vembu

True 15-Minute RTO for Mission-Critical VMs using Vembu BDR Suite

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

True 15-Minute RTO for Mission-Critical VM Systems with Vembu VM Replication

Most IT sites have a key system that is essential to the survival of their business. When processing on such a system is interrupted, essential Line of Business (LoB) operations cease to function and corporate business is significantly impacted. Moreover, the longer it takes to restore a mission-critical system, the greater the likelihood that the business will face substantial repercussions. As a result, nirvana for a Service Level Agreement (SLA) for business continuity centers on zero recovery time and zero recovery point objectives (RTO and RPO).

Mission-critical systems at the majority of IT sites fall into one of two major categories: database-driven financial and process-control manufacturing systems. For mission-critical systems that do not require specialized process-control hardware, high-performance VMs within a vSphere virtual environment provides IT with a distinct advantage in dealing with scenarios that impact business continuity.

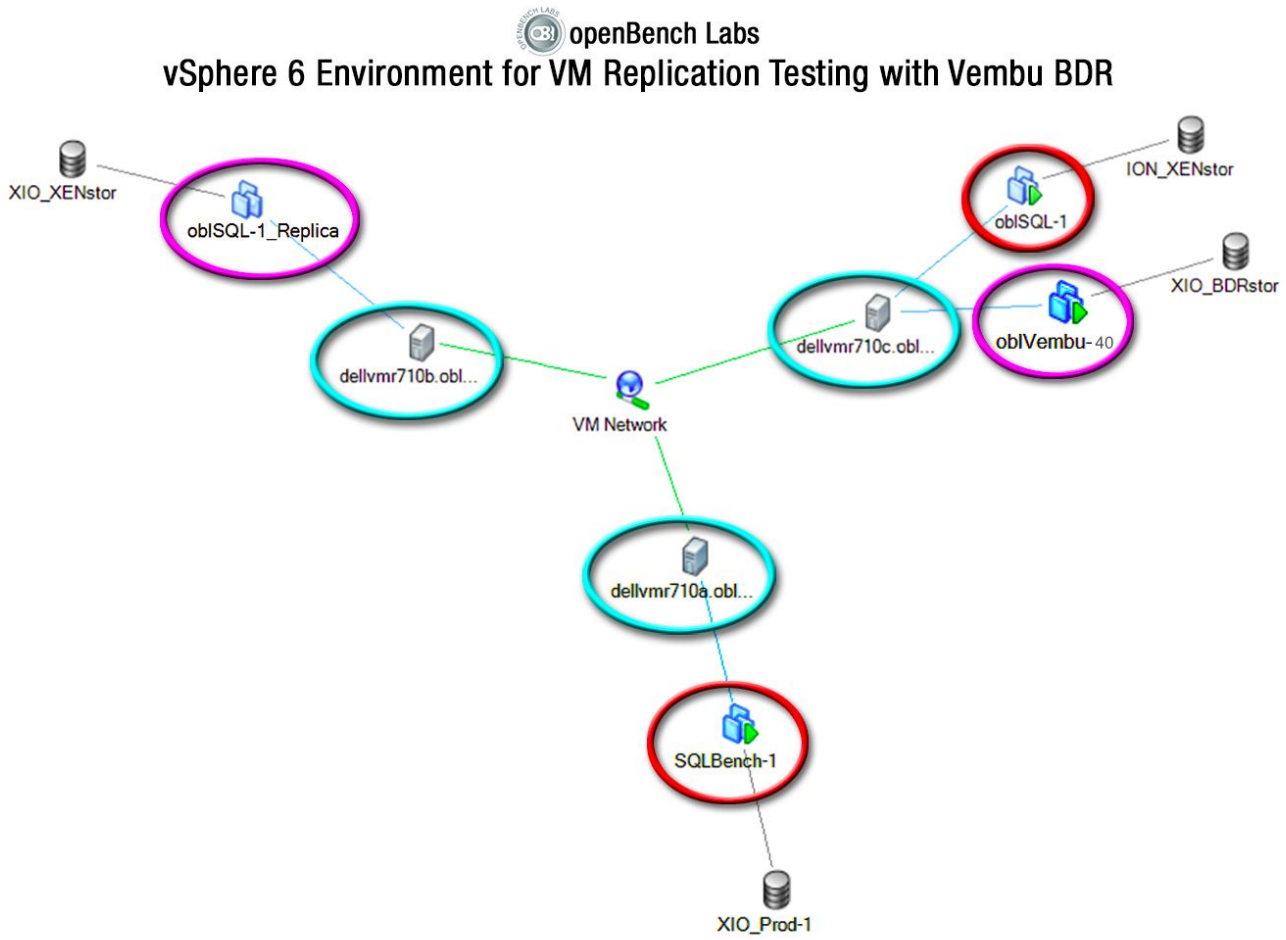
In such an environment, the new v4.0 of Vembu Backup & Disaster Recovery (BDR) provides IT with a Disaster Recovery Management (DRM) system capable of meeting even more aggressive RTO and RPO goals than the previous release. For highly active database-driven systems, Vembu BDR v4.0 leverages VMware tools and VMware Changed Block Tracking (CBT) to perform incremental backups in 15-minute intervals with minimal impact on query processing. As a result, IT can limit data loss to 15 minutes of processing on active mission-critical VMs.

