\_\_\_\_\_\_\_ \_\_\_\_\_\_ SQL Performance Testing

November 2018

Testing conducted by \_\_\_\_\_\_\_\_\_\_

Report prepared by \_\_\_\_\_\_\_\_\_\_

# 1.0.0 Summary

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ already supports Microsoft SQL Server. However, testing done earlier this year showed that performance on Azure SQL Database was below expectation and did not provide a substantive cost benefit. However, a number of things have changed since then:

* Azure SQL Database configurations are now primarily configured with separate options for vCores and storage.
* There is now a Business Critical SKU leveraging local SSD for very high IOPS and low latency.
* There is now a Hyperscale SKU for decoupling compute for storage and offering support for database sizes up to 100 TB.
* Azure SQL Database vCore options now give a choice between Gen 4 and Gen 5 architecture.

This round of testing was focused on performing the same performance tests (Emma, John, Mark, and Paul) leveraging these new options.

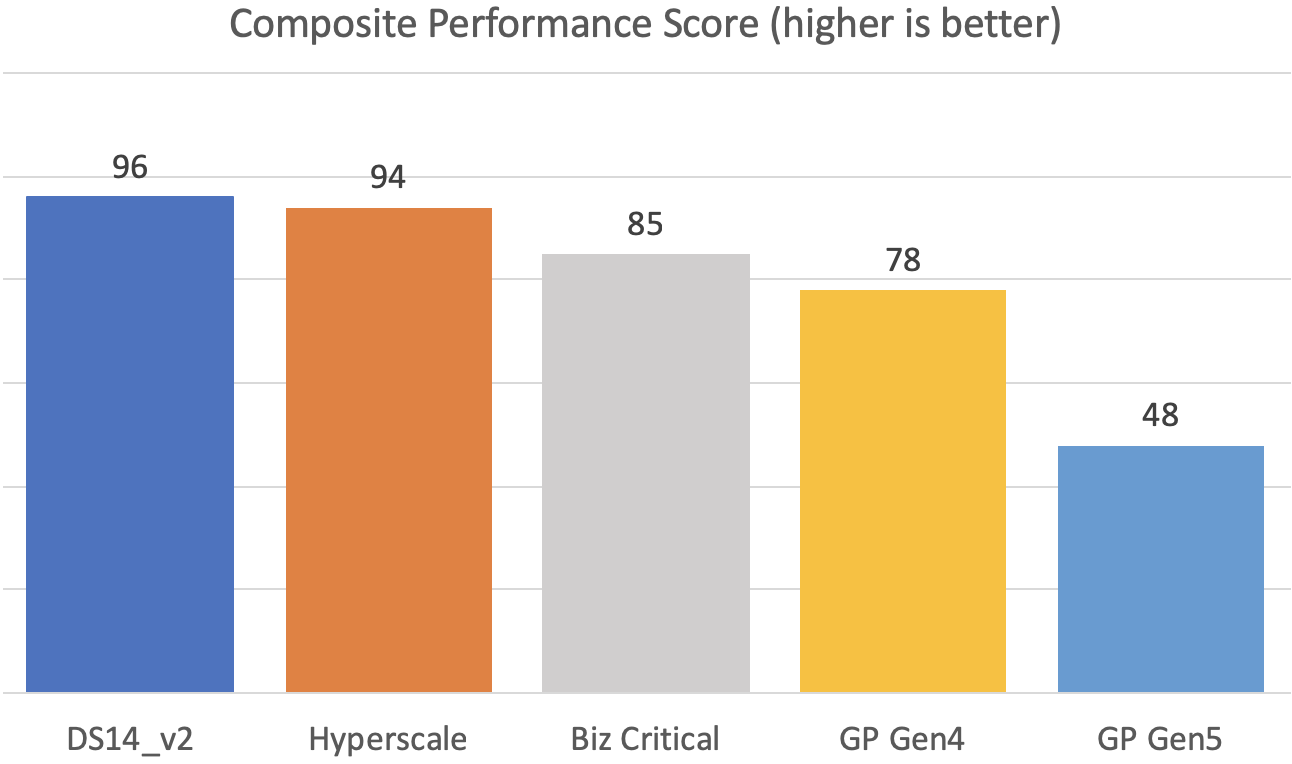
## 1.1.0 Maximum Degree of Parallelism

After testing it both ways, we discovered that \_\_\_\_\_ performs 2x faster or more when the maximum degree of parallelism setting is set to “1”, meaning there is no parallelism. It is likely that there are enough threads of activity going on during this testing to saturate the processors anyway and so breaking up queries is leading to longer query times.

Information about setting this configuration option can be found in Section 4.2.0.

## 1.2.0 Performance

The specific test details are included later in this report, but this chart shows the composite results weighing each test the same.



Performance is only one factor, in the next section we will look at the price/performance.

A “100” on this chart would be perfect performance – the configuration performed at the highest performance for every single scenario.

