

Reliant UNIX 5.45

High Availability

Comments... Suggestions... Corrections...

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Fax forms for sending us your comments are included at the back of the manual.

There you will also find the addresses of the relevant User Documentation Department

Copyright and Trademarks

Copyright © 1999 Siemens AG.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

This manual is printed on
paper treated with
chlorine-free bleach.

Preface

Systems operating under Reliant UNIX - terminology

Systems operating under Reliant UNIX - achieving high availability

Systems operating under Reliant UNIX - system-related measures

Software-related measures

Cluster systems

Reliant Cluster Server II (RCS II)

Glossary

Abbreviations

Related publications

Continued ►

Contents

1	Preface	1
1.1	Target group	1
1.2	Summary of contents	1
1.3	Changes made since the previous version	2
1.4	Notational conventions	2
1.5	Further reading	3
1.6	Telephone and fax numbers of the support center	3
2	Systems operating under Reliant UNIX – terminology	5
2.1	Availability	5
2.2	What influences the availability of a computer system?	6
2.3	High availability	7
2.4	Scalable availability	8
3	Systems operating under Reliant UNIX – achieving high availability	13
3.1	Hardware	13
3.1.1	Basic availability	13
3.1.1.1	Measures in design, development and production	14
3.1.1.2	Hardware tests on the RM models	14
3.1.1.3	Online replacement	15
3.1.1.4	Customer service	15
3.1.2	Higher availability	15
3.1.2.1	Protection against power failures	16
3.1.2.2	Protection against the failure of components	16
3.1.2.3	Protection against hard-disk crashes	16
3.1.2.4	Using remote disk cabinets	17
3.1.2.5	Using cluster systems	17
3.2	Software	18
3.2.1	Basic availability	18
3.2.1.1	Using the Veritas file system	18
3.2.1.2	Automatic restart after an “unrecoverable error”	19
3.2.2	Higher availability	19
3.2.2.1	Using cluster systems	19
3.2.2.2	Detecting problems in applications	20
3.3	Data	20
3.3.1	Basic availability	20
3.3.1.1	Protection against data corruption	20
3.3.1.2	Data integrity	21
3.3.1.3	Data backup	21
3.3.1.4	File system backup and installation (DSSI)	21

Contents

- 3.3.2 Higher availability 22
 - 3.3.2.1 Data integrity 22
 - 3.3.2.2 Data backup 24
- 3.4 Operation 25
 - 3.4.1 Basic availability 25
 - 3.4.2 Higher availability 26
 - 3.4.2.1 Avoiding operating errors 26
 - 3.4.2.2 Avoiding interruptions for backups 26
 - 3.4.2.3 Changing the operating system 26
- 3.5 Expanding the RM models 27
- 4 Systems operating under Reliant UNIX – system-related measures 29**
 - 4.1 Online replacement and lockout auto-recovery (OLR/LAR) . . . 29
 - 4.1.1 Online replacement (OLR) 30
 - 4.1.2 Lockout auto-recovery (LAR) 30
 - 4.1.2.1 Example of how LAR works 31
 - 4.1.2.2 LAR-capable components 32
 - 4.1.2.3 Threshold monitoring for recoverable hardware errors 32
 - 4.2 Using UPS and BBU 33
 - 4.3 Redundant hardware components 35
 - 4.3.1 Power supply units 35
 - 4.3.2 Fan Trays 36
 - 4.3.3 Disk controllers 36
 - 4.4 RAID systems 36
 - 4.4.1 RAID 0 36
 - 4.4.2 RAID 1 37
 - 4.4.3 RAID 2 40
 - 4.4.4 RAID 3 41
 - 4.4.5 RAID 4 42
 - 4.4.6 RAID 5 43
- 5 Software-related measures 45**
 - 5.1 The Veritas file system (VxFS) 45
 - 5.1.1 Performance characteristics 45
 - 5.1.2 Restoring file system integrity 46
 - 5.1.3 Online file system administration 46
 - 5.1.3.1 Restricting fragmentation 46
 - 5.1.3.2 Changing the size of the file system 47
 - 5.1.4 Online data backup 48
 - 5.2 Logging V3.0 49
 - 5.3 Data backup and archiving 50
 - 5.3.1 NetWorker 50
 - 5.3.1.1 Data backup 50

5.3.1.2	Archiving	52
5.3.2	The EMASS product family	53
5.3.3	Non-Siemens product HYPERTape (company Multistream) . . .	53
5.4	Transaction monitors	54
5.4.1	Overview	54
5.4.2	openUTM	57
5.4.2.1	Services of openUTM	57
5.4.2.2	The most important services of openUTM for high availability .	59
6	Cluster systems	61
6.1	Fundamental structure	61
6.2	Failover clusters	62
6.2.1	General features of a high-availability monitor	63
6.2.1.1	Sign-of-life monitoring of the nodes	63
6.2.1.2	Individual response scripts	63
6.2.1.3	IP address management	63
6.2.2	Redundant use of serial peripherals	64
6.2.3	Restrictions placed on applications	65
6.2.4	The OBSERVE software	66
6.2.4.1	Switching over disks	68
6.2.4.2	Logical switchover of IP addresses	69
6.2.5	Reliant Monitor Software (RMS)	71
6.3	OPS clusters (ORACLE Parallel Server)	73
6.4	Integrating OPS into a failover cluster	76
6.5	XPS Clusters (INFORMIX Extended Parallel Server)	78
6.6	SPS Clusters (Siemens Parallel Server)	78
6.7	Hardware configuration for clusters	79
6.7.1	General requirements for clusters	79
6.7.1.1	Cluster nodes	79
6.7.1.2	Shared disks	80
6.7.1.3	Additional requirements	80
6.7.1.4	Controller Area Network (CAN bus)	80
6.7.2	Special requirements for failover clusters with OBSERVE . . .	80
6.7.2.1	Serial connections for sign-of-life monitoring	80
6.7.2.2	LAN interface	80
6.7.2.3	Cluster console (optional)	80
6.7.3	Special requirements for RMS clusters and OPS clusters . . .	81
6.7.3.1	Virtual Ethernet	81
6.7.3.2	Cluster console	81
6.7.3.3	Cluster-console Ethernet (private)	81
6.8	Hardware Configuration for RCS II	81

Contents

7 Reliant Cluster Server II (RCS II) 83

7.1 The Concept 84

7.1.1 The building blocks of the Reliant Cluster Server II 85

7.1.2 The different editions 87

7.1.2.1 The features of the Scalability Edition 88

7.1.2.2 The features of the Failover Edition 90

7.1.2.3 The features of Domain Admin 91

7.1.2.4 The features of the high-availability SAP R/3 system
 (Turnkey Solution) 92

7.2 Hardware Architecture of the Reliant Cluster Server II 93

7.2.1 RM600 E systems 93

7.2.2 The cluster console 94

7.2.3 Connecting the cluster nodes 94

7.2.3.1 Cluster Interconnect 94

7.2.3.2 I/O Interconnect 96

7.3 Software Architecture of the Reliant Cluster Server II 97

7.3.1 Reliant Monitor Software (RMS) 97

7.3.2 Parallel databases OPS and XPS 97

7.3.3 Siemens Parallel Server (SPS) 97

7.3.4 Reliant UNIX 5.45 98

7.3.5 Additional cluster management products 98

7.4 Compatibility and Migration 98

Glossary 99

Abbreviations 105

Related publications 109

Index 117

1 Preface

This manual explains the term *availability* in connection with computer systems, how the high availability of systems running under Reliant® UNIX® is achieved, and how it can be further enhanced. The manual presents the various aspects of availability, describes the concepts underlying it, and explains the possible methods for achieving availability.

1.1 Target group

This manual is intended for DP managers, system organizers, system administrators, and other users who want to find out more about increasing the availability of a computer system.

1.2 Summary of contents

Chapter 2, “Systems operating under Reliant UNIX – terminology”, begins with a definition of the term *availability* and goes on to explain the importance of a computer system’s availability. This chapter applies to all systems running under Reliant UNIX. It is followed by chapters for the single-processor and symmetrical multiprocessor systems RM200, RM300, RM400 and RM600, and for cluster systems and Reliant Cluster Server II. This distinction is necessary because the differences between the system architecture of cluster systems of type Reliant Cluster Server II and those of the remaining RM models can have a major effect on the ways in which high availability is achieved.

Chapters 3 through 6 deal with the availability aspects of the single-processor and symmetrical multiprocessor systems RM200 through RM600. Many of the aspects described there are not system-specific and can apply to all RM models, including the Reliant Cluster Server. If you require information on the availability of individual products for special RM models, please contact our Sales department.

- The chapter “Systems operating under Reliant UNIX – achieving high availability” on page 13 gives you an overview of how the availability of the computer systems RM200 through RM600 is guaranteed, and how it can be increased still further.

- This is followed by the chapter “Systems operating under Reliant UNIX – system-related measures” on page 29, which describes in more detail various system-related measures that were mentioned in the previous chapter, such as hardware extensions or specific features of systems running under Reliant UNIX.
- The chapter “Software-related measures” on page 45 explains in more detail how availability can be increased using the supplied software or additional software.
- The chapter “Cluster systems” on page 61 then describes how standard systems can be made into a high-availability cluster with just a few additional components.

Finally, chapter 7, “Reliant Cluster Server II (RCS II)”, on page 83 deals with availability aspects of the Reliant Cluster Server II.

The manual is completed by a reference section containing a list of abbreviations, a glossary of technical terms, a list of related publications, and an index.

1.3 Changes made since the previous version

The chapter “Reliant Cluster Server” in the previous manual has been renamed “Reliant Cluster Server II” and completely reworked. Apart from describing the system architecture, it principally deals with the strong points of the Reliant Cluster Server II, which combines the advantages of high-availability and scalability.

1.4 Notational conventions

This manual uses the following notational conventions:

“Double quotes”

for references within the manual and to other documentation

italics

for command names and file names and for emphasis in the text



to indicate important information you should read

1.5 Further reading

The products and services mentioned in this manual have their own documentation, with detailed information on function, installation, configuration and operation. For the exact titles and notes on ordering, please see the “Related publications” section at the end of this manual.

1.6 Telephone and fax numbers of the support center

Germany

Siemens AG Call Management Center (CMC)

Telephone number: (0180) 54040

Fax number: (0180) 533 6 799

USA and Canada

Pyramid Technology Corporate Customer Support Center (CCSC)

Telephone number USA and Canada: (800) 695-2272

International telephone number : (408) 428-9000

Fax number: (408) 428-7450

International (except Germany, USA and Canada)

Ask your local Siemens provider or branch for the telephone and fax numbers of a nearby support center.

2 Systems operating under Reliant UNIX – terminology

This chapter begins with a definition of the term “availability” and explains which aspects of the concept are significant for computer systems.

2.1 Availability

The trend away from large centralized DP solutions toward distributed client/server applications based on open systems is continuing. Traditionally, what used to be called company-critical applications and data were kept predominantly on central mainframe systems. Nowadays, however, applications run more and more frequently on open systems in a client/server architecture. In addition, ever-increasing areas of public, economic and private life depend on the smooth running of these applications. If they are interrupted by errors, danger can result, be it economic because large amounts of money are dependent on the operation, or in extreme cases danger to people's lives.

The smooth functioning of complex systems, such as computer systems, is a matter of statistics. If you are lucky, you can unpack your new computer, plug it in and watch it run without malfunction to the end of its days. However, if access to the applications and data on this computer is very important, for example because the economic livelihood of a production firm depends on the smooth running of the production control, simply relying on luck is not enough - the risk of a malfunction must be made calculable.

This is where the concept of *availability* comes in. The availability of a computer system is defined as the percentage of its operating time during which all programs and data dependent on this system can be used in the regular way, completely, in up-to-date form, and at the time they are needed. So, if a computer runs without malfunction from the moment it is first switched on until its planned disposal, its availability is said to be 100%, which would be the ideal value.

In real life, however, this is rarely the case, which is why this manual deals with measures for increasing the availability of systems running under Reliant UNIX and the applications and data stored on them, so that potential malfunctions become less frequent and, in the worst case, do not lead to disaster.

2.2 What influences the availability of a computer system?

The availability of a computer system is influenced by many factors. The figure below illustrates what needs to be taken into account if the availability of a computer system is to be increased:

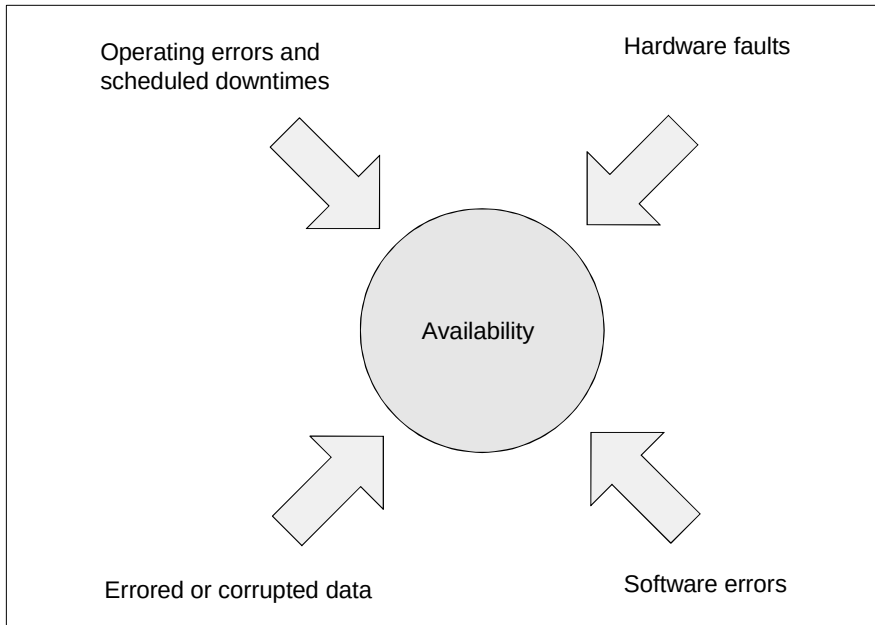


Figure 1: Factors influencing the availability of a computer system

Each of the four elements in the figure can be subdivided still further. The individual error sources and possible solutions are described in detail in the chapters that follow.

With computer systems the term *service interruption* usually signifies a hardware failure. A hardware component must be assumed to have failed as soon as the component involved stops functioning properly, even if it is still working. In principle an entire computer system can break down (e.g. due to a power failure) or each individual hardware component of the system can fail: power supply units, processors, storage units, controllers and hard disks are just as prone to wear and tear as all other parts of a system and its peripherals. The older a component, the greater the probability it will break down.

Apart from the hardware, software errors can also lead to system crashes, thus interrupting operation. But even if an error does not immediately cause the system or a program to crash, software errors can reduce the availability of a system if the wrong data is supplied or data is processed incorrectly.

As well as the hardware and software, one of the main elements of a computer system is the data which is stored on it. The whole purpose of a computer system is after all data processing. It thus follows that a computer system cannot be regarded as available if the data it is to process is unavailable, incomplete or contains errors.

Finally, the operating of the computer system by the system administrator or the users also has a considerable influence on the availability of the system. This is due not only to possible operating errors, but also to scheduled downtimes. Scheduled downtimes occur when an administrator switches off a system in order to carry out planned maintenance work of any kind. Much of this time is devoted to all data backup activities, but also software upgrades, preventive hardware maintenance or extending existing configurations (e.g. with additional CPUs or extra memory) can require scheduled deactivation.

Scheduled downtimes do not cause problems if the amount of time required is available; but if, for example, a company needs 20-hour availability every day and the daily offline data backup requires seven hours, something has to be done to bring the necessary availability into line with the equally important data backup procedure.

2.3 High availability

The availability percentage of a computer system with a standard configuration ex factory is known as the *basic availability* of the system. Without any additional precautions, systems running under Reliant UNIX have a basic availability of around 99%; assuming that they are used 365 days a year, 24 hours a day, this gives them a statistical downtime of approx. 88 hours per year. Bear in mind, however, that these 88 hours of downtime do not occur regularly each year or all together in one block.

Assuming a statistical downtime of 88 hours per year, distributed more or less equally over four system crashes, this means that the computer system is unavailable for less than four days per year.

In many cases, this 99% availability is sufficient. An automobile workshop, for example, which only uses its computer system to draw up offers and invoices, will not normally find its livelihood threatened if the invoices are occasionally written one or two days later. Networked PCs with a Siemens UNIX system as data backup server could be a perfectly adequate solution in this case.

For a manufacturing business with an annual revenue of \$1 billion, on the other hand, the situation could be quite different. If, for example, nothing could be produced for half a day due to a computer breakdown in the central racking depot, the downtime would cost around \$2 million. Clearly one cannot simply rely on luck in this situation. Controlling this racking depot with a conventional single-processor system would simply be too great a risk; a high-availability cluster system would be required here.

High availability in connection with computer systems is therefore concerned with the final percentage of availability. A computer system whose availability has been increased above the 99% basic availability is therefore known as a high-availability system. What degree of high availability is necessary for specific scenarios cannot be explained here in a few sentences. This requires a precise requirements analysis as well as a cost/benefit analysis, which has to be tailored exactly to the individual case.

If in a particular case availability has to be increased to 99.998% (i.e. only 10 minutes of downtime per year based on 24-hour operation as above), it is only important that this requirement is fulfilled and what this costs, not what the solution looks like technically. Here, Siemens offers customized solutions based on standard elements of open systems, which can normally meet the demands of commercial use at least as well as complicated and expensive special systems (often called *error-tolerant systems* or *non-stop systems*).

2.4 Scalable availability

As we have seen, availability requirements are wide-ranging. Our strategy is to achieve higher availability by adding more functionality to a base of standard components. The same tried-and-tested basic components - whose availability aspects are described in more detail below - are suitably supplemented and combined.

Availability thus becomes scalable and can be adjusted to meet the customer's exact requirements. This scalable availability can be represented as a configurable model of high availability, in which different availability requirements can be met by means of specific expansion of the system. The basic availability of

99% already constitutes a very good starting level. The figure below illustrates this configurable model in detail, with the possible extensions to the system listed in the form of keywords, not all of which need necessarily be used.

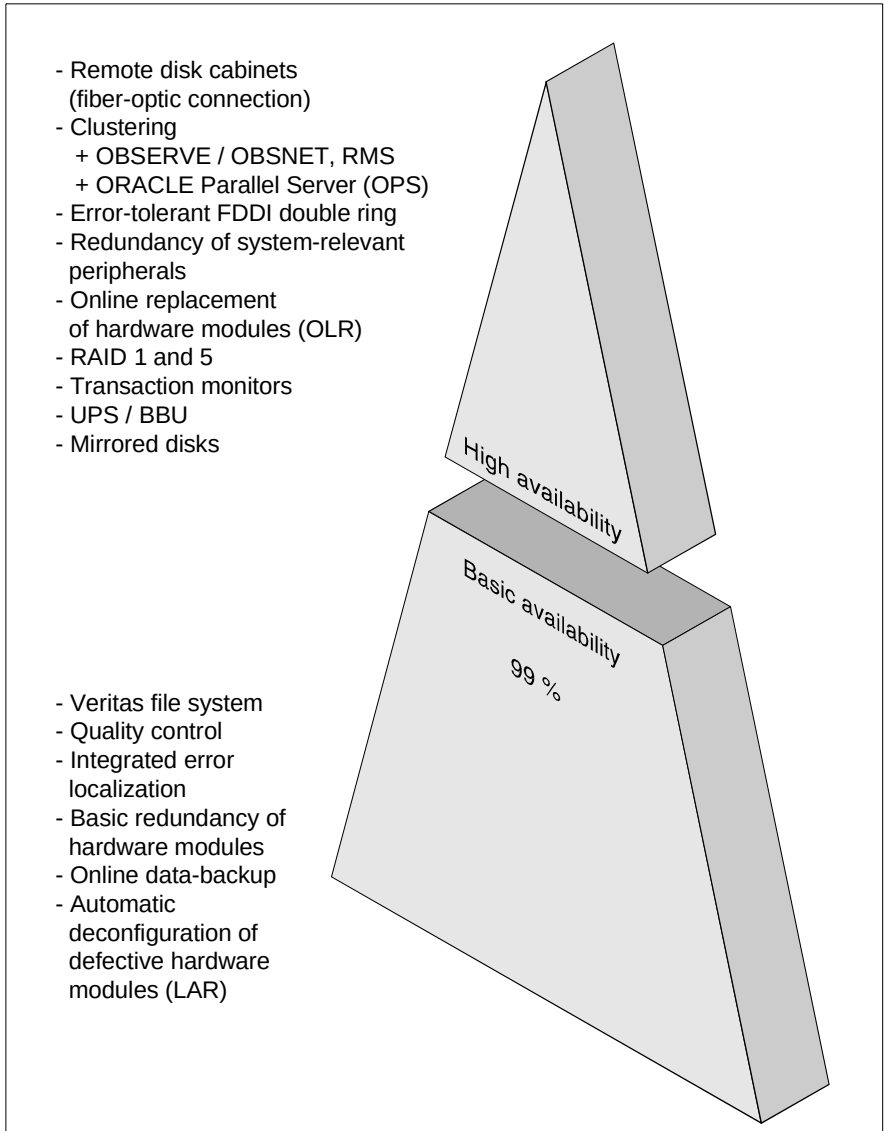


Figure 2: The configurable model of high availability

Because the above figure can only provide a very brief outline of what actually has to be done to achieve high availability, the following figure illustrates this in more detail, using as an example the availability of disk data: due to a faulty disk the “normal” conventional system on the far left must be repaired and the disk contents must be restored using conventional data backup methods. If disk mirroring is used, there is protection against a disk crashing, but there is no availability if other important components malfunction. For even higher requirements it is advisable to employ cluster concepts or an MPP system (massively parallel processing).

