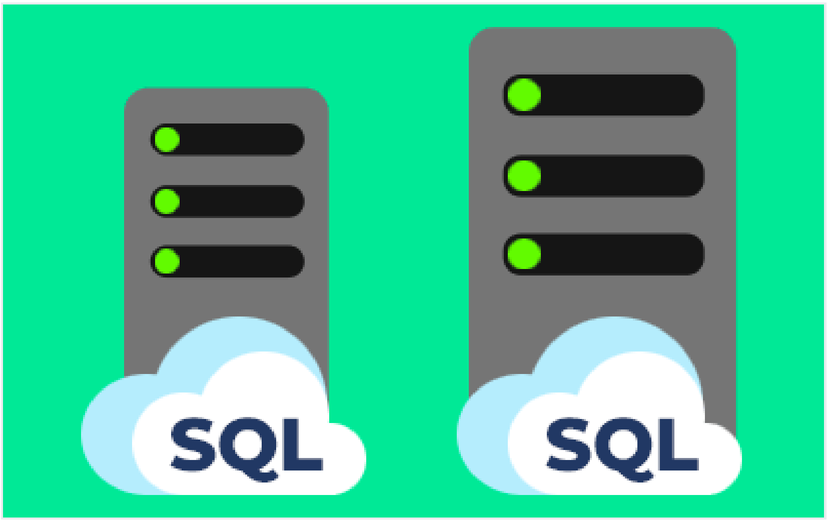
Azure SQL Managed Instance Performance Considerations

sqlperformance.com/2020/02/azure/sql-managed-instance-performance-considerations

Tim Radney

February 26, 2020

Azure SQL Database Managed Instance became generally available in late 2018. Since then, many organizations have started migrating to Managed Instance for the benefits of a managed environment. Organizations are taking advantage of having managed backups, lots of built-in security features, an uptime SLA of 99.99%, and an always up-to-date environment where they are no longer responsible for patching SQL Server or the operating system.



*One size does **not** always fit all.*
Managed Instance provides two tiers for performance. The **General Purpose** tier is designed for applications with typical performance and I/O latency requirements and provides built-in HA. The **Business Critical** tier is designed for applications that require low I/O latency and higher HA requirements. Business Critical also provides two non-readable secondaries and one readable secondary. The readable secondary is a great way to distribute the workload off of the primary, which can lower the service tier required for the primary — decreasing the overall spend for the instance.

When Managed Instance was first released, you could choose between Gen4 and Gen5 processors. Gen4 is still described in the documentation, but this option is mostly unavailable now. For this article, I'll only be covering configurations using the Gen5 processors.

Each service tier supports anywhere from 4 to 80 logical CPUs — also known as virtual cores, or vCores. Memory is allocated at approximately 5.1 GB per vCore. General Purpose provides up to 8 TB of high-performance Azure Blob storage, while Business Critical provides up to 4 TB of super-fast local SSD storage.

Memory

With only having 5.1 GB of memory per vCore, an instance with fewer vCores could struggle. The options for vCore configurations are 4, 8, 16, 24, 32, 40, 64, and 80 vCores. If you do the math on each of the vCore options (`(number of vCores) × (5.1 GB)`), you'll get the following core / memory combinations:

```
 4 vCores  =   20.4 GB
 8 vCores  =   40.8 GB
16 vCores  =   81.6 GB
24 vCores  =  122.4 GB
32 vCores  =  163.2 GB
40 vCores  =  204.0 GB
64 vCores  =  326.4 GB
80 vCores  =  408.0 GB
```

For many organizations I've helped transition from on-premises to Managed Instance, I've seen a significant reduction in memory. Typical configurations on-premises would be 4 vCores and 32 GB of memory, or 8 vCores and 64 GB. Both account for more than a 30% reduction in memory. If the instance was already under memory pressure, this can cause problems. In most cases, we've been able to optimize the on-premises instance to help alleviate the memory pressure prior to moving to Managed Instance, but in a few cases, the customer had to go with a higher vCore instance to alleviate the memory pressure.

