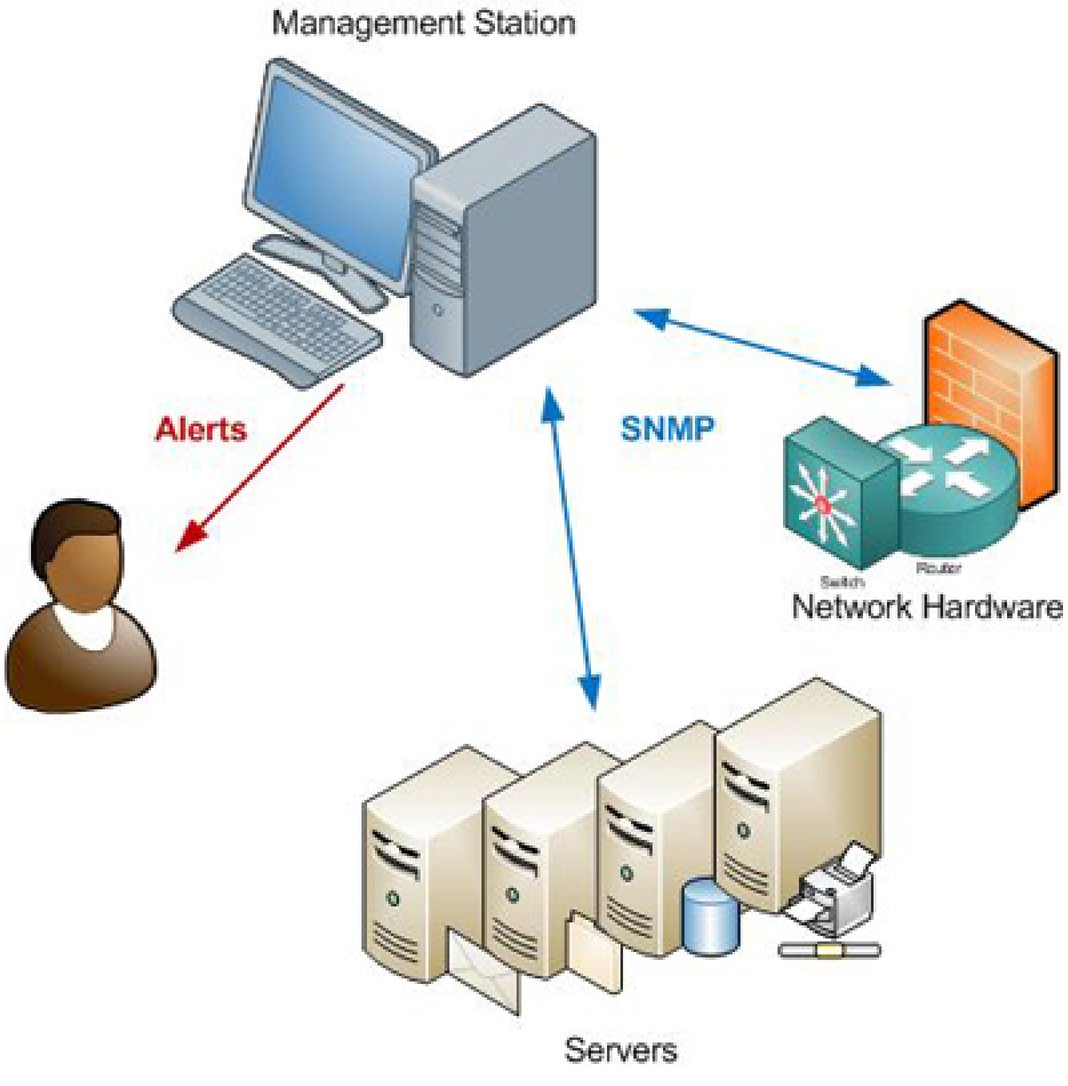
Network Management & Computer System Administration



Gwyneth Dionne

Misael Ratliff

First Edition, 2012

ISBN 978-81-323-0960-4

Published by:
Academic Studio
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter 1 - Out-of-Band Management

Chapter 2 - Intrusion Detection System

Chapter 3 - Computer Performance and Network Management

Chapter 4 - Automounter

Chapter 5 - Application Performance Management and Business Transaction Management

Chapter 6 - Deep Freeze (Software) and Multiseat Configuration

Chapter 7 - Intelligent Platform Management Interface

Chapter 8 - Magic SysRq Key

Chapter 9 - Password Cracking and Remote Administration

Chapter 10 - Software Deployment and System Console

Chapter 11 - System Monitor and System Profiler

Chapter 12 - Systems Management and Windows Management Instrumentation

Chapter 13 - Autonomic Networking

Chapter 14 - FCAPS

Chapter 15 - Simple Network Management Protocol and Java Management Extensions

Chapter 16 - Load Balancing (Computing) and Network Administrator

Chapter 17 - Management Information Base and NETCONF

Chapter 18 - NetCrunch and Netcat

Chapter 19 - Network Operations Center and OpenNMS

Chapter 20 - Opsi

Chapter 21 - RRDtool, Rsyslog and Syslog-ng

Chapter 1

Out-of-Band Management

In computing, **out-of-band management** (sometimes called **lights-out management** or **LOM**) involves the use of a dedicated management channel for device maintenance. It allows a system administrator to monitor and manage servers and other network equipment by remote control regardless of whether the machine is powered on.

By contrast, in-band management is the use of regular data channels (usually through Ethernet) to manage devices. A significant limitation of in-band management is its vulnerability to problems from the very devices that are being managed. To manage network servers and routers remotely, IT administrators need network access when problems occur. However, the same problems that cause the network to go down also result in the loss of management access to those devices.

Out-of-band management addresses this limitation by employing a management channel that is physically isolated from the data channel.

History

In the early 1980s, the concept of out-of-band was adapted for its natural application across the emerging **data** transmission network structures being introduced with the onset of Ethernet and cost-effective wide area networks. Network architects recognized that this out-of-band alternative pathway was a key requirement in service availability, and they could readily apply many of the lessons learned within the telecoms industry for the previous 30 years. Some of the earliest implementations of a **data** network out-of-band structure included the attachment of a single modem to any given server—in essence creating a very small out-of-band infrastructure (OOBI). Vendors such as IBM, DEC, HP, and Data General made very lucrative service businesses by providing such out-of-band tools as subscription-based services products. The ability to interact remotely with servers that were otherwise compromised was a powerful one which gave rise to the growth of out-of-band as a more general tool for data networks.

In the mid-1980s, Encore Computer released the Annex terminal server, later purchased by Xylogics. The Annex was capable of serving one parallel printer and up to 16 serial printers, terminals, modems, or serial consoles to various equipment. Later models supported more serial lines, making it the first OOB management server. It also supported a "reverse telnet" (aka. rtelnet) feature that, through a daemon, created a character device file on the Unix host where it ran. Opening this device created a connection to the pre-configured port on the Annex, thus supporting remote kernel debugging, remote modems, etc.

Beginning in the year 2000, the concept was formalized by many early out-of-band infrastructure data pioneers. It was quite clear that this technology was quickly becoming a core IT requirement when dealing with service levels across hundreds or thousands of geographically dispersed IT assets. **OOBI**, uses many of the same concepts and provides similar features to the telecom industry's out-of-band infrastructures. Vendors of **OOBI** solutions began offering these cost-effective alternatives to local administration for data system and network management.

In its original conception, OOBI referred to the physical architecture and components that were used to construct an out-of-band network. A more accurate description would be a *service port network* since the OOBI connected to the service ports rather than the data ports on the target devices. This network provided the platform to implement out-of-band management (or service port management). Just as in the past, a data OOBI provides alternate paths into the production infrastructure for the purpose of allowing disconnected assets to be remotely reconnected and subsequently returned to normal operation, in most cases eliminating the need for costly local administration. Some OOBI implementations include inherent enterprise-class security while others are constrained to the attributes of limited or proprietary mechanisms. An OOBI can improve operational efficiencies, cut costs, improve productivity and, in many cases, improve service levels and asset availability. Conceptually, data OOB infrastructures virtually guarantee a data "dial tone."

An example of a modern, open remote management interface is IPMI, which in physical terms uses data ports but allows their use as service ports, with a low-level route to a baseboard management controller of some sort configured such that OOB services can be accessed separately over the same IP network. While IPMI does provide remote access to computers even when the OS is down, it actually does not provide completely out-of-band access, because it shares the Network Interface Controller with the data pathway. True out of band access is provided by using a dedicated NIC . This is accomplished by using a **Remote Access Card** (RAC), or, more recently, through a Soft Processor leveraging a highly integrated BMC (iBMC).

Types of management systems

A complete LOM system consists of a hardware component called the LOM module and a program that facilitates the continuous monitoring of variables such as microprocessor temperature and utilization. The program also allows for such remote operations as rebooting, shutdown, troubleshooting, alarm setting, fan speed control, and operating system reinstallation. The program often integrates into traditional infrastructure in-band management tools such as HP Openview, Computer Associates, BMC, and Tivoli.

The most common out-of-band management solution involves connecting each device's serial console port to a console server. This implementation allows the monitoring of hardware self-test information and console access that is not available using typical in-band management.

Another type of management solution, a **Remote Access Card** (RAC), involves an expansion card for a computer which has its own processor, memory, battery, network connection, and access to the system bus. This system is effective but costly, and is being progressively supplanted by the use of dedicated Systems on Chip, also called Integrated Baseboard Management Controllers.

Some LOM systems function with more than one server, especially if combined with a KVM switch. When combined with a terminal server, administrators may access all serial console ports in a network or server farm from a single station. If the terminal server is also configured with network, Internet, and dial-up access, administrators will be able to manage network problems from any remote location, even if the network connection has been lost.

Communication between the controller and the remote servers sometimes takes place through an independent dial-up connection. More commonly nowadays, the LOM modules are connected by serial links to a separate management host; or the LOM module accepts telnet connections over an Ethernet connection. Either way, the LOM can then be remotely accessed over the Internet (through SSH to the management host, and/or a VPN). The LOM module keeps a record of all the operations (known as the event log), allowing the administrator to check instantly any or all of several hundred systems.

SoC based service Processors

Modern systems based on System-on-Chip , or iBMCs, usually have separate Ethernet connection that can be implemented either through dedicated or shared Ethernet port. This connection has its own IP address and other network settings. It remains functional also if the server is powered down and can power it up when required. Systems provide remote screen view (both graphical and text modes), remote mouse and keyboard and remote virtual media (including media that exists only as .iso images on the administrator machine). This access is not dependent from the operating system on the server and also works when managing remotely BIOS settings, for instance. IPMI connection is normally encrypted. Recent server boards have all this functionality built-in and do not require any additional extension cards. IPMI system can also be accessed from inside the server if required (for instance, to read the hardware sensor values). Interaction with the system on remote side can be implemented through web browser (including Java applets or JNLP) or specialized tools that may be cross platform. Open source software like FreeIpmi is also available. The most commonly used iBMCs include ASPEED AST 2050, Nuvoton WC450 (Hermon), Renesas 2164, Server Engines Pilot II and Pilot III.

Console redirection

Embedded firmware of most server motherboards support (BIOS) serial console redirection. And boot parameters of modern operating systems can be changed through the boot loader console which supports redirection as well (in Linux this is LILO, GRUB, or SYSLINUX). A Microsoft Windows feature is EMS. Furthermore, Unix-like systems can be configured to log kernel messages to their (serial) console too. The Linux kernel for example logs all messages to all its configured consoles, which can be a combination of virtual terminals (graphics card/keyboard combination), serial ports, parallel ports, etc. Management software such as the Conserver automatically captures this data, and can replay it if needed. When using serial console servers care should be taken not to send any unsolicited BREAK over the line (especially with Sun hardware, and also Linux if SysRq is enabled) as it can put the machine in "LOM mode" otherwise. Some solutions use a Java applet to display remote console view to the administrator.

Limitations

Servicing and managing computer servers in a remote data center can require the physical presence of a system administrator. For example, the loading or removal of media, or direct interaction with the server through a console and keyboard (which should only ever be needed if the CMOS NVRAM becomes corrupted). Such access requirements depend on a system administrator being co-located with the data center, often an additional business expense.

Specific implementations

- Advanced Lights Out Management (ALOM), Sun Microsystems-specific and comes standard on newer Sun servers (SunFire V125/V210/V215/V240/V245/V250/V440/T1000, Sun Netra 210/240/440)
- Integrated Lights Out Management (ILOM), Sun Microsystems's ALOM replacement on Sun x64 server SunFire X4100(M2)/X4200(M2)/X4600(M2)/X4140/X4240/X4440/X4150/X4250/X4450/X4170/X4270/X2250/X2270, Sun Blade 6000 Chassis Management Module/Blade Module(X6220/X6420/X6240/X6440/X6250/X6450/X6270/X6275), Sun CMT servers/blades (Sun T5120, T5220, T5240, T6340, T6320)
- American Megatrends' MegaRAC-SP firmware, which powers a number of OEM LOM implementations
- Apple Computer's Xserve, which provides lights-out management through the Ethernet network interface.
- Dell DRAC, the Dell-specific "Dell Remote Access Controller;" also called "Dell Remote Assistance Card" or "Dell Remote Access Card" depending on to which version one is referring
- Fujitsu Integrated Remote Management Controller (iRMC) (manual)
- HP Integrated Lights-Out (HP/Compaq specific)
- IBM Remote supervisor adapter (IBM specific)

- Intel Active Management Technology (iAMT)
- Intelligent Platform Management Interface
- LOM port, Sun Microsystems specific used on their older products (Netras)
- Open Platform Management Architecture (Generic interface)
- PC Weasel 2000 (Personal Computer hardware)
- Remote System Control (RSC) on Sun Microsystems SunFire 280R/V480/V490/V880/V890/VSP servers.
- Winbond Hermon
- Baseboard Management Controller (BMC) for IPMI
- Network Console on Acid (Unix tty redirection over SSH)

Chapter 2

Intrusion Detection System

An **intrusion detection system** (**IDS**) is a device or software application that monitors network and/or system activities for malicious activities or policy violations and produces reports to a Management Station. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

IDPSs typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDPSs can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attack's content.

Terminology

- **Alert/Alarm:** A signal suggesting that a system has been or is being attacked.
- **True Positive:** A legitimate attack which triggers an IDS to produce an alarm.
- **False Positive:** An event signaling an IDS to produce an alarm when no attack has taken place.
- **False Negative:** A failure of an IDS to detect an actual attack.
- **True Negative:** When no attack has taken place and no alarm is raised.
- **Noise:** Data or interference that can trigger a false positive.
- **Site policy:** Guidelines within an organization that control the rules and configurations of an IDS.
- **Site policy awareness:** The ability an IDS has to dynamically change its rules and configurations in response to changing environmental activity.
- **Confidence value:** A value an organization places on an IDS based on past performance and analysis to help determine its ability to effectively identify an attack.
- **Alarm filtering:** The process of categorizing attack alerts produced from an IDS in order to distinguish false positives from actual attacks.

- **Attacker** or **Intruder:** An entity who tries to find a way to gain unauthorized access to information, inflict harm or engage in other malicious activities.
- **Masquerader:** A user who does not have the authority to a system, but tries to access the information as an authorized user. They are generally outside users.
- **Misfeasor:** They are commonly internal users and can be of two types:
    1. An authorized user with limited permissions.
    2. A user with full permissions and who misuses their powers.
- **Clandestine user:** A user who acts as a supervisor and tries to use his privileges so as to avoid being captured.

Types

For the purpose of dealing with IT, there are two main types of IDS:

Network intrusion detection system (NIDS)

It is an independent platform that identifies intrusions by examining network traffic and monitors multiple hosts. Network intrusion detection systems gain access to network traffic by connecting to a network hub, network switch configured for port mirroring, or network tap. In a NIDS, sensors are located at choke points in the network to be monitored, often in the demilitarized zone (DMZ) or at network borders. Sensors captures all network traffic and analyzes the content of individual packets for malicious traffic. An example of a NIDS is Snort.

Host-based intrusion detection system (HIDS)

It consists of an agent on a host that identifies intrusions by analyzing system calls, application logs, file-system modifications (binaries, password files, capability databases, Access control lists, etc.) and other host activities and state. In a HIDS, sensors usually consist of a software agent. Some application-based IDS are also part of this category. An example of a HIDS is OSSEC.

Intrusion detection systems can also be system-specific using custom tools and honeypots. In the case of physical building security, IDS is defined as an alarm system designed to

Passive and/or reactive systems

In a **passive system**, the intrusion detection system (IDS) sensor detects a potential security breach, logs the information and signals an alert on the console and or owner. In a **reactive system**, also known as an intrusion prevention system (IPS), the IPS auto-responds to the suspicious activity by resetting the connection or by reprogramming the firewall to block network traffic from the suspected malicious source. The term IDPS is commonly used where this can happen automatically or at the command of an operator; systems that both "detect" (alert) and/or "prevent."

Comparison with firewalls

Though they both relate to network security, an intrusion detection system (IDS) differs from a firewall in that a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls limit access between networks to prevent intrusion and do not signal an attack from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system, and is another form of an application layer firewall.

Statistical anomaly and signature based IDSes

All Intrusion Detection Systems use one of two detection techniques:

Statistical anomaly-based IDS

A statistical anomaly-based IDS determines normal network activity like what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other- and alert the administrator or user when traffic is detected which is anomalous(not normal).

Signature-based IDS

Signature based IDS monitors packets in the Network and compares with preconfigured and predetermined attack patterns known as signatures. The issue is that there will be lag between the new threat discovered and Signature being applied in IDS for detecting the threat.During this lag time your IDS will be unable to identify the threat.

Limitations

- Noise can severely limit an Intrusion detection system's effectiveness. Bad packets generated from software bugs, corrupt DNS data, and local packets that escaped can create a significantly high false-alarm rate.
- It is not uncommon for the number of real attacks to be far below the false-alarm rate. Real attacks are often so far below the false-alarm rate that they are often missed and ignored.
- Many attacks are geared for specific versions of software that are usually outdated. A constantly changing library of signatures is needed to mitigate threats. Outdated signature databases can leave the IDS vulnerable to new strategies.

Evasion techniques

Intrusion detection system evasion techniques bypass detection by creating different states on the IDS and on the targeted computer. The adversary accomplishes this by manipulating either the attack itself or the network traffic that contains the attack.

Development

A preliminary concept of an IDS began and reviews of audit trails. An example of an audit trail would be a log of user access.

Fred Cohen noted in 1984 that it is impossible to detect an intrusion in every case and that the resources needed to detect intrusions grows with the amount of usage.

Dorothy E. Denning, assisted by Peter G. Neumann, published a model of an IDS in 1986 that formed the basis for many systems today. Her model used statistics for anomaly detection, and resulted in an early IDS at SRI International named the Intrusion Detection Expert System (IDES), which ran on Sun workstations and could consider both user and network level data. IDES had a dual approach with a rule-based Expert System to detect known types of intrusions plus a statistical anomaly detection component based on profiles of users, host systems, and target systems. Lunt proposed adding an Artificial neural network as a third component. She said all three components could then report to a resolver. SRI followed IDES in 1993 with the Next-generation Intrusion Detection Expert System (NIDES).

The Multics intrusion detection and alerting system (MIDAS), an expert system using P-BEST and Lisp, was developed in 1988 based on the work of Denning and Neumann. Haystack was also developed this year using statistics to reduce audit trails.

Wisdom & Sense (W&S) was a statistics-based anomaly detector developed in 1989 at the Los Alamos National Laboratory. W&S created rules based on statistical analysis, and then used those rules for anomaly detection.

In 1990, the Time-based Inductive Machine (TIM) did anomaly detection using inductive learning of sequential user patterns in Common Lisp on a VAX 3500 computer. The Network Security Monitor (NSM) performed masking on access matrices for anomaly detection on a Sun-3/50 workstation. The Information Security Officer's Assistant (ISOA) was a 1990 prototype that considered a variety of strategies including statistics, a profile checker, and an expert system. ComputerWatch at AT&T Bell Labs used statistics and rules for audit data reduction and intrusion detection.

Then, in 1991, researchers at the University of California, Davis created a prototype Distributed Intrusion Detection System (DIDS), which was also an expert system. The Network Anomaly Detection and Intrusion Reporter (NADIR), also in 1991, was a prototype IDS developed at the Los Alamos National Laboratory's Integrated Computing

Network (ICN), and was heavily influenced by the work of Denning and Lunt. NADIR used a statistics-based anomaly detector and an expert system.

The Lawrence Berkeley National Laboratory announced Bro in 1998, which used its own rule language for packet analysis from libpcap data. Network Flight Recorder (NFR) in 1999 also used libpcap. APE was developed as a packet sniffer, also using libpcap, in November, 1998, and was renamed Snort one month later, and has since become the world's largest used IDS/IPS system with over 300,000 active users.

The Audit Data Analysis and Mining (ADAM) IDS in 2001 used tcpdump to build profiles of rules for classifications.

In 2003, Dr. Yongguang Zhang and Dr. Wenke Lee argue for the importance of IDS in networks with mobile nodes.

Chapter 3

Computer Performance and Network Management

Computer performance

Computer performance is characterized by the amount of useful work accomplished by a computer system compared to the time and resources used.

Depending on the context, good computer performance may involve one or more of the following:

- Short response time for a given piece of work
- High throughput (rate of processing work)
- Low utilization of computing resource(s)
- High availability of the computing system or application
- Fast (or highly compact) data compression and decompression
- High bandwidth / short data transmission time

Performance metrics

Computer performance metrics include availability, response time, channel capacity, latency, completion time, service time, bandwidth, throughput, relative efficiency, scalability, performance per watt, compression ratio, instruction path length and speed up. CPU benchmarks are available.

Aspect of software quality

Computer software performance, particularly software application response time, is an aspect of software quality that is important in human–computer interactions.

Technical and non-technical definitions

The performance of any computer system can be evaluated in measurable, technical terms, using one or more of the metrics listed above. This way the performance can be

- compared relative to other systems or the same system before/after changes
- defined in absolute terms, e.g. for fulfilling a contractual obligation

Whilst the above definition relates to a scientific, technical approach, the following definition given by Arnold Allen would be useful for a non-technical audience:

The word performance *in computer performance means the same thing that performance means in other contexts, that is, it means "How well is the computer doing the work it is supposed to do?"*

Technical performance metrics

There is a wide variety of technical performance metrics that indirectly affect overall computer performance.

Because there are too many programs to test a CPU's speed on all of them, benchmarks were developed. The most famous benchmarks are the SPECint and SPECfp benchmarks developed by Standard Performance Evaluation Corporation and the ConsumerMark benchmark developed by the Embedded Microprocessor Benchmark Consortium EEMBC.

Some important measurements include:

- Instructions per second – Most consumers pick a computer architecture (normally Intel IA32 architecture) to be able to run a large base of pre-existing, pre-compiled software. Being relatively uninformed on computer benchmarks, some of them pick a particular CPU based on operating frequency.
- FLOPS – The number of floating-point operations per second is often important in selecting computers for scientific computations.
- Performance per watt – System designers building parallel computers, such as Google, pick CPUs based on their speed per watt of power, because the cost of powering the CPU outweighs the cost of the CPU itself.
- Some system designers building parallel computers pick CPUs based on the speed per dollar.
- System designers building real-time computing systems want to guarantee worst-case response. That is easier to do when the CPU has low interrupt latency and when it has deterministic response. (DSP)
- Computer programmers who program directly in assembly language want a CPU to support a full-featured instruction set.
- Low power – For systems with limited power sources (e.g. solar, batteries, human power).
- Small size or low weight - for portable embedded systems, systems for spacecraft.
- Environmental impact – Minimizing environmental impact of computers during manufacturing and recycling as well as during use. Reducing waste, reducing hazardous materials.
- Giga-updates per second - a measure of how frequently the RAM can be updated

Occasionally a CPU designer can find a way to make a CPU with better overall performance by improving one of these technical performance metrics without sacrificing any other (relevant) technical performance metric—for example, building the CPU out of better, faster transistors. However, sometimes pushing one technical performance metric to an extreme leads to a CPU with worse overall performance, because other important technical performance metrics were sacrificed to get one impressive-looking number—for example, the megahertz myth.

The total amount of time (**t**) required to execute a particular benchmark program is

$$t = N * C / f$$

where

- **N** is the number of instructions actually executed (the instruction path length). The code density of the instruction set strongly affects N. The value of N can either be determined **exactly** by using an instruction set simulator (if available) or by estimation—itself based partly on estimated or actual frequency distribution of input variables and by examining generated machine code from an HLL compiler. It cannot be determined from the number of lines of HLL source code. N is not affected by other processes running on the same processor. The significant point here is that hardware normally does not keep track of (or at least make easily available) a value of N for executed programs. The value can therefore only be accurately determined by instruction set simulation, which is rarely practiced.

- **f** is the clock frequency in cycles per second.

- **C** is the average cycles per instruction (CPI) for this benchmark.

Even on one machine, a different compiler or the same compiler with different compiler optimization switches can change N and CPI—the benchmark executes faster if the new compiler can improve N or C without making the other worse, but often there is a trade-off between them—is it better, for example, to use a few complicated instructions that take a long time to execute, or to use instructions that execute very quickly, although it takes more of them to execute the benchmark?

A CPU designer is often required to implement a particular instruction set, and so cannot change N. Sometimes a designer focuses on improving performance by making significant improvements in f (with techniques such as deeper pipelines and faster caches), while (hopefully) not sacrificing too much C—leading to a speed-demon CPU design. Sometimes a designer focuses on improving performance by making significant improvements in CPI (with techniques such as out-of-order execution, superscalar CPUs, larger caches, caches with improved hit rates, improved branch prediction, speculative execution, etc), while (hopefully) not sacrificing too much clock frequency—leading to a brainiac CPU design.

Network management

Network management refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems.

- Operation deals with keeping the network (and the services that the network provides) up and running smoothly. It includes monitoring the network to spot problems as soon as possible, ideally before users are affected.
- Administration deals with keeping track of resources in the network and how they are assigned. It includes all the "housekeeping" that is necessary to keep the network under control.
- Maintenance is concerned with performing repairs and upgrades—for example, when equipment must be replaced, when a router needs a patch for an operating system image, when a new switch is added to a network. Maintenance also involves corrective and preventive measures to make the managed network run "better", such as adjusting device configuration parameters.
- Provisioning is concerned with configuring resources in the network to support a given service. For example, this might include setting up the network so that a new customer can receive voice service.

A common way of characterizing network management functions is FCAPS—Fault, Configuration, Accounting, Performance and Security.

Functions that are performed as part of network management accordingly include controlling, planning, allocating, deploying, coordinating, and monitoring the resources of a network, network planning, frequency allocation, predetermined traffic routing to support load balancing, cryptographic key distribution authorization, configuration management, fault management, security management, performance management, bandwidth management, Route analytics and accounting management.

Data for network management is collected through several mechanisms, including agents installed on infrastructure, synthetic monitoring that simulates transactions, logs of activity, sniffers and real user monitoring. In the past network management mainly consisted of monitoring whether devices were up or down; today performance management has become a crucial part of the IT team's role which brings about a host of challenges—especially for global organizations.

Note: Network management does not include user terminal equipment.

Technologies

A small number of accessories methods exist to support network and network device management. Access methods include the SNMP, command-line interface (CLIs), custom XML, CMIP, Windows Management Instrumentation (WMI), Transaction Language 1, CORBA, NETCONF, and the Java Management Extensions (JMX).

