Linux on IBM Netfinity servers

A collection of papers

- Systems and network management
- High availability clustering
- Interoperability

Jonathan Follows
Dennis Hunter
Andreas Thomasch
Shawn Walsh

Redbooks

ibm.com/redbooks

SG24-5994-00

International Technical Support Organization

Linux on IBM Netfinity servers
A collection of papers

September 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special notices” on page 105.

First Edition (September 2000)

This document created or updated on October 9, 2000.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Preface

This redbook is a collection of three separate papers describing the theory and practice behind three “hot topics” in today’s Linux environment. They describe how Linux can be used on Netfinity hardware from IBM to implement solutions which were either not possible before or which would have been much more expensive to implement on earlier platforms and operating systems.

Linux is not for everybody, but it offers many interesting possibilities, and these papers seek to demystify some of the jargon around Linux and to explain what can and can not be done to implement stable and reliable solutions using a combination of the Linux operating system and IBM’s Netfinity hardware platform.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Jonathan Follows is an IBM-certified networking specialist at the International Technical Support Organization, Raleigh. He writes redbooks on all areas of IBM networking hardware and software, most recently on policy-based networking. Before joining the ITSO in 1998, he worked as a technical specialist providing sales and marketing support in the United Kingdom and has 15 years’ experience in all types of networking. Jonathan read Mathematics at Oxford University, England, and holds a degree in Computing Science from London University, England.

Dennis Hunter is a is an Lead Product Support Specialist at the IBM PC Helpcenter in Raleigh, NC. He is Red Hat RHCE trained, Microsoft Certified and also an IBM Certified Server Specialist. Before joining the IBM in 1997, he had over 20 years of experience in the computer electronics field. He was trained for six years in computers by the U.S. Navy, and has served as a engineer for twelve years for Grumman Aerospace onboard several aircraft carriers His areas of expertise include Netfinity PC Servers, RAID, LAN Hardware and Windows NT 2000/4.0, SCO Unix, Red Hat Linux and Microsoft clustering software.

Andreas Thomasch is a Linux support specialist and consultant at the Boeblingen Development Laboratory in Germany He has 7 years of experience in the Linux field and still remembers the times when a Linux distribution was delivered on dozens of floppy disks. He has studied and

worked at IBM for 4 years. His areas of expertise include Unix server support and consulting (AIX, Linux/Intel, Linux/390, and Solaris). Andreas is running the German Linux server and LTC FTP mirror http://tux.boeblingen.de.ibm.com. He holds a degree in Computing Science from Berufsakademie Sachsen, Staatlichen Studienakademie Dresden, Germany.

Shawn Walsh is a ????title??? in ???country???. He/she has ?? years of experience in ???? field. He/she holds a degree in ???? from ????. His/her areas of expertise include ?????? He/she has written extensively on ???????.

Thanks to the following people for their invaluable contributions to this project:

Gail Christensen
International Technical Support Organization, Raleigh Center

Kay Sintal, Pat Byers, ??????????
IBM Netfinity Marketing

Steve Quan
TurboLinux Product Marketing

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 119 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Part 1. High availability clustering

Chapter 1. Linux high availability cluster solutions

WIth the adoption of Linux as a mature server operating system by IBM in early 2000, what had been a relatively obscure “hacking” project, suddenly became the talk of the IT world. The approach taken by IBM towards Linux has “legitimized” Linux in the eyes of IBM’s more traditional customers, and has caused these customers to think seriously about Linux for the first time.

Linux now offers an alternative server operating system and is an ideal match for IBM’s range of Intel-based servers - the Netfinity range. Netfinity offers a range of scalable and highly reliable platforms on which to deliver server functions, and Netfinity has been working closely with Microsoft to deliver Windows solutions for some years now.

In this paper we are going to study specific Linux solutions for creating clusters of machines to provide high-availability configurations: software solutions using Linux to provide higher availability using multiple machines than single server solutions. The combination of the Linux operating system, sophisticated and reliable software clustering and Netfinity hardware offers high availability at a low price. Even in an Enterprise environment, where an obvious choice for a highly reliable back-end database server - for example - would be the S/390 Parallel Sysplex environment, Linux high-availability clustering solutions can provide a reliable front-end Web server.

The two primary benefits of a Linux high-availability cluster along the lines of the solutions discussed in this paper are:

1. High availability in the sense of fault tolerance: if a single Linux server in a cluster should fail then the server function of the total cluster solution is not impacted.
2. High availability in the sense of scalability: as workload demands grow it should be possible to add machines to an existing cluster to cope with the load. This compares with a single box solution in which at some point a total hardware replacement is required to upgrade the server function.

1.1 Why IBM Netfinity and Linux?

The promise of Linux is to provide a highly reliable and robust operating system running on relatively inexpensive hardware. In a sense it represents the best of both worlds: the reliability of UNIX on readily-available Intel platforms. Of these Intel platforms, IBM’s Netfinity line of servers is something of a one-off; although the Netfinity family of products is relatively inexpensive compared to IBM’s mid-range AS/400s and RS/6000s, not to mention its large

system S/390s, the Netfinity line extends that heritage of reliability and scalability which has made IBM machines famous worldwide, and as such stands alone in the PC market.

When you combine this reliability with the veritable “army” of support personnel at IBM’s disposal, you can well imagine that when you buy a Netfinity, you’re buying more than just another PC.

1.1.1 Our audience

When we think of high availability clusters it rather begs the question as to who might be interested in such designs. After all, setting up two servers ostensibly doing the same thing does require a fair bit of effort, and may not be for everyone. Internet service providers (ISPs), some corporate intranets and perhaps even some Application service providers (ASPs) might be interested in establishing an inexpensive entry way into their domains that is capable of tolerating the failure of one of those servers. One wouldn’t want to be caught like the chocolate manufacturer whose server crashed over the Easter holiday. If you can’t afford to keep all of your eggs in one basket then you’ll probably be interested in this topic. That said, it was primarily with clients such as ISPs in mind that the scenarios for this paper were developed as shown in Figure 1 on page 5.

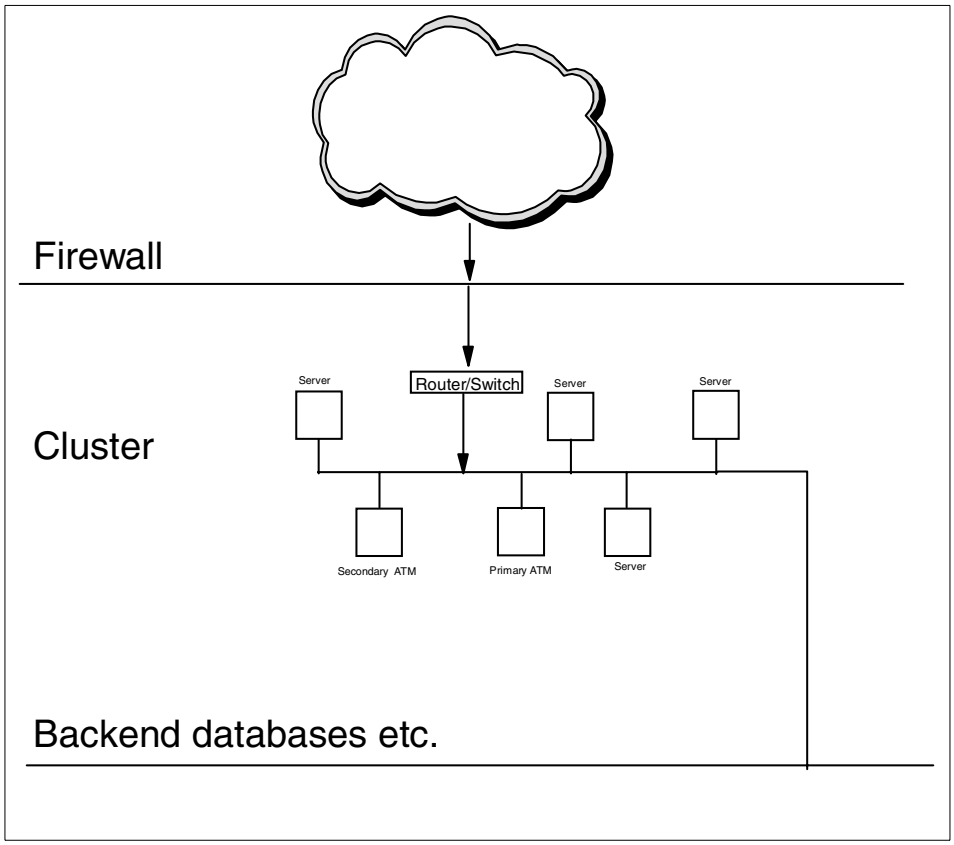


Figure 1. Linux/Netfinity cluster environment

Note that your Linux cluster is likely to serve as an entry point, or interface between your business and the outside world. The cluster resides behind a firewall but in front of your I/O intensive servers. In other words, this type of cluster will help ensure that as the volume of your online business grows you can keep up with it.

On the other hand it is important to keep in mind that once the entry into your Web site is highly available (via the solution described here) the next step you will likely take is to make the data and applications on your servers highly available. Linux solutions for data sharing are available via applications such as InterMezzo, but are beyond the scope of this paper and are not simple to implement. The view we are taking in this paper is that the actual database engines themselves are separate from the Linux cluster and that this database may already implement parallel clustering techniques of its own to present a high-performance and highly-available single image, perhaps using

S/390 Parallel Sysplex, DB2 and SAP/R3 or else another scalable UNIX server such as the Oracle (product name).

1.2 Linux cluster theory

To build, install, configure, and maintain a Linux cluster, you need some theoretical background covering cluster basics. Without that, it's a hard job to decide on the right clustering solutions and the implementation details. This chapter will provide you with the basic required knowledge about Linux clustering,

When talking about clustering we first want to say something about what clustering means at all, as there are different concepts and solutions for different scenarios or problems. We are not talking about all of these in detail in this paper; clustering covers three major topics:

- High Performance / Scientific Computing (Number Crunching)

  If somebody hears "Linux and clustering" what often comes to mind is "Beowulf". The Beowulf Project aims at High Performance Computing (HPC) using message-passing parallel programs. To really utilize the Beowulf concept, your application has to be (re)written using PVM (Parallel Virtual Machine) or MPI (Message Passing Interface). At least you should be able to run the processing in parallel using shell script front ends, so that each node works at a specific range of the whole task.

  Beowulf is not a single software package. It consists of different parts as e.g. PVM, MPI, Linux kernel, some kernel patches, etc. As this Redpaper doesn't focus on HPC, we won't cover Beowulf. For more information, please have a look at `http://www.beowulf.org/`.

- Load Balancing / Sharing (Scalability)

  This is a very important topic for any fast growing business as most of today's e-business sites are. Most of these sites start small with only a few web servers and a back-end database. So when they grow, they have to change their hardware more often, as their number of customers and the number or level of services they provide increases. Changing or upgrading your hardware means outages, downtimes, and after all lost money. That doesn't look professional nor does it provide the kind of service your business needs to grow.

  How about a cluster of regular Netfinity boxes running Web servers, sharing and balancing the load they get? So you can just add another box into the cluster if the demand or load you get increases. If one server fails, just change the cluster configuration automatically and take the broken

server out for service. Later you can reintegrate this server or a replacement box back into the cluster again. No downtime, no visible outages, no lost money.

Most of today's Linux load balancing cluster solutions are based on the Linux Virtual Server (LVS, http://www.linuxvirtualserver.org/) project. We will cover this kind of clustering and the TurboLinux Cluster Server as a major product available today in more depth later on in this paper.

Another very interesting approach to load sharing and distributed computing is called MOSIX (The Multicomputer OS for UNIX). It allows you to run processes distributed on a collection of clustered nodes transparently. MOSIX migrates processes from very loaded nodes to other less loaded nodes dynamically and scales very well. No special support from the application is necessary. It simply looks like SMP (Symmetric Multi Processing) but with more then one physical box.

To be honest, MOSIX doesn't exactly fit in any of these categories. It's something between HPC, and load sharing, and doesn't provide any additional availability at his time. That's why we won't cover it here in detail. If you are interested in additional information, just visit their web site http://www.mosix.org/.

- High Availability (HA, Fail-Over, Redundancy)

  Part of this topic was already discussed in the section above, as load balancing and sharing increases the availability of your servers. But there's one major topic left that's called Fail Over Service (FOS), the basic idea behind traditional HA solutions.

  It means, that in most cases you have two systems, one master and one standby system. In normal operation the master is running your service or application and the second system is just watching the master. If the master fails, the second system takes over the service immediately and shuts the master down (if this has not already happened). This provides you with a highly available system.

  Fail-Over-Service is also provided by the Linux Virtual Server project. One major commercial product is RedHat HA Server, covered in detail later on. The new release of TurboLinux Cluster Server will provide FOS, too.

  Right now, there's another project going on to port SGI's FailSafe HA solution to Linux. SuSE and SGI are the major contributors, but there's no real product available yet. So unfortunately, we aren't able to cover it in this Redpaper.

So, as you see, clustering is a very complex topic covering a whole lot of solutions for different problems using even more different solutions. The

following table tries to summarize the above items and to put it in a single picture.

Table 1. Cluster aspects and products

Products, Projects	High Performance Computing	Load Balancing, Sharing	High Availability	Covered by the Redpaper
Beowulf	XXX	‘		
MOSIX	XXX			
LVS[a]		XXX		XXX

a. RedHat HA Server and TurboLinux Cluster Server are both based on LVS.

This Redpaper’s focus is on load balancing/sharing and high availability, the major areas of interest of today’s fast growing e-businesses. That’s why we will only cover these two clustering aspects and leave scientific and high/performance clustering out of the picture.

Load balancing and fail-over have some things in common. So we discuss most of their common features together and point out the differences where appropriate.

Please note that while both Fail Over Service (FOS) and load balancing are provided by the Linux Virtual Server project, most people refer to LVS when talking about load balancing and FOS when they mean real high availability, and so do we.

1.3 How does clustering work?

Now we will take a deeper look at the technology that turns a bunch of powerful, rock-solid Netfinity servers into an even more powerful, highly available cluster. We will talk about heartbeat, services, networking, shared storage and other related terms.

1.3.1 Heartbeats

To form a cluster, its nodes have to communicate. First, they have to know, whether the other node or the other nodes are up and running, whether “their hearts are still beating”. This process is called the heartbeat process. The current products implement two ways for getting heartbeats:

- Ring heartbeats over serial line
- IP broadcasts over local area networks

While the serial line is a good and very reliable option for a two node system, it becomes more tricky with more nodes. Then serial line switches forming a ring structure are needed. That's the major reason for using broadcast heartbeats, e.g. using IP on Ethernet. . The next figures show the possible solutions.

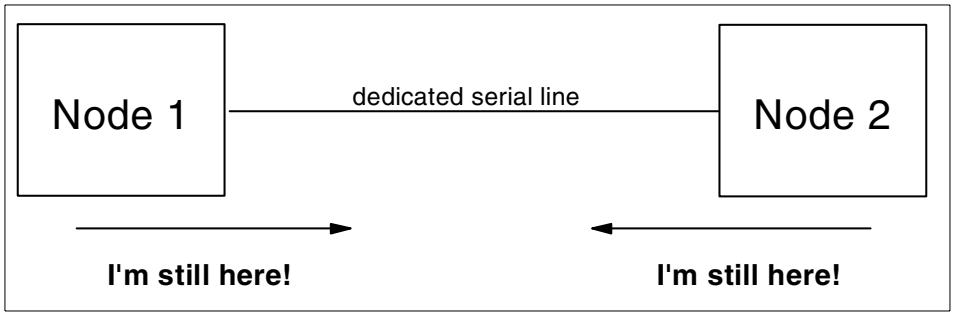


Figure 2. Serial line heartbeats between 2 nodes

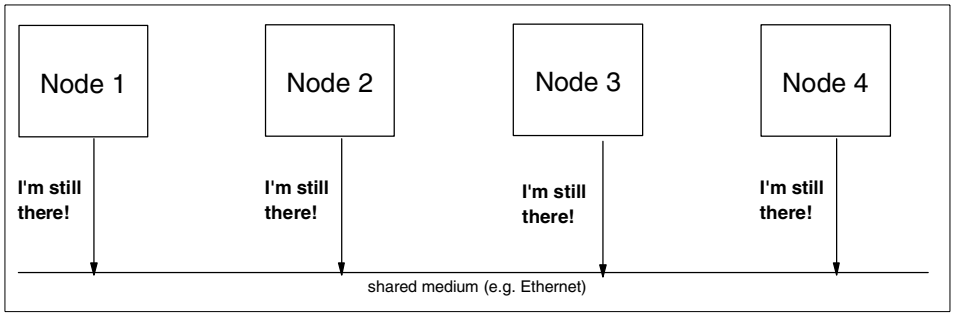


Figure 3. IP broadcast heartbeats

For enhanced availability you should consider a dedicated heartbeat medium for IP broadcasts: one in which the heartbeat traffic is separated from the regular network traffic to avoid heartbeats can get delayed or even lost due to high network traffic.

1.3.2 Networking for the real traffic

Now that the nodes know whether their neighbors are alive or not we should think about the network technology for the real network traffic. Depending on the amount of traffic and expected response times, you might choose from the entry level token-ring or fast Ethernet to the high-end Gigabit Ethernet or Myrinet connection. Switched topologies (as opposed to shared hubs) are

always preferred, as they provide higher throughput, are more reliable, and are no longer prohibitively expensive.

Myrinet is a very fast specialized network technology offering 2 Gbps speeds and low (< 10 µs) latency which has been used in BeoWulf Linux clusters and which is marketed by Myricom, Inc.

A comparison of the network technologies supported on Linux leads to the following table:

Table 2. Comparing network technologies

Technology	Reliability	Latency	Speed	Scalability	Cost
Fast Ethernet	low	poor	low (100 Mbps shared)	poor	low
Fast Ethernet (switched)	medium	poor	medium (100 Mbps dedicated)	medium	low
Token Ring	medium	medium	low (16 Mbps shared)	limited	medium
Token Ring (switched)	medium	medium	medium (16 Mbps dedicated)	medium	medium - high
Gigabit Ethernet (switched)	high	medium	high (1 Gbps dedicated)	high	high
Myrinet	high	best	high (2.0 + 2.0 Gbps)	high	high

So, for a low-cost Linux clustering solution, switched Fast Ethernet is the best compromise. It should scale well up to a certain point and offers a good price/value factor.

When going for one of the really high-end technologies as Gigabit Ethernet or Myrinet, please make sure that your servers are powerful enough to handle the traffic and don't take this for granted!

1.3.3 Virtualizing the server

Now that we have chosen the kind of heartbeat and network technology we are going to use, our servers are still individual machines. So there's still

something missing: we need to form one virtual server out of the separate hardware boxes.

While there's more then one way to accomplish this, we will only cover the "virtual IP address" mechanism, used by most of today's cluster products. Using this scheme, all incoming traffic is addressed to the one virtual IP address assigned to the cluster and all outgoing traffic originates from this same single virtual IP address. Anyone who is familiar with IBM's Network Dispatcher family of products will be familiar with much of the following discussion.

We will now split follow up this topic in two separate sections covering fail-over and load balancing, as their concepts become more different.

1.3.4 Fail-over service

Fail-over means that we most likely have two servers, one primary or master node and one secondary or backup node. Both know about each other via heartbeat and are attached to the "real" network as in Figure 4 below).

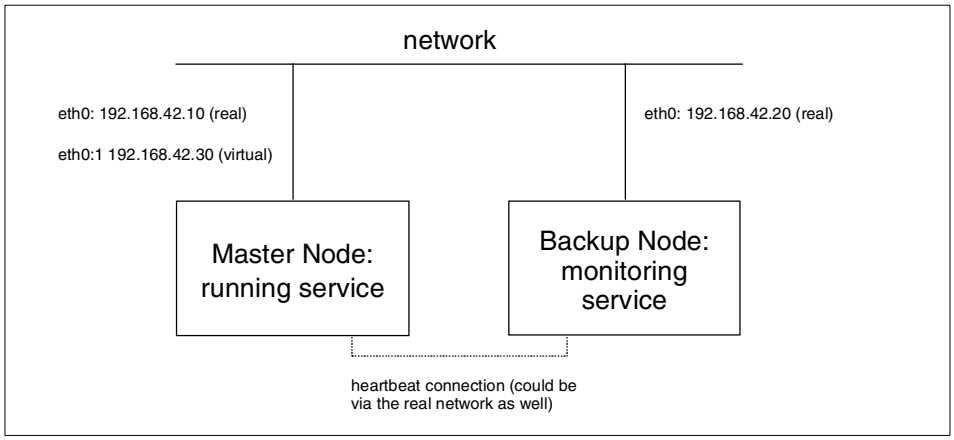


Figure 4. Fail-over service in normal operation

In normal operation, the master server is running and providing the service, a Web server for example. The backup node monitors the master such as by trying to connect to the master server's http port (80) every 10 seconds and retrieve a web page. Heartbeats are exchanged by both servers. As the picture implies, both servers have a real IP address assigned (192.168.42.10 for the master and 192.168.42.20 for the backup in this case) to their real interfaces (eth0). As the master node is active, it gets a second IP address,

the virtual cluster IP address. In Linux terms, this IP address is an alias address (eth0:1) defined on top of the real network interface (eth0).

Both real and virtual interfaces can be seen via ARP (Address Resolution Protocol), responsible for the IP to MAC address mapping. Actually, both eth0 and eth0:1 share the same MAC address, which is why eth0:1 is called an alias.

What happens if the service on the master server becomes unavailable or the server itself goes down? This situation will be noticed via monitoring (if only the service fails) or via heartbeat (if the complete machine goes down). Figure 5 shows what will happen then:

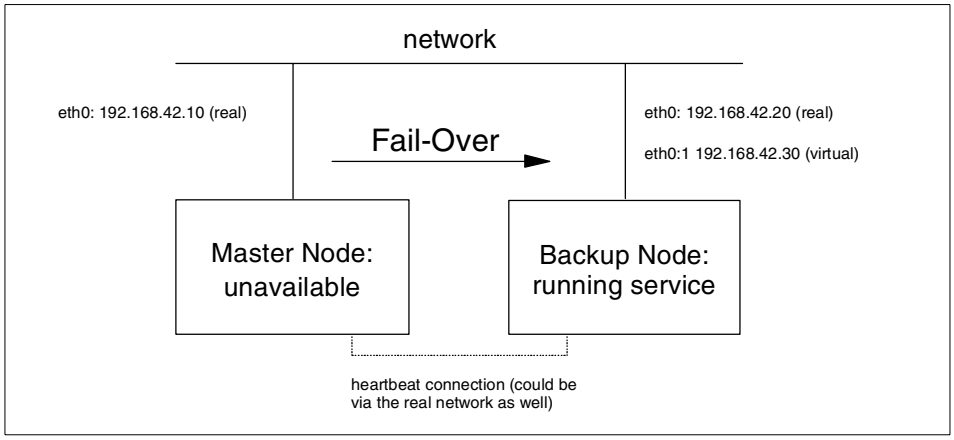


Figure 5. Fail-over service showing actual fail-over operation

The backup node takes over the virtual cluster IP address from the master node and gets its aliased eth1:0 up and running. After that it starts the service that was originally available on the master node and everything is fine again. This process is called “Fail-Over”.

As the virtual IP address is transferred to another real network interface, its associated MAC address changes too. To get this change reflected to all other computers on the network, the new active (and former backup) node broadcasts a ARP message for the IP address of the cluster containing the new MAC address. This process is known as “gratuitous ARP” or “courtesy ARP” and enables the other machines on the network to update their ARP tables with the new MAC address of the cluster.

If now the master becomes available again (observed via heartbeat), also called “Fall-Back” can take place (see figure below). The backup node stops

running the service, the master node takes over the virtual IP address, issues the gratuitous ARP broadcast and starts its service. At this time everything looks like no fail-over had happened at all.

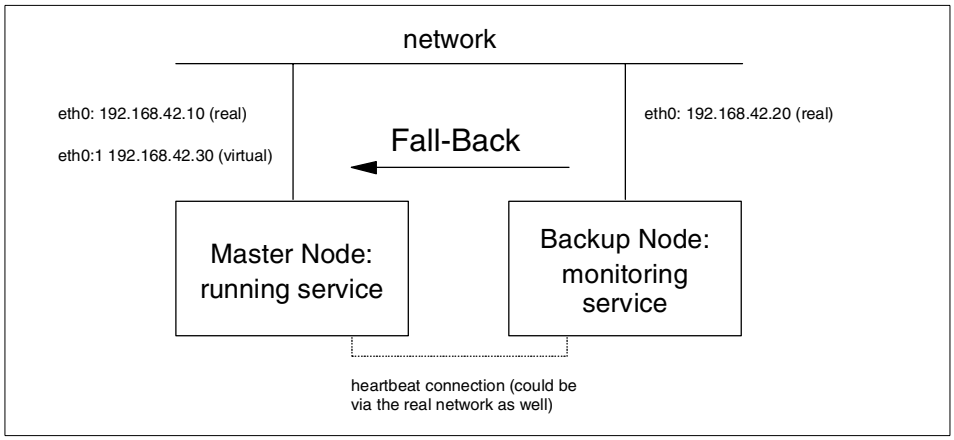


Figure 6. Fail-over service, resumption of normal operation

With the current Linux product implementations, some restrictions apply:

- Only 2 node FOS configurations are supported.
- No selective fail-overs (for individual services) are possible, all services are monitored and fail-over as a group.

1.3.5 Load balancing

Load balancing works similar to Fail-Over-Service, but aims at scalability and reducing system outages. It spreads incoming traffic to more than one server and lets all these servers look like one large server. It uses heartbeats like FOS, but implements another concept, unique to load balancing: traffic monitors or managers. A very simple LVS setup is shown in the figure below. There's no dedicated, internal cluster network, all machines are connected to the same physical network.

