8

Performance and Clusters

It is easy for system designers to become preoccupied by performance. It is good to know that a specific system configuration can cope with a specific workload, and you can run performance tests until the cows come home to demonstrate that Exchange is able to cope with thousands of mailboxes. Unfortunately, this position guarantees that you miss two facts. First, because it is difficult to assemble thousands of real-life users and get them to create a workload for a server, the normal course is to run performance tests with a simulated workload. Of course, this demonstrates that the server can cope with a simulated workload and therefore creates a certain amount of confidence before deployment, but it is no guarantee that the system will achieve the desired performance in production. Real users have an annoying habit of doing things that no self-respected simulated user would, such as deciding to send a message with a 10-MB attachment to a huge distribution list, and this type of behavior skews system predictability.

The second factor is raw CPU performance. An Exchange benchmark exceeded the 3,000 mailboxes per server level in 1997, yet few system designers rush to put more than 3,000 mailboxes on a production server even though CPU clock speed has increased dramatically since. Benchmarks now suggest that you can easily support tens of thousands of Exchange mailboxes on the latest servers, but few go past the 3,000 sticking point unless they are sure that they have the right combination of rock-solid operations procedures wrapped around industrial-strength server and storage hardware. The fact is that the steady increase in server speed reduced the need to worry about Exchange performance long ago. Most modern servers will cheerfully handle the load that your user population generates, and you can crank up the number of mailboxes on a server to levels that seemed impossible just a few years ago. Of course, increasing mailboxes on a server is not wise unless you know that you can manage them, but that is not the fault of either the software or the hardware.

A discussion about Exchange performance is, therefore, more about how you can run performance tests and the type of hardware configurations that you might deploy at the high end rather than a pursuit of the last possible piece of speed. Therefore, that is what we cover in this chapter: aspects of Exchange performance, the performance tools, high-end standard servers, the role of storage, and a discussion about clusters. Performance changes with hardware and developments in this area evolve rapidly, so it is best to use this chapter as a guideline for places you need to investigate rather than to expect the definitive text (which would fast become outdated).

8.1 Aspects of Exchange performance

The earliest Exchange servers were easy to configure. You bought the fastest processor, equipped the server with as much direct connected storage as it supported, and bought what seemed to be a huge amount of memory (such as 128 MB). System administration was easier, too, since the number of supported mailboxes on servers was not large. Table 8.1 charts the evolution of typical “large” Exchange server configurations since 1996 and especially illustrates the growth in data managed on a server. Today, we see a trend toward server consolidation, as companies seek to drive down cost by reducing the number of servers that they operate. The net result of server consolidation is an increase in the average number of mailboxes supported by the typical Exchange server, an increasing desire to use network-based storage instead of direct connected storage, and growing management complexity with an attendant need for better operational procedures and implementation.

The growing maturity of Windows and the hardware now available to us help server consolidation, but the new servers that we deploy still have to be balanced systems suited to the application in order to maximize results. A

Table 8.1 *The Evolution of Exchange Server Configurations*

Version	CPU	Disk	Memory
Exchange 4.0	Single 100-MHz/256-KB cache	4 GB	128 MB
Exchange 5.5	Single 233-MHz/512-KB cache	20 GB	256 MB
Exchange 2000	Dual 733-MHz/1-MB cache	>100 GB	512 MB
Exchange 2003	Quad 2-GHz/2-MB cache	SAN	4 GB

balanced system is one that has the right proportions of CPU power, storage, and memory. After all, there is no point in having the fastest multi-CPU server in the world if you cannot provide it with data to process. Storage and good I/O management are key points in building Exchange servers to support large user communities.

Exchange performance experts often aim to move processing to the CPU and keep it busy on the basis that a server that hums along at 10 percent CPU load may be underloaded because it is waiting for I/Os to complete. This illustrates the point that a system is composed of multiple elements that you have to balance to achieve maximum performance.

8.1.1 Storage

Storage becomes cheaper all the time, as disk capacity increases and prices drop. However, configuring storage is not simply a matter of quantity. Instead, for Exchange servers, you need to pay attention to:

- Quantity: You have to install enough raw capacity to accommodate the space you expect the O/S, Exchange, and other applications to occupy for their binaries, other support files, and user data. You also need to have sufficient capacity to perform maintenance operations and to ensure that the server will not run out of space on important volumes if users generate more data than you expect.
- Resilience: You have to isolate the important parts of the overall system so that a failure on one volume does not lead to irreversible damage. The basic isolation scheme is to use separate physical volumes to host the following files:
  - Windows O/S
  - Exchange binaries
  - Exchange databases
  - Exchange transaction logs
- Recoverability: Tools such as hot snapshots need a substantial amount of additional space to work.
- I/O: The sheer capacity does not matter if the storage subsystem (controller and disks) cannot handle the I/O load generated by Exchange.
- Manageability: You need to have the tools to manage the storage, including backups. Newer techniques such as storage virtualization may be of interest if you run high-end servers.

You can build these qualities into storage subsystems based on direct-connected storage to the largest SAN. The general rule is that the larger the server, the more likely it is to connect to a SAN in order to use features such as replication, virtualization, and business continuity volumes. Indeed, you can argue a case that it is better to concentrate on storage first and build servers around storage rather than vice versa, because it is easier to replace servers if they use shared storage.

Best practice for Exchange storage includes:

- Always keep the transaction logs and the Store databases isolated from each other on different physical volumes.
- Place the transaction logs on the drives with the optimal write performance so that the Store can write transaction information to the logs as quickly as possible.
- Protect the transaction logs with RAID 1. Never attempt to run an Exchange server in a configuration where the transaction logs are unprotected.
- Protect the Store databases with RAID 5 (minimum) or RAID 0+1. RAID 0+1 is preferred, because this configuration delivers faster performance (twice the speed of RAID 5) with good protection.
- Multispindle volumes help the system service the multiple concurrent read and write requests typical of Exchange. However, do not attempt to add too many spindles (no more than 12) to a RAID 5 volume. Deciding on the precise number of spindles in a volume is a balancing act between storage capacity, I/O capabilities, and the background work required to maintain the RAID 5 set.
- Use write cache on the storage controller for best performance for transaction log and database writes, but ensure that the controller protects the write cache against failure and data loss with features such as mirroring and battery backup. You also need to be able to transfer the cache between controllers if the controller fails and you need to replace it.

Storage technology evolves at a startling rate and we have seen the price per GB driven down dramatically since Exchange 4.0 appeared. New technologies are likely to appear, and you will have to make a decision regarding whether to use the technology with your Exchange deployment. Sometimes vendors make it unclear whether Microsoft fully supports the technology, and this is especially so with respect to database-centric applications such as Exchange and SQL. For example, Network Attached Storage (NAS) devices seem attractive because they are cheap and allow you to expand storage easily. However, at the time of writing Microsoft does not support NAS block-mode devices with Exchange and does not support any file-mode NAS devices. There are a number of reasons for this position, including network latency for write operations and redirectors introduced between the Store APIs and the Windows I/O Manager (see Microsoft Knowledge Base articles 314916 and 317173 for more information, including Microsoft's support policy for NAS devices). The Hardware Compatibility List (available from Microsoft's Web site) is the best place to check whether Microsoft supports a specific device, and it is also a good idea to ask vendors whether they guarantee that their device supports Exchange. Another good question is to ask the vendor to describe the steps required to recover mailbox data in the event of a hardware failure. However, technology changes and newer devices may appear that eliminate the problems that prevent Microsoft from supporting NAS and other storage technology. For this reason, you should consult a storage specialist before you attempt to build a storage configuration for any Exchange server.

8.1.2 Multiple CPUs

Given the choice, it is better to equip Exchange servers with multiple CPUs. Since Exchange 5.0, the server has made good use of multiple CPUs. Best practice is to use multi-CPU systems instead of single-CPU systems, with the only question being how many CPUs to use. Here is the logic:

- It does not cost much to equip a server with additional CPUs when you buy servers, and adding a CPU is a cheap way to extend the lifetime of the server.
- The extra CPU power ensures that servers can handle times of peak demand better.
- Add-on software products such as antivirus scanners consume CPU resources. The extra CPUs offload this processing and ensure that Exchange continues to service clients well.
- New versions of Windows and Exchange usually include additional features that consume system resources. If the new features support SMP, the extra CPUs may allow you to upgrade software without upgrading hardware.
- Adding extra CPUs after you build a server can force you to reinstall the operating system or applications.

- The performance of any computer degrades over time unless you perform system maintenance, and, even then, factors such as disk fragmentation conspire to degrade performance. Some extra CPU power offsets the effect of system aging.

Note that secondary cache is important for symmetric multiprocessing. Secondary cache is a high-performance area of memory that helps prevent front-side bus saturation. Large multi-CPU servers are inevitably equipped with a generous secondary cache, with the general rule that the more, the merrier.

With these points in mind, it is a good idea to equip small servers (under 1,000 mailboxes) with dual CPUs, and large mailbox servers with four CPUs. Going beyond this limit enters the domain of high-end systems and is probably not necessary for the vast majority of Exchange servers. Few people find that something like a 32-way server is necessary to support Exchange and that it is easier and cheaper to deploy servers with fewer CPUs. If you are tempted to purchase a system with more than eight CPUs, make sure that you know how you will configure the system, the workload it will handle, and the additional benefit you expect to achieve.

8.1.3 Memory

The various components of Exchange, such as the Store, DSAccess, Routing Engine, and IIS, make good use of memory to cache data and avoid expensive disk I/Os, so it is common to equip Exchange servers with large amounts of memory, especially since the price of memory has come down. It is always better to overspecify memory than install too little, since server performance is dramatically affected by any shortage of memory.

The Store is a multithreaded process implemented as a single executable (STORE.EXE), which runs as a Windows service and manages all the databases and storage groups on a server. As more users connect to mailboxes and public folders, the number of threads grows and memory demands increase.The Store is reputed to be a particular “memory hog,” because it uses as much memory as Windows can provide. However, this behavior is by design and is due to a technique called Dynamic Buffer Allocation, or DBA, which Microsoft introduced in Exchange 5.5. Before Exchange 5.5, administrators tuned a server with the Exchange Performance Wizard, which analyzed the load on a running server and adjusted system parameters. Specifically, the wizard tuned the number of buffers allocated to the Store to accommodate an expected number of connections. However, the wizard used no great scientific method, and much of the tuning was by guesswork and estimation. If the actual load on a server differed from the expected load, the tuning was inaccurate.

Microsoft implemented DBA to provide a self-tuning capability for the Store and ensure that the Store uses an appropriate amount of memory at all times, taking the demands of other active processes into account. DBA is an algorithm to control the amount of memory used by the Store and is analogous to the way that Windows controls the amount of memory used by the file cache and the working set for each process. To see the analogy, think of I/O to the Store databases as equivalent to paging to the system page file.

DBA works by constantly measuring demand on the server. If DBA determines that memory is available, the Store asks Windows for more memory to cache more of its data structures. If you monitor the memory used by the Store process, you will see it gradually expand to a point where the Store seems to use an excessive amount of memory, a fact that can alarm inexperienced system administrators who have not experienced it before. For example, in Figure 8.1 you can see that the Store process occupies a large amount of memory even though the system is not currently under much load. This is the expected situation and if another process becomes active that requires a lot of memory, the Store process shrinks if Windows cannot provide the required memory to that process.