Nonetheless, protecting data with closely interspersed data recovery points represents only half of the requirements spelled out in a business-continuity SLA. Losing only 15 minutes of past processing is just the starting point in the race to the recovery a fully functional VM. If it takes 60 minutes to restore a VM with a large volume of data, then from a LoB perspective, the organization has lost 75 minutes of processing time. That's why recovery time for mission-critical systems is such a pivotal issue.

Today, many data protection packages attempt to resolve the problem of meeting a stringent RTO goal by booting an ersatz production VM directly from a backup file. These techniques provide the ability to rapidly present a recovered VM capable of performing the same functions as the original VM without first performing a restore operation. Nonetheless, the recovered VM is not capable of sustaining the same level of application performance, which is a critical deficiency in the eyes of a LoB executive.

The only way to recover a VM with full functionality and full performance without performing an explicit restore operation is through VM replication. Maintaining a replica VM, however, requires frequent and potentially expensive update processes that involve both explicit backup and implicit restore operations.

To enable the extensive use of replication by IT, Vembu BDR v4.0 adds critical optimizations to both restore and replication operations that dramatically minimize overhead on ESXi hosts and production VMs to just VM snapshot processing. Specifically, a Vembu BDR server running on a VM is able to leverage hot-add SCSI transfer mode to write logical disk and logical disk snapshot files directly to a vSphere datastore, without involving the ESXi host for anything more than creating a VM snapshot.



What's more, Vembu BDR v4.0 has a replica management module that enables an IT administrator to fully manage an initial failover and later finalize failover or failback with consolidation. In addition, BDR 4.0 simplifies all management functions by eliminating the need to run a separate client module on a BDR server, which becomes its own client within the BDR reporting hierarchy. To enable our VM to best support the extended functionality of BDR Backup server, we provisioned a VM with 4 CPUs, 8GB of RAM, and a para-virtualized Ethernet NIC.

TESTING A MISSION-CRITICAL OLTP SCENARIO

To test VM replication, we utilized three Dell PowerEdge R710 servers with dual 6-core processors as ESXi hosts in a vSphere 6 environment. In addition, we set up a LOB application scenario that was highly sensitive to IO overhead to measure the overhead impact of replication.

Our LoB application simulated an OLTP stock trading application based on the TPC-E benchmark. Our objective was to measure how I/O load changes introduced by frequent incremental backup and replication operations changed the I/O equilibrium of the OLTP application. To that end, we monitored aggregate transaction data for all TPC-E SQL queries executed during a test, which we designated as the cumulative transaction processing rate (cTPS). Standard TPC-E performance tests focus specifically on the transaction rate of the business-oriented Trade-Result query, which handles the completion of stock trades.

To run our test application, we set up two VMs, dubbed "oblSQL-1" and "SQLBench-1." On oblSQL-1 we ran an instance of TPC-E benchmark database generated from 16 GB of initial table data. To support the TPC-E benchmark database, we installed SQL Server 2014 and provisioned eight CPUs, 32GB of RAM, and three, thin-provisioned, logical disks on an iSCSI SAN-based datastore dubbed "ION_XENstor." We generated customer and broker business transactions targeting the TPC-E database on SQLBench-1. Finally, we installed Vembu BDR v4.0 on a VM, which we dubbed "oblVembu-40," and replicated oblSQL-1 to a VM dubbed "oblSQL-1_Replica."

We provisioned all vSphere datastores in our test scenario on either an 8Gbps Fibre Channel (FC) SAN or a 10GbE iSCSI SAN. By using shared-storage for all VM datastores, we were able to leverage LAN-free technology during every VM backup and restore process. LAN-free technology enables a VM running BDR Backup to determine whether the datastore containing target VM logical disks can be accessed directly using either SAN or hot-add SCSI transport mode.

