Technical Report

# Deploying MSSQL Database Workloads on NetApp HCI

## Quants and Claims for MSSQL Workloads

Allen Cao, NetApp
November 2020 | TR-4870

## Abstract

This report describes why NetApp® HCI is the best solution for hosting Microsoft SQL Server database workload for NetApp sales representatives, sales engineers, interested NetApp partners, and customers.

We report lab tests on a NetApp HCI platform and a competing conventional HCI platform that demonstrate NetApp HCI advantages in database performance at scale; resilience in maintaining SQL performance; availability during maintenance, node failure, and recovery, and compelling TCO advantages derived from eliminating the HCI tax.

# Quants and Claims for MSSQL

Sowing fear, uncertainty, and doubt (FUD) is a common sales tactic that is countered by NetApp HCI quants and claims. In this technical report, we define the competitive claims that differentiate SQL Server 2017 on NetApp HCI.

## Customer Value

Each claim is supported with quantified test cases verified on competitive hardware by NetApp engineers. With these verified competitive claims, you will be able to address FUD with confident statements about NetApp HCI differentiation for SQL Server 2017. You will find these claims applicable and extendable to all SQL type databases.

For application owners, this paper will help you understand:

- The value of independently scaling compute and storage
- How to uncover hidden costs
- The real cause and effects of the HCI tax

Sales personnel reading this paper will be able to:

- Know how, why, and when we stand out and be prepared to address market FUD
- Have confidence, founded on verified test data, in the differentiation of NetApp HCI solutions and customer value
- Sell with boldness while forcing the competition to respond from a defense position with your customers
- Identify and understand the competitive claims and differentiators for SQL Server on NetApp HCI as compared to other HCI offerings in the market

SE personnel reading this paper will be able to:

- Provide technical background for our claims
- Answer sales and customer questions
- Understand the technical quantifications that support the competitive claims and the efforts undertaken to verify the quantifications on NetApp HCI and competitive product offerings
- Be a reference of compelling competitive claims that can help counter doubts posed by competitive sales teams

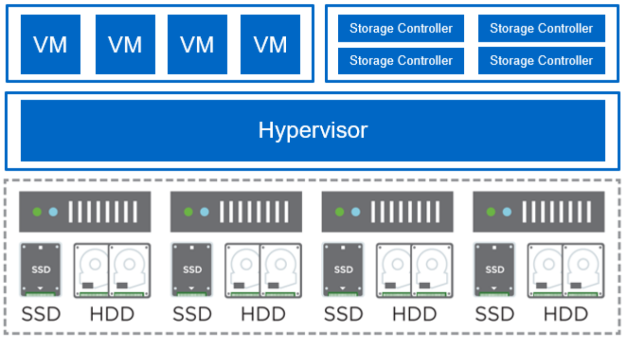
# Use Cases

NetApp HCI is designed to deliver superior value for enterprise applications such as SQL Server databases. This paper highlights these advantages and defines solution-level competitive differentiation against conventional HCI platforms.

Certain terms specific to this technical report are defined as follows:

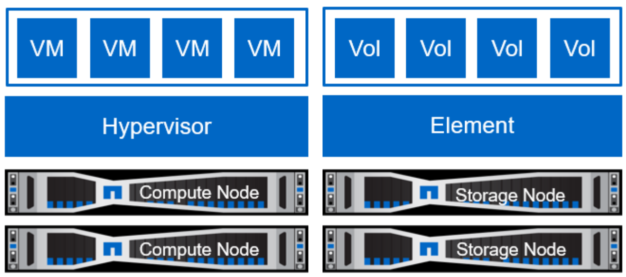
- **Conventional HCI.** HCI architecture with storage integrated within the compute node. HCI controller software requires vCPU resources that consume a definable percentage of the total compute resources available for applications. Therefore, they cannot scale compute and storage independently.

  The following figure depicts a high level conventional HCI platform.




- **NetApp HCI.** NetApp HCI architecture enables independently scale of compute and storage. HCI controller software and storage management runs on dedicated CPU cores on a storage node and does not consume any compute resources on the compute node.

  The following figure presents a high-level NetApp HCI system.




- **HCI Tax.** An HCI tax is the inability to scale compute and storage resources separately, which typically results in overbuying storage or compute resources. If an application is compute-hungry but does not need more storage, adding a new node means increasing storage capacity that is not needed. Likewise, if a data-intensive application requires HCI storage expansion, the new node includes unneeded compute capacity.

  An additional HCI tax is the overhead needed to run the storage service on an HCI compute node because there are no dedicated compute resources for the storage controller in conventional HCI.

## Target Audience

The target audience for this solution includes application owners, sales personnel, and SE personnel.

The following points can help you understand how to use this document:

- All competitive claims and quantifications are tested on NetApp HCI and competitive hardware. However, neither the vendor nor the product is identified.
- The Executive Summary is a quick sales-level reference. This section contains a summary of the NetApp HCI SQL 2017 solution's competitive claims and the quantifications that support each claim. These claims are applicable for any SQL server solution for NetApp HCI.
- The section "NetApp HCI Architecture Competitive Advantages" describes NetApp HCI architectural competitive advantages.
- The section "NetApp HCI for MSSQL Workloads" contains quantification details for engineers, but others will find methodology and detailed data here that justify NetApp claims and help them uncover hidden costs.
- The section "HCI Tax and TCO" is applicable to both sales and SE. Here you will find the financial values and quantifications that explain the HCI tax and why NetApp HCI delivers financial differentiation.

- The section “Technology and Methodology for Tests Validation” details how we tested in our lab to support various claims.
- The Conclusion summarizes competitive advantage claims for quick reference.

# Executive Summary

## Performance at Scale

Only NetApp HCI delivers consistent performance at scale for SQL Server across diverse use cases with predictable performance guarantees even as database and application instances increase within the HCI platform.

By not storing data in any compute node, NetApp HCI enables large number of databases in any VM on any compute node while maintaining a uniform performance level.

Competing platforms falter in database performance when deploying databases on nodes (or VMs running on nodes) that access data from another node. In conventional HCI, database performance is not the same when data is remote rather than local, and the I/O performance for a SQL Server database is poor at the cluster level.

| CLAIM #1 | Quantification: Section 5.1 |
| --- | --- |
| Only NetApp HCI delivers true performance at scale for SQL Server across diverse use cases with predictable performance guarantees even as database and application instances increase within the HCI platform. | On average, NetApp HCI delivered more than 30% better OLTP transaction performance than conventional HCI when you scale up the number of SQL Server databases on a single HCI node from one to three.<br>On average, NetApp HCI OLTP transaction performance is 26% better than conventional HCI when the number of SQL Server databases is scaled from one to six on two HCI nodes. |

## Performance Resilience

Resilience is important, especially during planned and unplanned maintenance. Only NetApp HCI seamlessly maintains SQL performance and availability during maintenance, VM migrations, node failure, or recovery.

NetApp HCI stores data in dedicated storage nodes, providing equal network bandwidth from all compute nodes for access to all copies of the data. In the case of a compute node failure, database VMs are migrated to another compute node without compromising data-access bandwidth.

In competitive HCI, performance suffers during node failure or when a VM is migrated by an admin to achieve compute-resource balancing to a node that must access data from another node. Due to these architectural disadvantages, skilled administration efforts are needed to tune the performance after every migration or failure scenario.

| CLAIM #2 | [Quantification: Section 5.2] |
|---|---|
| Only NetApp HCI seamlessly maintains SQL performance and availability during maintenance, node failure, and recovery. | During simulated maintenance via vMotion to move a DB VM from one node to another, SQL Server transaction performance was 16% better with NetApp HCI during a brief 10 to15 minute window while vMotion ran its course.<br>With completion of vMotion, the SQL transaction performance recovered fully with NetApp HCI while competitive HCI had a persistent SQL transaction performance reduction that averaged 9% less throughout the test run.<br>For a simulated node failure in which all three database VMs on a node were transferred via vMotion to another node, competitive HCI had as much as a 45% transaction rate drop off. The average SQL Server transaction performance hit with various numbers of vUsers was about 10% lower with a loss of data locality. NetApp HCI had zero SQL Server transaction performance effect because the DB server connects to central storage via direct iSCSI.<br>During node failure, NetApp HCI was at minimum three time faster in establishing data redundancy than competitive HCI in a validated test case. |

### HCI Tax and TCO

The TCO is important for any business, and software licensing and product design features can create hidden cost overhead that affects the TCO. Competitive HCI can create hidden HCI taxes for running the storage service via a virtual storage VM on every HCI node as well as the inability to scale out compute or storage independently.

| CLAIM #3 | [Quantification: Section 5.3] |
|---|---|
| NetApp HCI eliminates the HCI tax and delivers compelling financial differentiation for SQL Server 2017. | The storage VM CPU overhead for conventional HCI running SQL Server in a single node was as high as 34% and above 20% on average.<br>When heavy SQL Server workload is deployed to all nodes in a conventional HCI cluster, the average CPU overhead was over 40% of cluster capacity. This contributes as much as $48,000 in hidden software licensing costs per node.<br>Compared with NetApp HCI for SQL Server workloads, adding a node to cluster for storage capacity results in an extra software licensing cost of ~$122,000 with conventional HCI. Adding a node to cluster for compute capacity on the other hand results in extra hardware costs of ~85% for a conventional HCI node. |

## NetApp HCI Architecture Competitive Advantages

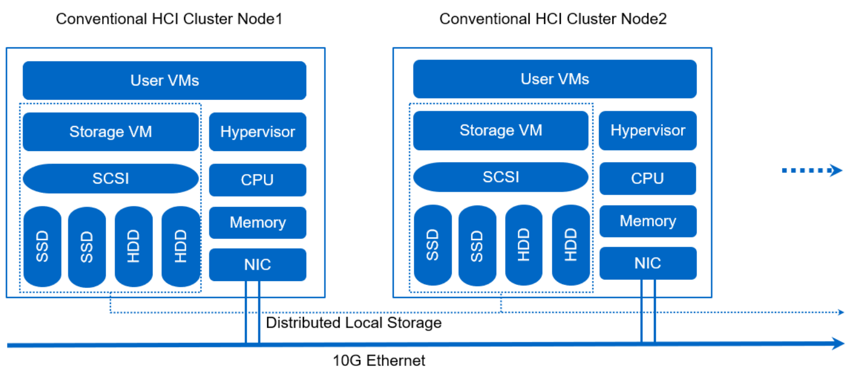
### Solution Technology

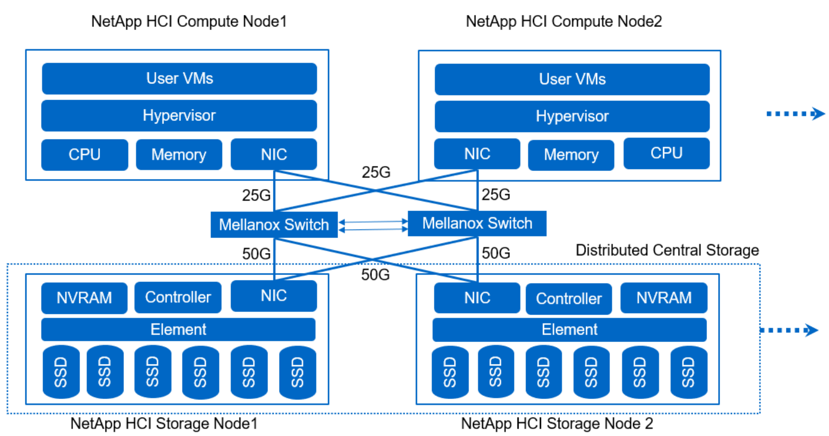
A typical HCI system is tightly packed with computing, networking, and storage components to create a functional infrastructure system to support business applications. It is easy to deploy, consume, and manage. However, not all HCI systems are created equal. The goal of this study is to validate NetApp HCI platforms performance advantages relative to a conventional HCI platform.

