SearchSQLServer

TechTarget

E-Guide

Best practices to maximize SQL Server performance

Every server's performance comes down to four basic components: memory, disk storage, network adapters and CPU. Your first step will be to maximize all four of these components in an existing server.

In this E-Guide, readers will learn insights into Solid-state storage devices and tips on how to effectively scale out a SQL Server environment across multiple systems.

Contents

Solid-state storage devices for SQL Server: Are they worth the cost?

By: Serdar Yegulalp, Contributor

Few people can deny the rising presence of solid-state drives (SSDs) in enterprise applications such as SQL Server. They have a few major advantages over their spinning-platter counterparts, namely, their increased read and random-access speeds. But given that conventional spinning-platter drives have been on the market for decades and have a great deal of proven technology behind them, is there a real incentive to push for a switch to solid-state storage devices for SQL Server -- especially given their cost?

SSDs have a number of attractive features that make them increasingly competitive against conventional disks. They consume little energy, they have fast random-access read modes, and they come in form factors (e.g., Serial Advanced Technology Attachment) that allow them to natively replace hard disks. For database administrators, SSDs' high read speeds are a major draw, since increasing those speeds can theoretically reduce a major I/O bottleneck.

But there are several valid reasons not to go with solid-state storage devices for SQL Server. The single biggest is their cost-effectiveness, whether or not they deliver better throughput for the dollar than conventional disks. When dealing with storage systems containing many disks—as you often do with databases -- it isn't just raw performance that matters but performance per

Contents

dollar. If you can solve most of your bandwidth problems with a broad array of cheap hard disk drives, go for it. With SSDs, you could be spending up to 10 times as much money, but unless you're getting 10 times better performance (and you typically don't), you're better off with hard disks.

A 2009 Microsoft Research paper, *Migrating Server Storage to SSDs: Analysis of Tradeoffs*, concluded that SSDs were not, at the time, a viable replacement for conventional hard drives in any of the server scenarios they tested. "The capacity/dollar of SSDs needs to improve by a factor of 3-3,000 for SSDs to be able to replace disks," the authors wrote. "The benefits of SSDs as an intermediate caching tier are also limited, and the cost of provisioning such a tier was justified for fewer than 10% of the examined workloads." SQL Server was not one of the workloads the authors tested explicitly, but they did test against a 5,000-user installation of Microsoft Exchange Server (which uses an embedded database) and didn't find the investment worthwhile.

One thing that should *not* be held against SSDs almost inevitably comes up in any discussion about their long-term use: that flash memory cells can withstand a limited number of write cycles. Users and IT administrators alike have been hyperconscious of this fact ever since flash drives came on the market. In a consumer setting, where the amount of I/O isn't as aggressive as in an enterprise environment, maybe write-cycle limit isn't such a big deal. But in an enterprise setting, especially for applications like databases, where reliability is crucial, people don't want to bank on a technology that might torch their data.

A closer look shows the "write endurance" problem is a lot worse on paper than in reality, and it has been mitigated to a great extent by good design. SSD market analyst Zsolt Kerekes did his own investigation of the issue and concluded, "In a well-designed flash SSD you would have to write to the whole disk the endurance number of cycles to be in danger." Even databases that sustain a great deal of writes don't pose a write-endurance problem to SSDs.

Contents

Given such a scenario, the write-endurance lifetime of the solid-state storage drive is many times longer than the likely deployment lifetime of the unit itself. In other words, you're far more likely to replace an SSD because a newer, larger, faster or more energy-efficient model of SSD comes on the market than because it runs out of write cycles.

And newer models are constantly arriving, although the prices have a long way to fall before they become cost-effective replacements for conventional drives. Consequently, if you're looking to spend the kind of money spent on flash SSD storage for a database system (easily on the order of thousands of dollars), you might be better off putting those resources toward other components in your database system. Increasing RAM, for instance, means less of the workload is I/O-bound, and may be a more cost-effective way to speed things up than dropping stacks of cash on SSDs. Your best bet is to use real-world statistics to find out how much of your database workload is irrevocably I/O-bound, and then determine if SSDs are worth the cost.

