ExpressCluster® X 1.0 *for Windows*

Getting Started Guide

6/22/2007
Third Edition

Revision History

Edition	Revised Date	Description
First	09/08/2006	New manual
Second	12/28/2006	Reflected the logo change Modified errors and document formats
Third	06/22/2007	Added descriptions for new resources added to the version 9.03 Updated resource note

Disclaimer

Trademark Information

Table of Contents

Preface

Who Should Use This Guide

The *ExpressCluster X Getting Started Guide* consists of two sections. Section I is intended for first-time users of the ExpressCluster. This section covers topics such as product overview of the ExpressCluster and its basic usage.
Section II is intended for users before installing ExpressCluster and those who update it after the installation. This section covers topics such as 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.
Chapter 2 **Using ExpressCluster**
Provides instructions on how to use ExpressCluster 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.

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 is 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*.

Alert Service Administrator's Guide
This guide is intended for system administrators who install ExpressCluster X Alert Service, operate and maintain a cluster system. The guide provides instructions for installing a cluster system that uses ExpressCluster X Alert Service.

Conventions

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]`
Monospace (courier)	Indicates path names, commands, system output (message, prompt, etc), directory, file names, functions and parameters.	`c:\Program files\CLUSTERPRO`
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.	`clpstat -s [-h host_name]`

Contacting NEC

For the latest product information, visit our website below:

http://www.ace.comp.nec.co.jp/CLUSTERPRO/clp/global-link.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:

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 had.

Introducing a cluster system allows you to minimize the period during which your system stops (down time) or to improve availability by load distribution.

As the word "cluster" represents, 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 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. The high availability cluster is available in the shared disk type and the mirror disk type.

- Load Distribution Cluster

  This is a cluster configuration where requests from clients are allocated to each of the nodes according to appropriate load distribution rules. This cluster configuration aims for high scalability. Generally, data cannot be passed. 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 the computation amount is huge and a single operation is performed with a super computer. CPUs of all nodes are used to perform a single operation.

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 nodes.

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 mirror disk type. The explanation for each type is provided below.

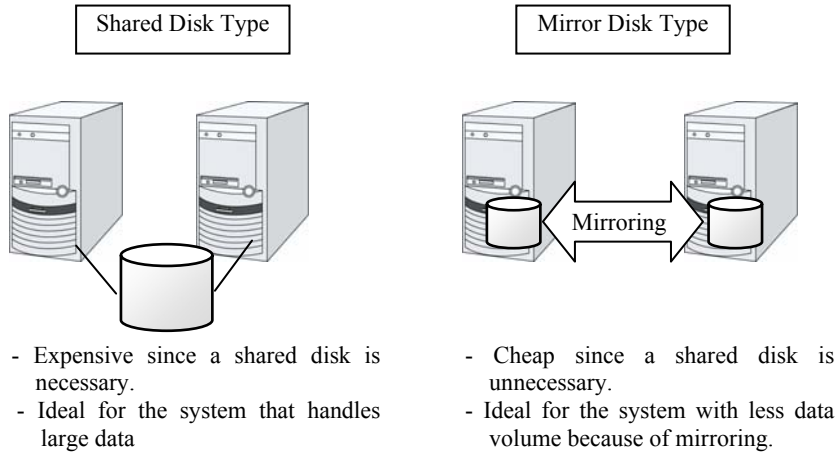


Figure 1-1: HA cluster configuration

Shared disk type

Data must be inherited from one server to another in cluster systems. A cluster typology where data is stored in an external disk (shared disk) accessible from two or more servers and inherited among them through the disk (for example, FibreChannel disk array device of SAN connection) is called shared disk type.

If a failure occurs on a server where applications are running (active server), the cluster system automatically detects the failure and starts applications in a stand-by server to take over operations. This mechanism is called failover. Operations to be inherited 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 of another network but not of an IP address given to a server 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.

If a failover occurs because an active server is down, data on the shared disk is inherited to a stand-by server without necessary application-ending processing being completed. For this reason, it is required to check logic of data on a stand-by server. Usually this processing is the same as the one performed when a non-clustered system is rebooted after its shutdown. 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.

After a failure occurs, 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. It is not necessary to failback a group to the original server when continuity of operations is important. If it is essentially required to perform the operations on the original server, move the group.

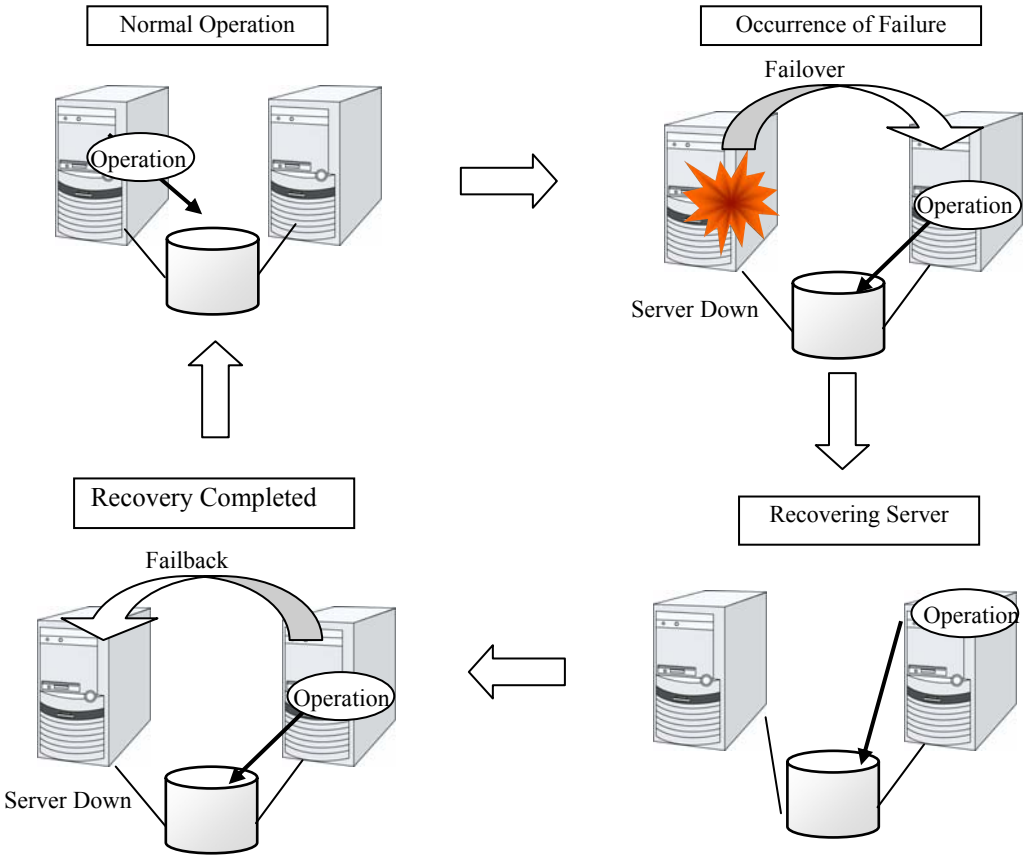


Figure 1-2: From occurrence of a failure to recovery

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, after finishing the recovery of the original node, stop the operations and start them again on the original node. Returning a failover group to the original server is called failback.

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 node of the cluster serving as both active and stand-by servers is referred to as multi-directional stand-by.

