System Management and Operation for Cloud Computing Systems

● Motomitsu Adachi ● Toshihiro Kodaka ● Motoyuki Kawaba
● Yasuhide Matsumoto

With the progress of virtualization technology, cloud systems have been started to be deployed on a full-scale basis. However, there are many issues in terms of managing cloud systems in a stable and high-quality way. This is because the number of servers becomes immense and dependencies between servers become complex. Conventionally, individual business applications and services have been operated in systems. However, in the cloud there is a degree of uniformity in the infrastructure that makes up systems. Consequently, there are hopes that it will be possible to prepare common management platforms and methods such as those to manage application life cycles and predictive failure detection technology. This paper introduces technology that integrates the development and operations management phases in the PaaS region by leveraging such characteristics of clouds. This technology functions according to the characteristics of the applications or individual service level agreements (SLAs), and makes it possible to configure applications that are deployed on the cloud. This paper also introduces technology that allows operators to automatically or simply build a test environment the same as the real environment when changing applications and run automated tests. In addition, this paper touches on technology to monitor and visualize work that is core technologies for the life cycles management. Moreover, this paper describes technology that can conduct statistical processing of the logs that are issued from the system during operation to detect the prediction of failure phenomenon.

1. Introduction

With the progress of virtualization technology, cloud computing is expected to be put to full-scale practical use. Cloud computing lets people use functions of networks, middleware and business applications including servers and storage as much as required when required via the Internet. In this way, there are hopes that cloud computing will give various benefits to users such as economic efficiency, flexibility and promptness. At the same time, however, it has given rise to new challenges in terms of the operation and management of systems. These include the continuity and stability of the services offered, and such issues arise because cloud systems must ensure safety of data management and continuously operate on a 24 hours a day, 7 days a week basis.

This paper presents operations management technologies in the cloud age. First, it describes the functions and roles expected of new operations management. It then describes technologies for realizing life cycle management, which is at the core of operations management, including application deployment and bottleneck analysis technologies. In particular, this paper explains operations-level visualization technology based on infrastructure visualization, which has been realized. This paper also discusses technology to allow failures to be prevented by detecting predictive signs of system failures.

2. Cloud operations management

This section describes the functions to be provided for cloud operations management and technologies to realize them.

2.1 Changes in roles of operations

Companies are reported to incur 70 percent of their information and communications technology (ICT) costs on operation and management, which means that they can allocate only 30 percent of their funds to new development. This is mainly because on-site operational know-how is personalized, hindering automation and skill-less operations. Conventionally, companies have operated their ICT systems by having an infrastructure operation manager construct and operate infrastructure based on the instructions given by the operations administrator. In a cloud environment, the operations administrator in person constructs and operates the infrastructure. This eliminates the need for an infrastructure operation manager specialized in certain operations and his or her roles are transferred to the center operation manager and operations administrator. The center operation manager is in charge of monitoring operations, maintaining large-scale infrastructure, and managing the relationships between infrastructure elements and between operations and infrastructure. The operations administrator manages the status of operations and billing, and monitors the quality of operations. Cloud users have high expectations that they will be able to construct and operate systems in an inexpensive way as compared with the conventional systems. This gives rise to a need for new and different operations management technology. The following sections describe new operations management technologies expected in the cloud age. Specifically, this paper has the premise that uniform infrastructure control has been made possible by visualization technology or such like. It then discusses technology to manage the entire life cycle of a system including its design, construction and operation from the perspective of PaaS, with the focus on its characteristics and points of differentiation.

2.2 Realization of life cycle

One benefit of cloud computing is that it allows the initial construction period of a system to be significantly shortened. This is because part of the environment that has already been built can be used on-demand to skip the processes of infrastructure construction, design and implementation for non-functional requirements. PaaS, which is provided as a cloud service, can automate operations in terms of non-functional requirement such as availability and performance. This can be done through techniques such as automating scale-out and data redundancy management by imposing restrictions on how applications are built.

However, service level agreements (SLAs) currently offered by a PaaS provider are fixed and cannot be freely specified or selected by application developers, who are users of the PaaS service. For this reason, when SLAs are required that exceed what are offered by a PaaS provider through the use of cloud, some measures are necessary. For example, users themselves can monitor the systems while using PaaS and exercise ingenuity for applications. Or users can give up using PaaS and carry out operational design and management on IaaS by themselves. It has been pointed out that, in this case, the processes of operational design and application development are divided. This poses a challenge: it is difficult to run the entire cycle from system development to management in a short time by using agile development techniques.

