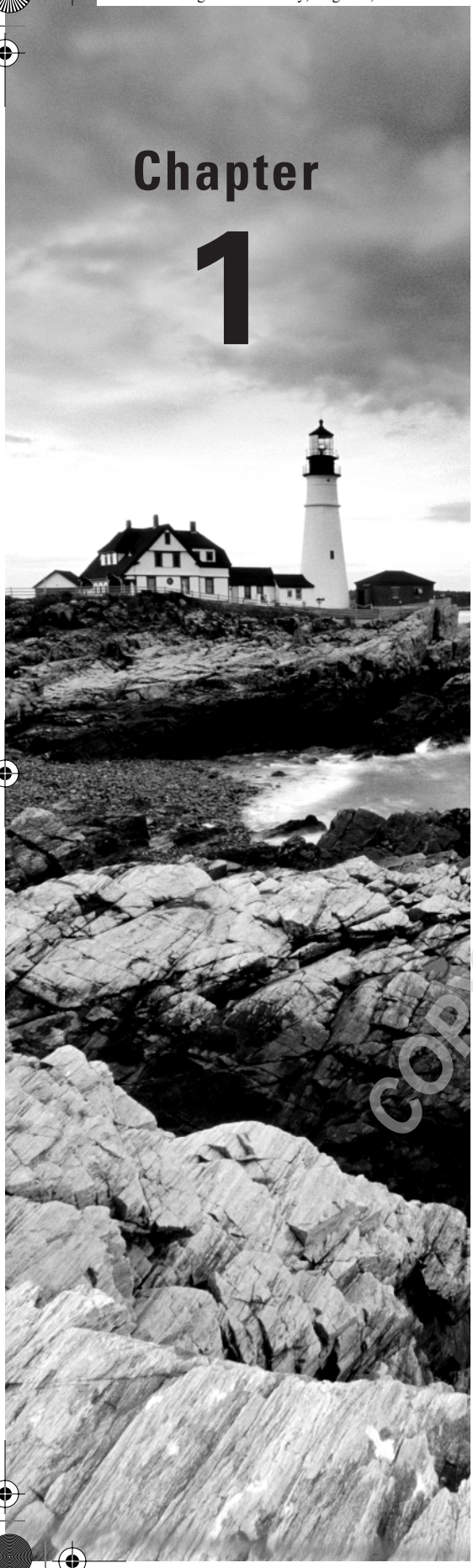


Chapter 1

Designing the Hardware and Software Infrastructure

MICROSOFT EXAM OBJECTIVES COVERED IN THIS CHAPTER:

- ✓ **Design for capacity requirements.**
  - Analyze storage requirements.
  - Analyze network requirements.
  - Analyze CPU requirements.
  - Analyze the current configuration.
  - Analyze memory requirements.
  - Forecast and incorporate anticipated growth requirements into the capacity requirements.
- ✓ **Specify software versions and hardware configurations.**
  - Choose a version and edition of the operating system.
  - Choose a version of SQL Server 2005.
  - Choose a CPU type.
  - Choose memory options.
  - Choose a type of storage.

There is an old saying among carpenters and woodworkers when preparing to work with a good or particularly special piece of hardwood: "Measure it twice, and saw it once." In other words, make sure of what you're doing and then get it right the first time, because it might be the only chance you have.

Added to that old saying are others: "No one plans to fail, they fail to plan"; "Failure to plan on your part does not constitute an emergency on mine"; and "A house built on sand will fall down."

The point is to emphasize the role that careful planning and design of the underlying support structure, the *infrastructure*, play in the successful completion of any project—from a child's dollhouse to a family vacation to a career.

If you were going to build a house, either for yourself or someone else, the first thing you'd want to know is how it was going to be used and how big it needed to be. To find out, you'd ask yourself (or a client) some key questions: How much land is there to build on? How many people will live there now? Are there plans for additional children? Will it be only a house, or will it serve as a home office? Is a separate section with a separate entrance required? How much money and resources are available? With that information, you'd then design and build accordingly. When it comes to a database server infrastructure, you need to do the same.

Let's re-emphasize that since it's very important that you grasp the underlying premise to this chapter and the rest of the book: If you understand how to plan and design a database infrastructure and how to successfully implement those plans, you will reap enormous benefits in terms of time saved and resources properly allocated while increasing the probability that your activities will succeed.

The process of designing infrastructure often depends more on your understanding of the underlying premises rather than on a single set of rules. Every infrastructure you design or work on will need to meet unique requirements—there is no "one size fits all." Unless you understand the hows and whys of that process, the end result will be far from satisfactory.

In this chapter, you'll take the first steps toward designing a database server infrastructure. Like anything you build, be it a birdhouse or the Great Pyramid, the foundation is the key. We'll first review strategies for assessing your current configuration and gathering data about the current capacity of key resources such as storage, CPU, memory, and network bandwidth. We'll then cover how to use this data, along with the business requirements of the organization, to estimate future capacity needs. The second part of the chapter will look at how to specify software versions and hardware configurations for use in the organization's requirements.

Analyzing the Current Configuration

As you almost certainly know, you'll rarely, if ever, be involved in designing a completely new database server infrastructure. You'll nearly always be working with an organization that has an existing infrastructure that needs to change to meet enterprise growth and to enhance performance.

In that case, the first step is not to reach for a piece of paper to draw your dream infrastructure. Instead, the first step is to evaluate the various subsystems of the existing infrastructure and figure out what you have to work with. This initial evaluation process will aid you in assessing how well the different subsystems interact and will also highlight potential trouble spots.

Next, you should gather the requirements you need to have in place for the modified infrastructure. These requirements may be technical or business-related (they're usually both), and they need to be prioritized in that context. Once you've established the requirements, set the priorities, and determined the funding levels, you can design modifications to the infrastructure.

When designing modifications, a good practice is to standardize the hardware and software configuration of database servers as much as possible. Doing so simplifies the design of the infrastructure and reduces the maintenance overhead. In addition, standardization results in significant cost savings.

Thinking Holistically

Whether before or after you've analyzed the capacity needs of the enterprise's individual database servers, you must at some point—the earlier the better—evaluate the existing database server infrastructure as a whole. This view can give you a quick assessment of the overall health of the infrastructure and help you determine any recurring trouble spots.

You should also think in terms of the ideal. Are the databases optimally designed? Are disk storage systems being used effectively? Are CPU and memory types and allocations appropriate? Is the network properly designed and prepared for the new infrastructure?

You should use your evaluations to determine what modifications should be made to the infrastructure to support business growth. And you should be able to make the business case for your recommendations.

Assessing How Good the Current Configuration Is

You should take a number of steps at the outset to assess the condition of the current database server configurations:

- Inventory the operating system versions, service packs, and hotfixes running on each database server.
- Verify whether the necessary OS service packs and hotfixes have been applied.

- Identify any compatibility issues between the operating environment and the applications running on the database server.
- Inventory the SQL Server versions, service packs, and hotfixes.
- Verify whether the latest service packs or hotfixes have been applied.

You can download the latest SQL Server service packs and hotfixes from the SQL Server website at `http://www.microsoft.com/sql/downloads/default.mspx`.

- Inventory what SQL Server services are running on each database server and what service accounts have been assigned to each. To do so, you can use SQL Server 2005 Configuration Manager. A short list of important services includes:
  - SQL Server Engine
  - SQL Server Agent
  - SQL Server Full-Text Search
  - SQL Server Reporting Services (SSRS)
  - SQL Server Analysis Services (SSAS)
  - SQL Server Browser
  - SQL Server Integration Services (SSIS)

For more information about the service accounts for SQL Server services, refer to the topic "Setting Up Windows Service Accounts" in SQL Server 2005 Books Online.

- Inventory hardware configurations, including disk subsystems, CPUs, memory, network cards, and power supplies on database servers. Make note of RAID or SCSI use. Identify all servers in a cluster, if a clustering environment is being used.
- Record SQL Server configuration settings. Record the minimum and maximum memory settings, the CPUs used by the instance, and the default connection properties for each SQL Server instance.

Use the stored procedure sp_configure with "show advanced option" to display the advanced options. SQL Server Configuration Manager can help you collect network configuration data, such as the libraries, protocols, and ports for each instance.

- Review the configuration management process for proposing, approving, and implementing configuration changes, and identify opportunities to make the process more efficient. What tools are used?

- Assess the quality of the database server documentation.
- Verify the capabilities of disk subsystems and physical storage. Determine whether the RAID levels of existing disk subsystems support data availability and performance requirements.

Chapter 2 discusses disk subsystems and physical storage design considerations in more detail.