At the infrastructure level, a conventional HCI platform consists of number of nodes with integrated compute, network, and local storage. Through a software-enabled storage VM, the local storage is pooled together and shared among many cluster nodes. As the cluster scales out horizontally, compute and storage are proportionally added to the cluster.

NetApp HCI on the other hand is a new category of HCI that IDC describes as disaggregated HCI, in that compute and storage can be added to the cluster disproportionally. The NetApp HCI storage node is centrally managed through a software-defined storage controller called NetApp Element® software. It combines performance and capacity on individual storage nodes into an homogenized storage fabric and evenly distributes storage workload among all available storage nodes.

The following architecture diagrams depict the architecture design differences between a conventional HCI platform (first diagram) and the NetApp HCI platform (second diagram).




Although virtualized compute and networking are very similar among all HCI platforms, the implementation of software-defined storage is significantly different.

In the diagram, the software-defined storage control services for conventional HCI are delivered through a storage VM on top of the hypervisor, which is in direct competition for resources with application VMs in an HCI node. NetApp HCI has dedicated compute resources for storage operations. It is not sharing any resources with user VMs in a compute node.

Many more differences between conventional HCI and NetApp HCI are summarized in the following table.

| HCI Platform → Features ↓ | Conventional HCI | NetApp HCI |
|---|---|---|
| Scale out | Proportional compute and storage | Independent compute and storage |
| Storage virtualization | Distributed local storage | Distributed central storage |
| Storage type | Hybrid SSD/HDD, SSD | SSD |
| Storage caching | SSD | NVRAM |
| Storage controller | Distributed storage VMs | Distributed storage software (Element) |
| Dedicate compute for Storage controller | No; shared compute with other VMs | Yes |
| Data distribution | Primary data copy locally | Even distribution of all data copies within cluster |
| Networking | Isolated network for local storage<br>Shared network for all cluster traffic, typically 10G | Dedicated application 25G<br>Dedicated storage networks 50G |
| Compression | Local, compute overhead | Global at storage node, no compute overhead |
| Deduplication | Local, compute overhead | Global at storage node, no compute overhead |
| Monitoring | On prem | Cloud-based SAAS Cloud Insight |

For database workloads, the key difference between conventional HCI and NetApp HCI is shared versus dedicated networking. Although conventional HCI uses an isolated network for local storage access, it relies on a single shared network link to carry all cluster traffic, such as database I/O, replication, compression, deduplication, backup, rebalancing, rebuilding, data lifecycle services, and so on.

To overcome the limitations of a single shared network link, conventional HCI stores the primary copy of data locally at the node where the database is installed to reduce data traffic. It is often referenced as the data locality feature for conventional HCI. Data locality improves performance because it reads and writes on the local copy of primary data. Therefore, it can deliver respectable performance in a small environment with small databases when workloads can be locally limited and cluster traffic is minimal. However, a real-life database workload in a shared infrastructure does not follow these limitations. As the database environment scales up and out, the shared network link can be overwhelmed, and the database I/O performance can drop off dramatically.

The unintended consequences of data locality are uneven distribution of data within a cluster that results in difficulty maintaining performance when the hosted database grows. Some HCI nodes could run hot while others run cold. Although virtualization or manual intervention can eventually balance the workload, the rebalancing could cause database performance to suffer because data locality must be re-established when a SQL Server VM is relocated for workload rebalancing. There is also a performance penalty during node failure or maintenance when database VMs are relocated to the other nodes, resulting in the loss of data locality.

The other drawback for the uneven distribution of data is the inefficient use of resources as the cluster scales out when database run out of capacity for either compute or storage at the local node, causing resource lockup or waste. This contributes to the HCI tax and higher TCO for operating your MSSQL environment.

NetApp HCI employs dedicated application and storage networks to provide high network throughput. Element software distributes data storage evenly using a random hash feature to ensure uniform data distribution among the storage cluster nodes and disks. As an incoming write occurs, the data is broken down into 4K blocks. A block ID is computed for each of these 4K blocks. If the data already exists, Element eliminates the need to write the data a second time. The data is then compressed and written to NVRAM on the primary node and replicated to the secondary node before sending a write acknowledge to the host. The primary node controls all iSCSI sessions for that volume.

In summary, although all HCI platforms use a distributed storage fabric to balance data, there are significant differences in how data distribution is accomplished. For database workloads that typically moves around large amount of data, there is a significant performance effect as a result. NetApp HCI is the only true evenly distributed platform for data storage. There is no potential network choke point or data locality, thus guaranteeing lower latency and linear scalability in I/O performance.

By conducting tests side by side in a similarly configured lab environment, this paper quantifies the differences between these two platforms in terms of database performance and TCO as it applies to Microsoft SQL Server workloads.

# NetApp HCI for MSSQL Workloads

## 5.1 SQL Server Performance at Scale

In our lab tests, we set up the equivalent of a four-node HCI cluster for each platform. We ran our load generators and management services on two HCI nodes and dedicated two HCI nodes as database servers for SQL Server.

To measure SQL Server performance at scale, we first tested how the SQL Server instance performed as the number of SQL Server databases scaled up from one to three within an HCI node. We then tested how the SQL Server performed as the number of SQL Server databases scaled out from one to six onto two HCI nodes.

We ran simulated OLTP (TPC-C) workloads via HammerDB with concurrent number of virtual users range from 1 to 128 on a SQL Server database using auto pilot in each test. This simulated SQL Server transaction performance in a gradually increasing workload with increasing numbers of concurrent virtual users.

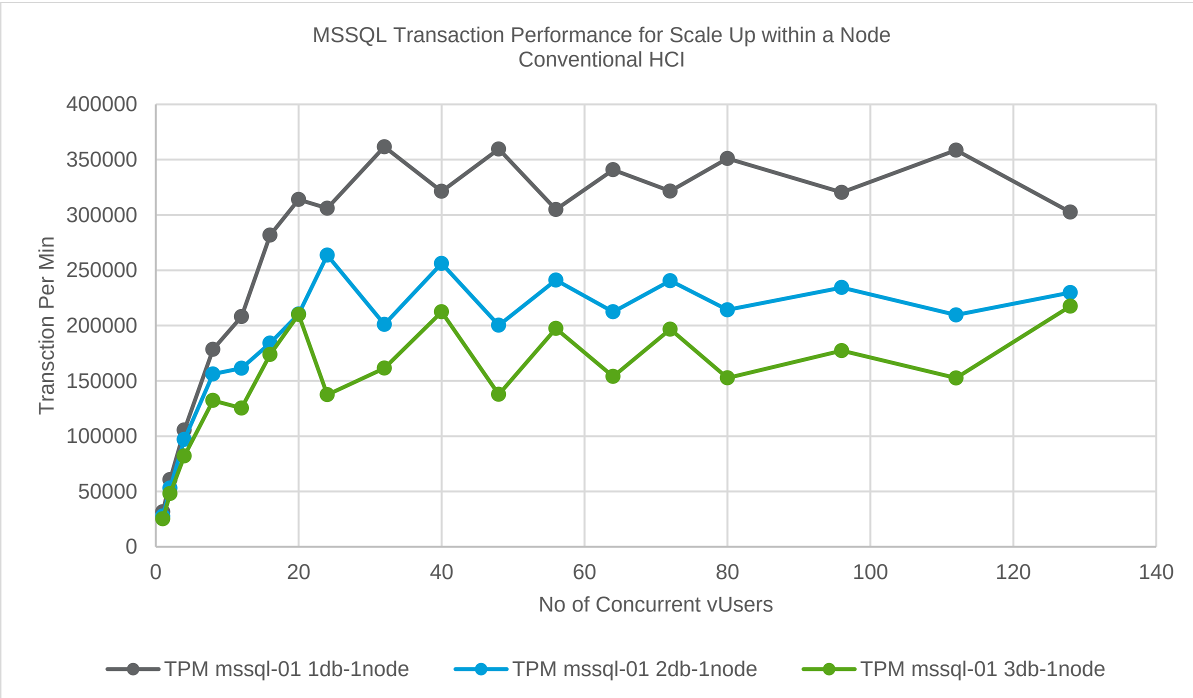
Each test-run ran for three hours. Each concurrent user level ran for 10 minutes and produce two OLTP performance metrics: transaction per minutes (TPM) and new order per minute (NOPM). The numbers were collected from SQL Server via the HammerDB client at each 10 minutes interval. We also measured storage I/O performance in terms of IOPs and latency for each corresponding run for comparison.

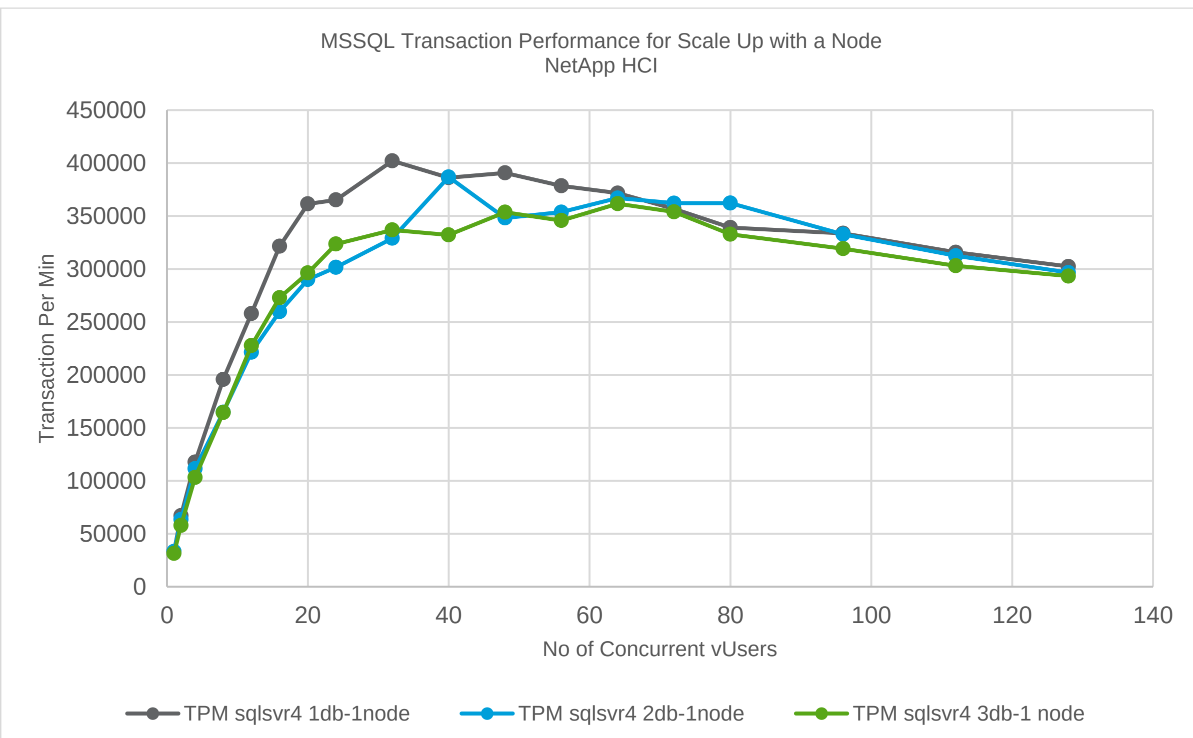
We charted the performance test results from test runs on a single platform against other test runs on the same platform to validate the relative performance as the number of databases or vUsers on the platform scaled up or out. We then compared the relative performance between two platforms to determine which platform scaled better for SQL Server workloads. With this methodology, we eliminated the relevance of underlying hardware for the comparison.