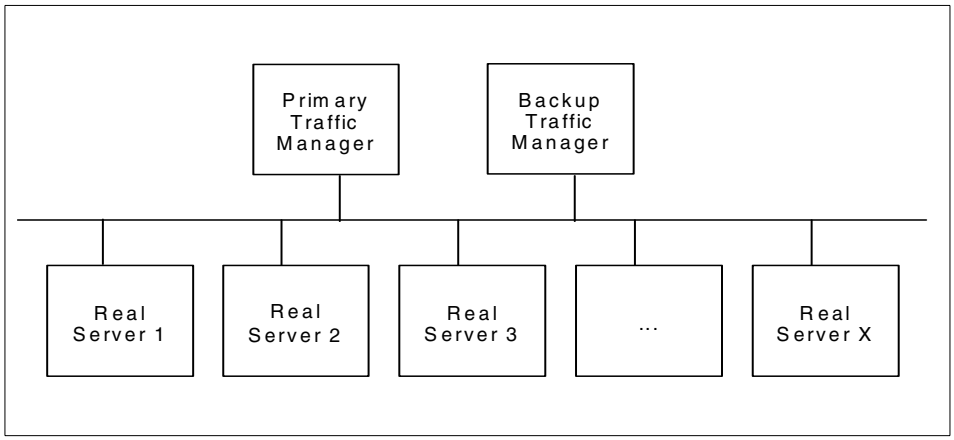


Figure 7. Simple Linux Virtual Server setup

As with FOS, there's a virtual server formed out of individual boxes. The primary and backup traffic manager behave like a FOS cluster concerning network connection and heartbeat service. The active traffic manager gets the virtual IP address assigned and redirects the incoming traffic to the real servers, based on the chosen load balancing and routing scheme. The traffic manager monitors the real servers for heartbeat, service, and load (if supported).

Scheduling mechanisms for distributing the incoming traffic can be (depending on the product):

- Round robin

  all traffic is equally distributed to all real servers

- Least connections

  more traffic is distributed to real servers with fewer active connections

- Weighted round robin

  more traffic gets distributed to the more powerful servers (as specified by the user) and dynamic load information is taken into account

- Weighted least connections

  more traffic is spread to the servers with fewer active connections (based on a user configured capacity) and dynamic load information is taken into account

Now that we have seen how the traffic manager works and how the traffic gets from the clients to the server, the next question is: How do we virtualize

the real servers inside the cluster or how do the real servers get their traffic from the traffic manager and respond to the clients? There are some options (depending on the product):

1.3.5.1 Direct routing

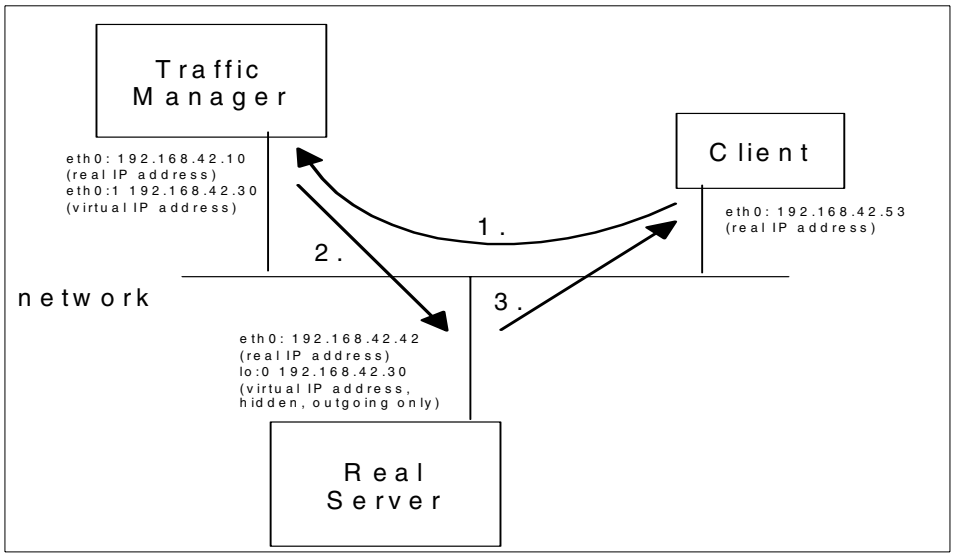


Figure 8. Direct routing of returned packets

In the figure above, the client accesses the virtual server (192.168.42.30). Its traffic gets routed to the traffic manager, which redirects it to the real server by simply changing the MAC address of the data frame and retransmitting on the LAN. The real server itself has a physical network interface (eth0) for the incoming traffic and one aliased, ARP-hidden network interface (lo:0) for the outgoing traffic. So the real server sends the response back directly to the requesting Client 1 using lo:0 as its source address, thus using the virtual IP address. From the perspective of the client, an IP packet has been sent to the virtual server's address and a response has been received from the same address. The client never sees any response to its request as coming from the server's "real" eth0 address, it only sees the virtual IP address.

The lo:0 address in the diagram above is called a "hidden" address because it must be configured in such a way that the server owning this network interface will not respond to ARP requests for the IP address. The only network device which should respond to ARP requests is the traffic manager. The traffic manager determines which actual server is to be used for the received packet and forwards the packet to the server by re-transmitting the

received packet onto the network but now with the destination Layer 2 MAC address of the packet now being the MAC address of the desired server. The server will receive the packet, because it is now destined to its hardware MAC address, and will examine the packet and discover that it contains an IP packet destined for an IP address known to the server as it's internal "hidden" IP address. It will then pass the packet to the IP application (such as a sockets application) bound to this IP address. The application will respond and the same IP address will be used as the source address in the response, and the response packet will be sent out over the network directly to the client. The response does not pass through the traffic manager.

In our cluster implementations we will primarily be examining all-Linux environments, in which the traffic managers and the servers themselves are running the same distribution of Linux code; this certainly eases implementation of the clusters but it should be noted that other operating system environments such as Windows NT, Windows 2000 or even S/390 can be used for the server environments in a Linux cluster: the only requirement is that the servers themselves must be configured with both "real" and "hidden" IP addresses in a similar manner to Linux servers.

Because only the traffic manager responds to ARP requests for the IP address of the cluster, a full implementation of a load-balancing cluster environment will include a backup traffic manager as shown in Figure 7 on page 14. There will now be an additional requirement for the backup traffic manager to maintain cluster state information such as information on the state of open TCP connections into the cluster, and this information will allow the backup traffic manager to take over operation of the cluster without disrupting existing connections.

Although not shown explicitly in Figure 7, the traffic manager function can also reside on the same physical server as one of the "real" servers: the function can be "co-located" with the server itself. This reduces the total number of machines required to implement a load-balancing cluster if this is an issue: a cluster could be implemented on only two machines, with one machine acting as primary traffic manager and the other as backup traffic manager and with the server functions themselves residing on the same machines.

This basic configuration is the easiest and fastest solution to implement, but has one major disadvantage: the traffic manager and the real servers must have interfaces to the same physical LAN segment. As traffic to the cluster increases, this may lead to congestion: each packet inbound to the cluster appears on the network twice (once to the traffic manager from outside and once from the traffic manager to the actual server) and then each response

packet also crosses the same network. It's a good idea to have a separate internal cluster network where possible like the one shown in Figure 9 next. In this network traffic between the traffic managers and the servers flows over the "internal" network, and this network could also be used for the flow of "heartbeat" information, meaning that all intra-cluster network traffic is isolated from the external "normal" network environment.

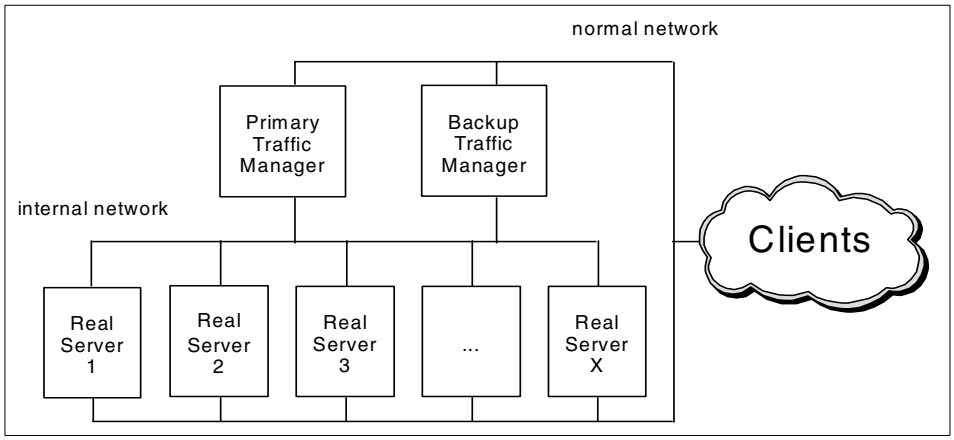


Figure 9. More sophisticated setup using an internal cluster network

1.3.5.2 NAT

Another option for hiding the internal cluster network is called Network Address Translation (NAT). NAT requires the traffic managers to take on one more job role; they have to translate the IP addresses of incoming traffic to direct it to one of the real servers and on the way back they have to re-translate the IP addresses of the outgoing traffic. Unlike the previous configurations, this requires that both inbound and outbound traffic has to flow through the traffic manager; Figure 10 on page 18 shows this process pictorially:

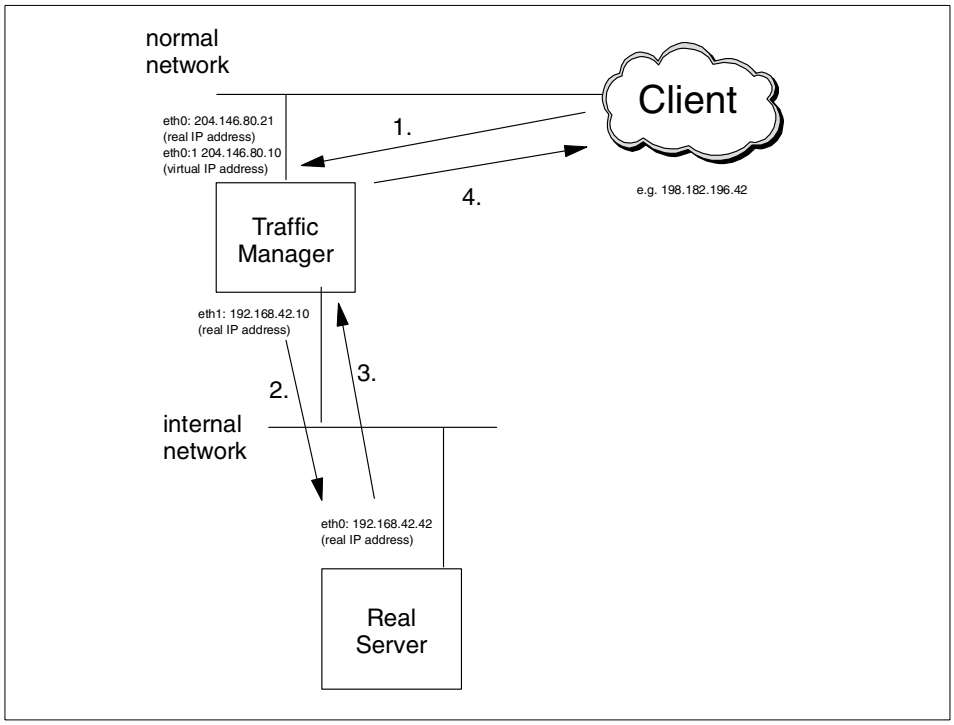


Figure 10. Network address translation

When the client talks to the virtual server represented by the traffic manager, its traffic looks like:

Source = 198.182.196.56, Destination = 204.146.80.10

Now the traffic manager selects a real server for this traffic and after translating the addresses passes on the traffic:

Source = 198.182.196.56, Destination = 192.168.42.42

After the real server did its job, it sends back the response using:

Source = 192.168.42.42, Destination = 198.182.196.56

Finally the traffic manager forwards the traffic to the outside world after a retranslation:

Source = 204.146.80.10, Destination = 198.182.196.56

The translation is done inside the traffic manager using a hash table and IP address - port mappings.

This is a very convenient way to implement a cluster because it only requires a single external IP address and that all the destination servers can be defined on an internal “private” IP network. It does have one really significant disadvantage, mentioned above: all outgoing traffic has to pass through the traffic manager now. One of the justifications for permitting inbound traffic to pass through the traffic manager in a basic cluster is that it’s usually the case that outgoing traffic is much more significant in volume than the incoming traffic, considering requests such as HTTP requests which are small in comparison to the volume of traffic sent back in response. So your traffic manager finally becomes the bottleneck of your cluster, and so NAT is suitable for a smaller cluster environment with not too much expected traffic.

And, in any case, what’s to stop the servers in Figure 8 from being configured with “real” IP addresses in a private IP network? There is no reason why even a simple cluster environment with a single network infrastructure can implement multiple IP networks over the same physical infrastructure. So NAT may not be required even in cases where multiple external IP addresses are not possible.

However, NAT has one other major attraction in that the destination servers themselves do not need to be configured with a “hidden” IP address at all. In the early days of clustering this was a problem on certain operating systems, and certainly adds complexity to the server configuration process even today. The NAT solution means that absolutely any IP server platform can be used as the target servers in a cluster without having to consider the quirks of IP addressing using “hidden” IP addresses configured on loopback interfaces.

1.3.5.3 Tunneling

Another interesting option for building up a LVS cluster is to use IP tunneling. It allows you to cluster real servers spread around the world, being part of different networks. But it needs the support of IP tunneling on each server of the cluster. Figure 11 shows the setup:

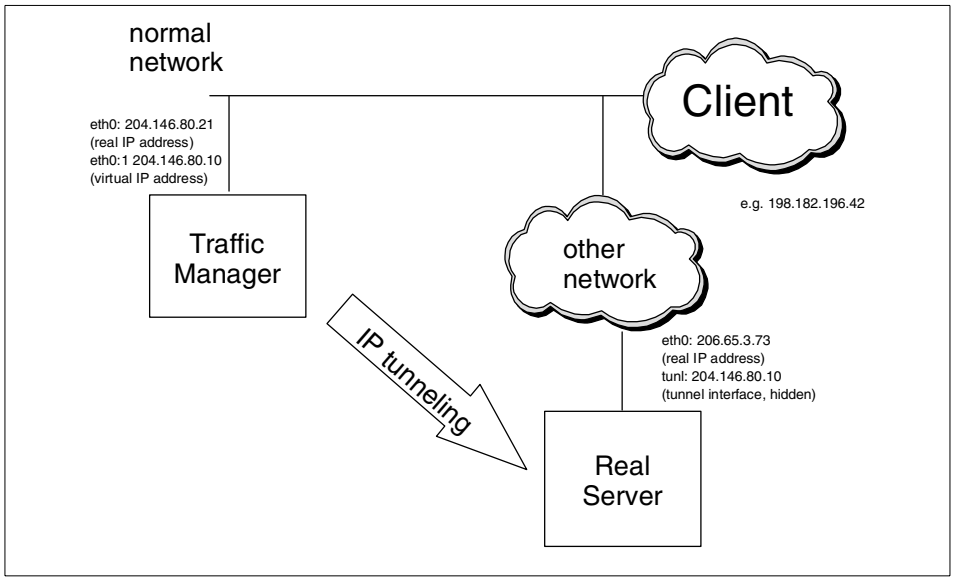


Figure 11. IP tunneling

Here, when a client accesses the virtual server this really means that the client sends a packet to the traffic manager, which advertises the IP address of the cluster and responds to ARP requests for it. Having received a packet from a client, the traffic manager encapsulates the packet into an IP datagram addressed to the real server, forwards it and stores this connection information in its hash table. All subsequent IP packets belonging to this connection end up at the same real server over the same IP tunnel. The real server itself de-encapsulates the packet and responds to the client directly using the virtual IP address as its source address.

IP tunneling is a very flexible way to build up a widespread cluster solution, but depends on the IP encapsulation protocol support of all participating cluster servers/nodes. In current implementations, this requires that all the servers be Linux servers whereas the other solutions we discussed can use a mix of server operating systems in a single cluster.

1.3.5.4 Summary

The ease you to select one of the possible three solutions explained above, we will provide you with a small table summarizing important features:

Table 3. Features of the different LVS implementations

Method	Scalability	Type of cluster network	Supported real server OS
Direct routing	medium	LAN	most (Linux, FreeBSD, NT, IRIX, HPUX, Solaris, S/390)
NAT	low	private	any
IP tunneling	large	LAN/WAN	Linux only

1.3.6 Services and monitoring

Now that we spoke about the different principles behind FOS and LVS clustering, which services are supported and how does the monitoring of these services work?

Basically all IP services using direct socket connection can be implemented with the current Linux clustering solutions. Here are some examples:

- http
- ftp (inetd)
- smtp
- pop
- imap
- ldap
- nntp
- ssh
- telnet

Services depending on a secondary port connection besides the listening port are not supported.

Monitoring only happens in a FOS solution, while LVS may use load information as feedback supplied by the real servers. To monitor means to very that a service is still available, and the easiest way the achieve this is to try to connect to the port of the specified service. If you get a connection

everything is fine, if not, the service is gone and you need to fail over. A more reliable approach is to connect to the port, send some string, get some response, compare this response to the expected result, and if it's different, fail over, otherwise everything's fine. While the later method works only with services sending back kind of clear text, the first mechanism probably works with any service.

1.3.7 What about the data?

We are talking about services a lot. But what about the data they serve? Where do the Web pages reside to which a Web server provides access? People want to access their mail through POP, but where are their mail folders stored? That's another very tricky part of clustering: sharing of data or common access to data. There are, as always, multiple solutions, mostly depending on the frequency of changes to your data and the amount of data involved. Let's start with the simplest solutions:

1.3.7.1 rsync

If your content is primarily static Web pages (contact information, for example) or a reasonably small ftp site, you can store all the data locally on each of the actual servers. Then to keep the data synchronized you can simply use a mirroring tool like rsync that runs periodically - say twice a hour. With this solution you get good availability, as all data is stored on each server individually. It doesn't matter if one server goes down for some reason, nor do you rely on a central storage server. Figure 12 shows this solution:

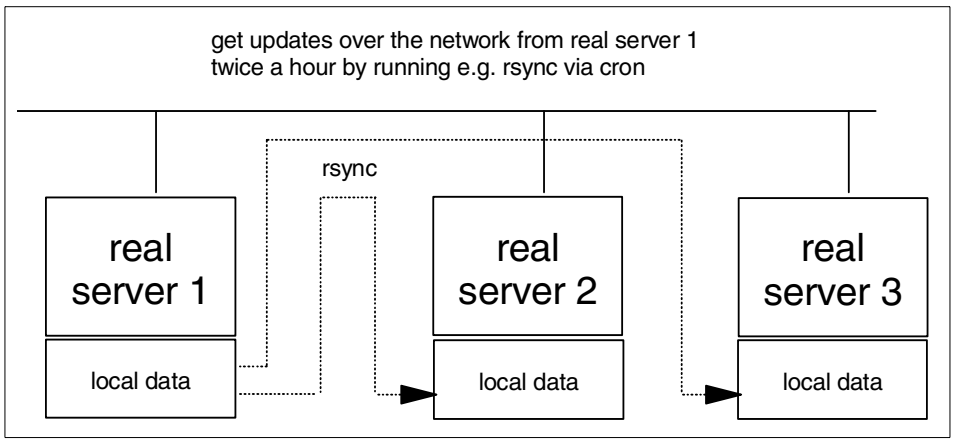


Figure 12. Using rsync for local server data synchronization

But this solution will not be suitable if you have really large amounts of data (more than a few gigabytes) changing more often (more then a few times in a

week) and you have to keep it synchronized. Now network or distributed file systems come into the picture.

1.3.7.2 Network File System

Again, starting with the simplest approach, we can NFS, a widely used, commonly know, stable network file system. It's easy to use and requires a central NFS server that exports the shared data. This data "mounted" by the real servers across the network. This approach looks like the one show in Figure 12:

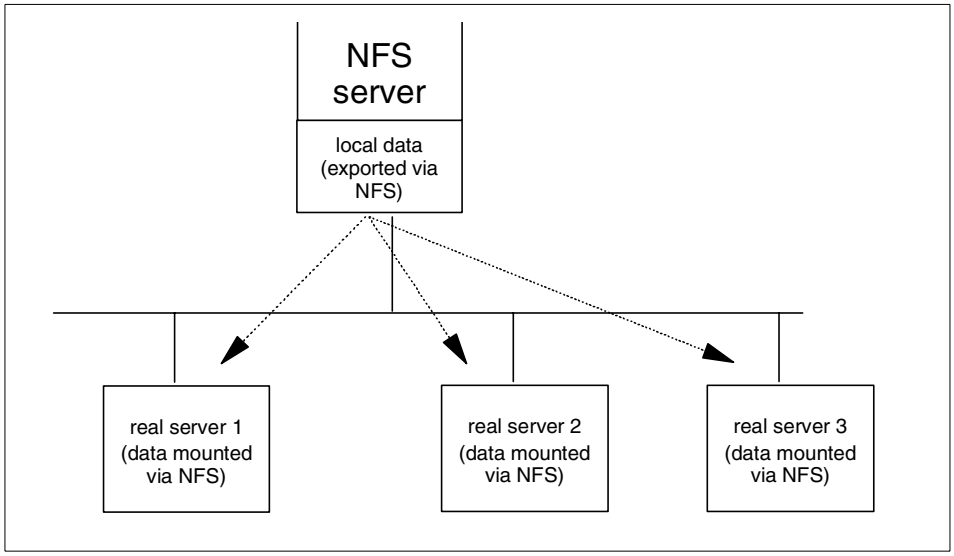


Figure 13. Using NFS for central data storing

Although NFS is simple to implement and use (on Linux), it has two major drawbacks:

- Slow performance
- Single point of failure

Although the performance may be acceptable for a small cluster solution, you should always be aware that if the NFS server dies then the real servers will no longer be able to get to your data and therefore will not be able to provide their service. This might make you think about setting up a redundant, highly available NFS server, but that's no trivial thing to attempt: you have to take care of the clients' file handles, keep the clustered NFS servers synchronized and there is no "out of the box" solution here. So after all: NFS is no real solution for a cluster environment.

That's why there are real cluster capable file systems like the Global File System (GFS) and Intermezzo, a different approach to a cluster file system.

1.3.7.3 Global File System

Let's have a look at the Global File System first. GFS implements the sharing of storage devices over a network. This includes Shared SCSI, Fibre Channel (FC), and Network Block Device (NBD). The Global File system sitting on top of these storage devices appears as a local file system for each box.

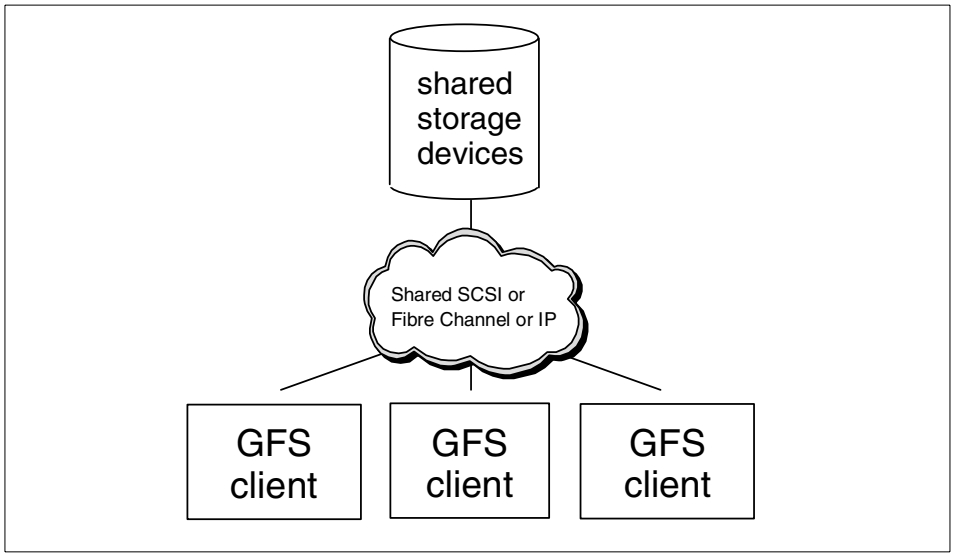


Figure 14. GFS

The Global File System is a 64 bit, shared disk file system focusing on:

- Availability; if one of the clients goes offline, the data can still be accessed by all the other GFS clients
- Scalability, which means it doesn't suffer from concepts based on a central file server, like NFS does

Furthermore GFS is able to pool separate storage devices into one large volume and to do load balancing between the workload generated by all the clients.

To setup GFS, you need to decide on the transport medium to use:

- Shared SCSI (a pool of disks physically attached to more than one computer), which gives you a low cost, but not very scalable solution using already deployed equipment

- Fibre Channel (providing you with a storage area network, to that the individual devices get attached), that is a real large scale setup and thus more expensive
- IP (akin to using tunneling to attach to your client over a traditional network), not yet a widely-used option and one which is limited by the network bandwidth but is one which allows you to attach any client without direct FC or SCSI connection to your storage pool.

GFS itself implements the storage pooling, the file locking, and the real file system. It's still under development, but should already be quite usable.

1.3.7.4 Intermezzo

As opposite to GFS, which is a shared file system, Intermezzo is an implementation of a distributed file system. This means that there's no central storage pool, but each machine has its own kind of storage locally. Their storage gets synchronized via traditional TCP/IP networks. Intermezzo still features a client-server model. The server holds the authoritative data, while the clients only have a locally cached version of the data, which is kept synchronized. Intermezzo even supports disconnected operation and is able to reintegrate when connected again. Figure 15 shows a simple Intermezzo configuration:

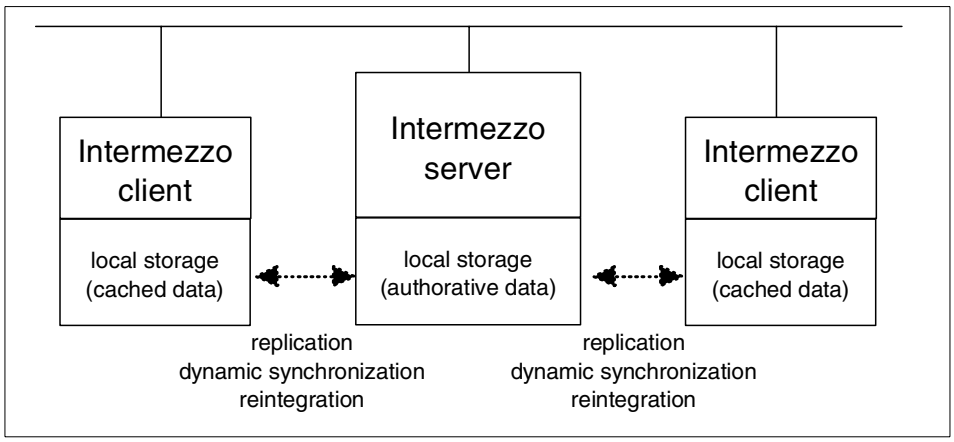


Figure 15. Sample Intermezzo setup

Intermezzo uses a traditional file system as ext2 to hold its data and puts a layer in between that is responsible for journaling updates and keeping the data synchronized. Please note that Intermezzo like GFS is still under development, but already usable (it almost certainly requires a Linux kernel recompilation to implement it today).

