
Set shorter backup and recovery windows for a virtual machine environment with a Dell EMC Integrated Data Protection Appliance solution

The Dell EMC solution backed up and recovered VMs in a heterogeneous environment faster than a similarly sized solution from a competitor

Shorter backup windows for production VMs

Less downtime from data loss or corruption

The science behind the report:

Set shorter backup and recovery windows for a virtual machine environment with a Dell EMC Integrated Data Protection Appliance solution

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 Set shorter backup and recovery windows for a virtual machine environment with a Dell EMC Integrated Data Protection Appliance solution.

We concluded our hands-on testing on January 13, 2021. 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 September 14, 2020 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Appendix A: Our results

Table 1: Time to complete backup of VMs in hours and minutes.

Simulated day	New VMs	Total VMs	Dell EMC™ IDPA DP8800 solution	Vendor Z solution	Backup time difference
1	100	100	0:53	1:01	0:07
2	100	200	1:19	1:19	0:00
3	100	300	1:25	1:33	0:07
4	100	400	1:35	1:43	0:08
5	100	500	1:39	2:00	0:21
6	100	600	1:57	2:29	0:31
7	100	700	1:57	2:31	0:34
8	100	800	2:01	2:42	0:40
9	100	900	2:12	2:58	0:45
10	100	1,000	2:37	3:18	0:40
11	0	1,000	1:46	2:28	0:42
12	0	1,000	1:39	2:20	0:40
13	0	1,000	1:40	2:17	0:36
14	0	1,000	1:40	2:25	0:44
Total time			24:27	31:10	6:43

Table 2: Physical storage consumed by each backup in TB.

Simulated day	New VMs	Total VMs	Dell EMC IDPA DP8800 solution	Vendor Z solution	Storage difference
1	100	100	3.59	4.50	0.91
2	100	200	7.19	8.50	1.32
3	100	300	10.84	12.31	1.47
4	100	400	14.53	16.57	2.04
5	100	500	18.28	20.72	2.44
6	100	600	22.09	24.95	2.86
7	100	700	25.94	29.38	3.44
8	100	800	29.83	33.84	4.01
9	100	900	33.77	38.64	4.87
10	100	1,000	38.11	43.19	5.08
11	0	1,000	38.64	44.00	5.36
12	0	1,000	39.18	44.89	5.71
13	0	1,000	39.69	45.77	6.08
14	0	1,000	40.21	46.65	6.44

Table 3: Physical storage growth for each backup in TB.

Simulated day	New VMs	Total VMs	Dell EMC IDPA DP8800 solution	Vendor Z solution	Storage growth difference
1	100	100	3.59	4.50	0.91
2	100	200	3.59	4.00	0.40
3	100	300	3.65	3.81	0.16
4	100	400	3.69	4.26	0.56
5	100	500	3.75	4.15	0.40
6	100	600	3.81	4.23	0.42
7	100	700	3.85	4.43	0.58
8	100	800	3.89	4.46	0.57
9	100	900	3.94	4.80	0.86
10	100	1,000	4.34	4.54	0.21
11	0	1,000	0.53	0.82	0.28
12	0	1,000	0.54	0.88	0.35
13	0	1,000	0.52	0.89	0.37
14	0	1,000	0.52	0.88	0.36

Table 4: Time to complete restores of instant access VMs in hours and minutes.

	Dell EMC IDPA DP8800 solution	Vendor Z solution	Restore time difference
Restore scenario with 5 instant access VMs			
Average instant access restore	0:00:31	0:02:02	0:01:31
Median instant access restore	0:00:32	0:01:52	0:01:20
Restore scenario with 10 instant access VMs			
Average instant access restore	0:00:33	0:07:40	0:07:07
Median instant access restore	0:00:33	0:05:00	0:04:27

Appendix B: Disclosures, diagram, and environment details

Server nodes

Table 5: Detailed configuration information for the servers under test.

