ExpressCluster® X 3.0 *for Solaris*

Getting Started Guide

10/01/2010
First Edition

Revision History

Edition	Revised Date	Description
First	10/01/2010	New manual

Disclaimer

Trademark Information

Table of Contents

Preface

Who Should Use This Guide

ExpressCluster Getting Started Guide is intended for first-time users of the ExpressCluster. The guide covers topics such as product overview of the ExpressCluster, how the cluster system is installed, and the summary of other available guides. In addition, latest system requirements and restrictions are described.

How This Guide is Organized

Section I **Introducing ExpressCluster**

Chapter 1 **What is a cluster system?**
Helps you to understand the overview of the cluster system and ExpressCluster.

Chapter 2 **Using ExpressCluster**
Provides instructions on how to use a cluster system and other related-information.

Section II **Installing ExpressCluster**

Chapter 3 **Installation requirements for ExpressCluster**
Provides the latest information that needs to be verified before starting to use ExpressCluster.

Chapter 4 **Latest version information**
Provides information on latest version of the ExpressCluster.

Chapter 5 **Notes and Restrictions**
Provides information on known problems and restrictions.

Chapter 6 **Upgrading ExpressCluster**
Provides instructions on how to update the ExpressCluster.

Appendix

Appendix A **Glossary**

Appendix B **Index**

ExpressCluster X Documentation Set

The ExpressCluster X manuals consist of the following four guides. The title and purpose of each guide are described below:

Getting Started Guide

This guide is intended for all users. The guide covers topics such as product overview, system requirements, and known problems.

Installation and Configuration Guide

This guide is intended for system engineers and administrators who want to build, operate, and maintain a cluster system. Instructions for designing, installing, and configuring a cluster system with ExpressCluster are covered in this guide.

Reference Guide

This guide is intended for system administrators. The guide covers topics such as how to operate ExpressCluster, function of each module, maintenance-related information, and troubleshooting. The guide is supplement to the *Installation and Configuration Guide*.

ExpressCluster X Integrated WebManager Administrator's Guide

This guide is intended for system administrators who manage the cluster system using ExpressCluster with ExpressCluster Integrated WebManager and for system engineers who are introducing Integrated WebManager. The details on the required items at the time of introducing the cluster system is explained in accordance with actual procedures.

Conventions

In this guide, **Note**, **Important**, **Related Information** are used as follows:

Note:
Used when the information given is important, but not related to the data loss and damage to the system and machine.

Important:
Used when the information given is necessary to avoid the data loss and damage to the system and machine.

Related Information:
Used to describe the location of the information given at the reference destination.

The following conventions are used in this guide.

Convention	Usage	Example
Bold	Indicates graphical objects, such as fields, list boxes, menu selections, buttons, labels, icons, etc.	In **User Name**, type your name. On the **File** menu, click **Open Database**.
Angled bracket within the command line	Indicates that the value specified inside of the angled bracket can be omitted.	`clpstat –s[-h` *`host_name`*`]`
#	Prompt to indicate that a Solaris user has logged in as root user.	`# clpcl -s -a`
Monospace (courier)	Indicates path names, commands, system output (message, prompt, etc), directory, file names, functions and parameters.	`/Solaris/3.0/eng/server/`
Monospace bold (courier)	Indicates the value that a user actually enters from a command line.	Enter the following: **`# clpcl -s -a`**
Monospace italic (courier)	Indicates that users should replace italicized part with values that they are actually working with.	`pkgadd -NECexpresscls-<`*`version_number`*`>-<`*`release_number`*`>-x86.pkg`

Contacting NEC

For the latest product information, visit our website below:

http://www.nec.co.jp/pfsoft/clusterpro/clp/overseas.html

Section I Introducing ExpressCluster

This section helps you to understand the overview of ExpressCluster and its system requirements. This section covers:

- Chapter 1 What is a cluster system?
- Chapter 2 Using ExpressCluster

Chapter 1 What is a cluster system?

This chapter describes overview of the cluster system.
This chapter covers the following items:

Overview of the cluster system

A key to success in today's computerized world is to provide services without them stopping. A single machine down due to a failure or overload can stop entire services you provide with customers. This will not only result in enormous damage but also in loss of credibility you once enjoyed.

A cluster system is a solution to tackle such a disaster. Introducing a cluster system allows you to minimize the period during which operation of your system stops (down time) or to avoid system-down by load distribution.

As the word "cluster" represents, a cluster system is a system aiming to increase reliability and performance by clustering a group (or groups) of multiple computers. There are various types of cluster systems, which can be classified into the following three listed below. ExpressCluster is categorized as a high availability cluster.

- **High Availability (HA) Cluster**

  In this cluster configuration, one server operates as an active server. When the active server fails, a stand-by server takes over the operation. This cluster configuration aims for high-availability and allows data to be taken over as well. The high availability cluster is available in the shared disk type, data mirror type or remote cluster type.

- **Load Distribution Cluster**

  This is a cluster configuration where requests from clients are allocated to load-distribution hosts according to appropriate load distribution rules. This cluster configuration aims for high scalability. Generally, data cannot be taken over. The load distribution cluster is available in a load balance type or parallel database type.

- **High Performance Computing (HPC) Cluster**

  This is a cluster configuration where CPUs of all nodes are used to perform a single operation. This cluster configuration aims for high performance but does not provide general versatility.
  Grid computing, which is one of the types of high performance computing that clusters a wider range of nodes and computing clusters, is a hot topic these days.

High Availability (HA) cluster

To enhance the availability of a system, it is generally considered that having redundancy for components of the system and eliminating a single point of failure is important. "Single point of failure" is a weakness of having a single computer component (hardware component) in the system. If the component fails, it will cause interruption of services. The high availability (HA) cluster is a cluster system that minimizes the time during which the system is stopped and increases operational availability by establishing redundancy with multiple servers.

The HA cluster is called for in mission-critical systems where downtime is fatal. The HA cluster can be divided into two types: shared disk type and data mirror type. The explanation for each type is provided below.

Shared disk type

Data must be taken over from one server to another in cluster systems. A cluster topology where data is stored in a shared disk with two or more servers using the data is called shared disk type.

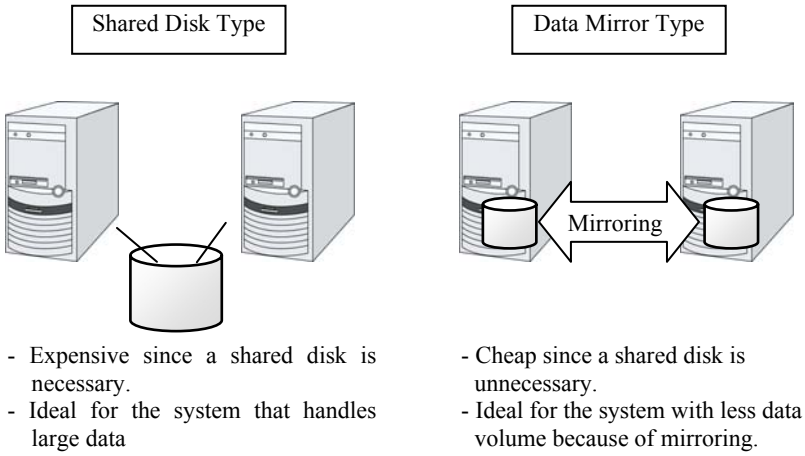


Figure 1-1: HA cluster configuration

If a failure occurs on a server where applications are running (active server), the cluster system detects the failure and applications are automatically started in a stand-by server to take over operations. This mechanism is called failover. Operations to be taken over in the cluster system consist of resources including disk, IP address and application.

In a non-clustered system, a client needs to access a different IP address if an application is restarted on a server other than the server where the application was originally running. In contrast, many cluster systems allocate a virtual IP address on an operational basis. A server where the operation is running, be it an active or a stand-by server, remains transparent to a client. The operation is continued as if it has been running on the same server.

File system consistency must be checked to take over data. A check command (for example, fsck in Solaris) is generally run to check file system consistency. However, the larger the file system is, the more time spent for checking. While checking is in process, operations are stopped. For this problem, journaling file system is introduced to reduce the time required for failover.

Logic of the data to be taken over must be checked for applications. For example, roll-back or roll-forward is necessary for databases. With these actions, a client can continue operation only by re-executing the SQL statement that has not been committed yet.

A server with the failure can return to the cluster system as a stand-by server if it is physically separated from the system, fixed, and then succeeds to connect the system. Such returning is acceptable in production environments where continuity of operations is important.

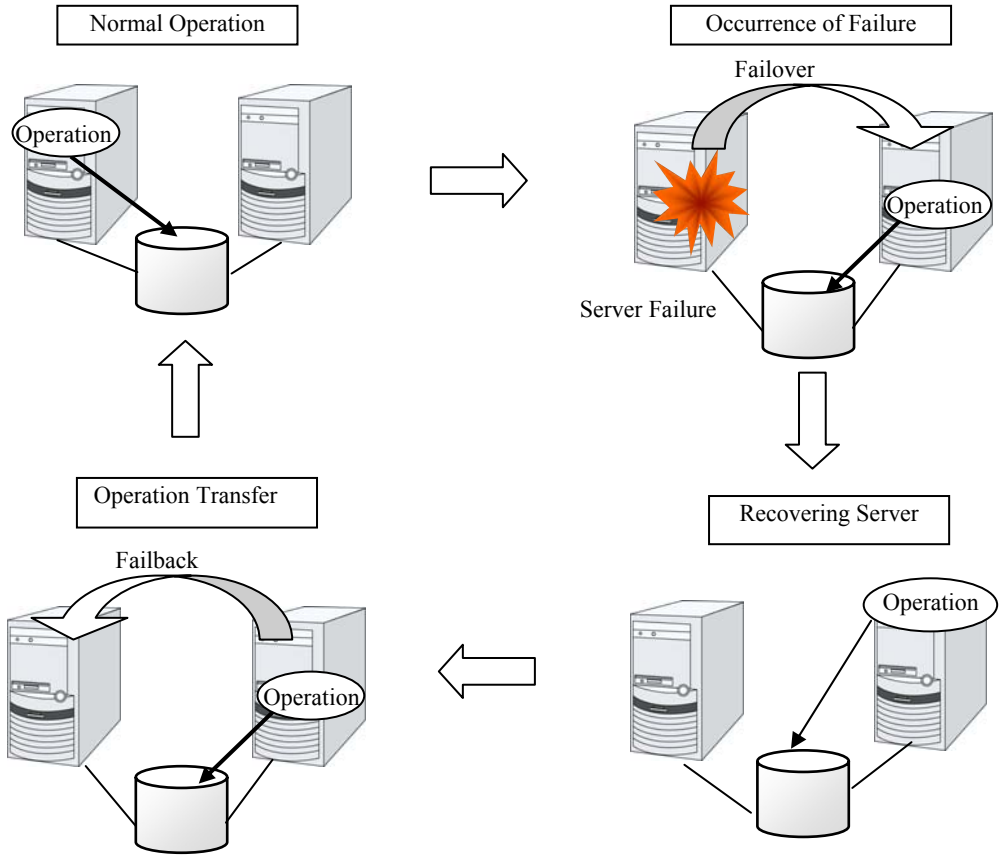


Figure 1-2: From occurrence of a failure to recovery

When the specification of the failover destination server does not meet the system requirements or overload occurs due to multi-directional stand-by, operations on the original server are preferred. In such a case, a failback takes place to resume operations on the original server.

