

## 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.

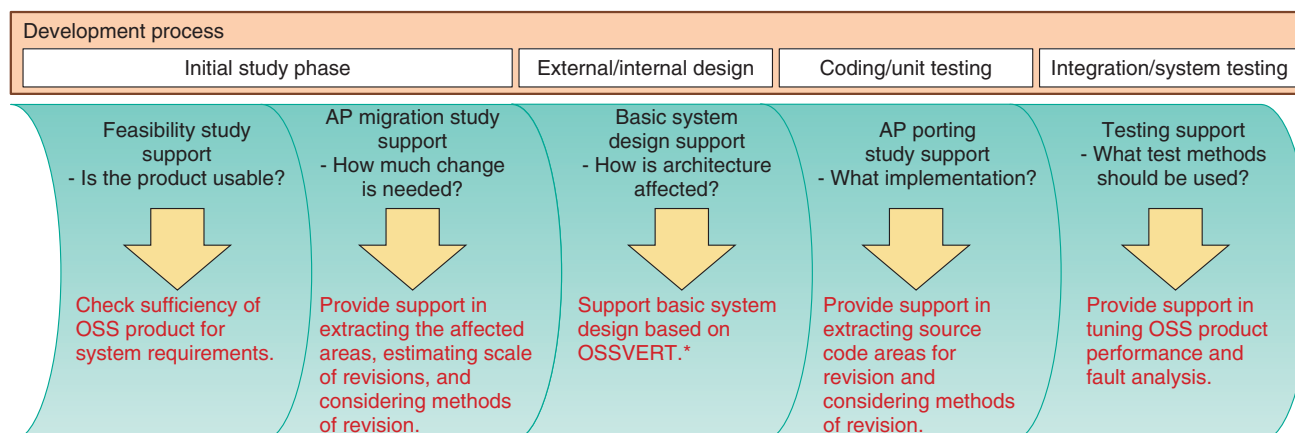
For system development in the NTT Group, we are actively introducing OSS for system upgrades as well as for new systems. Most system development proposals are for updates (over 80% of development proposals the NTT OSS Center is involved in), but most of these cases use particular functions of commercial 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*<sup>\*1</sup>. 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<sup>\*2</sup>, PostgreSQL<sup>\*3</sup>, and Crane<sup>\*4</sup>. 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.

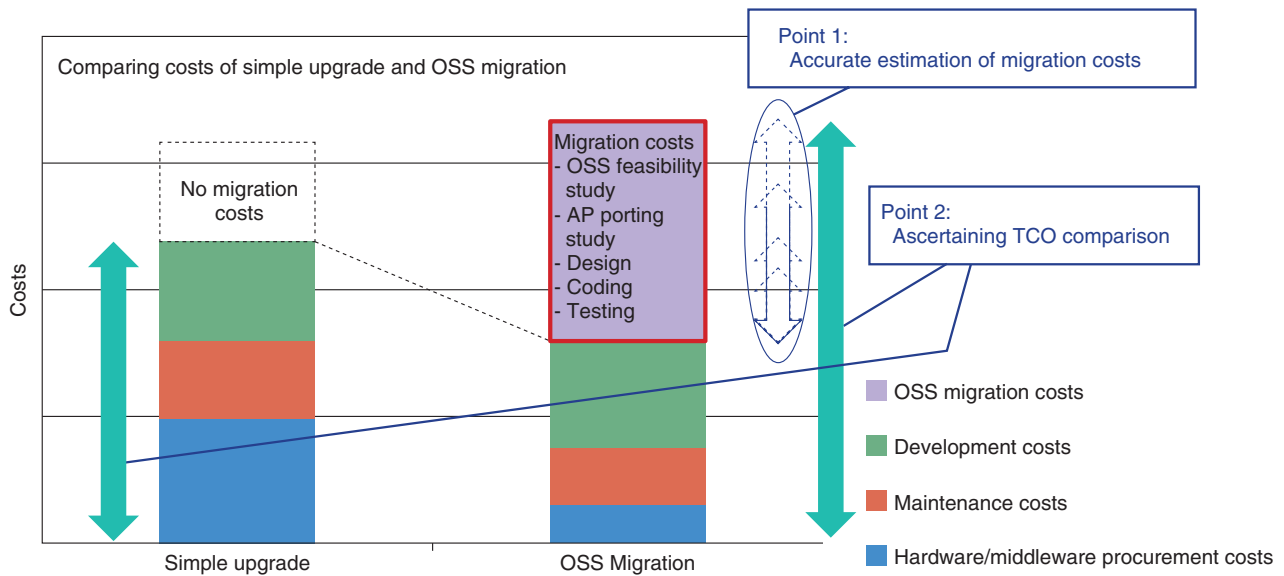


Fig. 2. Determining TCO reduction due to OSS migration.

programming language, differences in OS commands, system call specifications, and memory storage formats for different CPU architectures (big-endian, little-endian, etc.)<sup>\*5</sup> 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<sup>\*6</sup> 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<sup>\*7</sup> 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<sup>\*8</sup> 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

<sup>\*5</sup> 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.