Server configuration information	22 Dell EMC™ PowerEdge™ R740 servers hosting virtual machines (VMs)
Number of servers	22 (20 hosting test VMs, 2 hosting infrastructure VMs)
Processor	
BIOS name and version	Dell 1.4.9
Non-default BIOS settings	Intel® Turbo Boost enabled; Virtualization enabled
Operating system name and version/build number	VMware ESXi, 6.7.0, 13004448
Date of last OS updates/patches applied	April 29, 2019
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel Xeon® Platinum 8168
Core count (per processor)	24
Core frequency (GHz)	2.70
Stepping	4
Memory module(s)	
Total memory in system (GB)	256
Number of memory modules	8
Vendor and model	Samsung M393A4K40CB2-CTD
Size (GB)	32
Type	DDR4
Speed (MHz)	2,666
Speed running in the server (MHz)	2,666
Storage controller	
Vendor and model	Dell PERC H730p
Cache size (GB)	8
Firmware version	50.3.0-1512
Driver version	Dell-shared-perc8 06.806.90.00-1OEM.650.0.0.4598673

Server configuration information	22 Dell EMC™ PowerEdge™ R740 servers hosting virtual machines (VMs)
Local storage	
Number of drives	2
Drive vendor and model	Dell THNSF8120CCSE
Drive size (GB)	120
Drive information (speed, interface, type)	6Gbps, SATA, SSD
Host bus adapter	QLogic QLE2742 32Gb FC Adapter
Network adapter	
Vendor and model	Intel Ethernet Server Adapter X710-T
Number and type of ports	2 x 10GbE
Driver version	Intel 10 Gigabit Ethernet Network Driver 3.7.13.7.14iov-20vmw.670.0.0.8169922
Power supplies	
Vendor and model	Dell 80Y26KXA02
Number of power supplies	2
Wattage of each (W)	1,100 (1,260 max)

Dell EMC IDPA DP8800

Table 6: Detailed configuration information for the Dell EMC IDPA under test.

IDPA DP8800	
Platform	
Platform version	IDPA 2.5.0.645324
Avamar version	19.2.0-6
Data Domain version	7.1.0.5-643969
Date of last OS updates/ patches applied	September 14, 2020
ESXi nodes	
Number of nodes	3
VMware® ESXi™ version	ESXi 6.7.0 update 3 Build 15160138
VMware vCenter® version	6.7.0.42000
Backup server	
Vendor and model	Avamar Grid
Software version	Avamar 19.2.0-6
Utility nodes	1
M1200 Storage Nodes (16TB)	4
NDMP Accelerator Nodes	1

IDPA DP8800			
Protection storage			
Vendor and model	Dell Data Domain DD9800		
Software version	Data Domain OS 7.1.0.5-643969		
Total backup storage (TB)	340		
Number of drives	4	120	16
Drive size	400 GB	4 TB	800 GB
Drive information (speed, interface, type)	SHSAT6P400GLM21EMC, SATA, SSD	HITACHI HUS72604CLAR4000, 7200RPM, SAS, HDD	HITACHI H4SMR328CLAR800, SAS, SSD
DD Boost status	Enabled		

Vendor Z backup appliance

Table 7: Detailed configuration information for the Vendor Z storage appliance under test.

Vendor Z appliance		
Platform version	3.3.1	
Date of last OS updates/ patches applied	November 16, 2020	
Processors	Intel Xeon Gold 6138 CPU @ 2.00 GHz	
Number of Processors	2	
Cores per processor	20	
System memory (GB)	384	
Number of DIMMs	24	
DIMM type	Samsung DDR4 SH5724G4UNC26P2-SC	
Network interfaces	4 x 1Gbps, 8 x 10Gbps	
Network configuration	8 x 10Gbps LACP bonded	
Drives	5 x 1.5TB SAS HD (for control)	45 x 8TB (for storage)
Total backup storage (TB)	210	
Total number of drives	50	

Networking

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

Table 8: 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
Number and type of ports used in test	Varied x SFP+ 10GbE, 2 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 through all phases. We configured the array with one 300TB drive pool divided into 37 LUNs. The array held all source data for the VMs. We used the Dell EMC Unity toolset for LUN creation, snapshot creation and reversion, and other purposes for our test resets.

Table 9: 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
  - Microsoft Windows Server 2016 1607, OS Build 14393.3808
  - CentOS Linux 7, Kernel 3.10.0-957.21.3.el7.x86_64
- Database
  - Microsoft SQL Server 2016 version 13.0.5820.21
- IDPA backup software
  - Avamar 19.2.0-6

Testbed overview

The testbed consisted of 22 Dell PowerEdge R740 servers, each of which had dual-port 10Gb NICs and dual-port 32Gb Fibre Channel cards.

For networking, we connected 10 servers (for VMs) to one layer 2 switch, another 10 servers (also 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 (2x40Gb) link aggregated connections.

We connected each backup appliance to the networking infrastructure with 80 Gb of total connectivity. We connected the IDPA top of rack switch via two 40Gb connections to the core 40Gb switch. We connected the Vendor Z appliance to the third S4048-ON switch via eight 10Gb connections, all of which we configured as members of an LACP bond.