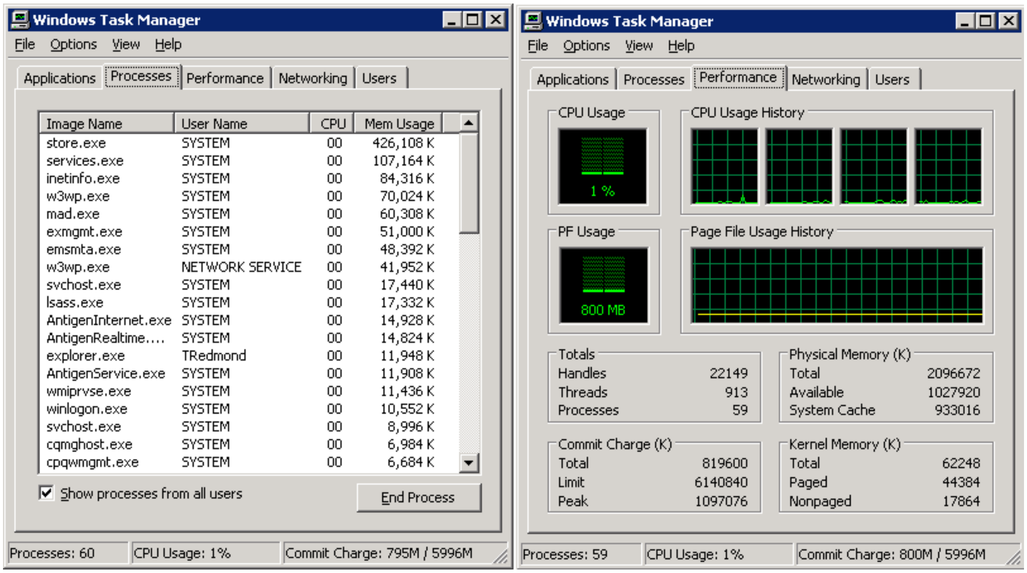


Image Name	User Name	CPU	Mem Usage
store.exe	SYSTEM	00	426,108 K
services.exe	SYSTEM	00	107,164 K
inetinfo.exe	SYSTEM	00	84,316 K
w3wp.exe	SYSTEM	00	70,024 K
mad.exe	SYSTEM	00	60,308 K
exmgmt.exe	SYSTEM	00	51,000 K
emsmta.exe	SYSTEM	00	48,392 K
w3wp.exe	NETWORK SERVICE	00	41,952 K
svchost.exe	SYSTEM	00	17,440 K
lsass.exe	SYSTEM	00	17,332 K
AntigenInternet.exe	SYSTEM	00	14,928 K
AntigenRealtime....	SYSTEM	00	14,824 K
explorer.exe	TRedmond	00	11,948 K
AntigenService.exe	SYSTEM	00	11,908 K
wmiprvse.exe	SYSTEM	00	11,436 K
winlogon.exe	SYSTEM	00	10,552 K
svchost.exe	SYSTEM	00	8,996 K
cqmghost.exe	SYSTEM	00	6,984 K
cpqwmgmt.exe	SYSTEM	00	6,684 K

☑ Show processes from all users

Processes: 60 | CPU Usage: 1% | Commit Charge: 795M / 5996M

Totals		Physical Memory (K)	
Handles	22149	Total	2096672
Threads	913	Available	1027920
Processes	59	System Cache	933016

Commit Charge (K)		Kernel Memory (K)	
Total	819600	Total	62248
Limit	6140840	Paged	44384
Peak	1097076	Nonpaged	17864

Processes: 59 | CPU Usage: 1% | Commit Charge: 800M / 5996M

Figure 8.1 *Memory used by the Store.*

There is no point in having memory sitting idle, so it is good that the Store uses available memory as long as it does not affect other processes. DBA monitors system demand and releases memory back to Windows when required to allow other processes to have the resources they need to work; it then requests the memory back when the other processes finish or release the memory back to Windows. On servers equipped with relatively small amounts of memory, you can sometimes see a side effect of DBA when you log on at the server console and Windows pauses momentarily before it logs you on and paints the screen. The pause is due to DBA releasing resources to allow Windows to paint the screen. DBA is not a fix for servers that are underconfigured with memory, but it does help to maximize the benefit that Exchange gains from available memory.

8.1.4 Using more than 1 GB of memory

Exchange servers running Windows 2000 Advanced Server (or any version of Windows 2003) that are equipped with 1 GB or more of physical memory require changes to the default virtual memory allocation scheme to take advantage of the available memory. Usually, Windows divides the standard 4 GB available address space between user and kernel mode. You can set the /3GB switch to tell Windows that you want to allocate 3 GB of the address space to user-mode processing, which allows Exchange to use the additional memory, especially within the single Store process that probably controls multiple Store instances on large servers (one for each storage group). Although using this switch allows you to provide more memory to Exchange and therefore scale systems to support heavier workloads, Windows may come under pressure as you reduce kernel-mode memory to 1 GB, which may cause Windows to exhaust page table entries—in turn, leading to unpredictable system behavior.

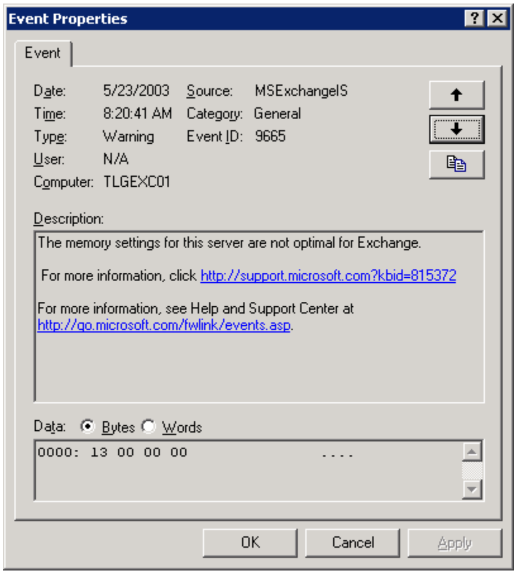
To make the necessary change, add the /3GB switch to the operating system section of boot.ini. For example:

```
[Operating Systems]
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Microsoft
Windows 2000 Server" /fastdetect /3GB
```

Windows 2003 provides an additional switch for boot.ini (USERVA). When used in conjunction with the /3GB switch, you can use the USERVA switch to achieve a better balance between the allocation of kernel- and user-mode memory. Microsoft recommends that you use a setting of /USERVA=3030 (the value is in megabytes) for Exchange 2003 servers. This value may change as experience grows with Exchange 2003 in different production configurations, so check with Microsoft to determine the correct value for your server configuration. Its net effect is to allocate an extra 40 MB of memory to the Windows kernel for page table entries, in turn allowing Exchange to scale and support additional users without running out of system resources.

Based on experience gained in the way the Store uses memory, Exchange 2003 attempts to use available memory more intelligently than Exchange 2000 when you set the /3GB switch, and you should set the switch on any server that has more than 1 GB of physical memory. If you do not, Exchange reports a nonoptimal configuration as event 9665 in the event log (Figure 8.2) when the Information Store service starts. This is just a pointer for you to remember to set the /3GB switch.

In Exchange 2000, the Store allocates a large amount of virtual memory (858 MB) for the ESE buffer. The Store always allocates the same amount of virtual memory, regardless of the system or memory configuration. The one size fits all approach is convenient, but it can lead to situations where smaller systems exhaust virtual memory. In Exchange 2003, the Store looks for the /3GB switch and uses it as the basis for memory allocation. If the switch exists, the Store assumes that lots of physical memory are available, so it allocates 896 MB for its buffer. If not, the Store tunes its virtual memory demand back to 576 MB.



Date: 5/23/2003 Source: MSExchangeIS
Time: 8:20:41 AM Category: General
Type: Warning Event ID: 9665
User: N/A
Computer: TLGEXC01

Description:

The memory settings for this server are not optimal for Exchange.

For more information, click http://support.microsoft.com?kbid=815372

For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

Data: ● Bytes ○ Words

```
0000: 13 00 00 00                ....
```

Figure 8.2
Exchange reports nonoptimal memory.

Finally, even though the Datacenter Edition of Windows 2003 supports up to 512 GB of memory, there is no point in equipping an Exchange server with more than 4 GB, since the current 32-bit version of Exchange cannot use the extra memory. This situation may change over time, so it is a good idea to track developments as Microsoft improves its 64-bit story.

8.1.5 Advanced performance

People want the best possible performance for their servers, so each new advance in server technology is eagerly examined to see whether it increases the capacity of a server to support more work. In the case of Exchange, this means more mailboxes. As we have discussed, other factors such as extended backup times or not wanting to put all your eggs in one basket (or all mailboxes on one server) can influence your comfort level for the maximum number of mailboxes on a server, but it is still true that extra performance always helps. Extra CPU speed can balance the inevitable demand for system resources imposed by new features, any lack of rigor in system management and operations, and the drain from third-party products such as antivirus scanners. Apart from speedier CPUs, the two most recent developments are hyperthreading and 64-bit Windows.

Hyperthreading (or simultaneous multithreading) is a technique that allows a CPU such as recent Intel Xeon processors to handle instructions more efficiently by providing code with multiple execution paths. In effect, to a program, a server seems to have more CPUs than it physically possesses. Not every program is able to take advantage of hyperthreading, just as not every program can take advantage of a system equipped with multiple CPUs, and not every program can exploit a grid computer. As it happens, Exchange has steadily improved its ability to use advanced hardware features such as multithreading since Exchange 5.0, and Exchange 2003 is able to use hyperthreaded systems. Indeed, experience shows that enabling hyperthreading on the 400-MHz front-side bus found in high-end servers creates some useful extra CPU "head room," which may allow you to support additional mailboxes on a server. Therefore, if you have the option, it is best to deploy a hyperthreaded system whenever possible.

With the arrival of the first native 64-bit Windows operating system,[1] people often ask how Exchange will take advantage of the extended memory space and other advantages offered by a 64-bit operating system. The

[1] Windows NT ran on the 64-bit Alpha chip from versions 3.1 to 4.0, but Windows 2000 was never ported to Alpha for production purposes. Microsoft used 64-bit versions of Windows 2000 on Alpha for development purposes only.

answer is that Exchange runs on the IA64 platform, but only as a 32-bit application running in emulation mode in the same manner as first-generation Exchange supports the Alpha platform. Porting a large application such as Exchange to become a native 64-bit application requires an enormous amount of work, and given that the third generation of Exchange uses a new database engine, it was always very unlikely that Microsoft would do the work in Exchange 2003. Thus, the next major release will be the first true 64-bit version of Exchange. In the meantime, you can certainly deploy Exchange 2003 on IA64 systems with an eye on the future.

Waiting for a future 64-bit version to appear does not mean that Microsoft will stop fixing problems in the current Store, nor will they stop adding features. Instead, the true meaning is that Microsoft is now dedicated to developing a new company-wide storage strategy that will accommodate Exchange alongside other products rather than focusing on the current ESE-base Store. The net effect is that we still have a while to wait before we can expect to use a version of the Store that fully exploits a 64-bit architecture to achieve better performance and higher scalability.

8.2 Measuring performance

As with any venture, the first question to ask is why you want to measure performance. Perhaps it is to validate different system configurations so that you can make a choice between one system configuration and another. For example, should you use a two-CPU system or a four-CPU system? Or perhaps it is to test a particular variant of an Exchange server under realistic working conditions, such as a cluster (to test how long failovers take after a Store failover) or how front- and back-end servers work together with clients accessing the servers through firewalls, a DMZ, and so on. For whatever reason, it is wise to begin the exercise with four points in mind:

- The “realistic” workload that you generate through simulation software is only representative of the workload that simulated users produce. Real users can do weird and wonderful things to upset system performance. In fact, they are rather good at doing this.
- A balanced system is more important than the fastest system on earth. This means that you have to concentrate on achieving the best balance between the speed of the CPU, the number of CPUs, the amount of storage, the type of storage and controller, and the amount of memory.

- Performance measured on a server that only runs Exchange is about as valid as a three-dollar bill in the real world of operations. No Exchange server simply runs Exchange. Instead, real servers run a mixture of Exchange and other software, including antivirus and antispam detectors, and so on, all of which steal some CPU cycles, memory, and I/O.
- New advances, such as hyperthreading and 64-bit Windows, will continue to appear to drive performance envelopes upward. However, operational considerations often limit the number of mailboxes that you want to support on a single server. The old adage of not putting all of your eggs in one basket holds true today. Against this argument, it is generally true that organizations operate far too many servers today and server consolidation is a trend that will continue for the foreseeable future.

Because its ability to deliver great performance decreases the further you get from the datacenter, even the best-balanced and most powerful server will not satisfy users all the time. The reasons for this include:

- Network speed and latency: If users connect across slow or high-latency links, their perceived access and performance are gated by the amount of data transferred across the link. You can install faster computers, but it will not make much difference to the users at the end of such links.
- Clients: Each client differs in its requirements. A POP client makes minimal demand when compared with Outlook, but the latest version of Outlook can work in cached Exchange mode to speed perceived performance on the desktop. Outlook Web Access is highly sensitive to bandwidth.
- User workload: If users are busy with other applications, some of which also use network links, the performance of their Exchange client might suffer and they might blame Exchange.