- Determine the locations of transaction log files and data files.
- Examine the use of filegroups.
- Are adequate data-file sizes allocated to databases, including the tempdb database?
- Verify that the AutoShrink property is set to False, to ensure that the OS files maintaining table and index data are resized downwardly.
- Determine whether disk-maintenance operations, such as defragmentation, are performed periodically.
- Assess Event Viewer errors to identify disk storage–related problems.

Forecasting and Incorporating Anticipated Growth Requirements into the Capacity Requirements

Requirements analysis is key to the process of designing modifications to a database server infrastructure. Just as you need to know the purpose of a house in order to build one that meets your needs, you must properly identify the business requirements in order to design your infrastructure. Otherwise, your design won't meet the needs of the organization; and not only can you forget professional pride, you'll be lucky if you still have a job.

It's essential that you always work in a collaborative way with company management, IT staff, and end users to identify both the technical and business requirements that the new database infrastructure is expected to support.

There is an intricate dance between the technical aspects and the nontechnical aspects of a project, and weaving them together seamlessly is one of your most important, if never really specified, jobs. Technical aspects and requirements typically focus on tasks such as capacity, archiving, distribution, and other needs. These are affected by business requirements that include budgetary and legal restrictions, company IT policies, and so on. Successful comprehension of *both* sets of requirements allows you to know precisely the scope of modifications to the infrastructure and establishes a valuable foundation on which to base your design and modification decisions.

When designing modifications to a database server infrastructure, you must consider its future capacity needs based on the projected business growth of the organization. In addition, you must consider requirements pertaining to data archiving, database server consolidation, and data distribution.

Technical Considerations

We'll spend the rest of this chapter talking about specific capacity needs, usually when talking about a specific server; but it's crucial to a successful design that you analyze and identify the capacity requirements of the database server infrastructure as a whole.

Because it's difficult to extrapolate the capacity needs of the entire infrastructure, you may not always be able to project growth except in qualitative and general terms. You should, nonetheless, answer these questions for your future planning estimates and projections:

- Is the enterprise planning to, or likely to, experience growth through increases in business operations, customer base, or increased demand for databases?
- Are there plans to utilize applications that require additional databases?
- Are there plans to improve the database server hardware?
- What cost variations, such as a decrease in the cost of servers and storage devices, will you see because of market forces?
- What data archiving requirements exist? Will these change? Do they differ by department?

Chapter 12 covers data in detail.

- What regulatory requirements are in place or are being contemplated? What security components do they involve?
- Are any database servers potential candidates for consolidation?

Chapter 2 covers database consolidation.

- Are there opportunities to optimize the data-distribution process through simplification and/or standardization?

Business Requirements to Consider

As we already mentioned, the nontechnical, business aspects of an organization or enterprise play a major role in determining the shape and scope of any infrastructure system you design. When considering business requirements, you should be aware of any and all budgetary constraints, IT policies, and industry-specific regulations. Additionally, you should consider the organization's data security and availability needs and requirements:

Budgetary constraints The amount of money an organization is willing to spend will obviously affect the sort of database server infrastructure you can design. You can budget monies for a project in a number of ways, but one of your key roles in the process is to design within

the constraints imposed by the bottom line. An ancillary role you play is to make business cases when the budget is unrealistic or when spending money now may produce a better return on investment (ROI) later.

Existing IT policies Any modifications to a database infrastructure must comply with existing IT in the organization. Normally, these policies cover the following:

- Remote access procedures and rules
- Security policies including encryption
- Service-level agreements (SLAs)
- Standard hardware and software configurations

Regulatory requirements With the collection of data come twin demands for greater privacy and security; at the same time, there are demands for data retention. All these demands, as you'll see throughout this chapter and the rest of the book, translate into infrastructure-related requirements. For example, the healthcare and banking industries have strict privacy requirements that translate into requirements for data security and auditing and maintaining certain data-specific time periods. These requirements primarily affect disk space storage and archiving needs and design considerations.

Data security An organization's data security requirements include the following:

- Confidentiality agreements with customers
- Privacy restrictions
- Data encryption needs
- External regulations

Chapter 4 covers database security. Chapters 5–7 cover other security-related issues.

Data availability Typically, the overall infrastructure's data-availability needs are a reflection of the data-availability requirements applied to individual database servers and then generalized for the entire infrastructure.

One other availability issue to consider applies only if the planned modification of the database infrastructure results in a planned or unplanned loss of data availability. Regardless of whether you anticipate a potential loss of availability across the infrastructure, it's important that you have solutions in place to prevent or minimize the loss. This may include placing mission-critical data on a redundant site that can be used as an emergency backup during infrastructure changes.

Now that we've examined some infrastructure-wide and general considerations for assessing, planning, and modifying a database server infrastructure, let's look at the process in more detail, especially as it relates to specific capacity requirements.

Designing for Capacity Requirements

There are two sources of capacity requirements: the business and technical requirements of the organization. The technical requirements are dictated by need and availability. You should also determine the business goals of the organization for which you're developing the database infrastructure. Without knowing those, you can't analyze or forecast its capacity needs, any more than you can build the best possible house without knowing what it will be used for.

With those two points in mind, you have two other key tasks to perform: assessing the current capacity of system resources; and identifying any information, such as growth trends, that you can use to forecast future needs. Most of the time, you can correlate the trends with a variable that can be measured, such as the *database transaction growth rate* (the rate at which the read/write activity on the database server grows.)

In the following sections, you'll learn how to gather data about the current capacity of key system resources such as storage, CPU, memory, and network bandwidth. Then, you'll learn how to use the data to estimate future capacity needs, using that information to design (or redesign when one exists) a database infrastructure.

Analyzing Storage Requirements

A lot of considerations go into analyzing the storage requirements of a database server. In addition to the physical size of the database, you need to consider the transaction growth rate and data-distribution requirements. Some industries, particularly financial and healthcare institutions, are subject to requirements regarding storage and security that must be taken into account in determining storage capacity. You'll now learn how to determine the current storage capacity of a database server and identify factors that affect its growth. We'll also look at how to forecast future disk-storage requirements, taking into account any relevant regulatory requirements that may apply to your business or enterprise.

Assessing Current Storage Capacity

Typically, you won't be starting from scratch when designing a database infrastructure; you'll be reviewing and upgrading an existing system. Even if the recommended upgrade calls for a complete overhaul of the system, you need to be fully aware of the database server's current storage capacity. In making your survey and determination of current storage capacity, consider the following factors:

Disk-space capacity Establishing disk-space capacity requires several steps. First, determine how much disk space is used by the database data files. Then, add the space required for the database's transaction log files, the portion of tempdb that supports database activity, and the space being used by full-text indexes. Look for any maintenance operations that may require extra disk space for the database files, particularly the transaction log files, such as index reorganization.

If you're examining an existing system, make sure you base your measurement of the current disk usage on a properly sized database and that adequate disk space is already allocated for data and log files. If adequate disk space is allocated for these files, SQL Server 2005 doesn't need to dynamically grow the database and request extra disk space from the operating system. The process of

allocating extra disk space for a file uses significant disk resources. In addition, the process can cause physical file fragmentation, because disk segments are added to an existing file.

Disk throughput capacity Next, assess the disk I/O rate that the database requires. You can use System Monitor's PhysicalDisk: Disk Read Bytes/sec and the PhysicalDisk: Disk Write Bytes/sec counters to measure disk I/O. If the database receives mostly reads, also check the Avg. Disk Read Queue Length counter value, which should be low for optimal performance. If the database primarily receives writes, check the Avg. Disk Write Queue Length counter value, which should also be low.

Locations and roles of database servers When you're working with a distributed environment, you should establish where the database servers are (and should be) and their different roles, because that may require a different disk-capacity assessment for each site and each server. For example, the servers at an organization's branch offices may store only a subset of the data that is stored on the main server at headquarters. Based on the roles of the servers, you may be able to identify which databases are most likely to experience growth in disk-space usage or have particularly high or low disk-space requirements.

