Provisioning Infrastructure Supporting Cloud Operations

Kenichi Sato[†], *Hideki Hayashi, and Ken Ojiri*

Abstract

With cloud technology, computing resources are provided dynamically in response to requests from users, so the function that manages normal operation of these resources (the management system) is very important. In this article, we describe a provisioning infrastructure, which is one of the main components of the management system within Common IT Bases over Cloud Computing (CBoC), a cloud infrastructure system being developed by the NTT Information Sharing Platform Laboratories (IT: information technology).

1. Introduction

NTT Information Sharing Platform Laboratories is conducting research and development (R&D) on a cloud for large-scale distributed processing that can act as societal infrastructure. This cloud is composed of an execution system and a management system (**Fig. 1**). The execution system is the set of functions providing specific computing resources to users, including virtual machines and storage, and the management system is the set of functions that manage the operation of the execution system so that services (computing resources) can be provided appropriately to the users. In this article, we describe the cloud management system.

This cloud can be regarded from several perspectives, including those of the cloud operator, the cloud services provider, and the cloud services users. In this article, we treat the cloud management system, so we focus on the operator's perspective and the services provider's perspective.

2. Cloud operation model

The cloud operations management task cycle, as conceived by NTT, is shown in **Fig. 2**.

In response to user requests, provisioning functions control the execution system to provide computing resources. Monitoring functions check the health of the computing resources being provided (whether continuous service is being provided) and the service level (whether the service quality is appropriate). The capacity planning function calculates the appropriate amount of resources on the basis of monitoring results and creates configuration change instructions for the provisioning functions. Initial provisioning is done on the basis of estimates calculated by the applications provider, so as the services being provided on the cloud mature, computing resources could become either excessive or insufficient compared with requirements owing to inaccuracies in the estimates. The purpose of the capacity planning function is to make these adjustments appropriately, but at present there is no well-defined method of accomplishing this. In current operation, the user re-calculates the estimate on the basis of the monitoring results.

Note that this cycle of provisioning, monitoring, and capacity planning is not a new concept with cloud computing. It is basically the same as has been done for various services in the past. In fact, such adjustments were done every few months or years in the past, but with cloud technology, they can now be done every few minutes or tens of minutes. The automation of this operational cycle has made it possible to operate with fine adjustments being made to equipment according to demand, which was difficult to do in the past.

Below, we describe mainly the provisioning function, which is one of the functions in the cloud operations management model.

