

Design Tip #102 Server Configuration Considerations

By Warren Thornthwaite

"How many servers do I need?" is a frequently asked question when technology is discussed in our Kimball University classes. The only right answer is the classic "it depends." While this may be true, it's more helpful to identify the factors upon which it depends. In this Design Tip, I briefly describe these factors, along with common box configurations.

Factors Influencing Server Configurations

Not surprisingly, the three major layers of the DW/BI system architecture (ETL, presentation server, and BI applications) are the primary drivers of scale. Any one of these may need more horsepower than usual depending on your circumstances. For example, your ETL system may be particularly complex with lots of data quality issues, surrogate key management, change data detection, data integration, low latency data requirements, narrow load windows, or even just large data volumes.

At the presentation server layer, large data volumes and query usage are the primary drivers of increased scale. This includes the creation and management of aggregates which can lead to a completely separate presentation server component like an OLAP server. At the query level, factors such as the number of concurrent queries and the query mix can have a big impact. If your typical workload includes complex queries that require leaf level detail, like COUNT(DISTINCT X), or full table scans such as the queries required to create a data mining case set, you will need much more horsepower.

The BI applications are also a major force in determining the scale of your system. In addition to the queries themselves, many BI applications require additional services including enterprise report execution and distribution, web and portal services, and operational BI. The service level requirements also have an impact on your server strategy. If it's OK to lock out user queries during your load window, you can apply most of your resources to the ETL system when it needs it, and then shift them over to the database and user queries once the load is finished.

Adding Capacity

There are two main approaches to scaling a system. Separating out the various components onto their own boxes is known as *scaling out*. Keeping the components on a single system and adding capacity via more CPUs, memory and disk is known as *scaling up*.

Start with the basic design principle that fewer boxes is better, as long as they meet the need from a data load, query performance and service level perspective. If you can keep your DW/BI system all on a single box, it is usually much easier to manage. Unfortunately, the components often don't get along. The first step in scaling is usually to split the core components out onto their own servers. Depending on where your bottlenecks are likely to be, this could mean adding a separate ETL server or BI application server, or both. Adding servers is usually cheaper up front than a single large server, but it requires more work to manage, and it is more difficult to reallocate resources as needs change. In the scale up scenario, it is possible to dedicate a portion of a large server to the ETL system, for example, and then reallocate that portion on the fly. At the high end, you can actually define independent servers that run on the same machine, essentially a combination of scaling up and scaling out.

Clustering and server farms offer ways to add more servers in support of a single component. These techniques usually become necessary only at the top end of the size range. Different database and

BI products approach this multi-server expansion in different ways, so the decision on when/if is product dependent.

Just to make it more interesting, your server strategy is also tightly integrated with your data storage strategy. The primary goal in designing the storage subsystem is to balance capacity from the disks through the controllers and interfaces to the CPUs in a way that eliminates any bottlenecks. You don't want your CPUs waiting idly for disk reads or writes.

Finally, once you get your production server strategy in place, you need to layer on the additional requirements for development and test environments. Development servers can be smaller scale, however, in the ideal world, your test environment should be exactly the same as your production system. We have heard from folks in the Kimball community who have had some success using virtual servers for functional testing, but not so much with performance testing.

Getting Help

Most of the major vendors have configuration tools and reference systems designed specifically for DW/BI systems based on the factors we described above. They also have technical folks who have experience in DW/BI system configuration and understand how to create a system that is balanced across the various factors. If you are buying a large system, they also offer test labs where they can set up full scale systems to test your own data and workloads for a few weeks. Search your vendor's website for "data warehouse reference configuration" as a starting point.

Conclusion

Hardware has advanced rapidly in the last decade to the point that many smaller DW/BI systems can purchase a server powerful enough to meet all their needs for relatively little money. The rest of you will still need to do some careful calculations and testing to make sure you build a system that will meet the business requirements.

This design tip focuses on the tangible decision of how much hardware to buy, and how to configure that hardware for the widely varying requirements of ETL, database querying and BI. This is an exciting time to make hardware decisions because of the explosive growth of server virtualization and the eventual promise of cloud computing. However, on reflection it is clear that neither of these approaches makes the problem of hardware configuration go away. DW/BI systems are so resource intensive, with highly specific and idiosyncratic demands for disk storage, CPU power, and communications bandwidth, that you cannot "virtualize it and forget it." Certainly, virtualization is appropriate in certain cases, such as flexing to meet increased demands for service. But those virtual servers still have to sit on real hardware that is configured with the right capacities.

Cloud computing is an exotic possibility that may one day change the DW/BI landscape. But we are still in the "wild ideas" stage of using cloud computing. Only the earliest adopters are experimenting with this new paradigm, and as an industry, we haven't learned how to leverage it yet. Thus far, a few examples of improved query performance on large databases have been demonstrated, but this is just a piece of the DW/BI pie.