北京希望电子出版社
www.bhp.com.cn

21世纪新工具软件开发指南丛书 3

非集群高可用性系统技术解决方案

[美] 斯蒂夫·拉塞尔 著

本光盘内容是:
本版电子书

High Availability Without Clustering

英文版

High Availability Without Clustering

北京希望电子出版社
www.bhp.com.cn

21世纪新工具软件开发指南丛书 3

非集群高可用性系统技术解决方案

[美] 斯蒂夫·拉塞尔 著

本光盘内容是：
本版电子书

High Availability Without Clustering

英文版

High Availability Without Clustering

内 容 简 介

越来越多的企业使用功能强大的联网计算系统。随着企业业务运作自动化程度的不断提高，数据处理系统的有效性便成了企业运作的关键。一台服务器的故障有可能严重影响到企业的收入和声誉。多数情况下，拥有 IBM xSeries 和 Netfinity 服务器的用户都是采用集群技术来获得其系统的高可用性，但是硬件成本会成倍增加，此外，集群系统还需要附加的软件与技能。

本版书为没有条件采用集群技术的用户提供了另一种解决方案。IBM 服务器以标准组件或选用组件的形式内置多种功能，可确保当子系统组件发生故障时服务器仍可正常工作。先进的管理功能还能帮助用户在服务器出现故障之前就采取预防措施。书中详述了服务器所采用的可用性技术，并教授用户如何使用这些技术，以便最大程度地提高服务器的可靠性。全书分 7 章，分述如下：第 1 章可用性入门；第 2 章可用性管理工具；第 3 章硬件可用性综述；第 4 章磁盘子系统的可用性；第 5 章联网硬件如何提高可用性；第 6 章各类操作系统的运行；第 7 章提高可用性的其他措施。

本书对各行业特别是中小型企业中使用服务器的用户颇有实用价值，可帮助他们优化硬、软件的可用性。本书也是高等院校师生教学、自学参考书，以及科研院所必备的馆藏读物。

本光盘内容包括本版电子书。

系　列　书：21 世纪新工具软件开发指南丛书（3）（英文版）
书　　名：非集群高可用性系统技术解决方案
总　策　划：北京希望电子出版社
文本著作者：斯蒂夫·拉塞尔等
责任编辑：谢建勋
CD 制作者：希望多媒体创作中心
CD 测试者：希望多媒体测试部
出版、发行者：北京希望电子出版社
地　　址：北京中关村大街 26 号，100080
网址：www.bhp.com.cn　E-mail：lwm@hope.com.cn
电话：010-62562329,62541992,62637101,62637102
010-62633308,62633309（发行）
010-62613322-215（门市）　010-62629581（编辑部）
经　　销：各地新华书店、软件连锁店
排　　版：希望图书输出中心　李毅
CD 生产者：北京中新联光盘有限责任公司
文本印刷者：北京广益印刷厂
开本 / 规格：787×1092 毫米　16 开本　14.25 印张　300 千字
版次 / 印次：2001 年 7 月第 1 版　2001 年 7 月第 1 次印刷
印　　数：1-2000 册
本 版 号：ISBN 7-900071-73-3/TP•72
定　　价：37.00 元（1CD，含配套书）

Introduction to availability

Contents

CHAPTER 1 Introduction to availability 1

1.1 Availability 1
1.1.1 Clustering for high availability 2
1.1.2 High availability without clustering 3
1.2 Terminology 3
1.3 Improving availability 5
1.3.1 The cost of the alternative 5
1.3.2 Availability for xSeries and Netfinity servers 7
1.4 The scope of this book 13

CHAPTER 2 Tools for availability management 14

2.1 IBM Director 16
2.1.1 Advanced System Management 18
2.1.2 Setting up alerting 18

JSP00 /08

2.1.3 Capacity Manager 27
2.1.4 System Availability Tool 31
2.2 Universal Manageability Services (UMS) client 33
2.3 Other tools 35
2.3.1 Windows performance counters 35
2.3.2 Windows resource kits 36
2.3.3 Linux performance counters 37