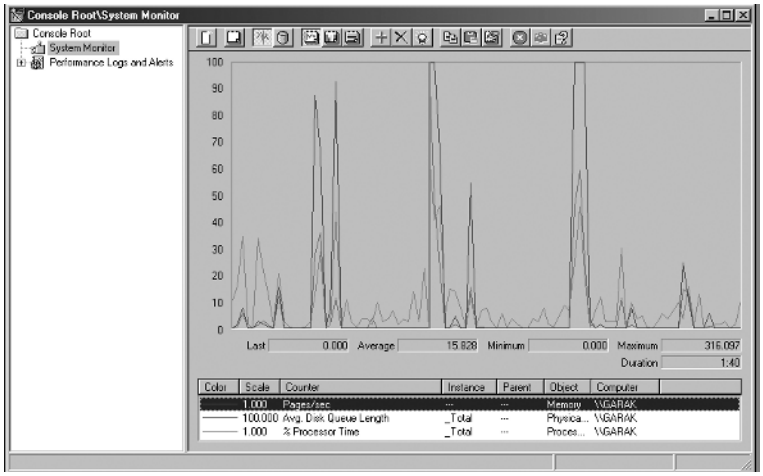
In Exercise 1.1, you'll learn to use System Monitor to assess current disk throughput.

EXERCISE 1.1

Using System Monitor to Assess Current Disk Throughput

To complete this exercise, you must be running a version of Windows that supports SQL Server 2005. You must also have sufficient privileges to access the System Monitor tool.

1. Click Start, select Run, and then type **perfmon** in the text box.
2. System Monitor opens. By default, it displays a line graph of Pages/sec, Avg. Disk Queue Length, and % Processor Time.



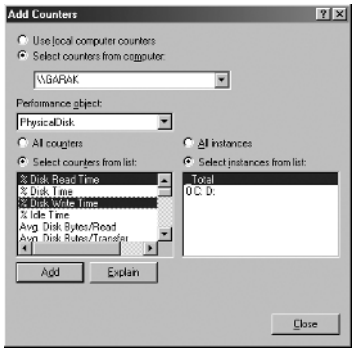
EXERCISE 1.1 *(continued)*

3. To assess the current disk throughput, select relevant counters and log the result. In this exercise, you'll remove the Pages/sec and % Processor Time counters. Highlight each counter, and press the Delete key on your keyboard.

4. Right-click Avg. Disk Queue Length, and select Add Counters from the pop-up context menu. The Add Counters dialog box opens.

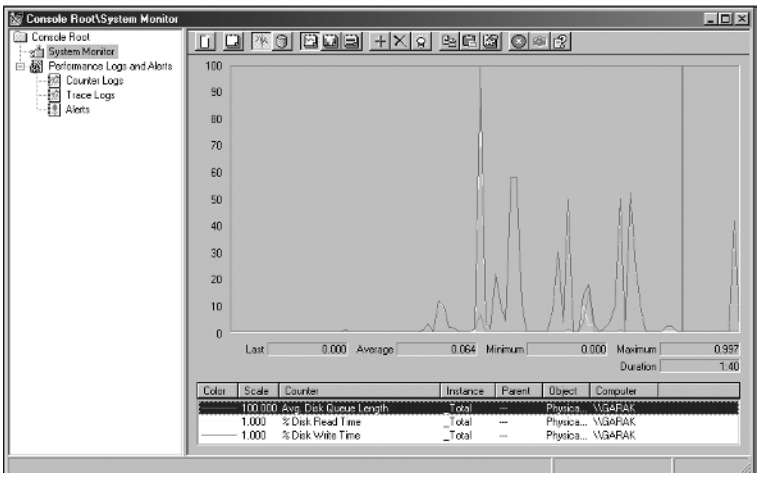
5. In the Performance Object drop-down box, select the value Physical Disk, and then highlight % Disk Read Time.

6. While holding down the Ctrl key, select % Disk Write Time.



7. Click Add. Click Close.

8. Note that there are three counters. Perform some simple tasks, such as opening or closing programs to generate disk activity, and watch how the counters change. Don't be alarmed or surprised if one or more of the counters reaches 100% for a short time.



EXERCISE 1.1 *(continued)*

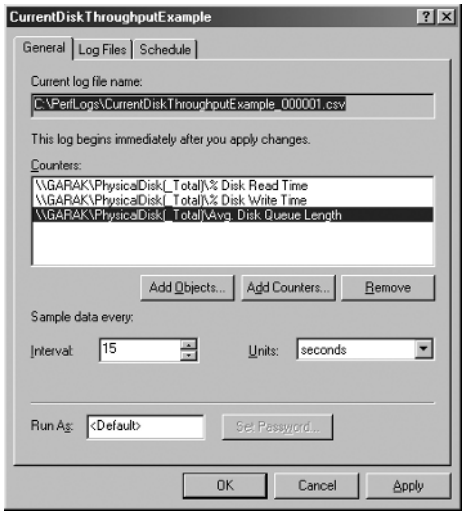
9. Although you can sit there and look at the table for hours to detect trends and periods of peak usage, it's not very practical. In order to be able to make decisions regarding disk usage, you should collect log data over time. To do so, expand the Performance Logs And Alerts node in the left window pane.

10. Right-click Counter Log, and choose New Log Settings from the pop-up menu

11. Name the new log settings **CurrentDiskThroughputExample**. Click OK.

12. Click the Add Counters button to open the Add Counters dialog box. Select PhysicalDisk from the Performance Object drop-down box. While holding down the Ctrl key, select % Disk Read Time, % Disk Write Time, and Avg Disk Queue Length.

13. Click Add. Click Close. Note that you can set the interval sampling time in the Sample Data Every section. The sample period will depend on the item you're measuring and how large a log file you wish. One key consideration is that the monitoring consumes resources and may as a result produce skewed readings, especially when monitoring the local computer. Click the Configure button on the Log Files tab to open a separate dialog box where you can specify the log file name, folder, and size..

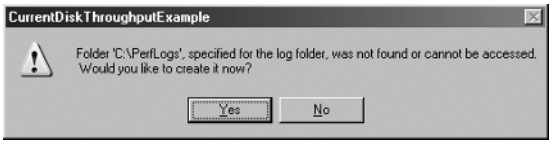


14. Click the Log Files tab. In this window, you can set the log-file type and make other configuration changes. The default log-file type is Binary. For this exercise, change the type to Text File (Comma Delimited).

EXERCISE 1.1 *(continued)*

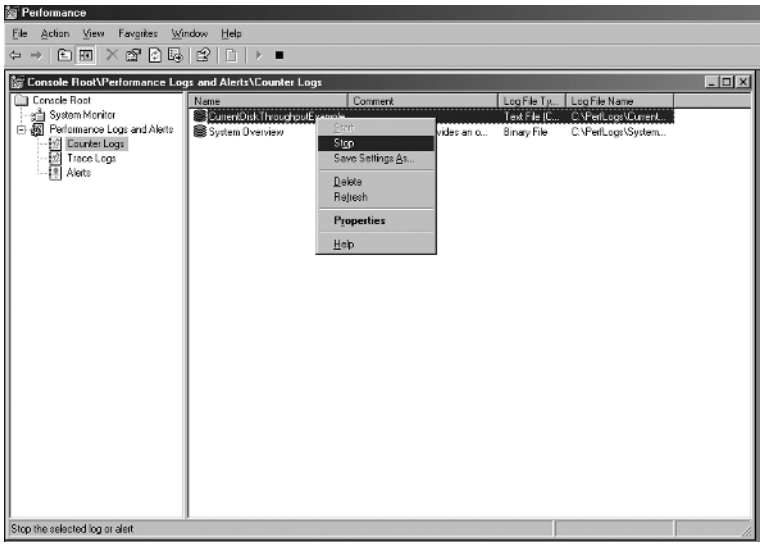
15. Click the Schedule tab. By default, the log file starts as soon as you click Apply or OK and runs until you stop it. Using this dialog box, you can set logging to start at a particular time and date and stop after a specified interval or at a time or date.

16. Click OK. If the specified folder doesn't exist, you receive a warning and are given the opportunity to let the program create the folder.



17. Click Yes. Let the program run for a few minutes. You can force disk activity by opening and closing programs, creating and deleting files, and performing other basic disk subsystem tasks.

18. After no less than three minutes, click the Counter Logs node in the left pane of System Monitor. Highlight CurrentDiskThroughputExample. Right-click, and select Stop from the context menu to end logging.



Note that the icon turns red to indicate "stopped."

19. Navigate to the folder where the log file is located. You can open it with Notepad, but you'll find that CSV files work best with Excel. If you chose the SQL Database log-file type, you can open the database with SQL Server 2005.

Forecasting and Planning Storage Requirements

When you're building a house, you want to design it with an eye on future needs and plans. The same applies to storage requirements. First, you need an idea of what the future plans are; armed with than information, you can begin planning storage requirements. You need to consider several key factors in planning for the future.