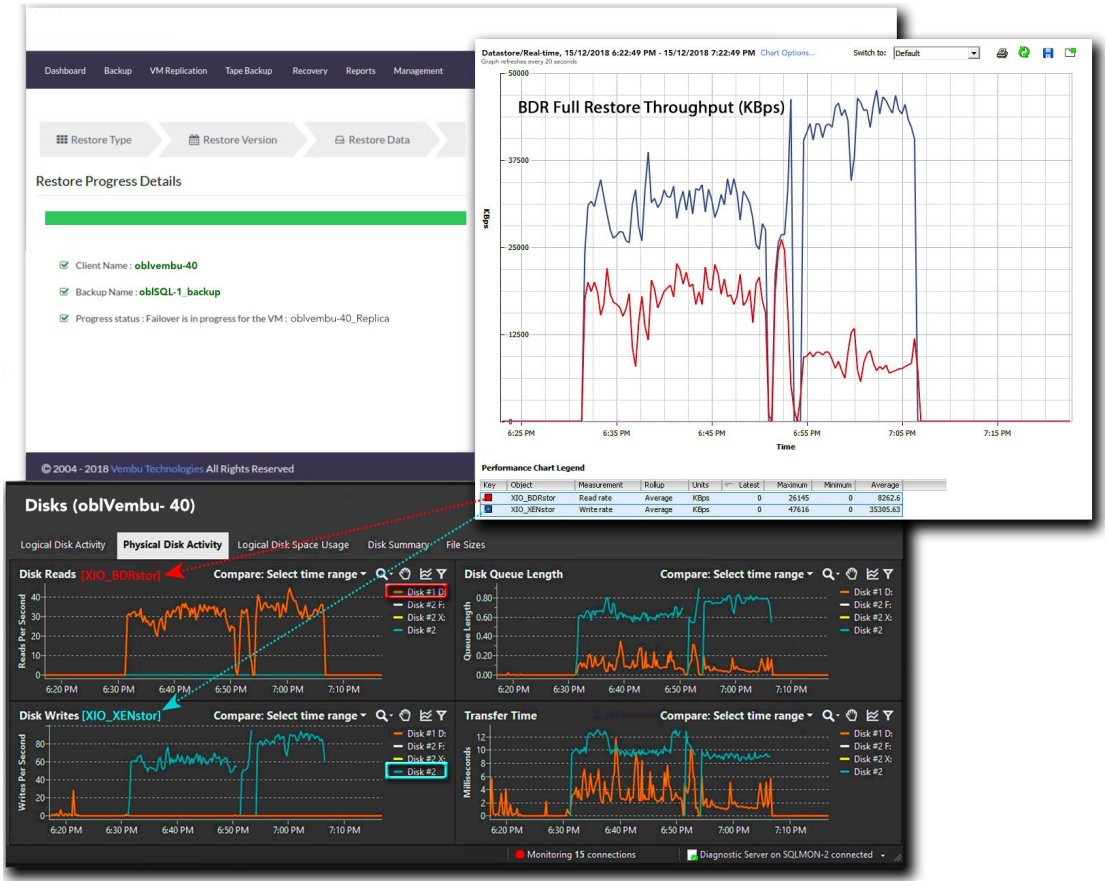
While many VM backup packages only utilize direct datastore access to read data during a VM backup, Vembu BDR utilizes hot-add SCSI mode to read and write data directly to vSphere datastores in backup, restore, and replication operations. In an end-to-end VM replication, Vembu BDR is able to read VMware Change Block Tracking (CBT) data from a source VM snapshot and write that data as a logical disk snapshot in the datastore of a replica VM. As a result, Vembu BDR server is able to perform frequent replication updates with very minimal overhead impact on an active VM.

At the start of a full backup of oblSQL-1, the Vembu BDR server, oblVembu-40, triggered oblSQL-1's host to create snapshots for each of the VM's three logical disks in ION_XENstor. Next, Vembu BDR Backup server sequentially mounts and reads block data from each of the three logical disk snapshots at upwards of 220 MB per second. At the same time, the BDR Backup server reformatted, deduplicated, compressed, and wrote the resulting backup data—about 20% of the VM's original datastore footprint—to VembuHIVE®, BDR's document-oriented database, at over 95 MB per second. By dramatically reducing the amount of data written to VembuHIVE, the initial full backup took just 11 minutes.

After completing the full backup, we next restored oblSQL-1 from VembuHIVE to the datastore XIO_XENstor. We began the restoration of oblSQL-1 by explicitly designating XIO_XENstor as the target datastore; however, we directed the backup to our vCenter server, rather than to a specific ESXi server.

By allowing oblVembu-40, the VM running BDR Backup, to restore the backup through our vCenter server, the process was able to leverage the configuration data for oblSQL-1. At the start of the restore process, vCenter was directed to provision a new VM that had an identical device infrastructure—CPUs, memory, storage and networks—to oblSQL-1. In particular, the new VM had identical SCSI controllers and disks with the same capacities as those on oblSQL-1, but were void of data.

