Toward Future Network Control Nodes

Takeshi Fukumoto[†]*, Masami Iio, and Kiyoshi Ueda*

Abstract

As a step toward reducing the cost of carrier networks, centralized adjustment and maintenance of various server systems is being studied. In this article, we describe a shared platform for network control nodes that NTT Network Service Systems Laboratories is studying as part of that cost reduction effort.

1. Introduction

There have been great changes in the environment of carrier networks over recent years, and network control nodes have also diversified. Because the degree of diversification varies with the required levels of performance, cost, quality, etc., control nodes are developed on their respective platforms. One result, however, is that maintenance procedures vary from system to system and there are restrictions on the range within which development expertise can be acquired. That creates circumstances that hinder plans for overall optimization. To break out of these constraints, we are investigating a shared platform for NTT network control nodes.

2. Shared platform and its architecture

In systems for which high reliability is required, one approach is to use a single server that has a high failure tolerance and to have a backup server to enable service provision to continue if a failure does occur (active-standby (ACT-SBY) architecture). In the case of session control servers for the Next Generation Network (NGN), it is expensive to use hardware that conforms to the Advanced-TCA (Telecom Computing Architecture) standard to ensure reliability and expensive to use the ACT-SBY architecture to ensure reliability. On the other hand, there are also systems for which cost reduction is desired, even if some degree of reliability must be sacrificed. Therefore, we chose to use the N-ACT (n active servers) architecture (**Fig. 1**), which provides functions for flexibly changing the degree of redundancy in backup systems. The N-ACT architecture features a high facility utilization rate because servers provide backup for other servers at the same time as performing their own processing. By contrast, the ACT-SBY architecture has dedicated backup servers, which are idle when on standby.

Furthermore, because the system scale (processing capability) is determined by the functions and services, there is great variation in size from small to large. Accordingly, a future network platform must be applicable to both large systems and small systems. With the consideration that "large is the sum of many smalls", there is also the approach of unifying on large systems, but if a large-scale platform is applied to small systems, the cost and energy consumption required for adjustment become problematic. Accordingly, our platform offers a way to provide performance, reliability, and scale according to demand by grouping small, low-cost servers. If this platform is shared by multiple systems, resources can be allocated among the systems. Specifically, as shown in **Fig. 2**, servers that do not have any application software installed are managed as a resource pool. The loads of the systems are monitored and high-load systems are given a capacity boost by installing application software on servers in the resource pool. For low-load systems, some servers are returned to the resource pool. This server adjustment is done automatically, without any interruption of services.