Establishing the Estimation Period

Establish at the outset how long you're planning for (in other words, the *estimation period* or the length of time for which the planning is valid). Are you establishing a plan that should be valid for one year, two years, five years, or more? Review your assumptions regarding database needs for the period. Are they valid? Do your estimates of the enterprise's future, such as its anticipated growth, match those of the enterprise? Determining how long the estimation period is and what forms it will require that you work with management and other key stakeholders in the enterprise. Often, they will need to reconcile conflicting ideas to come to a consensus on what the period should be. Don't be surprised if they turn to you for expert advice and mediation of internal disputes.

Projecting the Growth Rate of Required Disk Space

Obviously, you must estimate the amount of future disk space required. There are two ways to do this effectively; you can either base your projection on an existing source of data that correlates well with growth; or you can follow a rule-of-thumb estimate when you don't have a correlating variable.

Ideally, you should correlate the growth rate with a measurable variable such as increases in the transaction rate or user load. If you can identify such a variable, you can effectively estimate future growth in disk space. For example, a clear correlation may already exist between the growth in disk space for key tables and the number of new orders in a day. If so, and if other variables are insignificant, then you can use the rate of growth in new orders to estimate the rate of growth in disk space required.

If you don't have a correlating variable, you can use past growth trends to estimate future trends. In some cases, historical trends may be the only data you have for estimation.

You can make an estimate of future trend using any of a number of formulas where

F = Future disk space

C = Current disk space

T = Growth-rate time unit

A = Growth amount

R = Rate of growth

Linear growth If you expect disk space to grow by a constant amount in a specific period, the growth is *linear*. In that case, you can apply the following formula:

$$F = C + (A * T)$$

For example, if you have a 1000GB database that's expected to grow at 100GB per years, in four years the database is expected to be 1400GB: 1000GB + (100GB * 4) = 1400GB

Compound growth If you expect disk space to grow at a constant rate during a specific period (for example, at a certain percentage per month or per quarter), that growth is described as *compound*. In that instance, use the following formula:

F = C * (1 + R)^T

For example, if a 800GB database is expected to grow by 3 percent per quarter for two years, the resulting additional disk space required in eight quarters (two years) will be 800 * (1 + .03)^8 = 639GB.

In this type of formula, you should express the growth rate as a decimal translation of the percentage value. For example, if the growth rate is 3 percent per quarter, use the value 0.03 in the formula. In this example, the number of periods is specified in quarters so that it's consistent with the growth-rate unit.

Geometric growth If the disk space is expected to grow periodically by some increment, but the increment itself is also growing at a constant rate, the disk space requirement grows geometrically. In this case, use the sum of a series formula (also called a *geometric series)* to determine the projected size:

F = C + ((Initial Increment) * (1 – Increment Growth Rate)^(T + 1))) / (1 – Growth Rate))

For example, if a 1000GB database grows by an increment that starts at 3GB per month and increases at 2 percent per month, then in 24 months the total disk space required will grow to 1000 + (3 * (1 – 1.02^25)) / (1 – 1.02) = 1096GB.

In Exercise 1.2, you'll try your hand at forecasting future disk-storage requirements.

EXERCISE 1.2

Forecasting Future Disk Storage Requirements

You've been employed by the Yanni HealthCare Services Network to help modify the existing database server infrastructure. The company currently uses 600GB of disk storage. A review of company plans and relevant regulatory requirements related to HIPAA leads you to conclude that the company will experience a 2 percent increase per month through the four-year life of the estimation period.

Calculate the expected disk space requirement at the midpoint and end of the estimation period. How much extra disk space will be required beyond the current levels?

1. This is an example of compound growth, so apply the formula F = C * (1 + R)^T, where the following is true:

   F = Future disk space

   C = Current disk space

EXERCISE 1.2 *(continued)*

T = Growth rate time unit

A = Growth amount

R = Rate of growth

2. To assess the need at the midpoint, set C = 600, T = 24, and R = .02.
3. F = 600 * (1 + .02)^24 = 965GB. This is an extra disk-space requirement of 365GB at the midpoint (two-year point).
4. To assess the need at the end of the period, set the formula values to F = 600 * (1.02)^48 = 1552GB, or 952GB extra needed by the end of four years. As you can see, because of the compounding effect, almost twice as much disk space will be required in the second half of the period than in the beginning.

Understanding the Impact of Regulatory Requirements

As mentioned earlier, legal considerations and/or governmental and financial regulations, such as those for banks and the healthcare industry, can affect how long you need to retain data and how secure it must be. Both of these factors in turn affect infrastructure design not only in terms of security but also in terms of storage requirements:

Longevity Regulations may specify the length of time for which data must be maintained. Something as simple as immunization records, for example, needs to be retained for 20 years. Banks may also be required to store certain types of customer data for a specific number of years.

Before estimating disk-space capacity, determine what data must be available online. For any data you consider storing offline, assess how quickly the data must be available for online access.

Privacy/Security Regulations, industry guidelines, or legislation may mandate that security measures, including encryption, be undertaken to protect consumer data. For example, health insurers may be required to ensure the privacy of data. Such regulations affect the data distribution strategy and, consequently, the disk-space capacity of local and remote servers.

Privacy-related regulations may require data to be stored in an encrypted format. In SQL Server 2005, you can store data in an encrypted format by using several encryption algorithms. However, encrypted data requires more disk space than nonencrypted data. In addition, encryption increases CPU usage.

Analyzing Network Requirements

All database administrators and infrastructure designers should have a nuts-and-bolts understanding of the topology and capacity of the network supporting the database servers, because this impact infrastructural decisions. Available bandwidth, for example, plays a large part of

determining the backup strategies you use and the types of database services you utilize. In the following sections, you'll learn how to identify the database components of the network topology. We'll also look at factors you should consider when analyzing the current network traffic. Finally, you'll learn how to estimate future network-bandwidth requirements.

Assessing Current Network Capacity

You should review several things when assessing current network capacity.

Identifying the Parts of the Network That May Affect Database Traffic

Create a network diagram to identify the parts of the network that

- Deliver replicated data to other servers
- Back up files to network devices
- Provide data to client applications

Identify and assess the location of the following to determine weak points and potential bottlenecks in the topology, such as low-bandwidth wide area network (WAN) links. Also be aware of the security aspects of the network and the impact they have on traffic:

- Local and remote connections between database servers
- Firewalls
- Antivirus applications

Assess capabilities of the database servers on the network by gathering the following information about each:

- Number of SQL Server instances
- Instance names
- Installed SQL services
- Network protocols

In Exercise 1.3, you'll use SQL Server Configuration Manager to gather information about database servers.

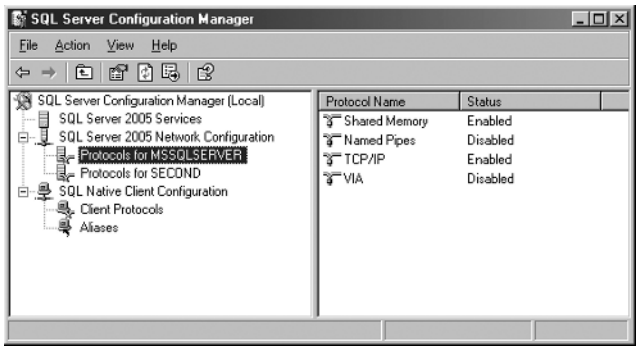
EXERCISE 1.3

Using SQL Server Configuration Manager to Gather Information about Database Servers

1. Select Start ➢ Programs ➢ Microsoft SQL Server 2005 ➢ Configuration Tools ➢ SQL Server Configuration Manager.
2. In the leftmost pane of the SQL Server Configuration Manager dialog box, select SQL Server 2005 Services to display the list of installed services on each instance and their current state.

EXERCISE 1.3 *(continued)*

3. Select SQL Server 2005 Network Configuration, and select any instance. The rightmost pane shows a list of installed network protocols and their status.



Analyzing Current Database Network Traffic

You should analyze current database network traffic to estimate whether, and how long, the existing network can support your database server infrastructure. If the network can't effectively handle current or future increases in traffic as a result of business growth, the performance of the database servers on the network will be adversely affected by traffic bottlenecks.

Analyze the traffic between servers and between clients and servers. Then, use the data you gather to identify potential bottlenecks. Key areas to review include:

Traffic between servers Use System Monitor counters to analyze the traffic caused by backup processes, database mirroring, and replication.

Backup processes The SQLServer: Backup Device:Device Throughput Bytes/sec counter specifies the number of bytes per second that the backup device currently supports. You should also review the backup strategy. If the amount of data is large and available network bandwidth is low, frequent backups to network devices can saturate the network.

Database mirroring The SQLServer:Database Mirroring:Bytes Sent/sec and Bytes Received/sec counters indicate the number of bytes transferred from the principal server to the mirror server.