Once the target VM was created, oblVembu-40 used vCenter to trigger a snapshot on the ESXi server that vCenter utilized to host the restored VM. Next oblVembu-40 mount the snapshot to expose the oblSQL-1 backup data as a set of three vmdk-formatted virtual disk images. BDR Backup was then able to stream the backup data from VembuHIVE, rehydrate that data by a factor of about 5X, and then write the rehydrated data to the three virtual disk files on XIO_XENstor.



By leveraging the capabilities of both vCenter and VembuHIVE, Vembu BDR reduced the process of restoring oblSQL-1 to streaming the VM's virtual disk images, as exposed by VembuHIVE, to the new datastore that was provisioned through vCenter. By directly streaming the logical disk data to the new VM's datastore, the entire restore process was able to complete in just 36 minutes.

In the final stage of baseline testing, we ran an initial replication of oblSQL-1. For any VM, the initial replication job is essentially executed as a full backup and restore in order to create a complete replica of the source VM. As a result, just as in a full restore of a VM from a traditional backup, we identified our vCenter server as the "host" of the new VM replica and designated XIO_XENstor as the target datastore.

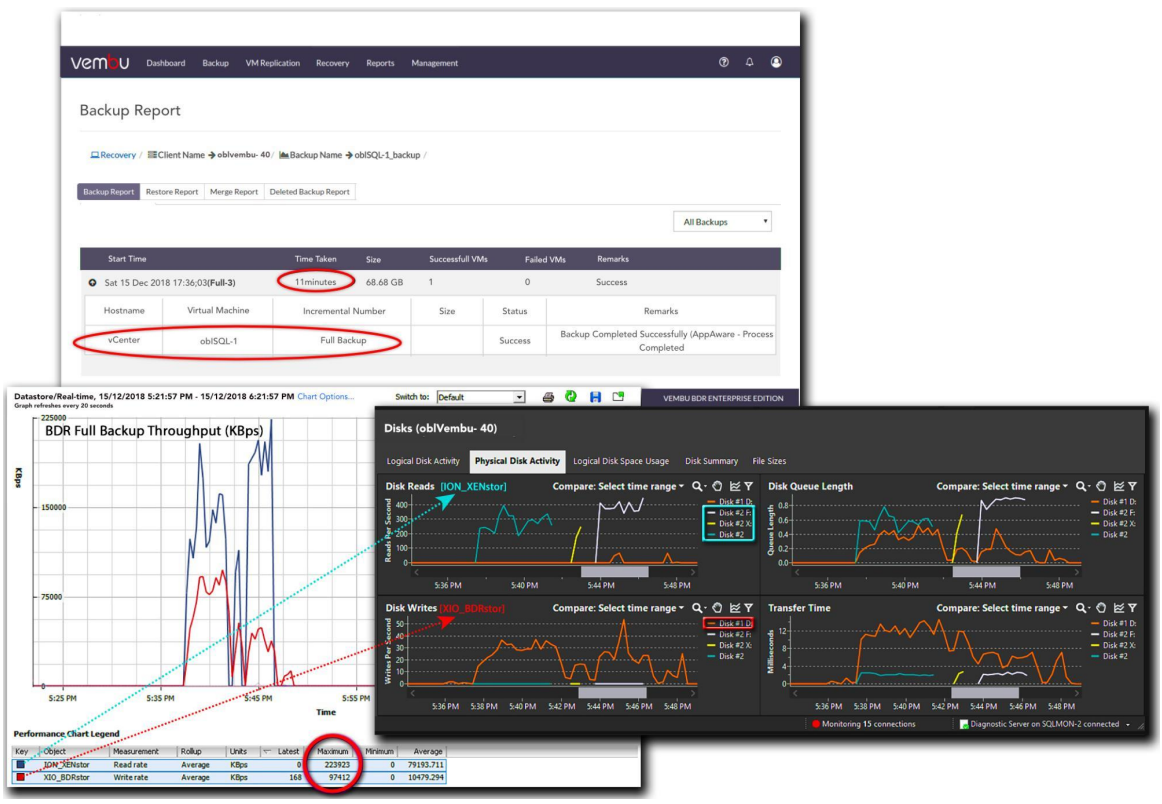
At the start of the replication process, vCenter server created and provisioned a new server named oblSQL-1_Replica. Once again, the new VM had the same device infrastructure as oblSQL-1. Next, in the most critical point of the replication process in terms of overhead, oblVembu-40, triggered our vCenter server to request both the ESXi host of oblSQL-1 and the ESXi host of oblSQL-1_Replica to create snapshots for each VM.