Yahoo's Flickr team has indicated there is a conflict between the development division, which generally demands addition of functions, and the operations management division, which aims at stable operation. This conflict causes the release cycle to be extended and increases the number of changes made in one release. As a result, there is increased risk at the time of release.[1] DevOps is attracting attention as a new development methodology that eliminates this conflict and integrates development, operations management and quality assurance. DevOps is intended to remove the barrier between development and operations management. It aims to achieve overall optimization and agile and continuous development, thereby reducing the risk mentioned above. A framework[2] for realizing agile operation on clouds has been also proposed.

We are targeting SaaS operators and promoting the development of PaaS management technology that builds a system on a cloud, in accordance with applications and SLAs, and automates operation from infrastructure to applications. Unlike other companies' PaaS such as Google App Engine, this system is characterized by its ability to handle different SLAs for different users while sharing infrastructure between multiple applications to reduce costs. It aims to integrate development and operations management, or to realize DevOps, by extracting from applications the information required for optimum operations management and giving feedback on the bottlenecks and problems generated during operation to developers. This allows the release cycle to be reduced and changes in operations and infrastructure environment to be promptly handled. Specifically, we intend to develop the following technologies by taking advantage of the characteristics of clouds that allow people to use computing resources as much as required when required.

1) Technology for optimum application deployment according to application characteristics and SLAs

This technology allows the applications on clouds to be deployed according to the application characteristics and SLAs. It conducts provisioning (allocates dynamic resources) for virtual machines (VMs), allocates memory and CPU resources and builds virtual systems that combine them, which are required for satisfying the SLAs. In addition, it allows the applications to automatically install the middleware required and then have the applications automatically deployed at the end. Dynamic configuration changes made in response to load variation or virtual system internal faults are automated as well.

2) Verification technology to ensure operation at the time of dynamic configuration change

With this technology, when the configuration of an application is changed, a test environment the same as the real environment is built automatically or by simple operation so that an automated test can be run. The new configuration will not be used unless this test has been passed. This ensures applications run properly even if the configurations of virtual systems, specifications of the individual VMs, and middleware tuning parameters are dynamically changed according to the situations or middleware or OS security patches are applied. Possible methods of conducting automated tests include capturing packets with the configuration before the change so that they can be played back in the test environment, or using an automated test tool in which the developer creates the test scenario.

3) Technology to feed back information from operation bottleneck analysis to development

With this technology, the users collect fundamental information for dynamic configuration changes of the deployed system to determine what to do specifically. First, configuration changes are done within a range that can be automatically handled by operations management such as scale-out and scale-up. However, handling changes with operations management has its limitations and the configuration changes made by operations management as described above may not be enough to solve the problem. In that case, this technology proposes methods of modification including modifying applications to the developer. To satisfy the performance requirements, for example, this technology analyzes the point where execution takes time at the program level and proposes methods of modification such as the addition of a distributed cache server.

In the future, we expect cloud will allow better services to be created one after another in a more efficient way than by installing pieces of middleware individually on IaaS. Regarding relational databases (RDBs), for example, using a service such as database.com[3] is more beneficial than installing MySQL on Amazon EC2. This is because there is no need to make back-ups, and the system has better scalability and improved efficiency thanks to its multitenancy. In this way, the principle of competition always comes into play between pieces of middleware and services and the cycle of creating and selecting new middleware and services is repeated. In this situation, applications must continuously change and always follow the latest service trends. Systems always need to make innovative changes in terms of their operations and functions as well because applications themselves will be eliminated unless users' way of using them is analyzed to identify needs.

Until now, operations management technology has been discussed under the theme of how to improve the efficiency of the processes to ensure stable operation of systems and services, or from the perspective of "defense." We intend to redefine the positioning of operations management technology and drive the development of "positive" operations management technology that realizes life cycle management (LCM) in which applications and operations continue to evolve on evolving infrastructure.

2.3 Shift from infrastructure monitoring to service monitoring

This section describes performance monitoring technologies to maintain application life cycles.