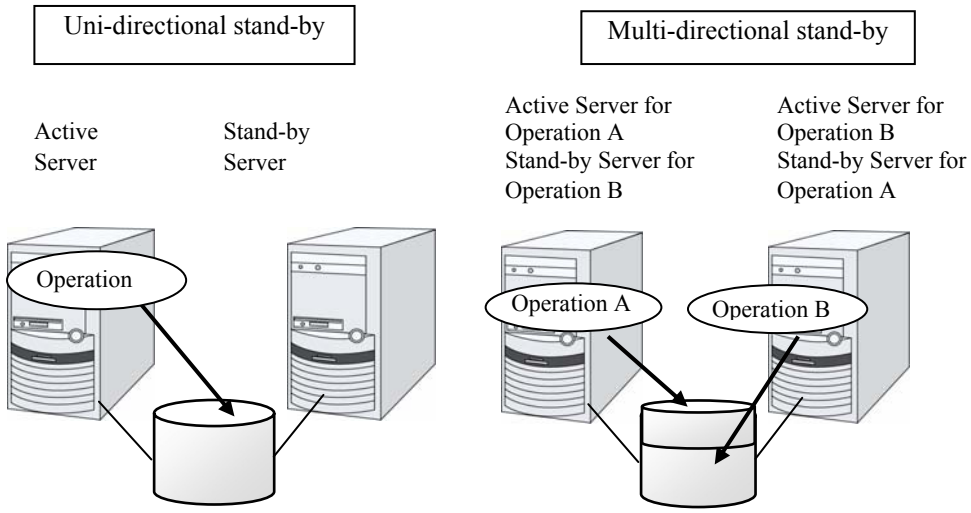


Figure 1-3: HA cluster topology

Mirror disk 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 mirror disk type cluster system provides the same functions as the shared disk type with smaller cost through mirroring of server disks.

The mirror disk 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 writes data in the local disk and sends the written data to the stand-by server via the interconnect. Interconnect is a cable connecting servers. It is used to monitor whether the server is activated or not 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.

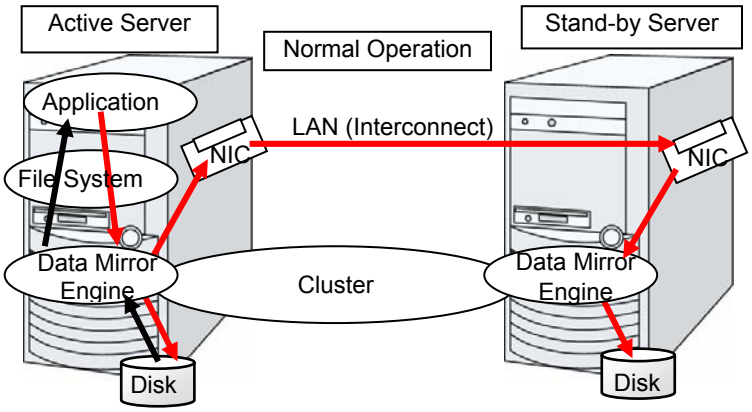


Figure 1-4: Data mirror mechanism

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 data of the stand-by server as snapshot backup by simply separating the server from the cluster.

System configuration

In a shared disk-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.

In the mirror disk type cluster, a data disk on the cluster server is mirrored via the network. When an error occurs on a server, the applications are taken over using the mirror data on the stand-by server. Data is mirrored for every I/O. Therefore, the mirror disk type cluster appears the same as the shared disk viewing from a high level application.

The following the shared disk type cluster configuration.

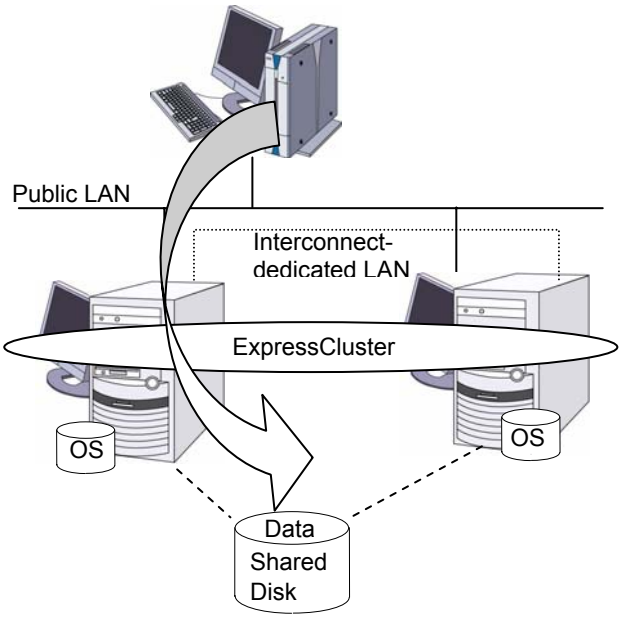


Figure 1-5: System configuration

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.

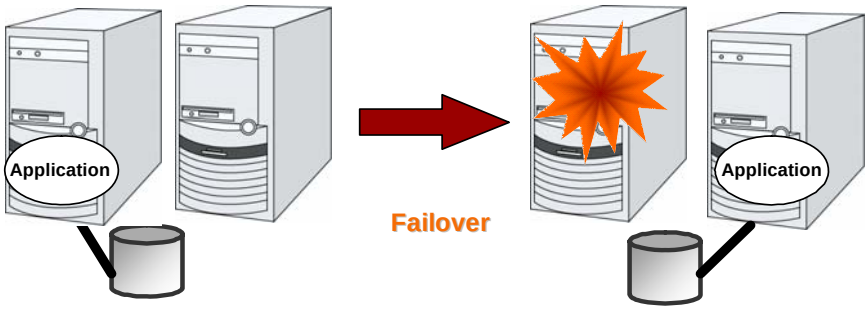
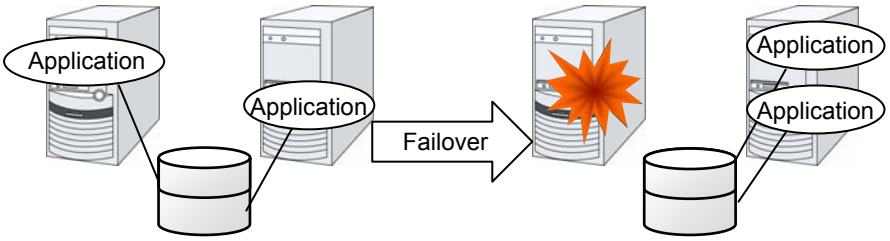


Figure 1-6: Uni-directional standby cluster system

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. These applications are operated on their own. When a failover occurs, the same applications are activated on one server. Therefore, the applications that can be activated by this operation need to be used. 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.

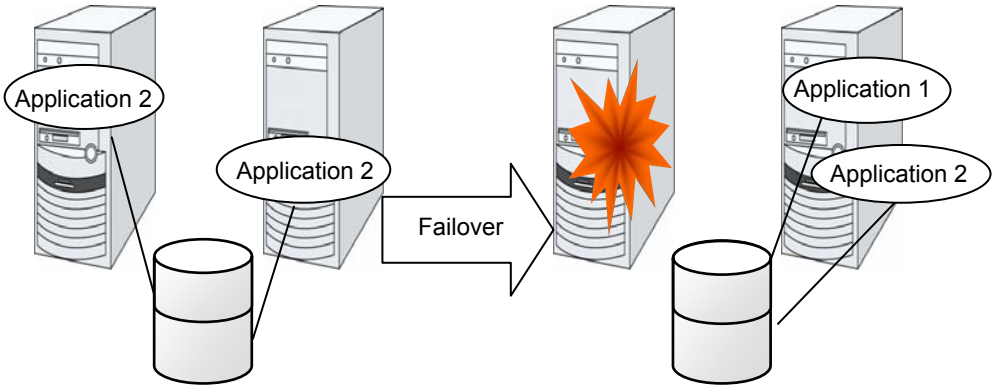


- The applications in the diagram are the same applications.
- Multiple application instances are run on a single server after failover.

Figure 1-7: 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 operate as standby servers. When a failover occurs, two or more applications are activated on one server. Therefore, these applications need to be able to coexist. You can build a load distribution system per application unit basis.



- Application 1 and application 2 are different applications.

Figure 1-8: Different application – multi directional standby cluster system

N Server to M Operation Configuration

The configuration can be expanded with more nodes by applying the configurations introduced thus far. In an N server to M operation 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, the stand-by server does not operate anything, so one of the two servers functions as a stand-by server. However, in an N server to M operation configuration, only one of the four servers functions as a stand-by server. Performance deterioration is not anticipated if an error occurs only on one server.