Storage

Storage is a bit more difficult to plan and make considerations for, due to having to consider multiple factors. For storage you need to account for the overall storage requirement for both storage size, and I/O needs. How many GBs or TBs are needed for the SQL Server instance and how fast does the storage need to be? How many IOPS and how much throughput is the on-premises instance using? For that, you must baseline your current workload using perfmon to capture average and max MB/s and/or taking snapshots of sys.dm_io_virtual_file_stats to capture throughput utilization. This will give you an idea of what type of I/O and throughput you need in the new environment. Several customers I've worked with have missed this vital part of migration planning and have encountered performance issues due to selecting an instance level that didn't support their workload.

This is critical to baseline because with on-premises servers, it is common to have storage provided from a super-fast SAN with high throughput capabilities to smaller size virtual machines. In Azure, your IOPS and throughput limits are determined by the size of the compute node, and in the case of Manage Instance, it is determined by the number of vCores allocated. For General Purpose there is a limit of 30-40k IOPS per instance or 500 up to 12,500 IOPS per file depending on the file size. Throughput per file is also based on size ranging starting at 100 MiB/s for up to 128 GB files, and up to 480 MiB/s for 4 TB and larger files. In Business Critical, IOPS range from 5.5K – 110K per instance or 1,375 IOPS per vCore.

Consumers must also account for log write throughput for the instance. General Purpose is 3 MB/s per vCore with a max of 22MB/s for the instance and Business Critical is 4 MB/s per vCore with a max of 48 MB/s for the entire instance. In my experience working with customers, many have far exceeded these limits for write throughput. For some it has been a showstopper, and for others, they have been able to optimize and modify their system to decrease the load.

In addition to needing to know overall throughput and I/O requirements, storage size is also tied to the number of vCores chosen. For General Purpose:

```
     4 vCores  =  2 TB max
8 - 80 vCores  =  8 TB max
```

For Business Critical:

```
 4 – 16 vCores  =  1 TB
     24 vCores  =  2 TB
32 - 80 vCores  =  4 TB
```

For General Purpose, once you get to 8 vCores, you can max out the available storage, which works well for those who only need General Purpose. But when you need Business Critical, things can be more challenging. I've worked with many customers who have multiple TBs allocated to VMs with 4, 8, 16, and 24 logical processors. For any of these customers, they would have to move up at least 32 vCores just to meet their storage requirement, a costly option. Azure SQL Database has a similar issue with max database size, and Microsoft came up with a Hyperscale option. We expect this to become an option for Managed Instance at some point to address the storage limits in a similar way.

The size of tempdb is also correlated to number of vCores. In the General Purpose tier, you get 24 GB per vCore (up to 1,920 GB) for the data files, with a tempdb log file size limit of 120 GB. For the Business Critical tier, tempdb can grow all the way up to the currently available instance storage size.

In-memory OLTP

For those who are currently taking advantage of In-Memory OLTP (or plan to), note that it is only supported in the Business Critical service tier. The amount of space available for In-Memory tables is also limited by vCores:

```
 4 vCores  =    3.14 GB
 8 vCores  =    6.28 GB
16 vCores  =   15.77 GB
24 vCores  =   25.25 GB
32 vCores  =   37.94 GB
40 vCores  =   52.23 GB
64 vCores  =   99.90 GB
80 vCores  =  131.86 GB
```

Conclusion

When planning a migration to Azure SQL Managed Instance, there are multiple considerations to take into account prior to deciding to migrate. First you need to fully understand your memory, CPU, and storage requirements, as this will determine the size of the instance. Just as important is knowing what your storage I/O requirements are. IOPS and throughput for the General Purpose tier are directly tied to vCores and the size of the database files. For Business Critical it is tied to the number of vCores. If you have a very I/O intensive workload, Business Critical is the more appealing service tier due to it providing higher IOPS and throughput numbers. The challenge with Business Critical is the lower storage capacity and only supporting 1TB for the entire instance up to 16 vCores.

With proper planning, and possible deconsolidation of larger instances to smaller Managed Instances, this offering can be a very viable migration option for many organizations. The appeal are the benefits of having managed backups, built-in HA with an SLA of 99.99%, added security features and options, and not having to worry about patching an OS or SQL Server instance.