

The science behind the report:

A Dell EMC Integrated Data Protection Appliance solution delivered faster backup and recovery times for databases and virtual machines

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report [A Dell EMC Integrated Data Protection Appliance solution delivered faster backup and recovery times for databases and virtual machines](#).

We concluded our hands-on testing on October 7, 2020. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on July 29, 2020 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Appendix A: Our results

Table 1: The detailed results from our testing and the percentages of savings for the Dell EMC Integrated Data Protection Appliance (IDPA) solution. We report times in hours:minutes:seconds. Lower numbers are better.

	Dell EMC IDPA DP5800 solution	Vendor Y solution	Percentage savings with Dell EMC solution
Multi-server environment testing			
Time to complete a full backup of a multi-server environment	8:42:28	10:03:57	13.49%
Average time to complete a backup with changes to the databases	1:57:57	6:16:48	68.70%
Time to complete all seven multi-server environment backups	20:30:11	47:44:45	57.06%
SQL Server database backup testing			
Time to complete a full backup of a 3TB SQL Server database	0:57:24	2:24:41	60.33%
Average time to complete a backup with changes to a SQL Server database	0:11:55	0:59:03	79.81%
Time to complete all seven SQL Server backups	2:08:55	8:18:57	74.16%
Total data transferred across the network (Gbps)	7,019	10,456	32.87%

	Dell EMC IDPA DP5800 solution	Vendor Y solution	Percentage savings with Dell EMC solution
Oracle database backup testing			
Time to complete a full backup of a 5TB Oracle database	1:30:58	6:37:00	77.09%
Average time to complete a backup with changes to an Oracle database	0:55:06	5:47:20	84.14%
Time to complete all seven Oracle database backups	7:01:35	41:21:00	83.01%
Recovery testing			
Time to perform a full recovery of a 3TB SQL Server database	0:44:27	1:15:35	41.19%
Time to restore a SQL Server database to a specific point in time	0:50:06	1:23:54	40.29%
Time to perform a full recovery of a 5TB Oracle database	2:00:34	3:35:30	44.05%
Time to restore an Oracle database to a specific point in time	3:35:35	3:46:01	4.62%

Appendix B: Disclosures, diagram, and environment details

Server nodes

Table 2: Detailed configuration information for the servers under test.

Server configuration information	Servers that hosted VMs	Oracle Database 12c server	SQL Server database server
Used in testing phases	Multi-server environment (backup)	Multi-server environment (backup), Oracle database testing (backup and recovery)	Multi-server environment (backup), SQL Server database testing (backup and recovery)
Number of servers	22 (20 that hosted test VMs, 2 that hosted infrastructure VMs)	1	1
BIOS name and version	Dell 1.4.9	Dell 2.4.8	Dell 2.3.10
Non-default BIOS settings	Intel® Turbo Boost enabled, virtualization enabled	Intel Turbo Boost enabled, virtualization enabled	Intel Turbo Boost enabled, virtualization enabled
Operating system name and version/build number	VMware® ESXi™ 6.7.0, 13004448	Red Hat® Enterprise Linux® 7.7, kernel 3.10.0-1062.9.1.el7.x86_64	Windows Server 2016 1607, (OS build 14393.3808)
Date of last OS updates/patches applied	April 29, 2019	March 26, 2020	July 29, 2020
Power management policy	Performance	Performance	Performance
Processor			
Number of processors	2	2	2
Vendor and model	Intel Xeon® Platinum 8168	Intel Xeon Platinum 8168	Intel Xeon Gold 6148
Core count (per processor)	24	24	20
Core frequency (GHz)	2.70	2.70	2.40
Stepping	4	4	4
Memory module(s)			
Total memory in system (GB)	256	256	224
Number of memory modules	8	8	7
Vendor and model	Samsung® M393A4K40CB2-CTD	Samsung M393A4K40CB2-CTD	Samsung M393A4K40CB2-CTD
Size (GB)	32	32	32
Type	DDR4 2666	DDR4 2666	DDR4 2666
Speed (MHz)	2,666	2,666	2,666
Speed running in the server (MHz)	2,666	2,666	2,666
Storage controller			
Vendor and model	Dell PERC H730p	Dell PERC H730p	Dell PERC H730p
Cache size (GB)	8	8	8
Firmware version	50.3.0-1512	50.3.0-1512	50.3.0-1512
Driver version	Dell-shared-perc8 06.806.90.00-1OEM.650.0.0.4598673	Not used – Storage area network (SAN) only	Not used – SAN only

