

Realtime
publishers

The Shortcut Guide[™] To



**Exchange
Server 2007
Storage Systems**

sponsored by



Jim McBee

Introduction to Realtimepublishers

by Don Jones, Series Editor

For several years, now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We’ve made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book’s production expenses for the benefit of our readers.

Although we’ve always offered our publications to you for free, don’t think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you \$40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the “realtime” aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We’re an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I’m proud that we’ve produced so many quality books over the past years.

I want to extend an invitation to visit us at <http://nexus.realtimepublishers.com>, especially if you’ve received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you’re sure to find something that’s of interest to you—and it won’t cost you a thing. We hope you’ll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Introduction to Realtimerepublishers..... i

Chapter 1: Choosing an Exchange Server Storage Platform1

Exchange Databases, Transaction Logs, and Other Files2

 Exchange Databases and Transaction Logs2

 Full-Text Index Files.....7

 Exchange Replication9

 Message Tracking and Protocol Log Files.....12

Internal vs. External Storage.....12

 Using Local Storage.....12

 Using External Storage13

Networked Storage Technologies14

 Basics of SANs and NAS14

 NAS.....15

 SANs.....16

 Grid or Clustered Storage19

 Tips for Choosing a SAN Solution20

Summary21

Copyright Statement

© 2007 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 1: Choosing an Exchange Server Storage Platform

Designing a server infrastructure for an Exchange organization has a lot of small obstacles that sometimes stand between you and an optimal design. Picking the right server hardware, security components, antivirus software, and anti-spam software can contribute to the success or the perceived failure of a particular design. Choosing the right storage platform seems like a no-brainer, but this task is frequently one of the choices that IT professionals and Exchange designers get wrong.

Choosing storage seems deceptively simple because it merely seems like you need to get enough disk space to ensure that you can support the number of mailboxes and the maximum mailbox sizes that you are planning to support. Scalability, input and output (I/O) capacity, deleted items, deleted mailboxes, and recoverability are frequently overlooked as factors that contribute to storage system requirements. Often a choice to use direct-attached storage (DAS) disks leaves you without room to grow, but choosing storage area network (SAN) technology can dramatically increase storage costs, limit your options with respect to configurability, and be unacceptably complex to manage for many Exchange shops.

Whether running Exchange Server 2000 or 2003, or Exchange 2007, picking storage for an Exchange system is a balance between performance, disk space requirements, scalability, and cost. Although the specifics of picking a system between Exchange 2000, 2003, or 2007 may be slightly different, the principles and most of the factors you must consider remain the same. Although this guide will mostly focus on Exchange 2007, it will note places where particular factors may affect Exchange 2000 or 2003.

This guide will cover the basics of Exchange Server storage including frequent mistakes that are made when sizing a storage system for Exchange data and how to properly size iSCSI-based SANs to provide you acceptable growth and performance. This chapter will cover the basics of Exchange Server storage including:

- The types of data storage that Exchange Server requires
- The differences between DAS and networked storage
- Storage technologies that Exchange Server can use
- The basics of Exchange input and output (I/O)

Exchange Databases, Transaction Logs, and Other Files

Let's start by looking at the different types of data that Exchange Server stores and how this may affect your scalability. A novice to Exchange might say "Exchange is just a big mail database, right?" For the most part, Exchange Server is just a big database engine for mail storage, but it is how that data gets into the mailbox that makes things a bit more complicated. In addition to databases, Exchange Server records all operations to transaction logs and has full-text indexes of the databases, local continuous replication databases, and logs that are used for auditing and troubleshooting.

Exchange Databases and Transaction Logs

The Exchange database engine is a specialized database engine that over the years has been tuned for use with hierarchical data such as mailboxes, folders, message items, message bodies, attachments, and so on. The database engine is called the Extensible Storage Engine (ESE). The ESE98 database engine is used with Exchange 2000/2003/2007 and the ESE97 database engine is used with Active Directory (AD). It is neither a fully relational database, nor a SQL database, nor a simple Access database, but a combination of these.



A popular misconception about Exchange Server is that it uses Microsoft SQL Server as a database engine. Although this possibility has been discussed by Microsoft many times as a future for Exchange Server, even Exchange Server 2007 continues to use the ESE database.

The ESE database engine provides Exchange Server with a robust database platform that can perform extremely well even under large loads, survive server failures, and allows for disaster recovery. The ESE database engine meets the ACID test for databases that was developed by IBM in the 1970s as a criterion for building a robust database and is described in ISO/IEC 10026-1:1992 Section 4. The ACID test means:

- All operations against a database are atomic. Either all changes are made to a database or none of them are made. If one part of a transaction fails, the entire transaction fails.
- The database is always taken from one consistent state to another. A transaction cannot move the database to an inconsistent state or a state that violates the rules of the database. When a transaction completes, the database must always be in a consistent state.
- Changes are always isolated. When multiple transactions are taking place against a single database, one set of transactions does not affect the other.
- All database changes are durable. Backups of databases and transaction logs ensure recovery from any type of failure that might occur.

The ACID test is frequently discussed in IT circles, even with respect to Exchange Server, but what does this really mean? The following points highlight the way I like to think of the ACID test with respect to Exchange:

- All changes are first recorded in to a transaction log before the database is updated.
- Any change to the database is broken into individual operations such as “save the To line,” “save the From line,” “save the message body,” and so on. The database is not changed until all the operations necessary to complete the transaction have been committed to the transaction log.
- If Exchange Server crashes after a transaction has been successfully logged but before it is recorded to the database, the transaction can be properly applied to the database when the server comes up.
- If an old backup is restored to an Exchange Server but all the transaction logs that have been recorded since the last backup are on the server’s disk, the database can be “played forward” until the last completed transaction found in the logs is recorded into the database.

I am surprised that frequently even experienced Exchange administrators don’t understand much about how the database works, but this type of knowledge can help you in storage sizing, disk configuration, backup, and most importantly data recovery. To better illustrate, consider Figure 1.1, which conceptualizes and simplifies the process of saving data, such as a mail message, to the database.