Replication No single set of counters in System Monitor helps you analyze all replication traffic; hence, what you need to use will depend on the type of replication being used. In the case of subscribers, for example, you can monitor commands delivered per second or transactions received per second.

Traffic between clients and servers Among other things, you must determine the client traffic on the network, assess how well the current network supports the user load, and identify the additional traffic that will be caused by an increase in user load or changes in the application.

A useful technique is to use the System Monitor counter Network Interface: Bytes Totals/sec counter to establish the number of bytes transferred across the database server's network interface. You need to do this for each network interface the server has. Look to see if a correlation exists between the Network Interface: Bytes Totals/sec counter and the SQLServer:General statistics:User connection counter. By doing this, you can determine the network traffic caused by users.

Potential bottlenecks Running Network Monitor on the database server lets you determine the number of bytes used, the percentage of the total bandwidth used, and the number of bytes broadcast in a specific period. You can also filter specific patterns and protocols for a more granular approach. Analyze these data to identify bottlenecks, and work with the network administrator on strategies to eliminate the bottlenecks.

To use Network Monitor for monitoring database traffic across servers, use the Network Monitor version included in System Management Server (SMS).

Forecasting and Planning Network Requirements

Now that you have a good grasp of the present network situation, you should take time to forecast network traffic growth for database servers. As with all forecasting, there are no hard and fast rules (nor guarantees of being 100% accurate—just ask a weatherman), but you should do the following:

Make a growth estimates for each network type Because different network types support varying data flows and volumes, it's essential that you assess each network type individually when determining growth estimates. A good tool is the network diagram mentioned earlier in this section. Use it to help determine expected network traffic that each segment may need to support.

Establish a baseline, and study the trends This should go without saying, but before you can even think about estimating future network traffic growth, you need to establish a baseline of network usage. As your network grows, you should use that baselines, and others collected at intervals, to determine changes in usage from the baseline(s) and identify growth trends.

Understand specific business needs and the expected workload for the estimation period Although understanding the technical aspects is a key to understanding design of the network, it's important to keep in mind the activity trends and growth projections for the enterprise. Review these business plans, and then estimate future usage and determine the network configuration that is required to support the plans. You should also make sure you confirm the plans and gather statistics periodically so that you can detect new trends early and adjust your estimates.

Analyzing CPU Requirements

The CPU is the heart of a computer and the heart of your database server infrastructure. You'll now learn about what you need to take into account when analyzing the CPU performance of

a database server and when choosing a processor type. We'll also look at what you should do to make meaningful estimates of future CPU requirements.

We'll review considerations for choosing a CPU, such as performance versus cost benefit of using processors with 32-bit and 64-bit architectures, and of using processors with multicore and hyperthreading technologies, later in this chapter.

Assessing Current CPU Performance

When you're analyzing the current CPU performance of a database server, consider the following factors:

Type of CPUs Identify the database servers in the system. For each, make a list of its current CPU, speed, architecture (32-bit or 64-bit), and whether the processor is multicore or capable of hyperthreading.

Affinity mask settings By default, each thread allocated by a SQL Server instance is scheduled to use the next available CPU. However, you can set the *affinity mask* to restrict an instance to a specific subset of CPUs. Additionally, setting the affinity mask ensures that each thread always uses the same CPU between interrupts. This reduces the swapping of a thread among multiple CPUs and increases the cache hit ratio on the second-level cache.

Dynamic affinity is tightly constrained by per-CPU licensing. When you set the affinity mask, SQL Server verifies that the settings don't violate the licensing policy.

Current CPU usage To identify any CPU performance issues, you should set a baseline of CPU usage in the current environment. To do so, first collect basic operations data, such as the number of user connections and the amount of application data. Next, establish the current CPU usage using monitoring tools such as System Monitor. Finally, correlate the operations data with the CPU usage.

Hardware bottlenecks, recompilation of stored procedures, and the use of cursors are some of the main causes of a decrease in CPU performance. To identify CPU performance problems, use the counters that are included in System Monitor's SQLServer:Plan Cache and SQLServer:SQL Statistics objects.

Forecasting and Planning CPU Requirements

Once you have a good understanding of the current CPU situation in your environment, you should next assess and forecast future CPU needs in order to ensure the efficient and effective operation of your database server infrastructure.

There are always a variety of indicators and factors to review, but the following are critical:

Determine the estimation period By now you may be tired of looking at this consideration, but you must keep this in mind in order to limit the scope of your forecast activities to a realistic level.

Establish a baseline of CPU usage by using historical data When analyzing data on CPU usage, consider the type of work the SQL Server instance performs.

Identify factors that affect CPU usage These factors obviously will include the number of users and the amount of application data on the servers. You should also observe variations over time and try to find a correlation with measurable variables.

Confirm your estimates by performing load tests and by using sizing tools Keep in mind that adding CPUs to a server doesn't necessarily increase the overall CPU power in a linear proportion.

Analyzing Memory Requirements

If the CPU is the heart of the computer, then memory is a combination of muscle, sinew, and brainpower. As a general rule, when assessing the memory requirements of a database server, you need to determine the amount of memory being used by the OS, other processes on the server, and SQL Server. It's also important to bear in mind that memory usage is affected by the type of CPU. In the following sections, you'll learn about determining the current memory usage of a database server, the interaction between memory usage and CPU type, and how to estimate future memory requirements.

Assessing Current Memory Usage

Assessing current memory usage isn't that difficult thanks to a number of tools, the most important of which is System Monitor. To determine current memory usage and assess the ability to satisfactorily meet the needs of the current environment, do the following:

- Establish how much physical memory is installed on the database server.
- In addition to the OS and SQL Server 2005, determine what other processes will be making use of the available memory.
- Use these System Monitor counters to determine how much memory is available and used on a server:
  - Memory:Available Bytes indicates how many bytes of memory are available.
  - Memory:Pages/sec specifies how many pages must be read from the disk or written to the disk to resolve page faults.
  - SQLServer:Memory Manager:Total Server Memory determines the amount of physical memory used by each instance of SQL Server.
  - Process:Working Set describes the set of memory pages that have been recently accessed by the threads running in the process and can be used to determine how much memory SQL Server is using.
  - SQLServer:Buffer Manager specifies the buffer cache-hit ratio. This counter identifies the percentage of pages that were found in the buffer pool without reading the disk.

The value of this counter should be over 90 percent. High values indicate good cache usage and minimal disk access when searching for data.

  - SQLServer:Buffer Manager:Page Life Expectancy specifies the average time spent by a data page in the data cache. A value of less than 300 seconds indicates that SQL Server needs more memory.

- In addition to the System Monitor tool, you can use the following dynamic management views to collect data about SQL Server memory:
  - sys.dm_exec_query_stats provides statistics on memory and CPU usage for a specific query.
  - sys.dm_exec_cached_plans returns a list of the query plans that are currently cached in memory.
  - sys.dm_os_memory_objects provides information about object types in memory, such as `MEMOBJ_COMPILE_ADHOC` and `MEMOBJ_STATEMENT`.
  - sys.dm_os_memory_clerks returns the set of all memory clerks that are currently active in the instance of SQL Server.

When you're trying to establish actual memory usage and peak usage, make sure you collect information over a complete business cycle in order to obtain the most accurate data. For example, if an organization generates a large number of reports the first week of each month, collect the peak usage data when those reports are generated.

- Determine whether the current database and memory size match is correct. If the database, including its indexes, fits completely into the available memory, there will be no page faults. When a large database can't fit in memory, some data must be retrieved from the disk when required. Page faults can be minimized by using the buffer cache efficiently.
- Determine the amount of memory being used by SQL Server connections.

In Exercise 1.4, you'll use System Monitor to assess memory requirements.

EXERCISE 1.4

Using System Monitor to Assess Memory Requirements

To complete this exercise, you must be running a version of Windows that supports SQL Server 2005. You must also have sufficient privileges to access the System Monitor tool.

1. Click Start, select Run, and type **perfmon** in the text box.
2. System Monitor opens. By default, it displays a line graph of Pages/sec, Avg. Disk Queue Length, and % Processor Time.

EXERCISE 1.4 *(continued)*

3. To assess the current memory throughput, you should select relevant counters and log the result. In this exercise, using the procedures you learned in Exercise 1.1, use System Monitor to log the following counters for each instance of SQL Server in a file named CurrentMemoryUsage:
   - Memory:Available Bytes
   - Memory:Pages/sec
   - SQLServer:Memory Manager:Total Server Memory
   - Process:Working Set
   - SQLServer:Buffer Manager:Buffer Cache Hit Ratio
   - SQLServer:Buffer Manager:Page Life Expectancy
4. Allow logging to run for five minutes.
5. Explain the meaning of each counter and how the counters relate to describing the system's current needs. Is there a difference between instances?