Server configuration information	Servers that hosted VMs	Oracle Database 12c server	SQL Server database server
Local storage (type A)			
Number of drives	2	All volumes were SAN based	All volumes were SAN based
Drive vendor and model	Dell THNSF8120CCSE	N/A	N/A
Drive size (GB)	120	N/A	N/A
Drive information (speed, interface, type)	6Gb SATA, SSD	N/A	N/A
Host bus adapter	QLogic® QLE2742 32Gb FC Adapter	QLogic QLE2742 32Gb FC Adapter	QLogic QLE2742 32Gb FC Adapter
Network adapter			
Vendor and model	Intel Ethernet Server Adapter X710-T	Intel Ethernet Server Adapter X710-T	Intel Ethernet Server Adapter X520-2
Number and type of ports	2 x 10GbE	2 x 10GbE	2 x 10GbE
Driver version	Intel 10 Gigabit Ethernet Network driver 3.7.13.7.14iov-20vmw.670.0.0.8169922	Intel 10 Gigabit PCI Express Network driver 5.1.0-k-rh7.7	Intel Ethernet Server Adapter X520-2 4.1.131.0
Power supplies			
Vendor and model	Dell 80Y26KXA02	Dell 80Y26KXA02	Dell 80Y26KXA02
Number of power supplies	2	2	2
Wattage of each (W)	1,100 (1,260 max)	1,100 (1,260 max)	1,100 (1,260 max)

Dell EMC IDPA DP5800

Table 3: Detailed configuration information for the Dell EMC IDPA under test.

IDPA DP5800	
Platform	
Platform version	IDPA 2.3.0.624689
Avamar version	18.2.0-134
Data Domain version	V6.2.0.5-608869
Date of last OS updates/patches applied	August 23, 2019
VMware ESXi virtualization nodes	
Number of nodes	3
VMware ESXi version	ESXi 6.5.0 build 10719125
VMware vCenter® version	6.5.0.23000
Backup server	
Backup metadata storage (TB)	3.46
NDMP accelerator nodes	0

IDPA DP5800	
Protection storage	
Vendor and model	Dell EMC Data Domain DD6800
Software version	Data Domain OS 6.2.0.5-608869
Total backup storage (TB)	90.926
Number of drives	30
Drive size (TB)	3.6
Drive information (speed, interface, type)	7,200 revolutions per minute (RPM), SAS, HDD
DD Boost status	Enabled

Vendor Y backup appliance

Table 4: Detailed configuration information for the Vendor Y storage appliance under test.

Vendor Y appliance	
Platform	
Platform version	5.0.3-p5-2548
Date of last OS updates/patches applied	August 23, 2019
Vendor Y nodes	
Number of chassis	2
Number of nodes	8 (4 per chassis)
Processors cores per node	10
Memory per node (GB)	96
Network interfaces per node	2 x 1Gbps, 2 x 10Gbps
Network configured per node	1 x 10Gbps
Drives per node	1 x 33.7GB SSD 1 x 471.8GB SSD 3 x 9.9TB, 7,200 RPM, SAS HDD
Total backup storage (TB)	238
Total number of drives	24

Networking

For networking, we used 10Gb connections from all servers to Dell S4048-ON switches. We connected each S4048 switch to a core Dell Networking S6000 40Gb switch via two 40 Gbps backhaul connections.

Table 5: Detailed configuration information for networking components.

Network switch configuration information	Dell Networking S6000
Firmware revision	2.0
Number and type of ports	32 x QSFP 40GbE
Number and type of ports used in test	10 x QSFP 40GbE
Non-default settings used	LACP port-channels for switch interconnections

Network switch configuration information	Dell Networking S4048-ON
Firmware revision	2.0
Number and type of ports	48 x SFP+ 10GbE, 6 x QSFP 40GbE
Non-default settings used	LACP port-channels for switch interconnections

Storage array

We used a Dell EMC Unity 650F all-flash array for storage in all phases. We configured the array with one 300TB drive pool that we subdivided into 37 LUNs. The array held all source data for the VMs, the SQL Server database server, and the Oracle Database server. We used the Dell EMC Unity toolset to create LUNs, create and revert snapshots, and perform similar tasks.

Table 6: Detailed configuration information for the storage array that held source data.