### SQL Server Scale Up

The ability to scale up the number concurrent databases while maintaining performance within an HCI node empowers you to run more database workloads with given local hardware resources. We deployed identical simulated OLTP workloads on both platforms to measure database performance in terms of transaction volume per minute (TPM) with different number of concurrent databases hosted within a node.

For the scale-up test results comparison, the following two graphs illustrate the TPM for test runs on one, two, and three SQL Server databases within an HCI node for each platform. These test results demonstrated that NetApp HCI was much better at maintaining SQL Server transaction performance when scaling up within a node.



SQL Server OLTP transaction performance is a cumulative effect of the compute and storage of an HCI system. As you scale up the number of databases on an HCI node, some shared resources are in contention such as CPU scheduling, storage IOPS, queue, and so on. Therefore, it is reasonable to expect the transaction rate to come down as the number of hosted databases goes up.

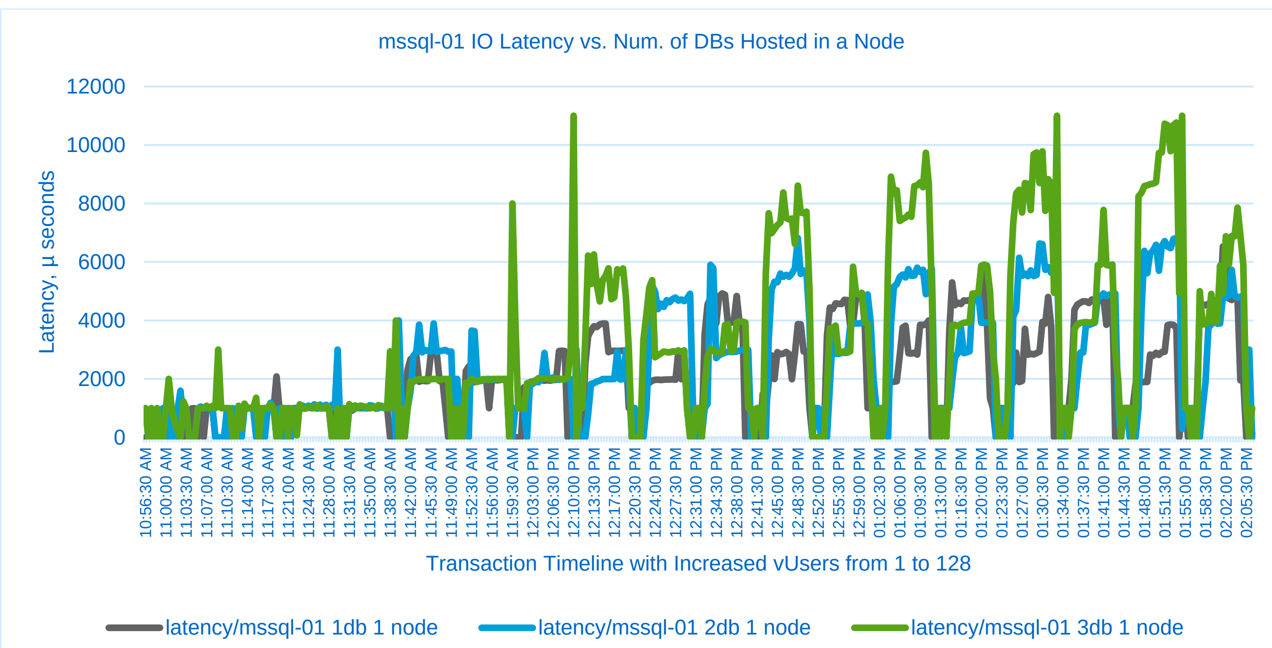
NetApp HCI delivered smooth and consistent transaction performance. The largest transaction rate drop-off for NetApp HCI as the number of databases scaled from 1 to 3 was 18.05%. The average transaction rate drop-off with a varied workload represented by the number of vUsers was 9.34%.

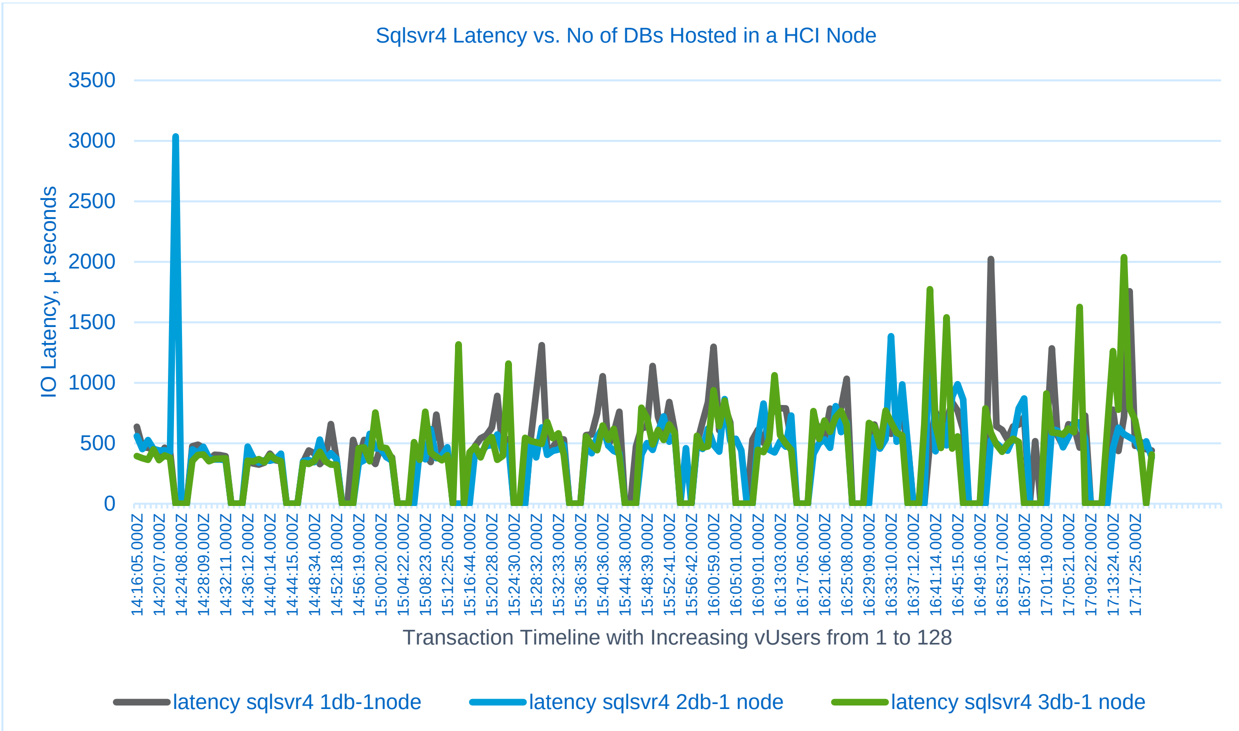
As a comparison, the transaction performance for conventional HCI was rather uneven and the largest transaction rate drop-off as number of databases scaled from one to three was 61.7%. The average transaction rate drop-off with a different number of vUsers was 40.1%. In fact, the minimum transaction rate drop-off for conventional HCI was 20.25%, which surpassed the maximum transaction rate reduction for NetApp HCI. On average, NetApp HCI delivered 30.76% (40.1-9.34) better performance when scaling up MSSQL on a single HCI node. The following table presents the relative SQL transaction performance comparison for scale up in an HCI node.

| | Average Transaction Rate Reduction with number of SQL Server DB scale up from 1 to 3 in a node | Relative Transaction Performance Comparison |
|---|---|---|
| NetApp HCI | 9.34% | 30.76% Higher |
| Conventional HCI | 40.1% | |

Database I/O performance dictates the transaction rates. We measured I/O latency for the corresponding runs to validate I/O performance. The lower the latency, the faster a transaction can be committed and, as a result, the larger the transaction volume.

The following two graphs show that the I/O performance for the corresponding transaction runs for each platform. The first graph shows the results for conventional HCI, and the second graph shows the results for NetApp HCI. The I/O performance results in terms of latency showed that NetApp HCI I/O performed much better during the transaction runs.





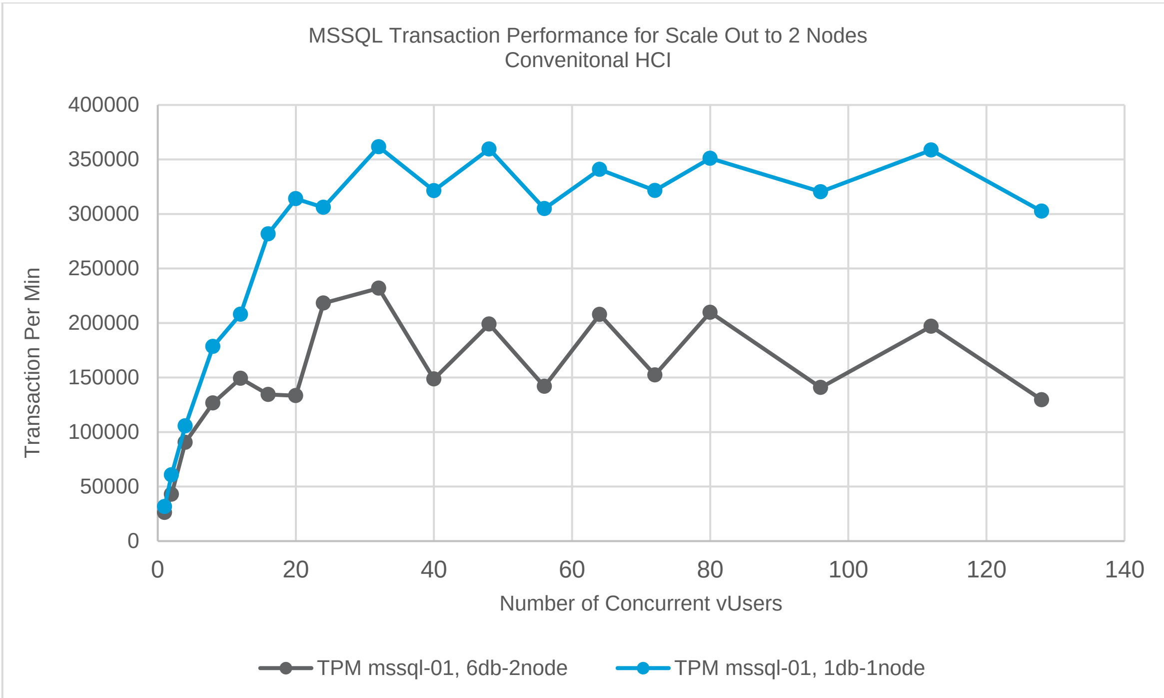

For most of the transaction runs, NetApp HCI I/O latency was under one millisecond with brief spikes up to two milliseconds. The exceptional storage I/O performance of NetApp HCI helped SQL transaction rates remain at a high level as the number of databases scaled up in an HCI node.