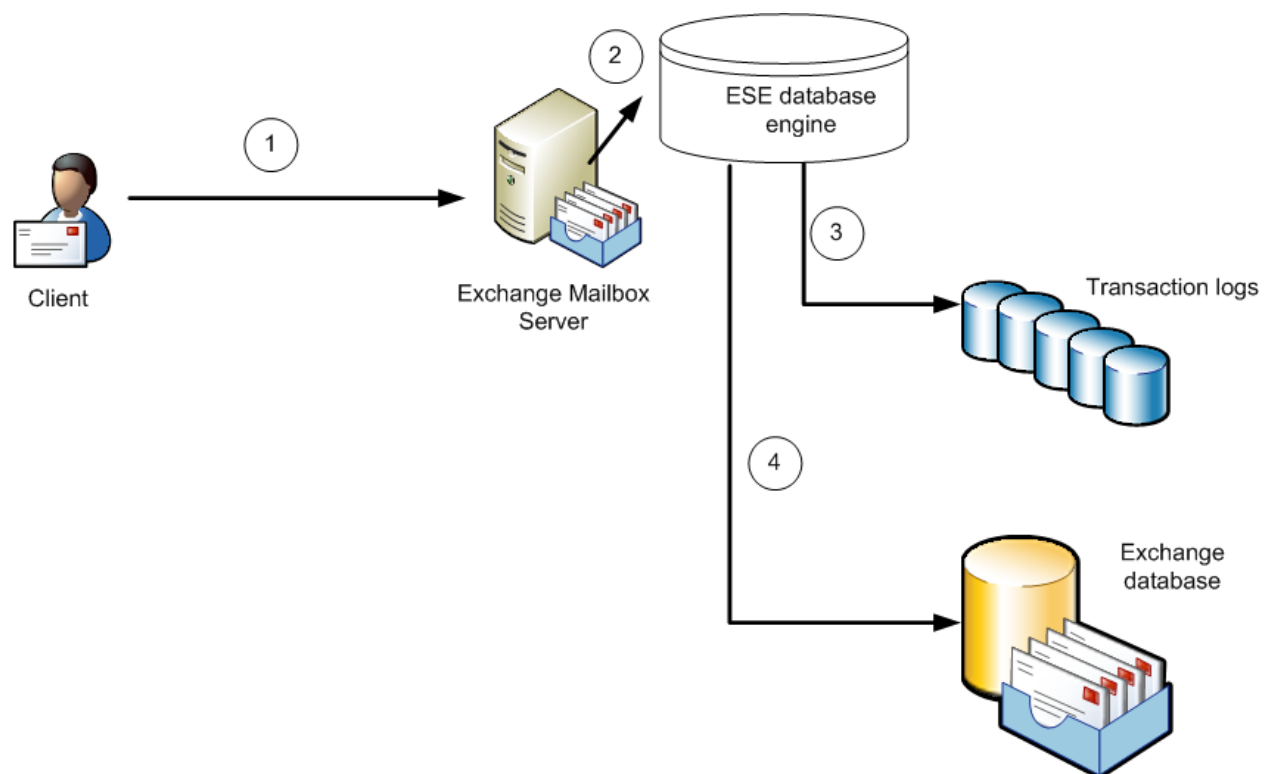


Figure 1.1: Simplified database operations.

Let's take as an example a new message arriving on the Exchange Mailbox server. In the first step, a client (Outlook, ActiveSync, Outlook Web Access, or the Exchange Hub Transport server role) sends a message to the Exchange Server. The Microsoft Exchange Information Store service (store.exe) is running the database engine (ESE). The message is moved into the database engine cache in memory in step 2.

The message is broken up into the individual operations necessary to save it as a transaction and, in step 3, is immediately written to the storage group's transaction logs. The message is completely written to the transaction logs usually within a second or two. Once the database engine has completely written the entire message transaction to the transaction logs, the same transactions will be applied to the actual database file (the EDB file for Exchange 2007 or the EDB and STM files for Exchange 2000/2003.) Once all the transactions are in the database, the message is present.



Transaction logs in Exchange 2000/2003 are 5MB in size while Exchange 2007 transaction logs are 1MB in size. The 1MB size allows for quicker replication of transactions if Exchange 2007 replication technologies are used.

The actual message transaction information may remain in the ESE database cache for a few additional seconds or even minutes before it is committed to the database. The message has been written to the transaction logs, so Exchange is not in a big hurry to write it to the database and will attempt to pick an optimal time when there is less server activity to commit the transaction to the database. In step 4, the actual writes to the database take place from memory (the ESE database engine cache), not by reading the transaction log files.

The actual operations that take place for other types of operations are similar for deletions, moves, folder creations, modifications, and so on. Figure 1.1 make this very simple; in reality, the database engine is far more complex—but the principles are important. There are a couple of important points to consider:

- All operations against an Exchange Server database are recorded to the transaction logs first.
- Transaction logs are not “read” during normal operation unless you have enabled local continuous replication or cluster continuous replication. Even then, Exchange doesn't read information from the logs in order to apply it to the database; logs are normally only read during recovery.
- Database caching is important for Exchange because the database engine can write to the database more efficiently if it has more cache. Recently, written messages can also be read from cache rather than requiring the Exchange database engine to re-read a message from cache. Exchange Server 2000/2003 are both limited in the amount of data that the ESE database can cache; the Exchange Server 2007 64-bit architecture dramatically improves the amount of memory that can be used.
- To improve the possibility of recovering from a failure, a storage group's transaction logs should be written to a separate physical disk or logical unit (LUN) from the one that stores the group's database files.
- Transaction log writes must occur as quickly as possible; performance on transaction log disks or LUNs is write intensive and sequential in nature.
- Database operations are both read- and write-oriented and are random in nature.

Another important point about how the database engine works is checksums and page size. The data in the Exchange databases and transaction logs is broken into pages. In Exchange 2007, these pages are 8KB while the page size for the Exchange 2000/2003 EDB file is 4KB and the STM file is 8KB. The STM file is not used by Exchange Server 2007; only the EDB file.

Each page contains mostly data, but at the beginning of the page is page header information that is critical to the functionality of the database. Figure 1.2 shows this idea conceptually. The page header contains a page identifier, a pointer to the next page, a pointer to the previous page, and a checksum.

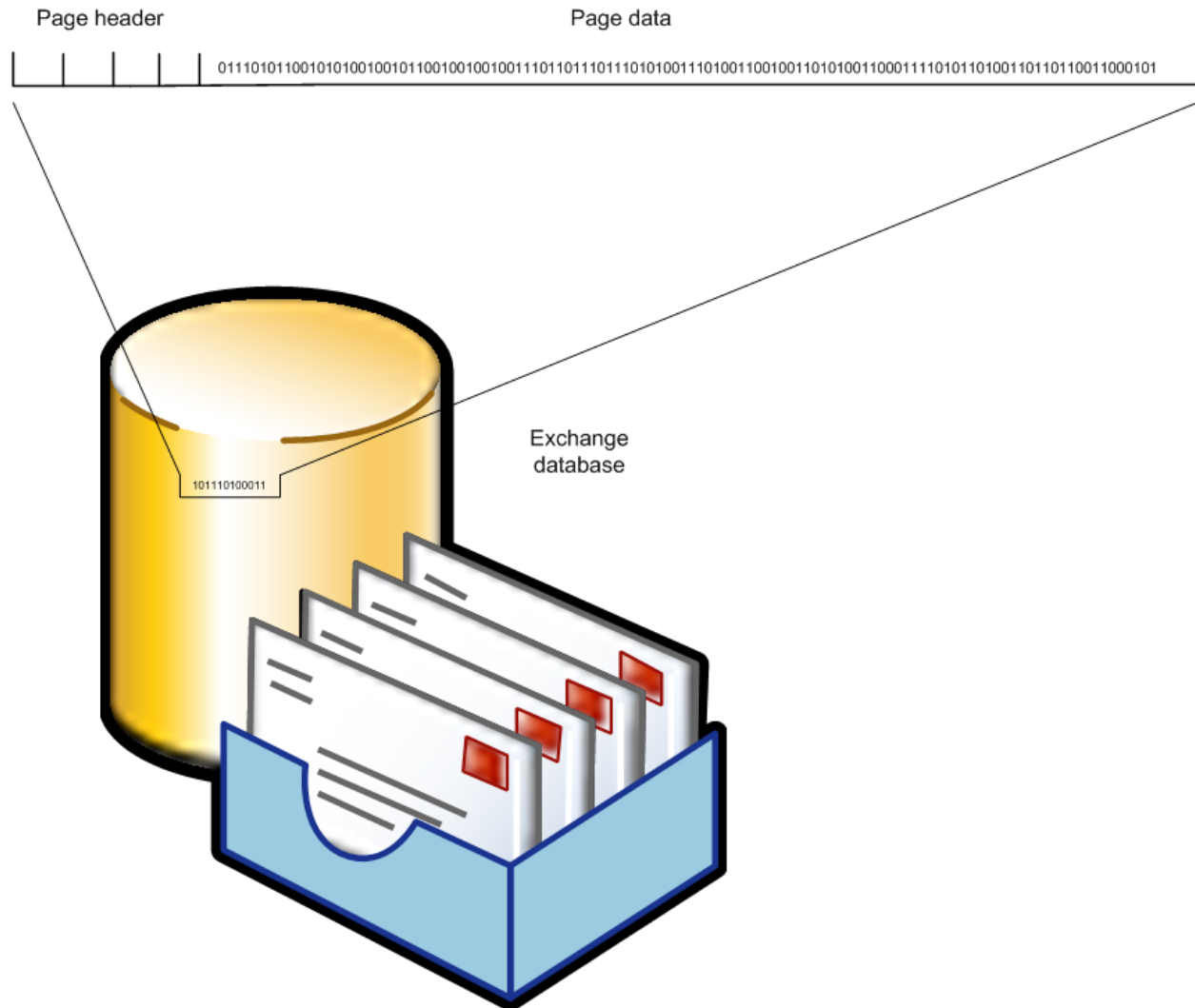


Figure 1.2: Examining a database page.