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

<sup>\*7</sup> db\_syntax\_diff: An OSS tool available at the following URL: [https://github.com/db-syntax-diff/db\\_syntax\\_diff](https://github.com/db-syntax-diff/db_syntax_diff)

<sup>\*8</sup> jobnet: Defines how and in what order the jobs should be executed so that multiple jobs (processes) can be run in sequence.

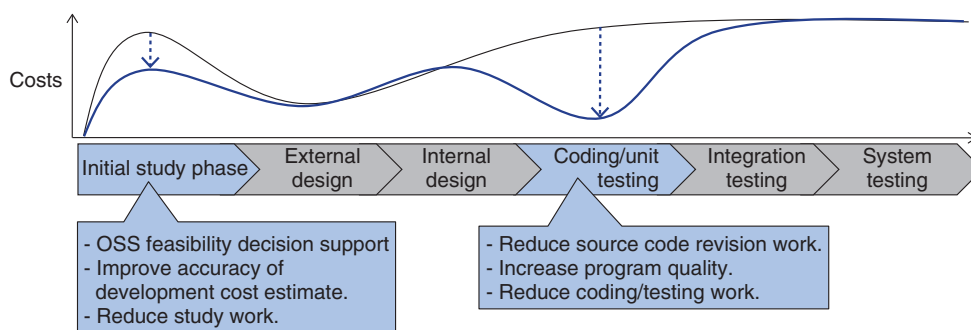


Fig. 3. Benefits of using OSSMA in the development process.

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<sup>\*9</sup> 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

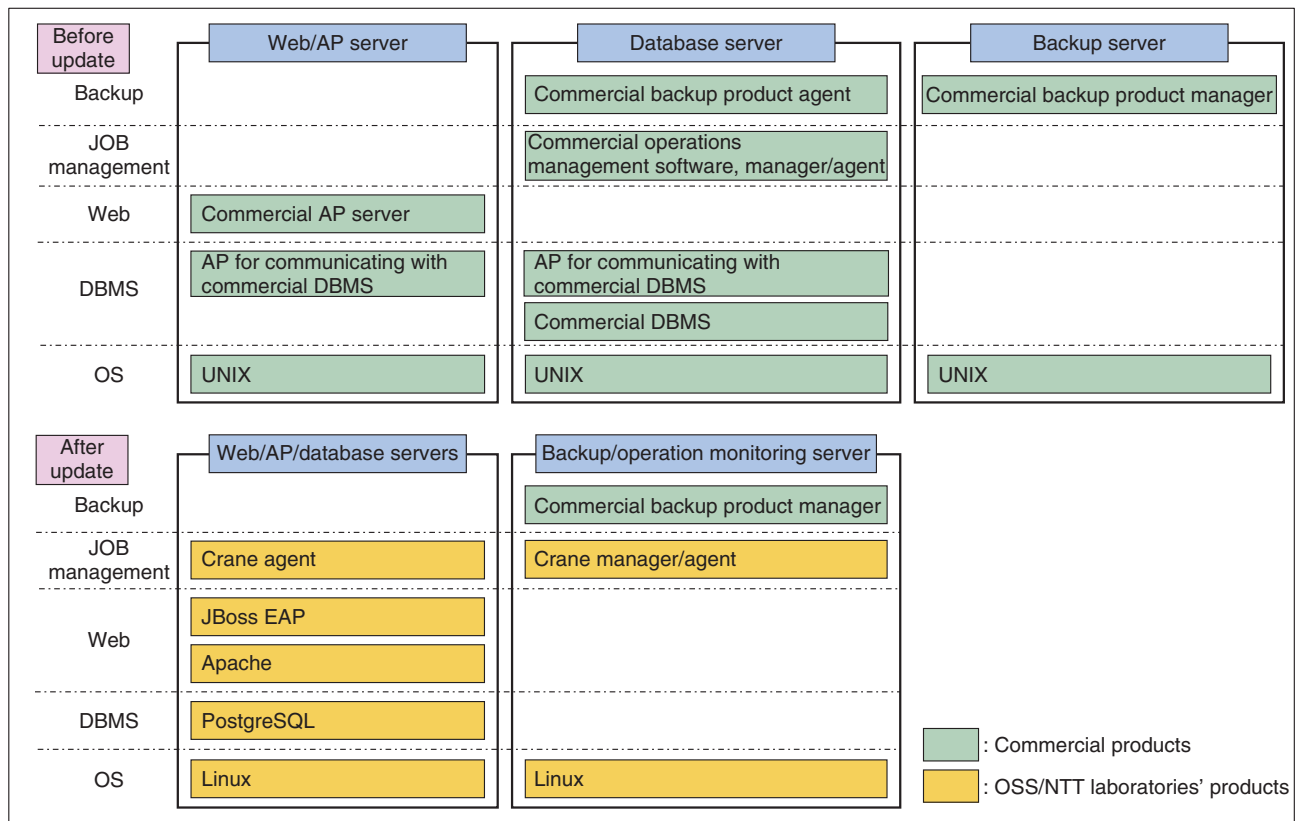
with respect to partitioning. The results of these tests are shown in Fig. 5. The figure shows that System A performance requirements will be met with PostgreSQL, eliminating the concern regarding the development project. It also shows that performance requirements will be met with PostgreSQL without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

<sup>\*9</sup> 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 Hat 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. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201406fa3.html>
- [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.

---