Schemas include the WBEM, the Common Information Model, and MTOSI amongst others.

Medical Service Providers provide a niche marketing utility for managed service providers; as HIPAA legislation consistently increases demands for knowledgeable providers. Medical Service Providers are liable for the protection of their clients confidential information, including in an electronic realm. This liability creates a significant need for managed service providers who can provide secure infrastructure for transportation of medical data.

Chapter 4

Automounter

An **automounter** is any program or software facility which automatically mounts filesystems in response to access operations by user programs. An automounter system utility (daemon under Unix), when notified of file and directory access attempts under selectively monitored subdirectory trees, dynamically and transparently makes local or remote devices accessible.

The automounter has the purpose of conserving local system resources and of reducing the coupling between systems which share filesystems with a number of servers. For example, a large to mid-sized organization might have hundreds of file servers and thousands of workstations or other nodes accessing files from any number of those servers at any time. Usually, only a relatively small number of remote filesystems (*exports*) will be active on any given node at any given time. Deferring the mounting of such a filesystem until a process actually needs to access it reduces the need to track such mounts, increasing reliability, flexibility and performance.

Frequently, one or more fileservers will become inaccessible (down for maintenance, on a remote and temporarily disconnected network, or accessed via a congested link). Administrators also often find it necessary to relocate data from one file server to another - to resolve capacity issues and balance the load. Having data mount-points automated makes it easier to reconfigure client systems in such cases. In addition, end-users should only have the ability to mount some removable media devices - such as floppies, CD-ROMs, and USB keys - when the device is attached to the system.

These factors combine to pose challenges to older "static" management methods of filesystem mount tables (the `fstab` files on Unix systems). Automounter utilities address these challenges and allow sysadmins to consolidate and centralize the associations of mountpoints (directory names) to the exports. When done properly, users can transparently access files and directories as if all of their workstations and other nodes attach to a single enterprise-wide filesystem.

One can also use automounters to define multiple repositories for read-only data; client systems can automatically choose which repository to mount based on availability, file-server load, or proximity on the network.

Home directories

Many establishments will have a number of file servers which host the home directories of various users. All workstations and other nodes internal to such organizations (typically all those behind a common firewall separating them from the Internet) will be configured with automounter services so that any user logging into any node implicitly triggers access to his or her own home directory which, consequently, is mounted at a common mountpoint, such as `/home/user`. This allows users to access their own files from anywhere in the enterprise, which is extremely useful in Unix environments, where users may frequently invoke commands on many remote systems via various job-dispatching commands such as `ssh`, `telnet`, `rsh` or `rlogin`, or via the X11 or VNC protocols.

/net

A very common default automounter local path is of the form `/net/hostname/nfspath` where `hostname` is the host name of the remote machine and `nfspath` is the path that is exported over NFS on the remote machine. This notation generally frees the system manager from having to manage each exported path explicitly via a central automounter map.

Software shares and repositories

In some computing environments, user workstations and computing nodes do not host installations of the full range of software that users might want to access. Systems may be "imaged" with a minimal or typical cross-section of the most commonly used software. Also, in some environments, users might require specialized or occasional access to older versions of software (for instance, developers may need to perform bug fixes and regression testing, or some users may need access to archived data using outdated tools).

Commonly, organizations will provide repositories or "depots" of such software, ready for installation as required. These also may include full copies of the system images from which machines have their operating systems initially installed, or available for repair of any system files that may get corrupted during a machine's lifecycle.

Some software may require quite substantial storage space or might be undergoing rapid (perhaps internal) development. In those cases the software may be installed on, and configured to be run directly from, the file servers.

Dynamically variant automounts

In the simplest case, a fileserver houses data and perhaps scripts which can be accessed by any system in an environment. However, certain types of files (executable binaries and shared libraries, in particular) can only be used by specific types of hardware or specific versions of specific operating systems.

For situations like this, automounter utilities generally support some means of "mapping" or "interpolating" variable data into the mount arguments.

For example, an organization with a mixture of Linux and Solaris systems might arrange to host their software package repositories for each on a common file-server using export names like `depot:/export/linux` and `depot:/export/solaris` respectively. Thereunder they might have directories for each of the OS versions that they support. Using the dynamic variation features in their automounter, they might then configure all their systems so that any administrator on any machine in their enterprise could access available software updates under `/software/updates`. A user on a Solaris system would find the Solaris compiled packages under `/software`, while a Red Hat or CentOS (Linux) user would find RPMs for their particular OS version thereunder. Moreover, a Solaris user on a SPARC workstation would have his `/software/updates` mapped to an appropriate export for that system's architecture, while a Solaris user on an x86 PC would transparently find his `/software/updates` directory containing packages suited to his system. Some software (written in scripting languages such as Perl or Python) can be installed and/or run on any supported platform without porting, recompilation or re-packaging of any sort. A systems administrator might conceivably locate such software in a `/software/common` export.

In some cases, organizations may also use regional or location-based variable/dynamic mappings — so that users in one building or site are directed to a closer file server which hosts replications of the resources that are hosted at other locations.

In all of these cases, automounter utilities allow the users to access files and directories without regard for the actual physical location. Using an automounter, the users and systems administrators can usually access files where they are "supposed to be" and find that they appear to be there.

Software

Tom Lyon developed the original automount software at Sun Microsystems: SunOS 4.0 made automounting available in 1988. Sun Microsystems eventually licensed this implementation to other commercial UNIX distributions. Solaris 2.0, first released in 1992, implemented its automounter with a pseudofilesystem called `autofs`, which communicates with a user-mode daemon that performs mounts. Other Unix-like systems have adopted that implementation of the automounter - including AIX, HP-UX, and Mac OS X 10.5 and later.

Linux has an independent implementation of an autofs-based automounter; version 5 of that automounter generally operates compatibly with the Solaris automounter.

In December 1989 Jan-Simon Pendry released *amd*, an automounter "based in spirit" on the SunOS automount program. *amd* has also become known as the Berkeley Automounter.

Some operating systems also support automatic mounting of external drives (such as disk drives or flash drives that use FireWire or USB connections) and removable media (such as CDs and DVDs). This technology differs from the automounting described here; it involves mounting local media when the user attaches them to or inserts them into the system, rather than mounting directories from remote file servers when a reference is made to them. Linux currently (as of Linux 2.6) uses the user-space program udev for this form of automounting. Some automounting functions have been implemented in the separate program HAL, but As of 2010 are being merged into udev. OpenBSD has hotplugd(8) which triggers special scripts on attach or detach of removable devices, so that user can easily add mounting of removable drives. In Mac OS X `diskarbitrationd` carries out this form of automatic mounting.

Disadvantages and caveats

While automounter utilities (and remote filesystems in general) can provide centrally managed, consistent and largely transparent access to an organization's storage services, they also can have their downsides:

- Access to automounted directories can trigger delays while the automounter resolves the mapping and mounts the export into place.
- Timeouts can cause the unmounting of mounted directories (which situation can later result in mount delays upon the next attempted access).
- The mapping of mountpoint to export arguments is usually done via some directory service such as LDAP or NIS, which constitutes another dependency (potential point of failure).
- When some systems require frequent access to some resources, while others only need occasional access, this can pose difficult or impossible problems in implementing a consistent, enterprise-wide mixture of locally "mirrored" (replicated) and automounted directories.
- When data is migrated from one file server (export) to another, there can be an indeterminate number of systems which, for various reasons, still have an active mount on the old location ("stale NFS mounts"); these can cause issues which may even necessitate the reboot of otherwise perfectly stable hosts.
- Organizations can find that they've created a "spaghetti" of mappings which can entail considerable management overhead and sometimes quite a bit of confusion among users and administrators.
- Users can become so accustomed to the transparency of automounted resources that they neglect to consider some of the differences in access semantics that may apply to networked filesystems, as compared to locally mounted devices. In particular, programmers may attempt to use "locking" techniques which are safe and provide the desired atomicity guarantees on local filesystems, but which are documented as inherently vulnerable to race conditions when used on NFS.

Chapter 5

Application Performance Management and Business Transaction Management

Application performance management

Application performance management, or **APM**, refers to the discipline within systems management that focuses on monitoring and managing the performance and service availability of software applications.

APM can be defined as process and use of related IT tools to detect, diagnose, remedy and report application’s performance to ensure that it meets or exceeds end-users’ and businesses’ expectations. Application performance relates to how fast transactions are completed on behalf of, or information is delivered to the end user by the application via a particular network, application and/or web services infrastructure.

Methods for Measuring Performance

There are two main methods by which applications performance is assessed for production applications. The first is measuring the resources used by the application, has been in use since computers have been used for business applications, and is still in use. The second is measuring the response time of applications from the perspective of the end user.

Application performance management is related to end-user experience management and real user management in that measuring the experience of real users in the use of an application in production is considered by many as being the most valid method of assessing the performance of an application in production.

Platforms

The use of application performance management is common for web applications written to JEE and Microsoft .NET platforms. All of the leading systems management vendors have JEE and .NET APM products in their portfolios. These APM for JEE and .NET based applications have the advantage of being able to measure response time from the perspective of the web server, and being able to provide root cause analysis for the likely causes of performance issues within the applications code executing in the JEE or .NET environment. Many of these products also have connectors that monitor the transaction flow from the business logic layer of the application to the database server, or to external interfaces like web services. Some of these vendors also have HTTP appliances in their product line that can decode transaction specific response times at the web server layer.

Dependency injection software development frameworks on JEE instrument an application to provide performance metrics automatically. For example, Spring-based JEE applications support management protocols to provide observed issues in application operation to a performance management tool/dashboard. SpringSource acquired APM-player Hyperic in 2009 to combine application development, automatic application instrumentation, and application performance management. Aspect Oriented Programming on JEE platforms enables automatic performance monitoring without instrumentation of the application. PushToTest TestMaker is an open source load testing solution that integrates with Glassbox, an open source application performance monitoring and troubleshooter application.

Current Issues

The difficult issues in APM currently revolve around two trends in the IT industry. The first is that for many enterprises, only a small fraction of their business critical applications are web based and written to JEE or .NET. For these enterprises who may have business critical applications like SAP that use "fat" Win32 clients, their APM need can only be met by engaging with vendors offering deep End User Experience monitoring for a specific set of enterprise applications. The second issue is that many applications systems are being virtualized, which has the effect of breaking the validity of time based performance metrics gathered within the guest OS where the application is running. This requires a totally new approach to APM tuned to the requirements of virtualized systems.

Five Functional Dimensions of APM

According to Gartner research, Application Performance Management includes 5 distinct functional dimensions:

- End-user experience monitoring
- Application runtime architecture discovery and modeling
- User-defined transaction profiling (Also called Business Transaction Management)
- Application component deep-dive monitoring
- Application data analytics

Business transaction management

Business transaction management (BTM), also known as **business transaction monitoring**, **application transaction profiling** or **user defined transaction profiling**, is the practice of managing information technology (IT) from a business transaction perspective. It provides a tool for tracking the flow of transactions across IT infrastructure, in addition to detection, alerting, and correction of unexpected changes in business or technical conditions. BTM provides visibility into the flow of transactions across infrastructure tiers, including a dynamic mapping of the application topology.

Using BTM, application support teams are able to search for transactions based on message context and content – for instance, time of arrival or message type – providing a way to isolate causes for common issues such as application exceptions, stalled transactions, and lower-level issues such as incorrect data values.

The ultimate goal of BTM is to improve service quality for users conducting business transactions while improving the effectiveness of the IT applications and infrastructure across which those transactions execute. The main benefit of BTM is its capacity to identify precisely where transactions are delayed within the IT infrastructure. BTM also aims to provide proactive problem prevention and the generation of business service intelligence for optimization of resource provisioning and virtualization.

A number of factors have led to the demand for the development of BTM software:

- Modern applications have become more complex, modular, distributed, interdependent and sensitive to environmental conditions.
- IT infrastructure has become a complex multi-tier environment.
- The rise of service-oriented architecture in systems development .
- The proliferation of service level agreements.

Applications

BTM solutions capture all of the transaction instances in the production environment and as such can be used for monitoring as well as for analysis and planning. Some applications include:

- Outage avoidance and problem isolation: Identification and isolation of tier-specific performance and availability issues.
- Service level management: Monitoring of SLAs and alerting of threshold breaches both at the end-user and infrastructure tier level.
- Infrastructure optimization: Modification of the configuration of data center infrastructure to maximize utilization and improve performance.
- Capacity planning: Analysis of usage and performance trends in order to estimate future capacity requirements.
- Change management: Analysis of the impact of change on transaction execution.

- Cloud management: Track the end-to-end transaction flow across both cloud (private, hybrid, public) and dedicated (on-premise, off-premise) infrastructure.

Transaction discovery methods

BTM systems track each of the hops in the transaction path using a variety of data collection methods including OS-level sockets, network packet sniffing, log parsing, agent-based middleware protocol sniffing, and others.

Relationship to application performance management

BTM is sometimes categorized as a form of application performance monitoring or management. It works alongside other IT monitoring systems including End-User Experience Monitoring, Synthetic Transaction Monitoring, Deep-Dive Monitoring and Business Activity Monitoring (BAM) solutions. According to Gartner, BTM and deep dive monitoring are "fundamentally distinct and their associated processes are typically carried out by different communities with different skill sets… The buyer should still implement multiple products, even if it means greater architectural complexity and apparent functional overlap."

Relationship to virtualization and cloud computing

BTM dynamically maps the execution of a user transaction as it traverses the data center. In both virtualized and cloud environments, the relationship between the application and infrastructure is to some degree dynamically allocated or defined. BTM discovers the infrastructure currently executing each transaction instance for purposes of problem identification, resolution, and infrastructure tuning. In public and hybrid cloud architectures, BTM has the ability to profile transactions from the datacenter, to the cloud provider, and back.

Chapter 6

Deep Freeze (Software) and Multiseat Configuration

Deep Freeze (software)

Deep Freeze, by Faronics, is an application available for the Microsoft Windows, Mac OS X, and SUSE Linux operating systems which allows system administrators to protect the core operating system and configuration files on a workstation or server by restoring a computer back to its original configuration each time the computer restarts.

Operation

Deep Freeze is a kernel-level driver that protects hard drive integrity by redirecting information being written to the hard drive or partition, leaving the original data intact. This redirected information is no longer referenced once the computer is restarted, thus restoring the system to its original state at the disk sector level. This allows users to make 'virtual' changes to the system, giving them the appearance that they can modify core files or even delete them, and even make the system unusable to themselves, but upon reboot the originally configured 'frozen' state of the operating system is restored.

To make changes, a system administrator must 'thaw' the protected partition by disabling Deep Freeze, make any needed changes, and then 'freeze' it again by re-enabling Deep Freeze. These changes become part of the protected partition and will be maintained after restarts. 'Freezing' and 'thawing' can be done at the workstation level or remotely via either the Faronics Core management platform or the Deep Freeze Enterprise Console. Users of the Enterprise version can also create virtual partitions called ThawSpaces (of up to 1 TB on an NTFS-formatted drive) to retain data on "frozen" hard drives after restarts.

Deep Freeze can also protect a computer from harmful malware as it automatically deletes (or rather, no longer 'sees') downloaded files when the computer is restarted. The advantage to using Deep Freeze as an antivirus/antimalware application is that it uses almost no system resources, and does not slow down the computer noticeably. The disadvantage is that it does not provide real-time protection, therefore an infected computer would have to be restarted in order to remove malware.

Limitations and security

Deep Freeze only protects workstations in a "fresh-booted" state. That is, Deep Freeze prevents permanent tampering with protected hard drives/partitions across reboots, but user activity between restarts is not limited by the program. For example, Deep Freeze does not prevent application installation; a user can install a modified version of a Web browser (but seemingly harmless to the unknowing user) designed to secretly send users' passwords to a server connected to the Internet. As a workaround, Deep Freeze can be configured to restart after user logout, shutdown after a chosen period of inactivity, or restart/shutdown at a scheduled time in an attempt to ensure that no such installations are retained (as rebooting the system returns the system to its original, unmodified state).

Deep Freeze cannot protect the operating system and hard drive upon which it is installed if the computer is booted from another medium (such as an external hard drive, a USB device, optical media, or network server). In such cases, a user would have real access to the contents of the (supposedly) frozen system. On a Windows-based computer, this scenario may be prevented by configuring the CMOS (nonvolatile BIOS memory) on the workstation to boot only to the hard drive to be protected, then password protecting the CMOS. This is a normal precaution for most public access computers. A further precaution would be to lock the PC case shut with a physical lock or tiedown cable system to prevent access to motherboard jumpers.

Deep Freeze can only protect hard drive partitions of up to a 2 TB capacity (using NTFS).

Competitors

There are sandboxing and virtualization products which have similar features to what Deep Freeze offers but do not employ the same redirection process. These include:

- Rollback Rx
- Clean Slate (Fortres Grand)
- HDGUARD
- Returnil Virtual System (Returnil)
- Sandboxie (Ronen Tzur)
- Shadow Defender (ShadowDefender.Net)
- SmartShield (Centurion Technologies)
- Windows SteadyState (Microsoft)
- System Revert

Multiseat configuration

A four-head multiterminal.

A **multiseat**, **multi-station** or **multiterminal** configuration is a single computer which supports multiple independent users at the same time. In modern usage the terms refer to multiple users using one personal computer, each with their own console, consisting of a keyboard a mouse, a monitor, and possibly headphones.

Motivation

With the increasing capacity of processors and memory, commodity personal computers can now perform significant numbers of tasks simultaneously without slowing down. However, using standard computer configurations, only one user is able to use the computer at a time, limiting the effectiveness of the system as it remains idle most of the time. With a multiterminal, a lot of users can share the same computer, so more of its total capacity is going to be used. For example, if someone is just using a web browser or word processor, no one else can use the computer and 90% of the system's resources may be idle - but with multiterminals, other people will be able to use the otherwise idle resources. However, if someone is using all of the system's resources (playing a resource-intensive computer game, for example) the other users will have a very slow system.

Multiseats are also more cost-effective: it is not necessary to buy separate motherboards, microprocessors, RAM, hard disks and other components for each user. For example, buying one high speed CPU usually costs less than buying several slower CPUs.

History

In the 1970s, it was very commonplace to connect multiple computer terminals to a single mainframe computer, even graphical terminals. Early terminals were connected with RS-232 type serial connections, either directly, or through modems. With the advent of Internet Protocol based networking, it became possible for multiple users to log in to a host using telnet or – for a graphic environment – a X Window "server". These systems would retain a physically secure "root console" for system administration and direct access to the host machine.

Support for multiple consoles in a PC running the X Window interface was implemented in 2001 by Miguel Freitas, using the Linux operating system and the X11 graphical system (in that age maintained by XFree86). This was done using a patch in the X server to execute several instances of X at the same time such that each one captures specific mouse and keyboard events and the graphical content. This method received the name of multiseat or multiterminal.

In 2002 a Canadian company, Userful Corporation, released Userful Multiplier, a multiseat Linux software solution that enables up to 10 users to simultaneously share one computer. Earlier they worked on a kernel-based approach to a multi-station platform computer, but abandoned the idea due to a problem with multiple video card support.

Other solutions appeared in 2003, such Svetoslav Slavtchev, Aivils Stoss and James Simmons worked, with the evdev and Faketty approach modifying the kernel Linux and letting more than one user independently use the same machine. In that time, the Linux Console Project also proposed an idea to use multiple independent consoles and then multiple independent keyboards and mice in a project called "Backstreet Ruby". Backstreet Ruby is a kernel patch for the Linux kernel. It is a back port to Linux-2.4 of the Ruby kernel tree. The aim of the Linux Console developers is to enhance and reorganize the input, the console and the framebuffer subsystems in the Linux kernel, so they can work independent from each other and to allow multi-desktop operation. The Backstreet Ruby idea was never finished.

In 2005, the team of C3SL (Center for Scientific Computing and Free Software), from Federal University of Parana in Brazil, created the solution based with nested X servers, such Xnest and Xephyr. With this solution, each nested X server runs in each screen of a host X server (e.g. Xorg) and a modification in the nested servers let it get the exclusivity of each set of mouse and keyboard. In 2008, the C3SL group releases the Multiseat Display Manager (MDM) to ease the process of installation and configuration of a multiseat box. This group, also in 2008, conceived a live-CD for tests purposes.

Multiseat was a planned feature for Fedora 12 but did not materialize and is currently pending.

Time line, commercial multiseat software evolution

- 1996, ThinSoft/BeTwin
- 2002, Userful Corporation
- 2004, Open-Sense Solutions (Groovix)
- 2006, NComputing
- 2010, Microsoft
- 2011, Black Box VirtuaCore

Requirements

Hardware requirements

Each monitor will need to be connected to a graphics output from a video card. For example, to make a *four-head* (four users), would require four monitors, four keyboards, four mice and two dual or one quad output video card (five users can be accommodated if onboard video off the motherboard is available). USB keyboards and mice are typically recommended instead of PS/2 connections, as they can be connected to a USB hub, or USB/audio hub. Additional devices and peripherals such as cameras, flash storage drives, card readers, touch screens, etc. could also be assigned to each seat. An alternative to multiple physical video cards and connections is DisplayLink over USB.

Windows

For Windows 2000, XP and Vista operating systems, there are several commercial products to implement multiseat configurations for two or more seats.

Windows MultiPoint Server 2010 was announced on February 24, 2010. It uses Remote Desktop (Terminal Services) technologies in Windows Server 2008 R2.

Case studies

Paraná Digital project

One of multiterminal's successful cases is happening at Paraná Digital project. It is creating multiterminal laboratories on 2000 public schools of the state of Paraná (Brazil). More than 1.5 million users will benefit from the 40,000 terminals when the project is finished. The laboratories have four-head multiterminals running Debian. The cost of all the hardware is 50% less than the normal price, and there is absolutely no cost with software. This project developer is C3SL (Center for Scientific Computing and Free Software).

Michigan State University Research in Tanzania

Since 2008, electrical and computer engineering students from Michigan State University have installed multiterminal systems with internet access in three schools in Mto wa Mbu, Tanzania. The purpose of the project is to study the impact of having computer systems with internet access in an education system that cannot afford other educational resources such as books. The computer systems run Ubuntu 8.04 32-bit and utilize the open source Multiseat Display Manager created by C3SL. The research will eventually be used to present to government officials of third world countries in effort to showcase the positive impact of having cost-effective computing systems in schools. The project is sponsored by George and Vickie Rock and the Dow Chemical Company.

Notable installations

- Userful announced a deployment of 356,800 Linux-based virtual desktops in Brazil (February 2009)
- NComputing provided 180,000 one to one computing seats for K–12 students in the country of Macedonia

Chapter 7

Intelligent Platform Management Interface

The **Intelligent Platform Management Interface** (IPMI) is a standardized computer system interface used by system administrators to manage a computer system and monitor its operation.

The development of this interface specification was led by Intel Corporation and is supported by more than two hundred computer systems vendors.. Dell, Hewlett-Packard, Intel, and NEC Corporation announced IPMI v1.0 on 1998-09-16, v1.5 on 2001-03-01, and v2.0 on 2004-02-14. The technology is now considered a de-facto standard for computer system management.

Functionality

An IPMI sub-system operates independently of the operating system and allows administrators to manage a system remotely in the absence of an operating system or of the system management software. The monitored system may be powered off, but must be connected to a power source and the monitoring medium, typically a local area network connection. IPMI can also function after the operating system has started, and exposes management data and structures to the system management software. IPMI prescribes only the structure and format of the interfaces as a standard, while detailed implementations may vary.

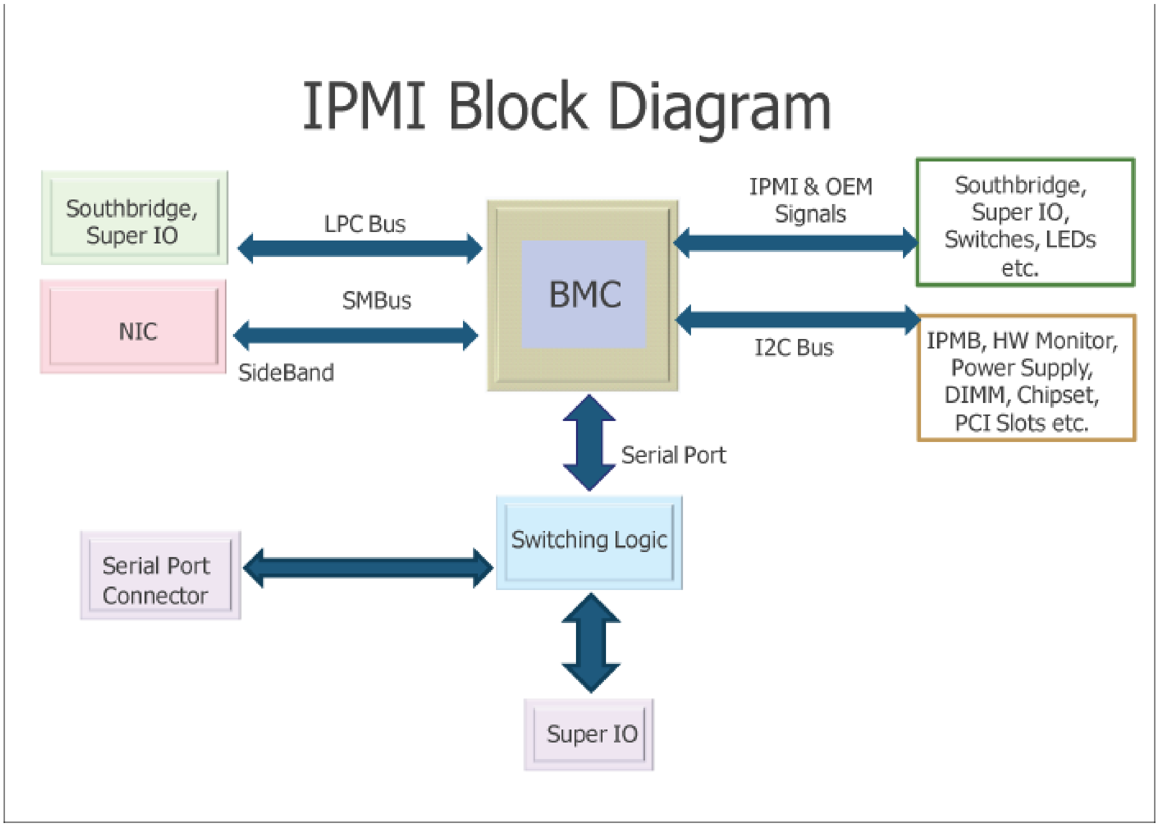
An implementation of IPMI version 1.5 can send out alerts via a direct serial connection or via a side-band local area network (LAN) connection to a remote client. The side-band LAN connection utilizes the board network interface controller (NIC). This solution is less expensive than a dedicated LAN connection but also has limited bandwidth. Systems compliant with IPMI version 2.0 can also send alerts via serial over LAN. System administrators can then use IPMI messaging to query platform status, to review hardware logs, or to issue other requests from a remote console through the same connections. The standard also defines an alerting mechanism for the system to send a simple network management protocol (SNMP) platform event trap (PET).