Each of the values in the page header must be correct; otherwise the database will be structurally unsound. The page must properly point to the next page in the database as well as to the previous page, but the most important value is the page's checksum. When the ESE database engine prepares to write the data to the disk (either to the database or a transaction log), the data in the page has a mathematical calculation performed on it called a checksum. The checksum value is written to the page header.

 A backup using the Exchange backup APIs verifies that each page of the data is good.

Each time the page is read by the database engine during normal operations or during an online backup, the checksum is performed on the data and compared with the checksum in the page header. If the checksum that is calculated during the read operation is different than the checksum that is in the header, this difference indicates the page is corrupt.

When planning for physical disks or LUN allocation for Exchange Server systems, the best possible design is to ensure that the transaction logs are on separate physical disks from the database files. This is partially for performance reasons but also for recoverability. If a problem occurs on the database volume and that volume is physically isolated from the transaction log volume, the transaction logs would not be affected thus guaranteeing very good recoverability. Separating transaction log I/O from database I/O also allows you to separate the types of I/O that are occurring on each disk. Database reads and writes are random while transaction log writes are sequential. Figure 1.3 shows an example of a disk layout for an Exchange 2007 Mailbox server. The operating system (OS) and the Exchange binaries are stored on the C drive which is a RAID 1 volume.

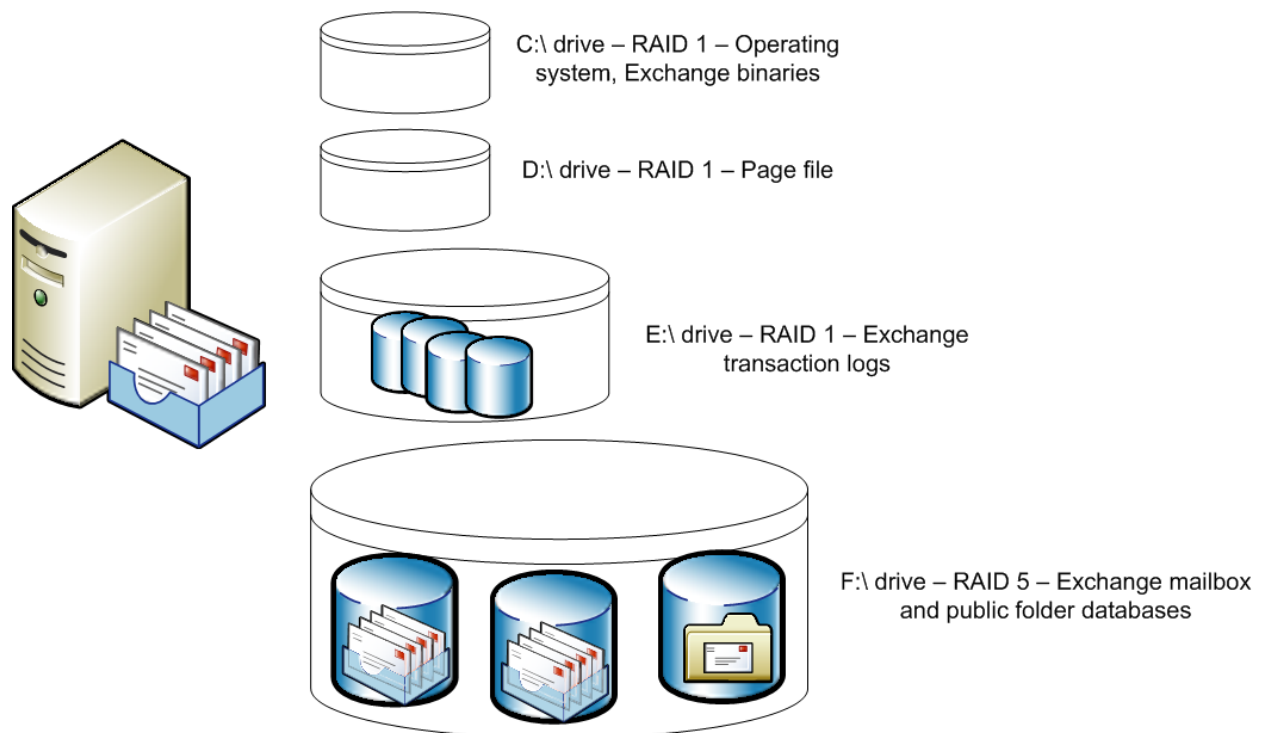




Figure 1.3: Example disk configuration.

The D drive is also a RAID 1 volume and holds the OS page file. For servers supporting less than 1000 mailboxes, putting the page file on separate physical partition is a “nice to have” feature but not entirely necessary. If this is not practical, you can always put the page file on the OS disk. If the Exchange Server is configured with sufficient memory (which it should be), it should not need to page frequently.

The E drive is a RAID 1 array and is for the Exchange transaction log files. Ideally, this volume should be on a separate physical RAID 1 array because RAID 1 will provide better write performance than RAID 5. The F drive is either a RAID 5 or RAID 1+0 array that supports the Exchange mailbox and public folder databases for this server.

 These RAID recommendations are somewhat storage vendor dependent, depending on what additional availability schemes are available. More generally, you want logs and database volumes to be protected, and log volumes should be optimized for sequential write performance.

 When planning database storage for an Exchange database, RAID 1+0 (striped arrays that are then mirrored to striped arrays) always provide better read-and-write performance than RAID 5 arrays alone provide.

I usually advocate splitting the transaction logs and database files onto separate volumes. For small organizations (less than 100 mailboxes), this setup may not be financially feasible, though. However, only 100 mailboxes will not put an extreme load on the storage subsystem, either. The example in Figure 1.3 is conceptual; the E and F drives could either be local physical disks or LUNs provided by a SAN.

Usually, SAN storage systems don't give you the luxury of separating one LUN onto one set of physical disks and another LUN on a different set of physical disks. The most important factor to consider when using SAN storage is to ensure that regardless of how the disks are configured within the SAN, it is capable of supporting the I/O load that the Exchange Server will place on it.

 This concept will be discussed in more detail later in this chapter and in later chapters.

Full-Text Index Files

By default, Exchange Server 2007 mailbox databases are fully indexed; the full-text index provides Outlook and Outlook Web Access users the ability to quickly search the content (message bodies and attachments) in their mailbox. The content index is found a folder below the mailbox database folder. Figure 1.4 shows the full-text index folder for an Exchange 2007 mailbox database.