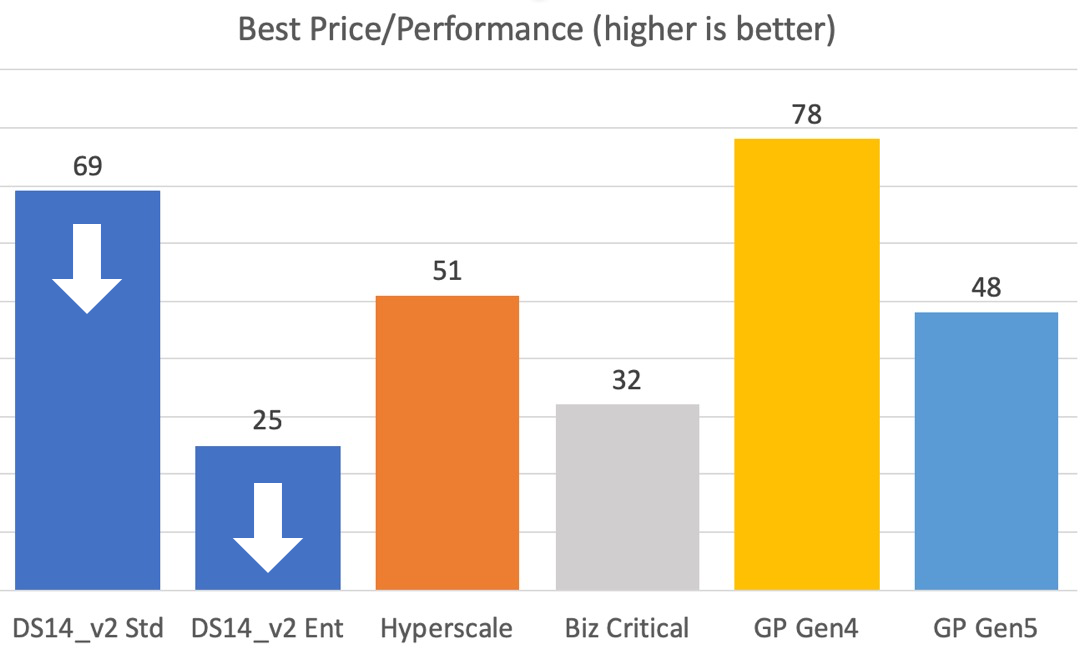
The composite performance score was comprised by taking the average performance of each test for each configuration. The highest performer was given 25 points while the others were given an appropriate percentage based on their performance. The total of all 4 tests were then added together for this composite score.

## 1.3.0 Value

Of course performance is only one factor, one must also consider how much it costs to run in each configuration. The following was used to generate the pricing shown:

* Pricing is for comparison only, your pricing will vary
* Pricing is retail without discounts, reserved instance pricing, or hybrid benefits
* Pricing is shown for “East US 2”
* Pricing is shown per month (730 hours)
* Pricing for backup storage is not included, but would be similar across all options
* Hyperscale is in “preview” so the pricing is based on 2x the current preview price
* All Azure SQL Database SKUs will have a cost for IO starting in 2019, but the price is currently unknown

|  |  |  |  |
| --- | --- | --- | --- |
| Configuration | Details | Availability | Pricing |
| DS14\_v2 Std | 2 nodes (primary, secondary)  16 vCore, 112 GB RAM  Windows License  SQL Server Standard License for primary  P4 for OS  P30 for Data/Log/TempDB Up to 5,000 IOPS | 1 offline replica | $4,265.60 |
| DS14\_v2 Ent | 2 nodes (primary, secondary)  16 vCore, 112 GB RAM  Windows License  SQL Server Enterprise License for primary  P4 for OS  P30 for Data/Log/TempDB Up to 5,000 IOPS | 1 offline replica | $11,853.60 |
| Hyperscale | Gen 4 – 16 vCore, 112 GB RAM  1 TB storage  Unknown but very high IOPS | 1 read-replica | ~$5,634.70 |
| Biz Critical | Gen 4 – 16 vCore, 112 GB RAM  1 TB storage  Up to 80,000 IOPS | 1 read replica  2 offline replicas  optional zone-redundant | $8,192.40 |
| GP Gen4 | Gen 4 – 16 vCore, 112 GB RAM  1 TB storage  Up to 7,000 IOPS | 1 offline replica | $3,060.55 |
| GP Gen5 | Gen 5 – 16 vCore, 88 GB RAM  1 TB storage  Up to 7,000 IOPS | 1 offline replica | $3,060.55 |



A “100” on this chart would be perfect price/performance – the configuration performed at the highest performance for every single scenario and was the cheapest. Unfortunately, no solution offered that, but consider the following:

* If you need SQL Enterprise features (not needed for PPSS currently), all Azure SQL DB SKUs are a better value.
* “General Purpose Gen 4” performed adequately in all tests, performed great on the most complex use case (Paul), and was the least expensive solution.
* If you only need SQL Standard features, using a “DS14\_v2” VM seems a good value. This report does not account for increased maintenance costs of running your own VM (threat detection, monitoring tools, patching, software upgrades, configuration management, etc.). There will be increase costs for running a VM, pushing that number down (the arrow above is a reminder).
* “Hyperscale” might be a consideration for very large customers. “General Purpose” and “Business Critical” support databases up to 4 TB whereas “Hyperscale” supports them up to 100 TB. “Hyperscale” is in preview and is missing quite a few features that will be coming in over the coming months.
* “Business Critical” might be a consideration for those customers that have increased availability requirements.
* “General Purpose Gen 5” is clearly not the right answer for this workload, it is the same cost as Gen 4 but performs the worst on every test.

# 2.0.0 Differences

There are feature and functionality differences between the different options. This section will attempt to detail some of the significant factors.

## 2.1.0 Azure SQL DB vs. SQL Server on a VM

This page goes into detail on the differences: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas>, but I will outline a few important ones here.

Azure SQL Database is optimized to reduce overall management costs to the minimum.