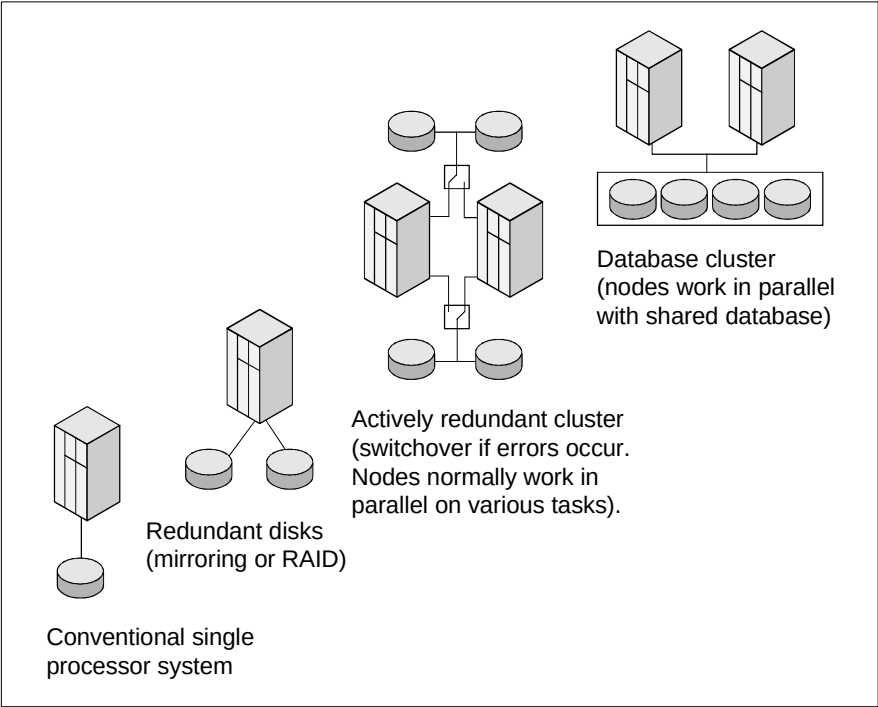


Figure 3: Hardware levels for higher availability

In this case customers decide which solution they want to use, depending on their requirements. From the expense point of view, they will need to weigh up the cost of additional redundancy in comparison to the possible costs of serious malfunctions.

Siemens' high-availability concept is based on the above-mentioned principle and offers various levels with a cost-effective entry level. Customers can begin with the high basic availability (contained in the operating system and in the hardware of RM systems) and can increase the availability of their computer systems via additional measures, such as using a battery backup unit (BBU), an uninterruptible power supply (UPS) or mirrored disks, right up to the highest level of cluster configuration.

Siemens' strategy for high availability can be broken down into many individual aspects and functions - all of which are both important in themselves and complement each other - and therefore requires more than a brief explanation. The following chapters attempt to delineate the most important facts and explain them in more technical detail for interested readers.

3 Systems operating under Reliant UNIX – achieving high availability

This chapter describes the means used to achieve the high basic availability of Siemens systems running Reliant UNIX and suggests how this availability can be increased still further. Many of the methods mentioned here are described in more detail in subsequent chapters.

The structure of this chapter reflects the possible impairment of availability through hardware faults, software errors, data problems and operating the system. For each of these areas, one of the following sections describes which measures have been taken by default to achieve the high basic availability and which additional measures can be applied.

Not all measures are available to the same extent for all RM models. Siemens' Sales and Marketing department will be happy to tell you which possibilities are available for specific models.

3.1 Hardware

A *hardware failure* is often the first thought that comes to mind when a *service interruption* occurs on a computer system. In actual fact, hardware failures are the cause of a very large proportion of all system stoppages. Preventing such malfunctions by guaranteeing high hardware availability, therefore, plays a particularly important role in achieving high availability for a system. There are various options available.

3.1.1 Basic availability

Even in their simplest versions, the RM models offer a range of options for achieving high basic availability. These are described below.

3.1.1.1 Measures in design, development and production

The high basic availability is achieved through special care in the design and development of the hardware components. This attention characterizes the entire development of hardware and software at Siemens. The design, development and production process is ISO 9001-compliant and largely eliminates errors.

Special importance is also given to the reliability of the hardware components used. Comprehensive testing ensures the high reliability of Siemens' hardware.

3.1.1.2 Hardware tests on the RM models

At every system start, RM systems conduct automatic tests on the hardware present (processors, memory, controllers etc.). If the test program finds a hardware fault, the relevant component is automatically deconfigured and the operating system starts up using only the intact components and employing one of the available redundancy facilities.

During operation also, an online diagnostics check can be performed on the system. This is available through the TDS2 utility, which can be requested free of charge by any customer with a maintenance contract.

The system environment of RM600 systems is protected by environmental interface processors (EIPs). You can find more detailed information about this in the manual "Reliant UNIX - Diagnostics and Troubleshooting", in the chapter "RM600 troubleshooting", section "System environment monitoring".

The system environment of RM200/300/400 systems is monitored by the daemons EIP and PTF (Reliant UNIX from version 5.44A) or the EEC daemon (Reliant UNIX from version 5.44B). You can find more detailed information about this in the manual "Reliant UNIX - Diagnostics and Troubleshooting", in the chapter "RM200/300/400 troubleshooting", section "System environment monitoring".

If important components such as hard disks should fail, it is not always necessary for the whole system to be deactivated. If, as with the Reliant UNIX operating system, the failed component can be pinpointed, it can often be simply deconfigured instead, enabling work to continue with a restricted configuration.

System components which become defective after the system's automatic self-test need not necessarily cause a system crash if lockout auto-recovery (LAR) is used. This procedure detects, at run time, both permanent and transient errors, and ensures that faulty components are properly identified and automat-

ically deconfigured, at the latest at the next system start, so that they will not impair operation any further. LAR functions for all components on the central bus and for the RM600 models on the I/O bus (see also section “Online replacement and lockout auto-recovery (OLR/LAR)” on page 29).

3.1.1.3 Online replacement

Online replacement (OLR) enables components that are found to be defective to be exchanged and configured back into the system while it is in operation (see also the section “Online replacement and lockout auto-recovery (OLR/LAR)” on page 29). It is therefore no longer necessary to shut down the whole system in order to repair the defective component. This online replacement is possible for hard disks, in-cabinet magnetic tape devices (only on the RM400 and RM600), power supply units and fans and is controlled via a simple, user-friendly user interface (see also the manuals “Reliant UNIX V5.4x - Hardware Configuration with Config under SINIX/windows” or “Reliant UNIX V5.4x - System Administration and Hardware Configuration Using the SYSADM User Interface”).

3.1.1.4 Customer service

Finally, the customer service department of Siemens AG constitutes a not inconsiderable factor influencing availability. Several thousand staff ensure the smooth running of all customer service requests, which go far beyond repairing defective computer systems. The wide-ranging service enables individual requirements to be met according to the customers' very specific requests. The customizable services offered by Siemens' customer service department range widely from the “classic” ones such as telephone and online support, repair and spare parts service, via additional services such as system installation, networking, system operation, cleaning and remote maintenance, to backup solutions in the event of a catastrophe.

3.1.2 Higher availability

There is a wide range of options available for reducing the possibility of hardware failures still further, or for lessening their consequences.

3.1.2.1 Protection against power failures

A basic means of protecting a computer system against frequent malfunctions is to absorb the effect of power loss or extreme voltage disturbances by means of battery backup units (BBUs) or uninterruptible power supplies (UPSs) (see also the section “Using UPS and BBU” on page 33). These can be used to prevent malfunctions based on undervoltage or overvoltage, as well as data loss due to power failures. Under- and overvoltage are filtered out and in the event of a power failure, the use of UPSs or BBUs ensures a controlled shutdown of the protected system; in conjunction with an emergency power supply, these devices even allow uninterrupted operation during power failures.

3.1.2.2 Protection against the failure of components

One possibility of protection against the failure of individual components is to use redundant components and modules. This means that an individual components of the computer system is supplemented by an individual component of the same type, which takes over the tasks of the original component should it fail. Typical components which can be designed redundantly are:

- power supply units
- fans
- disk controllers (in conjunction with disk mirroring)

For further information see the section “Redundant hardware components” on page 35.

3.1.2.3 Protection against hard-disk crashes

If availability is to be increased, additional protection against hard-disk crashes is especially important, because the data on the hard disks is often far more valuable than all the hardware of the computer system put together. If this is the case, data loss due to a hard-disk crash cannot be risked. Even if the data can be restored from a backup, this often means a prolonged service interruption because restoration can take some time if there is a lot of data involved.

To save this time, conventional protection measures can reproduce the data stored on the hard disks even if a hard disk crashes. The simplest option here, which is already standardly included in the operating system, is to set up mirrored disks (see “Disk-mirroring software” on page 38). This procedure involves a hard disk being supplemented by a (redundant) second disk, which

contains exactly the same data as the original disk. Continuous comparison of both disks ensures that their contents are always identical, even while the system is in operation.

Because this procedure requires twice as much hard-disk space, it may be unsuitable for very large quantities of data. In such cases, an alternative is to use disk arrays (see “The disk arrays PXRC and PXRE” on page 44), which enable redundant data storage according to RAID level 5 (or 3) (see the section “RAID systems” on page 36).

3.1.2.4 Using remote disk cabinets

To effectively protect programs and data against physical influences (e.g. catastrophes such as fire, earthquakes etc.), disk cabinets can be set up several hundred meters away from the computer system (e.g. in specially fireproof rooms or buildings) by means of fiber-optic (FO) cabling.

There are components available for fiber-optic cables for SCSI interfaces, Fibre Channel interfaces and the diagnostic bus (CAN bus).

3.1.2.5 Using cluster systems

Although the methods used up to now already provide a very high level of protection against hardware failure there are still components which are not protected, and the failure of a single, unprotected component can impair the entire system (single point of failure). To avoid system crashes, standard systems can be combined to form a high-availability cluster system, in which all single points of failure are eliminated. In such a system, therefore, the failure of any individual component can no longer cause the whole system to crash.

In cluster systems, several individual computers are combined to form a cluster with shared peripherals such as disk pools, terminals, printers and perhaps an ORACLE database.

If one of the computer systems (cluster nodes) involved crashes, the peripherals can be addressed via a different cluster node, whatever the reason for the crash. The peripherals can also be automatically switched over to another cluster node entirely transparently, so that the users of the defective node hardly notice its failure.

3.2 Software

Service interruptions can also be caused by software errors. There are therefore a whole series of measures which have either already been incorporated into systems as standard or which can be performed additionally in order to increase the availability of the system.

3.2.1 Basic availability

The software for systems running under the Reliant UNIX operating system, and the available application programs, have also already been optimized with regard to reliability at the design and development stages. In particular Siemens' UNIX operating system is subjected to intensive debugging tests.

Even the basic versions of Reliant UNIX systems offer a range of precautions for increasing system availability. These include the Veritas file system and the automatic restart facility.

3.2.1.1 Using the Veritas file system

Compared to standard UNIX file system type ufs, the Veritas file system (VxFS) supplied with the operating system offers a range of benefits (see also the section "The Veritas file system (VxFS)" on page 45). These benefits include:

- fast restoration of file system integrity following errors
- online file system administration
- online data backup
- extended options for the *mount* command (e.g. additional possibilities for ensuring data integrity)
- improved performance behavior

The availability of systems running under Reliant UNIX can thus be greatly increased by simple means. VxFS is the default file system for the Reliant UNIX operating system.

3.2.1.2 Automatic restart after an “unrecoverable error”

RM systems running under Reliant UNIX normally start up again automatically following a serious error. This restart behavior can be optimized.

The following measures can be used to ensure a quick restart of the system:

Reducing the startup times through an improved file system check (fsck -f) if you are using the ufs file system

It is possible to reduce the time the computer needs to check the file system after it has been switched on. Still more time can be saved by configuring these checks to run in parallel. Further improvements are possible using the Veritas file system (see also above and the section “The Veritas file system (VxFS)” on page 45).

Creating an alternative boot partition

If an alternative boot partition is set up, the operating system can be loaded even when the original boot partition is damaged and no longer readable.

Optimizing memory dumps

By defining various options for memory dumps it is possible to specify the dumps so as to reduce restart times for the system, depending on the respective configurations (e.g. by specifying a special dump partition).

3.2.2 Higher availability

A variety of individual measures are possible to increase the availability of the system still further.

3.2.2.1 Using cluster systems

As already described in the section “Using cluster systems” on page 17, cluster systems ensure that if one node fails the common peripherals of the cluster nodes remain available through the remaining nodes. This technology offers effective protection not only against hardware failures but also against errors occurring in the operating-system software, because part of the cluster can still be used.

3.2.2.2 Detecting problems in applications

There are two possibilities for safely and conveniently tracking problems and errors in programs during current operation. Both methods are based on the continuous monitoring of active applications while they are running, which allows error situations to be recognized quickly and reliably and reported.

One option is to use the program Logging V3.0, included with the operating system, which can collect and evaluate data on the execution and errors of application programs in log files, provided these applications send output to the log file (see the section “Logging V3.0” on page 49). The other option is to use the fail-over cluster software OBSERVE or RMS (see the section “Failover clusters” on page 62) in a special operating mode on an individual computer to recognize application and system states and respond to them.

Using the software OBSERVE or RMS for monitoring applications will be especially interesting if it is already being used in a cluster and the individual systems in the cluster also want to monitor applications.

3.3 Data

This section is primarily concerned with the possibility that data could be corrupted or destroyed as a result of system errors or external influences (e.g. hackers).

3.3.1 Basic availability

To achieve the high basic availability, various measures have been incorporated which largely preclude data errors.

3.3.1.1 Protection against data corruption

During operation the main memory and mass storage units as well as all buses work with automatic error detection and, if necessary, correction. For this purpose, all system buses are protected by a parity check for detecting transmission errors and avoiding other errors. The main memory of the system and the second-level cache operate internally using the error correction code (ECC), which recognizes all 2-bit errors and corrects all 1-bit errors during operation. Two-bit errors are avoided by correcting all recognized errors each time the

main memory is refreshed. In this way, the system prevents 1-bit errors in rarely-used cells from becoming 2-bit errors. The primary cache on the processor uses the parity check to detect errors.

The mass storage units, such as the hard disks, work with the usual cyclic redundancy procedures, which enable efficient error correction. The 1/2-inch magnetic tape drive and the 8-mm tape cartridge drive also use the read-after-write procedure during writing. With this procedure, the drive already checks while writing whether the data can also be read again, so as to avoid unpleasant surprises if it later has to be restored.

3.3.1.2 Data integrity

All RM systems already have a high degree of security built into the operating system. With this security, achieved through the mechanisms of the UNIX operating system for access control and password protection, the RM systems fulfill high security requirements without needing a special version of the operating system.

3.3.1.3 Data backup

The RM systems support various devices for backing up data. Siemens' operating system Reliant UNIX already includes several utility routines (e.g. *cpio*), which, with a bit of expert knowledge and shell-script programming, can be used to organize decent backups.

As of the operating system version SINIX V5.43, the Veritas file system is the standard file system (see also the section "The Veritas file system (VxFS)" on page 45).

This file system allows online data-backup during operation, thus increasing the possible availability of a system by simple means.

3.3.1.4 File system backup and installation (DSSI)

By booting the installation data-media, you can load a mini-root kernel. This mini-root kernel contains a program called DSSI (digital storage systems interconnect), which can be used to back up the data on the root partition and for system installation. This program is a service tool and runs only in the mini-root environment.

DSSI can be used to back up the partitions and disk-area tables of all physical hard disks present in the system both logically and physically, and to reload them. Because DSSI is started from the mini-root system, the root file system of the system disk can also be backed up without problems and reloaded (on the same disk type).

You can also use DSSI to duplicate the partitioning of a hard disk by loading another disk with the backed up data of the first one. This is useful if disks are to be exchanged or when cloning the system (1:1 image of the system).

As backup devices you can use all released drives for 1/4 inch, 8 mm and DAT which can be addressed via SCSI bus.

For further information on the DSSI program, see the manual “Reliant UNIX 5.44 - Installation and Operation”.

3.3.2 Higher availability

Increasing basic availability through data reliability means influencing above all data integrity and data backup.

3.3.2.1 Data integrity

A range of software products are available for helping to achieve this higher access protection:

AUDIT

AUDIT is a logging tool used to record as evidence processes which could affect the system security. The product AUDIT allows all accesses to system components to be stored and processed in log files. Version 2.0 of AUDIT allows convenient configuration by means of a configuration file and a single command. Via a system-oriented API, applications can also log their own data records for later use as evidence if required. The number and type of audit records can be individually defined by choosing from a multitude of selectable events. AUDIT provides a convenient evaluation system for selective analysis of the data (see also the “AUDIT V2.1” manual).



Together with the product AUDIT, SINIX V5.42 has been certified for all RM servers by the BSI (German Federal Office for Security in Information Technology) according to the security standard F-C2/E2 of the ITSEC criteria.

HORUS

HORUS is used for ongoing monitoring of a defined security strategy of a computer center. It offers both periodic checks on security-relevant system states which can be controlled via calendar functions, and continual monitoring and provision of alarms. In HORUS these are known as *event-oriented* and *state-oriented monitoring functions*. Here are two examples:

- Event-oriented monitoring functions include monitoring incoming TCP/IP requests (e.g. *rlogin*, *rsh*, *ftp*) or monitoring multiple failed login attempts. Naturally, existing system functions are used for this as far as possible. In these cases HORUS selects, summarizes and presents its records in standardized form to the user.
- An example of a state-oriented monitoring function is checking the quality of the passwords. This is done using the well-known program CRACK. CRACK tries to guess passwords using a dictionary. If it succeeds, access to the data of the user in question is wide open. One of the main criticisms of the password mechanism is that users, out of laziness, often choose passwords that are easy to guess (e.g. their own names). HORUS provides effective protection against this, which, to counteract CRACK, has been improved with a considerable extension of the dictionary. A further example is checking for so-called suid programs, which enable access to root (system administrator).

A highly user-friendly graphical interface makes configuration, i. e. selecting certain events or setting checkpoints, very easy. In addition the state changes set are indicated by alarms via the interface.

HORUS uses very few resources. Unlike Audit, it counts only specific events and triggers an alarm when certain threshold values are exceeded; this is not logged.

HORUS can be configured as a

- stand-alone product with its own interface and as a
- TransView Control Center agent

In the second case, it is possible to monitor several networked computers from a single point of administration.

ASECO

For all Siemens' Reliant UNIX platforms, ASECO supports network-wide chip-card access to the system and to applications. Authentication is by means of a cryptographic chip-card procedure, with no password being sent over the network. ASECO combines a high degree of security with the convenience of the single sign-on (SSO). Entering a personal identification number (PIN) once only is sufficient for all further network-wide authentication requests. ASECO supports the SICRYPT computer card as well as a range of other chip cards (see also the "ASECO V2.1" manual).

DCE

For networked client/server environments, a high level of data integrity can be guaranteed by integrating the Open Software Foundation's (OSF) Distributed Computing Environment (DCE): the security service of the DCE offers network-wide security for client/server systems. Because all other well-known manufacturers provide DCE security on their market-relevant operating systems, secure interoperability is also possible in heterogeneous networks.

3.3.2.2 Data backup

In this area too, there are some programs which can be used to increase availability:

Using the Veritas file system for operating system versions prior to SINIX V5.43

This file system allows online data-backup during operation, thus increasing the possible availability of a system by simple means (see also the section "The Veritas file system (VxFS)" on page 45).

NetWorker

NetWorker is the standard data-backup tool for systems running under Reliant UNIX (see the section "NetWorker" on page 50). In contrast to *cpio*, NetWorker offers convenient facilities for backup in the network. It is distinguished by its simple operation and outstanding integrability into heterogeneous environments. It is also notable for its performance values and high functionality.

If the NetWorker program is used in conjunction with changers for tape cartridges, regular data backups can also be automated, and devices with a total capacity of several terabytes can be connected.

Regular data backup is always necessary (even on systems with mirrored disks or RAID 5), because program or operating errors can lead to data corruption or can cause data to be created and stored with errors. In such cases, older data must be available.

The EMASS products AMASS and DataMgr

The products AMASS and DataMgr enable data backup or archiving to be implemented by means of a hierarchical storage management (see the section “The EMASS product family” on page 53). With this method, only one (virtual) file system is visible to the administrator and he or she need not bother about physical storage on specific media.

With DataMgr, you can also avoid operational errors resulting from overfilled hard disks by defining maximum “full states” for hard disks. If these are exceeded, parts of the data are automatically migrated to the archive.

3.4 Operation

Operators can restrict system availability with the following:

- operating errors (particularly by administrators)
- scheduled downtimes

3.4.1 Basic availability

To avoid a reduction in availability due to operating errors, all Siemens' systems that are to run under Reliant UNIX are delivered with comprehensive printed documentation, which enables users and administrators to find out the correct procedures for specific tasks even before the system is switched on. The hardware operating instructions provide detailed information on how to set up, cable and handle the hardware correctly. Observing the security and maintenance notes in these manuals also helps to keep the system functioning.

3.4.2 Higher availability

3.4.2.1 Avoiding operating errors

In order to avoid operating errors being the cause of service interruptions it is important that the staff involved have sufficient knowledge of the hardware and software used. To extend their knowledge Siemens AG offers a comprehensive, high-quality series of courses through its Training Center.

“Planned” operating errors (sabotage) cannot be avoided through system and software measures. In this case, anyone who uses a Reliant UNIX system must take appropriate action to deny unauthorized parties access to the hardware. Remote disk cabinets, which are physically separate from the system cabinets and connected to them via FO (fiber-optic) cabling, can be used to help protect the data stock.

3.4.2.2 Avoiding interruptions for backups

As already mentioned in the section “Data” on page 20, the Veritas file system can be used to enable online backups, as can the data-backup program NetWorker.

3.4.2.3 Changing the operating system

If you need to be able to change the version of the operating system even while the system is running, a cluster system is the solution. In a cluster system each system is switched over separately to the new operating system version, while the other systems remain available (see the chapter “Cluster systems” on page 61).

3.5 Expanding the RM models

Not all options listed in this chapter are available to the same extent for all RM models and operating system versions. Because of the rapid progress being made in the development of hardware and software, it makes little sense to provide detailed lists of the technical upgrade possibilities at this point. The general points given below should therefore only be taken as pointers. The up-to-date entries for the respective products in the Sales Handbook (SHB) are technically authoritative. If you require details of the upgrade possibilities for specific RM models, the Sales and Marketing department of Siemens AG will be happy to assist you.