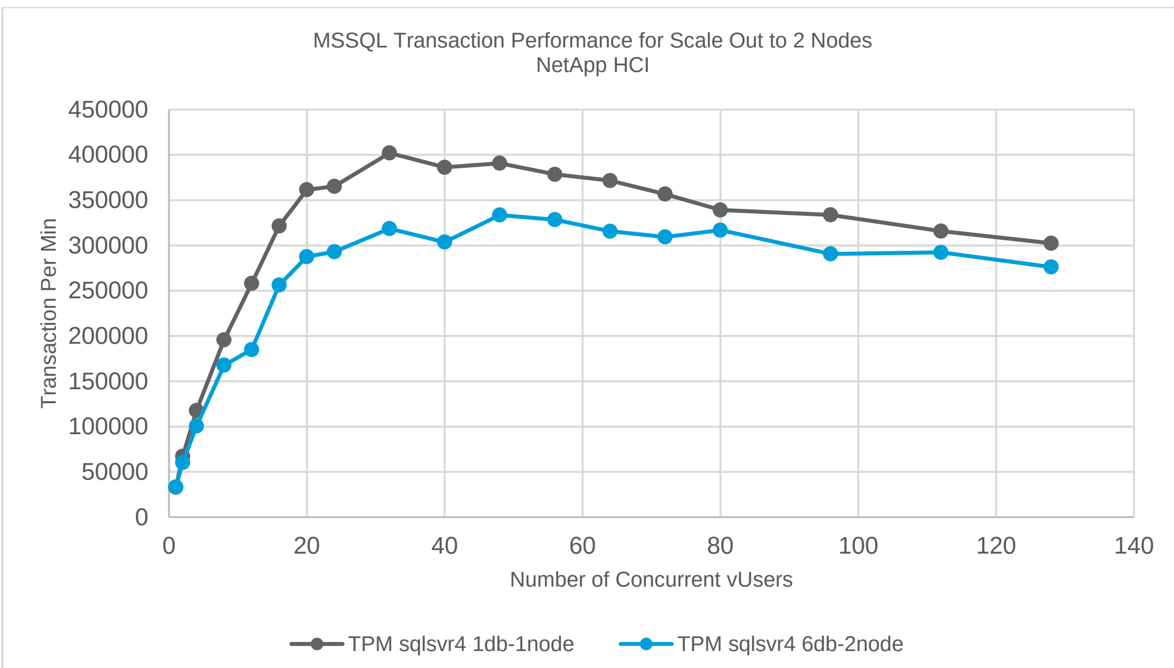
On the other hand, conventional HCI I/O latency was consistently over two milliseconds, even with minimal workload. As the workload increased when scaling up the number of vUsers, the I/O latency spiked much higher and remained elevated at up to 11 milliseconds. On average, conventional HCI I/O latency was four to seven times higher. This helped to explain why the SQL transaction rates on conventional HCI dropped off rather dramatically as the number of databases and vUsers scaled up in an HCI node.

## SQL Server Scale Out

For the MSSQL scale-out performance comparison, we deployed six SQL Server databases on two HCI nodes. Each node hosted three SQL Server VMs. We then ran identical workload via HammerDB auto pilot concurrently against six SQL Server databases. The scale-out tests measured the sustainability of shared resources in supporting SQL Server workloads at the HCI-cluster level.

The following two graphs illustrate how the same SQL Server performed on simulated OLTP transactions as the database environment scaled out from one DB on one node to six DBs on two nodes on both platforms. The first graph shows conventional HCI, and the second graph shows NetApp HCI. Again, the results demonstrated that NetApp HCI outperformed the competition by a large margin.

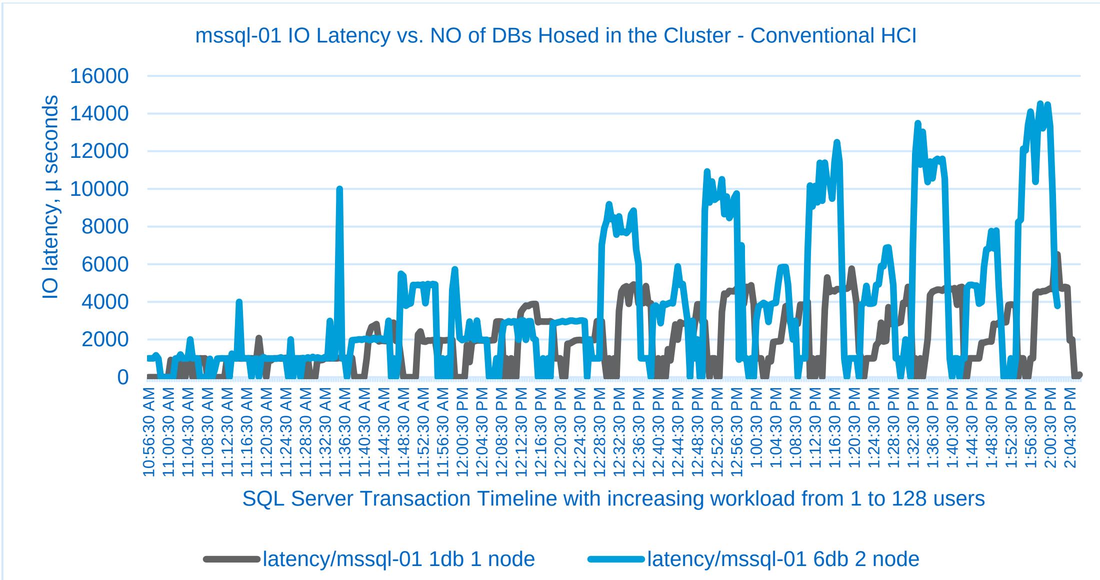


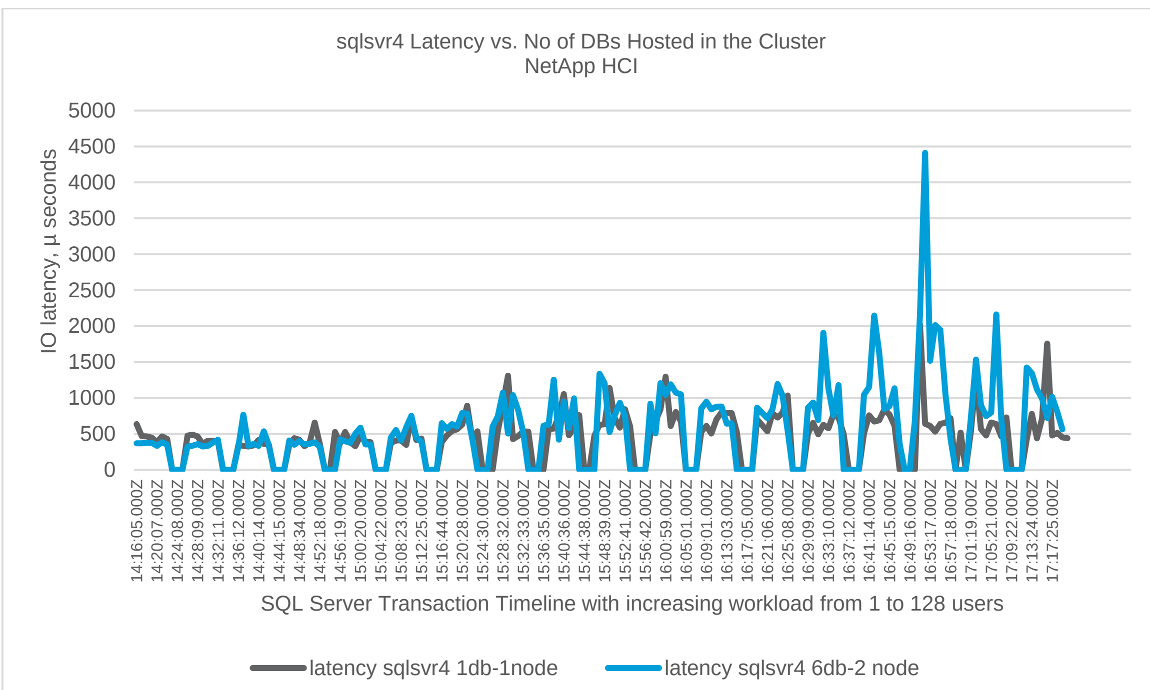




For scale-out performance, the average transaction rate drop-off for conventional HCI was 40.8% while the transaction rate reduction for NetApp HCI was 14.46%. The NetApp HCI relative transaction performance bested conventional HCI by 26.34% (40.8-14.46).

We also tallied SQL Server I/O performance for the respective test runs to measure the effect of resource sharing at the cluster level as is depicted in the next two graphs. The first graph shows conventional HCI, and the second graph shows NetApp HCI.

The results demonstrated that NetApp HCI was able to maintain I/O performance close to the level of a single-node with a single DB whereas conventional HCI I/O performance further deteriorated with increased workload at the cluster level.




When scaling out to six DBs on two cluster nodes, conventional HCI SQL Server I/O latency spiked as high as 14 milliseconds and remained elevated throughout the 10-minute transaction run at each workload level. For NetApp HCI, with exception of a brief spike to 4.5 milliseconds, I/O latency remained under 2 milliseconds even at a high concurrent user workload level. On average, conventional HCI I/O latency was more than six times that of NetApp HCI. This further validated the performance advantage of NetApp HCI in maintaining SQL Server transaction rate at high level with superior I/O performance.

## 5.2 SQL Server Performance Resilience

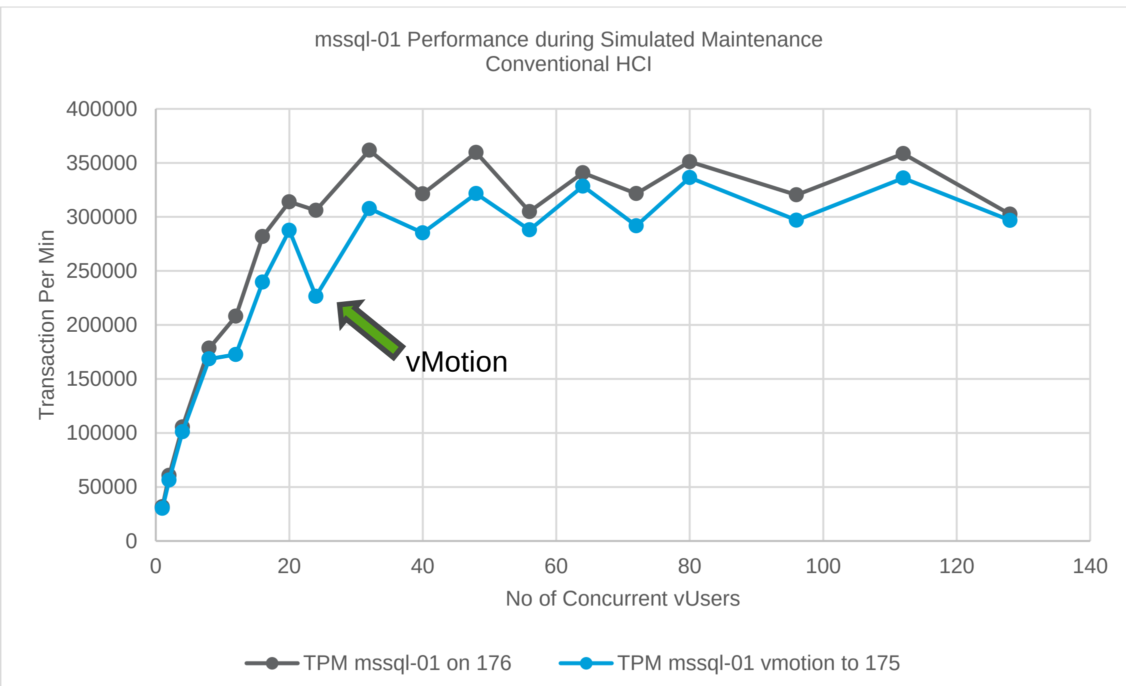
Having a high performing HCI system is import and so is the reliability and resilience of the platform.

To validate each HCI platform resilience in supporting SQL Server workloads, we simulated maintenance activity by an HCI admin to perform a vMotion operation to move a SQL Server VM from one node to another. We also simulated node failure in which all database VMs were moved with vMotion to a surviving HCI node. SQL Server performance during maintenance activity or the node failure scenario were then compared with baseline performance to quantify the effect of each activity.