Azure SQL Database offers the following key benefits:

* Best practice deployment
* Adjust performance and scale without downtime
* Extensive monitoring and alerting capabilities
* 99.99% availability SLA
* Automatic patching, backup, replication, failover, update, and other maintenance tasks.
* Point-in-time restore
* Active geo-replication
* Global failover groups
* Elastic pools to maximum resource utilization
* Automatic performance monitoring and tuning
* Adaptive query processing
* Advanced threat protection
* Auditing
* Data encryption
* Compliance certification

Many of the features that Azure SQL Database offers, you could duplicate on your own, but it would require significant investment of Database Administrators and IT Operations staff.

## 2.2.0 Enterprise vs. Standard

Azure SQL Database is built on SQL Server Enterprise and so offers additional features like these:

* > 24 cores
* Multiple replicas via Always on Availability Groups
* Readable secondaries
* Transparent Data Encryption

You can find more details on the differences between Enterprise and Standard licensing here: <https://www.microsoft.com/en-us/sql-server/sql-server-2017-editions>.

## 2.3.0 Gen 4 vs. Gen 5

Azure SQL Database vCore models are available with Gen 4 and Gen 5 CPUs:

* Gen 4 CPUs
  + Intel E5-2673 v3 (Haswell) 2.4 GHz processors
  + 1 vCore = 1 physical CPU
  + Up to 24 vCores
  + Up to 164 GB memory
* Gen 5 logical CPUs
  + Intel E5-2673 v4 (Broadwell) 2.3 GHz processors
  + 1 vCore = 1 hyper thread
  + Up to 80 vCores
  + Up to 396 GB memory

Gen 5 architecture is a refinement of the Gen 4 architecture that should be just a little bit better in every way. However, the move to hyperthreaded vCores seems to be a huge detriment for this workload. In performance testing we routinely saw CPU utilization staying at 100%.

## 2.4.0 Azure SQL Database vCore Service Tiers

Azure SQL Database is offered in these tiers:

* General Purpose – Budget-oriented, scalable compute and storage options for most business workloads.
* Business Critical – Local SSD for high IO and low latency; very high resiliency via multiple replicas.
* Hyperscale – Highly scalable storage and read-scale capabilities for very large databases.

You can read more here: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers-vcore>.

## 2.5.0 General Purpose vs. Business Critical

The tempdb is stored on local SSD in both cases, but General Purpose uses Premium storage for data and logs whereas Business Critical uses local SSD. General Purpose offers 500 IOPS per disk up to 7k IOPS maximum. Business Critical offers 5k IOPS per disk up to 200k IOPS maximum.

General Purpose offers 1 replica that cannot be used as a read intent target, whereas Business Critical has 4 replicas (1 configured as a read intent target).

Business Critical supports synchronous replication across Azure Availability Zones as an option.

You can read specifics here:

* <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers-vcore>
* <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers-general-purpose-business-critical>.

## 2.6.0 Hyperscale

Hyperscale is an architecture that separates compute, page servers, log servers, and storage into different nodes. This allows for some unique features:

* Up to 100 TB database sizes
* Near instantaneous backups via snapshots
* Fast database restores via snapshots
* High performance regardless of data size
* Rapid scale out of read-only nodes
* Rapid scale up without downtime

Hyperscale is in preview and there are many Azure SQL Database features not currently supported.

You can read specifics here:

* <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tier-hyperscale>
* <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tier-hyperscale-faq>

# 3.0.0 Tests

The following section describes the tests that were run and shows the results of each.

Emma, John, and Mark are all similar tests that are run simultaneously. They each filter and page on simpler \_\_\_\_\_ lists with only 6 elements.

* Emma: 66k products filtered and paged through 4 times, 396k \_\_\_\_\_ elements generated
* John: 115k products filtered and paged through 2 times, 690k \_\_\_\_\_ elements generated
* Mark: 440k products filtered and paged through 1 time, 2.6m \_\_\_\_\_ elements generated

When a workflow is run more than once, it will not duplicate the number of output \_\_\_\_\_ elements in the database, but will merge or overwrite the previous values.

Paul is run separately. This test conducts the following steps 2 times:

1. Navigates to a \_\_\_\_\_\_\_
2. Filters out ~94k \_\_\_\_\_\_\_
3. Retrieves a random page of 100 \_\_\_\_\_\_
4. Selects all the \_\_\_\_\_ affected by the filter
5. Triggers a \_\_\_\_\_ recalculation for the 94k items
6. Produces 94k \_\_\_\_\_ that contain several \_\_\_\_\_ element children (48 elements each = ~4m \_\_\_\_ elements)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

All test times are show as “minutes:seconds”.