Most of the measures listed are available for all RM models from RM200 through RM600. The following exceptions applied at the time this manual was printed:

- Redundant power supply units and fans, remote disk cabinets with FO cabling and LAR/OLR are possible for all RM models.
- Clusters can be set up using the system models RM300 through RM600.

For software extensions, only the following note applies:

- The Veritas file system comes with the operating system but it is only installed as default file system since SINIX V5.43. For earlier operating system versions, V5.4x, it has to be installed explicitly.

4 Systems operating under Reliant UNIX – system-related measures

This chapter provides additional information on the most important measures from the chapter “Systems operating under Reliant UNIX – achieving high availability” on page 13. It explains in more detail the system-related measures used to achieve high availability, i.e. those involving system hardware and software. Additional software products which can be used for this purpose are listed in the next chapter “Software-related measures” on page 45.

4.1 Online replacement and lockout auto-recovery (OLR/LAR)

Various RM models offer the possibility of online replacement (OLR) of some hardware components. This means exchanging the components while the system is running, without having to halt the entire system.

Before you can use online replacement, you have to be sure which of the components has failed. Both for OLR-capable components and all the other hardware components connected to the system bus or I/O bus, the system incorporates the lockout auto-recovery facility (LAR) which clearly identifies failed components.

With LAR the failed components can also be taken out of operation, either automatically or under the user's control, and the states of the monitored system components can be displayed graphically.

The availability of the overall system is thus increased through LAR/OLR especially when there are two or more of as many components as possible, so that the failure of an individual component is not so serious.

OLR and LAR are operated by means of the Config program, which is equipped with a graphical or an alphanumeric FMLI interface (see also the manuals “Reliant UNIX 5.4x - Hardware Configuration with Config under SINIX/windows” and “Reliant UNIX 5.4x - System Administration and Hardware Configuration Using the SYSADM User Interface”).

4.1.1 Online replacement (OLR)

OLR is offered for 3.5" SCSI drives, magnetic tape devices (only for the RM600), for cardcage and SCSI power supply units, and for fan trays. These components are normally replaced by Siemens service technicians. Certain peripheral cabinets also enable failed disks to be replaced by the customer's technicians. Please consult the operating instructions for your computer to see which hardware and software requirements apply to using the OLR facility for replacing components while the system is in operation.

4.1.2 Lockout auto-recovery (LAR)

Faulty hardware is detected at system startup by the test programs in the boot process, and while the operating system is running by the LAR facility.

LAR interacts here not only with the operator interface Config, but also with several different components of a Siemens system running under Reliant UNIX:

Information about defective boards is held in the HW_CONFIG defects table and is also stored in non-volatile RAM (NVRAM). Information about defective SCSI devices is held in the HW_CONFIG defects table and stored in the file */etc/opt/lar/scsi_dev.defect*. Components which are marked in this way as defective will then no longer be taken into operation, at the latest this will be at the next system start. You can find more detailed information about this in the manual "Diagnostics and Troubleshooting", in particular the sections "Activating the boards" in the chapters "RM600-E troubleshooting" and "RM600-xxx troubleshooting" and in the chapter "The LAR concept".

Error messages caused by defective hardware components are passed on to the Logging V3.0 software, provided the underlying fault does not lead to a system panic. Logging V3.0 then processes these error messages: it logs the errors and also performs threshold monitoring. With this type of monitoring, the frequency of an error can be checked and compared to a threshold value. If this value is reached, configurable, error-specific responses are initiated (see also the section "Logging V3.0" on page 49 and the chapter "Evaluation and administration of logbooks" in the manual "Diagnostics and Troubleshooting").

The interaction of the individual components is demonstrated below using an example of LAR:

4.1.2.1 Example of how LAR works

This example assumes an error from which the system can directly deduce that the board is defective.

In the system kernel a driver finds a board error which prevents the board from being used further. The driver supplements its error message with an error code that is unique across the whole system and passes it on to Logging V3.0.

Logging V3.0 records the message and then passes it on to the threshold monitor.

The threshold monitor has a threshold value of 1 for this particular error code, i. e. the response program which has been assigned to this error is called as soon as the error occurs for the first time. The response program sets the status *defective* for the relevant board, and the graphical interface is informed of this process.

The user is informed by the graphical interface that a defect has been discovered. At the highest level of the graphical display, the cabinet of the system configuration that contains the defective board is identified. The user can then display the slot containing the defective component step by step via the corresponding cardcage.

At the latest at system shutdown, the error information is written to the NVRAM and is thus available at the next system startup in the earliest possible phase.

At system startup the error found by the operating system and any errors found by the test programs in the boot process are incorporated into the configuration. This prevents the components marked as defective from being put into operation, and the system uses only the remaining components.

4.1.2.2 LAR-capable components

On RM600 systems, LAR is applied to the following system components:

- Boards and board components
- Hard disks, CD-ROM drives and magnetic tape drives
- BBUs, power units and fans (this also covers temperature problems)

On RM200/300/400 systems, LAR is applied to the following system components:

- Main memory (from RM300E on, RM400E)
- Hard disks, CD drives and magnetic tape drives (RM400E)
- Power units (including loss of the primary power supply) and fans (including temperature problems)

Components that are marked as defective will not be taken into operation, at the latest when the system is next booted.

When boards, board components and SCSI devices are analyzed, their position in the chassis is taken into account, as is the serial number if it can be read out by the software. If a board marked as defective at runtime is found to have a different serial number after rebooting, the marking indicating a defect will be deleted and the board will be activated. The same thing will also happen if a board marked as defective (with the same serial number) is found in a different slot.

4.1.2.3 Threshold monitoring for recoverable hardware errors

As well as hard errors, which prevent the affected hardware from being used, there are errors which can be recovered by repeating the operation or through error detection/correction measures.

Threshold monitoring is available for SCSI disk drives, magnetic tape devices and 1-bit memory corrections. If a threshold value is reached once or more than once (configurable), the affected component is marked as *error* or *defective*.

Error means that the component will not be taken out of operation, but will be marked via the interface (Config) so that preventive maintenance can be initiated if necessary.

Defective means that the component will be taken out of operation, at the latest at the next system start.

4.2 Using UPS and BBU

A common cause of unscheduled service interruptions is disruption of the energy supply. Even “normal” voltage fluctuations in the electricity network can, under unfavorable conditions, lead to malfunctions by destroying data or, in extreme cases, system components. If the power supply is interrupted, even just for a short time, such disturbances always occur. That is why even hardware insurers recommend the use of so-called battery backup units (BBU cabinets) and uninterruptible power supplies or UPS, which can remedy the situation.

In the event of a power outage, a BBU maintains the cabinet’s internal power supply. BBU cabinets are available as optional extras for RM600 systems. A BBU cabinet is equipped with a diagnostics processor (environmental interface processor, EIP2) and is thus integrated into the system’s cabinet-monitoring (see also the section “Hardware” on page 13). The system is therefore kept constantly informed of the charge of the BBU.

A UPS is a device that is linked into the power supply circuit between the socket and the end user, i.e. a computer, for example. While BBUs are intended exclusively for the cabinets of the RM400 and RM600, a UPS gives you the option of supplying emergency power not only to the cabinets but also to other devices, such as RM200s/RM300s, consoles, VDUs, terminal servers, and tape cartridge changers. A UPS can be used in addition to a BBU (to supply external devices).

In general there are two types of uninterruptible power supply, which work with different techniques: the standby technique or the online technique. With the standby technique, which only provides minor protection and will not be dealt with in any more detail here, the voltage runs from the socket directly and unfiltered to the computer (or another protected system component). In the event of a power failure, the batteries built into the UPS take over the power supply of the device within a few milliseconds.

By comparison, a UPS with online technique provides better protection because here the input voltage is internally filtered and is buffered via the batteries before it is passed on to the attached device (e.g. a computer). This completely filters out fluctuations in the operating voltage or the input frequency and provides a constant, interrupt-free operating voltage of, say, 230V. In the event of a power failure, the voltage is supplied immediately by the batteries without the need to switch over explicitly.

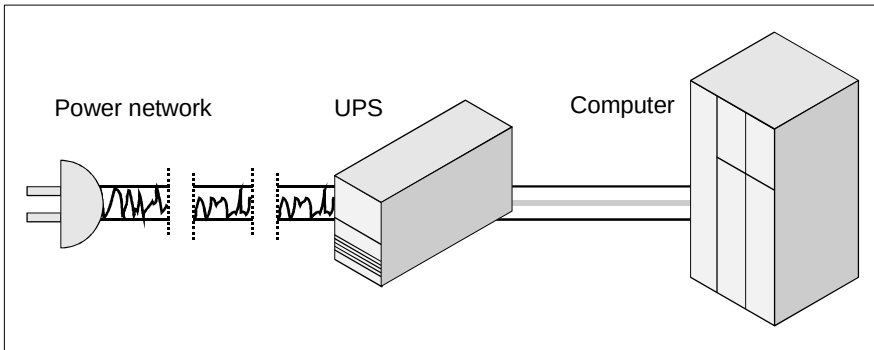


Figure 4: Functioning of a UPS with online technique

Because in both cases the batteries can only maintain the voltage for a short length of time, it is useful to be able to respond to a power outage. For this purpose, UPSs have an additional communication output, which can be connected to a computer and provides information on the current state of the power supply as well as the charge of the batteries. The system can use this information to take appropriate measures for protecting the data and applications.

In the simplest case, after a preset wait time the system can begin with an organized system shutdown in order to avoid data loss. More complex actions are possible, however, which are tailored to the individual requirements of the respective systems. If the system is also protected by an emergency power supply for critical applications, the use of the UPS may be restricted to bridging the brief period between the power failing and the emergency power supply being activated, without the need for further responses.

4.3 Redundant hardware components

Just like mechanical components, electrical and electronic components of a computer system are also subject to wear and tear, which can cause them to fail. The average operating time of a component until an error occurs (mean time between failure, MTBF) constitutes, as the name suggests, only an average value, which, in individual cases, can be way off the mark in either direction. To combat the failure of such components, more than one of each system component can be built in so that if one of these fails a substitute is available to take over the tasks of the original component.

4.3.1 Power supply units

In addition to the use of UPSs/BBUs, the power supply of important computer systems can be further protected by using redundant power supply units. If a computer system is only equipped with only one power supply unit and this unit fails, the whole computer system will fail, even if there is a UPS.

The principle of redundant power supply units is quite simple. Instead of just one, there are two or more completely independent power supply units, which supply the computer system with the necessary operating voltage. If one of these fails, the remaining ones will take over the supply.

Almost all system and extension cabinets of the RM series can be equipped with a redundant system power supply unit (some already have one by default). The defective power supply unit can be replaced during operation (online replacement, OLR). In addition, each peripheral chassis has its own power supply unit, which also supplies the accompanying fans and hard disk controllers. A fault in the power supply unit, therefore, causes only a few drives to fail. To make OLR available for these peripheral power supply units also, a second (redundant) unit can be connected parallel to each peripheral one.

4.3.2 Fan Trays

Each fan tray has three fans. If one fan fails, the remaining two are sufficient to cool the chassis. The defective tray can be exchanged without interrupting operation (OLR).

4.3.3 Disk controllers

Redundant disk controllers are particularly effective when mirrored disks are used (see also the section “RAID 1” on page 37).

For the RM300 and RM400 series there are PCI RAID controllers available which enable you to implement the functions of RAID levels 1 and 5 for system disks and disks in peripherals cabinets. This is an economical way of guarding against disk failures.

4.4 RAID systems

The RAID concept (redundant array of inexpensive disks) involves combining several hard-disk drives to form so-called disk arrays. In these disk arrays a proportion of the physical storage capacity is used to store redundant information on data. If one disk drive within an array fails, the missing data can then be reconstructed.

The RAID concept is divided into levels 0 through 5, of which levels 1, 3 and 5 can be used with RM systems. The following sections aim to explain these levels.

4.4.1 RAID 0

RAID 0 is a RAID level which does not effect the availability of data. At level 0, the data is distributed equally over several disks (disk striping) and no redundancy information is kept.

This level only provides a higher speed of access to the data, because the data can be written to or read by several drives in parallel. RAID 0 is therefore not described further here.

4.4.2 RAID 1

RAID 1 is also known as the mirrored-disk technique. At this level, all data is written in duplicate to two separate drives, which can be read independently of each other.

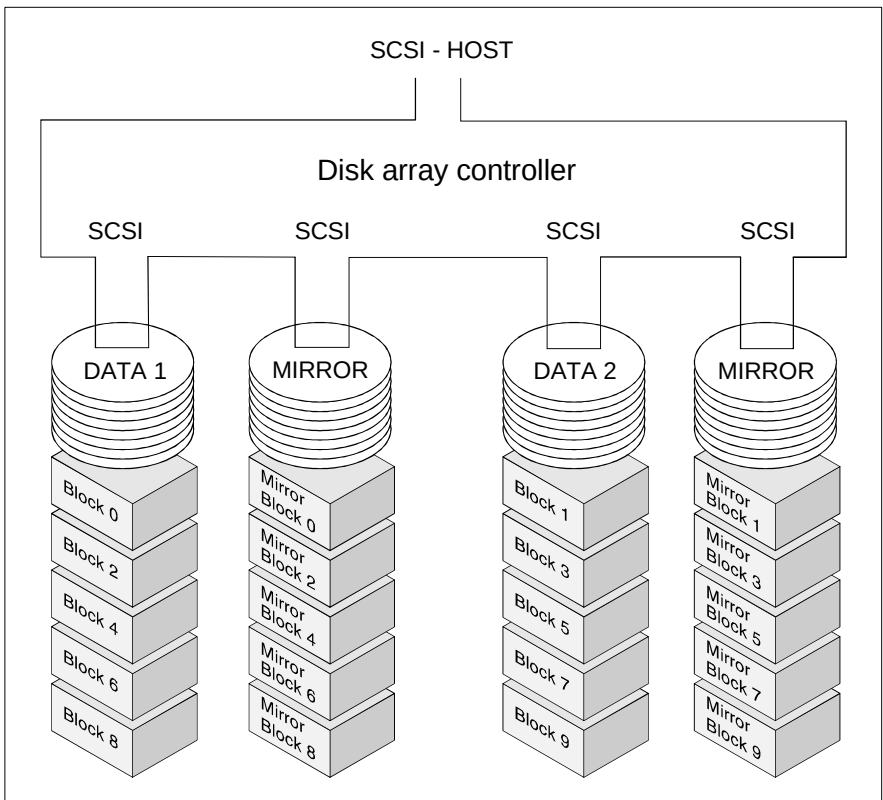


Figure 5: Schematic representation of RAID 1

If one disk fails (e. g. DATA 1), the information it contained is still available (on the mirrored disk of DATA 1). The advantage over RAID 0 lies in the higher availability of the data. The disadvantage over RAID 0 is that all the information must be present twice, so RAID 1 is highly disk-intensive. RAID 1 is therefore particularly suitable for applications which require very high availability.

Systems running under Reliant UNIX can use RAID 1 in a number of ways: externally with the help of an appropriately configured RAID subsystem, internally using disk-mirroring software or using PCI RAID controllers.

RAID subsystem in RAID 1 mode

A RAID subsystem is an external cabinet which contains a series of disks with the accompanying controllers (e.g. the RAID subsystems PXRC and PXRE). It is connected to the computer, which addresses it just like a regular external hard disk. Unlike the disk-mirroring software, RAID subsystems provide internal disk mirroring, whereby the computer addresses the (mirrored) disks in the RAID subsystem like an individual disk and the RAID subsystem then performs the mirroring. The computer therefore requires no additional computing power for the disk mirroring (see also “The disk arrays PXRC and PXRE” on page 44).

Disk-mirroring software

The Reliant UNIX operating system is based on an administration concept for hard disks, which allows the configuration of what are called virtual disks. A virtual disk functions in the same way as a conventional physical hard disk, but is made into a logical hard-disk partition by combining pieces of physical hard disks. If this process creates virtual disks which enable all I/O operations to be performed simultaneously on two or more physical devices, then you have mirrored disks.

The higher availability according to RAID 1 is here achieved by means of a software solution; the computing power necessary for its implementation must be provided by the computer itself. The computer must execute write processes in duplicate, which can lead to a performance reduction compared to unmirrored access if the number of write processes is very high. This disadvantage can be compensated for if there are many read accesses, because read accesses are only performed on one drive and this drive can have parallel access.

The gain in availability is especially high if each of the disks involved in forming a mirrored disk is addressed via a separate disk controller. In this case the data remains available not only if a disk fails but also if a disk controller fails:

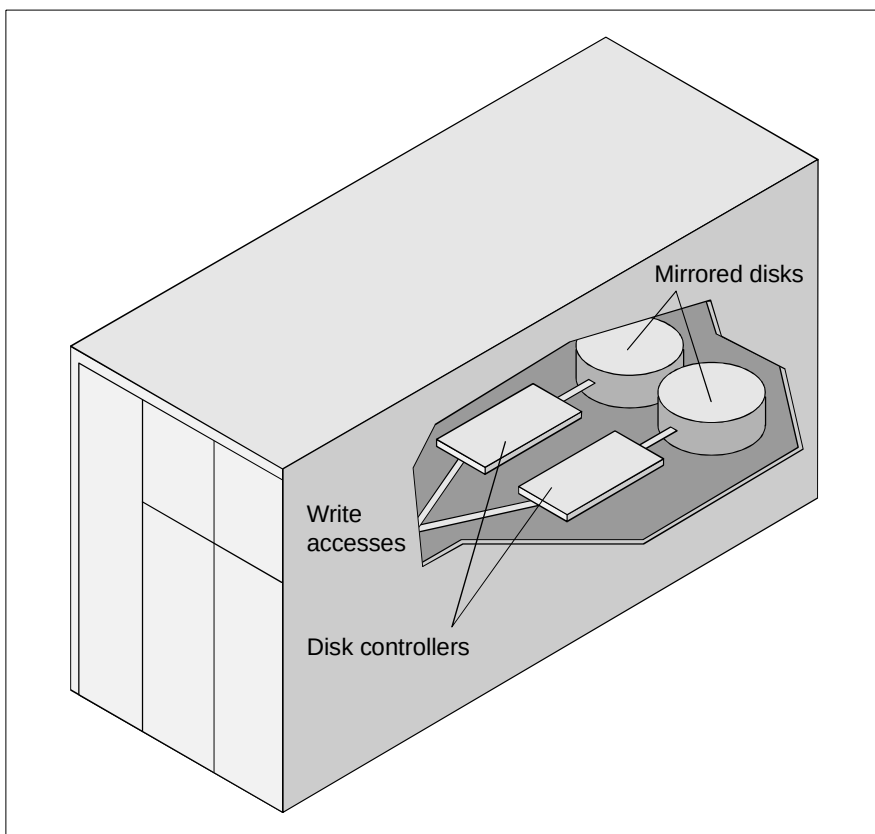


Figure 6: Internal disk-mirroring with redundant controllers

The availability can be increased still further if the various disks are also built into different cabinets.

If the hard disks can be exchanged during operation by means of OLR, the data remains available while the error that occurred is recovered. The system administrator can then reintegrate the deactivated part of the mirrored disk into the system.

Multiple mirroring is also possible. In practice, mirrored disks comprising a maximum of three mirror images are most common.

The root partition and the swap partitions can also be mirrored and thus protected against drive failures.

For further information on the disk-mirroring software, see the manuals “Reliant UNIX V5.4x - Virtual Disks”.

4.4.3 RAID 2

With RAID 2 data blocks are distributed bit by bit over several parallel drives. Generation of a complicated checksum (Hamming code) produces checksum bits on several additional disks, thereby guaranteeing the restoration of data after disk errors.

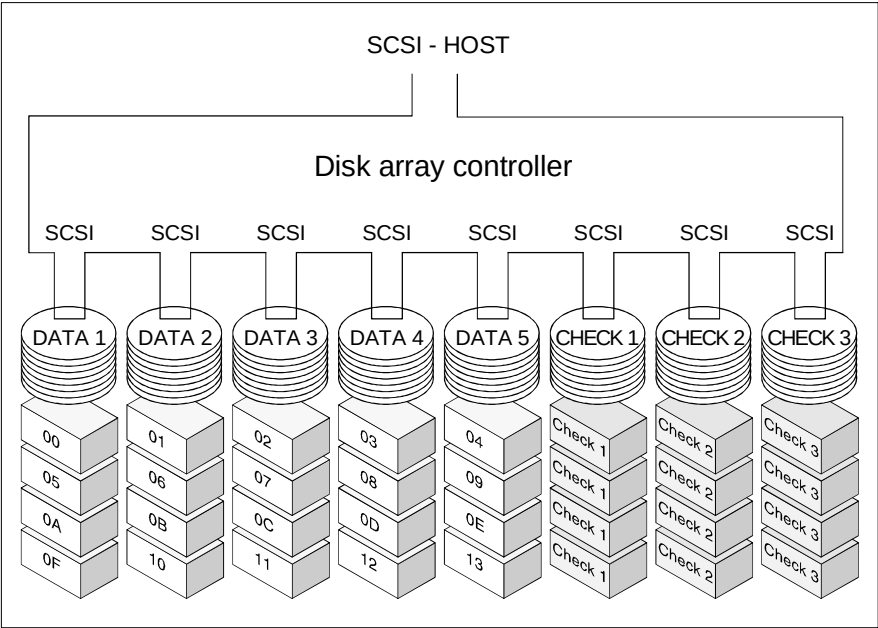


Figure 7: Schematic representation of RAID 2

The RAID 2 configuration has over RAID 1 the advantage that it requires less disk space.

RAID 2 is especially suitable for transferring large quantities of data, because for each write access all hard disks in the group are accessed in parallel, thus producing high transfer rates. RAID 2 is not so suitable for transferring small quantities of data.

