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

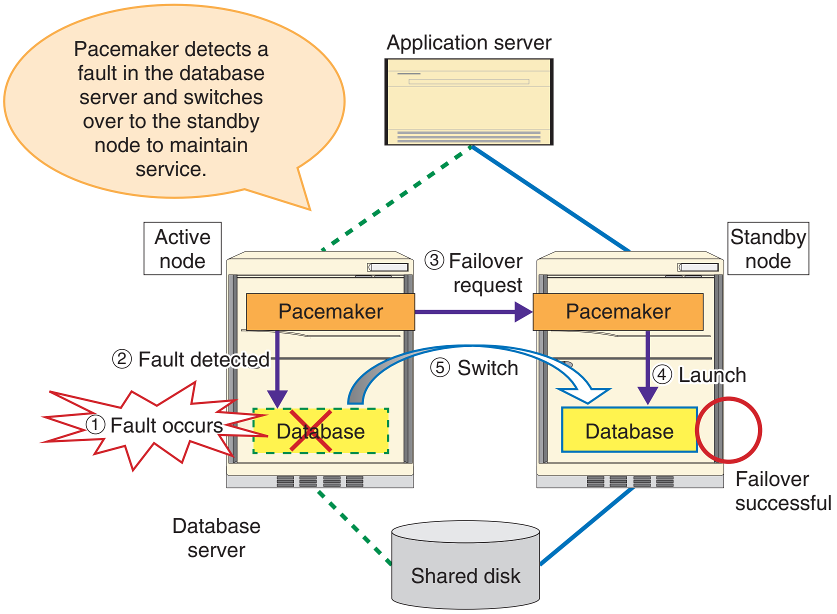

Fig. 1. Overview of Pacemaker operation.

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.

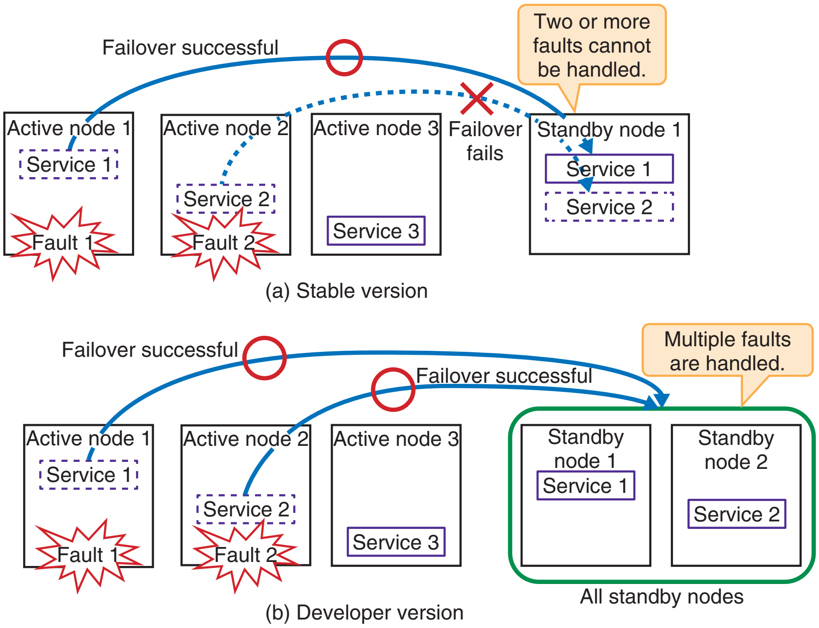

Fig. 2. Pacemaker failover.