A stand-by mode where there is one operation and no operation is active on the stand-by server, as shown in Figure 1-3, is referred to as uni-directional stand-by. A stand-by mode where there are two or more operations with each server of the cluster serving as both active and stand-by servers is referred to as multi-directional stand-by.

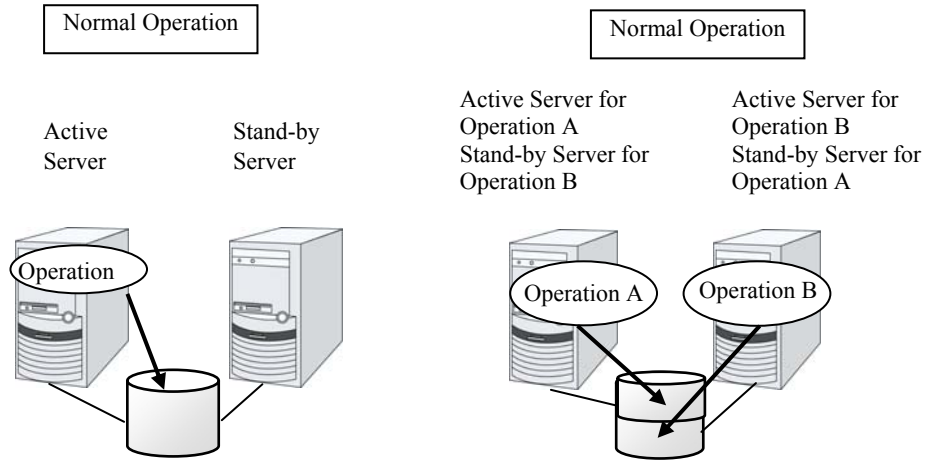


Figure 1-3: HA cluster topology

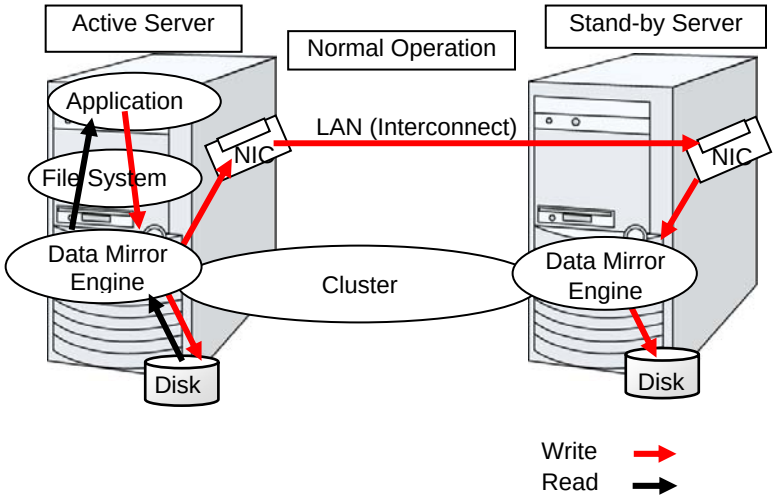
Data mirror type

The shared disk type cluster system is good for large-scale systems. However, creating a system with this type can be costly because shared disks are generally expensive. The data mirror type cluster system provides the same functions as the shared disk type with smaller cost through mirroring of server disks.

The data mirror type is not recommended for large-scale systems that handle a large volume of data since data needs to be mirrored between servers.

When a write request is made by an application, the data mirror engine not only writes data in the local disk but sends the write request to the stand-by server via the interconnect. Interconnect is a network connecting servers. It is used to monitor whether or not the server is activated in the cluster system. In addition to this purpose, interconnect is sometimes used to transfer data in the data mirror type cluster system. The data mirror engine on the stand-by server achieves data synchronization between stand-by and active servers by writing the data into the local disk of the stand-by server.

For read requests from an application, data is simply read from the disk on the active server.



Note:
In ExpressCluster X3.0 for Solaris, you cannot configure data mirror type cluster.

Figure 1-4: Data mirror mechanism

Snapshot backup is applied usage of data mirroring. Because the data mirror type cluster system has shared data in two locations, you can keep the disk of the stand-by server as snapshot backup without spending time for backup by simply separating the server from the cluster.

Failover mechanism and its problems

There are various cluster systems such as failover clusters, load distribution clusters, and high performance computing (HPC) clusters. The failover cluster is one of the high availability (HA) cluster systems that aim to increase operational availability through establishing server redundancy and passing operations being executed to another server when a failure occurs. How to implement clusters and the problems related to them are explained below.

Error detection mechanism

Cluster software executes failover (for example, passing operations) when a failure that can impact continued operation is detected. The following section gives you a quick view of how the cluster software detects a failure.

Heartbeat and detection of server failures

Failures that must be detected in a cluster system are failures that can cause all servers in the cluster to stop. Server failures include hardware failures such as power supply and memory failures, and OS panic. To detect such failures, heartbeat is employed to monitor whether or not the server is active.

Some cluster software programs use heartbeat not only for checking whether or not the target is active through ping response, but for sending status information on the local server. Such cluster software programs begin failover if no heartbeat response is received in heartbeat transmission, determining no response as server failure. However, grace time should be given before determining failure, since a highly loaded server can cause delay of response. Allowing grace period results in a time lag between the moment when a failure occurred and the moment when the failure is detected by the cluster software.

Detection of resource failures

Factors causing stop of operations are not limited to stop of all servers in the cluster. Failure in disks used by applications, NIC failure, and failure in applications themselves are also factors that can cause the stop of operations. These resource failures need to be detected as well to execute failover for improved availability.

Accessing a target resource is a way employed to detect resource failures if the target is a physical device. For monitoring applications, trying to service ports within the range not impacting operation is a way of detecting an error in addition to monitoring whether or not application processes are activated.

Problems with shared disk type

In a failover cluster system of the shared disk type, multiple servers physically share the disk device. Typically, a file system enjoys I/O performance greater than the physical disk I/O performance by keeping data caches in a server.

What if a file system is accessed by multiple servers simultaneously?

Since a general file system assumes no server other than the local updates data on the disk, inconsistency between caches and the data on the disk arises. Ultimately the data will be corrupted. The failover cluster system locks the disk device to prevent multiple servers from mounting a file system, simultaneously caused by a network partition.

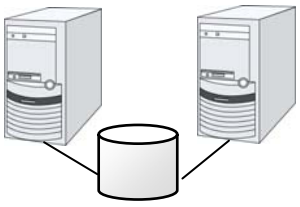


Figure 1-5: Cluster configuration with a shared disk

Network partition (split-brain-syndrome)

When all interconnects between servers are disconnected, failover takes place because the servers assume other server(s) are down. To monitor whether the server is activated, a heartbeat communication is used. As a result, multiple servers mount a file system simultaneously causing data corruption. This explains the importance of appropriate failover behavior in a cluster system at the time of failure occurrence.

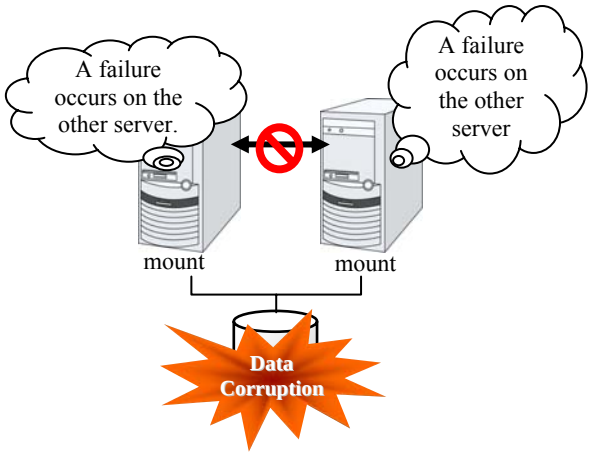


Figure 1-6: Network partition problem

The problem explained in the section above is referred to as “network partition” or “split-brain syndrome.” The failover cluster system is equipped with various mechanisms to ensure shared disk lock at the time when all interconnects are disconnected.

Taking over the cluster resources

As mentioned earlier, resources to be managed by a cluster include disks, IP addresses, and applications. The functions used in the failover cluster system to take over these resources are described below.

Taking over the data

Data to be passed from a server to another in a cluster system is stored in a partition on the shared disk. This means taking over the data is re-mounting the file system of files that the application uses on a healthy server. What the cluster software should do is simply mount the file system because the shared disk is physically connected to a server that taking over the data.

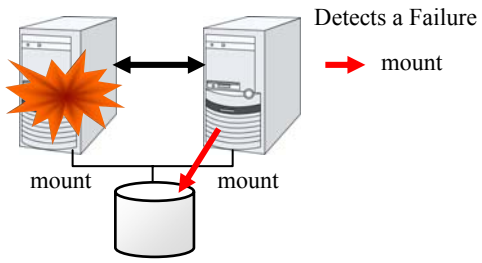


Figure 1-7: Taking over the data

The figure 1-7 may look simple, but consider the following issues in designing and creating a cluster system.

One issue to consider is recovery time for a file system. A file system to be taken over may have been used by another server or being updated just before the failure occurred and requires a file system consistency check. When the file system is large, the time spent for checking consistency will be enormous. It may take a few hours to complete the check and the time is wholly added to the time for failover (time to take over operation), and this will reduce system availability.

Another issue you should consider is writing assurance. When an application writes important data into a file, it tries to ensure the data to be written into a disk by using a function such as synchronized writing. The data that the application assumes to have been written is expected to be taken over after failover. For example, a mail server reports the completion of mail receiving to other mail servers or clients after it has securely written mails it received in a spool. This will allow the spooled mail to be distributed again after the server is restarted. Likewise, a cluster system should ensure mails written into spool by a server to become readable by another server.

Taking over the applications

The last process of the operation to be taken over by cluster software is to take over the applications. Unlike fault tolerant computers (FTC), no process status such as contents of memory is taken over in typical failover cluster systems. The applications running on a failed server are taken over by rerunning them on a healthy server.