Storage configuration information	Dell EMC Unity 650F all-flash array
Controller firmware revision	5.0.2.0.5.009
Number of storage controllers	2
Number of storage shelves	2
Number of drives per shelf	24
Drive vendor and model number	Dell EMC Unity 005052556 / D3f-2SFXL2-7680
Drive size (TB)	7.68
Drive information (speed, interface, type)	12Gbps, SAS, Flash 4

Software versions across all phases

- Hypervisor
 - VMware ESXi 6.7.0, 13004448
 - VMware vCenter Server 6.7.0, build 10244857
- VMs
 - Windows Server 2016 1607, OS build 14393.3808
 - CentOS Linux 7, kernel 3.10.0-957.21.3.el7.x86_64
- Databases
 - SQL Server 2016 version 13.0.5820.21
 - Oracle Database 12c Enterprise Edition, release 12.2.0.1.0 64 bit
- Bare-metal servers
 - Windows Server 2016 1607, OS build 14393.3808
 - Red Hat Enterprise Linux 7.7, kernel 3.10.0-1062.9.1.el7.x86_64
- IDPA backup software
 - Data Domain Boost for Enterprise Applications (DDBEA) for Oracle (Recovery Manager (RMAN) Agent Version: 19.2.0.0(41))
 - DDBEA for SQL Server (19.4.0.0 build 21)
 - Avamar 18.2.0-134

Testbed overview

The testbed consisted of 24 Dell PowerEdge R740 servers, each of which had dual-port 10Gb NICs and dual-port 32Gb Fibre Channel cards.

For networking, we connected 12 servers (10 for VMs, 2 for bare-metal databases) to one layer 2 switch, 10 servers (for VMs) to a second layer 2 switch, and two infrastructure host servers to a third layer 2 switch. We connected the three 10Gb switches upstream using 80Gb (two 40Gb) link aggregated connections.

We connected each backup appliance to the networking infrastructure with 80Gb of total connectivity. We connected the IDPA top of rack switch to the core 40Gb switch via two 40Gb connections. We connected the Vendor Y appliance to the third of the S4048-ON switches via eight 10Gb connections (one 10Gb connection per node).

For the storage network, we connected a Dell EMC Unity 650 all-flash storage array to each host via 32Gb Brocade Fibre Channel switches.

Dell EMC IDPA setup

The IDPA DP5800 arrived fully assembled in the rack. Dell EMC engineers completed the initial configuration to connect the IDPA switch to the core 40Gb switch with the assistance of our lab staff. We designed all resets in our tests to roll back to the original configuration parameters.

Post configuration/rollback tasks included logging into the IDPA System Manager web interface and verifying that the single sign-on component functioned properly. We also created a storage unit in Data Domain for the Oracle database and another for the SQL Server database.

For VMware backups, we added VMware vCenter to the IDPA, performed discovery of VM guests, and deployed Avamar proxy VMs on the discovered ESXi hosts. Finally, we created schedules, backup and retention policies, and datasets in Avamar for VMs.

For the Oracle bare-metal server, we installed and configured the DDBEA agent to target the storage unit created for the Oracle Database backups. For the SQL Server bare-metal server, we installed the DDBEA plugin and configured it to target the storage unit defined on the Data Domain for SQL backups. We downloaded the DDBEA software directly from the Dell EMC website.

Vendor Y setup

The vendor Y engineers installed two units, each containing four nodes. They connected each of the eight nodes to one of the 10Gb access switches and clustered them together into the initial solution using networking information we provided. We designed all resets in our tests to use the same configuration parameters in the original deployment.

Post configuration/reset tasks included re-registering vCenter, Oracle, and SQL servers within the Vendor Y interface, performing discovery of VM guests, redeploying any necessary agents for proper functionality, and defining service-level agreements (SLAs)/backup policies for those defined and discovered entities.

We installed the agents provided by Vendor Y for the Oracle database server and for the SQL database server, both of which we downloaded directly from Vendor Y.