All of this goes to prove that no matter how well you measure performance and then configure systems before you deploy Exchange, user perception remains the true test.

8.2.1 Performance measuring tools

Microsoft provides three tools to assist you in measuring the performance of an Exchange server:

- LoadSim

- Exchange Stress and Performance (ESP)
- JetStress

LoadSim is the oldest tool, since Microsoft first engineered it for Exchange 4.0 to generate a measurable workload from MAPI clients. ESP serves roughly the same purpose for Internet clients (including Outlook Web Access), while JetStress generates low-level database calls to exercise the I/O subsystem. You can download these tools from Microsoft's Exchange site at www.microsoft.com/exchange.

LoadSim and ESP both work by following a script of common operations that you expect users to take (creating and sending messages, scheduling appointments, browsing the GAL, and so on). You can tweak the scripts to create heavier or lighter workload. Usually, one or more workstations generate the workload to exercise a server, each of which follows the script and generates the function calls to perform the desired operations. The workstations do not have to be the latest and greatest hardware, since even a 700-MHz Pentium III-class machine is capable of generating the equivalent workload for 600 or so clients. Note that LoadSim does some things that make it very unsuitable for running on any production server. For example, when LoadSim creates accounts and mailboxes to use during the simulation, it gives the new accounts blank passwords. You can imagine the opinion of your security manager if you create hundreds of accounts with blank passwords in your production environment. For this reason, always run LoadSim on test servers, but equip those servers with hardware that is as close as possible, if not identical, to the configuration used in production.

JetStress falls into a different category, because you do not use this tool to measure the overall performance of a server. Instead, JetStress exercises the storage subsystem by generating calls to stress the physical disks, controllers, and cache to identify if a configuration is capable of handling a specified workload. Another way of thinking about JetStress is that it mimics the work done by the Store process, whereas the other tools aim to exercise a complete Exchange server. While the Store is central to Exchange, many other components affect the overall performance of an Exchange server, such as the Routing Engine. The Store places the heaviest load on the storage subsystem and that is what JetStress attempts to measure. Unlike the other tools, JetStress does not come with a pretty interface and does not generate nice reports. You have to be prepared to interrogate the system performance monitor to capture data that you later analyze. In addition, while LoadSim and ESP work on the basis of operations (such as sending a message to two recipients) that you can easily associate with time, JetStress requires detailed knowledge of Windows performance and storage fundamentals if you are to make sense of its results. It is probably fair to say that any Exchange system administrator can run and understand LoadSim, but JetStress requires you to do more work to understand how to change hardware configurations to improve performance based on the data it generates.

8.2.2 The difference between vendor testing and your testing

Hardware vendors typically use a standard benchmark workload called MMB2 for Exchange 2000 and MMB3 for Exchange 2003[2] when they test new servers. MMB2 is a modification of the original MMB workload and represents the workload generated by average office workers, if you could ever find one of these strange beasts. MMB3 is an evolution of MMB2, but differs in that it attempts to reproduce the different load generated by Outlook 2003 clients that use cached Exchange mode. Client-side caching changes server workload and may affect overall system performance, but it is only one aspect of Exchange 2003 performance. Microsoft has incorporated other factors into MMB3 (such as the use of rules, query-based distribution groups, and search folders) that increase client demand on a server, so a typical MMB3 result (in terms of number of mailboxes supported by a server) is lower than MMB2. Therefore, you cannot take a server result for Exchange 2000 and compare it with a result reported for Exchange 2003, because it is not an apple-to-apple comparison. You need to use the LoadSim 2003 version to perform benchmarks based on the MMB3 workload. A similar situation occurred when Microsoft changed the original MMB benchmark to MMB2 with the introduction of Exchange 2000.

All benchmarks attempt to prove one thing: that a server can support many more Exchange mailboxes than any sane administrator would ever run in production. To some extent, the benchmarks are a game played out by hardware vendors in an attempt to capture the blue riband of Exchange performance. It is nice to know that a server will support 12,000 mailboxes, but you always have to realize that, despite Microsoft's best effort to refine the MMB workloads, real users generate workloads very different from simulations for the following reasons:

- Real servers run inside networks and experience all of the different influences that can affect Exchange performance, such as losing connectivity to a GC.

[2] Microsoft does not endorse any benchmark results gained by running MMB2 against Exchange 2003 servers.

- Real servers run much more than Exchange. For example, antivirus detection software can absorb system resources that inevitably affect overall system performance. Some informal benchmarking of leading antivirus software shows that it can absorb 20 percent to 25 percent CPU, as well as virtual memory, with an attendant reduction on the number of supported mailboxes. Multi-CPU systems tend to be less affected by add-on software, because the load is spread across multiple processors.
- Benchmarks usually test the performance of single servers and ignore complex configurations such as clusters.
- Benchmarks do not usually incorporate complex storage configurations such as SANs, but shared storage is a prerequisite for any server consolidation exercise. Storage can significantly affect server performance, especially for database applications such as Exchange, which is the reason why vendors avoid complex storage configurations in benchmarks. They also tend to use RAID 0 volumes to hold the Store databases. This ensures performance, but you would never use RAID 0 for Store databases on production servers.
- The Store databases on real-world servers include a much wider variety of attachment types than found in the measured setup of a test database. For example, you do not typically use test databases that include huge PowerPoint attachments, yet any corporate Exchange server is littered with these files.
- The performance of all servers degrades over time due to factors such as disk fragmentation.

If you just read these points, you might conclude that there is no point in paying any attention to vendor benchmarks and running your own benchmark tests may not deliver worthwhile results. This is an oversimplification of the situation. The results of a vendor benchmark performed using a standard workload (remember that the MMB3 workload is preferred for Exchange 2003) gives you a baseline to measure different system configurations against each other. You can understand the impact of installing multiple CPUs in a server, the difference an increase in CPU speed makes, or how different storage controllers and disks contribute to overall system performance. All of this assumes that vendors perform the benchmarks according to the "rules" laid down by Microsoft and do not attempt anything sneaky to improve their results. Many consultants take the benchmark results reported by vendors and adjust them based on their own experience to create recommendations for production-quality server configurations.

Other factors, such as the “keeping all your eggs in one basket” syndrome and the time required to take backups (and, more importantly, restores) of large databases, reduce the tens of thousands of mailboxes that some benchmarks report to a more supportable number. For example, an HP quad-CPU (2 GHz) Proliant DL580 with 4 GB of memory benchmarks at 13,250 mailboxes, but you would never run this number in production. Experience of most corporate-style deployments indicates that 4,000 is closer to a long-term supportable number.

The decision to run your own benchmarks is harder to make because of the effort required. You can run LoadSim on a server just to see how it responds, but this will not generate a measurement that you can use in any serious sense. To create a proper benchmark you need:

- Dedicated servers to host Exchange and the Active Directory (DC and GC), as well as the workstations to generate the workload.
- A similar software configuration on the Server Under Test (SUT) that you intend to run in production. In other words, if you want to run a specific antivirus agent on your production servers, it should be installed and running on the SUT too. The same is true if you intend to host file and print services or any other application on the production servers—these have to be factored into the equation.
- Access to the same storage configuration that you plan to deploy in production. If you want to use a SAN, then you must connect the SUT to the SAN and use the same controller and disk layout as planned for production. Because Exchange performance is so dependent on the Store, you can drastically affect overall performance by changing storage characteristics.
- Apart from basic disk layout of the Exchange files (database, logs, and binaries), you should place the SMTP and MTA work directories and the message tracking logs in the same locations that they have in production.

Apart from all of this, all you need is time to prepare the test, run the benchmark, and then analyze the captured data. Do not expect to get everything right on the first run, and be prepared to run several tests, each of which lasts six hours or more (one hour to normalize the load, four hours to measure the SUT being exercised, one hour to complete the test).

During the same time, you may want to take advantage of the realistic configuration you create for the test servers to validate that assumptions about backup and restore times are correct, that antivirus and archiving tools work, that operational procedures are viable, and so on. You may also want to use other tools, such as Intel's IOmeter, to measure the base performance of storage volumes or other components.

Of course, you can take the view that every server available today is easily capable of supporting thousands of Exchange mailboxes and ignore benchmarks completely. This is a viable option if you then buy high-quality server hardware based on attributes other than just speed, including:

- Vendor support
- Additional features, such as the ability to boot or otherwise manage the server remotely
- Form factor (some sites prefer blade servers because of their reduced rack size or form factor)
- Server compatibility with the storage infrastructure

Even if you do ignore benchmarks, it is still worthwhile to build some test servers based on the desired configuration and validate it before proceeding to full deployment.

8.3 Cloning, snapshots, and lies

The traditional approach to backups saves an offline or online copy of data to tape. Tape throughput, better software, and increased automation have all improved and increased the ability to manage backups, but the amount of data to be processed has increased at a faster rate. The tape drives and libraries deployed to back up Exchange 5.5 servers with 10-GB Mailbox Stores may struggle to handle the demands of a modern Exchange server with multiple large databases. If you cannot back up your databases in a reasonable time or, more importantly, quickly restore your databases from a backup set should a disaster occur, then you inevitably need to limit database sizes. Limiting database size then restricts the number of mailboxes you can support on a server or limits the size of the mailboxes you can provide to users. Given that most users struggle to cope with a small mailbox, especially in corporate environments, and that administrators want to consolidate small servers into larger servers, we need a way to back up and restore large databases as quickly as possible. This has been a requirement since Exchange 5.5 lifted the 16-GB limit for a database, and the need has become increasingly urgent as servers become more powerful and storage costs decrease.

Backups to disk are always faster than tape and you can use disk instead of tape if that is your preferred backup medium. However, it is difficult to keep enough free disk space available to process backups, so it is unusual to find this implementation. Volume cloning and snapshots—sometimes referred to as "hot backups"—have attracted a lot of attention in the last few years. The marketing term for cloning is Business Continuance Volumes (BCV), which describes the intention behind the technology: reduce the possible downtime due to data unavailability to a minimum. Up to now, you need to deploy specific storage technology to be able to take snapshots, and some limitations exist in application support. Now, the availability of Volume ShadowCopy Services (VSS) in Windows 2003 brings this technology into the mainstream.

Clones and snapshots use different mechanisms to duplicate data so that you can create point-in-time copies. Clones are physical copies of data and are based on RAID 0+1 technology, so the technology to create a clone has been in storage architectures for a long time. You create a clone by establishing a new member of a RAID 0+1 mirror set so that the controller duplicates any data written to the mirror set automatically to all member drives. To create the clone, you split one of the members from the mirror set. The application can continue working, because the mirror set still exists and you can proceed to back up the newly created clone to create a backup save set for long-term storage. When the backup is complete, you can then reintroduce the drive into the mirror set to restart the cycle. Clones hold a complete copy of the data on a volume, so you can use a clone to very rapidly recover data should the need arise. The right-hand pane in Figure 8.3 shows the concept in action after you split off the clone from the mirror RAID set containing production data.

As its name implies, a snapshot is a point-in-time picture, or mapping, of the physical blocks on a volume. You can also think of a snapshot as a logical copy of data. When you create a snapshot, the blocks mapping the original file on disk are maintained, and the snapshot is created from a combination of the original blocks (the starting point) and any blocks of data that hold data that has changed since. The left-hand panel in Figure 8.3 illustrates how a snapshot is built from original and changed blocks.

Implementing hot backup technology is not simply a matter of plugging in appropriate storage technology. Storage systems commonly support many operating systems, so they include the necessary hardware and firmware support for cloning and snapshots. The operating system then implements the necessary support within its file system and drivers and then applications come along to take advantage of whatever they can from the new facilities. Applications include backup utilities as well as the applications that generate data. A complex collection of relationships and depend-

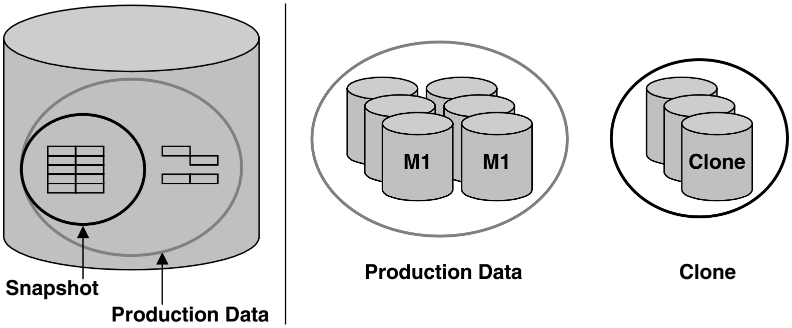


Figure 8.3 *Snapshots and clones.*

encies needs to come together before you can truly generate and depend on hot backups.