## 3.1.0 Index Fragmentation and Statistics

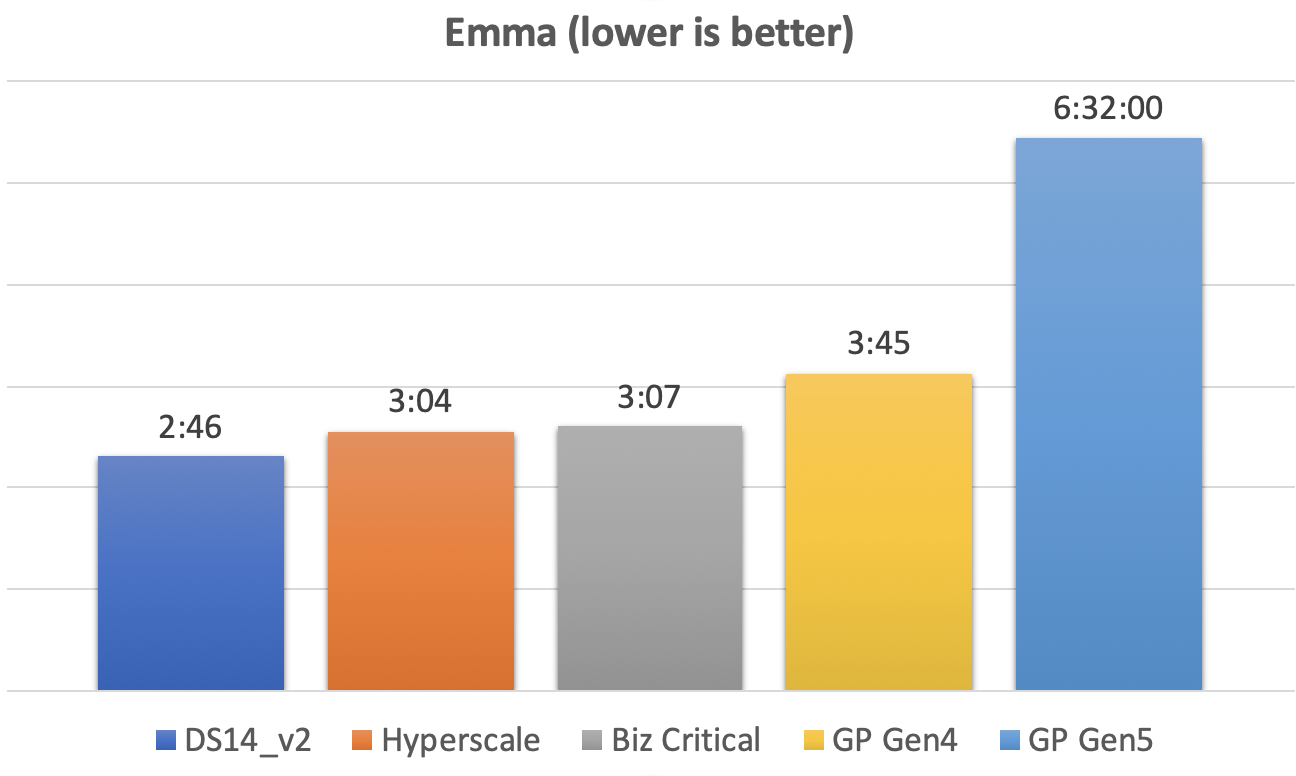
We found a lot of fragmentation in some of the indexes (a few were as high as 99%) so we did a REBUILD on all the indexes that were used by the testing that were above 5% fragmentation. After REBUILD, we recalculated statistics.

* <https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-db-index-physical-stats-transact-sql?view=sql-server-2017>
* <https://docs.microsoft.com/en-us/sql/t-sql/statements/update-statistics-transact-sql?view=sql-server-2017>

## 3.1.0 Emma

Best Performers: DS14\_v2, Hyperscale, Business Critical

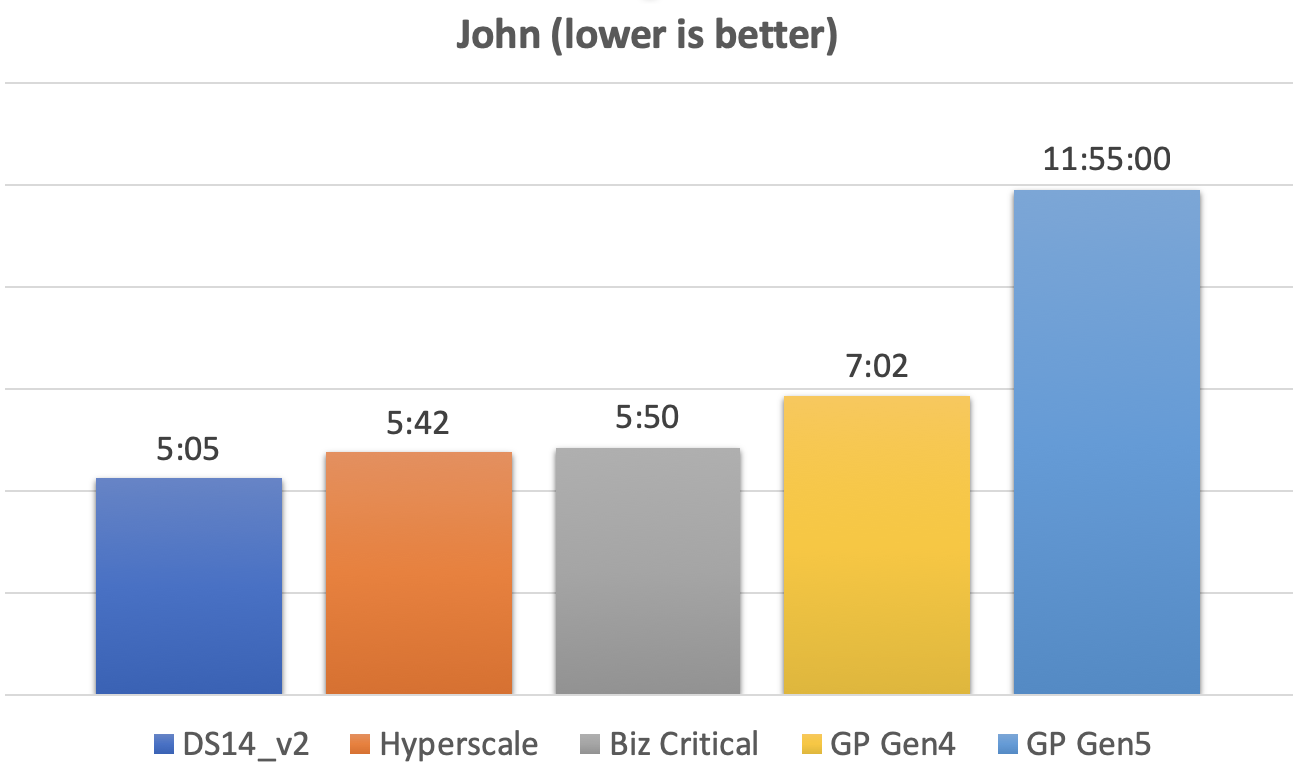
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **DS14\_v2** | **Hyperscale** | **Biz Critical** | **Gen Purpose** | **Gen Purpose** |
| Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen5 16 vCores  88 GB RAM |
| Run 1 | 3:15 | 3:47 | 3:51 | 2:16 | 14:10 |
|  | 2:22 | 2:30 | 2:31 | 3:51 | 5:05 |
|  | 3:23 | 3:42 | 4:01 | 6:51 | 8:38 |
|  | 2:03 | 2:04 | 2:10 | 3:16 | 5:13 |
| Run 2 | 3:20 | 3:48 | 3:48 | 2:15 | 8:37 |
|  | 2:21 | 2:42 | 2:32 | 3:41 | 3:18 |
|  | 3:19 | 3:50 | 3:56 | 5:10 | 8:24 |
|  | 2:09 | 2:03 | 2:09 | 3:27 | 3:07 |
| Run 3 | 3:11 | 3:56 | 3:46 | 4:21 | 8:39 |
|  | 2:23 | 2:37 | 2:31 | 3:24 | 3:40 |
|  | 3:13 | 3:52 | 3:57 | 4:29 | 8:07 |
|  | 2:04 | 2:03 | 2:10 | 2:21 | 3:35 |
| Run 4 | 3:11 | 3:53 | 3:43 | 4:08 | 8:40 |
|  | 2:25 | 2:25 | 2:38 | 3:25 | 3:06 |
|  | 3:24 | 3:40 | 3:59 | 4:32 | 8:45 |
|  | 2:07 | 2:05 | 2:09 | 2:27 | 3:22 |
| Average | **2:46** | **3:04** | **3:07** | **3:45** | **6:32** |
| Score | **25** | **23** | **22** | **18** | **11** |



