coding work. This investigation must be done methodically and involves quite a lot of work when done manually, so the probability of human error (omissions and errors) increases as the number of these sections increases. This has resulted in increasing demand for a tool that automatically investigates dependencies, such as selecting sections for revision and computing the revision costs.

The NTT OSS Center provides support services for

this sort of migration. The service includes summarizing the sections in the source code and configuration files of an existing Java application that needs revision when performing a migration and estimating the cost of making the revisions. We also systematically summarize the knowledge accumulated from supporting projects in documents and share them internally as guidelines, which have increased the efficiency of this work. Using these guidelines to study a migration has enabled us to conduct more accurate preliminary investigations. Because of this, we began developing a tool that incorporates functions for both dependency analysis and degree of difficulty analysis that uses our knowledge base.

2. The TUBAME migration support tool

The TUBAME Java application migration support tool was developed based on the background described above. This tool reduces the amount of work involved in migrating an existing Java application to another AP server by automatically extracting and presenting information regarding the cost of revisions needed and how the revisions need to be made. It also confirms

^{*1} JBoss: An OSS Java Platform Enterprise Edition (JavaEE) application server.

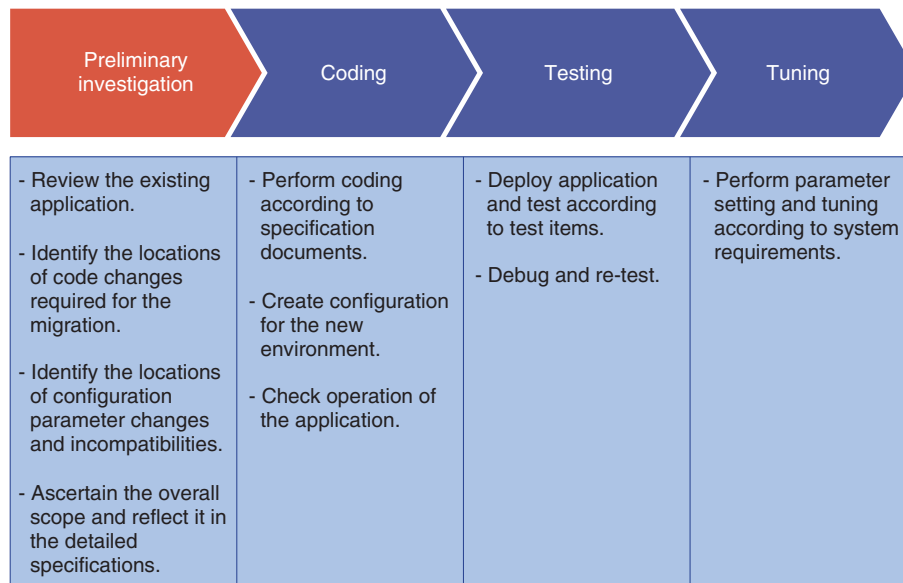


Fig. 1. Migration process.

how revisions should be made for each section to be revised, so the actual work of porting the Java application is more efficient. The tool has been made available as OSS on a community site [2] on GitHub^{*2} since October 2013.

3. Operation overview

Here, we describe the operation of TUBAME.

3.1 Overall organization

TUBAME has two main components: the dependency analysis tool and the knowledge search tool (Fig. 2). Each tool can be used independently, and both can be run to check the effects of migrating a Java application. The knowledge search tool also incorporates a knowledge management tool, which provides an intuitive GUI (graphical user interface) enabling users to compile the knowledge necessary for a search.

3.2 Dependency analysis tool organization

This tool takes the application source code and configuration files as input and automatically extracts the Java packages and classes that cannot be used on the AP server after migration and displays the results (Fig. 3). By ascertaining the propensity to use code that is strongly dependent on the current AP server in this way, the user can examine the degree of difficulty

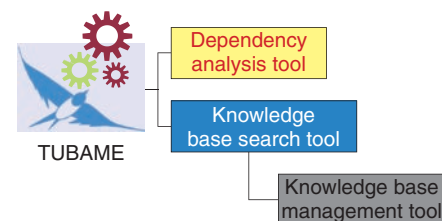


Fig. 2. Overall architecture of TUBAME.

and scope for revising the Java application in concrete terms. The tool can also analyze the effects of multiple migrations such as to different AP servers and to a different framework at the same time. For example, when the AP server is being migrated to JBoss, and the application framework is being upgraded to Struts^{*3} [3], decisions can be made taking the overall effects into consideration. If the scope of revisions for the framework is greater than expected, priority could be given to the JBoss migration first, and the framework upgrade could be deferred to the next development cycle. The dependency results mentioned earlier can also be saved as comma

*2 GitHub: A web sharing service for software development projects.

*3 Struts: A Web application development framework using the Java programming language.

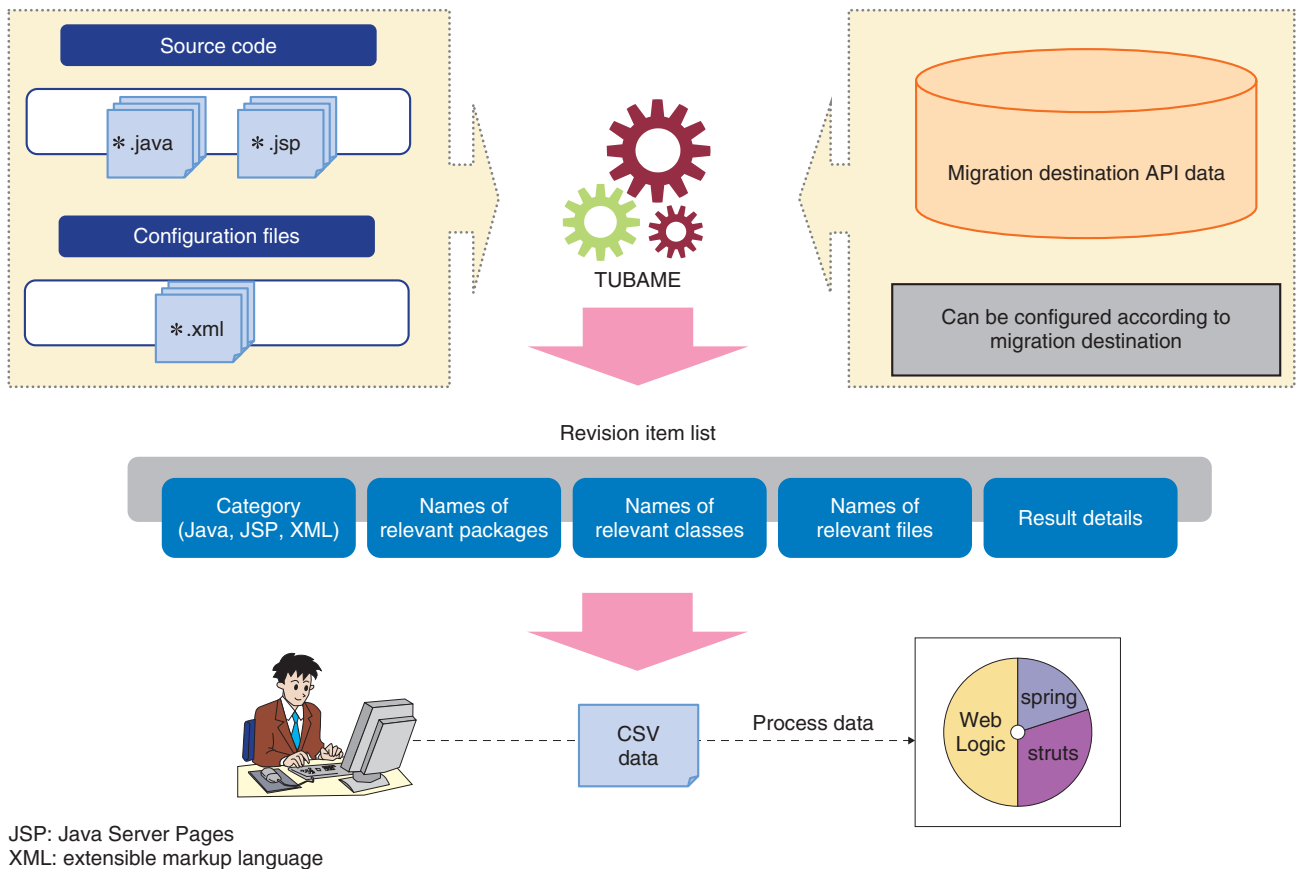


Fig. 3. Organization of dependency analysis tool.

separated value (CSV) data, making it easier to create graphs and other reports showing the proportion of Java packages or classes needing revision.

3.3 Knowledge-base search tool organization

This tool extracts sections for revision based on migration knowledge, which cannot be extracted using the dependency analysis tool described above, and presents revision methods (Fig. 4). For example, the formats of configuration files for different AP servers can be very different, so it is important to know beforehand whether a similar configuration method will be available after the migration. If it will, the porting will be easier, but if not, possible alternative approaches must be studied, which may make the porting more difficult. Thus, it is necessary to know the actual method to be used and to compute the cost. The knowledge base is composed of such search methods and keywords for finding the sections needing revision. The tool uses this knowledge to extract the sections requiring revision from source code and

configuration files automatically and presents the level of difficulty and the cost of revisions for each section. As with the dependency analysis tool discussed earlier, this information can be saved as CSV data, so data on sections needing revision can be aggregated, and the total cost of revisions can be estimated.

4. TUBAME features

A major strength of TUBAME is its generality; the tool and the knowledge are independent, so users can create their own knowledge and customize it appropriately. Moreover, it can support users who would like to study a variety of migrations. For example, the configuration and other aspects of different versions of JBoss can vary quite a bit. In one case, when an application that worked with an earlier JBoss version was migrated to the latest version, the application worked without major revisions but did not deliver the performance of the earlier version under heavy

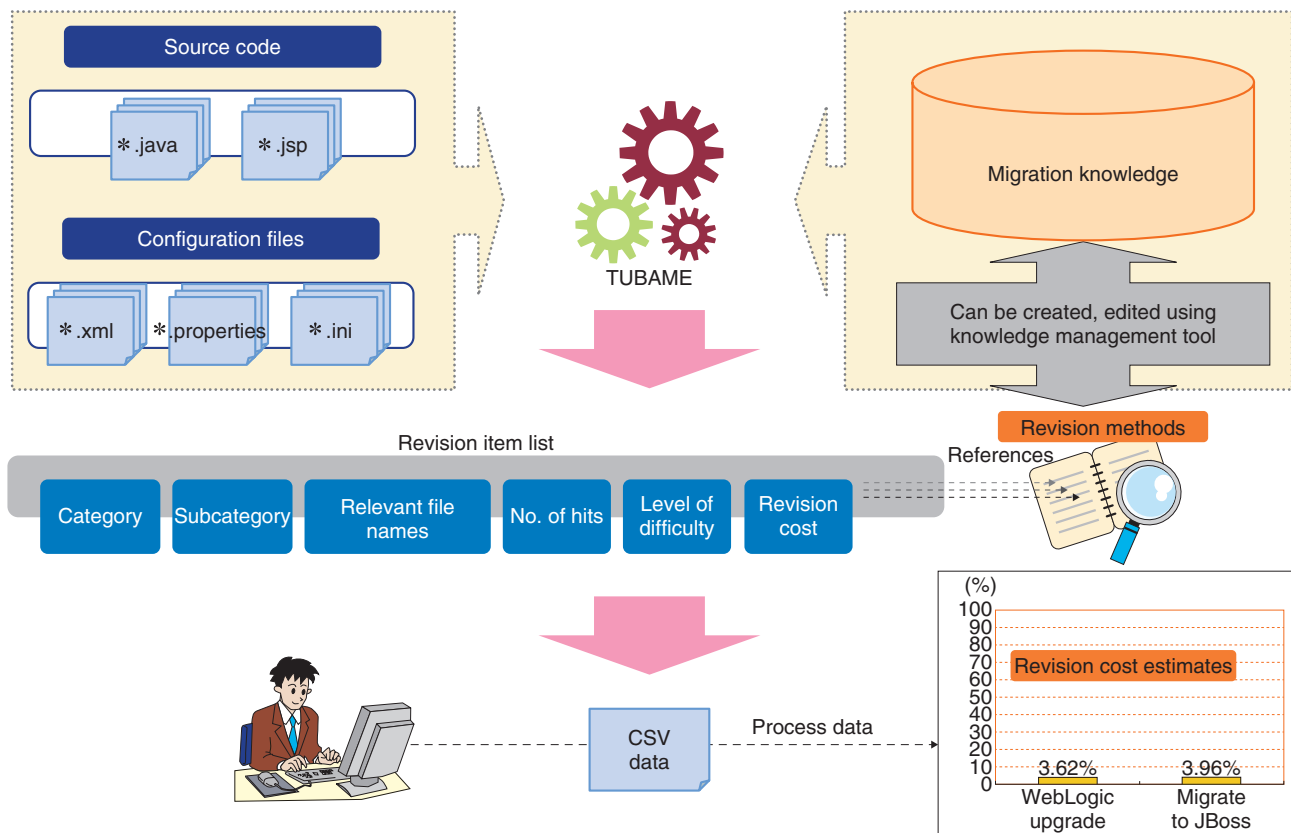


Fig. 4. Organization of knowledge base search tool.

load. This was because the default size of the instance pool*⁴ had changed. This provided a lesson that testing can go more quickly if default values are set to those of the previous version. We are adding such lessons to our knowledge regarding upgrades (migration) to new JBoss versions.

We have also added mechanisms that store API data used with the dependency analysis tool in the past, which can be reused when investigating similar AP server migrations in the future, so that the knowledge base search tool can check the sections needing revision and the methods for making the revisions.

5. Target users and usage

We expect that TUBAME will be used for projects involving the actual migration or possible migration of Java applications. It operates as a plug-in for Eclipse [4], an integrated development environment widely used around the world for development of Java applications, so it can be installed easily, even on compact, lightweight work machines.

6. Example application and effects

We present here the results of comparing the amount of work that is necessary to investigate a migration using TUBAME with that using conventional manual techniques (Fig. 5). A comparison of the work done by the NTT OSS Center for migration support services using the tool and conventional techniques is shown in Fig. 5 and shows an average reduction of approximately 40% by introducing the tool. Currently, the tool has been used to investigate more than ten migrations. These projects vary in the amount of source code by a factor of approximately 20 from the smallest to the largest. When working manually, the amount of time required is proportional to the scale of the source code, but with the tool, revision sections of even a relatively large Java application of over two million lines of code can be extracted in less than three hours, and this effect increases as the scale increases (Fig. 6). The tool can also be used

*⁴ Instance pool: An entity that pools a given data set in memory.

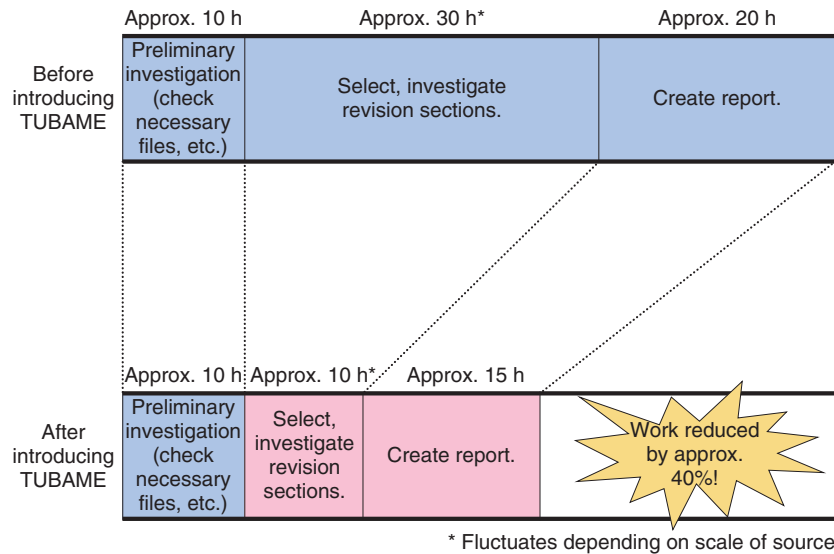


Fig. 5. Amount of work required using TUBAME and conventional techniques.

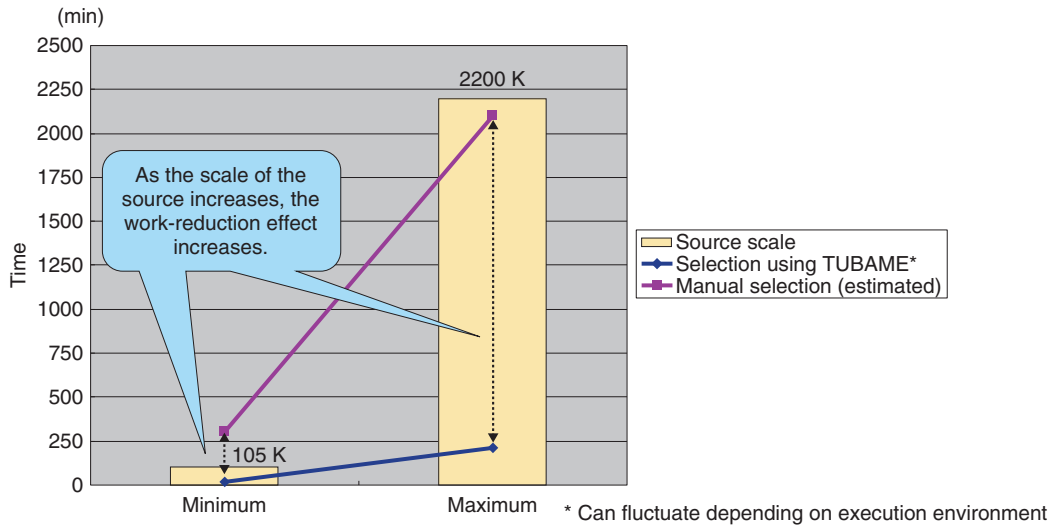


Fig. 6. Relation between revision ratio and selection time.

by people who are not migration experts and enables them to partition the work; thus, they can conduct a rough analysis first and have a specialist perform a more detailed analysis later.

7. Community activity

Currently in the TUBAME community, two knowledge bases that can be downloaded and used with the tool have been published: one for migrating from

WebLogic to JBoss and another for upgrading JBoss. We plan to continue to update the migration knowledge mainly for migration to JBoss OSS. We can also receive requests to add or revise knowledge discovered by users in this community. We will investigate such user requests and periodically update the knowledge to increase its accuracy. We are also considering expanding the knowledge to add information regarding migration to JBoss from commercial AP servers besides WebLogic (WebSphere etc.) to our current

knowledge, in cooperation with engineers in the JBoss community.

8. Future development

The NTT OSS Center intends to continue to provide support in the migration of applications to OSS AP servers by promoting this tool to those in development projects who are migrating, or considering migrating, Java applications, and by actively expanding our migration support service. As of December 2013, we had had more than 1000 accesses from more than ten countries, so we are attracting substantial interest. With such knowledge provision and feedback from around the world, we hope to enrich

and improve a wide body of knowledge and develop it as a tool that can dramatically reduce the costs of migration from various commercial AP servers to OSS AP servers and migration between different versions of AP servers.

References

- [1] NTT OSS Center (in Japanese).
<http://www.oss.ecl.ntt.co.jp/oss/techno/jboss.html>
- [2] TUBAME, GitHub.
<https://github.com/TUBAME>
- [3] Struts, The Apache Software Foundation.
<http://struts.apache.org/>
- [4] Eclipse, the Eclipse Foundation.
<http://www.eclipse.org/>



Masato Urano

Senior Expert, Application Software Technology Unit, NTT Open Source Software Center.