[†] NTT Network Service Systems Laboratories
Musashino-shi, 180-8585 Japan

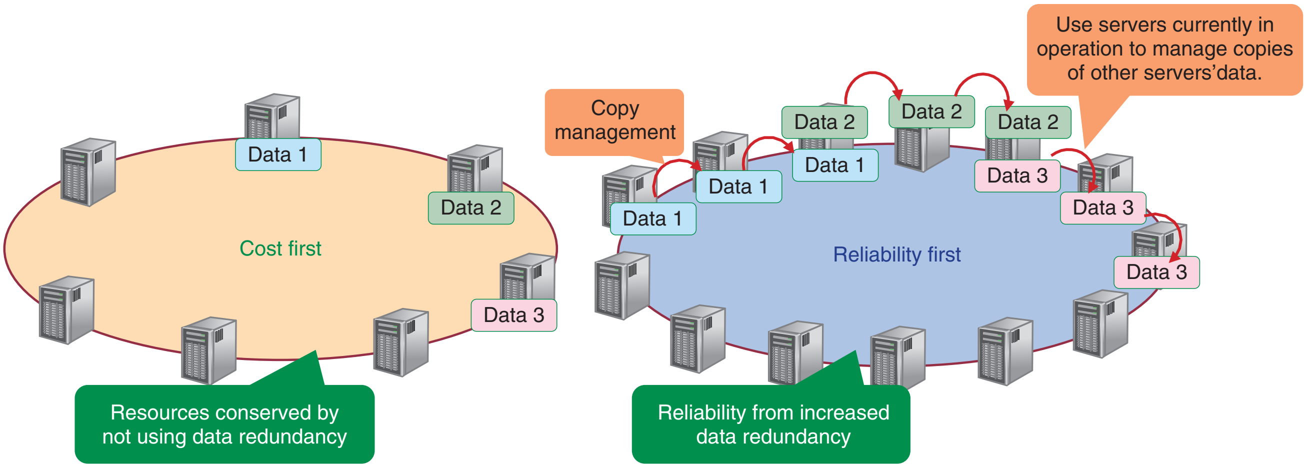
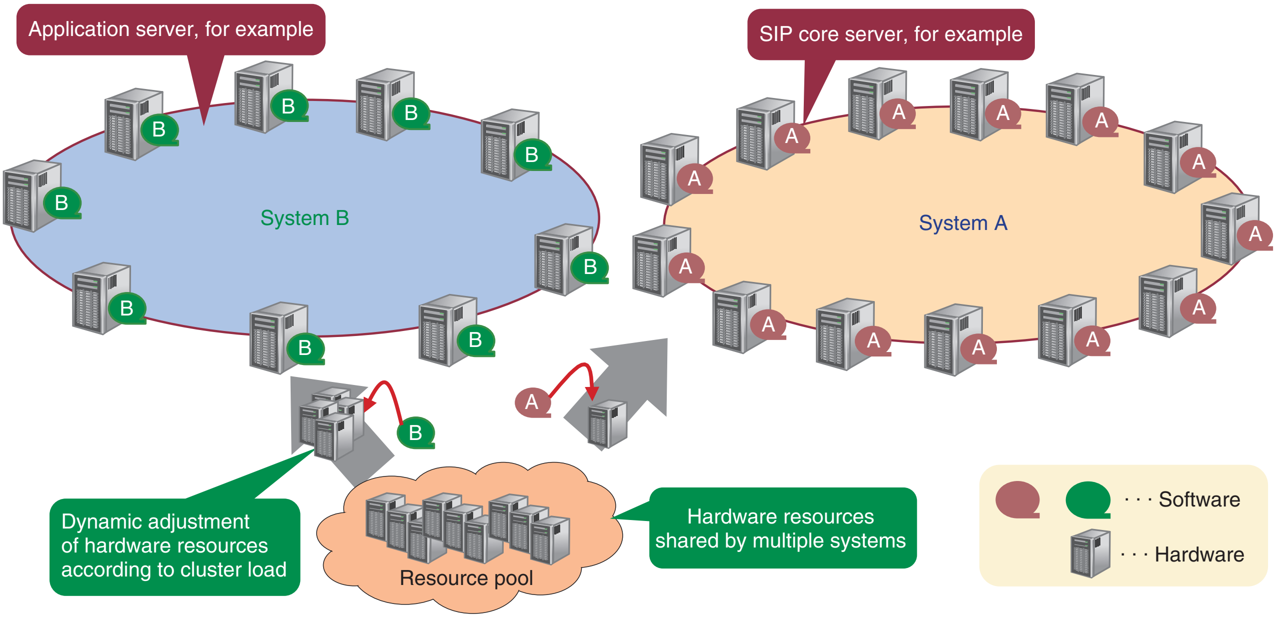

Use servers currently in operation to manage copies of other servers'data.
Copy management
Data 1
Data 2
Data 3
Cost first
Reliability first
Resources conserved by not using data redundancy
Reliability from increased data redundancy

Fig. 1. Maintaining reliability in an N-ACT architecture.


Application server, for example
SIP core server, for example
System B
System A
B
A
Dynamic adjustment of hardware resources according to cluster load
Resource pool
Hardware resources shared by multiple systems
· · · Software
· · · Hardware

SIP: session initiation protocol

Fig. 2. Dynamic resource control.