## 3.2.0 John

Best Performers: DS14\_v2, Hyperscale, Business Critical

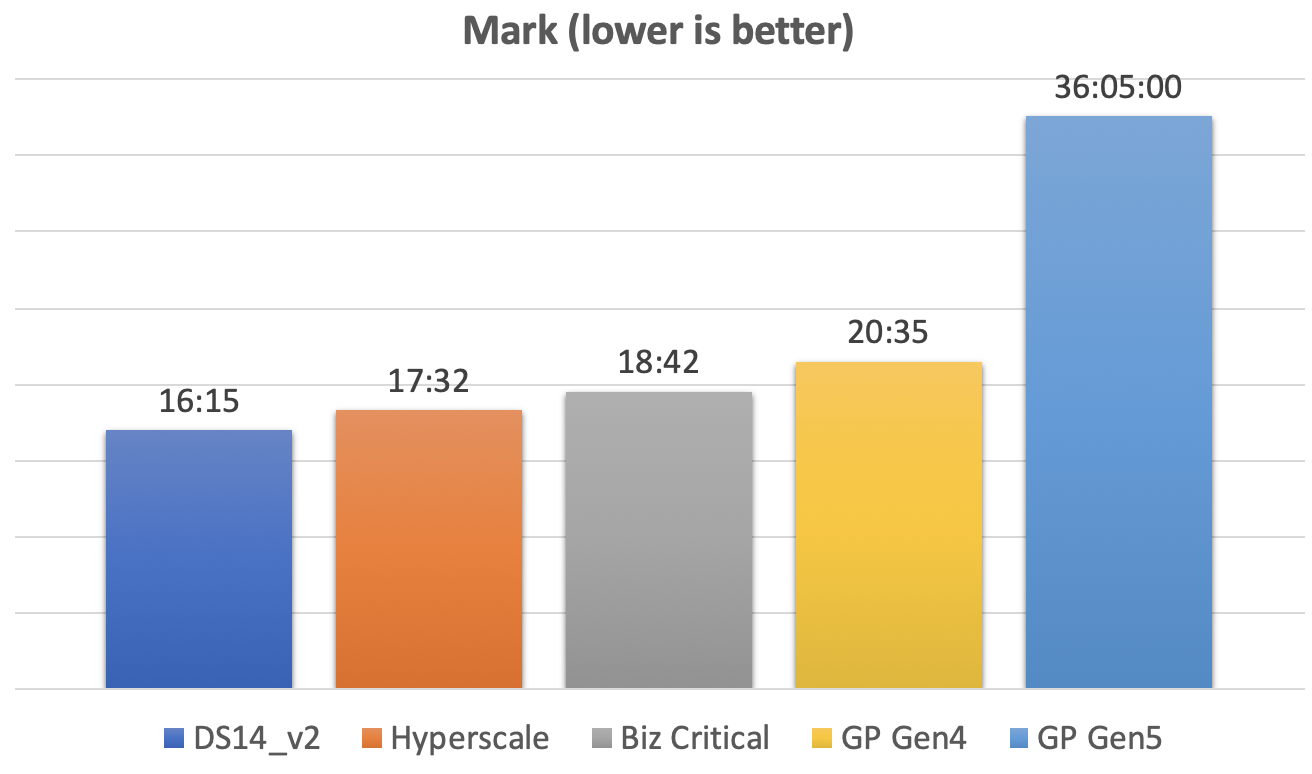
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **DS14\_v2** | **Hyperscale** | **Biz Critical** | **Gen Purpose** | **Gen Purpose** |
| Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen5 16 vCores  88 GB RAM |
| Run 1 | 5:08 | 5:46 | 5:49 | 7:12 | 9:11 |
|  | 5:03 | 5:31 | 5:50 | 7:02 | 14:04 |
| Run 2 | 5:10 | 5:50 | 5:49 | 7:23 | 11:55 |
|  | 5:07 | 5:43 | 5:48 | 7:24 | 11:45 |
| Run 3 | 4:59 | 5:56 | 5:49 | 7:12 | 12:20 |
|  | 5:02 | 5:38 | 5:49 | 6:27 | 11:56 |
| Run 4 | 5:01 | 5:45 | 5:54 | 6:52 | 11:52 |
|  | 5:09 | 5:25 | 5:49 | 6:42 | 12:15 |
| Average | **5:05** | **5:42** | **5:50** | **7:02** | **11:55** |
| Score | **25** | **23** | **22** | **18** | **11** |