1.3.7.5 Backend database

Another option for storing and accessing your data, and one that might already be in place, is a backend database, like DB/2. This database itself can be highly available, but that's not part of our Linux clustering solution. All that your real servers have to be capable of is to connect to this database and put data into it or get data from it using, for example, remote SQL queries inside PHP featuring dynamic web pages. This is a very convenient and widely-used option. An example of such a configuration is given in Figure 16; consider the backend database as the existing enterprise database server running on S/390 or other UNIX platforms, for example:

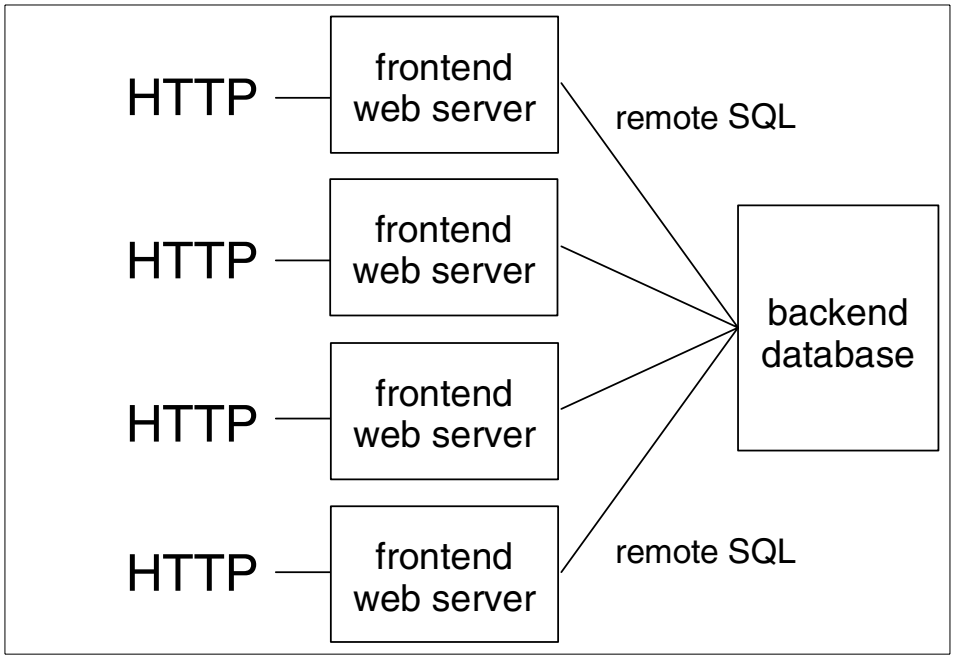


Figure 16. A cluster of front-end servers in conjunction with a fault-tolerant back-end database

1.3.8 Putting all together

After discussing the different aspects of clustering, we now put all these things together to get kind of a complete picture. The first question to ask is:

Why do I want to do clustering? The possible answers are:

- We want a scalable solution. So we go for Load Balancing.
- We want a highly available solution. So we go for Fail-Over-Service.

The important things about load balancing are:

- Make sure your services are able to run in a cluster environment (single listening TCP/IP port) and can be balanced (servers can act in parallel).
- Keep your system scalable from the point of network technology and cluster implementation.
- Think about a backup traffic manager. Otherwise all your real servers get useless if the traffic manager dies.
- Based on the amount of data and change frequency, select a appropriate method to access and store your data.

Talking about high availability consider the following thoughts:

- Make sure you can monitor the services accordingly.
- Think about storing and accessing your data safely and available. There's only little sense in building up a HA web server if the database it connects to does not offer comparable high availability.
- Be paranoid (up to a certain point), e.g. consider a second, backup internet provider if you want to offer internet services. Otherwise you may end up with a really highly available internet service locally that is not accessible if your provider goes offline.
- Think about HA from the hardware side (UPS).

Sure, there's a lot more things to talk about, but not really related to Linux clustering especially. You may e.g. want to set up your HA cluster spread among two physically separated buildings in case of fire or flooding or things like that.

One way to think about what kind of setup you need, is to know what kind of disasters who want your solution to survive and how much money you may loose from such an event otherwise. Finally you should consider how likely such a event may happen and what other services might be affected, that don't allow your solution to survive. Don't forget to test your setup on a regular basis

1.3.9 Where to go for more information

- Linux HA project: `http://linux-ha.org/`
- Linux Virtual Server (LVS) project: `http://www.linuxvirtualserver.org/`
- RedHat HA server project: `http://people.redhat.com/kbarrett/HA/`
- Global File System (GFS): `http://www.globalfilesystem.org/`
- Intermezzo: `http://www.inter-mezzo.org/`

- Please RTFM and don't forget about man pages, mailing lists, news groups, web search engines, ...

1.4 Some questions:

1.4.1 What are some of the applications that a cluster server would be useful for?

- Web Servers
- Secure Web Servers and E-Commerce Sites
- Mail Servers
- FTP Servers
- Telnet Servers
- Most other standard and custom IP based client/server applications

1.4.2 Is HA Clustering the same as Failover Clustering?

With failover solutions, only one instance of a particular application and configuration is usually active within the cluster. When that application or server fails, it is simply restarted on another node. This is what is meant by "failover". With HA Clustering, the same application and the same configuration is active on all the nodes. Because of this, there is nothing to "failover" - the applications are already active.

1.4.3 How is HA Clustering different from Beowulf clusters?

Beowulf is a parallel processing cluster system used mainly for scientific applications. HA Cluster systems are intended to be used for web servers, mail servers and other services.

1.5 Implementing a Linux cluster

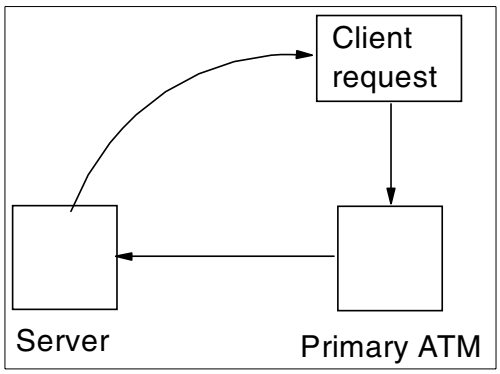
1.5.1 TurboLinux TurboCluster Server 4.0

1.5.1.1 What is TurboCluster Server?

TurboCluster Server is an Enterprise-class clustering solution for Linux. It allows the construction of a complete cluster environment (both the traffic manager and server components as shown in Figure 8 on page 15) on Intel servers such as Netfinity. The TurboCluster also allows existing servers, such as those implemented using Windows NT or Windows 2000, to be included in the "server farm".

A combination of TurboCluster software and Netfinity hardware allows the creation of a stable, reliable and scalable load-balancing server environment. TurboCluster Server includes integrated Application Stability Agents for popular protocols such as Web traffic, mail, ftp and news services which monitor the status and health of the individual servers to allow the Advanced Traffic Manager node to determine the most appropriate target server for each receiving client service request. It also provides a framework allowing network administrators to customize and install their own Application Stability Agents for other services than the basic ones provided.

1.5.1.2 Concepts



ATM = Advanced Traffic Manager

Note that ATM “routes” incoming traffic to available servers

Figure 17. Logical view of request through ATM to server, and back to client

A client request comes into the network via some router/switch, directed to ATM, which in turn directs the request to an available server (say a Web server) using some kind of scheme which may be as simple as round-robin or may be based on actual end server load statistics. The server in turn responds directly to the client. In other words the response does not go back through the ATM, but directly to client. This is an important feature, since in other distributions the request is routed back out through an ATM, which can “bottleneck” the solution.

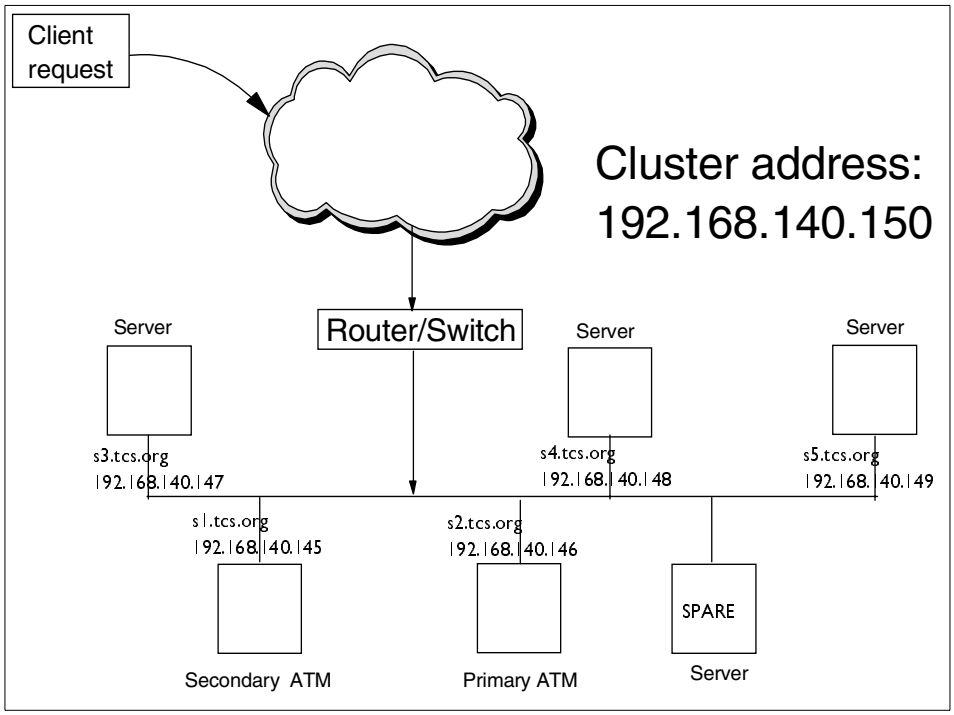


Figure 18. Our TurboCluster network diagram

In our test environment we actually had our clients on the same network as the servers; in a real environment the clients would connect across the Internet and reach our server network across a firewall and router/switch.

1.5.1.3 How we set it up

The purpose of this section is to document in outline the steps we took to set up our own TurboCluster. More details are provided in the product documentation, and we are not trying to duplicate that work here. The key aim here is to point out the choices we made, explain these and refer the reader back to the architectural discussion earlier in this paper for more information.

1.5.1.4 Starting the installation

Once you've collected the hardware and networking information for each machine in the cluster, installation starts just like a regular Linux installation on each machine. Each machine should be configured with the base Linux distribution; in the case of TurboLinux 4.0 this base TurboLinux distribution is supplied as part of the TurboCluster product but the next release of TurboCluster Server will be independent of the base Linux distribution

installed on the server. This initial installation process should simply consider each server as a stand-alone server, so each server should be configured with its own network interface addresses and not with the cluster address at all. Cluster configurations come later.

In our particular installation we used static IP addresses with each host name defined in the `/etc/hosts` file on each machine. We also set all the machine clocks to GMT, but consideration should be given to synchronizing the clocks on all the machine, perhaps by using one of the Linux machines as a master time server (using the network time protocol, ntp).

We are not going to show the base Linux installation process in this paper, our examples which follow start from the point at which the base Linux operating system has been installed on all nodes in the cluster. The only action we took after the base installation had completed on each node was to disable services which we knew we would not be using in our cluster, PCMCIA and SQL for example. This is purely good practice.

1.5.1.5 Configuring IP addresses: the loopback trick

From the outside, the cluster appears as a single entity with a single IP address, but the reality is that the cluster is composed of multiple machines. To ensure that the machines in the cluster are “cluster aware”, all of the machines must also be aware of the cluster IP address, but not be able to respond to it, and to achieve this with Linux the servers within the cluster are configured with the cluster address as a loopback address at the same time as being configured with their “real” addresses on the actual network interface. This configuration actually takes place automatically as part of the cluster configuration process we’re going to describe next, but the cluster IP address should be identified before starting this process as an IP address which is not being used by any of the servers as its primary IP address.

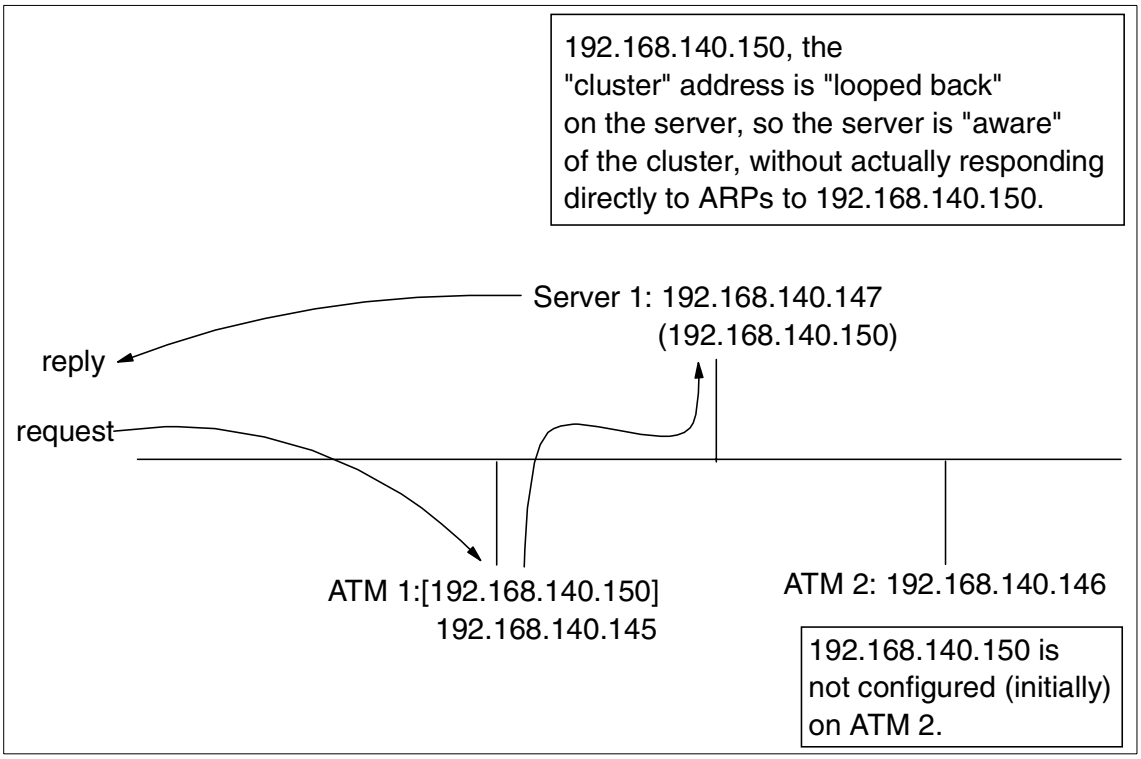


Figure 19. The loopback trick

Figure 19 shows some of the addresses we used in our network: Server 1 is configured to have IP address 192.168.140.147 on its Ethernet interface eth0 but is additionally configured to have IP address 192.168.140.150, the cluster address, on its loopback interface. This all happens automatically as part of the TurboCluster installation process, so as long as we have installed Linux on all the machines, know their unique IP addresses and know the IP address we are going to use for the cluster itself we can proceed to set up the cluster.

1.5.1.6 Configuring the cluster

A nice feature of the TurboCluster environment is that the cluster configuration file is the same for each machine in the cluster. All IP addresses and roles of all machines are included in the one file, which is read by each machine when it starts up: comparing the contents of this file with the machine's own fixed IP address allows each machine to determine its role in the cluster and configure itself accordingly. The cluster configuration task is made much easier, since the cluster now only needs to be configured once, on any one machine in the network, and the configuration file is then "pushed"

to all the other servers in the network, which can be reloaded and automatically configure themselves appropriately for their roles in the cluster.

Note: In the product documentation, cluster controllers are referred to as ATMs (Advanced Traffic Managers), routers, and cluster controllers with reckless abandon...

We chose to set up the primary ATM first in our network, as it happens. For two ATMs, three IP addresses are needed: one for each ATM, and then a “spare” that initially resides on the primary, but will transfer to the secondary when the primary fails. We established a private IP network between our ATMs and nodes, but also provided each ATM with a connection to the outside world for management etc.

1.5.1.7 Editing turbonetcfg

To configure your cluster, you will need to enter information about the machines and their relationships. You do so in the turbonetcfg file. To get to this file, type `turbonetcfg` at a command prompt. You will see a screen similar to that shown in Figure 20.

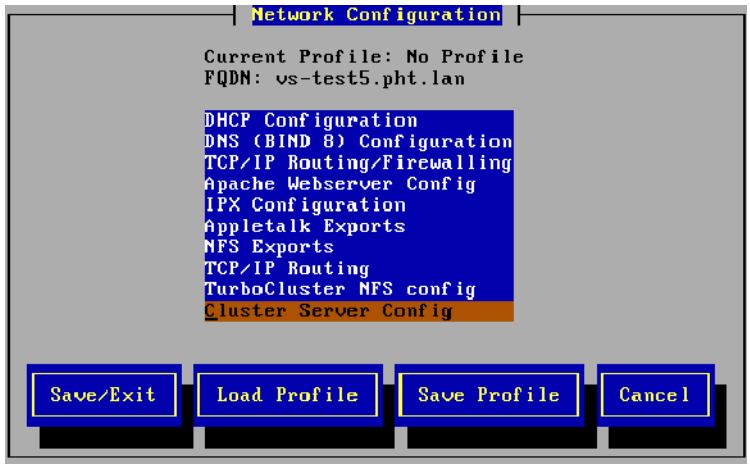


Figure 20. turbonetcfg interface

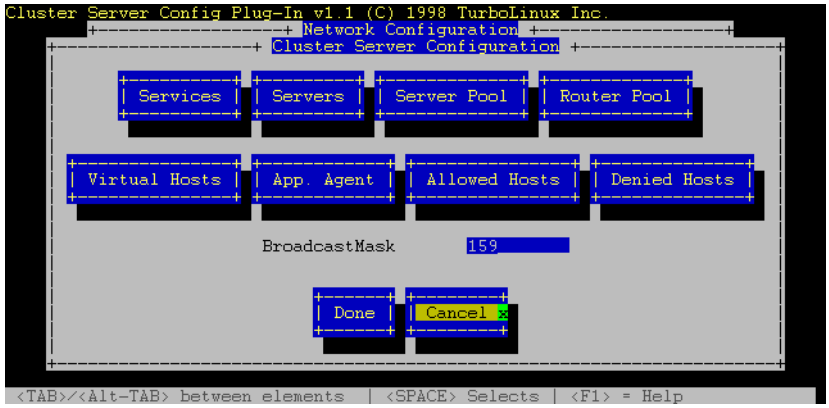


Figure 21. Start by configuring the services you will require

The cluster is then configured in a sequence of logical steps. We started with Figure 21 and went through the menu items from the top left of the screen to the bottom right of the screen, so our first step was to define the service which the cluster will be providing, in this case a Web server which uses tcp on port number 80:

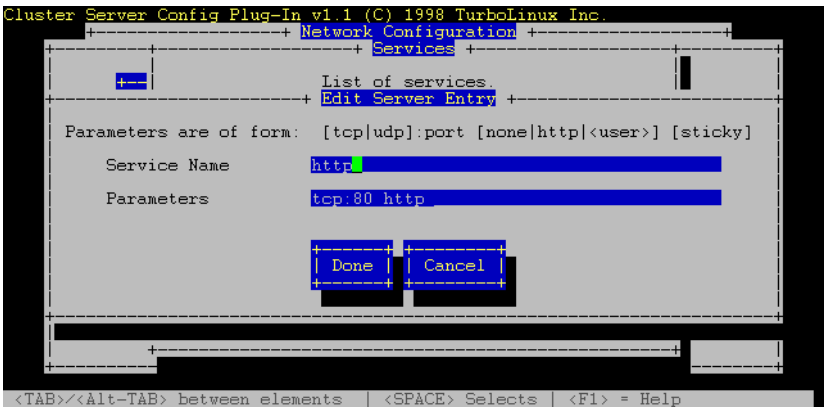


Figure 22. Define a Web server in the cluster

Having defined the Web server service, the next step is to define the actual servers which are going to be providing the service. The servers can be defined as one of three types:

1. tunnel, which means that the server must be a Linux server and that the traffic manager will forward traffic for the server over an ipip point-to-point tunnel

2. direct, which will apply to the majority of configurations, where the traffic manager will forward traffic for the server directly over the LAN
3. local, where the server is co-located on the same machine as the traffic manager

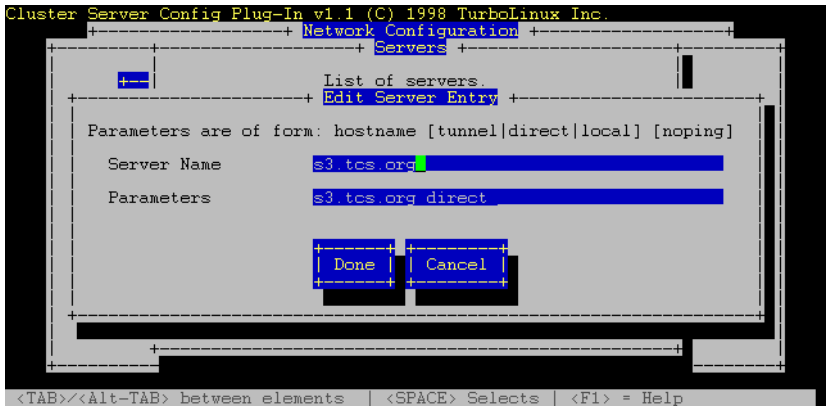


Figure 23. Define one of the actual servers in the cluster

Having defined all the Web servers in our cluster (s3, s4 and s5 in our case) we then proceed to define a server pool, which we call “webservers” in Figure 24. We took the default values for the delay and time-out parameters on this screen, which are probably appropriate for the directly-connected environment we had.

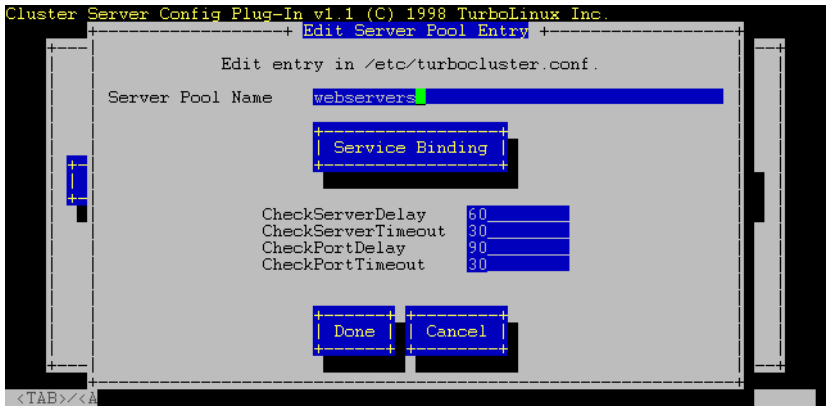


Figure 24. Define a server pool and give it a name

Having defined a pool and given it a name, Figure 25 shows the screen we used to add a server to a particular pool. We defined all our servers with the same value - 10 - for “weight’; these values have no explicit meanings but are

used to distribute traffic based on relative weights between servers, so if we had given a value of “20” to a particular server we would have seen that it would be given twice as much traffic as the other servers in our cluster.

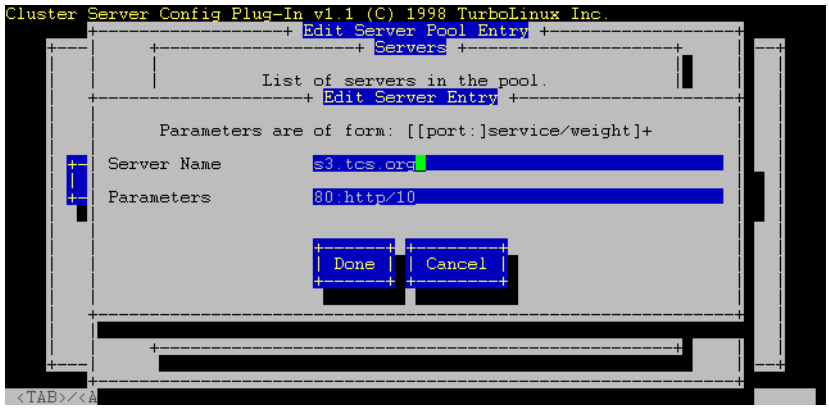


Figure 25. Add a server to a pool

Figure 26 simply shows all the servers in our pool once we had repeated the last step for all three servers.

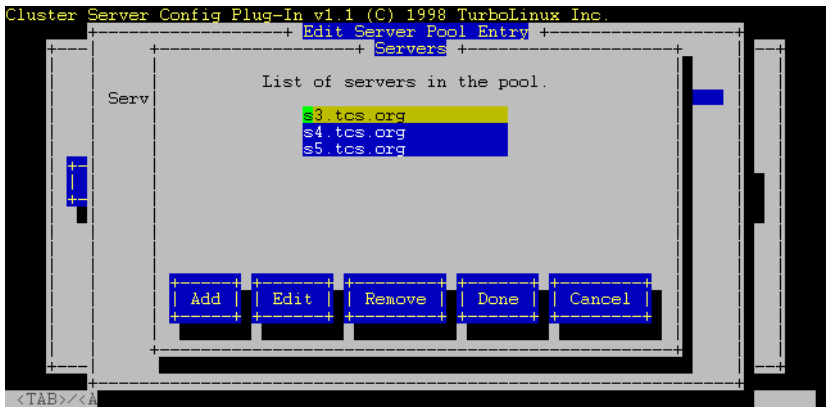


Figure 26. Show all the servers in the pool

Having defined all the actual servers, the next step is to define the “routers”, or the Advanced Traffic Manager systems themselves. In our network we define a primary and a backup ATM, so Figure 27 shows how we defined a pool to encompass all the ATMs; the values on this panel have no defaults and should be set based on the expected load for the cluster.