4.4.4 RAID 3

With RAID 3 the data is distributed bit by bit or byte by byte over several equal-sized drives (striping at bit or byte level) and each group has only one disk for the checksums (parity drive).

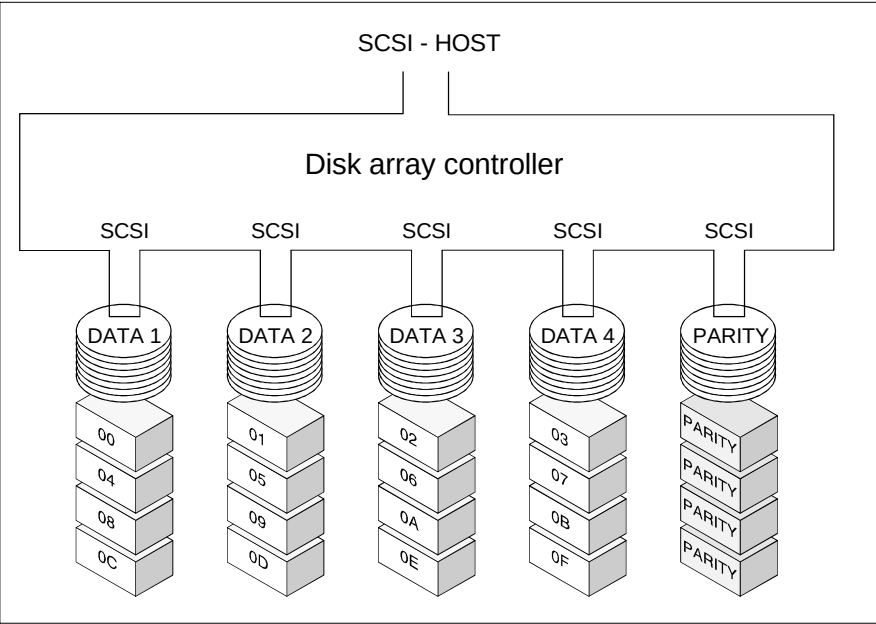


Figure 8: Schematic representation of RAID 3

This configuration has over RAID 2 the advantage that it saves some disk space because only one hard disk is required for redundancy information.

For the transfer rates of this configuration the same applies as for RAID 2, namely that particularly when large quantities of data are transferred performance improves due to parallel access via all hard disks, while parallel access is more of a disadvantage with smaller transfers. This is because the transfers to the disks occur sector by sector (512 bytes each time). In the configuration shown above, to write four bytes, for example, a total of five sectors of 512 bytes each must be read, modified and written back.

4.4.5 RAID 4

With RAID 4 the data is distributed in blocks over several drives. Here too there is only one parity drive.

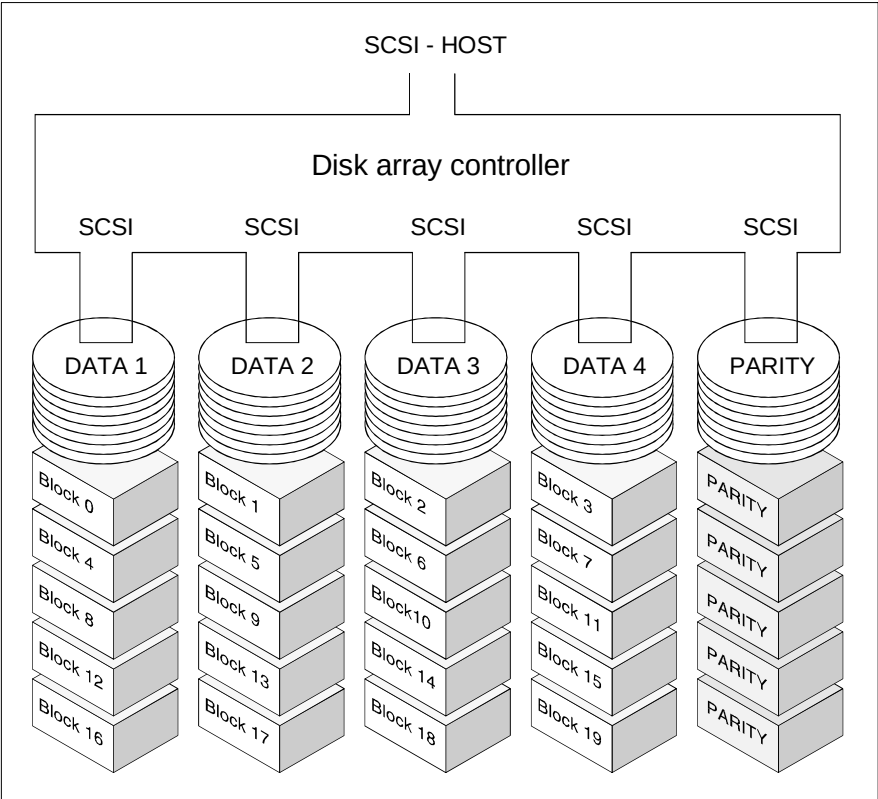


Figure 9: Schematic representation of RAID 4

The advantage of RAID 4 over RAID 3 is that with RAID 4 disk drives have to be accessed block by block rather than bit by bit or byte by byte. This means that, with small quantities of data, several parallel accesses are possible on different files, which enables high transaction rates. However, because with write accesses the parity drive also has to be accessed every time, bottlenecks may occur here in this case.

4.4.6 RAID 5

With RAID 5, as with RAID 4, the data is stored in blocks over several drives. A parity block is formed from these data blocks but is not stored on a specific parity drive. With RAID 5 the parity blocks are distributed equally over all disks in the group.

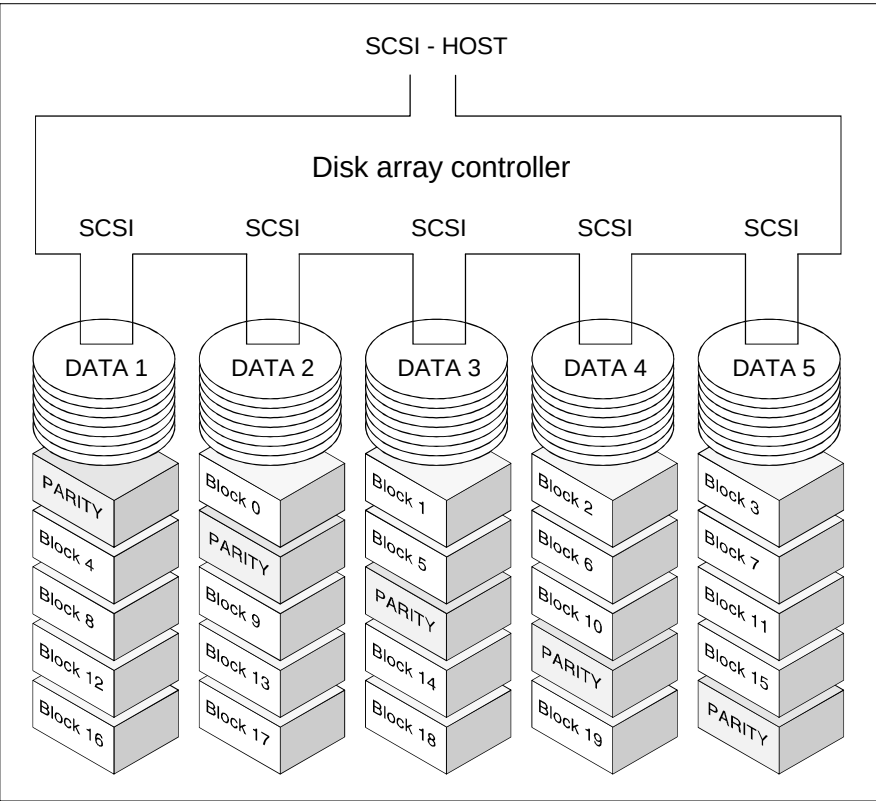


Figure 10: Schematic representation of RAID 5

The advantage of RAID 5 over RAID 4 is that when small quantities of data are transferred to the disks in parallel, a different disk contains the parity block each time. This eliminates the possibility of bottlenecks. RAID level 5 is therefore suitable for the transfer of both smaller datasets with high transaction rates and larger datasets with high transfer rates. For RM systems RAID 5 is provided by the RAID systems PXRC and PXRE.

The disk arrays PXRC and PXRE

The disk arrays PXRC and PXRE provide particularly efficient controllers with an additional battery-buffered 32 MB (PXRC) or 128 MB (PXRE) cache memory which on average more than doubles the data throughput and I/O rate compared to previous models. The available 4 GB and 9 GB disk drives, which, thanks to their hot-swap capability, can be replaced while the system is running, mean that up to 216 GB (overall) can be used on one controller. The disk arrays PXRC and PXRE each require only one SCSI ID. Individual groups of three to six disks are controlled via logical units (LUNs) and addressed as autonomous, logical RAID subsystems.

To increase availability and data integrity, you can configure a global hot spare drive, i.e. an additional idling drive which immediately takes over the function of any disk in the array that fails and automatically reconstructs its data. This also guarantees higher availability for the time until the defective device is replaced, because the failure of a second disk within the group is also absorbed.

As well as RAID levels 0 and 1, the RAID controllers of the PXRC and PXRE offer an adaptive RAID level 3/5, which automatically stores the data using the most suitable RAID procedure. This combines the advantages of RAID level 3 (high transfer rates for large jobs) with those of RAID level 5 (high transaction rates for small jobs).

As well as adding redundant components such as power supply units and fans, you can use a second controller which can then also be operated as an active controller.

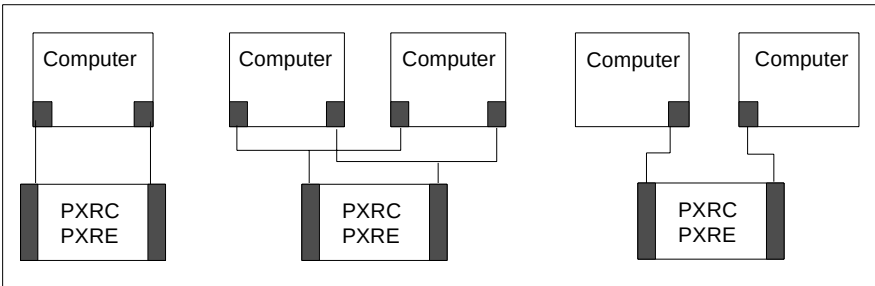


Figure 11: Ways of connecting the disk arrays PXRC and PXRE (with redundant controllers)

5 Software-related measures

This chapter describes in more detail the various ways of using software extensions to improve the availability of systems running under Reliant UNIX. The various sections describe software products which were already mentioned in the chapter “Systems operating under Reliant UNIX – achieving high availability” on page 13.

5.1 The Veritas file system (VxFS)

The Veritas file system (VxFS) is supplied together with the operating system. From V5.43, VxFS is the default file system for RM systems running under Reliant UNIX (except for the system file system, which is a file system of type ufs).

Within the context of the high availability concept, the Veritas file system is used as the standard file system although it remains compatible with ufs file systems. VxFS contains significant improvements over the traditional ufs file system implementation (Berkeley fast file system) which make it ideal for use on server systems. VxFS also permits file systems and individual files to have a size of 16 Terabytes, whereas a ufs file system may only go up to 4 Gigabytes and individual files are restricted to 2 Gigabytes.

5.1.1 Performance characteristics

VxFS offers extensions over ufs which increase the availability and performance of the Reliant UNIX operating system. The main features of the VxFS file system with regard to high availability are:

- Fast restoration of the file system integrity
- Online file system administration
- Online data backup
- Extended options of the *mount* command (extended possibilities for safeguarding data integrity, special mode for temporary file systems)

5.1.2 Restoring file system integrity

To perform structural checks after a system error, the ufs file system uses only the *fsck* command. The *fsck* command for ufs file systems checks the structure of the whole file system. Any inconsistencies found are eliminated. This process can take up a great deal of time for systems with a lot of hard disk capacity.

VxFS enables the file system integrity to be restored with the help of a modification log (intent logging). This is written to a reserved area on the hard disk and records all modifications to the file system structure which have not yet been completed.

For restoration following a system error, the VxFS command *fsck* reads this log and deletes or terminates the file system operations which were active at the time the system error occurred. The file system can then be mounted without a complete consistency check. Except for a short wait time of a few seconds, neither the users nor the system administrator notice when the file system integrity is restored using the modification log.



A complete consistency check on the file system which uses the *fsck* command is also available, and this must be used to effect repairs after a hardware error on the hard disk.

5.1.3 Online file system administration

The VxFS file system enables a so-called defragmentation during operation as well as a change in the file system size without the *mkfs* command having to be used again. In this context, online means that the file system involved does not have to be “dismounted”. The following sections describe this functionality.

5.1.3.1 Restricting fragmentation

Free hard-disk resources are initially arranged efficiently. When files are created and extended, the hard-disk resources are allocated by the system in such a way as to guarantee optimum performance. When a file system is used intensively over a long period, files become larger or smaller, are deleted and new files are created. The original arrangement of free resources is therefore lost. The occupied resources are distributed over the whole file system and unused gaps or fragments appear between used hard-disk areas. This process, known as fragmentation, impairs the performance of the system, because the possi-

bility of assigning contiguous free memory areas (extents) for files becomes smaller and smaller. This causes additional positioning of the read/write heads of these hard disks when files are accessed.

The VxFS file system provides the *fsadm* command for online system administration in order to solve the problem of fragmentation. One of the functions of the *fsadm* command is to defragment a mounted file system.

The *fsadm* command can be called as required and should regularly be started automatically via the program *cron*.

5.1.3.2 Changing the size of the file system

When a file system is created, it is assigned a specific number of blocks and inodes. Changes in the system workload can mean that file systems become either too large or too small for the intended tasks.

When a ufs file system is too small, there are traditionally three solutions:

- some users are assigned a different file system
- a subdirectory of the file system is moved to a different file system
- the entire file system is copied to a larger file system

With standard file systems, if a file system is too large, all files and directories must be removed before it can be made smaller. The solutions offered by the ufs file system require the file systems to first be unmounted, so that no user can access them while their size is being changed.

The *fsadm* command allows the size of a VxFS file system to be changed without it having to be unmounted and the users' work interrupted. However, because a VxFS file system can be mounted on only one device, it also has to be physically possible to enlarge the file system on the relevant device.

The only restriction applies to virtual disks. These cannot be reconfigured while they are active (i.e. kept open by VxFS). If a virtual disk is to be extended, the VxFS file system must first be unmounted. An additional partition can then be added to the end of the virtual disk. After a reconfiguration with *dkconfig*, the VxFS-file system can be mounted again and its size can be adjusted with *fsadm*.

5.1.4 Online data backup

The VxFS file system allows online data backup using a so-called snapshot. This method uses the snapshot function of VxFS. A snapshot of a file system is generated by means of the snapshot option of the *mount* command. The resulting snapshot file system then enables access to the data of the original file system at the time the snapshot was taken. The original file system remains active and can also be modified again immediately.

The snapshot file system is at first empty. If data of the original file system is modified after a snapshot, the original data is copied into the data area of the snapshot beforehand. If the snapshot file system is read, the copied data of this file system is supplemented by the rest of the data (as yet unchanged) of the original file system.

Data backups of snapshot file systems can be performed with all kinds of different common backup methods.

Another, more complicated method of online data backup under VxFS employs mirrored disks (see also “Disk-mirroring software” on page 38). This involves mirroring the disk that is to be backed up. For the backup the mirroring must be canceled and the second disk mounted as a new file system, which can then be backed up.

5.2 Logging V3.0

The software product Logging V3.0 is available for monitoring the (system and application) software of systems running under Reliant UNIX. Its job is to collect and analyze information on the proceedings, the state and possible error situations. The analysis can be automated, whereby tolerance limits for certain states or errors can be defined and if these values are exceeded alarms can be generated.

The product Logging offers the following advantages:

- a central system log file
this enables Logging to substantially simplify error analysis, thus reducing the time required for an error search
- uniform structure of all log files
- interfaces with long-term guarantee
- easy to program via a program library
- the log files have many and varied uses extending well beyond pure error diagnostics (thanks to individual programming)
- NLS-capable (native language support), i.e. can be set for almost any language (NLS is standardized in accordance with the X/Open Portability Guide).

A menu-driven operator interface is provided for administration and analysis purposes. Logging V 3.0 is a component of Siemens' UNIX operating system and is supplied together with it.

Logging V3.0 can be integrated into the TransView Control Center, where it also provides a uniform network-wide view of SNMP networks. Its diagnostics are reported to the TransView Control Center in the form of warnings and alarms.

You will find a full description of Logging V3.0 in the section "Evaluation and administration of logbooks" in the manual "Diagnostics and Troubleshooting".

5.3 Data backup and archiving

Siemens offers three products for backing up or archiving data on their UNIX systems: the standard product NetWorker, AMASS and DataMgr. Another possibility is the non-Siemens product HYPERtape.

5.3.1 NetWorker

NetWorker is the standard product for network-wide backups. The NetWorker architecture is based on the client/server model, whereby the clients communicate with the server via TCP/IP (or, more specifically, via a remote procedure call). As well as data backups, NetWorker also offers facilities for archiving data.

5.3.1.1 Data backup

The most important attributes of NetWorker for data backup are:

Extremely simple to operate and install

Self-explanatory menus make operation extremely simple, even for users who do not want to get to grips with the particulars of data backup. Both graphical and alphanumeric interfaces are available. Even if several NetWorker servers are in use, they can all be administered from one client.

Supports heterogeneous platforms

NetWorker supports almost all UNIX servers and workstation and PC platforms, including OS/2 and Windows NT.

Supports different backup media

NetWorker supports a wide range of magnetic tapes, tape cartridges and optical disks (including the corresponding jukeboxes), right up to large robot systems, and is thus scalable and suitable for different areas of application.

Performance

NetWorker's general process architecture allows multiplexing of incoming and outgoing data streams at the most varied levels. For example, the data streams of several clients which may be connected to the server via a relatively slow line can be combined to form a composite data stream for storage on a backup

medium, and vice versa. Of course it is also possible under certain conditions to write a data stream to several backup drives in parallel, thus drastically increasing the throughput. The separation of the data streams is an important element of effective throughput planning.

With relatively slow lines between client and server, it is advisable to use the NetWorker compression or encryption mode which has added security.

High functionality

Despite its simple operation, NetWorker offers high functionality. The time required by the backup, taking into account the network topology and use of data, can be conveniently controlled via the calendar. The objects to be backed up (files, directories on client machines) can be combined into groups and assigned to so-called device pools, i.e. groups of peripherals. As well as full and incremental backups, so-called level backups are available. With this technique the quantity of data to be backed up can be effectively reduced (thus increasing performance) without noticeably impairing the fast availability of data in the event of a restoration (following an error). So, for example, once a month you could do a full backup, once a week a level backup, based on the full backup at the beginning of the month, and once a day an incremental backup. In the event of a disaster recovery, you would then reload the last full backup, the last level backup, and all incremental backups since the last level backup.

Unmanned operation

If jukeboxes/auto-changers of the appropriate dimensions are used, NetWorker offers ideal conditions for unmanned operation via the functions and interface elements available.

Database backups

Thanks to its modular structure, NetWorker allows smooth integration of so-called application-specific modules (ASMs). In the case of the databases ORACLE and INFORMIX, this technique implements integration of the DB backup tools available in NetWorker. The solution enables the administrator to effectively organize an online and offline backup and restoration of the database with little database know-how. The administrator can use the familiar NetWorker interface for this. As data is checked on entry, operating errors are largely ruled out.

5.3.1.2 Archiving

As of version 4.1 of NetWorker, not only backup but also archiving functionality is supported. In archiving, NetWorker stands out above all by virtue of:

- its high data throughput
- the possibility of robot operation

Users may need special know-how for dimensioning the components involved in the backup and for drawing up corresponding backup plans in order to reduce the data streams or to correct their times. Siemens AG offers help with any tuning that may be required.

The following robot systems are currently supported, whereby the robot servers are provided by strategic partners and must also be obtained from these:

- ABBA systems from Grau Storage Systems GmbH, Schwäbisch Gmünd, Germany. Server software is Siemens' ROBAR or Grau's DAS/ACI.
- 3494 systems (Cashion) from IBM, Stuttgart, Germany. Robot server software can be obtained from ESG GmbH, Munich, Germany (sharing with BS2000)
- STK silos from Storage Technology, Frankfurt. Server software also from Storage Technology; sharing with mainframes (MVS, BS2000)

5.3.2 The EMASS product family

Siemens' EMASS product family is one of the most successful HSM (hierarchical storage management) products available on UNIX systems.

HSM allows various storage media (hard disks, streamer tapes etc.) to be organized as a single storage system.

While many HSM packages are for the most part implemented as a monolithic block, making expansion very difficult, the EMASS family is made up of three packages which have clearly defined interfaces to each other:

- the archiving component AMASS
- the migration component DataMgr
- the robot/silo server component VOLSERV

The archiving system AMASS is organized as a file system that can be addressed via the virtual file system layer (VFS). The user, e.g. the product DataMgr, sees only this file system; it is completely irrelevant to the user which medium of the robot peripherals its data is located on.

DataMgr organizes the transparent migration of the magnetic disk overflow of any UNIX system into an AMASS archive. When the level of utilization of a magnetic disk passes a certain level files are automatically migrated to the AMASS archive.

5.3.3 Non-Siemens product HYPERtape (company Multistream)

Because BS2000/OSD systems are supported as backup nodes, HYPERtape is available as a solution wherever heterogeneous environments exist alongside BS2000 systems and the latter are to be enlisted for backup.

5.4 Transaction monitors

Modern data processing more and more often involves client/server processing. Here processing power is distributed over numerous application servers, and in particular the presentation of the application on PCs or workstations is relocated. Pure PC networks are linked to UNIX servers, and mainframes are incorporated into the network. In these circumstances it is no longer sufficient to rely solely on the security mechanisms of the operating systems or network software involved.