## 3.3.0 Mark

Best Performers: DS14\_v2, Hyperscale

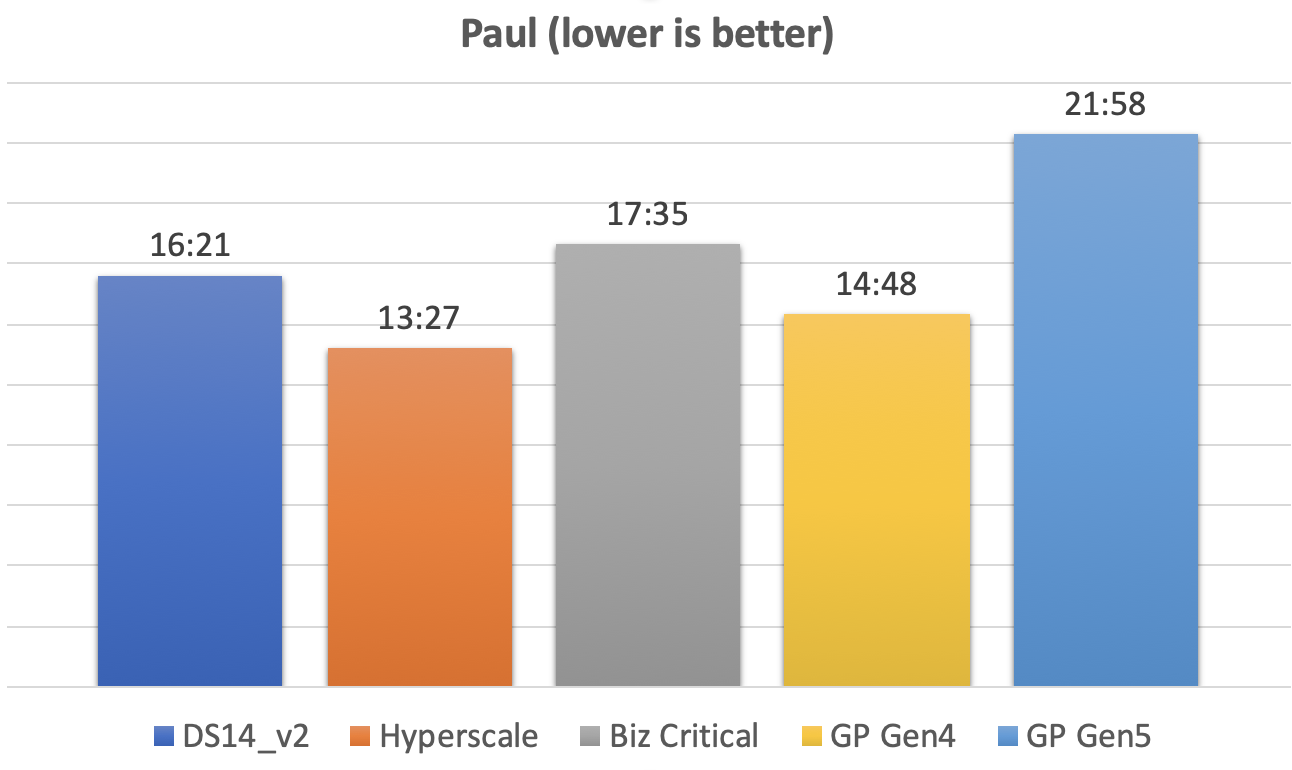
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **DS14\_v2** | **Hyperscale** | **Biz Critical** | **Gen Purpose** | **Gen Purpose** |
| Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen5 16 vCores  88 GB RAM |
| Run 1 | 16:13 | 17:26 | 18:50 | 18:46 | 43:54 |
| Run 2 | 16:25 | 17:52 | 18:44 | 20:20 | 33:03 |
| Run 3 | 16:10 | 17:42 | 18:32 | 21:28 | 33:44 |
| Run 4 | 16:14 | 17:10 | 18:41 | 21:44 | 33:41 |
| Average | **16:15** | **17:32** | **18:42** | **20:35** | **36:05** |
| Score | **25** | **23** | **22** | **19** | **11** |