For example, when instances of a database management system (DBMS) are taken over, the database is automatically recovered (roll-forward/roll-back) by startup of the instances. The time needed for this database recovery is typically a few minutes though it can be controlled by configuring the interval of DBMS checkpoint to a certain extent.

Many applications can restart operations by re-execution. Some applications, however, require going through procedures for recovery if a failure occurs. For these applications, cluster software allows to start up scripts instead of applications so that recovery process can be written. In a script, the recovery process, including cleanup of files half updated, is written as necessary according to factors for executing the script and information on the execution server.

Summary of failover

To summarize the behavior of cluster software:

- Detects a failure (heartbeat/resource monitoring)
- Resolves a network partition (NP resolution)
- Switches cluster resources
  - Pass data
  - Pass IP address
  - Application taking over

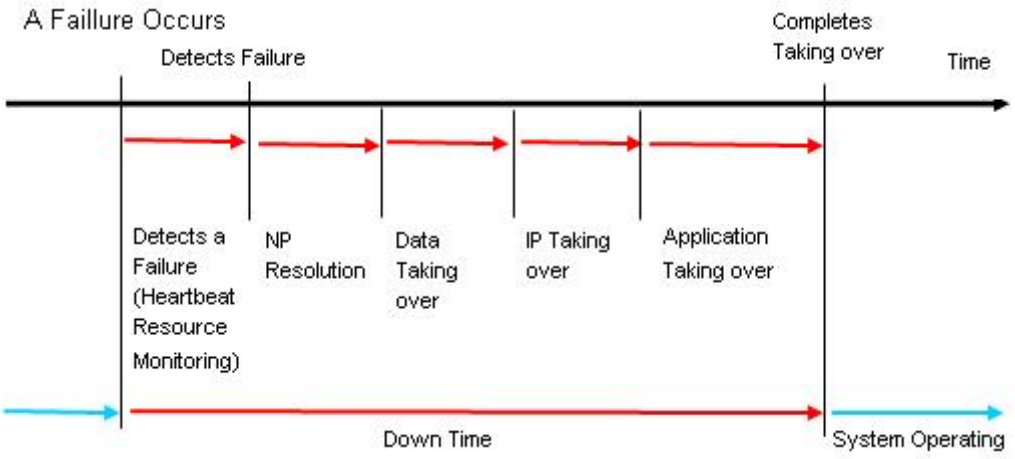


Figure 1-8: Failover time chart

Cluster software is required to complete each task quickly and reliably (see Figure 1-8.) Cluster software achieves high availability with due consideration on what has been described so far.

Eliminating single point of failure

Having a clear picture of the availability level required or aimed is important in building a high availability system. This means when you design a system, you need to study cost effectiveness of countermeasures, such as establishing a redundant configuration to continue operations and recovering operations within a short period of time, against various failures that can disturb system operations.

Single point of failure (SPOF), as described previously, is a component where failure can lead to stop of the system. In a cluster system, you can eliminate the system's SPOF by establishing server redundancy. However, components shared among servers, such as shared disk may become a SPOF. The key in designing a high availability system is to duplicate or eliminate this shared component.

A cluster system can improve availability but failover will take a few minutes for switching systems. That means time for failover is a factor that reduces availability. Solutions for the following three, which are likely to become SPOF, will be discussed hereafter although technical issues that improve availability of a single server such as ECC memory and redundant power supply are important.

- Shared disk
- Access path to the shared disk
- LAN

Shared disk

Typically a shared disk uses a disk array for RAID. Because of this, the bare drive of the disk does not become SPOF. The problem is the RAID controller is incorporated. Shared disks commonly used in many cluster systems allow controller redundancy.

In general, access paths to the shared disk must be duplicated to benefit from redundant RAID controller. There are still things to be done to use redundant access paths in Linux (described later in this chapter). If the shared disk has configuration to access the same logical disk unit (LUN) from duplicated multiple controllers simultaneously, and each controller is connected to one server, you can achieve high availability by failover between nodes when an error occurs in one of the controllers.

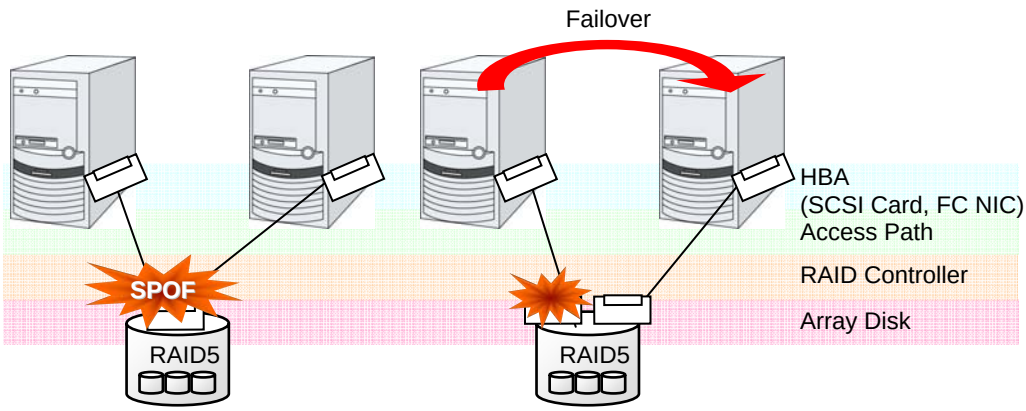


Figure 1-9: Example of the shared disk RAID controller and access paths being SPOF (left) and an access path connected to a RAID controller

With a failover cluster system of data mirror type, where no shared disk is used, you can create an ideal system having no SPOF because all data is mirrored to the disk in the other server. However you should consider the following issues:

- Disk I/O performance in mirroring data over the network (especially writing performance)
- System performance during mirror resynchronization in recovery from server failure (mirror copy is done in the background)
- Time for mirror resynchronization (clustering cannot be done until mirror resynchronization is completed)

In a system with frequent data viewing and a relatively small volume of data, choosing the data mirror type for clustering is a key to increase availability.

Access path to the shared disk

In a typical configuration of the shared disk type cluster system, the access path to the shared disk is shared among servers in the cluster. To take SCSI as an example, two servers and a shared disk are connected to a single SCSI bus. A failure in the access path to the shared disk can stop the entire system.

What you can do for this is to have a redundant configuration by providing multiple access paths to the shared disk and make them look as one path for applications. The device driver allowing such is called a path failover driver. It is important to configure multiple access paths for the shared disk by using this path failover to improve availability.

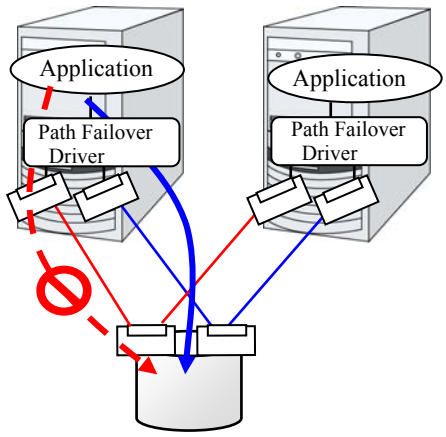


Figure 1-10: Path failover driver

LAN

In any systems that run services on a network, a LAN failure is a major factor that disturbs operations of the system. If appropriate settings are made, availability of cluster system can be increased through failover between nodes at NIC failures. However, a failure in a network device that resides outside the cluster system disturbs operation of the system.

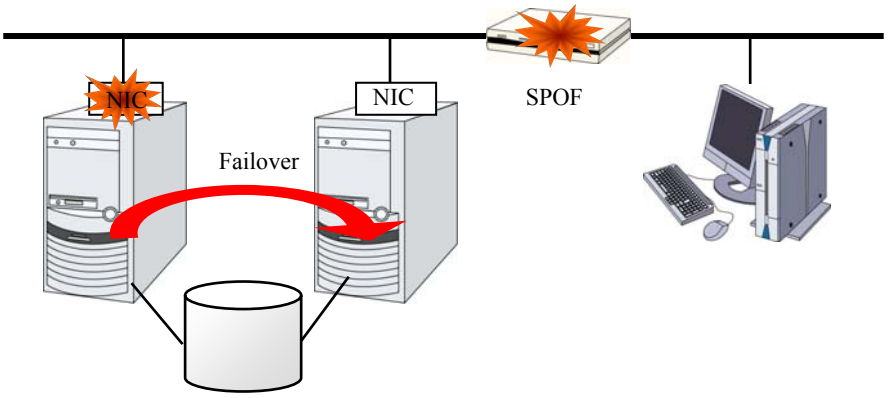


Figure 1-11: Example of router becoming SPOF

LAN redundancy is a solution to tackle device failure outside the cluster system and to improve availability. You can apply ways used for a single server to increase LAN availability. For example, choose a primitive way to have a spare network device with its power off, and manually replace a failed device with this spare device. Choose to have a multiplex network path through a redundant configuration of high-performance network devices, and switch paths automatically. Another option is to use a driver that supports NIC redundant configuration such as Intel's ANS driver.

Load balancing appliances and firewall appliances are also network devices that are likely to become SPOF. Typically they allow failover configurations through standard or optional software. Having redundant configuration for these devices should be regarded as requisite since they play important roles in the entire system.

Operation for availability

Evaluation before staring operation

Given many of factors causing system troubles are said to be the product of incorrect settings or poor maintenance, evaluation before actual operation is important to realize a high availability system and its stabilized operation. Exercising the following for actual operation of the system is a key in improving availability:

- Clarify and list failures, study actions to be taken against them, and verify effectiveness of the actions by creating dummy failures.
- Conduct an evaluation according to the cluster life cycle and verify performance (such as at degenerated mode)
- Arrange a guide for system operation and troubleshooting based on the evaluation mentioned above.

Having a simple design for a cluster system contributes to simplifying verification and improvement of system availability.

Failure monitoring

Despite the above efforts, failures still occur. If you use the system for long time, you cannot escape from failures: hardware suffers from aging deterioration and software produces failures and errors through memory leaks or operation beyond the originally intended capacity. Improving availability of hardware and software is important yet monitoring for failure and troubleshooting problems is more important. For example, in a cluster system, you can continue running the system by spending a few minutes for switching even if a server fails. However, if you leave the failed server as it is, the system no longer has redundancy and the cluster system becomes meaningless should the next failure occur.

If a failure occurs, the system administrator must immediately take actions such as removing a newly emerged SPOF to prevent another failure. Functions for remote maintenance and reporting failures are very important in supporting services for system administration. Solaris is known for providing excellent remote maintenance functions. Mechanism for reporting failures are coming in place. To achieve high availability with a cluster system, you should:

- Remove or have complete control on single point of failure.
- Have a simple design that has tolerance and resistance for failures, and be equipped with a guide for operation and troubleshooting.
- Detect a failure immediately and take appropriate actions against it.

