

## Tech Tip #4: Recommendations for Your Database Server

Overview: Recommendations for hardware for your PSQL database environment

One of the most common questions we get is “What kind of hardware should I buy for my server?” As you can imagine, this question is kind of like “Which new car should I buy?” – and it has just as many different aspects and idiosyncrasies to consider, too.

Let’s start with the basics – a Windows or Linux computer is needed for PSQL v11, so starting with hardware capable of running these environments is a hard necessity. Beyond that, however, the PSQL engine will typically run with whatever you give it. If you elect to run it on a server with 256MB of RAM, it will run – but VERY slowly. (Remember that early PC’s had limited resources, and yet the database engine still ran in those days – this has not changed.) In most cases, you’ll have (and want) much more than this.

So, let’s start with the key factors to consider when buying a database server:

- **CPU Clock Speed:** A single user running a database on a server only needs one core, but needs a fast clock speed to get fast performance. As such, the faster the clock speed you can get, the better. For best results, look at 2.6GHz or faster. Modern Haswell CPU’s (and newer) have over-clocking capabilities that can speed the server up over the native rate when power is available, and this is a great option for new hardware.
- **Number of Cores:** The number of cores on a server is more related to throughput, not performance. Just like a 4-lane freeway can carry more cars than a 2-lane road, having more cores will allow for better multi-tasking on the database, allowing work from multiple users to be done in parallel. (This was NOT the case in PSQLv10 or older, which suffered from some limitations on multi-core servers.) If you will have only a few users, then core count is not critical, and 4 should be more than enough. If you will have 50+ users, then 8+ cores are recommended. (Of course, all of these numbers are relative – 4 users all running reports will generate a LOT more traffic than 100 users running data entry applications.)
- **Memory:** System memory is the BEST way to increase performance, especially as your database gets larger and larger. A database will perform best if there are NO disk I/O operations needed to satisfy user requests, which means sizing the memory based on the expected size of your database, plus memory for the OS. We usually recommend reserving 4GB for the OS itself, but for large systems (over 24GB total), you may wish to reserve 8GB or more for the OS. We’ve seen companies get up to 384GB for their server, and we have heard of companies researching boxes with 1TB of RAM to cache extremely large data sets.
- **Network:** A fast network card with a TCP Offload Engine (TOE) is recommended. Be sure to use the latest drivers, and enable the TOE if needed. We recommend a minimum of 1 Gbps, but consider a 10Gbps NIC if your network infrastructure will support it now or in the near future.
- **Disk:** The disk subsystem is the most complicated piece, and there are many different factors to consider at the same time. Let’s look at each of these factors separately:
  - **Capacity:** Put simply, get as much disk space as you estimate you need for your environment. If in doubt, check with the developer of your application for typical database sizes for a company with your expected usage pattern.
  - **Disk Type:** Spinning disks? Solid state drives? Bus-attached flash memory? Spinning disks are the most common, but many are considering SSD storage today. Our biggest reservation on SSD components is longevity – databases tend to write the same pages over and over again, which may not be good for the hardware. If you can afford SSD or

flash memory systems like the Fusion I/O device, then go for it – but remember to get top quality components, or you may find yourself replacing them in a year or two as they get worn out.

- **Disk Speed:** More important to database performance (especially when memory is limited) is the disk throughput and latency ratings. Avoid SATA drives for servers, which have a lower maximum throughput than SAS. Disks that spin faster (i.e. 15,000 RPM) can also minimize latency over a slower drive.
- **RAID Level:** The configuration of the drives can matter, too. RAID6 should be avoided for database systems due to the slow disk write speeds. RAID 5 is acceptable – RAID10 is substantially better (though more costly), and RAID0 should NEVER be used. The RAID controller matters, too -- some of the fastest performance numbers we have seen are coming from the Dell PERC H710 controllers.

There are other factors to consider in a server, too. Most importantly, you should get a 64-bit operating system (and use the 64-bit database engine) whenever possible. Do not buy any new server with a 32-bit OS, because it severely limits the accessible memory and other resources the server can use.

Another big question is whether to virtualize a server or not. While there are many good reasons to virtualize a database server, performance is definitely NOT one of those reasons. Remember that the process of using a virtual server will slow I/O operations by as much as 30-40% -- and any database server will be mostly doing I/O operations. If your hardware is superior, however, then you should be able to virtualize with no problem. Here is a paper that explains virtualizing database servers in more detail:

<http://www.goldstarsoftware.com/papers/VirtualizingDatabaseServers.pdf>

Finally, the biggest single factor for most organizations is total price. You can build a REAL nice server for \$100,000! However, this is probably overkill for most environments. Essentially, buying a server becomes a constrained resource problem – find the **best** combination you can afford in your given budget. Of course, “best” is a tricky word. For our money, we would recommend these top items: Fastest core clock speed, largest memory (relative to data size), best overall disk speed, largest number of cores. Beyond that, you can tweak to your heart’s content within your budget constraint.