[†] NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan

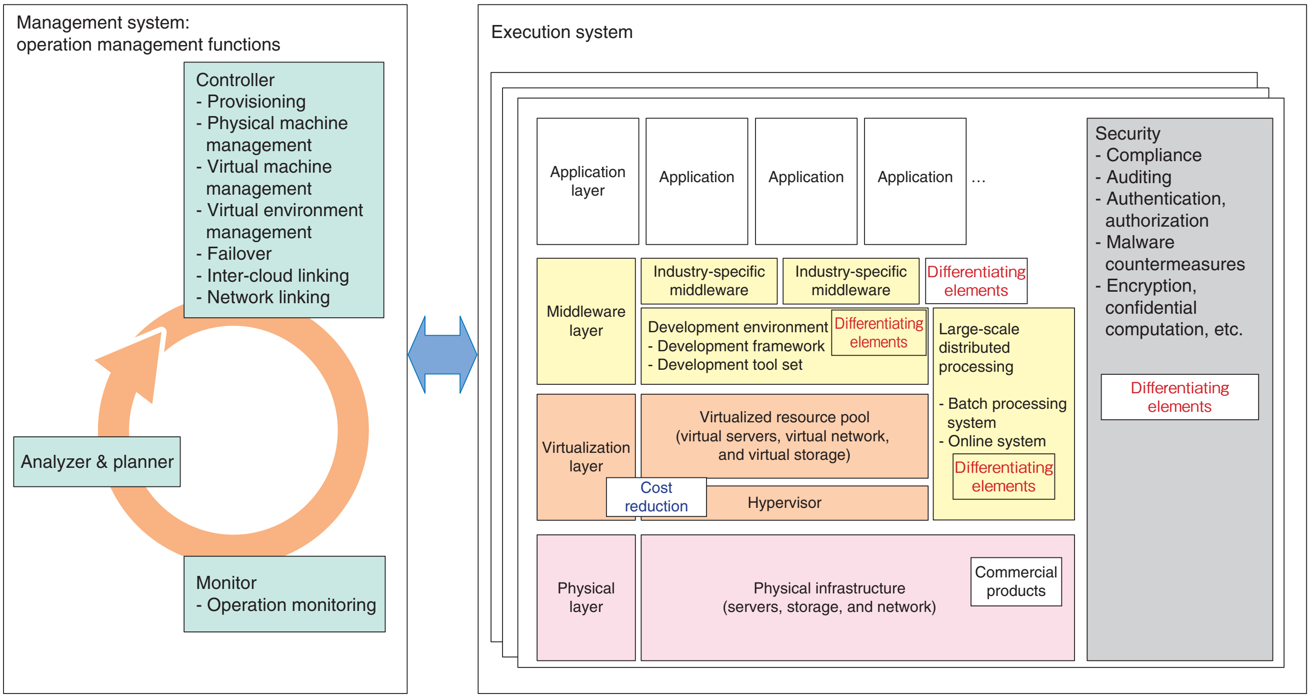


Fig. 1. Cloud configuration.

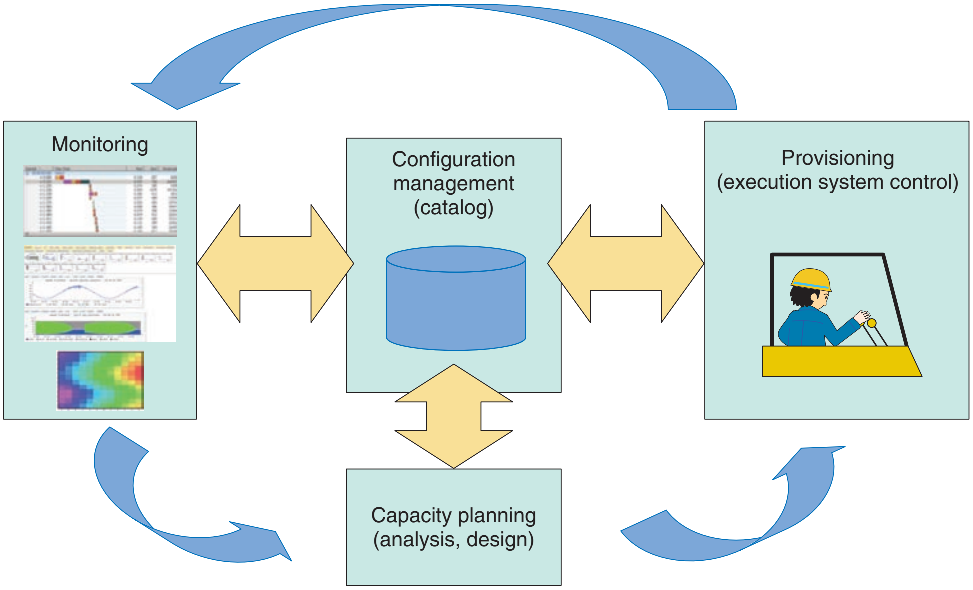

Monitoring
Configuration management (catalog)
Provisioning (execution system control)
Capacity planning (analysis, design)

Fig. 2. Cloud operations-management task cycle.

3. CBoC provisioning infrastructure

Next, we describe the Common IT Bases over Cloud Computing (CBoC) provisioning infrastructure being developed by NTT (IT: information technology).

3.1 Development goals

There are already several cloud services in existence, including that of the NTT Group. On-demand self-service [1] is a widely accepted property of clouds, performing resource provisioning according to instructions received from users, so the management system, and in particular the provisioning function, is a basic feature of a cloud service.

These existing cloud services provide various types of services at various layers, according to business needs. This trend is expected to accelerate in the future [2].