5.4.1 Overview

In such an environment a logically completed operation frequently consists of a series of individual steps which often involve several systems. Let us take the example of operation in a bank. Assuming a bank customer wants to withdraw a certain amount from her account over the counter. The process this requires comprises several steps (simplified):

- entering the account number and sum at the counter terminal
- printing a withdrawal slip
- debiting the amount from the account
- registering this transaction for the account statement
- possibly automatic provision of the cash sum at the counter and paying it out

All these steps belong together logically and if the withdrawal is to be processed correctly all steps must have been executed before the transaction can be seen as complete. If the last step (paying out the money) is not performed due to a computer error, the other steps cannot be regarded as having been successfully completed. The customer would probably complain that an amount had been debited from her account although she had not received it.

To cope with such problems with complex logical operations, transaction monitors are used (also called transaction processing monitors or TP monitors). Using a transaction monitor is one of the most important measures for increasing the availability on the client side. Transaction monitors monitor the progress of sequences of logically connected processing steps. Such sequences are called transactions. The term “transaction” in this area has a different meaning from the one in database processing. A database transaction is a sequence of individual steps within the context of a database action (e.g. changing a data record), which is located at a much lower level than a trans-

action in a distributed application system. In actual fact, one or more database transactions can be elements of an individual transaction within a distributed application. To clearly distinguish between the two different types of transaction, transactions within a database will be referred to as database transactions in the remainder of this section. The word “transaction” on its own, therefore, always means a sequence of processing steps in a distributed application.

If a transaction cannot be fully completed due to hardware- or software-related problems then no part of it should be executed, i. e. it should have no effect on the user's data (e.g. if in the above example the money is not paid out, the amount should also not be debited from the bank account and no entry should be made in the bank statement).

Applications which can safely process such transactions are known as transaction-oriented application systems (shortened to transaction systems). These transaction systems consist of the following three components:

1. A fixed set of functions from application programs which may be distributed over several applications, and the data that goes with them.
2. A number of terminals, client computers such as PCs or workstations and/or data devices such as automatic cash dispensers and robots, subdivided into I/O devices (e.g. terminals) and output-only devices (e.g. printers).
3. A transaction monitor as the core, which manages operating resources 1 and 2.

Individual actions such as

- initiating a request via a message from a terminal,
- accessing data,
- outputting a result, or
- communication between applications

are only possible via services of the transaction monitor. There is no direct communication between components 1 and 2 above without intervention by the transaction monitor. The sequence of a transaction is fully controlled by the transaction monitor – the applications do not need to concern themselves with this.

Transaction systems are typically used in areas involving large numbers of information, booking and data entry processes. Well-known examples include:

- Account information and booking of sums of money at the bank
- Buying tickets and reserving seats
- Reserving flights, ferries and hotels at travel agents
- Order entry and processing, optimizing inventory management in commercial companies
- Controlling supply to assembly lines, entering production data in industrial companies
- “paperless” administration at insurance companies, savings and loan associations etc., and administrative processes (registration, vehicle licensing, land registry proceedings) at public authorities

Application systems which become transaction systems through the use of a transaction monitor offer not only the immediate advantage of always-consistent data but also a substantially quicker resumption of work following a system error.

After an error during a transaction, the end user can resume his dialog with consistent data - there is no data loss or duplication. He does not need to reenter the data he entered before the end of the last transaction. The amount of time saved here can be considerable.

There are several transaction monitors available for Siemens UNIX systems: Siemens' own openUTM, Tuxedo from BEA, etc.

The following section briefly introduces openUTM, the transaction monitor from Siemens AG.

5.4.2 openUTM

The transaction monitor openUTM provides outstanding middleware technology for the most varied function requirements and user profiles in distributed applications and has (according to the Gartner group) the best functionality on the market. openUTM can be used as a basis for modern solutions such as mobile computing, workflow management, Internet and WWW connections and object orientation in a distributed transaction processing environment.

The range of heterogeneous, distributed execution environments for which openUTM can be regarded as a framework stretches from communication and processing with transaction-protection to client/server processing across different types of network. Due to the identical availability of all supported platforms, the user has a uniform programming environment, which helps to reduce the time and cost of application development.

openUTM is available for platforms from all major hardware providers and runs, for example, on hardware from Siemens, Pyramid, HP, SUN, IBM, ICL or BULL, and under the operating systems UNIX, BS2000, Windows NT and Windows.

5.4.2.1 Services of openUTM

The universal transaction monitor openUTM is the core of a transactions-oriented application system, the openUTM application, as defined by DIN standard 66265. openUTM is linked to the individual application programs to form a so-called openUTM application program. The individual application programs are called openUTM subprograms. The core openUTM then performs the following tasks:

Ensuring data consistency

openUTM guarantees the consistency of the user data by – possibly in conjunction with a database system – making sure that a transaction with all accompanying data modifications is executed either completely or not at all. Synchronization with the database is handled by the database coupling interface XA defined in X/Open.

After a system crash openUTM automatically brings its data and the data of the database system up to a common consistent state. When the openUTM-application is restarted, all users can continue their dialogs at the point they had reached by the end of the last successful transaction.

Controlling the communication

In distributed transaction processing openUTM controls the communication between the applications and the execution of the transactions. In so doing, openUTM allows distributed transaction processing over several Siemens UNIX systems or between Siemens UNIX systems and BS2000 or MVS[®] systems, as well as to client systems under Siemens' UNIX operating system, UNIX or MS Windows[®] using the product openUTM-Client.

Connecting CMX provides a simple option for other transport-system applications, e.g. in MacOS or OS/2 or in data devices such as POS terminals or robots, to communicate with openUTM applications running under Siemens' UNIX operating system, Windows NT or BS2000 computers. A transaction-protected and restartable link from openUTM applications on Siemens UNIX systems with CICS or IMS applications is also possible.

Managing operating resources

openUTM ensures that many end users can use the same operating resources (processes, programs, data, printers etc.) at the same time. This allows the same processing operation to be executed several times in parallel.

openUTM provides special call interfaces which applications can use to request openUTM system services. One of these interfaces is identical to the corresponding interface in BS2000, KDCS (conforming to the German DIN standard), which simplifies the porting of openUTM applications between BS2000 and Siemens' UNIX operating system.

Other interfaces are XATMI and CPI-C, as defined in X/Open for the communication of transaction applications. Together with the communication protocol OSI TP from ISO, these interfaces also allow communication with applications running in other systems and controlled by other transaction monitors. The other systems must, however, use the same interfaces and protocols (e.g. Tuxedo/OSI TP applications).

Communication with IBM CICS/IMS-TM applications is via gateways and is such that the CICS/IMS-TM application does not require any special add-ons.

Simple administration of the transaction system

openUTM provides an easy-to-use generation and administration interface. The openUTM administrator can modify certain configuration data during operation.

Extensive data protection functions

All users of openUTM applications are subject to access controls. Moreover, it can be specified for each user, which subprogram runs he or she can start and from which clients.

Support for client-server architectures

Elements of applications which lie outside transaction brackets (e.g. data preparation, screen I/O, validity checks) can run on a client that is outside openUTM. The actual processing of a request (e.g. database accesses) occurs in the server application under openUTM control. The clients can run on a UNIX system, a workstation or a PC.

Message queuing

openUTM offers active message queues, which permit especially long-running or asynchronous transactions. They are particularly suitable for many modern applications such as data warehousing, mobile computing etc.

5.4.2.2 The most important services of openUTM for high availability

openUTM provides the following contributions toward increasing the availability of Siemens' UNIX systems:

Continuous operation (7x24-hour operation)

openUTM enables uninterrupted operation because it allows programs to be exchanged during operation. Moreover, errors do not cause the entire application to crash (there is no single point of failure).

Time savings with system errors

After system errors (e.g. network failure, loss of connection, a crash of a participating system) openUTM automatically restores the state as it was at the end of the last successfully completed transaction. The process can then be resumed. By synchronizing its own transactions with the transactions they contain from the database systems used, openUTM enables a coordinated restart of process and application, which guarantees that the data and processing states are consistent. This also applies if several different databases are being used in the transaction.

Even if the whole application crashes, openUTM updates the aborted processes to the state of the last end-of-transaction; they can then be resumed from that point. All asynchronous jobs or print jobs that have not yet been processed are retained.

There is, therefore, no need to check which operations have already been executed and which have not. This can save a great deal of time in the case of very large transaction systems with several hundred or even thousand clients.

Duplication of openUTM files

The openUTM files can be split up and duplicated. This enables an application to continue running even if a write or read error occurs in a file.

Integrated data protection functions

In openUTM you can specify which users and which clients can call certain services. For the authorization check when accessing openUTM applications, a badge reader or chip-card reader can be used.

If openUTM is working in synchronization with a database system, the latter's protection mechanisms can also be used. In client/server environments the access protection is guaranteed by the product openUTM-Client.

This reduces the danger of malfunctions due to corrupted data.

Use in cluster systems

Its performance characteristics make openUTM predestined for use in cluster systems (see the chapter "Cluster systems" on page 61). If one system should fail, the openUTM application can be started automatically on the cluster's remaining intact system. Using its restart mechanisms, openUTM then restores the system to a consistent state. Clients can be included in a restart, so that operation can continue without losses and with safeguarded transactions.

6 Cluster systems

When particularly high availability is required, the best solution is often to use cluster systems.

Cluster formation also makes it possible to compensate for scheduled downtime, e.g. when upgrading to a new operating system.

The following sections describe the basics of cluster formation, the different types of cluster and other aspects of cluster systems.



This chapter will provide you with basic information on the subject of “Clusters”, but chapter 7 contains details of a special Siemens solution, known as “Reliant Cluster Server II” (RCS II). The RCS II is not an individual product, the name designates a specified set of products and components that are geared to a variety of business requirements.

6.1 Fundamental structure

A cluster always comprises at least two RM systems of type RM400 or RM600. The individual systems in the cluster will be referred to as *nodes*. Each of these, seen in isolation, is an arbitrarily configured symmetrical multiprocessor system, whose own availability can be increased by means of the measures described so far in this manual. By means of special hardware and software add-ons, several such systems with their own peripherals and external disk subsystems are combined to form a cluster. Redundancy is thus further increased.

The fundamental principle is that even if individual components or a whole node fail, the basic functionality of the cluster is retained. This is possible with a cluster because the failure of any one component in any one participating system does not alone cause the whole system to crash. In other words, there is no single point of failure.

For this reason the disk subsystems, at least, must be external so that they are available to more than one node. Furthermore, special software must be used on the nodes which recognizes malfunctions and reacts accordingly. There are two types of access to shared peripherals:

- exclusive, switchable access (failover cluster) for any desired applications, and
- parallel access (shared database cluster), for instance to an ORACLE database (ORACLE Parallel Server).

To enable the software to respond immediately and flexibly if individual components or an entire node should fail, the nodes in the cluster communicate constantly with each other (sign-of-life monitoring). This allows the failure of a node to be recognized immediately.

This chapter will first explain how a high-availability cluster (failover cluster) is constructed, describe the two high-availability monitors OBSERVE and Reliant Monitor Software (RMS) and then examine the special case of parallel access to an ORACLE database.

This is followed by a section which describes how to integrate OPS into a failover cluster, i.e. how to combine the two types of cluster.

Finally, you will be given a number of hints regarding the necessary hardware configuration for cluster systems.

6.2 Failover clusters

A high-availability or failover cluster is a cluster in which each participating node works actively on its own tasks but takes over another node's tasks as well if that node fails. On RM systems, the software which is provided for implementing failover clusters consists of the two high-availability monitors

- OBSERVE and
- RMS (Reliant Monitor Software).

The first of these enables affected applications to be switched to a different cluster node if one of the participating cluster systems should fail, and the second can switch server resources over to a different cluster node.

The term resource is used for any component which is to be monitored, for instance a mirror disk, a RAID subsystem, a printer, etc.

During normal operation, the switchable applications and resources are specifically assigned to one of the cluster nodes and can be used only by this one. Only if this node fails will they be switched to a different one and then used exclusively by that one.

Each cluster node is at the same time also a productive system, i.e. in normal operation all the nodes work independently on various tasks using their own application data and other peripherals. To protect all the cluster nodes from failure, the cluster should be created symmetrically, i.e. in a configuration with two nodes, each of the nodes must have access to all the necessary resources, and it must be possible to switch these over to the other node if necessary.

6.2.1 General features of a high-availability monitor

6.2.1.1 Sign-of-life monitoring of the nodes

At regular intervals the nodes in a cluster exchange sign-of-life messages. This exchange, via two independent redundant connections, makes it possible to distinguish between the failure of a system and the failure of a connection. If the software registers that there are no further sign-of-life messages it will conclude that the corresponding computer has failed.

In this case another system will initiate taking over the corresponding applications and peripherals (external disks, terminals, printers, etc.). When the failed system is available again, the takeover can be reversed (either automatically or by command from the administrator).

6.2.1.2 Individual response scripts

Application systems are controlled by executing predefined shell scripts. These response scripts, which must be made known when the system is configured, enable the user to initiate individual responses to malfunctions, such as transferring an application which has encountered a problem onto some other machine, or switching over peripherals.

6.2.1.3 IP address management

To ensure that the transfer of tasks to another cluster node remains transparent for whoever is using the resources, high-availability monitors make use of a mechanism which automatically reassigns IP addresses in the event of a failover. Access from the user application always goes via the same IP address.

6.2.2 Redundant use of serial peripherals

Not only disk subsystems but any serial peripherals, such as terminals or printers, can also be used redundantly. This means that these resources will also remain available if one node in the cluster fails.

In this case the switchover is effected using TACLAN as the switching mechanism. The figure below illustrates the fundamental principles of switching over serial peripherals:

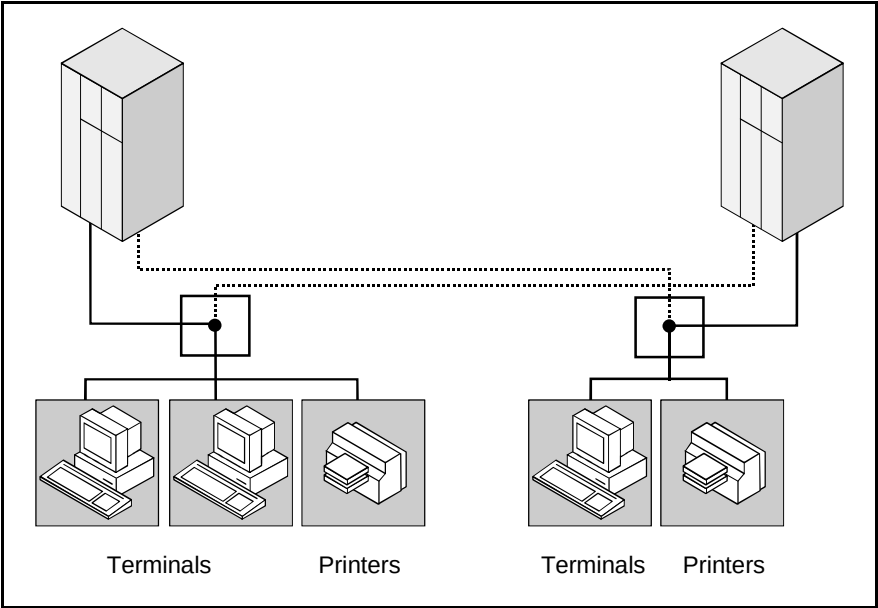


Figure 12: Switchover of serial peripherals

If one of the two nodes in the figure fails, the other can take over the terminals and printers of the failed node. The users of the failed system, therefore, do not have to find a new terminal on which to continue their work.

6.2.3 Restrictions placed on applications

To guarantee the availability of applications during cluster formation, the applications which are to run in a failover cluster must satisfy certain conditions. However, this should not cause any problems because these conditions should in principle apply to all applications, even if they are not used in a cluster environment.

The following rules must be observed for applications which are to run in a cluster:

- It must be possible to install the application on the hard disks of an add-on cabinet, and it must also store its internal processing and configuration data there, otherwise it cannot be switched over to another system.
- The application must not use any system-specific attributes (e.g. system name) in its internal data if processing is to continue on the other system after a switchover.
- The application should be able to be terminated and restarted within a few seconds in order to keep the switchover time to a minimum.
- When it terminates, the application should release all file systems used so that these can be dismounted.
- If the response scripts for the switchover are to be able to function correctly, it is important that all the commands which make up the application return a zero exit code when successfully executed and a non-zero exit code when unsuccessful.

6.2.4 The OBSERVE software

OBSERVE is a monitoring system for dual computer configurations, with sign-of-life monitoring and switchover control for switchable peripherals (such as disks, terminals or printers). After a system crash, the redundant system switches the peripherals over to itself and takes over the tasks of the failed system (see also the manual “OBSERVE”). The sections “Switching over disks” on page 68 and “Logical switchover of IP addresses” on page 69 will give you more details of the various possibilities for a switchover.

The nodes use redundant communication connections to maintain mutual sign-of-life monitoring, which is effected by exchanging appropriate sign-of-life messages. If no such messages are received from a partner system for a certain length of time, this will be interpreted as a system failure and the private peripherals of the failed node will automatically be switched over to the surviving node. Additional responses to such a failure, such as restarting specific applications on the replacement system, can be specified individually and flexibly by means of user-defined response scripts. The applications on the node which is taking over can continue running undisturbed during this phase, there is no need to restart the system.

Within a system also, malfunctions can be detected by means of the monitoring programs (observers). If only this functionality is used, OBSERVE can also usefully be run on a single-computer system.

To make sure the clusters are properly monitored, and to distinguish a communication failure from a system crash, the nodes involved are monitored for signs of life on the LAN and simultaneously over serial connections. In the event of a system crash the resources administered by OBSERVE will be switched to the other system.

The switchover time, which is the interrupt time required to completely switch over the resources when a node fails, consists of two components:

- the time required for switching over the resources, which is of the order of a few seconds, and
- the time for carrying out recovery measures on the private data that has been switched over.

As regards the second of these components, the user has a decisive influence on the time required for these measures. Depending on the type of file system used (ufs or VxFS) and the sizes of the disks, the file system check, for example, can take anything from a few seconds to 20-30 minutes.

Another way in which the user can influence the switchover time is in choosing the size of the log files in a database application. If you use very large log files, so as to improve performance, a switchover will take correspondingly longer than with small log files because of the numerous recovery procedures.

The degree of availability of an OBSERVE cluster is therefore largely determined by the user.

OBSERVE offers the following functions, in addition to the normal functionality of high-availability monitors as described in section 6.2:

Command interface and administration menus

Individual commands are provided for controlling the monitoring process and outputting information. The administration menu provides a convenient superstructure for all the functions offered by the command interface.

Simple configuration

In a configuration file the user specifies the parameters required for the sign-of-life monitoring, as well as for those components of the system that are to be monitored, and also specifies the changeover switches and scripts which are to be started by OBSERVE in response to malfunctions.

Switchover control

OBSERVE switches all the configured changeover switches to the required position and starts user-specific response scripts.

Standard observers and observer API

Prefabricated standard monitoring programs (observers) are supplied for printers and processes. OBSERVE comes with five standard observers. If one of the monitored objects fails or its status changes (e.g. failure of a process), a previously defined response script is initiated.

A C-API is provided for creating individual observers which are to monitor any desired objects. An object can be an item of hardware, such as a changeover switch, or it can be a process. The API contains functions to facilitate communications between OBSERVE and the special observer program.

6.2.4.1 Switching over disks

If it is to be possible to switch disks belonging to one system over to another, they must be linked to an external disk subsystem. This can be a peripheral cabinet or a RAID subsystem (see also “The disk arrays PXRC and PXRE” on page 44).

Peripheral cabinets are connected via a SCSI changeover switch (SCSI switch) both to the node that normally uses these disks and to the one which is to take over these disks should the first one fail. RAID subsystems do not require an additional SCSI switch; instead there must be a second RAID controller installed which is connected to the second node. The OBSERVE software ensures that this second node only accesses the RAID subsystem if the normally assigned node has failed.

Moreover, the RAID subsystem can also be “split up” between two nodes so that in normal operation each node is only assigned a share of the RAID subsystem. Then if one node fails, the other takes over its share also.

Basically, the configuration for switching over private disk subsystems looks like this:

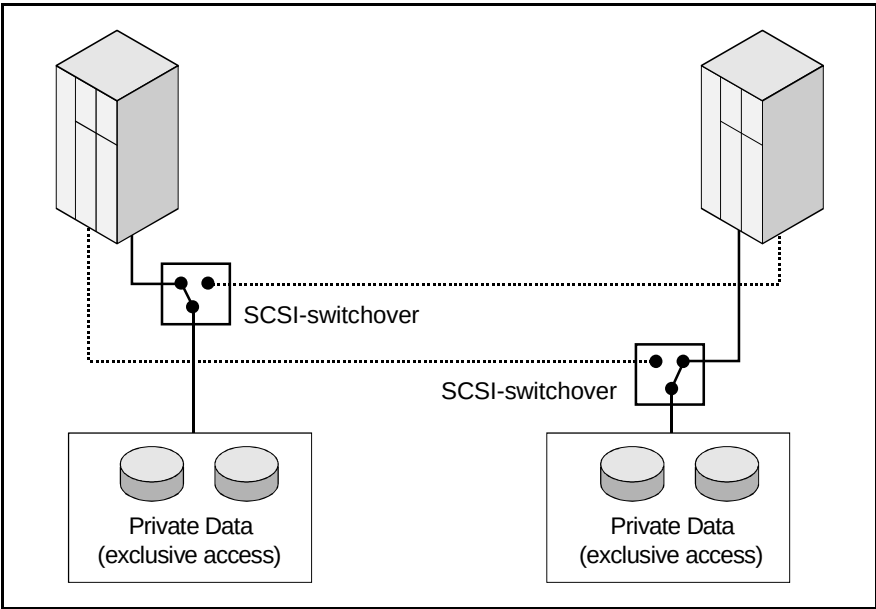


Figure 13: Switchover of private disks

If one of the two systems in this figure fails, the second one can take over the disks of the failed system in addition to its own by switching over the external disk subsystems.

The users of the failed node can resume their work via the second node. However, if only the external disks are switched over, users whose terminal (say) is connected to the failed node will have to change to a different terminal.

To prevent this from happening, serial peripherals can also be switched over.