## 3.4.0 Paul

Best Performers: Hyperscale, General Purpose Gen4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **DS14\_v2** | **Hyperscale** | **Biz Critical** | **Gen Purpose** | **Gen Purpose** |
| Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen4 16 vCores  112 GB RAM | Gen5 16 vCores  88 GB RAM |
| Run 1 | 17:35 | 15:10 | 18:48 | 17:46 | 23:30 |
|  | 17:05 | 14:56 | 18:51 | 15:59 | 22:59 |
| Run 2 | 15:54 | 12:54 | 17:12 | 14:06 | 21:44 |
|  | 15:47 | 12:34 | 16:52 | 13:41 | 21:19 |
| Run 3 | 15:57 | 12:33 | 16:53 | 13:39 | 21:06 |
|  | 15:46 | 12:33 | 16:51 | 13:34 | 21:07 |
| Average | **16:21** | **13:27** | **17:35** | **14:48** | **21:58** |
| Score | **21** | **25** | **19** | **23** | **15** |



# 4.0.0 Configuration options of each configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | name | DS14v2 | Azure SQL DB | description |
| 101 | recovery interval (min) | 0 | 0 | Maximum recovery interval in minutes |
| 102 | allow updates | 0 | 0 | Allow updates to system tables |
| 103 | user connections | 0 | 0 | Number of user connections allowed |
| 106 | locks | 0 | 0 | Number of locks for all users |
| 107 | open objects | 0 | 0 | Number of open database objects |
| 109 | fill factor (%) | 0 | 0 | Default fill factor percentage |
| 114 | disallow results from triggers | 0 | 0 | Disallow returning results from triggers |
| 115 | nested triggers | 1 | 1 | Allow triggers to be invoked within triggers |
| 116 | server trigger recursion | 1 | 1 | Allow recursion for server level triggers |
| 117 | remote access | 1 | 1 | Allow remote access |
| 124 | default language | 0 | 0 | default language |
| 400 | cross db ownership chaining | 0 | 0 | Allow cross db ownership chaining |
| 503 | max worker threads | 0 | 0 | Maximum worker threads |
| 505 | network packet size (B) | 4096 | 4096 | Network packet size |
| 518 | show advanced options | 1 | 0 | show advanced options |
| 542 | remote proc trans | 0 | 0 | Create DTC transaction for remote procedures |
| 544 | c2 audit mode | 0 | 0 | c2 audit mode |
| 1126 | default full-text language | 1033 | 1033 | default full-text language |
| 1127 | two digit year cutoff | 2049 | 2049 | two digit year cutoff |
| 1505 | index create memory (KB) | 0 | 0 | Memory for index create sorts (kBytes) |
| 1517 | priority boost | 0 | 0 | Priority boost |
| 1519 | remote login timeout (s) | 10 | 10 | remote login timeout |
| 1520 | remote query timeout (s) | 600 | 600 | remote query timeout |
| 1531 | cursor threshold | -1 | -1 | cursor threshold |
| 1532 | set working set size | 0 | 0 | set working set size |
| 1534 | user options | 0 | 0 | user options |
| 1535 | affinity mask | 0 | 0 | affinity mask |
| 1536 | max text repl size (B) | 65536 | 65536 | Maximum size of a text field in replication. |
| 1537 | media retention | 0 | 0 | Tape retention period in days |
| 1538 | cost threshold for parallelism | 75 | 5 | cost threshold for parallelism |
| 1539 | max degree of parallelism | 1 | 0 | maximum degree of parallelism |
| 1540 | min memory per query (KB) | 1024 | 1024 | minimum memory per query (kBytes) |
| 1541 | query wait (s) | -1 | -1 | maximum time to wait for query memory (s) |
| 1543 | min server memory (MB) | 0 | 0 | Minimum size of server memory (MB) |
| 1544 | max server memory (MB) | 2147483647 | 2147483647 | Maximum size of server memory (MB) |
| 1545 | query governor cost limit | 0 | 0 | Maximum estimated cost allowed by query governor |
| 1546 | lightweight pooling | 0 | 0 | User mode scheduler uses lightweight pooling |
| 1547 | scan for startup procs | 0 | 0 | scan for startup stored procedures |
| 1549 | affinity64 mask | 0 | 0 | affinity64 mask |
| 1550 | affinity I/O mask | 0 | 0 | affinity I/O mask |
| 1551 | affinity64 I/O mask | 0 | 0 | affinity64 I/O mask |
| 1555 | transform noise words | 0 | 0 | Transform noise words for full-text query |
| 1556 | precompute rank | 0 | 0 | Use precomputed rank for full-text query |
| 1557 | PH timeout (s) | 60 | 60 | DB connection timeout for full-text protocol handler (s) |
| 1562 | clr enabled | 0 | 1 | CLR user code execution enabled in the server |
| 1563 | max full-text crawl range | 4 | 4 | Maximum  crawl ranges allowed in full-text indexing |
| 1564 | ft notify bandwidth (min) | 0 | 0 | Number of reserved full-text notifications buffers |
| 1565 | ft notify bandwidth (max) | 100 | 100 | Max number of full-text notifications buffers |
| 1566 | ft crawl bandwidth (min) | 0 | 0 | Number of reserved full-text crawl buffers |
| 1567 | ft crawl bandwidth (max) | 100 | 100 | Max number of full-text crawl buffers |
| 1568 | default trace enabled | 1 | 0 | Enable or disable the default trace |
| 1569 | blocked process threshold (s) | 0 | 20 | Blocked process reporting threshold |
| 1570 | in-doubt xact resolution | 0 | 0 | Recovery policy for DTC transactions with unknown outcome |
| 1576 | remote admin connections | 0 | 0 | Dedicated Admin Connections are allowed from remote clients |
| 1577 | backup compression default |  | 0 | Enable compression of backups by default |
| 1578 | filestream access level |  | 0 | Sets the FILESTREAM access level |
| 1579 | optimize for ad hoc workloads | 0 | 0 | When this option is set, plan cache size is further reduced for single-use adhoc OLTP workload. |
| 1580 | access check cache bucket count | 0 | 0 | Default hash bucket count for the access check result security cache |
| 1581 | access check cache quota | 0 | 0 | Default quota for the access check result security cache |
| 1582 | backup checksum default | 0 | 0 | Enable checksum of backups by default |
| 1583 | automatic soft-NUMA disabled | 0 | 0 | Automatic soft-NUMA is enabled by default |
| 1584 | external scripts enabled | 0 | 0 | Allows execution of external scripts |
| 1585 | clr strict security | 0 | 0 | CLR strict security enabled in the server |
| 1586 | column encryption enclave type | 0 | 0 | Type of enclave used for computations on encrypted columns |
| 1587 | Agent XPs |  | 1 | Enable or disable Agent XPs |
| 1588 | Database Mail XPs |  | 0 | Enable or disable Database Mail XPs |
| 16384 | SMO and DMO XPs | 1 | 0 | Enable or disable SMO and DMO XPs |
| 16386 | Ole Automation Procedures | 0 | 0 | Enable or disable Ole Automation Procedures |
| 16387 | xp\_cmdshell | 1 | 1 | Enable or disable command shell |
| 16388 | Ad Hoc Distributed Queries | 0 | 0 | Enable or disable Ad Hoc Distributed Queries |
| 16390 | Replication XPs | 0 | 0 | Enable or disable Replication XPs |
| 16391 | contained database authentication | 0 | 0 | Enables contained databases and contained authentication |
| 16392 | hadoop connectivity | 0 | 0 | Configure SQL Server to connect to external Hadoop or Microsoft Azure storage blob data sources through PolyBase |
| 16393 | polybase network encryption | 0 | 0 | Configure SQL Server to encrypt control and data channels when using PolyBase |
| 16394 | remote data archive | 0 | 0 | Allow the use of the REMOTE\_DATA\_ARCHIVE data access for databases |
| 16395 | allow polybase export | 1 | 1 | Allow INSERT into a Hadoop external table |
| 16396 | allow filesystem enumeration | 0 | 0 | Allow the use of the REMOTE\_DATA\_ARCHIVE data access for databases |
| 16397 | polybase enabled | 0 | 0 | Allow INSERT into a Hadoop external table |
| 16398 | allow filesystem enumeration |  | 1 | Allow enumeration of filesystem |
| 16399 | polybase enabled |  | 0 | Configure SQL Server to connect to external data sources through PolyBase |