He received the M.E. in industrial engineering from Yokohama National University in Yokohama, in 2000. He joined NTT in 2011. He provides JBoss support to NTT Group companies and develops tools related to JBoss such as TUBAME.



Masahiro Nagafusa

Leader, Application Software Technology Unit, NTT Open Source Software Center.

He received the B.L.S. from the University of Library and Information Science (currently, the School of Informatics at the University of Tsukuba, Ibaraki), in 1989. He joined the NTT Group in 1989. He manages the support operations of more than 15 web/OSS products including JBoss.



Kyohei Nakamura

Expert, Application Software Technology Unit, NTT Open Source Software Center.

He received the BSIT in computer science from Chiba Institute of Technology in Chiba, in 2010. He joined NTT in 2013. He is the developer of TUBAME and provides Tomcat and JBossEAP support for NTT Group companies.

Faster System Building with No Reworking! OSSVERT Smart Builder

Shigenari Suzuki, Hiroharu Toyoda, Ryo Sugahara, Hiroyuki Yamada, and Hirohisa Tezuka

Abstract

The scale of systems that provide numerous and diverse services continues to increase. The NTT Open Source Software Center has developed the OSSVERT Smart Builder tool to reduce the amount of work required and the number of errors that occur when building and managing OSS (open source software) systems. In this article, we describe various aspects of the tool, including its features and technical details, and we report on its effectiveness verified using real equipment.

Keywords: system building, automation, puppet

1. Introduction

In the NTT Group, we are consolidating the work of building and managing systems by building and migrating various internal systems that have been operated separately onto an integrated infrastructure. In doing this, as the number of servers comprising the integrated infrastructure increases, the need for ever more efficient operation in system building and management tasks increases.

Various commercial tools that support system building and management have been offered to satisfy such needs. However, there are barriers to using them such as being restricted to vendor-specific hardware, or a very high cost.

To solve these issues, the NTT Open Source Software (OSS) Center has been studying and developing the OSS Suites Verified Technically (OSSVERT) Smart Builder (O-SB) tool. Our objectives are focused on reducing the effort required and the human error involved in building systems by using open source software (OSS), and maximizing the use of our accumulated OSSVERT system-building know-how.

2. OSSVERT Smart Builder

2.1 Overview

System building and management work can be divided into three processes: design, installation and configuration, and testing (**Fig. 1**). Of these, the processes of installing (or upgrading) the operating system (OS) and middleware, configuration, and testing increase proportionately with the number of servers, and are prone to human error.

In developing the O-SB, we focused our study on the process of installing and configuring the OS and middleware, which consumes a particularly large amount of time, and we implemented functions to automate these processes. A concrete representation of O-SB operation is shown in **Fig. 2**. The O-SB server automatically installs and configures the target server over the network according to a predefined procedure. It is fully automated from the start through the completion of installation and configuration and requires no operations by the user. The O-SB server also has a function to maintain the target server in a correct state after the initial system building process is completed, in case an error occurs in the environment changes or in another area.

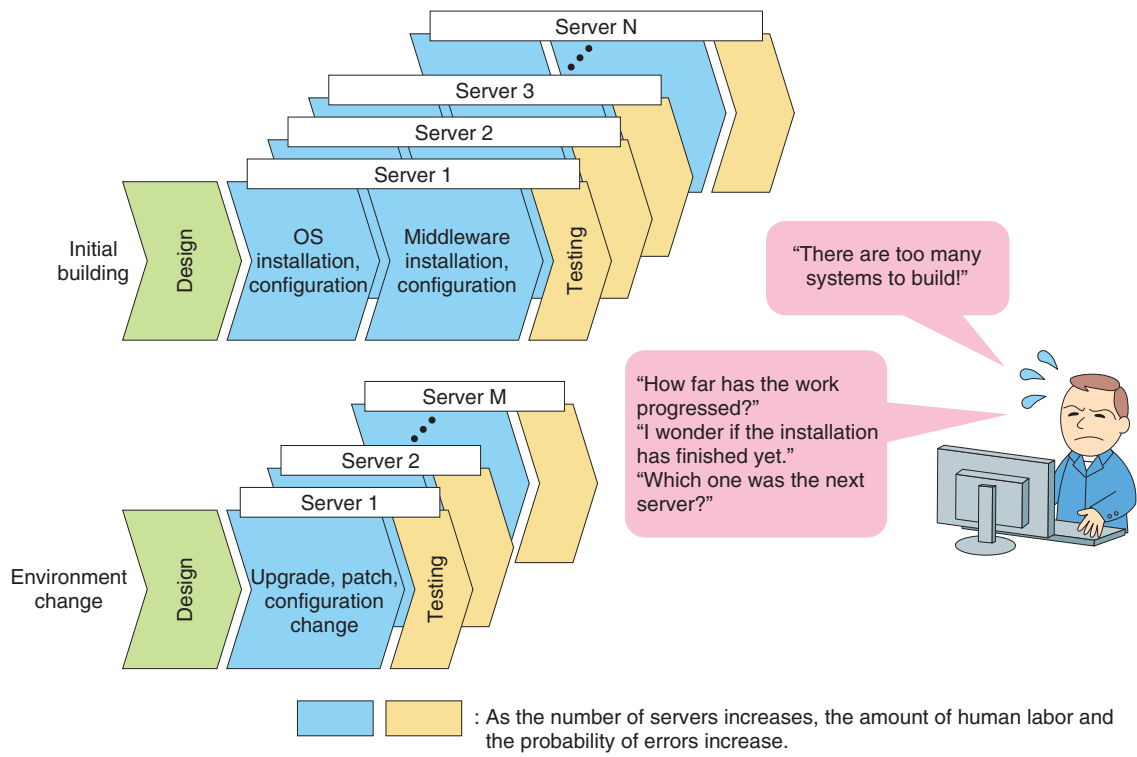


Fig. 1. System building and management process.

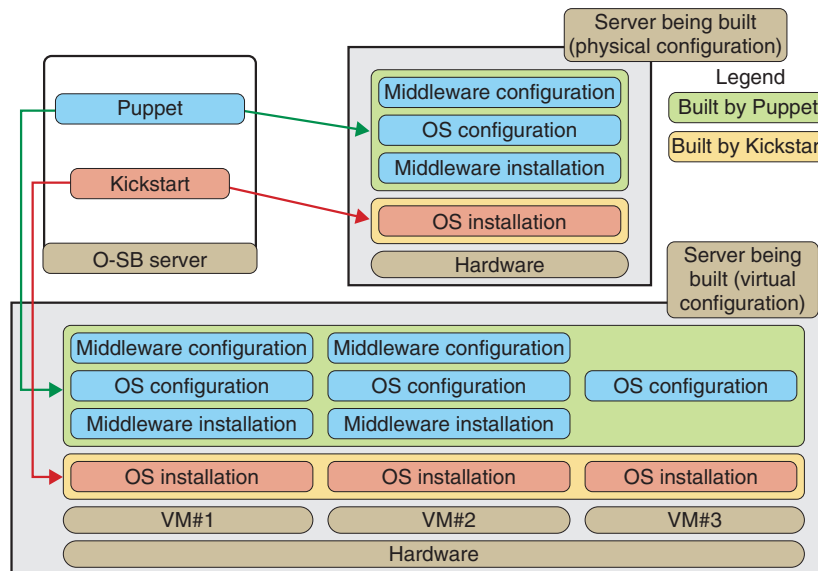
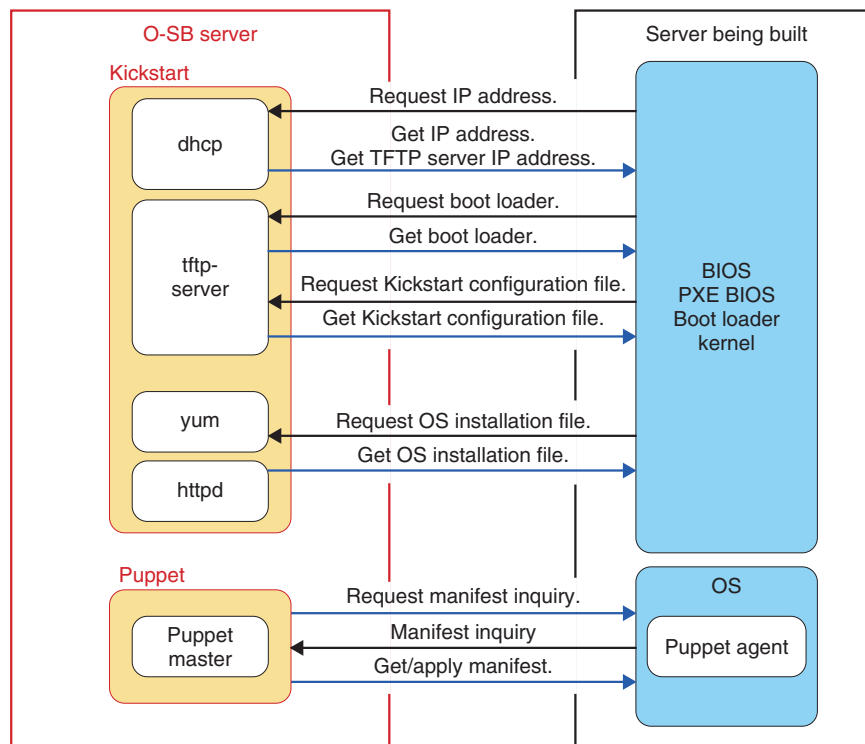


Fig. 2. O-SB basic operation.



BIOS: basic input/output system
 IP: Internet Protocol
 PXE: preboot execution environment
 TFTP: Trivial File Transfer Protocol

Fig. 3. Sequence of operations between the O-SB server and the server being built.

2.2 Technical points

The sequence of operations between the O-SB server and the server being built is shown in Fig. 3. O-SB has two technical features, as follows.

- (1) It uses OSS (Puppet^{*1}, Kickstart^{*2}) that has a proven track record of building and managing real systems used in a variety of fields such as finance, information, and SNS (social networking services).
- (2) It uses OSSVERT system-building know-how. Puppet and OSSVERT are particularly important and are therefore explained in more detail.

Puppet is an OSS automatic system configuration management tool. The user describes the desired state of the system in a configuration file called a *manifest*. The *puppet master* reads the manifest and uses a *puppet agent* placed on the server to configure the system on the server being built according to the manifest (Fig. 4). This allows installation and configuration of middleware and other software, and various types of configuration for the OS. The description in the

manifest is extremely flexible, so there are multiple ways to describe any given configuration. This could cause confusion about how to describe a configuration, so O-SB provides an outline of the description created in order to reduce the work required by users.

O-SB also provides puppet manifest templates that allow various OS and middleware servers to be built immediately. These templates have been created by adding OSSVERT know-how to the aforementioned outline descriptions. OSSVERT is the OSS stack recommended by the NTT OSS Center when building a web-based three-tier system. OSSVERT validates the behavior (performance, long-term stability, maintainability, etc.) of an entire web-based three-tier system, including benchmark applications, and gathers

*1 Puppet: An open source system-configuration framework and tool set that is capable of UNIX-type system configuration and management.

*2 Kickstart: A mechanism for automating the installation of Red Hat OS versions such as Red Hat Linux and Fedora Core.

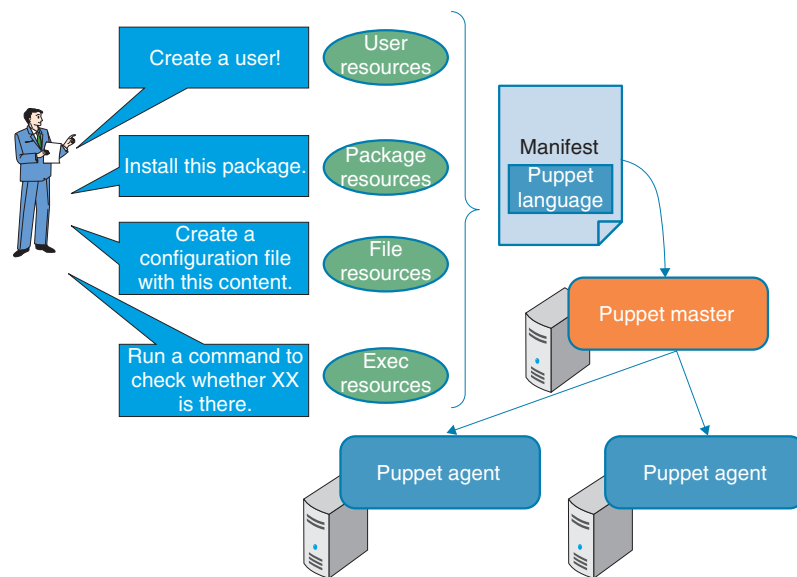


Fig. 4. Organization of Puppet.

technical materials including installation manuals, environment definition documents (e.g., a list of recommended settings), and test reports.

O-SB automatically builds systems that have been verified to operate correctly, according to the OSSVERT installation procedure and environment configuration documents and using the described manifest. However, some values (addresses etc.) need to be changed in real scenarios, so a document is provided that summarizes the set of parameters to be changed and where in the templates the changes can be made.

3. Application example and results

We measured the effectiveness of the tool when building a web-based three-tier system of an OSSVERT configuration with six physical servers (web, application, and database servers, each in a double or redundant configuration). The building procedure was done in two steps, starting with the six physical servers in their initially purchased state and physical connections in place: (1) installing and configuring the OS (including drivers), and (2) installing and configuring various types of middleware. We confirmed that this would require approximately 60 hours of manual work by a typical technician, but this was reduced to approximately 8 hours (one-seventh the time required to do it manually) by using O-SB. This 8-hour period includes the time to build the O-SB

server itself and to code the definition file (manifest) for the servers to be built. To rebuild the same environment (second and subsequent times), the O-SB server can be reused, so the systems can be built in about 2 hours (in one-thirtieth the time) (Fig. 5). We also confirmed through simulation that if the number of servers increases further, O-SB can run in parallel, so for up to a certain number of servers, the building time will not increase, depending on the specifications of the network and servers.

This confirms that O-SB is effective in reducing the time required for installation and configuration. We also confirmed other beneficial effects such as a reduction in verification work necessary during the design process by incorporating OSSVERT design know-how into the manifests, a reduction in human errors in the configuration process, and elimination of the need for operator intervention during the installation and configuration process.

4. Future plans

The software that can be built using O-SB includes typical items in OSSVERT configurations, as indicated in Table 1. For the software listed in the table, users can create manifests to install and configure the desired software in a relatively short time using templates, even if versions are somewhat different.

NTT OSS Center is preparing training materials for users to guide them in areas such as revising templates

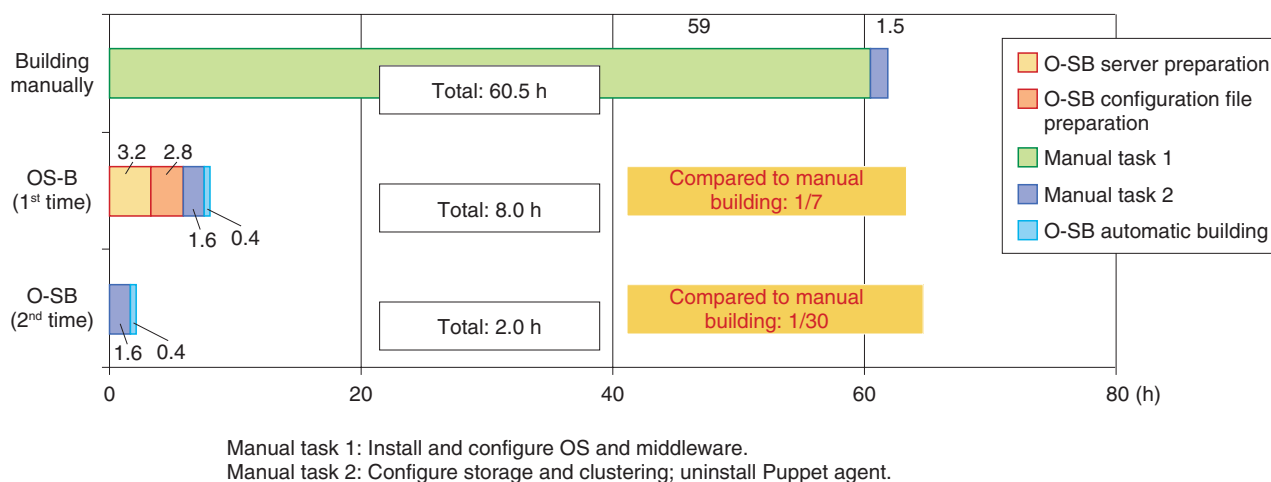


Fig. 5. Potential time savings by using O-SB.

Table 1. Software that can be installed and configured using O-SB.

Category	Product name
OS	Red Hat Enterprise Linux
Database	PostgreSQL
	Oracle
Application server	JBoss EAP
	Tomcat
	OpenJDK
	JDBC
	HeapStats
Web server	Apache
	mod_jk
Clustering software	Pacemaker
Load balancer	UltraMonkey-L4

and creating a manifest from scratch. We are also providing support for inquiries regarding Puppet through a help desk.

As the scale of systems becomes even larger, we expect the number of scenarios in which O-SB can be used to increase even more, so we will continue our research and development while anticipating applications of O-SB in various environments in the future.

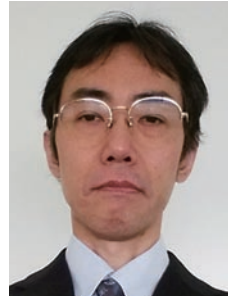
References

- [1] Y. Hirakawa, “RHEL6 in Practice! Anyone can Build a Home Cloud (Pt. 4)—Building a Provisioning Server,” Nikkei Linux, July 2011 issue, 2011 (in Japanese).
- [2] S. Uchikawa, “The OSSVERT Initiative: For Using Open Source Skillfully,” NTT Technical Journal, Vol. 19, No. 3, pp. 60–62, 2007 (in Japanese).
- [3] H. Toyoda and M. Okumura, “OSSVERT Extensions Supporting Introduction of Open Source Software,” NTT Technical Journal, Vol. 23, No. 8, pp. 8–13, 2011 (in Japanese).


Shigenari Suzuki

Senior Research Engineer, NTT Open Source Software Center.

He received the B.E. and M.E. in engineering from Kanazawa University, Ishikawa, in 1993 and 1995, respectively. Since joining NTT in 1995, he has been engaged in the development of an ATM transmission service monitoring system, large-scale system development to provide television relay service, and development and support of OSSVERT and OSSVERT Smart Builder.


Hiroyuki Yamada

Expert, Application Software Technology Unit, NTT Open Source Software Center.

He received the LL.B. from Tokyo University, and then joined NTT DATA in 1995. Most of his past career is related to open source software (OSS), starting from OSS advocate back in the end of the previous century. He used to get involved in OSS system deployment/migration projects ever since.


Hiroharu Toyoda

Research Engineer, Open Source Software SE Project, NTT Software Innovation Center.

He received the B.E. and M.S. in engineering from Nagoya University, Aichi, in 1990 and 1992, respectively. Since joining NTT Network Information Systems Laboratories in 1992, he has been developing a real-time operating system for switching systems and information systems. Since 2006, he has been developing and maintaining OSSVERT, an OSS stack model to check the interconnectivity and quality of multiple types of OSS. He has also been involved in promoting several types of OSS to the NTT Group. He is a member of the Institute of Electronics, Information and Communication Engineers.