Side-band and out-of-band

IPMI implements what is often called a side-band management LAN connection. This connection utilizes a System Management Bus (SMBUS) interface between the BMC (Baseboard Management Controller) and the board Network Interface Controller (NIC). This solution has the advantage of reduced costs but also provides limited bandwidth – sufficient for text console redirection but not for video redirection. In other words, when a remote computer is down the system administrator can access it through IPMI and utilize a text console. This will be sufficient for a few vital functions, such as checking the event log, accessing the BIOS setup and perform power on, power off or power cycle. However, more advanced functions, such as remote re-installation of an operating system, may require an out-of-band management approach utilizing a dedicated LAN Connection.

IPMI components

An IPMI sub-system consists of a main controller, called the baseboard management controller (BMC) and other management controllers distributed among different system modules that are referred to as satellite controllers. The satellite controllers within the same chassis connect to the BMC via the system interface called Intelligent Platform Management Bus/Bridge (IPMB) — an enhanced implementation of I²C (Inter-Integrated Circuit). The BMC connects to satellite controllers or another BMC in another chassis via the Intelligent Platform Management Chassis (IPMC) bus or bridge. It may be managed with the Remote Management Control Protocol (RMCP), a specialized wire protocol defined by this specification.



Interfaces to the baseboard management controller

Several vendors develop and market BMC chips. A BMC utilized for embedded applications may have limited memory and require optimized firmware code for implementation of the full IPMI functionality. Highly integrated BMCs can provide complex instructions and provide the complete out-of-band functionality of a service processor. The firmware implementing the IPMI interfaces is provided by various vendors. A field replaceable unit (FRU) holds the inventory, such as vendor ID and manufacturer, of potentially replaceable devices. A sensor data record (SDR) repository provides the properties of the individual sensors present on the board. For example, the board may contain sensors for temperature, fan speed, and voltage.

Baseboard management controller

The *baseboard management controller* is the intelligence in the IPMI architecture. It is a specialized microcontroller embedded on the motherboard of a computer, generally a server. The BMC manages the interface between system management software and platform hardware.

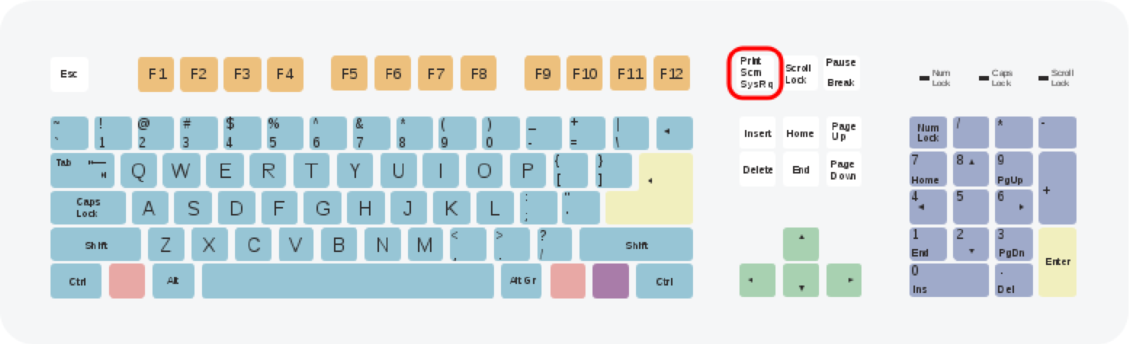
Different types of sensors built into the computer system report to the BMC on parameters such as temperature, cooling fan speeds, power status, operating system (OS) status, etc. The BMC monitors the sensors and can send alerts to a system administrator via the network if any of the parameters do not stay within preset limits, indicating a potential failure of the system. The administrator can also remotely communicate with the BMC to take some corrective action such as resetting or power cycling the system to get a hung OS running again. These abilities save on the total cost of ownership of a system.

Physical interfaces to the BMC include SMBus busses, an RS-232 serial console, address and data lines and an Intelligent Platform Management Bus (IPMB), that enables the BMC to accept IPMI request messages from other management controllers in the system.

A direct serial connection to the BMC is not encrypted as the connection itself is secure. Connection to the BMC over LAN may or may not use encryption depending on the security concerns of the user.

Chapter 8

Magic SysRq Key



The SysRq key

The **magic SysRq key** is a key combination understood by the Linux kernel, which allows the user to perform various low level commands regardless of the system's state. It is often used to recover from freezes, or to reboot a computer without corrupting the filesystem.

To be able to use this functionality the `CONFIG_MAGIC_SYSRQ` option has to be enabled at kernel compile time.

Purpose

Much like Sun Microsystems's Open Firmware (OpenBoot), this key combination provides access to powerful tools for software development and disaster recovery. In this sense, it can be considered a form of escape sequence. Principal among the offered commands are means to forcibly unmount file systems, kill processes, recover keyboard state, and write unwritten data to disk. With respect to these tasks, this feature serves as a tool of last resort.

Magic commands

The key combination consists of Alt, SysRq and another key, which controls the command issued (as shown in the table below). Users with a keyboard layout other than QWERTY have to remember that their layout becomes QWERTY when they use one of these combinations. For example, on a Dvorak keyboard, the key below '9' and '0' counts as an 'o', not as an 'r', so it shuts the system down instead of switching the keyboard to raw mode. Furthermore, some keyboards may not provide a separate SysRq key. In this case, a separate "Print Screen" key should be present. Under graphical environments (such as Gnome or KDE) 'Alt'+'PrintScrn/SysRq'+key combination generally only leads to a screenshot being dumped. To avoid this Print Screen feature the magic SysRq combination should include the Ctrl, becoming 'Ctrl'+'Alt'+'SysRq'+key. For the same purposes the AltGr key, if present, can be used in place of the Alt key. The magic SysRq can also be accessed from the serial console.

Action	QWERTY	Dvorak	AZERTY
Set the console log level, which controls the types of kernel messages that are output to the console	**0** through **9**	**0** through **9**	**0** through **9** (*without* using shift)
Immediately reboot the system, without unmounting partitions or syncing	**b**	**x**	**b**
Reboot kexec and output a crashdump	**c**	**j**	**c**
Display all currently held Locks	**d**	**e**	**d**
Send the SIGTERM signal to all processes except init (PID 1)	**e**	**.**	**e**
Call oom_kill, which kills a process to alleviate an OOM condition	**f**	**u**	**f**
When using Kernel Mode Setting, provides emergency support for switching back to the kernel's framebuffer console	**g**	**i**	**g**
Output a terse help document to the console Any key which is not bound to a command should also perform this action	**h**	**d**	**h**
Send the SIGKILL signal to all processes except init	**i**	**c**	**i**
Kill all processes on the current virtual console (Can be used to kill X and svgalib programs, see below) This was originally designed to imitate a Secure Access Key	**k**	**t**	**k**
Output current memory information to the console	**m**	**m**	**,**
Reset the nice level of all high-priority and real-time tasks	**n**	**b**	**n**
Shut off the system	**o**	**r**	**o**
Output the current registers and flags to the console	**p**	**l**	**p**

Display all active high-resolution timers and clock sources.	**q**	**'**	**a**
Switch the keyboard from raw mode, the mode used by programs such as X11 and svgalib, to XLATE mode	**r**	**p**	**r**
Sync all mounted filesystems	**s**	**o**	**s**
Output a list of current tasks and their information to the console	**t**	**y**	**t**
Remount all mounted filesystems in read-only mode	**u**	**g**	**u**
Output Voyager SMP processor information	**v**	**k**	**v**
Display list of blocked (D state) tasks	**w**	**,**	**z**

Common usage

Command line access and configuration

While this was originally implemented as part of the kernel's keyboard handler for debugging, the functionality has been also exposed via the proc filesystem and is commonly used to provide extended management capabilities to headless and remote systems. As an example, shell script can be simply used:

```
echo b > /proc/sysrq-trigger
```

This is equivalent to the key combination Alt + SysRq + B which reboots the machine.

The feature is controlled both by a compile-time option in the kernel configuration, CONFIG_MAGIC_SYSRQ, and a sysctl kernel parameter, kernel.sysrq.

"Raising Elephants" mnemonic device

A common use of the magic SysRq key is to preform a safe reboot of a Linux computer which has otherwise locked up. This can prevent a fsck being required on reboot and gives some programs a chance to save emergency backups of unsaved work. The QWERTY (or AZERTY) mnemonic "**R**aising **E**lephants **I**s **S**o **U**tterly **B**oring", "**R**eboot **E**ven **I**f **S**ystem **U**tterly **B**roken" or simply remembering the word "BUSIER" backwards, are often used. It stands for:

```
unRaw      (take control of keyboard back from X),
 tErminate (send SIGTERM to all processes, allowing them to terminate
gracefully),
 kIll      (send SIGKILL to all processes, forcing them to terminate
immediately),
  Sync     (flush data to disk),
  Unmount  (remount all filesystems read-only),
reBoot.
```

In practice, each command may require a few seconds to complete, especially if feedback is unavailable from the screen due to a freeze or display corruption.

Remote access

The linux daemon sysrqd provides a method of accessing SysRq features over TCP/IP port 4094 after authenticating with a plain-text password.

Graphical programs

When magic SysRq keys are used to kill a frozen graphical program, the program has no chance to restore text mode. This can make everything unreadable. The commands `textmode` (part of SVGAlib) and `reset` can restore text mode and make the console readable again.

In hypervisors

The Xen hypervisor has functionality to send magic commands to hosted domains via its "xm sysrq" command.

Security concerns

Some people view this key as giving access to dangerous system-level commands to anyone who has physical access to the keyboard or serial console. It has been argued that this perceived security is illusory, as anyone with physical access to the computer would already have the capability to compromise its security. The advent of the procfs interface has rekindled debate over this subject.

Disabling SysRq key

The SysRq key can be disabled with the following command:

```
echo 0 > /proc/sys/kernel/sysrq
```

To re-enable:

```
echo 1 > /proc/sys/kernel/sysrq
```

On newer kernels (exact version unknown), it is possible to have a more fine-grained control. On these machines, the number written to /proc/sys/kernel/sysrq can be zero, one, or a number greater than one which is a bitmap indicating which features to allow.

Possible values are:

- 0 - disable sysrq
- 1 - enable sysrq completely
- >1 - bitmask of enabled sysrq functions:
    - 2 - control of console logging level
    - 4 - control of keyboard (SAK, unraw)
    - 8 - debugging dumps of processes etc.
    - 16 - sync command
    - 32 - remount read-only
    - 64 - signalling of processes (term, kill, oom-kill)
    - 128 - reboot/poweroff
    - 256 - nicing of all RT tasks

Chapter 9

Password Cracking and Remote Administration

Password cracking

Password cracking is the process of recovering passwords from data that has been stored in or transmitted by a computer system. A common approach is to repeatedly try guesses for the password. The purpose of password cracking might be to help a user recover a forgotten password (though installing an entirely new password is less of a security risk, but involves system administration privileges), to gain unauthorized access to a system, or as a preventive measure by system administrators to check for easily crackable passwords. On a file-by-file basis, password cracking is utilized to gain access to digital evidence for which a judge has allowed access but the particular file's access is restricted.

Time needed for password searches

The time to crack a password is related to bit strength, which is a function of the password's information entropy. Most methods of password cracking require the computer to produce many candidate passwords, each of which is checked. Brute force cracking, in which a computer tries *every* possible key or password until it succeeds, is the lowest common denominator of password cracking. More common methods of password cracking, such as dictionary attacks, pattern checking, word list substitution, etc., attempt to reduce the number of trials required and will usually be attempted before brute force.

The ability to crack passwords using computer programs is a function of the number of possible passwords per second which can be checked. If a hash of the target password is available to the attacker, this number can be quite large. If not, the rate depends on whether the authentication software limits how often a password can be tried, either by time delays, CAPTCHAs, or forced lockouts after some number of failed attempts.

Individual desktop computers can test anywhere between one million to fifteen million passwords per second against a password hash for weaker algorithms, such as DES or LanManager. See: John the Ripper benchmarks A user-selected eight-character password with numbers, mixed case, and symbols, reaches an estimated 30-bit strength, according to NIST. 2^{30} is only one billion permutations and would take an average of 16 minutes to crack. When ordinary desktop computers are combined in a cracking effort, as can be done with botnets, the capabilities of password cracking are considerably extended. In 2002, distributed.net successfully found a 64-bit RC5 key in four years, in an effort which included over 300,000 different computers at various times, and which generated an average of over 12 billion keys per second. Graphics processors can speed up password cracking by a factor of 50 to 100 over general purpose computers. As of 2011, commercial products are available that claim the ability to test up to 2,800,000,000 passwords a second on a standard desktop computer using a high-end graphics processor. Such a device can crack a 10 letter single-case password in one day. Note that the work can be distributed over many computers for an additional speedup proportional to the number of available computers with comparable GPUs.

If a cryptographic salt is not used in the password system, the attacker can pre-compute hash values for common passwords variants and for all passwords shorter than a certain length, allowing very rapid recovery. Long lists of pre-computed password hashes can be efficiently stored rainbow tables. Such tables are available on the Internet for several common password authentication systems.

Another situation where quick guessing is possible is when the password is used to form a cryptographic key. In such cases, an attacker can quickly check to see if a guessed password successfully decodes encrypted data. For example, one commercial product claims to test 103,000 WPA PSK passwords per second.

Despite their capabilities, desktop CPUs are slower at cracking passwords than purpose-built password breaking machines. In 1998, the Electronic Frontier Foundation (EFF) built a dedicated password cracker using FPGAs, as opposed to general purpose CPUs. Their machine, Deep Crack, broke a DES 56-bit key in 56 hours, testing over 90 billion keys per second. In 2010, the Georgia Tech Research Institute developed a method of using GPGPU to crack passwords, coming up with a minimum secure password length of 12 characters.

Perhaps the fastest way to crack passwords is through the use of pre-computed rainbow tables. These encode the hashes of common passwords based on the most widely used hash functions and can crack passwords in a matter of seconds. However, they are only effective on systems that do not use a salt, such as Windows LAN Manager and some application programs.

Prevention

The best method of preventing password cracking is to ensure that attackers cannot get access even to the encrypted password. For example, on the Unix operating system, encrypted passwords were originally stored in a publicly accessible file `/etc/passwd`. On modern Unix (and similar) systems, on the other hand, they are stored in the file `/etc/shadow`, which is accessible only to programs running with enhanced privileges (ie, 'system' privileges). This makes it harder for a malicious user to obtain the encrypted passwords in the first instance. Unfortunately, many common network protocols transmit passwords in cleartext or use weak challenge/response schemes.

Modern Unix systems have replaced traditional DES-based password hashing with stronger methods based on MD5 and Blowfish. Other systems have also begun to adopt these methods. For instance, the Cisco IOS originally used a reversible Vigenère cipher to encrypt passwords, but now uses md5-crypt with a 24-bit salt when the "enable secret" command is used. These newer methods use large salt values which prevent attackers from efficiently mounting offline attacks against multiple user accounts simultaneously. The algorithms are also much slower to execute which drastically increases the time required to mount a successful offline attack.

Many hashes used for storing passwords, such as MD5 and the SHA family, are designed for fast computation and efficient implementation in hardware. Using key stretching algorithms, such as PBKDF2, to form password hashes can significantly reduce the rate at which passwords can be tested.

Solutions like a security token give a formal proof answer by constantly shifting password. Those solutions abruptly reduce the timeframe for brute forcing (attacker needs to break and use the password within a single shift) and they reduce the value of the stolen passwords because of its short time validity.

Software

Main category: Password cracking software.

There are many password cracking software tools, but the most popular are Cain and Abel, John the Ripper, Hydra, ElcomSoft and Lastbit. Many litigation support software packages also include password cracking functionality. Most of these packages employ a mixture of cracking strategies, with brute force and dictionary attacks proving to be the most productive.

Remote administration

Remote administration refers to any method of controlling a computer from a remote location.

Software that allows remote administration is becoming increasingly common and is often used when it is difficult or impractical to be physically near a system in order to use it, or in order to access web material that is not available in one's location, for example viewing the BBC iPlayer from outside the United Kingdom. A remote location may refer to a computer in the next room or one on the other side of the world. It may also refer to both legal and illegal (i.e. hacking) remote administration.

Requirements

Internet connection

Any computer with an Internet connection, TCP/IP or on a Local Area Network can be remotely administered.

For non-malicious administration, the user must install or enable server software on the host system in order to be viewed. Then the user/client can access the host system from another computer using the installed software.

Usually, both systems should be connected to the internet, and the IP address of the host/server system must be known. Remote administration is therefore less practical if the host uses a dial-up modem, which is not constantly online and often has a Dynamic IP.

Connecting

When the client connects to the host computer, a window showing the Desktop of the host usually appears. The client may then control the host as if he/she were sitting right in front of it.

Certain versions of Windows XP have a built-in remote administration package called Remote Desktop Connection. A free cross-platform alternative is VNC, which offers similar functionality.

Common tasks for which remote administration is used

General

Controlling one's own computer from a remote location (e.g. to access the software on a personal computer from an internet café).

Shutdown

- Shutting down or rebooting another computer over a network

Accessing Peripherals

- Using a network device, like printer
- Retrieving streaming data, much like a CCTV system

Modifying

- Editing another computer's registry settings
- Modifying system services
- Installing software on another machine
- Modifying logical groups

Viewing

- Remotely assisting others
- Supervising computer or internet usage
- Access to a remote system's "Computer Management" snap-in

Popular software

Windows

Windows Server 2003/2008, Tablet PC Editions, and Windows Vista Ultimate, Enterprise and Business editions come with Microsoft's Microsoft Management Console, Windows Registry Editor and various command-line utilities that may be used to administrate a remote machine. One form of remote administration is remote desktop software, and Windows includes a Remote Desktop Connection client for this purpose.

Windows XP comes with a built-in remote administration tools called Remote Assistance and Remote Desktop, these are restricted versions of the Windows Server 2003 Terminal Services meant only for helping users and remote administration. With a simple hack/patch (derived from the beta version of Windows XP) it's possible to "unlock" XP to a fully featured Terminal Server, one good and easy example is Sala´s Terminal Server Patch. With this patch it is possible to make a terminal server out of Windows XP Professional, Multimeda Center, and Tablet PC Edition. Windows XP Home can be made to run a full-featured Terminal Service as well, with additional patching. .

Windows Server 2003 comes with built-in remote administration tools, including a web application and a simplified version of Terminal Services designed for Remote administration.

Active Directory and other features found in Microsoft's Windows NT Domains allow for remote administration of computers that are members of the domain, including editing the registry and modifying system services and access to the system's "Computer Management" Microsoft Management Console snap-in.

Some third-party remote desktop software programs perform the same job.

Remote Server Administration Tools for Windows 7 enables IT administrators to manage roles and features that are installed on remote computers that are running Windows Server 2008 R2

Non-Windows

VNC can be used for remote administration of computers, however it is increasingly being used as an equivalent of Terminal Services and Remote Desktop Protocol for multi-user environments.

Back Orifice, whilst commonly used as a Script Kiddie tool, claims to be a remote-administration and system management tool. Critics have previously stated that the capabilities of the software require a very loose definition of what "administration" entails.

Linux, UNIX and BSD support remote administration via remote login, typically via SSH (The use of the Telnet protocol has been phased out due to security concerns). X-server connection forwarding, often tunnelled over SSH for security, allows GUI programs to be used remotely. VNC is also available for these operating systems.

Apple Remote Desktop provides Macintosh users with remote administration capabilities.

Scriptlogic's Desktop Authority encompasses remote control as a part of remote management. This solution includes: secure web-based access to client machines, real-time diagnostics and troubleshooting, management of the file system, users/groups, registry, virtual memory, reboots and more - without user interaction, interactive remote monitoring and control of the desktop, supports clients running Windows 98 through XP/2003/Vista.

NX and its Google fork NeatX are free graphical Desktop sharing solutions for the X Window System with Clients for different platforms like Linux, Windows and Mac OS X. There is also a enhanced commercial version of NX Server available.

Wireless Remote Administration

Remote administration software has recently started to appear on wireless devices such as the BlackBerry, Pocket PC, and Palm devices, as well as some mobile phones.

Generally these solutions do not provide the full remote access seen on software such as VNC or Terminal Services, but do allow administrators to perform a variety of tasks, such as rebooting computers, resetting passwords, and viewing system event logs, thus reducing or even eliminating the need for system administrators to carry a laptop or be within reach of the office.

Chapter 10

Software Deployment and System Console

Software deployment

Software deployment is all of the activities that make a software system available for use.

The general deployment process consists of several interrelated activities with possible transitions between them. These activities can occur at the producer site or at the consumer site or both. Because every software system is unique, the precise processes or procedures within each activity can hardly be defined. Therefore, "deployment" should be interpreted as a *general process* that has to be customized according to specific requirements or characteristics. A brief description of each activity will be presented later.

Deployment activities

Release

> The release activity follows from the completed development process. It includes all the operations to prepare a system for assembly and transfer to the customer site. Therefore, it must determine the resources required to operate at the customer site and collect information for carrying out subsequent activities of deployment process.

Install and activate

> Activation is the activity of starting up the executable component of software. For simple system, it involves establishing some form of command for execution. For complex systems, it should make all the supporting systems ready to use.
>
> In larger software deployments, the working copy of the software might be installed on a production server in a production environment. Other versions of the deployed software may be installed in a test environment, development environment and disaster recovery environment.

Deactivate

> Deactivation is the inverse of activation, and refers to shutting down any executing components of a system. Deactivation is often required to perform other deployment activities, e.g., a software system may need to be deactivated before

an update can be performed. The practice of removing infrequently used or obsolete systems from service is often referred to as application retirement or application decommissioning.

Adapt

The adaptation activity is also a process to modify a software system that has been previously installed. It differs from updating in that adaptations are initiated by local events such as changing the environment of customer site, while updating is mostly started from remote software producer.

Update

The update process replaces an earlier version of all or part of a software system with a newer release.

Built-In

Mechanisms for installing updates are built into some software systems. Automation of these update processes ranges from fully automatic to user initiated and controlled. Norton Internet Security is an example of a system with a semi-automatic method for retrieving and installing updates to both the antivirus definitions and other components of the system. Other software products provide query mechanisms for determining when updates are available.

Version tracking

Version tracking systems help the user find and install updates to software systems installed on PCs and local networks.

- Web based version tracking systems notify the user when updates are available for software systems installed on a local system. For example: VersionTracker Pro checks software versions on a user's computer and then queries its database to see if any updates are available.
- Local version tracking system notifies the user when updates are available for software systems installed on a local system. For example: Software Catalog stores version and other information for each software package installed on a local system. One click of a button launches a browser window to the upgrade web page for the application, including auto-filling of the user name and password for sites that require a login.
- Browser based version tracking systems notify the user when updates are available for software packages installed on a local system. For example: wfx-Versions is a Firefox extension which helps the user find the current version number of any program listed on the web.

Uninstall

Uninstallation is the inverse of installation. It is the removal of a system that is no longer required. It also involves some reconfiguration of other software systems in order to remove the uninstalled system’s files and dependencies.

Retire

Ultimately, a software system is marked as obsolete and support by the producers is withdrawn. It is the end of the life cycle of a software product.

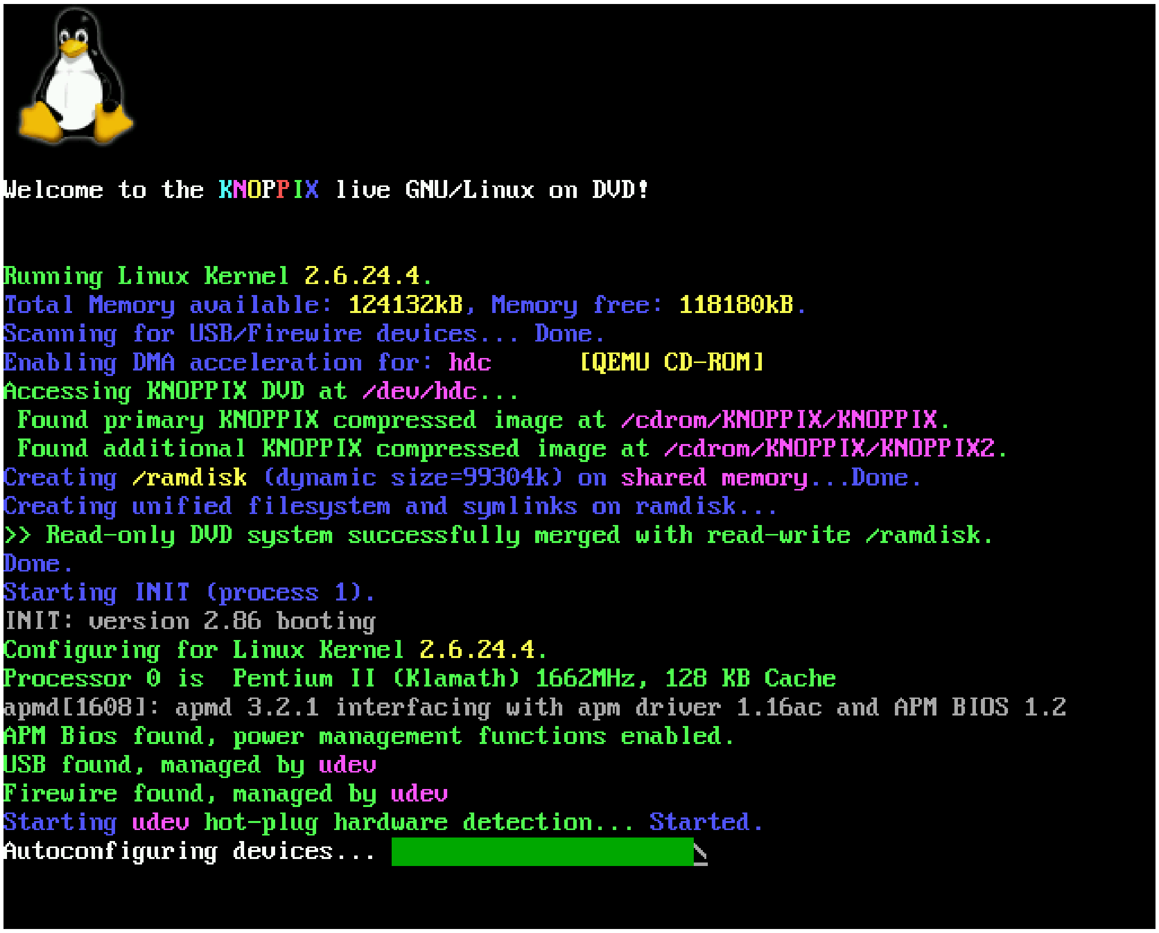
Deployment roles

The complexity and variability of software products has necessitated the creation of specialized roles for coordinating and engineering the deployment process. For desktop systems, an end user is frequently also the "software deployer" when they install the software package on their machine. For enterprise software, there are many more roles involved. Additionally, the roles involved typically change as the application progresses from test (pre-production) to production environments. The typical roles involved in software deployments for enterprise applications are:

- Pre-production environments
    - Application developers
    - Build and release engineers
    - Release managers
    - Deployment coordinators

- Production environments
    - System administrator
    - Database administrator
    - Release coordinators
    - Operations project managers

System console



```
Welcome to the KNOPPIX live GNU/Linux on DVD!


Running Linux Kernel 2.6.24.4.
Total Memory available: 124132kB, Memory free: 118180kB.
Scanning for USB/Firewire devices... Done.
Enabling DMA acceleration for: hdc      [QEMU CD-ROM]
Accessing KNOPPIX DVD at /dev/hdc...
 Found primary KNOPPIX compressed image at /cdrom/KNOPPIX/KNOPPIX.
 Found additional KNOPPIX compressed image at /cdrom/KNOPPIX/KNOPPIX2.
Creating /ramdisk (dynamic size=99304k) on shared memory...Done.
Creating unified filesystem and symlinks on ramdisk...
>> Read-only DVD system successfully merged with read-write /ramdisk.
Done.
Starting INIT (process 1).
INIT: version 2.86 booting
Configuring for Linux Kernel 2.6.24.4.
Processor 0 is  Pentium II (Klamath) 1662MHz, 128 KB Cache
apmd[1608]: apmd 3.2.1 interfacing with apm driver 1.16ac and APM BIOS 1.2
APM Bios found, power management functions enabled.
USB found, managed by udev
Firewire found, managed by udev
Starting udev hot-plug hardware detection... Started.
Autoconfiguring devices...
```

Knoppix system console showing the boot process

The **system console**, **root console** or simply **console** is the text entry and display device for system administration messages, particularly those from the BIOS or boot loader, the kernel, from the init system and from the system logger. It is a physical device consisting of a keyboard and a screen.