The management of application life cycles on cloud environments is demanding the shift from infrastructure monitoring to service monitoring. That means that performance monitors should provide not only system availability from the viewpoint of system operators but also performance metrics from the viewpoint of end users.

End users that run their applications on clouds have a tendency to pay more attention to the behavior of the applications than that of infrastructure itself, since the details of the infrastructure are invisible to them. Regarding SLA assurance, for example, end users tend to emphasize maintaining end-to-end performance such as business application response, rather than managing infrastructure availability. However, infrastructure monitoring still plays a key role because the availability of the infrastructure is imperative for sustaining stable performance of business applications. Thus, the shift to service monitoring should be conducted based on existing technologies of infrastructure monitoring.

In our view, "information integration" and "monitoring extension" are of importance to the shift of technologies. Information integration means processing various pieces of monitoring information from the infrastructure and organizing them into information friendly to application developers. Monitoring extension allows application developers themselves to extend items to be monitored by using an application framework (**Figure 1**). In either case, the monitoring system will be able to deliver information about service quality to developers more directly than ever.

Information integration necessitates new mechanisms to integrate existing monitoring technologies, because no existing technologies can afford to pinpoint faulty application codes from the observation of a behavioral anomaly detected in the infrastructure. The conventional

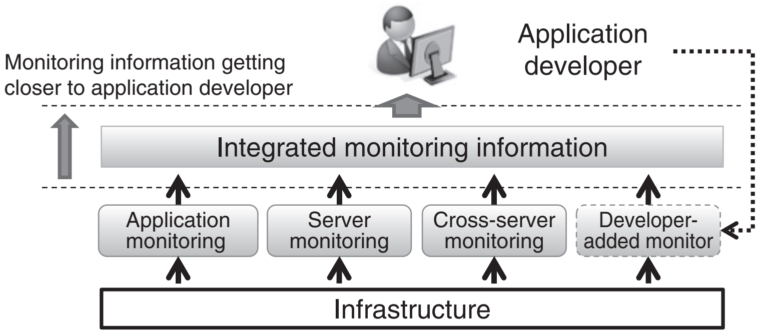

Figure 1
Integrated monitoring information.

Table 1
Existing monitoring technologies.

Application monitoring	Method profile, function profile, application log (Web access log, SQL log)
Server monitoring	Resource meter (CPU, I/O, memory usage)
Cross-server message monitoring	Network traffic analysis, system behavior visualization technology

monitoring technologies are listed in **Table 1**.

Most of the application monitoring and intra-server monitoring shown in Table 1 monitor individual components of server hardware and software separately. Therefore, although they can locate the delay in processing time in case of any performance deterioration, it is still hard to identify the application codes that caused the problem. Meanwhile, cross-server message monitoring cannot delve into application codes, even though it is able to follow problems spreading across servers.

However, integrating these types of monitoring information allows us to create information which is friendly to application developers. An example of information integration that can pinpoint causal application codes is illustrated in **Figure 2**. This monitoring system employs an application monitoring technique, method profile, and a cross-server message monitoring technique, system behavior visualization technology.[4),5)]

System behavior visualization technology, which we have developed, allows users to monitor end-to-end behavior of individual requests from clients by heuristically exploring relationships among network messages. Since it uses only network data between servers in the exploration of the relationships, the technology can chase causes of problems across servers whereas it is platform-independent. By combining this technology and the method profile, the users can trace application codes causing problems even if they are extending over multiple servers. With a simple contrivance provided for the existing method profile mechanism, information integration is feasible.

Monitoring extension entails an application framework that permits application developers to embed their own monitoring codes into application codes. The framework should allow them to obtain useful information that they cannot extract in ordinary monitoring systems and to focus on the most interesting codes by monitoring partial codes and altering items to be monitored dynamically depending on the situation. Incorporating this framework into the architecture of integrated monitoring information in Figure 1 will make it easier for application developers to grasp the behavior of their applications and pinpoint problematic application codes.

We intend to work on the architecture of integrated monitoring information described here to accomplish service monitoring that realizes life cycles of development and operation.

