vembu

Optimize RPO & RTO objectives while enhancing DR Resilience with Vembu OffsiteDR

Dr. Jack Fegreus

Founder of openBench Labs.

Copyright Notice

This whitepaper was initially published in December 2015 based on the test results and metrics obtained by using the Vembu BDR Suite v3.5 and these test results and metrics were updated on December 2018 by conducting the same set of tests using the Vembu BDR Suite v4.0

Optimize RPO & RTO objectives while enhancing DR Resilience with Vembu OffsiteDR

The ability to configure and deploy high-performance VMs within a vSphere virtual environment continues to put CIOs under increasing pressure to deal with the rampant bête noire of IT: business continuity. What started with Line of Business (LoB) driven Service Level Agreements (SLAs) requiring IT to meet rigorous Recovery Time and Recovery Point Objectives (RTO and RPO) has grown into an auditable ISO standard (ISO22301) and an emerging software niche for Disaster Recovery Management (DRM) systems.

LoB executives have played a critical role in the advent of DRM systems. For these key consumers of IT services, backup operations remain just a means to achieve business continuity. Their focus is entirely on the continuity of business processing and the restoration of processing in the event of a disruption. LoB executives expect CIOs to implement rigorous internal operations to minimize disruptions in business processing and when a disruption occurs the issue must be resolved in as short a time as possible and with minimal loss of data.

The introduction of OffsiteDR Server, a Vembu Backup & Disaster Recovery (BDR) data protection solution, enhances DRM operations by eliminating all potential single points of failure for restore functions. Using Vembu BDR Suite, IT admins are able to replicate backup data from multiple BDR Backup servers to a system running OffsiteDR Server within their own data center. As a result, IT garners an alternate system from which to recover protected VMs and physical servers using the same procedures that IT administrators employ on a BDR Backup server.

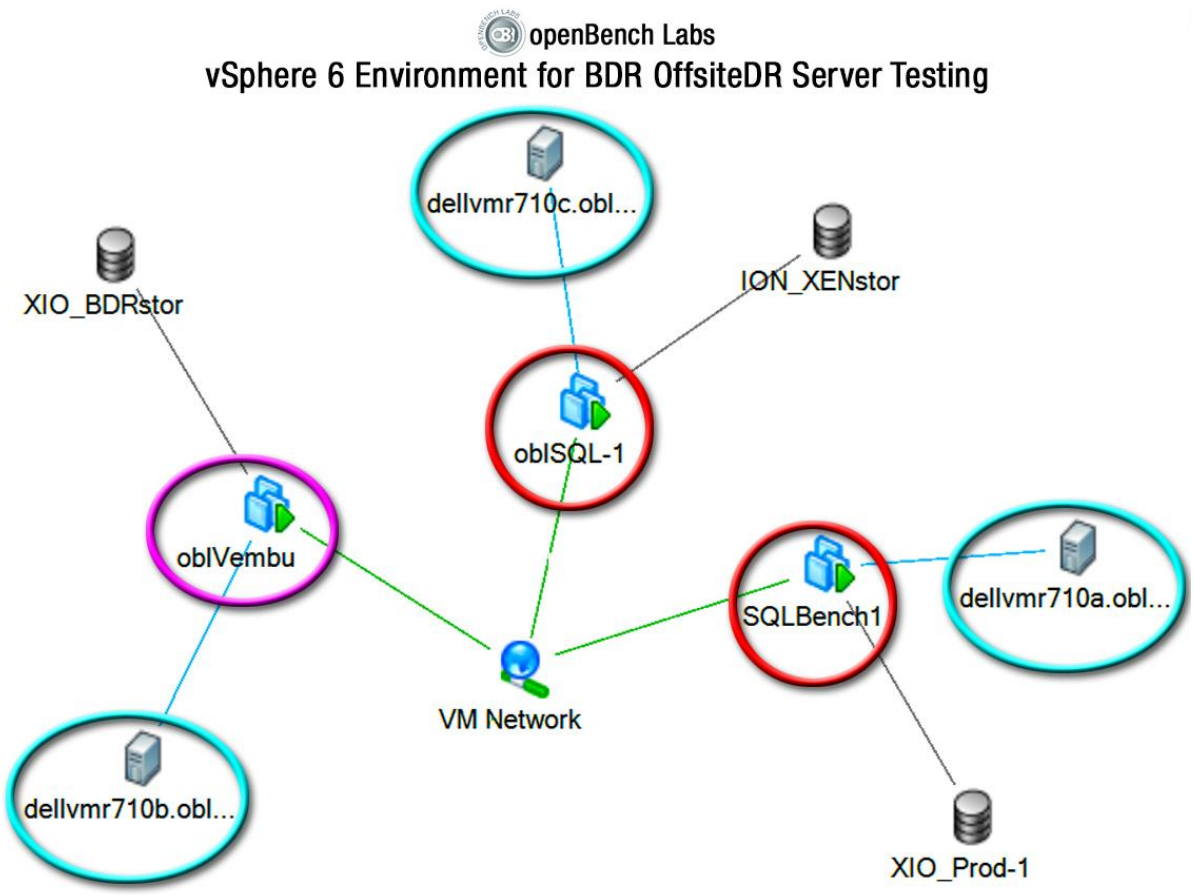
For this analysis, openBench Labs assessed the performance and functionality of the Vembu OffsiteDR Server, a DRM device that increases the resilience of recovery processes. Our initial intent was to examine the ability to restore data in the event of a catastrophic failure in our vSphere environment, including:

- A VM running BDR Backup server,
- an ESXi host, and
- a SAN device.

The full capabilities of Vembu OffsiteDR Server, however, quickly revealed that the device had a much broader operational impact. With the installation of OffsiteDR Server on an external physical server, we were free to restore operations in a way that optimized RTO and RPO for all business-critical application scenarios running in our vSphere test environment.

BUSINESS CONTINUITY FOR AN ACTIVE MISSION-CRITICAL VM

In testing Vembu OffsiteDR Server, we utilized three host Dell PowerEdge R710 servers with dual 6-core processors within a vSphere 6 test data center. In this data center, we set up two VMs, dubbed "oblSQL-1" and "SQLBench1," to support the simulation of a mission-critical LoB application that modeled an online stock trading application based on the TPC-E benchmark. On a third VM, named "oblVembu," we installed both the VMBackup client and BDR Backup server modules to create a self-contained BDR data protection environment. We provisioned Vembu on an FC SAN-based datastore, dubbed "XIO_BDRstor," and configured the VM with 4 CPUs, 8 GB RAM, and a paravirtualized Ethernet NIC (VMXNET3).