3. Reduction of maintenance cost

As mentioned above, the need to reduce costs is increasing, and for future networks, there is a demand to reduce maintenance to the maximum extent possible. Our objective is to reduce maintenance costs by unifying the maintenance system applicable to a platform shared by multiple systems.

There is concern that many of the systems currently in operation may be incapable of providing services in a normal way if the load exceeds their processing capability, so the trend is toward autonomous invocation of system regulation. The system is given some degree of surplus capability at the capacity design stage so that services can be provided in the unregulated range. However, if the possibility of the load surpassing that assumed during the design arises, a switching of accommodation, which is referred to as location division, is done. Location division involves momentary halting of services during the night when there are fewer users so that various types of data can be moved. High-risk tasks such as location division involve the temporary stoppage of service, and the effects of this task failing are great. Therefore, the work is done only after verification by multiple prior tests conducted in a testing environment to minimize the risk of failure from bugs or operational errors. However, such a cautious procedure increases costs, so to address that problem we have implemented a mechanism for flexibly changing capacity to provide automatic server recovery. That mechanism can dynamically increase or decrease server capacity by adding servers automatically when the processing load becomes large or by removing servers when the processing load becomes small.

4. Tolerance of severe disasters and congestion

Last year's major earthquake in eastern Japan caused serious damage to NTT buildings. If a similar disaster were to damage entire buildings, it would become impossible to provide services because most of the current systems have the online and backup servers located in the same building. Although there are some systems with remote backup servers to avert that problem, the switchover is done manually, so restoration after a severe disaster takes from a few hours to several days.

Our objectives are to minimize the effects of regional damage such as that as caused by the 2011 Great East Japan Earthquake by distributing servers over a wide area and to enable rapid restoration of services in undamaged areas. Moreover, because the effects of that earthquake extended over a wide range, phone calls made by people to check on the safety of their family and friends were concentrated: as a result, servers entered a regulated state that extended over a long period of time. For that problem, too, we would like to use the dynamic server addition mechanism described above to prevent prolonged service regulation due to load concentration.

5. Modeling approach

Next, we describe the technologies used. A distributed system is a group of multiple servers regarded as a single virtual server. This group of servers is referred to as a cluster. The individual servers in a cluster cooperate to perform processing, but the volume of signals that must be exchanged among the servers generally increases with the number of servers. That prevents the abovementioned scalability. We chose to use a method [1] in which the signals for cooperation between servers are regularly distributed in advance, and signals received from outside are divided up among the servers that hold the data. The processing that determines the location of the data required at a particular time uses a consistent hash algorithm, which is explained below.

6. Consistent hash algorithm

The function that distributes received signals to the servers must satisfy the following conditions.

(1) The request signals shall be distributed evenly over multiple servers.

(2) Multiple related request signals shall be assigned to the same server.

To satisfy these two conditions, we use a hash function, which is a computational method for generating a pseudorandom number of fixed length from given variable-length data (key information). By using a hash function to compute the parameters that relate the requests contained in the received signals in order to determine the processing server, we can satisfy the two conditions.

However, the platform that we aim for requires dynamic adjustment of the number of servers to which the request signals are distributed. When a simple hash function is used, changing the number of destination servers also changes all of the request distribution conditions. Because the number of servers is expected to change frequently, the cost of data reorganization processing when conditions are changed cannot be neglected. One technique for minimizing that cost is to use a consistent hash algorithm (**Fig. 3**).

Using a consistent hash algorithm [2] enables the effects of an increase or decrease in the number of destination servers to be limited to the adjacent servers. This method is also effective for determining the backup servers in an N-ACT architecture such as shown in Fig. 3. With this method, the mechanism for implementing the processing for signal distribution to the backup system servers when a failure occurs can be simpler than that in the conventional ACT-SBY configuration, which does it by changing the IP (Internet protocol) addresses.