The Exchange developers knew that Windows would eventually include the necessary support for hot backups, so they never sought to build their own implementation into the Exchange backup API or ESE. From 1997 onward, storage vendors decided not to wait for Microsoft and began to build their own implementations. These solutions enabled pseudo hot backups for both Exchange 5.5 and 2000 by working around the essential fact that the Store is a transactional database with no way to become fully consistent unless it shuts down. Some solutions addressed the problem by closing down the Store process before they take a hot backup. This forces the Store to flush any outstanding transactions and creates a consistent version of the database on disk. The normal flow is to break the mirror set as soon as the Store process shuts down, followed by a fast restart of the Store process to restore service to users. You can then mount the disk that contains the clone on another server and begin a tape-based backup to take a copy of the database. Users usually do not notice the interruption in service if you schedule it early in the morning or at a time of low demand. The time taken to close down the Store, break the mirror set, and restart is a matter of a few minutes, and these operations are usually automated through vendor-provided scripts that can handle backup and restore operations.

Recent variations on this theme involve breaking the mirror set while the Store is still running and then taking a backup of the database copy plus all its transaction logs. While this technique works—most of the time—it exhibits inherent flaws, because the Exchange Store does not support backups made in this manner. In effect, what you are doing is creating a backup of an inconsistent copy of the database from disk. In this scenario, the database copy is always inconsistent, because the Store has not had the chance to commit outstanding transactions, so you rely on the roll-forward capability to capture transactions in the logs whenever you need to make the database consistent—if you ever need to restore and use this copy of the database. In addition, because you halt Store processing in a very abrupt manner, there is no guarantee that the internal structures are intact, because you never know what the Store was doing at the exact moment that you take the hot backup. Finally, unlike normal backups, the Store does not perform checksum validation on pages as it writes them out to the backup media, so you could easily take a copy of an already corrupt database and make it even more corrupt. It is not surprising that Microsoft does not support this technique for Exchange 2000 and leaves any support issues that occur, such as recovering a corrupt database, to the vendor that supplied the storage, scripts, and other technology used to take the backup. If you insist on using such a method to take backups, you should also take frequent full tape backups, so that you are sure that you always have something to recover. Or, even better, upgrade to Windows 2003 and Exchange 2003 and find storage and backup vendors that support the Volume ShadowSet-Copy Services API and take proper, fully supported hot backups.

Once you have split a mirror set to create a copy of the database, you need to take a file-level backup to copy the data to tape. In this respect, backups take roughly the same amount of time as they would when you take an online backup of the Store, but restores can be dramatically faster if you can use an on-disk copy of the database. For example, tests performed by HP using its Rapid Restore Solution for Exchange demonstrated that it is possible to restore a 41-GB storage group (two Mailbox Stores and one Public Folder Store) in 6 minutes from a clone, compared with 1 hour 45 minutes from tape. Clearly, restoring a database or storage group takes the same amount of time if the disk clone is not available and you have to use the tape backup.

There is no doubt that hot backups work when properly implemented and integrated into a backup and restore plan. Everything works extremely well at the storage hardware level and all of the issues occur with the application. Exchange knows nothing about the hot backup, because you must stop the Store to remove the clone of the database. The other problem is the variety of approaches and implementations taken by vendors, because they build off their own platform. This is not a problem when you always work with a single technology, but it can be a problem where different groups deploy a variety of technologies inside a large organization.

8.3.1 Volume ShadowCopy Services

VSS provides the necessary architecture for application and storage vendors to support hot backups using a common API. VSS incorporates application synchronization (how to control the taking of a hot backup), discovery and listing of shadow copies (both clones and snapshots), and a plug-and-play framework for backup components from different vendors.

Vendors use the VSS API to develop provider processes that maintain data about physical clones and snapshots and can expose them to the operating system and applications. VSS processes can contain kernel-mode (device drivers) and user-mode code (the user interface and processing to take a hot backup). Windows 2003 contains a software-based VSS provider as part of the operating system. Storage vendors such as HP and EMC have hardware providers to allow their storage technology to participate in VSS backups.

In addition to VSS providers, vendors also develop VSS requesters. These applications coordinate the processing required to take a backup or perform a restore. VSS providers and writers do the actual processing to create a backup or perform a restore. Typical processing incorporated in a requester includes the identification of the volumes to include in a backup, requesting data from different writers (Exchange, SQL, Oracle, other applications), and communication through a user interface. You can think of the requester as the center coordination point for hot backup operations. It has to communicate with the applications to ask them to provide data and with the providers to identify how to back up the data from volumes under their control. Traditional backup applications such as Legato and Backup Exec are likely VSS requesters.

In VSS terminology, the applications that control data permit writers to allow VSS requesters to include data from the applications in shadow set copies. Exchange 2003 is the first version to support VSS and incorporates the necessary support to be a VSS writer in the Store process. Each application controls its own data and may use dramatically different ways to access and maintain that data, but this complexity is hidden from requesters by the ShadowCopy interface, which ensures data integrity and consistency across applications. During ShadowCopy processing, application writers perform operations such as prepare (get data ready for processing), freeze (prevent writes while processing proceeds), thaw (allow I/O to begin again), and normalize (complete operations). Each writer defines its needs through a set of XML metadata that the requester can interpret and process during a backup or restore operation. For example, the metadata published by Exchange 2003 includes details of the databases and storage groups to include in an operation (the components), whether a reboot is required after the operation completes, what type of restore operations are possible, and so on.

The fact that the VSS framework exists means that a common way of implementing hot backups exists, but you must upgrade many components before you can take hot backups. The storage hardware, operating system, applications, and backup software must work together, so it is important that you have the correct versions installed to make everything work.

8.3.2 Using VSS with Exchange 2003

Assuming that you have the right combination of backup software and hardware, you can plan to incorporate snapshots into your backup strategy. VSS backups for Exchange are always taken at the storage group level, so a full VSS backup contains the complete set of databases in the storage group and the transaction logs, and the log set is truncated (old logs are deleted) after a successful backup. Exchange also supports VSS copy backups at the storage group level, with the major difference being that the transaction logs are not truncated. Incremental and differential backups only copy transaction logs, and, once again, everything is done at the storage group level, with the difference being that the logs are either truncated (incremental) or not (differential). Exchange writes the information about the components assembled into the backup in an XML document that it then provides to the VSS requester, which then proceeds to copy the information stated in the document.

In a restore situation, the backup application is only responsible for extracting information from the backup and restoring it to disk. The Store is responsible for making the database consistent through log replays using the normal soft recovery process, which the Store initiates when you remount databases after recovering them from a backup set.

You should view snapshots as complementary to tape backups, but not as a complete replacement. Streaming to tape retains some advantages that snapshot backups do not have. For example, the Store calculates a checksum for each page as it streams data out to the backup media, but obviously this does not happen when you take a snapshot. The same is true for empty page zeroing (or scrubbing), which the Store can do as it processes pages during a backup, but it cannot be done for a snapshot. As long as you are confident that your databases are in good order, you can proceed to take snapshots with confidence, but it is still a good idea to take an old-fashioned tape backup from time to time, just to be sure.

Software and hardware vendors are likely to collaborate to create special packages to handle VSS-based Exchange backup and restores in an automated manner. Because every hardware and software backup vendor will now use a common API to take hot backups, the benefit in these packages comes from the ease and speed in which the software and hardware combination processes backup and restore situations. Before buying any solution, investigate how easily it handles various disaster recovery situations, such as corrupt databases or logs, as well as the routine of daily backup operations. If a disaster occurs, you will be grateful if the solution automates the entire recovery process instead of leaving you to figure out how best to restore service.

8.4 Virtual Exchange servers

VMware's[3] ESX Server is a popular option for server consolidation on Intel systems. The idea is simple. Buy the biggest server you can find and then run software that creates logical partitions that support virtual servers. The software (VMware) runs on top of either Windows or Linux and allows you to install different operating systems and applications to form the virtual servers. Once you have virtual servers going, you can install applications to make the virtual servers productive. In concept, this seems very similar to the way that Windows clusters work. After all, Exchange runs as a virtual server supported by a physical server that is part of the cluster.

Is it a good idea to deploy some very large multi-CPU servers and consolidate smaller Exchange servers onto the system, running each as a virtual server? Different people will give different answers, but, at the end of the day, supportability rather than feasibility will probably influence your decision more.

There is no doubt that you can build Exchange servers (including clusters) on top of a VMware virtual server, so the question of feasibility does not occur. Many companies use VMware to test server operating systems and applications, but they do not necessarily take the next step to deploy the same configuration into production. This is where the question of supportability occurs.

As of mid-2003, Microsoft's position on the subject is clear.[4] It will only support a problem reported on a virtual server if you can replicate the same problem on a standard server. Microsoft does not include virtual servers in its test procedures, so it is difficult for it to provide support for complex applications such as Exchange and SQL in an environment that it does not test. In addition, most production Exchange servers do not simply run

[3] www.vmware.com.

[4] See Knowledge Base article 273508 for details.

Exchange. Instead, they support other utilities, such as migration tools, messaging connectors, antivirus checkers, backup products, and so on. It would be possible for Microsoft to test and validate Exchange on a virtual server, but including all possible permutations into a test plan for a platform that it does not build is asking a little much.

The answer today is that virtual servers are a good idea for testing complex applications, and they have a role to play for server consolidation projects for simple facilities, such as file and print services. However, until Microsoft fully supports virtual servers without the requirement to replicate problems on standard servers, it is difficult to argue a case to use virtual servers in production. Microsoft's purchase of the Connectix technology (to become the Microsoft Virtual Server product) in early 2003 will generate some interesting scenarios as product groups grapple with the need to support a Microsoft-branded virtual server. Interesting days lie ahead.

Server consolidation is a good idea, and, because many Exchange servers support relatively small user populations, Exchange is definitely a candidate for consolidation. This is especially true since network costs have come down, because it can be cheaper to pay for the increase in bandwidth to bring clients back to a small set of large servers in a datacenter than to keep a set of smaller servers distributed to multiple locations. Early Exchange deployments, those that have run since Exchange 4.0 and 5.0, are specific candidates for consolidation, a project that you might care to undertake in conjunction with a deployment of Outlook 2003 so that you can take advantage of its more efficient network use.

8.5 A brief history of clustering Exchange

> *A cluster is a parallel or distributed system that consists of a collection of interconnected whole computers that are utilized as a single, unified computing resource.*
>
> —Gregory Pfister, *In Search of Clusters,* 2d ed. 1998.

Microsoft introduced Exchange clusters in November 1997, when it released Exchange 5.5, the Enterprise Version of which supported Wolfpack 1.0, or Windows NT cluster services. Exchange's implementation as a clustered application—at least one that could take full advantage of active-active clustering—was incomplete and not every Exchange service could run on a cluster. In particular, a cluster could not host many of the connectors to legacy messaging systems such as Microsoft Mail. However, you could deploy the basic messaging infrastructure on clusters and use them as mailbox servers.

The two servers in an Exchange 5.5 cluster must match in terms of CPU and memory, and you need licenses for the enterprise editions of Windows NT 4.0 (or Windows 2000) and Exchange 5.5 for both servers. Additionally, the hardware must be certified by Microsoft and be included on the cluster Hardware Compatibility List (HCL). Because Exchange 5.5 only supports active-passive clusters, one of the servers is usually inactive, although some customers used the passive server either to run file and print services or host another application. The net result was that an Exchange 5.5 cluster is an expensive solution that requires substantial expertise to deploy. Cluster state transitions were often extended, and although Microsoft worked to eliminate the causes (RPC timeouts and other software glitches) and improved matters in service packs, Exchange clusters never took off, and only a small percentage of customers evaluated them—with an even smaller percentage (estimated at less than 1 percent of corporate customers) moving into deployment.