For our vSphere test scenario, all datastores used to store VM logical disks were provisioned on either 8Gbps Fibre Channel (FC) SAN or 10GbE iSCSI SAN volumes. By requiring a shared-storage topology for all VM datastores, we were able to leverage LAN-free technology during every VM backup. LAN-free technology enables Vembu BDR server to automatically determine if it can access and read VM backup data from a datastore directly using either SAN or hot-add SCSI transport mode.

A Vembu BDR server only calls on the ESXi host to initiate a LAN-mode backup, if both the SAN and hot-add SCSI mode tests fail. What's more, when we ran a full backup of a VM, we measured the same backup throughput using hot-add SCSI mode with a VM-based Vembu BDR server, as we did use a Vembu BDR server running on an independent physical server provisioned with HBA hardware for SAN connectivity.

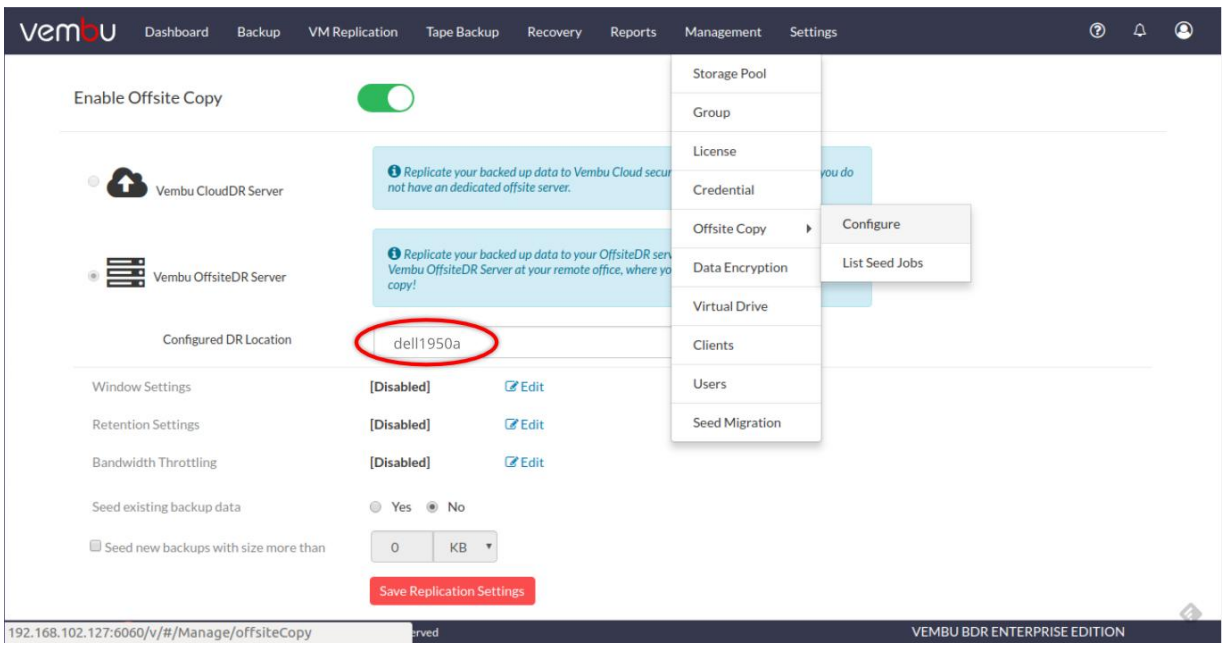
By adding an independent Offsite Server to our Vembu BDR data protection environment, we effectively decoupled restore configuration issues from backup configuration issues. From the perspective of a CIO, we had cut the Gordian knot tying RPO to RTO.

With all recent backups replicated across two servers—one VM-based and the other based on an independent physical server—we eliminated all server-related limitations to restore functionality. As a result, to meet an aggressive RPO, we were able to choose a backup server that maximized backup throughput and minimized VM application disruption in the most cost-effective manner without regard to restore issues. In addition, to meet an aggressive RTO, we were always able to select a recovery server to support a chosen Vembu BDR restore option and optimize network throughput with either an ESXi or Hyper-V host without regard to backup issues.

To model a full DRM scenario, we extended our data protection environment by adding a Dell PowerEdge 1950 server with dual 4-core processors. We located the new server outside of our vSphere environment, named the server "Dell1950A," and installed Windows Server 2012 R2, Hyper-V, and Vembu OffsiteDR Server. Our OffsiteDR Server provided all of the recovery features of a full BDR Backup server; however, the OffsiteDR Server could not be utilized as a primary backup server.

To replicate backup data to our OffsiteDR Server, we had to enable Offsite Copy Management using the Web GUI of a BDR Backup server. Once Offsite Copy Management was enabled from the Vembu BDR Backup server, we were able to set up global offsite copy parameters for all backups stored on that VM. From the perspective of our OffsiteDR Server on Dell1950A, Vembu was functioning as its virtual client.

We identified Dell1950A as the OffsiteDR Server for oblVembu, set new backup data on Vembu to be transferred immediately to Dell1950A, and set the maximum number of full backups, which includes all incremental chained backups, to be retained on Dell1950A, at two. Given the GFS retention schedules configured for each backup profile on oblVembu, the retention plan extended to Dell1950A effectively maintained two weeks of data on our OffsiteDR Server for each protected system.



GETTING DOWN TO BUSINESS

Our business test application, which was based on the TPC-E benchmark, focused on a simulation of stock trading at a brokerage firm. From a LoB perspective, the business model underpinning our application involved customers generating trades, making account inquiries, and initiating market research, while the brokerage firm was interacting with financial markets to execute customer orders and track market activity. In addition, the brokerage firm periodically ran a special CPU-intensive query that collected data about the trading process for internal BI analysis.

From an IT perspective, all of the critical processing for our database-driven application was focused on oblSQL-1, which was the VM running SQL Server 2014 to support the TPC-E benchmark database. We configured oblSQL-1 with eight CPUs; 32 GB of RAM; and three, thin-provisioned, logical disks using an iSCSI SAN-based datastore dubbed "ION_XENstor." In particular, we provisioned oblSQL-1's system volume (C:) with 100 GB, oblSQL-1's SQL Server deployment volume (D:) with 50 GB, and oblSQL-1's production TPC-E database volume (E:) with 100 GB. As a result, oblSQL-1 had a 142 GB storage footprint on ION_XENstor.