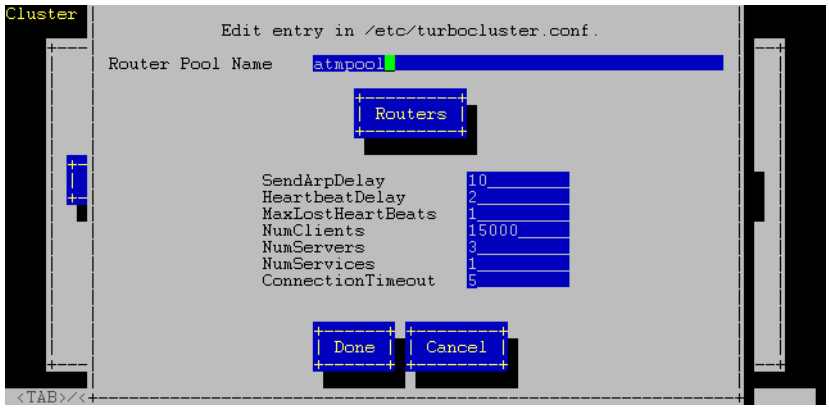


Figure 27. Define a traffic manager pool

Figure 28 then shows how each of the ATMs is added to the pool, in our case these ATMs are s1 and s2.

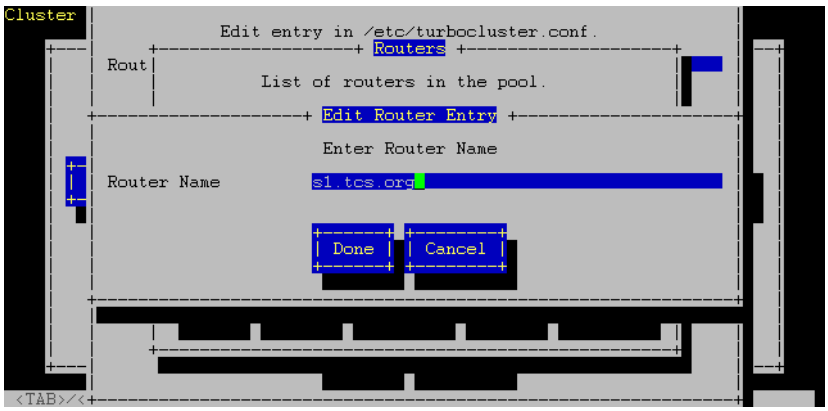


Figure 28. Add a single ATM to the pool

Then, in the same way as for a server pool, Figure 29 shows the list of all the ATMs in the pool of traffic managers once we have defined both s1 and s2:

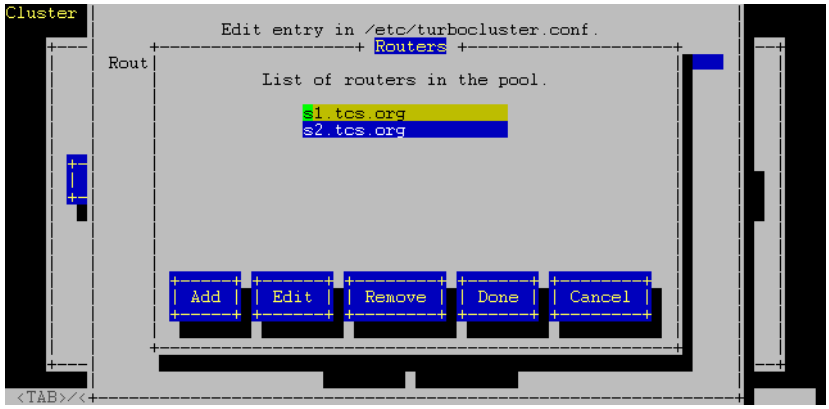


Figure 29. All the ATMs in this cluster

This is almost all the configuration work that is required; all that remains is for the IP address of the cluster itself to be defined, which in our network is 192.168.140.150. We chose to use a name of “s.tcs.org” in our network, which required us to define this name in the `/etc/hosts` file of our client computers (or we could have used DNS) but we could equally well have configured the raw IP address of the cluster itself in Figure 30 and used this explicitly.

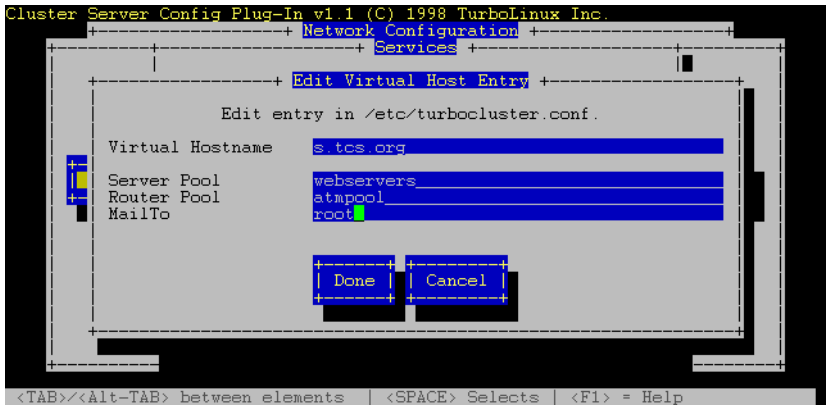


Figure 30. Defining the IP address/name of the cluster itself

Having completed the configuration for the cluster, remember that this work was only performed on one of the machines in the cluster. It doesn’t matter which of the machines we chose to perform this configuration work, but now we can use the nice facility to “push” the configuration information for the cluster out to all the other machines in the cluster. Figure 31 shows the panel

which appears, which simply serves to confirm the names of all the servers in the cluster - we can see the three Web servers themselves followed by the two ATM servers in the list. Assuming that all these machines are actively connected to the network, pressing “Start” will automatically push the configuration information out to all the other 4 machines in the cluster and restart the cluster services on each machine immediately.

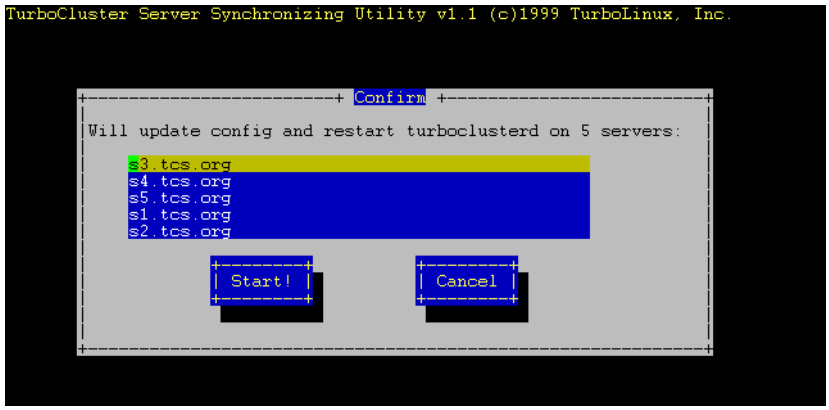


Figure 31. Pushing out configuration information to all the members of the cluster

The `turbocluster_sync` (case sensitive) command can also be used to push this configuration information to the other servers and “routers” in the cluster.

Attention

Contrary to the product documentation, whichever machine you put first in turbonetcfg will be the primary ATM, which means for example that if the primary goes down and the secondary takes over, when the primary comes back up, it will take over, or re-establish itself as primary.

1.5.1.8 Summary

We ended up with two ATMs and three servers. Using two ATMs means that we have no single point of failure in our cluster other than the networking infrastructure itself: if the primary ATM should fail then the backup ATM will take over all its sessions. One of the roles of the ATM is to maintain a table of active sessions established between clients and servers, which ensures that later packets on the same session get directed to the same server. In order for the backup ATM to take over from the primary ATM in case of failure, this state information is automatically synchronized between primary and backup ATMs. So when the backup ATM takes over the role it will issue a gratuitous

ARP and then appear to all intents and purposes just as the primary ATM did, with no impact to existing sessions and no awareness of the failure by the clients. Of course, client TCP sessions to specific servers will be broken if the actual server in the cluster should fail, but users will automatically be redirected to another Web server and in many cases will not even notice the failure: in the worst case the user will at least be able to re-submit his transaction and not be presented with a “server down” failure message which may well cause him to take his business elsewhere immediately.

1.5.1.9 How we used it

The main point about the high-availability cluster is that it is transparent to all the users: users connect to a single name or IP address and receive service just as if a single machine were serving all their needs. So it’s not immediately easy to demonstrate the cluster in action! Also, the standard caveat “your mileage may vary” must be made here - our tests were made in a particular test environment and many of our test components do not necessarily mirror real world cluster environments, and in particular our test clients were directly connected to the same switch to which all the servers and ATMs were attached. But we were able to make comparisons between connecting to a single server directly and connecting to a cluster instead.

Our first test of the cluster used MicroSoft’s Web Application Stress V1.1 load tester, available from their Web site, to see how the load balancing feature of TurboLinux TurboCluster Server handles multiple requests.

From an NT client running a stress test script it appeared that the ATM was sending requests to our servers 3, 4, and 5 respectively. That is, as a request would come in from the client, if 3 was busy processing a request, then the ATM routed the next request to 4, then to 5 and so on. This was observed via a simple `netstat` command, which was executed on each of the servers while the test was running.

Another test was run with Microsoft’s WCAT (Web Capacity Analysis Tool). Our first test with this tool was to stress the Web server directly, but as Table 4 shows, the total time from when the first byte to the last byte of a request (TTFB and TTLB) was received improved dramatically (less than half) when going through the ATM. Again, using the `netstat` command, we noticed that the ATM was distributing the load amongst three servers, no doubt accounting for the improved times. This tends to suggest that via ATM the servers are not only receiving requests more rapidly, but that throughput is also significantly enhanced. This makes sense when you consider that a

server not being bombarded with requests will likely have more resources available for processing.

Table 4. Comparison using WCAT

	Page Summary	Hits	TTFB Avg	TTLB Avg
via ATM	GET/	4685	82.08	82.80
direct to server	GET/	4157	167.45	171.61

1.5.2 Red Hat High Availability Server 1.0

We also took the opportunity to use a different software distribution, still based on the same Linux Virtual Server architecture. The Red Hat Server is more recent that TurboLinux TurboCluster Server 4.0 but it should be pointed out that a new version of the TurboCluster Server will shortly be available and that we simply did not have enough time to test a beta version of this code ourselves during the preparation of this paper.

1.5.2.1 How we set it up.

We chose to setup a fail-over solution as opposed to the load balancing solution we implemented using TurboLinux, but again this should not imply that the RedHat software can't also implement a load balancing solution.

Our setup looks like the following figure:

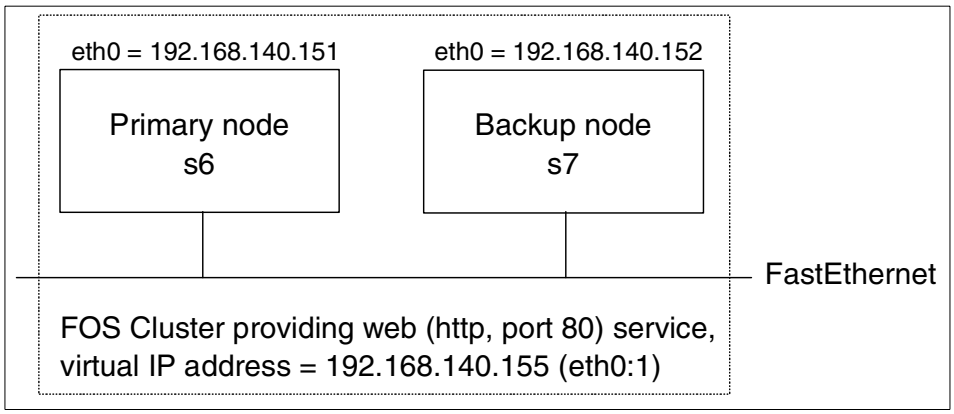


Figure 32. FOS setup/scenario

The installation process itself is straightforward and involved booting our servers from the installation CD, selecting the "Install" option (the "upgrade" option is not yet used, in fact, given that this is the first release of the

high-availability server) and we performed no customizing of the package selection, allowing all the “standard packages and services” to be installed.

Since this is a new product, we also chose to upgrade the installation by using the updates available from the RedHat ftp site.

Some minimal configuration pre-requisite work was required: we had to set up `piranha-passwd` <your password> on all cluster machines and had to enable remote network access by editing /etc/hosts.allow and .rhosts for root.

Each time the configuration files are set up or changed, they must be copied to all nodes in the cluster. Either rcp (in the case of rsh) or scp (in the case of ssh) will be used to perform the actual copy. Note that this copy is done as root. You must decide whether to use rsh or ssh, and although the installation panels allow for either you will have to install ssh manually (using rpm) if you want to use it because it is not installed by default. SSH should be used unless all the network access will be over a private, trusted network.

After setting the initial piranha password, connect to one of your servers pointing your browser to `http://<hostname>/piranha`. If this doesn’t seem to work, make sure it is actually running the http server; start this server by issuing `/etc/rc.d/init.d/httpd start` if not.

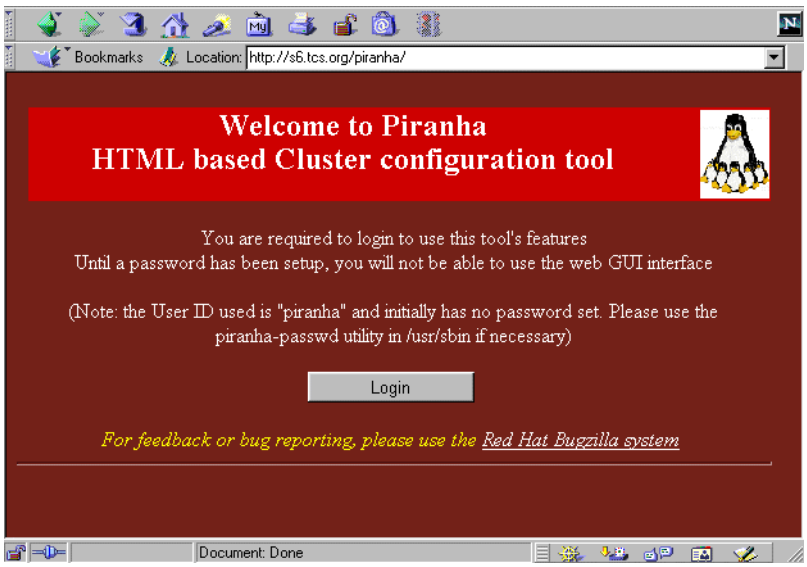


Figure 33. Piranha welcome screen

Now you can login using the previously-defined password.

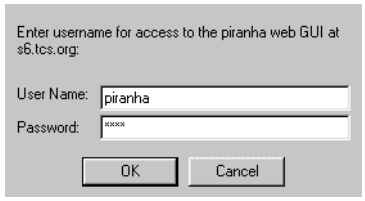


Figure 34. Logging on to Piranha

Now you will see the control/monitoring page of piranha as in Figure 35. Since we haven't setup anything yet, there's nothing running and the daemon's stopped.

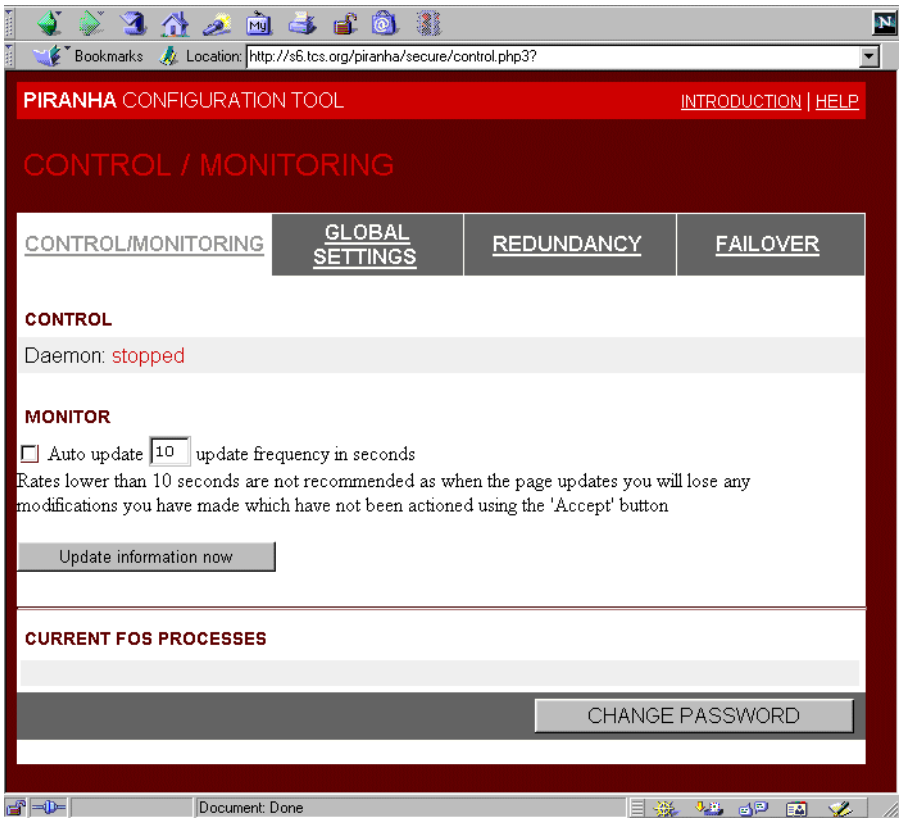


Figure 35. Control/monitoring

To configure the cluster, go to the "Global Settings page" and select the type of cluster environment to be configured. We chose "fos" (fail-over) or rather than "lvs" (load balancing) for our example. In this environment there will be a

primary node and a backup node; just as for the load-balancing environment there will also be a cluster IP address which is adopted by the active server as well. First of all, provide the ip address of the primary server/node (we used 192.168.140.151 = s6) and select rsh or ssh for performing file synchronization. As mentioned earlier, you'll have to install ssh separately if you want to use it but there are security advantages to doing so Click "Accept" to save your settings. Figure 36 shows the configuration information we provided:

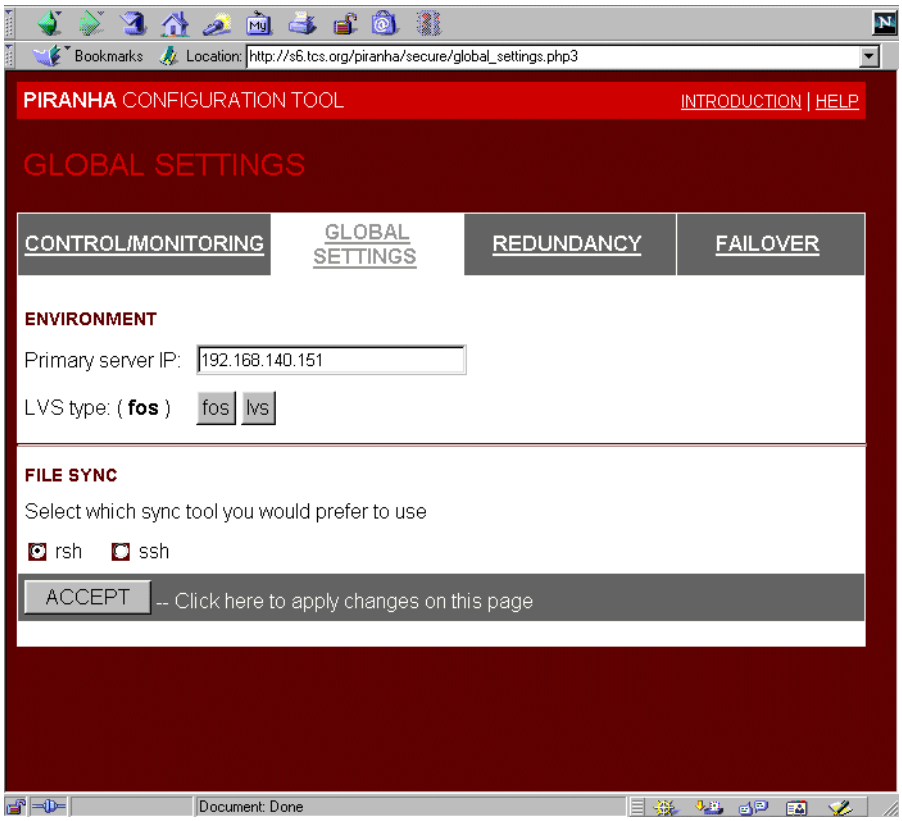


Figure 36. Global settings

Having defined the primary server we now need to define the backup server; go to the "Redundancy" section (see Figure 37) and enter the ip address of the backup server/node (we used 192.168.140.152 = s7).

You have to decide on some time-outs/intervals here too:

- Heartbeat interval: the number of seconds between every heartbeat packet

- Dead time-out: after how many seconds without receiving a heartbeat a server is assumed dead, thus making a fail-over occur
- The port number to use for the heartbeat protocol; normally the default value of 539 will be acceptable

Values depend on your expectations of fail-over time-outs. Smaller values mean smaller time-outs but more heartbeat traffic on the network; however heartbeat packets are really very small so this should not be an over-riding consideration. By having a dead time-out greater then the heartbeat interval you allow a certain amount of heartbeats to get lost before considering failing over, which would allow for temporary network traffic overload or network errors to be tolerated without initiating automatic fail-over.

Don't forget to press "Accept" at the end and your page should say "Backup: active" now, as in Figure 37.

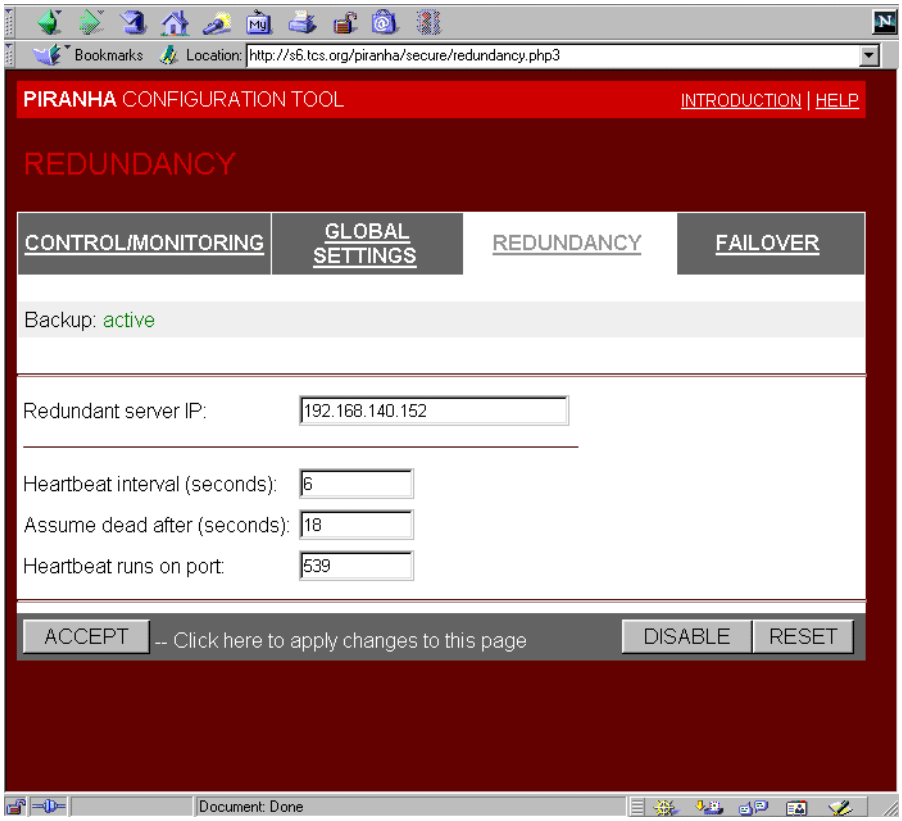


Figure 37. Backup server is now active

Finally we configure the fail-over parameters as in Figure 38, where we configure the services covered by the fail-over setup. This panel requires the completion of:

- a meaningful name for the service
- a virtual IP address for this service (we used 192.168.140.155), distinct from the actual IP addresses of the servers themselves
- the application port (e.g. 80 for http, 21 for ftp, 25 for smtp)
- the name of the aliased interface associated with the virtual IP address
- the allowed service time-out until it's considered non-functioning.

If you want, you can edit the generic monitoring scripts for watching the service.

Press "Accept" and you're done; you can add more services the same way by choosing "Add" instead of "Edit" on the Failover page.