On traditional minicomputers, the console was a **serial console**, an RS-232 serial link to a terminal such as a DEC VT100. This terminal was usually kept in a secured room since it could be used for certain privileged functions such as halting the system or selecting which media to boot from. Large midrange systems, e.g. those from Sun Microsystems, Hewlett-Packard and IBM, still use serial consoles. In larger installations, the console ports are attached to multiplexers or network-connected multiport serial servers that let an operator connect a terminal to any of the attached servers. Today, serial consoles are often used for accessing headless systems, usually with a terminal emulator running on a laptop. Also, routers, enterprise network switches and other telecommunication equipment have RS-232 serial console ports.

On PCs and workstations, the computer's attached keyboard and monitor have the equivalent function. Since the monitor cable carries video signals, it cannot be extended very far. Often, installations with many servers therefore use keyboard/video multiplexers (KVM switches) and possibly video amplifiers to centralize console access. In recent years, KVM/IP devices have become available that allow a remote computer to view the video output and send keyboard input via any TCP/IP network and therefore the Internet.

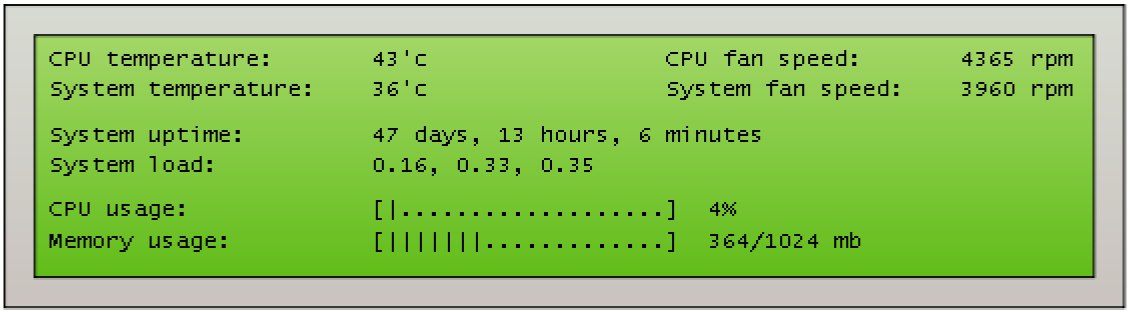
Some PC BIOSes, especially in servers, also support serial consoles, giving access to the BIOS through a serial port so that the simpler and cheaper serial console infrastructure can be used. Even where BIOS support is lacking, some operating systems, e.g. FreeBSD and Linux, can be configured for serial console operation either during bootup, or after startup.

It is usually possible to log in from the console. Depending on configuration, the operating system may treat a login session from the console as being more trustworthy than a login session from other sources.

Chapter 11

System Monitor and System Profiler

System monitor



```
CPU temperature:       43'c                  CPU fan speed:       4365 rpm
System temperature:    36'c                  System fan speed:    3960 rpm

System uptime:         47 days, 13 hours, 6 minutes
System load:           0.16, 0.33, 0.35

CPU usage:             [|....................]  4%
Memory usage:          [|||||||..............]  364/1024 mb
```

Simulated LCD panel for display of resource usage.

A **system monitor** is a hardware- or software- based system used to monitor resources and performance in a computer system.

Software monitors occur more commonly, sometimes as a part of a widget engine. These monitoring systems are often used to keep track of system resources, such as CPU usage and frequency , or the amount of free RAM. They are also used to display items such as free space on one or more hard drives, the temperature of the CPU and other important components, and networking information including the system IP address and current rates of upload and download. Other possible displays may include the date and time, system uptime, computer name, username, hard drive S.M.A.R.T data, fan speeds, and the voltages being provided by the power supply.

Less common are hardware-based systems monitoring similar information. Customarily these occupy one or more drive bays on the front of the computer case, and either interface directly with the system hardware or connect to a software data-collection system via USB. With either approach to gathering data, the monitoring system displays information on a small LCD panel or on series of small analog or LED numeric displays. Some hardware-based system monitors also allow direct control of fan speeds, allowing the user to quickly customize the cooling in the system.

A few very high-end models of hardware system monitor are designed to interface with only a specific model of motherboard. These systems directly utilize the sensors built into the system, providing more detailed and accurate information than less-expensive monitoring systems customarily provide.

Software versions are becoming more common, with even the Microsoft Windows Vista sidebar including a meter to monitor CPU and RAM usage. Hardware versions are very rare on OEM computers, with the exception of high-end servers. However, computer enthusiasts often install such hardware-based monitors.

Software

- CPU-Z, detects live changes in CPU attributes
- Conky
- frysk - analyzes and monitors system processes
- Ganglia (software)
- GKrellM
- htop (Unix)
- Iostat
- Motherboard Monitor
- Nmon
- Sar in UNIX
- SpeedFan
- top (software)
- Vmstat
- Windows Sidebar on Windows Vista
- WinBar
- Activity monitor on Mac OS X.

System profiler

A **system profiler** is a program that can provide detailed information about the software installed and hardware attached to the computer. In older versions of Apple Computer's Mac OS, this was done by an application called Apple System Profiler. Mac OS X's profiler is simply called System Profiler. In Microsoft Windows some similar information may be found by getting properties on My Computer on the desktop.

List of system profiler software

Microsoft Windows

- Sisoft SANDRA (System Analyser, Diagnostic and Reporting Assistant) system profiling software
- HWiNFO32 - Freeware system information tool.
- Lavalys EVEREST Ultimate Edition (formerly AIDA32) - Shareware System information and benchmarking tool.
- FreshDiagnose - Freeware system information tool.
- CPU-Z, useful when overclocking processors
- SIW - System Information for Windows - portable freeware (does not require installation) - software, hardware, network information, tools and real-time monitors
- Belarc Freeware for personal use PC Auditing Software lists hardware, as well as software installed on the local machine and displays as a local webpage. Belarc also makes a security assessment for checking how secure a system is, and links missing updates directly to a Microsoft website for download.
- systeminfo native windows command line, returns OS version, uptime, CPU, physical memory, network cards, etc ...
- msinfo32 or winmsd comprehensive view of hardware, system components, and software environment
- PsInfo command line, from the Sysinternals suite
- SekChek Local - an automated security audit tool which scans multiple Windows workstations and servers, from the network. It creates a security assessment report file which is presented as an Microsoft Access dataset.
- CPU Speed Pro is a Microsoft Windows software application which will test the speed of the central processing unit (CPU)
- SlimWare Utilities - PC data and registry cleaner, driver diagnostic, system analyzer and optimization software.

Linux

- LSHW
- HardInfo
- SekChek Classic - The UNIX extract tool extracts control data regarding security policies and objects defined on the target host.

DOS

- HWiNFO - still updated DOS version of HWiNFO32

OS Independent

- Hardware Detection Tool (HDT)

Chapter 12

Systems Management and Windows Management Instrumentation

Systems management

Systems management refers to enterprise-wide administration of distributed systems including (and commonly in practice) computer systems. Systems management is strongly influenced by network management initiatives in telecommunications.

Centralized management has a time and effort tradeoff that is related to the size of the company, the expertise of the IT staff, and the amount of technology being used:

- For a small-business startup with ten computers, automated centralized processes may take more time to learn how to use and implement than just doing the management work manually on each computer.
- A very large business with thousands of similar employee computers may clearly be able to save time and money, by having IT staff learn to do systems management automation.
- A small branch office of a large corporation may have access to a central IT staff, with the experience to set up automated management of the systems in the branch office, without need for local staff in the branch office to do the work.

System management may involve one or more of the following tasks:

- Hardware inventories.
- Server availability monitoring and metrics.
- Software inventory and installation.
- Anti-virus and anti-malware management.
- User's activities monitoring.
- Capacity monitoring.
- Security management.
- Storage management.
- Network capacity and utilization monitoring.
- Anti-manipulation management

Functions

Functional groups are provided according to International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Common management information protocol (X.700) standard. This framework is also known as Fault, Configuration, Accounting, Performance, Security (FCAPS).

Fault management

- Troubleshooting, error logging and data recovery

Configuration management

- Hardware and software inventory

- *As we begin the process of automating the management of our technology, what equipment and resources do we have already?*
- *How can this inventorying information be gathered and updated automatically, without direct hands-on examination of each device, and without hand-documenting with a pen and notepad?*
- *What do we need to upgrade or repair?*
- *What can we consolidate to reduce complexity or reduce energy use?*
- *What resources would be better reused somewhere else?*
- *What commercial software are we using that is improperly licensed, and either needs to be removed or more licenses purchased?*

- Provisioning

- *What software will we need to use in the future?*
- *What training will need to be provided to use the software effectively?*

- Software deployment

- *What steps are necessary to install it on perhaps hundreds or thousands of computers?*

- Package management

- *How do we maintain and update the software we are using, possibly through automated update mechanisms?*

Accounting management

- Billing and statistics gathering

Performance management

- Software metering

- *Who is using the software and how often?*
- *If the license says only so many copies may be in use at any one time but may be installed in many more places than licensed, then track usage of those licenses.*
- *If the licensed user limit is reached, either prevent more people from using it, or allow overflow and notify accounting that more licenses need to be purchased.*

- Event and metric monitoring

- *How reliable are the computers and software?*
- *What errors or software bugs are preventing staff from doing their job?*
- *What trends are we seeing for hardware failure and life expectancy?*

Security management

- Identity management
- Policy management

However this standard should not be treated as comprehensive, there are obvious omissions. Some are recently emerging sectors, some are implied and some are just not listed. The primary ones are:

- Business Impact functions (also known as Business Systems Management)
- Capacity Management
- Real-time Application Relationship Discovery (which supports Configuration Management)
- Security Information and Event Management functions (SIEM)
- Workload Scheduling

Performance Management functions can also be split into end-to-end performance measuring and infrastructure component measuring functions. Another recently emerging sector is Operational Intelligence which focuses on real-time monitoring of business events that relate to business processes, not unlike Business Activity Monitoring.

Standards

Distributed Management Task Force (DMTF)

Alert Standard Format (ASF)
Common Information Model (CIM)
Desktop and mobile Architecture for System Hardware DASH
Directory Enabled Networking (DEN)
Systems Management Architecture for Server Hardware (SMASH)
Java Management Extensions (JMX)

Windows Management Instrumentation

Management Instrumentation (**WMI**) is a set of extensions to the Windows Driver Model that provides an operating system interface through which instrumented components provide information and notification. WMI is Microsoft's implementation of the Web-Based Enterprise Management (WBEM) and Common Information Model (CIM) standards from the Distributed Management Task Force (DMTF).

WMI allows scripting languages like VBScript or Windows PowerShell to manage Microsoft Windows personal computers and servers, both locally and remotely. WMI is preinstalled in Windows 2000 and newer OSs. It is available as a download for Windows NT, Windows 95 and Windows 98.

Microsoft also provides a command line interface to WMI called **Windows Management Instrumentation Command-line** (**WMIC**).

Purpose of WMI

The purpose of WMI is to define a non-proprietary set of environment-independent specifications which allow management information to be shared between management applications. WMI prescribes enterprise management standards and related technologies that work with existing management standards, such as Desktop Management Interface (DMI) and SNMP. WMI complements these other standards by providing a uniform model. This model represents the managed environment through which management data from any source can be accessed in a common way.

Development process

Because WMI abstracts the manageable entities with CIM and a collection of providers, the development of a provider implies several steps. The major steps can be summarized as follows:

Step 1 – Create the manageable entity model

- Define a model
- Implement the model

Step 2 – Create the WMI Provider

- Determine the provider type to implement
- Determine the hosting model of the provider
- Create the provider template with the ATL wizard
- Implement the code logic in the provider
- Register the provider with WMI and the system

Step 3 – Test the provider

Step 4 – Create consumer sample code

Importance of WMI providers

Since the release of the first WMI implementation during the Windows NT 4.0 SP4 era (as an out-of-band download), Microsoft has consistently added WMI providers to Windows. Under Windows NT 4.0, Microsoft had roughly 15 WMI providers available once WMI was installed. When Windows 2000 was released, there were 29 WMI providers as part of the operating system installation. With the release of Windows Server 2003, Microsoft included in the platform more than 80 WMI providers. Windows Vista includes 13 new WMI providers, taking the number close to around 100 in all, and Windows Server 2008 includes some more including providers for IIS 7, PowerShell and virtualization. This has been a sign for many customers that WMI became at Microsoft, the "ubiquitous" management layer of Windows, even if this commitment has never been explicit from Microsoft.

During these last years, due to a constant increasing exposure of management data through WMI in Windows, more and more people in the IT systems management field started to develop scripts and automation procedures based on WMI. Beyond the scripting needs, most leading management software in the world, such as MOM, SMS, ADS, HP OpenView for Windows (HPOV), BMC Software or CA, Inc. are WMI-enabled and capable to consume and provide WMI information through various *User Interfaces*. This enables administrators and operators not capable of scripting or programming on top of WMI to enjoy the benefits of WMI without even learning about it. However, if they want to, because WMI is scriptable, it gives them the opportunity to consume WMI information from scripts or from any Enterprise Management software that is WMI-aware.

Features

For someone willing to develop one or many WMI providers, WMI offers many features out of the box. Here are the most important advantages:

1. *Automation interfaces:* Because WMI comes with a set of automation interfaces ready to use, all management features supported by a WMI provider and its set of classes get the scripting support for free out-of-the box. Beyond the WMI class design and the provider development, the Microsoft development and test teams are not required to create, validate and test a scripting model as it is already available from WMI.
2. *.NET Management interfaces:* Because the System.Management namespace relies on the existing COM/DCOM plumbing, the created WMI provider and its set of WMI classes becomes automatically available to all .NET applications independently of the language used (e.g. C#, VB.NET). Beyond the WMI class design and the provider development, like for scripting, the Microsoft

development and test teams are not required to create, validate and test new assemblies to support a new namespace in the .NET Framework as this support is already available from WMI for free.

3. *C/C++ COM/DCOM programming interfaces:* Like most components in Windows, COM/DCOM programmers can leverage the features of the provider they develop at the COM/DCOM interfaces level. Like in previous environments (scripting and .NET Framework), a COM/DCOM consumer just needs to interact with the standard set of WMI COM interfaces to leverage the WMI provider capabilities and its set of supported WMI classes. To make all management information available from the native APIs, the WMI provider developer just needs to interact with a set of pre-defined WMI COM interfaces. This will make the management information available at the WMI COM level automatically. Moreover, the scripting COM interface object model is very similar to the COM/DCOM interface object model, which makes it easy for developers to be familiar with the scripting experience.
4. *Remoting capabilities over DCOM and SOAP:* More than simply offering local COM capabilities, as management is all about remoting, WMI offers the DCOM transport. In addition, SOAP transport will be available in Windows Server 2003 **R2** through the WS-Management initiative led by Microsoft, Intel, Sun Microsystems and Dell. This initiative allows to run any scripts remotely or to consume WMI data through a specific set of interfaces handling SOAP requests/responses. The advantage for the WMI provider developer is that when he exposes all his features through WMI, *Windows Remote Management*/WS-Management can in turn consume that information as well (embedded objects in WMI instances are not supported in Windows Server 2003 R2. It is however a target for Vista). All the layering to WS-Management and the mapping of the CIM data model to SOAP comes for free out of the WMI/WS-Management solution. In the event DCOM must be used, implementing DCOM requires the presence of a proxy DLL deployed on each client machine. As WMI is available in the Windows operating system since Windows 2000, these issues are eliminated.
5. *Support for Queries:* WMI offers support for WQL queries out of the box. This means that if a provider is not designed to support queries, WMI supports it by using an enumeration technique out of the provider.
6. *Eventing capabilities:* WMI offers the capability to notify a subscriber for any event it is interested in. WMI uses the WMI Query Language (WQL) to submit WQL event queries and defines the type of events to be returned. The eventing mechanism, with all related callbacks, is part of the WMI COM/DCOM and automation interfaces. Anyone writing a WMI provider can have the benefit of this functionality at no cost for his customers. It will be up to the consumer to decide how it wants to consume the management information exposed by the WMI provider and its related set of WMI classes.
7. *Code template generator:* To speed up the process of writing a WMI provider including all COM/DCOM interfaces and related definitions, the WMI team developed the *WMI ATL Wizard* to generate the code template implementing a provider. The code generated is based on the WMI class model initially designed

by the developer. The WMI provider developer will be able to interface the pre-defined COM/DCOM interfaces for the WMI provider with its set of native APIs retrieving the management information to expose. The exercise consists in filling the "gaps" in the provider code to create the desired interfacing logic.

8. *Predictability:* Predictability is an important concern for IT professionals because it defines the capability of someone having an experience with a set of interfaces managing a Windows component to apply this knowledge right away, intuitively, to any other manageable Windows component without having relearn everything from ground up. Predictability for a customer is a real gain as it increases the Return of Investment (ROI). A person facing such a situation simply expects things to work the same way based on his previous experience. The constant increase of COM programming/scriptable interfaces has a huge impact on the predictability, as this makes it difficult for customers to automate, manage Windows and leverage their existing knowledge. WMI with CIM address this problem by always exposing the same programming object model (COM/DCOM, Automation, .NET) whatever the manageable entity is.
9. *Protect existing customer investments:* Protecting customers and partners investment motivates customers to invest in technologies. As Microsoft did invest a lot these past years in writing WMI providers, customers and partners invested in tools leveraging the WMI capabilities of Windows. Therefore, they naturally continue to exploit these capabilities instead of having to use a new set of specific interfaces for each Windows manageable component. A specific set of interfaces means having a specific set of agents or in-house developed software based on a new model or set of interfaces especially dedicated to a component or technology. By leveraging the capabilities of WMI today, customers and partners can leverage the work investment made in the past while minimizing their costs in developments, learning curves and new discoveries. This will also have a great impact on the stability and reliability of their infrastructure as they continue to leverage an existing implementation with an improved technology.
10. *Provide a logical and unified administration model:* As briefly described before in the introduction, this model is based on an industry standard called CIM defined by the DMTF. The CIM class-based schema is defined by a consortium of constructors and software developers that meets the requirements of the industry. This implies that not only Microsoft leverages the WMI capabilities, but also any other third party constructors or developers write their own code to fit into the model. For instance, Intel is doing this for some their network driver adapters and software. HP is leveraging existing WMI providers and implementing their own WMI providers in their HP Open View Enterprise Management software. IBM consumes WMI from the Tivoli management suite, MOM and SMS are also consuming and providing WMI information. Lastly, Windows XP SP2 leverages WMI to get information status from anti-virus software and firewalls.

WMI tools

Some WMI tools can also be useful during the design and development phases. These tools are:

- *The MOF compiler (MOFComp.exe):* The Managed Object Format (MOF) compiler parses a file containing Managed Object Format statements and adds the classes and class instances defined in the file to the CIM repository. The MOF format is a specific syntax to define CIM class representation in an ASCII file (e.g. MIB are to SNMP what MOF files are to CIM). MOFComp.exe is included in every WMI installation. Every definition existing in the CIM repository is initially defined in an MOF file. MOF files are located in %SystemRoot%\System32\WBEM. During the WMI setup, they are loaded in the CIM repository.
- *The WMI Administrative Tools:* The WMI Administrative Tools are made of four tools: WMI CIM Studio, WMI Object Browser, WMI Event Registration and WMI Event Viewer. WMI Administrative Tools can be downloaded here. The most important tool for a WMI provider developer is WMI CIM Studio as it helps in the initial WMI class creation in the CIM repository. It uses a web interface to display information and relies on a collection of ActiveX components installed on the system when it runs for the first time. WMI CIM Studio provides the ability to:
    - Connect to a chosen system and browse the CIM repository in any namespace available.
    - Search for classes by their name, by their descriptions or by property names.
    - Review the properties, methods and associations related to a given class.
    - Perform Queries in the WQL language.
    - Generate an MOF file based on selected classes.
    - Compile an MOF file to load it in the CIM repository.
- *WinMgmt.exe:* WinMgmt.exe is not a tool; it is the executable that implements the WMI Core service. Under the Windows NT family of operating systems, WMI runs as a service. On computers running Windows 98, Windows 95 or Windows Me, WMI runs as an application. Under the Windows NT family of operating systems, it is also possible to run this executable as an application, in which case, the executable runs in the current user context. For this, the WMI service must be stopped first. The executable supports some switches that can be useful when starting WMI as a service or as an application. WMI provider developers who may want to debug their providers essentially need to run the WMI service as an application.
- *WBEMTest.exe:* WBEMTest.exe is a WMI tester tool, which is delivered with WMI. This tool allows an administrator or a developer to perform most of the tasks from a graphical interface that WMI provides at the API level. Although available under all Windows NT-based operating systems, this tool is not officially supported by Microsoft. WBEMTest provides the ability to:
    - Enumerate, open, create and delete classes.
    - Enumerate, open, create and delete instances of classes.
    - Select a namespace.
    - Perform data and event queries.
    - Execute methods associated to classes or instances.

    - Execute every WMI operation asynchronously, synchronously or semi-asynchronously.
- The WMI command line tool (WMIC): WMIC is a command-line tool designed to ease WMI information retrieval about a system by using some simple keywords (aliases). WMIC.exe is only available under Windows XP Professional, Windows Server 2003, Windows Vista and Windows Server 2008. By typing “WMIC /?” from the command-line, a complete list of the switches and reserved keywords is available.
    - There is a Linux port of WMI command line tool, written in Python, based on Samba4 called 'wmi-client'
- *WBEMDump.exe:* WBEMDump is a tool delivered with the Platform SDK. This command line tool comes with its own Visual C++ project. The tool can show the CIM repository classes, instances, or both. It is possible to retrieve the same information as that retrieved with WMIC. WBEMDump.exe requires more specific knowledge about WMI, as it doesn’t abstract WMI as WMIC. However, it runs under Windows NT 4.0 and Windows 2000. It is also possible to execute methods exposed by classes or instances. Even if it is not a standard WMI tool delivered with the system installation, this tool can be quite useful for exploring the CIM repository and WMI features.

Wireless networking example

In the .NET framework, the ManagementClass class represents a Common Information Model (CIM) management class. A WMI class can be a Win32_LogicalDisk in the case of a disk drive, or a Win32_Process, such as a running program like Notepad.exe.

This example shows how "MSNdis_80211_ServiceSetIdentifier" WMI class is used to find the SSID of the Wi-Fi network that the system is currently connected to in the language C#:

```
ManagementClass mc = new ManagementClass("root\\WMI",
"MSNdis_80211_ServiceSetIdentifier", null);
ManagementObjectCollection moc = mc. GetInstances();

foreach (ManagementObject mo in moc)
{
   string wlanCard = (string)mo["InstanceName"];
   bool active;
   if (!bool.TryParse((string)mo["Active"], out active))
   {
      active = false;
   }
   byte[] ssid = (byte[])mo["Ndis80211SsId"];
}
```

The "MSNdis_80211_ServiceSetIdentifier" WMI class is only supported on Windows XP and Windows Server 2003.

WMI driver extensions

The WMI extensions to WDM provide kernel-level instrumentation such as publishing information, configuring device settings, supplying event notification from device drivers and allowing administrators to set data security through a WMI provider known as the *WDM provider*. The extensions are part of the WDM architecture; however, they have broad utility and can be used with other types of drivers as well (such as SCSI and NDIS). The WMI Driver Extensions service monitors all drivers and event trace providers that are configured to publish WMI or event trace information. Instrumented hardware data is provided by way of drivers instrumented for WMI extensions for WDM. WMI extensions for WDM provide a set of Windows device driver interfaces for instrumenting data within the driver models native to Windows, so OEMs and IHVs can easily extend the instrumented data set and add value to a hardware/software solution. The WMI Driver Extensions, however, are not supported by Windows Vista and later operating systems.

Chapter 13

Autonomic Networking

Autonomic Networking follows the concept of Autonomic Computing, an initiative started by IBM in 2001. Its ultimate aim is to create self-managing networks to overcome the rapidly growing complexity of the Internet and other networks and to enable their further growth, far beyond the size of today.

Increasing size and complexity

The ever-growing management complexity of the Internet caused by its rapid growth is seen by some experts as a major problem that limits its usability in the future.

What's more, increasingly popular smartphones, PDAs, networked audio and video equipment, and game consoles need to be interconnected. Pervasive Computing not only adds features, but also burdens existing networking infrastructure with more and more tasks that sooner or later will not be manageable by human intervention alone.

Another important aspect is the price of manually controlling huge numbers of vitally important devices of current network infrastructures.

Autonomic nervous system

The autonomic nervous system (ANS) is the part of the nervous system of the higher life forms that is not consciously controlled. It regulates bodily functions and the activity of specific organs. As proposed by IBM, future communication systems might be designed in a similar way to the ANS.

Components of autonomic networking

As autonomics conceptually derives from biological entities such as the human autonomic nervous system, each of the areas can be metaphorically related to functional and structural aspects of a living being. In the human body, the autonomic system facilitates and regulates a variety of functions including respiration, blood pressure and circulation, and emotive response. The autonomic nervous system is the interconnecting fabric that supports feedback loops between internal states and various sources by which internal and external conditions are monitored.

Autognostics

Autognostics includes a range of self-discovery, awareness, and analysis capabilities that provide the autonomic system with a view on high-level state. In metaphor, this represents the perceptual sub-systems that gather, analyze, and report on internal and external states and conditions – for example, this might be viewed as the eyes, visual cortex and perceptual organs of the system. Autognostics, or literally "self-knowledge", provides the autonomic system with a basis for response and validation.

A rich autognostic capability may include many different "perceptual senses". For example, the human body gathers information via the usual five senses, the so-called sixth sense of proprioception (sense of body position and orientation), and through emotive states that represent the gross wellness of the body. As conditions and states change, they are detected by the sensory monitors and provide the basis for adaptation of related systems. Implicit in such a system are imbedded models of both internal and external environments such that relative value can be assigned to any perceived state - perceived physical threat (e.g. a snake) can result in rapid shallow breathing related to fight-flight response, a phylogenetically effective model of interaction with recognizable threats.

In the case of autonomic networking, the state of the network may be defined by inputs from:

- individual network elements such as switches and network interfaces including
    - specification and configuration
    - historical records and current state
- traffic flows
- end-hosts
- application performance data
- logical diagrams and design specifications

Most of these sources represent relatively raw and unprocessed views that have limited relevance. Post-processing and various forms of analysis must be applied to generate meaningful measurements and assessments against which current state can be derived.