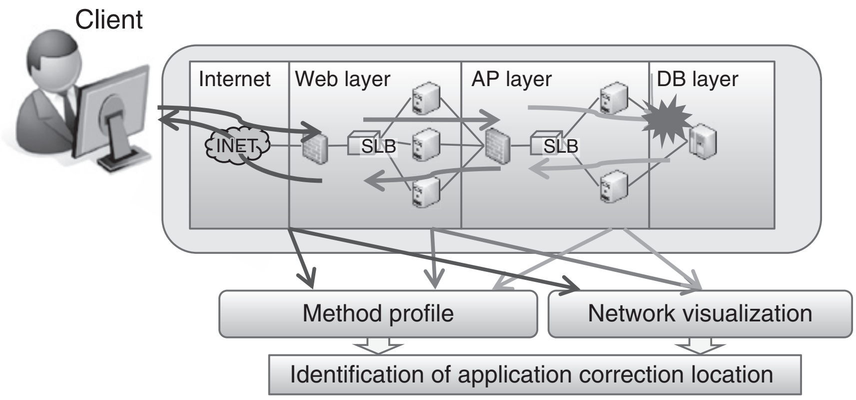

Figure 2
Locating causal application codes.

3. Stable system

This section presents new failure management technology for stably operating large-scale systems.

3.1 Approach to failures

When a large number of failures are generated in cloud computing, an area that is growing larger in scale, burdens on the operation manager in charge of failure response become enormous. One approach for reducing the failure response burdens is advance detection. This aims to detect occurrence of failure in advance so that workarounds can be taken before the failure occurs and so that initial action can be taken more quickly in response to failures. It thereby prevents failures and restricts the spread of their impact.

3.2 Failure prediction and response: predictive detection

To secure sufficient time for executing workaround operations, we need to know beforehand what kind of failure will occur and when the failure will occur. Here, we refer to the technology for identifying the type and time of occurrence of failure as predictive failure detection technology. The period between the time of advance detection and time of failure occurrence should be as long as possible because advance detection would be worthless if a failure is detected too late to be avoided.

We target advance detection at a level of a few tens of minutes to a few hours rather than a span of three months or one year. This is because we assume systems will offer predictive detection that allows users to appropriately use automation functions for operations management such as live migration and diagnostic functions.

Various studies have been conducted up to now on predictive failure detection technology. Fu et al. focus on performance metrics measured in an ICT system such as CPU and memory usage.[6] They classify the performance metrics measured for individual failures by temporal locality and spatial dependence. Then they extract the characteristics of the respective failure types from the results of classification and then predict the failure by using the extracted models. This technique uses the failure occurrence interval to characterize failure types in terms of the temporal distributions. For that reason, it is capable of predicting failures that characteristically occur regularly, but it is not suitable for those generated irregularly. Salfner et al. have assumed that predictive signs of failures can be characterized by patterns of specific error events. They model the state transition of error events from their detection to the occurrence of a failure including the time required for transition.[7),8)] With these techniques, a failure that has a lead time of Δt_l after the present time t can be predicted. However, the prediction time is in seconds, which is not long enough to allow users to make a response after predictive signs of failures have been detected.

We have developed a failure detection technology capable of promptly detecting occurrence of the failure. It identifies message patterns that characterize various failures from the past failure records and system log messages and detects occurrence of failures by comparing current log messages of target system with identified message patterns.[9)] We have assumed that, with failures that present predictive signs, message patterns (predictive patterns) characteristic of the failures appear before the failure occurs as well. Accordingly, our technology can identify predictive patterns by learning in the time before the failure occurs. To learn in this way, the time between the detection of predictive patterns and the time of occurrence (lead time) should be associated with predictive patterns. This lead time can be used to calculate the predicted time of occurrence (**Figure 3**). By using this technology, if a predictive pattern is detected in system log messages being observed, the time of failure occurrence can be estimated based on the lead time associated with the predictive pattern.

To evaluate this technique, we used past log messages and past failure records of the in-house system under evaluation. We examined individual cases of two failure types that occurred particularly often [type 1: threshold exceeded (30 cases) and type 2: process down (166 cases)]. We divided the 60 minute-period immediately before the time of occurrence into time slots of 10 minutes and our technology learned the predictive patterns by using the log messages mentioned above as the input (**Figure 4**). As a result, in terms of the precision (the ratio of the correct detection), the value was considerably high at 0.7 or higher in most of the time slots for both failure types. However, the value did not increase as the time of failure occurrence approached. In terms of the recall (the cover ratio of the correctly predicted failures including the expected time to all the occurrence failures), the percentage for 50 to 60 minutes before process down-type failures was relatively low at around 50% but the value was high at