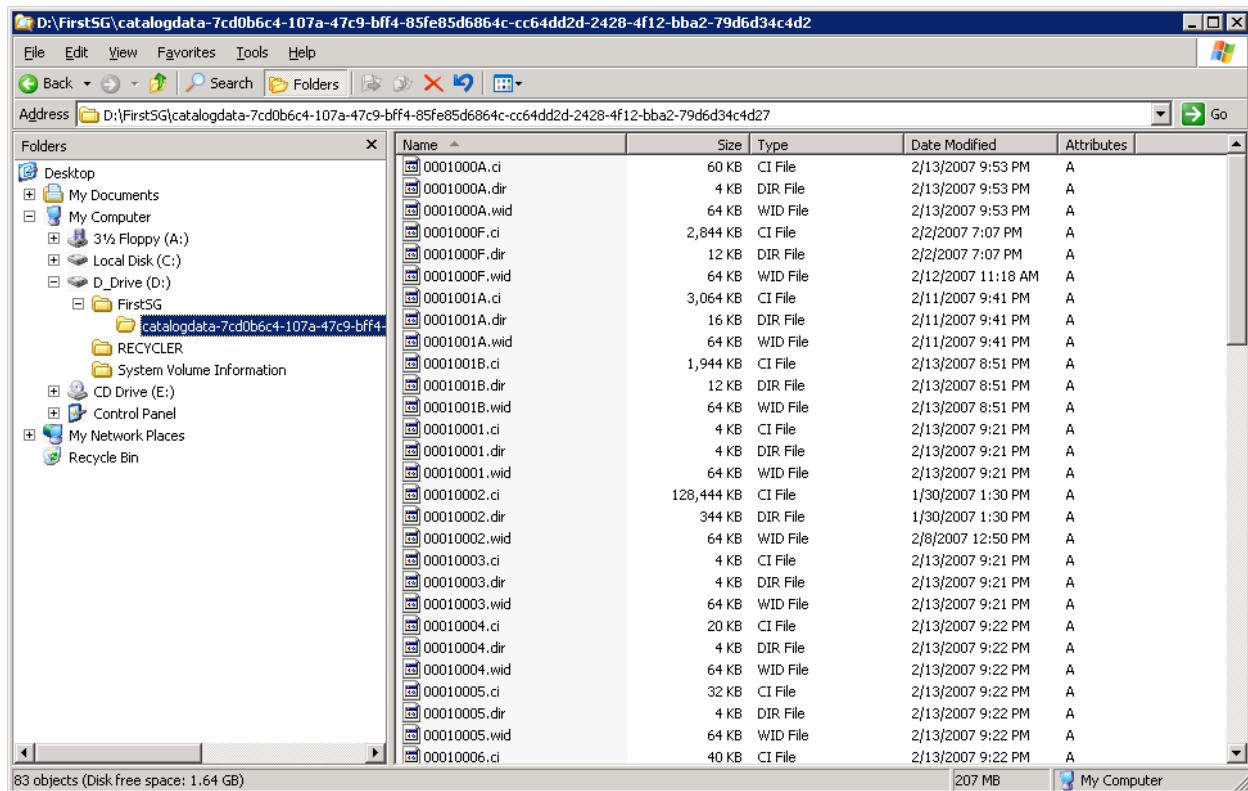


Figure 1.4: Microsoft Search service index files.

The full-text index files are found in a folder that begins with

catalogdata-

and includes the GUID of the database for which it was created. You can verify that with the Exchange Management Shell cmdlet

```
Get-MailboxDatabase
```

Here is an example:

```
Get-MailboxDatabase "HNLEX03\Mailbox Database" | fl
name,*index*,GUID
```

```
Name      : Mailbox Database
```

```
IndexEnabled : True
```

```
Guid      : 7cd0b6c4-107a-47c9-bff4-85fe85d6864c
```

You can disable the full-text indexing feature for a particular database using the

```
Set-MailboxDatabase
```

cmdlet. Here is an example of disabling full-text indexing for a database called Mailbox Database on server HNLEX03:

```
Set-MailboxDatabase "HNLEX03\Mailbox Database" -IndexEnabled
$False
```

I recommend you keep the indexing feature enabled for Exchange 2007 mailbox databases. The overhead is only about 5 percent of the total database size. In Figure 1.4, the total size of the full-text index files is 207MB for a mailbox database that is 4.4GB in size.

If you are using Exchange 2000 or 2003, I recommend enabling full-text indexing only if there is a specific need. If full-text indexing is necessary for only a subset of the user community, create an additional mailbox store that hosts only those users and enable full-text indexing for that specific mailbox store. Exchange 2000/2003 full-text indexing is both CPU and disk intensive. An Exchange 2000/2003 full-text index catalog can consume as much as 40 percent of the total size of the database. By default, the full-text index files are located on the same disk volume as the Exchange binaries (usually C:\Program Files\Exchsrvr); this volume may not be sized appropriately to store the full-text index files or allow for the performance overhead that the indexing process requires.

Exchange Replication

A new feature has been introduced for Exchange Server 2007 that allows for near-real-time replication of Exchange databases. Replication of databases can either be used on a standalone mailbox server or in a 2-node active-passive cluster. Replication on a standalone mailbox server is called local continuous replication and is used for database recovery purposes. Figure 1.5 illustrates this concept. The Exchange mailbox server requires additional disk drives that hold replicated transaction logs and backup database files.

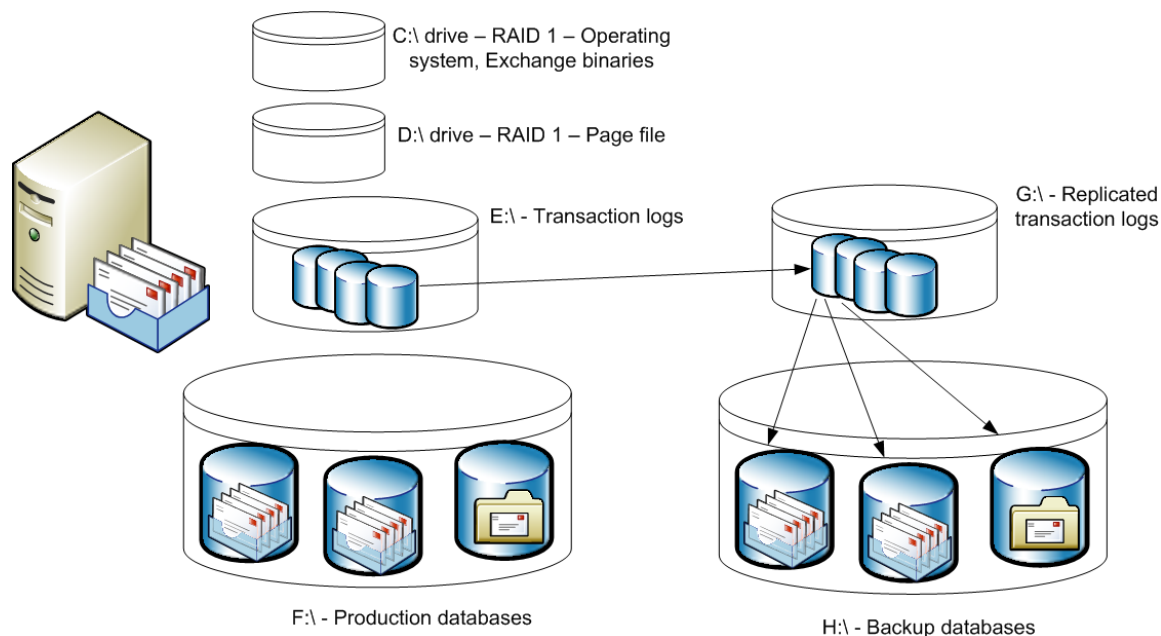


Figure 1.5: Local continuous replication.

When local continuous replication (LCR) is configured for a production database, a copy of that database is copied to the LCR database location. This process is called seeding. As transaction log files are filled and committed to the disk (the E drive), Exchange copies the log files to the LCR log file location (in this case, the G drive). Exchange then reads the transactions in the log file that was copied and commits them to the backup copies of the databases on the H drive. At any given point, the backup copies of the database are only a few transactions behind the production databases.

The advantage of this is that you have local copies of databases that are always ready to be put in to production in the event of a production database becoming corrupted. If you implement snapshot database backup technology, snapshots can be taken of the backup databases rather than the production databases; doing so will reduce the overhead of snapshot backups. Exchange 2007 replication LCR and cluster continuous replication (CCR) do not require any sort of special replication technology such as disk-level or page-level replication. This will simplify replication for organizations that require it because either local disk or SAN attached can be used and will be supported by Microsoft. I strongly advise you to use separate physical disks or LUNs for the LCR transaction logs and databases.

Keep in mind that adding LCR (or CCR) replication to your environment will create an additional I/O load on your disk subsystem. The transaction logs must be read after they have been committed so that they can be copied to the LCR transaction log location. Once written to the LCR transaction log location, they must be read and replayed to the relevant LCR copies of the databases. These factors must be considered when designing disk capacity for an LCR solution.

☞ One requirement for using the Exchange 2007 continuous replication solutions is that each storage group must contain only a single database. You must take this into consideration when planning storage groups, volumes or LUNs, and disk space requirements.

The other replication technology is CCR. This technology works in a 2-node, active-passive Windows cluster. Figure 1.6 shows an example of an active-passive Exchange 2007 cluster.