Hirohisa Tezuka

Leader, Application Software Technology Unit, NTT Open Source Software Center.

He received the B.E. and M.E. in engineering from Kyoto University in 1991 and 1993, respectively. He joined NTT in 1993. His research interests include system management, networked robotics, and lifelog systems. He is a member of the Society of Instrument and Control Engineers.


Ryo Sugahara

Senior Expert and Project Management Professional, Open Source Business Unit, Solution Business Division, NTT DATA INTELLILINK.

He received the B.E. in science from Toho University, Tokyo, in 1995. Since joining NTT DATA in 2005, he has been developing large-scale web information systems. He joined NTT DATA INTELLILINK in 2011, where he has been researching IT automation systems.

Pacemaker: Increasing System Reliability

Kengo Fujioka, Yuta Takeshita, Keisuke Mori, Takayuki Tanaka, Kazuhiko Higashi, and Kazuyoshi Mii

Abstract

The NTT Group builds and operates many systems that provide various services internally and externally. These systems are required to have less down time. In this article, we introduce the latest trends concerning Pacemaker, the high-availability cluster software that enables highly reliable systems. We also report on the efforts of the NTT Open Source Software Center in this area.

Keywords: OSS, Pacemaker, high-availability cluster

1. Introduction

As the importance of a service increases, the emphasis on system reliability increases. System reliability can be measured using various parameters such as mean time to failure or mean time to repair. One index for these parameters is downtime. Systems with low downtime, meaning that the proportion of time that the system can be used is high, are called high-availability systems. One way to implement a high-availability system is to use a high-availability cluster, which achieves service continuity by having redundant servers, and automatically switching between servers when a fault occurs. Pacemaker is open source software (OSS) for high-availability clustering. An overview of Pacemaker operation is shown in **Fig. 1**.

2. Pacemaker stable version

The development community currently provides the 1.0 series of Pacemaker versions as the stable product version, with periodic revisions focused mainly on fixing bugs. This is referred to as the Pacemaker stable version in this article. The stable version is comparable to commercial products in features and quality, and has already been used in many systems. It has been used for many network service operation systems within the NTT Group as well.

The stable version was developed mainly for small-scale systems. As shown in Fig. 1, its basic configuration is a simple 1:1 configuration with a single active node corresponding to a single standby node. It can also be used with N active nodes and one standby node (N:1) for a small number of nodes.

3. Pacemaker developer version

The success of the Pacemaker stable version resulted in greater demand for a version that was applicable over a wider domain. The stable version was originally intended for use in small-scale systems, but as the application domain expanded, functions supporting larger-scale systems were developed. These functions are implemented in the developer versions of Pacemaker (Ver. 1.1 series). The three main features of the developer version are: (1) it accommodates an increased number of nodes; (2) it enables efficient use of standby nodes; and (3) it implements geoclusters. These are introduced below.

(1) Increased number of nodes

With the stable version, clusters with more than two nodes are possible, but there are limitations. The NTT Open Source Software (OSS) Center sets a guideline of a maximum of six nodes (e.g.: five active nodes and one standby node). This limitation is due to the use of Heartbeat, a cluster infrastructure software, for the node management function^{*1}. In contrast, the

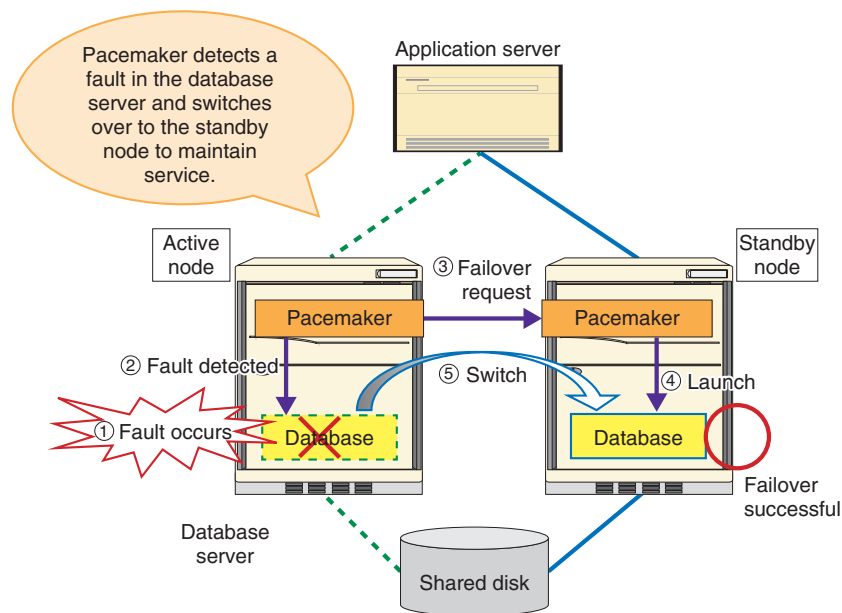


Fig. 1. Overview of Pacemaker operation.

Pacemaker developer version uses Corosync as the node management function, and its communication functions were designed for many nodes. Thus, many more nodes can be accommodated with the developer version. Specifically, there have been cases reported within the community in which 16 nodes (e.g., 15 active nodes and 1 standby node) were used without difficulty.

(2) Efficient use of standby nodes

Several issues arise when the number of nodes is increased and they are applied in a real system. For example, with the stable version of Pacemaker, only configurations with N active nodes and one standby node are possible, so only a single fault can be handled^{*2}(Fig. 2(a)). On the contrary, with the developer version of Pacemaker, multiple standby nodes can be applied for N active nodes, so multiple faults can be handled. Each time there is a fault in an active node, one of the standby nodes not in use is selected for failover^{*3}(Fig. 2(b)).

With large-scale clusters, the possibility that multiple active nodes will experience a fault increases. This functionality allows the number of standby nodes to be designed appropriately according to how many faults the system will support or the level of reliability required, and the overall system server resources can be used efficiently.

(3) Geoclusters

To ensure continuous service even in the event of a

major disaster, high-availability clusters spanning separate sites were developed in the developer version. These are referred to as geoclusters.

An important function of high-availability clusters is preventing services such as the database from being launched more than once, even during faults such as a network failure or system runaway. For example, if the communication between active and standby nodes is lost for some reason, and both active and standby nodes start the database, data corruption or other serious failures could result. This sort of situation is called *split-brain*.

With a local cluster, *split-brain* is prevented by measures such as forcibly terminating the server with the fault or exclusively controlling the service using a shared disk. These measures ensure that the service running in the cluster is unique. However, these measures cannot be used for a geocluster over a wide-area

*1 Node management function: A functional component of Pacemaker that handles communication between nodes and manages cluster members. There is also a *resource management function* that controls resources being managed; this resource management controller is called Pacemaker in a narrow sense.

*2 Strictly speaking, a single standby node can accept failover from multiple active nodes, but if failover occurs for services exceeding their processing capacity, the services will not be able to continue normally.

*3 Failover: The high-availability clustering software detects that a fault has occurred in an active server and switches operation of the service to a standby server.

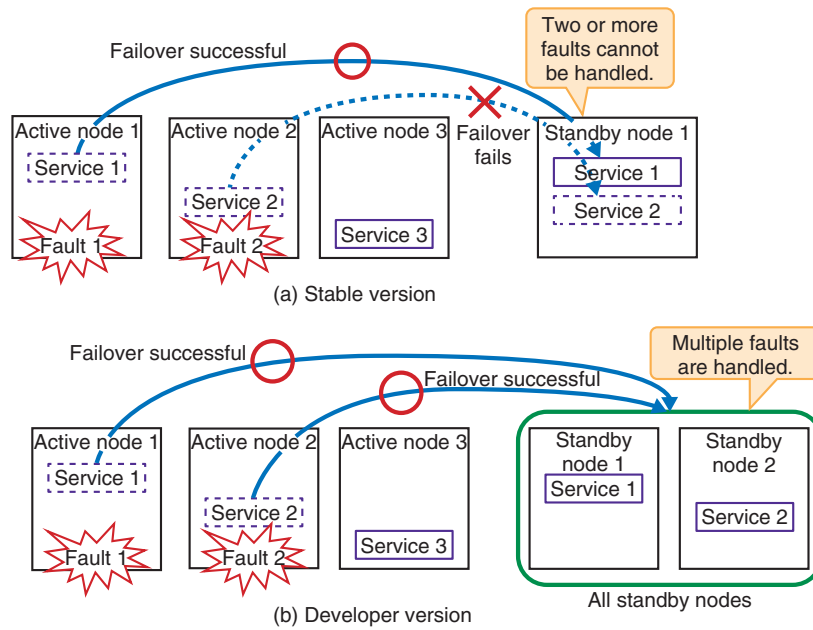


Fig. 2. Pacemaker failover.

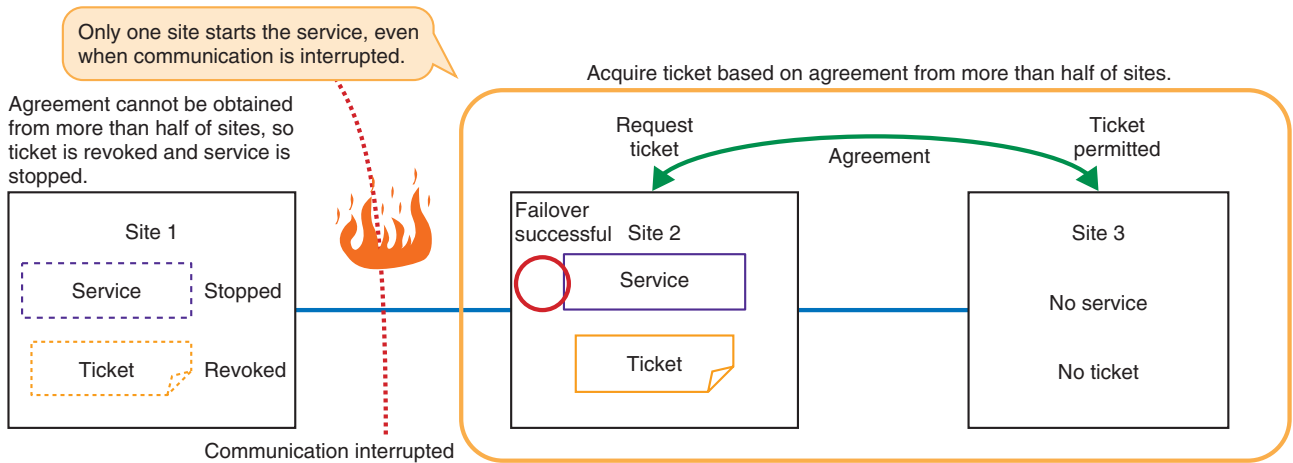


Fig. 3. Split-brain solution for geocluster.

network.

Instead, the developer version introduces a concept called the *ticket*, which gives the right to run the service. To obtain the ticket, agreement from more than half of the sites comprising the geocluster is needed. If agreement from more than half of the sites can no longer be obtained, the ticket becomes invalid, and the service cannot continue. This guarantees that no more than one service is running in the geocluster at

a time.

The countermeasure for split-brain using a ticket is shown in **Fig. 3**. In the figure, site 1 loses communication and can no longer communicate with more than half of the sites, so it discards the ticket and stops the service. On the other hand, site 2 loses communication with site 1, so it obtains agreement from more than half of the sites (site 3 and itself), gets the ticket, and starts the service.

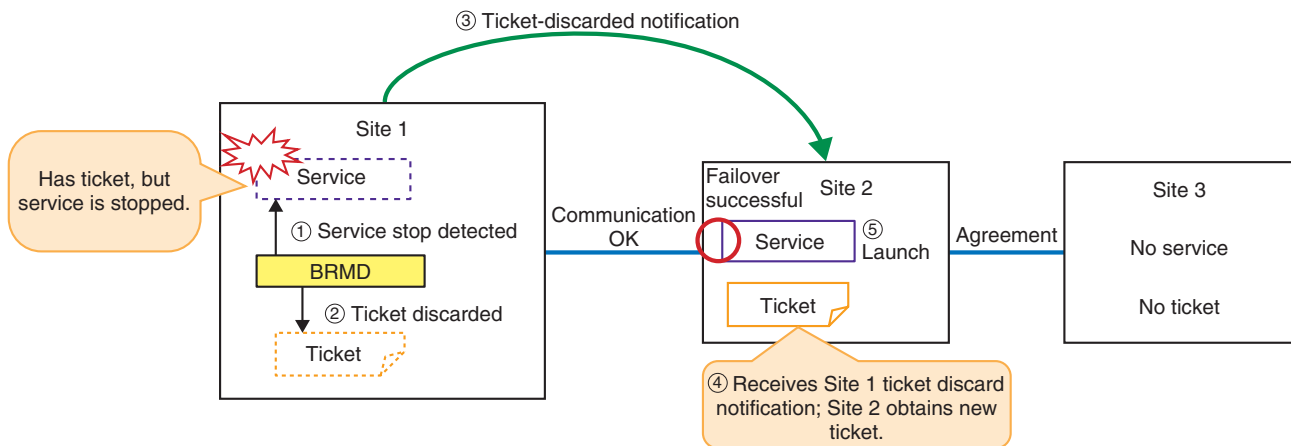


Fig. 4. Overview of operation of Booth_Resource_Monitor.

Since only one ticket can ever exist, even if communication is lost between sites, only one site running the service can exist. Thus, failover between sites can occur without multiple services being launched, even if communication is lost due to a disaster or other cause.

4. NTT OSS Center initiatives

Many of the services offered within the NTT Group are provided 24 hours a day and 365 days a year, so systems with low down time are needed. The NTT OSS Center is working to implement high-availability systems using Pacemaker in order to meet these requirements. We introduce some of these activities below.

(1) Participation in the development community

Developers from around the world participate in the Pacemaker development community. The NTT OSS Center has also developed many functions and provided them to the community [1]. We are also helping to improve quality by conducting testing to verify operation with each product release, reporting any bugs discovered to the community, and posting patches as necessary. As a result of these contributions, one of the authors of this article, Mr. Mori, was appointed maintainer of Pacemaker in 2010 and given a central role in preparing the releases of the Pacemaker stable version.

(2) Development activity in cooperation with the community

To support geoclusters in the developer version, we developed a function that we called Booth_Resource_

Monitor, or BRMD, and provided it to the community. Initially, the developer version geocluster function had a problem that it did not check the service state. Therefore, if the service stopped in the site with the valid ticket, failover between sites was not possible, and as a result, no sites ran the service. To solve this problem, we discussed it with the community and developed functionality to monitor the state of the service on the server with the valid ticket, and to revoke the ticket if the service stops. This enables the service to continue by failing over to another site if the service stops on the site with the valid ticket (Fig. 4). The importance of this function was recognized, and in June 2013, the development community accepted it in the product.

(3) Promoting use within the NTT Group

To promote the use of Pacemaker, we are providing a great deal of support that ranges from consulting on the introduction of Pacemaker to troubleshooting during operation. When problems arise, we handle them using the know-how cultivated through community activity, even analyzing the source code if necessary. Through these efforts, the use of Pacemaker within the NTT Group has been increasing every year. By spreading the use of Pacemaker, the NTT OSS Center is contributing to reducing the total cost of ownership (TCO) within the NTT Group.

5. Future work

The NTT OSS Center is currently working to improve the quality of Pacemaker by enhancing its functionality and is also consolidating the Japanese

documentation so that we can apply the developer version for NTT Group's systems. We are cooperating with the developer community and contributing to the development of Pacemaker in order to implement high-reliability systems in the future. We will support the high-availability needs of new platforms as virtualization, cloud, and other technologies devel-

op, with the goal of expanding its use in business.

References

- [1] T. Sakata, K. Mii, S. Kihara, and T. Iizuka: "Infrastructure OSS Technology supporting OSSVERT," NTT Technical Journal, Vol. 23, No. 8, pp. 14–18, 2011 (in Japanese).



Kengo Fujioka

Manager, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the M.S. in physics from the University of Tokyo in 1995. He joined NTT in 1995. He has worked at NTT Open Source Software Center since 2010. His research interests include high-availability systems and open source software.



Takayuki Tanaka

Senior Expert, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the B.E. in visual communication design from Kyushu Institute of Design, Fukuoka, in 1995. He joined NTT in 1995. His research interests include high-availability systems and open source software.



Yuta Takeshita

Expert, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the M.E. in intelligent engineering from Doshisha University, Kyoto, in 2010. He joined NTT Comware in 2010. He moved to NTT Open Source Software Center in 2013. His research interests include high-availability systems and open source software.



Kazuhiko Higashi

Expert, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the B.E. in physical science and engineering from Nagoya University, Aichi, in 2004. He joined NTT Comware in 2004 and moved to NTT Open Source Software Center in 2012. His research interests include high-availability systems and open source software.



Keisuke Mori

Manager, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the M.S. in information engineering from Yamagata University in 1994. He worked at NTT Software Corporation and Phoenix Technologies K.K. prior to joining NTT DATA INTELLILINK in 2004. He has worked at NTT Open Source Software Center since 2009. His research interests include high-availability systems and open source software.



Kazuyoshi Mii

Leader, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the M.E. in applied systems science from Kyoto University in 1993. He joined NTT in 1993. He has worked at NTT Open Source Software Center since 2008. His research interests include open source software development and business strategies. He is a member of the Information Processing Society of Japan and the Database Society of Japan.

PostgreSQL: Leader in Cost Reduction

Akiyoshi Kawada, Go Nishimura, Junji Teramoto, Etsuro Fujita, Takeshi Yamamuro, and Takuma Wakamori

Abstract

The NTT Open Source Software Center is conducting research and development on the PostgreSQL open-source database management system in order to develop and spread open source software (OSS). Recent advances in the development of PostgreSQL have brought it to a level of usability adequate for commercial systems, and due to its great cost-reducing effects relative to commercial products, it is receiving much attention and even being introduced for core systems. In this article, we introduce new functionality in the latest version, PostgreSQL 9.3, released in September 2013, and outline NTT's OSS development policies. We also introduce the SQL/MED (Structured Query Language/Management of External Data) indexing engine, which was developed by the authors as infrastructure for log analysis and an area for expanding applications of PostgreSQL.

Keywords: DBMS, PostgreSQL, open source

1. Introduction

Database management systems (DBMSs) store, manage, and make use of various types of data, and they have been under continuous research and development (R&D) for several decades. DBMSs hold a very important place in ICT (information and communication technology) systems and play many different roles. For example, in the area of data storage management, they provide synchronization control so that updates are processed correctly without creating inconsistencies if several people update the data at the same time, and recovery functions so that the results of completed operations are not lost if the system goes down.

In the area of data search, they provide a declarative language enabling applications to search data easily simply by describing what data they need and without having to implement detailed search algorithms. This simplifies application development.

A DBMS is a sophisticated mechanism that requires advanced technical ability to develop it. Furthermore,

it will be used continuously by many users and systems for five, ten, or more years, so it must be maintained continuously to preserve stability and quality. PostgreSQL has been developed by technologists from around the world that have formed a community, and it currently has functionality and performance that compares favorably with commercial DBMS products. NTT is also a member of this community and has contributed by operating and supporting the portal site for the Japan PostgreSQL user's group (JPUG). We participate in development discussions and provide many patches for functionality and performance improvements. In fact, NTT was the top patch contributor from Japan for the current main versions, 9.x (9.0 to 9.3) (**Fig. 1**).