Testbed diagram

- 40Gbps
- 10Gbps
- 16Gb fibre channel

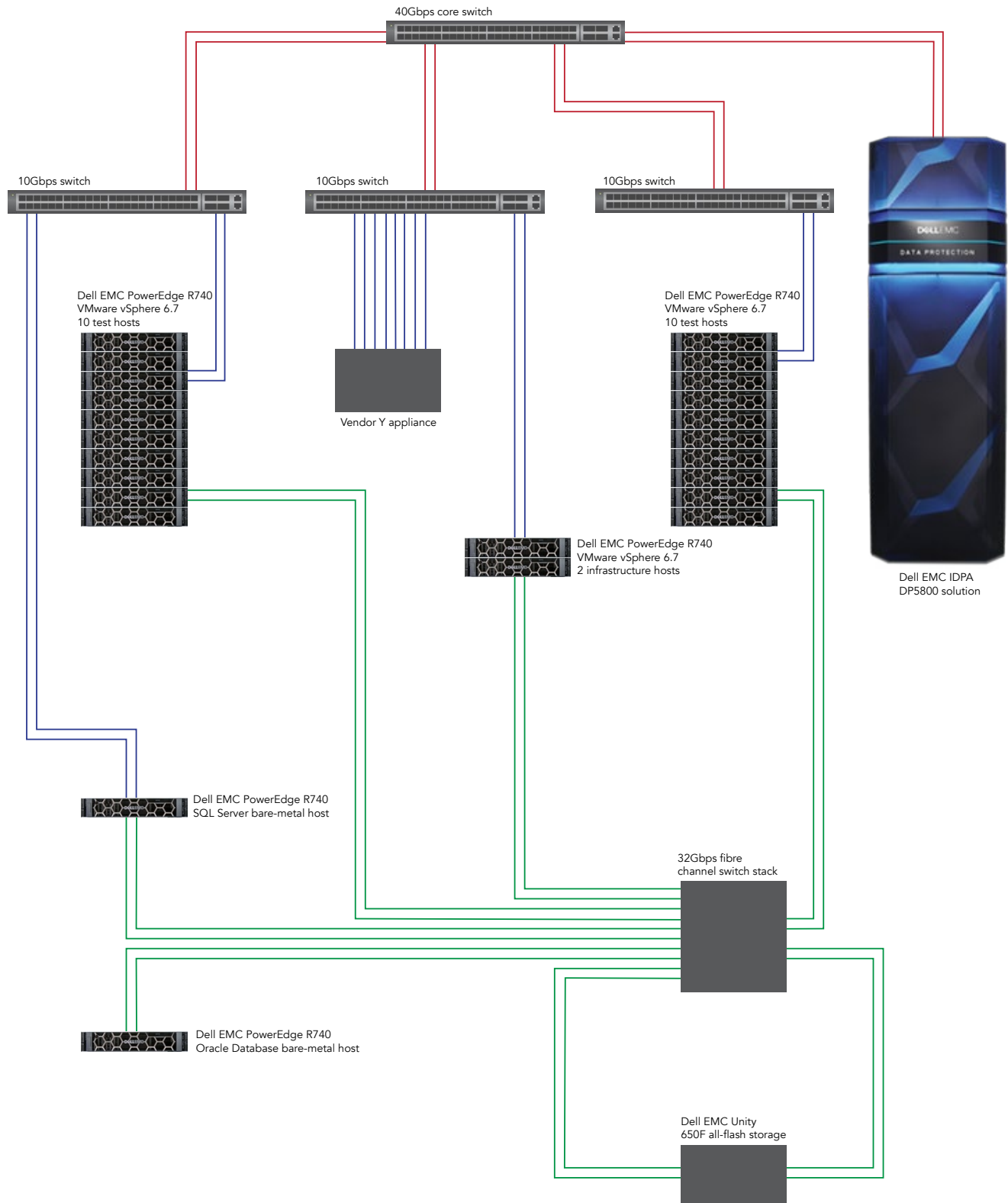


Figure 1: A diagram of our test bed for both solutions.

Appendix C: Project phases and general methodology overview

Definition and description of project phases

For this study, we tested both Vendor Y and the Dell IDPA 5800 solution in three phases, described below. We strive to include sufficient disclosure information in this document, but where additional detail is required, please contact us at info@principledtechnologies.com for more information or specifics on scripts, configurations, and settings.

Phase 1: Combined VMs and bare-metal database servers (backup only)

In this scenario, we tested a combination of a VMware virtual environment at scale combined with two large bare-metal database servers – one running Microsoft SQL Server and one running Oracle Database. We ran the 500 VM backups, the SQL Server database backup, and the Oracle Database backup concurrently, and we staggered the VMs by minute in 10 groups of 50. Our testing in this phase studied backup times and consisted of seven simulated days.

Phase 2: Microsoft SQL Server bare-metal server (backup and recovery)

In this scenario, we tested a single bare-metal server running Microsoft SQL Server. Our testing in this phase studied backup and recovery times and consisted of seven simulated days.

Phase 3: Oracle Database bare-metal server (backup and recovery)

In this scenario, we tested a single bare-metal server running Oracle Database. Our testing in this phase studied backup and recovery times and consisted of seven simulated days.

DDBEA for SQL Server and Oracle Database

For all IDPA testing, we used DDBEA for both SQL Server and Oracle Database in bare-metal scenarios. For more information, see Dell EMC documentation.

General methodology overview

This section describes, at a high level, the overall processes we used.

For Phase 1, Phase 2, and Phase 3 tests, we used combinations of Ansible, PowerShell, Python, and Bash scripts, as well as some manual work through GUIs when necessary.

After we performed a one-time configuration of the entire testbed and all scenarios, each of the three phases followed the same basic pattern:

1. For each test iteration, reset all resources.
2. Run phase-specific tests with data collection and analysis.

One-time configuration

Before testing, we configured all resources to a consistent start state and captured applicable snapshots, gold template VMs, and so forth to ensure repeatable testing.