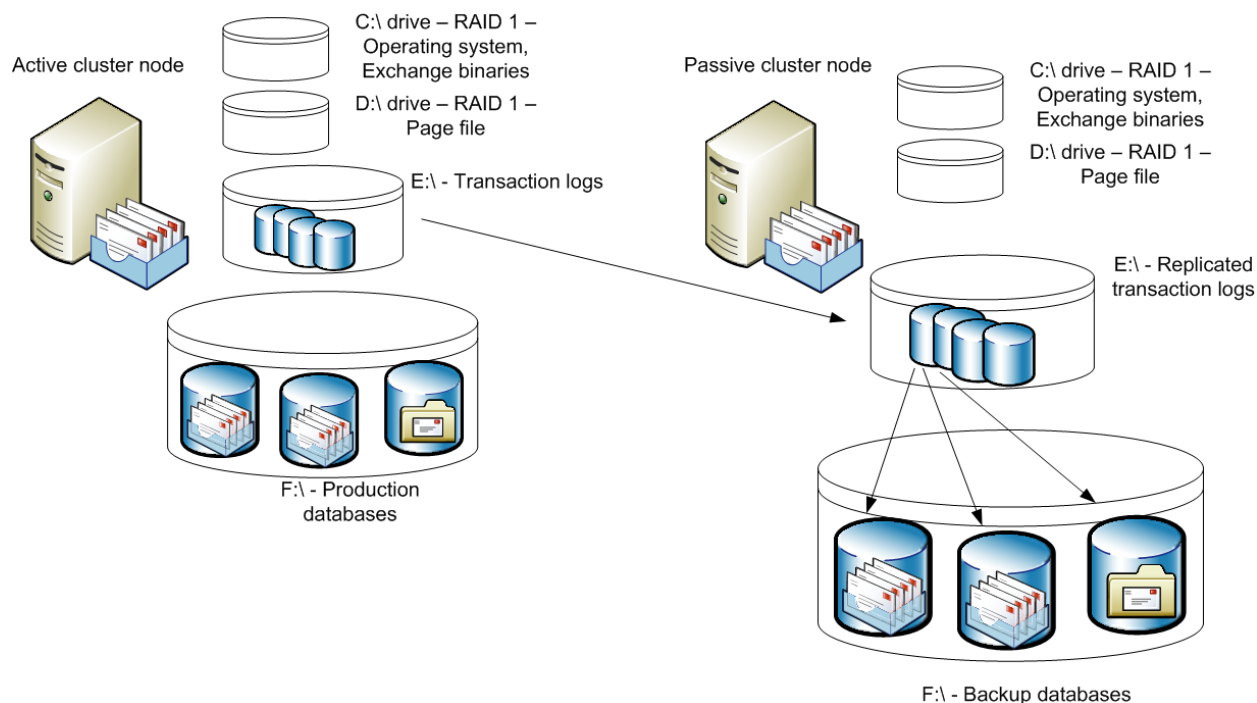




Figure 1.6: Cluster continuous replication.

CCR works very similar to LCR. As transaction logs are filled and committed on the active node of the cluster, the passive node copies them to the E drive of the passive node. Once they are written to the passive node, they are read and the transactions are replayed into the CCR copies of the databases.

 Exchange 2007 CCR does not require shared storage. This makes implementing a cluster quite a bit simpler than in previous versions.

In the event of a failure of the active node of the cluster, the passive node will perform a failover and assume responsibility for the clustered mailbox servers. This is different from Exchange 2000/2003 clusters (now called a Single Copy Cluster) where there is only one copy of the databases and the transaction log files; this single copy usually resides on a SAN. With CCR, the clustered databases can either be on a local disk or on SAN storage; if the data is stored on a SAN, it does not need to be shared with the other node of the cluster.

 LCR/CCR in a SAN environment will not protect against hardware failures unless the two copies are on different physical arrays (or different clusters). You might still want to apply block-level replication for array-level protection. Also, array-based replication is almost always more efficient and has less impact from an IOPS perspective. However, LCR/CCR gives greater logical protection than array-based protection. Some consider LCR/CCR as more of a replacement for tape backups than as a replacement for array replication.

Message Tracking and Protocol Log Files

Exchange Server also has a variety of tracking and protocol logs that can be enabled. For servers that function as a Hub Transport server, these logs include the message tracking logs, routing table logs, send protocol logs, receive protocol logs, and intra-organization protocol logs. For servers that have the Client Access server role configured, Internet Information Server (IIS) will (by default) record HTTP protocol logs. The protocol and message tracking log files are found on the same volume as the Exchange binaries (for example, C:\Program Files\Microsoft\Exchange Server\TransportRoles\LogFiles) and the IIS HTTP protocol logs are found in the C:\Windows\System32\LogFiles folder. I recommend keeping 10 to 15 days worth of these logs at a minimum, as they always come in handy when troubleshooting.

All these log files can be moved, but keep in mind that a busy Exchange 2007 Client Access and Hub Transport server can easily generate a few hundred megabytes of these logs each day. Although writing to the log files is not as disk intensive as writing to databases or transaction logs files, ideally, you should not put these log files on a disk or LUN with the databases or transaction logs.

Internal vs. External Storage

One of the biggest decisions you will have to make with respect to storage is whether you should use internal or external storage. In the past, the term “external storage” would have indicated disks that were in an external storage unit but directly attached to the server via a SCSI cable. This is commonly known as DAS even if the disk enclosure is separate from the actual server machine.

Now external storage more commonly describes disks that are managed separately from a single server on your network and are accessed via some type of storage network. These are most commonly called SANs or network attached storage (NAS). This section will look at the reasons DAS or SAN storage might be right for you.

Using Local Storage

Local storage is considered any disk storage that is on the internal IDE, SATA, or SCSI bus. Certainly using DAS for Exchange is one of the most common methods of storage for Exchange Server. DAS is simple to implement and requires very little additional experience beyond that required to deploy the server hardware. Let’s take a quick look at the advantages of using locally attached storage:

- Simple and quick to deploy
- Server and storage system are “self contained”
- Fault tolerance is handled either by the OS (not recommended) or a locally installed RAID controller—I don’t recommend using the Windows RAID 1 or RAID 5; recovery is complicated and performance is not as good as hardware RAID controller solutions

However, the simplicity has its cost in terms of growth, centralized management, and scalability. The following list highlights disadvantages of using locally attached storage:

- Most servers have a limit to the number of drives that can be supported internally, which can limit both the amount of space that the server has as well as the I/O capacity; even external disk enclosures will eventually reach the maximum capacity
- Depending on the server, reconfiguring the local storage can be difficult and thus makes it difficult to adapt to an organization's changing requirements
- All the storage capacity on a server must be allocated at the time it is installed
- Problems with the server hardware will also affect the storage subsystem
- Most server's RAID controllers and local storage systems offer fewer fault-tolerance and backup options than centralized storage

Using External Storage

Over the past 8 years or so, we have seen networked storage become commonplace for many applications in the data center. There are certainly some advantages to using networked storage even for small and medium-sized businesses. Some of these advantages include:

- Better scalability and performance than DAS
- Adding or reallocating storage to a specific server or application is much easier
- Storage can be centrally managed and protected, and allocated based on an organization's needs
- Networked storage systems usually have much higher availability and better fault-tolerance features than local storage systems provide
- Network storage systems often offer improved backup technologies such as snapshot and replication technologies that are incorporated into the storage system

With these additional features, though, network storage systems have a few downsides. Though the cost of networked storage has dropped dramatically over the past few years, there is still a pretty significant cost and the upfront cost is only part of the total storage costs. The second downside is additional complexity. Networked storage systems such as SANs require additional training and expertise to operate properly and take advantage of their added functionality. However, both of these disadvantages depend entirely on the technology and the vendor. Some networked storage solutions are less expensive than you might think and are often not much more complicated than learning how to configure your server vendor's RAID controller.