Figure 38. Fail-over parameters

Now distribute the configuration file /etc/lvs.conf using rcp (or scp) to the other (backup) server: rcp is a remote copy protocol but scp is more secure and will probably prompt for a password before allowing the copy to take place. Not copying these configuration files (so that the primary and backup server are both configured identically) is the most likely cause of cluster failure.

```
[root@s6 /etc]# rcp lvs.cf s7:/etc/lvs.cf
s7.tcs.org: Connection refused
Trying krb4 rcp...
s7.tcs.org: Connection refused
trying normal rcp (/usr/bin/rcp)
[root@s6 /etc]#
```

Figure 39. Configuration file distribution using rsh

(The warnings about permission denied are fine. The 3rd try - using /usr/bin/rcp succeeded whereas the first two failed by trying to use kerberos, which we didn't set up)

1.5.2.2 How we used it

Now start your cluster by running:

```
/etc/rc.d/init/pulse start
```

on both machines. If everything goes fine, you should see the following messages (see `/var/log/messages`):

```
Aug 10 16:21:19 s6 pulse[31468]: STARTING PULSE AS MASTER
Aug 10 16:21:19 s6 pulse[31468]: Starting Failover Service Monitors
Aug 10 16:21:19 s6 pulse[31469]: running command  "/sbin/ifconfig" "eth0:1" "down"
Aug 10 16:21:19 s6 pulse[31468]: running command  "/usr/sbin/fos" "--monitor" "-c" "/etc/lvs.cf"
"--nofork"
Aug 10 16:21:19 s6 fos[31470]: Stopping local services (if any)
Aug 10 16:21:19 s6 fos[31470]: Shutting down local service 192.168.140.155:80
Aug 10 16:21:19 s6 fos[31470]: running command  "/etc/rc.d/init.d/httpd" "stop"
Aug 10 16:21:19 s6 pulse: pulse startup succeeded
Aug 10 16:21:19 s6 httpd: httpd shutdown failed
Aug 10 16:21:19 s6 fos[31470]: running command  "/usr/sbin/nanny" "-c" "-h" "192.168.140.152" "-V"
"192.168.140.155" "-p" "80" "-s" "GET / HTTP/1.0\r\n\r\n" "-x" "HTTP" "-R" "/etc/rc.d/init.d/httpd
start" "-D" "/etc/rc.d/init.d/httpd stop" "-t" "2"
Aug 10 16:21:19 s6 fos[31470]: Starting monitor for 192.168.140.155:80 running as pid 31470
Aug 10 16:21:19 s6 nanny[31485]: Failover service monitor for 192.168.140.155:80 started
Aug 10 16:21:19 s6 nanny[31485]: No service active & available...
Aug 10 16:21:21 s6 pulse[31468]: partner dead: activating failover services
Aug 10 16:21:21 s6 fos[31470]: Shutting down due to signal 15
Aug 10 16:21:21 s6 fos[31470]: Shutting down monitor for webservice 192.168.140.155:80 running as pid
31485
Aug 10 16:21:21 s6 nanny[31485]: terminating due to signal 15
Aug 10 16:21:21 s6 fos[31470]: will now exit to notify pulse...
Aug 10 16:21:21 s6 pulse[31468]: running command  "/usr/sbin/fos" "--active" "-c" "/etc/lvs.cf"
"--nofork"
Aug 10 16:21:21 s6 fos[31488]: Stopping local services (if any)
Aug 10 16:21:21 s6 fos[31488]: Shutting down local service 192.168.140.155:80
Aug 10 16:21:21 s6 fos[31488]: running command  "/etc/rc.d/init.d/httpd" "stop"
Aug 10 16:21:21 s6 pulse[31491]: running command  "/sbin/ifconfig" "eth0:1" "192.168.140.155" "up"
Aug 10 16:21:21 s6 pulse[31490]: running command  "/usr/sbin/send_arp" "-i" "eth0" "192.168.140.155"
"0004AC6EE826" "192.168.140.159" "ffffffffffff"
Aug 10 16:21:21 s6 httpd: httpd shutdown failed
Aug 10 16:21:21 s6 fos[31488]: Starting local service 192.168.140.155:80 ...
Aug 10 16:21:21 s6 fos[31488]: running command  "/etc/rc.d/init.d/httpd" "start"
Aug 10 16:21:22 s6 httpd: httpd startup succeeded
Aug 10 16:21:26 s6 pulse[31487]: gratuitous fos arps finished
```

Figure 40. Master starting (s6)

and the backup node should log to following to `/var/log/messages`:

```
Aug 10 16:20:13 s7 pulse[24524]: STARTING PULSE AS BACKUP
Aug 10 16:20:13 s7 pulse[24524]: Starting Failover Service Monitors
Aug 10 16:20:13 s7 pulse[24525]: running command  "/sbin/ifconfig" "eth0:1" "down"
Aug 10 16:20:13 s7 pulse[24524]: running command  "/usr/sbin/fos" "--monitor" "-c" "/etc/lvs.cf" "--nofork
Aug 10 16:20:13 s7 fos[24526]: Stopping local services (if any)
Aug 10 16:20:13 s7 fos[24526]: Shutting down local service 192.168.140.155:80
Aug 10 16:20:13 s7 fos[24526]: running command  "/etc/rc.d/init.d/httpd" "stop"
Aug 10 16:20:13 s7 pulse: pulse startup succeeded
Aug 10 16:20:13 s7 httpd: httpd shutdown failed
Aug 10 16:20:13 s7 fos[24526]: running command  "/usr/sbin/nanny" "-c" "-h" "192.168.140.151" "-V" "192.16
"-p" "80" "-s" "GET / HTTP/1.0\r\n\r\n" "-x" "HTTP" "-R" "/etc/rc.d/init.d/httpd start" "-D" "/etc/rc.d/in
stop" "-t" "2"
Aug 10 16:20:13 s7 fos[24526]: Starting monitor for 192.168.140.155:80 running as pid 24526
Aug 10 16:20:13 s7 nanny[24541]: Failover service monitor for 192.168.140.155:80 started
Aug 10 16:20:13 s7 nanny[24541]: Remote service 192.168.140.151:80 is available
```

Figure 41. Backup starting (s7)

The messages “httpd shutdown failed” are because the httpd server was not actually running in the first place.

Now the “Control/Monitoring” page should state “Daemon: running” and list the current fail-over processes as in Figure 42:

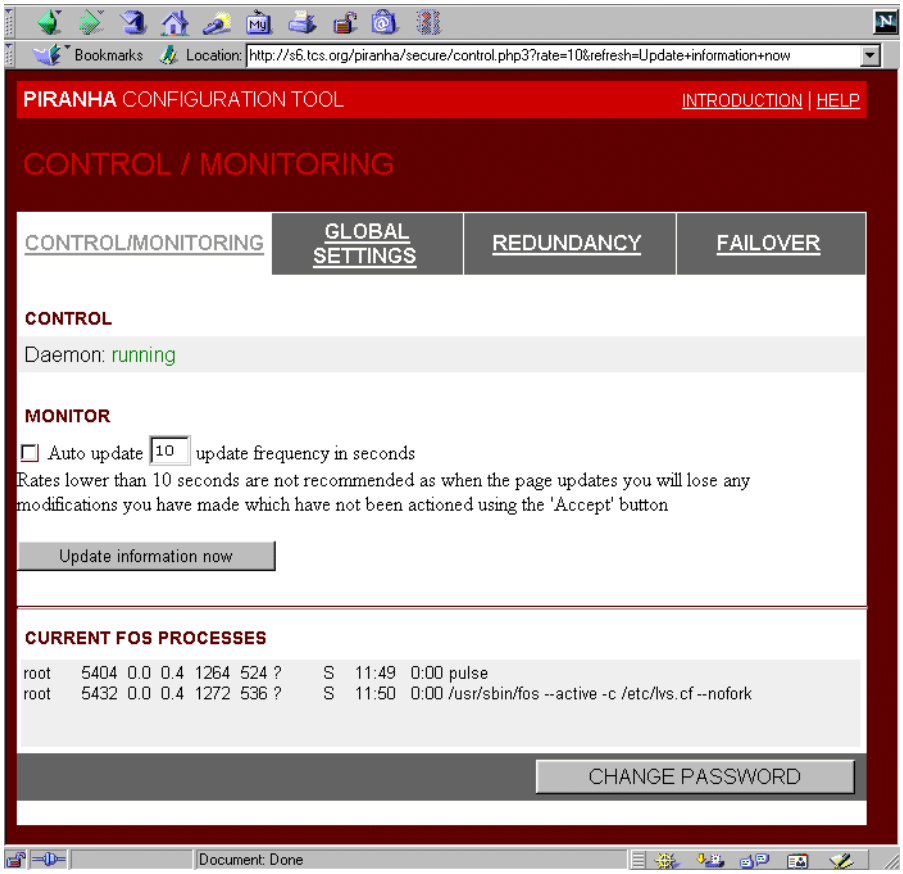


Figure 42. The fail-over cluster operating

1.5.2.3 How we showed it in operation

We adopted a simple approach: we pulled the network cable from the master server and watched the `/var/log/messages` log file (`tail -f /var/log/messages`). After the configured time-out period we saw messages to the effect that the master was no longer available and the backup took over the role of active server; the messages shown in Figure 43 are taken from “s7” which was our backup server (see Figure 32 on page 41 for our network diagram again).

```
Aug 10 16:21:28 s7 pulse[24524]: partner dead: activating failover services
Aug 10 16:21:28 s7 fos[24526]: Shutting down due to signal 15
Aug 10 16:21:28 s7 fos[24526]: Shutting down monitor for webservice 192.168.140.155:80 running as pid 2454
Aug 10 16:21:28 s7 nanny[24541]: terminating due to signal 15
Aug 10 16:21:28 s7 fos[24526]: will now exit to notify pulse...
Aug 10 16:21:28 s7 pulse[24524]: running command  "/usr/sbin/fos" "--active" "-c" "/etc/lvs.cf" "--nofork"
Aug 10 16:21:28 s7 fos[24550]: Stopping local services (if any)
Aug 10 16:21:28 s7 fos[24550]: Shutting down local service 192.168.140.155:80
Aug 10 16:21:28 s7 fos[24550]: running command  "/etc/rc.d/init.d/httpd" "stop"
Aug 10 16:21:28 s7 pulse[24553]: running command  "/sbin/ifconfig" "eth0:1" "192.168.140.155" "up"
Aug 10 16:21:28 s7 pulse[24552]: running command  "/usr/sbin/send_arp" "-i" "eth0" "192.168.140.155" "0006
"192.168.140.159" "ffffffffffff"
Aug 10 16:21:28 s7 httpd: httpd shutdown failed
Aug 10 16:21:28 s7 fos[24550]: Starting local service 192.168.140.155:80 ...
Aug 10 16:21:28 s7 fos[24550]: running command  "/etc/rc.d/init.d/httpd" "start"
Aug 10 16:21:29 s7 httpd: httpd startup succeeded
Aug 10 16:21:33 s7 pulse[24549]: gratuitous fos arps finished
```

Figure 43. Fail-Over to backup node (s7)

All services remained reachable using the virtual ip address 192.168.140.155. We replaced the cable again and observed a “fail-back” in which the primary server s6 resumes its role again.

```
Aug 10 16:24:12 s7 pulse[24524]: partner active: deactivating services
Aug 10 16:24:12 s7 fos[24550]: Shutting down due to signal 15
Aug 10 16:24:12 s7 fos[24550]: Shutting down local service 192.168.140.155:80
Aug 10 16:24:12 s7 fos[24550]: running command  "/etc/rc.d/init.d/httpd" "stop"
Aug 10 16:24:12 s7 httpd: httpd shutdown succeeded
Aug 10 16:24:12 s7 fos[24550]: will now exit to notify pulse...
Aug 10 16:24:12 s7 pulse[24635]: running command  "/sbin/ifconfig" "eth0:1" "down"
Aug 10 16:24:12 s7 pulse[24524]: running command  "/usr/sbin/fos" "--monitor" "-c" "/etc/lvs.cf" "--nofork
Aug 10 16:24:12 s7 fos[24636]: Stopping local services (if any)
Aug 10 16:24:12 s7 fos[24636]: Shutting down local service 192.168.140.155:80
Aug 10 16:24:12 s7 fos[24636]: running command  "/etc/rc.d/init.d/httpd" "stop"
Aug 10 16:24:12 s7 httpd: httpd shutdown failed
Aug 10 16:24:12 s7 fos[24636]: running command  "/usr/sbin/nanny" "-c" "-h" "192.168.140.151" "-V" "192.16
"-p" "80" "-s" "GET / HTTP/1.0\r\n\r\n" "-x" "HTTP" "-R" "/etc/rc.d/init.d/httpd start" "-D" "/etc/rc.d/in
stop" "-t" "2"
Aug 10 16:24:12 s7 fos[24636]: Starting monitor for 192.168.140.155:80 running as pid 24636
Aug 10 16:24:12 s7 nanny[24649]: Failover service monitor for 192.168.140.155:80 started
Aug 10 16:24:12 s7 nanny[24649]: Remote service 192.168.140.151:80 is available
```

Figure 44. Fall-Back to master (log from backup node = s7)

1.6 Other solutions for clustering

We also looked at some other clustering solutions, but didn’t actually implement any of them: this section should be considered as pointers to further reading elsewhere rather than a comprehensive discussion of each of

the products presented. Most of these clustering solutions are available for download in trial 30-day limited versions.

According to the paper, available from IDC, on “Clustering software and Load Balancing software submarkets”

“End-user organizations often build their computing solutions upon interconnected servers or clustered system nodes. This ongoing trend continues to produce an increase in end-user requirements for easy-to-use yet reliable, scalable, and manageable distributed solutions.

Software extending each single system's operations, coordinating its resources, and “virtualizing” the environment into a single-system view has become increasingly important. The infrastructure software that extends, coordinates, and “virtualizes” the operating system is called serverware.

This market includes file movement software, file system/volume replication software, clustering software, load balancing software, distributed file systems, Web server software, distributed naming or directory services, and virtual user interface software.

The market for clustering software has grown from $85 million in 1997 to $367.7 million in 1999.

The market for load balancing software has grown from $17.3 million in 1997 to $90.3 million in 1999."[1]

1.6.1 Polyserve’s Understudy

Understudy is a software-only server clustering solution for high availability and load balancing. It provides web, file, and e-mail server service monitoring, failure detection, failover and a load-balancing cluster utility that ensures constant server availability. It can monitor and failover IP services including: HTTP, FTP, SMTP, or TCP on Linux, NT, Solaris and FreeBSD.

SuSE Inc. has announced that it will ship an evaluation version of PolyServe as part of its SuSE 7.0 Linux distribution.

In September 2000, PolyServe announced an additional product, “LocalCluster”. LocalCluster allows automatic data replication in a cluster environment; it uses replication policies which are designed to synchronize Web content in a Web server farm. This seems to be an automated version of the design discussed in 1.3.7.1, “rsync” on page 22.

[1] High Availability Software Market Forecast and Analysis, 2000-2004, Dan Kusnetsky, Report #W22433 June 2000

For more information, see http://www.polyserve.com.

1.6.2 RSF-1 (Resilient Software Facility - 1) from StarFire Technology

Note

The following information was reprinted from the High Availablity Project web site at http://linux-ha.org/commercial.html with permission from the webmaster Alen Robertson, with Dennis Hunter's enhancements . Text is from the assoacited websites, with enhancements.

RSF-1 for Linux is another software high availability Linux server solution. RSF-1 (Resilient Server Facility) Release 1.3 provides backup and failover capabilities to pairs of servers, allowing each to take over and restart its sibling's applications following a failure.

Requirements:

Two servers, each with spare network, disk and serial interfaces.

Dual ported, resilient disk subsystem (e.g. RAID) Linux 2.x kernel with glibc (Redhat 5.1 or later recommended)

For further information on RSF-1 and to obtain their white paper, please see the web site at: http://www.high-availability.com

1.6.3 Net/Equater

Net/Equater is a load-balancing solution which refers to "logically partitioned applications". It offers one additional function over the "basic" load-balancing cluster environments we have already been discussing: it offers the ability of a client computer to connect to the "Configuration Server" prior to establishing a connection to a server. This API call would need to be incorporated in a specific client/server application program, but it means that the client can ask the Configuration Server for the address of the most appropriate server to use and then use it directly. This avoids the single point of failure and overhead of routing all inbound traffic through a "traffic manager" as shown in Figure 8 on page 15, although the single point of failure would normally be avoided by using more than one traffic manager.

This additional function requires that the *client* perform a query of the cluster before initiating a connection and therefore requires special client application code. A "proxy server" is provided which performs the same cluster query on behalf of standard application clients, which comes down to much the same

thing as the load-balancing server environments we have been discussing before.

Net/Equater was developed by Budnik & Sporner Computer Software LLC and their Web site containing more information can be found at `http://www.bcsoft.com/wlman.html`.

1.6.4 IBM's WebSphere Performance Pack

IBM® WebSphere™ Performance Pack for Multiplatforms, V3.0 is IBM's collection of software products which includes many of the products and capabilities discussed earlier in this paper. It supports Linux (Red Hat 5.2 or 6.0 & SuSE 6.0 or 6.1, all with a 2.2.x Linux kernel) and - from the perspective of this paper - it includes IBM SecureWay Network Dispatcher load-balancing software. Network Dispatcher works in exactly the same way as the Linux load-balancing solutions discussed above, and the Performance Pack also includes the IBM Web Traffic Express (a Web caching and proxy server to increase performance) and the IBM AFS Enterprise File System, all of which can be managed by the Tivoli Global Enterprise Manager or the Tivoli Enterprise Console.

For more information on the WebSphere Performance Pack, go to the Web site at `http://www.ibm.com/software/webservers/perfpack/`.

1.6.5 TurboLinux High Availability Cluster

On Oct. 19, 1999 TurboCluster Server 4.0 Beta 7 / RC3 was released. This version included major updates to the installer, clustering software, and support packages. We had access to this code but didn't have the time to install and test it during our residency.

One significant change is that the TurboCluster is no longer built on a specific Linux distribution, which means that - for example - TurboCluster could be implemented on a RedHat base distribution.

The following features are included in this release:

Users can now run both the cluster application as well as the cluster manager on the same node

TCSWAT security has been significantly improved

XFree86 has been upgraded to 3.3.5, providing enhanced support for a wider range of video cards

TCS will now automatically create the virtual interfaces on all cluster nodes, if using TCS throughout the cluster. It is no longer necessary to create lo:0 or tunl0 interfaces manually!

FTP and NFS installation now works

Certain SCSI issues have been resolved

Numerous bugs have been squashed

More information onTurboCluster Server can be found at
`http://www.turbolinux.com/products/tcs`.

1.6.6 Wizard Watchdog Service Cluster Software

Another fail-over cluster solution, designed for a high-availability environment by using hot spare servers, with shared storage (between the primary and spare) servers such as dual-attached disks, RAID, SAN and NAS solutions all being supported.

More information can be found at
`http://www.solutions6000.com/uk_framesoftware.htm`.

1.6.7 Resonate's Central Dispatch by Penguin Computing

Resonate Central Dispatch™ is a load-balancing cluster solution supported on RedHat Linux 6.0 with 2.2.12 kernel or later. It provides “class of service” capabilities which allow different resource levels to be provided to different users, applications, transactions and systems.

More information is available at
`http://www.resonate.com/products/central_dispatch/`.

1.6.8 Twincom's Network Disk Mirror

Twincom’s Dual Disc Mirror ® concept and the fault tolerant Network Disc Mirror ® software minimizes data loss and system downtime by duplicating all disk data on two separate hosts. One of the systems operates as a “hot standby” system and maintains a copy of the disk running on the active system: each write operation to the primary system’s disk is mirrored to the mirror disk running on the hot standby system. NDM is therefore designed for environments in which the data sharing requirement between active and “hot standby” systems is more than just period refresh/updates of static Web pages because NDM allows both systems to be kept in synchronization all the time.

NDM does not appear to be coupled to the fail-over Linux code directly; if the primary system should fail then messages are sent to a system administrator and the stand-by system automatically takes over ownership of the mirrored drive, and is then in a position to take over from the failed primary system. So it appears that NDM can run in conjunction with Linux fail-over solutions discussed above, allowing live data to be shared between two systems.

More information on NDM can be found at
`http://www.twincom.com/netdisc.html`.

1.6.9 Mod_Redundancy:

Mod_Redundancy is an add-on module for the Apache Web Server specifically for the purpose of creating a high-availability Apache Web Server environment. Now the single point of failure for a Web service can be avoided by setting up the extended Web Server on several physical machines and by providing a priority to each of the server instances. The server with the highest priority initially starts to work and the server with the next highest priority takes over in case of failure. Several new Apache configuration directives are provided by this new Apache module.

The minimum setup for Mod_Redundancy is two servers, one as master and one as slave; this achieves high availability but no load balancing between the servers.

The next level setup extends the high availability achieved through Mod_Redundancy with load balancing function using a combination of Mod_Proxy and Mod_Rewrite. What is done is to configure a “reverse proxy”, so called because it functions in the opposite direction than a normal proxy: the reverse proxy is not serving client requests but is in front of the servers. It dispatches the requests - for example in a random manner - to one of them. The Mod_Redundancy documentation suggests that you configure two servers to serve dynamic Web pages and the third server to serve static ones. Usually, this reverse proxy setup has a single point of failure in the proxy machine itself, and if it crashes, you can't reach your servers anymore. By protecting the reverse proxy with Mod_Redundancy, you achieve a very professional and logical solution. All the intelligence of the solution is within the Webserver and its configuration file.

The highest level setup is an extension to the medium variant in that,you double not only the reverse proxy, but also all the webservers. With this, you achieve the maximum High Availability.

Mod_Redundancy can be found at `http://www.ask-the-guru,com/`.

Part 2. Systems and network management

Chapter 2. System and Network Management with Linux

While systems management may initially involve the installation and configuration of hardware and software, inevitably it also involves the monitoring and maintenance of system hardware and software, as well as being responsive to user requests. This is true whether you are running a Linux-based, Windows-based, or even a mainframe-based network. What has changed dramatically in the Linux world, in an amazingly short period of time, is the ease and depth with which these tasks now can be performed.

In its first iteration, Linux was completely dependent on a command line interface for which an in-depth understanding of UNIX was, if not mandatory, at least highly recommended. After the IPO of Red Hat and subsequent emergence of other distributions such as SuSE, Caldera, and TurboLinux (to name but a few) as well as the coincident development of KDE and Gnome, a more graphical, and some would argue more intuitive, interface was developed that has made it possible for those more familiar with Web and Windows interfaces to effectively manage Linux networks. Tools, such as Linuxconf and WebMin, for example, have made adding users and configuring resources fairly straightforward for even the most UNIX illiterate.

In this paper, we will attempt to ascertain the viability of Linux-based systems management today in terms of ease of use and scope of function, and then briefly explore the next iteration of these tools.

2.1 What can you do today: management OF Linux

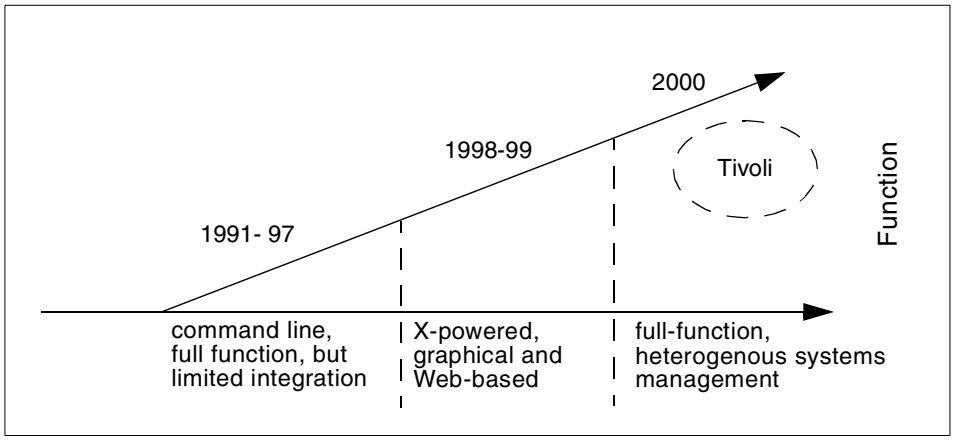


Figure 45. Trajectory of Linux management capabilities and ease of use over time

Today, following a relatively brief development period, a single network controlled by a Linux server can effectively manage itself, as well as additional Linux and Windows-based file and print servers via command line, graphical, and/or Web-based interfaces. Some of the more common functionality found within today's Linux management tools includes:

- User establishment and authentication
- Remote server maintenance
- Peripheral management
- Security
- Network status

2.1.1 Command line is still viable

To reduce overhead most servers are run in non-graphical mode, until the system is being administered, at which time graphics can be switched "on". To hard core users, however, the command line is always the most efficient interface (if not a badge of honor amongst true geeks). If you are either a geek who has momentarily forgotten a command or a geek "wannabe", simply type `whereis command_name` and Linux will show you the path...obviously have to know the command first, but...Here are some of the more useful Linux commands for systems management:

Table 5. Common Linux systems management commands

Command	Use and brief explanation
`whereis`	tells you where (in which directory) commands reside
`man`	man pages are like readme files in Windows world
`netstat`	state of TCP connections and IP routing table
`vmstat`	check virtual memory performance
`ifconfig`	state of IP interfaces
`top`	state of all processes running on the system
`mount`	attach a file system to a mount point
`unmount`	detach a file system from a mount point
`fsck`	check the state of a file system
`mkfs`	create a file system
`procinfo`	gather process information
`meminfo`	current core memory and swap usage

Command	Use and brief explanation
`loadavg`	system load average

2.1.2 Graphical interfaces: fact and fancy

Whereas the command line remains a useful and efficient medium, those of us who are non-“native” UNIX users will probably find the world of X-windows, and the KDE/Gnome interfaces built atop that world, much more convenient until we do master the intricacies of *vi* and/or *cat.* In fact, both KDE and Gnome come with a set of X-powered admin tools, but it is more common to find that the distro has imposed another layer on top of either KDE or Gnome for systems administration.

Some people take a position on which of KDE or Gnome is “better”; Gnome made much of the fact that KDE was based on the “Qt” toolkit which was not “open source” software whereas KDE made much of the fact that nothing much of substance had been released by Gnome. “Qt” has been covered by the “Q Public License” since March 1999 and Gnome is now formally supported and available on the major Linux distributions, and today the two desktop environments can be considered equivalent in many ways and equally well supported on most Linux distributions. The KDE/Gnome position today is as much one of diversity than it is one of competition or of which one is “better”.

2.1.3 Systems management overview

One of the criticisms of Linux in the past has been “It’s too hard to administer.” While this may have been true in 1995, improvements in both the range of available functions and ease of use (primarily via graphical and Web interfaces) have made this rather a non-issue. What may still be true, however, is the criticism that Linux is not quite ready for the enterprise due to a limited range of management function.[1] True, you can now manage both Windows and Linux clients from a Linux “console” to some extent, but the full range of function available in say Netfinity Director is still missing, to say nothing of a fully integrated, heterogenous management solution such as Tivoli. With that in mind, let’s consider for a moment what a “fully fledged” single system management solution *should* be able to do.

Management can be described as the ongoing attempt to configure, diagnose, and tune for performance an individual server or a network of servers. This can be as simple as adding a user to a logical network of

[1] When one says “today” in the Linux world it must be taken literally, since tomorrow things may be completely different. The rate of change within the free software/open source community is truly staggering: if a Web year is three months, then a Linux year may well be one week ;-)

computers, to pinging the host server to see why that same user could not log on, to observing data packets and measuring response times on that same network. The goal of systems management is to provide the maximum availability of system resources to the maximum number of approved users in a usable format while minimizing both the time of access to those resources and the total cost of ownership of the networked systems.