CHAPTER 3 Integrated hardware availability features 40

3.1 Techniques to improve availability 41
3.2 Redundant power 42
3.3 Redundant cooling 44
3.4 Processors 46
3.5 Memory 46
3.5.1 Error checking and correcting memory (ECC) 47
3.5.2 Chipkill memory 48
3.6 Light Path Diagnostics 49
3.7 Advanced Systems Management hardware 52
3.7.1 Using a direct connection 53
3.7.2 Using an Ethernet connection 56
3.7.3 Automatic Server Restart 58
3.7.4 Remote Supervisor adapter 59
3.8 ActivePCI 63
3.8.1 Prerequisites for ActivePCI 63
3.8.2 Installing ActivePCI software for Windows 2000 64
3.8.3 Installing IBM Hot Plug for Windows NT 4.0 Package 67

CHAPTER 4 Disk subsystem availability 74

4.1 IBM xSeries and Netfinity server disk subsystems 75
4.2 ServeRAID controllers 76
4.2.1 ServeRAID controller families 76
4.3 Standard RAID levels 78
4.3.1 RAID-0 79
4.3.2 RAID-1 79
4.3.3 RAID-1E 80
4.3.4 RAID-5 81
4.3.5 RAID-5E 82
4.4 Advanced RAID levels 82
4.4.1 RAID-00 83

4.4.2 RAID-10 83
4.4.3 RAID-1E0 84
4.4.4 RAID-50 84
4.5 Servers with integrated RAID controllers 85
4.6 Choosing the right RAID Level 85
4.6.1 Important parameters 86
4.7 Failover RAID adapter pairs 86
4.7.1 Configuring a ServeRAID adapter failover pair 86
4.7.2 How failover adapter pairs work 87
4.8 FlashCopy 89

CHAPTER 5 Networking hardware for availability 94

5.1 Network adapters 95
5.2 Network adapter failover pairs 96
5.2.1 How to configure a failover adapter pair 96
5.2.2 Failover adapter pairs in action 115
5.3 Load balancing 116
5.3.1 Types of balance 117
5.3.2 Where to deploy load balancing 117
5.3.3 Benefits of load balancing 118
5.3.4 Adapters that support load balancing 119
5.3.5 onfiguring load-balancing adapters 120

CHAPTER 6 Keeping the operating system up and running 112

6.1 Windows NT 4.0 127
6.1.1 Hardware compatibility 128
6.1.2 Recovering from a blue screen 129
6.1.3 Automatically updating recovery information 130
6.1.4 Generating system documentation 131
6.1.5 Restarting failed services 132
6.1.6 Best practices 137
6.2 Windows 2000 137
6.2.1 Designed for reliability 138
6.2.2 Fewer reboots required 139
6.2.3 System file protection 139
6.2.4 Hardware compatibility 141
6.2.5 Recovering from a blue screen 145
6.2.6 Reducing recovery time 147
6.2.7 Restarting failed services 153

6.2.8 High availability storage 155
6.2.9 Application repair 156
6.2.10 Best practices 157
6.3 Linux 158
6.3.1 xSeries and Netfinity servers and Linux 159
6.3.2 IBM Director client for Linux 159
6.4 NetWare 162
6.4.1 The NetWare Management Portal 162
6.4.2 Portal functions 164

CHAPTER 7 Additional availability enhancements 172

7.1 Power 173
7.1.1 Uninterruptible power supplies 174
7.1.2 Power fault management software 174
7.1.3 Power conditioners 178
7.1.4 Independent power circuits 178
7.2 Electronic Service Agent 179
7.2.1 What it can do for you 179
7.2.2 Installation of ESA 181
7.2.3 Configuring and testing ESA 187
7.2.4 Enabling a system for ESA 191
7.2.5 Working with ESA 193
7.3 Update Connector Manager 198
7.3.1 Installation and configuration of UCM 198
7.3.2 Discovering updates 201
7.3.3 Applying updates 203
7.4 The other 80% of availability 205
7.4.1 Planning and testing 205
7.4.2 Operations management 207
7.4.3 Training and support 208
7.4.4 Change management 209
7.4.5 Backup 212
7.4.6 Disaster recovery 214

Chapter 1

To a business that has become dependent upon its data processing systems, availability of those systems is of crucial importance. Clustering technologies of various pedigrees have provided an answer to availability requirements, but at a price. How can you achieve maximum availability of your computers without the investment of a lot of money? The purpose of this book is to make you aware of the tools and options for IBM xSeries and Netfinity servers that can be used to increase system availability without resorting to clustering.