For the storage network, we connected a Dell Storage Unity 650 All-Flash storage array to each host via 32Gb Brocade fibre channel switches.

Dell EMC IDPA setup

The IDPA DP8800 appliance arrived fully assembled in the rack. Dell Engineers completed the initial configuration, with the assistance of our lab staff to connect the IDPA switch to the core 40Gb switch with redundant connections for a total of 80Gbps in an LACP bond. Any appliance reset that occurred during testing rolled back the system 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.

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.

Vendor Z setup

The vendor engineers installed the controller, shelf, and all disks. We connected each of the eight 10Gb NICs to one of the 10Gb access switches, and LACP bonded together into the initial solution using networking information that we provided. Throughout our tests, any resets returned the configuration parameters to the original deployment settings.

Post configuration/reset tasks included re-registering vCenter within the Vendor Z interface, performing discovery of VM guests, and defining SLAs/backup policies for defined and discovered entities.

Testbed diagram

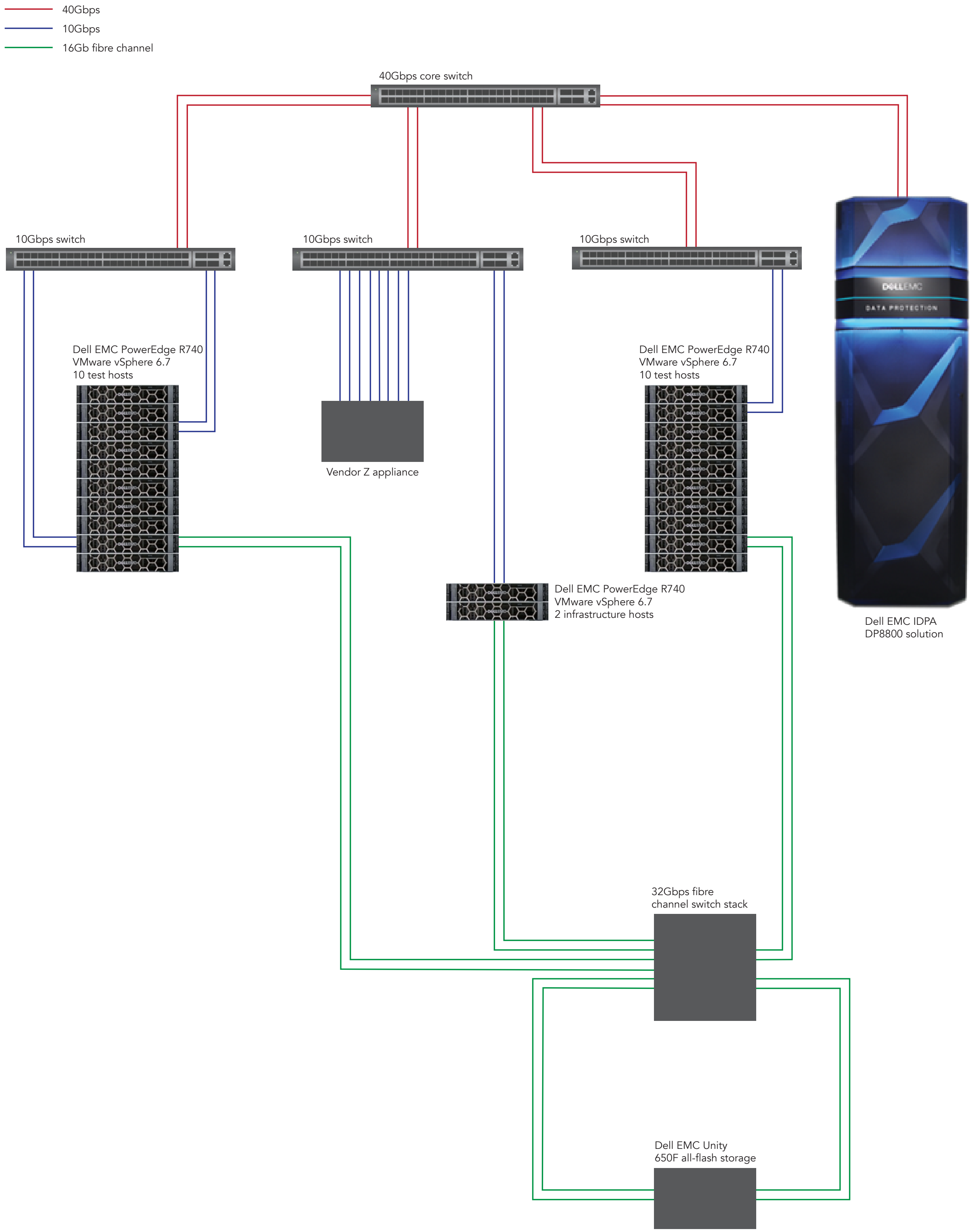

Figure 1: A diagram of our test bed for both solutions.

