Chapter 10

Introduction to Performance Monitoring and Tuning: Java and .NET

For any applications to be performance conscious, its performance must be monitored continuously. Monitoring performance is a necessary part of the preventive maintenance of the application. By monitoring, we obtain performance data which are useful in diagnosing performance problems under operational conditions. Based on data collected through monitoring, one can define a baseline — a range of measurements that represent acceptable performance under typical operating conditions. This baseline provides a reference point that makes it easier to spot problems when they occur. In addition, during troubleshooting system problems, performance data give information about the behavior of system resources at the time the problem occurs, which is useful in pinpointing the cause. In order to monitor the system, the operational environment provides various parameters implemented through counters for collection of performance data.

Applications developed must ultimately be installed and run on a specific operating system. Hence, applications performance also depends on factors that govern the operating system. Each operating system has its own set of performance parameters to monitor and tune for better performance. Performance of applications also depends on the architectural level monitoring and tuning. However, architectural design depends on specific technology. Hence, technology level monitoring and tuning must be addressed for better results. To achieve all these, proper guidelines must be enforced at various stages for monitoring and tuning.
All the previous chapters, together, described the performance testing from concept to reality whereas this chapter highlights aspects of monitoring and tuning to specific technologies. This chapter provides an overview of monitoring and tuning applications with frameworks in Java and Microsoft .NET technologies. Before addressing the technology specific performance issues, we need to know the overall bottlenecks that arise in Web applications.

Areas of Bottlenecks in Web-Based Applications

Web applications are developed in a multi-operating system (OS) environment. Server modules may run in Unix OS whereas client modules may run in Windows OS. The overall design of architecture includes Web server, application server, network environment, firewalls, and so on. When a Web server is installed on the system (Figure 10.1), there are many aspects that need to be addressed which can impact performance of the system by blocking the throughput. Obviously, the network is one of the factors but so are all the other servers that are being accessed. It is important to understand how the Web applications are developed, what optimization flags have been used, how many servers are in place as well as how performance is being measured. Several of the parameters chosen in the Web may cause performance issues. Causes could be because of server resources like CPU, memory, disk I/O, and network bandwidth. As shown in Figure 10.1, there could be performance bottlenecks at server side, introduction of firewall and routers, slowness of load balancers, and could be due to poor performance of database servers. Server configuration can have a serious impact on performance and may need to be reevaluated.

Other possible bottlenecks in a Web application could be in the different application layers like:

- Poor database design during the application development;
- Poor standards followed in table design;

Figure 10.1. Anticipated bottleneck in Web application system
E-mail Mining: Emerging Techniques for E-Mail Management
www.igi-global.com/chapter/mail-mining-emerging-techniques-mail/31103?camid=4v1a