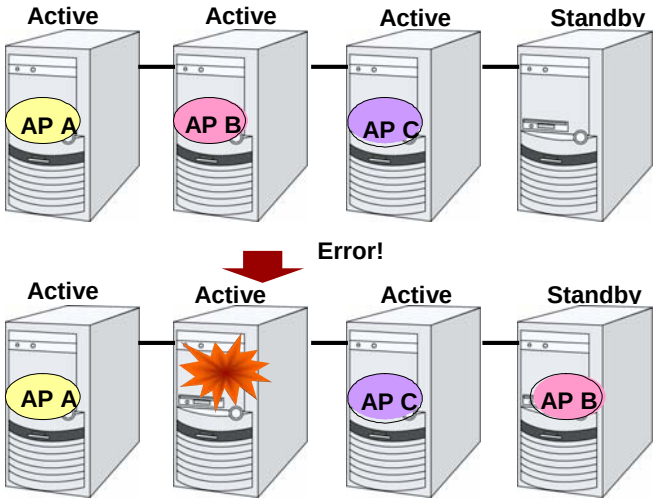


Figure 1-9: N Server to M Operation configuration

Error detection mechanism

Cluster software executes failover (for example, passing operations) when a failure that can affect 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, the heartbeat is used to monitor whether the server is active or not.

Some cluster software programs use heartbeat not only for checking if 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 used to detect resource failures if the target is a physical device. For monitoring applications, trying to service ports within the range not affecting operation is a way of detecting an error in addition to monitoring if application processes are activated.

Shared disk lock

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 destroyed. The failover cluster system locks the disk device to prevent multiple servers from mounting a file system, simultaneously caused by a network partition problem explained next.

Network partition (split-brain-syndrome)

When all interconnects between servers are disconnected, it is not possible to tell if a server is down, only by monitoring if it is activated by a heartbeat. In this status, if a failover is performed and multiple servers mount a file system simultaneously considering the server has been shut down, data on the shared disk may be corrupted.

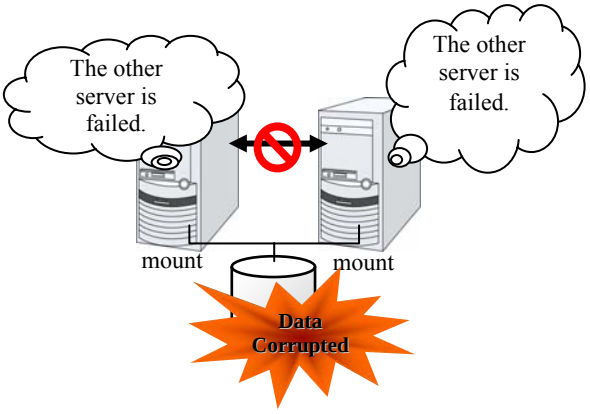


Figure 1-10: Network partition

The problem explained in the section above is referred to as “network partition” or “Split Brain Syndrome.” To resolve this problem, the failover cluster system is equipped with various mechanisms to ensure shared disk lock at the time when all interconnects are disconnected.

Inheriting 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 inherit these resources are described below.

Inheriting data

In the shared disk type cluster, data to be passed from a server to another in a cluster system is stored in a partition in a shared disk. This means inheriting data is re-mounting the file system of files that the application uses from 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 inherits data.

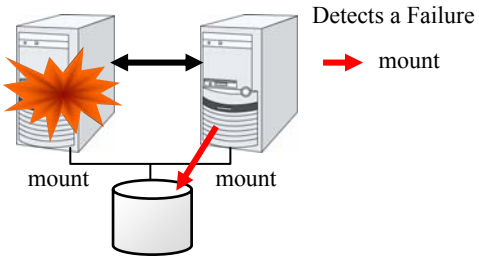


Figure 1-11: Inheriting data

The diagram above (Figure 1-11) may look simple. Consider the following issues in designing and creating a cluster system.

One issue to consider is recovery time for a file system or database. A file to be inherited may have been used by another server or to be updated just before the failure occurred. For this reason, a cluster system may need to do consistency checks to data it is moving on some file systems, as well as it may need to rollback data for some database systems. These checks are not cluster system-specific, but required in many recovery processes, including when you reboot a single server that has been shut down due to a power failure. If this recovery takes a long time, 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 data into the shared disk, usually the data is written through a file system. However, even though the application has written data – but the file system only stores it on a disk cache and does not write into the shared disk – the data on the disk cache will not be inherited to a stand-by server when an active server shuts down. For this reason, it is required to write important data that needs to be inherited to a stand-by server into a disk, by using a function such as synchronous writing. This is same as preventing the data becoming volatile when a single server shuts down. Namely, only the data registered in the shared disk is inherited to a stand-by server, and data on a memory disk such as a disk cache is not inherited. The cluster system needs to be configured considering these issues.

Inheriting IP addresses

When a failover occurs, it does not have to be concerned which server is running operations by inheriting IP addresses. The cluster software inherits the IP addresses for this purpose.

Inheriting applications

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

For example, when the database instance is failed over, the database that is started in the stand-by server can not continue the exact processes and transactions that have been running in the failed server, and roll-back of transaction is performed in the same as restarting the database after it was down. It is required to connect to the database again from the client. 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
    - Pass applications

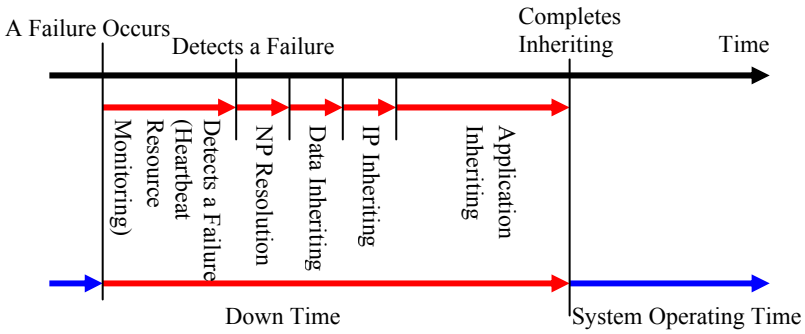


Figure 1-12: Failover time chart

Cluster software is required to complete each task quickly and reliably (see Figure 1-12.) 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, 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.

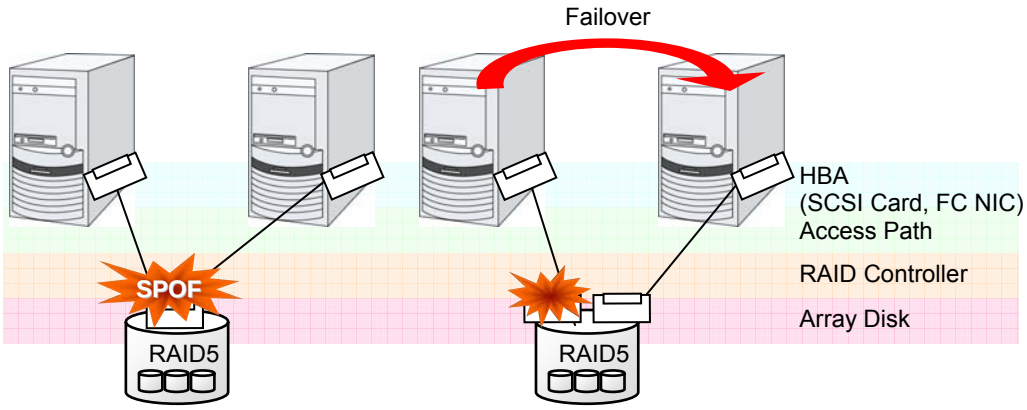
A cluster system can improve availability but failover will take a few minutes for switching systems. That means time for failover is a factor that reduces availability. Solutions for the following three, which are likely to become SPOF, will be discussed hereafter although technical issues that improve 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.



- HBA stands for Host Bus Adapter. This is an adapter of the server not of the shared disk.