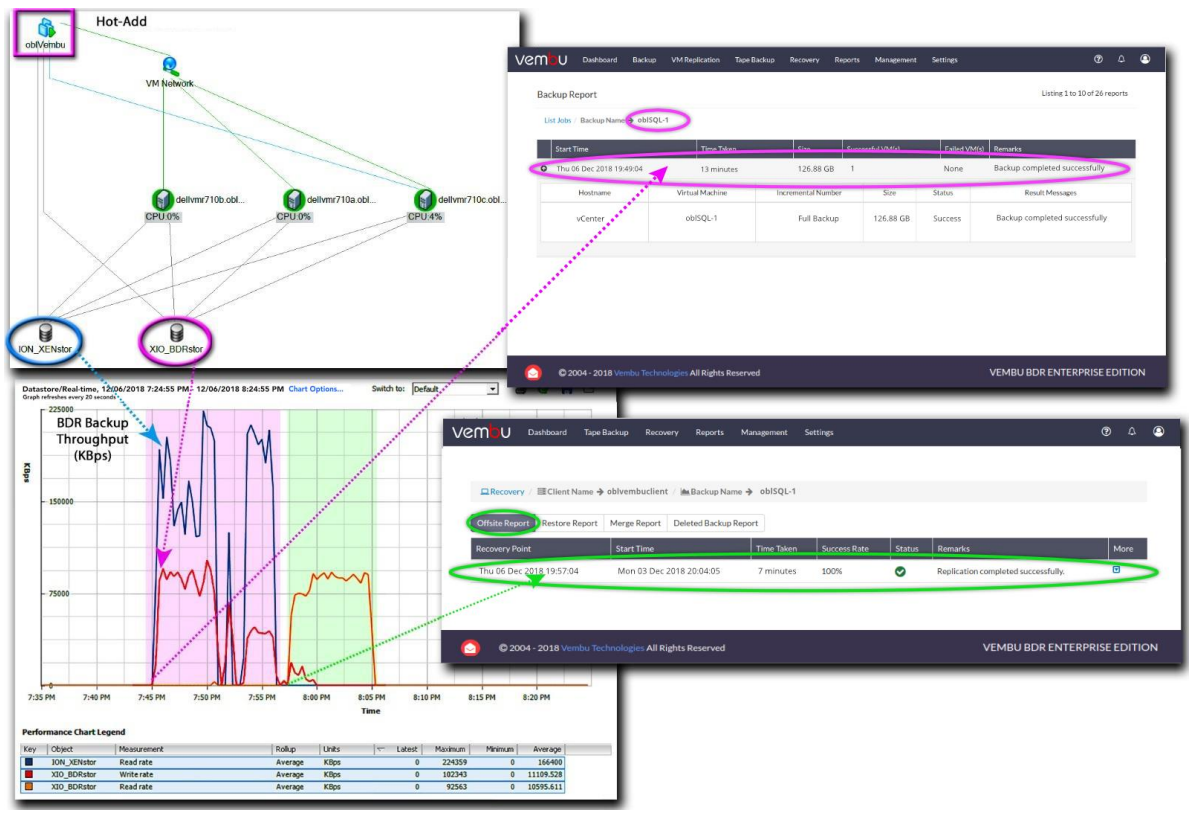
On the second VM, SQLBench1, we generated customer and broker transactions directed at the TPC-E database on oblSQL-1. While we utilized an instance of the TPC-E benchmark to model a mission-critical OLTP application, our goal in testing was very different from the benchmark's intent. We used the TPC-E benchmark model as a means to place a stable heavy write I/O load on a VM in order to measure how changes in the I/O load introduced by frequent incremental backups changed the existing application I/O equilibrium. To that end, we monitored aggregate transaction data for all TPC-E SQL queries executed during a test, which we designated the cumulative transaction processing rate (cTPS). In contrast, a traditional LoB performance perspective focuses on just the number of times that the business-oriented Trade-Result query, which handles the completion of stock market trades, was executed.

To ensure that all database reads associated with our test queries would be satisfied by SQL Server's buffer cache, we provisioned oblSQL-1 with 32 GB of memory. As a result, all physical I/O for the TPC-E database consisted entirely of write operations, which are the only I/O operations that affect the processing of snapshots for both VM logical disks and MS applications running on a VM.

ENHANCING RPO VIA SNAPSHOT REDIRECTION

We began our evaluation of BDR's performance in a DRM system by setting an end-to-end performance baseline for a full backup of oblSQL-1 with no active database processing. At the start of the oblSQL-1 backup, the Vembu BDR server on oblVembu triggered the ESXi host to create VMFS logical disk and VSS database snapshots, which took between one to two minutes to complete. Next, the Vembu BDR Server leveraged LAN-free mechanism to reconfigure oblVembu by attaching ION_XENstor, the datastore on which oblSQL-1 had been provisioned. Following hot-SCSI attachment of ION_XENstor, oblVembu sequentially mounted and processed the snapshots of oblSQL-1's three logical disks.

The Vembu BDR Server reads the data on the logical disk snapshots at upwards of 220MB per second. At the same time, it reformatted, deduplicated, compressed, and stored backup data for oblSQL-1—about 20% of the VM's original datastore footprint—at up to 100MB per second in VembuHIVE®, BDR's document-oriented file system. The initial full backup on oblVembu took just over 13 minutes and was followed by copying the oblSQL-1 backup data in VembuHIVE to the OffsiteDR Server on Dell1950A, which took less than 8 minutes.



Following our initial full backup of oblSQL-1, we launched our mission-critical LoB application by generating TPC-E queries on SQLBench1 for execution on oblSQL-1. In particular, we configured SQLBench1 to generate a steady stream of SQL queries with randomized data. The TPC-E database on oblSQL-1 handled those queries at an average rate of 850 cTPS. From a storage performance perspective, our query rate translated into a load of disk write operations equal to about 600 IOPS. Moreover, our queries, which were adding and updating database records, changed over 20% of the total VMFS block data—about 125 GB—for oblSQL-1 every hour. To comply with an RPO limiting data loss in the event of a processing failure to 10%, we scheduled an incremental backup every 30 minutes.

For a VM as active as oblSQL-1, running incremental backups every 30 minutes can be a very disruptive process. In particular, the creation and removal of VM disk and Windows' application snapshots during a backup can add significant I/O overhead and negatively impact I/O

processing of mission-critical applications with high I/O transaction rates. The main issue is the handling of new disk writes that occur during a backup. In our test case, write operations were proceeding at 600 IOPS in tandem with our backup. What's more, our test scenario further complicated snapshot processing by creating VSS application snapshots within the VM in order to quiesce SQL Server for database consistency and log truncation.