NTT also proposed the streaming replication function and developed it together with the community, to greatly improve PostgreSQL availability for use in internal and external mission-critical applications. This feature is currently used by many users around the world.

The NTT Open Source Software (OSS) Center is

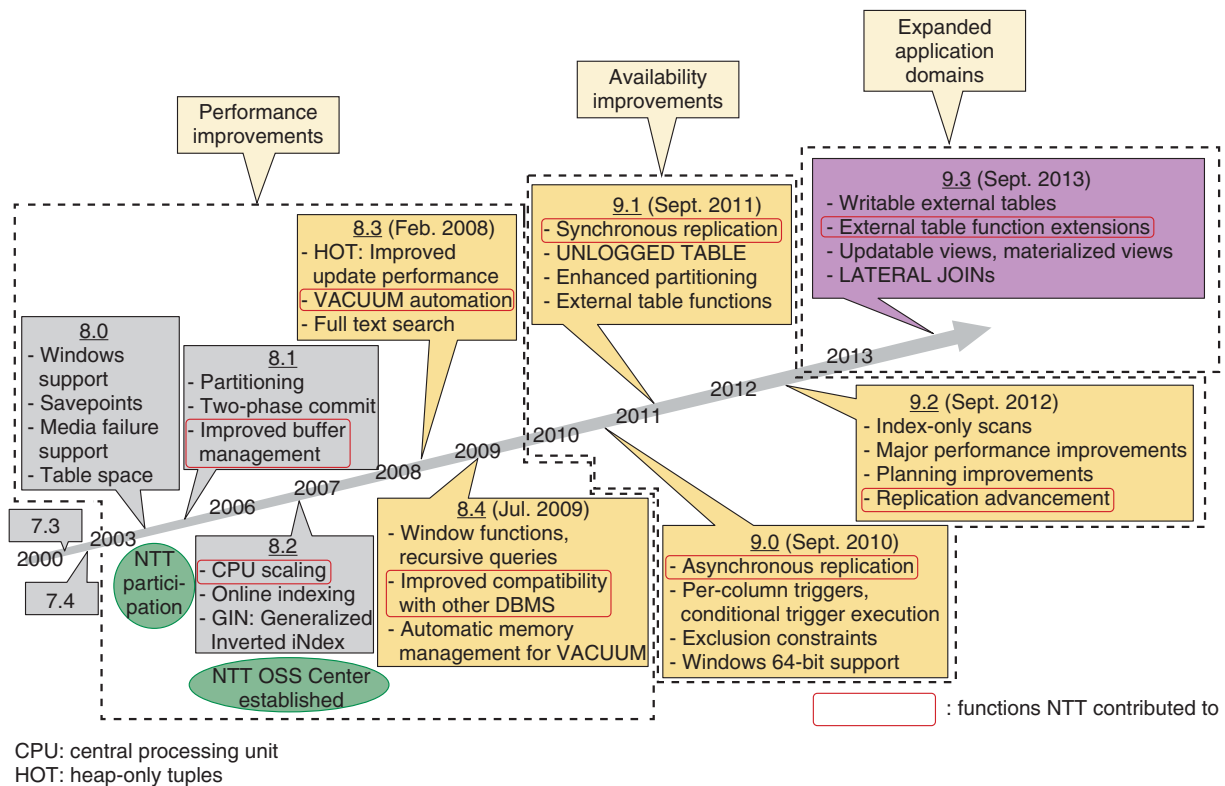


Fig. 1. Main functions contributed by NTT to each version of PostgreSQL.

addressing the themes of enhancing performance and functionality as databases increase in scale, and improving portability from commercial DBMSs. Initiatives include developing external table functions for use with large-scale databases, conducting R&D on distributed databases, and creating functions for improving compatibility that will enable Structured Query Language (SQL) in dialects particular to commercial DBMSs to run as-is on PostgreSQL as well.

2. PostgreSQL Version 9.3

A major revision of PostgreSQL is released once a year, and the most recent was Version 9.3, released in September 2013. This version includes many functional and performance enhancements. The three main enhancements are as follows.

(1) View function extensions

New additions to these functions include materialized views and updatable views. Views are virtual tables that could in fact consist of a query from multiple tables (a SELECT statement). A materialized view stores the result of executing the query as a table so it can be reused when the same query is called

again to increase query processing speed. An updatable view allows updates to be applied to a view without creating complex definitions. Until now, view functions were a weak point when compared with commercial DBMSs, so implementation of these functions should simplify migration from commercial DBMSs.

(2) Replication function improvements

To improve fault tolerance in the system, PostgreSQL has a replication function that enables two servers to operate as active and standby servers. If the active server experiences a fault, the standby server must be restarted as the active server, but improvements have been made to reduce the time for this process. This has increased system availability and enabled operation rates to approach 100%. It is also one of the functions that NTT has put effort into developing and contributing.

(3) External table function extensions

PostgreSQL functions for managing external data as external tables have been enhanced. The external table functions enable data external to the DBMS (such as a comma-separated values (CSV) file) to be handled as a table, just like an internal table. This

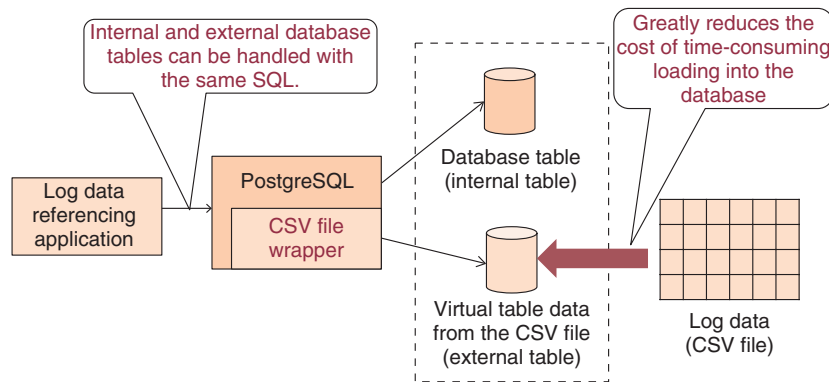


Fig. 2. Overview of PostgreSQL SQL/MED technology.

enables many large logs to be analyzed using SQL without loading them into the DBMS; instead, they are kept externally. External tables were introduced in PostgreSQL 9.1, and in the latest version 9.3, they have been extended to handle data on other PostgreSQL servers as external tables in addition to data in files. Update functions to add and delete records from external tables have also been added, so large amounts of data can now be easily distributed over multiple PostgreSQL servers. NTT’s contributions have helped to advance the distributed query optimization technology since external table functions were first added, and these results were used in the latest functional enhancements.

3. SQL/MED indexing engine development initiatives

We now introduce the SQL/MED (Management of External Data) indexing engine developed by the authors as a log analysis infrastructure using PostgreSQL external table functions.

Conventionally, DBMSs have primarily been used for processing online transactions. However, management decision-making using business intelligence has taken center stage, and DBMSs are being used more and more for analyzing large-scale data such as logs. Service history data (log data) are now increasingly being managed as CSV files for the purpose of analyzing customer preferences. Even on information systems operating in the background of services, more detailed system operation and error logs are being collected, and these logs are often being analyzed in order to improve operations.

In some cases, most of the analysis of acquired logs

is done manually using editors and scripts, but this manual analysis work can be difficult if the logs being processed are large. If the logs are placed in a DBMS, the large volume of data can be summarized and analyzed from various perspectives, and sophisticated searches with complex conditions can be done by the DBMS, leaving the developer to focus on the task of analysis. There are also issues with this approach, however. Loading many large logs into a DBMS incurs a great cost in terms of time and management. Using external tables is one way of resolving this issue (Fig. 2).

At NTT, we have implemented SQL/MED as defined in SQL2003 (the SQL language international standard set in 2003) for PostgreSQL, as well as an extended implementation of external table indexing not in the SQL standard, called SQL/MED indexing [1].

Currently, NTT is also conducting R&D to implement partitioning technology and index-only scan technology to further increase processing speed (Fig. 3). These technologies were introduced into a contact-center voice-mining platform for call analysis (a voice mining system), which had required quite a long time to process database updates with earlier technology. The results indicated that the technologies reduced the time to one-half to one-third that of earlier values. The analyst was able to allocate the time saved on processing to other types of analysis, and to provide more data.

We now introduce the two key technologies to increase speed: partitioning technology and index-only scan technology.

(1) Partitioning technology

Partitioning technology partitions tables in order to

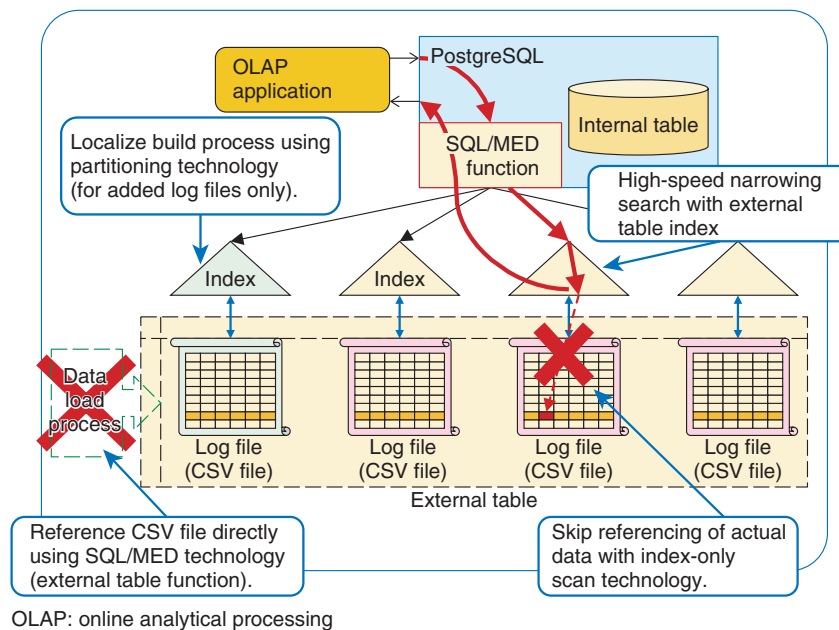


Fig. 3. Partitioning and index-only scan technologies.

speed up table referencing, index creation, and other operations. Partitioning tables reduces the range of key values handled when processing a table reference. In a scenario where newly arrived log entries are always added to the end of the log, the creation of indices can be completed by processing only the latest log file that caused the update. Speed is increased by processing only some of the partition tables. Table partitioning in PostgreSQL uses a function called table inheritance, which can now also be used with external tables. For example, consider a table that is partitioned by months and operated cyclically, with tables added for the latest month and deleted when they are a year or more old. When table partitioning is introduced, operations such as deletion, addition, and index creation only apply to tables for particular months, so the work and processing time for these operations is reduced. When large logs are analyzed by processing external tables, the disk I/O (input/output)—rather than the CPU (central processing unit)—tends to be the bottleneck, so using partitioning technology to reduce disk I/O can be particularly effective in increasing speed.

(2) Index-only scan technology

Index-only scan technology speeds up the processing of table references by only scanning an index (and not accessing the data itself) to obtain search results in cases when the query satisfies certain conditions.

Normally, index sizes are small and can be kept in memory, so disk I/O can be reduced dramatically when processing external tables. This results in dramatic increases in the speed of table reference processing. The query condition for this technology to be effective is that the columns specified in the **SELECT** list and **WHERE** condition in the query must all be present in the columns used to create the index. (Multiple columns can be used.) It is possible to consider that all of the columns needed for a search result are included when designing an index, so index-only scan technology can be effective in many circumstances.

By combining the above technologies, the SQL/MED indexing engine can access large amounts of data at high speed without loading it into the DBMS and can be used in business for applications such as data analysis.

4. Future developments

This article presented some trends in the latest version of PostgreSQL and the development of a log analysis infrastructure at the NTT OSS Center. In addition, the NTT OSS Center is putting effort into providing support for introduction and operation of PostgreSQL so that it can be confidently used for business. The technical expertise gained through

development is used in providing support, and the requests for improvements received when providing support are fed back into development, contributing to further improvements in PostgreSQL. We will continue to contribute to reducing business costs and expanding business domains through the two-pronged approach of support and development.

References

- [1] T. Hitaka, N. Kotani, Y. Suga, E. Fujita, A. Kawada, and T. Yamamuro: "PostgreSQL SQL/MED Technology—Infrastructure for Efficient Analysis of Service and System Logs," NTT Technical Journal, Vol. 23, No. 8, pp. 30–34, 2011 (in Japanese).



Akiyoshi Kawada

Expert, Infrastructure Technology Unit, NTT Open Source Software Center.

He received the M.S. in theoretical physics from Hokkaido University, Sapporo, in 1996. He joined NTT Comware in 1996. He then moved to NTT Open Source Software Center in 2012. His research interests include database as platforms of log analysis and open source software.



Etsuro Fujita

Senior Expert, Infrastructure Technology Unit, NTT Open Source Software Center.

He received the PhD. in informatics from The Graduate University for Advanced Studies (SOKENDAI), Kanagawa, in 2013. He joined NTT in 1996. He then moved to NTT Open Source Software Center in 2012. His research interests include database systems, information integration, information retrieval, and open source software.



Go Nishimura

Senior Manager, Infrastructure Technology Unit, NTT Open Source Software Center.

He received the M.M.G. from Keio University, Tokyo, in 1996. He joined NTT in 1996. He then moved to NTT Open Source Software Center in 2013. His research interests include information systems and applications.



Takeshi Yamamuro

Researcher, Distributed Computing Technology Project, NTT Software Innovation Center (since April, 2014).

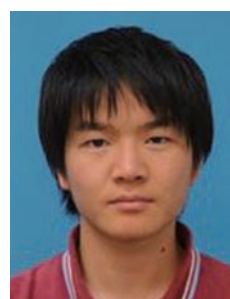
He received the M.E. in science and engineering from Sophia University, Tokyo, in 2008. He joined NTT in 2008, and then moved to NTT Open Source Software Center in 2012. He is interested in columnar-db topics such as vectorized pipelining, type-specific compression, and hardware-aware techniques.



Junji Teramoto

Manager, Infrastructure Technology Unit, NTT Open Source Software Center.

He received the M.E. in electronics, information and communications engineering from Waseda University, Tokyo, in 1998. He joined NTT in 1998. He then moved to NTT Open Source Software Center in 2012. His research interests include relational database management systems and open source software.



Takuma Wakamori

Researcher, Distributed Computing Technology Project, NTT Software Innovation Center (since April, 2014).

He received the M.E. in electronic and information engineering from Yokohama National University, Kanagawa, in 2013. He joined NTT Open Source Software Center in 2013. He then moved to NTT Software Innovation Center in 2014. His research interests include DBMS, distributed computing, and open source software.

Linux and KVM Activities Providing Foundational Support for Use of OSS

Kenji Yoshida, Kumiko Goto, Toshiaki Makita, Toshio Nakazawa, and Seiji Kihara

Abstract

The NTT Open Source Software Center is working on various aspects of Linux and Kernel-based Virtual Machine (KVM) to support the use of open source software (OSS) within the NTT Group. In this article, we introduce some of these activities, which include building expertise in order to use OSS with confidence, improving the quality of OSS products by cooperating with development communities, and promoting OSS virtualization technology.

Keywords: Linux support, OSS community activity, server virtualization

1. Introduction

When speaking of software, we tend to focus on applications, which are directly visible to users. However, there are other elements that act as “unsung heroes” operating behind the scenes. Operating systems (OSs) and virtualization software are examples of this type of software that operates closest to the hardware.

An OS provides the infrastructure for running various other types of software such as applications. It also provides functions such as process management, memory management, file systems, and networking. The NTT Open Source Software (OSS) Center provides technical support to NTT Group companies for Red Hat Enterprise Linux (RHEL), which is an industry standard Linux OS distribution provided by Red Hat, Inc. and used widely within our group companies. We also consult separately regarding other distributions such as CentOS when requested.

To improve the quality of this support, we investigate and test new versions of RHEL when they are released. In doing so, we actively contribute patches to the OSS development community when we find bugs in order to improve software quality.

Recently, server virtualization technology has been focused on as a way of reducing total cost of ownership (TCO) and achieving more flexible system

operations by integrating servers and systems. Kernel-based Virtual Machine (KVM) is an OSS hypervisor, the standard virtualization technology in Linux, that is included in RHEL. As such, we are actively working on it.

Initially, the operations management features of KVM were limited, so it was not used much within our group. However, Red Hat Enterprise Virtualization (RHEV) has since been introduced, and it provides more substantial operations management features. We are therefore working with RHEV to promote the use of KVM (Fig. 1).

2. Testing new RHEL versions and providing feedback

2.1 Investigation and testing of new RHEL releases

For each major release^{*1} of RHEL, Red Hat offers minor releases^{*1} to fix bugs and security vulnerabilities approximately every six months. Red Hat conducts testing before releasing them so they can be used safely for enterprise services.

However, system administrators tend to be conservative when considering whether applying software updates is necessary because doing so can affect the

^{*1} The latest version is RHEL 6.5; ‘6’ indicates the major release and ‘5’ the minor release.

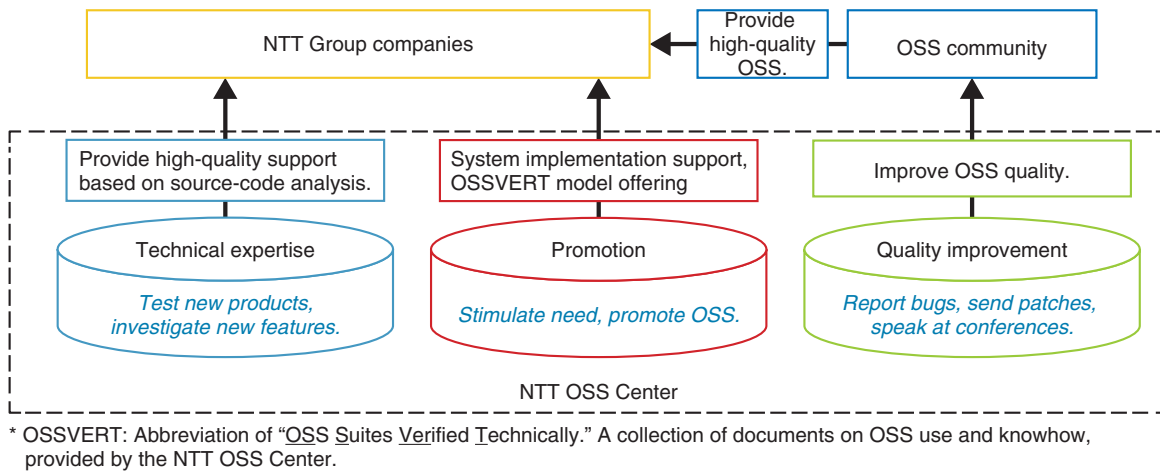


Fig. 1. NTT OSS Center initiatives for use of OSS within the NTT Group.

Table 1. Benchmark tools used in our performance testing.

Measurement items	Benchmark tools
System call, glibc performance	LMbench (http://www.bitmover.com/lmbench/)
Network performance	netperf (http://www.netperf.org/)
Task scheduler performance	ocbench (http://linux.1wt.eu/sched)
Disk I/O performance	IOzone (http://www.iozone.org/)

end users of their enterprise services and incur a substantial cost. This is why an even stronger guarantee of stability is required in our group.