BASELINES FOR REPLICATION PROCESSING

We had provisioned oblSQL-1 on the ESXi datastore, ION_XENstor, with three logical disk volumes:

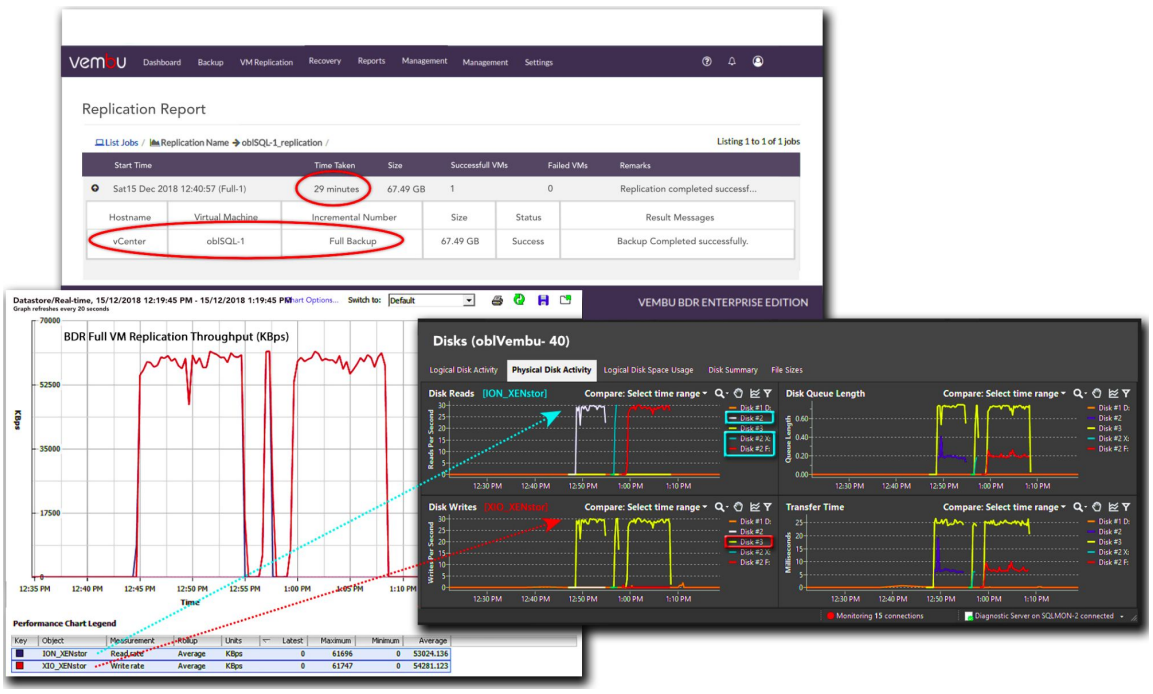
- a system volume (C:) provisioned with 100 GB of thin-provisioned storage,
- a SQL Server deployment volume (D:) with 50 GB of thin-provisioned storage,
- and a TPC-E database volume (E:) with 100 GB of thin-provisioned storage.

When active, oblSQL-1 had a storage footprint on the ION_XENstor datastore ranging between 100 and 140 GB, which included a 30 GB vSphere cache file for VM page swapping. Variation in the storage footprint was primarily associated with the growth of table data in the TPC-E database and to a lesser degree, growth in system database tempdb as a result of the test queries initiated on SQLbench-1.



We began our evaluation of VM replication performance by setting end-to-end performance baselines for a full backup and a full recovery of oblSQL-1 with no SQL query activity. In this process, we backed up oblSQL-1 from its initial datastore, ION_XENstor, and then restored the VM to the datastore that we planned for replicating of oblSQL-1, XIO_XENstor.

Following the successful creation of snapshots for oblSQL-1 and oblSQL-1_Replica, oblVembu-40 once again leveraged LAN-free mechanism to reconfigure its disk infrastructure. Using hot-add SCSI mode, the datastore snapshot of oblSQL-1_Replica along with the snapshots of oblSQL-1's three logical disks were all mounted on oblVembu-40. Our BDR Backup server was then able to sequentially read the logical block data from each of oblSQL-1's three volumes and write that data as three disk image files in the datastore of oblSQL-1_Replica. As a result, all I/O during the end-to-end replication process was handled directly by oblVembu-40 in a process that took 29 minutes to complete.



Once the replica VM was created, we were able to update the replica with a continuous schedule of incremental backups. On each incremental backup, the Vembu BDR server proceeded to:

Reorganize the CBT data transferred from each logical disk snapshot on the original VM,

- Format that data as a disk snapshot for the corresponding disk on the replica,
- And stream the snapshot file to the replica datastore.