With most distributions today, you will find a “built-in” management tool, which allows you to do most or some of the following, which have been arranged in part to answer a query from an attendee at the Linux World Conference held in San Jose Ca, the week of August 14, 2000, who noted the lack of tuning, optimization and troubleshooting sessions:

- Tuning

  For the sake of this discussion regarding systems management, we assume the system has been installed and configured; thus, any additional configuration might rightly fall under the heading of tuning, or of adding something “extra” to a working system to improve availability and/or function, including the following tasks:

  - Network configuration
  - Package management
  - Services, including downloading and installing device drivers and BIOS updates

- Optimizing

  If tuning can make your system more reliable and/or available, then optimizing should make it more efficient; that is, it should provide users with better response times and increased throughput, and will include the following considerations:

  - Asset mgmt: software and hardware inventory and performance relationship
  - Performance enhancements
  - Storage management

- Troubleshooting

  No matter how well you set up your system, problems are bound to appear from time to time, necessitating the following:

  - Service mgmt: alerts/monitoring/security
  - Remote help desk
  - Routine maintenance

2.1.4 Tuning

Networks and machines must be available: Availability management...first step is to get your machine(s) “hooked”.

2.1.4.1 Network configuration

The Linux kernel supports a number of hardware drivers for various types of equipment. This section gives a short overview of the driver families available and the interface names they use.

There is a number of standard names for interfaces in Linux, which are listed here. Most drivers support more than one interface,in which case the interfaces are numbered, as in eth0 and eth1:

- lo: This is the local loopback interface. It is used for testing purposes, as well as a couple of network applications. It works like a closed circuit in that any datagram written to it will immediately be returned to the host's networking layer. There is always one loopback device present in the kernel, and there's little sense in having more.
- eth0, eth1...: These are the Ethernet card interfaces. They are used for most Ethernet cards, including many of the parallel port Ethernet cards.
- tr0, tr1...: These are the Token Ring card interfaces. They are used for most Token Ring cards, including non-IBM manufactured cards.
- sl0, sl1...: These are the SLIP interfaces. SLIP interfaces are associated with serial lines in the order in which they are allocated for SLIP.
- ppp0, ppp1...: These are the PPP interfaces. Just like SLIP interfaces, a PPP interface is associated with a serial line once it is converted to PPP mode.
- plip0, plip1...: These are the PLIP interfaces. PLIP transports IP datagrams over parallel lines. The interfaces are allocated by the PLIP driver at system boot time and are mapped onto parallel ports. In the 2.0.x kernels there is a direct relationship between the device name and the I/O port of the parallel port, but in later kernels the device names are allocated sequentially, just as for SLIP and PPP devices.
- ax0, ax1...: These are the AX.25 interfaces. AX.25 is the primary protocol used by amateur radio operators. AX.25 interfaces are allocated and mapped in a similar fashion to SLIP devices.

There are many other types of interfaces available for other network drivers, and we've listed only the most common ones. There are several ways to check and manage the interface you’re using. One is by clicking **AnotherLevel menus --> Administration --> Network configuration -->**

Interfaces, which will result in the screen shown in Figure 46. Another way to get to the same screen is by clicking **System --> Control Panel**, which will bring up an iconic representation of many system functions including network configuration.

.

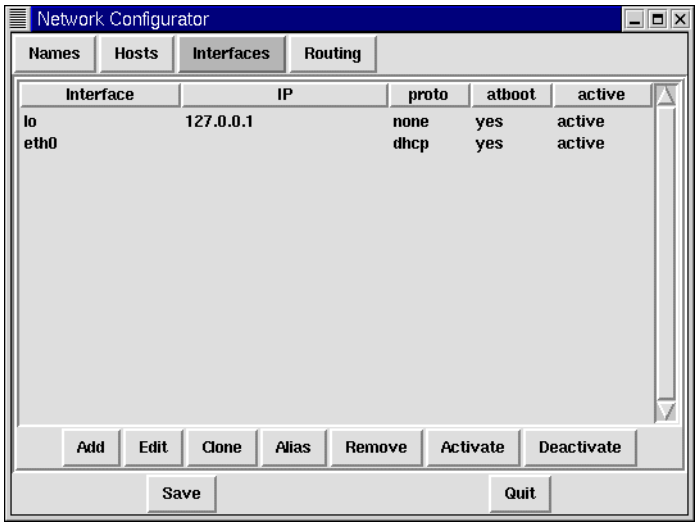


Figure 46. Gnome network administration interface

Linuxconf is an administration and configuration system used by many distributions, most notably Red Hat, used both to configure Linux systems and to activate configuration options. As shown in the example in Figure 47 on page 65, it can be used to configure networking options but it can be used for more general configuration tasks, including the addition of users, network interfaces, network routes, file systems, Apache (the Web server), Samba (for interoperability with Windows and OS/2 networks), domain name server, and both DHCP server and client. The alternative to Linuxconf is the manipulation of many different text configuration files; Linuxconf is simply an alternative method of configuration which many people will prefer to this.

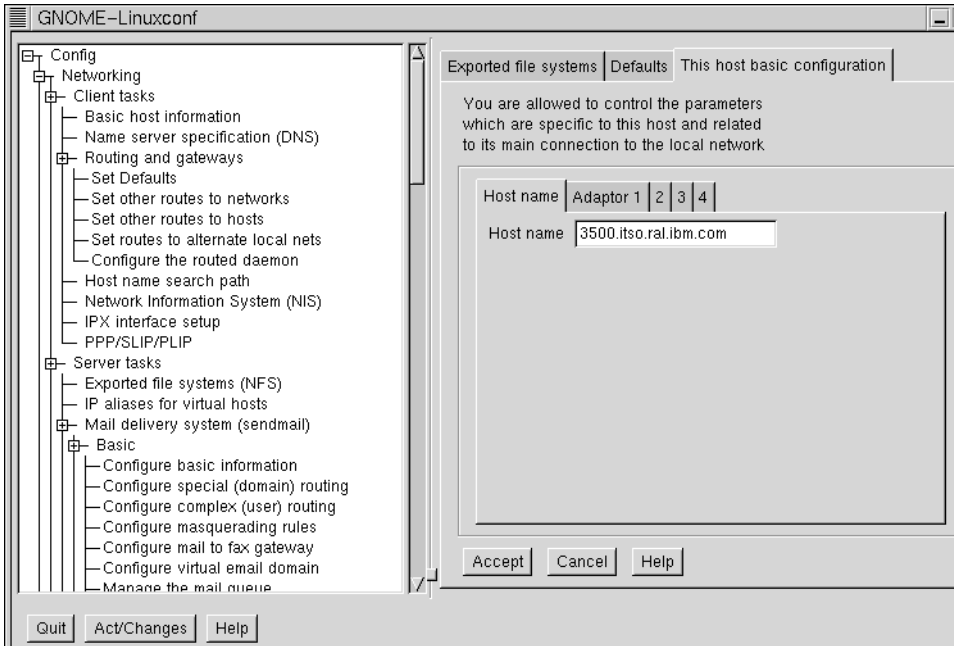


Figure 47. Linuxconf under Gnome

2.1.4.2 Package management

In Linux, much software comes in a “package” and the *de facto* industry standard of package installation and management is RPM (Red Hat Package Manager). Select **System --> GnoRPM** to access the following:

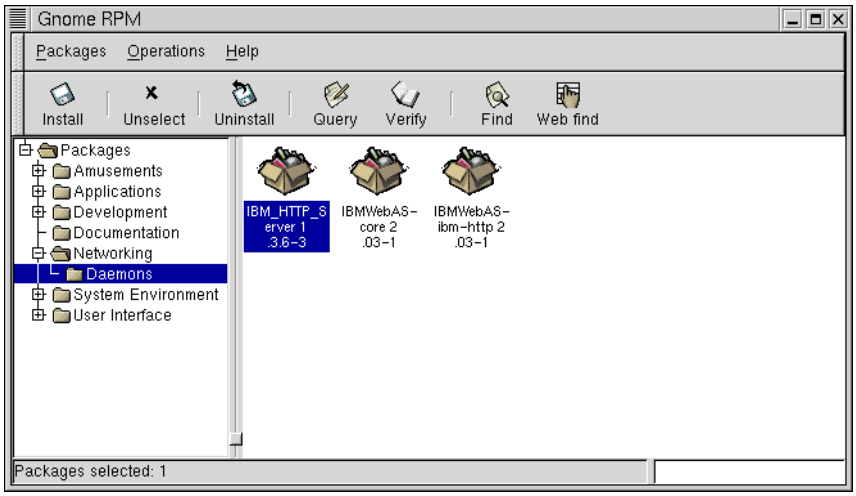


Figure 48. Gnome RPM will display installed packages

Select the package you want to work with, and then, for example click **Query**:

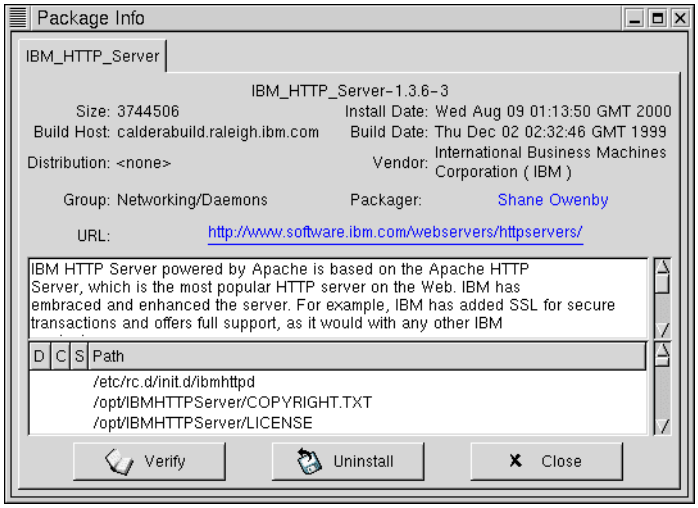


Figure 49. Select Query under Gnome RPM for information about the package

RPM allows the installation and maintenance of software packages. Not all software comes in RPM format: software which does is often easier to install and use, not least because it has been tested and used on specific Linux distributions. The alternative method of software distribution is one in which source code is shipped, and packaged in some way, and the user often has to compile and install this code more manually using the command line interface. RPM packages of source code, SRPMs, are almost always available (remember, this is “open source” software) corresponding to binary executable RPM packages: for many users installing and running the binary pre-compiled code will be quite adequate, but SRPMs allow developers and tinkerers to get hold of and modify the source code to suit their specific needs.

2.1.4.3 Services

Linux services or daemons are usually background tasks which listen for connections to be made to them before taking action. In the simplest method possible, a daemon can be started from the command line by using the `&` character as a suffix to the command, which instructs the shell to fork a subsidiary process and run the command in the background. So, for example, the command `dhcpd&` would start the dhcp daemon on the background. But many Linux distributions use schemes such as “System V” to use system scripts to start daemons: the standard Red Hat distribution defines 7 modes of execution or “runlevels”:

0. The system is halted
1. Single user mode with the root filesystem mounted read-only
2. Most services but not network services such as httpd, named etc.
3. Normal mode with all services running
4. Unused on most distributions
5. All services and graphical logon (X) enabled
6. The system will be rebooted

The scripts themselves reside in the `/etc/init.d` directory but there are separate runlevel directories `/etc/rc.d/rc3.d` (for example, for runlevel 3) which contain symbolic links to the actual daemon scripts which start with the letter “S” to denote scripts to be started when the system enters this runlevel and links starting with the letter “K” denoting scripts to be run when leaving this runlevel. So, for example, if “`S80sendmail`” exists in `/etc/rc.d/rc2.d` as a symbolic link to `/init.d/sendmail` then the sendmail daemon will be started when runlevel 2 is entered by the `/init.d/sendmail start` command.

This scheme will automatically be used by all the “normal” daemon processes supplied with the distribution but it should be understood and used by anyone wanted to add new daemon processes or modify the automatic starting/stopping of existing ones.

`/etc/inittab` will contain the default system runlevel setting, which will usually be 3 or 5 (it had better not be 0 or 6!). The `telinit` command can be used to change to a different runlevel, and this command would most probably be used to switch the system to single-user mode for specific maintenance operations which require no other users to be active on the system. The `runlevel` command will display the current and, if applicable, prior runlevel.

2.1.5 Optimizing

After you have tuned your network for availability, you can start tweaking it for improved performance. One of the first steps is to gather all of the necessary information regarding hardware and software. In other words, you’ll need to know what you’ve got before you can “improve” upon it. Tools such as Lothar and Harddrake (from Mandrake) can assist in the identification and installation of Linux hardware.

2.1.5.1 Performance

Performance management asks the question “How is My System Doing?” Some of the things it does in a network is to look at data package traffic

patterns for bottlenecks, file transfer times, update time-outs. It sets known good statistics for failure comparisons.

The Gnome System Monitor shown below is the Gnome version of the `top` command which shows the status and activity of all the processes in the system, the amount of memory being used and the status of all the monted filesystems:

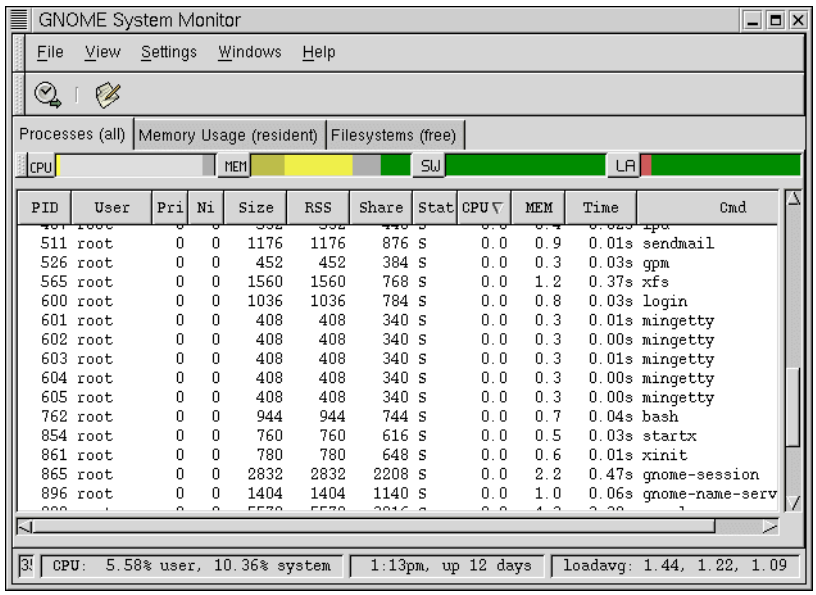


Figure 50. Gnome System Monitor

2.1.6 Troubleshooting

2.1.6.1 Monitoring

Some simple monitoring commands are used at the command line: `netstat -r` to show the IP routing table, `ifconfig tr0` to show the status of the tr0 network interface, `ps -A` to show all the processes, `tail /var/log/messages` to show the last 20 lines of the system messages log, `ipchains -L` to show the status of the firewall/IP forwarding rules, to give a few examples. Many of these commands will be unfamilar to current Windows system administrators but most of them will be familiar to current Unix system administrators.

Other monitoring tools are available though KDE and Gnome menus; many of them (as well as the command line options) require root authority although some will work without this.

2.1.6.2 Maintenance

Some of the same basic commands as in the previous section can be used to maintain and alter the system configuration: `ifconfig tr0 down` will bring down the tr0 interface, `killall dhcpd` will kill the dhcpd daemon, `route add` will add a static route to the IP routing table.

2.1.6.3 Security

System security starts with good system administration. This includes checking the ownership and permissions of all vital files and directories and monitoring use of privileged accounts. The COPS program, for instance, will check your file system and common configuration files for unusual permissions or other anomalies. It is also wise to use a password suite that enforces certain rules on the users' passwords that make them hard to guess. The shadow password suite, for instance, requires a password to have at least five letters and to contain both upper- and lowercase numbers, as well as non-alphabetic characters.

When making a service accessible to the network, make sure to give it “least privilege”; don't permit it to do things that aren't required for it to work as designed. For example, you should make programs set uid to root or some other privileged account only when necessary. Also, if you want to use a service for only a very limited application, don't hesitate to configure it as restrictively as your special application allows. For instance, if you want to allow diskless hosts to boot from your machine, you must provide Trivial File Transfer Protocol? (TFTP) so that they can download basic configuration files from the /boot directory. However, when used unrestrictively, TFTP allows users anywhere in the world to download any world-readable file from your system. If this is not what you want, restrict TFTP service to the /boot directory.

Another source of concern should be programs that enable login or command execution with limited authentication. The rlogin, rsh, and rexec commands are all very useful, but offer very limited authentication of the calling party. Authentication is based on trust of the calling host name obtained from a name server (we'll talk about these later), which can be faked. Today it should be standard practice to disable the r commands completely and replace them with the ssh suite of tools. The ssh tools use a much more reliable authentication method and provide other services, such as encryption and compression, as well.

You can never rule out the possibility that your precautions might fail, regardless of how careful you have been. You should therefore make sure you detect intruders early. Checking the system log files is a good starting point,

but the intruder is probably clever enough to anticipate this action and will delete any obvious traces he or she left. However, there are tools like tripwire, written by Gene Kim and Gene Spafford, that allow you to check vital system files to see if their contents or permissions have been changed. tripwire computes various strong checksums over these files and stores them in a database. During subsequent runs, the checksums are recomputed and compared to the stored ones to detect any modifications.

Most current Linux kernels include support for the addition of firewall rules: meaning that Linux can act as a firewall on behalf of other machines as well as protecting itself. In this age of “broadband” “always-on” networking it is of increasing importance that connections to the Internet be regarded as unsafe and that all machines attached to the public Internet should be protected from malicious attack by the use of firewalls. The following screen capture shows an example of the rule sets which can be implemented on a standard Linux distribution: essentially this rule set allows the establishment of new connections from the Internet only for known services (such as the Web server service) but rejects and logs unknown and unathorised connections. Examination of system logs at frequent intervals should also be made to police this ruleset.

Note also that many of today’s Linux distributions no longer install and enable many applications by default - ftp and telnet may need to be re-configured from the default installation configuration to enable Internet access, if this is actually what is required. This can be a source of annoyance and frustration for new Linux users but probably protects the majority from unwittingly leaving too many doors open.

```
ipchains -L
Chain input (policy REJECT):
target     prot opt     source               destination          ports
ACCEPT     all  ------  10.0.0.0/8           anywhere             n/a
ACCEPT     all  ------  10.0.0.0/8           anywhere             n/a
REJECT     all  ----l-  10.0.0.0/8           anywhere             n/a
ACCEPT     tcp  ------  0.0.0.0/24           255.255.255.0/24     bootpc ->   bootps
ACCEPT     udp  ------  0.0.0.0/24           255.255.255.0/24     bootpc ->   bootps
ACCEPT     tcp  ------  0.0.0.0/24           255.255.255.0/24     bootpc ->   bootps
ACCEPT     udp  ------  0.0.0.0/24           255.255.255.0/24     bootpc ->   bootps
ACCEPT     all  ---fl-  anywhere             atl-tgn-ydc-vty4.as.wcom.net  n/a
ACCEPT     icmp ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  echo-reply
DENY       icmp ------  Serial1-8.GW8.ATL1.ALTER.NET atl-tgn-ydc-vty4.as.wcom.net  any
ACCEPT     icmp ----l-  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   any
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   dom
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   dom
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  domain ->
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  domain ->
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   www
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   www
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   ftp
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  ftp-data ->
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   tel
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   ssh
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   smt
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   pop
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   pop
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   pop
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   pop
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   ima
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   ima
ACCEPT     tcp  ----l-  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   aut
ACCEPT     tcp  ------  9.0.0.0/8            atl-tgn-ydc-vty4.as.wcom.net  any ->   tim
ACCEPT     tcp  ------  32.0.0.0/8           atl-tgn-ydc-vty4.as.wcom.net  any ->   tim
ACCEPT     all  ------  anywhere             anywhere             n/a
REJECT     tcp  -y--l-  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   any
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   102
ACCEPT     tcp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  ntp ->   ntp
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  ntp ->   ntp
ACCEPT     udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   610
DENY       udp  ------  anywhere             atl-tgn-ydc-vty4.as.wcom.net  any ->   697
REJECT     all  ----l-  anywhere             anywhere             n/a
Chain forward (policy DENY):
target     prot opt     source               destination          ports
ACCEPT     all  ------  10.0.0.0/8           10.0.0.0/8           n/a
ACCEPT     all  ------  10.0.0.0/8           10.0.0.0/8           n/a
MASQ       all  ------  10.0.0.0/8           anywhere             n/a
REJECT     all  ----l-  anywhere             anywhere             n/a
Chain output (policy REJECT):
target     prot opt     source               destination          ports
ACCEPT     all  ------  anywhere             10.0.0.0/8           n/a
ACCEPT     all  ------  anywhere             10.0.0.0/8           n/a
REJECT     all  ----l-  anywhere             10.0.0.0/8           n/a
REJECT     all  ----l-  10.0.0.0/8           anywhere             n/a
ACCEPT     all  ------  atl-tgn-ydc-vty4.as.wcom.net anywhere             n/a
ACCEPT     all  ------  anywhere             anywhere             n/a
REJECT     all  ----l-  anywhere             anywhere             n/a
```

2.1.6.4 Support

You can now count on IBM for the most comprehensive software, services and technical support in the Linux environment. Depending on customer need, IBM offers 24-hour a day, 7-days a week Internet and voice support, ranging from answering usage questions to identifying problems.

IBM Global Services also provides consulting, planning and implementation services for Linux. IBM consultants can help you evaluate whether Linux is appropriate for your particular environment. If so, they will help you implement and optimize your Linux solutions. We are now offering the same kind of operating system support for Linux as we do for AIX, OS/2, OS/390, OS/400 and NT.

Because Linux is still an evolving environment, you want a service partner to help you properly configure and implement, as well as enhance, your Linux solutions. Now, customers can turn to IBM Global Services as a one-stop shop for Linux support. For information on these or additional service and support offerings please visit our Web site or call 1-888-426-4343.

2.1.6.5 Support Line for the Linux Operating System

IBM Operational Support Services -- Support Line for the Linux Operating System offers:

- 7x24 Enterprise Level remote support for your Linux OS environment.
- Fast and accurate problem resolution.
- A way to supplement your internal staff with IBM's skilled services specialists.
- Defect support for supported distributions of the Linux OS and Linux applications.
- Electronic support and problem submission that saves you time and allows you to track your open support issues.

2.1.6.6 IBM's premier remote technical support for Linux

IBM has long set the standard for high quality, responsive and professional software support. With IBM Operational Support Services -- Support Line for the Linux Operating System, you can now leverage IBM's Enterprise support capabilities into your Linux operating environment. IBM supports major distributions of the Linux OS as well as all IBM and some non-IBM applications that operate in a Linux environment.

We help answer your how-to questions, help you define problems and determine their source. Additionally, by leveraging our partnerships with the

key distributors of the Linux operating system, IBM is able to provide defect-level support for the Linux OS. Remote assistance is available through toll-free telephone access and electronic access.

For all eligible distributions of the Linux operating system, IBM can help you with:

- usage and installation questions
- product compatibility and interoperability questions
- interpretation of product documentation
- a diagnostic information review to help isolate the cause of a problem
- configuration samples
- IBM and multi vendor database searches
- planning information for software fixes
- defect support

Electronic Support allows you to submit and get answers to your problems electronically.

2.1.6.7 What OS Distributions are Supported?

IBM provides how-to and defect support for the four major distributions of the Linux OS: Red Hat Linux, Caldera OpenLinux, TurboLinux, and SuSE Linux.

2.1.6.8 Standard Coverage

Basic prime shift support includes coverage during normal business hours, Monday through Friday, excluding national holidays. Only designated callers can request support. With our standard coverage option, you may select two named callers who can submit unlimited service requests for products covered under your agreement.

2.1.6.9 Coverage Options

We have several options that you may choose to enhance your Linux support coverage. Full Shift Coverage provides you 24-hours a day, 7-days a week coverage. In addition, the number of named callers increases to six per covered platform. Additional Named Callers can be purchased if you require more than the allotted number.

2.1.6.10 Other services to help ensure your business success

Other IBM technical support services that can be of value to you:

IBM Operational support Services -- Account Advocate is an option that provides a single support interface for remote support. With this service, you are assigned your own Account Advocate team that becomes thoroughly familiar with your business and systems environment. This team serves as your single interface for software support at IBM.

IBM Operational Support Services -- Advanced Support is the highest level of remote support provided by IBM. This service is tailored to meet your unique needs for continuous, business critical system operation.

IBM Operational Support Services — Consult Line lets you schedule telephone consultations with our technical experts to resolve in-depth issues important to your business needs.

2.1.6.11 Total solutions through comprehensive services

IBM provides the most extensive array of IT services in the industry. We offer product support services through both prepackaged and customized solutions, supporting both IBM and multi vendor systems

2.1.7 Working from home: remote capabilities

Talk about geographically separate systems, be they on different floors of the same building, different states, or different continents. Then discuss what is possible...does distance make a difference? Bandwidth and latency limitations of current tools...

Is there a Linux “Wake on LAN” solution?

For the sake of this discussion, assume that remote installation and configuration are management components, whereas remote administration is more concerned with end users than machines per se.??????

Remote management/installation/configuration is sort of a separate issue...that is which of these mgmt tasks can be done remotely is different than which can be done at all...(to enable remote access may require some diddling with the /etc/services.../etc/inetd.conf...and then /etc/hosts.allow and hosts.deny files...this will allow such functions as ftp, telnet etc.) and “everything” can be done using telnet ;-)

2.1.7.1 Remote installation

With remote installation the system administrator may install, remove, and upgrade both applications and operating system components without having to visit each physical system on the network.

2.1.7.2 Remote management

Remote management is the interplay of local device management agents and network management software to provide requested information on system status gained from management information bases (MIBs). It can also include the ability to transparently control a physical system and or logical network on an ongoing basis as if standing in front of the systems's main local interface.

2.1.8 Migration

NT2Linux is a migration kit that picks up various settings in NT and installs them on a new Linux box. There are two components: one is an NT program that allows you to select the topics to export, and the other is a Linuxconf module, adapting the information to Linux.

You can export:

- Network configuration
- User accounts
- Groups
- Disk shares
- Printers
- Data access control list

2.1.9 Summary

Having outlined some of the major aspects of systems management, let's now take a brief look at how the tools within the "major" distributions (in this case, "major" means those formally supported by IBM) have addressed these functions.