For this reason, with the release of each minor version, certain items need to be checked on the kernel and virtualization software to prevent problems from occurring. Specifically, we check:

- (1) Whether there are significant changes such as added or removed functionality by reading release notes and related documentation.
- (2) Whether it can be installed on a PC server without difficulty or can be upgraded from the previous version.
- (3) Whether there are any significant changes in performance.

On the basis of the results of these checks, we decide whether the new version can be used safely in our group, and if so, we recommend it as a *recommended version*. As an example, for RHEL 6.4 in February 2013, we temporarily deferred making it a recommended version since it failed to pass item (3). The following is a detailed description of how we reached that decision.

2.2 Performance testing

For the newly released RHEL 6.4, as we always do for each minor release, we did some performance tests on a bare-metal server and virtual machines, using KVM for example, to determine whether there were any significant performance changes in the basic functions of the OS. The details of the measurements taken for these tests are given in **Table 1**.

2.3 Disk I/O problem with RHEL 6.4 KVM

On a KVM virtual machine, the disk I/O (input/output) performance test result showed that throughputs for a particular I/O pattern dropped 20% compared to the RHEL 6.3 environment. Since this was a significant change in performance, we temporarily deferred making RHEL 6.4 a recommended version in cases where it would be used for KVM virtualization, and we further investigated the issue.

As a result of the investigation, we reached the conclusion that what had triggered the performance regression was a bug in QEMU—the machine emulator and virtualizer—that could disable a disk I/O optimization.

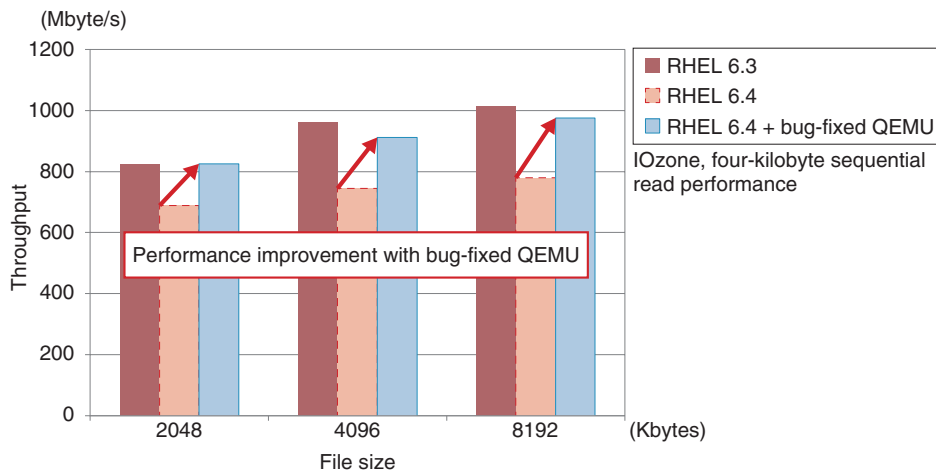


Fig. 2. Disk I/O performance test results (KVM environments).

We also confirmed that by using a later version of QEMU that had received a bug fix patch, we could gain the same performance as RHEL 6.3 (Fig. 2). Finally, on the basis of this result, we decided to recommend the use of the QEMU version that included the bug fix for RHEL 6.4 KVM virtualization [1].

This is an example of how our testing can prevent problems from occurring in real systems. We plan to continue this kind of testing so that our group companies can use RHEL with confidence for enterprise services.

2.4 The next major release (RHEL7)

Red Hat is now preparing to release RHEL 7, the next major version. RHEL 7 is expected to have improved scalability and many new features, so it will most likely become one of the most widely used operating systems in the future.

Accordingly, in preparation for its release, we are strengthening our know-how of RHEL 7, checking its stability, and feeding back any problems found to Red Hat and the OSS community to ensure that it is reliable.

3. Contributing to OSS development community and improving software quality

In addition to giving feedback to Red Hat, we also contribute to the Linux community to improve the quality of the software. Specifically, we submit patches and have discussions with developers through mailing lists and community meetings. Because RHEL is based on OSS products developed by the

Linux community, providing feedback to the community leads to improving the quality of RHEL.

We recently submitted patches to fix bugs in KVM and networking, and they were merged into the latest Linux kernel. We also gave presentations at various conferences related to Linux and KVM, for example, LinuxCon and KVM Forum, to publicly discuss problems with developers around the world. These contributions to the community led to our receiving the Japan OSS Contribution Award and the Japan OSS Incentive Award in February 2013.

We are also testing and improving new features that have not been supported on RHEL. As an example, the latest Linux kernels incorporate Open vSwitch, a software switch that supports OpenFlow^{*2}, which is attracting attention in the cloud computing and network virtualization fields. We have already investigated its internal structure and have tested its performance to enhance our knowledge.

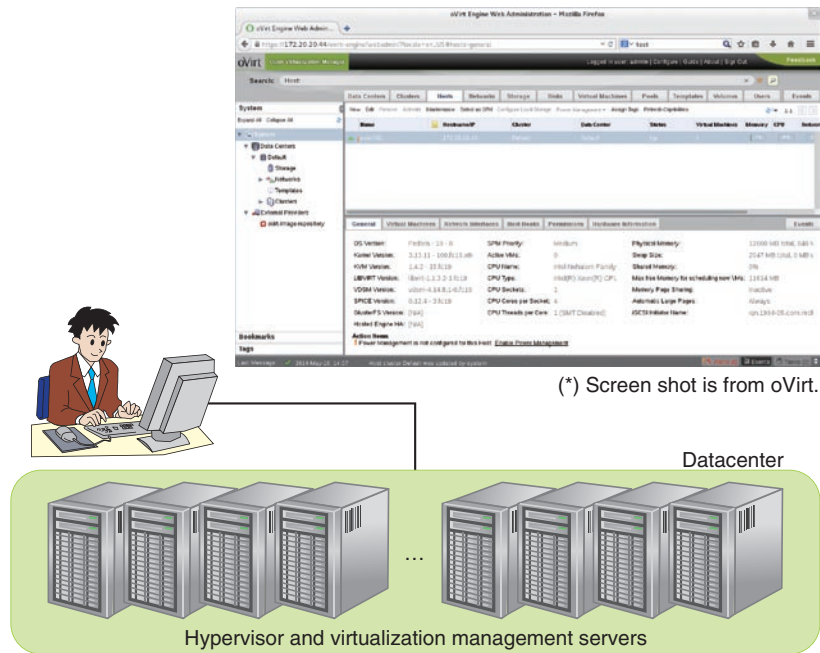
We will continue contributing to the community and trying to improve the quality of OSS products.

4. Further promoting OSS virtualization technology

4.1 Issues in promoting KVM

Although we have been working on KVM, it is not widely used within our group. One of the reasons for this is that the operations management features for

^{*2} OpenFlow: A standard protocol that allows direct access to and manipulation of the forwarding plane of network devices. It is attracting attention as one of the core technologies for network virtualization and SDN (software-defined networking).



(*) Screen shot is from oVirt.

Fig. 3. Centralized management using a virtualization management product.

KVM virtual environments are still inadequate.

Current popular proprietary (non open-source) products that have operations management features generally provide a GUI (graphical user interface) enabling various tasks to be performed easily even if they are complex. On the contrary, there was no such product for KVM, and depending on the task, many commands had to be entered through a CUI (command-line user interface). In this way, the lack of operations management features had been an obstacle to promoting KVM.

4.2 OSS virtualization management products: RHEV and oVirt

The oVirt community was created to develop an open source management product for KVM. Red Hat is testing and improving the product developed by the oVirt community to achieve the quality necessary for enterprise use and is releasing it as RHEV (Fig. 3). Many of the operations management features needed for KVM virtual environments are already implemented in RHEV, and efforts are continuing to add new functions and to improve its quality. We have confirmed that RHEV has no major bugs in its frequently used features and is ready for practical use.

To promote KVM, we will continue working on RHEV and on improving the quality of RHEV and oVirt by feeding back the know-how we gain to the oVirt community.

5. Promoting OSS in the future

As we have explained in this article, using OSS in our group is not just for avoiding vendor lock-in, but for gaining various benefits such as cost reductions and analyzability of software.

We will continue to carry out the functions of the NTT OSS Center by assessing the maturity of OSS technology, promoting the use of OSS within our group, and providing high-quality support. Furthermore, the experience gained through these activities will help us contribute to the OSS community extending throughout the world, which in turn will lead to the further development of OSS.

References

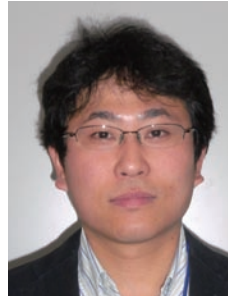
- [1] QEMU-KVM bug fix update, <https://rhn.redhat.com/errata/RHBA-2013-0857.html>



Kenji Yoshida

Manager, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the B.E. and M.E. in electronic engineering from University of Fukui in 1993 and 1995, respectively. He joined NTT in 1995 and has been engaged in research on IP service network engineering. He then moved to NTT Open Source Software Center in 2010, where he has been providing Linux technical support.



Toshio Nakazawa

Senior Expert, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the B.E. in electrical and electronic engineering from Meiji University, Tokyo, in 2002. He joined NTT Software Corporation in 2002 and was involved in developing systems for mobile communication. He then moved to NTT Open Source Software Center in 2010, where he has been providing technical support for Linux. He is now interested in operations management products for KVM and is promoting them within the NTT Group.



Kumiko Goto

Engineer, SE Department Core Technology, Quality Management and Engineering Division, NTT Comware.

She received the B.E. in information and system engineering from Chuo University, Tokyo, in 2005. She joined NTT Comware in 2005, and then moved to NTT Open Source Software Center in 2010, where she was responsible for providing Linux technical support. She moved back to NTT Comware in 2014.



Seiji Kihara

Senior Manager, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the B.S. and M.S. in information science from the Tokyo Institute of Technology in 1990 and 1992, respectively. He joined NTT in 1992. His current research interests include operating systems, computer networks, and open source software. He is a member of the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Association for Computing Machinery (ACM).



Toshiaki Makita

Senior Expert, Infrastructure Software Technology Unit, NTT Open Source Software Center.

He received the B.S. and M.S. in informatics from Kyoto University in 2004 and 2006, respectively. He joined NTT WEST in 2006 and studied Ethernet VPN. Since moving to NTT Open Source Software Center in 2011, he has been interested in networking technologies and developing a Linux kernel networking subsystem.

Dramatic Changes in Java Troubleshooting! —Pursuing Uncompromising Low-overhead Implementations

Yasumasa Suenaga

Abstract

This article focuses on the work of one of the researchers who contributed to a Feature Article in this issue.

Keywords: HeapStats, OSS, Java

In April 2013, the NTT Open Source Software (OSS) Center released HeapStats, a Java Virtual Machine (JVM) monitoring and analysis support tool, as open source software (OSS). This attracted a great deal of interest from various people inside and outside the NTT Group. We also had the gratifying experience of being awarded the Prize for Excellence among Java technologists representing Japan at the Java Day Tokyo 2013 event. Installations of HeapStats within the NTT Group are increasing and have begun to produce good results in analyzing the sources of problems when and where they occur.

The NTT OSS Center provides support and problem analysis for organizations adopting various types of OSS. I primarily handle problem analysis related to Java, and I quite often encounter bugs related to regions of Java heap memory that are not being released properly by garbage collection (GC), the mechanism that recovers unused memory. These are called memory leaks. Analyzing such problems used to be very difficult because to identify the cause, it was necessary to follow the state of the Java heap memory over the time sequence leading to the memory leak. However, in most cases, there was no information indicating when the memory leak occurred, and to get such information, appropriate settings had to be set before waiting for the leak to be reproduced.

Because of this, we began developing HeapStats as a way of gathering information that can be analyzed immediately, without waiting to reproduce the problem.

Java provides an application programming interface (API) for collecting information, so we initially used it to create a prototype. However, when we tried to use it, we found that it caused a large drop in system performance. From past experience supporting development projects, we knew we could not allow any drop in overall system performance when gathering information about problems that occur with low probability. In the field of motor sports, a mechanic uses his utmost skill in tuning his machine to ensure high performance in various environments. Similarly, our goal with HeapStats was to operate with low overhead in various environments (applications). Therefore, we considered every detail down to the central processor unit (CPU) architecture in developing it. We studied the source code of OpenJDK (an open-source implementation of the Java Platform), but also analyzed the JVM executable binary by following processing at the CPU instruction level, and we thoroughly investigated the movement of data within JVM by checking the actual placement in memory and the instruction sequences using a debugger. As a result, we synchronized HeapStats perfectly

with the GC, and in doing so, we established a method for gathering information with the least possible load on the system. Furthermore, we selected CPU instructions more efficiently by introducing SIMD (single-instruction multiple-data) instructions and branch prediction, which are difficult to support using only compiler optimization options, and we repeatedly tested and fine-tuned the implementation. As a result, HeapStats operates efficiently and fully uses

the features of the CPU.

The key to growth for HeapStats is to increase the number of users and developers, so we plan to promote it to Java technologists around the world. We will also strive to improve the features and quality of HeapStats, which will contribute to improving the quality of systems that use Java. My goal is to see engineers around the world turning to HeapStats for Java memory analysis.



Yasumasa Suenaga

Senior Expert, OSS Promotion Unit, NTT Open Source Software Center.

He received the B.E. in information and communication engineering from Tokyo Denki University, Tokyo, in 2006. He joined the NTT Group in 2006. He is a committer of IcedTea (HeapStats), and an author of the JDK9 development community. In April 2014, he moved to NTT Comware.

Inspection Prioritizing Technique for Early Discovery of Defective Facilities—Manhole Facility Inspection Prioritization

Kazuyoshi Kawabata, Jiro Mori, Hiroshi Masakura, Ryo Yamakado, Sho Ashikaga, and Yasuhiro Katsuki

Abstract

There are approximately 680,000 manholes throughout Japan, and as most of these facilities were built more than 30 years ago, aging-related problems have begun to appear. Consequently, the maintenance and management of communications-infrastructure facilities carried out from this point on will become increasingly important and will have to be done in a more efficient manner. In this article, we introduce a technique for analyzing data previously collected during facility inspections and for giving priority to inspecting facilities with a high probability of deterioration in order to achieve early discovery of defective manholes.

Keywords: manholes, communications infrastructure facilities, deterioration

1. Introduction

Communications infrastructure facilities constitute a vitally important social infrastructure for accommodating the underground communication cables that support our advanced information society. Most of these facilities, which include conduits, manholes, and tunnels, were built in the 15-year period from the latter half of the 1960s to the end of the 1970s, which means that about 80% of them are now more than 30 years old. Left as such, these facilities will be more than 50 years old in about 20 years time and will begin to age in earnest (Fig. 1). Concrete structures such as manholes can weaken with age as a result of cracks, peeling, corrosion of reinforcing bars, and other problems.

The ideal way of dealing with aging facilities would be to renovate them in a systematic manner. However, renovating a huge number of underground manholes would be costly and time-consuming and is therefore unrealistic for all practical purposes. In addition, dig-

ging up roads is obviously troublesome for the surrounding environment. For these reasons, there is a need to conduct appropriate maintenance without having to carry out complete renovation so that such facilities can continue to be used for a long period of time.

At present, the inspection of a manhole located under a road requires that approval be obtained for using that road and that safety measures be put into place. It also requires considerable labor such as ventilating the manhole, performing a toxic-gas check, and pumping out standing water before inspectors can enter the manhole and begin the inspection. The current system, moreover, involves inspecting all manhole facilities in a uniform and exhaustive manner on an area-by-area basis.

From here on, defective manholes*¹ must be

*1 Defective manhole: Based on the concept of preventive maintenance, a manhole in early-stage deterioration that can lead to a drop in safe performance.

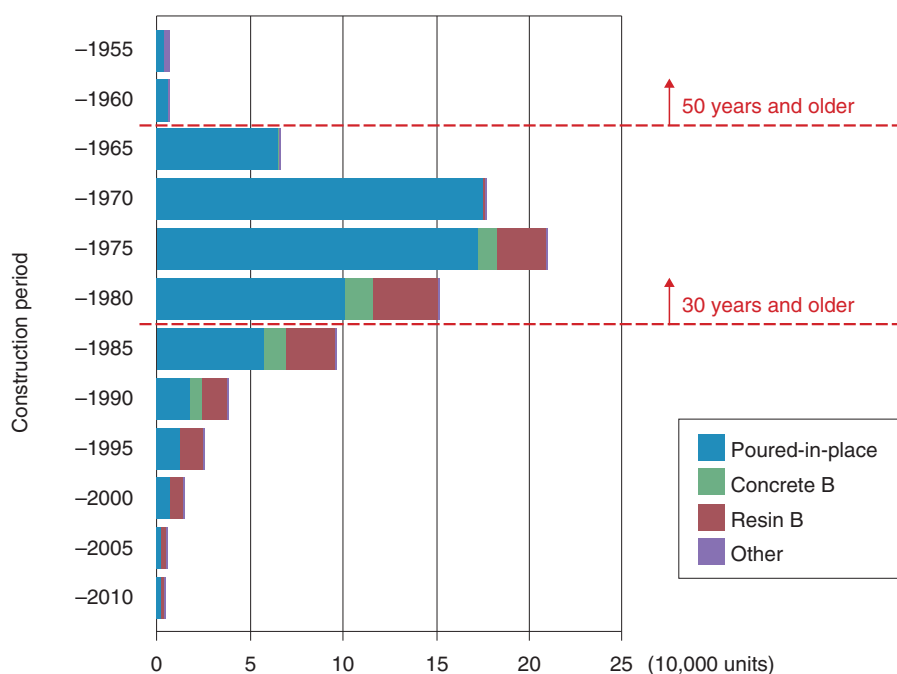


Fig. 1. Number of manholes built in 5-year periods.

discovered early from the huge number of aging facilities and repaired accordingly, and inspections must be tailored to the conditions of individual facilities. With this in mind, we have been developing a condition-based maintenance and management technique that enables us to predict the deterioration of a manhole from its installation environment and facility structure (Fig. 2).

2. Overview of inspection prioritizing technique

Causes of manhole deterioration can be broadly divided into external and internal factors. The former consists of external forces such as soil pressure and vehicle loads, and the latter refers to material deterioration such as the corrosion of reinforcing bars and neutralization of concrete.

In our research, we first clarified the relationship between the generation of cracks—a form of deterioration caused by external factors—and the installation environment. Then, we analyzed past manhole inspection data taking those factors of crack generation into consideration. We used the results to develop an inspection prioritizing technique for extracting facilities likely to be in a state of deterioration and prioritizing them for repair.

3. Relationship between cracks caused by external forces and installation environment

Two types of forces act on manholes buried underground: soil pressure and the load from vehicles passing on the road above. However, the impact on the manhole does not depend solely on the magnitude of those two forces; it will depend greatly on the installation environment such as the type of soil, underground depth, and thickness of the road pavement as well as on the size and shape of the manhole itself. We reproduced these diverse conditions using finite element method (FEM) analysis^{*2}. In creating a FEM model for manholes, we examined the behavior of actual manholes when subjected to realistic conditions by conducting a pressure-application test using a full-size test specimen, and we verified the reproducibility of that behavior (Figs. 3 and 4). We also narrowed down variables of concern in the installation environment to nine items that have a major affect on the external forces acting on the manhole. We combined those items with data on the facility

*2 FEM analysis: A technique for solving engineering problems by approximating a continuum having an infinite number of degrees of freedom using a set of subdomains or elements that each include a finite number of specific unknown quantities.