In this introductory chapter we discuss what is meant by high availability. We have included a brief overview of clustering for those unfamiliar with this technique. The latter part of the chapter includes some definitions of terms used in discussions about availability and summarizes the most important availability features found in IBM Intel processor-based servers.

1.1 | Availability

Availability means money in today's global, competitive business environment. Many organizations need almost continuous availability of their mission-critical server resources. Loss of service (sometimes called an *outage*) of an important server often translates directly into lost revenue or, even worse, lost customers.

Outages can be broadly classified into two categories. One encompasses the planned outages that take place when the operations staff take a server offline to perform backups, upgrades, maintenance, and other scheduled events. The other type are unplanned outages that occur due to unforeseen events such as power loss, a hardware or software failure, system operator errors, security breaches, or natural disasters.

Maximizing system availability means taking steps to minimize the impact on your users of any of these planned or unplanned events.

1.1.1 Clustering for high availability

A cluster, in its most general form, comprises two or more interconnected computers that are viewed and used as a single, unified computing resource. By using multiple systems, the impact of a failure of any individual system is kept low by passing the failed system's workload to the remaining members of the cluster. Clusters may offer other benefits, such as high performance and scalability, but here we are focusing on the availability features of clustering.

One popular clustering solution is based on Windows 2000 and Microsoft Cluster Services (MSCS). This high availability solution supports a two-node cluster in which both nodes, or servers, have access to a common disk storage subsystem. Only one server, however, may control an individual disk at any time.

If one of the servers in the cluster fails, the remaining server automatically assumes the control of the disks that the failed server was using in addition to its own disks. This server also starts up any applications and services (such as a Web server) that the failed server was running. The failed server can then be repaired offline without the loss of time or work efficiency, because access to that server's data and applications is still available. You should bear in mind, also, that the workload on the surviving server has increased and this may result in slower response times for your users.

When the failed server is operating again, it can be placed back into the cluster. At a convenient time, the resources are then reallocated between the two servers and the cluster can resume normal operation.

MSCS is just one clustering solution of several that are available. Other solutions, such as Lotus Domino clustering, use techniques that do not rely on specific hardware. However, multiple machines are still required and, as with MSCS, you have to buy servers that are powerful enough to handle the additional workload during a failure.

While clustering is a powerful and useful technology, it can be expensive and difficult to justify for systems that may not be critical to your business. Loss of such servers can still be inconvenient, and downtime should be kept to a minimum. In this situation, it makes sense to make the most of the facilities available to you to ensure that your servers are kept running. The rest of this book examines those features provided and available for IBM xSeries and Netfinity servers.

1.1.2 High availability without clustering

In contrast with clustering, you can improve system availability without adding additional servers by taking advantage of features that are included either as standard or as options for IBM xSeries and Netfinity servers. These features address different subsystems of the server, such as the memory, processor, and disk subsystems, for example. Redundant components such as fans, power supplies, and adapters can also provide higher availability, particularly when used with software that provides monitoring and alerting capability to your system administrators.

The following chapters in this redbook introduce you to these features and suggest ways in which they can be used to achieve the goal of high availability.

1.2 | Terminology

Before we begin to delve into the main topic of this book, we felt it worthwhile to define some of the common terms used when discussing features that provide high availability. Having a common understanding of these terms simplifies the descriptions used later in the book and helps to ensure that no misconceptions arise.

- Failure

  For the purpose of this book, a failure is defined as a deviation from expected behavior of any component of the IT infrastructure. If a system has time contraints then it fails when it is not available to end users at that time. A failure may also be behavior below defined performance levels.