Table 6. Major distros and their sys mgmt tools

	Red Hat	**Caldera**	**SuSE**	**TurboLinux**
Tool	linuxconf	COAS	YaST	Enighten
Interface				
Window mgr	X	KDE	KDE	Gnome
Tuning				
Configuration				
Packages	RPM			
Services				
Maintenance				

	Red Hat	Caldera	SuSE	TurboLinux
Optimizing				
Assets	weak/limited			
Perfprmance				
Troubleshooting				
Monitoring				
Security				
Support	full from IBM	full from IBM	full from IBM	full from IBM
Remote capabilities				
Management				
Installation				

2.2 Where do you want to go tomorrow: management ON Linux

Currently Linux-based solutions are fairly limited to single system, more or less homogenous networks, but all of that is about to change, at which point Linux might truly be ready for the enterprise.

Let's first take a look at what an enterprise systems management solution will have to live up to

2.2.1 Netfinity Director shows the way

Currently Netfinity Director supports a vast array of functions for systems management, some of which are unique to Netfinity's hardware architecture, but which, nevertheless, have become the benchmark by which other's must be judged. Before discussing those functions which are available today for the Linux platform, we will take a brief look at Netfinity Director's functionality and assume to some extent that all of these functions will be available to Linux systems in the near future.

IBM's Netfinity line of Intel-based PC servers provide a unified platform designed from the bottom up to provide you with a system of servers that have "lessons learned" from IBM's leadership roll in mainframe X architecture. IBM almost invented the concept of systems management for

networked systems and now brings that background to their other innovation the PC.

Netfinity servers have systems management hardware built into the system design enabling the systems information to be gathered in a format that is ready to be leveraged by Linux management systems.

Netfinity Director: heart of Universal Manageability (UM)

- supports up to 1500 licensed clients
- multiple systems management
- Java-based Explorer-like user interface (drag and drop)
- rules based grouping (groups can be dynamic, static, or task-based)
- integrated database
- Supports Common Information Model (CIM), Desktop Management Interface (DMI), and Simple Network Management Protocol (SNMP)
- Three components: server, console, clients
- Supports Life Cycle Tools (UM Server Extensions...plug-ins)
- client/server configuration (as opposed to peer-to-peer) with dedicated server

Netfinity Director tasks :

- Inventory
- Monitors
- Events
- Software distribution
- Process Manager
- Remote control
- Cluster Manager
- File transfer
- DMI browser
- CIM browser
- Upward integration (point-to-point management only
- Alerts

2.2.2 Tivoli endpoint

As the industry's leading supplier of *open, cross-platform management solutions*, Tivoli has a large number of customers with increasingly heterogeneous IT environments, to which Linux is rapidly being added.

Currently, Linux is supported as an endpoint in the Tivoli Enterprise Architecture, where an endpoint is simply a system running the Tivoli Management Agent software

The first Tivoli product lines to support Linux will be Tivoli Enterprise products and the Tivoli Management Suites. In addition to the Tivoli Management Framework, applications supporting Linux will include Tivoli Software Distribution, Tivoli Inventory, Tivoli Distributed Monitoring, Tivoli User Administration, Tivoli Security Management and Tivoli Enterprise Console adapters. These products will initially support the Red Hat distribution, in order to fulfill the requests of Tivoli customers. Tivoli will support other versions of Linux and other applications in the future as demand warrants.

2.3 Other tools and solutions

Some of the following tools combine one or more of the management functions discussed earlier into one convenient package. Some tools can be used together to provide the solution that you require. It is beyond the scope of this document to explain in detail how you can use each tool or all tools together, although IBM can and will help you with that need. Instead what we want to do is expose your mind to a new way of thinking about management of your systems and the endless possibilities that with stable, Unix like tools, you, and IBM Netfinity PC servers can achieve.

For the latest on what you can do with Netfinity and Linux contact your local IBM reseller or go to the IBM Linux web site at `http://ibm.com/linux`

For up and running support, installing Linux on Netfinity PC Servers, visit the Support@IBM site at http://www.pc.ibm.com/support and select Servers->Family (i.e. Netfinity 7000-M10)->OS Installation (i.e. Linux) or call the IBM PC Help Center at 1-800-772-2227

If your Netfinity PC Server ever fails to operate, you have live support available, 24/7, at the IBM PC Help Center.

Linux “bugs” don’t live long. The light of constant peer review on the Web means that only the best survive. If there is a “bug” in the program either your programmers can find and fix it themselves because you have the source code, or you can contact the writer of the program and ask him to address it (offer lots of pizza and much kudos, although they will trade in dollars).

The collection of Linux Management tools and “solutions” referenced in this document are but some of those available, new ones are being created every day. Old ones are always being perfected.

The following are some of the more well known system management solutions in the Linux world:

- ACUA - Access Control and User Administration tool
- apc_up - daemon that supports APC UPS's under Linux
- Avatar - job processing application suite, with a highly sophisticated built-in scheduler
- cfengine - very powerful and easy-to-use network con fig and admin tool
- Chklogs - PERL script to help maintain system logs
- Clustor - a task crunching utility
- ComBase - Web-based SysAdmin Tools add/modify/delete user accounts
- DevAlloc - emulates device allocation mechanism Sun provides in Solaris 2.x
- Evolution Scheduler - scheduler based on genetic algorithms and evolutionary programming
- FakeBO - fakes BackOffice server and logs every attempt to a log file or stdout
- Figurine - configuration system for UNIX
- Generic NQS - batch processing system
- Gr_monitor - draws 3D color bar graphs of all processes on a Linux system
- grub - Grand Unified Bootloader boots multiple OS's on PC's
- IRQTUNE - a Linux x86 IRQ priority optimizer
- jaZip - program for maintaining Zip and/or Jaz drive(s) and disks
- LanSafe III - UPS Power Management Software UPS app provides automatic shutdown
- Linux-PAM - a flexible mechanism for authenticating users
- Loadmeter - an enhanced X11 system resource meter
- lsof - list open files for running Unix processes
- Mfsm - a Motif utility that monitors free space and user file system quotas
- morepkgtools - supplemental scripts for Slackware's package tools
- Nessus - a SATAN-like sys admin tool for security auditing
- NIST - used to synchronize the system's time with a known timeserver
- pam - Pluggable Authentication Modules for Linux
- phalanx - security daemon will only allow 'cleared' users from 'cleared' domains

- ProcMeter - simple X Windows based performance meter, reading the info from /proc
- Qps - visual process manager, an X11 version of "top" or “ps”
- qtime - time tracking software written in Tcl using Qddb and its Fx library
- Queue - load-balancing system that lets users control their remote jobs
- radius - Context Remote Authentication Dial In User Services accounting log analysis package
- Sawmill - a powerful, hierarchical log analysis tool
- SUDO - superuser do utility to allow restricted root access
- sysdaemon - a system monitor written in generic perl5
- System Commander - manage multiple operating systems on one PC
- WOTS - log file monitoring program with extensible and configurable actions
- XLoadTime - /proc based load meter that also shows current time
- XUser - create, modify and delete user-info from an interactive X shell
- Tivoli - Management tool from IBM to be ported to Linux Networking Tools

2.4 Other Network Management Tools

- Angel Network Monitor tool to monitor the status of services on your network
- Big Brother monitor multiple Unix systems, network connectivity, disk space, processes, etc
- bootpc boot protocol client used to grab ip number and set up DNS name servers
- CFEngine script based system for managing a heterogeneous network of UNIX systems
- Clobberd allows ISP's or Linux/Unix operator to regulate their users
- CVS client/server configuration management system
- Gxsnmp full featured snmp network management application for GNOME
- Hifs handy information for sysadmins, a tool for system monitoring
- IETF Mobile IPv4 a mobile IP implementation for Linux
- IPAcct program/kernel extension to do per user ip accounting
- IP NAT IP network address translation under Linux

- ipsnat another Linux IP Network Address Translation implementation
- ippfvs scalable and highly available virtual server built on cluster of real servers
- Mon general-purpose network resource monitoring system
- MPAS interface allows Win/DOS users access to files stored on network
- MRTG multi-routing traffic grapher monitors network traffic load
- MTR (Matt's Trace Route) combines the functionality of ping and traceroute into one app
- Multi Router Traffic Grapher tool to monitor traffic load on network-links creates HTML pages
- Netatalk for Linux let your Unix box look like an Appletalk server on a LAN
- net-tools collections of programs to control Linux' networking
- Network Management Software a catalog of network management tools for Linux
- ntop tool that shows network usage, similar to top
- REGULUS management system for Internet Service Providers (ISPs)
- Remote Reboot use a network connection to reboot a machine remotely
- Ruplist shows all sites on that LAN with their uptimes and their ranking
- Scotty Tcl extensions for network management applications
- SecureNet PRO a full featured network administration and diagnostic tool
- snmpsniff a Linux SNMP sniffer (a tool for network management debugging)
- Sniffit a Linux packet sniffer
- serialmon serial line monitor program with statistics
- SPONG a simple network/host system monitoring package
- Tkined a network drawing editor running on top of the Tcl/Tk toolkit
- traf a network traffic monitor program suitable for ethernets
- TZO dynamic DNS solution for Linux users, creates domain names for PC's
- ZanNet combo Win95 client/Unix server provides a Win95 network drive to access server files

2.4.1 Relevant Web sites

- http://www.redhat.com/support/manuals/RHL-6.2-Manual/ref-guide/

  See especially the Systems Administration chapter.

- http://www.linuxdoc.org/LDP/nag2/index.html

  This is an online Linux network administrator's guide and part of the LDP (Linux Documentation Project)...quite good

Part 3. Interoperability

Chapter 3. Interoperability of Linux solutions

The purpose of this paper is to explore how to add Linux servers to existing configurations of clients and servers. These Linux servers may replace existing servers on other operating systems or they may provide additional functions complementary to existing servers.

We will show some examples of how we took a new Linux server running on Netfinity hardware and added it to an existing network, but the purpose of this paper is not to be a “cookbook”: the main purpose of this paper is to discuss the roles which Linux servers can play in mixed client/server environments, to explore some of the possibilities and to point the reader in the right direction to explore these possibilities further.

This paper will provide you with an overview of today’s interoperability issues and how they can be solved using Linux. We will not cover the pure Linux client/server environment, what we are talking about are the possibilities and issues in a mixed environment. Linux allows many possibilities for adding a Linux server to an existing environment in which no Linux devices exist.

3.1 Where are we now?

Looking inside today’s companies we normally find a heterogeneous client/server environment: a mix of both hardware and software. There may be large UNIX or S/390 back-end database servers surrounded by application servers which may well be running on UNIX platforms. There are probably many workgroup file/print servers deployed, running Windows NT or Windows 2000. Windows 95/98/NT dominates the client/workstation platforms. In addition, or instead, there may be AS/400 mid-range servers as well. Putting this into a picture leads to the next figure:

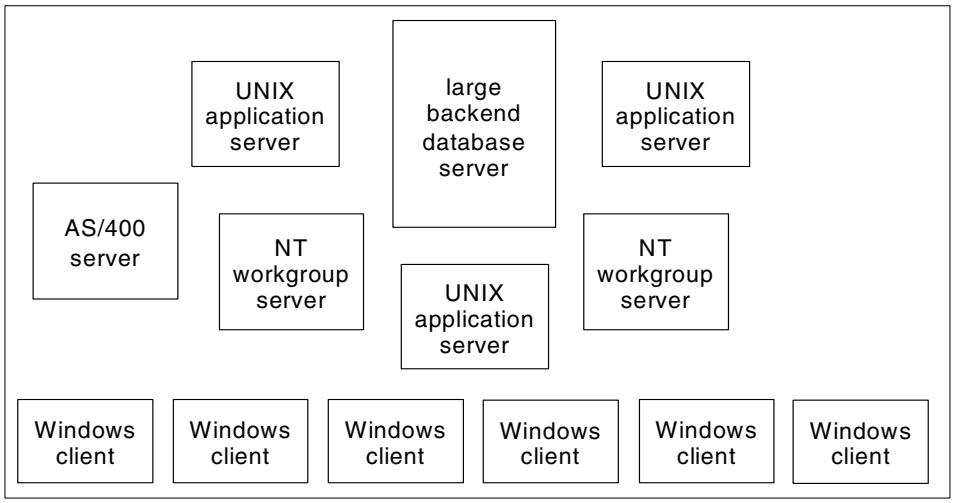


Figure 51. Typical current environment

3.2 Where to integrate Linux into your current environment

Where can Linux fit into the above picture? Right now, Linux doesn't really have all the features required to serve as a large database server for a large company because it doesn't - yet - have the reliability and scalability required for such large database servers. But you can run DB/2 on Linux, providing one of the best databases available for small or medium business environments or as part of a large company's smaller database environments. The attraction of Linux here is low software and hardware cost coupled with a stable operating environment.

The next layer is the application server layer. This is one of the best places to put Linux today. Linux delivers a rock-solid, fully-featured UNIX-style operating system which can be used to its full benefit when coupled with rock-solid, well-supported Netfinity servers. There are many different kinds of application servers available for Linux, all the way from free solutions such as Zope[1]to fully-supported commercial products such as Websphere from IBM. Even the SAP/R3 application server is available in a Linux version today. Furthermore you can cluster these Linux servers for load balancing and high availability.

Most of today's workgroup servers are running Windows NT and provide file sharing and printer sharing abilities. This is what Samba is designed for. Linux running Samba acts as a full-featured Windows file and print server.

[1] Zope in an Open Source Web application server, see `http://www.zope.org` for more details

Since Linux already includes many other server services, you can use Linux to enhance your workgroup server to act as a file/print server, provide content via Web for your department and to act as a mail server, fax server and so on. For some businesses the low software cost coupled with high Linux reliability may be enough of a reason to do this - and, as already noted, IBM is happy to support customers running Linux on Netfinity servers. Even if software cost is not a primary issue, the reliability and stability of Linux coupled with the ability to provide many complementary server functions on a single reliable hardware platform make Linux an attractive option today.

What's finally left are the Windows clients. While this is not the focus of this paper, it's nevertheless a interesting aspect to watch. WINE (which allows you to run Windows applications inside Linux without installing Windows) gets better and better from release to release. Major office suites are available. More and more companies use Web-based applications, which only require a Web browser such as Netscape on the client workstation, and with desktop environments such as KDE and GNOME the end user has a working environment at least comparable to the Windows desktop.

3.3 Why Linux?

While Linux is a heavily discussed and very "trendy" topic today, we shouldn't forget that there are serious reasons behind this trend towards "Linux everywhere".

- Linux can run on different hardware platforms in a way that no other operating system can. Linux is available for hardware platforms including IA-32 (existing Intel), IA-64 (future Intel), PowerPC, S/390, Alpha (formerly DEC, now Compaq). You can port and run your applications on different hardware platforms just as long as they support Linux, and the reason for wanting to run applications on different platforms include scalability and the price/value factor of Linux solutions. Linux allows PC (IA-32) applications to be moved to PowerPC or S/390 platforms in a way that was not possible before.
- Huge amounts of know-how of many experts went into the development of Linux, leading to one of the stablest, full-featured, and fastest operating systems available. The design point of many of the Linux developers is one of speed, stability and efficiency. Linux is not designed to be "user friendly" in the sense that it is not designed for home users or people who do not understand computers; having said that, much effort has recently been expended in making Linux easier to configure and use.
- Linux is an Open-Source operating system, allowing you to take part of its development and modify it to fit your needs. It's easy to find out how

things are implemented and how they work: Linux is not simply “object code”. Furthermore, you can fix bugs by yourself, if you are experienced enough. You don’t become dependent on one company owning the operating system, the applications the know-how and the support structure.

- Linux is a well-supported platform and the support gets better every day. There are free support options such as news groups, mailing lists, HOWTOs, and FAQs, as well as commercial support offerings for large scale projects or more important solutions. “Free support” may not seem appropriate for mission-critical applications and operating systems, but Linux today has achieved “critical mass” in which enough people are operating and supporting Linux to make this the best option for most purposes.

Depending on your needs there will be even more reasons for integrating Linux into your business. So let’s have a look at the Linux integration solutions.

3.4 Solutions for integrating Linux

As Linux becomes more pervasive it will become even more important to integrate it with existing platforms. Naturally, one would expect Linux to interoperate fairly seamlessly with an RS/6000 running AIX or a similar UNIX system, but what about the AS/400, S/390, or even Windows? In the following sections we will provide an overview of the more popular methods available for achieving just such an integrated environment.

3.4.1 Samba does Windows

If you already happen to be the administrator of an environment where the engineers are using UNIX boxes, some of your programmers are working with NT, and the managers are demanding Windows laptops, then you probably already know about Samba. If you happen to be responsible for such an environment and haven’t deployed Samba yet, then you may want to consider the following:

1. Samba was first published by Australian, Andrew Tridgell in 1992. It was his solution to the problem of sharing disk space on a UNIX server with his PC. This in turn accurately defines what Samba is: a piece of code that allows UNIX machines to share resources with other systems, primarily (in this day and age) Windows machines with TCP/IP loaded[2]. In other words, Samba running on your UNIX/Linux machine allows Windows clients to access shared resources that physically reside on a UNIX or Linux-based

[2] Samba requires that the clients use NetBIOS over TCP/IP (RFC 1001 and RFC 1002) and not NetBIOS over NETBEUI.

server. Samba allows a Linux machine to be a file/print server for Windows clients. Of course it works the other way as well, allowing Linux or UNIX clients to access resources on say a Windows NT server.

From the point of view of the clients, Samba allows the UNIX server to appear just as a PC server.

3.4.1.1 How does Samba work?

Samba allows Linux and Windows machines to share resources by utilizing a networking protocol that is common to both: SMB (Server Message Block), which in turn is built upon the NetBIOS API.

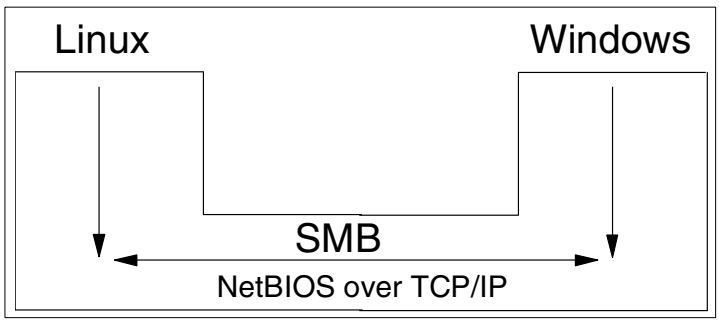


Figure 52. The common ground for Linux and Windows machines is SMB, which is built on top of the NetBIOS API

As noted above, Samba is intimately related to NetBIOS and SMB. SMB was originally defined by IBM and Microsoft in the middle 1980s and has since been developed further by Microsoft; in its most recent incarnation it is now known as CIFS, or the Common Internet File System. The following table is a representation of the OSI and TCP/IP layers, and shows where SMB resides. Both OS/2 and Windows *can* use SMB over the NetBEUI transport (which uses connection-oriented LLC2 protocols), but this transport method is not supported by Linux's SMB implementation. As long as this is understood, it

should not cause a problem, because both OS/2 and Windows can be configured to use SMB over TCP/IP.

Table 7. SMB over multiple protocols

OSI			**TCP/IP**
Application	**SMB**		Application
Presentation			
Session	NetBEUI	**NetBIOS**	
Transport		**TCP/UDP**	TCP/UDP
Network		**IP**	IP
Link	802.2 802.3, 802.5	LAN: Ethernet, token-ring and others	
Physical		UI MAC frames	

3.4.1.2 What can Samba do?

Samba translates file attributes, file names, user authentication and network architecture between UNIX/Linux and Windows systems, which in part explains why it is so large and complex. At its most elemental level, Samba is composed of two key programs: `smbd` and `nmbd`. Together these provide the basic file and print services, authentication/authorization, name resolution, and browsing, and ensure that existing Windows machines can use the Linux Samba server just as if it were another Windows machine - no additional client software is required.

Table 8. Key Samba programs and their roles

daemon	**service**
smbd	file and print; share mode and user mode authentication and authorization
nmbd	name resolution and browsing

This is the basic picture; now we will discuss the function provided by Samba a bit more into detail.

File and printer sharing is used to provide multiple machines and users access to the same, central resource for storing and printing their files. Shares exported by Samba servers look like normal, regular shares given by Windows machines.

Different mechanisms can be used to grant access to resources on the Samba server. Users can be given access to a resource based on their username and password or based on one password assigned to this specific share. Samba can manage the users on its own or trust an established Windows domain controller. It can even act as a domain controller managing authentication and authorization for a whole domain of Windows clients replacing Windows NT servers completely. This means that Samba on Linux can replace an existing Windows NT server completely or else can be installed alongside an existing NT server and can use the existing userid/password mechanism already in place on the NT server without the need to define a separate, new access control mechanism.

NetBIOS name resolution, in which computer names are matched to IP addresses, can be done in one of two ways:

1. Using broadcasts, which means that whenever one machine wants to talk to another machine and only knows its name, it shouts out “What’s the address of the machine called ‘XYZ’?” throughout the local network and XYZ answers saying “I’m at ‘A.B.C.D’”. These machines are know as “broadcast nodes” or “B nodes” and the broadcast frames are UDP datagrams, normally limited to a single IP subnet but capable of generating a large amount of network traffic.

- Using NetBIOS name and datagram distribution servers, usually called WINS (Windows Internet Name Service) servers, in which every device registers its name/IP address pair with the WINS server. Name resolution is now performed by querying the WINS server, providing the name of the NetBIOS session partner and receiving the IP address of the partner in return. This is where Samba comes into the picture again, because a WINS server can be implemented using Samba. Although the IP address of the WINS server must be pre-configured in each client, this address will typically be provided by the DHCP server in the network - and, again, the DHCP server function can be implemented on Linux.

Last, but not least, Samba can provide browse services. These browse services refer to a list of available shares who can be seen when opening the “Network Neighborhood” from Windows. So, again, a Linux machine running Samba appears in an identical manner to a Windows NT server to other clients on the network.

Rather than configuring Samba by editing plain text configuration files, two major tools are available with most Linux distributions:

1. SWAT (Samba Web Administration Tool) - a web based GUI tool and
2. linuxconf - a tool included in some of the major distributions.

SWAT is attractive because it can be used by any client with a Web browser and with network connectivity to the Linux machine: SWAT itself acts as just like a Web server. SWAT is configured by defining a TCP port in `\etc\services` on which the application listens for connections (typically 901 or 910; since the port is numbered lower than 1024 this means that the SWAT application must run with root privileges) and by defining the application program itself in `\etc\inetd.conf`.

Linuxconf is attractive because it bundles configuration of many different Linux features into a single configuration tool, and therefore provides a single mechanism for configuring all the services on a Linux machine.

Screenshots of these tools are provided below.

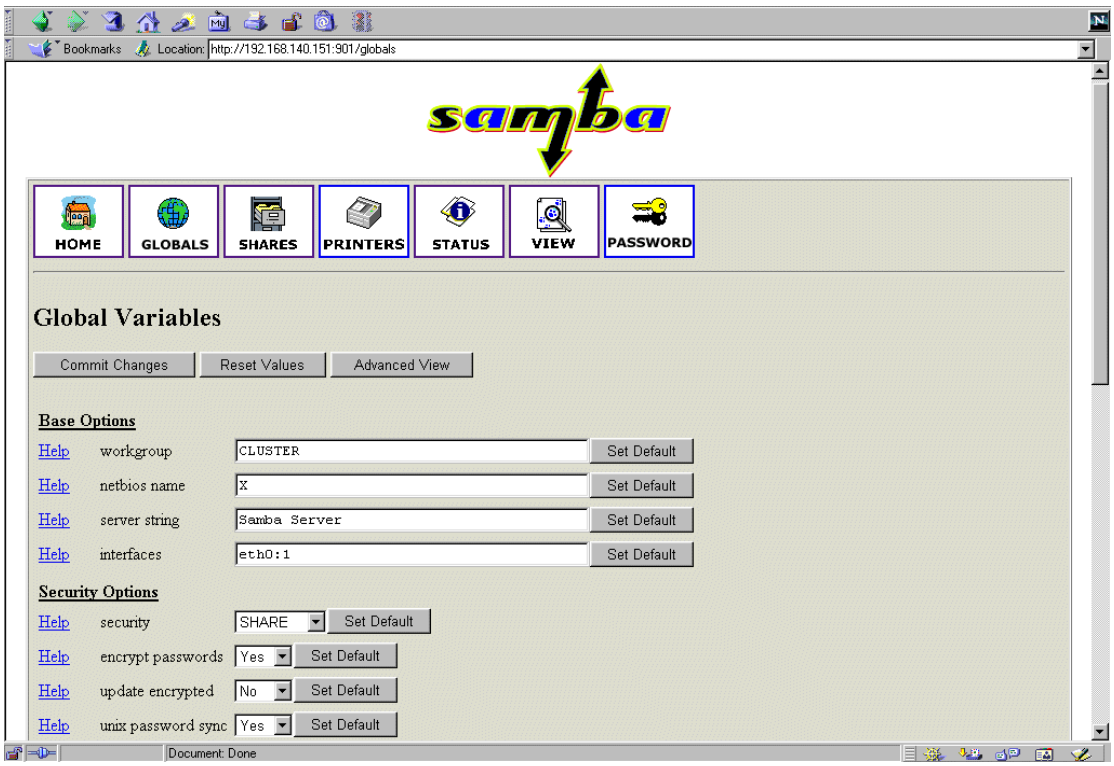


Figure 53. SWAT - managing Samba over the web

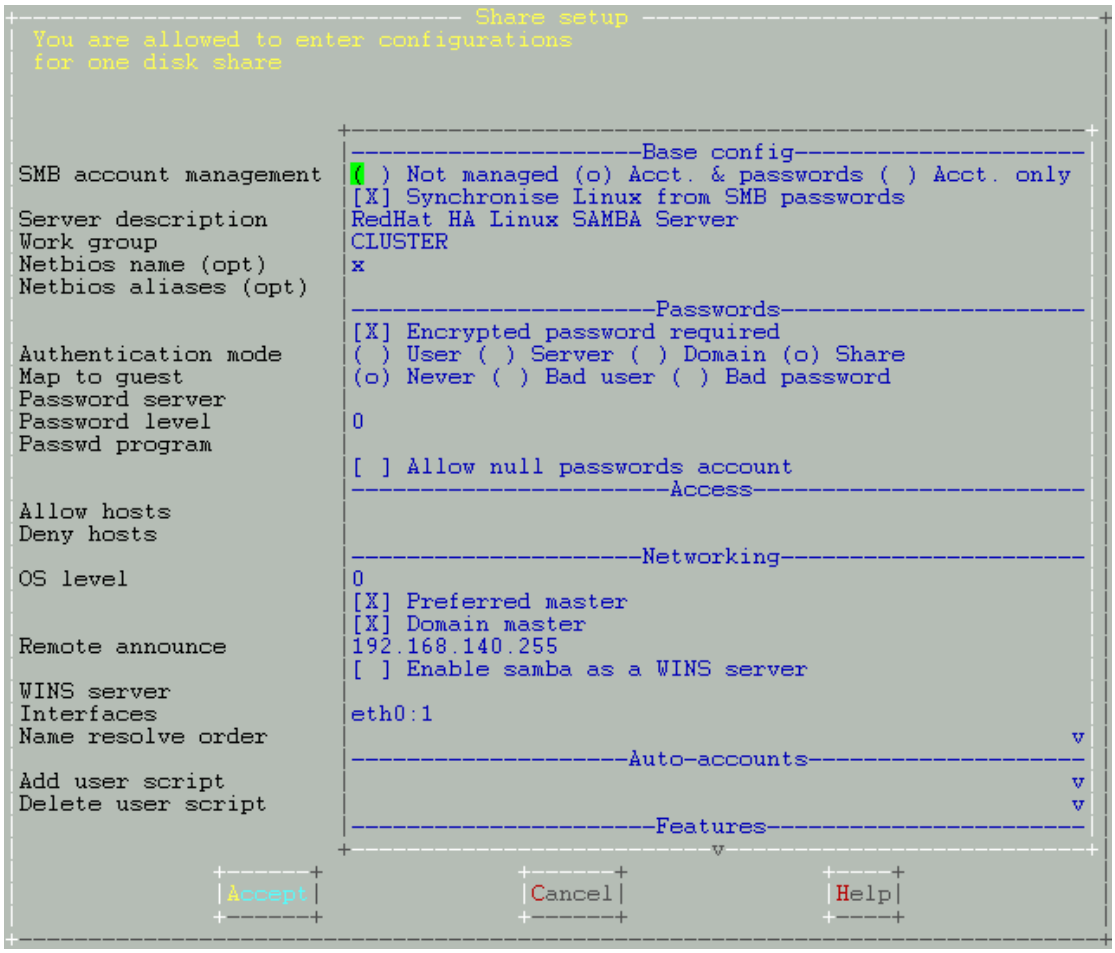


Figure 54. Samba configuration via linuxconf

Linux also provides a SMB file system client: Smbfs. Smbfs allows a Linux system to act as a client to Windows machines, allowing Linux machines to mount file system shares exported by Windows machines. Smbfs can even be used for one Linux machine to mount a share exported by another Linux machine using Samba, although NFS would normally be the preferred method of sharing file systems between Linux machines.