Chapter 2 Using ExpressCluster

This chapter explains the components of ExpressCluster, how to design a cluster system, and how to use ExpressCluster.

This chapter covers:

What is ExpressCluster?

ExpressCluster is software that enhances availability and expandability of systems by a redundant (clustered) system configuration. The application services running on the active server are automatically taken over to a standby server when an error occurs in the active server.

ExpressCluster modules

ExpressCluster consists of following three modules:

ExpressCluster Server

A core component of ExpressCluster. Includes all high availability function of the server. The server function of the WebManager is also included.

ExpressCluster X WebManager (WebManager)

A tool to manage ExpressCluster operations. Uses a Web browser as a user interface. The WebManager is installed in ExpressCluster Server, but it is distinguished from the ExpressCluster Server because the WebManager is operated from the Web browser on the management PC.

ExpressCluster X Builder (Builder)

A tool for editing the cluster configuration data. The Builder also uses Web browser as a user interface. The following two versions of Builder are provided: the offline version, which is installed on your terminal as software independent of ExpressCluster Server, and the online version, which is opened by clicking the **setup mode** icon on the WebManager screen toolbar or **Setup Mode** on the **View** menu. Usually, it is not required to install this. Install this separately only when this is used offline.

Software configuration of ExpressCluster

The software configuration of ExpressCluster should look similar to the figure below. Install the ExpressCluster Server (software) on a Solaris server, and the Builder on a management PC. Because the main functions of WebManager and Builder are included in ExpressCluster Server, it is not necessary to separately install them. However, to use the Builder in an environment where ExpressCluster Server is not accessible, the offline version of Builder must be installed on the PC. The WebManager or Builder can be used through the Web browser on the management PC or on each server in the cluster.

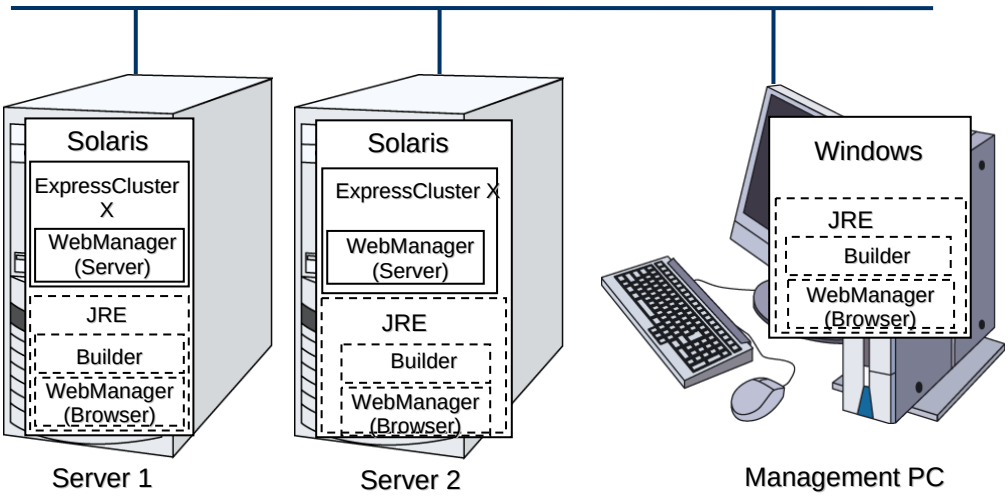


Figure 2-1 Software configuration of ExpressCluster

How an error is detected in ExpressCluster

There are three kinds of monitoring in ExpressCluster: (1) server monitoring, (2) application monitoring, and (3) internal monitoring. These monitoring functions let you detect an error quickly and reliably. The details of the monitoring functions are described below.

What is server monitoring?

Server monitoring is the most basic function of the failover-type cluster system. It monitors if a server that constitutes a cluster is properly working.

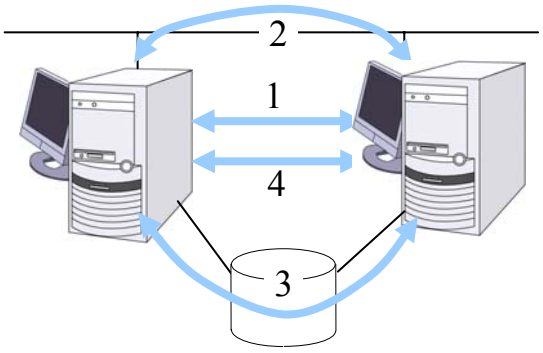
ExpressCluster regularly checks whether other servers are properly working in the cluster system. This way of verification is called “heartbeat communication.” The heartbeat communication uses the following communication paths:

Interconnect-dedicated LAN

Uses an Ethernet NIC in communication path dedicated to the failover-type cluster system. This is used to exchange information between the servers as well as to perform heartbeat communication.

Public LAN

Uses a communication path used for communication with client machine as an alternative interconnect. Any Ethernet NIC can be used as long as TCP/IP can be used. This is also used to exchange information between the servers and to perform heartbeat communication.



1. Interconnect-dedicated LAN
2. Public LAN
3. Shared disk
4. COM port

Shared disk

Creates an ExpressCluster-dedicated partition (ExpressCluster partition) on the disk that is connected to all servers that constitute the failover-type cluster system, and performs heartbeat communication on the ExpressCluster partition.

COM port

Performs heartbeat communication between the servers that constitute the failover-type cluster system through a COM port, and checks whether other servers are working properly.

Having these communication paths dramatically improves the reliability of the communication between the servers, and prevents the occurrence of network partition.

Note:
Network partition (also known as “split-brain syndrome”) refers to a condition when a network gets split by having a problem in all communication paths of the servers in a cluster. In a cluster system that is not capable of handling a network partition, a problem occurred in a communication path and a server cannot be distinguished. As a result, multiple servers may access the same resource and cause the data in a cluster system to be corrupted.

What is application monitoring?

Application monitoring is a function that monitors applications and factors that cause a situation where an application cannot run.

Activation status of application monitoring

An error can be detected by starting up an application from an exec resource in ExpressCluster and regularly checking whether a process is active or not by using the pid monitor resource. It is effective when the factor for application to stop is due to error termination of an application.

Note:

An error in resident process cannot be detected in an application started up by ExpressCluster. When the monitoring target application starts and stops a resident process, an internal application error (such as application stalling, result error) cannot be detected.

Resource monitoring

An error can be detected by monitoring the cluster resources (such as disk partition and IP address) and public LAN using the monitor resources of the ExpressCluster. It is effective when the factor for application to stop is due to an error of a resource which is necessary for an application to operate.

What is internal monitoring?

Internal monitoring refers to an inter-monitoring of modules within ExpressCluster. It monitors whether each monitoring function of ExpressCluster is properly working. Activation status of ExpressCluster process monitoring is performed within ExpressCluster.

- Critical monitoring of CLUSTERPRO process

Monitorable and non-monitorable errors

There are monitorable and non-monitorable errors in ExpressCluster. It is important to know what can or cannot be monitored when building and operating a cluster system.

Detectable and non-detectable errors by server monitoring

Monitoring condition: A heartbeat from a server with an error is stopped

Example of errors that can be monitored:

- Hardware failure (of which OS cannot continue operating)
- System panic

Example of error that cannot be monitored:

- Partial failure on OS (for example, only a mouse or keyboard does not function)

Detectable and non-detectable errors by application monitoring

Monitoring conditions: Termination of applications with errors, continuous resource errors, and disconnection of a path to the network devices.

Example of errors that can be monitored:

- Abnormal termination of an application
- Failure to access the shared disk (such as HBA1 failure)
- Public LAN NIC problem

Example of errors that cannot be monitored:

- Application stalling and resulting in error. ExpressCluster cannot monitor application stalling and error results. However, it is possible to perform failover by creating a program that monitors applications and terminates itself when an error is detected, starting the program using the exec resource, and monitoring application using the PID monitor resource.

Network partition resolution

When the stop of a heartbeat is detected from a server, ExpressCluster determines whether it is an error in a server or a network partition. If it is judged as a server failure, failover (activate resources and start applications on a healthy server) is performed. If it is judged as network partition, protecting data is given priority over taking over the operations, so processing such as emergency shutdown is performed.
The following is the network partition resolution method:

- ping method

Related Information:

For the details on the network partition resolution method, see Chapter 7, "Details on network partition resolution resources" in Section II of the Reference Guide.

Failover mechanism

When an error is detected, ExpressCluster determines whether an error detected before failing over is an error in a server or a network partition. Then a failover is performed by activating various resources and starting up applications on a properly working server.

The group of resources which fail over at the same time is called a "failover group." From a user's point of view, a failover group appears as a virtual computer.

Note:

In a cluster system, a failover is performed by restarting the application from a properly working node. Therefore, what is saved in an application memory cannot be failed over.

From occurrence of error to completion of failover takes a few minutes. See the figure 2-2 below:

[1] HBA is an abbreviation for host bus adapter. This adapter is not for the shared disk, but for the server.

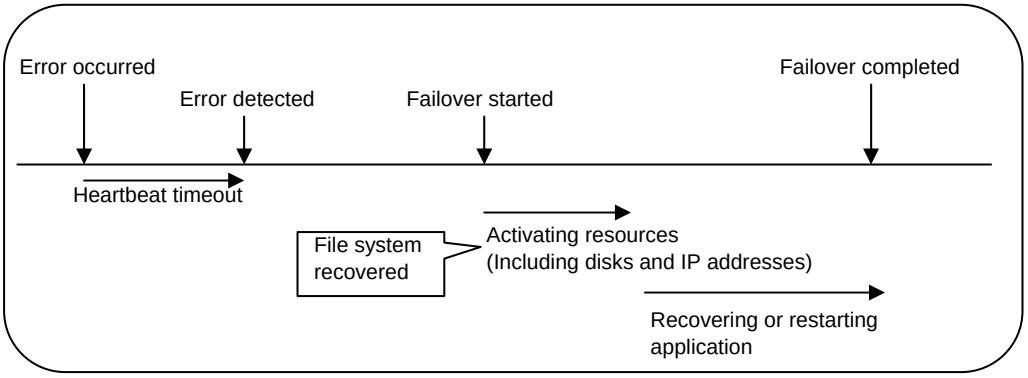


Figure 2-2 Failover time chart

Heartbeat timeout

- The time for a standby server to detect an error after that error occurred on the active server.
- The setting values of the cluster properties should be adjusted depending on the application load. (The default value is 90 seconds.)