The low penetration achieved by clusters had a domino effect, since ISVs were reluctant to make their code work on an Exchange 5.5 cluster. Thus, while you could deploy clusters as mailbox servers, you could not protect them against viruses or install other popular ISV software, such as backup agents, fax connectors, and so on. This situation gradually improved, as ISVs updated their code to support clusters, but the lack of third-party software presented a huge hurdle for potential cluster deployments to overcome.

8.6 Second-generation Exchange clusters

The release of Exchange 2000 promised a new beginning for Exchange clusters. Many factors had changed to improve matters, including a new version of the underlying cluster software provided by Windows, the partitioning of the Store into storage groups to allow greater granularity during transitions, support of four-way active-active clusters, and the experience of almost three years of real-life deployments. Unfortunately, some of the same hurdles to customer acceptance of clusters remain, including:

- Complexity
- Cost
- Lack of support for third-party products

The lack of support from third parties is entirely due to market acceptance of clusters. Because clusters remain strictly a minority interest within the general Exchange community, third-party developers focus their efforts on supporting mainstream standard servers. The result is that it is often difficult to find a version of an add-on product for Exchange that supports a clustered environment.

Soon after Exchange 2000 shipped, customers began to report problems with memory management on clusters. The problems appeared on active-active clusters and caused Exchange to freeze and be unable to service client requests. Administrators also reported similar problems on high-end standard Exchange 2000 servers that handle heavy workloads over extended periods. SharePoint Portal Server 2001, which uses a modified version of the Exchange database engine and Store, can also run into memory management problems under heavy load.

Microsoft has steadily improved the quality and robustness of Windows clusters and the applications that support clusters, including Exchange. However, even six years after Microsoft first shipped Exchange clusters, the two biggest problems that cause operating issues with Exchange clusters are still the overall complexity of the solution and operational management. Microsoft now says that all support above two nodes must be active/passive. In other words, you must always keep a passive node available to handle failovers. To those used to other cluster implementations (such as VMSclusters), the need to keep a passive node around is a condemnation of Microsoft clustering technology.

8.6.1 The complexity of clusters

Successful operation of Exchange clusters requires:

- Appropriate hardware
- Attention to detail
- Administrator knowledge of Windows, Exchange, and cluster services
- Cluster-aware operational procedures and third-party products

Cluster hardware should include high-quality servers with balanced configuration of CPU and memory to allow the servers to handle workload equally. A SAN is almost mandatory for anything but entry-level clusters, so you need to pay attention to controller configuration and resilience, basic disk technology (speed and placement), file layout across available volumes, and so on. Commissioning procedures differ across storage technologies, so be sure that you take the right approach for the chosen technology.

Clusters depend on a complex interaction between hardware, operating system, applications, and people. You need to pay attention to detail to ensure that you properly install and configure the cluster before you introduce any application into the equation. This is particularly important when you deal with a SAN, since technology differs greatly across SANs provided by different vendors—and almost every production-quality cluster uses a SAN.

Applications often follow a different installation procedure on clusters. It is not enough to assume that Windows and Exchange behave only slightly differently on a cluster—practice makes perfect! For example, in the case of Exchange, you must manage the basic services that make up the application through the cluster administration tool rather than performing actions such as stop and start services through the Services Manager utility or ESM. However, because clusters are expensive, it is often difficult for administrators to get the necessary experience on clusters before deploying the first production cluster. Few test environments incorporate a fully configured production-quality cluster, but it is perfectly possible to commission an entry-level cluster and use that for testing. Many companies use virtual systems to test clusters, which is an effective approach to solve the need.

Administrators must understand how Microsoft has implemented cluster services for Windows and then what modifications occur for applications to support cluster services. For example, you can only install Exchange on a cluster in a mixed-mode site if another Exchange 2000/2003 server is already present, because some of the services (such as SRS) required for mixed mode cannot run on a cluster. Administrators must understand the differences between standard servers and clusters and understand how to manage and troubleshoot both environments, including how to correctly back up and restore a cluster, as well as how to cope with various disaster recovery scenarios, such as a catastrophic hardware failure.

It is essential that you modify operational procedures developed for standard servers for clusters. The software used to monitor servers and applications may not support clusters and may require a change or replacement. Some third-party software may not be supported and may force you to change the operational procedures to accommodate a different package. In addition, clusters are sensitive to change, so you must carefully plan and test any upgrades and installations of new software before you make changes to production environments.

8.7 Microsoft cluster basics

Microsoft clusters use the shared-nothing model, which means that each server owns and manages local devices (e.g., disks) as specific cluster resources. Clusters include common devices that are available to all of the nodes, but these are owned and managed by only one node at one time. For example, an Exchange virtual server that supports one storage group usually places its transaction logs on a volume, which we will call L: for the moment. The L: volume is visible to all of the servers in the cluster, but only the server that currently hosts the Exchange virtual server running the storage group can access L: at one time. If a failure occurs and the cluster transitions the virtual server to another physical server in the cluster, that server takes ownership of the L: volume.

8.7.1 Resources

Microsoft cluster management services take care of the complex interaction between the physical servers in the cluster, the virtual servers they host, and the resources such as disks that they use, including the management of the different network addresses (names and IP addresses) used by clients to access cluster resources. In this context, a resource is any physical or logical component that you can bring online or take offline within the cluster, but only a single server can own or manage the resource at one time. A network interface card (NIC) is an example of a physical resource, while an IP address is an example of a logical resource.

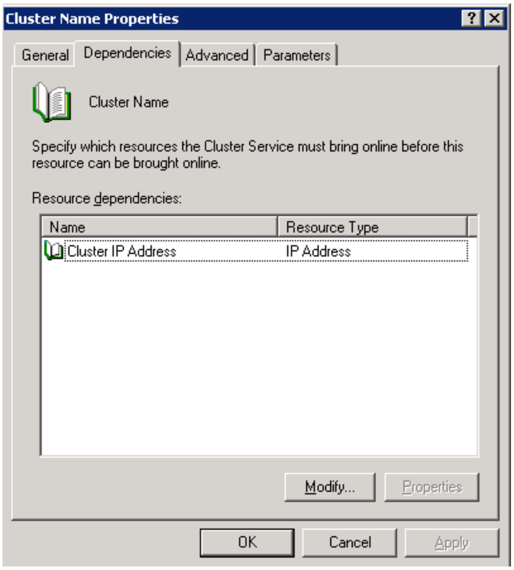
Each server in the cluster has its own system disk, memory, and copy of the operating system. Each server is responsible for some or all of the resources owned by the cluster, depending on the current state of the cluster. For example, in a two-node cluster, where one node has just failed, the single surviving node hosts its own unique resources (such as its system disk) as well as all the shared cluster resources and the applications that depend on those resources. When the failed server is available again, the cluster redistributes the shared resources to restore equilibrium.

8.7.2 Resource groups and other cluster terms

The number of resources used in a cluster can be quite large, so cluster services use resource groups as the fundamental unit of management within cluster; they also represent the smallest unit that can fail over between nodes in a cluster. Resource groups hold a collection of resources for both the cluster (its network name, IP address, etc.) itself as well as applications. For management purposes, clusters define Exchange virtual servers as resource groups. The shared-nothing model prevents the different nodes within the cluster from attempting to own resources or resource groups simultaneously, so all the resources that make up an Exchange virtual server must run on a single node. In fact, if you have enough processing power, you can run multiple Exchange virtual servers on a single physical computer—something that is interesting in the software laboratory but not recommended for production.

Resource groups can contain both logical and physical resources. For Exchange, the logical resources include the name of the virtual server and its IP address as well as the set of services that make up Exchange. The physical resources include details of any shared disks (used to hold the binaries, Store, and logs). Resource groups often have dependencies on other resource groups—conditions that must be satisfied before the resource group can come online. The properties of a resource or resource group state any dependencies that exist. For example (Figure 8.4), an Exchange virtual server cannot come online unless it has a valid IP address to allow clients to connect. You can only bring Exchange resources online in dependency order.

Dependencies also exist on standard Exchange servers, the best example being the Information Store service, which cannot start if the System Attendant is not running. Note that dependencies cannot span resource group boundaries, since this would complicate cluster management enormously



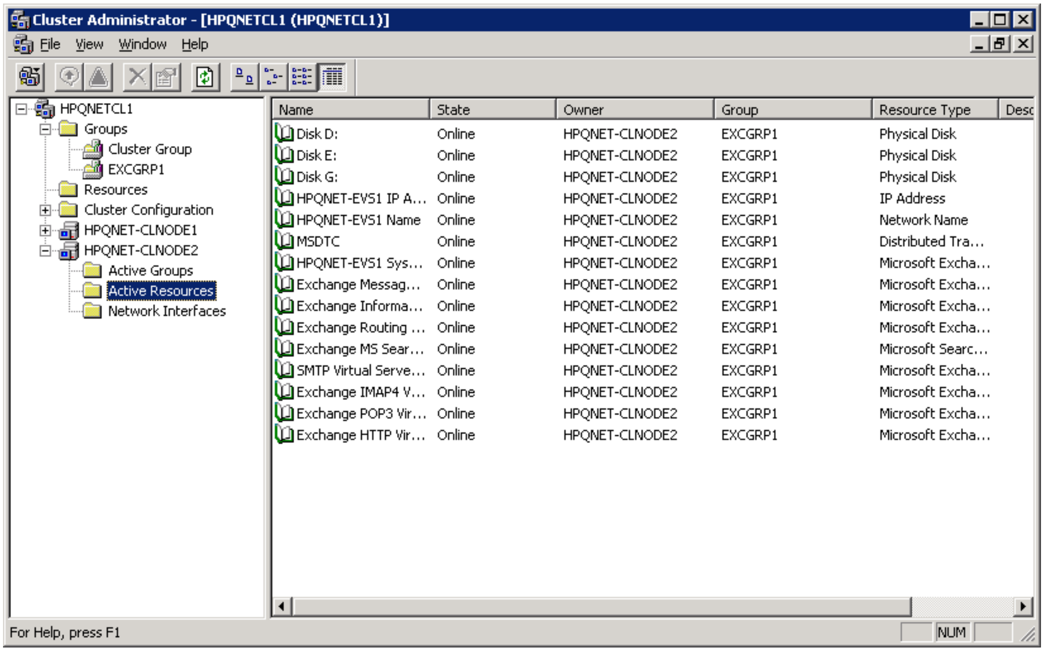
Cluster Name

Specify which resources the Cluster Service must bring online before this resource can be brought online.

Resource dependencies:

Name	Resource Type
Cluster IP Address	IP Address

Figure 8.4 *Resource dependency.*



Name	State	Owner	Group	Resource Type	Desc
Disk D:	Online	HPQNET-CLNODE2	EXCGRP1	Physical Disk	
Disk E:	Online	HPQNET-CLNODE2	EXCGRP1	Physical Disk	
Disk G:	Online	HPQNET-CLNODE2	EXCGRP1	Physical Disk	
HPQNET-EVS1 IP A...	Online	HPQNET-CLNODE2	EXCGRP1	IP Address	
HPQNET-EVS1 Name	Online	HPQNET-CLNODE2	EXCGRP1	Network Name	
MSDTC	Online	HPQNET-CLNODE2	EXCGRP1	Distributed Tra...	
HPQNET-EVS1 Sys...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	
Exchange Messag...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	
Exchange Informa...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	
Exchange Routing ...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	
Exchange MS Sear...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Searc...	
SMTP Virtual Serve...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	
Exchange IMAP4 V...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	
Exchange POP3 Vir...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	
Exchange HTTP Vir...	Online	HPQNET-CLNODE2	EXCGRP1	Microsoft Excha...	

Figure 8.5 *Cluster groups and resources.*

and create situations where resource dependencies might be scattered across various physical servers. In our example, you could not create a dependency for an Exchange virtual server on an IP address that is part of a different resource group.

Figure 8.5 shows the resource groups and resources for a very simple cluster. In this case, the cluster consists of one physical server. Even on a single-node cluster, the basic principles of a cluster still apply, so we can see details of cluster resources as well as the resources that make up an Exchange virtual server. Notice that Exchange represents all of the services that you would expect to see on a standard Exchange server to the cluster as resources. The resources also include some elements that are under the control of IIS, such as the different protocol virtual servers used by Exchange (IMAP, SMTP, POP3, and HTTP).