7. Separation of logic and data

Having explained a method for maintaining

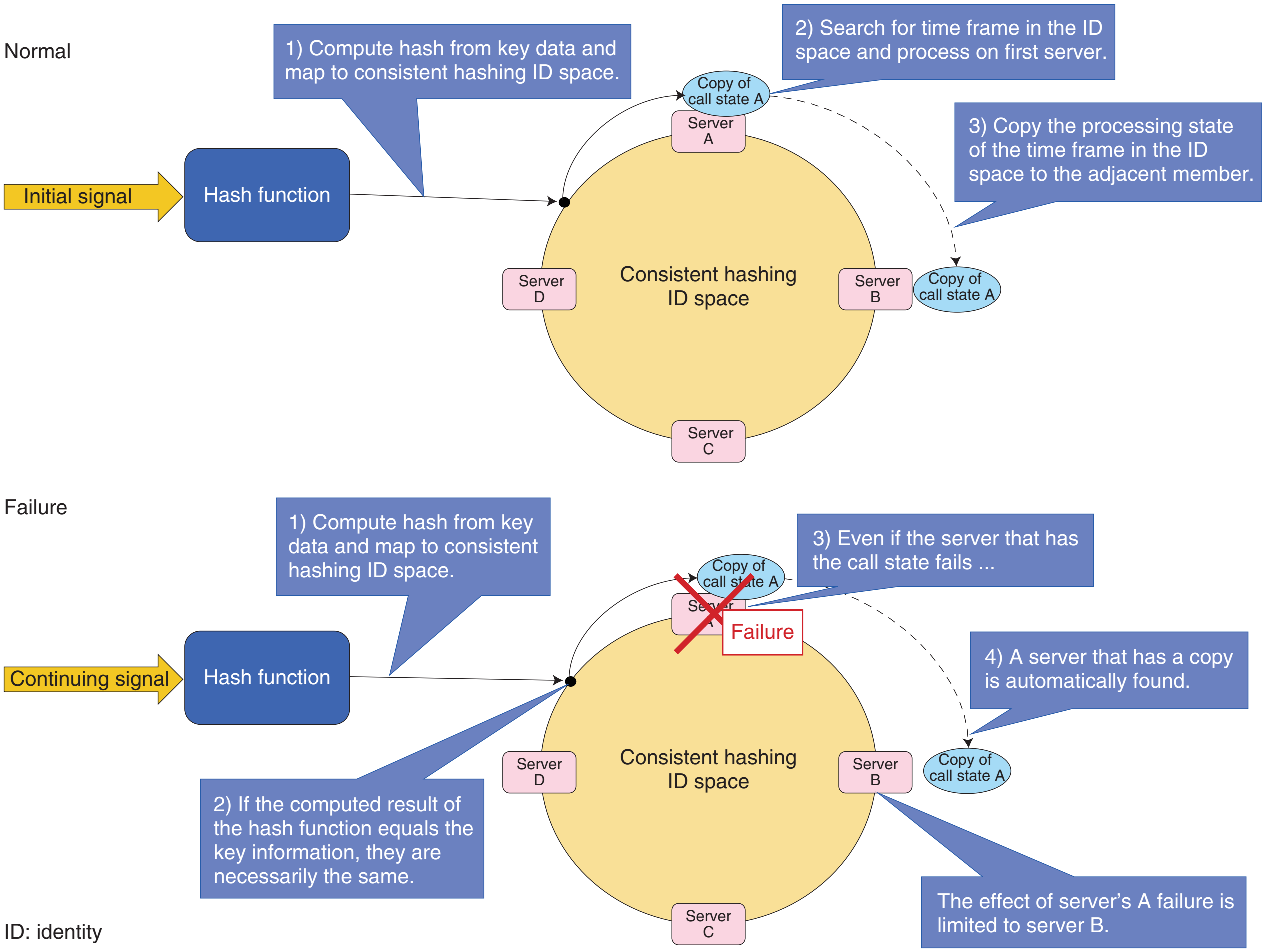

Normal
1) Compute hash from key data and map to consistent hashing ID space.
2) Search for time frame in the ID space and process on first server.
3) Copy the processing state of the time frame in the ID space to the adjacent member.
Initial signal
Hash function
Copy of call state A
Server A
Server D
Consistent hashing ID space
Server B
Copy of call state A
Server C
Failure
1) Compute hash from key data and map to consistent hashing ID space.
3) Even if the server that has the call state fails ...
Copy of call state A
Server A
Failure
4) A server that has a copy is automatically found.
Continuing signal
Hash function
Server D
Consistent hashing ID space
Server B
Copy of call state A
2) If the computed result of the hash function equals the key information, they are necessarily the same.
Server C
The effect of server's A failure is limited to server B.
ID: identity

Fig. 3. Using a consistent hash algorithm to determine the server.

organization and reliability of the dynamic data generated by the request processing, we now explain the handling of static data stored on the servers in advance, such as service conditions and user contract information.

Our platform is not of the distributed capacity type: the requests sent by a particular user do not necessarily all arrive at the same server. Therefore, it is assumed that static data can be stored on all servers. If the amount of data is small, it is not a problem for the same data to be stored on all of the servers; however, for a large amount of data, the data distribution is an important issue. Therefore, we have proposed a split configuration that involves a cluster for performing services (logic cluster) and a cluster for the management of static data (data cluster) (**Fig. 4**). The logic cluster retrieves the data required for processing from the data cluster each time it is needed, which allows the use of a distributed load architecture. However, with this method, there is concern about the delay until data is retrieved. To address that concern, the data cluster uses the hypertext transfer protocol (HTTP), which has a relatively light processing overhead, and a key-value data store, which permits high-speed retrieval. Although their roles differ, any of the clusters described so far can be configured on the basis of the new network control node configuration technology. Concerning the processing delay in data clusters, we are currently constructing prototypes for the purpose of investigating the issue.

8. Concluding remarks

Many technical problems remain regarding commercial introduction of the new network control node configuration technology. Work to characterize the