6.2.4.2 Logical switchover of IP addresses

OBSERVE clusters are often also networked with other computers and communicate with these via TCP/IP. In most cases, the computers in the OBSERVE cluster function as servers and the other computers in the network as clients which use the functions of the server.

When the clients communicate with the cluster nodes, the following problems arise:

- The clients do not always notice immediately if the cluster node has failed
- If the server fails and its function is taken over by a redundant system, the client must turn to this system.

At this point the OBSNET software intervenes and offers the following possible solutions:

- Switching over a network address by using a second network interface on every server
- Informing the clients when a server fails
- Ascertaining the current state of the server from the client.

With the first solution, the server monitored by OBSERVE and the second server are each equipped with a second network interface. On the monitored system, one interface is used to monitor and log in to the network, and one to access the clients (the requisite IP address for this is also called the service address). When the second server is booted, only the first interface is configured and parameterized with the Internet address corresponding to the server name.

For configuring the second interface there is a command which can be used to assign the address of the failed node to the redundant cluster node. The substitute (redundant) server can then be addressed in the network under the

service address of the failed server. In the case of a restart the client applications will then find their server again under the old address. The fact that this is now a different system is not noticed by the clients.

If the monitoring is to be symmetrical, i.e. mutual, there will have to be three network interfaces on each of the two servers.

For this solution, OBSNET need only be installed on the servers. This solution is especially recommended when OBSNET cannot be used on clients (e.g. when the clients are not Siemens systems running Reliant UNIX).

For the second solution, a command and programming interface is provided, via which an application on the client can query the current state of all systems of the cluster configuration.

For this purpose, OBSNET must also be installed on the client. With the help of OBSNET, daemons are started on all clients. If a cluster node now fails, the redundant node can inform the daemons on all clients known to it of this failure.

Furthermore, applications on a client can tell that client's daemon further actions which are to be performed if a server fails. This can be used to effect an automatic restart of an application. A C interface and a command interface are provided for specifying these actions.

Binaries of OBSNET are available for Siemens' UNIX operating system. So that the OBSNET functionality is also available for clients that are not Siemens systems running under Reliant UNIX, the OBSNET client part is supplied as source code in C. This enables the user to port this client part onto the desired destination systems.

You will find further information on OBSNET in the manuals "OBSERVE" and "OBSNET".

6.2.5 Reliant Monitor Software (RMS)

Like OBSERVE, RMS is a high-availability monitor for monitoring computers and applications in a cluster. The fundamental principles which it uses to maintain availability, that is, implementing sign-of-life monitoring via redundant communications connections, reacting to the failure of a node by using predefined response scripts to restart specified applications on a replacement computer and the use of monitoring programs to monitor local objects, are the same as those used by OBSERVE (see also the manuals “Configuring and Administering RMS” and “Operating RMS”).

RMS also offers the following additional facilities:

Symmetrical takeover in clusters consisting of more than two nodes

Each of the nodes in a cluster (which may consist of up to eight nodes) can take over the functions of a failed node. In order to achieve this, it is essential that all the nodes in the cluster have access, at hardware level, to all the critical devices, most particularly the disks.

Multihosted SCSI

All the nodes in a cluster have competitive access to all the disks comprising the common data set, via multihosted SCSI. This obviates the need for hardware switchovers.

Switchover only under software control

In the case of an RMS cluster, switching over and exclusive access to the data are controlled only by software.

Cluster console

To prevent critical situations such as temporary system hang-ups from endangering the consistency of the data, each cluster possesses a cluster console (RM 200, UNIX PC LAN console) which functions as a supervisor.

If no further sign-of-life messages are received from a particular node, which is interpreted by other nodes as indication that that system has failed, the affected applications, file systems and other relevant resources will not be switched over to a replacement server until the cluster console has confirmed the failure. To prevent inconsistencies arising, the cluster console must also ensure, before giving such confirmation, that a system which is temporarily hanging will not subsequently revive.

Flexible monitoring granularity through object-orientation

While OBSERVE focuses on monitoring the partner system, RMS centers on monitoring individual resources (objects). The availability of relevant data and applications is not only seen to be disrupted by a system failure, but also by the failure of (say) one of the system's SCSI controllers.

Switching the relevant application over to a replacement server, on which the relevant SCSI ports are intact, restores that application's availability without immediately having to switch over all the applications on the server which is partially impaired due to the defect in its SCSI controller.

RMS is able to maintain availability with a finer granularity than OBSERVE. You can configure your own specific requirements to determine the actual degree of granularity.

Graphical user interface

As well as a command interface, there is a graphical user interface for administration purposes, which can also be used to view the current status of the overall configuration, including all the monitored resources.

Redundant Ethernet connection

Monitoring of the cluster is guaranteed by sign-of-life monitoring of the participating nodes by means of a Virtual Ethernet. This is a logical LAN which interconnects all the computers and is implemented via at least two independent LAN connections.

IP aliasing

To ensure that the transfer of tasks to another cluster node remains transparent for whoever is using the resources, RMS uses a method known as IP aliasing, whereby one network interface is assigned a number of IP addresses (aliases). This makes it possible for a computer to react to inquiries that are directed not at its own IP address but to any one of its alias addresses.

6.3 OPS clusters (ORACLE Parallel Server)

In an OPS cluster all nodes access a common ORACLE database at the same time (parallel access to a global database). An OPS cluster thus combines the performance and availability of a multiple-server system with the consistency of a separate database. As well as the general advantages of scalability and higher availability of a cluster, this provides the following benefits:

- Higher performance:

Because the database is accessed via several nodes, the load can be distributed equally, resulting in higher performance.

- Lower costs:

Instead of several databases on different servers having to be administered, there is now only one database to be managed and maintained.

All nodes of a cluster have competing access to all disks of the shared data of the database via multihosted SCSI. Only the database data can be stored on the shared hard disks of the OPS cluster, private file systems cannot. If a node of the cluster fails, the database remains fully functional and is available without interruption via the remaining nodes; the only exception is the accesses which were performed last via the failed node and which had not been completed. These accesses are fully reversed by OPS so as to ensure the integrity of the database.

To guarantee that the cluster is monitored, the participating nodes are monitored for signs of life on the Virtual Ethernet. This consists of a logical LAN connection between the systems, which is implemented via at least two independent LAN connections.

Together with the functionality of OBSERVE/OBSNET which was described in the previous section, OPS also allows you to integrate a high-availability database into a failover cluster (see also the section “Integrating OPS into a failover cluster” on page 76).

The following figure illustrates the fundamental structure of such an OPS cluster in much-simplified form:

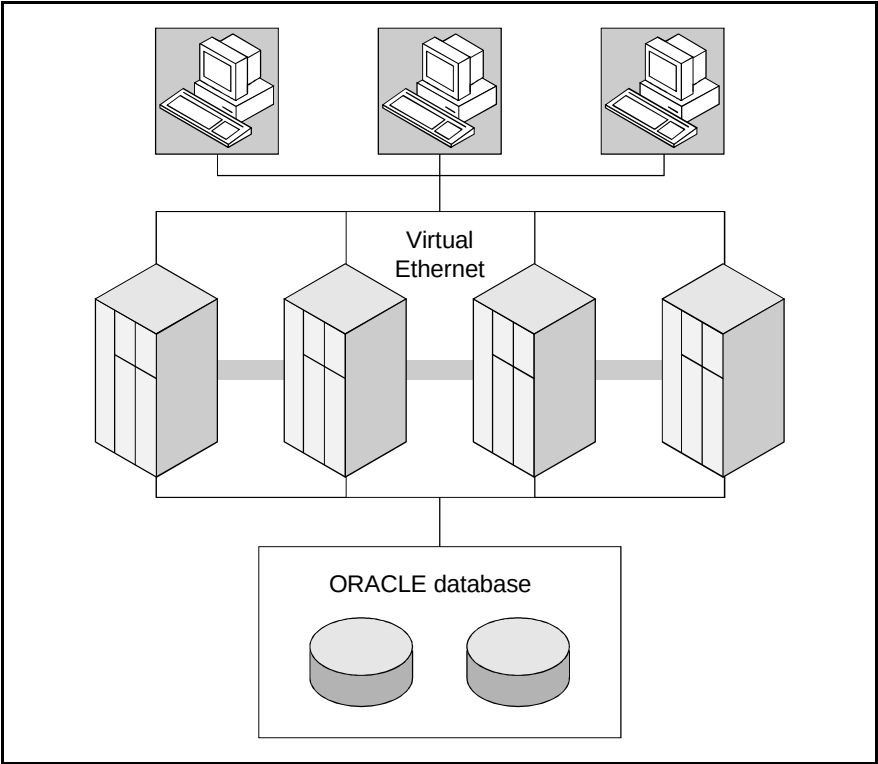


Figure 14: Fundamental structure of an OPS cluster

The following subsection provides a summary of the most important information on OPS clusters. For more detailed information, especially on the attributes that are mentioned, see the manual “ORACLE Parallel Server for Siemens RM600 Clusters - White Paper”.

The software ORACLE Parallel Server

The required software for an OPS cluster comprises Siemens' UNIX operating system, the database software ORACLE Parallel Server and a range of additional packages, called "SIDLM cluster software". This contains packages for administering distributed accesses to the shared database, the distributed lock manager (DLM), the Virtual Ethernet software, which is responsible for the communications connections between the nodes, and software for the cluster console, which enables the cluster to be centrally administered and monitored via the cluster console.

The ORACLE Parallel Server software, which is specially optimized for database clusters, offers a range of features for increasing availability and performance, which are especially tailored to use in a cluster:

- Shared buffer caches, which help prevent superfluous disk accesses. Data blocks are loaded into these buffer caches, which enables several database accesses to the same data block without additional disk accesses.
- Parallel management of the buffer caches (parallel cache management) ensures that even parallel accesses by several nodes to the same data block are consistently processed.
- Together with transaction isolation, the internal access control (concurrency control) ensures that accesses to different records within a data block can be executed in parallel.
- ORACLE Parallel Server allows the automatic restoration of database integrity if a node fails. This ensures that the failure of a node does not leave behind "half-finished" transactions, even during a database operation.
- ORACLE Parallel Server also offers the possibility of unlocked read access to the database, inserting data records from different nodes in parallel, generating sequence numbers that are unique across the whole cluster (e.g. for unique invoice numbers), and performing online backups.

Further information on OPS and the OPS cluster is provided in the ORACLE Parallel Server documentation and in the manual "SIDLM V1.2 - Installation and Administration Guide".

6.4 Integrating OPS into a failover cluster

The two architectures failover cluster and OPS are fully self-sufficient, i.e. they can be used independently of each other. With most applications, however, it is highly beneficial to combine both architectures and use OPS as a special application within a failover cluster.

With ORACLE Parallel Server, emphasis is on the constant availability of the database. A high-availability monitor, on the other hand

- is completely independent of the file systems and database systems used
- guarantees parallel operation of the individual nodes of the cluster without overhead, provided the applications and data are distributed appropriately
- allows any desired objects to be monitored
- supports IP address switching and the switchover of other peripherals such as terminals or printers.

The figure below which shows the interaction of OPS with OBSERVE will illustrate this. To give you a better overview, the switchover is only shown for one system in each case, although of course every system may be equipped with private data or terminals which can be switched over.

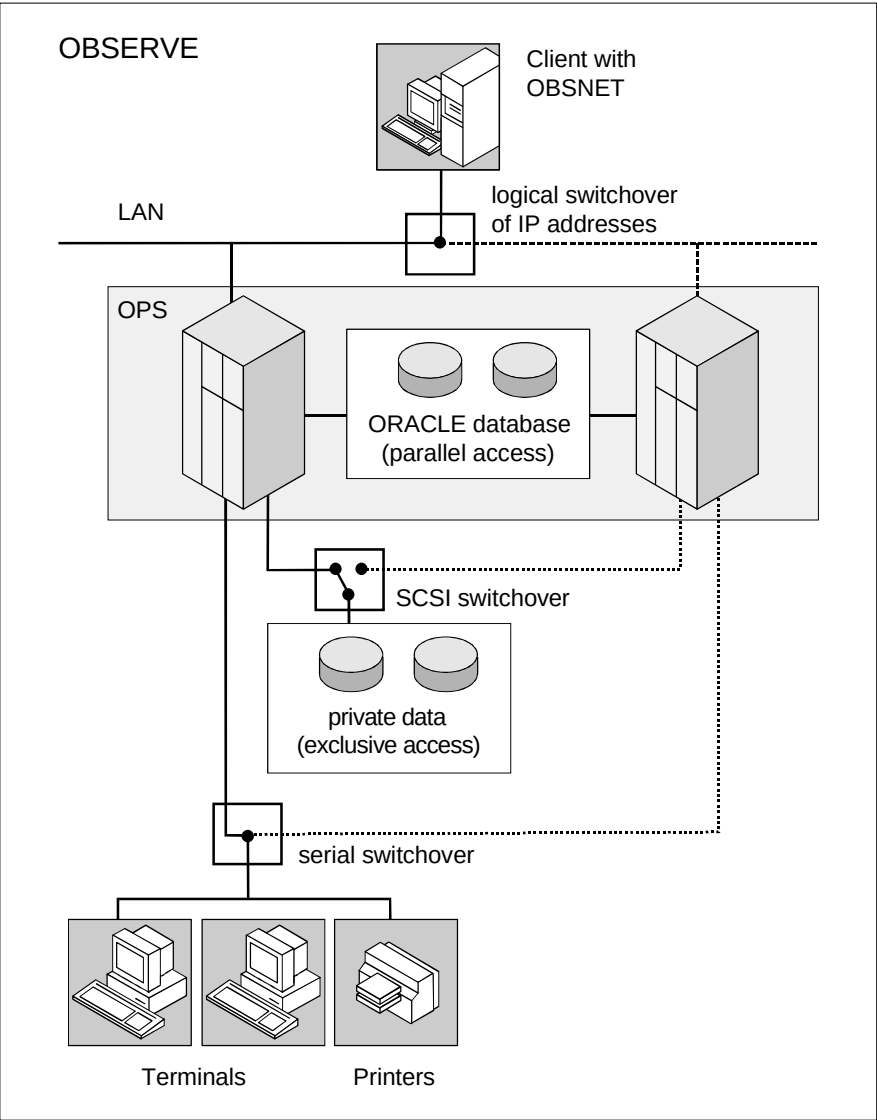


Figure 15: Interaction of OBSERVE and OPS

6.5 XPS Clusters (INFORMIX Extended Parallel Server)

XPS is based on what is called “shared nothing” architecture. As its name implies, XPS does not require any shared resources (such as disks); each node administers only the resources assigned to it.

This means that there is no need to implement a lock manager, nor do we need common parallel disk accesses. The entire database is distributed between the cluster nodes so that each server is responsible for one particular section. In such cases we say the database is “partitioned”. From the point of view of the applications, however, the entire database remains a self-contained entity, just as it does with OPS.

Requests for data which belongs to a different node are automatically forwarded via the *cluster interconnect* (CI) (see also chapter 7).

If a node fails, the data belonging to it will be temporarily unavailable. The disks will have to be switched over to one of the remaining nodes and taken over by that node's INFORMIX instance. This procedure is controlled by RMS.

6.6 SPS Clusters (Siemens Parallel Server)

In clusters of type *Reliant Cluster Server II* (see chapter 7), the use of the parallel database solutions ORACLE (OPS) and INFORMIX (XPS: extended parallel server) is supported by the product SPS together with the Reliant UNIX operating system.

SPS has the task of implementing the cluster communications of the database system, for which it uses a redundantly installed private network. The hardware and software components for this must be able to exchange large volumes of data extremely rapidly and with the utmost reliability, so that the entire cluster can, if necessary, survive the failure of any individual component. In the interests of maximum scalability, SPS uses a protocol geared to short latencies for passing messages between the nodes.

As the two database systems OPS and XPS have quite different requirements as to the number and length of the messages that have to be exchanged, and since these are also highly dependent on which applications are used, SPS has been given an extremely flexible architecture, which is largely independent of the underlying cluster interconnect. This makes it possible to support different

cluster interconnect (CI) technologies by replacing the hardware and the driver layer, while the remaining components of SPS can be retained unchanged. The product supports *Fast Ethernet* and *SCI* (scalable coherent interface).

6.7 Hardware configuration for clusters

This section describes the components comprising a cluster. All specifications given below are based on a configuration with two nodes; for configurations with more nodes the corresponding number of additional components must be used.

You can obtain detailed specifications on your hardware configuration from your sales representative and the relevant sales literature.

6.7.1 General requirements for clusters

Every cluster, whether a failover cluster or an OPS cluster, has certain requirements regarding the hardware used. This section will attempt to present a brief overview of these requirements for all types of cluster.

6.7.1.1 Cluster nodes

Every cluster consists of at least two cluster nodes. The maximum number of cluster nodes with OPS is four, with OBSERVE it is two, and with RMS eight. Possible systems are the RM600 and the RM400.

When selecting a suitable model, there are certain aspects you must consider in order to determine the required I/O expandability and the CPU and storage possibilities:

- The expected number of users. (With OPS you should take into account the number of users for each node and the expected number of transactions.)
- The minimum controller configuration for the system as a cluster node: for example, with OPS you will need at least two SCSI controllers and two LAN controllers just for using OPS.
- Possible future extensions with additional I/O components (FDDI, Token Ring etc.).

6.7.1.2 Shared disks

Shared disks must be operated exclusively in external disk subsystems. The maximum number and the type of these external disk subsystems (I/O cabinets and peripherals cabinets) depend on the server systems used.

Instead of these conventional disk subsystems you can use one or more RAID subsystems. While conventional disk subsystems are cheaper to buy, RAID systems are much simpler to configure and, in failover clusters, can also be used jointly by two systems.

6.7.1.3 Additional requirements

Both the computers themselves and the disk subsystems used must be protected against power failure by means of BBUs or UPSs.

6.7.1.4 Controller Area Network (CAN bus)

The CAN bus connects all the cabinets in the cluster (i.e. the system cabinets of the cluster nodes as well as the peripheral cabinets and I/O cabinets).

6.7.2 Special requirements for failover clusters with OBSERVE

6.7.2.1 Serial connections for sign-of-life monitoring

In addition to the public LAN, a failover cluster also requires a serial connection between the nodes in order to guarantee sign-of-life monitoring.

6.7.2.2 LAN interface

If IP aliasing is to be implemented via OBSNET, one additional LAN interface must be used for the monitored system and another for the second system.

6.7.2.3 Cluster console (optional)

You can optionally use a cluster console for centralized administration of the cluster (single point of administration). The cluster console is a special system (usually an RM200) with the additional software package Slxscn.

6.7.3 Special requirements for RMS clusters and OPS clusters

6.7.3.1 Virtual Ethernet

You will need a Virtual Ethernet to control intercommunication between the nodes. The minimum number of LAN controllers required per cluster node for exclusive use by OPS is two (with two LAN ports per controller). The Virtual Ethernet needs one port from each LAN controller.

6.7.3.2 Cluster console

Each OPS cluster and RMS cluster must have a cluster console, a special system (usually an RM200 or a UNIX PC, known as a LAN console) with the additional software package Slxscon. The cluster console is linked via serial cables to every node of the cluster; it manages the nodes and monitors their availability. If one of the nodes fails, the cluster console restarts the cluster and ensures the availability of the remaining nodes.

6.7.3.3 Cluster-console Ethernet (private)

The OPS cluster console requires its own private Ethernet for monitoring the nodes. Where an Ethernet controller is used for the virtual Ethernet, its second port can be used for this Ethernet connection (this can be dispensed with if RMS is also used).

6.8 Hardware Configuration for RCS II

The special features of the Reliant Cluster Server II (RCS II) are described in the appropriately names chapter.

7 Reliant Cluster Server II (RCS II)

The mounting complexity of business processes and their ever-growing demands for the availability of systems and data, as well as for scalable performance, increasingly often prove to be more than an individual computer system can handle.

The cost of a system failure may be considerable, depending on the particular application and market sector involved. The Gartner group estimates that 25 percent of companies would not survive a one-day total failure of their IT infrastructure, and conclude further that almost all enterprises would go to the wall after four days without their computer systems. In addition, corresponding studies have shown that 90 percent of companies which suffer a total catastrophic loss of data will be bankrupt before very long.

This is why Siemens is now offering a comprehensive multistage concept of system extensions, based on the RM group of systems, which increase the availability and performance of the overall system step by step: *Reliant Cluster Server II (RCS II)* - this product embodies manifold possible extensions to suit different business requirements.

With Reliant Cluster Server II, Siemens is setting a new standard for availability, scalability and administrability. Reliant Cluster Server II makes full use of existing products and ideas, and thus represents an on-going development of the existing concepts of high-availability (*failover*) clusters and database clusters. It incorporates the latest technology to fulfill the requirements for:

- increased availability of systems, components, applications and data
- distribution of the load represented by users, applications and data management
- simplified cluster setup and migration of our existing cluster solutions, and
- convenient and comprehensible administration of the cluster infrastructure.

The following diagram will illustrate once again these advantages, as they affect a company using the Reliant Cluster Server II.

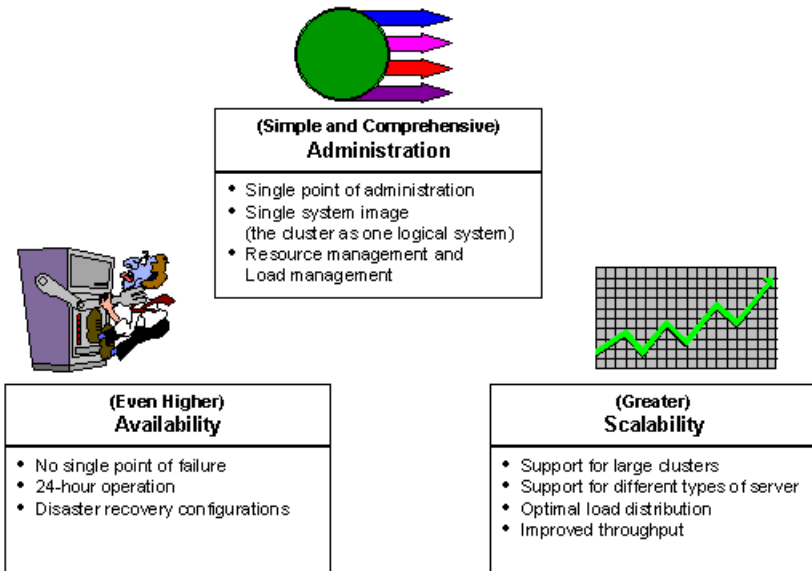


Figure 16: Advantages of Reliant Cluster Server II

7.1 The Concept

The Reliant Cluster Server II is not a single delivery unit which can be ordered as such, it is represented by a specified set of products and components. To use our term, they are the “building blocks” whose optimal interaction is the very essence of the RCS II.