The autognostic system interoperates with:

- configuration management - to control network elements and interfaces
- policy management - to define performance objectives and constraints
- autodefense - to identify attacks and accommodate the impact of defensive responses

Configuration management

Configuration management is responsible for the interaction with network elements and interfaces. It includes an accounting capability with historical perspective that provides for the tracking of configurations over time, with respect to various circumstances. In the biological metaphor, these are the hands and, to some degree, the memory of the autonomic system.

On a network, remediation and provisioning are applied via configuration setting of specific devices. Implementation affecting access and selective performance with respect to role and relationship are also applied. Almost all the "actions" that are currently taken by human engineers fall under this area. With only a few exceptions, interfaces are set by hand, or by extension of the hand, through automated scripts.

Implicit in the configuration process is the maintenance of a dynamic population of devices under management, a historical record of changes and the directives which invoked change. Typical to many accounting functions, configuration management should be capable of operating on devices and then rolling back changes to recover previous configurations. Where change may lead to unrecoverable states, the sub-system should be able to qualify the consequences of changes prior to issuing them.

As directives for change must originate from other sub-systems, the shared language for such directives must be abstracted from the details of the devices involved. The configuration management sub-system must be able to translate unambiguously between directives and hard actions or to be able to signal the need for further detail on a directive. An inferential capacity may be appropriate to support sufficient flexibility (i.e. configuration never takes place because there is no unique one-to-one mapping between directive and configuration settings). Where standards are not sufficient, a learning capacity may also be required to acquire new knowledge of devices and their configuration.

Configuration management interoperates with all of the other sub-systems including:

- autognostics - receives direction for and validation of changes
- policy management - implements policy models through mapping to underlying resources
- security - applies access and authorization constraints for particular policy targets
- autodefense - receives direction for changes

Policy management

Policy management includes policy specification, deployment, reasoning over policies, updating and maintaining policies, and enforcement. Policy management is required for:

- constraining different kinds of behavior including security, privacy, resource access, and collaboration
- configuration management
- describing business processes and defining performance
- defining role and relationship, and establishing trust and reputation

It provides the models of environment and behavior that represent effective interaction according to specific goals. In the human nervous system metaphor, these models are implicit in the evolutionary "design" of biological entities and specific to the goals of survival and procreation. Definition of what constitutes a policy is necessary to consider what is involved in managing it. A relatively flexible and abstract framework of values, relationships, roles, interactions, resources, and other components of the network environment is required. This sub-system extends far beyond the physical network to the applications in use and the processes and end-users that employ the network to achieve specific goals. It must express the relative values of various resources, outcomes, and processes and include a basis for assessing states and conditions.

Unless embodied in some system outside the autonomic network or implicit to the specific policy implementation, the framework must also accommodate the definition of process, objectives and goals. Business process definitions and descriptions are then an integral part of the policy implementation. Further, as policy management represents the ultimate basis for the operation of the autonomic system, it must be able to report on its operation with respect to the details of its implementation.

The policy management sub-system interoperates (at least) indirectly with all other sub-systems but primarily interacts with:

- autognostics - providing the definition of performance and accepting reports on conditions
- configuration management - providing constraints on device configuration
- security - providing definitions of roles, access and permissions

Autodefense

Autodefense represents a dynamic and adaptive mechanism that responds to malicious and intentional attacks on the network infrastructure, or use of the network infrastructure to attack IT resources. As defensive measures tend to impede the operation of IT, it is optimally capable of balancing performance objectives with typically over-riding threat management actions. In the biological metaphor, this sub-system offers mechanisms comparable to the immune system.

This sub-system must proactively assess network and application infrastructure for risks, detect and identify threats, and define effective both proactive and reactive defensive responses. It has the role of the warrior and the security guard insofar as it has roles for both maintenance and corrective activities. Its relationship with security is close but not identical – security is more concerned with appropriately defined and implemented access and authorization controls to maintain legitimate roles and process. Autodefense deals with forces and processes, typically malicious, outside the normal operation of the system that offer some risk to successful execution.

Autodefense requires high-level and detailed knowledge of the entire network as well as imbedded models of risk that allow it to analyze dynamically the current status.

Corrections to decrease risk must be considered in balance with performance objectives and value of process goals – an overzealous defensive response can immobilize the system (like the immune system inappropriately invoking an allergic reaction). The detection of network or application behaviors that signal possible attack or abuse is followed by the generation of an appropriate response – for example, ports might be temporarily closed or packets with a specific source or destination might be filtered out. Further assessment generates subsequent changes either relaxing the defensive measures or strengthening them.

Autodefense interoperates closely with:

- security - receives definition of roles and security constraints, and defines risk for proactive mitigation
- configuration management - receives details of network for analysis and directs changes in elements in response to anticipated or detected attack
- autognostics - receives notification of detected behaviors

It also may receive definition of relative value of various resources and processes from policy management in order to develop responses consistent with policy.

Security

Security provides the structure that defines and enforces the relationships between roles, content, and resources, particularly with respect to access. It includes the framework for definitions as well as the means to implement them. In metaphor, security parallels the complex mechanisms underlying social interactions, defining friends, foes, mates and allies and offering access to limited resources on the basis of assessed benefit.

Several key means are employed by security – they include the well-known 3 As of authentication, authorization, and access (control). The basis for applying these means requires the definition of roles and their relationships to resources, processes and each other. High-level concepts like privacy, anonymity and verification are likely imbedded in the form of the role definitions and derive from policy. Successful security reliably supports and enforces roles and relationships.

Autodefense has a close association with security – maintaining the assigned roles in balance with performance exposes the system to potential violations in security. In those cases, the system must compensate by making changes that may sacrifice balance on a temporary basis and indeed may violate the operational terms of security itself. Typically the two are viewed as inextricably intertwined – effective security somewhat hopefully negating any need for a defensive response. Security’s revised role is to mediate between the competing demands from policy for maximized performance and minimized risk with auto defense recovering the balance when inevitable risk translates to threat. Federation represents one of the key challenges to be solved by effective security.

The security sub-system interoperates directly with:

- policy management - receiving high-level directives related to access and priority
- configuration management - sending specifics for access and admission control
- autodefense - receiving over-riding directives under threat and sending security constraint details for risk assessment

Connection fabric

The connection fabric supports the interaction with all the elements and sub-systems of the autonomic system. It may be composed of a variety of means and mechanisms, or may be a single central framework. The biological equivalent is the central nervous system itself – although referred to as the autonomic system, it actually is only the communication conduit between the human body's faculties.

Principles of autonomic networking

Consequently, it is currently under research by many research projects, how principles and paradigms of mother nature might be applied to networking.

Compartmentalization

Instead of a layering approach, autonomic networking targets a more flexible structure termed compartmentalization.

Function re-composition

The goal is to produce an architectural design that enables flexible, dynamic, and fully autonomic formation of large-scale networks in which the functionalities of each constituent network node are also composed in an autonomic fashion

Atomization

Functions should be divided into atomic units to allow for maximal re-composition freedom.

Closed control loop

A fundamental concept of Control theory, the closed control loop, is among the fundamental principles of autonomic networking. A closed control loop maintains the properties of the controlled system within desired bounds by constantly monitoring target parameters.

Chapter 14

FCAPS

FCAPS is the ISO Telecommunications Management Network model and framework for network management. FCAPS is an acronym for *Fault, Configuration, Accounting, Performance, Security*, the management categories into which the ISO model defines network management tasks. In non-billing organizations *Accounting* is sometimes replaced with *Administration*.

The comprehensive management of an organization's information technology (IT) infrastructure is a fundamental requirement. Employees and customers rely on IT services where availability and performance are mandated, and problems can be quickly identified and resolved. Mean time to repair (MTTR) must be as short as possible to avoid system downtimes where a loss of revenue or lives is possible.

History

In the early 1980s the term FCAPS was introduced within the first Working Drafts (N1719) of ISO 10040, the Open Systems Interconnection (OSI) Systems Management Overview (SMO) standard. At that time the intention was to define five separate protocol standards, one for each functional area. Since initial experiences showed that these protocols would become very similar, the ISO working group responsible for the development of these protocols (ISO/TC97/SC16/WG4, later renamed into ISO-IEC/JTC1/SC21/WG4) decided to create a single protocol for all five areas instead. This protocol is called common management information protocol (CMIP). In the 1990s the ITU-T, as part of their work on Telecommunications Management Network (TMN), further refined the FCAPS as part of the TMN recommendation on Management Functions (M.3400). The idea of FCAPS turned out to be very useful for teaching network management functions; most text books therefore start with a section that explains the FCAPS.

Fault management

A fault is an event that has a negative significance. The goal of fault management is to recognize, isolate, correct and log faults that occur in the network. Furthermore, it uses trend analysis to predict errors so that the network is always available. This can be established by monitoring different things for abnormal behavior.

When a fault or event occurs, a network component will often send a notification to the network operator using a proprietary or open protocol such as SNMP, or at least write a message to its console for a console server to catch and log/page. This notification is supposed to trigger manual or automatic activities. For example, the gathering of more data to identify the nature and severity of the problem or to bring backup equipment on-line.

Fault logs are one input used to compile statistics to determine the provided service level of individual network elements, as well as sub-networks or the whole network. They are also used to determine apparently fragile network components that require further attention.

The leading Fault Management systems are EMC Smarts, CA Spectrum, HP Software, NetIQ, IBM Tivoli Netcool, TTI Telecom Netrac, CA Clarity, Objective Systems Integrators NETeXPERT etc. Fault Isolation tools like Delphi are also available, which are basically used to isolate the fault in any telecom network.

Configuration management

The goals of configuration management include:

- to gather and store configurations from network devices (this can be done locally or remotely).
- to simplify the configuration of the device
- to track changes that are made to the configuration
- to configure ('provision') circuits or paths through non-switched networks

Accounting management

Accounting is often referred to as billing management. The goal is to gather usage statistics for users.

Using the statistics the users can be billed and usage quota can be enforced.

Examples:

- Disk usage
- Link utilization
- CPU time

RADIUS, TACACS and Diameter are examples of protocols commonly used for accounting.

For non-billed networks, "administration" replaces "accounting". The goals of administration are to administer the set of authorized users by establishing users, passwords, and permissions, and to administer the operations of the equipment such as by performing software backup and synchronization.

Performance management

Performance management enables the manager to prepare the network for the future, as well as to determine the efficiency of the current network, for example, in relation to the investments done to set it up. The network performance addresses the throughput, percentage utilization, error rates and response times areas.

By collecting and analysing performance data, the network health can be monitored. Trends can indicate capacity or reliability issues before they become service affecting.

Performance thresholds can be set in order to trigger an alarm. The alarm would be handled by the normal fault management process (see above). Alarms vary depending upon the severity.

Security management

Security management is the process of controlling access to assets in the network. Data security can be achieved mainly with authentication and encryption. Authorization to it configured with OS and DBMS access control settings.

Chapter 15

Simple Network Management Protocol and Java Management Extensions

Simple Network Management Protocol

Simple Network Management Protocol (**SNMP**) is an "Internet-standard protocol for managing devices on IP networks. Devices that typically support SNMP include routers, switches, Servers, workstations, printers, modem tracks, and more.”. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention. SNMP is a component of the Internet Protocol Suite as defined by the Internet Engineering Task Force (IETF). It consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects.

SNMP exposes management data in the form of variables on the managed systems, which describe the system configuration. These variables can then be queried (and sometimes set) by managing applications.

Overview and basic concepts

In typical SNMP use, one or more administrative computers called managers have the task of monitoring or managing a group of hosts or devices on a computer network. Each managed system executes, at all times, a software component called an *agent* which reports information via SNMP to the manager.

Essentially, SNMP agents expose management data on the managed systems as variables. The protocol also permits active management tasks, such as modifying and applying a new configuration through remote modification of these variables. The variables accessible via SNMP are organized in hierarchies. These hierarchies, and other metadata (such as type and description of the variable), are described by Management Information Bases (MIBs).

An SNMP-managed network consists of three key components:

- Managed device

- Agent — software which runs on managed devices
- Network management system (NMS) — software which runs on the manager

A *managed device* is a network node that implements an SNMP interface that allows unidirectional (read-only) or bidirectional access to node-specific information. Managed devices exchange node-specific information with the NMSs. Sometimes called network elements, the managed devices can be any type of device, including, but not limited to, routers, access servers, switches, bridges, hubs, IP telephones, IP video cameras, computer hosts, and printers.

An *agent* is a network-management software module that resides on a managed device. An agent has local knowledge of management information and translates that information to or from an SNMP specific form.

A *network management system* (NMS) executes applications that monitor and control managed devices. NMS's provide the bulk of the processing and memory resources required for network management. One or more NMSs may exist on any managed network.

Management information base (MIB)

SNMP itself does not define which information (which variables) a managed system should offer. Rather, SNMP uses an extensible design, where the available information is defined by management information bases (MIBs). MIBs describe the structure of the management data of a device subsystem; they use a hierarchical namespace containing object identifiers (OID). Each OID identifies a variable that can be read or set via SNMP. MIBs use the notation defined by ASN.1.

Protocol details

SNMP operates in the Application Layer of the Internet Protocol Suite (Layer 7 of the OSI model). The SNMP agent receives requests on UDP port 161. The manager may send requests from any available source port to port 161 in the agent. The agent response will be sent back to the source port on the manager. The manager receives notifications (*Traps* and *InformRequests*) on port 162. The agent may generate notifications from any available port.

SNMPv1 specifies five core protocol data units (PDUs). Two other PDUs, *GetBulkRequest* and *InformRequest* were added in SNMPv2 and carried over to SNMPv3.

All SNMP PDUs are constructed as follows:

IP header	UDP header	version	community	PDU-type	request-id	error-status	error-index	variable bindings

The seven SNMP protocol data units (PDUs) are as follows:

GetRequest

A manager-to-agent request to retrieve the value of a variable or list of variables. Desired variables are specified in variable bindings (values are not used). Retrieval of the specified variable values is to be done as an atomic operation by the agent. A *Response* with current values is returned.

SetRequest

A manager-to-agent request to change the value of a variable or list of variables. Variable bindings are specified in the body of the request. Changes to all specified variables are to be made as an atomic operation by the agent. A *Response* with (current) new values for the variables is returned.

GetNextRequest

A manager-to-agent request to discover available variables and their values. Returns a *Response* with variable binding for the lexicographically next variable in the MIB. The entire MIB of an agent can be walked by iterative application of *GetNextRequest* starting at OID 0. Rows of a table can be read by specifying column OIDs in the variable bindings of the request.

GetBulkRequest

Optimized version of *GetNextRequest*. A manager-to-agent request for multiple iterations of *GetNextRequest*. Returns a *Response* with multiple variable bindings walked from the variable binding or bindings in the request. PDU specific *non-repeaters* and *max-repetitions* fields are used to control response behavior. *GetBulkRequest* was introduced in SNMPv2.

Response

Returns variable bindings and acknowledgement from agent to manager for *GetRequest*, *SetRequest*, *GetNextRequest*, *GetBulkRequest* and *InformRequest*. Error reporting is provided by *error-status* and *error-index* fields. Although it was used as a response to both gets and sets, this PDU was called *GetResponse* in SNMPv1.

Trap

Asynchronous notification from agent to manager. Includes current *sysUpTime* value, an OID identifying the type of trap and optional variable bindings. Destination addressing for traps is determined in an application specific manner typically through trap configuration variables in the MIB. The format of the trap message was changed in SNMPv2 and the PDU was renamed *SNMPv2-Trap*.

InformRequest

Acknowledged asynchronous notification from manager to manager. This PDU uses the same format as the SNMPv2 version of *Trap*. Manager-to-manager notifications were already possible in SNMPv1 (using a *Trap*), but as SNMP commonly runs over UDP where delivery is not assured and dropped packets are not reported, delivery of a *Trap* was not guaranteed. *InformRequest* fixes this by sending back an acknowledgement on receipt. Receiver replies with *Response* parroting all information in the *InformRequest*. This PDU was introduced in SNMPv2.

Development and usage

Version 1

SNMP version 1 (SNMPv1) is the initial implementation of the SNMP protocol. SNMPv1 operates over protocols such as User Datagram Protocol (UDP), Internet Protocol (IP), OSI Connectionless Network Service (CLNS), AppleTalk Datagram-Delivery Protocol (DDP), and Novell Internet Packet Exchange (IPX). SNMPv1 is widely used and is the de facto network-management protocol in the Internet community.

The first RFCs for SNMP, now known as SNMPv1, appeared in 1988:

- RFC 1065 — Structure and identification of management information for TCP/IP-based internets
- RFC 1066 — Management information base for network management of TCP/IP-based internets
- RFC 1067 — A simple network management protocol

These protocols were obsoleted by:

- RFC 1155 — Structure and identification of management information for TCP/IP-based internets
- RFC 1156 — Management information base for network management of TCP/IP-based internets
- RFC 1157 — A simple network management protocol

After a short time, RFC 1156 (MIB-1) was replaced by more often used:

- RFC 1213 — Version 2 of management information base (MIB-2) for network management of TCP/IP-based internets

Version 1 has been criticized for its poor security. Authentication of clients is performed only by a "community string", in effect a type of password, which is transmitted in cleartext. The '80s design of SNMP V1 was done by a group of collaborators who viewed the officially sponsored OSI/IETF/NSF (National Science Foundation) effort (HEMS/CMIS/CMIP) as both unimplementable in the computing platforms of the time as well as potentially unworkable. SNMP was approved based on a belief that it was an interim protocol needed for taking steps towards large scale deployment of the Internet and its commercialization. In that time period Internet-standard authentication/security was both a dream and discouraged by focused protocol design groups.

Version 2

SNMPv2 (RFC 1441–RFC 1452), revises version 1 and includes improvements in the areas of performance, security, confidentiality, and manager-to-manager communications. It introduced *GetBulkRequest*, an alternative to iterative GetNextRequests for retrieving large amounts of management data in a single request. However, the new party-based security system in SNMPv2, viewed by many as overly complex, was not widely accepted.

Community-Based Simple Network Management Protocol version 2, or *SNMPv2c*, is defined in RFC 1901–RFC 1908. In its initial stages, this was also informally known as *SNMPv1.5*. SNMPv2c comprises SNMPv2 *without* the controversial new SNMP v2 security model, using instead the simple community-based security scheme of SNMPv1. While officially only a "Draft Standard", this is widely considered the *de facto* SNMPv2 standard.

User-Based Simple Network Management Protocol version 2, or *SNMPv2u*, is defined in RFC 1909–RFC 1910. This is a compromise that attempts to offer greater security than SNMPv1, but without incurring the high complexity of SNMPv2. A variant of this was commercialized as *SNMP v2**, and the mechanism was eventually adopted as one of two security frameworks in SNMP v3.

SNMPv1 & SNMPv2c interoperability

As presently specified, SNMPv2 is incompatible with SNMPv1 in two key areas: message formats and protocol operations. SNMPv2c messages use different header and protocol data unit (PDU) formats from SNMPv1 messages. SNMPv2c also uses two protocol operations that are not specified in SNMPv1. Furthermore, RFC 2576 defines two possible SNMPv1/v2c coexistence strategies: proxy agents and bilingual network-management systems.

Proxy agents

A SNMPv2 agent can act as a proxy agent on behalf of SNMPv1 managed devices, as follows:

- A SNMPv2 NMS issues a command intended for a SNMPv1 agent.
- The NMS sends the SNMP message to the SNMPv2 proxy agent.
- The proxy agent forwards `Get`, `GetNext`, and `Set` messages to the SNMPv1 agent unchanged.
- GetBulk messages are converted by the proxy agent to `GetNext` messages and then are forwarded to the SNMPv1 agent.

The proxy agent maps SNMPv1 trap messages to SNMPv2 trap messages and then forwards them to the NMS.

Bilingual network-management system

Bilingual SNMPv2 network-management systems support both SNMPv1 and SNMPv2. To support this dual-management environment, a management application in the bilingual NMS must contact an agent. The NMS then examines information stored in a local database to determine whether the agent supports SNMPv1 or SNMPv2. Based on the information in the database, the NMS communicates with the agent using the appropriate version of SNMP.

Version 3

Although SNMPv3 makes no changes to the protocol aside from the addition of cryptographic security, its developers have managed to make things look much different by introducing new textual conventions, concepts, and terminology.

SNMPv3 primarily added security and remote configuration enhancements to SNMP.

Security has been the biggest weakness of SNMP since the beginning. Authentication in SNMP Versions 1 and 2 amounts to nothing more than a password (community string) sent in clear text between a manager and agent. Each SNMPv3 message contains security parameters which are encoded as an octet string. The meaning of these security parameters depends on the security model being used.

SNMPv3 provides important security features:

- Confidentiality - Encryption of packets to prevent snooping by an unauthorized source.
- Integrity - Message integrity to ensure that a packet has not been tampered with in transit.
- Authentication - to verify that the message is from a valid source.

As of 2004 the IETF recognizes *Simple Network Management Protocol version 3* as defined by RFC 3411–RFC 3418 (also known as STD0062) as the current standard version of SNMP. The IETF has designated SNMPv3 a full Internet standard, the highest maturity level for an RFC. It considers earlier versions to be obsolete (designating them "Historic").

In practice, SNMP implementations often support multiple versions: typically SNMPv1, SNMPv2c, and SNMPv3.

Applications

Environmental monitoring

Server, rack and appliance operating temperatures and room humidity could be monitored remotely for SNMP-enabled HVAC devices.

Implementation issues

SNMP implementations vary across platform vendors. In some cases, SNMP is an added feature, and is not taken seriously enough to be an element of the core design. Some major equipment vendors tend to over-extend their proprietary command line interface (CLI) centric configuration and control systems.

SNMP's seemingly simple tree structure and linear indexing may not always be understood well enough within the internal data structures that are elements of a platform's basic design. As a result, processing SNMP queries on certain data sets may result in higher CPU utilization than necessary. One example of this would be large routing tables, such as BGP or IGP.

Resource indexing

Modular devices may dynamically increase or decrease their SNMP indices (aka instances) whenever slotted hardware is added or removed. Although this is most common with hardware, virtual interfaces have the same effect. Index values are typically assigned at boot time and remain fixed until the next reboot. Hardware or virtual entities added while the device is 'live' may have their indexes assigned at the end of the existing range and possibly reassigned at the next reboot. Network inventory and monitoring tools need to have the device update capability by properly reacting to the cold start trap from the device reboot in order to avoid corruption and mismatch of polled data.

Index assignments for an SNMP device instance may change from poll to poll mostly as a result of changes initiated by the system admin. If information is needed for a particular interface, it is imperative to determine the SNMP index before retrieving the data needed. Generally, a description table like ifDescr will map a user friendly name like Serial 0/1 (Blade 0, port 1) to a SNMP index.

Security implications

- SNMP versions 1 and 2c are subject to packet sniffing of the clear text community string from the network traffic, because they do not implement encryption.
- All versions of SNMP are subject to brute force and dictionary attacks for guessing the community strings, authentication strings, authentication keys, encryption strings, or encryption keys, because they do not implement a challenge-response handshake. Entropy is an important consideration when selecting keys, passwords and/or algorithms.
- Although SNMP works over TCP and other protocols, it is most commonly used over UDP that is connectionless and vulnerable to IP spoofing attacks. Thus, all versions are subject to bypassing device access lists that might have been implemented to restrict SNMP access, though SNMPv3's other security mechanisms should prevent a successful attack.
- SNMP's powerful configuration (write) capabilities are not being fully utilized by many vendors, partly due to lack of security in SNMP versions before SNMPv3 and partly due to the fact that many devices simply are not capable of being configured via individual MIB object changes.
- SNMP tops the list of the SANS Institute's Common Default Configuration Issues with the issue of default SNMP community strings set to ‘public’ and ‘private’ and was number ten on the SANS Top 10 Most Critical Internet Security Threats for the year 2000.

Autodiscovery

SNMP by itself is simply a protocol for collecting and organizing information. Most toolsets implementing SNMP offer some form of discovery mechanism, a standardized collection of data common to most platforms and devices, to get a new user or implementor started. One of these features is often a form of automatic discovery, where new devices discovered in the network are polled automatically. For SNMPv1 and SNMPv2c, this presents a security risk, in that your SNMP read communities will be broadcast in cleartext to the target device. While security requirements and risk profiles vary from organization to organization, care should be taken when using a feature like this, with special regard to common environments such as mixed-tenant datacenters, server hosting and colocation facilities, and similar environments.

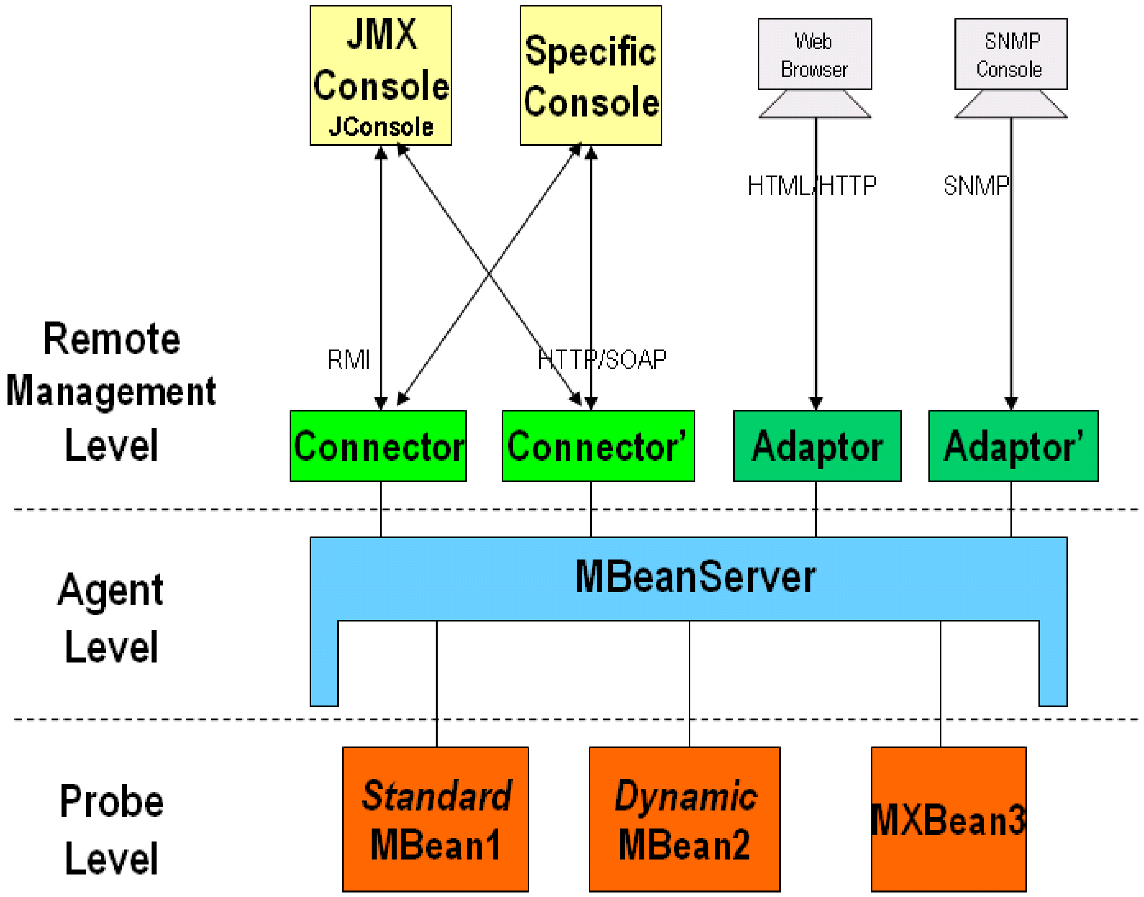
Java Management Extensions

Java Management Extensions (**JMX**) is a Java technology that supplies tools for managing and monitoring applications, system objects, devices (e. g. printers) and service oriented networks. Those resources are represented by objects called MBeans (for *Managed Bean*). In the API, classes can be dynamically loaded and instantiated. Managing and monitoring applications can be designed and developed by Java Dynamic Management Kit.