Figure 1-13: 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:

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

In a system with frequent data viewing and a relatively small volume of data, choosing the failover cluster of data mirror type is effective 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.

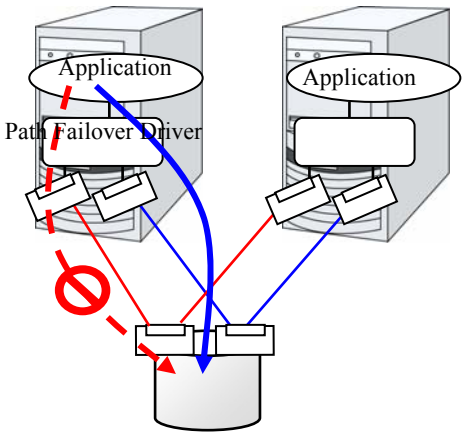


Figure 1-14: Path failover driver

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.

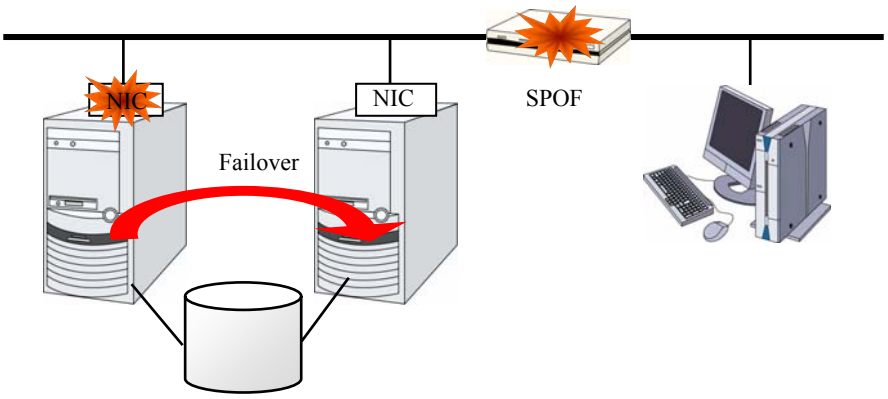


Figure 1-15: Example of router becoming SPOF

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.

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 quickly and take appropriate action 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 enables the HA cluster system.

ExpressCluster modules

ExpressCluster consists of following three modules:

- ExpressCluster Server

  A core component of ExpressCluster. Install this to the server machines that constitute the cluster system. The ExpressCluster Server includes all high availability functions of ExpressCluster. The server functions of the WebManager and Builder are 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 browsers as its user interface. There are offline and online versions. In the offline version, you can install and use the Builder on the machine where you use it, in addition to the ExpressCluster Server. In the online version, use the Builder included in the ExpressCluster Server through the WebManager. 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 Linux server that constitutes a cluster. This function of the WebManager or Builder does not need to be installed separately, because it is included in the ExpressCluster Server. However, to use the Builder in the environment where you cannot access the ExpressCluster Server, it is required to install the offline version Builder on the management PC. The WebManager and the Builder can be used from the Web browser on the management PC, so they can be used from the Web browser on the servers that constitute a cluster.

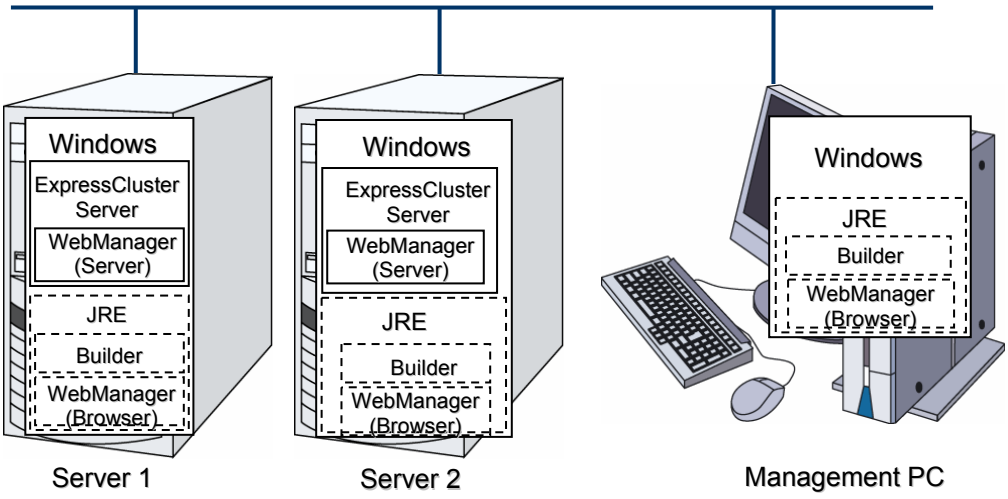


Figure 2-1: Software configuration of ExpressCluster

Note:
JRE stands for Java Runtime Environment.

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.

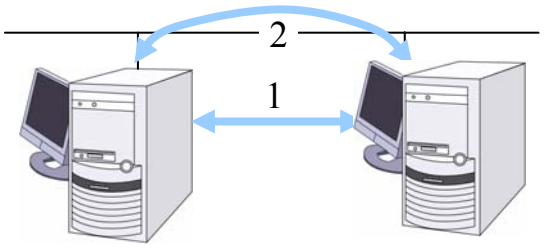
ExpressCluster regularly checks whether other servers are properly working in the cluster system. This way of verification is called “heartbeat communication.” The heartbeat communication uses the following communication paths:

- Interconnect-dedicated LAN

  LAN dedicated to communication between the cluster servers. This is used to exchange information between the servers as well as to perform heartbeat communication.

- Public LAN

  This is used as a path to be used for the communicating with a client. This is used for exchanging data between the servers as well as for a backup interconnects.



1. Interconnect-dedicated LAN
2 Public LAN

What is application monitoring?

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

- Monitoring applications and/or protocols to see if they are stalled or failed by using the monitoring option.

  In addition to the basic monitoring of successful startup and existence of applications, you can even monitor stall and failure in applications including specific databases (such as Oracle, DB2) ,protocols (such as FTP, HTTP) and / or application servers (such as Websphere, Weblogic) by introducing optional monitoring products of ExpressCluster. For the details, see Chapter 7, “Monitor resource details” in Section II of the *Reference Guide*.

- Monitoring activation status of applications

  An error can be detected by starting up an application by using an application-starting resource (called application resource and service resource) of ExpressCluster and regularly checking whether the process is active or not by using application-monitoring resource (called application monitor resource and service 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.
  - An internal application error (for example, application stalling and 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 that 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.

Monitorable and non-monitorable errors

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

Detectable and non-detectable errors by server monitoring

Monitoring conditions: 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)
  - Stop error
- 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 application with errors, continuous resource errors, 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 HBA 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[1]. 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 application resource, and monitoring application using the application monitor resource.

[1] Stalling and error results can be monitored for the database applications (such as Oracle, DB2), the protocols (such as FTP, HTTP) and application servers (such as Websphere, Weblogic) that are handled by a monitoring option.

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 inheriting operations, so processing such as emergency shutdown is performed.
The following are the network partition resolution methods:

- COM method
- ping method
- Shared disk method
- COM + shared disk method
- Ping + shared disk method
- Majority method
- Not solving the network partition

Related Information:

For the details on the network partition resolution method, see Chapter 9, "Details on network partition resolution resources" in Section II of the *Reference Guide*.