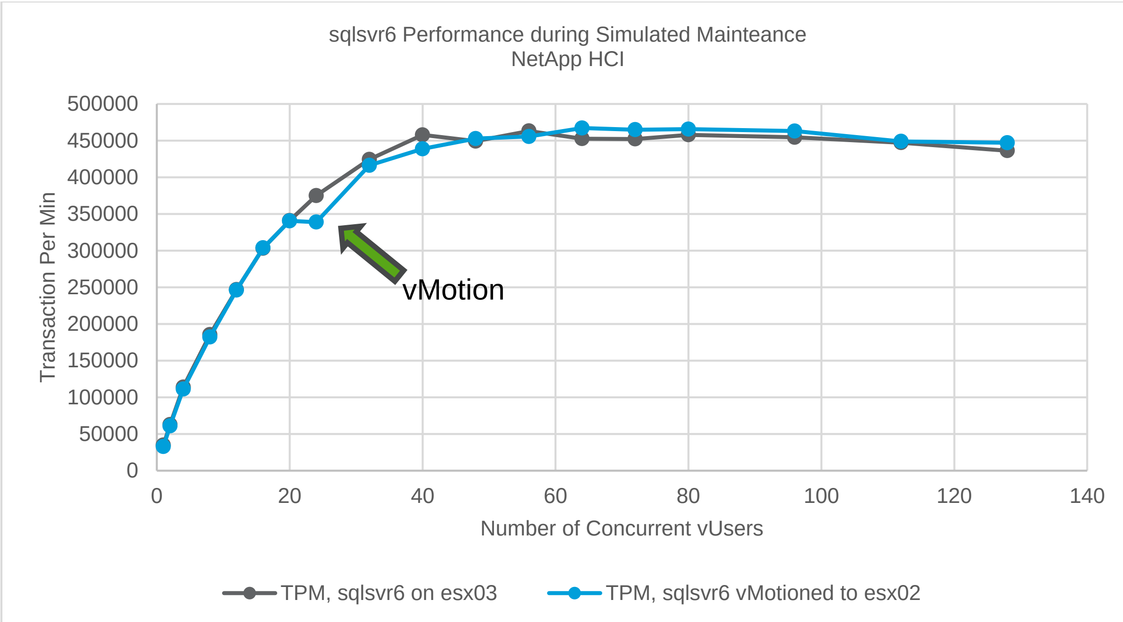
### SQL Server Performance During Maintenance

To simulate admin-triggered maintenance, we ran vMotion to migrate a SQL Server database in the middle of an auto pilot test run at 24 concurrent vUsers. The database server began the transactional run on one ESXi host and ended the transactional run on another ESXi host. The transactional performance was then compared with a full transaction run without interruption on the same database server and the same ESXi host.

The following two graphs demonstrate the SQL Server performance in TPM with and without vMotion during a transaction run. The first graph shows conventional HCI, and the second shows NetApp HCI. Ideally you want to achieve same SQL Server transaction performance during maintenance activity such as vMotion operations.






The test results show that there was a minimal effect on SQL Server performance for NetApp HCI during simulated maintenance, which caused a brief 9.6% drop in transaction volume. The performance hit was transient and lasted only about 10-15 minutes before full recovery with the completion of vMotion. Other than that, the transaction performance curves were nearly the same.

There was no app/client disconnection observed while vMotion took place. The NetApp HCI architecture dictates that vMotion activity triggered by maintenance has no effect beyond a brief hiccup while vMotion is in progress. After the completion of vMotion, SQL Server transaction performance was fully recovered. Because the database I/O was serviced from a centrally distributed storage pool via iSCSI connections, it did not matter where the iSCSI connections were initiated. The dual dedicated, redundant 50G paths iSCSI networks provided high levels of database performance during the maintenance activity.

For conventional HCI, moving a SQL Server VM via vMotion from one node to another had a greater effect on database performance relative to NetApp HCI. Beyond the effect of vMotion itself, there was a persistence transaction performance effect through the entire test run. The transaction performance did not recover fully after the completion of the vMotion operation. The primary reason was that conventional HCI relies on data locality to achieve performance. When a SQL Server VM was moved via vMotion to another node, data locality was lost. SQL Server had to service its user I/O requests from remote storage across a network link rather than from a cached local copy. There was also overhead to re-establish data locality at the new host.

The quantified performance effect in SQL Server TPM for conventional HCI was 26% lower during vMotion and transaction performance after completion of vMotion was 9.17% lower.

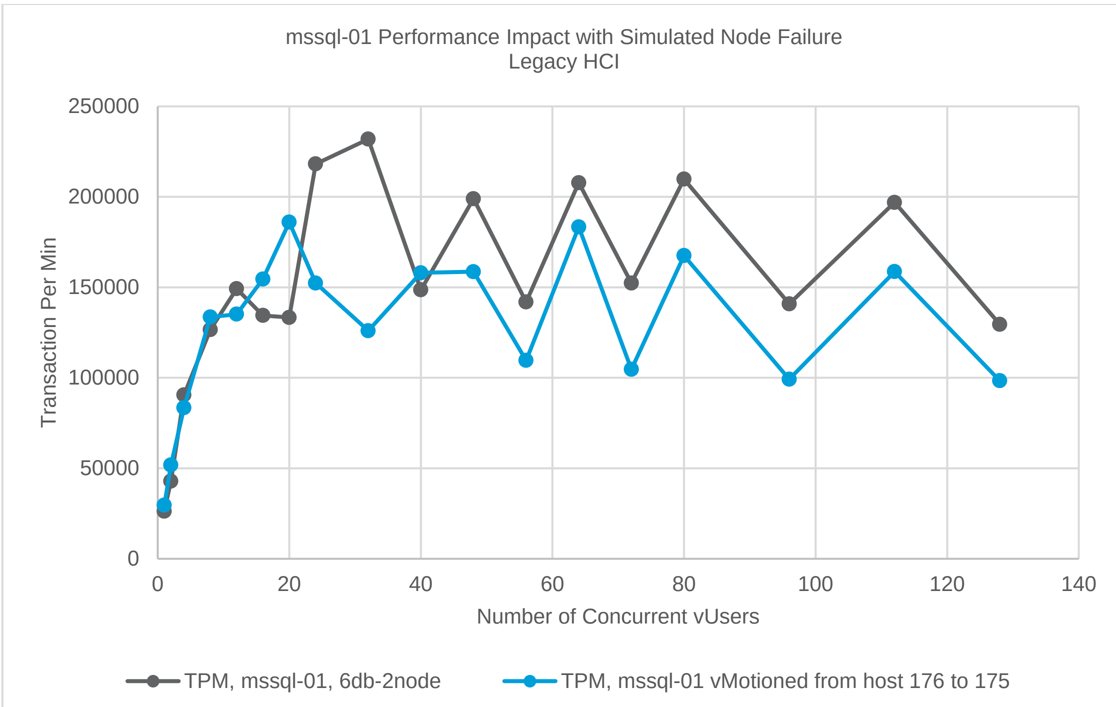
## SQL Server Performance with Node Failure

To simulate a node failure scenario, we moved all three SQL VMs on an HCI node to another node with a vMotion operation. Then we deployed identical HammerDB OLTP workloads on six concurrent SQL Servers with auto pilot to measure the performance effect of node failure within the cluster.

In general, a virtualized HCI platform has HA configured for important workloads like SQL Server in which SQL VMs on a failed node are restarted on a surviving node, and applications can reconnect to SQL server and resume transaction processing. Platform performance resilience hinges on how well SQL Server transaction processing proceeds on the survival node when compared with performance at the home node.

For NetApp HCI, the storage is centrally distributed. Therefore, there is no difference in SQL Server transaction performance when SQL VMs are restarted on another node in the case of node failure because SQL VMs reconnect to same central storage via iSCSI links. There are virtually no changes compared with before failure because the surviving node is equivalent to the home node in configuration and compute resources.

For conventional HCI, the situation is quite different in that SQL Server data is stored locally. When failed SQL VMs are restarted on a surviving node, SQL Server now needs to access its data through network compared with local data access before failure. The data locality effect is amplified when more concurrent SQL Server workloads are deployed within the cluster because the single network link limits transaction processing. This full performance effect on SQL Server transactions was quantified and demonstrated in in the following graph.



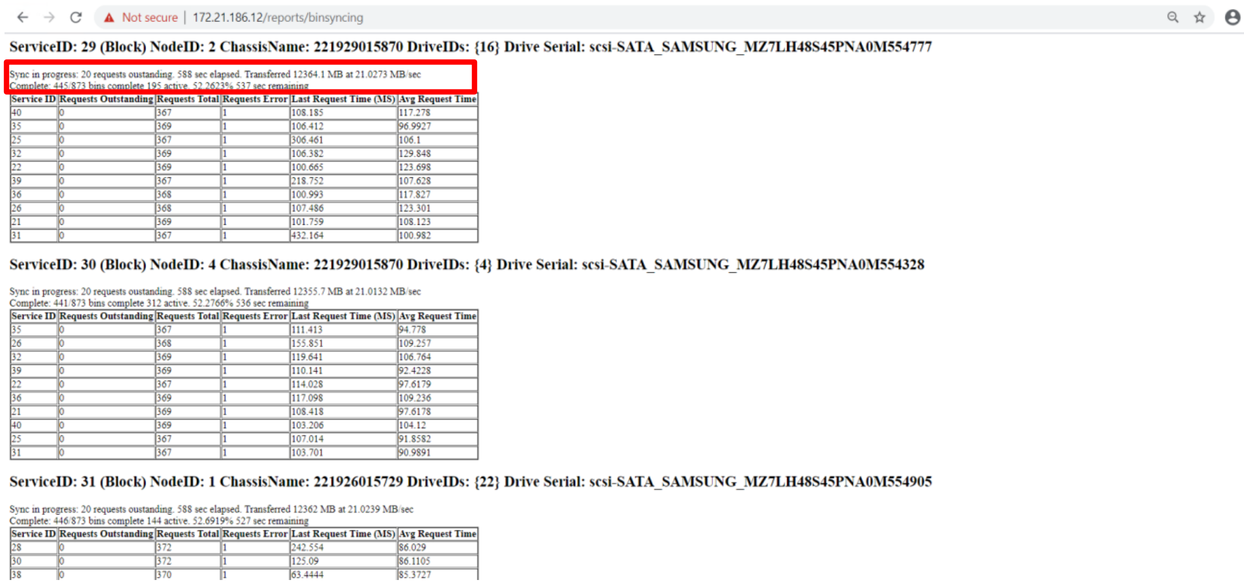

Here you can see that the transaction rate was reduced by as much as 45%. The average transaction performance hit or drop-off at all tested concurrent user levels was about 10% lower.

## Cluster Rebuilding with a Node Failure

Typical HCI stores a minimum of two data copies for redundancy in case of a node failure. When node failure occurs, a built-in automation algorithm rebuilds lost redundancy. This process is called cluster rebuilding. The cluster is vulnerable to permanent data loss if additional failure occurs while rebuilding is in progress. Therefore, the faster the cluster can complete its rebuilding, the more resilient the platform is for hosting database workloads.

To validate the NetApp Element cluster auto-healing capability, we added an additional two storage nodes to the cluster and loaded approximately 4TB of SQL Server data. We then intentionally shut down storage node 6 to trigger a simulated failure scenario and captured how the cluster recovered from the failure by rebuilding.

From the web API interface, we monitored that the failure immediately triggered the rebuilding process. It took approximately 18 mins to redistribute around 652G SQL data on node 6 to the other nodes as demonstrated in the following screenshot.