The point of all this collecting and reviewing is to identify trouble spots that need to be addressed in the current configuration or that will play a role in modification and future growth of the infrastructure.

Consequently, you should track memory usage values regularly and establish a baseline. To gather data for establishing the baseline, you can use the System Monitor counters for SQL Server memory usage. When they're present, you can also use management tools such as Microsoft Operations Manager (MOM) to gather data on memory usage across a set of enterprise database servers.

You should also establish minimum and maximum usage values. Using these data ensures that the memory usage for the current period doesn't exceed the established limits. If you compare current memory usage values with the baseline, you can assess whether SQL Server has sufficient memory for normal operations. If memory is insufficient, the database server is said to be under *memory pressure*, a circumstance that needs to be addressed sooner rather than later.

Forecasting and Planning Memory Requirements

Once you have a good understanding of the current memory requirements, you should next estimate future memory needs. In order to do so, establish the following information:

Determine the number of SQL Server instances Ensure that the server has the optimal number of SQL Server instances. (For more information on instances, see Chapter 8.) Running too many instances can cause memory pressure. In some cases, such as in a multiple-instance cluster during a failover, multiple instances may be required.

Estimate database growth Because the memory needed by a database may grow if the size of the database and the volume of data that is queried increases, adding new databases to the same SQL Server instance may also create memory pressure.

Specify the number of concurrent users An increase in the number of user connections may result in a wider range of queries with varied parameters. This increases the pressure on memory because more data is cached in response to the queries.

Use baseline data Once you have the baseline data, you can collect current usage information and then compare it to historical data to identify growth trends.

Determine the rate of growth in memory usage By correlating the rate of growth in memory usage with a measurable variable, such as user connections, you can estimate future memory requirements. Let's say that you determine the data cache increases by 50 percent for every 500 users. Using this correlation, you can estimate the additional memory requirements of the server if the number of users increases by 50.

Specifying Software Versions and Hardware Configurations

In this section of the chapter, we won't specify every single aspect of hardware and software selection that you should apply. Given the rate of hardware and software change, any specific recommendations would likely be out of date by the time this book reached the shelves.

However, you're going to need to make these decisions every step of the way. As we've said, there are no hard and fast rules, but there are best practices. Apply them, and you can't go wrong.

Following Best Practices

You should apply the following best practices when selecting database server hardware and software:

Meet or exceed the design requirements Based on your assessment, the safest and most effective course is to at least meet design requirements not only for the present but also for the future. Be wary of technology obsolescence. Often, the best way to do that is to deliberately select hardware and software that exceed the requirements.

Perform cost-benefit analyses When choosing hardware or software, always perform a cost-benefit analysis. The purpose is to ensure that the benefit you'll obtain from the new item is justifiable in terms of increased throughput that offsets the cost. Spending money on a new high-speed memory bus may be warranted on a server with a heavy workload because the new hardware will increase the server's performance; but it wouldn't be appropriate for a small server with little workload, such as a server used to store archival data.

Choose from approved hardware and software configurations Most organizations have a list of approved hardware and software products and configurations that restricts your choices. One benefit is that the standardization of hardware and software reduces the complexity of the database server infrastructure and thereby simplifies maintenance and reduces implementation costs. Therefore, you should try to make selections that are within the framework of the hardware and software standards established by your organization. Keep in mind that it's better to choose database-server hardware and software products that have already been successfully used in a given environment.

Be prepared to justify variations from standards Although it's better to stay with a standardized setup, it isn't uncommon for standards to lag behind improvements and upgrades in database server hardware and software products. Consequently, you may need to make hardware and software choices that vary from the standards.

Design requirements should supersede standards, but these variations need to be approved by management. The best way to justify a variation is to clearly demonstrate that the existing standards don't allow you to meet specific design requirements, and to describe how the hardware and software products you've proposed meet those requirements.

Bear in mind that just because the variation may be justified doesn't always mean a variation will be approved; variations from standards are frequently rejected, usually for budgetary reasons. If that occurs, you need to identify alternative hardware and software products that come closest to fulfilling the requirements.

Choosing a Version and Edition of the Operating System

The version and editions of the operating system you use, if not predetermined by organizational standards, depend on the version of SQL Server 2005 you select. Table 1.1 specifies the minimum versions and editions of the operating system required for each edition of SQL Server 2005.

TABLE 1.1 SQL Server 2005 Editions and Minimum Operating System Versions and Editions

Edition	Operating System Version and Edition
Enterprise 64-bit	Windows XP Professional 64 or later; Windows 2003 Server: Standard x64, Enterprise x64, or Datacenter x64 edition with Service Pack 1 or later
Enterprise 32-bit	Windows XP with Service Pack 2 or later; Windows 2000 Server with Service Pack 4 or later; Windows 2003 Server: Standard, Enterprise, or Datacenter edition with Service Pack 1 or later; Windows Small Business Server 2003 with Service Pack 1 or later; Windows 2000 Professional with Service Pack 4 or later

TABLE 1.1 SQL Server 2005 Editions and Minimum Operating System Versions and Editions *(continued)*

Edition	Operating System Version and Edition
Enterprise Itanium	Microsoft Windows Server 2003 Enterprise Edition or DataCenter edition for Itanium-based systems with SP 1 or later
Standard 64-bit	Windows XP Professional 64 or later; Windows 2003 Server: Standard x64, Enterprise x64, or Datacenter x64 edition with Service Pack 1 or later
Standard 32-bit	Windows XP with Service Pack 2 or later; Windows 2000 Server with Service Pack 4 or later; Windows 2003 Server: Standard, Enterprise, or Datacenter edition with Service Pack 1 or later; Windows Small Business Server 2003 with Service Pack 1 or later; Windows 2000 Professional with Service Pack 4 or later
Standard Itanium	Microsoft Windows Server 2003 Enterprise Edition or Datacenter edition for Itanium-based systems with SP 1 or later
Workgroup	Windows XP with Service Pack 2 or later; Windows 2000 Server with Service Pack 4 or later; Windows 2003 Server: Standard, Enterprise, or Datacenter edition with Service Pack 1 or later; Windows Small Business Server 2003 with Service Pack 1 or later; Windows 2000 Professional with Service Pack 4 or later
Developer 64-bit	Windows XP Professional 64 or later; Windows 2003 Server: Standard x64, Enterprise x64, or Datacenter x64 edition with Service Pack 1 or later
Developer 32-bit	Windows XP with Service Pack 2 or later; Windows 2000 Server with Service Pack 4 or later; Windows 2003 Server: Standard, Enterprise, or Datacenter edition with Service Pack 1 or later; Windows Small Business Server 2003 with Service Pack 1 or later; Windows 2000 Professional with Service Pack 4 or later
Developer Itanium	Microsoft Windows Server 2003 Enterprise Edition or Datacenter edition for Itanium-based systems with SP 1 or later
Express	Windows XP with Service Pack 2 or later; Windows 2000 Server with Service Pack 4 or later; Windows 2003 Server: Standard, Enterprise or Datacenter edition with Service Pack 1 or later; Windows Small Business Server 2003 with Service Pack 1 or later

Choosing an Edition of SQL Server 2005

Because SQL Server 2005 is used by a vast audience of different people, businesses, school, government agencies, and so on, each of whom has different needs as well as diverse requirements, SQL Server 2005 comes in different editions. Each targets a group of people based on creating a good match to the unique performance, runtime, and price requirements of organizations and individuals.

There are five different editions of SQL Server 2005: Microsoft SQL Server 2005 Enterprise Edition, Microsoft SQL Server 2005 Standard Edition, Microsoft SQL Server 2005 Workgroup Edition, SQL Server 2005 Developer Edition, and SQL Server 2005 Express Edition. The most common editions used are Enterprise, Standard, and Workgroup, because only these can be installed and used in production server environments:

SQL Server 2005 Enterprise Edition (32-bit and 64-bit) This edition comes in 32- and 64-bit varieties. This is the ideal choice if you need a SQL Server 2005 edition that can scale to limitless size while supporting enterprise-sized On-Line Transaction Processing (OLTP), highly complex data analysis, data-warehousing systems, and websites.

Enterprise Edition has all the bells and whistles and is suited to provide comprehensive business intelligence and analytics capabilities. It includes high-availability features such as failover clustering and database mirroring. It's ideal for large organizations or situations with the need for a SQL Server 2005 that can handle complex situations.