Failover mechanism

When the stop of heartbeat is detected from other servers, ExpressCluster determines whether it is an error in a server or a network partition before starting a failover. 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 time-chart below:

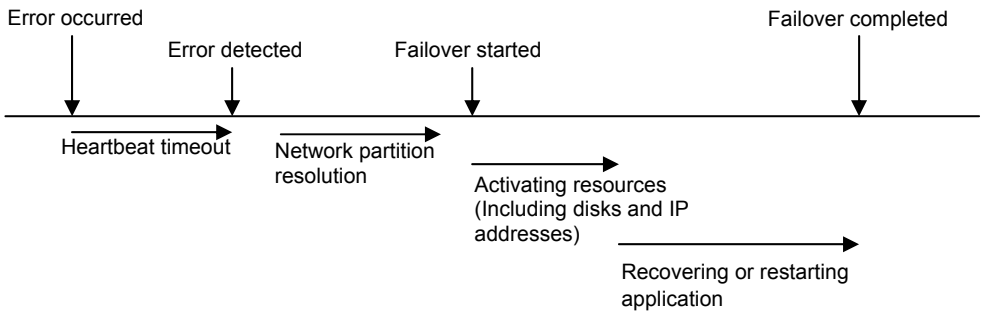


Figure 2-2: Failover time chart

- 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 delay caused by application load. (The default value is 30 seconds.)

- Network partition resolution
  - This is the time to check whether stop of heartbeat (heartbeat timeout) detected from the other server is due to a network partition or an error in the other server.
  - When the shared disk method is specified as the network partition method, 30 to 60 seconds of wait time is set by default because the time considering the disk I/O delay needs to be set. The required time changes in accordance with the time to access to a cluster partition and the heartbeat timeout value. When other method is specified, confirmation completes immediately.

- Activating 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, see the *Installation and Configuration Guide*.

- Recovering and restarting applications
  - The startup time of the application to be used in operation. The data recovery time such as a roll-back or roll-forward of the database is included.
  - 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.

Hardware configuration of the shared disk type cluster configured by ExpressCluster

The hardware configuration of the shared disk type cluster 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.

Sample of cluster environment when a shared disk is used:

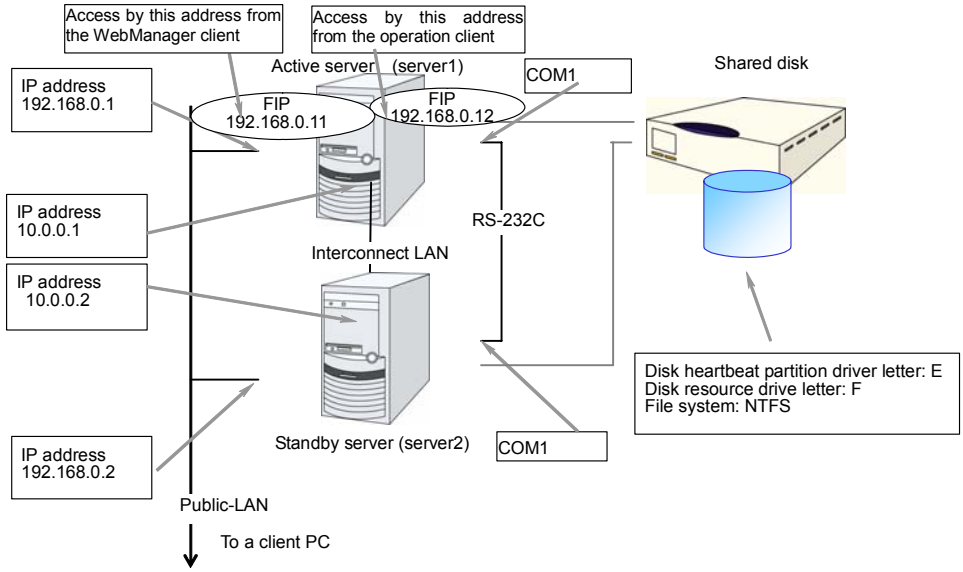


Figure 2-3: Sample of a cluster environment when a shared disk is used

Hardware configuration of the mirror disk type cluster configured by ExpressCluster

The mirror disk type cluster is an alternative to the shared disk device, by mirroring the partition on the server disks. This is good for the systems that are smaller-scale and lower-budget, compared to the shared disk type cluster.

Note:
To use a mirror disk, it is a requirement to purchase the Replicator option.

A network for copying mirror disk data is required, but normally interconnect (NIC for ExpressCluster internal communication) is used for this purpose.

The hardware configuration of the data mirror type cluster configured by ExpressCluster is described below.

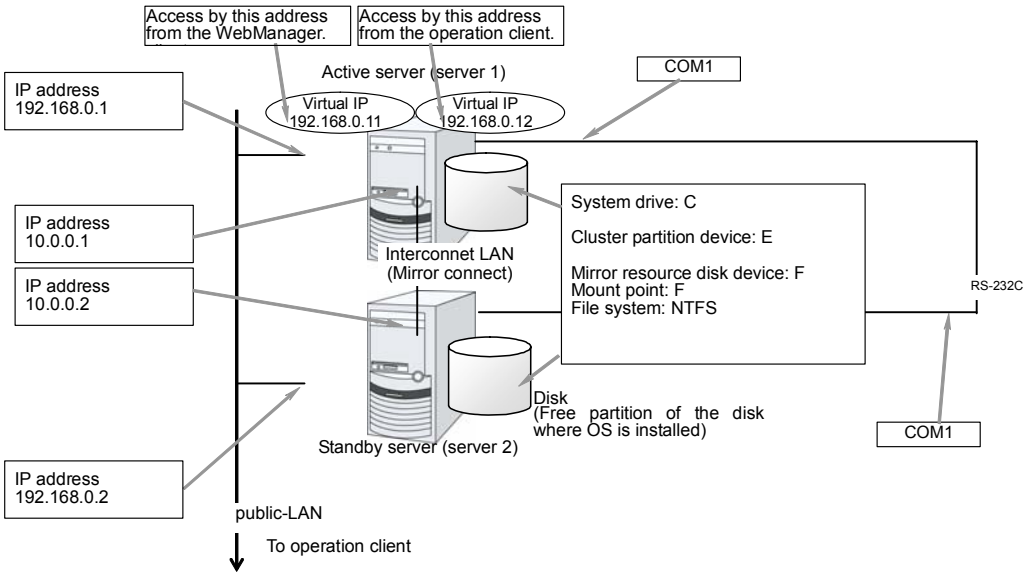


Figure 2-4: Sample of a cluster environment when mirror disks are used (when allocating cluster partition and data partition on the disk where OS is installed)

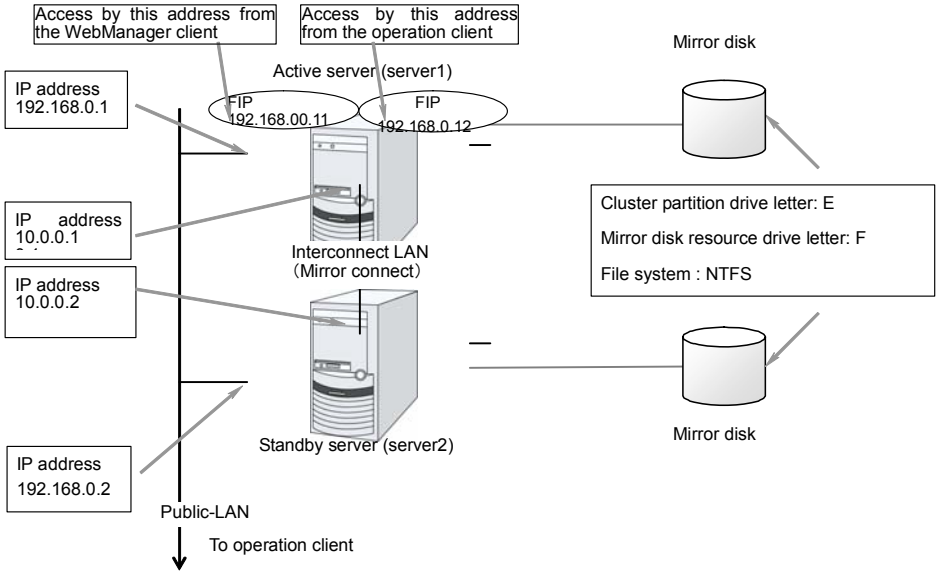


Figure 2-5: Sample of cluster environment when mirror disks are used (in case of preparing disks for cluster partition and data partition)

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.” The resources that perform monitoring and those to be monitored are classified into two groups and managed. 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.