ESXi hosts implement a Copy on Write (CoW) snapshots on VM logical disks in VMFS datastores. CoW snapshots are highly space efficient and can be very quickly deployed as an empty file. Only when data needs to be written to a logical disk, does the host actually write data into the snapshot. In particular, the host reads the current data, writes that data to the snapshot, and then writes the new data to the original disk location. As a result, the host performs three I/O operations for each new write to the file representing a VM logical disk.

The overhead for writes associated with a CoW snapshot escalates dramatically when a business-critical application with high write I/O activity runs on a VM with a Windows guest OS. As part of the backup process, a VSS requestor agent will need to invoke the VSS Writer to create application snapshots to quiesce processing and prevent data corruption. For this process, most VSS requestor agents double down on backup I/O overhead by implementing CoW application snapshots within the Windows OS and then encapsulating those snapshots within a CoW snapshot for each logical disk on the host. In contrast, Vembu's AppAware extension to VMware tools implements Redirect on Write (RoW) snapshots within a Windows guest OS.

RoW snapshots provide the same space-efficiency as CoW snapshots but do not double the number of write operations. Specifically, RoW snapshots do not copy existing data into a snapshot file before writing new data to the original location. Instead, new data is written directly to the RoW snapshot file and a pointer is setup to redirect access around the old data, which remains in place. RoW snapshot overhead is only manifested when unwinding pointers associated with a long chain of snapshots. Since AppAware utilizes only one RoW snapshot during a backup, there is no chain of snapshots to unwind.

AGGRESSIVE RPO WITH RESILIENCY

We tested the efficiency of the Vembu BDR's RoW snapshot scheme in our end-to-end DRM environment, which began with BDR Backup running on a vSphere VM and ended with OffsiteDR Server hosted on an independent Windows server. Our test DRM environment, like every DRM environment, was driven by two fundamental constructs: We needed to limit data loss during a recovery operation and we needed to perform any recovery operation as quickly as possible.

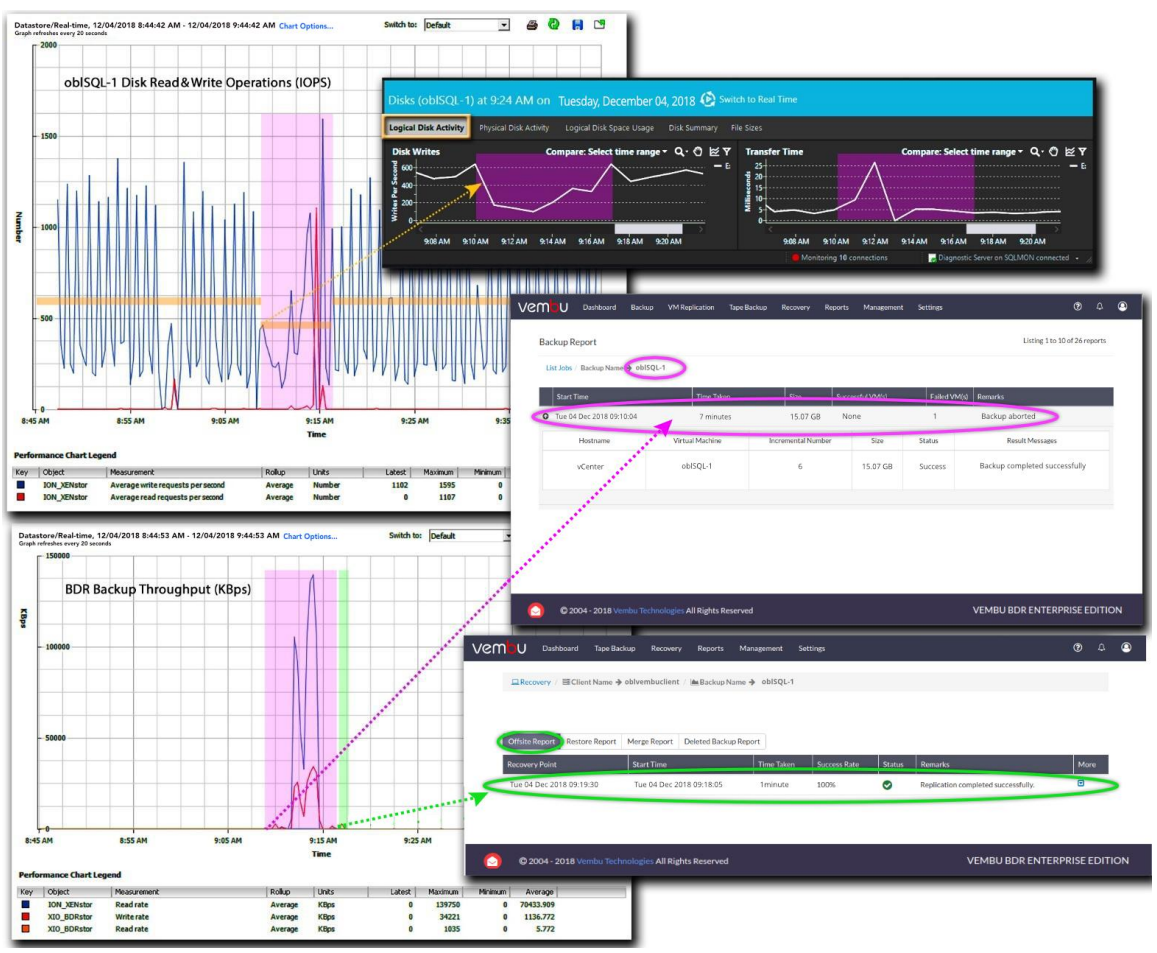
Limiting data loss on an active VM is synonymous with the execution of frequent incremental backups. Through the use of Changed Block Tracking (CBT), VMware limits the data collected in an incremental VM backup to logical disk blocks that have changed since the previous backup. For a backup package, such as BDR, the VMware CBT mechanism limits the problem of implementing frequent incremental backups to quiescing an active VM with minimal disruptive overhead before initiating a CBT-based backup. As a result, the I/O write load on an active VM is a primary driver of the time needed to complete an incremental backup.