Before going too far, we should first explain the various names used in a cluster, which include:

- The name of the cluster (in this case, HPQNETCL1)
- The names of each of the physical servers (nodes) that make up the cluster—here we have two physical servers (HPQNET-CLNODE1

and HPQNET-CLNODE2), which are the computers that Windows, cluster services, and applications such as Exchange run on.

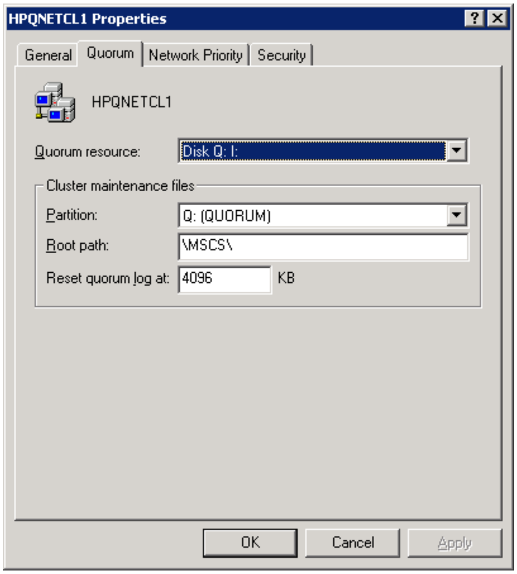
- The names of each of the virtual servers that the cluster hosts: Clients do not connect to the cluster, nor do they connect to a physical computer. Logically, they look for the name of the Exchange server that holds their mailboxes. This cluster supports only one Exchange virtual server (HPQNET-EVS1), which runs on a physical server that is part of the cluster. Cluster services move a virtual server from one physical server to another within the cluster. Moves do not affect clients, because the cluster services take care of redirecting incoming client requests to the combination of hardware and software that represents the virtual server within the cluster at that point in time.

It makes sense to decide upon and use naming conventions for cluster systems and virtual servers so that their purpose is obvious at a glance. Some practical definitions of other important cluster terms include:

- Generically cluster aware: A mode where an application is cluster aware by using the generic cluster support DLL, meaning that the application is not specially upgraded to support clusters and can only operate on one node of the cluster at a time. Microsoft supplies the generic cluster support DLL to allow vendors (including its own development groups) to run applications on a cluster with minimum effort.
- Purpose-built cluster aware: A mode where an application is cluster aware through special application-specific code, which enables the application to take full advantage of cluster capabilities, and the application can run on all nodes of the cluster concurrently. Exchange implements its support for clusters through EXRES.DLL, which the setup program installs when you install Exchange on a cluster. EXRES.DLL acts as the interface between the Exchange virtual server and the cluster. At the same time, setup installs EXCLUADM.DLL to enable the cluster administration program to manage Exchange components so that they respond to calls such as “come online,” “go offline,” and so on. With these components installed, the core of Exchange can run on all nodes in a cluster (active-active mode), but some older or less frequently used code does not support this mode or cannot run at all on a cluster.
- Cluster registry: A separate repository to the standard system registry used to track the cluster configuration and details about resources and resource groups. The quorum resource holds the cluster registry.

A mechanism called “global update” publishes information about cluster changes to members of the cluster.

- Members (or nodes): The physical computers that make up the cluster. In production, clusters range from a two-node cluster to an eight-node cluster (on Windows Server 2003 Enterprise Edition), although you can build a single-node cluster for training or test purposes.
- Quorum resource (Figure 8.6): Most Windows clusters use a disk quorum, literally a physical disk that holds the registry and other data necessary to track the current state of the cluster plus the necessary information to transfer resource groups between nodes. While Exchange 2003 does not have any direct involvement with quorums (this is the responsibility of the OS), you can install Exchange clusters with disk quorums as well as local and majority node set quorums. A local quorum is only available to a single-node cluster (also known as a “lone wolf” cluster), which you would typically use in a disaster recovery scenario, while a majority node set quorum is usually found in stretched clusters where multiple systems use a disk fabric to communicate across several physical locations. In this situation, network interrupts may prevent all the systems from coming online at the same time, so majority set quorums allow the cluster to function once a majority of the nodes connect. For example, once five nodes in an eight-node cluster connect, a quorum exists.



HPQNETCL1

Quorum resource: Disk Q: I:

Cluster maintenance files

Partition: Q: (QUORUM)

Root path: \MSCS\

Reset quorum log at: 4096 KB

Figure 8.6 *The cluster quorum.*

Cluster purists may not agree with some of the definitions offered here. However, they are functional rather than precise and provide enough foundation to proceed.

8.7.3 Installing Exchange on a cluster

You must have the following resources to install Exchange on a cluster:

- An IP address and a network name for each virtual server. You cannot use dynamic IP addresses.
- The physical hardware for the cluster nodes, ideally balanced in terms of CPU (number and speed) and memory.
- Physical shared disk resources configured to hold the Store databases and transaction logs.

It is best to create a separate resource group for each Exchange virtual server in the cluster and then move the storage used for the databases and so on into the resource group. Installing Exchange on a cluster is no excuse to ignore best practice for the Store, so make sure that you place the databases and the transaction logs on separate physical volumes. Interestingly, the number of available drive letters may cause some design problems on very large Exchange clusters, since you have to allocate different drive letters to each storage group and perhaps the volume holding the transaction logs for each storage group. This problem does not occur when you deploy Exchange 2003 on Windows 2003 clusters, because you can use mount points to overcome the lack of available drive letters. By convention, clusters use drive Q: for the quorum resource and M: for ExIFS (such as all other Exchange servers).

Remember that on Windows 2003 clusters, you have to install components such as IIS and ASP.NET on each node before you can install Exchange. Exchange 2003 requires Microsoft DTC, so you have to create it as a cluster resource before you install Exchange.

Equipped with the necessary hardware, you can proceed to install the cluster and elect for an active-passive or active-active configuration (for a two-node cluster) up to an eight-node cluster where seven nodes are active and one is passive. Installing the cluster is reasonably straightforward, and defining the number of storage groups and databases is the only issue that you have to pay much attention to afterward. The enterprise edition of Exchange 2000 or 2003 supports up to four storage groups of five databases. Each virtual server running in a cluster can support up to these limits, but such a configuration runs into problems when a failure occurs, because Exchange cannot transfer the storage groups over to another cluster node. Consider this scenario: You have two virtual servers, each configured with three storage groups of three databases. A failure occurs and Exchange attempts to transfer the three storage groups from the failed server to the virtual server that is still active. The active virtual server can accept one storage group and its databases and then encounters the limit of four storage groups, so a full transition is impossible. Cluster designs, therefore, focus on failure scenarios to ensure that remaining virtual servers can take the load and never exceed the limits. In a very large cluster, where each virtual server supports two storage groups, you may only be able to handle a situation where two or three servers fail concurrently, depending on the number of storage groups each virtual server supports.

8.7.4 What clusters do not support

The vast majority of Exchange code runs on a cluster, but you should think of clusters as primarily a mailbox server platform, because of some limitations on connector support. In addition, you never think of clusters for front-end servers, because these systems do not need the high level of resilience and failover that clusters can provide and they are too expensive.

Most of the components not supported by clusters are old or of limited interest to the general messaging community. These are:

- NNTP
- Exchange 2000 Key Management Server
- Exchange 2000 Instant Messaging
- Exchange 2000 Chat
- MTA-based connectors (GroupWise, Lotus Notes, cc:Mail, Microsoft Mail, IBM PROFS, IBM SNADS)
- Exchange 2000 Event Service
- Site Replication Service

You can break down these components into a set of old connectors, which, depending on the MTA, are being phased out in favor of SMTP connections; subsystems such as Instant Messaging, which Exchange 2003 does not support; and the Site Replication Service, which is only needed while you migrate from Exchange 5.5. The exception is NNTP, but very few people use Exchange as an NNTP server or to accept NNTP newsfeeds simply because other lower-cost servers are better at the job. In addition, using a cluster for NNTP is total overkill.

8.7.5 Dependencies

Figure 8.7 illustrates the resource models implemented in Exchange 2000 and Exchange 2003. The resource model defines dependencies between the various components that run in a cluster. The Exchange 2000 resource model centers on the System Attendant and the Store, so if either of these processes fails, it affects many other processes. By comparison, the Exchange 2003 resource model removes many of the previous dependencies on the Store and makes the System Attendant process the sole "must-be-alive" process for a cluster to function. The change improves failover times by reducing the processes that have to be stopped and restarted if a problem occurs; this is entirely logical, because the protocol stacks have a dependency on IIS rather than the Store.

8.7.6 Clusters and memory fragmentation

When Microsoft released Exchange 2000, system designers looked forward to a new era of high-end email servers built around active-active clusters, a promise that was further embellished when Exchange 2000 SP1 provided the necessary support for Windows 2000 Datacenter Edition to enable four-way active-active clusters. System designers look to clustering to provide high degrees of both system resilience and availability and often as a

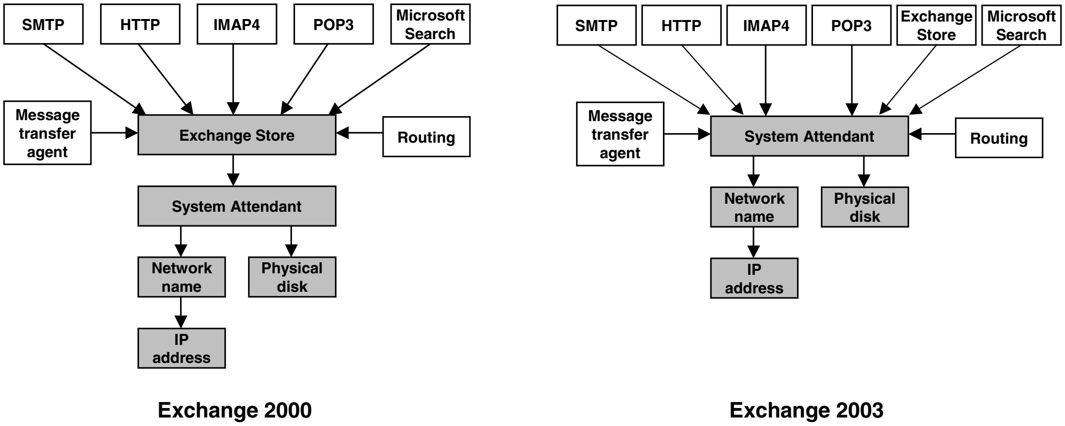


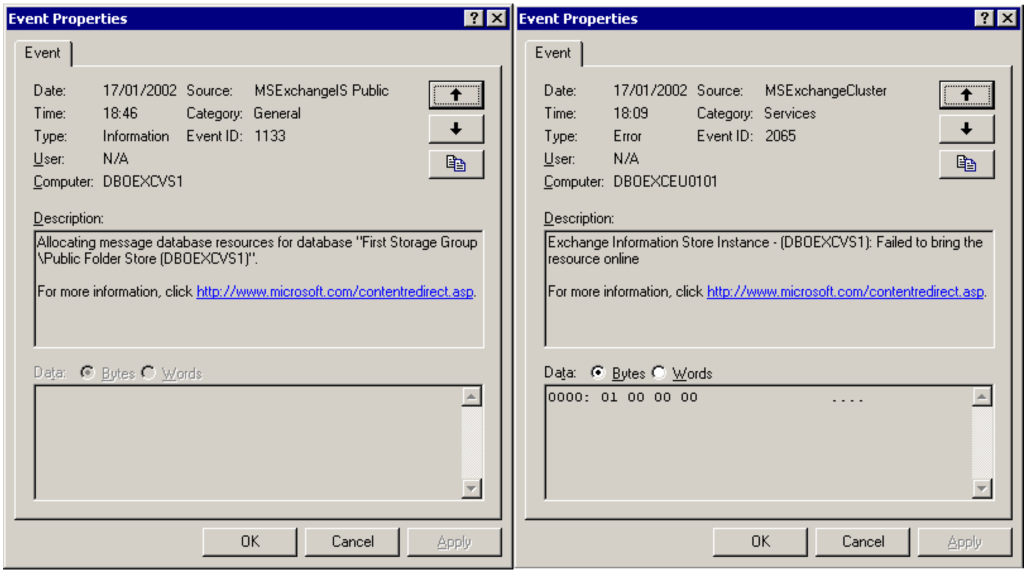
Figure 8.7 *Exchange cluster resource models.*

way to consolidate a number of servers into a smaller set of large clusters. Exchange 5.5 supports active-passive two-node clustering, meaning that one physical system or node actively supports users while its mate remains passive, waiting to be brought into action through a cluster state transition should the active system fail. This is an expensive solution, because of the need for multiple licensed copies of the application, operating system, and any associated third-party utilities (e.g., backup or antivirus programs), as well as the hardware. Active-active clusters provide a better "bang" for your investment, because all of the hardware resources in the cluster are available to serve users.

