In the first chapter, I stressed the importance of having a performance baseline that you can use to measure performance changes. In fact, this is one of the first things you should do when starting the performance tuning process, since without a baseline you will not be able to quantify improvements. In this chapter, you will learn how to use the Performance Monitor tool to accomplish this and how to use the different performance counters that are required to create a baseline. Other tools necessary for establishing baseline performance metrics for the system will also be addressed when they can help you above and beyond what the Performance Monitor tool can do.

Specifically, I cover the following topics:

- The basics of the Performance Monitor tool
- How to analyze hardware resource bottlenecks using Performance Monitor
- How to retrieve Performance Monitor data within SQL Server using dynamic management views
- How to resolve hardware resource bottlenecks
- How to analyze the overall performance of SQL Server
- Considerations for monitoring virtual machines
- How to create a baseline for the system

Performance Monitor Tool

Windows Server 2008 provides a tool called Performance Monitor. Performance Monitor collects detailed information about the utilization of operating system resources. It allows you to track nearly every aspect of system performance, including memory, disk, processor, and the network. In addition, SQL Server 2012 provides extensions to the Performance Monitor tool to track a variety of functional areas within SQL Server.

Performance Monitor tracks resource behavior by capturing performance data generated by hardware and software components of the system, such as a processor, a process, a thread, and so on. The performance data generated by a system component is represented by a performance object. A performance object provides counters that represent specific aspects of a component, such as % Processor Time for a Processor object. Just remember, when running these counters within a virtual machine (VM), the performance measured for the counters in most instances is for the VM, not the physical server.
There can be multiple instances of a system component. For instance, the Processor object in a computer with two processors will have two instances represented as instances 0 and 1. Performance objects with multiple instances may also have an instance called Total to represent the total value for all the instances. For example, the processor usage of a computer with four processors can be determined using the following performance object, counter, and instance (as shown in Figure 2-1):

- **Performance object:** Processor
- **Counter:** % Processor Time
- **Instance:** _Total

System behavior can be either tracked in real time in the form of graphs or captured as a log (called a data collector set) for offline analysis. The preferred mechanism on production servers is to use the log.

To run the Performance Monitor tool, execute perfmon from a command prompt, which will open the Performance Monitor suite. You can also right-click the Computer icon on the desktop or the Start menu, expand Diagnostics, and then expand the Performance Monitor. Both will allow you to open the Performance Monitor utility.

You will learn how to set up the individual counters in the “Creating a Baseline” section later in this chapter. First, let’s examine which counters you should choose in order to identify system bottlenecks and how to resolve some of these bottlenecks.