JMX 1.0, 1.1 and 1.2 were defined by JSR 003 of the Java Community Process. As of 2006, JMX 2.0 is being developed under JSR 255. The JMX Remote API 1.0 for remote management and monitoring is specified by JSR 160. An extension of the JMX Remote API for Web Services is being developed under JSR 262.

Adopted early on by the J2EE community, JMX has been a part of J2SE since version 5.0. It is a trademark of Sun Microsystems, Inc.

Architecture



JMX architecture.

JMX is based on a 3-level architecture:

- The *Probe* level contains the probes (called MBeans) instrumenting the resources. Also called the *Instrumentation* level.
- The *Agent* level, or MBeanServer, is the core of JMX. It is an intermediary between the MBean and the applications.
- The *Remote Management* level enables remote applications to access the MBeanServer through Connectors and Adaptors. A connector provides full remote access to the MBeanServer API using various communication frameworks (RMI, IIOP, JMS, WS-* …), while an adaptor adapts the API to another protocol (SNMP, …) or to Web-based GUI (HTML/HTTP, WML/HTTP, …).

Applications can be generic consoles (such as JConsole and MC4J), or domain-specific (monitoring) applications. External applications can interact with the MBeans through the use of JMX connectors and protocol adapters. Connectors are used to connect an agent with a remote JMX-enabled management application. This form of communication involves a connector in the JMX agent and a connector client in the management application.

Protocol adapters provide a management view of the JMX agent through a given protocol. Management applications that connect to a protocol adapter are usually specific to the given protocol.

Managed Bean

A **managed bean** - sometimes simply referred to as an *MBean* - is a type of JavaBean, created with dependency injection. Managed Beans are particularly used in the Java Management Extensions technology. But, with the Java EE 6 specification provides for a more detailed meaning of a managed bean.

The MBean represents a resource running in the Java virtual machine, such as an application or a Java EE technical service (transactional monitor, JDBC driver, etc.). They can be used for getting and setting applications configuration (pull), for collecting statistics (pull) (e.g. performance, resources usage, problems) and notifying events (push) (e.g. faults, state changes).

Java EE 6 provides that a managed bean is a bean that is implemented by a Java class, which is called its bean class. A top-level Java class is a managed bean if it is defined to be a managed bean by any other Java EE technology specification (for example, the JavaServer Faces technology specification), or if it meets all of the following conditions:

1. It is not a non-static inner class.
2. It is a concrete class, or is annotated @Decorator.
3. It is not annotated with an EJB component-defining annotation or declared as an EJB bean class in ejb-jar.xml.

No special declaration, such as an annotation, is required to define a managed bean.

An MBean can notify the MBeanServer of its internal changes (for the attributes) by implementing the javax.management.NotificationEmitter. The application interested in the MBean's changes registers a listener (javax.management.NotificationListener) to the MBeanServer. Note that JMX does not guarantee that all notifications will be received by the listeners.

Types

There are two basic types of MBean:

- *Standard MBeans* implement a business interface containing setters and getters for the attributes and the operations (i.e., methods).
- *Dynamic MBeans* implement the javax.management.DynamicMBean interface which provides a way to list the attributes and operations, and to get and set the attribute values.

Additional types are *Open MBeans*, *Model MBeans* and *Monitor MBeans*. *Open MBeans* are dynamic MBeans that rely on the basic data types. They are self-explanatory and more user friendly. *Model MBeans* are dynamic MBeans that can be configured during runtime. A generic MBean class is also provided for dynamically configuring the resources during program runtime.

An MXBean (*Platform MBean*) is a special type of MBean that reifies Java Virtual Machine subsystems such as memory pools, garbage collection, multi-threading, JIT compilation, etc.

An MLet (*Management applet*) is a utility MBean to load, instantiate and register MBeans in the MBeanServer from a XML description. The format of the XML descriptor is:

```
<MLET CODE = ''class'' | OBJECT = ''serfile''
  ARCHIVE = ''archiveList''
  [CODEBASE = ''codebaseURL'']
  [NAME = ''objectName'']
  [VERSION = ''version'']
>
  [arglist]
</MLET>
```

Support

JMX is supported at various levels by different vendors:

- JMX is supported by Java application servers such as OpenCloud Rhino Application Server , JBoss, JOnAS, WebSphere Application Server, WebLogic, SAP Netweaver Application Server, Oracle Application Server 10g and Sun Java System Application Server.

- Systems management tools that support the protocol include IBM Director, HP OpenView, Zyrion, Zenoss, Zabbix, Hyperic, Empirix OneSight and GroundWork Monitor.
- JMX is also supported by servlet containers such as Apache Tomcat.
- MX4J is Open Source JMX for Enterprise Computing.
- jManage is an open source enterprise-grade JMX Console with web and command-line interfaces.
- MC4J is an open source visual console for connecting to servers supporting JMX

Chapter 16

Load Balancing (Computing) and Network Administrator

Load balancing (computing)

In networking, **load balancing** is a technique to distribute workload evenly across two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The load balancing service is usually provided by a dedicated program or hardware device (such as a multilayer switch or a DNS server).

It is commonly used to mediate internal communications in computer clusters, especially high-availability clusters. If the load is more on a server, then the secondary server takes some load while the other is still processing requests.

Internet-based services

One of the most common applications of load balancing is to provide a single Internet service from multiple servers, sometimes known as a server farm. Commonly, load-balanced systems include popular web sites, large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, Network News Transfer Protocol (NNTP) servers and Domain Name System (DNS) servers.

For Internet services, the load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting backend servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports.

Some load balancers provide a mechanism for doing something special in the event that all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying a message regarding the outage.

An alternate method of load balancing, which does not necessarily require a dedicated software or hardware node, is called **round robin DNS**. In this technique, multiple IP addresses are associated with a single domain name; clients themselves are expected to choose which server to connect to. Unlike the use of a dedicated load balancer, this technique exposes to clients the existence of multiple backend servers. The technique has other advantages and disadvantages, depending on the degree of control over the DNS server and the granularity of load balancing desired.

Another technique for load-balancing using DNS, which is far more intelligent than the simple "round robin", is to delegate `www.example.org` as a sub-domain whose zone will be served out by each of the same servers that are serving the web site. This technique works particularly well where individual servers are spread around the Internet. For example,

```
one.example.org A 1.1.1.1
two.example.org A 2.2.2.2
www.example.org NS one.example.org
www.example.org NS two.example.org
```

However, the zone file for `www.example.org` on each server will be different such that each server will give out its own IP Address as the A-record. On "one" the zone file for `www.example.org` will say:

```
@ in a 1.1.1.1
```

On "two" the same zone file will say:

```
@ in a 2.2.2.2
```

This way, if a server is down, its DNS will not respond and so the web service will not receive any traffic. Also if the line to one server becomes congested the unreliability of DNS will ensure less HTTP traffic will reach that server, further the DNS response that gets back to the resolver the quickest will nearly always be from the network closest server, ensuring geo-sensitive load-balancing. A short TTL on the A-record will also help to ensure traffic is quickly diverted if a server goes down. Consideration must be given the possibility that this technique may cause individual clients to switch between individual servers mid-session.

A variety of scheduling algorithms are used by load balancers to determine which backend server to send a request to. Simple algorithms include random choice or round robin. More sophisticated load balancers may take into account additional factors, such as a server's reported load, recent response times, up/down status (determined by a monitoring poll of some kind), number of active connections, geographic location, capabilities, or how much traffic it has recently been assigned. High-performance systems may use multiple layers of load balancing.

In addition to using dedicated hardware load balancers, software-only solutions are available, including open source options. Examples of the latter include the Apache web server's mod_proxy_balancer extension and the Pound reverse proxy and load balancer.

In a Multitier architecture, terminology for designs behind a load balancer or network dispatcher may include **Bowties** and **Stovepipes**. A stovepipe presents a situation such that a transaction that is dispatched at a top tier follows a static path through the stack of devices and software behind the load balancer to its final destination. Alternatively, if Bowties are used, at each tier the transaction could take one of many paths after being serviced by the applications at a particular tier. Network diagrams with transaction flows resemble Stovepipes or Bowties, or hybrid architectures based on need at each tier.

Persistence

An important issue when operating a load-balanced service is how to handle information that must be kept across the multiple requests in a user's session. If this information is stored locally on one backend server, then subsequent requests going to different backend servers would not be able to find it. This might be cached information that can be recomputed, in which case load-balancing a request to a different backend server just introduces a performance issue...

One solution to the session data issue is to send all requests in a user session consistently to the same backend server. This is known as "persistence" or "stickiness". A significant downside to this technique is its lack of automatic failover: if a backend server goes down, its per-session information becomes inaccessible, and any sessions depending on it are lost. The same problem is usually relevant to central database servers; even if web servers are "stateless" and not "sticky", the central database is (see below).

Assignment to a particular server might be based on a username, client IP address, or random assignment. Owing to DHCP, Network Address Translation, and web proxies, the client's IP address may change across requests, and so this method can be somewhat unreliable. Random assignments must be remembered by the load balancer, which creates a storage burden. If the load balancer is replaced or fails, this information can be lost, and assignments may need to be deleted after a timeout period or during periods of high load to avoid exceeding the space available for the assignment table. The random assignment method also requires that clients maintain some state, which can be a problem, for example when a web browser has disabled storage of cookies. Sophisticated load balancers use multiple persistence techniques to avoid some of the shortcomings of any one method.

Another solution is to keep the per-session data in a database. Generally this is bad for performance since it increases the load on the database: the database is best used to store information less transient than per-session data. To prevent a database from becoming a single point of failure, and to improve scalability, the database is often replicated across multiple machines, and load balancing is used to spread the query load across those replicas. Microsoft's ASP.net State Server technology is an example of a session database. All servers in a web farm store their session data on State Server and any server in the farm can retrieve the data.

Fortunately there are more efficient approaches. In the very common case where the client is a web browser, per-session data can be stored in the browser itself. One technique is to use a browser cookie, suitably time-stamped and encrypted. Another is URL rewriting. Storing session data on the client is generally the preferred solution: then the load balancer is free to pick any backend server to handle a request. However, this method of state-data handling is not really suitable for some complex business logic scenarios, where session state payload is very big or recomputing it with every request on a server is not feasible, and URL rewriting has major security issues, since the end-user can easily alter the submitted URL and thus change session streams.

Load balancer features

Hardware and software load balancers can come with a variety of special features.

- **Asymmetric load:** A ratio can be manually assigned to cause some backend servers to get a greater share of the workload than others. This is sometimes used as a crude way to account for some servers being faster than others.
- **Priority activation:** When the number of available servers drops below a certain number, or load gets too high, standby servers can be brought online
- **SSL Offload and Acceleration:** SSL applications can be a heavy burden on the resources of a Web Server, especially on the CPU and the end users may see a slow response (or at the very least the servers are spending a lot of cycles doing things they weren't designed to do). To resolve these kinds of issues, a Load Balancer capable of handling SSL Offloading in specialized hardware may be used. When Load Balancers are taking the SSL connections, the burden on the Web Servers is reduced and performance will not degrade for the end users.
- **Distributed Denial of Service (DDoS) attack protection:** load balancers can provide features such as SYN cookies and delayed-binding (the back-end servers don't see the client until it finishes its TCP handshake) to mitigate SYN flood attacks and generally offload work from the servers to a more efficient platform.
- **HTTP compression:** reduces amount of data to be transferred for HTTP objects by utilizing gzip compression available in all modern web browsers
- **TCP offload:** different vendors use different terms for this, but the idea is that normally each HTTP request from each client is a different TCP connection. This feature utilizes HTTP/1.1 to consolidate multiple HTTP requests from multiple clients into a single TCP socket to the back-end servers.
- **TCP buffering:** the load balancer can buffer responses from the server and spoon-feed the data out to slow clients, allowing the server to move on to other tasks.

- **Direct Server Return:** an option for asymmetrical load distribution, where request and reply have different network paths.
- **Health checking:** the balancer will poll servers for application layer health and remove failed servers from the pool.
- **HTTP caching:** the load balancer can store static content so that some requests can be handled without contacting the web servers.
- **Content Filtering:** some load balancers can arbitrarily modify traffic on the way through.
- **HTTP security:** some load balancers can hide HTTP error pages, remove server identification headers from HTTP responses, and encrypt cookies so end users can't manipulate them.
- **Priority queuing:** also known as rate shaping, the ability to give different priority to different traffic.
- **Content aware switching:** most load balancers can send requests to different servers based on the URL being requested.
- **Client authentication:** authenticate users against a variety of authentication sources before allowing them access to a website.
- **Programmatic traffic manipulation:** at least one load balancer allows the use of a scripting language to allow custom load balancing methods, arbitrary traffic manipulations, and more.
- **Firewall:** direct connections to backend servers are prevented, for network security reasons
- **Intrusion Prevention System:** offer application layer security in addition to network/transport layer offered by firewall security.

In telecommunications

Load balancing can be useful when dealing with redundant communications links. For example, a company may have multiple Internet connections ensuring network access even if one of the connections should fail.

A failover arrangement would mean that one link is designated for normal use, while the second link is used only if the first one fails.

With load balancing, both links can be in use all the time. A device or program decides which of the available links to send packets along, being careful not to send packets along any link if it has failed. The ability to use multiple links simultaneously increases the available bandwidth.

Major telecommunications companies have multiple routes through their networks or to external networks. They use more sophisticated load balancing to shift traffic from one path to another to avoid network congestion on any particular link, and sometimes to minimize the cost of transit across external networks or improve network reliability.

Relationship with failover

Load balancing is often used to implement failover — the continuation of a service after the failure of one or more of its components. The components are monitored continually (e.g., web servers may be monitored by fetching known pages), and when one becomes non-responsive, the load balancer is informed and no longer sends traffic to it. And when a component comes back on line, the load balancer begins to route traffic to it again. For this to work, there must be at least one component in excess of the service's capacity. This is much less expensive and more flexible than failover approaches where a single "live" component is paired with a single "backup" component that takes over in the event of a failure. Some types of RAID systems can also utilize hot spare for a similar effect.

Network administrator

A **network administrator** is a person responsible for the maintenance of computer hardware and software that comprises a computer network. This normally includes deploying, configuring, maintaining and monitoring active network equipment. A related role is that of the network specialist, or network analyst, who concentrates on network design and security.

The network administrator (or "network admin") is usually the level of technical/network staff in an organization and will rarely be involved with direct user support. The network administrator will concentrate on the overall integrity of the network, server deployment, security, and ensuring that the network connectivity throughout a company's LAN/WAN infrastructure is on par with technical considerations at the network level of an organization's hierarchy. Network administrators are considered tier 3 support personnel that only work on break/fix issues that could not be resolved at the tier 1 (helpdesk) or tier 2 (desktop/network technician) levels.

Depending on the company, the Network Administrator may also design and deploy networks. However, these tasks may be assigned to a network engineer should one be available to the company. A network engineer designs, implements, and/or troubleshoots computer networks. In general, a network engineer will not regularly perform system administration tasks, but will instead concentrate on high-level network related duties such as network architecture, network design, choosing of network devices, and network policies. Network engineers will rarely be involved with direct user support, as they are normally placed as tier three support.

The actual role of the Network Administrator will vary from company to company, but will commonly include activities and tasks such as network address assignment, assignment of routing protocols and routing table configuration as well as configuration of authentication and authorization – directory services. It often includes maintenance of network facilities in individual machines, such as drivers and settings of personal computers as well as printers and such. It sometimes also includes maintenance of certain network servers: file servers, VPN gateways, intrusion detection systems, etc.

Network specialists and analysts concentrate on the network design and security, particularly troubleshooting and/or debugging network-related problems. Their work can also include the maintenance of the network's authorization infrastructure, as well as network backup systems.

The administrator is responsible for the security of the network and for assigning IP addresses to the devices connected to the networks. Assigning IP addresses gives the subnet administrator some control over the personnel who connect to the subnet. It also helps ensure that the administrator knows each system that is connected and who is personally responsible for the system.

Duties of a network administrator

Many organizations use a three tier support staff solution, with tier one (help desk) personnel handling the initial calls, tier two (technicians and pc support analysts) and tier three (network administrators). Most of those organizations follow a fixed staffing ratio, and being a network administrator is either the top job, or next to top job, within the technical support department.

Network administrators are responsible for making sure that the computer hardware and network infrastructure for an IT organization is properly maintained. They are deeply involved in the procurement of new hardware (For example: Does it meet existing standardization requirements? Does it do the job required?), rolling out new software installs, maintaining the disk images for new computer installs (usually by having a standardized OS and application install), making sure that licenses are paid for and up to date for software that need it, maintaining the standards for server installations and applications, and monitoring the performance of the network, checking for security breaches, poor data management practices and more.

Most network administrator positions require a breadth of technical knowledge and the ability to learn the ins and outs of new networking and server software packages quickly. While designing and drafting a network is usually the job of a network engineer, many organizations roll that function into a network administrator position as well.

One of the chief jobs of a network administrator is connectivity. Network administrators are in charge of making sure that connectivity works for all users in their organization, and for making sure that data security for connections to the internet is properly handled. (For network administrators doing security aspects, this can be a full time job.)

Trouble tickets work their way through the help desk, then through the analyst level support, before reaching the network administrator's level. As a result, in their day-to-day operations, network administrators should not be dealing directly with end users as a routine function. Most of their jobs should be on scheduling and implementing routine maintenance tasks, updating disaster prevention programs, making sure that network backups are run and doing test restores to make sure that those restores are sound.

Chapter 17

Management Information Base and NETCONF

Management information base

A **management information base (MIB)** is a virtual database used for managing the entities in a communications network. Most often associated with the Simple Network Management Protocol (SNMP), the term is also used more generically in contexts such as in OSI/ISO Network management model. While intended to refer to the complete collection of management information available on an entity, it is often used to refer to a particular subset, more correctly referred to as MIB-module.

Objects in the MIB are defined using a subset of Abstract Syntax Notation One (ASN.1) called "Structure of Management Information Version 2 (SMIv2)" RFC 2578.The software that performs the parsing is a MIB compiler.

The database is hierarchical (tree-structured) and entries are addressed through object identifiers. Internet documentation RFCs discuss MIBs, notably RFC 1155, "Structure and Identification of Management Information for TCP/IP based internets", and its two companions, RFC 1213, "Management Information Base for Network Management of TCP/IP-based internets", and RFC 1157, "A Simple Network Management Protocol".

Abstract Syntax Notation One (ASN.1)

In telecommunications and computer networking, Abstract Syntax Notation One (ASN.1) is a standard and flexible notation that describes data structures for representing, encoding, transmitting, and decoding data. It provides a set of formal rules for describing the structure of objects that are independent of machine-specific encoding techniques and is a precise, formal notation that removes ambiguities.

ASN.1 is a joint ISO and ITU-T standard, originally defined in 1984 as part of CCITT X.409:1984. ASN.1 moved to its own standard, X.208, in 1988 due to wide applicability. The substantially revised 1995 version is covered by the X.680 series.

An adapted subset of ASN.1, Structure of Management Information (SMI), is specified in SNMP to define sets of related MIB objects; these sets are termed MIB modules.

MIB hierarchy

The MIB hierarchy can be depicted as a tree with a nameless root, the levels of which are assigned by different organizations. The top-level MIB OIDs belong to different standards organizations, while lower-level object IDs are allocated by associated organizations. This model permits management across all layers of the OSI reference model, extending into applications such as databases, email, and the Java reference model, as MIBs can be defined for all such area-specific information and operations.

A managed object (sometimes called a MIB object, an object, or a MIB) is one of any number of specific characteristics of a managed device. Managed objects are made up of one or more object instances (identified by their OIDs), which are essentially variables.

Two types of managed objects exist:

- Scalar objects define a single object instance.
- Tabular objects define multiple related object instances that are grouped in MIB tables.

An example of a managed object is *`atInput`*, which is a scalar object that contains a single object instance, the integer value that indicates the total number of input AppleTalk packets on a router interface.

An object identifier (or object ID or OID) uniquely identifies a managed object in the MIB hierarchy.

SNMPv1 and SMI-specific data types

The first version of the SMI (SMIv1) specifies the use of a number of SMI-specific data types, which are divided into two categories:

- Simple data types
- Application-wide data types

Simple data types

Three simple data types are defined in the SNMPv1 SMI, all of which are unique values:

- The integer data type is a signed integer in the range of -2^{31} to $2^{31}-1$.
- Octet strings are ordered sequences of 0 to 65,535 octets.
- Object IDs come from the set of all object identifiers allocated according to the rules specified in ASN.1.

Application-wide data types

The following application-wide data types exist in the SNMPv1 SMI:

- *Network addresses* represent addresses from a particular protocol family. SMIv1 supports only 32-bit (IPv4) addresses (SMIv2 uses Octet Strings to represent addresses generically, and thus are usable in SMIv1 too. SMIv1 had an explicit IPv4 address datatype.)
- *Counters* are non-negative integers that increase until they reach a maximum value and then roll over to zero. SNMPv1 specifies a counter size of 32 bits.
- *Gauges* are non-negative integers that can increase or decrease between specified minimum and maximum values. Whenever the system property represented by the gauge is outside of that range, the value of the gauge itself will vary no further than the respective maximum or minimum, as specified in RFC 2578.
- *Time ticks* represent time since some event, measured in hundredths of a second.
- *Opaques* represent an arbitrary encoding that is used to pass arbitrary information strings that do not conform to the strict data typing used by the SMI.
- *Integers* represent signed integer-valued information. This data type redefines the integer data type, which has arbitrary precision in ASN.1 but bounded precision in the SMI.
- *Unsigned integers* represent unsigned integer-valued information, which is useful when values are always non-negative. This data type redefines the integer data type, which has arbitrary precision in ASN.1 but bounded precision in the SMI.

SNMPv1 MIB tables

The SNMPv1 SMI defines highly structured tables that are used to group the instances of a tabular object (that is, an object that contains multiple variables). Tables are composed of zero or more rows, which are indexed in a way that allows SNMP to retrieve or alter an entire row with a single `Get`, `GetNext`, or `Set` command.

SMIv2 and structure of management information

The second version of the SMI (SMIv2) is described in RFC 2578 - RFC 2579. It makes certain additions and enhancements to the SMIv1-specific data types, such as including bit strings, network addresses, and counters. Bit strings are defined only in SMIv2 and comprise zero or more named bits that specify a value. Network addresses represent an address from a particular protocol family. Counters are non-negative integers that increase until they reach a maximum value and then return to zero. In SMIv1, a 32-bit counter size is specified. In SMIv2, 32-bit and 64-bit counters are defined.

SMIv2 also specifies information modules, which specify a group of related definitions. Three types of SMI information modules exist: MIB modules, compliance statements, and capability statements.

- MIB modules contain definitions of interrelated managed objects.

- Compliance statements provide a systematic way to describe a group of managed objects that must be implemented for conformance to a standard.
- Capability statements are used to indicate the precise level of support that an agent claims with respect to a MIB group. A NMS can adjust its behavior toward agents according to the capabilities statements associated with each agent.

Updating MIBs

MIBs are periodically updated to add new functionality, remove ambiguities and to fix defects. These changes are made in conformance to section 10 of RFC 2578. An example of an MIB that has been updated many times is the important set of objects that was originally defined in RFC 1213 "MIB-II". This MIB has since been split up and can be found in MIBs such as RFC 4293 "Management Information Base for the Internet Protocol (IP)", RFC 4022 "Management Information Base for the Transmission Control Protocol (TCP)", RFC 4113 "Management Information Base for the User Datagram Protocol (UDP)", RFC 2863 "The Interfaces Group MIB" and RFC 3418 "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)".

MIBs index

There are a large number of MIBs defined by both standards organizations like the IETF, private enterprises and other entities.

IETF maintained

There are 318 RFCs in the first 5000 RFCs from the IETF that contain MIBs. This list is merely a fraction of the MIBs that have been written:

- **SNMP - SMI**:RFC 1155 - Defines the Structure of Management Information (SMI)
- **MIB-I**: RFC 1156 - Historically used with CMOT , not to be used with SNMP
- **SNMPv2-SMI**: RFC 2578 - Structure of Management Information Version 2 (SMIv2)
- **MIB-II**: RFC 1213 - Management Information Base for Network Management of TCP/IP-based internets
- **SNMPv2-MIB**: RFC 3418 - Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)
- **TCP-MIB**: RFC 4022 - Management Information Base for the Transmission Control Protocol (TCP)
- **UDP-MIB**: RFC 4113 - Management Information Base for the User Datagram Protocol (UDP)
- **IP-MIB**: RFC 4293 - Management Information Base for the Internet Protocol (IP)
- **IF-MIB**: RFC 2863 - The Interfaces Group MIB
- **ENTITY-MIB**: RFC 4133 - Entity MIB (Version 3)
- **ENTITY-STATE-MIB**: RFC 4268 - Entity State MIB
- **ALARM-MIB**: RFC 3877 - Alarm Management Information Base (MIB)

- Fibre Channel
  - **FC-MGMT-MIB**: RFC 4044 Fibre Channel Management MIB
  - **FIBRE-CHANNEL-FE-MIB**: RFC 2837 Definitions of Managed Objects for the Fabric Element in Fibre Channel Standard
- **HPR-IP-MIB**: RFC 2584 - Definitions of Managed Objects for APPN/HPR in IP Networks

IEEE maintained

The IETF and IEEE have agreed to move MIBs relating to IEEE work (for example Ethernet and bridging) to their respective IEEE workgroup. This is in process and a few items are complete.

- Network bridge
  - IEEE 802.1ap-2008 consolidated the IEEE and IETF RFCs related to bridging networks into eight related MIBs.

NETCONF

The Network Configuration Protocol, **NETCONF**, is an IETF network management protocol. It was developed in the NETCONF working group and published in December 2006 as RFC 4741.

NETCONF provides mechanisms to install, manipulate, and delete the configuration of network devices. Its operations are realized on top of a simple Remote Procedure Call (RPC) layer. The NETCONF protocol uses an Extensible Markup Language (XML) based data encoding for the configuration data as well as the protocol messages. This in turn is realized on top of the transport protocol.

The NETCONF protocol can be conceptually partitioned into four layers:

```
    Layer                            Example
+-------------+      +-------------------------------------------+
|   Content   |      |     Configuration data                    |
+-------------+      +-------------------------------------------+
          |                           |
+-------------+      +-------------------------------------------+
| Operations  |      |<get-config>, <edit-config>, <notification>|
+-------------+      +-------------------------------------------+
          |                           |                     |
+-------------+      +------------------------------+       |
|     RPC     |      |    <rpc>, <rpc-reply>        |       |
+-------------+      +------------------------------+       |
          |                           |                     |
+-------------+      +-------------------------------------------+
|  Transport  |      |   BEEP, SSH, SSL, console                 |
|  Protocol   |      |                                           |
+-------------+      +-------------------------------------------+
```

Operations

Basic Operations

The base protocol includes the following protocol operations: <get>, <get-config>, <edit-config>, <copy-config>, <delete-config>, <lock>, <unlock>, <close-session>, <kill-session>.