Unfortunately, active-active clusters ran into virtual memory fragmentation problems within the Store, and this issue prevents Exchange from taking full advantage of clustering. The way that Exchange implements Store partitioning is by establishing a storage group as a cluster resource that is transitioned (along with all its associated databases and transaction logs) if a problem occurs. However, while everything looked good on the theoretical front, clustering has not been so good in practice. Exchange uses dynamic buffer allocation (DBA) to manage the memory buffers used by the Store process. DBA sometimes gives administrators heart palpitations, because they see the memory used by STORE.EXE growing rapidly to a point where Exchange seems to take over the system. This behavior is by design since DBA attempts to balance the demands of Exchange to keep as many Store buffers and data in memory as possible against the needs of other applications. On servers that only run Exchange it is quite normal to see the Store take large amounts of memory and keep it, because there is no other competing applications that need this resource.

During normal operation, Windows allocates and deallocates virtual memory in various sizes to the Store to map mailboxes and other structures. Virtual memory is sometimes allocated in contiguous chunks, such as the approximately 10 MB of memory that is required to mount a database, but as time goes by it may become difficult for Windows to provide the Store with enough contiguous virtual memory, because it has become fragmented. In concept, this is similar to the fragmentation that occurs on disks, and usually it does not cause too many problems—except for cluster state transitions.

During a cluster state transition, the cluster must move the storage groups that were active on a failed node to one or more other nodes in the cluster. Storage groups consist of a set of databases, so the Store has to be able to initialize the storage group and then mount the databases to allow users to access their mailboxes. You can track this activity through event



Date: 17/01/2002 Source: MSExchangeIS Public
Time: 18:46 Category: General
Type: Information Event ID: 1133
User: N/A
Computer: DBOEXCVS1

Description:
Allocating message database resources for database "First Storage Group \Public Folder Store (DBOEXCVS1)".

For more information, click http://www.microsoft.com/contentredirect.asp.

Data: ● Bytes ○ Words

Date: 17/01/2002 Source: MSExchangeCluster
Time: 18:09 Category: Services
Type: Error Event ID: 2065
User: N/A
Computer: DBOEXCEU0101

Description:
Exchange Information Store Instance - (DBOEXCVS1): Failed to bring the resource online

For more information, click http://www.microsoft.com/contentredirect.asp.

Data: ● Bytes ○ Words

```
0000: 01 00 00 00        ....
```

Figure 8.8 *Allocating resources to mount a database, and a failure.*

1133 in the application event log (see left-hand screen shot in Figure 8.8). On a heavily loaded cluster, it may be possible that the Store is not able to mount the databases, because no contiguous virtual memory or not enough contiguous virtual memory is available, in which case you will see an event such as 2065, shown in the right-hand screen shot in Figure 8.8. Thus, we arrive at the situation where the cluster state transition occurs but the Store is essentially brain dead, because the databases are unavailable.

Now, it is worth noting that this kind of situation only occurs on heavily loaded systems, but you will remember that server consolidation and building big, highly resilient systems is one of the prime driving factors for system designers to consider clusters in the first place. After receiving problem reports, Microsoft analyzed the data and realized that it had a problem. It began advising customers to limit cluster designs to lower the numbers of concurrently supported clients (1,000 in Exchange 2000, 1,500 in SP1, and 1,900 in SP2, going a little higher with SP3[5]) when running in active-active mode.

Because MAPI is the most functional and feature-rich protocol, MAPI clients usually generate the heaviest workload for Exchange, so these num-

[5] At the time of writing, Microsoft has not yet completed its testing to identify suggested levels of mailbox support for Exchange 2003.

bers reflect a MAPI load. Outlook Web Access clients generate much the same type of demand as MAPI. The functions exercised through other client protocols (such as IMAP4 and POP3) typically generate lower system demand and may result in a lesser workload for the server, so it is possible that you will be able to support more client connections before the virtual memory problem appears. Your mileage will vary, and a solid performance and scalability test is required to settle on any final cluster configuration. The test must be realistic and include all of the software incorporated in the final design.

From Exchange 2000 SP3 onward, the Store includes a new virtual memory management algorithm, which changes the way it allocates and frees virtual memory. The key changes are:

- JET top-down allocation: Prior to SP3, the JET database engine allocates virtual memory for its needs from the bottom up in 4-K pages. Other processes that require virtual memory (Store, epoxy, IIS, etc.) are also allocating virtual memory from the bottom up, but they allocate memory in different sizes. This method of managing memory can result in virtual memory fragmentation when multiple processes are continuously requesting and releasing virtual memory. SP3 changed the JET virtual memory allocation to a top-down model to eliminate contention for resources with other system processes. In practical terms, the top-down model results in less virtual memory fragmentation, because small JET allocations pack together tightly. It also allows the Store process to access larger contiguous blocks of virtual memory over sustained periods of load.
- Max open tables change: When the JET database engine initially starts, it requests the virtual memory necessary to maintain a cache of open tables for each storage group. The idea is to have tables cached in memory to avoid the need to go to disk and page tables into and out of memory as the Store services client requests. SP2 allocates enough memory for each storage group to hold 80,000 tables open, which requires a sizable amount of virtual memory. SP3 reduces the request to 27,000 open tables per storage group. The reduction in the request for memory does not seem to affect the Store's performance and increases the size of the virtual memory pool available to other processes. In addition, lowering the size of MaxOpenTables leads to fewer small allocations by JET.

Experience to date demonstrates that servers running SP3 encounter less memory problems on high-end clusters. Thus, if you want to run a cluster or any high-end Exchange server, make sure that you carefully track the latest release of the software in order to take advantage of the constant tuning of the Store and other components that Microsoft does in response to customer experience.

The problems with virtual memory management forced Microsoft to express views on how active clusters should be. Essentially, Microsoft's advice is to keep a passive node available whenever possible, meaning that a two-node cluster is going to run in active-passive mode and a four-node cluster will be active on three nodes and be passive on the fourth. Of course, this approach is most valid if the cluster supports heavy load generated by clients, connectors, or other processing. Clusters that support a small number of clients and perhaps run only a single storage group with a few databases on each active node usually operate successfully in a fully active manner, because virtual memory fragmentation is less likely to occur.

By definition, because a "fresh" node is always available in an active-passive configuration, clusters can support higher numbers of users per active node, perhaps up to 5,000 mailboxes per node. The exact figure depends on the system configuration, the load generated by the users, the type of clients used, and careful monitoring of virtual memory on the active nodes as they come under load. There is no simple and quick answer to the "how many users will a system support" question here, and you will need to work through a sizing exercise to determine the optimum production configuration. See the Microsoft white paper on Exchange clustering posted on its Web site for more details about how to monitor clustered systems, especially regarding the use of virtual memory.

8.7.7 Monitoring virtual memory use

Exchange incorporates a set of performance monitor counters that you can use to check virtual memory use on a cluster. Table 8.2 lists the essential counters to monitor.

Figure 8.9 shows the performance monitor in use on a cluster. In this case, there is plenty of virtual memory available, so no problems are expected. If available virtual memory begins to decline as the load on a cluster grows, Exchange logs a warning event 9582[6] when less than 32 MB of available memory is present and then flags the same event again, this time with an error status, when no contiguous blocks of virtual memory larger than 16 MB exist inside STORE.EXE. After the Store reaches the

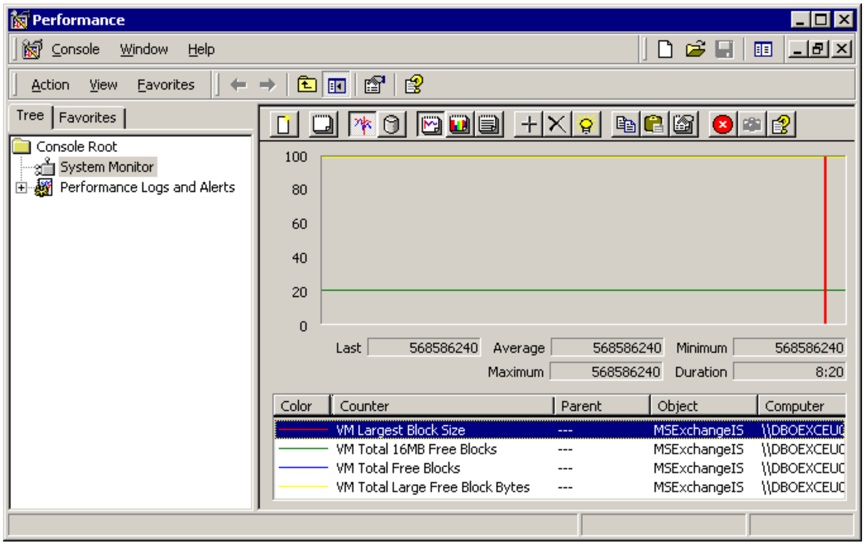
[6] Article 314736 describes how incorrect use of the /3GB switch in BOOT.INI on Exchange 2000 servers can also generate event 9582.

Table 8.2 *Performance Counters to Monitor Virtual Memory*

Performance Object	Performance Counter	Description
MSExchangeIS	VM largest block size	Size in bytes of the largest free virtual memory block
MSExchangeIS	VM total free blocks	Total number of free virtual memory blocks
MSExchangeIS	VM total 16 MB free blocks	Total number of free virtual memory blocks larger than or equal to 16 MB
MSExchangeIS	VM total large free block bytes	Total number of bytes in free virtual memory blocks larger than or equal to 16 MB

threshold, the cluster can become unstable and stop responding to client requests, and you will have to reboot. Microsoft Knowledge Base article 317411 explains some of the steps that you can take to capture system information to assist troubleshooting if virtual memory problems occur.

You may also see event 9582 immediately after a failover to a passive node, if the passive node has ever hosted the same virtual server that the cluster now wishes to transition. Each node maintains a stub STORE.EXE process, and the memory structures within the Store process may already be



Last 568586240 Average 568586240 Minimum 568586240
Maximum 568586240 Duration 8:20

Color	Counter	Parent	Object	Computer
🟥	VM Largest Block Size	---	MSExchangeIS	\\DBOEXCEU(
🟩	VM Total 16MB Free Blocks	---	MSExchangeIS	\\DBOEXCEU(
🟦	VM Total Free Blocks	---	MSExchangeIS	\\DBOEXCEU(
🟨	VM Total Large Free Block Bytes	---	MSExchangeIS	\\DBOEXCEU(

Figure 8.9 *Monitoring virtual memory.*

fragmented before a transition occurs, leading to the error. You can attempt to transition the virtual server to another node in the cluster and then restart the server that has the fragmented memory, or, if a passive node is not available, you will have to restart the active node. The rewrite of the virtual memory management code included in Exchange 2000 SP3 generates far fewer problems of this nature, and you are unlikely to see event 9582 triggered under anything but extreme load.

Microsoft made many changes to virtual memory management in Exchange 2003, and, generally speaking, the situation is much better and you should not see 9582 events logged as frequently as on an Exchange 2000 server. In addition, Microsoft incorporated a new safety valve into the Store process that kicks in if the Store signals the warning 9582 event. When this happens, the Store requests a one-time reduction (or back-off) of the ESE buffer to free up an additional 64-MB block of virtual memory. The net effect is that the Store can use this memory to handle the demand that caused the amount of free virtual memory to drop to critical limits. However, because the Store releases the virtual memory from the ESE buffer, server performance is affected and you cannot ignore the event. Instead, you should schedule a server reboot as soon as convenient. The advantage of the one-time reduction is that you have the opportunity to schedule the server reboot in a graceful manner, but it is not an excuse to keep the server up and running, because the 9582 error event will eventually occur again and you have to conduct an immediate reboot.