Note:
For the details of each resource, see Section II of the *Reference Guide*.

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.

Network partition resolution resources

Resources used for solving the network partition are shown below:

- **COM network partition resolution resource**
  This is a network partition resolution resource by the COM method.
- **DISK network partition resolution resource**
  This is a network partition resolution resource by the DISK method and can be used only for the shared disk configuration.
- **PING network partition resolution resource**
  This is a network partition resolution resource by the PING method.
- **Majority network partition resolution resource**
  This is a network partition resolution resource by the majority method.

Group resources

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

- **Application resource (appli)**
  Provides a mechanism for starting and stopping an application (including user creation application.)
- **Floating IP resource (fip)**
  Provides a virtual IP address. A client can access a virtual IP address the same way as accessing a regular IP address.
- **Disk resource (sd)**
  Provides a function to control access to a specific partition on the shared disk. This can be used only when the shared disk device is connected.
- **Mirror disk resource (md)**
  Provides a function to perform mirroring a specific partition on the local disk and control access to it. It can be used only on a mirror disk configuration.
- **Registry synchronization resource (regsync)**
  Provides a mechanism to synchronize specific registries of more than two servers, to set the applications and services in the same way among the servers that constitute a cluster.
- **Script resource (script)**
  Provides a mechanism for starting and stopping a script (BAT) such as a user creation script.
- **Service resource (service)**
  Provides a mechanism for starting and stopping a service such as database and Web.
- **Print spooler resource (spool)**
  Provides a mechanism for failing over print spoolers.
- **Virtual computer name resource (vcom)**
  Provides a virtual computer name. This can be accessed from a client in the same way as a general computer name.
- **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.
- **CIFS resource (cifs)**
  Provides a function to disclose and share folders on the shared disk and mirror disks.
- **NAS resource (nas)**
  Provides a function to mount the shared folders on the file servers as network drives.

Note
To use a mirror disk resource, the ExpressCluster X Replicator license is required.

Monitor resources

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

- **Application monitor resource (appliw)**
  Provides a monitoring mechanism to check whether a process started by application resource is active or not.
- **Disk RW monitor resource (diskw)**
  Provides a monitoring mechanism for the file system and function to perform a failover by resetting the hardware or an intentional stop error at the time of file system I/O stalling. This can be used for monitoring the file system of the shared disk.
- **Floating IP monitor resource (fipw)**
  Provides a monitoring mechanism of the IP address started by floating IP resource.
- **IP monitor resource (ipw)**
  Provides a mechanism for monitoring the network communication.
- **Mirror disk monitor resource (mdw)**
  Provides a monitoring mechanism of the mirroring disks.
- **Mirror connect monitor resource (mdnw)**
  Provides a monitoring mechanism of the mirror connect.
- **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.
- **Registry synchronization monitor resource (regsyncw)**
  Provides a monitoring mechanism of the synchronization process by a registry synchronization resource.
- **Disk TUR monitor resource (sdw)**
  Provides a mechanism to monitor the operation of access path to the shared disk by the TestUnitReady command of SCSI. This can be used for the shared disk of FibreChannel.
- **Service monitor resource (servicew)**
  Provides a monitoring mechanism to check whether a process started by a service resource is active or not.
- **Print spooler monitor resource (spoolw)**
  Provides a monitoring mechanism of the print spooler started by a print spooler resource.
- **Virtual computer name monitor resource (vcomw)**
  Provides a monitoring mechanism of the virtual computer started by a virtual computer name resource.
- **Virtual IP monitor resource (vipw)**
  Provides a monitoring mechanism of the IP address started by a virtual IP resource.
- **CIFS resource (cifsw)**
  Provides a monitoring mechanism of the shared folder disclosed by a CIFS resource.
- **NAS resource (nasw)**
  Provides a monitoring mechanism of the network drive mounted by a NAS resource.
- **DB2 monitor resource (db2w)**
  Provides a monitoring mechanism for the IBM DB2 database.

- **ODBC monitor resource (odbcw)**
  Provides a monitoring mechanism for the database that can be accessed by ODBC.
- **Oracle monitor resource (oraclew)**
  Provides a monitoring mechanism for the Oracle database.
- **PostgreSQL monitor resource (psqlw)**
  Provides a monitoring mechanism for the PostgreSQL database.
- **SQL Server monitor resource (sqlserverw)**
  Provides a monitoring mechanism for the SQL Server database.
- **FTP monitor resource (ftpw)**
  Provides a monitoring mechanism for the FTP server.
- **HTTP monitor resource (httpw)**
  Provides a monitoring mechanism for the HTTP server.
- **IMAP4 monitor resource (imap4w)**
  Provides a monitoring mechanism for the IMAP server.
- **POP3 monitor resource (pop3w)**
  Provides a monitoring mechanism for the POP server.
- **SMTP monitor resource (smtpw)**
  Provides a monitoring mechanism for the SMTP server.
- **Tuxedo monitor resource (tuxw)**
  Provides a monitoring mechanism for the Tuxedo application server.
- **Weblogic monitor resource (wlsw)**
  Provides a monitoring mechanism for the Weblogic application server.
- **Websphere monitor resource (wasw)**
  Provides a monitoring mechanism for the Websphere application server.

Note:
To use the DB2 monitor resource, ODBC monitor resource, Oracle monitor resource, PostgreSQL monitor resource, and SQL Server monitor resource, the ExpressCluster X Database Agent license is required.
To use the FTP monitor resource, HTTP monitor resource, IMAP4 monitor resource, POP3 monitor resource and SMTP monitor resource, the ExpressCluster X Internet Server Agent license is required.
To use the Tuxedo monitor resource, Weblogic monitor resource and Websphere monitor resource, the ExpressCluster X Application Server Agent license is required.

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, “Designing 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*

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 3 Installation requirements for ExpressCluster

This chapter provides information on system requirements for ExpressCluster.

This chapter covers:

System requirements for hardware

ExpressCluster operates on the following server architectures:

- IA-32
- X86-64

General server requirements

Required specifications for the 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
- Mirror disk or empty partition for mirror (required when the Replicator is used)
- CD-ROM drive

  To use the Builder off-line, one of the following is required to send the cluster configuration data. (This is not required for using the Builder online.)
- Removable media (for example, floppy disk drive or USB memory)
- The method to share the file with the server machine on which the ExpressCluster Server has been installed.

System requirements for the ExpressCluster Server

Supported operating systems

ExpressCluster Server only runs on the operating systems listed below.

IA-32 version

OS	Replicator support
Microsoft Windows Server 2003: Standard Edition SP1 or later	Yes
Microsoft Windows Server 2003: Enterprise Edition SP1 or later	Yes
Microsoft Windows Server 2003: Standard Edition R2	Yes
Microsoft Windows Server 2003: Enterprise Edition R2	Yes

EM64T version

OS	Replicator support
Microsoft Windows Server 2003: Standard x64 Edition SP1or later	Yes
Microsoft Windows Server 2003: Enterprise x64 Edition SP1 or later	Yes
Microsoft Windows Server 2003: Standard x64 Edition R2	Yes
Microsoft Windows Server 2003: Enterprise x64 Edition R2	Yes

Required memory and disk size

Following is the memory size and disk size required for mirror disk resource.

When changing to asynchronous method or changing the queue size, it is required to add more memory of the size specified at the time of configuration. Memory size increases as disk load increases because memory is used corresponding to mirror disk I/O.

	Required memory size		Required disk size	
	User mode	Kernel mode	Right after installation	Max. during operation
IA-32	35MB	32MB + 4MB x number of mirror resources	40MB	600MB
x86-64	40MB	32MB + 4MB x number of mirror resources	18MB	600MB

System requirements for the Builder

Supported operating systems and browsers

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

Operating system	Browser	Language
Microsoft Windows XP SP2	IE6 SP2	English/Japanese
Microsoft Windows Vista™	IE7	English/Japanese
Microsoft Windows Server 2003 SP1 or later	IE6 SP1	English/Japanese
Microsoft Windows Server 2003 R2	IE6 SP1	English/Japanese

Note:
The Builder cannot be operated on 64-bit and x86_64 machines. When constructing or changing a cluster system, prepare the 32-bit machine.