172.21.186.12/reports/binsyncing

**ServiceID: 29 (Block) NodeID: 2 ChassisName: 221929015870 DriveIDs: {16} Drive Serial: scsi-SATA_SAMSUNG_MZ7LH48S45PNA0M554777**

Sync in progress: 20 requests oustanding. 588 sec elapsed. Transferred 12364.1 MB at 21.0273 MB/sec
Complete: 445/873 bins complete 195 active. 52.2623% 537 sec remaining

| Service ID | Requests Outstanding | Requests Total | Requests Error | Last Request Time (MS) | Avg Request Time |
|---|---|---|---|---|---|
| 40 | 0 | 367 | 1 | 108.185 | 117.278 |
| 35 | 0 | 369 | 1 | 106.412 | 96.9927 |
| 25 | 0 | 367 | 1 | 306.461 | 106.1 |
| 32 | 0 | 369 | 1 | 106.382 | 129.848 |
| 22 | 0 | 369 | 1 | 100.665 | 123.698 |
| 39 | 0 | 367 | 1 | 218.752 | 107.628 |
| 36 | 0 | 368 | 1 | 100.993 | 117.827 |
| 26 | 0 | 368 | 1 | 107.486 | 123.301 |
| 21 | 0 | 369 | 1 | 101.759 | 108.123 |
| 31 | 0 | 367 | 1 | 432.164 | 100.982 |

**ServiceID: 30 (Block) NodeID: 4 ChassisName: 221929015870 DriveIDs: {4} Drive Serial: scsi-SATA_SAMSUNG_MZ7LH48S45PNA0M554328**

Sync in progress: 20 requests oustanding. 588 sec elapsed. Transferred 12355.7 MB at 21.0132 MB/sec
Complete: 441/873 bins complete 312 active. 52.2766% 536 sec remaining

| Service ID | Requests Outstanding | Requests Total | Requests Error | Last Request Time (MS) | Avg Request Time |
|---|---|---|---|---|---|
| 35 | 0 | 367 | 1 | 111.413 | 94.778 |
| 26 | 0 | 368 | 1 | 155.851 | 109.257 |
| 32 | 0 | 369 | 1 | 119.641 | 106.764 |
| 39 | 0 | 369 | 1 | 110.141 | 92.4228 |
| 22 | 0 | 367 | 1 | 114.028 | 97.6179 |
| 36 | 0 | 369 | 1 | 117.098 | 109.236 |
| 21 | 0 | 369 | 1 | 108.418 | 97.6178 |
| 40 | 0 | 369 | 1 | 103.206 | 104.12 |
| 25 | 0 | 367 | 1 | 107.014 | 91.8582 |
| 31 | 0 | 367 | 1 | 103.701 | 90.9891 |

**ServiceID: 31 (Block) NodeID: 1 ChassisName: 221926015729 DriveIDs: {22} Drive Serial: scsi-SATA_SAMSUNG_MZ7LH48S45PNA0M554905**

Sync in progress: 20 requests oustanding. 588 sec elapsed. Transferred 12362 MB at 21.0239 MB/sec
Complete: 446/873 bins complete 144 active. 52.6919% 527 sec remaining

| Service ID | Requests Outstanding | Requests Total | Requests Error | Last Request Time (MS) | Avg Request Time |
|---|---|---|---|---|---|
| 28 | 0 | 372 | 1 | 242.554 | 86.029 |
| 30 | 0 | 372 | 1 | 125.09 | 86.1105 |
| 38 | 0 | 370 | 1 | 63.4444 | 85.3727 |

This very fast rebuilding can be attributed to the even distribution of data (more disks for rebuilding) within the cluster as well as a dedicated storage network with as much as five times the throughput bandwidth compared with a typical shared 10G ethernet configuration.

We also tested the rebuilding capability of conventional HCI for comparison by shutting down one of HCI node from the hypervisor. The built-in automation for conventional HCI did not start rebuilding immediately. Instead, it delayed the cluster rebuilding process by going through a node detaching process. It would not start the rebuilding process until a failed node was fully detached from cluster. The logic behind the node detaching was likely to minimize data traffic within the cluster. We observed it took about one hour before a failed node was fully detached from the cluster.

Without factoring into the actual time it took for conventional HCI to complete the rebuilding process, NetApp HCI was at minimum three time faster at establishing data redundancy during failure for the validated test cases.

## 5.3 HCI Tax and TCO Evaluation

The HCI tax includes wasted compute or storage resources not needed when scaling out a conventional HCI cluster as well as the overhead from running the storage controller function with storage VMs in a conventional HCI cluster. Storage VMs have priority reservation and consume a large portion of CPU cycles. Because SQL Server is normally licensed on the number of CPU cores in an HCI node or cluster, a certain portion of SQL Server licensing cost is wasted for storage services instead of running SQL Server.

NetApp HCI on the other hand eliminates all HCI tax because you can individually scale compute and storage as needed. NetApp HCI uses dedicated compute for storage services within isolated storage nodes. Application workloads such as SQL Server are isolated from compute on the storage node. Therefore, those dedicated storage cores are not licensable for SQL Server services.

In this section, we quantify the HCI taxes that are attributable to storage VM CPU overhead as well as HCI cluster scale-out for either compute or storage and their contribution to TCO.

### Storage VM Compute Overhead

Conventional HCI relies on a storage VM on each HCI node to handle storage I/O services. This puts the storage VM in direct competition for CPU cycles with applications such as SQL Server hosted on the same node.

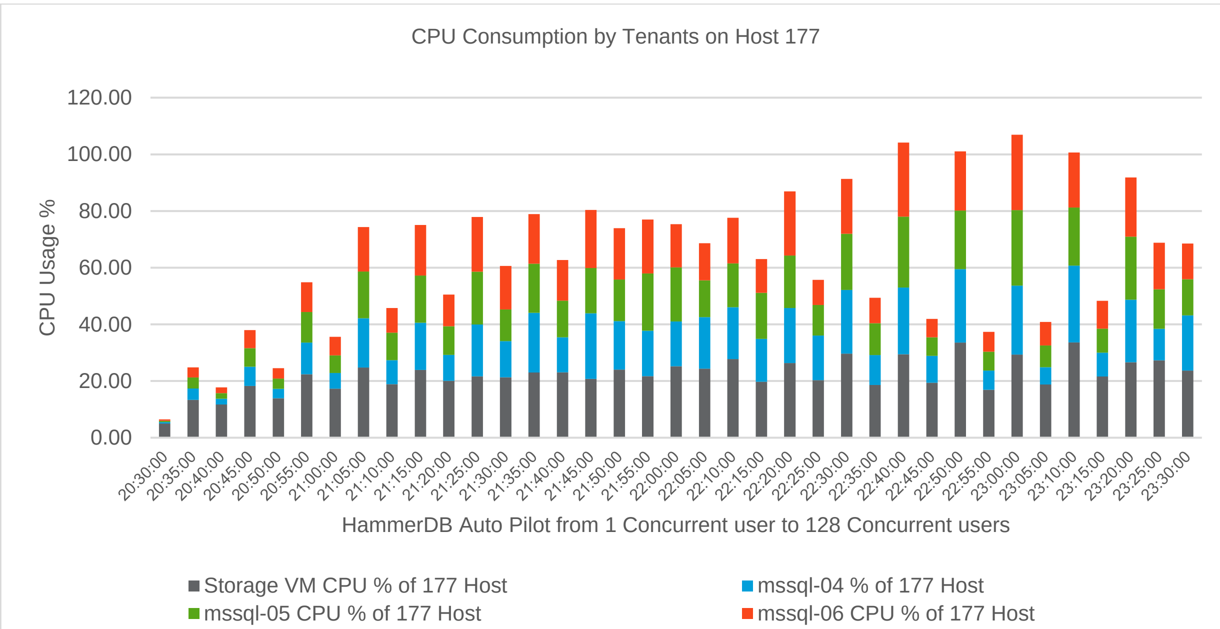
 

It is meaningful to quantify the cost overhead for the storage VM when deploying SQL Server workloads on conventional HCI. For that purpose, we deployed simulated SQL Server OLTP workloads on three concurrent SQL Server databases running on a conventional HCI node. From VMware, we measured CPU cycles consumed by all tenants on the node, which includes three SQL Servers and the storage VM. VMware reported CPU consumption by each VM in megahertz. We then calculated the total capacity of CPU cycles in megahertz for an HCI node using the following formula:

- Total CPU cycles in megahertz = (number of sockets) x (number of cores/socket) x (clock speed in MHz/core)

We calculated the CPU consumption for each tenant during a test run as a percentage of the total CPU capacity in a node by dividing the reported CPU consumption by a tenant by the total CPU capacity. The quantified storage VM CPU use provided a measure of the proportion of CPU cycles in an HCI node or cluster that was wasted for non-SQL processing.

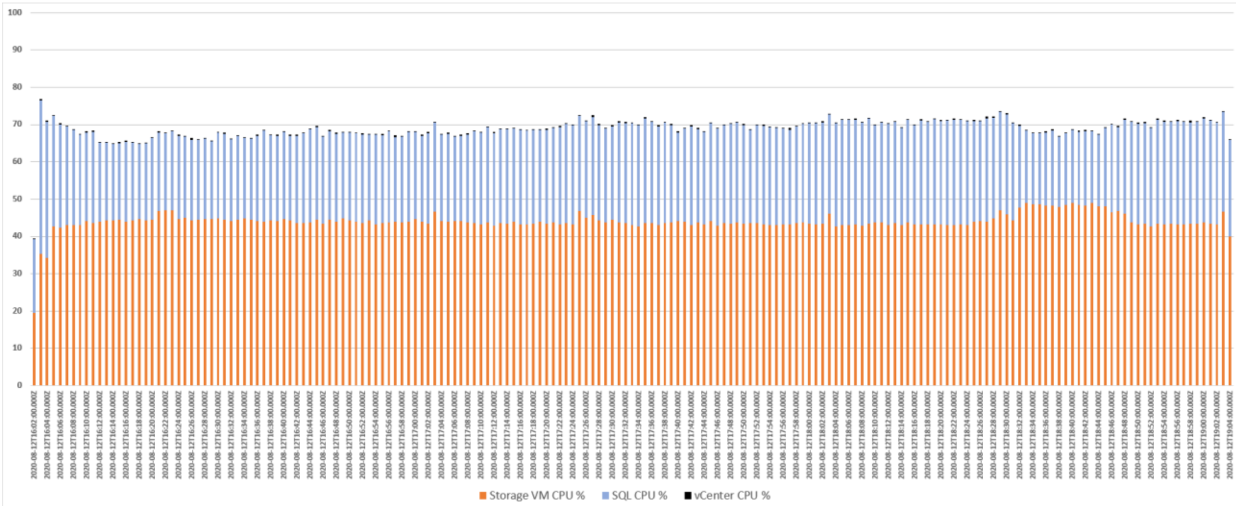
The following graph illustrates CPU consumption by each tenant in an HCI node hosting three SQL Server VMs during a test run.




The formula we used to calculate CPU consumption did not figure in hyperthreading, which can normally provide a 15% boost on the CPU capacity in a dual socket system. Therefore, we saw some data points exceeded 100%, which was the net effect of hyperthreading.