Networked Storage Technologies

Recently, networked storage technologies have emerged, matured, and become reasonably priced, even for small and medium-sized businesses. Originally, SANs were both expensive and complex and thus only available to very large customers. As the technology has matured, both vendor-specific and open standards solutions have emerged. This section will explore some of the basics of network storage and introduce you to some of the key concepts as well as the more advanced topics of SANs.

Basics of SANs and NAS

Let's start with some conceptual information about networked storage. Essentially, networked storage consists of some type of basic OS coupled with large amounts of storage. The OS may be a scaled-back and hardened version of UNIX, Linux, or even Windows. Consider the system illustrated in Figure 1.7. The networked storage system consists of a CPU running an OS that has been customized for sharing storage and a series of disks. The disk array in Figure 1.7 shows one massive drive array; this could in reality be RAID 0, 1, 5, 1+0, or some other striping or striping with parity configuration.

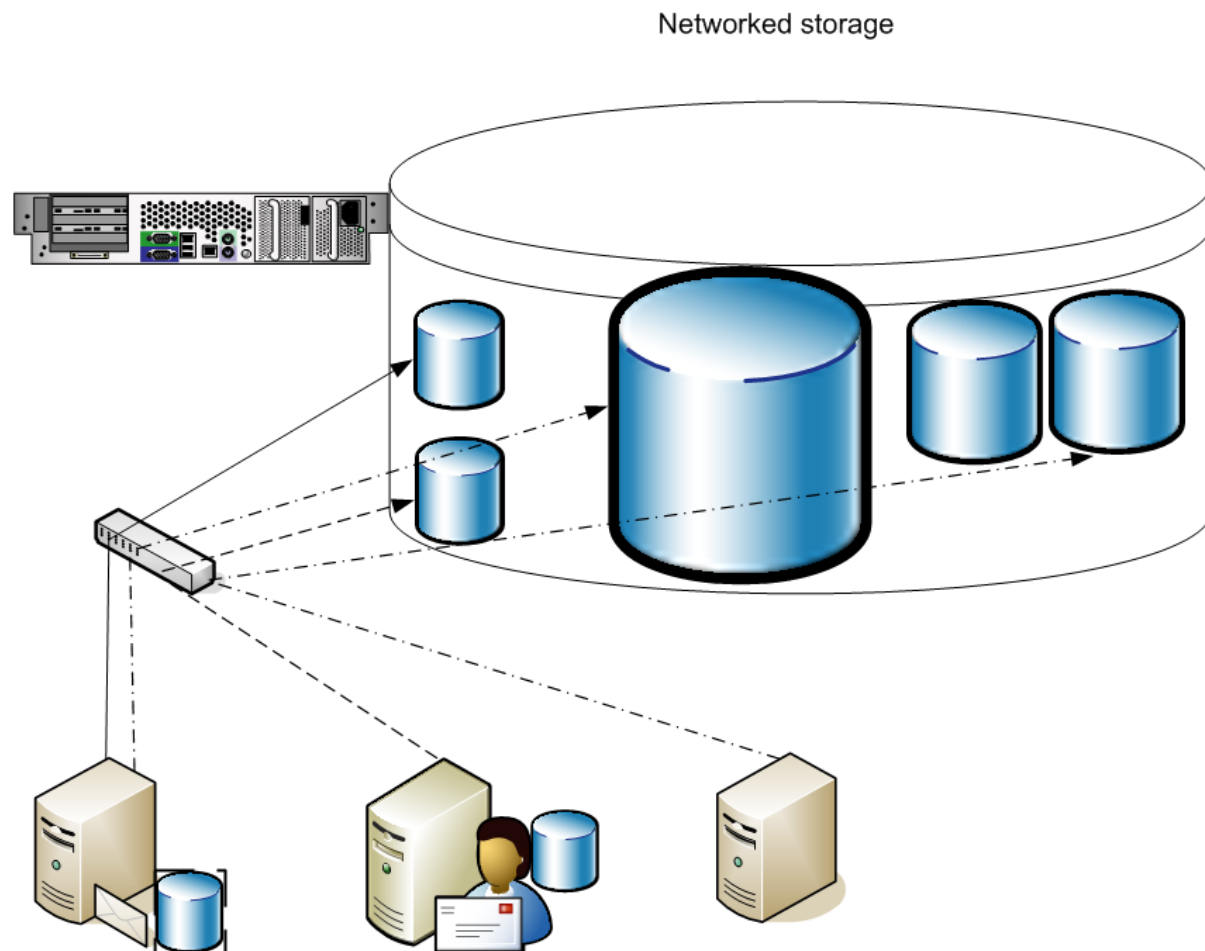


Figure 1.7: Conceptual view of networked storage.

The disk space on the networked storage system is split or carved up into smaller pieces called LUNs. The LUNs are sized based on the servers and functions to which they will be assigned. The servers connect to the networked storage using either Fibre Channel or Ethernet, depending on the SAN protocol in use. Once they are connected to the storage network and have the right device drivers or software installed, the administrator of the server connects to the LUNs designated for a particular server and assigns them drive letters. At this point, the LUN looks to the server just like a local disk. Only one server can use a LUN at a time, although SAN protocols support moving a LUN from one server to another.

Notice in Figure 1.7 that a couple of the servers have local disk drives; this is a normal configuration. However, one of the servers has no local disk drives. Some network storage technologies support something called SAN boot; the server can boot directly off a LUN on the SAN rather than having its own local storage. SAN booting has its own advantages and disadvantages, but they're outside the scope of this guide.

NAS

Before delving into more about SANs, I want to first define NAS. NAS storage generically means just about any storage system that can be accessed via a network. This is certainly not a new concept and Novell's NetWare servers, Sun's network file system (NFS), IBM's LANManager, and the Windows Server family of OSs are all network attached storage systems. These systems provide access to storage via a storage protocol such as NetWare core protocol (NCP), NFS, server message block (SMB), and common Internet file system (CIFS).

OSs such as NetWare and UNIX are general-purpose OSs and the servers on which these are deployed are often capable of serving many purposes. For example, a Windows server could support both Exchange Server and SQL Server and provide file sharing at the same time. Over the years, vendors have taken variants of UNIX and Windows and stripped out many of the "general purpose" features to create platforms that are storage centric. A number of vendors on the market have taken their NAS solutions and allow them to support multiple file-sharing protocols such as NFS and SMB. This allows both UNIX and Windows clients to access the same shared storage. NAS servers can potentially provide both better fault tolerance as well as increased performance.

NAS differs significantly in the implementation of how data is accessed on the shared storage when compared with a SAN. NAS storage uses file-based protocols such as NFS or CIFS; from the Windows point of view, this means that the shared storage must be accessed via disk drive letters that are mapped. SAN storage is accessed directly through NTFS and uses block-level access and is accessed by a connected host using SCSI over Fibre Channel connections or iSCSI over Ethernet.


 NAS systems are not supported in Exchange 2007. You should use Fibre Channel-attached SANs or iSCSI SANs.

Applications such as Exchange Server and SQL Server do not allow you to store data on drive letters that are considered “networked drives.” In the past, many vendors created workarounds using device drivers or services that “faked out” Exchange so that it would see the networked drives as local disks, but Microsoft doesn’t support these with Exchange 2000, 2003, or 2007.

SANs

The most common network storage system in use in enterprise environments today is SAN solutions. SANs come in many shapes and sizes, but they have a few things in common. First is that the clients of the SAN (the servers that have LUNs on the SAN) connect to it over a storage network and use block-level access. The I/O on the storage network is very similar to the types of I/O you would see on internal disk drives. Even though the LUNs that are allocated to a server are on the storage network; the OS and applications see the LUN as a local disk.

Originally, the most common SAN deployment was to have all the server nodes that require connectivity to the SAN use Fibre Channel connectivity. Typical SAN interfaces include 1Gb, 2Gb, and 4Gb Fibre Channel interfaces. In Figure 1.8, the SAN includes two Fibre Channel switches. Each node of the SAN (meaning the servers that will have LUNs assigned to them) have two Fibre Channel cards connected to two different Fibre Channel switches. The switches then connect to the SAN storage hardware. The redundant connectivity provides fault tolerance in the event of a failure.