The provisioning functions in existing services can be considered to optimize specific computing resources according to their properties. On the other hand, the speed with which the provisioning function can be developed when providing a new computing resource must not become an obstacle to the business.

Thus, with the CBoC provisioning infrastructure, we have abstracted the control model so that various computing resources can be supported very quickly, helping to accelerate business development. The operation of various computing resources can also be optimized daily (automatically in the future) through the operational cycle discussed above.

3.2 Central concepts

Below, we describe the central concepts implemented by the features of the CBoC provisioning infrastructure, which allow various computing resources to be added quickly and applications to be integrated easily.

3.2.1 Resource abstraction and connection between resources

One question that arises is whether particular computing resources will behave differently, though we talk about all resource types together. However, from a provisioning perspective, the important things are the fixed operations: create, allocate, initialize, activate, deactivate, finish, free, and delete. If we consider a specific example, these fixed operations apply to both virtual machines (kernel-based virtual machine (KVM) and VMware) and virtual local area networks (VLANs), allowing them to be handled in a unified way by the provisioning infrastructure. The CBoC provisioning infrastructure uses this concept to abstract all of the objects that it handles as resources (of course, in addition to the fixed operations, it also provides ways to handle attributes and operations particular to a given resource).

Another question that arises is whether the process of connecting a virtual machine to a LAN can be abstracted. However, connections between resources are also abstracted, including definitions for a connection's source and destination as well as connection operations, and these definitions regulate what connections and what connection operations are possible. In this way, virtual machines and VLANs can connect, and higher-layer connections, such as those between an application and a database, can also be automated.

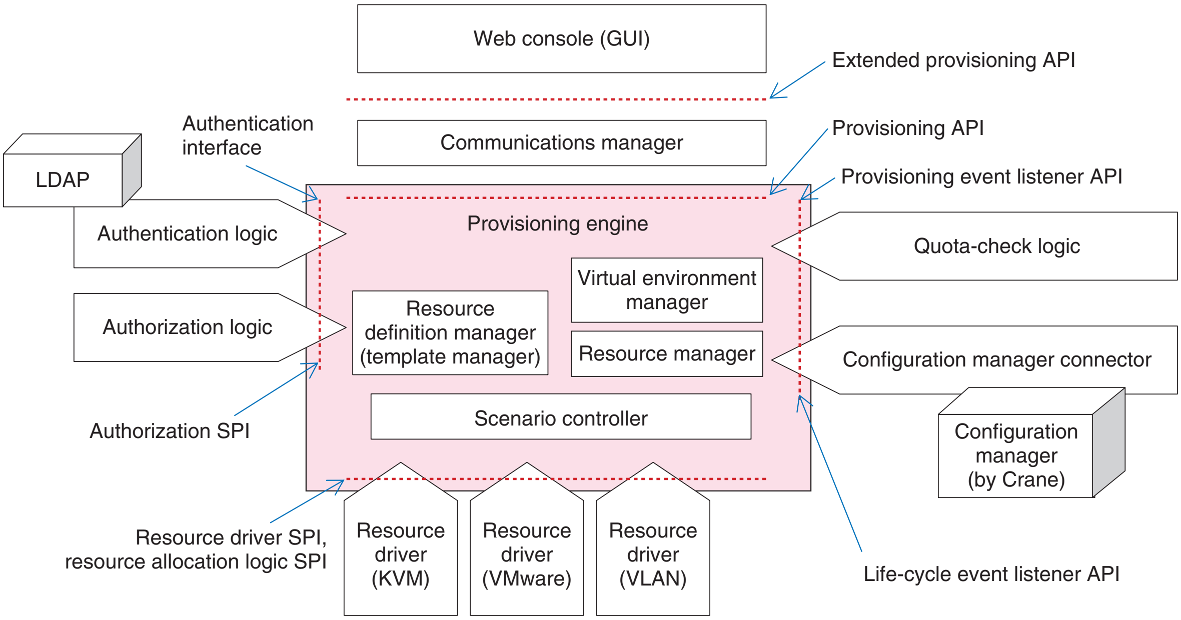
This abstraction enables new resources to be added easily to the CBoC provisioning infrastructure. By creating a driver that operates the execution system and by writing template data (resource definitions, connection definitions, and connection operation definitions), one can integrate a new resource into the system.