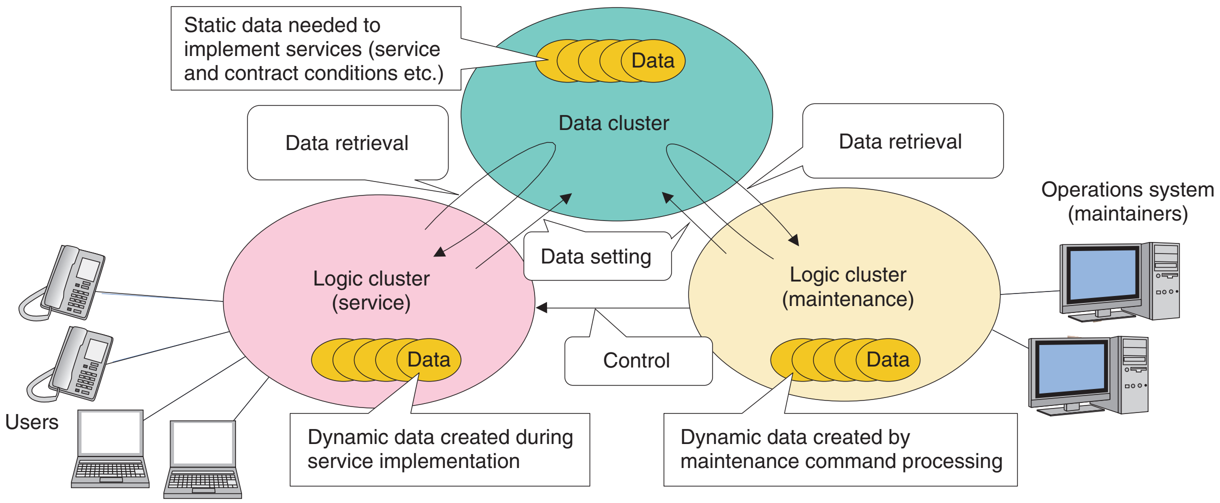

Static data needed to implement services (service and contract conditions etc.)
Data
Data cluster
Data retrieval
Data retrieval
Operations system (maintainers)
Data setting
Logic cluster (service)
Logic cluster (maintenance)
Data
Data
Control
Users
Dynamic data created during service implementation
Dynamic data created by maintenance command processing

Fig. 4. Logic and data clusters.

delay in static data retrieval is in progress, but revision according to circumstances and additional study are required. It is also necessary to investigate methods for implementing file updating, failure detection, analytical methods, and other functions available in current systems. Furthermore, advanced control technology based on automatic control theory for dynamic control of servers has also become necessary.

We will continue to investigate the feasibility of various application-dependent requirements, including the extent to which implementation is possible, and continue with technical studies of application to commercial networks.

References

[1] M. Irie, H. Nishimura, M. Kaneko, and H. Bessyo, "Session Control Server Cluster Model for Scale-out and Flexible Reconfiguration," IEICE General Conference, Feb. 2011 (in Japanese).

[2] M. Irie, H. Nishimura, M. Kaneko, H. Bessyo, and M. Iio, "Load Distribution that is Aware of Data Copying in the Consistent Hash Algorithm," IEICE Technical Report, Vol. 110, No. 224, pp. 69–74, IN2010-77, 2010 (in Japanese).

Takeshi Fukumoto

Senior Research Engineer, Emerging Communication Architecture Project, NTT Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in computer science and systems engineering from Kyushu Institute of Technology, Fukuoka, in 1994 and 1996, respectively. Since joining NTT Network Service Systems Laboratories in 1996, he has been engaged in R&D of communication node system software, especially in designing the distributed communication node management software. After having been engaged in the development of application software for the switchboard of NTT's public backbone, he became a development member of NTT's Next Generation Network (NTT-NGN). In particular, he was in charge of the application software of a Subscriber Session Control server (SSC) and Interrogate Session Control server (ISC) in NGN development. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).

Masami Iio

Senior Research Engineer, Supervisor, Emerging Communication Architecture Project, NTT Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from Nihon University, Tokyo, in 1988 and 1990, respectively. He joined NTT Switching Systems Laboratory in 1990 and studied the Advanced Intelligent Network, especially dynamic feature interaction between services. In 1995, he was transferred to the staff section in the head office; he investigated network open strategies for telecommunication competitions in Europe and North America and voice communication over packet network technologies, e.g., VoIP and VTOA, in order to study the effects on NTT's business. After that, he developed the commercial system of the Advanced Intelligent Network for IMT-2000 and high-availability middleware for the network system on general-purpose or COTS servers. He is currently studying the system architecture for the future network. He is a member of IEICE and the Technical Program Committee in ICIN.

Kiyoshi Ueda

Senior Research Engineer, Supervisor, Emerging Communication Architecture Project, NTT Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from Keio University, Kanagawa, and the Ph.D. degree in information sciences from Kyushu University, Fukuoka, in 1987, 1989, and 2010, respectively. Since joining NTT Switching Systems Laboratories in 1989, he has been engaged in R&D of communication node system software, especially in designing distributed communication node management software. He was an active member of ITU-T SG11 and SG4, where he contributed to the TMN. He was responsible for R&D of peer-to-peer network service software and grid computing software. He was also responsible for the development of an NGN application server and some network operation systems. He received the Best Paper Award from the IEICE Communication Society in 2008. He is a member of IEEE, IEICE, and the Information Processing Society of Japan.