In this one-time configuration step, we deployed all physical resources, and created “Gold State” snapshots that allowed us to return the testbed to a consistent state. The configuration consisted of the following high-level steps:

1. Deploy and configure infrastructure.
2. Deploy and configure the appliances.
3. Deploy and configure testbed resources.
4. Generate data changesets.
5. Populate the initial data on testbed resources.
6. Create “Gold State” snapshots of infrastructure, the appliances (where possible), and testbed resources.

Resetting step (between test runs)

The "Reset" step brought the testbed back to a known, consistent state (also known as "Gold State") from prior to testing. The general reset methodology for all phases followed the same basic pattern:

1. Shut down testbed resources.
2. Restore "Gold State" snapshots.
3. Reset or restore the appliances.
4. Power on testbed resources.
5. Reconnect testbed resources and the appliances.
6. Perform any further appliance or testbed resource configuration.

Backup testing (all phases)

Backup testing in all phases followed this general process:

1. Perform the reset process.
2. Deploy test source code to any testbed resources.
3. Test connectivity to all infrastructure, testbed, and appliance resources for each simulated day.
4. If the day is greater than one, apply data changeset updates for testbed resources.
5. Collect appliance "pre" statistics.
6. Start any infrastructure or testbed resource monitoring.
7. Record backup start time.
8. Trigger appliance-specific mechanisms for backup of testbed resources.
9. Record backup completion time.
10. Stop any infrastructure or testbed resource monitoring.
11. Collect any infrastructure or testbed resource monitoring data.

Recovery testing (phases 2 and 3 only)

Recovery testing in both phases followed this general process:

1. While following the backup testing methodology, interrupt the process after the targeted simulated day backup completes.
2. Use the appliance's facilities to choose the desired type of recovery (e.g., point-in-time or full).
3. Recover the database to the desired state or time.
4. By using appropriate tools, ensure row count and contents are correct per the recovered state.
5. Stop any infrastructure or testbed resource monitoring.
6. Collect any infrastructure or testbed resource monitoring data.

Appendix D: Phase 1 methodology (combined VMs and bare-metal database servers)

Virtual infrastructure and VM information

Our virtual environment consisted of 20 hypervisor nodes for testing and a cluster of two hypervisor nodes with VMs for Active Directory, DNS, automation servers and a single vCenter Server. Spread across the 20 hypervisor nodes were 500 VMs: 450 Windows Server VMs and 50 CentOS VMs. A subset (50) of the Windows Server VMs also ran smaller virtualized SQL Server databases. We used the VMware native toolsets and best practices for creating templates, deploying VMs between test scenarios, creating virtual networks, storage connectivity, creating datastores, collecting performance statistics, and so on. For brevity, we do not include the detailed installation and configuration steps of all these supporting components.

SQL Server bare-metal server information

On the SQL Server bare-metal server, we installed Windows 2016 Datacenter Edition, Microsoft SQL Server 2016 Enterprise Edition, and SQL Server SP2 and CU13.

We connected the physical SQL Server machine via Fibre Channel to a Dell EMC Unity all-flash array for its storage and configured it for Boot-from-SAN. The SQL Server had a QLogic QLE2742 HBA with two ports mapped to four Fibre Channel ports across both storage processors, and we assigned eight of the array's LUNs to it. In addition, we installed Dell EMC PowerPath on the host, which provided multipathing and failover to the LUNs. We configured LUNs with PowerPath in an Active/Active mode.

Using SQL Server native tools, we loaded the day-1 SQL Server database from flat text files, which we had generated on another server. The SQL Server held a 3TB TPC-H-like database that we used for testing. We configured the SQL Server machine to the appropriate backup appliance for the appropriate phases of testing.

For Phase 1, we defined the TPC-H-like database as having one primary file, four 1TB data files, and one 34GB log file. We used standard data types, indexing, and other schema definition for the database itself, with the exception of modifying certain integer columns to BigInt data type to handle the large amount of rows involved in the testing. We distributed those files across their respective LUNs (see below).

Oracle Database bare-metal server information

On the Oracle Database bare-metal server, we installed Red Hat Enterprise Linux 7.7 and Oracle Database 12c release 2 (12.2.0.1.0) Enterprise Edition.

We connected the physical Oracle Database machine via Fibre Channel to a Dell EMC Unity 650F all-flash array for its storage and configured it for Boot-from-SAN. The Oracle Database server had a QLogic QLE2742 HBA installed with two ports mapped to four Fibre Channel ports across both storage processors, and we assigned 15 of the array's LUNs to it. In addition, we installed Dell EMC PowerPath on the host, which provided multipathing and failover to the LUNs. We configured LUNs with PowerPath in an Active/Active mode.