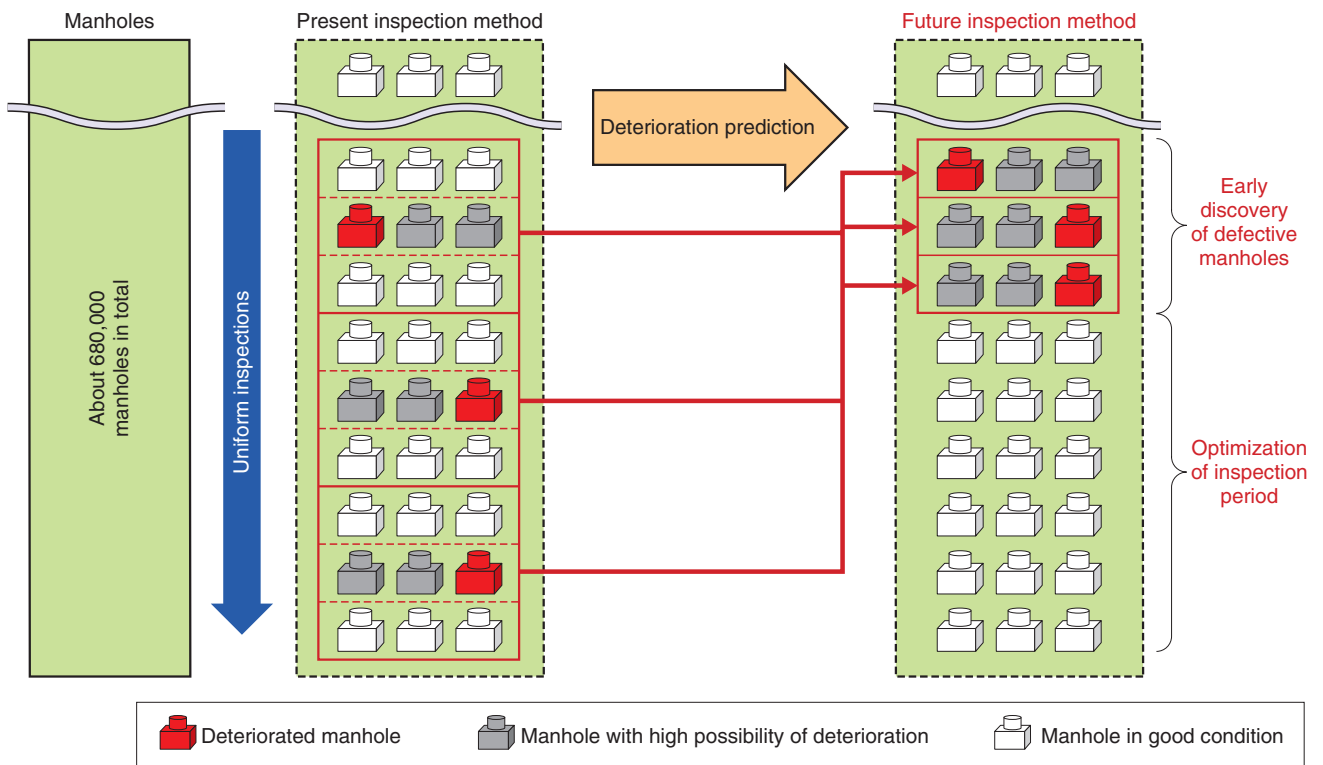


Fig. 2. Future inspection method based on deterioration prediction.

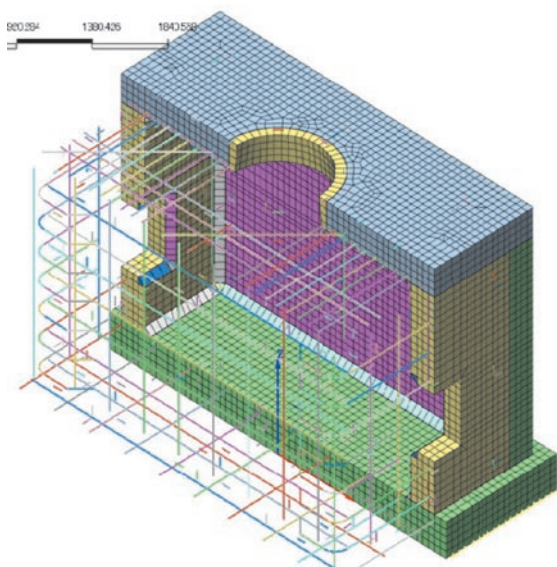


Fig. 3. Manhole FEM model.



Fig. 4. Apparatus for conducting pressure application test.

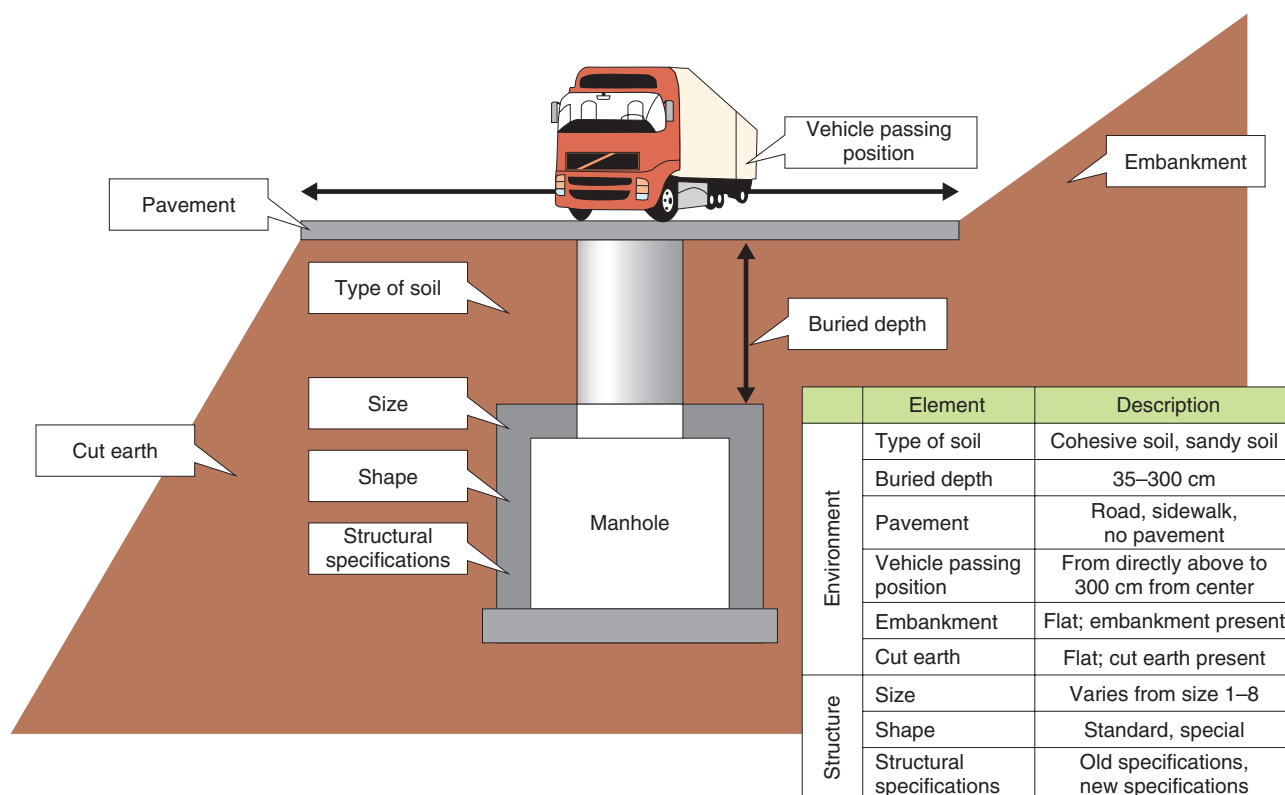


Fig. 5. Elements affecting external forces on a manhole (9 items).

structure and analyzed about 300 patterns of manhole installation states (Fig. 5). Here, on the basis of knowledge obtained through past experiments with concrete specimens that a crack will be generated at a tensile strain of $100 \mu\epsilon$, we set the threshold for crack generation to be the value obtained by multiplying the above value of tensile strain by a safety factor. This factor takes into account an *impact coefficient* related to the difference in grade level between the road and the manhole cover, and *repeated loading*, as obtained from wheel-load experiments.

The results of analysis revealed that the threshold for crack generation set as described above would be exceeded for certain combinations of the underground depth of the manhole, referred to as the *manhole buried depth*, and the *vehicle passing position*. To easily determine those combinations, we created a crack generation map with one axis assigned to the buried depth and the other to the vehicle passing position (Fig. 6). This map revealed that, in general, a shallowly buried manhole receives the load of a vehicle passing directly above and that its upper floor slab (ceiling portion of the manhole interior) experi-

ences a large amount of strain. It also revealed that a deeply buried manhole suffers from the effects of soil pressure and that its lower floor slab (floor portion of the manhole interior) experiences a large amount of strain regardless of the vehicle passing position. We tested these analysis results using actual manhole inspection data from a certain area and found that the defect rate of manholes corresponding to the crack generation areas of this map was indeed 1.5 times greater than that of other manholes.

4. Analysis of inspection data

At first, we analyzed inspection data using only inspection items and the facility information, but identifying the inspection items that had a high possibility of influencing a defective inspection result proved difficult. We therefore decided to focus our attention on items such as vehicle passing position and defects in manhole cover and peripheral pavement, which were found to have a relationship with external forces from the FEM analysis. We then performed data mining. Here, we adopted the Bayesian

Example of a crack generation map

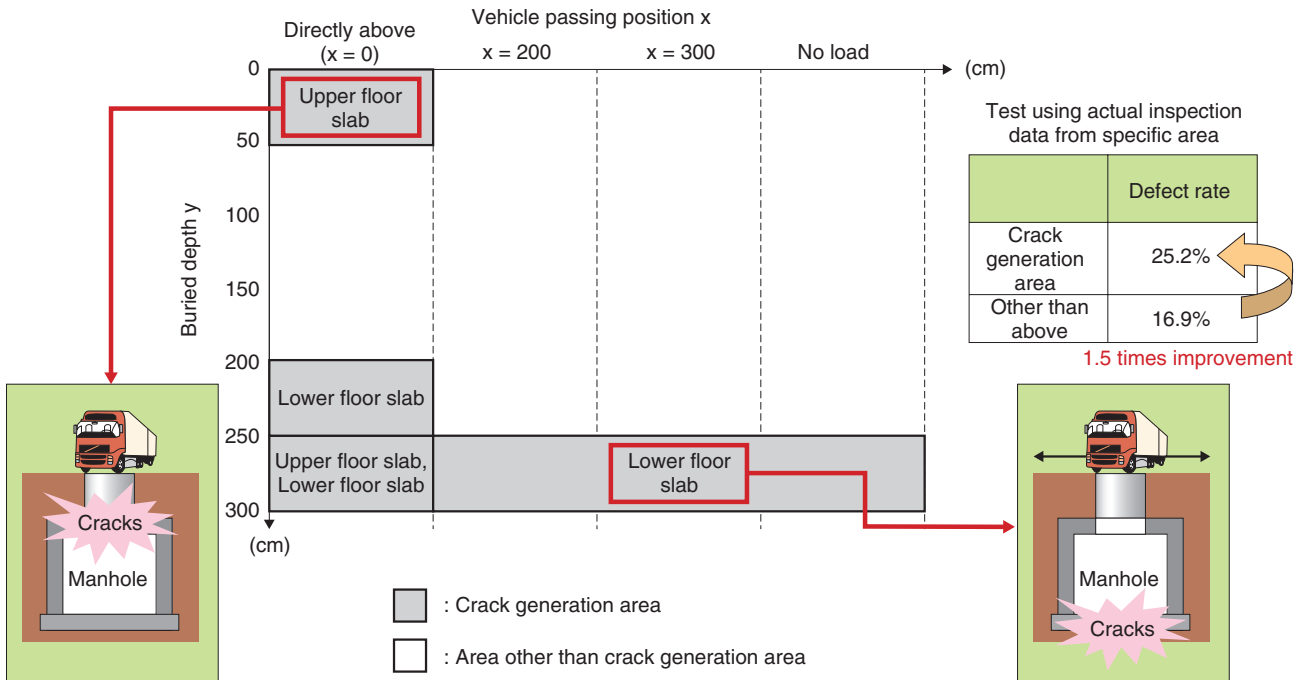


Fig. 6. Crack generation map.

network technique^{*3}, which makes it possible to visualize dependency relationships among data items and flexibly incorporate tacit knowledge of experienced inspectors in the model-creation process.

In this way, we first prepared a network model showing defective inspection results and the dependency relationship among data items. We then improved the acceptance of this model by holding discussions with local maintenance departments and making modifications as needed, and we checked the validity of the defect rates in this network model through a process of trial and error (Fig. 7). In the end, we identified four items having a dependency relationship with an inspection result of *defective* (that is, features common to a defective manhole). These items are *thickness of pavement*, *age of manhole*, *result of inspecting manhole cover (has the manhole cover been defective in the past?)*, and *vehicle passing position*, the last of which also played a role in the FEM analysis. These items included those that had not been obtained as inspection data in the past, and we adjusted for their future acquisition with the NTT business companies of concern.

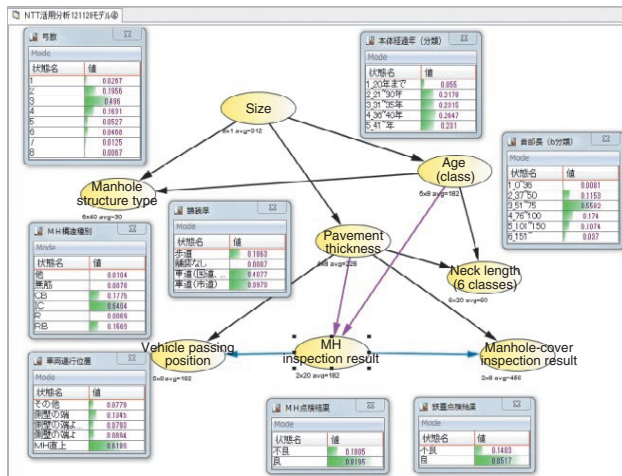
5. Inspection prioritizing technique

Inspection prioritizing begins by dividing up the manholes targeted for inspection according to whether a manhole falls within a crack generation area in the crack generation map prepared by FEM analysis. This step gives priority to manholes that are heavily affected by external forces. Next, we take the four items found to be common to defective facilities in the analysis of actual inspection data, classify each manhole according to a certain combination of those items, and sort the manholes according to the defect rates previously calculated for those combinations. In this way, priority can be given to inspecting those manholes that are heavily affected by external forces and that have a high defect rate (Fig. 8).

We conducted a case study using about 9000 sets of manhole data. Assuming that about 20% of all manholes can be inspected in one year, we prioritized their inspection using the technique described above. Compared to the conventional inspection method, the

*3 Bayesian network technique: A type of probabilistic model that can be used for predicting phenomena, making rational decisions, and identifying faults under uncertainty.

(a) Network model



(b) Defect rates (example)

Pavement thickness	Age (years)	Vehicle passing position	Cover inspection result	Defect rate
No pavement	41 +	Directly above	Bad	90.0%
No pavement	36-40	Directly above	Bad	80.0%
Sidewalk	41 +	Directly above	Bad	70.0%
No pavement	30-36	Directly above	Bad	60.0%
No pavement	41 +	Within 200 m	Bad	50.0%
Sidewalk	41 +	Within 200 m	Bad	40.0%
No pavement	26-30	Directly above	Bad	30.0%
No pavement	20-36	Directly above	Bad	20.0%
No pavement	41 +	Directly above	Good	10.0%
No pavement	36-40	Directly above	Good	8.0%
Sidewalk	36-40	Directly above	Bad	6.0%
Sidewalk	30-36	Directly above	Bad	4.0%
Roadway	41 +	Directly above	Bad	2.0%

Fig. 7. Inspection data analysis.

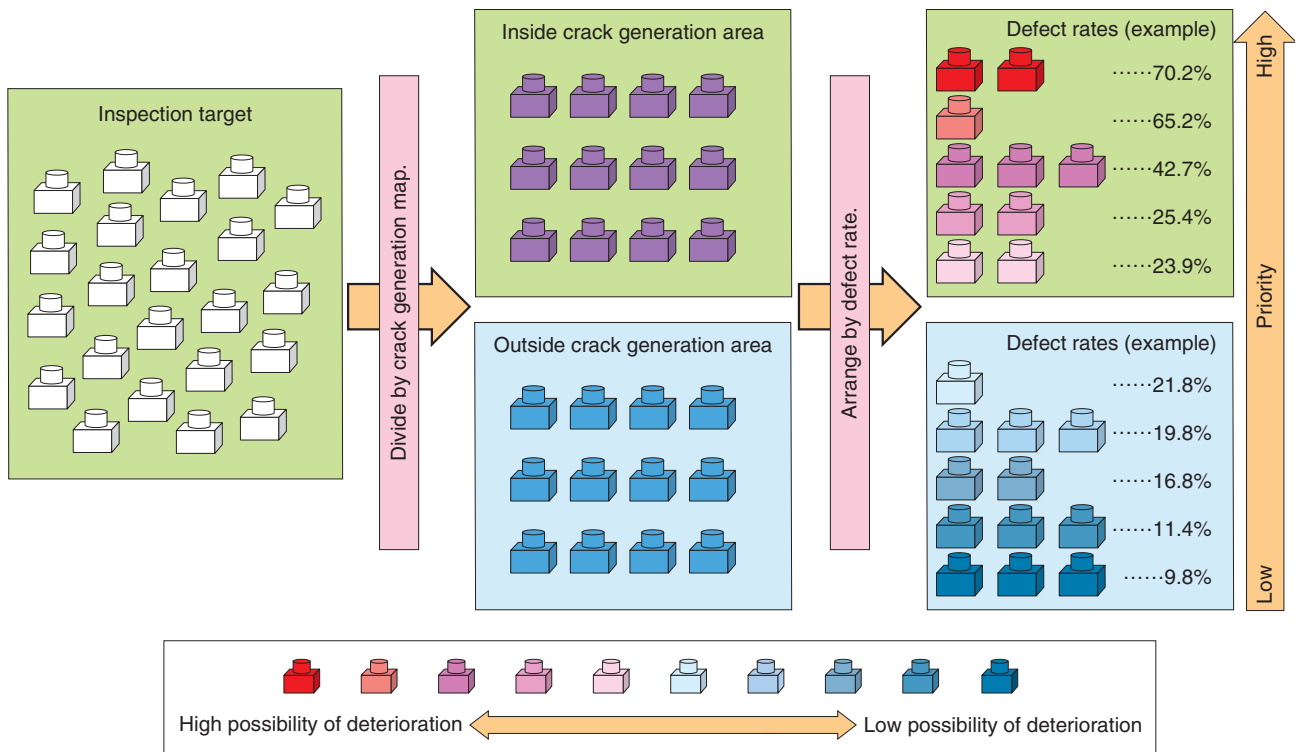


Fig. 8. Application scenario.

proposed technique improved the discovery rate of defective manholes by up to two times in the first year, and by the third year, we were able to discover 80% of the defective manholes.

