CHAPTER 18

Tuning for Peak Load

By Ben DeBow

Today is historically the busiest day of the year for your online shopping business. At 11 AM, the system is starting to show signs of strain and load that exceeds that of the previous year. Customers are having problems completing their orders. Some fail, others are just slow. The operations center calls you in to see what’s going on with SQL Server, and you observe that processes are starting to queue up because of large blocking chains, which ultimately affects the checkout process. You didn’t code the application, but your job as a database administrator (DBA) is to fix the problem. As this event is unfolding, upper management is concerned and the Chief Information Officer (CIO) is standing over your shoulder watching you try to resolve the issue—no pressure there. Things eventually calm down later in the day, but it is unknown how many sales or potential customers were lost because of the slowdown or failed checkouts. In addition to the orders being affected, replication to the reporting environment now has 12 hours of latency affecting the reports to internal consumers of the raw sales data. Can you prevent this scenario from happening? In many cases, you can.

Every organization has mission-critical applications that run the business. These applications must scale and perform to handle load when the system is at its busiest—otherwise known as its peak load. Peak loads might occur daily or at varying intervals, often during the peak season; it’s not necessarily just a once-a-year occurrence such as the holiday season (which could be considered a peak season) described in the preceding example. To address these challenges, organizations must plan for these events well in advance so that they can have peace of mind that their systems will meet the increased traffic demands. Failure or downtime is often not an option because most businesses either depend on a large percentage of revenue from these peak times or have mission-critical business events, such as an accounting department’s year-end close.

The way your solution looks the day it is put in place is not the way it will look 10, 100, or 1,000 days later. More users are added and traffic increases. You need to anticipate the change not only in daily usage (which might represent your peak time), but also for those special times when system performance is crucial. This chapter will show you how to plan for the times when your servers will need to scale and perform to handle this additional traffic.

Define the Peak Load

Do you know how many transactions you expect during the peak hour of your peak day? If so, how accurate is that number? Are there documented performance requirements for the core transactions to

B. Ball et al., Pro SQL Server 2012 Practices
meet the service level agreements (SLA)? Have you measured how many transactions the current system can handle day to day? These and many other questions are important to answer before the business can feel comfortable about going into the peak season.

Every application and environment is different. However, there is one thing that is constant: no single action will make the system scale. A combination of different things must happen to achieve the desired result. For example, the business cannot assume that purchasing newer, bigger hardware will solve its performance woes and make the application scale. Think about your critical application and how a large blocking chain during the core business hours might affect the application and, ultimately, the business itself. Hardware will not account for, nor fix, blocked internal resources in SQL Server. Blocked resources is an application issue, even if it winds up being as simple as fixing an index or something as complex as rewriting application logic. The process to prepare the application for the peak load is a repetitive one that often must be repeated several times before the environment is ready to scale to meet the additional load.

Before you can do any analysis to possibly remediate the application to handle your peak load, you need to put a multifunctional project team in place. The project team should consist of various groups from across the enterprise, including the technical and nontechnical sides of the business.

Here is the makeup of a typical project team, listed by function:

- Project manager
- Several application representatives
- DBA lead
- Storage resource
- Network resource
- Server resource
- Business representative

The project team’s major responsibility is to clearly define the goals that must be met during the peak usage time (whether it’s an event, day-to-day use, seasonal use, and so forth) and to implement the changes necessary to enable the systems to meet or exceed these goals. These goals must be formally documented and agreed upon so that everyone is on the same page when it comes to expectations. If this is not done, you might wind up in the scenario that started the chapter.

Start by defining what is expected during the peak window. Depending on the environment and the company, this exercise might be easier for you than others. The expectation should always start as a nontechnical requirement from the business. Discussions to get to that expectation often involve looking at the current market conditions and reviewing a lot of data, both current and historical. The goal is to define how many business transactions the company expects to process within a certain period of time. The transactions represent the core processes the consumer or end user will use or initiate during the peak season. For example, if your company is an online retailer, the core transactions might be product searches, adding products to the shopping cart, the checkout process, processing payments, and reviewing orders. What you’re really trying to understand is this: when will the transactions occur, and what is the expected peak volume? How many orders does the company anticipate confirming within an hour or day? Is it 1,000 per hour or 500,000 per day? Does that volume change at certain times of the day, month, or year? Planning for 1,000 transactions per hour is very different than planning for a workload of 500,000 transactions. You need to design solutions to meet the maximum number of transactions during the peak time frame.

Applications are the key, so mapping something like X sales per hour to the number of transactions in the system to how many transactions need to occur in SQL Server is important. This gives the technical side of the team an idea of where to start and what to look for and architect to.