James Hamilton of the Data Center Futures team at Microsoft crunched some numbers on when SSDs make sense in server applications and produced a useful formula for figuring the cost-effectiveness of SSDs. His formula uses a database server (a "red-hot transaction processing system," in his words) as a test case for when SSDs might be justified. From what he's found, random I/O to and from disks have consistently lagged behind other kinds of I/O, so it's tempting to replace disks with solid-state storage devices on this note alone. But, again, there's how cost effective it is to do so, and if you gather real-world data from your own systems and do the numbers, you may find the costs don't justify the gains.

While SSDs are on the way to overtaking their spinning-disk counterparts in many environments, it's still hard to justify their use in a SQL Server environment from a cost perspective. That will change as the prices on SSDs come down, or your workloads change, or both. But before you drop the cash, do the math; for the time being, your money may be put to better use somewhere else.

Contents

ABOUT THE AUTHOR
Serdar Yegulalp *has been writing about computers and IT for more than 15 years for a variety of publications, including* InformationWeek and Windows Magazine.

Four tips on boosting SQL Server scalability

By: Don Jones, Contributor

Scaling out a SQL Server environment across multiple systems can be a difficult and complicated project, involving partitioned databases, federation and more. So, when it comes to SQL Server scalability, most organizations prefer to scale individual systems *up* as much as possible before trying to tackle the *out* option. Here are four tips for making the scale-up process easier and more effective.

Maximize SQL Server performance components. Every server's performance comes down to four basic components: memory, disk storage, network adapters and CPU. Your first step will be to maximize all four of these components in an existing server.

Start with memory – it can have the biggest impact and it's usually the most easily expanded piece of a server. There is just one limitation: On servers running a 32-bit version of Windows, there's no reason to have more than 4 GB of total RAM installed, because the operating system can't make use of more than that. On 64-bit machines running 64-bit versions of Windows and SQL Server, install as much RAM as the server can hold to get the best performance from SQL Server.

In fact, if you have any SQL Server instances running on 32-bit versions of Windows, migrate them to a 64-bit machine as your very first step, because enabling your systems to address that additional memory is one of the biggest performance improvements you can make. And don't be cheap on memory; buy whatever your server manufacturer recommends, which is often more expensive, error-correcting, high-speed memory. It's worth the expense.

Contents

Storage is the next thing you'll look at. It's a big topic, and it takes up its own tip later in this article. Suffice to say that faster storage performance is always a good thing.

Network connectivity is third. Plenty of SQL Server computers get performance-bound at the network adapter level. If you can afford them, multiple network adapters will provide multiple paths to your network. Gigabit Ethernet (GbE) adapters should be a minimum for SQL Server computers, and if your network can support 10 GbE, then go for it. It's especially important that your computers use a separate network and have at least one network adapter for every major use.

For example, if you're relying on iSCSI for storage communications, data transfers should be happening over a dedicated network interface controller (NIC) and a dedicated network, not over networks that are shared by client traffic.

Finally, examine your server's processors. They are last on this list for a reason: It's rare that you can upgrade them in a cost-effective fashion. That's because of the way processors are matched to their motherboards, which typically are designed for a specific family and generation of CPUs. To install significantly faster processors, you often have to get a new motherboard – and that usually means new memory and new everything else. In other words, a whole new server.

Adding more processors, if your server has room for them, will add some kick – with upgrades, *more* is always preferable to *slightly faster*. But here again, your options might be limited: Most servers are purchased with fully populated sockets, leaving no room for additional processors.

Virtualize when it makes sense. Believe it or not, virtualization can be a clever way to bring about performance benefits from SQL Server. It seems counterintuitive; after all, the point of virtualization is to run multiple workloads on a single host computer. So, by dedicating a single computer entirely to SQL Server, wouldn't you get better SQL Server performance?

Contents

Generally speaking, you would. If you have one SQL Server database that can keep an entire physical server completely occupied, then do just that. Many organizations, however, load their SQL Server systems with multiple databases or even multiple instances. There's nothing wrong with that – it's what SQL Server was designed for, in fact – but it doesn't give you much flexibility for matching workloads with the available hardware resources.

Instead, move those databases that need less than an entire physical server into a virtual machine. SQL Server is a great virtual-machine guest application, and by separating your databases across multiple SQL Server instances, each hosted in different virtual machines, you gain flexibility. Using live migration technologies, you can quickly move virtual machines from host to host, rearranging the virtual machines to best utilize available hardware based on current workload demands.