The development required when new resources are added is also a concern. It is true that creating a resource driver involves development. However, a resource re-definition procedure can be used, even for small-scale cases. Resource re-definition is a function for creating new template data from the existing resource status and its template data, without writing programs or creating template data from scratch. This function makes it easy to perform operations such as creating a new template of a virtual machine with installed applications from a virtual machine with only the installed operating system.

3.2.2 Virtual environment operation, asynchronous scenario processing

In many cases with cloud services like Amazon's, the focus is on handling individual virtual machines. On the other hand, in ordinary system development, it is instead more common to bundle multiple processors and treat them as a single system or development environment. The CBoC provisioning infrastructure uses a virtual environment approach, modeling the system being provided by the user. When a system is being developed or a service is being provided, multiple resources can be conveniently bundled in a virtual environment, enabling batch operations (start, end, backup, etc.).

Resource operations are basically considered to be asynchronous. Operations on real resources such as



API: application programming interface
GUI: graphical user interface
LDAP: Light Directory Access Protocol
SPI: service provider interface

Fig. 3. CBoC provisioning infrastructure components.

launching a virtual machine usually take time, so processing is done asynchronously with respect to user requests.

Asynchronous processing is relatively easy to implement for individual virtual machine operations, but for batch operations in a virtual environment with multiple virtual machines, it is more complex. As an example, what would need to be done in a state where one virtual machine was stopped, another virtual machine was assigned but was not processing yet, and you want to transition them to a completed state all at once? With the CBoC provisioning infrastructure, operating procedures are generated automatically, taking into account the current state of resources bundled in the virtual environment and the state of connections between them.

3.2.3 CBoC provisioning infrastructure configuration

Keeping in mind the need to accelerate business development, the various components of the CBoC provisioning infrastructure were designed to be pluggable, with well-defined interfaces, so that only the required parts for a given application need be developed or modified. These components are shown in **Fig. 3**. The types of interfaces and their applications are also listed in the **Table 1**.

4. Application examples and effect

Next, we describe some examples of applying the CBoC provisioning infrastructure and the effect of these applications.