In this process, the Vembu BDR server consistently wrote snapshots to the replica VM that were 50 to 60% smaller than the original CBT data transferred during the backup.

MISSION CRITICAL DRM

To test the effectiveness of implementing replication on an active VM in a comprehensive DRM solution, we configured our TPC-E benchmark generator on SQLbench-1 to drive a load of TPC-E transactions at an average rate of 850 cTPS. In processing this business-transaction load, SQL Server executed approximately 400 atomic SQL TPS on the TPE-E database and 1,000 atomic SQL TPS on SQL Server's tempdb.

While the volume of SQL transactions on tempdb was well over twice the volume of SQL transactions on the TPC-E database, the pattern of physical I/Os on ION_XENstor, the datastore containing oblSQL-1, was dramatically different. All I/O activity on tempdb consisted of logical reads, which contributed no physical I/Os on ION_XENstor. In contrast, the SQL transactions on the TPC-E database were entirely write operations. As a result, our business transaction load translated into a steady stream of approximately 600 physical write IOPS on the TPC-E database and, in turn, ION_XENstor. These patterns in physical and logical SQL transactions and the resulting IOPS load on the underlying ESXi datastore had important implications for instituting a comprehensive DRM scheme for oblSQL-1.

The prominence of logical reads in the processing of our TPC-E business transactions on SQL Server makes our OLTP application less sensitive to external I/O operations, such as the reading and writing of snapshot data on the underlying datastore. Nonetheless, the high rate of write operations for the datastore—600 IOPS—makes oblSQL-1 particularly sensitive to the creation and unwinding of logical disk snapshots, particularly with respect to the logical disk containing the TPC-E database.

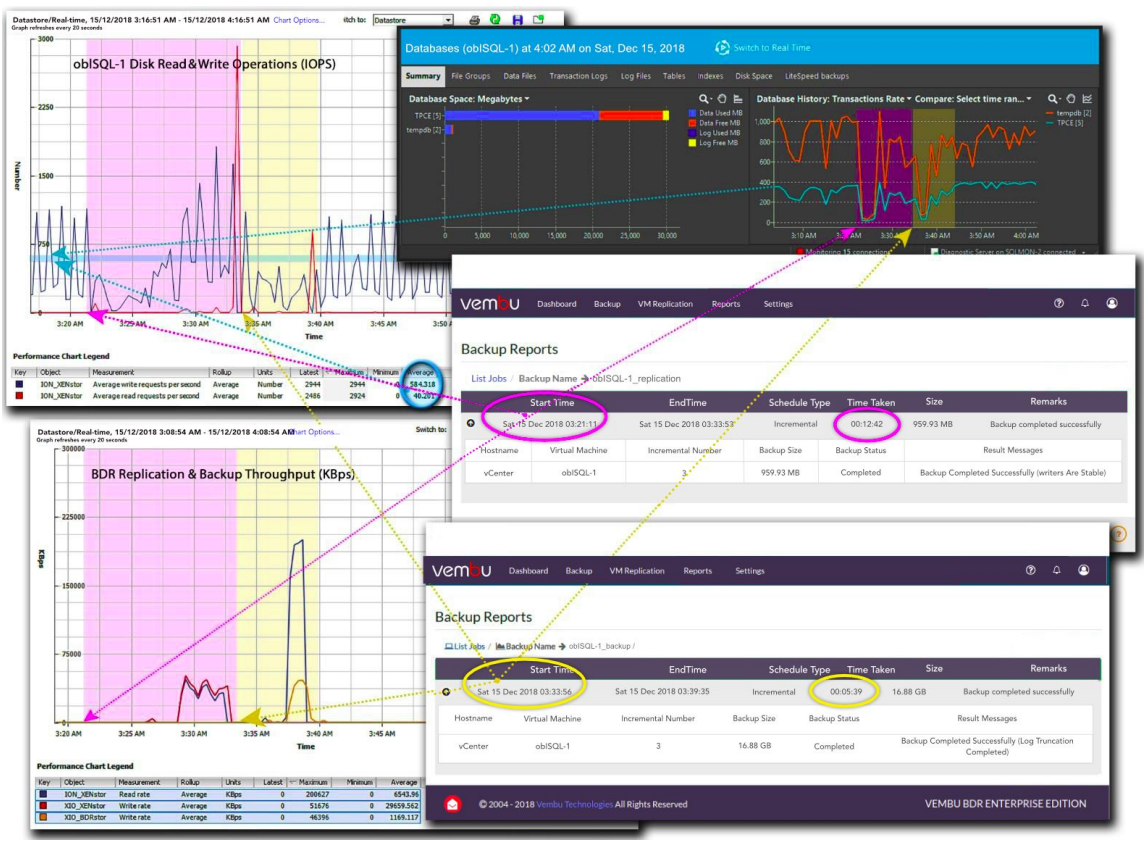
More importantly, our business transactions, which were adding and updating TPC-E database records, were generating about 25 GB of CBT data every hour. To comply with an RPO limiting data loss to 10% of oblSQL-1's total data and an RTO of 15 minutes, it would be necessary to schedule an incremental replication every 30 minutes.