 Not only disk drives can be made available to members of the SAN; tape drives or tape libraries connected to the SAN can be bridged to servers that use the SAN.

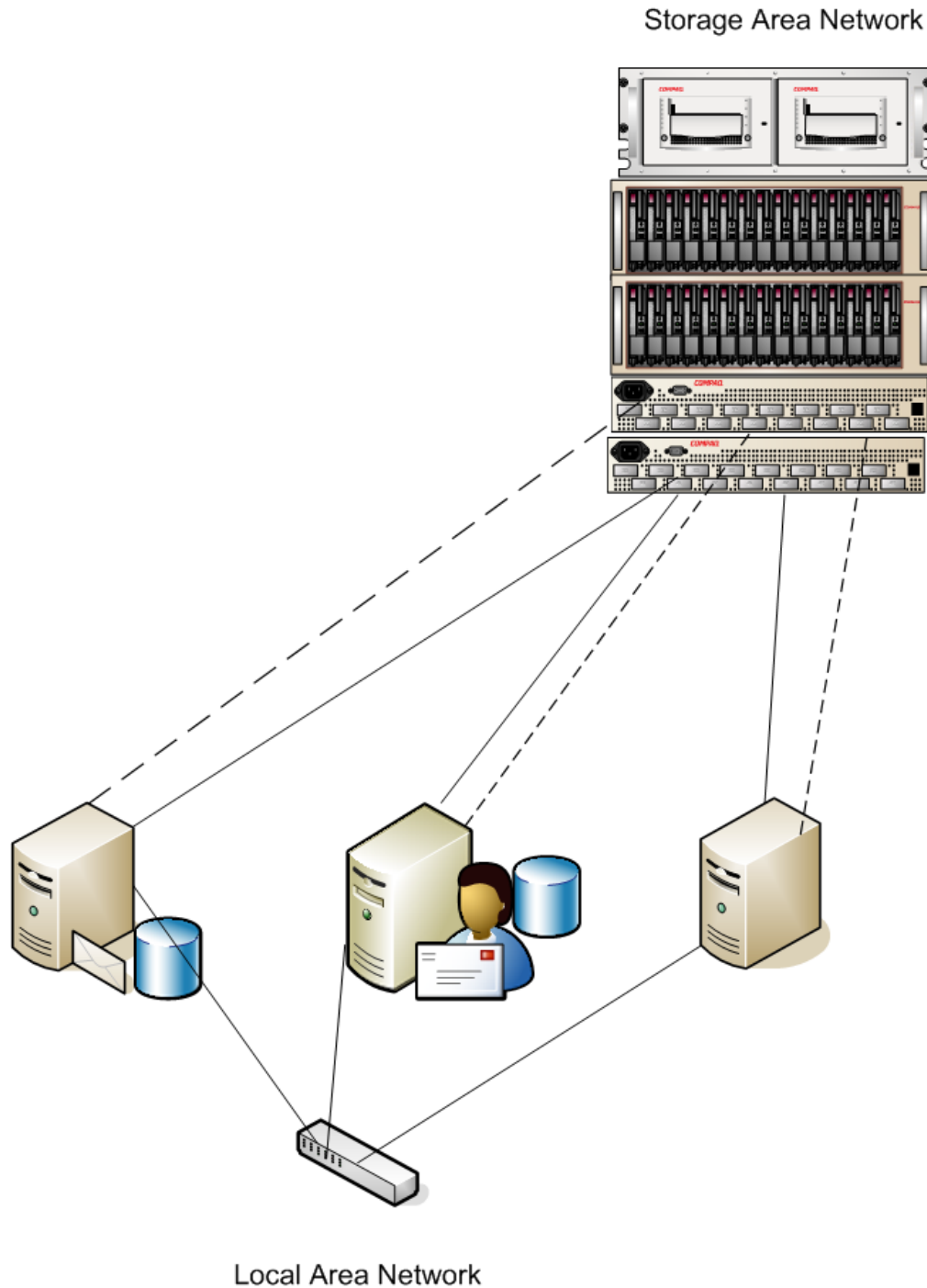


Figure 1.8: SAN using Fibre Channel.

In addition to having Fibre Channel adapters, the servers connect to the local area network (LAN) using traditional network interface cards such as Ethernet adapters. One of the downsides to this type of configuration is that each server that uses the SAN must have at least one (preferably two) Fibre Channel cards. This can get expensive. Providing multiple paths of I/O (multi-path I/O) allows for redundant connectivity to the SAN network; this is an essential part of fault tolerance when designing a SAN.

SANs are often implemented for disaster recovery reasons. Many SAN technologies include replication solutions that can replicate the data stored on one SAN to a different SAN, possibly even in a different location. This makes SAN technologies attractive for organizations looking for a business continuance solution.

Snapshot technology in the SAN allows backup software on the SAN to duplicate the contents of an entire LUN. The snapshots can be used for disaster recovery or replicated to another location for business continuance reasons. Snapshots are attractive disaster recovery tools because only the changes (delta blocks) are recorded since the last snapshot and they are “instantaneous,” eliminating the backup window. However, snapshot solutions that backup Exchange Server databases should use the Windows Volume Shadow Copy Service (VSS) and they must use the Exchange APIs for VSS backups. Backups that are made using non-Exchange API backup systems may not restore properly and may not be supported by Microsoft Customer Support Services.

One major advancement in SAN technology is the standardization of SCSI over IP, also called iSCSI. iSCSI allows for the connectivity of shared storage using a low-level access technique (SCSI) but over an IP network. Chapter will go into more detail on what iSCSI is and how to implement it; for now, let’s just look at the concepts. Figure 1.9 shows a simple iSCSI SAN deployment. A SAN storage module is deployed and (we are assuming) has direct gigabit Ethernet connectivity. All the servers that need a LUN allocated from the SAN connect to a gigabit Ethernet switch.

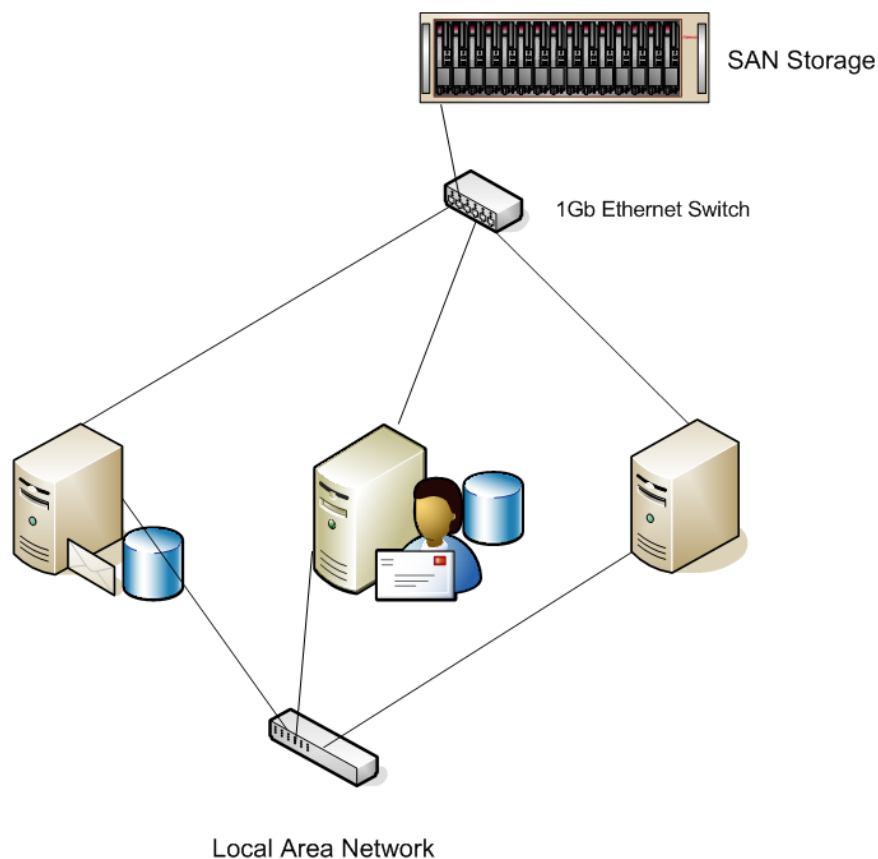


Figure 1.9: Using an iSCSI SAN.