In our OLTP test scenario, oblSQL-1 processed SQL queries at a rate of 850 cTPS. Under this query load, SQL Server generated steady streams of 2,400 logical read IOPS and 600 physical write IOPS just on the logical VM disk supporting the TPC-E database. At that level of I/O activity, the ESXi host took from three to four minutes to complete the generation of VMFS and VSS snapshots for oblSQL-1's three logical disks and active SQL Server databases under the direction of the Vembu BDR Server. In addition, releasing the VMFS snapshots at the completion of the backup added an addition 30 seconds to the incremental backup window.

Once oblSQL-1's host had prepared a snapshot for each VM logical disk, the Vembu BDR Server sequentially mounted each snapshot to read the VMware CBT data, which the client then streamed to VembuHIVE at a rate similar to that measured in a full backup. What's more, the amount of processed data added to VembuHIVE was so small that it typically took just one minute to transfer a copy to the OffsiteDR Server. As a result, it took just 8 minutes to perform an end-to-end incremental backup with an offsite copy for either a 25 GB incremental backup every 60 minutes or a 12 GB incremental backup every 30 minutes: Setting up snapshots under our heavy I/O load conditions consumed 60 to 65% of the total backup time.

More importantly, from the perspective of a LoB executive, the only impact frequent incremental backups had on our business application was a minimal drop of 5% in transaction processing during the backup, as all reads were offloaded to cache. Specifically, TPC-E query processing dropped to about 805 cTPS during the initial backup stage and rebounded to 850 cTPS by the time the backup had completed.

From an IT operations perspective, physical write operations on the TPC-E database volume dropped precipitously to around 200 IOPS as the VMFS and VSS snapshots were created. Simultaneously, data transfer latency for writes ballooned from 3.5 to 25 ms as the Vembu BDR Server quiesced SQL Server. Following the quiescence of SQL Server, the write load on the database disk steadily rebounded to a level of 600 IOPS. More importantly, once the Vembu BDR server had quiesced the VM and SQL Server, there was no impact on transaction processing from either reading incremental backup data, unwinding VMFS and VSS snapshots, or copying VembuHIVE data to the OffsiteDR Server.`



RESTORE EVERYWAY ANYWHERE

The other important construct of our test DRM environment was the need to perform any recovery operation as quickly as possible. With respect to recovery functionality, Vembu BDR implements a number of important well-known features with an ingenious twist. Rather than store block-level backups of VMs as a collection of backup files, Vembu BDR utilizes a document-oriented NoSQL database, dubbed VembuHIVE, as a backup repository. Within VembuHIVE, backup data is stored as documents, which encapsulate information encoded in value-key pairs without a strict schema.

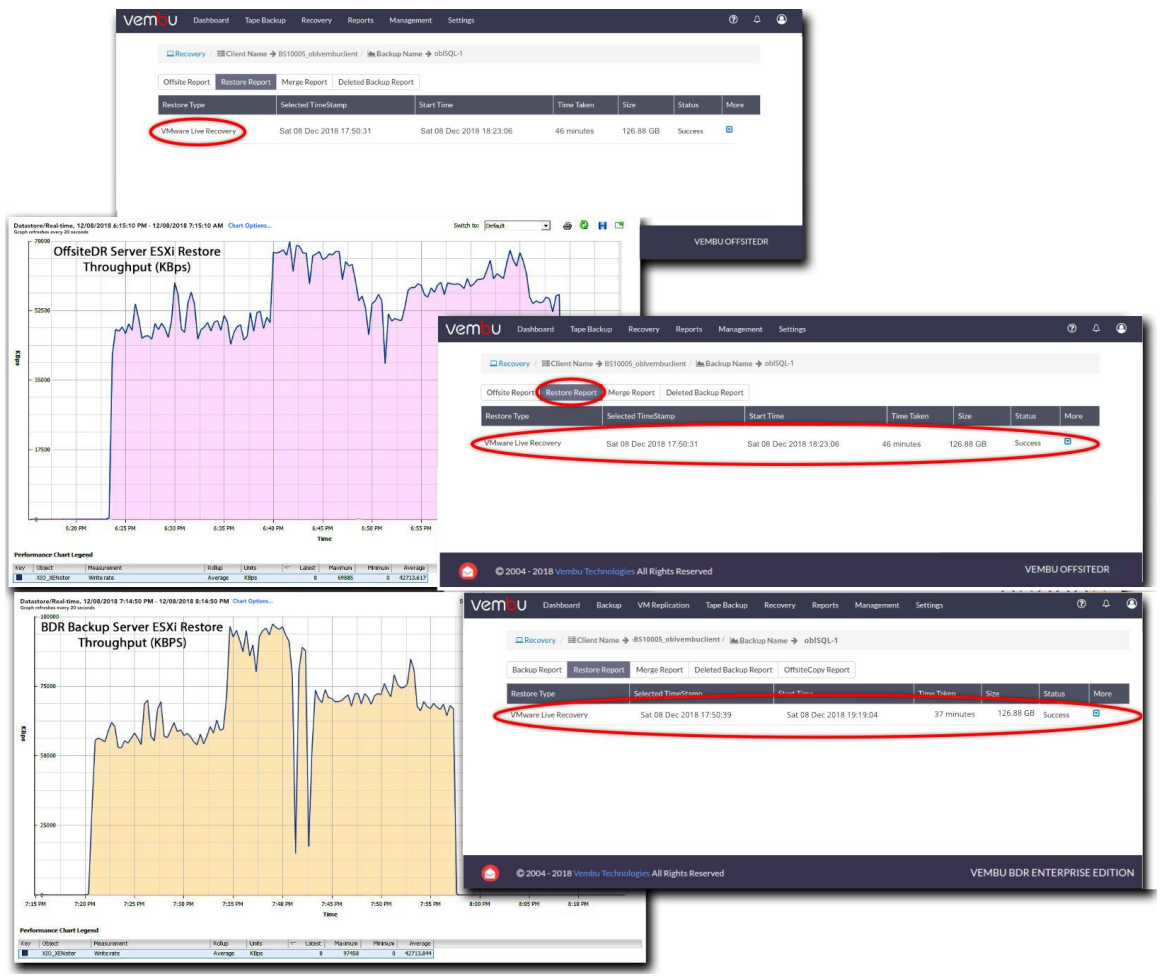
The VembuBDR service handles all of the backup and restore functions on a Vembu BDR Backup server, while the VembuOffsiteDR service handles restore functions on an Vembu OffsiteDR Server. During a backup, the VembuBDR service removes all file-system metadata, adds content-related metadata, and restructures the resulting data in a hypervisor-neutral format. In addition, the VembuBDR service compresses and de-duplicates all value-key pairs before storing the data in VembuHIVE. While our test VM, oblSQL-1, had a datastore footprint of 170 GB, which included three thin-provisioned virtual disks, VembuHIVE utilized just 46 GB to store one full and six incremental backups of oblSQL-1. In terms of recoverable data, the 46 GB of data in VembuHIVE represented seven distinct recovery points that would consume 1.15 TB of storage, if restored to our production datacenter.