Using Oracle Database native tools, we loaded the day 1 5TB TPC-H-like dataset from flat text files. We configured the Oracle Database machine to the appropriate backup appliance for the appropriate phases of testing.

We used 480 32GB datafiles for the database. We distributed those files across 11 LUNs (see below).

Storage layout and configuration

See the hardware disclosure in Appendix A for the storage array information. In addition to the LUNs specified below, we used many other LUNs for scratch and working space.

- Virtual infrastructure LUNs
- 10TB LUN for Infrastructure systems (AD, vCenter, etc.)
- 10 30TB LUNs for VMs
- 8 LUNs for SQL Server
- 100GB operating system LUN (Boot-from-SAN)
- 2TB log file LUN, containing SQL Server LDF file
- 6TB scratch LUN, containing change files
- 1TB temp file LUN, containing 8 SQL Server tempdb files
- 4 2TB data LUNs (.NDF files)
- 15 LUNs for Oracle Database
- 100GB LUN for operating system (Boot-from-SAN)
- 512GB LUN for software and alert area
- 512GB LUN for undo files
- 11 2TB LUNs for datafile
- 2 10TB LUNs for temporary tablespace and archive logs

Reset description

Between test simulations, we first disconnected the applicable backup appliance from VMware vCenter, all VMs, and both bare-metal database servers. Then we powered down all VMs and bare-metal servers, and then restored the Unity LUNs, which provided backing storage. We then reset each appliance to its initial state. In the case of IDPA, this consisted of removing DDBEA storage units for database instances, rolling back Avamar to a "Gold State" checkpoint, and performing a filesystem clean. In the case of Vendor Y, we performed a factory reset.

We then powered on all testbed resources and connected the testbed resources to the backup appliance. For IDPA, we added VMware vCenter and VM clients to Avamar, deployed Avamar proxies, and created backup policies for 10 groups of 50 hosts each. For Vendor Y, the reconnection process was similar, although we did not need to deploy proxies, and the test script handled the grouping of VMs instead of Vendor Y's native policy mechanism. Finally, in the case of IDPA, we recreated DD Boost storage units for Oracle Database and Microsoft SQL bare-metal servers and connected DDBEA on SQL Server and likewise configured dd-util and Oracle RMAN.

Backup testing

For backup testing with IDPA, we backed up VMs using the mccli command line utility to trigger backup policies on Avamar. We backed up bare-metal Oracle Databases using RMAN with DDBEA. We backed up bare-metal Microsoft SQL Server databases using DDBEA by way of the ddbmsqlsv.exe command line utility. For backup testing with Vendor Y, we triggered all backups using the appliance's REST API.

For both appliances, backups for all 500 VMs ran concurrently with each other and with backups for databases on bare-metal. We staggered VM backup start times in 10 groups of 50 VMs each, one minute apart. We backed up the bare-metal databases on a per-database basis.

Appendix E: Phase 2 methodology (Microsoft SQL Server bare metal)

SQL Server bare-metal server installation overview

On the SQL Server bare-metal server, we installed Windows 2016 Datacenter Edition, Microsoft SQL Server 2016 Enterprise Edition, and SQL Server SP2 & CU13.

We connected the physical SQL Server machine via Fibre Channel to a Dell EMC Unity all-flash array for its storage and configured it for Boot-from-SAN. The SQL Server had a QLogic QLE2742 HBA installed with two ports mapped to four Fibre Channel ports across both storage processors and was assigned to eight of the array's LUNs. In addition, we installed Dell EMC PowerPath on the host, which provided multipathing and failover to the LUNs. We configured LUNs with PowerPath in an Active/Active mode.

Using SQL Server native tools, we loaded the day 1 SQL Server database from flat text files, which we generated on another server. The SQL Server held a 3TB TPC-H-like database that we used for testing. We configured the SQL Server machine to the appropriate backup appliance for the appropriate phases of testing.

For Phase 2, we defined the TPC-H-like database as having one primary file, four 1TB data files and one 34GB log file. We used standard data types, indexing, and other schema definition for the database itself, with the exception of modifying certain integer columns to Bigint data type, to handle the large number of rows involved in the testing. We distributed those files across their respective LUNs (see below).

SQL Server storage layout and configuration

See the hardware disclosure in Appendix A for the storage array information.

We configured and mapped our SQL Server LUNs as follows:

- 1 100GB Operating System LUN (boot from SAN)
- 1 2TB Log file LUN (.LDF files)
- 1 1TB Temp file LUN (8x temp files)
- 1 6TB Scratch LUN (DB & change files)
- 4 2TB Data LUNs (.NDF files)

Reset description

Between test simulations, we reset the applicable backup appliance by rolling the appliance back to factory default settings. We then configured the appliance with our network and naming conventions. We reset the SQL Server bare-metal database by restoring its Gold snapshot on the storage array, bringing it back to a basic SQL Server state and then adding in the appropriate appliance backup agent software.