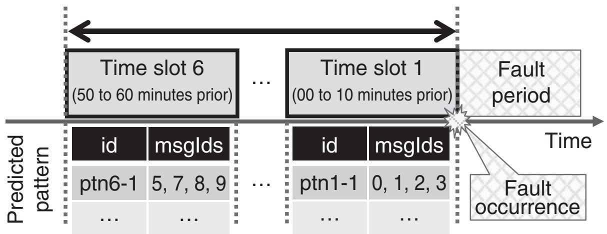

Figure 4
Method of learning predictive patterns in this evaluation.

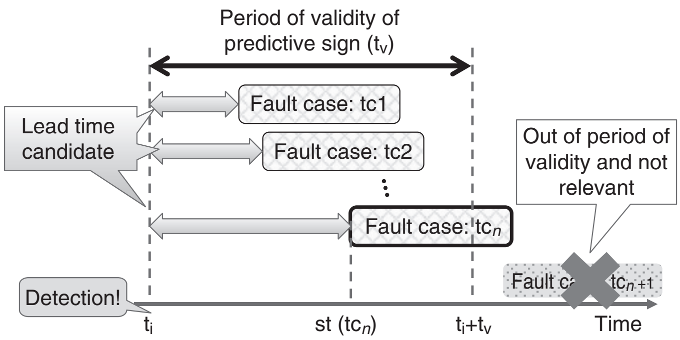

Figure 3
Calculation of lead time candidates for each identification.

0.7 or higher for other time slots. In addition, the cover rate showed a tendency to increase as the time of failure occurrence approached. We have confirmed the effectiveness of the lead time estimation with these results.

In the future, we plan to work on improving the processing performance so that our technology can be applied to large-scale systems and the design of management processes including human operation for actual operation.

4. Conclusion

This paper has verified the optimum way to deploy applications that is best suited to the application's characteristics and SLAs, and their operations during dynamic configuration changes, as operations management technologies for a cloud environment. It has also introduced technology to manage the life cycles of applications, and integrated information monitoring technology that is used to monitor applications that support such life cycle management technology. This paper also discussed predictive failure detection technology for stable operations management. We intend to mature the technologies as the middleware platform of Fujitsu's cloud systems.

References

1) J. Allspaw et al.: 10 deploys per day: Dev&Ops cooperation at Flickr. Velocity 2009.
2) N. Harada et al.: Operation Framework for Cloud Application Systems Based on IaaS with Continuous Integrations. (in Japanese), IPSJ SIG Technical Report, 2011.
3) Salesforce database.com. *http://database.com*
4) R. Take: Real-time System Behavior Visualization Technology for Managing Complex Distributed Application Systems. (in Japanese), *FUJITSU*, Vol. 59, No. 1, pp. 33–38 (2008).
5) M. Morinaga et al.: Network Sensing—Network Monitoring and Diagnosis Technologies. (in Japanese), *FUJITSU*, Vol. 60, No. 4, pp. 381–386 (2009).
6) S. Fu et al.: Quantifying Temporal and Spatial Correlation of Failure Events for Proactive Management. SRDS2007, pp. 175–184.
7) F. Salfner et al.: Predicting Failures of Computer Systems: A Case Study for a Telecommunication System. IPDPS2006.
8) F. Salfner et al.: Using Hidden Semi-Markov Models for Effective Online Failure Prediction. SRDS2007, pp. 161–174.
9) Y. Watanabe et al.: Trouble Detection with Message Pattern Learning. IOTS2009.

Motomitsu Adachi
Fujitsu Laboratories Ltd.
Mr. Adachi is currently engaged in research and development of system management technology.

Motoyuki Kawaba
Fujitsu Laboratories Ltd.
Mr. Kawaba is currently engaged in research and development technology in relation to monitoring and visualization of items ranging from infrastructure to applications.

Toshihiro Kodaka
Fujitsu Laboratories Ltd.
Mr. Kodaka is currently engaged in research and development of technology for managing development and operation life cycles of applications.

Yasuhide Matsumoto
Fujitsu Laboratories Ltd.
Mr. Matsumoto is currently engaged in research and development of operation and maintenance in relation to standardization of cloud-related technology.