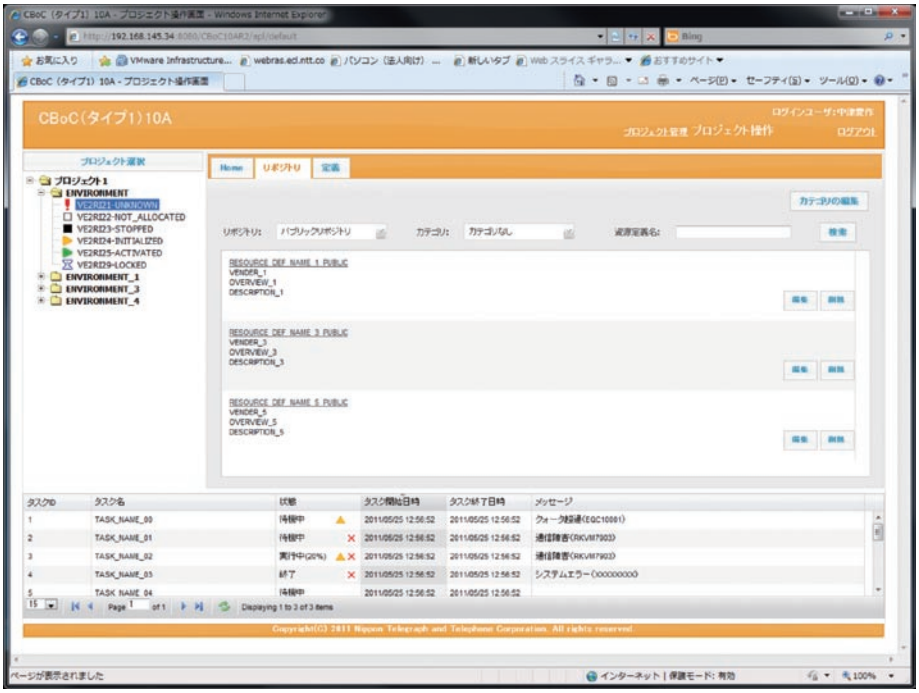
4.1 Application to the R&D cloud

Starting in October 2011, we plan to apply the CBoC provisioning infrastructure to the R&D cloud as a development environment lending service components. The configuration for the R&D cloud includes authentication linked with the Open Light Directory Access Protocol (OpenLDAP), project management, role-based authorization, and group quota checking. It uses a web-based console screen (**Fig. 4**) and allows users to build and operate their own development and testing environments using virtual machines and VLANs.

Prior to the R&D cloud, NTT was operating a development environment lending service manually. Experience with it showed that the actual work time required was approximately 5.5 hours, from applying to use the system to actually beginning to use it, and that other work was also required, such as 8.5 hours to create a virtual machine template. The CBoC provisioning infrastructure should reduce this work significantly. Since these figures are based on actual

Table 1. CBoC provisioning-infrastructure external interfaces.

Interface	Application/objective
Resource SPI	A new resource driver should conform to this interface when being created and integrated.
Resource allocation logic SPI	Modules conforming to this interface can control resource allocation (allocating virtual machines to physical machines etc.).
Authentication interface	When the authentication method used for accessing the provisioning infrastructure is changed, the authentication module must satisfy this interface.
Authorization SPI	When a change is made to the method by which operators authorize operations, an authorization module implementing this interface must be created and integrated. The authorization module checks who, in what role, is authorized to perform what actions.
Provisioning API	Basic API provided by the CBoC provisioning infrastructure. Allows creation and deletion of resources, querying and changing of resource states, and other actions.
Extended provisioning API	Provides the provisioning API as a web service. Bundles primitive API requests according to application requirements.
Provisioning event listener API	To enable the introduction of processing before or after the execution of a provisioning API routine, an event listener can be implemented using this interface.
Life-cycle event listener API	To enable the introduction of any additional processing when the state of a resource changes, an event listener can be implemented using this interface.



CBoC(タイプ1)10A

プロジェクト選択

プロジェクト1
- ENVIRONMENT
  - VE2RI21-UNKNOWN
  - VE2RI22-NOT_ALLOCATED
  - VE2RI23-STOPPED
  - VE2RI24-INITIALIZED
  - VE2RI25-ACTIVATED
  - VE2RI29-LOCKED
- ENVIRONMENT_1
- ENVIRONMENT_3
- ENVIRONMENT_4

リポジトリ: パブリックリポジトリ　カテゴリ: カテゴリなし　資源定義名:

RESOURCE_DEF_NAME_1_PUBLIC
VENDER_1
OVERVIEW_1
DESCRIPTION_1

RESOURCE_DEF_NAME_3_PUBLIC
VENDER_3
OVERVIEW_3
DESCRIPTION_3

RESOURCE_DEF_NAME_5_PUBLIC
VENDER_5
OVERVIEW_5
DESCRIPTION_5

タスクID	タスク名	状態		タスク開始日時	タスク終了日時	メッセージ
1	TASK_NAME_00	待機中	▲	2011/05/25 12:56:52	2011/05/25 12:56:52	ウォーク超過(EQC10001)
2	TASK_NAME_01	待機中	✕	2011/05/25 12:56:52	2011/05/25 12:56:52	通信障害(RKVM7903)
3	TASK_NAME_02	実行中(20%)	▲✕	2011/05/25 12:56:52	2011/05/25 12:56:52	通信障害(RKVM7903)
4	TASK_NAME_03	終了	✕	2011/05/25 12:56:52	2011/05/25 12:56:52	システムエラー(xxxxxxxx)
5	TASK_NAME_04	待機中		2011/05/25 12:56:52	2011/05/25 12:56:52	

Displaying 1 to 3 of 3 items

Copyright(C) 2011 Nippon Telegraph and Telephone Corporation. All rights reserved.