Testing

Backup testing

We backed up the SQL Database for the IDPA by running full backups and transaction log backups every simulated day. For Vendor Y, we followed its recommended procedure for backing up SQL Server databases to capture one full backup on day 1, and incremental and transaction log backups for the succeeding days.

Recovery testing

For IDPA, we first dropped the database, then used DDBEA to restore the database. We used this procedure for both types of recoveries.

For Vendor Y, we had two types of recovery procedures. For full recovery, after dropping the database, we used its recommended procedure to recover the database. For the point-in-time recovery, we chose a specific time value on the appliance and recovered the database to that specific time.

Appendix F: Phase 3 methodology (Oracle Database bare metal)

Oracle Database bare-metal server information

On the Oracle Database bare-metal server, we installed Red Hat Enterprise Linux 7.7 and Oracle Database 12c Release 2 (12.2.0.1.0) Enterprise Edition.

We connected the physical Oracle Database machine via Fibre Channel to a Dell EMC Unity 650F all-flash array for its storage and configured it for Boot-from-SAN. The Oracle Database server had a QLogic QLE2742 HBA installed with two ports mapped to four Fibre Channel ports across both storage processors, and we assigned 15 of the array's LUNs to it. In addition, we installed Dell EMC PowerPath on the host, which provided multipathing and failover to the LUNs. We configured LUNs with PowerPath in an Active/Active mode.

Using Oracle Database native tools, we loaded the day 1 5TB TPC-H-like dataset from flat text files. We configured the Oracle Database machine to the appropriate backup appliance for the appropriate phases of testing.

We used 480 32GB datafiles for the database. We distributed those files across 11 LUNs (see below).

Storage layout and configuration

See the hardware disclosure in Appendix A for the storage array information.

We configured and mapped our Oracle Database LUNs as follows:

- 100GB LUN for operating system (Boot-from-SAN)
- 512GB LUN for software and alert area
- 512GB LUN for undo files
- 11 2TB LUNs for datafile
- 2 10TB LUNs for temporary tablespace and archive logs

Reset description

Between test simulations, we reset the applicable backup appliance by erasing all backup data and performing a garbage collection. When that completed, we configured the appliance for the Oracle Database. We reset the Oracle Database by shutting down the server, applying the gold-snapshots for the LUNs for Oracle Database and restarting the server.

Testing

Backup testing

We backed up Oracle Database for the IDPA by running full (level 0) backups and archive-log backups every day. For Vendor Y, we followed its recommended procedure for backing up Oracle Databases to capture one full backup on day 1 and subsequent incremental and archive-log backups for the succeeding days.

Recovery testing

For IDPA recovery testing, we deleted its datafiles for the data under test and used RMAN to restore the data and return the database to service. We used this procedure for both full and point-in-time recoveries.

For Vendor Y, we used two recovery procedures. For full recovery, after deleting the datafiles, we used the vendor's recommended procedure to restore the data and recover the database. For the point-in-time recovery, we added one additional step of copying datafiles from the network file system to local LUNs.

Appendix G: Data changeset information

Building datasets for VMs (Phase 1)

We created initial data and changesets for Microsoft Windows and CentOS Linux VMs with proprietary command line utilities. The utilities generated an initial fileset (in '/fs1' for Linux and 'X:' for Windows), and subsequent invocations applied a fixed percentage change in the generated files. We used the following steps to create the initial 65GB dataset and apply changes for Phase 1 VMs. The following commands are for CentOS Linux, but the commands for Microsoft Windows are identical except for folder paths.

Notes:

- \$SEED varied from 1,000 to 1,500, indicating the VM's unique index in the list of VMs.
- \$DAY is the number of simulated day minus one, from 1 through 6 (we applied no updates on the first day).

1. When preparing the environment, invoke the datagen-v4 binaries:

```
datagen-v4 --generate-fs-v4 --fs-total-length=69793218560 --fs-mean-file-length=1000000 --client-number=$(echo "$SEED/10" | bc) --base-rate=50 --local-dest-dir=/fs1/ --base-change-rate=1 --change-rate=3 --fs-standard-deviation=50 --fs-min-files-per-dir=50 --fs-max-files-per-dir=500 --fs-min-dirs-per-level=10 --fs-max-dirs-per-level=20 --base-compression-factor=210 --compression-factor=210 --fs-change-cluster-probability=999 --fs-base-change-cluster-probability=500 --revision=$REV --base-seed=$SEED --loop-delay=7 --degen-mode=false --debug=false
```