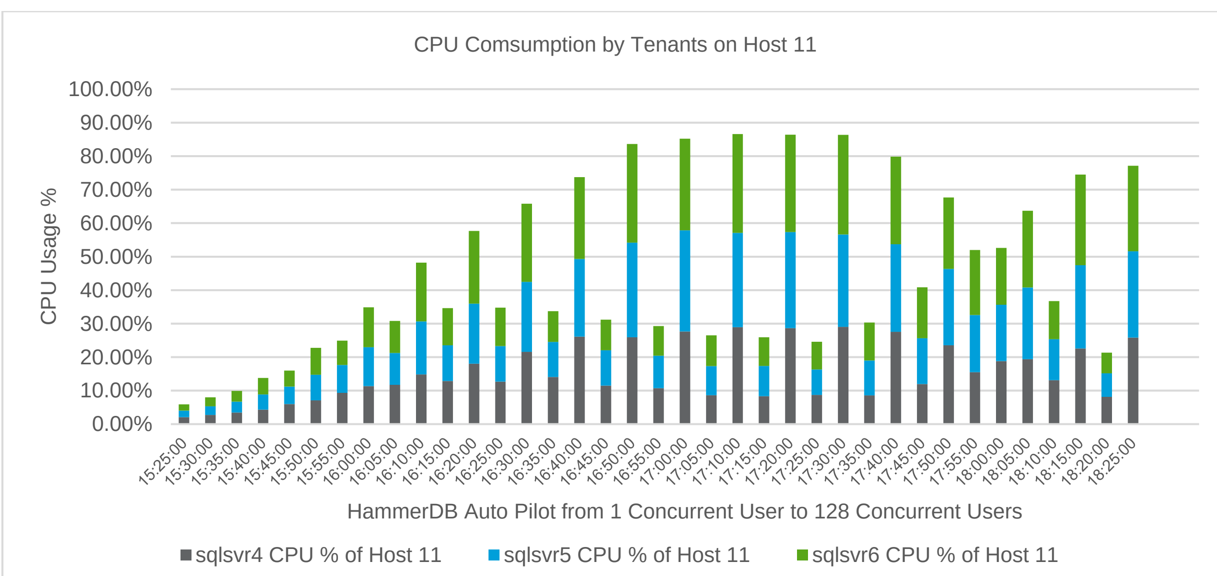
The highest CPU utilization reached by the storage VM during test run was 33.61%. The average CPU utilization was 22.09% during the entire test run or roughly a quarter of the CPU capacity on an HCI node was consumed by the storage VM when hosting SQL Server workloads.

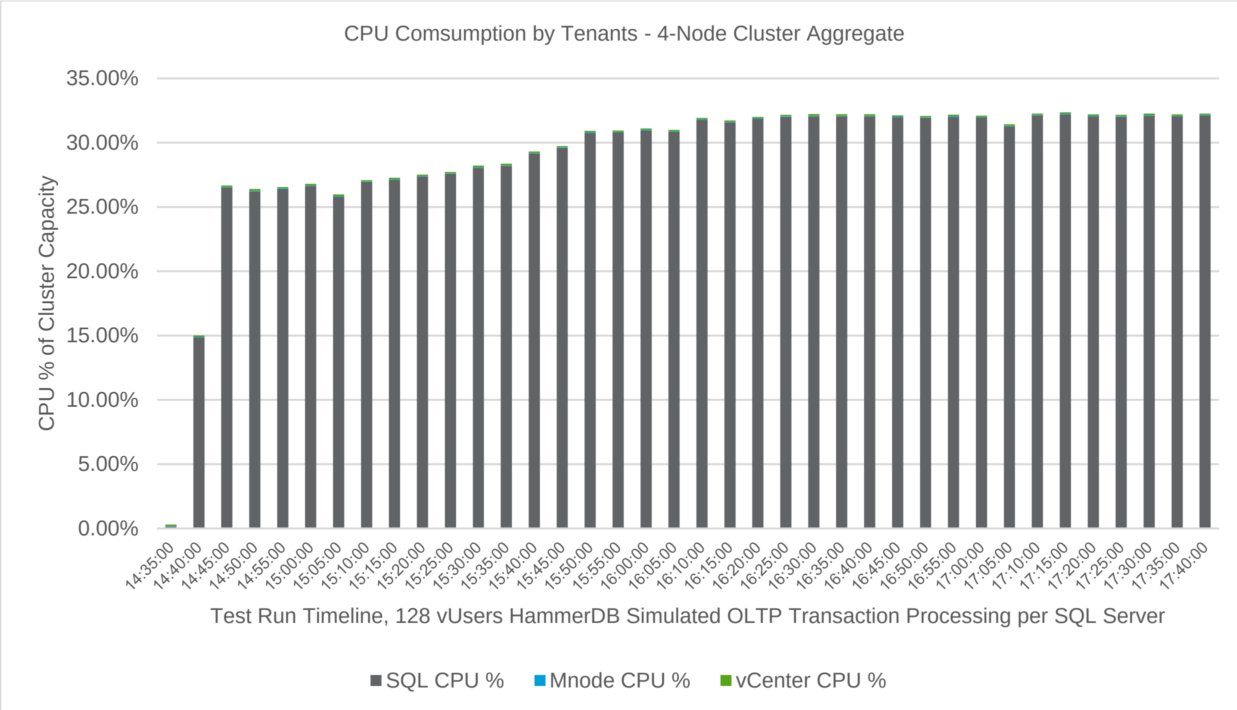
This storage VM CPU utilization measurement was limited to workloads on a single conventional HCI node. To measure the storage VM CPU utilization at the cluster level, we deployed four SQL Servers concurrently to a four-node cluster with one SQL Server VM on each node. We then ran HammerDB with 128 vUsers against each DB for a 3-hour test run. The storage VM CPU utilization at the cluster level can be seen in the following graph.




The preceding graph shows the aggregate of all tenants' CPU consumption for a four-node cluster as a percentage of total cluster CPU capacity. When broken out on the individual node level, the CPU consumption picture was almost identical. The storage VM consumed an average of 44% of the node CPU or cluster capacity. The test showed that the CPU overhead for the storage VM was much higher when the cluster was under heavy SQL Server workload.

NetApp HCI does not use a storage VM for I/O processing. At the cluster level, there is minimal overhead to run the management node service or mnode at 0.05% of cluster CPU capacity. As such, close to 100% of CPU cycles are devoted to SQL processing. For comparison, the first of the following two graphs demonstrate CPU usage on one HCI node hosting three SQL Server VMs and the second shows CPU usage on a four-node cluster hosting one SQL Server on each node. CPU utilization on the four-node cluster was aggregated CPU usage.






CPU consumption on the NetApp HCI test runs showed that CPU utilization was lower and much more efficient compared similar test runs with conventional HCI.

## TCO Analysis Consideration

For a true TCO comparison, all contributing costs, including hidden costs, should be taken into consideration. For SQL Server workloads, HCI-platform transaction-performance scalability might also contribute significantly to TCO, because the SQL Server transaction performance differential between competing platforms is too big to ignore.

To simplify the cost analysis, we first looked at how the storage VM compute overhead of conventional HCI contributes to the TCO. In this analysis, we calculated the hidden software licensing cost overhead of deploying SQL Server VMs on a conventional HCI node.

The following table shows the published list pricing for software we used for our TCO calculation.

| Microsoft SQL Server 2017 licensing price | Microsoft SQL Server Enterprise $14,256 per 2-core pack | $7,128/per core |
|---|---|---|
| VMware licensing | vSphere Enterprise Plus, $899.00 support/year | $3,595 / per CPU |
| VMware licensing | vCenter Server Standard, $1,544.00 support/year | $6,175.00 / per instance |

The two HCI systems are similarly configured, so we assumed that the systems can deliver similar transaction volume per SQL Server database VM. However, we have quantified that the storage VM CPU overhead for conventional HCI is between 20% to 40% for SQL Server workloads. Therefore, if you license an HCI node for SQL Server workload processing, 20% to 40% of the license cost is wasted on I/O services for the storage VM. In other words, in order to deliver the same SQL Server transaction volume required by business, you need to invest 20% to 40% more in software licensing to make up the shortfall, which increases the TCO by the same amount. The following table shows the TCO overhead for running SQL Server on conventional HCI in a single HCI node.

| 2x Intel XEON Silver Processor 8 cores. | Software licensing cost | Extra HCI tax is 20-40% of each software licensed to the compute cores (doesn't exist for NetApp HCI) |
|---|---|---|
| Microsoft SQL licensing price/node | 16 x 7218 = $115K | @20% approximately $23K<br>@40% approximately $46K |
| VMware vSphere Enterprise Plus | 2 x 3595 = $7K | @20% approximately $1.4K<br>@40% approximately $2.8K |

This hidden software licensing cost is further compounded by every HCI node that you are deploying to run SQL Server.

## Cluster Scale Out Cost

It is very common for database workloads that you sometime need additional storage to accommodate a large database. The ability of NetApp HCI to scale out storage independently eliminates the cost of compute when scaling out and thus, substantially reduces the TCO for SQL Server workloads. Conventional HCI, however, is unable to scale independently, which results in additional hidden expense. The following table shows the extra cost or waste on compute resources with conventional HCI when additional storage is needed.

| | Extra VMware licensing | Extra SQL Server Licensing | Extra SW TCO |
|---|---|---|---|
| Conventional HCI | 2 x 3595 = $7K | 16 x 7218 = $115K | $122K |
| NetApp HCI | $0 | $0 | $0 |

The extra $122K in software was not needed and contributes significantly to the TCO.

For supporting higher numbers of concurrent application users within the HCI cluster, you might sometimes need additional computing cycles. The following table shows the potential hidden cost with conventional HCI on storage not needed that contributes to TCO when adding a compute node to the cluster.

| | Compute Cost | Storage Cost (not needed) | Extra HW TCO |
|---|---|---|---|
| Conventional HCI | 15% of HCI cost per node | 85% of HCI cost per node | 85% of HCI cost per node or about 5X |
| NetApp HCI | $X | $0 | $0 |

The cost of the storage portion of an HCI node is generally a much larger proportion (85%) of the total cost or about five times the cost of a compute node. The inability to scale out compute independently by conventional HCI results in roughly five times the extra cost in hardware TCO than for NetApp HCI.

## The Big Picture

The TCO analysis above does not figure into the SQL Server transaction performance differential between the platforms. As we have demonstrated, NetApp HCI is much more scalable and resilient to deliver consistent SQL Server transaction performance relative to conventional HCI.

For larger SQL Server deployments that spans multiple nodes, NetApp HCI performance advantages can be even more significant and thus translate into a much lower TCO. In fact, based on the independent third-party Evaluator Group report "How Architecture Design Can Lower Hyperconverged Infrastructure Total Cost of Ownership", NetApp HCI TCO is as much as 59% lower than competing platforms.

# Technology and Methodology for Tests Validation

This section explains what hardware and software were used in test cases and how tests were performed on the HCI platforms.