SQL Server 2005 Standard Edition (32-bit and 64-bit) Standard Edition includes the essential functionality needed for e-commerce, data warehousing, and line-of-business solutions without some advanced features such as Advanced Data Transforms, Data-Driven Subscriptions, and DataFlow Integration using Integration Services. Standard Edition is best suited for the small- to medium-sized organization that needs a complete data-management and analysis platform without many of the advanced features found in the Enterprise Edition.

SQL Server 2005 Workgroup Edition (32-bit only) Workgroup Edition is the data-management solution for small organizations that need a database with no limits on size or number of users. It includes only the core database features of the product line (it doesn't include Analysis Services or Integration Services, for example). It's intended as an entry-level, easy-to-manage database.

SQL Server 2005 Developer Edition (32-bit and 64-bit) Developer Edition has all the features of Enterprise Edition, but it's licensed only for use as a development and test system, not as a production server. This edition is good choice for persons or organizations that build and test applications but don't want to pay for Enterprise Edition.

SQL Server 2005 Express Edition (32-bit only) SQL Server Express is a free, easy-to-use, simple-to-manage database without many of the features of other editions (including Notification Services, Analysis Service, Integrations Services, and Report Builder). SQL Server Express is free and can functions as the client database as well as a basic server database. It's a good option if all that's needed is a stripped-down version of SQL Server 2005, typically among low-end server users, non-professional developers building Web applications, and hobbyists building client applications.

Choosing a CPU Type

SQL Server 2005 supports both 32-bit and 64-bit CPUs. In addition, it supports multicore CPUs and CPUs that use hyperthreading. When estimating CPU requirements, you should consider the benefits of using different processor types.

The benefits of using a 64-bit CPU instead of a 32-bit CPU include the following:

- Larger direct-addressable memory. A database server running Microsoft Windows Server 2003 on a 64-bit architecture can support up to 1,024 terabytes of physical memory and 512GB of addressable memory. In contrast, a Windows server with a 32-bit architecture can directly address a maximum of 3GB of physical memory. The server can indirectly address memory beyond this limit if you enable the Address Windowing Extensions (AWE) switch.
- Better on-chip cache management. A 64-bit CPU allows SQL Server memory structures such as the query cache, connection pool, and lock manager to use all available memory. A 32-bit CPU doesn't.
- Enhanced on-processor parallelism. The 64-bit architecture can support 64 processors, allowing SQL Server to support more concurrent processes, applications, and users on a single server. A 32-bit architecture can support only 32 processors.

A *multicore* CPU includes two or more complete execution cores that run at the same frequency and share the same packaging and interface with the chipset and the memory. In addition, the cores contain two or more physical processors and two or more L2 cache blocks. On a Windows system running SQL Server, each core can be used as an independent processor to increase the multithreaded throughput.

Hyperthreading lets a CPU to execute multiple threads simultaneously. Consequently, the CPU throughput increases. A CPU that supports hyperthreading contains two architectural states on a single physical core. Each state acts as a logical CPU for the operating system. However, the two logical CPUs use the same execution resources, so you don't get the performance benefits of using two physical CPUs.

Currently, 64-bit, multicore, and hyperthreading CPUs are more expensive than 32-bit CPUs. You should assess the tradeoff between the higher cost for a CPU when compared to the benefits of newer technology and increase in performance.

SQL Server requires a variety of different CPUs depending on edition, as summarized in Table 1.2.

TABLE 1.2 SQL Server 2005 Editions and Minimum CPU Type and Speed

Edition	Minimum CPU Type and Speed
Enterprise 64-bit	1 Gigahertz (GHz) AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support processor
Enterprise 32-bit	32-bit, 600 megahertz (MHz) Pentium III-compatible or faster processor; 1GHz or faster processor recommended

TABLE 1.2 SQL Server 2005 Editions and Minimum CPU Type and Speed *(continued)*

Edition	Minimum CPU Type and Speed
Enterprise Itanium	Itanium processor: 1GHz or faster processor
Standard 64-bit	1GHz AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support processor
Standard 32-bit	32-bit, 600MHz Pentium III-compatible or faster processor; 1GHz or faster processor recommended
Standard Itanium	1GHz or faster processor
Workgroup	32-bit, 600MHz Pentium III-compatible or faster processor; 1GHz or faster processor recommended
Developer 64-bit	1GHz AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support processor
Developer 32-bit	32-bit, 600MHz Pentium III-compatible or faster processor; 1GHz or faster processor recommended
Developer Itanium	1GHz or faster processor
Express	32-bit, 600MHz Pentium III-compatible or faster processor; 1GHz or faster processor recommended

Choosing Memory Options

This may sound simplistic, but the best memory option still follows the oldest rule of thumb: Buy as much RAM as you can and the fastest possible RAM you that is appropriate for the system you're installing it on.

Increasing memory often solves what may initially appear to be a CPU bottleneck. Minimum and recommended RAM requirement for the different editions of SQL Server 2005 are presented in Table 1.3.

TABLE 1.3 SQL Server 2005 Editions and Minimum RAM

Edition	Minimum and Recommended RAM
Enterprise 32-bit and 64-bit and Itanium	512MB of RAM or more; 1GB or more recommended
Standard 32-bit and 64-bit and Itanium	512MB of RAM or more; 1GB or more recommended

TABLE 1.3 SQL Server 2005 Editions and Minimum RAM *(continued)*

Edition	Minimum and Recommended RAM
Workgroup	512MB of RAM or more; 1GB or more recommended
Developer 32-bit and 64-bit and Itanium	512MB of RAM or more; 1GB or more recommended
Express	192MB of RAM or more; 512MB or more recommended

Choosing a Type of Storage

As with memory options, the best rule of thumb is to buy as much hard disk space as you can afford as fast as you can. We'll discuss physical storage in more detail in the next chapter. See Table 1.4 for the minimum recommendations.

TABLE 1.4 SQL Server 2005 Editions and Hard Disk Space

Edition	Hard Disk Space
Enterprise 32-bit and 64-bit and Itanium	350MB of available hard disk space for the recommended installation with approximately 425MB of additional space for SQL Server Books OnLine, SQL Server Mobile Books Online, and sample databases
Standard 32-bit and 64-bit and Itanium	350MB of available hard disk space for the recommended installation with approximately 425MB of additional space for SQL Server Books OnLine, SQL Server Mobile Books Online, and sample databases
Workgroup	350MB of available hard disk space for the recommended installation with approximately 425MB of additional space for SQL Server Books OnLine, SQL Server Mobile Books Online, and sample databases
Developer, 32-bit and 64-bit and Itanium	350MB of available hard disk space for the recommended installation with approximately 425MB of additional space for SQL Server Books OnLine, SQL Server Mobile Books Online, and sample databases
Express	350MB of available hard disk space for the recommended installation with approximately 425MB of additional space for SQL Server Books OnLine; SQL Server Mobile Books Online and sample databases

Case Study: Assessing a Database Infrastructure

Thylacine Savings & Loan Association is a large financial institution serving approximately 1.6 million customers over a broad geographic area. The company is headquartered in the city of Trevallyn, which also serves as northern headquarters, with 407 employees. Three branch offices are located in Stratford (Eastern operations), Belleville (Western), and Rock Hill (Southern).

The company currently has a 3 terabyte OLTP database that tracks more than 2 billion transactions each year. The main database for all transactions and operations is located in Trevallyn. Regional databases contain deposit/withdrawal information only, and the headquarters database is updated daily from the regional offices.

Thylacine's departmental database servers are dispersed throughout the headquarters location.

The company is currently experiencing 4 percent annual growth and plans to expand into four new markets at the rate of one new market every two years. The database is growing at a rate of 6 percent per year and will exceed available hard disk space in the future. Additionally, server capacity is overloaded, resulting in poor performance and long delays. A large portion of the database data is historical information.

After lengthy consideration, Thylacine Savings & Loan has decided to upgrade its database system to SQL Server 2005 and has hired you as a consultant database project architect to address the company's current and future needs.

1. Briefly summarize the initial steps you should take before beginning capacity planning.
2. Do you need to consider regulatory factors? If so, describe the impact they're likely to have on the various components of the infrastructure's capacity.
3. Which would you give greater weight: the observed growth rate of a database or the projected business growth rate of Thylacine Savings & Loan? Why?

Summary

In this chapter, we reviewed the factors you need to consider when you're assessing the capacity requirements of a database server. We reviewed a variety of methods for collecting information about current capacity and how to forecast future needs.

In addition, you familiarized yourself with the techniques and skills you'll need to achieve a balance between business and technical requirements. Finally, you learned about the various hardware and software considerations you need to factor in to your design plans, including hardware, operating system, and software versions.