The iSCSI SAN reduces the complexity associated with Fibre Channel SANs and, because many of the components such as Fibre Channel adapters and switches are not required, they are much more economical. Where Fibre Channel SANs require expensive host bus adapters (HBAs), iSCSI SANs can use a server's built-in network adapters. iSCSI SANs also leverage the existing Ethernet networking expertise that virtually all IT departments have to some degree. From a networking perspective, setting up an iSCSI SAN is no different than setting up any IP subnet. Higher-performance network adapters are available that are designed specifically with iSCSI SANs.

Grid or Clustered Storage

When you are looking at SAN features, you may come across a term called grid storage or clustered storage. This is a fairly new concept in SAN design and deployment. Instead of deploying a single large SAN enclosure, multiple smaller, self-contained storage nodes are used. Each of the storage nodes are completely self-contained and include processor, disk storage, and network connectivity.

Multiple storage units can be combined into a single logical storage unit. Figure 1.10 shows a logical SAN that consists of four separate individual storage units or storage modules; each of these storage units could function as a standalone SAN. When combined, the multiple storage units appear as a single logical SAN.

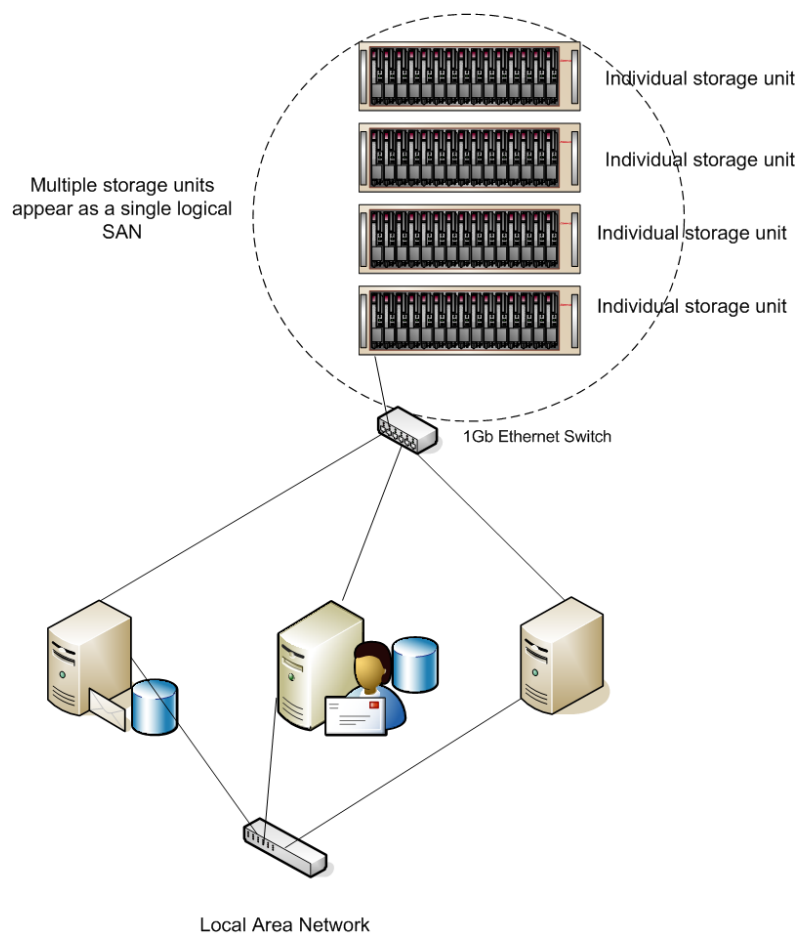


Figure 1.10: Multiple independent SAN systems can function as one logical SAN.

Depending on the SAN vendor, some SANs can even virtualize the storage and balance the load between the storage and the client. Data can be replicated so that the data is found on multiple storage units. LUNs can be assigned to clients on the network and the requests for data can be load balanced across any node in the storage grid. The client does not have to know specifically which storage units in the grid contain the data for its LUN.

Grid storage provides additional redundancy and fault tolerance. Data can reside on multiple storage units in the grid SAN to provide fault tolerance and redundancy. This can also help reduce downtime because an entire storage unit can be taken offline for maintenance, but remaining storage units in the storage grid can continue to service requests.

One of the biggest advantages of grid storage is scalability. If you need to add disk capacity or I/O capacity, you do so by adding storage units. As new storage units are added to the grid, they are automatically recognized by the rest of the grid. grid and the existing LUNs are “restriped” to take advantage of the new units. Thus, even performance problems can be easily solved by the addition of a new unit.

Tips for Choosing a SAN Solution

Over the years, I have seen many companies wooed by slick pitches of storage consolidation, centralized management, and promises of easy storage growth. Even as soon as a year later, these companies have several racks full of equipment that now must be upgraded due to insufficient capacity or outdated hardware or because the system does not fully meet the organization’s needs.

With respect to Exchange, I frequently see SAN storage systems that are not configured correctly to support the I/O load (more on that in Chapter 2), and SANs that do not have the features necessary to support factors such as snapshot backups and restores. With that in mind, I have put together a list of questions that should be asked of any storage vendor you are considering:

- How well does the SAN scale?
- If a LUN is out of I/O capacity, how do you scale for additional I/O capacity?
- If you add drives for increased capacity or performance, what’s the process for allocating those new drives to existing volumes?
- How do you add capacity to the SAN? How is this capacity allocated to individual volumes? What downtime is required?
- How do you add capacity to existing volumes? Is this process automated?
- Does the SAN support snapshots? What are the limits per array? Per volume? Per LUN?
- How is the snapshot disk space management done? How much space should be allocated for snapshots?
- Are the snapshot solutions compatible with Exchange Server backups (for example, VSS support)?
- What types of redundant connectivity and multi-path I/O is supported and how many network connections?
- If multi-path I/O is supported, can more than one path be used at once?

- Are all hardware components redundant? What combinations of failures cause data loss?
- How are disks replaced when they fail? Does this require downtime? Is it automatic? What happens if you have multiple drive failures?
- What notification and monitoring systems are in place for SAN events such as disk or power supply failures?
- Is there replication technology available on the SAN?
- What types of fault-tolerant solutions are available? Are they configurable or not?
- Can the various protection mechanisms (RAID, replication) be applied on a per volume basis or do they apply to the entire array?
- Are there triple and quadruple protection mechanisms available for highly critical volumes?
- How is multi-site disaster recovery accomplished for the SAN? What additional products are required?
- How is the SAN firmware/software upgraded? What downtime is required for upgrades?

In addition to being comfortable with the answers that your prospective SAN vendors provide, you should be concerned with the experience that the SAN vendor has with your applications. Some additional information I want to be comfortable with when working with a storage vendor includes:

- Experience with the applications I am supporting, including the specific versions
- 24 × 7 support
- Understanding I/O capacity requirements for applications such as Exchange Server and SQL Server

Summary

This chapter discussed some of the basics of Exchange Server data storage, data files, and how these files might affect your storage and I/O capacity requirements. It has also introduced to you the concept of using disk storage that is not directly attached to your server. Finally, the chapter discussed some of the basics of networked storage, specifically SANs.

The coming chapters will apply this knowledge of Exchange requirements and storage planning to learn more about how to adequately plan not only for storage capacity but also for sufficient I/O capacity. The next chapter will introduce to you the concept of I/Os per second (IOPS) and how to plan for sufficient IOPS capacity. The chapter will also look at some of the most common mistakes that administrators and messaging engineers make when estimating storage and I/O capacity.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.