## 4.1.0 Show Advanced Options

This feature is turned off on Azure SQL Database and may not be turned on. You can find out more about the setting here: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/show-advanced-options-server-configuration-option?view=sql-server-2017>.

This should not have any impact on performance.

## 4.2.0 Maximum Degree of Parallelism

If the estimated cost to run a query is below the “1538: cost threshold for parallelism” threshold, then the query will not be parallelized. If it is above this threshold, it *could* be parallelized if 1539 is set to something other than “1” and there are multiple logical processors.

“1539: max degree of parallelism” determines how many threads the query can be broken into for processing if the threshold for 1538 is met. The default of “0” allows the query to use all available processors.

While Azure SQL Database does not allow these server configuration settings to be changed, you can change them via database scoped configurations: <https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-database-scoped-configuration-transact-sql?view=sql-server-2017/>.

There are guidelines for setting the degree of parallelism here: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-the-max-degree-of-parallelism-server-configuration-option?view=sql-server-2017>.

This setting had a tremendous impact on performance. Running with “1” (no parallelism) doubled the performance of the application.

## 4.3.0 CLR

In reference to “1562: clr enabled”, only Managed Instance and Business Critical tiers have limited support for CLR. You can find out full details:

* <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-features>
* <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-transact-sql-information#clr>

To find out more about the 1562 setting, you can reference: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/clr-enabled-server-configuration-option?view=sql-server-2017>.

## 4.4.0 Default Trace

In reference to “1568: default trace enabled”, this feature has been deprecated in favor of Extended Events per <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/default-trace-enabled-server-configuration-option?view=sql-server-2017>.

## 4.5.0 Blocked Process Threshold

Setting “1569: blocked process threshold” determines how often blocked process reports are generated. The \_\_\_\_ configuration of “0” means those reports are not generated. The Azure SQL DB configuration of “20” means those reports are generated every 20 seconds. Generating the reports will consume some system resources, but it cannot be disabled in Azure SQL DB per <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/blocked-process-threshold-server-configuration-option?view=sql-server-2017>.

## 4.6.0 Agent XPs

Setting “1587: Agent XPs” determines whether or not the SQL Server Agent extended stored procedures are available. This setting cannot be changed on Azure SQL DB, but should not affect performance. You can read more here: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/agent-xps-server-configuration-option?view=sql-server-2017>.

## 4.7.0 SMO and DMO XPs

Setting “16384: SMO and DMO XPs” determines whether or not the SQL Server Management Object extended stored procedures are available. DMO XPs have not been available beginning in SQL Server 2012. This setting cannot be changed on Azure SQL DB, but it should not affect performance. You can read more here: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/smo-and-dmo-xps-server-configuration-option?view=sql-server-2017>.

## 4.8.0 Filesystem Enumeration

Setting “16398: allow filesystem enumeration” allows for enumeration of DFS which did not enter into this test scenario. This setting cannot be changed on Azure SQL DB, but should not affect performance.