7.1.1 The building blocks of the Reliant Cluster Server II

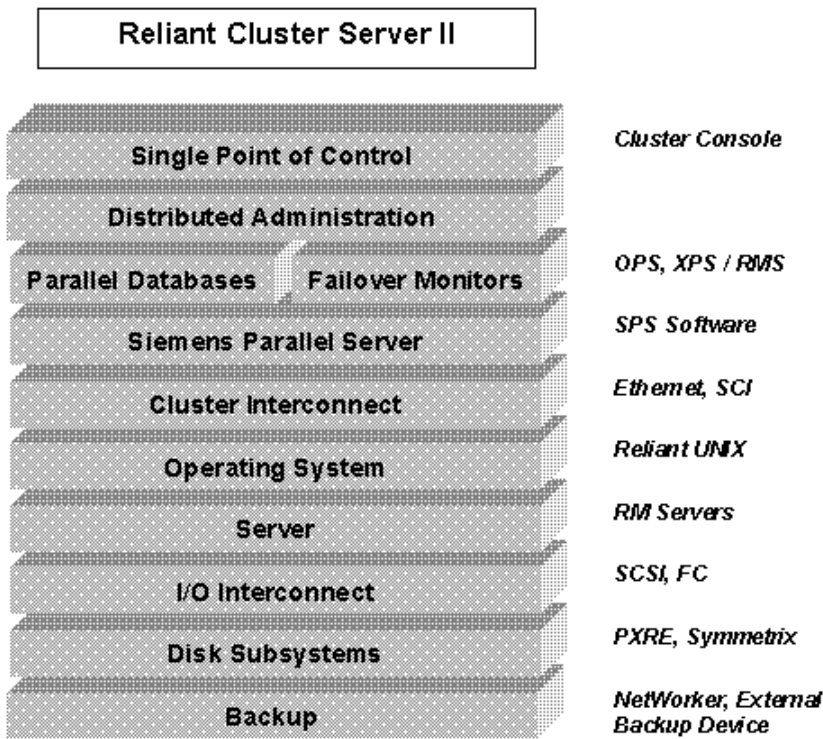


Figure 17: The building blocks of the RCS II

- *Single Point of Control:* One single console for the entire cluster
- All the administrative tasks for a cluster are carried out from a cluster console. This is an RM200 or UNIX PC, which is connected via LAN (local area network) and via serial connections (one V.24 connection per node). All console functions can thus be carried out from a single location.

- *Distributed Administration*: simple and convenient cluster administration

This functionality, which is provided with the operating system Reliant UNIX 5.45, gives the administrator a uniform view of the entire cluster. This means that administrative activity is no longer focused on individual systems but can operate on the whole of the integrated system.

- *Parallel Databases*: using resources for distributed data

The new “parallel databases” from ORACLE and INFORMIX, OPS and XPS, provide a whole series of powerful functions for securing the data consistency and distributing the load within a cluster. These products receive optimal support from the RCS II.

- *Failover Monitor* (high-availability monitor): guarantees the availability of the system

RMS (Reliant Monitor Software) is a product offering comprehensive functionality drawn from experience of over 1500 failover installations, all the way from object-level monitoring to workload.

- *Siemens Parallel Server*: the intelligent instance for parallel databases

SPS (Siemens Parallel Server) has the task of providing optimal support for such products as ORACLE, OPS, and INFORMIX, XPS. SPS has been adapted from the software technology used in massively parallel systems for symmetrical multiprocessor systems such as the RM600 E.

- *Reliant UNIX*: the operating system for company-wide use

The individual cluster nodes utilize the full 64-bit functionality of Reliant UNIX 5.44C. The version Reliant UNIX 5.45 also provides the convenience and extensive integration of domain administration.

- *RM systems*: the powerful nodes which make up the cluster

The nodes for the RCS II are the RM servers belonging to the RM group of systems. The corresponding follow-on systems will also be supported by RCS II.

- *Cluster Interconnect (CI)*: fast and reliable communications between the nodes

In addition to the familiar Ethernet connections between the nodes, a communications controller of type SCI (scalable coherent interface) is also provided. This consists of an SCI adapter for the well-known EHIOS

technology used in the RM600E system. With an EHIOS CI board and FO (fiber optic) cable between the systems, it can achieve transfer rates of up to 400 Mbyte/s.

- *I/O Interconnect*: faster over long distances with Fibre Channel

In addition to the familiar method of connecting storage subsystems via SCSI, Fibre Channel technology opens up the possibility of transmitting data from subsystems up to 10 km away at a transfer rate of up to 100 Mbyte/s.

- *Disk Subsystems*: accessing and storing data using the most up-to-date technology

The storage subsystems (RAID subsystems) PXRE-HF and Symmetrix 3000 have high-availability functions and modern connection facilities (such as Fibre Channel) which make them suitable for storing large volumes of mission-critical data.

- *Backup*: securing data throughout the company.

The product *NetWorker* coupled with an external backup device (such as the PXT7) enables backup of the entire data stock while the system remains in operation.

7.1.2 The different editions

The concept of the Reliant Cluster Server II opens up a variety of possible applications. For the sake of simplicity we will first introduce appropriate solution packages for the areas *scalability with high-availability* and *high-availability alone (failover)*. These packages are listed and explained below.

- *Scalability Edition*:

This package comprises all the necessary products for a two-node cluster permitting the operation of a parallel database cluster using OPS or XPS. These include, among others, the components SPS, RMS, EHIOS CI boards for the RM600 E and a cluster console.

- *Failover Edition*:

This package comprises the necessary products and components for a two-node cluster using RMS in a purely high-availability (failover) configuration.

Both packages, the *Scalability Edition* and the *Failover Edition*, also contain all the necessary hardware and software components for implementing a high-availability backup and recovery strategy. These components include the software product *Networker*, an external backup device, and the necessary infrastructure for controlling and switching over this device.

- *Domain Admin:*

Domain Admin is an add-on product belonging to the operating system Reliant UNIX 5.45. It is used to administer a cluster, or a number of individual systems congregated into a domain, via a common console. From this *single point of control* you have a uniform view of the domain or cluster (single system image).

- *Turnkey Solution:* (the package “High-availability SAP R/3 System”)

RCS II is an “easy configuration” landmark for SAP R/3 . Together with the configuration tool *RCWizard* (RMS configuration wizard) it includes a turnkey high-availability (failover) solution for SAP R/3 applications) within the product RMS.

The essential features of all these packages are explained in the following sections.

7.1.2.1 The features of the Scalability Edition

This solution package places the emphasis on scalability. In addition to the availability of data and applications provided by the components of the *Failover Edition*, it guarantees even higher availability for the database and enables a number of nodes to operate in parallel on a single database.

A number of systems can access the database simultaneously. The users of a failed system can continue working on one of the other systems, and the consistency of the database is always secure. This even applies if several of the nodes in a cluster should fail. All the necessary activities - such as the immediate reconfiguration of the cluster - are carried out transparently and without operator intervention.

Competing accesses to identical data areas are synchronized by special software products. The advantage of this configuration is increased availability of the database. While the *Failover Edition* carries out a switchover followed by a recovery of the database, which may take some time, the database of a parallel database cluster such as the *Scalability Edition* remains continuously

available from the intact nodes, even when a system has failed. As soon as the remaining nodes have reached agreement on the reconfiguration, one of the intact nodes will perform an online recovery for the failed node.

The fact that this takes only the database into account means that a parallel database cluster can only be an additional option in a high-availability (failover) cluster, because usually applications also have to be kept available and IP addresses for clients must be switched over. This necessity is mirrored in the proposed packages Failover Edition and Scalability Edition.

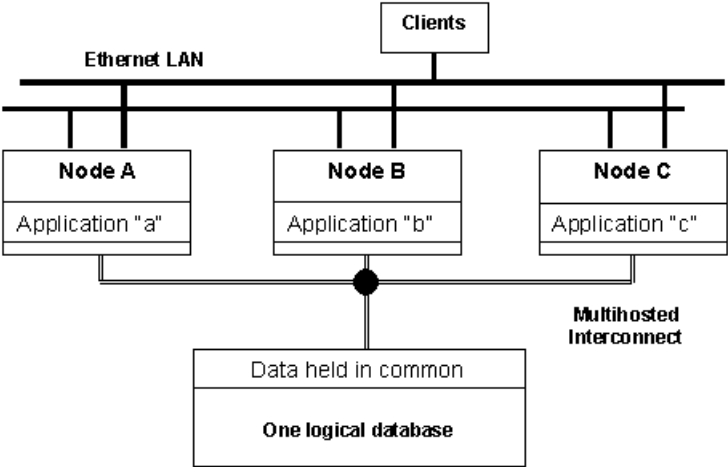


Figure 18: Data consistency in a parallel database cluster

In addition to increased availability, a database cluster configuration also aims to increase the throughput of the database beyond that of an isolated system by enabling parallel access from several systems. If scalability and increased throughput are of primary importance, the individual nodes will have to be equipped for maximum performance.

7.1.2.2 The features of the Failover Edition

All the nodes of the Failover Edition can be used productively, i.e. they are **not** merely available as “standby”. Each node operates with its own specific applications and data areas.

The nodes are redundantly interconnected and carry out “sign-of-life monitoring” with each other at regular intervals. When a node is recognized to have failed, the affected data areas and applications are automatically switched over to a functioning node.

In addition to this, the workstations connected via the LAN are switched over transparently (“IP aliasing”). In the Failover Edition, the availability of mission-critical applications is guaranteed by the failover procedure which restarts the applications on an intact node. The applications already running on the “takeover” system can continue unaffected operation during this time. There is no need for the system to be rebooted.

The length of time needed for such a switchover is determined by the time needed to recover the applications, and is thus largely within the customer’s remit. Switching over the necessary system components is a matter of seconds, or minutes, particularly with *Veritas* file systems, but the procedure for recovering a database may take a very long time, depending on its size and configuration.

In addition to the reciprocal monitoring of the set of nodes there are also programs for the local monitoring of other objects in the system, such as printers and processes. The reaction to the failure of such an object can be specified in the same way as for a system failure.

One of the outstanding features of the Failover Edition is its flexibility: any desired objects can be monitored, and the customer can decide which objects these should be and how often they should be checked. He or she selects the system components and applications that are to be monitored and specifies the action that is to be taken in response to the failure of a node.

In the Failover Edition these functions and mechanisms are implemented by the *High-Availability Monitor* (HA monitor).

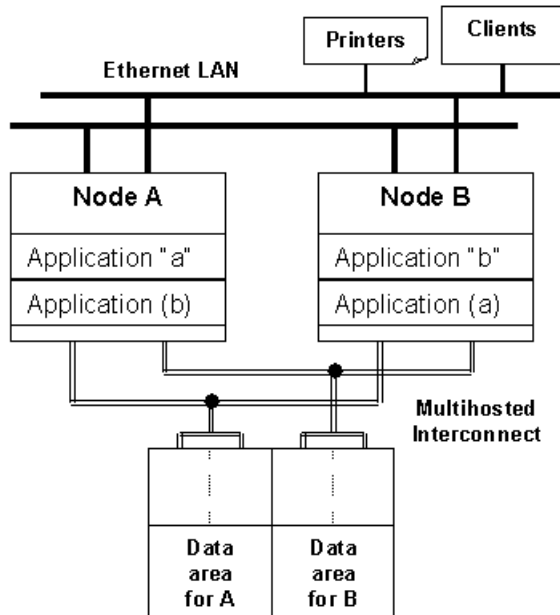


Figure 19: Transfer of applications and data in the Failover Edition

7.1.2.3 The features of Domain Admin

Domain Admin is a Web-based add-on for Reliant UNIX 5.45, which extends the administration beyond the confines of the individual system. It permits a uniform view of a number of systems which form a domain. This can (but need not) be a cluster.

The core element of this package is comprehensive domain management, offering - among other things - functions for domain-wide configuration and the distribution and installation of software. It also includes tools for performance and workload management, enabling the overall system resources to be used as effectively, efficiently and flexibly as possible.

Among other things, the domain management (*single point of administration*) comprises:

- functions for incorporating and removing systems
- dynamic reconfiguration of hardware components (such as boards, controllers, etc.)

- domain-wide distribution and installation of software
- domain-wide user administration
- domain-wide event handling, based e.g. on a centralized log-book. Naturally, this can also involve *Teleservice* or *TransView Control Center*.

Additional resource management and performance management products round off the domain administration concept (see also section 7.3.5).

7.1.2.4 The features of the high-availability SAP R/3 system (Turnkey Solution)

This package comprises switchover scripts and specific monitoring programs. You specify just a few SAP R/3-specific parameters and the complete high-availability environment is created automatically. The resultant configuration file contains definitions of the system components that are to be monitored and their mutual interdependencies.

This failover process also includes the handling of application systems that are not controlled by RMS. Such application systems have to be halted and subsequently restarted when the database system is switched over.

With the help of the *RCWizard*, this high-availability product package for SAP R/3 systems also supports this procedure for “non-RMS application systems”.

This applies equally to NT-based application systems.

Over and above the purely “high-availability” approach, this product package enables you to compensate for a number of known malfunctions within R/3 and to avoid downtime by means of special monitoring programs (“detectors”). The monitoring of R/3 and database processes and the failover which initiates switchover procedures can be disabled when necessary to allow for administrative intervention.

7.2 Hardware Architecture of the Reliant Cluster Server II

The following figure shows the hardware architecture of the RCS II.

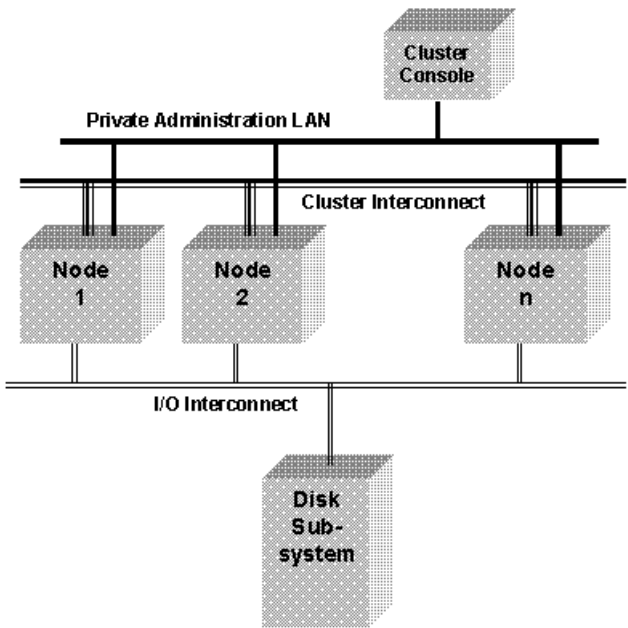


Figure 20: The hardware architecture

All the components are explained below.

7.2.1 RM600 E systems

At present, it is the models in the RM600 E group of systems that are principally used as nodes for the Reliant Cluster Server II.

7.2.2 The cluster console

The cluster console can either be an appropriately equipped RM200 running the operating system Reliant UNIX (version 5.44 C or higher) or a high-performance PC running the operating system Solaris 7. The cluster console is connected to the nodes by means of a private standard Ethernet LAN.

7.2.3 Connecting the cluster nodes

The nodes are connected together either by a redundantly installed Ethernet LAN (100 Mbit/s), or via SCI (scalable coherent interface).

Fiber optic cables with appropriate “switches” are used.

7.2.3.1 Cluster Interconnect

A cluster interconnect is a private connection network between the cluster nodes, which is used exclusively by the parallel databases, or rather, by the component SPS. The following media are available for implementing the cluster interconnect:

- Fast Ethernet with a transfer rate of 100 Mbit/s
- SCI (scalable coherent interface) with 6400 Mbit/s
- Gigabit Ethernet 1000 Mbit/s

Redundancy

To prevent this cluster interconnect from failing due to the malfunction of a single component it is configured with complete redundancy. This means that each node in the cluster is normally connected to each other node via two independent lines. These can also be used in parallel so long as there are no malfunctions, that is, for over 99 % of the time the cluster is in operation.

Scalable Coherent Interface (SCI)

Cluster interconnects were previously usually implemented using Fast Ethernet, but extremely heavy use of the database often means that the 100 Mbit/s throughput of this medium is no longer adequate. For this reason, and for optimal support of the latest parallel database system architectures, you can now use the high speed cluster interconnect SCI to integrate systems belonging to the RM600 E group into a highly scalable high-performance group.

SCI is based on the IEEE standard and specifies a bidirectional ring, currently with a transfer rate of 400 Mbyte/s in each direction, i.e. 800 Mbyte/s for both directions.

The cables used for SCI are either 5m copper cables or fiber optic cables of up to 150m in length. If there are only two nodes they are connected directly; if there are more than two they are connected using “switches”. Such a switch is a small box which is connected to the individual cluster nodes. The advantage of this technology over a ring connection is that individual cluster nodes can easily be switched out and back in without disturbing the communications of the remaining nodes.

SCI Adapters

A new communications controller has been developed for the RM600 models E30 and E70 (and subsequent models) based on the SCI specification.

This controller actually consists of an EHIOS board (extended high input output system) and a plugable daughterboard, which is the SCI adapter. The EHIOS CI (cluster interconnect) controller provides two CI connections.

SCI Switches (Distributors)

Separate “switches” (distributors) are provided for connecting more than two nodes. These distributors each have four ports and can be cascaded via two extension ports. Copper cables are attached to the four ports and, with the appropriate adapters, FO cables (fiber optic) can be connected to the extension ports.

SCI distributors are supplied as desktop units and can be installed in a 19" peripherals cabinet.

SCI Connections

As connections between the systems, or between system and SCI distributor we support copper cables (up to 5m), or FO cables for greater distances.

Copper Cables

A redundant CI connection between two systems (with no SCI distributor) consists of four copper cables including eight connectors.

If this configuration were to be implemented using an SCI distributor we would need a total of eight copper cables (four connecting the SCI distributor to each system).

FO Cables

A redundant CI connection between two systems (with no SCI distributor) consists of four FO cables including eight cable adapters. These Paroli adapters are available in 20m, 50m and 150m lengths.

A configuration using a SCI distributor would double the number of cables and Paroli adapters in each case.

7.2.3.2 I/O Interconnect

Connections to the storage subsystems PXRE-HF or Symmetrix 3000 are made via the appropriate Fibre Channel controller (FC) using the appropriate FO cables.

Connections to the storage subsystems in an RCS II configuration are made with full redundancy and crossed over. The following diagram shows a two-node configuration with a common storage subsystem.

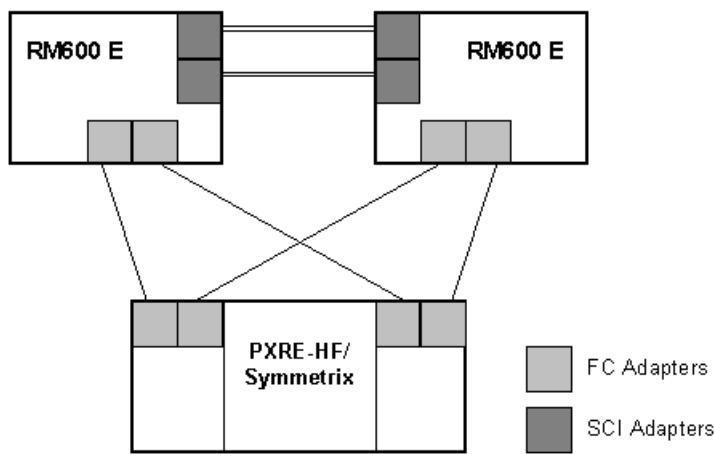


Figure 21: A two-node cluster with direct FC connection to the common storage subsystem

If more than two nodes are to be connected to a common storage subsystem you can use distributors (switches).

7.3 Software Architecture of the Reliant Cluster Server II

The following diagram shows a sample software architecture of the Reliant Cluster Server II.

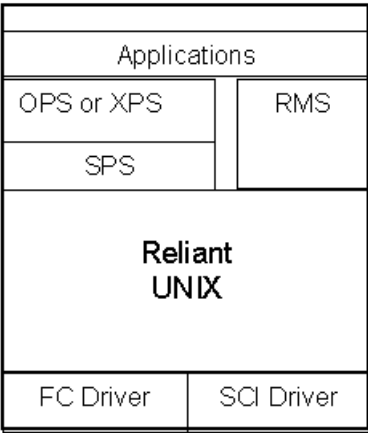


Figure 22: The software architecture

7.3.1 Reliant Monitor Software (RMS)

The high-availability (failover) concept requires the component RMS.

7.3.2 Parallel databases OPS and XPS

The additional components OPS V8.xx and XPS V8.xx, from ORACLE and INFORMIX, plus SPS, are required for scalability.

7.3.3 Siemens Parallel Server (SPS)

SPS here constitutes the software layer for the support of parallel database systems. In this context it supplies the cluster-wide interprocess communications interface which these database systems require.

7.3.4 Reliant UNIX 5.45

Reliant UNIX 5.45 provides additional cluster functionality together with the operating system and its add-ons, making Reliant UNIX 5.45 the recommended UNIX operating system for RCS II. Appropriate drivers are provided for the cluster interconnect controller and I/O interconnect controller (SCI and FC).

7.3.5 Additional cluster management products

The following products round off the concept of Domain Admin for cluster management.