Activating various resources

- The time to activate the resources necessary for operating an application.
- The resources can be activated in a few seconds in ordinary settings, but the required time changes depending on the type and the number of resources registered to the failover group. For more information, refer to the Installation and Configuration Guide.

Start script execution time

- The data recovery time for a roll-back or roll-forward of the database and the startup time of the application to be used in operation.
- The time for roll-back or roll-forward can be predicted by adjusting the check point interval. For more information, refer to the document that comes with each software product.

Failover resources

ExpressCluster can failover the following resources:

Switchable partition

- Resources such as disk resources and others.
- A disk partition to store the data that the application takes over.

Floating IP Address

- By connecting an application using the floating IP address, a client does not have to be conscious about switching the servers due to failover processing.
- It is achieved by dynamic IP address allocation to the public LAN adapter and sending ARP packet. Connection by floating IP address is possible from most of the network devices.

Script (exec resource)

- In ExpressCluster, applications are started up from the scripts.
- The file failed over on the shared disk may not be complete as data even if it is properly working as a file system. Write the recovery processing specific to an application at the time of failover in addition to the startup of an application in the scripts.

Note:
In a cluster system, failover is performed by restarting the application from a properly working node. Therefore, what is saved in an application memory cannot be failed over.

System configuration of the failover type cluster

In a failover-type cluster, a disk array device is shared between the servers in a cluster. When an error occurs on a server, the standby server takes over the applications using the data on the shared disk.

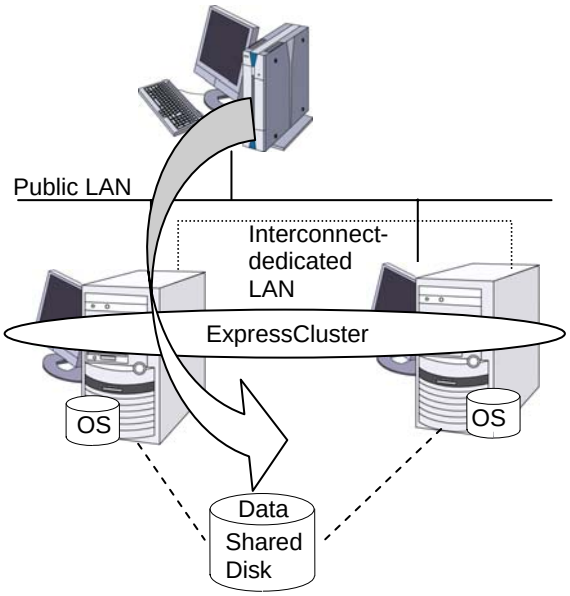


Figure 2-3 System configuration

A failover-type cluster can be divided into the following categories depending on the cluster topologies:

Uni-Directional Standby Cluster System

In the uni-directional standby cluster system, the active server runs applications while the other server, the standby server, does not. This is the simplest cluster topology and you can build a high-availability system without performance degradation after failing over.

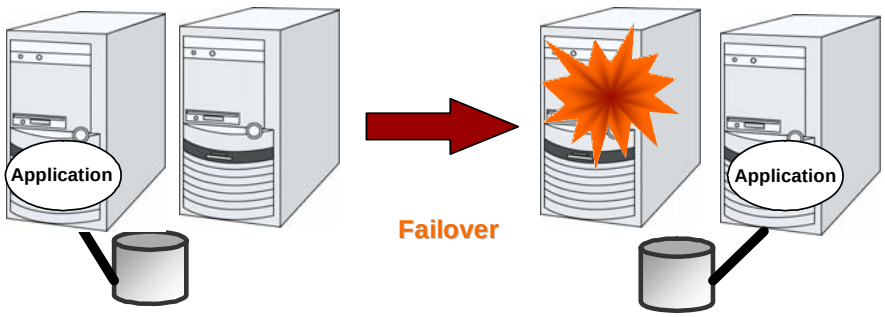


Figure 2-4 Uni-directional standby cluster system

Same Application Multi Directional Standby Cluster System

In the same application multi-directional standby cluster system, the same applications are activated on multiple servers. These servers also operate as standby servers. The applications must support multi-directional standby operation. When the application data can be split into multiple data, depending on the data to be accessed, you can build a load distribution system per data partitioning basis by changing the client's connecting server.

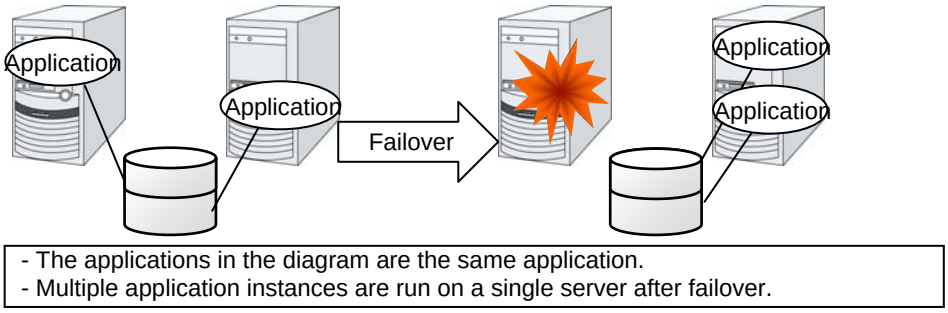


Figure 2-5 Same application multi directional standby cluster system

Different Application – Multi Directional Standby Cluster System

In the different application multi-directional standby cluster system, different applications are activated on multiple servers and these servers also operate as standby servers. The applications do not have to support multi-directional standby operation. A load distribution system can be built per application unit basis.

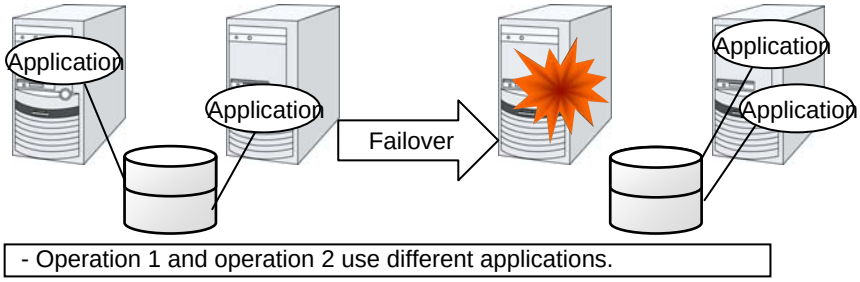


Figure 2-6 Different application multi directional standby cluster system

Node to Node Configuration

The configuration can be expanded with more nodes by applying the configurations introduced thus far. In a node to node configuration described below, three different applications are run on three servers and one standby server takes over the application if any problem occurs. In a uni-directional standby cluster system, one of the two servers functions as a standby server. However, in a node to node configuration, only one of the four server functions as a standby server and performance deterioration is not anticipated if an error occurs only on one server.

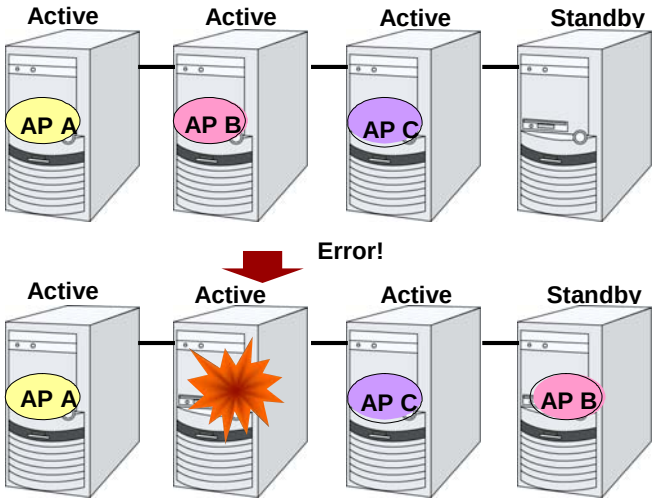


Figure 2-7 Node to Node configuration

Hardware configuration of the shared disk type cluster

The hardware configuration of the shared disk in ExpressCluster is described below. In general, the following is used for communication between the servers in a cluster system:

- Two NIC cards (one for external communication, one for ExpressCluster)
- COM port connected by RS232C cross cable
- Specific space of a shared disk

SCSI or FibreChannel can be used for communication interface to a shared disk; however, recently FibreChannel is more commonly used.

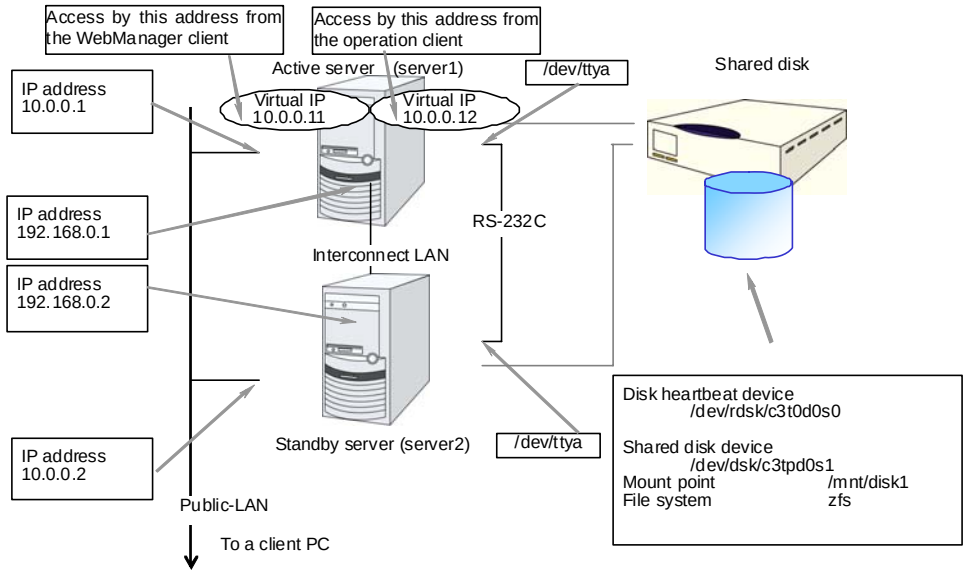


Figure 2-8 Sample of cluster environment when a shared disk is used

What is cluster object?

In ExpressCluster, the various resources are managed as the following groups:

Cluster object
Configuration unit of a cluster.

Server object
Indicates the physical server and belongs to the cluster object.

Heartbeat resource object
Indicates the network part of the physical server and belongs to the server object.

Network partition resolution resource object
Indicates the network partition resolution mechanism and belongs to the server object.

Group object
Indicates a virtual server and belongs to the cluster object.

Group resource object
Indicates resources (network, disk) of the virtual server and belongs to the group object.