In fiscal year 2013, we used this technique to prioritize facilities targeted for inspection in a certain area and conducted manhole inspections. At first, we were concerned that inspection efficiency would decrease since the manholes targeted for inspection were extracted without regard to contiguous installation locations and were consequently scattered in various areas. However, since many causes of manhole deterioration originate in the characteristics of roads, the facilities extracted using the proposed technique tended to be located along the same routes. Although only half the number of targeted manholes has so far been inspected in that area, the discovery rate of defective manholes is running at about twice that of the conventional method. These results demonstrate that the proposed technique is practical and

effective for early discovery of defective manholes.

6. Future plans

At present, we are focusing our studies on the speed of corrosion in concrete reinforcing bars as a type of deterioration caused by internal factors. Our objective here is to set an optimal inspection period based on the progression of deterioration. We will continue to research and develop techniques for appropriately maintaining and managing infrastructure facilities at low cost with the aim of contributing to enhanced operations in the NTT Group.

Reference

- [1] F. Sugino, "R&D Trends in Communications Infrastructure to Achieve Safe and Secure Access Networks," NTT Technical Review, Vol. 11, No. 6, 2013.
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201306fa2.html>



Kazuyoshi Kawabata

Group Leader, Facilities Management Group, Civil Engineering Project, NTT Access Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in civil engineering from Tokyo Institute of Technology in 1988 and 1990, respectively. He joined NTT in 1990. He has been engaged in R&D of communications-infrastructure facilities.



Ryo Yamakado

Research Engineer, Concrete Structure Group, Civil Engineering Project, NTT Access Network Service Systems Laboratories.

He received the B.E. degree in mechanical engineering from Keio University, Kanagawa, in 1992. He joined NTT the same year. He has been engaged in R&D of communications-infrastructure facilities.



Jiro Mori

Senior Research Engineer, Facilities Management Group, Civil Engineering Project, NTT Access Network Service Systems Laboratories.

He received the B.E. degree in civil engineering from Osaka City University in 1990. He joined NTT the same year. He has been engaged in R&D of communications-infrastructure facilities.



Sho Ashikaga

Engineer, Facilities Management Group, Civil Engineering Project, NTT Access Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in civil engineering from Hokkaido University in 2009 and 2011, respectively. He joined NTT in 2011. He has been engaged in R&D of communications-infrastructure facilities.



Hiroshi Masakura

Senior Research Engineer, Civil Engineering Project, NTT Access Network Service Systems Laboratories.

He received the B.E. degree in civil engineering from Ehime University in 1989. He joined NTT the same year. He has been engaged in R&D of communications-infrastructure facilities.



Yasuhiro Katsuki

Engineer, Facilities Management Group, Civil Engineering Project, NTT Access Network Service Systems Laboratories.

He received the B.E. degree in electronics from Doshisha University, Kyoto, in 2007. He joined NTT the same year. He has been engaged in R&D of communications-infrastructure facilities.

Standardization Trends Relating to Smart TV

Kiyoshi Tanaka

Abstract

Smart TV (television), which implements advanced TV services, has been receiving a lot of attention throughout the world recently. Standardization of smart TV as technology relating to receivers and services is underway by both the International Telecommunication Union and the World Wide Web Consortium. These standardization trends are explained in this article.

Keywords: smart TV, IPTV, Web and TV

1. Introduction

Smart TV (television) has recently become a popular topic with electrical appliance retailers and in the media such as newspapers and magazines. It is instinctively understood that smart TV is an enhanced version of what we call TV, but the emphasis depends on the standpoint as to whether TV receivers themselves should be enhanced or whether TV broadcast services should be enhanced. In other words, the former standpoint is mainly held by TV receiver manufacturers who sell their own smart TVs. It concerns the addition of functions to TV receivers by installing a variety of applications in them, similar to the way smartphones are enhanced mobile phones. The latter standpoint tries to diversify program delivery services by combining broadcast services and communication services, for example, Hybridcast^{*1}, a broadcast service that started in Japan in September 2013.

What does Internet Protocol TV (IPTV), which implements broadcast services on IP networks, have to do with smart TV? According to the definition of IPTV given by the International Telecommunication Union (ITU) [1], IPTV is defined as “multimedia services such as television/video/audio/text/graphics/data delivered over IP based networks managed to provide the required level of QoS/QoE (quality of service/quality of experience), security, interactivity and reliability.” On the basis of this definition, the services that will implement smart TV are expected to

be handled the same way IPTV services are handled. That is why there have been discussions at the ITU on the relevance to IPTV of the standardization of smart TV.

A recent development is the increasing use of web technology to provide video delivery services over the Internet in response to the conversion of access lines to broadband and the increasing speeds of the Internet. In June 2012, the basic strategy towards the promotion of smart TV [2] was published by the Ministry of Internal Affairs and Communications to promote smart TV as a new service that combines broadcasting and the web. The objectives are to fulfill three basic functions: combining broadcasting and the web, providing a wide range of applications and content, and establishing links between terminals, and also to exploit these basic functions to the maximum extent in order to increase user-friendliness and broaden user options, and to expand the market. However, the key to all this is thought to be web technology, starting with HTML5 (Hypertext Markup Language, 5th revision). From these viewpoints, attention is drawn to web technology relating to HTML5, and its standardization is centered on the World Wide Web Consortium (W3C).

We will now introduce trends in standardization

^{*1} Hybridcast: TV service that brings together broadcasting and broadband networks; standards are being drawn up by the IPTV Forum in Japan.

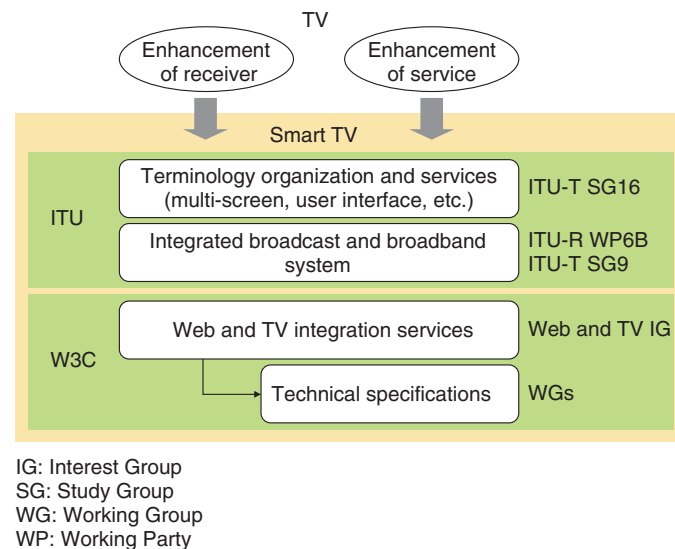


Fig. 1. Standardization of smart TV.

relating to smart TV in the ITU and W3C (see **Fig. 1**).

2. Standardization relating to smart TV by ITU

2.1 Studies by leading IPTV groups

In the ITU-Telecommunication Standardization Sector (ITU-T) Study Group (SG) 16 (multimedia coding, systems, and applications), investigations relating to IP-based TV-related multimedia services, including smart TV, were added to the Terms of Reference of question Q13/16, which concerns multimedia application platforms and end systems for IPTV, in the new session that began with a meeting in January 2014. To that end, Q13/16 prompted the start of investigations of terminology, since many services that resemble smart TV are provided around the world.

The ITU-T draft technical paper HSTP.IPTV-Gloss^{*2} is a technological document that was drawn up with the purpose of clarifying and compiling terminology, and section 7.2 covers smart TV. In the current description, features such as “a new hybrid terminal device reflecting the current trend of applying Internet and Web 2.0 technologies to television sets and set-top boxes” and “social networking, games, interactive advertising, and IPTV as well as traditional broadcast TV channels” appear, but the parts relating to terminals and services are seen as expansions of IPTV.

Note that with Q13/16, recommendations for IPTV multimedia services are created in harmony with

ISDB-T^{*3} (Integrated Services Digital Broadcasting-Terrestrial) standards. The upstream activities of specifications of the IPTV Forum in Japan have been achieved by Japanese participants who are playing a central role. This means that the ITU-T standards have been developed with a strong correlation with the broadcast standards of Japan. Recently, a Q13/16 discussion has shifted to work items related to multi-screen, multi-device services focused on by China, and to enhanced user interfaces such as voice recognition, gesture inputs, and touchpad controls other than remote controls, focused on by South Korea. However, there are calls for an assessment of future trends from the viewpoints of conformity with Japanese standards and the effects on new services.

2.2 Standardization from the broadcast viewpoint

When smart TV is viewed with a focus on broadcasting, its aspect as an integrated service of broadcast and broadband networks becomes stronger. From that viewpoint, recommendations relating to the Integrated Broadcast-Broadband digital television system (IBB system) are in progress in Working Party (WP) 6B (Broadcast service assembly and access) under

*2 HSTP.IPTV-Gloss: Technical Paper: Glossary and terminology of IP-based TV-related multimedia services.

*3 ISDB-T: Terrestrial digital TV broadcast standards drawn up by the Association of Radio Industries and Businesses (ARIB), which is used in terrestrial digital broadcasting within Japan and other countries.

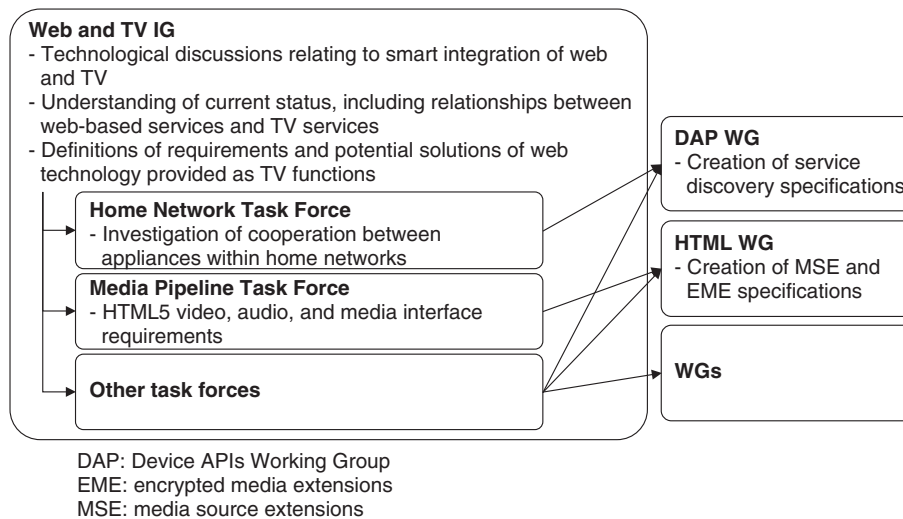


Fig. 2. Relationships between Web and TV IG and WGs.

SG6 (Broadcasting service) of the ITU-Radio Communication Sector (ITU-R) and ITU-T SG9 (Broadband cable and TV). In ITU-T SG9, ITU Recommendation J.205 on the system requirements was consented in January 2012, and ITU-T Recommendation J.206, related to architecture, was consented in January 2013. In ITU-R WP6B, a draft new recommendation for the requirements of an IBB system corresponding to ITU-T recommendation J.205 was drawn up, and at a meeting in October 2012, information related to various IBB systems was collected, and a work item toward a new report was drawn up. At a meeting in April 2013, a proposal from Japan to include Hybridcast services in the report was submitted and it was included in the report ITU-R BT.2267 together with information on the Broadcast Markup Language (BML) Type 2, which is used in Japanese terrestrial digital broadcasting. This was issued in August 2013.

3. Discussions relating to smart TV in W3C

3.1 Web and TV

W3C discussions relating to smart TV were vigorous in the first workshop held in Tokyo in September 2010; these discussions concerned studies related to services that integrate the web and TV. The participants at this meeting were stakeholders with various standpoints such as telecom operators, broadcasters, and manufacturers. The constitution of a Web and TV Interest Group (IG) for the discussion of technologi-

cal requisites and the relationships between web-based services and TV services was agreed upon. As a result of this constitution, the second workshop was held in Berlin in February 2011, where it became clear that it was necessary to investigate areas such as cooperation between appliances within home networks, adaptive bit rate streaming, and content protection. The third workshop was held in Hollywood in September 2011, where content providers and service providers mingled, selected topics from their own viewpoints, and pitched their implementation through demo sessions. Through these workshops, task forces (TFs) were set up within the group, and studies into use cases and requirements proceeded (Fig. 2).

3.2 Investigations

Details of the main investigations conducted by TFs are introduced as follows.

(1) Home Network Task Force (HNTF)

Multi-screen scenarios involving the use of a number of screens within the home, for example, TVs and tablets, have been studied, but it is still necessary to discover and manage devices and services within local home networks. Since gap analysis of existing technology is required, the HNTF has been discussing various use cases in consideration of factors such as Universal Plug and Play (UPnP)^{*4} and APIs

^{*4} UPnP: Protocol for cooperation between network devices, established by the UPnP forum.

specified by CableLabs, a non-profit research and development consortium. During this time, they studied topics including the discovery of services and content, the control of content players and recorders, content protection, and security and privacy, and a requirements document [3] was drawn up. Since the IG of the W3C only implemented requisite investigations, the drawing up of concrete specifications was implemented by a working group (WG), but discussions relating to these requirements, such as an investigation of service discovery APIs by the Device APIs WG (DAP WG), have continued.

(2) Media Pipeline Task Force (MPTF)

This task force was set up with the objectives of investigating the requirements for HTML5 video, audio, and media interfaces, and proposing APIs for them, in order to implement services that bring together the web and TV. The MPTF discussed studies concerning the insertion of advertisements, event synchronization, and requirements such as audio sub-channels and closed captioning, in order to implement each country's regulations and also transfer the existing TV services required by consumers onto HTML5 browsers. They also collected the requirements for adaptive bit rate streaming [4] and content protection [5] to ensure that content can be delivered

stably and securely, even on networks that cannot guarantee reliability, such as the Internet. These requirements are linked to discussions of specifications such as those of media source extensions (MSE) and encrypted media extensions (EME) by the HTML WG.

4. Future expansion

As the standardization relating to smart TV in the ITU and W3C continues, it seems that interest in smart TV will increase throughout the world. In Japan, the provision of full-fledged smart TV services is expected in the near future.

References

- [1] Recommendation ITU-T Y.1901: "Requirements for the Support of IPTV Services," Geneva, Jan. 2009.
- [2] News release of the Ministry of Internal Affairs and Communications (MIC) (in Japanese).
http://www.soumu.go.jp/main_sosiki/joho_tsusin/hyojun/smart.html
- [3] W3C, Requirements for Home Networking Scenarios.
<http://www.w3.org/TR/hnreq/>
- [4] W3C, MPTF Requirements for Adaptive Bit Rate Streaming.
<https://dvcs.w3.org/hg/webtv/raw-file/tip/mpreq/adbreq.html>
- [5] W3C, MPTF Requirements for Content Protection.
<https://dvcs.w3.org/hg/webtv/raw-file/tip/mpreq/cpreq.html>



Kiyoshi Tanaka

Senior Research Engineer, Supervisor, Service Harmonization Project, NTT Service Evolution Laboratories.

He received the B.E., M.E., and Ph.D. in communication engineering from Osaka University in 1992, 1994, and 2005, respectively. He joined NTT in 1994 and since then has been engaged in research on video-on-demand systems and metadata-related interactive video systems and services, especially those related to IPTV systems and services. His current interests include standardization of IPTV, digital signage, and multimedia broadcasting for mobile devices. He joined the standardization efforts of ARIB STD-B38 (v2.1, 2.2, and 2.3), STD-B45 (v2.0), and TR-B33 as an editor. He has also contributed to ITU-T's international standardization efforts, mainly in ITU-T SG16 and IPTV-GSI. He is a member of W3C Web and TV IG, the Institute of Electronics, Information and Communication Engineers, and the Human Interface Society in Japan.

Fault Cases and Countermeasures for Optical Fiber Cables in Optical Network Facilities

Abstract

This article introduces case studies of failures that have occurred in optical fiber cables as well as some countermeasures against such failures. This is the twenty-third of a bimonthly series on the theme of practical field information on telecommunication technologies. This month's contribution is from the Access Network Engineering Group, Technical Assistance and Support Center, Maintenance and Service Operations Department, NTT EAST.

Keywords: optical fiber cables, access equipment, fault cases

1. Introduction

The number of FLET'S HIKARI optical fiber broadband service subscribers reached over 17.8 million as of the end of March 2014. In contrast, the number of subscribers using PSTN (Public Switched Telephone Network) consisting of metallic cables was recorded at 27 million as of the end of the same period. The Access Network Engineering Group has been investigating failures that occur in access network facilities including optical fiber cables and metallic cables.

We investigated a range of failure items in 2012, as shown in **Fig. 1**. The figure indicates that optical fiber cable and optical connection devices account for nearly half of the total failures. We reported fault cases of field assembly connectors in optical access facilities in a previous issue (July 2011) [1]. Therefore, we report here the fault cases of optical fiber cables investigated in 2012.

2. Fault case of damage to an optical drop cable

One case involved an optical drop cable that was damaged during the installation of a new metallic cable along a conduit line. We investigated the cause of the damage to the optical drop cable.

2.1 Overview of damage

A new metallic cable was installed in the conduit line connecting a handhole to a multi-dwelling unit (**Fig. 2**). The installation involves first laying a rod for cables and using the rod to lay a polypropylene string. The rod is then removed, and the polypropylene string is used to install a metallic cable. Finally, the polypropylene string is removed.

Just after the installation, we received a report about an interruption in the optical communication service provided by the previously installed optical drop cable. That cable had four fibers in the same conduit line. An investigation revealed damage to the optical drop cable in the conduit line about 30 cm from the handhole entrance.

2.2 Results of investigation

The damaged optical drop cable that was removed from the conduit line is shown in **Fig. 3**.

The form of the damage (damage point 1) consisted of a severe bend in the cable and a split and detached cable jacket. Another point of damage consisting of a trench (damage point 2) was located 15 cm from damage point 1.

We confirmed using X-ray imaging that two optical fibers were broken at damage point 1. Moreover, we also confirmed with a video microscope that the UV (ultraviolet) coating on the broken optical fibers was

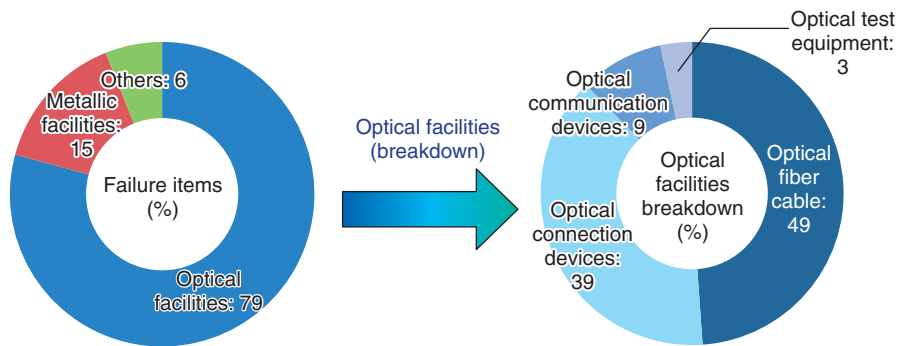


Fig. 1. Failures of access network facilities the Access Network Engineering Group investigated in 2012.