## Cluster Configuration and Deployment

The lab environment consists of two four-node clusters for comparable tests. For NetApp HCI, this included four compute nodes and four storage nodes. The NetApp HCI cluster was deployed using the automated deployment engine (NDE). The details of the test environment for NetApp HCI, including hardware and software components, are listed in the following two tables.

| Hardware Components | Details |
|---|---|
| Storage node1: mssql-rtp01-stg-01 | H300S, 6x480G SSD drive |
| Storage node2: mssql-rtp01-stg-02 | H300S, 6x480G SSD drive |
| Storage node3: mssql-rtp01-stg-03 | H300S, 6x480G SSD drive |
| Storage node4: mssql-rtp01-stg-04 | H300S, 6x480G SSD drive |
| Compute node1: mssql-rtp01-esx-01 | H500E, two 12-core Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, 512G Ram |
| Compute node2: mssql-rtp01-esx-02 | H500E, two 12-core Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, 512G Ram |
| Compute node3: mssql-rtp01-esx-03 | H410C, two 14-core Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz, 512G Ram |
| Compute node4: mssql-rtp01-esx-04 | Supermicro SYS-2029BT-HNC0R, two 8-core Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz, 384G Ram |
| Data Switch | Dual Mellanox SN2010 25G |
| Management switch | Dual 1GB |

| Software Components | Details |
|---|---|
| Storage management stack | NetApp Element 11.7 |
| DB server operating system | Microsoft Windows 2016 |
| SQL Server version | 2017 |
| Hypervisor | VMware vSphere 6.7 |

| | |
|---|---|
| Load simulator | HammerDB 3.3 |
| Cluster monitoring | NetApp Active IQ |

The conventional HCI system contained a four-node packaged HCI block. The details of the test environment for conventional HCI including hardware and software are listed in the following two tables.

| Hardware Components | Details |
|---|---|
| Node1 - 174 | 2 x E5-2620 (2.1 GHz) v4 Intel Processors, 384 GB RAM, 6 x Samsung MZ7KM480HMHQ 480GB SSDs |
| Node2 - 175 | 2 x E5-2620 (2.1 GHz) v4 Intel Processors, 384 GB RAM, 6 x Samsung MZ7KM480HMHQ 480GB SSDs |
| Node3 - 176 | 2 x E5-2620 (2.1 GHz) v4 Intel Processors, 384 GB RAM, 6 x Samsung MZ7KM480HMHQ 480GB SSDs |
| Node4 - 177 | 2 x E5-2620 (2.1 GHz) v4 Intel Processors, 384 GB RAM, 6 x Samsung MZ7KM480HMHQ 480GB SSDs |

| Software Components | Details |
|---|---|
| Management and monitoring stack | Platform management version 5.15 |
| DB server operating system | Microsoft Windows 2016 |
| SQL Server version | 2017 |
| Hypervisor | VMware vSphere 6.7 |
| Load simulator | HammerDB 3.3 |

## Test Goal and Methodology

Tests on NetApp HCI and conventional HCI were performed to measure and compare the performance at scale, performance resilience during maintenance or failure scenarios, and storage VM overhead needed to run Microsoft SQL Server workloads on a four-node HCI cluster. These test cases are as following:

- SQL Server performance as the number of databases scaled up from one to three within a HCI node
- SQL Server performance as the number of databases scaled out from one to six on two HCI nodes
- SQL Server performance while a database VM is moved with vMotion to another node during maintenance
- SQL Server performance in a simulated node failure where all database VMs are moved via vMotion to a surviving node
- Time to rebuild cluster redundancy after node failure
- Measure storage VM CPU overhead on an HCI node

For SQL Server scalability testing, we started with one SQL Server database running on one compute node, and we then increased to three databases on a node for a scale-up comparison. We then scaled out to six databases on two compute nodes for comparison. For each scalability test, the SQL Server transaction performance was measured with a simulated OLTP workload via a HammerDB client running on a remote node. The workload was deployed using auto pilot with step-by-step increases in the number of concurrent users starting from one user up to 128 users. Each user level ran for 10 minutes. Performance metrics were reported by HammerDB and collected at 10 minutes intervals. An entire test run took approximately three hours. HammerDB tests produce two metrics: transactions per minute (TPM) and new order per minute (NOPM). The TPM numbers were compared with other runs from same platform to measure the platform scalability as the number of databases increased within a node or at the cluster level. The TPM performance numbers were then graphed against the number of concurrent virtual users for analysis and comparison in between runs.

For validating SQL Server performance resilience, we simulated maintenance activity by moving a SQL Server VM to another node via vMotion in the middle of a HammerDB test run. The performance result was then compared with a test run without interruption. An HCI node failure scenario was simulated by moving three SQL Server VM to a survival node with vMotion and then initiating a HammerDB test run to measure the SQL Server TPM performance while VMs remained on the surviving node. The test results were then compared with TPM performance metrics before failure to measure the performance effect on the SQL Server database.

Finally, Storage VM CPU overhead was measured with either three SQL VMs deployed on an HCI node or with one SQL VM each on a node. An HCI node was intentionally shutdown to time cluster rebuilding activity as a measure of platform resilience for hosting SQL Server workloads.

## SQL Server Configuration

For NetApp HCI, six database server VMs were deployed using a preconfigured VM template. The OS storage was provisioned from a pre-allocated VMware datastore created by NDE. Each DB server was configured with eight vCPUs and 16G of RAM. We follow VMware best practices for SQL Server for hosting three SQL Server databases on a node. The primary goal was to ensure a 1:1 pCore to vCore ratio for performance.

Each database VM was bonded with four NIC interfaces, one for private VM traffic, one for public internet access, and the remaining two for multipath iSCSI connections to the Element virtualized storage cluster.

SQL Server database storage volumes were created using the NetApp Element plug-in for VMware, and an access group was created for controlling volume access security. Each database server could only see its own storage volume. The database volume was then directly imported into the corresponding DB server VMs bypassing the hypervisor layer to provide improved performance. Jumbo frames were enabled on iSCSI interfaces. Multipath was configured for the iSCSI connection to provide redundancy in case of path failure. After detection in the Windows OS, the database volume was activated and formatted as drive E on Windows server. SQL Server 2017 software was then installed on drive C. After a tpcc user database was created and application schema were created and loaded, the DB server was ready for testing.

For conventional HCI, both the OS volume and the SQL Server data volume were provisioned off the VMware datastore on the NFS storage mount. All configurations inside the SQL Server VM were identically configured besides SQL Server data volume access, in which there were no direct iSCSI connections, and SQL Server data volume access was through VMware hypervisor.

For this solution, SQL Server was configured using only default settings. The goal was not to set any performance number targets but rather to evaluate database performance relevance in between test runs.

## Test Workload Configuration

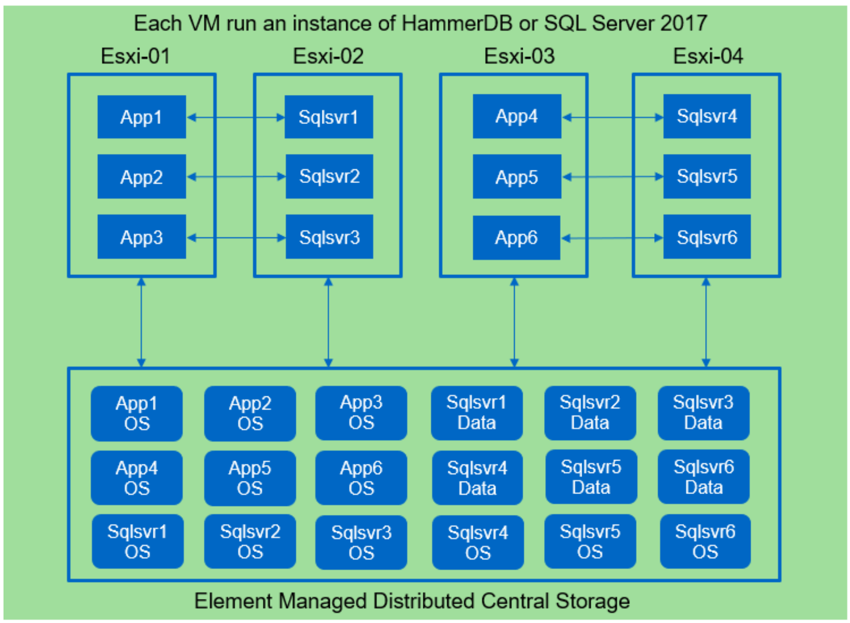
The SQL Server OLTP transactional workload was simulated using HammerDB auto pilot. Auto pilot loops through a preconfigured number of virtual users for a 10-minute test run. The number of virtual users per test run was configured to be 1, 4, 8, 12, 16, 20, 24, 32, 40, 48, 56, 64, 72, 80, 96, 112, and 128 to deliver a varied SQL Server transaction workload.

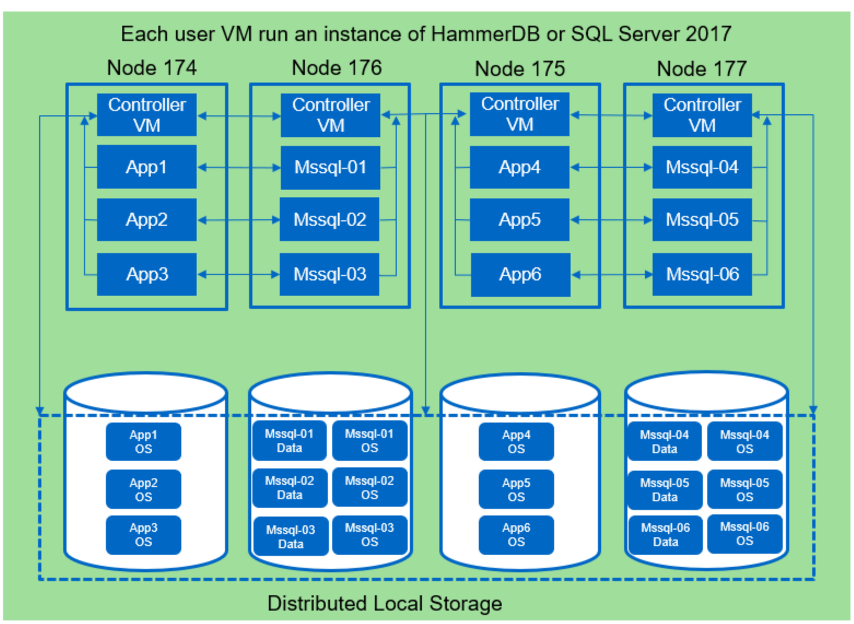
We created six app servers with each running an instance of HammerDB against a SQL Server DB server on two dedicated nodes. The app servers connected to database servers via TCP connections. This avoided unpredictable results if HammerDB was running on the same VM as the database server.

Each app server VM was configured with four vCPU and 8G RAM so that it has enough resources to accommodate the required number of concurrent users. We also used Windows Performance Monitor to collect system-resource usage to make sure that system resources were not maxed out.

## Test Environment Configuration and Management

The test environment was managed using the VMware vSphere web client or the platform-specific UI interface. The details of test environment for each platform are illustrated in the following two figures (NetApp HCI and conventional HCI respectively).



# Conclusion

Although all HCI platforms from this report are built on modern virtualization technology and software-defined storage data management, they are not created equal. NetApp possesses the following advantages over convention HCI:

- Shared and distributed central storage
- Dedicated high bandwidth application and storage networks
- Fine grained and evenly distributed cluster data
- Highly automated storage data management
- Efficient inline and global storage efficiency
- Unique disaggregated architecture

As a result, NetApp HCI outperformed conventional HCI for MSSQL database workloads in the following areas:

- Performance at scale
- Resilience in maintaining SQL performance and availability during maintenance, node failure and recovery
- Compelling financial differentiation in TCO by eliminating the HCI Tax

The exceptional I/O performance of NetApp HCI makes it stand out as a HCI platform of choice for mission-critical, latency-sensitive database workloads like MSSQL. The many benefits and advantages of NetApp HCI contribute to a reduction in the TCO when compared with conventional HCI.

In conclusion, if you are looking for an HCI platform to run MSSQL workloads, look no further than NetApp HCI.

# Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- Disaggregated HCI
  https://blog.netapp.com/disaggregated-hci/
- NetApp HCI 1.7 Deployment Guide
  http://docs.netapp.com/hci/index.jsp?topic=%2Fcom.netapp.doc.hci-ude-17P1%2Fhome.html
- Active IQ - Actionable Intelligence for Optimal Data Management
  https://www.netapp.com/us/products/data-infrastructure-management/active-iq.aspx
- SQL Server Technical Documentation
  https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15
- HammerDB User's Guide
  https://www.hammerdb.com/docs/index.html
- SQL Server on VMware Best Practice Guide
  https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-best-practices-guide.pdf

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.