You can pull off similar tricks with clustering technologies, including Windows Cluster Service. Partition your databases into different clustered SQL Server instances (a SQL Server instance is, after all, a form of virtualization). You can then move those instances around at will, with very little interruption in database availability. If Database A needs to be scaled up one afternoon, shift other instances off of that database's cluster node, freeing up resources for use by the instance that's getting hammered. This kind of dynamic scaling can produce impressive performance results, although it does require your organization to develop a mature performance monitoring and response model.

If you're going to be spending big money on new server hardware, it makes sense to do so in conjunction with some kind of shared-resources scheme, such as clustering or virtualization. That way you'll have your expensive new hardware working as close as possible to its maximum capacity at all times while still delivering the performance your users require.

Upgrade storage performance. The *size* of your storage systems is driven by the capacity needs of your databases; the *speed* of those storage systems is something too often overlooked. In the SQL Server world, disk

Contents

storage speed is king. SQL Server is most likely to become I/O-bound before memory, network adapters or processors become an issue. You want storage that's *fast.* That means putting storage area networks (SANs) on the other end of hyper-fast fiber-optic connections, and using fast SAN protocols such as iSCSI to communicate with your disks.

Pay close attention to the exact kind of workload your database handles and match storage technologies such as RAID to that workload. RAID 5, for example, offers recoverability in the event of a device failure, but it can slightly slow down write times because the extra striping information, which enables data recovery, must be written along with each update.

Fast disk-controller electronics can help resolve that issue by caching data as fast as the server sends it, and then quickly dumping the data to platters. Every element of the storage subsystem plays a crucial role in performance – platter rotational speed, raw device I/O, average seek times, communications media (copper or fiber). Work with an experienced storage vendor to put together a system that offers the best performance for SQL Server.

If money is an issue, skimp on processor speed, and even memory, before you skimp on storage. Storage performance will go a long way toward scaling SQL Server up, up, up.

Invest in new servers. At some point, you're going to look at your existing servers and realize that you can't squeeze any more performance out of them. Your storage I/O is as fast as it's going to get. You've maxed out the memory. Every processor slot is full. The boxes are bristling with 10 GbE NICs, and they have enough cooling fans to power a hovercraft. In those cases, it's easy to talk yourself into new servers (although your chief financial officer might still argue the point with you).

What's tougher is to look at a server that clearly has room for expansion and decide to ditch it for a new one. When you're looking at empty memory slots, available CPU sockets and desolate PCI backplanes, it seems much more cost-effective to start filling in those holes with new, performance-enhancing components.

Contents

Sometimes that's a great idea. Sometimes you need to resist the urge.

First of all, never spend a dime upgrading a 32-bit computer. Replace it with a 64-bit one and install 64-bit versions of Windows and SQL Server. Load the new server with RAM aplenty, four or more multicore processors, and fast new NICs.

One last thought on SQL Server scalability: Any server more than four or five years old should be replaced, not upgraded. The cost to add memory or processors to an older server is often too close to what a brand-new server would cost, and a new server comes with lots of performance upgrades you couldn't buy a la carte: faster BIOS circuits, faster chipsets, faster memory bridges. That old server can still find useful life as a file server, or it can handle some other, less-intense workload than running SQL Server.

ABOUT THE AUTHOR

Don Jones *is a senior partner and principal technologist at strategic consulting firm Concentrated Technology. Contact him via www.ConcentratedTech.com.*

TechTarget

Contents

Free resources for technology professionals

TechTarget publishes targeted technology media that address your need for information and resources for researching products, developing strategy and making cost-effective purchase decisions. Our network of technology-specific Web sites gives you access to industry experts, independent content and analysis and the Web's largest library of vendor-provided white papers, webcasts, podcasts, videos, virtual trade shows, research reports and more —drawing on the rich R&D resources of technology providers to address market trends, challenges and solutions. Our live events and virtual seminars give you access to vendor neutral, expert commentary and advice on the issues and challenges you face daily. Our social community IT Knowledge Exchange allows you to share real world information in real time with peers and experts.

What makes TechTarget unique?

TechTarget is squarely focused on the enterprise IT space. Our team of editors and network of industry experts provide the richest, most relevant content to IT professionals and management. We leverage the immediacy of the Web, the networking and face-to-face opportunities of events and virtual events, and the ability to interact with peers—all to create compelling and actionable information for enterprise IT professionals across all industries and markets.

Related TechTarget Websites

- SearchDataManagement
- ebizQ.net
- SearchOracle
- SearchSAP