What's more, by storing value-key data in a hypervisor-neutral format, VembuHIVE can be virtualized as a logical file system. As a result, both the VembuBDR and VembuOffsiteDR services are able to extend restore functionality by mimicking advanced OS file system utilities. Specifically, these services are able to expose all backups as a set of disk images in the native file system formats of a number of virtual environments, including vSphere, Hyper-V, and KVM.

We began testing traditional restore functionality by running a full-restore of oblSQL-1 from both our BDR Backup server and our OffsiteDR Server. Since every restore operation needs to interact directly with an ESXi host, all data traffic transferred during a restore has to be over LAN. Nonetheless, shared-SAN connectivity still provided a significant, albeit indirect, advantage in our environment.

We were able to boost throughput while streaming network data to an ESXi host by an average of 20% when we transferred the data from a VM that was configured with a paravirtualized NIC and attached to the target ESXi host. As a result, we were able to minimize restore time by performing a fast host migration with vMotion before running a restore from our VM-based BDR Backup server to minimize restore time.

On a full restore of oblSQL-1, peak throughput using the BDR Backup server on oblVembu spiked 40% higher than using OffsiteDR Server on Dell1950A, an external physical server. More importantly, both VM restore operations took well under the typical RTO goal of 1 hour to complete. Specifically, we completed a full restore in less than 38 minutes using the VM-based BDR Backup server and 46 minutes using OffsiteDR Server on an external server.`

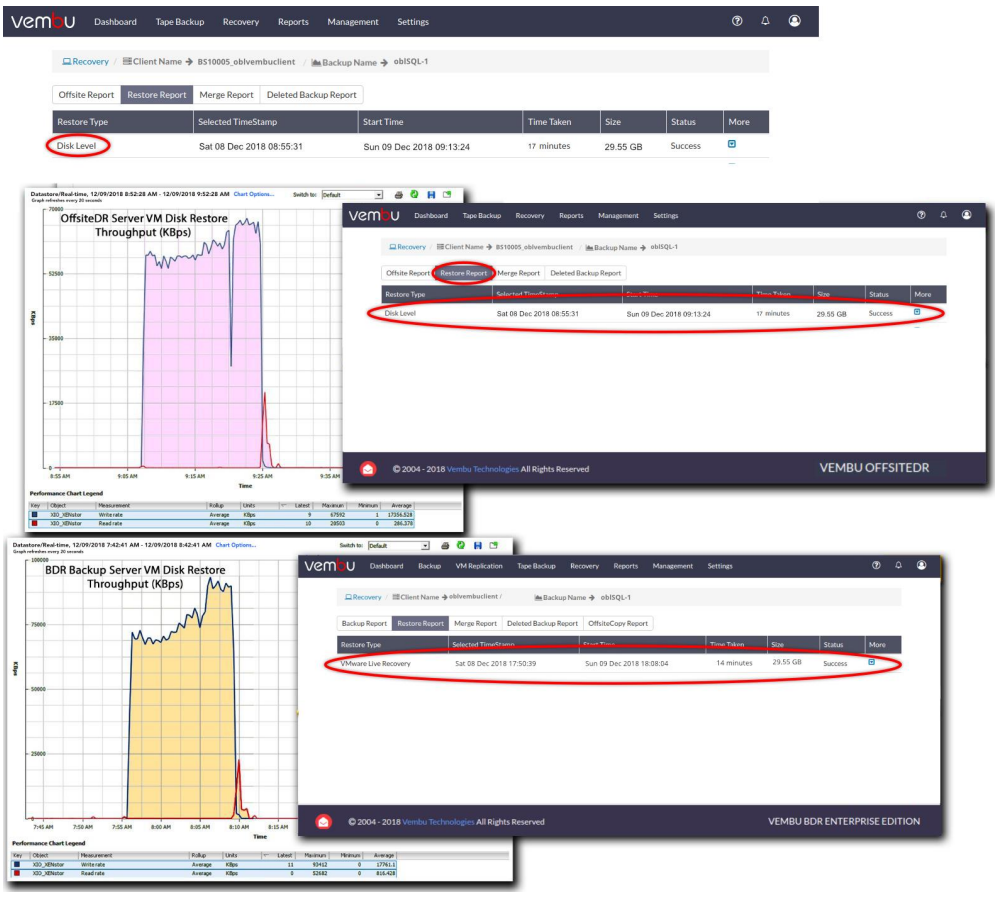


We measured the same differential in network throughput running Vembu BDR's unique restore option for restoring a disk from a recovery point. The Disk Level Recovery option leverages the ability of VembuHIVE to expose a restore point as a set of disk images to recover a specific corrupted disk on a vSphere VM without making any other changes to the VM. For most IT recovery incidents, a VM disk restore is all that is needed to resolve a data recovery issue in significantly less time than it takes to restore an entire VM.

To enhance the power of VM restore for IT operations, the Disk Level Recovery adds a new disk on the target VM and leaves the original disk intact. As a result, a local system administrator on the VM garners a number of follow-up options once a VM disk restore has completed, including the ability to use the newly restored disk to recover data and repair the corrupted volume.

In our OLTP test configuration, all unique business data was located on a thin-provisioned 100 GB volume that contained TPC-E database file and logs, which consumed between 25-to-35 GB of storage during our tests. Using Disk Level Recovery from either our OffsiteDR Server or BDR Backup server, we were able to recover that volume and resume full OLTP processing in 15-to-20 minutes. What's more, given our schedule of incremental backups, we were able to limit data loss to 30 minutes of processing.

Just as in a full restore, we were able to leverage VMware's networking optimizations to recover our database volume more quickly from oblVembu, our VM running Vembu BDR Backup server, than from our physical server running Vembu OffsiteDR Server. In particular, we recovered a restore point representing 29.55 GB instance of oblSQL-1's TPC-E database disk in just over 14 minutes from oblVembu and in just under 18 minutes from Dell1950A.