Java runtime environment

Required:

Sun Microsystems, Java(TM) Runtime Environment, Version 5.0 Update 6 (1.5.0_06) or later.

Required memory size and disk size

Required memory size: 32MB or more

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

Supported ExpressCluster versions

Builder version	ExpressCluster Server version
1.0	1.0

System requirements for the WebManager

Supported operating systems and browsers

Currently the following operating systems and browsers are supported:

Operating system	Browser	Language
Windows® XP SP2	IE6 SP2	English/Japanese
Microsoft Windows Vista™	IE7	English/Japanese
Microsoft Windows Server 2003 SP1 or later	IE6 SP1	English/Japanese
Microsoft Windows Server 2003 R2	IE6 SP1	English/Japanese

Note:
The WebManager cannot be operated on 64-bit and x86_64 machines. P When constructing or changing a cluster system, prepare the 32-bit machine. For the latest information, refer to ExpressCluster website, *http://www.ace.comp.nec.co.jp/CLUSTERPRO/clp/global-link.html*.

Java runtime environment

Required:

Sun Microsystems, Java(TM) Runtime Environment, Version 5.0 Update 6 (1.5.0_06) or later

Required memory size and disk size

Required memory size: 40MB or more

Required disk size: 300KB or more (excluding the size required for Java runtime environment)

Chapter 4 Latest version information

This chapter provides the latest information on ExpressCluster. The latest information on the upgraded and improved functions is described in details.

This chapter covers:

The latest version

As of June 2007, the latest version of ExpressCluster is 9.03.

Check the version of ExpressCluster by using WebManager.

You can see the detailed version of a server by selecting the icon for the server on the tree view of the WebManager.

By applying CPRO-XW010-02, versions 9.02 or earler can be updated to 9.03. For more information about the steps for updating and improved failure funtions, see the update manual.

Function upgrade information

Upgrade has been performed on the following minor versions.

Number	Version (in detail)	Upgraded section
1	9.03	The following group resources and monitor resources have been added. Group resources: cifs, nas Monitor resources: cifsw, nasw, tuxw, wlsw, wasw
2	9.03	Automatic registration of virtual computer name with DNS is available.
3	9.03	The upper limits of the communication band that mirror disks use for mirror connection communication in the asynchronous mode and the size of the temporary files to be stored in the history file storage folder are configurable .
4	9.03	The clpvolsz command to configure the size of the data partition that is mirrored by mirror resources has been added.
5	9.03	Staring and stopping of group resources from the batch file by using the clprsc command is available.
6	9.03	The command “clprsc” is added to start/stop group resources.
7	9.03	WebManager and Builder can work together with Windows Vista + IE7 and Java(TM) Runtime Environment Version 6.0.
8	9.03	Supports IPv6/IPv4 coexistence environment. (Note that IPv4 is necessary for the communications between cluster servers)
9	9.03	EXPRESSCLUSTER X Application Server Agent 1.0 for Windows is added.

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.

Supported operating systems for the Builder and WebManager

- To operate the Builder and WebManager on a x86-64 machine, Java runtime for 32-bit is required.

Hardware requirements for mirror disks

- Dynamic disks cannot be used. Use basic disks.
- GUID partition table (GPT) disks cannot be used.
- Disks to be used as a mirror disk resource do not support a Linux and and/or LVM stripe set, volume set, mirroring, and stripe set with parity.
- There are no specific limitations on locating partitions for mirroring, but the data partition sizes need to be perfectly matched with one another on a byte basis. A cluster partition also requires space of 17 MB or larger.
- When making data partitions as logical partitions on the extended partition, make sure to select the logical partition for both servers. Even when the same size is specified on both primary partition and logical partition, their actual sizes may different from each other.
- It is recommended to create a cluster partition and a data partition on different disks for the load distribution. (There are not any problems to create them on the same disk, but the writing performance will slightly decline, in case of asynchronous mirroring or in a state that mirroring is suspended.)
- Use the same type of disks for reserving data partitions that perform mirroring by mirror resources on both of the servers.

Example

Combination	server1	server2
OK	SCSI	SCSI
OK	IDE	IDE
NG	IDE	SCSI

- Partition size reserved by **Disk Management** is aligned by the number of blocks (units) per disk cylinder. For this reason, if disk geometries used as disks for mirroring differ between servers, the data partition sizes cannot be matched perfectly. To avoid this problem, it is recommended to use the same hardware configurations including RAID configurations for the disks that reserve data partitions on server1 and server2.
- When you cannnot synchronize the disk size or geometry on the both servers, make sure to check the exact size of data partitions by using the clpvolsz command. If they do not match, make the larger partition small by using the clpvolsz command.
- When RAID-disk is mirrored, it is recommended to use writeback mode because writing performance decreases a lot when the disk array controller cache is set to write-thru mode. However, when writeback mode is used, it is necessary to use disk array controller with battery installed or use with UPS.
- A partition with the OS page file cannot be mirrored.

Hardware requirements for shared disks

- Dynamic disks cannot be used. Use basic disks.
- Software RAID (stripe set, mirror set, stripe set with parity) and volume set cannot be used.
- GPT disks cannot be used.

NIC link up/down monitor resource

Some NIC boards and drivers do not support required DeviceIoControl. You cannot use this monitor resource in such a case. To use this monitor resource, use the trial license and check the operation in advance.

Write function of the mirror resource

There are 2 types of disk mirroring of mirror resources: synchronous mirroring and asynchronous mirroring.

In synchronous mirroring, data is written in the disks of both servers for every request to write data in the data partition to be mirrored and its completion is waited. Data is written in each of the servers along with this, but it is written in disks of other servers via network, so writing performance declines more significantly compared to a normal local disk that is not to be mirrored. In case of the remote cluster configuration, since the network communication speed is slow and delay is long, the writing performance declines drastically.

In asynchronous mirroring, data is written to the local server immediately. However, when writing data to other server, it is saved to the local queue first and then written in the background. Since the completion of writing data to other server is not waited for, even when the network performance is low, the writing performance will not decline significantly. However, in case of asynchronous mirror, the data to be updated is saved in the queue for every writing request as well, so the writing performance declines more significantly, compared to the normal local disk that is not to be mirrored and the shared disk. For this reason, it is recommended to use the shared disk for the system (such as the database system with lots of update systems) that is required high throughput for writing data in disks.

In case of asynchronous mirroring, the writing sequence will be guaranteed, but the data that has been updated to the latest may be lost, if an active server shuts down. For this reason, if it is required to inherit the data immediately before an error occurs for sure, use synchronous mirroring or the shared disk.

History file of asynchronous mirroring

In mirror disk with asynchronous mode, data that cannot afford to be written in memory queue is recorded temporarily in a folder specified to save history files. When the limit of the file is not specified, history files are written in the specified folder without limitation. In this case, the line speed is too low, compared to the disk update amount of application, writing data to other server cannot catch up with updating the disk, and history files will overflow from the disk.

For this reason, it is required to reserve a communication line with enough speed in the remote cluster configuration as well, in accordance with the amount of disk application to be updated.

In case the folder with history files overflows from the disk because the communication band gets narrowed or the disk is updated continuously, it is required to reserve enough empty space in the drive and specify the limit of the history file size. This space will be specified as the destination to write history files, and to specify the drive different from the system drive as much as possible.

Data consistency among multiple asynchronous mirror disks

- In mirror disk with asynchronous mode, writing data to the data partition of the active server is performed in the same order as the data partition of the standby server.
- This writing order is guaranteed except during the initial mirror disk configuration or recovery (copy) period after suspending mirroring the disks. The data consistency among the files on the standby data partition is guaranteed.
- However the writing order is not guaranteed among multiple mirror disk resources. For example, if a file gets older than the other and files that cannot maintain the data consistency are distributed to multiple asynchronous mirror disks, an application may not run properly when it fails over due to server failure.
- For this reason, be sure to place these files on the same asynchronous mirror disk.