Capabilities

Basic NETCONF functionality can be extended by the definition of NETCONF capabilities. The set of additional protocol features an implementation supports is communicated between the server and the client during the capability exchange portion of session setup. Mandatory protocol features are not included in the capability exchange since they are assumed. RFC 4741 defines a number of optional capabilities including :xpath and :validate.

A capability to support subscribing and receiving asynchronous event notifications is published in RFC 5277. It defines the <create-subscription> operation, which enables creating real-time and replay subscriptions. Notifications are then sent asynchronously using the <notification> construct. The RFC also defines the :interleave capability, which when supported with the basic :notification capability facilitates the processing of other NETCONF operations while the subscription is active.

A capability to support partial locking of the running configuration is defined in RFC 5717. This allows multiple sessions to edit non-overlapping sub-trees within the running configuration. Without this capability, the only lock available is for the entire configuration.

The working group is also working on a new capability to retrieve the schema definitions (XML Schema, Relax NG, etc) that define NETCONF content.

Transport Protocols

NETCONF defines four transport mappings

- SSH (RFC 4742), which is mandatory to implement
- SOAP (RFC 4743)
- BEEP (RFC 4744)
- TLS (RFC 5539)

Content

The content of NETCONF operations is well-formed XML. Most content is related to network management.

The NETMOD working group has completed work to define a "human-friendly" modeling language for defining the semantics of operational data, configuration data, notifications, and operations, called YANG. YANG is defined in RFC 6020, and is accompanied by the "Common YANG Data Types" found in RFC 6021.

During the summer of 2010, the NETMOD working group was re-chartered to work on core configuration models (system, interface, and routing) as well as work on compatibility with the SNMP modeling language.

History

The IETF developed SNMP in the late 1980s and it proved to be a very popular network management protocol. In the early part of the 21st century it became apparent that in spite of what was originally intended, SNMP was not being used to configure network equipment, but was mainly being used for network monitoring. In 2002, the Internet Architecture Board and key members of the IETF's network management community got together with network operators to discuss the situation. The results of this meeting are documented in RFC 3535. It turned out that operators were primarily using proprietary Command Line Interfaces (CLI) to configure their boxes. This had a number of features that the operators liked, including the fact that it was text-based, as opposed to the BER-encoded SNMP. In addition, many equipment vendors did not provide the option to completely configure their devices via SNMP. As operators generally liked to write scripts to help manage their boxes, they did find the CLI lacking in a number of ways. Most notably was the unpredictable nature of the output. The content and formatting of output was prone to change in unpredictable ways.

Around this same time, Juniper Networks had been using an XML-based network management approach. This was brought to the IETF and shared with the broader community.

Collectively, these two events led the IETF to the creation of a protocol which it hopes will better align with the needs of network operators and equipment vendors.

Chapter 18

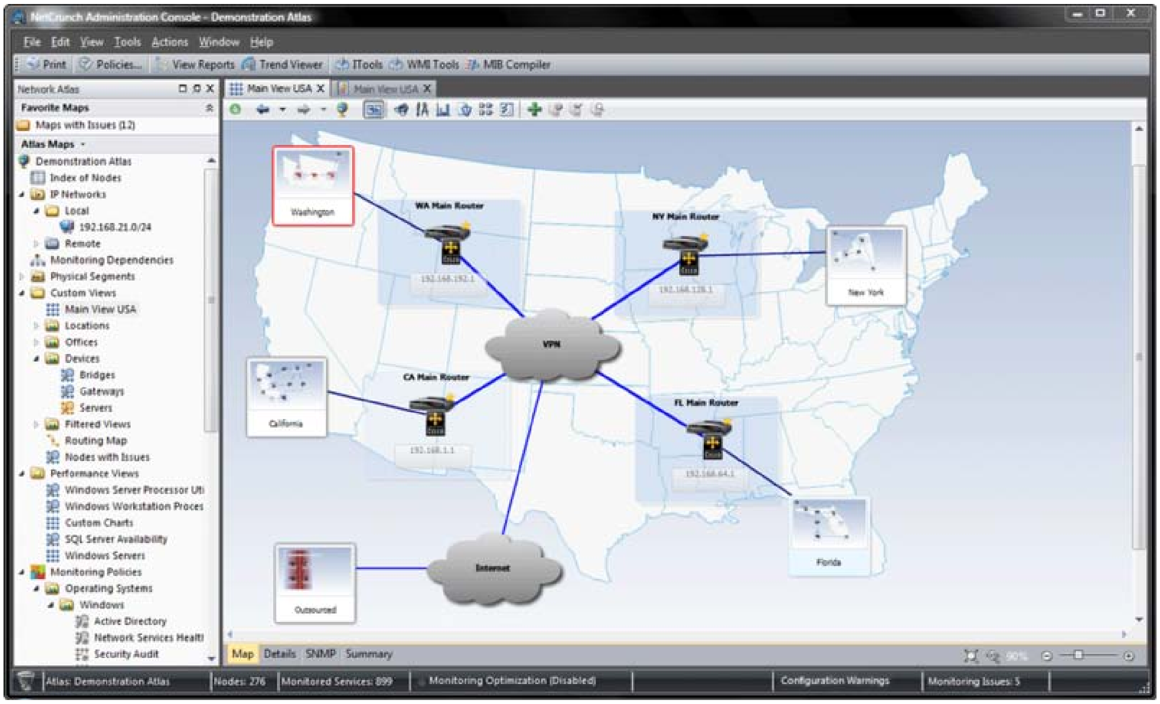
NetCrunch and Netcat

NetCrunch

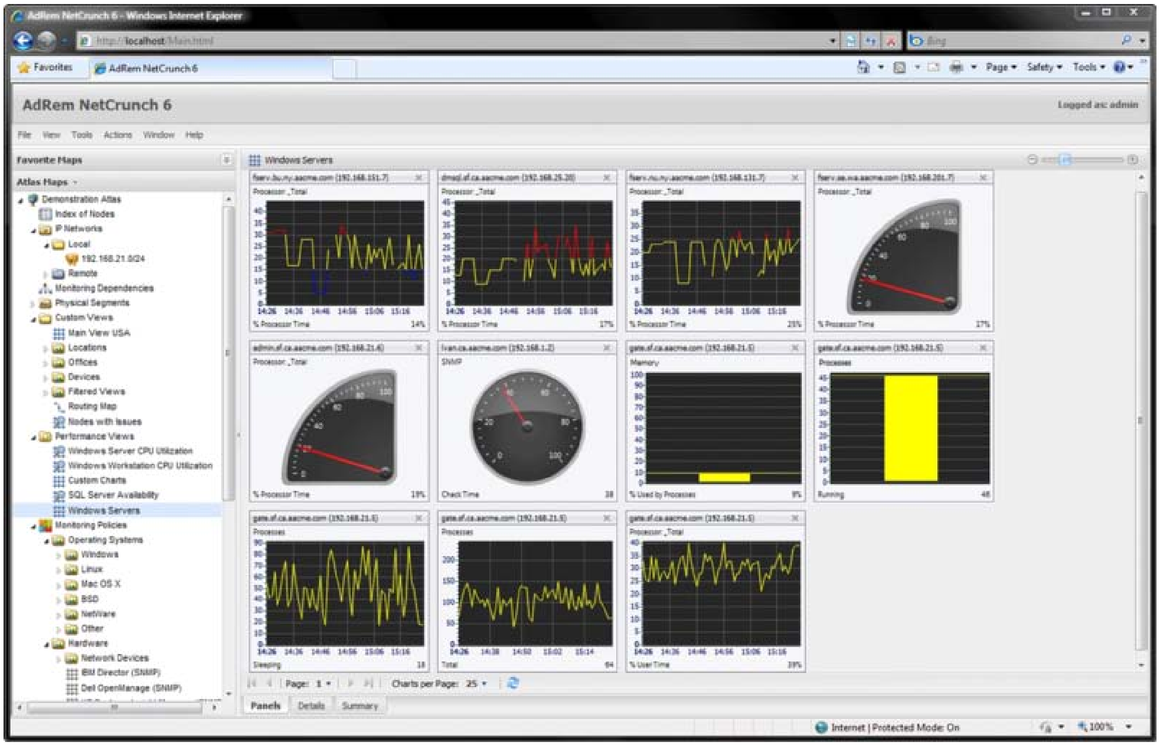
NetCrunch

Developer(s)	AdRem Software, Inc.
Type	Network management system
License	Commercial

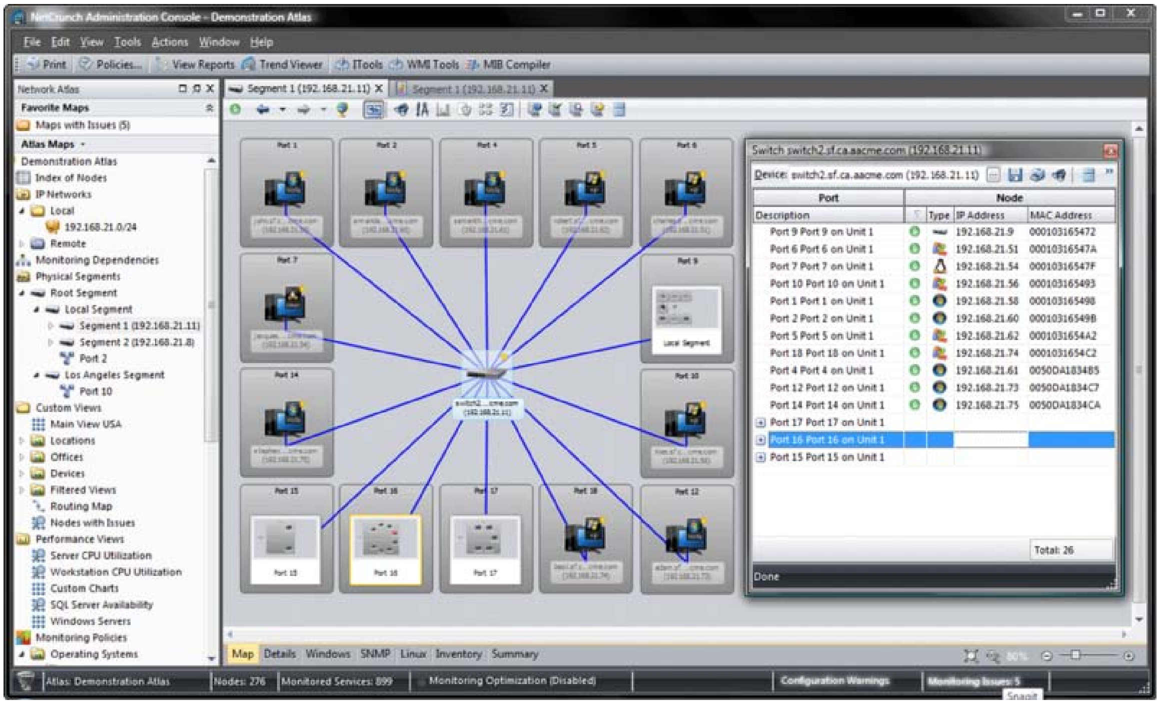
AdRem **NetCrunch** is a commercial software solution for agentless, cross-platform network monitoring developed by AdRem Software, Inc. The program monitors 65 network services, Windows applications; Windows, Linux, NetWare, BSD, Mac OS X systems and SNMP (v1-3) devices without agents; centralizes fault management by collecting and alerting on events from sources including Windows Event Log, syslogs, and SNMP traps; presents physical and logical network topology as automatically updated dynamic graphical views.



AdRem NetCrunch 6 custom network map



AdRem NetCrunch 6 performance view



Switch switch2.sf.ca.aacme.com (192.168.21.11)

Device: switch2.sf.ca.aacme.com (192.168.21.11)

Port		Node		
Description		Type	IP Address	MAC Address
Port 9 Port 9 on Unit 1	●		192.168.21.9	000103165472
Port 6 Port 6 on Unit 1	●		192.168.21.51	00010316547A
Port 7 Port 7 on Unit 1	●		192.168.21.54	00010316547F
Port 10 Port 10 on Unit 1	●		192.168.21.56	000103165493
Port 1 Port 1 on Unit 1	●		192.168.21.58	000103165498
Port 2 Port 2 on Unit 1	●		192.168.21.60	00010316549B
Port 5 Port 5 on Unit 1	●		192.168.21.62	0001031654A2
Port 18 Port 18 on Unit 1	●		192.168.21.74	0001031654C2
Port 4 Port 4 on Unit 1	●		192.168.21.61	0050DA1834B5
Port 12 Port 12 on Unit 1	●		192.168.21.73	0050DA1834C7
Port 14 Port 14 on Unit 1	●		192.168.21.75	0050DA1834CA
Port 17 Port 17 on Unit 1				
Port 16 Port 16 on Unit 1				
Port 15 Port 15 on Unit 1				

Total: 26

Done

AdRem NetCrunch 6 physical network map

Features

AdRem NetCrunch key features:

Auto Discovery - initial and scheduled discovery and classification of network resources

Network Views - physical and logical network topology including predefined views like: IP networks, Routing Map, Physical Segments, Servers, Maps with Issues, and others.

Event Management - events from Windows, Syslog and SNMP traps

Monitoring Policies - sets of rules defining events to be monitored and data to be collected for later reporting, including predefined policies for:

1. *Operating System* - Windows (Active Directory, Network Services Health, Security Audit, Terminal Services, Windows Server, Basic Windows Monitoring), Linux, Mac OS, BSD, Netware, Other (AIX, AS/400, MIB-II host resources).

2. *Hardware* – Network Devices (Nortel, Alcatel OmniSwitch, Cisco), IBM Director, Dell OpenManage, HP Systems Insight Manager, APC PowerChute.

3. *Applications* – Microsoft (Exchange 2003, IIS 5.0/6.0, ISA Server 2000, ISA Server 2004,MS SQL Server 7.0/2000, MS SQL Server 2005), APC Windows Events, ARCServe, CA eTrust Alert Manager, Lotus Notes 6, McAfee AlertManager, Norton

AntiVirus Corporate Edition, Oracle 9i, Sophos Enterprise Manager, Trend Micro ServerProtect, Veritas Backup Exec.

Availability Monitoring - network devices and services (HTTP, POP3, SMTP, etc.)

Performance Monitoring - real-time statistics, multi-server charts, and performance trends (devices, systems, and applications)

Long-term Trend Analysis – reports generated on demand or delivered on schedule via email.

Remote Access – Remote Administration Console and access via web browser.

User Experience Monitors – advanced monitoring of crucial network services. The program simulates user's action e.g. sends an email for POP3 service monitoring, etc.

Smart Monitoring - limiting monitoring traffic for specific sub networks.

Inventory - Ability to gather the basic inventory information of Windows machines (i.e. mainboard, processor type and memory), with options to schedule inventory audits to be performed, automatically discover and view installed software

Platform

Recommended platforms for NetCrunch Server are Windows 2008 x32/x64 or Windows 2003 SP2 x32/x64. AdRem NetCrunch Remote Administration Console runs on Windows 7/Vista SP2/XP SP3 or Windows Server 2003/2008 (x32/x64).

Editions

There are two editions: Premium and Premium XE. The Premium XE edition is designed for large networks and intended to run on a dedicated Windows 2008 server.

Common Premium XE Usage Scenarios include:

- Network with more than a 1000 nodes for availability monitoring
- Network with more than a 1000 network services for availability monitoring
- Network with more than a 100 servers and routers for performance monitoring
- Network with sub networks
- Network with external servers connected over WAN Links
- Network with CISCO or Nortel switches
- MS SQL, Exchange application performance monitoring and trending

Language versions

There are several language versions of AdRem NetCrunch: english, japanese, polish, german. French, spanish and italian are being prepared.

Technology overview

Client-Server Architecture - user manages NetCrunch server using Remote Administration Consoles.

Multithreading - NetCrunch Premium XE uses multithreading to take advantage of multi-core x64 machines’ performance characteristics.

Prioritized monitoring - the program automatically sets up node monitoring order and time upon monitoring dependencies hierarchy.

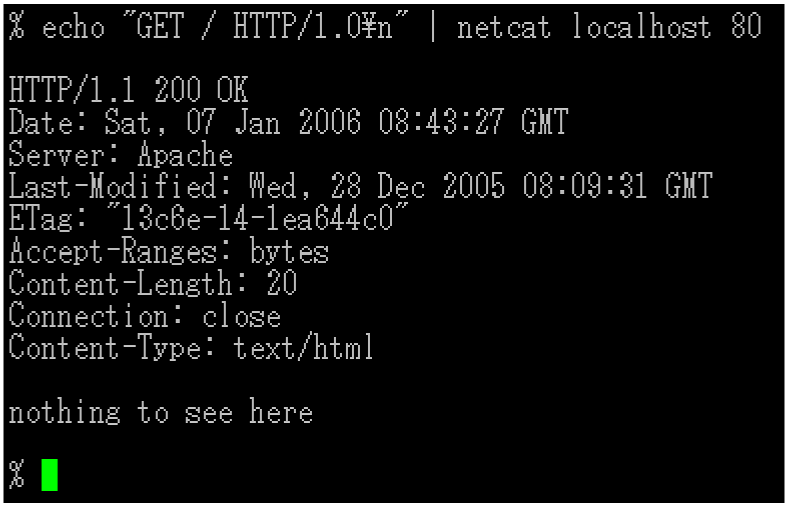
Event Suppression - in case of failure of an intermediate node, the program suppresses alerts from nodes located beyond that node.

SQL event database (up to 128GB)

Web Access - by implementing AJAX technology the server/browser interactions run asynchronously without reloading the GUI Web Access page

Netcat

netcat



```
% echo "GET / HTTP/1.0¥n" | netcat localhost 80

HTTP/1.1 200 OK
Date: Sat, 07 Jan 2006 08:43:27 GMT
Server: Apache
Last-Modified: Wed, 28 Dec 2005 08:09:31 GMT
ETag: "13c6e-14-1ea644c0"
Accept-Ranges: bytes
Content-Length: 20
Connection: close
Content-Type: text/html

nothing to see here

% 
```

Developer(s)	*Hobbit*

Stable release	1.10 / March 20, 1996
Operating system	UNIX
Type	Network utility
License	Permissive free software

Netcat is a computer networking service for reading from and writing network connections using TCP or UDP. Netcat is designed to be a dependable “back-end” device that can be used directly or easily driven by other programs and scripts. At the same time, it is a feature-rich network debugging and investigation tool, since it can produce almost any kind of correlation you would need and has a number of built-in capabilities.

In 2000, according to www.insecure.org, **Netcat** was voted the second most functional network security tool. Also, in 2003 and 2006 it gained fourth place in the same category. Netcat is often referred to as a "Swiss-army knife for TCP/IP." Its list of features includes port scanning, transferring files, and port listening, and it can be used as a backdoor.

Features

Some of netcat's major features are:

- Outbound or inbound connections, TCP or UDP, to or from any ports
- Full DNS forward/reverse checking, with appropriate warnings
- Ability to use any local source port
- Ability to use any locally-configured network source address
- Built-in port-scanning capabilities, with randomization
- Built-in loose source-routing capability
- Can read command line arguments from standard input
- Slow-send mode, one line every N seconds
- Hex dump of transmitted and received data
- Optional ability to let another program service established connections
- Optional telnet-options responder
- Featured tunneling mode which allows also special tunneling such as UDP to TCP, with the possibility of specifying all network parameters (source port/interface, listening port/interface, and the remote host allowed to connect to the tunnel.

Examples

Opening a raw connection to port 25 (like telnet)

```
nc mail.server.net 25
```

Setting up a one-shot webserver on port 8080 to present a file

```
{ echo -ne "HTTP/1.0 200 OK\r\n\r\n"; cat some.file; } | nc -l 8080
```

The file can then be accessed via a webbrowser under http://servername:8080/. Netcat only serves the file once to the first client that connects and then exits.

Checking if UDP ports (-u) 80-90 are open on 192.168.0.1 using zero mode I/O (-z)

```
nc -vzu 192.168.0.1 80-90
```

PS: UDP tests will always show as “open”. The -uz argument is useless.

Pipe via UDP (-u) with a wait time (-w) of 1 second to 'loggerhost' on port 514

```
echo '<0>message' | nc -w 1 -u loggerhost 514
```

Port scanning

An uncommon use of netcat is port scanning. Netcat is not considered the best tool for this job, but it can be sufficient (a more advanced tool is Nmap)

```
nc -v -n -z -w 1 192.168.1.2 1-1000
```

The “-n” parameter here prevents DNS lookup, “-z” makes nc not receive any data from the server, and “-w 1″ makes the connection timeout after 1 second of inactivity.

Making any process a server

On a computer A with IP 192.168.1.2:

```
nc -l -p 1234 -e /bin/bash
```

The “-e” option spawns the executable with its input and output redirected via network socket. It connects to computer A from any other computer on the same network:

```
nc 192.168.1.2 1234
ls -las
total 4288
4 drwxr-xr-x 15 imsovain users 4096 2009-02-17 07:47 .
4 drwxr-xr-x 4 imsovain users 4096 2009-01-18 21:22 ..
8 -rw------- 1 imsovain users 8192 2009-02-16 19:30 .bash_history
4 -rw-r--r-- 1 imsovain users 220 2009-01-18 21:04 .bash_logout
...
```

The consequences are that nc is a popular cracker tool as it is so easy to create a backdoor on any computer. On a Linux computer you may spawn /bin/bash and on a Windows computer cmd.exe to have total control over it.

Port Forwarding or Port Mapping

On Linux, NetCat can be used for port forwarding. Below are nine different ways to do port forwarding in NetCat (-c switch not supported though):

```
nc -l -p port1 -c ' nc -l -p port2'
nc -l -p port1 -c ' nc host2 port2'
nc -l -p port1 -c ' nc -u -l -p port2'
nc -l -p port1 -c ' nc -u host2 port2'
nc host1 port1 -c ' nc host2 port2'
nc host1 port1 -c ' nc -u -l -p port2'
nc host1 port1 -c ' nc -u host2 port2'
nc -u -l -p port1 -c ' nc -u -l -p port2'
nc -u -l -p port1 -c ' nc -u host2 port2'
```

Variants

There are several implementations on POSIX systems, including rewrites from scratch like GNU netcat or OpenBSD netcat (this last has also new features like IPv6 support). Mac OS X users can use the Netcat Darwin Port. There is also a Microsoft Windows version of netcat created by Chris Wysopal, and a Cygwin version is available.

Known ports for embedded systems includes versions for the Windows CE (named Netcat 4 wince) or for the iPhone.

BusyBox includes by default a lightweight version of netcat.

Socat is a more complex cousin of netcat. It is larger and more flexible and has more options that must be configured for a given task.

Cryptcat is a version of netcat with integrated transport encryption capabilities.

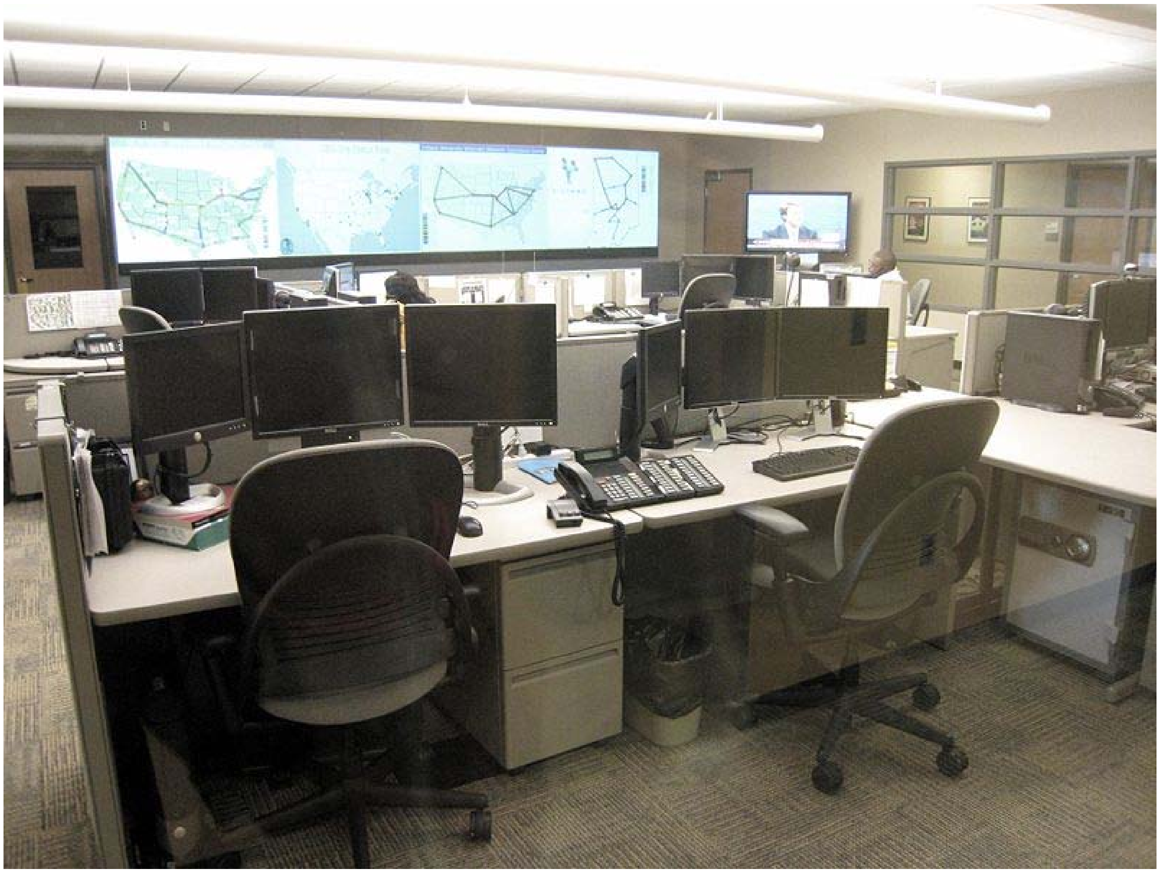
Middle 2005 the Nmap announced another netcat incarnation called Ncat. It features new possibilities such as "Connection Brokering", TCP/UDP Redirection, SOCKS4 client and server support, ability to "Chain" Ncat processes, HTTP CONNECT proxying (and proxy chaining), SSL connect/listen support and IP address/connection filtering. Like Nmap, Ncat is cross-platform.

On some systems, modified versions or similar netcat utilities go by the command name(s) *nc*, *ncat*, *pnetcat*, *socat*, *sock*, *socket*, *sbd*.

Chapter 19

Network Operations Center and OpenNMS

Network operations center



Overview of a typical NOC. Lot of monitors (front), backbone overview (back) and news broadcast on TV-set (right)

A **network operations center** (or **NOC**, pronounced "nok," like the word "knock") is one or more locations from which control is exercised over a computer, television broadcast, or telecommunications network.

Large organizations may operate more than one NOC, either to manage different networks or to provide geographic redundancy in the event of one site being unavailable or offline.

