CHAPTER 15

Selecting and Sizing the Server

By Glenn Berry

With the release of SQL Server 2012, Microsoft has moved to a new licensing model that is quite different from previous releases. Because of this, it is quite important to revisit how you go about selecting database server hardware in order to get good performance and scalability while still keeping your SQL Server 2012 license costs under control. SQL Server 2012 Enterprise Edition uses core-based licensing, where you pay for licenses based on physical processor cores, with a minimum of four physical core licenses required for each physical processor. This is quite a change from the socket-based licensing used by SQL Server 2008 R2 and all previous releases of SQL Server.

This new model came as an unpleasant shock to many people when it was first revealed in November 2011, causing some database professionals to immediately assume a worst-case scenario of a fourfold price increase for SQL Server 2012 licenses. This chapter will show you how to properly select your database server hardware to maximize your performance and scalability, while keeping your SQL Server licensing costs under control.

Depending on your workload type, you will want to make different hardware choices to get the best performance and scalability for your workload.

Tip  Read the entire chapter before you actually spec out any hardware. The interaction between workloads, hardware, and SQL Server licensing is a complex subject. It is easy to make an expensive mistake if you don’t understand the concepts involved.

The hardware I look at in this chapter is specifically the server itself, meaning the form factor, and also the CPU and memory. Your CPU choice most strongly affects your licensing costs, and I spend a good deal of time looking at CPU choices with respect to SQL Server’s new licensing model. Microsoft’s new pricing scheme changes the game, and it’s important to revisit assumptions you might be holding that are no longer valid.

Understanding Your Workload

SQL Server places different demands on its underlying hardware depending on what type of database workload is running against the instance of SQL Server. Database engine workloads are quite different
than file server, web server, or application server workloads. SQL Server instances that are running the
database engine generally see one of two main types of workloads. One type is an online transaction
processing (OLTP) workload, while the other type is a relational data warehouse (DW) workload.

OLTP workloads are characterized by a high number of short-duration transactions and queries that
are usually executed on a single thread of execution. They can have a higher percentage of write activity,
and the data in some tables can be extremely volatile. These characteristics have important implications
for the hardware selection and configuration process. For example, you would want a processor with
excellent single-threaded performance, as most queries use only a single thread of execution. You would
also want an I/O subsystem that can supply a high number of input/output operations per second (IOPS),
and one that has excellent write performance because of the volatility of the data and the high transaction
log write activity.

DW workloads are characterized by longer running queries against more static data. These queries are
often parallelized by the query optimizer, so having a higher number of physical cores in your processors
can be very beneficial. Having a large amount of physical RAM is very useful for DW workloads, because
you will be able to have more data in the SQL Server buffer cache, which will reduce the read pressure on
the I/O subsystem. Outside of data loads, there tends to be very little write activity with a DW workload,
which means the I/O subsystem would be provisioned and configured differently than for a typical OLTP
workload. Many DW-type queries read large amounts of data as they calculate aggregates, so good
sequential read I/O performance is very important. There is much less write activity (outside of data
loads), which will also affect how you configure your I/O subsystem in terms of storage type and RAID
level.

You should try to determine what type of workload your server will be supporting before you decide
what type of hardware and storage subsystem to purchase. You also have to keep in mind that very few
database workloads are pure OLTP- or pure DW-type workloads, so you will often have to deal with mixed
workload types. You also might have to host multiple databases on a single SQL Server instance, where
each database has a different type of workload. Another possibility is multiple databases on a single SQL
Server instance where each database has the same type of workload, but the aggregate workload on the
server ends up looking quite different from what you might expect. Given all of this, how can you
determine your expected workload type?

If you have an existing server, running a production workload, you can run a number of Dynamic
Management View (DMV) and Dynamic Management Function (DMF) queries, and monitor some key
PerfMon counters to get a better idea about your existing workload type. For example, you can query the
sys.dm_os_wait_stats DMV to see your top cumulative wait types since the last time that SQL Server was
restarted (or the wait statistics were cleared) to get a better idea of your current bottlenecks.

For a new application that is under development, you can run these same queries and counters on
your development and test servers, along with reviewing documentation and talking to the developers,
business analysts, and end users to get a better idea about the expected workload type. The point of all this
detective work is to have a better idea of the type of workload your hardware and storage subsystem is
going to be dealing with, so you can make the best hardware selection and configuration decisions.

**SQL Server 2012 Enterprise Edition Consideration Factors**

One critical question that you must answer early on in the design process is whether you are going to be
able to use SQL Server 2012 Enterprise Edition, or whether you will be forced to use a lower-end edition,
such as SQL Server 2012 Standard Edition or SQL Server 2012 Business Intelligence Edition. Speaking as a
database professional, I strongly believe that Enterprise Edition is very much worth the extra license costs,
but I know that it can be a tough sell when you have a limited budget. Following is a listing of some of the
more valuable features that you get in SQL Server 2012 Enterprise Edition:

- Data compression