Note that some third-party products—particularly virus checkers—can affect how the Store uses virtual memory. If you run into problems, check that you have the latest version of any third-party product and monitor the situation with the product enabled and then disabled to see if it makes a difference.

Even though Exchange 2003 has improved virtual memory management, this is still not an area that an administrator can ignore, especially on heavily used servers. Once a server supports more than 1,000 concurrent mailbox connects (a rule of thumb, because server configurations vary dramatically), you should monitor virtual memory use to determine whether fragmentation is likely to be an issue for the server.

8.7.8 RPC client requests

The RPC Requests performance counter for the Store (MSExchangeIS) tracks the number of outstanding client requests that the Store is handling. On very large and heavily loaded clusters, the workload generated by clients may exceed the capacity of the Store and requests begin to queue. Normally, if the server is able to respond to all the client workload, the number of outstanding requests should be zero or very low. If the value of the RPC Requests counter exceeds 65, you may encounter a condition where Exchange may lose connectivity to the Global Catalog Server, resulting in clients experiencing a "server stall." Outlook 2003 clients that operate in cached Exchange mode experience fewer interruptions during cluster transitions or when servers have other problems, so you may want to deploy Outlook 2003 clients alongside Exchange 2003 clusters to isolate users as much as possible from server outages.

8.7.9 Upgrading a cluster with a service pack

Upgrading clusters always seems to be a stressful activity and the application of service packs to clusters is probably the worst culprit, possibly because of a now normal dependency on hot fixes. Microsoft has published a reasonable guide regarding how to apply service packs (see Microsoft Knowledge Base article 328839) that should be your first port of call for information. Exchange 2003 SP1 is a good example of how to go about applying a service pack to an Exchange cluster. The steps are:

1. Investigate any hot fixes for Windows that you need to apply to the cluster and install the fixes before beginning the upgrade.
2. Make a backup.
3. Install the service pack on the passive node of the cluster and reboot (if required).
4. Once the upgraded passive node is online, use the Cluster Administrator to move an Exchange virtual server over from an active node to the node that is now running the service pack.
5. Upgrade the inactive node using the "upgrade Exchange Virtual Server" option in the Cluster Administrator.
6. Continue to move virtual servers around to allow upgrading of each node in the cluster until they are all done.

After all nodes are complete, take another backup (just in case) and check that failover works correctly.

8.7.10 Stretched clusters and Exchange

Given the importance of email to many corporations and the need to ensure resilience against disaster, it comes as no surprise that there is an interest in using stretched clusters with Exchange. Stretched, or geographically dispersed, clusters use virtual LANs to connect SANs over long distances (usually between 10 KM and 20 KM, but sometimes over longer distances). For the cluster to function correctly, the VLAN must support connectivity latency of 500 ms or less. Exchange generates quite a specific I/O pattern, so if you want to use a stretched cluster, it is important that you deploy storage subsystems that support Exchange's requirements:

- The hardware must be listed in the multicluster section of Microsoft's Windows Server Catalog (search microsoft.com for "Windows Catalog" and navigate to "multicluster").
- The replication mechanism in the disk storage system must be synchronous (at the time of writing, Microsoft had not completed testing of systems that use asynchronous writes).
- The disk storage system must honor the ordering of writes.

In most cases, if Exchange suffers problems on the cluster, Microsoft will look at the storage system first, so it is good to have a high-quality relationship with your storage vendor and have some members of your team skilled in the technology.

Stretched clusters deliver resilience, but only in terms of geographical isolation. By themselves, they add nothing to the regular reliability functions delivered by running Exchange on any cluster. Indeed, stretched clusters come with a downside, because they can support fewer users than their regular counterparts due to the synchronous nature of the I/O. The exact reduction depends on workload and configuration, but you can expect to cut the number of supported users by half. This aspect of stretched clusters may improve over time as the technology gets smarter, but, for now, it is an issue to consider.

Hardware-based stretched clusters are the only implementation method that Microsoft supports. Software-based stretched cluster implementations do exist, but some evidence indicates that these solutions are less than perfect within an Exchange environment, so they are not recommended.

8.7.11 Deciding for or against a cluster

Assuming that you have the knowledge to properly size, configure, and manage an Exchange cluster, Table 8.3 lists some of the other factors that companies usually take into account before they decide to put clusters into production.

Table 8.3 *Pros and Cons of Exchange Clusters*

Pros	Cons
Clusters allow you to update software (including service packs) on a rolling basis, one node at a time. This ensures that you can provide a more continuous service to clients, because you do not have to take the cluster totally offline to update software.	If you plan software upgrades properly, schedule them for low-demand times (e.g., Sunday morning), and communicate the necessary downtime to users well in advance, so you can take down a server to apply an upgrade without greatly affecting users. Routine maintenance is necessary for all systems, so planning a software upgrade at the same time is not a big problem. Microsoft hot fixes are often untested on clusters when released to customers, so it is a mistake to assume that you can apply every patch to a cluster. In addition, third-party product upgrades do not always support rolling upgrades, and you can only apply the upgrade to the active node.
Clusters provide greater system uptime by transitioning work to active members of the cluster when problems occur.	Clusters are expensive and may not justify the additional expense over a well-configured standard server in terms of additional uptime.
Active-active clusters are a great way to spread load across all the servers in a cluster.	Memory management problems limit the number of concurrent clients that an active-active cluster supports, so many clusters run in active-passive mode to ensure that transitions can occur.
Clusters provide protection against failures in components such as motherboards, CPUs, and memory.	Clusters provide no protection against storage failures, so they have an Achilles heel.
	Because clusters are not widely used, a smaller choice of add-on software products is available for both Windows and Exchange.
	Clusters require greater experience, knowledge, and attention to detail from administrators than standard servers.
	Clusters do not support all Exchange components and therefore are only useful as mailbox servers.
	Failures in the shared disk subsystem remain the Achilles heel of clusters: A transition from one node to another that depends on a failed disk will not work.

When many companies reviewed their options for Exchange server configurations, they decided not to use clusters and opted for regular servers instead. Common reasons cited by administrators include:

- Not all locations in the organization require (or can fund) the degree of uptime that a cluster can provide. Deployment and subsequent support is easier if standard configurations are used everywhere and the total investment required to support Exchange is less.

- Administrators can be trained on a single platform without having to accommodate “what if” scenarios if clusters are used.
- The choice of third-party products is much wider if clusters are not used.
- The hardware and software used by the cluster are expensive.
- Experience of Exchange 5.5 clusters had not been positive.

Every company is different, and the reasons why one company declines to use clusters may not apply elsewhere. Compaq was the first large company to achieve a migration to Exchange 2000 and opted not to use clusters. As it happens, the Exchange organization at Compaq does include a couple of clusters, but they only support small user populations and support groups that have the time and interest to maintain the clusters. In addition, none of the clusters at Compaq uses active-active clustering. On the other hand, many companies operate two-node and four-node production-quality clusters successfully. In all cases, these companies have dedicated the required effort and expertise to deploy and manage the clusters.

8.7.12 Does Exchange 2003 make a difference to clusters?

The combination of Windows 2003 and Exchange 2003 introduces a new dimension to consider when you look at clusters. The major improvements are:

- The dependency on Windows 2000 Datacenter Edition is gone, so you can now deploy up to eight-node clusters without the additional expense that Windows 2000 Datacenter edition introduces. Now that the Enterprise Edition of Exchange 2003 supports up to eight nodes in a cluster, administrators have a lot more flexibility in design.
- Windows 2003 and Exchange 2003 both make changes that contribute to better control of memory fragmentation, which may increase the number of MAPI clients that a cluster supports. Windows and Exchange also make better use of large amounts of memory, because Microsoft has gained more experience of how to use memory above 1GB when it is available.
- You can use drive mount points to eliminate the Windows 2000/ Exchange 2000 restriction on the number of available drive letters,

which limits the number of available disk groups in a cluster. This is important when you deploy more than ten storage groups spread across multiple cluster nodes.

- Assuming that you use appropriate hardware and backup software, you can use the Volume ShadowCopy Services (VSS) API introduced in Windows 2003 to take hot snapshot backups. This is critical, because clusters cannot attain their full potential if administrators limit the size of the databases they are willing to deploy, in turn limiting the number of mailboxes that a cluster can host.
- The Recovery Storage Group feature lets administrators recover from individual database failures more quickly and without having to deploy dedicated recovery servers.
- The Store is faster at moving storage groups from failed servers to active nodes.

In addition, if you deploy Outlook 2003 clients in cached Exchange mode, there is potential to support more concurrent MAPI clients per cluster node because the clients generate less RPC operations against the server, since much of the work that previous generations of MAPI clients did using server-based data is now executed against client-side data. However, we are still in the early days of exploring this potential and hard results are not yet available.

To Microsoft's credit, it is using clusters to test the technology and help consolidate servers. For its Exchange 2003 deployment, Microsoft has a "datacenter class" cluster built from seven nodes that support four Exchange virtual servers. Four large servers (HP Proliant DL580 G2 with quad 1.9-GHz Xeon III processors and 4 GB of RAM) take the bulk of the load by hosting the Exchange virtual servers, each supporting 4,000 mailboxes with a 200-MB default quota. A passive server is available to handle outages, and two other "auxiliary" servers are available to perform backups and handle other administrative tasks. Microsoft performs backups to disk and then moves the virtual server that owns the disks holding the backup data to the dedicated backup nodes, a technique necessary to handle the I/O load generated when they move the data to tape for archival. All the servers connect to an HP StorageWorks EVA5000 SAN, and the storage design makes heavy use of mount points to allocate disk areas for databases, transaction logs, SMTP work areas, and so on. Supporting 16,000 mailboxes on a large cluster demonstrates that you can deploy clusters to support large numbers of users. Of course, not all of the users are active at any time, and the administrators pay close attention to memory defragmentation in line with best practice, along with normal administrative tasks.

One thing is certain: It is a bad idea simply to install a cluster because you want to achieve highly reliable Exchange. An adequately configured and well-managed standalone server running the latest service pack is as likely to attain a "four nines" SLA as a cluster.

8.7.13 Clusters—in summary

Microsoft did its best to fix the problems with memory fragmentation, but there is no doubt that Exchange 2000 clusters have been a disappointment. As with Exchange 5.5 clustering, which initially promised a lot and ended up being an expensive solution for the value it delivered, the problems have convinced many who considered Exchange clusters to look at other alternatives, notably investing in standalone servers that share a Storage Area Network (SAN). In this environment, you devote major investment into building resilience through storage rather than clusters. If you have a problem with a server, you still end up with affected users, but the theory is that the vast majority of problems experienced with Exchange are disk related rather than software or other hardware components. Accordingly, if you take advantage of the latest SAN technology to provide the highest degree of storage reliability, you may have a better solution to the immediate need for robustness. Going with a SAN also offers some long-term advantages, since you can treat servers as discardable items, planning to swap them out for newer computers as they become available, while your databases stay intact and available in the SAN.

Fools rush in to deploy clusters where experienced administrators pause for thought. There is no doubt that Exchange clusters are more complex than standard servers are. Experience demonstrates that you must carefully manage clusters to generate the desired levels of uptime and resilience. Those who plunge in to deploy clusters without investing the necessary time to plan, design, and deploy generally encounter problems that they might avoid with standard servers after they have installed some expensive hardware. On the other hand, those who know what they are doing can manage clusters successfully and attain the desired results. At the end of the day, it all comes down to personal choice.

The early reports of successful deployments of Exchange 2003 clusters, including Microsoft's own, are encouraging and we can hope that the changes in Windows 2003, Exchange 2003, and Outlook 2003, as well as improvements in server and storage technology and third-party software products, all contribute to making Exchange clusters a viable option for more deployments. The challenge for Microsoft now is to continue driving complexity out of cluster software and administration so that it becomes as easy to install a cluster as it is to install a standard server. That day is not yet here.

I remain positive about clusters. Providing that you carefully plan cluster configurations and then deploy those configurations, along with system administrators who have the appropriate level of knowledge about the hardware, operating system, and application environment, clusters do a fine job; I am still content to have my own mailbox located on an Exchange cluster. The problem is that there have been too many hiccups along the road, and clusters have not achieved their original promise. Work is continuing to improve matters, but in the interim, anyone who is interested in clustering Exchange servers should consider all options before making a final decision.