Monitor resource object
Indicates monitoring mechanism and belongs to the cluster object.

What is a resource?

In ExpressCluster, a group used for monitoring the target is called “resources.” There are four types of resources and are managed separately. Having resources allows distinguishing what is monitoring and what is being monitored more clearly. It also makes building a cluster and handling an error easy. The resources can be divided into heartbeat resources, network partition resolution resources, group resources, and monitor resources.

Heartbeat resources

Heartbeat resources are used for verifying whether the other server is working properly between servers. The following heartbeat resources are currently supported:

LAN heartbeat resource
Uses Ethernet for communication.

COM heartbeat resource
Uses RS232C (COM) for communication.

Disk heartbeat resource
Uses a specific partition (cluster partition for disk heartbeat) on the shared disk for communication. It can be used only on a shared disk configuration.

Network partition resolution resources

The resource used for solving the network partition is shown below:

PING network partition resolution resource
This is a network partition resolution resource by the PING method.

Group resources

A group resource constitutes a unit when a failover occurs. The following group resources are currently supported:

Floating IP resource (fip)
Provides a virtual IP address. A client can access virtual IP address the same way as the regular IP address.

EXEC resource (exec)
Provides a mechanism for starting and stopping the applications such as DB and httpd.

Disk resource (disk)
Provides a specified partition on the shared disk. It can be used only on a shared disk configuration.

NAS resource (nas)
Connect to the shared resources on NAS server. Note that it is not a resource that the cluster server behaves as NAS server.

Virtual IP resource (vip)
Provides a virtual IP address. This can be accessed from a client in the same way as a general IP address. This can be used in the remote cluster configuration among different network addresses.

Volume Manager resource (volmgr)
Controls logical disks managed by volume manager.

VM resource (vm)
Starts, stops, or migrates the virtual machine.

Dynamic DNS resource (ddns)
Registers the virtual host name and the IP address of the active server to the dynamic DNS server.

Monitor resources

A monitor resource monitors a cluster system. The following monitor resources are currently supported:

IP monitor resource (ipw)
Provides a monitoring mechanism of an external IP address.

Disk monitor resource (diskw)
Provides a monitoring mechanism of the disk. It also monitors the shared disk.

PID monitor resource (pidw)
Provides a monitoring mechanism to check whether a process started up by exec resource is active or not.

User mode monitor resource (userw)
Provides a monitoring mechanism for a stalling problem in the user space.

NIC Link Up/Down monitor resource (miiw)

Provides a monitoring mechanism for link status of LAN cable.

Multi target monitor resource (mtw)

Provides a status with multiple monitor resources.

Virtual IP monitor resource (vipw)

Provides a mechanism for sending RIP packets of a virtual IP resource.

Custom monitor resource (genw)

Provides a monitoring mechanism to monitor the system by the operation result of commands or scripts which perform monitoring, if any.

MySQL monitor resource (mysqlw)

Provides a monitoring mechanism for MySQL database.

nfs monitor resource (nfsw)

Provides a monitoring mechanism for nfs file server.

Oracle monitor resource (oraclew)

Provides a monitoring mechanism for Oracle database.

PostgreSQL monitor resource (psqlw)

Provides a monitoring mechanism for PostgreSQL database.

samba monitor resource (sambaw)

Provides a monitoring mechanism for samba file server.

Volume Manager monitor resource (volmgrw)

Provides a monitoring mechanism for logical disks managed by volume manager.

VM monitor resource (vmw)

Checks whether the virtual machine is alive.

Message receive monitor resource (mrw)

Specifies the action to take when an error message is received and how the message is displayed on the WebManager.

Dynamic DNS monitor resource (ddnsw)

Periodically Registers the virtual host name and IP address of the active server to the dynamic DNS server.

Getting started with ExpressCluster

Refer to the following guides when building a cluster system with ExpressCluster:

Latest information

Refer to Section II, “Installing ExpressCluster” in this guide.

Designing a cluster system

Refer to Section I, “Configuring a cluster system” in the *Installation and Configuration Guide* and Section II, “Resource details” in the *Reference Guide*.

Configuring a cluster system

Refer to the *Installation and Configuration Guide.* When using an optional monitoring command, refer to the *Administrator’s Guide* that is available for each target monitoring application.

Troubleshooting the problem

Refer to Section III, “Maintenance information” in the *Reference Guide*.

Section II Installing ExpressCluster

This section provides the latest information on the ExpressCluster. The latest information on the supported hardware and software is described in detail. Topics such as restrictions, known problems, and how to troubleshoot the problem are covered.

- Chapter 3 Installation requirements for ExpressCluster
- Chapter 4 Latest version information
- Chapter 5 Notes and Restrictions
- Chapter 6 Upgrading ExpressCluster

Chapter 3 Installation requirements for ExpressCluster

This chapter provides information on system requirements for ExpressCluster.
This chapter covers:

Hardware

ExpressCluster operates on the following server architectures:

- i86pc(x86)
- i86pc(x86_64)

General server requirements

Required specifications for ExpressCluster Server are the following:

- RS-232C port 1 port (not necessary when configuring a cluster with 3 or more nodes)
- Ethernet port 2 or more ports
- Shared disk
- CD-ROM drive

When using the off-line Builder upon constructing and changing the existing configuration, one of the following is required for communication between the off-line Builder and servers:

- Removable media (for example or USB flash drive)
- A machine to operate the off-line Builder and a way to share files

Software

System requirements for ExpressCluster Server

Supported OS versions

There are driver modules unique to Express Cluster. The driver module versions are listed below.

i86pc(x86)

Version	Run clpka and support	ExpressCluster Version	Remarks
Solaris10 10/08	Yes	3.0.0-1~	
Solaris10 10/09	Yes	3.0.0-1~	

i86pc(x86_64)

Version	Run clpka and support	ExpressCluster Version	Remarks
Solaris10 10/08	Yes	3.0.0-1~	
Solaris10 10/09	Yes	3.0.0-1~	

Applications supported by monitoring options

Version information of the applications to be monitored by monitor resources is described below.

For the support information on the monitoring options of command type (that are registered as script resources at setup), which is provided on ExpressCluster 1.0.x-x, see the administrator's guide of each option.

i86pc (x86)

Monitor resource	Monitored application	ExpressCluster version	Remark
Oracle monitor	Oracle Database 10g Release 2 (10.2)	3.0.0-1~	
PostgresSQL monitor	PostgresSQL 8.3	3.0.0-1~	
	PostgresSQL 8.4	3.0.0-1~	
MySQL monitor	MySQl 4.0	3.0.0-1~	
	MySQL 5.1	3.0.0-1~	
Samba monitor	Samba 3.2	3.0.0-1~	
nfs monitor	No specified version	3.0.0-1~	

i86pc (x86_64)

Monitor resource	Monitored application	ExpressCluster version	Remark
Oracle monitor	Oracle Database 10g Release 2 (10.2)	3.0.0-1~	
Samba monitor	Samba 3.2	3.0.0-1~	
nfs monitor	No specified version	3.0.0-1~	

Required memory and disk size

	Required memory size	Required disk size	Remark
	User mode	Right after installation	
i86pc(x86)	64MB	20MB	
i86pc(x86_64)	64MB	20MB	

System requirements for the Builder

Supported operating systems and browsers

Refer to the website, *http://www.ace.comp.nec.co.jp/CLUSTERPRO/global-link.html*, for the latest information. Currently supported operating systems and browsers are the following:

Operating system	Browser	Language
Microsoft Windows® XP SP2 (IA32)	IE6 SP2	English/Japanese/Chinese
	IE7	English/Japanese/Chinese
Microsoft Windows Vista® (IA32)	IE7	English/Japanese/Chinese
	IE8	English/Japanese/Chinese
Microsoft Windows® 7(IA32)	IE8	English/Japanese/Chinese
Microsoft Windows Server 2003 SP1 or later (IA32)	IE6 SP1	English/Japanese/Chinese
Microsoft Windows Server 2008 (IA32)	IE7	English/Japanese/Chinese

Note:

The Builder does not operate on 64-bit machines. Use 32-bit machine when constructing and changing a cluster configuration.

Java runtime environment

Required:

Sun Microsystems, Java ™ Runtime Environment, Version 6.0 Update 21 (1.6.0_21) or later

Required memory and disk size

Required memory size: 32MB or more

Required disk size: 5MB (excluding the size required for Java runtime environment)

Supported ExpressCluster versions

Offline Builder version	ExpressCluster X package version
3.0.0-1	3.0.0-1

Note:

When you use the Offline Builder and the ExpressCluster package, a combination of their versions should be the one shown above. The Builder may not operate properly if they are used in a different combination.

System requirements for the WebManager

Supported operating systems and browsers

Refer to the website, *http://www.ace.comp.nec.co.jp/CLUSTERPRO/global-link.html,* for the latest information. Currently the following operating systems and browsers are supported:

Operating system	Browser	Language
Microsoft Windows® XP(IA32)	IE6 SP2	English/Japanese/Chinese
	IE7	English/Japanese/Chinese
Microsoft Windows Vista™ (IA32)	IE7	English/Japanese/Chinese
	IE8	English/Japanese/Chinese
Microsoft Windows® 7(IA32)	IE8	English/Japanese/Chinese
Microsoft Windows Server 2003 SP1 or later (IA32, x86_64)	IE6 SP1	English/Japanese/Chinese
Microsoft Windows Server 2008 (IA32)	IE7	English/Japanese/Chinese

Java runtime environment

Java runtime environment is required to use the WebManager.

Required:

Sun Microsystems, Java ™ Runtime Environment, Version 6.0 Update 21 (1.6.0_21) or later

Required memory and disk size

Required memory size: 40MB or more

Required disk size: 600KB (excluding the size required for Java runtime environment)

Chapter 4 Latest version information

This chapter provides the latest information on ExpressCluster. It also introcudes the enhanced and improved points/faunctions on this latest version.
This chapter covers:

Correspondence list of ExpressCluster and a manual

This book has explained on the assumption that ExpressCluster of the following version. Be careful of the number of versions of the version of ExpressCluster, and a manual.

ExpressCluster Version	Manual	Manual Version	Remarks
3.0.0-1	Installation and Configuration Guide	First Edition	
	Getting Started Guide	First Edition	
	Reference Guide	First Edition	
	Integrated WebManager Administrator's Guide	First Edition	

Enhanced functions

Upgrade has been performed on the following minor versions.