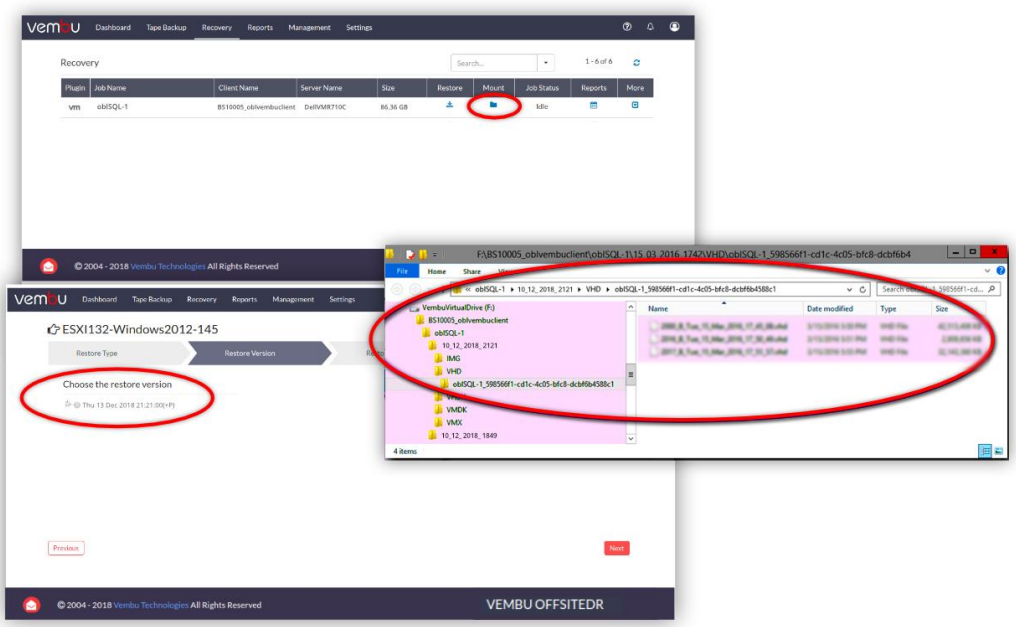
REALTIME RECOVERY FROM VIRTUAL MEDIA

Both Vembu BDR and Vembu OffsiteDR servers are capable of supporting all of the Vembu BDR restore features. Included among the important BDR restore options is Vembu's Quick VM Recovery, which leverages tight integration with Hyper-V on a Windows server along with the ability to virtualize backup data in VembuHIVE as a set of disk images, to provide IT with the ability to meet a 10-minute RTO goal for any VM, without regard to VM's original host server.

Nonetheless, the dependence of Quick VM Recovery on Hyper-V makes this fast recovery feature available only when BDR Backup server or OffsiteDR Server is installed on a physical server. By running the BDR Backup server on a vSphere VM and immediately copying all backup data to an OffsiteDR Server installed on a physical system, we garnered three crucial advantages for our DRM environment:

- We leveraged all vSphere networking optimizations between a VM and its ESXi host;
- utilized all restore features provided by Vembu BDR server; and
- ensured our ability to restore backed up data, if our backup server was unavailable.

Key to the implementation of Quick VM Recovery and a number of other restore features is the ability of the Vembu BDR and Vembu OffsiteDR server to create a virtual device, dubbed "VembuVirtualDrive", on the host system. Using our OffsiteDR Server running on Dell1950A, we were able to mount read-only (M) virtual disk images in all supported formats for a restore point. We were also able to invoke a more sophisticated restore option (VHD), which mounted the three logical disk images associated with oblSQL-1 on the Vembu virtual device with write access. The Vembu BDR and Vembu OffsiteDR servers preserve the integrity of restore points mounted with write access, by adding metadata to VembuHIVE with the initial write. In particular, the new metadata creates an independent persistent restore point (+P) with the same timestamp as the original restore point.



Vembu leverages its ability to create new synthetic restore points to provide aggressive features for booting a VM directly from VembuHIVE documents without using special write caches and redo logs, which have to be managed by a system administrator. Vembu simplifies its Quick VM Recovery feature by integrating a local Hyper-V environment with VembuVirtualDrive in order to provide system administrators with a single-click restore option.

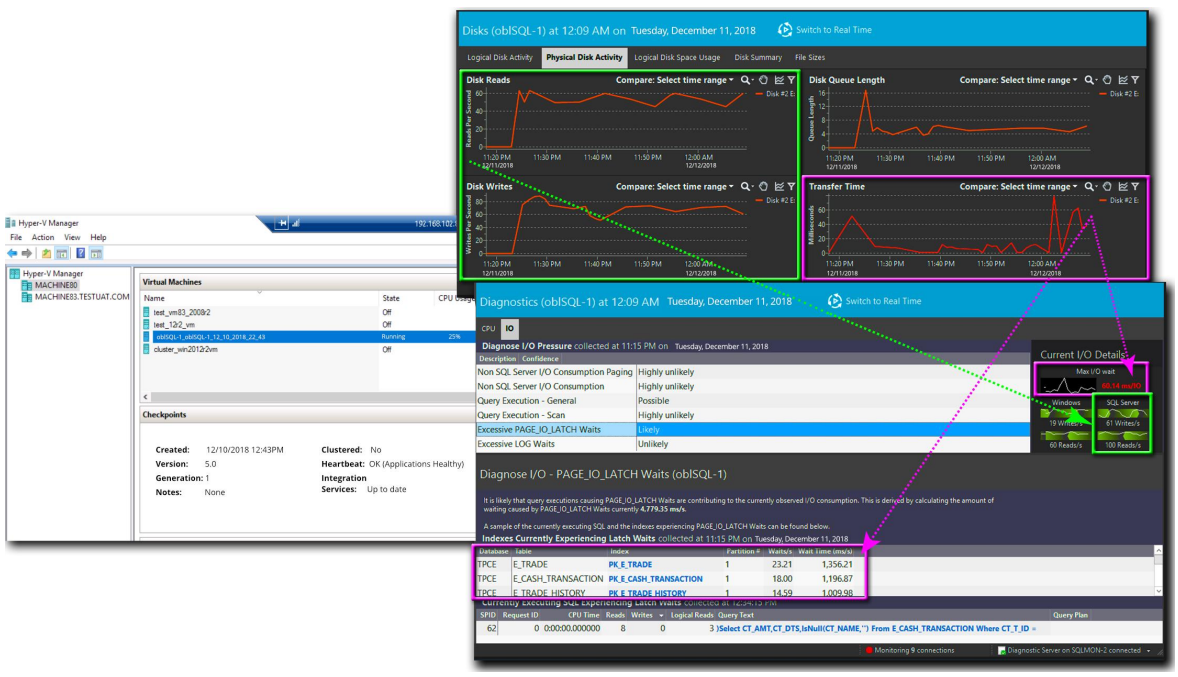
When a system administrator launches Quick VM Recovery, the local Vembu BDR or OffsiteDR server automatically:

- mounts virtual disks associated with a restore point with write access on VembuVirtualDrive,
- boots a 1st-generation Hyper-V VM using vhd-formatted disk images on VembuVirtualDrive,
- and creates an independent synthetic VM restore point in VembuHIVE.