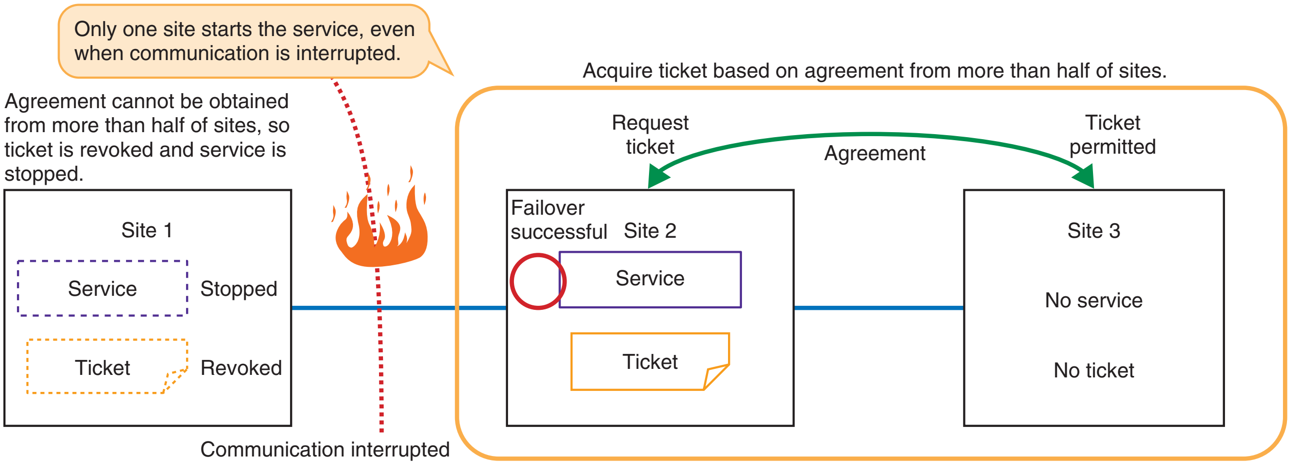
Only one site starts the service, even when communication is interrupted.
Agreement cannot be obtained from more than half of sites, so ticket is revoked and service is stopped.
Site 1
Service
Stopped
Ticket
Revoked
Communication interrupted
Acquire ticket based on agreement from more than half of sites.
Request ticket
Agreement
Ticket permitted
Failover successful
Site 2
Service
Ticket
Site 3
No service
No ticket

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.

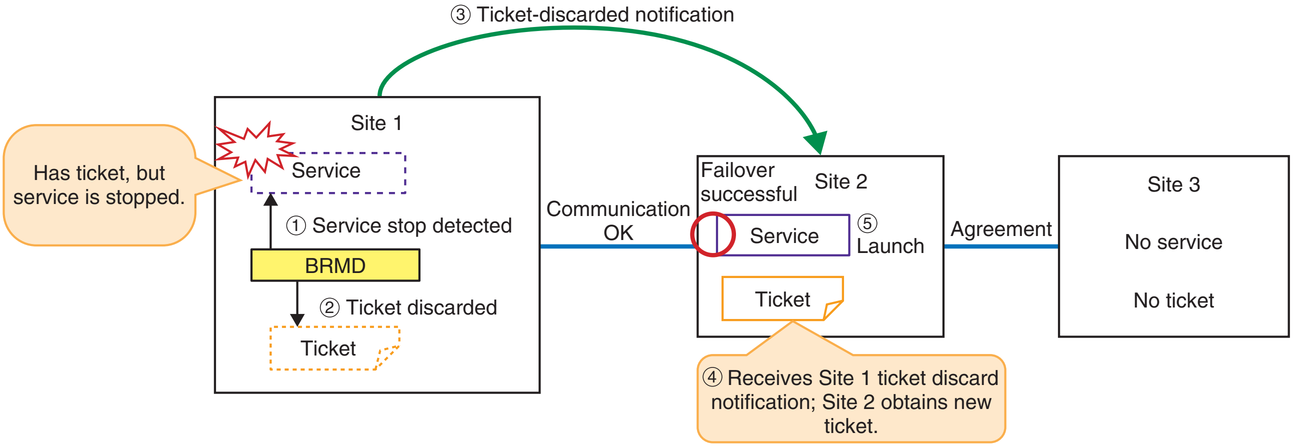

Fig. 4. Overview of operation of Booth_Resource_Monitord.

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_Monitord, 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 develop, 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.

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.

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.

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.

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.

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.