- Mean time to failure (MTTF)

  Hardware reliability can be predicted by statistically analyzing historical data. The longer a component operates, the more it is likely to fail due to aging. The mean time to failure of a component is just that: a statistical forecast. The greater the MTTF of a component, the less likely it is to fail. You can use the MTTF of a component (if it is known) as useful information for establishing preventative maintenance procedures. MTTF of an overall system can be improved by carefully selecting your hardware and software. The table below lists MTTFs for typical components. Note that the MTTF of hardware components is influenced by a lot of external parameters such as location, temperature, electrical environment, and so on.

Table 1 MTTF of some hardware components

Component	MTTF
Cables	1000 years
Logic boards	3-20 years
Disks (without using RAID)	1-50 years
LAN 3	weeks
Power (North America)	about 5 months

- Mean time to recovery (MTTR)

  This is also a statistical measurement that indicates the average time it takes to recover from a certain failure. The lower the MTTR, the better. Self-healing systems may recover from a software error within seconds. A disaster such as an earthquake or a fire might take your systems down permanently. Having a *disaster recovery plan* is essential to handle this type of catastrophic event. Maintaining a pool of spare parts on site and taking frequent full backups will decrease the MTTR.

- Availability

  Availability is usually measured as the percentage of time a system or application is available. But even then, availability can be viewed from different perspectives:

  - What is the time frame during which availability is measured?
  - Is downtime measured as a failure of the physical component, as unavailability from the user's perspective, or of the application as a whole?
  - Are there performance or throughput constraints? (For example, 80% of online transactions needs to be completed within a maximum of two seconds.)

  We define availability as an acceptable or agreed-to level of end-user service during scheduled periods. It is common to measure it in terms of percentage of uptime. Table 2 illustrates one typical way in which availability is classified:

Table 2 System availability classification

Percent available	Downtime per year	Classification
99.5%	3.7 days	Conventional
99.9%	8.8 hours	Available
99.99%	52.6 minutes	Highly available
99.999%	5.3 minutes	Fault resilient
99.9999%	32 seconds	Fault tolerant

- Planned and unplanned outages

  Planned outages include maintenance, offline backups and upgrades. These can often

be scheduled outside periods when high availability is required. Unplanned outages will impact availability most significantly, and may be due to component failures, power failure, or human or procedural error.

1.3 | Improving availability

In this book, we take a broad view of availability, outlining how to make your IBM xSeries and Netfinity servers as available as possible without using clustering technologies. Our perspective is that of the end user, and the time window for availability measurement is when the system is supposed to provide its services to them. For an e-business Web server, this could be 24 hours a day, seven days a week. In the case of a departmental file server, this will be during normal business hours.

As we have explicitly eliminated clustering from the scope of this book, we are concerned with availability of a single box and the software components that execute on it. Our efforts to improve availability are focused in two areas:

- Eliminating or minimizing outages by introducing redundancy and fault-tolerant technologies (increasing MTTF).
- Minimizing recovery time when things do go wrong (reducing MTTR).

Availability doesn't just happen. It takes quality hardware and software, along with careful planning, testing, and implementation, well-trained people, proper and updated procedures, and frequent evaluation to achieve it.

1.3.1 The cost of the alternative

We can safely state that today's Intel-based servers will achieve about 95% or more availability out of the box, depending on what features are built into the machine. To justify investment to achieve better availability, you need to consider the cost of the alternative. What are the consequences of losing your server?

- Business revenue

  Revenue for business transactions that would have been completed in that time frame are lost.
- Productivity

  While the system is unavailable users cannot work on it, thus impacting their productivity. They may not be able to continue working for as long as the system or application is down. An example could be call-center employees who need to interrogate a database. Another productivity impact is on the support staff that has to

diagnose and solve the problem.

- Reputation

  A lost customer never comes back. Frequent outages or downtime at peak hours will create negative publicity and will possibly result in the loss of acustomer, which is far worse than the loss of a single purchase. Reputation also includes the credibility of the IT department with the rest of the company.

- Regulatory and legal concerns

  Deadlines may be missed due to lack of availability, or customer response time may be increased. In some industries, this may result in penalties and/or fines. It may even lead to legal action from those affected.