Share II (“share two”)

Share II is a resource management tool which offers the system administrator more control over the allocation of system resources in a Reliant UNIX operating system environment. This makes it possible to guarantee the availability of certain resources for users and mission-critical applications.

ViewPoint

ViewPoint is a solution for monitoring and analyzing system performance. It is versatile and easy to use. From a PC, in combination with the latest client/server and Microsoft Windows technology, ViewPoint monitors and analyzes the activity on a number of computers running various operating systems and databases, either in real-time or in history mode.

ViewPoint is capable of monitoring and analyzing over 50 systems simultaneously.



These products are independent of RCS II and can also be used on their own.

7.4 Compatibility and Migration

The systems RM600 E30 and E70 can be combined in any desired way within an RCS II configuration provided the identical operating system version is used.

If requested, the systems RM600 E20 and E60 can be configured as a two-node cluster using Ethernet as the cluster interconnect.

RM400 E systems can also be used as cluster nodes for RCS II but this is subject to a special release procedure in each case. This configuration supports Fast Ethernet as cluster interconnect.

Glossary

availability

The availability of a system is determined by the relationship of the actual to the planned usage time. In contrast to error tolerance, which is a system attribute, availability is a system extension, which is added on to the standard version of the operating system SINIX.

backup

Backup refers to any form of static data-saving. Backing up data increases availability. The more often data is backed up, the more quickly the current data set can be used again after a failure. Mirroring guarantees that data is always up to date. Mirroring takes place online and does not impair operation of the system.

A backup that can be performed while the system is running and therefore does not interrupt current operation is called an online backup. In contrast to mirroring, online backup saves data statically, i.e. it records the state that existed at the time of the save.

Backups are logical or physical. With logical backups, files or file systems are saved. With physical backups, a 1:1 image of the disk is stored.

basic availability

Basic availability is the basis for high availability. It is guaranteed by early quality control, system-internal fault-tolerant measures, sophisticated service strategies and the UNIX commands that support availability.

block-oriented access

Data is accessed in blocks. If, for example, files of a file system are written to the hard disk, the data is stored block by block first in a buffer, then on the disk. Devices which are accessed in this way have block-oriented special files.

boot slice

The boot slice is part of the hard disk. It ensures that UNIX can be loaded when the file system / (root) is present. If an alternative boot slice is set up, ideally on a second disk, the operating system can still be loaded even if a disk fails.

character-oriented access

Data is accessed character by character. If, for example, data is written to the hard disk, it is stored on the disk as a data stream without taking a detour via a buffer. Character-oriented access is also called raw mode.

cluster

A cluster always consists of at least two RM systems of type RM400 or RM600. Each of the systems in the cluster, seen in isolation, is an arbitrarily configured symmetrical multiprocessor system. Special hardware and software add-ons are used to combine a number of such systems, each with their own peripherals and external disk subsystems, to form a cluster. This gives the redundancy an additional boost.

cluster interconnect (CI)

Cluster Interconnect enables fast and secure communication between the individual systems of a cluster, the so- called nodes. In addition to the well known Ethernet connections, an SCI (Scalable Coherent Interface) communications controller is also provided, consisting of an SCI adapter for the EHIOs technology of the RM600 E system. With an EHIOs CI board and an FO cable (Fiber Optic), transfers of up to 400 MB/sec are possible.

data consistency

Data is said to be consistent if it is all in the same state and does not contain any contradictions.

data integrity

Protecting data from corruption and unauthorized access is one way of maintaining the data integrity.

downtime

-> *interrupt time*

error handling

Once errors are detected, they can be handled in different ways. Faulty components are excluded from operation - they are no longer addressed, isolated. Compensating for errors restores a correct result, for example when errored data is overwritten by correct data. Recovering errors restores an error-free state, although this does not eliminate the cause of the error.

error tolerance

Error tolerance is a feature of systems whose prior aim is to tolerate any errors which occur. Multiple errors are always intercepted, usually without the users noticing the error handling. Error tolerance guarantees that planned and actual operating time are virtually identical.

expansion cabinet

Expansion cabinets contain additional, external disks, which can be accessed by a second system even if the first system which looks after the disks crashes. The term expansion cabinet is often used synonymously with extension box or peripherals cabinet.

interrupt time

In this manual, interrupt time is the same as downtime. Depending on the context, the terms mean the duration of a single failure or the sum of all downtimes for the defined observation period, usually one year. The interrupt time begins with the failure and ends when the user can resume work again.

failover cluster

The term failover cluster is used for a cluster in which each of the participating nodes is actively processing its own tasks, but also takes over the tasks of another node if this should fail. The software which is available for implementing failover clusters on RM systems consists of the two high-availability monitors OBSERVE and RMS (Reliant Monitor Software). If one of the participating cluster system should fail, this software makes it possible for the affected applications to switch over to some other node in the cluster, or to switch resources over to a different cluster node.

high availability

The increase in the average availability of a system through special hardware and software measures is known as high availability. *Clusters*, a special combination of systems, is one way of increasing availability. With Reliant Cluster Server II, Siemens is offering solutions packages for the sectors of *scalability with high availability* and *pure high availability*. The system which can be expanded to form such a cluster is the RM600 model CS42, the consolidation server.

log file

To safeguard database actions, all user actions are written to log files as they occur. These files can be used to reset incomplete user actions.

mirrored disks

In this manual, the term mirrored disks refers to all disks containing mirrored data, regardless of whether the disks are internal or external or whether the data is primary or secondary.

mirroring

At least two copies of all data are kept: we speak of original data (primary data) and mirrored data (secondary data). This means that if a data record becomes unavailable or corrupt, the user does not notice as the system continues to work with the duplicate record.

mixed configuration

Standby configurations with different system models are called mixed configurations.

multiple error

A multiple error occurs when more than one error is present at the same time (-> *single error*).

object response script

-> *response script*

observer

Observers are monitoring programs in OBSERVE. There are standard observers, provided by OBSERVE in finished form, and user observers, which users can program themselves.

online backup

-> *backup*

online repair

All repair work carried out while the system is running is called online repair. Unlike quasi-online repair, the system tasks do not need to be handed over to a redundant system while online repair is performed.

primary data

-> *mirroring*

quasi-online repair

A faulty component can be repaired without interrupting system operation if operation during the repair time is controlled by another system. Quasi-online repair is thus possible whenever productive tasks can be passed over to a redundant system.

raw mode

-> *character-oriented access*

redundancy

Redundancy is said to exist when there are two or more ways of solving a task. The redundant use, for example, of data and hardware components increases the availability of a system.

Reliant Cluster Server II (RCS II)

Reliant Cluster Server II is not one single deliverable item; rather it is made up of a defined set of products and components, the building blocks of the RCS II. Along with the operating system and the server (hardware), these include parallel databases, high availability monitors, Cluster Interconnect and disk subsystems.

The RCS II concept offers many possibilities that are tailored to the needs of each individual company. As a first step, suitable solution packages are offered for the sectors *scalability with high availability* and *pure high availability (failover)*. The RM600 model that can be expanded into a cluster on which these solutions packages run is the CS42.

response script

OBSERVE uses shell scripts to define appropriate responses for standard and unexpected situations. Scripts which are used in standard situations, such as system startup, handing over the system functionality, or system shutdown, are called system response scripts. Object response scripts are created to respond to the failure of individual observers and monitored components, to support switchovers, or to start and terminate applications.

resource

The term resource is used for any component which is to be monitored by a cluster system, for instance a mirror disk, a RAID subsystem, a printer, etc.

restart (time)

The restart time is the final phase of the interrupt time. It begins with rebooting after a system crash.

secondary data

-> *mirroring*

single error

Every error that occurs is a single error in its own right. If a second error occurs, we speak of a multiple error. Sometimes a single error can be a single point of failure.

single point of failure

A single point of failure is a single error which can bring down the whole system by itself.

slice

Disks are divided into slices. These are logical disk drives which are uniquely defined by special files.

special file

Under UNIX all devices are treated as files. Data is passed on to a device via a special file. Devices include whole disks and the slices on the disks.

Depending on the access type, a distinction is made between block-oriented (-> *block-oriented access*) and character-oriented (-> *character-oriented access*) special files.

standby configuration

Configurations with two or more systems are called standby configurations if one system is kept ready for taking over the tasks of a failed production system. To what extent the standby system is also used for other tasks depends on its capacity. Standby configurations can be implemented as dual system configurations or as 1:n configurations.

switchover

Switchover, as defined in this manual, occurs when all tasks controlled by one system are passed on to another system or are taken over by it. The tasks are switched over manually, by command or via OBSERVE. For some hardware components, OBSERVE provides its own switchover programs, for others switchover programs can be written in OBSERVE.

Abbreviations

CEN
Control and Environmental Node

CI
Cluster Interconnect

CLI
Command Line Interface

CPU
Central Processing Unit

DLM
Distributed Lock Manager

DMA
Direct Memory Access

DSS
Decision Support System

ESCON
Enterprise Systems Connection

FC
Fibre Channel

FIFO
First In First Out

FO
Fiber Optic

FRU
Field Replaceable Unit

Gbyte
Gigabytes

Abbreviations

GUI	Graphical User Interface
HA	High Availability
ICF	Internode Communication Facility
I/O	Input/Output
IOCF	Input Output Control Facility
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
Mbyte	Megabytes
MIA	Mesh Interface ASIC
MPP	Massively Parallel Processing
MRC	Mesh Router Chip
MVM	Mesh Visual Manager
OLM	Online Maintenance

OLTP

Online Transaction Processing

POD

Pyramid Object Database

RAS

Reliability Availability Serviceability

RCS II

Reliant Cluster Server II

RDBMS

Relational Database Management System

RMC

Remote Mesh Network Connection

SCI

Scalable Coherent Interface

SCSI

Small Computer System Interface

SGA

System Global Area (buffer area of an ORACLE database)

SMP

Symmetrical Multi Processor

Tbyte

Terabytes

TCP

Transport Control Protocol

VCP

Virtual Console Protocol

Related publications

ASECO V2.1

User Guide

Target group

System administrators

Content

The manual describes how chipcards can be used in a network to protect Reliant UNIX computers and applications against unauthorized access.

AUDIT V2.1

(Reliant UNIX)

System Administrator Guide

Target group

System administrators

Content

The manual describes the administration of the AUDIT subsystem on Reliant UNIX computers.

AUDIT is a logging tool used to record as evidence processes which could affect the system security.

OBSERVE V1.1

(SINIX)

User Guide

Target group

System administrators, Programmers

Content

The manual describes the installation and configuration of OBSERVE in configurations with several computers.

With OBSERVE the availability of one or more SINIX computers can be significantly increased, since OBSERVE is capable of recognizing many different types of system failure and responding accordingly. If one computer fails completely, OBSERVE can transfer its tasks to another.

OBSERVE V1.2

(Reliant UNIX)

User Guide

Target group

System administrators, Programmers

Content

The manual describes the installation and configuration of OBSERVE in configurations with several computers.

With OBSERVE the availability of one or more reliant UNIX computers can be significantly increased, since OBSERVE is capable of recognizing many different types of system failure and responding accordingly. If one computer fails completely, OBSERVE can transfer its tasks to another.

OBSNET V1.1

(Reliant UNIX)

Product Manual

Target group

System administrators

Content

Additional optimizing components for OBSERVE together with the command interface and API interface.

openUTM V4.0

Concepts and Transactions

User Manual

Target group

Users

Content

The manual contains a general description of all functions and a introduction for beginners.

**ORACLE Parallel Server for SNI RM600 Clusters
White Paper**

Overview

Target group

System administrators

Content

The manual describes the administration of a global data base for all nodes in a cluster system.

**PXRC-S
Installing, Operating and Maintaining the RAID System**

Firmware V3.0/V3.1

Target group

System administrators

Content

The manual describes the installation, operation and maintenance of the RAID system PXRC-S.

**PXRC-S
Configuring the RAID System**

Firmware V3.0/V3.1

Target group

System administrators

Content

The manual describes the configuration of the RAID system PXRC-S.

**PXRE-S
Starting up and Operating the RAID System**

Target group

System administrators

Content

The manual describes the start-up and operation of the RAID system PXRE-S.

PXRE-S
Configuring the RAID System

Target group
System administrators

Content
The manual describes the configuration of the RAID system PXRE-S.

Reliant UNIX 5.45
Diagnostics and Troubleshooting
System Administrator Guide

Target group
System administrators, Service technicians

Content
Part 1 of the manual describes errors that can occur during system start-up and while the system is running, how the computer will respond to them as standard, and what action you can take to deal with them.

Part 2 describes the user interfaces for system diagnostics, e. g. Logging V3.0, Config and the diagnosis menu.

Part 3 contains descriptions of various routines that may assist you in analyzing and resolving errors, information on errors in conjunction with the cabling and start-up of peripherals, a description of the effects of a power failure, an explanation of the LAR concept, and a list of error messages in the boot 1 phase on RM600 systems.

Reliant UNIX 5.44
Hardware Configuration with Config under SINIX/windows
Product Manual

Target group
System administrators

Content
The manual describes the functionality of Config to configure RM systems using the graphical SINIX/windows user interface.

Reliant UNIX 5.45
Reliant UNIX Installation and Operation
RM600-xxx, RM600-E
Installation Guide

Target group
System administrators

Content
The manual describes the re-installation and updating of Reliant UNIX as well as initial installation of it when there is a new system disk.

Reliant UNIX 5.45
Reliant UNIX Installation and Operation
RM200, RM300, RM400, RM600
Installation Guide

Target group
System administrators

Content
The manual describes the re-installation and updating of Reliant UNIX as well as initial installation of it when there is a new system disk.

Reliant UNIX 5.44
System Administration and Hardware Configuration Using the
SYSADM User Interface
System Administrator Guide

Target group
System administrators

Content
The manual describes the administration and configuration of RM systems using the alphanumeric SYSADM user interface.

SINIX V5.45
VERITAS File System (VxFS)
System Administrator Guide

Target group
System administrators

Content
The manual describes the VxFS file system. It names diagnostic and error messages and explains what to do in each case.

Reliant UNIX 5.45

Virtual Disks

System Administrator Guide

Target group

System administrators

Content

The manual describes the various types of virtual disks and how to configure them.

Reliant Monitor Software V3.1D

Installing RMS

Target group

System administrators

Content

Installation and Operation of the HA Software RMS.

Reliant Monitor Software V3.1D

RMS Configuration and Administration

Target group

System administrators

Content

Installation and Operation of the HA Software RMS.

Reliant Monitor Software V3.1D

Administering RMS using RCVM

Target group

System administrators

Content

Installation and Operation of the HA Software RMS.

SIDLM V1.2

Installation and Administration Guide

Target group

System administrators, Service technicians

Content

The manual describes the installation of the SNI Distributed Lock Manager software package and how to administer the SIDLM cluster.

**Siemens Parallel Server
Installation, Configuration and Administration of SPS**

Target group

System administrators

Content

This manual describes the installation and operation of SPS.

To order manuals please contact your local branch office.

Index

A

- access speed 36
- adapter 95
- administration domain 91
- alternative boot partition 19
- AMASS 25, 53
- architecture
 - hardware 93
 - software 97
- ASECO 24
- AUDIT 22
- availability 5
 - basic availability 7
 - data 20
 - hardware 13
 - high availability 7
 - influences 6
 - operation 25
 - scalable 8
 - software 18
- availability, see also
 - basic availability
 - higher availability

B

- basic availability 7
 - data 20
 - hardware 13
 - operation 25
 - software 18
- Battery Backup Unit (BBU) 16, 33
- boot partition, alternative 19
- buffer caches 75
- building blocks 84
 - of RCS II 85

C

- cables
 - copper 95
 - FO 95
- cache

- primary 21
- second-level 20
- checksum generation (Hamming code) 40
- cluster console 75, 94
- cluster formation 61
- cluster interconnect 94
- cluster nodes 61
- cluster systems 61
- cluster, hardware configuration 79
- cluster, see also
 - failover cluster
 - OPS cluster
- concurrency control (internal access control) 75
- connections 95
- copper cable 95
- CRACK 23
- cron daemon (see cron (1M)) 47
- cyclic redundancy procedure 21

D

- data
 - basic availability 20
 - higher availability 22
- data archiving 50
- data backup 21, 50
 - AMASS 25, 53
 - DataMgr 25, 53
 - mirrored disks 48
 - NetWorker 24, 50
 - online 45, 48
 - snapshot 48
 - Veritas file system (VxFS) 24
- data corruption, protection against 20
- data integrity
 - ASECO 24
 - AUDIT 22
 - DCE 24
 - HORUS 23
- data integrity 21

Index

database integrity, restoring 75
DataMgr 25, 53
dataq backup
 HYPERtape 53
DCE 24
disk array
 use of 17
disk cabinets, remote 17
Distributed Lock Manager (DLM) 75
dkconfig command (see dkconfig(8))
 47
Domain Admin 91

E
emergency power supply 16, 34
Environmental Interface Processor
 (EIP) 14
Error Correction Code (ECC) 20

F
failover cluster 62
 IP address management 63
 OBSERVE software 66
 redundant use of peripherals 64
 response scripts 63
 restrictions placed on applica-
 tions 65
 RMS 71
 sign-of-life monitoring 63
 with OBSERVE 80
Failover Edition 90
fan trays 36
fiber optic cabling 17
file system
 backup (DSSI) 21
 changing size 47
 fast restoration of integrity 18, 46
 online administration 46
 restricting fragmentation 46
 Veritas (VxFS) 18, 24, 45
file system ufs 45
FO cable 95
fsadm command (see fsadm(1M)) 47
fsck command (see fsck(1M)) 46

G
global hot spare drive 44

H
Hamming Code 40
hard disk
 redundant 16
hardware
 basic availability 13
 higher availability 15
 tests 14
hardware architecture 93
hierarchical storage management 25
high availability 7, 13
high-availability monitors
 OBSERVE 62
 RMS 62
higher availability
 data 22
 hardware 15
 operation 26
 software 19
HORUS 23
hot-swap capability 44
HYPERtape 53

I
I/O bus 29
I/O Interconnect 96
internal access control (concurrency
 control) 75
IP address, switching 72
IP aliasing 90

L
Lockout Auto Recovery (LAR) 14, 29
log files, size 67
Logging V3.0 30, 49

M
main memory
 refresh 21
memory dumps, optimizing 19
migration 98

mirrored disk
 software 38
 technique 37
 mirrored disks 38, 48
 mkfs command (see mkfs(1M)) 46
 modification log (intent logging) 46
 monitoring functions
 event-oriented 23
 state-oriented 23
 mount command (see mount(1M)) 45

N

NetWorker 24, 50

O

OBSERVE 66
 disks, switching over 68
 IP address, switchover 69
 SCSI changeover switch (SCSI switch) 68
 OBSNET software 69
 online
 backup 26
 data backup 18, 48
 diagnostics TDS2 14
 file system administration 18, 46
 replacement (OLR) 15, 29
 online data backup 45
 openUTM transaction monitor 57
 operating errors, avoiding 26
 operating system, changing the 26
 operation
 basic availability 25
 higher availability 26
 OPS 97
 OPS cluster 73
 integrating into a failover cluster 76
 Virtual Ethernet 73
 optimizing memory dumps 19
 ORACLE
 OPS cluster 73
 parallel server 75

P

parallel cache management 75
 parity
 block 43
 check 20
 drive 41, 42
 Paroli adapter 95
 performance reduction 38
 power supply
 BBU 16, 33
 emergency power supply 16, 34
 USV 16, 33
 power supply units, redundant 35
 primary cache 21
 PXRC
 RAID system 43
 PXRE
 RAID system 43

R

RAID
 subsystem 38
 RAID 0 36
 RAID 1 37
 RAID 2 40
 RAID 3 41
 RAID 4 42
 RAID 5 43
 RAID controller 44
 RAID systems 36
 RCS II 83
 read-after-write procedure 21
 redundancy 94
 redundancy procedure
 cyclic 21
 redundant components and modules 16
 redundant power supply units 35
 refresh, main memory 21
 Reliant Cluster Server II 83
 Reliant Monitor Software (RMS) 71
 Reliant UNIX 5.45 98
 restart
 alternative boot partition 19

Index

- of the system 19
- optimizing memory dumps 19
- reducing the startup times 19

RMS 97

root partition 39

S

SAP R/3 92

Scalability Edition 88

scalable coherent interface 94

SCI 94

SCI adapter 95

SCI connections 95

SCI switch 95

second-level cache 20

Share II 98

shared database cluster 62

SIDLM cluster software 75

Siemens Parallel Server 97

sign-of-life monitoring 90

single point of failure 17, 61

Single Sign-On (SSO) 24

Slxscn software package 81

snapshot 48

SNMP networks 49

software

- basic availability 18

- higher availability 19

software architecture 97

SPS 97

SPS clusters 78

striping 41

suid programs 23

swap partition 39

switch 95

system bus 29

system environment monitoring 14

system shutdown 34

T

TACLAN 64

TDS2 online diagnostics 14

threshold monitoring 30, 32

transaction rates 42, 43

transfer rates 40, 41

TransView Control Center agent 23

Transaction monitor openUTM 57

U

ufs file system (Berkeley Fast File System) 45

Uninterruptible Power Supply (UPS)
16, 33

V

Veritas file system (VxFS) 18, 24, 45

ViewPoint 98

virtual disks 38

Virtual Ethernet software 75

X

XPS clusters 78

Siemens AG
Information and Communication Products
Computer Systems
User Documentation UNIX
33094 Paderborn
Germany

Fax: (++ 49) 52 51 8-1 52 48

Internet: DOCetc@pdb.siemens.de
<http://manuals.mchp.siemens.de>

Submitted by

Comments
Suggestions
Corrections

Comments on Reliant UNIX 5.45
High Availability



Siemens Business Services
Training Center, Consulting Service
81730 Munich
Germany

Fax: (++49) 89636-52003

Courses Consulting Self-tuition media

The Siemens Training Center offers consulting for in-service training, courses and self-tuition media on almost all aspects of information technology, either on your premises or at a Training Center near you. Write to us for information and course programs.

Submitted by

send me information/course programs on the following topics