2. For each simulated day's data update (days 2-7), run the following:

```
datagen-v4 --generate-fs-v4 --fs-total-length= 69793218560 --fs-mean-file-length=1000000 --client-number=$(echo "$SEED/10" | bc) --base-rate=50 --local-dest-dir=/fs1/ --base-change-rate=1 --change-rate=3 --fs-standard-deviation=50 --fs-min-files-per-dir=50 --fs-max-files-per-dir=500 --fs-min-dirs-per-level=10 --fs-max-dirs-per-level=20 --base-compression-factor=210 --compression-factor=210 --fs-change-cluster-probability=999 --fs-base-change-cluster-probability=500 --revision=$DAY --base-seed=$SEED --loop-delay=7 --degen-mode=false --debug=false
datagen --generate-fs-v3 --fs-total-length=104689827 --fs-mean-file-length=1000000 --client-number=$(echo "$SEED/10" | bc) --base-rate=400 --local-dest-dir=/fs1/ --base-change-rate=3 --change-rate=3 --fs-standard-deviation=400 --fs-min-files-per-dir=50 --fs-max-files-per-dir=500 --fs-min-dirs-per-level=10 --fs-max-dirs-per-level=20 --base-compression-factor=210 --compression-factor=210 --fs-change-cluster-probability=600 --fs-base-change-cluster-probability=600 --revision=0 --base-seed=$DAY
```

Building datasets for 50 SQL Server VMs (Phase 1)

We used the following steps to create the initial dataset and changesets for the 50 Phase 1 VMs running SQL Server.

1. Increase the update-percentage parameter to 5 percent, and compile the TPC-H-like dbgen program (version 2.18.0):

```
# increase "update %" to 5%
cp dss.h dss.h-orig
sed -i 's/^#define\s*UPD_PCT\s*[0-9]*\s*/#define UPD_PCT 500/' \
    dss.h
make clean all CC=gcc WORKLOAD=TPCH MACHINE=WINDOWS DATABASE=SQLSERVER
```

2. Create the initial 65GB TPC-H-like dataset:

```
./dbgen -s 65 -f
mkdir -p /media/share/tmp/sql/seed
mv *.tbl /media/share/tmp/sql/seed
```

3. Create the TPC-H-like update sets:

```
./dbgen -s 65 -U 6 -f
mkdir -p /media/share/tmp/sql/update
mv *.tbl.u? delete.? /media/share/tmp/sql/update/
```

Building datasets for Oracle Database bare-metal server (phases 1 and 3)

We used the following steps to create the initial dataset and the changesets for scenarios with Oracle Database bare-metal testing.

1. Increase the update-percentage parameter to 5 percent, and compile the TPC-H-like dbgen program (version 2.18.0):

```
# increase "update %" to 5%
cp dss.h dss.h-orig
sed -i 's/^\#define\s*UPD_PCT\s*[0-9]*\$/\#define UPD_PCT 500/' \
    dss.h
make clean all CC=gcc WORKLOAD=TPCH MACHINE=LINUX \
    DATABASE=ORACLE
```

2. Create the initial 5TB TPC-H-like dataset:

```
./dbgen -f -s 5000
mkdir -p /mnt/data_5000/update/
mv *.tbl /mnt/data_5000/
```

3. Create the TPC-H-like update set:

```
./dbgen -f -s 5000 -U 7
mv *.tbl.u? delete.? /mnt/data_5000/update/
```

Building changesets for SQL Server bare-metal server (phases 1 and 2)

We used the following steps to create the initial dataset and the changesets for scenarios with SQL Server bare-metal testing.

1. Increase the update-percentage parameter to 5 percent, and compile the TPC-H-like dbgen program (version 2.18.0):

```
cp dss.h dss.h-orig
sed -i 's/^\#define\s*UPD_PCT\s*[0-9]*\$/\#define UPD_PCT 500/' dss.h
# create dbgen executable for MSSQL
make clean all CC=gcc WORKLOAD=TPCH MACHINE=LINUX DATABASE=SQLSERVER
```

2. Create the initial 3TB TPC-H-like dataset:

```
./dbgen -f -s 3000
mkdir -p /mnt/data_3000/update/
mv *.tbl /mnt/data_3000/
```

3. Create the TPC-H-like update set:

```
./dbgen -f -s 3000 -U 7
mv *.tbl.u? delete.? /mnt/data_3000/update/
```

Read the report at <http://facts.pt/3jZYw3K> ►

This project was commissioned by Dell Technologies.



Facts matter.®

Principled Technologies is a registered trademark of Principled Technologies, Inc. All other product names are the trademarks of their respective owners.

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:

Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.