Always remember that, if your server is business-critical (such as for an e-business Web site), you might want to consider clustering after all.

Table 3 lists some cost estimates of downtime for various applications:

Table 3 Downtime costs by application

Application	Cost per minute
Call location	$27,000
Enterprise resource planning	$13,000
Supply chain management	$11,000
e-Commerce	$10,000
Customer service center	$3,700
Automated teller/Point-of-sale	$3,500

Source: "Average cost per minute of downtime" The Standish Group, 1998

Depending on your estimates of the cost of downtime to you, you may want to spend more money and effort preventing it. The higher the level of availability you need, the greater the costs involved, as illustrated in Figure 1, which shows the dramatic increase in relative cost as systems approach 100% availability:

Very high availability will usually involve some sort of clustering technology. The topics covered in this book aim to keep your systems as far to the right of this graph as possible without investing in clustering.

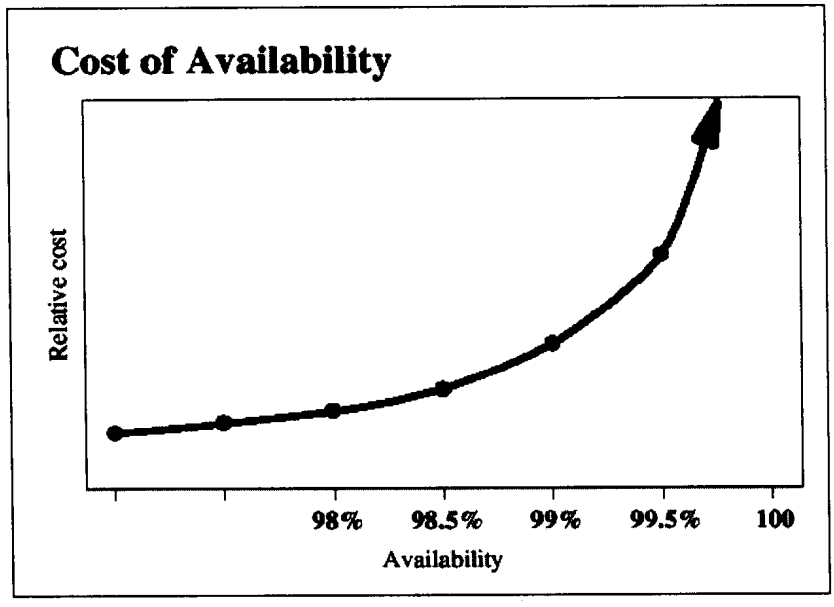

Figure 1. Relative cost of availability

1.3.2 Availability for xSeries and Netfinity servers

IBM xSeries and Netfinity systems are industry-standard, Intel processor-based, server products. They cover a whole range of price/performance points from entry-level, easy-to-use, low-cost servers to enterprise-level, high-performance systems. Power, control, scalability, reliability and serviceability are terms used by many manufacturers when describing their server offerings. IBM has designed the xSeries and Netfinity servers to make these terms more than mere words. Leading benchmarks results, award-winning management software and innovative technical solutions such as Advanced System Management, Light Path Diagnostics and Active PCI combine to differentiate these systems from Intel-based servers offered by other vendors.

Many of these features result from the implementation of X-architecture technologies and the OnForever initiative. The X-architecture design blueprint was launched to leverage existing innovative IBM technologies found in high-end systems, such as mainframes and UNIX-based servers, and introduce them into xSeries and Netfinity server products. X-architecture features available today include Predictive Failure Analysis, Active PCI, and Chipkill memory.