Number	Version (in detail)	Upgraded Section
1	3.0.0-1	The WebManager and Builder can now be used from the same browser window.
2	3.0.0-1	The cluster configuration wizard has been upgraded.
3	3.0.0-1	Some settings can now be automatically acquired in the cluster configuration wizard.
4	3.0.0-1	The Integrated WebManager can now be used from a browser.
5	3.0.0-1	A function has been implemented to check settings when uploading configuration data.
6	3.0.0-1	ExpressCluster can now automatically select the failover destination when an error occurs.
7	3.0.0-1	A function has been implemented to control failovers across server groups.
8	3.0.0-1	**All Groups** can now be selected as the failover target when an error is detected.
9	3.0.0-1	The start wait time can now be skipped.
10	3.0.0-1	ExpressCluster can now manage external errors.
11	3.0.0-1	Dump information can now be acquired when the target monitoring application times out.
12	3.0.0-1	Detailed information about an Oracle database can now be acquired if an error is detected while monitoring it.
13	3.0.0-1	A function has been implemented to register a virtual host name to the dynamic DNS server.

14	3.0.0-1	When global Solaris container zones are in a cluster, non-global zones can now be handled as resources.
15	3.0.0-1	Additional OSs are now supported.
16	3.0.0-1	Additional applications are now supported.
17	3.0.0-1	Additional network warning lights are now supported.

Corrected information

There is no corrected information because this is the first edition.

Chapter 5 Notes and Restrictions

This chapter provides information on known problems and how to troubleshoot the problems.
This chapter covers:

Designing a system configuration

Hardware selection, system configuration, and shared disk configuration are introduced in this section.

Function list and necessary license

The following option products are necessary as many as the number of servers.

Necessary function	Necessary license
Oracle monitor resource	CLUSTERPRO X Database Agent 3.0
PostgreSQL monitor resource	CLUSTERPRO X Database Agent 3.0
MySQL monitor resource	CLUSTERPRO X Database Agent 3.0
samba monitor resource	CLUSTERPRO X File Server Agent 3.0
nfs monitor resource	CLUSTERPRO X File Server Agent 3.0

Supported operating systems for the Builder and WebManager

- The Builder does not run on 64-bit machines. Use a 32-bit machine when configuring and changing the configuration of a cluster system.

Hardware requirements for shared disks

- A shared disk does not support a Solaris md stripe set, volume set, mirroring, and stripe set with parity.

NIC link up/down monitor resource

Some NIC boards and drivers do not support required ioctl( ).

To check if NIC Link Up/Down monitor resource can be used by using ExpressCluster on an actual machine, follow the steps below to check the operation.

1. Register NIC Link Up/Down monitor resource with the configuration information.
   Select **No Operation** for the configuration of recovery operation of NIC Link Up/Down monitor resource upon failure detection.
2. Start the cluster.
3. Check the status of NIC Link Up/Down monitor resource.
   If the status of NIC Link Up/Down monitor resource is abnormal while LAN cable link status is normal, NIC Link Up/Down monitor resource cannot be operated.
4. If NIC Link Up/Down monitor resource status becomes abnormal when LAN cable link status is made abnormal status (link down status), NIC Link Up/Down monitor resource can be operated.
   If the status remains to be normal, NIC Link Up/Down monitor resource cannot be operated.

Before and at the time of installing operating system

Notes on parameters to be determined when installing an operating system, allocating resources, and naming rules are described in this section.

/opt/nec/clusterpro file system

It is recommended to use a file system that has journaling functions to improve tolerance for system failure.

Dependent library

- SUNWlxml

Before installing ExpressCluster and after installing OS

Notes after installing an operating system, when configuring OS and disks are described in this section.

Communication port number

In ExpressCluster, the following port numbers are used. You can change the port number by using the Builder.

Make sure not to access the following port numbers from a program other than ExpressCluster.

Configure to be able to access the port number below when setting a firewall on a server.

Server to Server **Loopback in servers**					
					Used for
Server	Automatic allocation [1]	-	Server	29001/TCP	Internal communication
Server	Automatic allocation	-	Server	29002/TCP	Data transfer
Server	Automatic allocation	-	Server	29002/UDP	Heartbeat
Server	Automatic allocation	-	Server	29003/UDP	Alert synchronization
Server	Automatic allocation	-	Server	icmp	duplication check for FIP/VIP resource
Server	Automatic allocation	-	Server	XXXX [2]/TCP	Internal log communication

WebManager to Server					
					Used for
WebManager	Automatic allocation	-	Server	29003/TCP	http communication

Server connected to the Integrated WebManager to Target server					
					Used for
Server connected to the Integrated WebManager	Automatic allocation	-	Server	29003/TCP	http communication

Others					
					Used for
Server	Automatic allocation	-	Network warning light	514/TCP	Network warning light control
Server	Automatic allocation	-	Management LAN of server BMC	623/UDP	BMC control (Forced stop / Chassis lamp association)
Server	Automatic allocation	-	Monitoring target	icmp	IP monitor
Server	Automatic allocation	-	NFS server	icmp	Checking if NFS server is active by NAS resource
Server	Automatic allocation	-	Monitoring target	icmp	Monitoring target of Ping method network partition resolution resource

In automatic allocation, a port number not being used at a given time is allocated.

Select **UDP** for the **Communication Method for Internal Logs** in **Cluster Properties**, **Port No. (Log)** tab. Use the port number configured in Port No. Communication port is not used for the default log communication method **UNIX Domain**.

Changing the range of automatic allocation for the communication port numbers

- The range of automatic allocation for the communication numbers managed by OS and the communication numbers used by Express Cluster may be duplicated.
- Change the OS settings to avoid duplication when the range of automatic allocation for the communication numbers managed by OS and the communication numbers used by Express Cluster are duplicated.

Examples of checking and displaying OS setting conditions.

The range of automatic allocation for the TCP communication port numbers can be checked with the following commands.

```
# ndd -get /dev/tcp tcp_smallest_anon_port
32768
# ndd -get /dev/tcp tcp_largest_anon_port
65535
```

This is the condition to be assigned for the range from 32768 to 65535 when the application to conduct TCP communication requests automatic allocation for the communication port numbers to the OS.

The range of automatic allocation of the UDP communication port number can be checked with the following commands as well.

```
# ndd -get /dev/udp udp_smallest_anon_port
32768
# ndd -get /dev/udp udp_largest_anon_port
65535
```

This is the condition to be assigned for the range from 32768 to 65535 when the application to conduct UDP communication requests automatic allocation for the communication port numbers to the OS.

Examples of OS settings change

Execute the following commands when changing the range of automatic allocation for TCP port to the range of from 30000 to 65000.

```
# ndd -set /dev/tcp tcp_smallest_anon_port 30000
# ndd -set /dev/tcp tcp_largest_anon_port 65000
```

Clock synchronization

In a cluster system, it is recommended to synchronize multiple server clocks regularly. Synchronize server clocks by using ntp.

Shared disk

When you continue using the data on the shared disk at times such as server reinstallation, do not allocate a partition or create a file system.

The data on the shared disk gets deleted if you allocate a partition or create a file system.

ExpressCluster controls the file systems on the shared disk. Do not include the file systems on the shared disk to /etc/fstab in operating system.

See the Installation and Configuration Guide for steps for shared disk configuration.

Adjusting OS startup time

It is necessary to configure the time from power-on of each node in the cluster to the server operating system startup to be longer than the following:

- The time from power-on of the shared disks to the point they become available.
- Heartbeat timeout time

See the *Installation and Configuration Guide* for configuration steps.

Verifying the network settings

The network used by Interconnect is checked. It checks on all the servers in a cluster.

See the Installation and Configuration Guide for configuration steps.

Ipmiutil and OpenIPMI

The following functions use ipmitool command.

- Final Action at Activation Failure / Deactivation Failure
- Monitor resource action upon failure
- Forced Stop
- Chassis ID lamp link

Users are responsible for making decisions and assuming responsibilities. NEC does not support or assume any responsibilities for:

- Inquires about ipmitool command
- Tested operation of ipmitool command
- Malfunction of ipmitool command or error caused by such malfunction.
- Inquiries about whether or not ipmitool command is supported by servers.

- Check whether or not your server (hardware) supports ipmitool command in advance.
- Note that even if the machine complies with ipmi standard as hardware, ipmitool command may not run if you actually try to run them.

nsupdate and nslookup

- The following functions use nsupdate and nslookup.
  - Dynamic DNS resource of group resource (ddns)
  - Dynamic DNS resource of monitor resource (ddns)
- ExpressCluster does not include nsupdate and nslookup. Therefore, install the rmp files of nsupdate and nslookup, in addition to the ExpressCluster installation.
- NEC does not support the items below regarding nsupdate and nslookup. Use nsupdate and

nslookup at your own risk.

- Inquiries about nsupdate and nslookup
- Guaranteed operations of nsupdate and nslookup
- Malfunction of nsupdate or nslookup or failure caused by such a malfunction
- Inquiries about support of nsupdate and nslookup on each server

Notes when creating ExpressCluster configuration data

Notes when creating a cluster configuration data and before configuring a cluster system is described in this section.

Environment variable

The following scripts cannot be executed under the environment where more than 255 environmental variables are set. When using the following function of resource, set the number of environmental variables less than 256.

Start/Stop script executed by EXEC resource when activating/deactivating

Script executed by Custom monitor Resource when monitoring

Script before final action after the group resource or the monitor resource error is detected.

Force stop function, chassis identify lamp linkage

When using forced stop function or chassis identify lamp linkage, settings of BMC IP address, user name and password of each server are necessary. Use definitely the usename to which the password is set.

Server reset, server panic and power off

When ExpressCluster performs “Server Reset,” “Server Panic,” or “Server power off,” servers are not shut down normally. Therefore, the following may occur.

- Damage to a mounted file system
- Lost of unsaved data
- Suspension of OS dump collection

“Server reset” or “Server panic” occurs under the following settings:

- Action at an error occurred when activating/inactivating group resources
  -Keepalive Reset
  -Keepalive Panic
  -BMC Reset
  -BMC Power Off
  -BMC Power Cycle
  -BMC NMI
- Final action at detection of an error in monitor resources
  -Keepalive Reset
  -Keepalive Panic
  -BMC Reset
  -BMC Power Off
  -BMC Power Cycle
  -BMC NMI
- Action at detection of user space monitor timeout
  - Monitoring method keepalive
- Shutdown stall monitoring
  - Monitoring method keepalive
- Operation of Forced Stop
  -BMC reset

-BMC power off
-BMC cycle
-BMC NMI

Final action for group resource deactivation error

If you select **No Operation** as the final action when a deactivation error is detected, the group does not stop but remains in the deactivation error status. Make sure not to set **No Operation** in the production environment.

Stack size of the application executed by EXEC resource

- Exec resource is executed while the stack size is configured as 2MB. Thus, if an application which is started from exec resource requires the stack size of more than 2MB, stack overflow occurs.
  If stack overflow occurs, configure the stack size before starting the application.