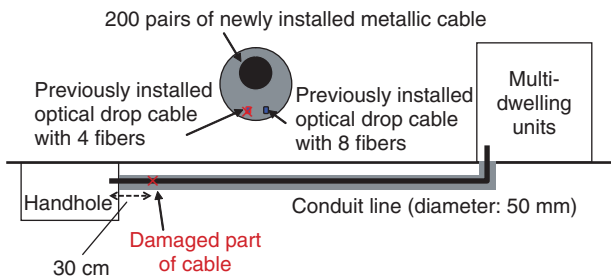


Fig. 2. Configuration of underground facilities where optical drop cable was damaged.

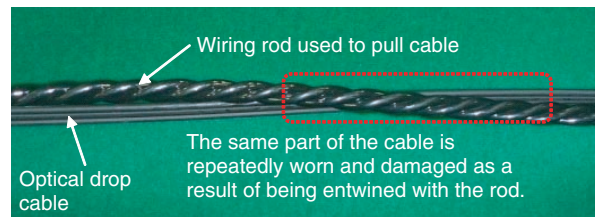


Fig. 4. Reproduction of cable damaged by entwining the rod and cable together.

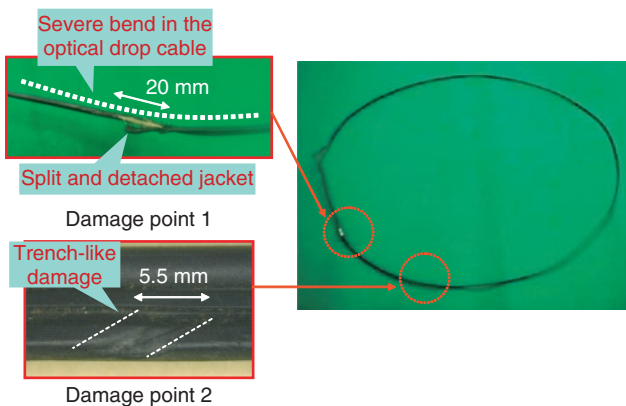


Fig. 3. External view of damaged optical drop cable.

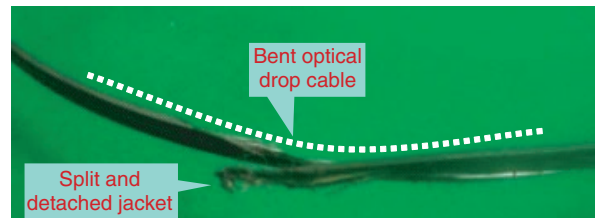


Fig. 5. Reproduced damage in optical drop cable.

stripped, leaving the bare fibers exposed.

2.3 Reproduction of cable damage

We investigated whether installation of metallic cable can damage the optical drop cable. Specifically, we entwined and tensioned a piece of optical drop

cable with wiring rod, with polypropylene string, and with metallic cable, respectively, as shown in **Fig. 4**, in order to intentionally damage the cable. We then compared the resulting damage with the damage from the removed cable to see if the same kind of cable bend, jacket detachment, and optical fiber breakage occurred. Finally, we confirmed that the damage that occurred when the wiring rod was used most resembled that of the removed cable (**Fig. 5**).

2.4 Discussion

In our examination, we confirmed that the optical drop cable was damaged when it was pulled after

being entwined with the wiring rod. Therefore, we recommended two countermeasures to prevent this damage: (i) Install a new conduit line to avoid using multiple threading cables in the same conduit, and (ii) in the inevitable case that new metallic cable is installed in the conduit line containing previously installed optical drop cable (in the manner described in this report), the metallic cable must be placed very carefully. Then the installer should check whether excessive force has been applied or whether existing cables have been moved.

3. Fault case of damage to aerial optical fiber cable caused by fire

We describe here a fault case involving the interruption of optical telecommunication service via an aerial optical fiber cable damaged by fire six years previously.

3.1 Overview of damage

After receiving a report of a disruption of the optical telecommunication service, maintenance personnel searched for the fault location in the aerial optical network. They found a breakage in the sheath of two optical fiber cables. Upon closer inspection, they found that one optical fiber cable had bulges and a hole in the sheath, and the other cable had breakages, as shown in Fig. 6.

3.2 Investigation at location of the damaged optical fiber cable

We interviewed the local fire department and found that three wooden houses had been completely destroyed by fire about six years earlier in an area near the damaged optical fiber cable. The damaged cables were directly downwind of the fire when the

fire occurred.

3.3 Reproduction of the damaged cable

Fires of the type that consumed the wooden buildings typically have temperatures exceeding 1000°C (according to a fire handbook published by Kyoritsu Shuppan Co., Ltd. in 1997). The cable sheath material generally begins to deform at temperatures over 150°C. Therefore, we heated the same type of aerial optical cable in a heater in order to replicate the deformation of the optical cable by heat. When the surface temperature of the cable exceeded 150°C, the cable sheath started to soften. After 15 minutes, the cable deformed, and after 60 minutes, it started to

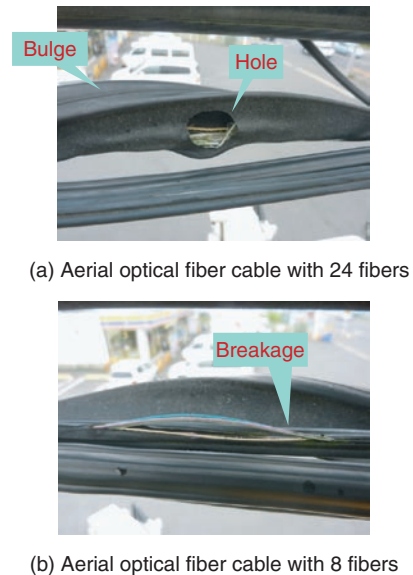


Fig. 6. External view of damaged aerial optical fiber cables.

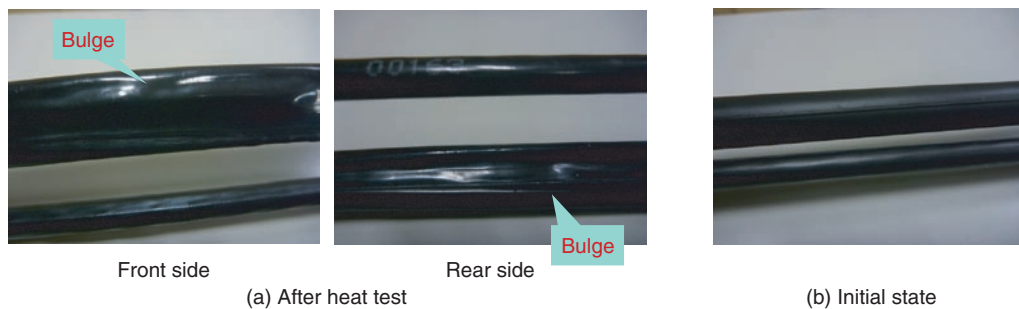


Fig. 7. Replicated damage in aerial optical fiber cable.

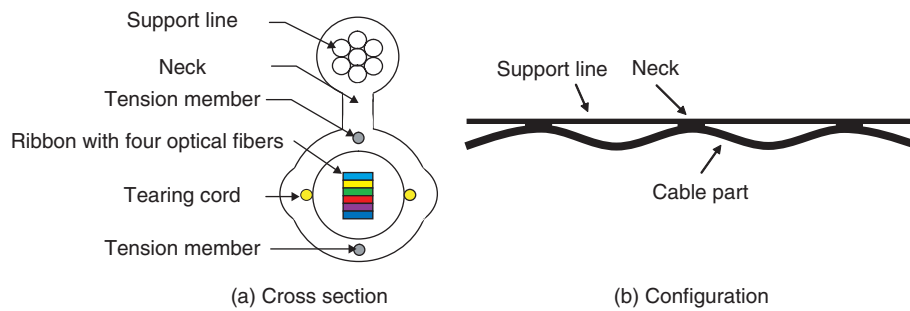


Fig. 8. Configuration of aerial optical fiber cable with 24 fibers.

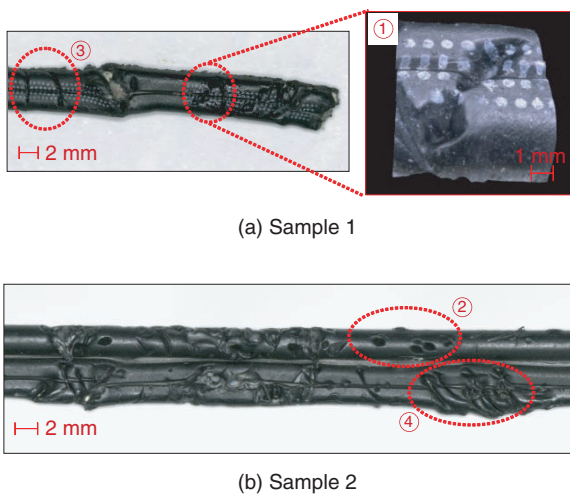


Fig. 9. Samples of damaged optical drop cables.

bulge (Fig. 7), and its shape resembled that of the piece of removed cable described in the previous section. The configuration of this type of aerial optical fiber cable with 24 fibers is shown in Fig. 8.

3.4 Discussion

From the results of the above investigation, we presumed that the change in the shape of the cable sheath and the support-line surface was caused by the high surface temperature, which reached about 150°C due to the heat from the fire. Moreover, as a result of the cable surface being stretched by the force of the tension member inside the optical fiber cable, the sheath starts to tear, and after many years of exposure to weather and UV light, it degrades and eventually breaks. It is difficult to confirm the state of damage to an aerial optical fiber cable after a fire by conducting only a visual inspection from the ground. Therefore,

as a countermeasure, it is necessary to carry out inspections from close up using an aerial work platform.

4. Fault cases of damage to optical fiber cable by wildlife

We introduce two fault cases of damage to optical fiber cables caused by wildlife.

4.1 Damage by arboreal mammals (such as martens and civets)

The first fault case involved an optical drop cable that was cut during the night. A lot of scarring was found on the sheath of the damaged cable, and the optical fibers inside the damaged cable were completely severed. These damaged optical drop cables are shown in Fig. 9. The appearance of damage on the sheath of the optical drop cable was different from the damage caused by the wildlife we had investigated in previous cases. These fault cases happened at night, but they were in areas where there were no nocturnal bird species with beaks capable of pecking through optical drop cables. The damage was also quite different from the damage caused by the incisors of rodents such as squirrels and mice. Therefore, we presumed that the damage to the cable was caused by an arboreal mammal such as a marten or civet.

4.2 Damage due to powder-post beetles (*Bostrichidae*)

The other fault case involved an optical drop cable that had damage with a different appearance. The damaged area was littered with small punctures that had a diameter of 1 mm. The *Cryptotympana facialis* (a kind of cicada) is a familiar culprit of puncture damage [2], but we had not received any reports of damage to optical drop cables due to *Cryptotympana*

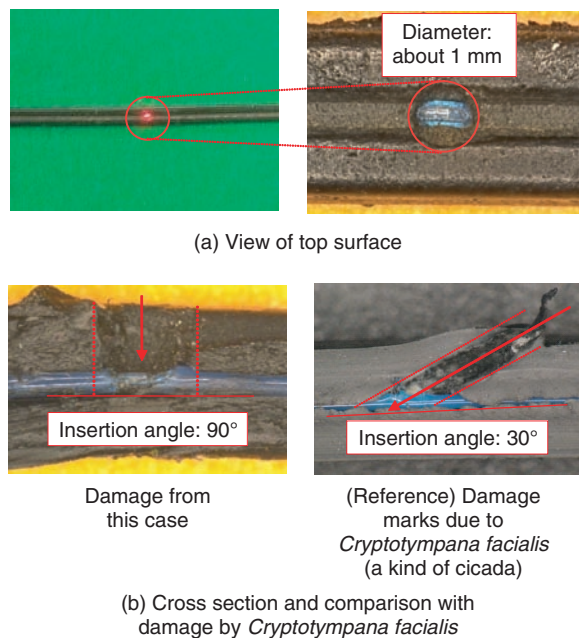


Fig. 10. Damage marks on optical drop cables.

facialis in the area where this fault occurred. The traces of damage to the optical drop cable were near the center section of the cable. The punctures had a circular hole-like shape with a diameter of 1 mm and were in the notch section of the optical drop cable (**Fig. 10**).

We examined the punctures and found that they had an insertion angle of 90°. In contrast, a typical puncture caused by the ovipositor of *Cryptotympana facialis* is characterized as a stab that occurs at an insertion angle of about 30°. This difference in the insertion angle led us to conclude that the damage was not caused by *Cryptotympana facialis*. Moreover, we confirmed that no carcasses or eggs of these insects were found in the extracted section of the optical drop cable.

We also found five semicircles 0.1 mm in diameter where the UV coating had been stripped away. The damage was similar to that caused by a bamboo powder-post beetle (*Bostrichidae*) examined in previous investigations. In particular, the width of the tip of the mandible of a bamboo powder-post beetle is about 0.1 mm, and the surface of the damaged sections is characterized by marks that appear to be less than or equal to 0.1 mm in size. Moreover, the damage marks on the optical drop cable had an insertion angle of 90°. In light of these facts, we inferred that the insect that inflicted the damage was *Bostrichidae*.

5. Concluding remarks

In this article, we reported fault cases of damage to optical fiber cables and proposed some countermeasures against such damage. The Technical Assistance and Support Center reached its 25th anniversary in 2012, and last year marked the 50th year of our technical collaborations (including those of our previous incarnation, the Technical Collaboration Department). Utilizing the knowledge and experience that we have accumulated up until the present, we will strive to continuously improve the reliability of access facilities and equipment while implementing measures to reduce malfunctions and service failures.

References

- [1] NTT EAST, "Fault Cases and Countermeasures for Field Assembly Connectors in Optical Access Facilities," NTT Technical Review, Vol. 9, No.7, 2011.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201107fa2.pdf&mode=show_pdf
- [2] K. Nakano, K. Shiraishi, Y. Maehara, O. Inoue, A. Daido, K. Takamizawa, and T. Numata, "Lower-cost Cicada-resistant Optical Fiber Drop Cable with Better Workability," NTT Technical Review, Vol. 9, No. 6, 2011.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201106ra2.pdf&mode=show_pdf



New NTT Colleagues

—We welcome our newcomers to the NTT Group

Here, we welcome newcomers to the NTT Group. This is a corner of the NTT Technical Review where we introduce our new affiliate companies.

NextiraOne **Established in 2002, Headquartered in France, ICT solutions service provider**

NextiraOne is a European multinational company that designs, installs, maintains, and supports business solutions and communications services for private and public sector customers throughout Europe. Using its expertise in leading-edge communications, including datacenters, contact centers, unified communications, secure network infrastructures, and managed services, NextiraOne helps its customers to transform their organizations, making the complex simple. NextiraOne has 1850 permanent employees in Austria, Belgium, the Czech Republic, Germany, Hungary, Ireland, Luxembourg, the Netherlands, Poland, Portugal, Slovakia, Spain, and the United Kingdom.

Contacts:

Dimension Data

<http://www.dimensiondata.com/Global/Downloadable%20Documents/Dimension%20Data%20Completes%20Acquisition%20of%20NextiraOne.pdf>

Papers Published in Technical Journals and Conference Proceedings

Scalability Analysis of Source Routing Multicast for Huge Numbers of Groups

Y. Katayama, T. Inoue, N. Takahashi, and R. Kawamura
IEICE Trans. Fundamentals, Vol. E96-B, No. 11, pp. 2784–2794, 2013.

Source routing multicast has been gathering more attention than traditional Internet Protocol (IP) multicast, since it is thought to be more scalable in terms of the number of groups at the cost of higher traffic loads. This paper introduces a mathematical framework to analyze the scalability of source routing multicast and IP multicast by leveraging previous multicast studies. We first analyze the amount of data traffic based on the small-world nature of networks, and show that source routing multicast can be as efficient as IP multicast if a simple header fragmentation technique (subgrouping) is utilized. We also analyze scalability in terms of group numbers, which are derived under the equal budget assumption. Our analysis shows that source routing multicast is competitive for low bit-rate streams, like those in the publish/subscribe service, but we find some factors that offset the advantage. This is the first work to analytically investigate the scalability of source routing multicast.

Study on MIMO Transmission Using Orthogonal Directivities Obtained from Higher Order Microstrip Antenna Modes

M. Arai, T. Seki, K. Hiraga, K. Sakamoto, H. Toshinaga, T. Nakagawa, and K. Uehara

IEEE Electron. Lett., Vol. 50, No. 7, pp. 562–564, 2014.

This paper proposes a simpler multiple-input multiple-output (MIMO) transmission method by using orthogonal directivities. Higher data rates will be required since large amounts of data will be transmitted and received. MIMO technology is useful in order to achieve the higher data rates. However, since MIMO detection becomes complicated when the number of the antennas becomes large, a simpler method should be considered. To solve the problem, we previously proposed a spatial division method using orthogonal directivities. In this paper, we expand our previous method to N streams' transmission. Furthermore, we propose using higher order antenna modes at the same frequency to more easily obtain the orthogonal directivities. To evaluate the proposed method, we confirm that the capacity of the proposed method increases without MIMO detection when the number of antennas becomes large.

Performance Evaluation of a Simple Decoding Method for Millimeter-wave Short-range MIMO Transmission through a Wall

K. Sakamoto, K. Hiraga, T. Seki, T. Nakagawa, and K. Uehara
IEICE Communications Express, Vol. 3, No. 4, pp. 131–137, 2014.

The simple decoding method we have proposed for short-range

multiple-input multiple-output (SR-MIMO) transmission is a promising means for reducing power consumption. The method performs MIMO detection with analog devices, thus reducing the number of quantization bits required in the analog-to-digital converter (ADC) of the receiver and the amount of signal processing calculation for MIMO detection. However, when the method is applied to a wall-transmissive wireless repeater on a multilayered wall, the transmission performance degrades due to multipath generated by the multilayered structure. In this letter, we evaluate the method's performance using data for a millimeter wave propagation channel that we measured from wall samples and the measured S-parameters of the method's analog circuit. As a result, we quantify the influence of multipath generated by a wall's multilayered structure on transmission performance.

Permutation-free Clustering Method for Underdetermined Blind Source Separation Based on Source Location Information

N. Ito, S. Araki, K. Kinoshita, and T. Nakatani
IEICE Trans. Fundamentals, Vol. J97-A, No. 4, pp. 234–246, 2014.

We discuss the framework of underdetermined blind source separation (BSS) based on source location information. Conventional techniques in this framework work even for unknown microphone and source positions, if the number of sources is known. However, they cannot deal with an unknown number of sources, and this has significantly limited the real-world application of BSS techniques. We propose a permutation-free clustering method, which extends the above framework in general to the case of an unknown number of sources. Experiments show that the proposed method can separate sources effectively and enumerate sources perfectly, even under an underdetermined (two microphones and three sources), highly reverberant (440-ms reverberation time) condition.

Anticorrelated Bidirectional Output from Quasi-stadium-shaped Microlasers

T. Harayama, K. Arai, K. Yoshimura, S. Shinohara, T. Fukushima, and S. Sunada

Optical Review, Vol. 21, No. 2, pp. 113–116, 2014.

Time-domain properties of the output from quasi-stadium-shaped microcavity semiconductor lasers are studied. Ring modes generating a bidirectional output are selectively excited by partial pumping. We observe a high anticorrelation between the two beams output from the ring modes. This can be considered as the generalization of alternate oscillations reported previously. We find that the outputs exhibit a robust slow modulation of 4–10 MHz, and explain it by the quasi-degeneracy of the resonator modes.