ページが表示されました

Fig. 4.   Web console for the R&D cloud.

working time, the waiting time for users will be reduced even more, which will increase user satisfaction.

4.2 Evaluating ease with which resource drivers for products on the market can be created

As part of the process of evaluating virtual network products in NTT, we have created a resource driver that handles the product controller and have linked it with a KVM driver for evaluation. The evaluator was not particularly familiar with virtual network products, but was still able to create the resource driver in about one person-month, including testing and integrating the virtual network product into the self-provisioning environment.

Considering that this resource driver was produced internally by laboratory staff, without the quality control of a commercial product, that there is currently no development guide, and that there are many other variable factors, this evaluation result shows that it is easy to create resource drivers. It also shows that it will be easy to quickly integrate various new computing resources provided in the future into the system.

5. Future directions

(1) Linking with capacity planning

There is no definitive method for capacity planning yet, but in the future this area will expand beyond simply the so-called performance optimization to include aspects like controlling power consumption in response to the demands of society. Linking this with the provisioning function is also important, and we intend to pursue this R&D in a unified way.

(2) Enhancing network functions

Network system functions must be enhanced in order to provide the type of cloud infrastructure expected from a network carrier. This requires us to advance network driver development, centered on the laboratory’s core technologies, and expand into the area of performance optimization, including that of the network.

(3) Inter-cloud linking

Cloud linking from the perspective of distributing resources is also important considering the effects of disasters. When clouds are linked, the management of large amounts of widely distributed data and cooperative behavior over a wide-area network are very important. Such wide-ranging cooperation is an important area, even in the laboratory. In the future, we will continue to work on these elemental technologies.

(4) Commodity function initiatives

Virtual machines and VLANs, which were the initial cloud resources, are now familiar, and this area will continue to become commoditized in the future. In commodity fields, influential open source software such as OpenStack has appeared, and this may contribute to a unification of operational interfaces in the future. For the CBoC provisioning infrastructure, resource drivers have been developed according to standardized interfaces, so we leave integration as commodity resources to organizations like OpenStack and focus our R&D efforts on differentiating functions.

References

[1] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” NIST, Vol. 53, No. 6, p. 50, 2009.

[2] A. Abe, T. Ochi, A. Shirahase, and F. Kumada, “NTT Group Initiatives for Achieving Societal Cloud Infrastructure,” NTT Technical Review, Vol. 9, No. 12, 2011.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201112fa1.html

Kenichi Sato

Senior Research Engineer, Supervisor, Cloud Computing SE Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in precision mechanical engineering from the University of Tokyo in 1991 and 1993, respectively. He joined NTT Network Information Systems Laboratories in 1993. He studied intelligent agent communication including agent communication platforms and agent application development platforms. After that, he joined an electronic payment system trial project. From 2002 to 2005, he developed and operated an ASP (application service provider) service (Business goo), targeted at small-office home-office businesses. In 2005, he returned to the Information Sharing Platform Laboratories and engaged in the development of a high reliability transaction processing monitor on Linux. Since 2008, he has been studying cloud computing systems, including cloud distributed data management, cloud application frameworks, and cloud operation platforms.

Hideki Hayashi

Senior Research Engineer, Supervisor, Development Project Leader, Cloud Computing SE Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from Tokyo Institute of Technology in 1987 and 1989, respectively. He joined NTT Telecommunication Networks Laboratory in 1989 and studied network control technology. He developed ATM network integration manager systems, security gateway systems, and authentication authorization accounting systems. He is currently studying CBoC Type 1. He is a member of the Institute of Electronics, Information and Communication Engineers.

Ken Ojiri

Research Engineer, Cloud Computing SE Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in communication engineering from Osaka University in 1994 and 1996, respectively. He joined NTT Network Service Systems Laboratories in 1996 and developed intelligent network systems until 2002. From 2002 to 2010, he developed, provided, and operated identity management systems for web-based Internet services in NTT Information Sharing Platform Laboratories and NTT Resonant Inc. He is currently studying and developing CBoC provisioning infrastructure.