While a replica VM can satisfy the RPO and RTO requirements of an SLA for business continuity, it simply cannot meet any of the record keeping requirements of a traditional backup schedule. Specifically, a replica can be configured to maintain a maximum of seven snapshots, which are automatically deleted as new snapshots are added. As a result, we also needed to set up an ongoing incremental backup schedule, which sets up a potentially serious conflict with respect to VMware's CBT mechanism.

Running two distinct incremental backup jobs on the same VM with no controls corrupts all incremental backups. Each job will independently reset the VMware CBT data after each backup, which leaves subsequent incremental backups with incomplete data. As a result, there will be unrepairable errors whenever the incremental backups are rolled up into a full backup.

To avoid the corruption of incremental backup data, Vembu BDR gives all control of CBT resets to BDR clients. As a result, both replication and the backup job can be scheduled on the same VM by using the same BDR Backup server. In particular, the server recognizes the data protection configuration and synchronizes CBT rests to keep the backup data of both jobs consistent. As a result, we scheduled incremental replication and backup processes for oblSQL-1 and never incurred data corruption issues restoring the VM from either a backup or a replica.

In our test scenario, we ran an incremental replication followed by an incremental backup every 30 minutes. For both processes, we enabled AppAware VSS snapshots to ensure the consistency of all SQL Server databases. Moreover, with oblSQL-1's datastore incurring 600 write IOPS during the snapshot processes, the most significant overhead occurred at the start of each process when the database was quiesced and snapshots were created.



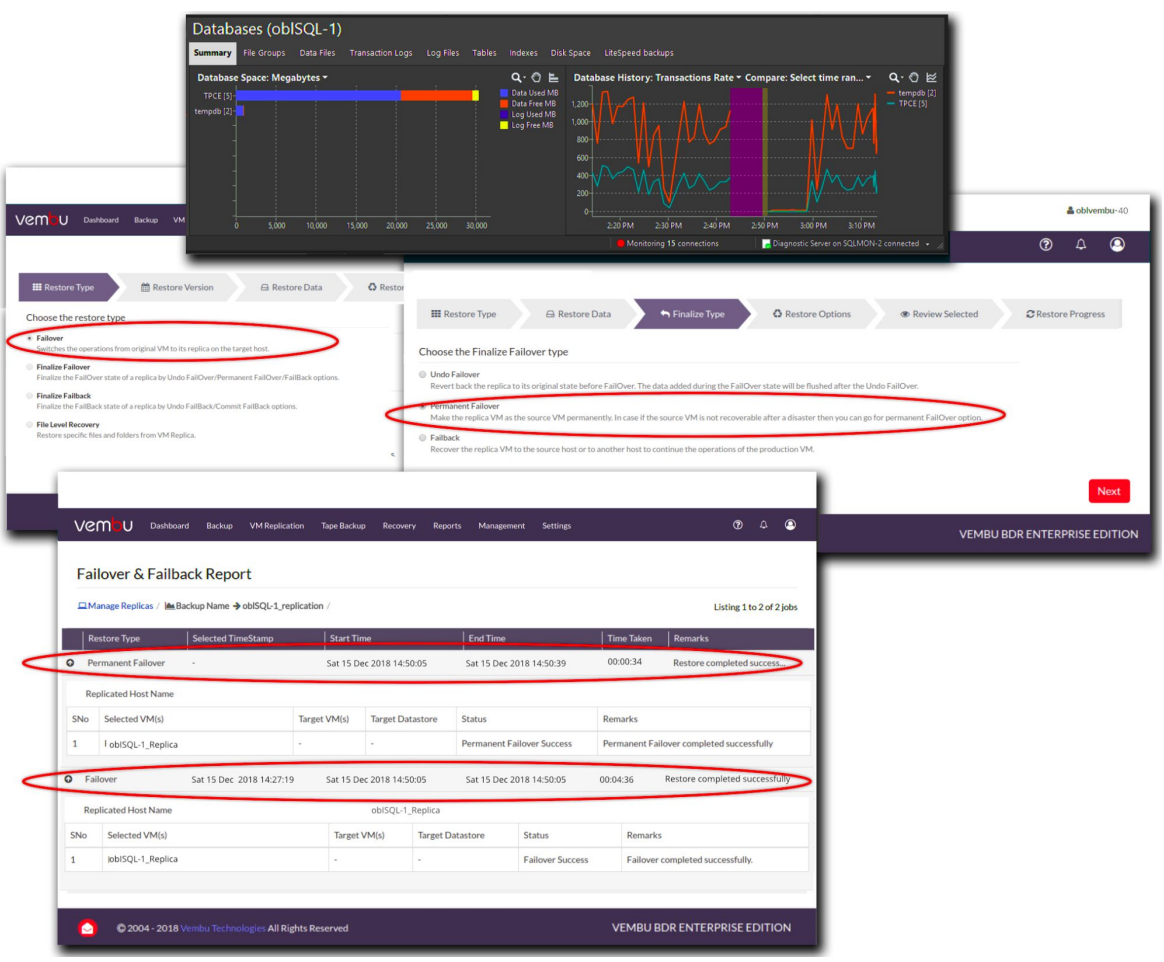
While the IOPS rate on oblSQL-1 remained burdened throughout the initialization of both processes, SQL Server transaction processing levels, with their strong dependence on cached data, quickly rebounded following the quiescence of all SQL Server databases in both processes. As a result, our OLTP business transaction processing application was minimally impacted as the TPC-E transaction rate dropped by about 5% to about 805 cTPS.