The OnForever initiative is one of the major cornerstones of X-architecture. Its goal is to design xSeries and Netfinity servers so that they can deliver close to uninterrupted computing. To achieve this goal, this initiative will, within time, extend beyond existing high availability features such as Active PCI and Chipkill memory, to include technologies such as hot-plug CPUs, hot-swap memory, and memory mirroring. Table 4 lists the various features of X-architecture and

categorizes them into four key areas:

Table 4 X-architecture implementation

OnForever Computing	Low-cost Enterprise Technologies	Driving Industry Standards	Making IT Easy
• Chipkill Memory	• Windows 2000 Datacenter	• PCI-X	• ServerGuide
• Active PCI	• 8-node MSCS	• InfiniBand I/O	• Light Path Diagnostics
• Common Diagnostics Model	• Capacity Manager	• Intel • Monterey	• Remote Connect
• Software Rejuvenation	• Netfinity SP Switch	• Linux	• Update Connector
• Advanced System Management Processor	• ESCON Channel Connection	•Microsoft Windows	• User-centered Design
	• Integrated Netfinity for AS/400	• Novell NetWare	•LANClient Control Manager
• IBM Center for Microsoft Technologies	• FlashCopy	• SCO UnixWare	• System Migration Assistant
	• RAID 1E & 5E	• IBM Director	• Software Delivery Assistant
• OPS Cluster Enabler	• Active Security	• Fibre Channel	• FAStT Storage Manager
• Predictive Failure Analysis	• Logical Drive Migration		• ServeRAID Manager
• SAF-TE			• Tivoli Storage Manager
• Capacity Manager			
• Remote Mirroring			

For more information on X-architecture, please visit

- http://www.pc.ibm.com/us/eserver/xseries/xarchitecture.html
- http://www.pc.ibm.com/us/techlink/wtpapers/index.html

In this book we will cover many of these technologies in detail in the upcoming chapters as they greatly influence the overall availability of our

servers. As mentioned earlier, we focus on two major determining factors that affect availability. The first is reducing the number of outages experienced. Technologies such as redundant components, update connector, PFA, Chipkill memory, software rejuvenation and Active PCI all contribute to that goal. The second factor is reducing recovery and aintenance time when outages do occur. X-architecture provides technologies such as hot-swap components, Advanced System Management (ASM) processors, Netfinity Director, Electronic Services Agent and Flashcopy to help achieve this second goal.

1.3.2.1 Overview of major X-architecture features

This section gives a brief description of the main X-architecture technologies that affect server availability:

- Predictive Failure Analysis

  IBM xSeries and Netfinity servers provide Predictive Failure Analysis (PFA), an

IBM-developed technology that periodically measures selected attributes of a component or its activity. If a predefined threshold is exceeded, PFA sends a warning message that enables timely replacement of the failing component before the failure actually occurs. IBM Director can be set up to alert an administrator to this impending failure so that corrective action can be taken—often from a remote location. Specific PFA-enabled components on Netfinity servers include hard-disk drives, power supplies, VRMs, cooling fans, processors and memory.

- Active PCI

  Active PCI lets you upgrade your hardware and software, replace hardware and make other changes to your servers without having to shut down your Netfinity servers. Active PCI features are designed to increase total server availability and can be described in three major categories:

  – Hot Add, allowing you to add adapters to the server to expand capacity
  – Hot Replace, allowing you to replace an adapter in the system that is no longer operating correctly
  – Failover, so that if the first adapter fails, a second adapter can pick up the workload

- Chipkill Error Correction Code (ECC)

  Chipkill ECC memory and automatic server restart features work to minimize server downtime. IBM Chipkill memory, initially developed for NASA's space mission, is an excellent example of IBM's commitment to provide systems that remain highly available. Chipkill memory is more reliable than standard ECC DIMMs at preventing certain system memory errors. With the latest Chipkill memory technology, select xSeries and Netfinity servers will be protected from any single memory chip that fails and any number of multi-bit errors from any portion of a single memory chip.

- Light Path Diagnostics

  Light Path Diagnostics contribute to advanced manageability. xSeries and Netfinity servers are designed with quick problem isolation as a goal: implementing a light-path service panel in conjunction with component LEDs. Lights are attached to specific components on selected servers. These lights are switched on when a failure occurs. Components monitored in this way include memory, processors, voltage regulator modules (VRMs), hard disk drives, power supplies, and cooling fans. Service personnel can quickly and easily identify a failing component by following the light path, potentially without even running diagnostics.

- Advanced System Management

  Using ASM, you can diagnose and resolve many problems without having to send an