Exam Essentials

Be familiar with System Monitor counters. Know how to use the System Monitor tool and which counters provide relevant information about system status. Know the techniques of collecting a baseline, and when and how to use it.

Understand your business requirements. Know your different business requirements and the subsystems they impact. Make sure you understand the effect of regulatory requirements on your storage needs.

Know the prerequisites. Know the prerequisites for installing the various editions of SQL Server 2005, including what operating system, how much memory, and how fast a CPU you need.

Understand the cost-benefit relationship between 32-bit and 64-bit processors. Be familiar with the advantages and disadvantages of the two processor types.

Review Questions

1. Which of the following factors should be considered when projecting disk storage requirements?
   - **A.** Forecasted business growth
   - **B.** Historical trends
   - **C.** Index maintenance space requirements
   - **D.** All of the above
2. Which of the following file types should not be considered when determining the amount of disk space used by the database files?
   - **A.** Database files
   - **B.** Database paging file
   - **C.** Database transaction logs
   - **D.** Full-text indexes
3. What can result if improper disk-space allocation causes SQL Server 2005 to dynamically grow the database by requesting extra disk space from the operating system? (Choose all that apply.)
   - **A.** Truncating of log files
   - **B.** Reduced network bandwidth
   - **C.** Disk/file fragmentation
   - **D.** Processor bottleneck
4. Which of the following are System Monitor counters that can be used to assess disk I/O rates? (Choose all that apply.)
   - **A.** PhysicalDisk:Disk Read Bytes/sec
   - **B.** PhysicalDisk:Disk Write Bytes/sec
   - **C.** PhysicalDisk:Avg. Disk Queue Length
   - **D.** PhysicalDisk:Disk Modify Bytes/sec
   - **E.** All of the above
5. The length of time data must be retained is also referred to as what?
   - **A.** Lifetime of data
   - **B.** Data Retention Period (DRP)
   - **C.** Data estimation period
   - **D.** Longevity of data

6. In order to start the System Monitor tool, you should type which of the following commands in the Run text box?
   - **A.** `perfmon`
   - **B.** `sysmon`
   - **C.** `sysinfo`
   - **D.** `mssysmon`
   - **E.** `mmc`

7. If regulatory requirements or internal procedures require the encryption of data, which subsystems are directly impacted? (Choose all that apply.)
   - **A.** Physical storage
   - **B.** Memory
   - **C.** CPU
   - **D.** Network
   - **E.** SQL Server version

8. If you calculate future disk-space requirements based on a constant amount in a specified period, you are calculating what?
   - **A.** Linear growth
   - **B.** Compound growth
   - **C.** Trigonometric growth
   - **D.** Geometric growth
   - **E.** Incremental growth

9. Which of the following will not affect the CPU performance of a database server?
   - **A.** Affinity mask settings
   - **B.** Number of connections
   - **C.** Available memory
   - **D.** Network bandwidth
   - **E.** Processor type

10. Affinity masks can be used to do what? (Choose all that apply.)
    - **A.** Change the bit speed of a CPU.
    - **B.** Restrict a SQL Server instance to a specific subset of CPUs.
    - **C.** Ensure that each thread always uses the same CPU between interrupts.
    - **D.** Free up RAM that was locked earlier.
    - **E.** Restrict CPU operation to specific file types.

11. Which of the following is not a benefit of using a 64-bit CPU?

A. Larger direct-addressable memory

B. Better on-chip cache management

C. Lower cost per chip

D. Enhanced on-processor parallelism

E. None of the above

12. Characteristics of a multicore CPU include which of the following? (Choose all that apply.)

A. Executes multiple cores simultaneously

B. Includes two or more completion execution cores

C. All cores run at different frequencies

D. Can increase multithreaded throughput in SQL Server 2005

E. Contains two architectural states on each core

13. Which of the following are the most important factors in estimating CPU requirements? (Choose all that apply.)

A. Establishing a baseline

B. Business plans

C. Historical trends

D. Correlation between CPU usage and a measurable variable

E. Longevity of data

14. Which of the following counters indicates how many bytes of memory are available?

A. Memory:Available Bytes

B. Memory:Pages/sec

C. Memory:Available RAM

D. Memory:Available Pages

E. Memory:Bytes/sec

15. Which of the following dynamic management views can be used to gather data about memory usage by SQL Server? (Choose all that apply.)

A. sys.dm_exec_query_stats

B. sys.dm_exec_cached_plans

C. sys.dm_os_memory_pages

D. sys.dm_os_memory_objects

E. sys.dm_exec_query_calls

16. Which one of the following counters is used to determine the amount of memory used by SQL Server connections?
    - **A.** SQLServer:MemoryManager:Total Server Memory
    - **B.** SQLServer:MemoryManager:Working Set
    - **C.** SQLServer:Buffer Manager:Connection Memory (KB)
    - **D.** SQLServer:Page Manager:Connection Memory (KB)
    - **E.** SQLServer:Memory Manager:Connection Memory (KB)

17. Which of the following may affect network traffic?
    - **A.** Backup schedules
    - **B.** Firewalls
    - **C.** Antivirus applications
    - **D.** Enabled network protocols
    - **E.** All of the above

18. Which of the following business requirements should be considered when modifying or designing a database infrastructure?
    - **A.** Budgetary constraints
    - **B.** IT policies
    - **C.** Data security
    - **D.** Data availability
    - **E.** All of the above

19. During your survey, you determine that one of the existing database servers has an 800MHz Pentium III processor with 256MB of RAM and a 400GB hard drive, running Windows XP2. Which version of SQL Server can you install on this machine?
    - **A.** Workgroup
    - **B.** Standard
    - **C.** Enterprise
    - **D.** Developer
    - **E.** None of the above

20. You want to install SQL Server 2005 Enterprise Edition on a 32-bit CPU machine. Which operating systems can this machine use? (Choose all that apply.)
    - **A.** Windows 2003, Service Pack 1
    - **B.** Windows 2000 Professional, Service Pack 3
    - **C.** Windows XP, Service Pack 1
    - **D.** Windows 2000 Server, Service Pack 5
    - **E.** All of the above

Answers to Review Questions

1. D. All of these need to be assessed and considered.
2. B. There is no such thing as a database paging file. This question emphasizes the many factors you need to consider.
3. C. Only C occurs. Neither B nor D is relevant. Although A *might* occur in some circumstances, it has nothing to do with SQL Server 2005 growing dynamically.
4. A, B, C. The counter PhysicalDisk:Disk Modify Bytes/sec does not exist.
5. D. Only D is correct. The other answers are distracters.
6. A. Only `perfmon` is correct. The rest of the answers are distracters or fictitious.
7. A, C. Regulatory requirements typically affect the length of retention of data and whether it should be encrypted. This impacts the CPU because more processing is required for encrypted data. Physical storage is affected because encrypted files take up more space than unencrypted files.
8. A. Straight-line projections are linear. B and D are other types of growth projections, whereas C and E are fictitious names.
9. D. Network bandwidth has no direct or indirect impact on CPU performance, but all the other listed items do.
10. B, C. None of the other answers is in the least related to affinity masks.
11. C. 64-bit CPUs cost more than 32-bit CPUs, although there may be a performance trade-off in some instance that makes spending more money on a chip cheaper in the long run.
12. B, D. A is incorrect because there is no such capability. C is incorrect because the cores run at the same frequency. E refers to a hyperthreaded CPU. B and D are correct answers.
13. A, B, C, D. Establishing a baseline is critical as well as finding any correlations that exist to use as a predictor. Historical trends are more likely to be valuable than business forecasts, but both must be considered. Although longevity of data will impact data-storage issues, it is not relevant to CPU requirements.
14. A. C, D, and E refer to counters that don't exist. The counter Memory:Pages/sec indicates how many pages need to be read from or written to the disk to resolve hard page faults.
15. A, B, D. C and E refer to nonexistent dynamic management views and are therefore incorrect.
16. E. Only E is correct. A is used to collect information on the amount of physical memory used by each instance of SQL Server. The remaining counters do not exist.
17. E. All of these items impact directly on network traffic.
18. E. In addition, you should consider regulatory requirement, be they external or internal in origin.

19. E. None of these editions can run on the machine, because they all require a minimum of 512MB of RAM. You could install SQL Server Express Edition, which requires a minimum of 192MB RAM.

20. A, D. The minimum upgrade for Windows 2000 is Service Pack 4; for Windows XP, Service Pack 2 is required. The required operating systems are summarized in Table 1.1.