1. If you select **Script created with this product**
   Please change stack size using ulimit command before the application is executed.

2. If you select User Application (Do not use this mode)
   Please select Script created with this product and edit script file to execute the application by the script. Also, please change stack size using ulimit command before the application is executed.

   Example of start script (start.sh)

```
-------------------------------------------------
#! /bin/sh
#*****************************************
#*                    start.sh                    *
#*****************************************

ulimit -s unlimited    # Change stack size (unlimited)

" the application to be executed"

-------------------------------------------------
```

- When you will change scripts for exec resource, please refer to Reference Guide Section II “Chapter 4 Group resource details – Understanding EXEC resources”.

Delay warning rate

If the delay warning rate is set to 0 or 100, the following can be achieved:

- When 0 is set to the delay monitoring rate

  An alert for the delay warning is issued at every monitoring.
  By using this feature, you can calculate the polling time for the monitor resource at the time the server is heavily loaded, which will allow you to determine the time for monitoring time-out of a monitor resource.

- When 100 is set to the delay monitoring rate

  The delay warning will not be issued.

Be sure not to set a low value, such as 0%, except for a test operation.

Disk monitor resource (monitoring method TUR)

- You cannot use the TUR methods on a disk or disk interface (HBA) that does not support the Test Unit Ready (TUR) of SCSI. Even if your hardware supports these commands, consult the driver specifications because the driver may not support them.
- TUR methods burdens OS and disk load less compared to Read methods.
- In some cases, TUR methods may not be able to detect errors in I/O to the actual media.

WebManager reload interval

- Do not set the “Reload Interval” in the WebManager tab for less than 30 seconds.

LAN heartbeat settings

- You need to set at least one LAN heartbeat resource. It is recommended to set two or more LAN heartbeat resources.
- It is recommended to set both LAN heartbeat resource and kernel mode LAN heartbeat resource together.

COM heartbeat resource settings

- It is recommended to use a COM heartbeat resource if your environments allows. This is because using COM heartbeat resource prevents activating both systems when the network is disconnected.

Double-byte character set that can be used in script comments

- Scripts edited in Solaris environment are dealt as EUC code, and scripts edited in Windows environment are dealt as Shift-JIS code. In case that other character codes are used, character corruption may occur depending on environment.

Failover exclusive attribute of virtual machine group

- When setting virtual machine group, do not set **Normal** or **Absolute** to **Failover exclusive attribute**.

After starting ExpressCluster operation

Notes on situations you may encounter after start operating ExpressCluster are described in this section.

Limitations during the recovery operation

Do not control the following commands, clusters and groups by the WebManager while recovery processing is changing (reactivation → failover → last operation), if a group resource is specified as a recovery target and when a monitor resource detects an error.

- Stop and suspend of a cluster
- Start, stop, moving of a group

If these operations are controlled at the transition to recovering due to an error detected by a monitor resource, the other group resources in the group may not be stopped.

Even if a monitor resource detects an error, it is possible to control the operations above after the last operation is performed.

Executable format file and script file not described in the manuals

Executable format files and script files which are not described in Chapter 4, ”ExpressCluster command reference” in the *Reference Guide* exist under the installation directory. Do not run these files on any system other than ExpressCluster. The consequences of running these files will not be supported.

Scripts in EXEC resources

EXEC resource scripts of group resources stored in the following location.

/opt/nec/clusterpro/scripts/*group-name*/*resource-name*/

The following cases, old EXEC resource scripts are not deleted automatically.

- When the EXEC resource is deleted or renamed
- When a group that belongs to the EXEC resource is deleted or renamed

Old EXEC resource scripts can be deleted when unnecessary.

Monitor resources that monitoring timing is “Active”

When monitor resources that monitoring timing is “Active” have suspended and resumed, the following restriction apply:

- In case stopping target resource after suspending monitor resource, monitor resource becomes suspended. As a result, monitoring restart cannot be executed.
- In case stopping or starting target resource after suspending monitor resource, monitoring by monitor resource starts when target resource starts.

Notes on the WebManager

- The information displayed on the WebManager does not necessarily show the latest status. If you want to get the latest information, click the **Reload** button.
- If the problems such as server shutdown occur while the WebManager is getting the information, acquiring information may fail and a part of object may not be displayed correctly. Wait for the next automatic update or click the **Reload** button to reacquire the latest information.
- Collecting logs of ExpressCluster cannot be executed from two or more WebManager simultaneously.
- If the WebManager is operated in the state that it cannot communicate with the connection destination, it may take a while until the control returns.
- If you move the cursor out of the browser in the state that the mouse pointer is displayed as a wristwatch or hourglass, the cursor may be back to an arrow.
- When going through the proxy server, make the settings for the proxy server be able to relay the port number of the WebManager.
- When updating ExpressCluster, close the browser. Clear the Java cache and open the browser.

Notes on the Builder (Config mode of Cluster Manager)

- Closing the Web browser (by clicking **Exit** from the menu) discards the edited data. Even if the configuration is changed, the dialog box to confirm to save is not displayed. When you need to save the edited data, select **File** from the menu of the Builder and click **Export** before terminating.
- Reloading the Web browser (by selecting **Refresh** button from the menu or tool bar) discards the current editing data. Even if the configuration is changed, the dialog box to confirm to save is not displayed. When you need to save the editing data, select **File** from

the menu bar of the Builder and click **Export** before reloading.

- When creating the cluster configuration data using the Builder, do not enter the value starting with 0 on the text box. For example, if you want to set 10 seconds for a timeout value, enter "10" but not "010."

Service startup time

ExpressCluster services might take a while to start up, depending on the wait processing at startup.

- clusterpro_evt
  Servers other than the master server wait up to two minutes for configuration data to be downloaded from the master server. Downloading usually finishes within several seconds if the master server is already operating. The master server does not have this wait process.
- clusterpro_trn
  There is no wait process. This process usually finishes within several seconds.
- clusterpro
  Although there is no wait process, ExpressCluster might take several tens of seconds to start up. This process usually finishes within several seconds.
- clusterpro_webmgr
  There is no wait process. This process usually finishes within several seconds.
- clusterpro_alertsync
  There is no wait process. This process usually finishes within several seconds.

In addition, the system waits for cluster activation synchronization after the ExpressCluster daemon is started. By default, this wait time is five minutes.

For details, see Chapter 9, "The system maintenance information" in the *Reference Guide*.

Chapter 6 Upgrading ExpressCluster

This chapter provides information on how to upgrade ExpressCluster.
This chapter covers:

How to update ExpressCluster

How to update from X2.1 to X3.0

Install the ExpressCluster Server package as root user.

1. Get the configuration information by the online Builderor the clpcfctrl command.
2. Stop the cluster by the WebManager or the clpcl command.
3. Disable the services by running the **`svcadm disable`** ***name*** in the following order. Specify one of the following services to be disabled in *name*.

   clusterpro_alertsync

   clusterpro_webmgr

   clusterpro

   clusterpro_trn

   clusterpro_evt

4. Confirm that ExpressCluster services are not running, and then uninstall the package file by executing the pkgrm command.

   ```
   pkgrm NECclusterpro
   ```

5. Mount the installation CD-ROM media.
6. Install the package file by executing the pkgadd command.
   The package file is under /Solaris/3.0/jp/server in the CD-ROM. The pkg for installation is different depending on architecture. For architecture, there are i686 and x86_64. Select architecture according to the installation destination environment.

   ```
   pkgadd -d NECclusterpro-<version>-<architecture>.pkg
   ```

   The installation directory of ExpressCluster is /opt/nec/clusterpro. Be careful because ExpressCluster cannot be uninstalled if this directory is changed.
7. Unmount the CD-ROM media after installation and remove it.
8. Repeat the steps 3-7 on all the servers.
9. Register the license. For details on registering license, see “Chapter 4 Registering the license” in the Installation and Configuration Guide.
10. Connect the WebManager to one of the servers of the cluster.
11. The initial construction dialog box is displayed on the connected WebManager, and then read the cluster configuration information gotten in the step 1 by selecting **Import cluster configuration file**.
12. Confirm that all servers of the cluster are started, and then apply the configuration data. For details on how to operate the online Builder, see the *Reference Guide*. **Restart Manager** is executed automatically.
13. Reboot all the servers of the cluster.

Appendix

- Appendix A Glossary
- Appendix B Index

Appendix A Glossary

Interconnect	A dedicated communication path for server-to-server communication in a cluster. (Related terms: Private LAN, Public LAN)
Virtual IP address	IP address used to configure a remote cluster.
Management client	Any machine that uses the WebManager to access and manage a cluster system.
Startup attribute	A failover group attribute that determines whether a failover group should be started up automatically or manually when a cluster is started.
Shared disk	A disk that multiple servers can access.
Shared disk type cluster	A cluster system that uses one or more shared disks.
Switchable partition	A disk partition connected to multiple computers and is switchable among computers. (Related terms: Disk heartbeat partition)
Cluster system	Multiple computers are connected via a LAN (or other network) and behave as if it were a single system.
Cluster shutdown	To shut down an entire cluster system (all servers that configure a cluster system).
Active server	A server that is running for an application set. (Related term: Standby server)
Secondary server	A destination server where a failover group fails over to during normal operations. (Related term: Primary server)
Standby server	A server that is not an active server. (Related term: Active server)
Disk heartbeat partition	A partition used for heartbeat communication in a shared disk type cluster.
Data partition	A local disk that can be used as a shared disk for switchable partition.
Network partition	All heartbeat is lost and the network between servers is partitioned. (Related terms: Interconnect, Heartbeat)
Node	A server that is part of a cluster in a cluster system. In networking terminology, it refers to devices, including computers and routers, that can transmit, receive, or process signals.
Heartbeat	Signals that servers in a cluster send to each other to detect a failure in a cluster. (Related terms: Interconnect, Network partition)

Public LAN	A communication channel between clients and servers. (Related terms: Interconnect, Private LAN)
Failover	The process of a standby server taking over the group of resources that the active server previously was handling due to error detection.
Failback	A process of returning an application back to an active server after an application fails over to another server.
Failover group	A group of cluster resources and attributes required to execute an application.
Moving failover group	Moving an application from an active server to a standby server by a user.
Failover policy	A priority list of servers that a group can fail over to.
Private LAN	LAN in which only servers configured in a clustered system are connected. (Related terms: Interconnect, Public LAN)
Primary (server)	A server that is the main server for a failover group. (Related term: Secondary server)
Floating IP address	Clients can transparently switch one server from another when a failover occurs. Any unassigned IP address that has the same network address that a cluster server belongs to can be used as a floating address.
Master server	The server displayed on top of the **Master Server** in **Cluster Properties** in the Builder.

Appendix B Index