Multi boot

Avoid using multi boot if either of mirror disk or shared disk is used because if an operating system is started from another boot disk, access restrictions on mirroring and the shared disk become ineffective. The mirror disk consistency will not be guaranteed and data on the shared disk will not be protected.

Before installing ExpressCluster

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

File system

Use NTFS for file systems of a partition to install OS, a partition to be used as a disk resource of the shared disk, and of a data partition of a mirror disk resource.

Communication port number

In ExpressCluster, the following port numbers are used by default. 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					
From			**To**		**Used for**
Server	Automatic allocation[2]	→	Server	29001/TCP	Internal communication
Server	Automatic allocation	→	Server	29002/TCP	Data transfer
Server	Automatic allocation	→	Server	29003/UDP	Alert synchronization
Server	Automatic allocation	→	Server	29004/TCP	Communication between disk agents
Server	Automatic allocation	→	Server	29005/TCP	Communication between mirror drivers
Server	29106/UDP	→	Server	29106/UDP	Heartbeat

Server to Client					
From			**To**		**Used for**
Client	Automatic allocation	→	Server	29007/TCP 29007/UDP	Client service communication

WebManager to Server					
From			**To**		**Used for**
WebManager	Automatic allocation	→	Server	29003/TCP	http communication

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

Server that the browser is connected to the Integrated WebManager to Target server					
From			**To**		**Used for**
Server that the browser is connected to the Integrated WebManager	Automatic allocation	→	Server	29003/TCP	http communication

If mirror connect monitor resources are going to be used, you need to let icmp packets through because EXPRESSCLUSTER checks if ping reaches between servers. If mirror connect monitor resources are going to be used, modify firewall settings so that ping reaches between servers.

Clock synchronization

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

Partition for shared disk

- If multiple servers that are connected to the shared disk are started while access is not restricted by ExpressCluster, data on the shared disk may be corrupted. When the access is restricted, make sure to start only one of the servers.
- When the shared disk method is used to solve network partition, create a raw partition (disk heartbeat partition) with space larger than 17 MB that disk network partition resolution resources use on the shared disk.
- Format the partition (switchable partition) used to transfer data between servers as disk resources with NTFS.
- For each partition on the shared disk, assign the same drive letter on all servers.
- Partitions on the shared disk can be formatted and created from one of the servers. It is not necessary to recreate or reformat a partition on each server. However, the drive letter needs to be set in each server.
- When you continue using the data on the shared disk at times such as server reinstallation, do not create or format a partition. The data on the shared disk gets deleted if you allocate or format a partition.

Partition for mirror disk

- Create a raw partition with larger than 17 MB space on local disk of each server as a management partition for mirror disk resource (cluster partition.)
- Create a partition (data partition) for mirroring on local disk of each server and format it with NTFS. It is not necessary to recreate a partition when the existing partition is mirrored.
- Set the same data partition size to both servers.
- Set the same drive letter to both servers for a cluster partition and data partition.

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[3]:

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

Verifying the network settings

- On all servers in the cluster, verify the status of the following networks using the ipconfig or ping command.
    - Public LAN (used for communication with all the other machines)
    - Interconnect-dedicated LAN (used for communication between servers in ExpressCluster)
    - Mirror connect LAN (used with interconnect)
    - Host name
- The IP address does not need to be set as floating IP resource in the operating system.

Notes when creating the cluster configuration data

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

Final action for group resource deactivation error

If 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.

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.

3OS start up time setting may be ignored when there is only one OS to select at boot time. In this case, edit the boot.ini file and add the second entry to Operating System. The copy of the first entry can be used for the second entry.

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) command 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” on the WebManager tab or less than 30 seconds. If you set it for less than 30 seconds, it may affect the performance of ExpressCluster.

Heartbeat resource settings

- You need to set at least one kernel mode heartbeat resource.
- It is recommended to register an interconnect-dedicated LAN and a public LAN as heartbeat resources. (It is recommended to set more than two heartbeat resources.)

After start operating ExpressCluster

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

Limitations during the recovery operation

Do not perform the following operations by the WebManager or from the command line while recovery processing is changing (reactivation → failover → last operation), if a group resource (disk resource or application 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 manuals

Executable format files and script files which are not described in Chapter 3, “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.

Cluster shutdown and cluster shutdown reboot

When using a mirror disk, do not execute cluster shutdown or cluster shutdown reboot from the clpstdn command or the WebManager while a group is being activated. A group cannot be deactivated while being activated. OS may shut down while mirror disk resource is not properly deactivated and mirror break may occur.

Shutdown and reboot of individual server

When using a mirror disk, if you shut down the server or run the shutdown reboot command from the command or the WebManager, a mirror break occurs.

Recovery from network partition status

The servers that constitute a cluster cannot check the status of other servers if a network partition occurs. Therefore, if a group is operated (started/stopped/moved) or a server is restarted in this status, a recognition gap about the cluster status occurs among the servers. If a network is recovered in a state that servers with different recognitions about the cluster status are running like this, a group cannot be operated normally after that. For this reason, during the network partition status, shut down the server separated from the network (the one cannot communicate with the client) or stop the ExpressCluster Server service. Then, start the server again and return to the cluster after the network is recovered. In case that a network is recovered in a state that multiple servers have been started, it becomes possible to return to the normal status, by restarting the servers with different recognitions about the cluster status.

When a network partition resolution resource is used, even though a network partition occurs, emergent shut-down of a server (or all the servers) is performed. This prevents two or more servers that cannot communicate with one another from being started. When manually restarting the server that emergent shut down took place, or when setting the operations during the emergent shut down to restarting, the restarted server performs emergent shut down again. (In case of ping method or majority method, the ExpressCluster Server service will stop.) However, if two or more disk heartbeat partitions are used by the disk method, and if a network partition occurs in the state that communication through the disk cannot be performed due to a disk failure, both of the servers may continue their operations with being suspended.

Notes on the WebManager

- If the client data update method settings of the WebManager is set to “Polling,” the information displayed on the WebManager is regularly updated and the latest status is not immediately displayed even if the status has changed. 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. If the client data update method settings of the WebManager is set to “Polling,” wait for the next automatic update or click the **Reload** button to reacquire the latest information. If “Realtime” is set, the information is automatically updated to the latest status.
- 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, configure the settings for the proxy server be able to relay the port number of the WebManager.
- When updating ExpressCluster, close the Web browser. Clear the Java cache and open the browser.

Notes on the Builder

- ExpressCluster does not have the compatibility of the cluster configuration data with the following products.
    - The Builder of other than ExpressCluster X1.0 for Windows
    - The Builder for ExpressCluster for Linux
    - The Builder of ExpressCluster for Windows Value Edition
- 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 **Save** 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 **Save** before reloading.

ExpressCluster Disk Agent Service

Make sure not to stop the ExpressCluster Disk Agent Service. This cannot be manually started once you stop. Restart the OS, and then restart the ExpressCluster Disk Agent Service.

Changing the cluster configuration data during mirroring

Make sure not to change the cluster configuration data during the mirroring process including initial mirror configuration. The driver may malfunction if the cluster configuration is changed.

Appendix

Appendix A. Glossary

Cluster partition	A partition on a mirror disk. Used for managing mirror disks. (Related term: Disk heartbeat partition)
Interconnect	A dedicated communication path for server-to-server communication in a cluster. (Related terms: Private LAN, Public LAN)
Virtual IP address[4]	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. Data partition for mirror disks. (Related term: Cluster partition)
Network partition	All heartbeat is lost and the network between servers is partitioned. (Related terms: Interconnect, Heartbeat)

[4] This applies only for Windows version.

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.
Mirror connect	LAN used for data mirroring in a data mirror type cluster. Mirror connect can be used with primary interconnect.
Mirror disk type cluster	A cluster system that does not use a shared disk. Local disks of the servers are mirrored.

Appendix B. Index