There is no need for IT to configure or maintain a complex IP subnet to isolate and masquerade VMs booted from backup data with respect to interactions with VMs on the production network.

We tested Quick VM Recovery on our OffsiteDR Server using a backup of oblSQL-1. The OffsiteDR Server automatically generated a unique name for the Hyper-V VM by appending the name and creation date of the backup to the VM's original vSphere node name, oblSQL-1. To test I/O performance on the new VM, we determined the highest query rate that would not generate a large queue of pending I/O requests or SQL execution errors from blocked queries. Specifically, we found a rate of 125 cTPS to be a safe limit for our test configuration.

With our Hyper-V VM booted from VembuHIVE representing oblSQL-1, we easily sustained a TPC-E query load of 85 cTPS. This query load translated into a disk I/O load of about 120 IOPS—65 write IOPS and 55 read IOPS—on the logical disk containing the TPC-E database. While sustaining queries at 85 cTPS, the disk I/O queue for the TPC-E database volume typically contained five pending operations and disk I/O transfer latencies hovered around 15 ms. Nonetheless, we observed frequent spikes of up to 80 ms in disk transfer latency, which increased query execution time. Specifically, we observed long wait times while taking out and releasing I/O page latches to access the indices of active TPC-E database tables.



The initial boot time of our Hyper-V VM, which completely determines restore time for the Quick VM Recovery option, took approximately nine minutes as new devices were initialized—in particular, the three logical disks virtualized from VembuHIVE. Less complicated vSphere VMs typically took less than five minutes to complete an initial Hyper-V boot. Once we completed an initial boot of our test VM, it took less than two minutes to reboot that VM from its persistent restore point; however, for meeting an RTO commitment, only the initial boot time matters.

While we were able to meet an RTO of ten minutes for our test VM via Quick VM Recovery, we were not able to sustain the very high OLTP rate sustained on the original vSphere VM. Nonetheless, for a majority of database-driven LoB applications, the ability to sustain 125 TPS will adequately maintain normal processing levels.

More importantly, we were able to select on demand the persistent recovery point associated with the live Hyper-V VM in VembuHIVE and restore an entire fully-functional vSphere VM—a process that took about one hour—or just a particular disk, such as the TPC-E database volume, to an existing vSphere VM, with all of the data changes generated while running as a Hyper-V VM. Unlike competitive solutions, there was no need to run a lengthy special process to consolidate new data on the VM from redo logs or a write cache.

NFS vs HYPER-V

Without the tight integration that Vembu BDR leverages with a local Hyper-V environment, there is no equivalent to the Quick VM Recovery restore function that configures and boots a VM on a vSphere host from documents in VembuHIVE. As an alternative to Quick VM Recovery, Vembu provides a Virtual Drive Management function that integrates an open source NFS network server with the VHD mount option.

Virtual drive management presents system administrators with a simple set of options to share the local VembuVirtualDrive volume on a LAN via NFS. An administrator is able to select any backup job defined on a Vembu BDR Server and mount all recovery points for VMs protected by that job as writable disk images in the local VembuVirtualDrive volume. Once these disk images are mounted, they become accessible on any ESXi host importing the NFS share for the VembuVirtualDrive volume.

From the BDR Backup server running on oblVembu, we invoked Virtual Drive Management to share the VembuVirtualDrive volume on the VM as an NFS share, which was automatically dubbed “VembuNFS.” Next, we imported the VembuNFS share on each of our ESXi hosts as a new datastore, dubbed “NFS_VembuHIVE”. For performance testing, we created a new VM, which we named oblSQL-NFS, and attached three vmdk-formatted logical drives associated with a restore point for oblSQL-1.

We safely ran SQL queries at a rate 85 cTPS when we targeted the VM restored by Quick VM Recovery in a Hyper-V environment; however, while running our ersatz NFS-based VM, we were able to sustain processing TPC-E database queries at a rate of 25 cTPS. Even with a query load of 25 cTPS, we observed intermittent transient spikes in pending I/O operations that placed upwards of 1,200 I/O requests in the disk operations queue. Moreover, average data transfer time on to the TPC-E database volume increased by an order of magnitude—ballooning from 15 to 150 ms. Meanwhile, the average wait time for a SQL query rose from 65 to 245 ms.

While an NFS-based restore provided tepid performance compared to an Instant Boot into a Hyper-V environment, an NFS-based restore does provide a solid functional addition to the resilience of restore operations. While sharing the VembuVirtualDrive via NFS does not provide a primary way to quickly restore a VM into production, it does provide reliable access to data archived in backup restore points. What's more, when changes to NFS-shared drives are made, VembuHIVE stores those changes in an independent, recoverable, synthetic restore point.

In our test environment, the combination of Vembu OffsiteDR Server deployed on a physical server with a Vembu BDR Backup server deployed on a VM provided a value proposition that extended far beyond the enhancement of DRM recovery resilience. With OffsiteDR Server installed on a physical server, we were able to optimally leverage VM and physical server platforms to easily implement all of the data protection functionality provided by Vembu BDR Suite, leverage all of the performance optimizations available to VMs in a vSphere environment, and do so in the most cost-effective system configuration.

From an IT overhead perspective, manually setting up an NFS-based vSphere recovery configuration with vCenter that paralleled the automatically generated Quick VM Recovery configuration turned out to be relatively simple. The real issue, however, turned out to be network overhead and throughput, which we discovered while performing an initial boot of oblSQL-NFS.

While an initial boot of the Hyper-V VM automatically configured with Quick VM Recovery took just over 9 minutes to complete, it took over 28 minutes to complete the initial boot of oblSQL-NFS—including discovery and internal device configuration of the VM's three NFS-shared logical disks. More importantly, that 3-to-1 differential in boot time was reflected in I/O latency while running TPC-E queries.