NOCs are responsible for monitoring the telecommunication network for alarms or certain conditions that may require special attention to avoid impact on the network's performance. For example, in a telecommunications environment, NOCs are responsible for monitoring for power failures, communication line alarms (such as bit errors, framing errors, line coding errors, and circuits down) and other performance issues that may affect the network. NOCs analyse problems, perform troubleshooting, communicate with site technicians and other NOCs, and track problems through resolution. If necessary, NOCs escalate problems to the appropriate personnel. For severe conditions that are impossible to anticipate – such as a power failure or optical fiber cable cut – NOCs have procedures in place to immediately contact technicians to remedy the problem.

Technicians in Architel NOC

NOCs are frequently laid out with several rows of desks, all facing a video wall, which typically shows details of highly significant alarms, ongoing incidents and general network performance; a corner of the wall is sometimes used for showing a news or weather TV channel, as this can keep the NOC technicians aware of current events which may have an impact on the network or systems they are responsible for.

The back wall of the NOC is sometimes glazed; there may be a room attached to this wall which is used by members of the team responsible for dealing with serious incidents to meet whilst still able to watch events unfolding within the NOC.

Individual desks are generally assigned to a specific network, technology or area. A technician may have several computer monitors on their desk, with the extra monitors used for monitoring the systems or networks covered from that desk.

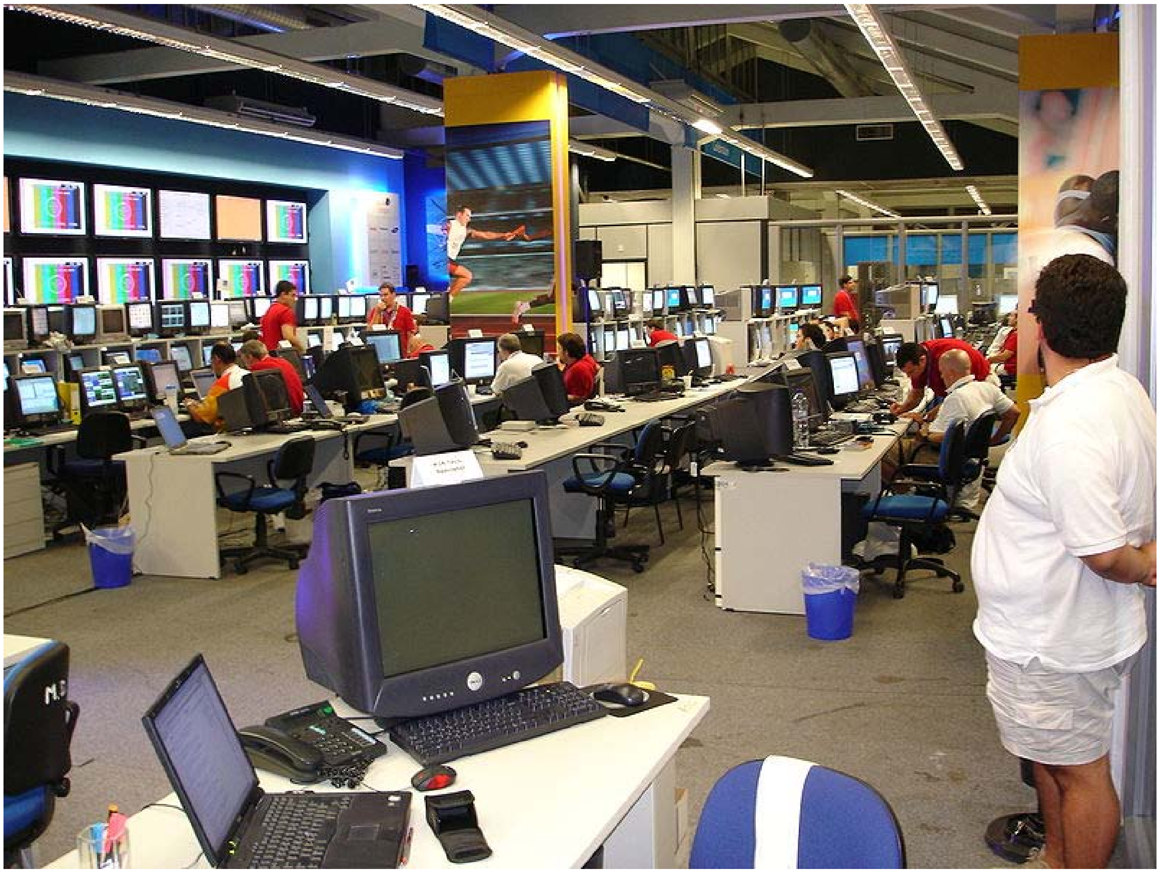
NOCs often escalate issues in a hierarchic manner, so if an issue is not resolved in a specific time frame, the next level is informed to speed up problem remediation. Many NOCs have multiple "tiers", which define how experienced/skilled a NOC technician is. A newly-hired NOC technician might be considered a "tier 1", whereas a technician that has been there for several years may be considered a "tier 3" or "tier 4". As such, some problems are escalated within a NOC before a site technician or other network engineer is contacted.

Additionally, the NOC staff may perform extra duties; a network with equipment in public areas (such as a mobile network Base Transceiver Station) may be required to have a telephone number attached to the equipment for emergencies; as the NOC may be the only continuously staffed part of the business, these calls will often be answered there.

The term *NOC* is normally used when referring to telecommunications providers, although a growing number of other organizations such as public utilities (e.g., SCADA) and private companies also have such centers, both to manage their internal networks and to provide monitoring services.

The location housing a NOC may also contain many or all of the primary servers and other equipment essential to running the network, although it is not uncommon for a single NOC to monitor and control a number of geographically dispersed sites.

In broadcast television



Operations center during the Athens 2004 Olympic games

NOCs at television broadcast facilities are responsible for the technical and operational overview of all broadcast network services, including monitoring, correcting, and troubleshooting day-to-day issues.

Duties that fall under broadcast NOCs include:

- Serial Digital Video, ASI, Multiplexed and DVB data streams technical monitoring
- Networking
- RF and IF distribution
- Monitoring turnaround video services

OpenNMS

OpenNMS is an enterprise grade network monitoring and network management platform developed under the free software or open source model. It consists of a community-supported, free-software project as well as an organization offering commercial services, training and support.

The goal is for OpenNMS to be a truly distributed, scalable platform for all aspects of the FCAPS network management model, and to make this platform available to both free software / open-source and commercial applications.

All code associated with the project is available under the GNU General Public License.

OpenNMS is currently maintained by Tarus Balog, The OpenNMS Group, and The Order of The Green Polo.

Features

- Service polling - determining service availability and latency, including distributed measurement of availability and latency, and reporting on the results
- Data collection - collecting, storing and reporting on data collected from nodes via protocols including SNMP, JMX, HTTP, Windows Management Instrumentation, JDBC, and NSClient
- Thresholding - evaluating polled latency data or collected performance data against configurable thresholds, creating events when these are exceeded or rearmed
- Event management - receiving events, both internal and external, including via SNMP traps
- Alarms and automations - reducing events according to a reduction key and scripting automated actions centered around alarms
- Notifications - sending notices regarding noteworthy events via e-mail, XMPP, or other means

Supported platforms

As OpenNMS is written mainly in Java; it can theoretically run on any system that supports a 1.5 (or higher) SDK.

The following operating systems are supported:

- Linux
- Solaris
- Mac OS X
- Microsoft Windows
- FreeBSD

Awards

On August 11, 2005, OpenNMS won the *Product Excellence Award* in the category *Best Systems Management Tools* at LinuxWorld Conference and Expo . .

The other 3 nominees were:

- Userful DiscoverStation 4.0
- IBM Tivoli Intelligent Orchestrator
- Novell ZENworks 7 Linux Management

OpenNMS won the Gold award in the *Network and IT management platforms* category of SearchNetworking.com's Product Leadership Awards 2007 , beating out HP OpenView and IBM Tivoli.

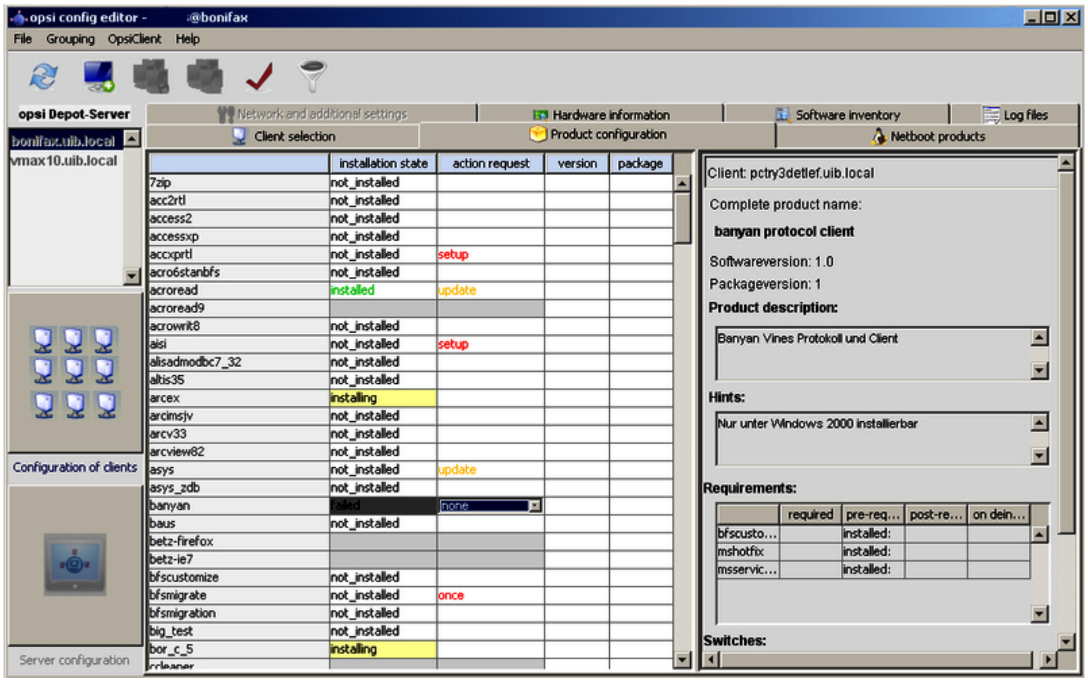
In March 2008, OpenNMS again appeared in the SearchNetworking.com Product Leadership Awards, this time taking bronze or third place in the *Applications and network management* category.

In August 2009, OpenNMS won in the *Networking and network management* category of InfoWorld's BOSSIE (Best of Open Source Software) awards.

Chapter 20

Opsi

opsi



	installation state	action request	version	package
7zip	not_installed			
acc2rtl	not_installed			
access2	not_installed			
accessxp	not_installed			
accxprtl	not_installed	setup		
acro6stanbfs	not_installed			
acroread	installed	update		
acroread9				
acrowrit8	not_installed			
aisi	not_installed	setup		
alisadmodbc7_32	not_installed			
altis35	not_installed			
arcex	installing			
arcimsjv	not_installed			
arcv33	not_installed			
arcview82	not_installed			
asys	not_installed	update		
asys_zdb	not_installed			
banyan	failed	none		
baus	not_installed			
betz-firefox				
betz-ie7				
bfscustomize	not_installed			
bfsmigrate	not_installed	once		
bfsmigration	not_installed			
big_test	not_installed			
bor_c_5	installing			
ccleaner				

Client: pctry3detlef.uib.local

Complete product name:

banyan protocol client

Softwareversion: 1.0

Packageversion: 1

Product description:

Banyan Vines Protokoll und Client

Hints:

Nur unter Windows 2000 installierbar

Requirements:

	required	pre-req...	post-re...	on dein...
bfscusto...		installed:		
mshotfix		installed:		
msservic...		installed:		

Switches:

opsi management interface

Developer(s)	uib gmbh, Mainz, Germany
Stable release	4.0 / 1. oct 2010
Written in	Python Java
Operating system	Linux, Windows
Available in	English, French, German, Spanish, Turkish
Type	Network management System administration
License	GPL

Opsi (open pc server integration) is a software distribution and management system for Windows Clients, based on Linux servers.

Features

The core features of opsi are:

- Automatic operating system installation (OS Deployment)
- Software distribution
- Patch-Management
- Inventory (Hardware and Software)
- License Management / Software Asset Management
- Administrative Tasks (Configuration Management)

Developer and maintainer is the company uib gmbh in Mainz, Germany. The product is Open Source licensed under the GNU General Public License.

Supported client operating systems are Windows 2000, Server 2000, Windows XP, Server 2003. Support for Windows Vista, Server 2008 and Windows 7 is available, but subject to a Co-funding Project and not released as open source yet. For the installation of an opsi-server there are packages available for the Linux distributions Debian, Ubuntu and Suse.

Automatic operating system installation

Via management interface a client may be selected for OS-Installation. If the client boots via PXE it loads a boot image from the opsi-depotserver. This bootimage prepares the hard disk, copies the required installation files, drivers and the opsi client agent and starts finally an unattended OS-Installation. An OS-installation via Disk image is also supported.

Software distribution

For the automatic software distribution a client agent (the opsi-preloginloader) has to be installed on the client. This client agent starts at boot time, connects to the opsi configuration-server and starts if scheduled a script driven installation program (opsi-winst) which installs the required software on the client. During the installation process the user login can be blocked for integrity reasons. To integrate a new software packet into the software deployment system, a script must be written to specify the installation process. This script provides all the information on how this software packet has to be installed silent or unattended or by using tools like AutoIt or Autohotkey.

Patch-Management

The mechanism of the software deployment can also be used to deploy software patches and hotfixes.

Inventory (Hardware and Software)

The Hard- and Software Inventory uses also the opsi client agent. The Hardware Information is collected via calls to WMI while the Software Information is gathered from the registry. The inventory data are sent back to the opsi-configuration-server by web service.

License Management / Software Asset Management

The opsi License Management module supports the administration of different kinds of licenses like Retail, OEM and Volume licenses. It keeps track of the licenses that are used while the software deployment. Using the connection between the License Management and the software inventory, Software Asset Management reports on the number of free and installed licenses can be generated. The License Management module is part of a Co-funding Project and not released as open source yet.

Administrative tasks (Configuration Management)

The opsi client runs a script in a administrative context, which can be used also for configuration purpose. The opsi script interpreter supports:

- Start of programs and exit code detection
- Detection of the running OS, language and national settings as well as evaluation of Ini-files, text files, registry entries and environment variables
- Editing of registry, start menu and desktop entries, Ini-files, XML files and text files
- Editing of user specific profile registry entries and files (in case of not using 'roaming profiles')
- Calling external programs and scripts, catch and provide their output as variables for further processing
- File copy with or without version control

opsi-server

The opsi server provides the following services:

- The configuration-server stores the configuration data for the clients and the software packages. For the data storage a file- or LDAP based storage can be chosen. For administration of these data the configuration-server provides a graphical management interface via https as well as a command line interface.
- The depot-server stores software packages that may be installed by the clients. To provide support for multiple locations, multiple depot-servers may be controlled by one configuration-server.
- A TFTP-Server provides the boot images for the OS-Installations.
- A DHCP-Server may be integrated in the opsi-server

Management-Interface

For the management of opsi there is a graphical user interface, which is available as application as well as Browser Applet. Management is also possible with a command line tool or via web service.

Co-funding Projects

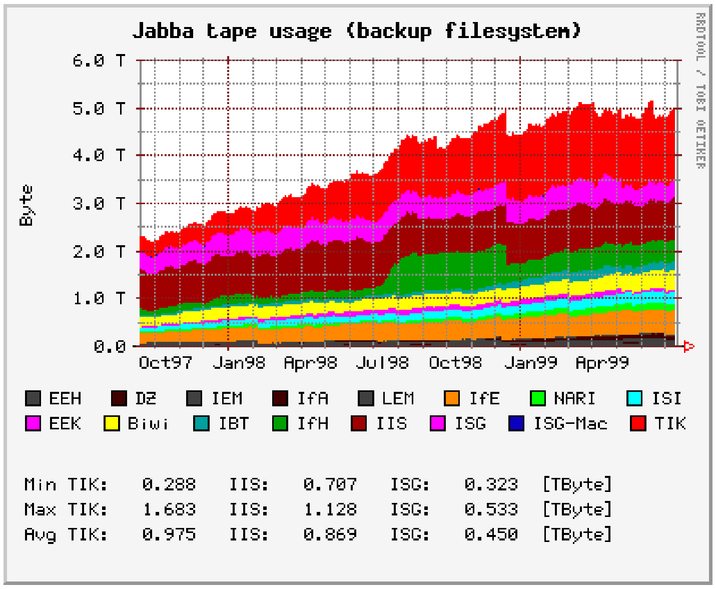
Even though opsi is open source, there are some components which are not free at the moment. These components are developed in a co-funding project which means that until the complete development costs are paid by co-funders, they are only allowed to use by the co-funders or for evaluation purposes.

Chapter 21

RRDtool, Rsyslog and Syslog-ng

RRDtool

	RRDtool
Original author(s)	Tobi Oetiker
Stable release	1.4.4 / July 5, 2010
Written in	C
License	GNU General Public License



RRDtool has a graph function, which presents data from an RRD in a customizable graphical format

RRDtool (acronym for **r**ound-**r**obin **d**atabase **tool**) aims to handle time-series data like network bandwidth, temperatures, CPU load etc. The data are stored in a round-robin database (circular buffer), thus the system storage footprint remains constant over time.

It also includes tools to extract **RRD** data in a graphical format.

Tobi Oetiker wrote RRDtool as a replacement for MRTG and licenses it as free software under the terms of the GNU General Public License (GPL).

Bindings exist for Perl, Python, Ruby, Tcl, PHP and Lua.

General data storage

RRDtool assumes time-variable data in intervals of a certain length. This interval, usually named **step**, is specified upon creation of an RRD file and cannot be changed afterwards. Because data may not always be available at just the right time, RRDtool will automatically interpolate any submitted data to fit its internal time-steps.

The value for a specific step, that has been interpolated, is named a primary data point (**PDP**). Multiple primary data points may be consolidated according to a consolidation function (**CF**) to form a consolidated data point (**CDP**). Typical consolidation functions are average, minimum, maximum.

After the data have been consolidated, the resulting CDP is stored in a round-robin archive (**RRA**). A round-robin archive stores a fixed amount of CDPs and specifies how many PDPs should be consolidated into one CDP and which CF to use. The total time covered by an RRA can be calculated as follows:

```
 time covered = (#CDPs stored) * (#PDPs per CDP) * step
```

After this time the archive will "wrap around": the next insertion will overwrite the oldest entry. This behavior is sometimes referred to as "round-robin" and is the reason for the program's name.

To cover several timespans and/or use several consolidation functions, an RRD file may contain multiple RRAs. The data retrieval function of RRDtool automatically selects the archive with the highest resolution that still covers the requested timespan. This mechanism is also used by RRDtool's graphing subsystem.

Release history

Colour	Meaning
🟥 Red	Release no longer supported
🟩 Green	Release still supported
🟦 Blue	Future release

RRDTool is sponsored since 1.2, each release comes with a list of sponsors.

The following table contains the **release history of RRDtool**, showing most of its release versions.

Version number	Date	Links	Notable changes
1.0	July 16, 1999	Full release notes, Announce	First release. Basically MRTG "done right".
1.1	April 25, 2005	Full release notes, Announce	libart; output EPS, PDF & SVG; VDEF; trends; percentiles; updatev; Holt-Winters Forecasting; COMPUTE; .rrd format change.
1.3	June 11, 2008	Full release notes, Announce	Safer & faster file access; cairo/pango; anti-aliasing; TEXTALIGN; dashed lines; new HWPREDICT; libxml; i18n; XML dump;
1.4	October 27, 2009	Full release notes, Announce	Caching daemon; VDEF PERCENTNAN; CDEF PREDICT & PREDICTSIGMA; libDBI; graph legends positioning; Lua bindings; 3D border width; and more ...

Rsyslog

Rsyslog	
Original author(s)	Rainer Gerhards
Stable release	5.6.2 / November 30, 2010; 12 days ago
Preview release	6.1.1 / November 30, 2010; 12 days ago
Written in	C
Operating system	Unix-like
Type	System logging
License	GNU General Public License v3

Rsyslog is an open source program for forwarding log messages in an IP network for UNIX and Unix-like systems. It implements the basic syslog protocol, extends it with content-based filtering, rich filtering capabilities, flexible configuration options and adds important features such as using TCP for transport.

Protocol

Rsyslog uses the quasi-standard BSD syslog protocol, specified in RFC 3164. As the text of RFC 3164 is just a vague informational description and not a standard, various incompatible extensions of it emerged. Rsyslog supports many of these extensions. The format of relayed messages can be customized.

The most important extensions of the original protocol supported by rsyslog are:

- ISO 8601 timestamp with millisecond granularity and timezone information
- the addition of the name of relays in the host fields to make it possible to track the path a given message has traversed
- reliable transport using TCP
- support GSS-API and TLS
- logging directly into various database engines.
- support for the upcoming new IETF syslog RFC series
- support for buffered operation modes where messages are buffered locally if the receiver is not ready

History

The rsyslog project began in 2004, when Rainer Gerhards, the primary author of rsyslog, decided to write a new strong syslog daemon to compete with syslog-ng, because; and according to the author "A new major player will prevent monocultures and provide a rich freedom of choice."

Distributions

rsyslog is available for a number of Unix systems and Linux distributions, among others:

- Fedora (In November 2007, rsyslog has become the default syslogd for the Fedora project) Fedora was the first major distribution to adopt this software.
- openSUSE (default since 11.2; November 2009)
- Debian GNU/Linux (As of Debian 5.0, rsyslog has become the default syslog)
- Ubuntu
- Red Hat Enterprise Linux (from the upcoming version 6)
- Solaris
- FreeBSD
- OpenBSD
- Gentoo

Related RFCs and working groups

- RFC 3164 - The BSD syslog Protocol (obsoleted by RFC 5424)
- RFC 5424 - The Syslog Protocol (obsoletes RFC 3164)
- RFC 5425 - Transport Layer Security Mapping for Syslog

- RFC 5426 - Transmission of Syslog Messages over UDP

Syslog-ng

syslog-ng	
Original author(s)	Balázs Scheidler
Initial release	1998
Stable release	3.2.1 / November 30, 2010; 12 days ago
Operating system	Unix-like
Type	System logging
License	GNU Lesser General Public License(core) GNU General Public License version 2(plugins)

syslog-ng is an open source implementation of the Syslog protocol for Unix and Unix-like systems. It extends the original syslogd model with content-based filtering, rich filtering capabilities, flexible configuration options and adds important features to syslog, like using TCP for transport. As of today syslog-ng is developed by Balabit IT Security Ltd. It have two editions with common codebase. The fist is called syslog-ng OSE (with the license LGPL) and have additional plugins (modules) under proprietary license. This edition is called Premium Edition (PE).

Protocol

syslog-ng uses the quasi-standard BSD syslog protocol, specified in RFC 3164. As the text of RFC 3164 is vague and is just an informational description and not a standard, various incompatible extensions of it emerged. Since version 3.0 also supports the standard syslog protocol specified in RFC 5424 which was released in 2009. syslog-ng tries hard to interoperate with a wide variety of devices, and the format of relayed messages can be customized.

The most important extensions of the original protocol endorsed by syslog-ng are:

- ISO 8601 timestamp with millisecond granularity and timezone information

- the addition of the name of relays in the host fields to make it possible to track the path a given message has traversed
- reliable transport using TCP
- TLS encryption (Since 3.0.1 in OSE)

History

The syslog-ng project began in 1998, when Balázs Scheidler, the primary author of syslog-ng, ported the existing nsyslogd code to Linux. The 1.0.x branch of syslog-ng was still based on the nsyslogd sources and are available in the syslog-ng source archive.

Right after the release of syslog-ng 1.0.x, a reimplementation of the code base started to address some of the shortcomings of nsyslogd and to address the licensing concerns of Darren Reed, the original nsyslogd author. This reimplementation was named stable in the October of 1999 with the release of 1.2.0. This time around, syslog-ng depended on some code originally developed for lsh by Niels Möller.

Three major releases (1.2, 1.4 and 1.6) were using this code base, the last release of the 1.6.x branch in February 2007. In this period of about 8 years, syslog-ng became one of the most popular alternative syslog implementations.

In a volunteer based effort, yet another rewrite was started back in 2001, dropping lsh code and using the more widely available GLib library. This rewrite of the codebase took its time, the first stable release of 2.0.0 happened in October 2006.

Development efforts are focused on improving the 2.0.x branch; support for 1.6.x is expected to be dropped in the near future (as of May 2007). Balabit, the company behind syslog-ng, started a parallel, commercial fork of syslog-ng, called syslog-ng Premium Edition. Portions of the commercial income are used to sponsor development of the free version.

Syslog-ng version 3.0 was released in the fourth quarter of 2008.

Starting with the 3.0 version developments efforts were parallel on the Premium and on the Open Source Editions. PE efforts were focused on quality, transport reliability, performance and encrypted log storage. The Open Source Edition efforts focused on improving the flexibility of the core infrastructure to allow more and more different, non-syslog message sources.

Both the OSE & PE forks produced two releases (3.1 and 3.2) in 2010.

Features

syslog-ng has a much larger scope than merely transporting syslog messages and storing them to plain text log files:

- the ability to format log messages using UNIX shell-like variable expansion;
- the use of this shell-like variable expansion when naming files, thus covering thousands of destination files with a single statement;
- the ability to send log messages to local applications;
- ability to message flow-control in network transport;
- logging directly into a database (since syslog-ng OSE 2.1);
- rewrite portions of the syslog message with set and substitute primitives (since syslog-ng OSE 3.0);
- classify incoming log messages and at the same time extract structured information from the unstructured syslog message (since syslog-ng OSE 3.0);
- generic name-value support: each message is just a set of name-value pairs, which can be used to store extra information (since syslog-ng OSE 3.0);
- the ability to process structured message formats transmitted over syslog, like extract columns from CSV formatted lines (since syslog-ng OSE 3.0);
- the ability to correllate multiple incoming messages to form a more complex, correllated event (since syslog-ng OSE 3.2);

Distributions

syslog-ng is part of a number of different GNU/Linux and Unix distributions. Some distributions install it as the default system logger, others only provide a package and an upgrade path from the standard syslogd.

Among others:

- openSUSE used it prior to openSUSE 11.2
- Debian GNU/Linux used before version 5.0 syslogd and klogd (Lenny (5.0) uses Rsyslog)

- Gentoo Linux
- Fedora used it prior to Fedora 10
- Arch Linux
- Hewlett-Packard's HP-UX
- FreeBSD
- A Cygwin port is available for Microsoft Windows

Portability

syslog-ng is highly portable to many Unix systems, old and new alike. A list of the currently known to work Unix versions are found below:

- Linux on i386, SPARC and x86-64 CPUs
- FreeBSD 6.x, 7.x on i386 CPUs
- AIX 5 on IBM POWER CPUs
- HP-UX 11iv1, 11iv2 on PA-RISC and Itanium CPUs
- Solaris 8, 9, 10 on SPARC, x86-64 and i386 CPUs

- Tru64 5.1b on Alpha CPUs

The list above is based on BalaBit's current first hand experience, other platforms may also work, but your mileage may vary.

Related RFCs & working groups

- RFC 3164 - The BSD syslog protocol
- RFC 5424 - The Syslog Protocol
- RFC 5425 - Transport Layer Security (TLS) Transport Mapping for Syslog
- RFC 5426 - Transmission of Syslog Messages over UDP

Log Viewers

Name	Free	Description
XLog-Solution	Free and Commercial	Log aggregation from distributed systems. Web-based dashboard with control over filters. Email alerts. Interface with Jira.