Having discussed some of the technical features of Samba, we will now go on to demonstrate some environments in which Samba could be used.

The simplest Samba setup is a Linux box using only the client functionality to access a file share or printer provided by another SMB-based server. This allows a Linux system to share existing Windows resources in the network:

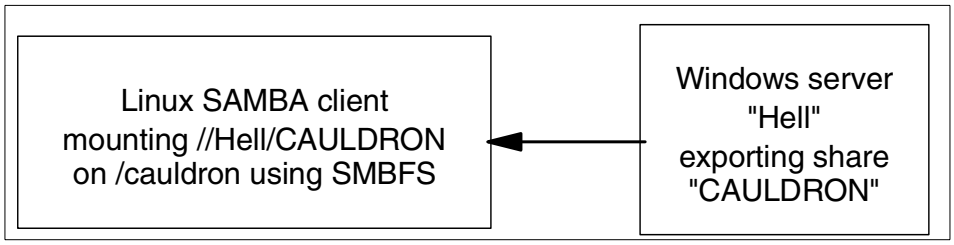


Figure 55. Linux acting as a Samba client

Where Samba really gets interesting is the environment in which Linux acts as a Samba server, sharing its resources to existing Windows (and OS/2) clients in the network. In the following diagram authentication is still handled by an existing Windows domain controller:

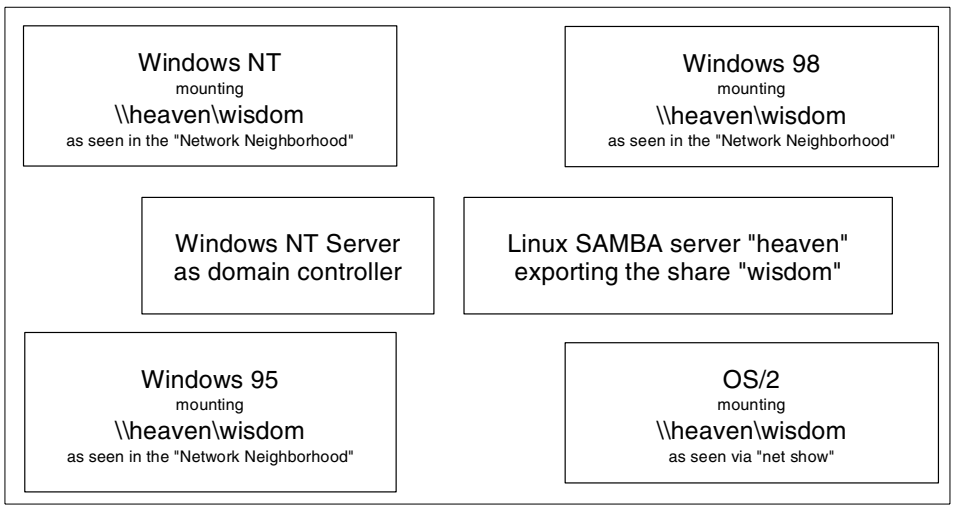


Figure 56. Linux exporting a share for some workstations, Windows used as domain controller

Ultimately, of course, even the Windows domain controller can be replaced by a Linux box, as the latest version of Samba is able to mime a domain controller. A *Windows domain* is a workgroup of machines using SMB with the addition of a Windows server acting as *domain controller*. The protocols which Microsoft has devised for communication between the domain

controller and the other devices in the network is proprietary, and is also different for:

- Domain controller to Windows 95/98 communications
- Domain controller to Windows NT communications
- Domain controller to Windows 2000 communications

Samba has *reverse-engineered* the proprietary Microsoft protocol, and initially provided primary domain controller (PDC) support for Windows 95/98 clients alone. Samba release 2.0 provided rudimentary NT PDC support as well, but this code is still under development with a fully stable release due to be available in Samba release 2.1. At the time of writing, Samba 2.0.7 appears to be the latest release of Samba available for Intel Linux operating systems.

Samba does not currently work as a backup domain controller (BDC). This is because, again, the protocol Microsoft uses to synchronize information between primary and backup domain controllers is proprietary. Each backup domain controller must maintain its own copy of the domain's security account manager (SAM) data, and currently only Windows operating systems are capable of implementing this function.

Samba can act as a *domain master browser* and also as a *local master browser*. Master browsers are used to cut down on broadcast network traffic; if a Windows user wants to view a list of machines available on a network then in fact the request is sent to a master browser which maintains a list of supposedly active machines on the network. Almost any Windows machine as well as Samba can act as a local master browser and Windows machines can also act as a backup local master browser; the actual local master browser in use is determined by an election process which favours operating systems such as NT 4.0 over Windows 98, amongst other things. Samba can be configured so that it is always the master browser, if required, or it can be configured to yield the role to other machines - the election can be "rigged"! Domain master browsers, on the other hand, are always the same machine as the primary domain controller, and can be used to maintain a browse list for a single domain spanning multiple IP subnets, if required. It happens, however, that neither Windows 95 nor Windows 98 clients can even contact a domain master browser, let alone actually become one, which means that if a workgroup is required to span multiple IP subnets then a Windows NT or Samba server is *required* in order for Windows 95/98 clients to be able to see all machines on the single workgroup or domain.

What all this means is that identification of a machine in a network ("Network Neighborhood" under Windows) is performed without actually going to each

and every machine in the network: the list of machines is supplied by the master browser. Browsing the actual resources on a particular machine, however, is accomplished by connecting to the machine in question directly. And this, in turn, means that the Windows Network Neighborhood may in fact show machines which have crashed, or may not show machines which are in fact connected to the network. However, once a particular machine has been selected, any shares and printers subsequently displayed on that machine really must exist on the network at that time.

As previously discussed, Samba can function as a WINS server, otherwise known as a NetBIOS name server (nbns). WINS implements a flat database containing only pairs of machine names and IP addresses. WINS is not specific to a domain or workgroup - WINS servers can reside anywhere in the IP network and can serve anyone. In other words, a single WINS server can serve machine in many different domains. Now, Microsoft allows for multiple WINS servers in a network by synchronizing information between primary and backup servers periodically. This is theoretically elegant, allowing clients to be configured with the IP address of the closest mirrored WINS server, but in practice it can be quite cumbersome, with multiple WINS servers struggling to keep themselves synchronized all the time. For most networks it will probably be simplest to implement a single, authoritative WINS server, and Samba can certainly perform this function. Microsoft also allows the configuration of a secondary WINS server, which will take over the WINS server function if the primary WINS server should fail. This configuration is pre-determined (there is no election process, unlike with master browsers) and Samba does not support this function.

In summary, then: Samba implements most of the functions implemented by Windows servers, allowing file/print resources to be shared with Windows clients of all types. Samba has had to reverse-engineer protocols proprietary to Microsoft and does not yet implement absolutely all of the domain/browsing and WINS functions offered by Microsoft software, but Samba does provide all the important functions, allowing Linux to be dropped in to an existing Windows network with ease. Samba can’t be a backup domain controller, a local backup browser or a secondary WINS server and Samba implementations of primary domain controllers should use the most current Samba code release (ideally 2.1or higher).

Much of the preceding discussion has considered Linux in the role of a server, providing file and printer sharing capabilities to network clients. But there’s also a role for Linux as a client: one useful capability is the ability of a Linux system to send WinPopup messages to other machines on the network. Even if no other capabilities of Samba were implemented, this would allow a

Linux machine to notify one or all of the Windows machines of the network of events such as:

- The arrival of e-mail on a Linux mail server
- An alarm for a particular time of day
- An alert from a network management event

It is easy to construct a script on a Linux machine to send a message as shown in the “man” page for `smbclient`:

```
cat mymessage.txt | smbclient -M FRED
```

An alternative approach to Samba would be one in which all the Windows machines use native TCP/IP protocols such as network file system (NFS). NFS would allow Windows clients to `mount` Linux file systems across a network, achieving the same end as connecting to a Samba share across the same network. The drawback of using NFS is that all the clients have to have additional code installed, and furthermore the code is not free. In a typical environment in which Linux is to be added to a Windows network there are likely to be many Windows clients and one or few Linux servers: it normally makes most sense to install Samba on the Linux servers rather than NFS clients on all the Windows clients.

3.4.1.3 Benefits and drawbacks

- Supposedly robust, but SMB relies on the client to re-establish lost connections.
- Much more complex than NFS
- Less network traffic than NFS, at the cost of hundreds of commands
- SMB servers must be large and complex
- SMB can’t be implemented in the kernel (too big), which will be a negative performance impact on the Linux machine
- SMB combines many of the features of NFS, lpd, and Kerberos (or DCE).
- Ease of administration: server-side install of Samba vs. each client having to be installed with networking software
- Some Windows apps rely on features not found in NFS
- Printers on UNIX hosts can be shared with Windows clients
- user level or share level security...user level means users must provide unique ID and password to gain specific rights. Share level means password provided per service rather than per user ID.

- available on very different hardware platforms, from small, old i386 box to large, powerful S/390 mainframe

As operating systems such as DOS, Windows and OS/2 matured, there was the requirement to continually enhance the networking capabilities of SMB. Samba has kept pace with these developments and even provided for back level implementations of the various versions of Windows by providing support for the following extensions.

Table 9. SMB protocol extensions supported by Samba

Core	PC Network Program 1.0
Core Plus	Microsoft Networks 1.03
LAN Manager 1.0	Support for OS/2 multitasking (and opportunistic locks)
LANManager 2.0	Enhanced support for OS/2 (long file names)
NT LM 0.12	New commands to support Windows NT
CIFS 1.0	Essentially a Microsoft version of NFS

http://samba.anu.edu.au/pub/samba/

3.4.2 Linux-SNA gets you to the mainframes

Unix servers are not the only servers available today, nor is TCP/IP the only networking protocol. Another very important and much used protocol suite is Systems Network Architecture (SNA) developed by IBM in the 1970s and very widely used even today in IBM mainframe environments. In particular, although SNA is described as “legacy” and “old” (in fact, SNA and TCP/IP were both developed at roughly the same time) the real importance of SNA is that the large majority of “mission critical” mainframe applications still rely on SNA. One of the challenges of today’s networks is in combining the reliability of SNA networking with the availability and widespread adoption of TCP/IP networking and the Internet.

Linux-SNA allows Linux to implement a SNA protocol stack. This then means that Linux machines can be included in an existing SNA environment and - because Linux already uses TCP/IP - Linux can act as a gateway between the SNA mainframe “legacy” world and the TCP/IP Internet.

Linux-SNA supports APPN End Node and Network Node functions over token-ring and Ethernet LAN connections. The APPN support is a subset of all the defined option sets, and today does not include high performance routing (HPR), dependent LU requestor (DLUR), border node (PBN or EBN) or parallel TG support, for example. Indeed, the APPN NN implementation

does not support many of the directory and topology options (registration of end nodes, broadcast/directed searches, topology exchanges) or transport options such as intermediate session routing (ISR). As it stands today, Linux-SNA is not an APPN router.

On the other hand, there are many related SNA functions which Linux-SNA can support, and perhaps the most important role Linux-SNA can perform is to act as a gateway between TCP/IP clients and SNA servers. Linux-SNA is currently being developed, but the first full and stable release (1.0) should include TN3270 and TN5250 client and server support and the ability to use SNA over LAN connections, S/390 host channel connections and SDLC connections; at the time of writing the channel and SDLC datalink controls have not been completed.

Linux-SNA is a project led by Jay Schulist of TurboLinux to implement SNA support into Linux. It consists of the following parts:

Table 10. Linux-SNA components

Component name	Functionality
net-fun	kernel patch to get SNA kernel support
libsna	SNA support libraries
sna-tools	base set of SNA networking tools
asuite	APPC tools, like aftp, aping etc.
etc-sna	contains the SNA config files going into /etc
llcping	LLC test program
tn5250	TN5250 client
tn5250d	TN5250 server
tn3270	TN3270 client
tn3270d	TN3270 server

Linux-SNA currently requires a relatively recent Linux kernel (2.2.x) to which a patch must be applied so that additional SNA options can be added. Once the new kernel has been loaded and configuration files have been edited,

Linux-SNA should be up and running. An overview of the planned and supported devices, protocols, and APIs is shown in the following table:

Table 11. Linux-SNA supported devices, protocols, and APIs

Devices	Protocols	APIs
Ethernet, Token Ring	APPN	CPI-C V2
ATM, FIDDI, SDLC	LU 6.2	APPC
ESCON, Bus/Tag Channel		LU API
5250 Twinaxial, 3270 Coax		HILLAPI
X.25		Linux-SNA native

But note that at the time of writing this code is under development and test and that support is only currently available for Ethernet and token-ring devices, for example.

Linux-SNA provides a low-cost entry point into SNA networking from the Linux world. Of course there are existing commercial solutions, not least of which is IBM's Communications Server, which runs on IBM's UNIX (AIX on RS/6000) as well as on Windows NT on Netfinity hardware, and which provides many more features and functions than Linux-SNA does. Furthermore, SNA servers such as the IBM S/390 and AS/400 both provide native TCP/IP access to their applications and services, allowing TCP/IP clients to connect to them directly and obviating the need for a gateway such as Linux-SNA. But there are still places where it would be convenient to have Linux-SNA support, and in due course Linux-SNA will be able to implement:

- TCP/IP - SNA gateways, opening routes from one network world to the other. This is a Linux-SNA implementation of AnyNet, allowing IP traffic to be transported over SNA networks or vice-versa.
- TN3270/5250 gateways, enabling todays clients to connect to your mainframe and pull and push data
- SNA protocol converters, providing your new applications a way to talk your applications on the mainframe

and more solutions, fitting your needs to bring the mainframe worlds and the Linux worlds together, combining the advantages from both sides.

- using existing applications get new interfaces to the fast growing Linux world
- using of cheap commodity access servers to your installed business backbone

- customizing Open Source solutions to fit your needs and get up to date software
- building stable and affordable bridges between formerly separated architectures

At the bottom line, Linux-SNA proves once again, that Linux is business ready, flexible and easily integrated in today's computing environments in an affordable manner.

3.4.3 File systems and other computing environments

There are a more interoperability issues then the ones Samba and Linux-SNA aim to solve. Some of these include:

3.4.3.1 AFS

A widespread and commonly-used distributed network file system is the Andrew File System (AFS), named after a research project at Carnegie-Mellon University which originally developed it. It's used in companies and universities to provide transparent access to data stored on one or more file servers forming a single AFS cell. A cell is a collection of servers grouped together administratively and managed as a single system, not as individual file servers. AFS is an Enterprise File System product supplied by Transarc Corporation and is available for a wide majority of current operating systems, primarily UNIX systems:

- Linux
- AIX
- HP-UX
- Solaris
- Windows (as a client for Windows NT Workstation and Server)
- IRIX
- Digital Unix

You can setup your Linux box to be either an AFS server, providing access to an AFS cell or to act as an AFS client, taking part in an already established AFS cell. Using AFS, you can share data transparently, securely, easily, quickly and reliably between different platforms. AFS is available from IBM for all these different platforms, making it easy for you to build a tightly connected, integrating file and data sharing environment for your business.

As a "native" TCP/IP solution, AFS compares well with NFS in that it scales better and is faster in medium-sized and larger installations. Windows clients

can connect directly to AFS or can use Samba to connect to an exported SMB share provided by the Linux server.

3.4.3.2 Linux & Novell

Novell NetWare solutions are still very widespread and have delivered stable network environments for many years. Novell Netware services provides printer and file sharing to its clients, for example, in a similar manner to SMB in the Windows environment except that Novell uses the IPX protocol instead of NetBEUI or TCP/IP.

Linux has included IPX support in the kernel in all 2.x kernel versions and can take on the following roles:

- Netware file/print client (using ncpfs)
- Netware print server (using ncpfs and/or mars_nwe)
- Netware file server (using mars_nwe)

The Linux kernel does not yet support additional related protocols used by NetWare for other purposes such as SAP or IPX/RIP or NCP, but these protocols can be supported by other operating system software. With these enhancements, Linux can operate as a full IPX bridge or router. `ipxripd` and `mars_nwe` both provide software which maintaints the IPX routing table.

As with most networking implementations, there are alternative solutions available. In order to access data stored on a Novell file server from a Linux client there are two options:

1. Make the file server compatible with Linux by implementing a NFS file service on the Novell platform, allowing Linux to access data using standard NFS “mount” commands.
2. Make the Linux client compatible with the Novell file server by adding Linux code to be able to mount Novell volumes into the Linux filesytem.

Novell has long provided an NFS server package, but there may be insufficient justification (too little potential usage?) for the purchase cost of this software. Linux, of course, provides the `ncpfs` package which will work with Novell 3.x and later servers and is essentially free.

Table 12. Parts needed to access a Novell Netware server

Component	Functionality
Linux kernel 2.2.x or greater	Base IPX and NCP file system support
ncpfs (e.g. as RPM)	NCP mount and print tools

When compiling or installing the kernel, make sure it has support for IPX and the NCP file system. Then, when everything is installed correctly, the

`ncpmount`

and

`ncpumount`

can be used (analogous to the NFS `mount` and `umount` commands) to define remote Novell file server shares using IPX. Once the Novell file system has been mounted locally it will appear at the specified mount point in the local Linux file system; permissions and ownership will be determined by the Novell server based on the userid which issued the `ncpmount` command.

The `ipxutils` package will generally be required on the Linux machine as well; this includes utilities such as a utility for sending messages to other users of the Novell network.

The `ncpfs` package also provides a simple way of using Linux as a Novell print server: a print queue can be defined on a Novell server and the ncpfs can be used to retrieve print jobs from the Novell server and print them on the Linux system, for example on a printer using `lpr`.

On the other hand, the `mars_nwe` package (which stands for Martin Stover's NetWare Emulator) provides full Novell NCP support for file and print sharing as a NetWare server. Again, the Linux kernel needs to be compiled with IPX support, and `mars_nwe` installs a configuration file `/etc/nwserv.conf`. Its printing services are independent from the ones provided by ncpfs.

Besides Novell NetWare, there's another very important product from Novell: The Novell Directory Service (NDS). NDS eDirectory is a full featured directory service based on open standards for managing and securing network resources. It's able to integrating multiple platforms into one single directory structure. NDS eDirectory is available for Linux and many different other platforms in competition with Microsoft's Active Directory.

Novell announced its NDS eDirectory in February 2000, along with NDS Corporate Edition for Linux. The eDirectory product can be used to build a custom directory, and the major alternative products are the iPlanet Directory Server (formerly Netscape Directory Server) and various forms of Lightweight Directory Access Protocol (LDAP) servers. Active Directory is the main competitor for the Corporate product, and NDS for Linux can be used to integrate NetWare, Linux and Windows NT/2000 environments into a single directory. Of particular interest is the capability to provide a single sign-on

(SSO) environment for all network resources: users log on once and are then authenticated to use all the applications, domains and shares in the entire network. NDS is not free software, it currently costs $26 per use for NDS Corporate Edition.

Implementing NDS is not simple, but the benefits may be substantial. Using any sort of directory service for the first time requires careful consideration of the existing network model and how these services are to be used. NDS is only available for certain operating system platforms, but these include all the major file/print server platforms currently available.

3.5 Outlook

To give an outlook, let's first take a look a few years back. At this time companies were developing and selling mostly proprietary protocols, products, and whole proprietary solutions. It were complete, out-of-the box products with a certain, but because of this concept limited functionality. So if you wanted to enhance this solution by e.g. integrating another machine or service you were depended on your one, single solution provider to enable this.

Now the picture is changing and has already changed a lot, because of Linux and all the Open-Source software available. Today Linux is used to provide fast to implement, low-cost solutions, connecting very different platforms. It's mostly acting as a gateway or front end to your already deployed solutions.

Linux offers a wide variety of supported network protocols, file systems, and server services making it one of the number one choices for your future business integration platform. It's growing as fast as your business, being as much flexible to fit your changing demands and keeps you prepared for the future.

3.5.1 Where to go for more information

- Samba: `http://www.samba.org/`
- Linux-SNA: `http://www.linux-sna.org/`
- AFS: `http://www.transarc.com/`
- ncpfs: `ftp://ftp.gwdg.de:/pub/linux/misc/ncpfs/`
- Caldera: `http://www.calderasystems.com/`
- mars_nwe: `http://www.compu-art.de/mars_nwe/`
- Novell NDS: `http://www.novell.com/products/nds/`

Appendix A. Special notices

This publication is intended to help anyone wanting to know more about some interesting Linux topics and in particular how to implement Linux solutions using Netfinity hardware. The intention of these papers is to document both “why” and “how” certain solutions were implemented by us, with the intent that this information will help anyone else attempting the same sort of thing.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
APPN
AT
CUA
DB2
IBM
Netfinity
OS/390
Parallel Sysplex
RS/6000
SecureWay
System/390
Wake on LAN
Wizard
Lotus
Manage. Anything. Anywhere.
NetView
Tivoli Ready
IBM ®
AnyNet
AS/400
CT
Current
ESCON
Manage. Anything. Anywhere.
OS/2
OS/400
PS/2
S/390
SP
TCS
WebSphere
400
Tivoli
TME
Cross-Site
Tivoli Certified

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United

States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 113.

- *Netfinity and TurboLinux Integration Guide*, SG24-5862
- *Netfinity and Red Hat Linux Server Integration Guide*, SG24-5853
- *Implementing Linux in your Network using Samba*, REDP0023

B.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

B.3 Other resources

These publications are also relevant as further information sources:

- *????full title???????*, xxxx-xxxx
- *????full title???????*, xxxx-xxxx
- *????full title???????*, xxxx-xxxx

B.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- http://tux.boeblingen.de.ibm.com
- http://www.ask-the-guru,com
- http://www.beowulf.org
- http://www.linuxvirtualserver.org
- http://www.mosix.org
- http://www.polyserve.com
- http://www.bcsoft.com/wlman.html
- http://www.ibm.com/software/webservers/perfpack
- http://www.turbolinux.com/products/tcs
- http://www.solutions6000.com/uk_framesoftware.htm
- http://www.resonate.com/products/central_dispatch
- http://www.twincom.com/netdisc.html
- http://linux-ha.org/commercial.html
- http://www.high-availability.com
- http://people.redhat.com/kbarrett/HA
- http://www.globalfilesystem.org
- http://www.inter-mezzo.org
- http://home.xnet.com/~blatura/linapps.shtml
- http://www.redhat.com/support/manuals/RHL-6.2-Manual/ref-guide
- http://ibm.com/linux
- http://www.pc.ibm.com/support
- http://www.linuxdoc.org/LDP/nag2/index.html
- http://www2.linuxjournal.com/cgi-bin/frames.pl/index.html
- http://samba.anu.edu.au/pub/samba
- http://www.transarc.com
- http://www.compu-art.de/mars_nwe
- http://www.samba.org
- http://www.linux-sna.org
- http://www.novell.com/products/nds
- http://www.calderasystems.com
- http://www.zope.org
- http://w3.itso.ibm.com
- http://w3.ibm.com

- http://www.elink.ibmlink.ibm.com/pbl/pbl

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

T

U

V

W

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5994-00 Linux on IBM Netfinity servers A collection of papers
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

Linux on IBM Netfinity servers A collection of papers

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Linux on IBM Netfinity servers

A collection of papers

Redbooks

Systems and network management

High availability clustering

Interoperability

This book contains three separate papers:
1.) Linux systems and network management, showing remote management and monitoring capabilities including remote installation and upgrade of Linux systems.
2.) High availability clustering, the theory and practice of clustering solutions, in particular TurboCluster using TurboLinux to set up a load-balancing environment for several services and Red Hat High Availability Server to set up a fail-over environment.
3.) Interoperability: how to use Linux servers in conjunction with servers and clients on other platforms, Samba in particular (for Windows environments) but also NetWare, SNA and other considerations.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5994-00 ISBN