What's more, IOPS processing on oblSQL-1 was unencumbered as oblVembu-40 streamed data directly from the logical disk snapshots created at the start of each process. There was no impact on transaction processing from either reading incremental backup data or unwinding VMFS and VSS snapshots.

To complete replication testing, we performed a VM Failover using the Failover and Failback option within Vembu BDR Backup server. We began by shutting down oblSQL-1, which was in the process of running our OLTP application. Next, we opened Failover and Failback option to initiate a restore operation on our VM replica of oblSQL-1.

From the top menu of the Manage Replicas utility, a system administrator can invoke one of four functions:

- **Initialize a failover of production processing to a standby replica,**
- **Finalize the failover process on a running replica, or**
- **Finalize a failback from an active replica to the original production system.**
- **Recover the individual files and folder from the replicated VM.**



In an initial failover, the replica is powered on and booted from a VM snapshot chosen by a system administrator. For our replica of oblSQL-1, the total time to power on and boot using one of the snapshots created in an incremental backup was just over four and one-half minutes.

On completion of the initial failover, an IT administrator can choose to immediately run the replica as a production system. This strategy offers the greatest flexibility for final recovery; however, it also comes with a distinct performance cost. VM I/O performance is burdened with the overhead of running from a VMware copy on write (CoW) snapshot for each logical disk. Specifically, a CoW snapshot doubles the number of I/O operations required to write any new data as existing data must first be written to a snapshot before it can be replaced by new data.

The rationale for this strategy of running from a snapshot is rooted in the use of physical systems. The underlying idea is that the replica is a temporary system, which will only be used until IT is able to recover the original physical system. At that time, the replica, which is masquerading as the production system, would failback—including all data changes incurred while it acted as the production system—to the original physical system. While this strategy makes sense in a world of asymmetric physical servers, in a virtual environment, this strategy is the equivalent of washing paper cups.

The value of a VM is that it has no value. As virtual infrastructure, VMs are consumable objects. As a result, once we had booted our VM and confirmed that the replica was operating with the most recent data, we used the Failover and Failback option to immediately finalize the failover state. In finalizing the failover state, the Vembu BDR Backup server removed all snapshots from the replica VM and eliminated the ability to continue using that VM as a replication target. Nonetheless, in removing all CoW snapshots, failover finalization ensured that all further I/O processing would proceed without any impediments.

Immediately following the booting of oblSQL-1_Replica, we re-launched our transaction processing generator on SQLbench-1. Within 10 minutes, SQL Server had rebuilt its buffer and procedure caches for TPC-E database processing on the new VM. As a result, within 15 minutes of shutting down the original VM, we were processing business transactions at the original production rate of 850 cTPS.

A key value proposition for Vembu BDR Backup server is its ability to directly read and write all backup and restore data directly to and from a datastore snapshot. As a result, Vembu BDR offloads all I/O overhead from production VMs and ESXi hosts, which is critical for maintaining an aggressive DRM strategy in a highly active virtual environment. What's more, the performance of Vembu BDR server in our test environment made it possible to enhance support for a mission-critical OLTP application running on a VM using a combination of incremental backups for backup and replication. As a result, we were able to comply with a 30-minute RPO, restore the VM to a production environment in 5 minutes, and return to full-production level processing of business transactions—850 cTPS—in under 15 minutes.