Appendix C: General methodology overview

For this study, we tested both Vendor Z and the Dell IDPA DP8800 solution as described in the following sections. We strive to include sufficient disclosure information in this document, but please contact us at info@principledtechnologies.com for additional information or specifics about scripts, configurations, and settings.

1,000 VM scale-up test description

In this scenario, we tested appliance backup performance by measuring the time each solution required to complete backups of VMware VMs. This test simulated 14 days of data growth across 1,000 virtual machines, starting with 100 VMs on the first day with an additional 100 new VMs on the second day, as well as the original 100. To simulate a day of data growth, we ran a script that executed changes to the VMs, followed by a backup that would capture those changes. We continued growing the VM count through day 10. On day 11, we stopped adding new VMs. On each day, the solutions performed backups of all VMs concurrently (limited by the scheduling capability of the appliance), but with a small delay between successive groups of 100 VMs.

General methodology overview

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

We used combinations of Ansible, PowerShell, Python, and Bash scripts, as well as some manual work through GUIs when necessary.

At a high level, the testing consisted of the following:

1. A one-time configuration of the testbed
2. For each iteration, we:
    a. Reset all testbed resources and appliance
    b. Ran the test
    c. Collected and analyzed data

One-time configuration

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

We deployed all physical resources, and created "Gold State" snapshots that allowed us to return the testbed to a baseline state. The configuration process 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, baseline 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 the testbed resources and appliances.
6. Perform any further appliance or testbed resource configuration.

Backup testing

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. On each day after day 1, 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

Recovery testing in both solutions followed this general process:

1. While following the backup testing methodology, run the backups until the full 14-day cycle is complete.
2. Use the appliance's facilities to choose the desired type of recovery (e.g., point-in-time or Instant Recovery).
3. Recover the indicated number of VMs to the desired time.
4. By using the appropriate tools, ensure VM contents and state are correct per the recovery type and time.
5. Stop any infrastructure or testbed resource monitoring.
6. Collect any infrastructure or testbed resource monitoring data.

Appendix D: Phase 1 methodology

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 VMware vCenter Server. Spread across the 20 hypervisor nodes were 1,000 VMs: 600 Windows Server VMs and 400 CentOS VMs. A subset (200) 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 for each of these supporting components.

Storage layout and configuration

For the storage array information, see the hardware disclosure in Appendix B: Disclosures, diagram, and environment details. In addition to the following LUNs, we used many other LUNs for scratch and working space:

- 1 x virtual infrastructure LUNs
- 1 x 10TB LUN for infrastructure systems (AD, vCenter, etc.)
- 10 x 30TB LUNs for VMs

Reset description

Between test simulations, we first disconnected the applicable backup appliance and all VMs from VMware vCenter. Then we powered down all VMs, and restored the Unity LUNs, which provided backing storage. We then reset each appliance to its initial state. In the case of IDPA, we rolled back Avamar to a "Gold State" checkpoint and performed a filesystem clean. In the case of Vendor Z, 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 100 hosts each. For Vendor Z, the reconnection process was similar, although we did not need to deploy proxies, the test script handled the grouping of VMs using Vendor Z's native policy mechanisms.

Backup testing

For the IDPA solution, we backed up VMs using the mccli command-line utility to trigger backup policies on Avamar. For backup testing with the Vendor Z solution, we triggered all backups using the appliance's command-line utilities targeting each group of 100 VMs.

Both appliances performed backups for all VMs concurrently. We staggered VM backup start times in groups of 50 VMs each, one minute apart.

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.

Recovery testing

For IDPA, we used the AUI to select the appropriate number of VMs to restore and to the desired point-in-time. Additionally, the IDPA solution performed Instant Access restores using the same time value as the full recoveries.

For Vendor Z, we had similar types of recovery procedures. For the point-in-time recovery, we chose the same number of VMs and time value on the appliance as on the IDPA and recovered the VMs to new VMs. In addition, the IDPA solution performed Instant Access recoveries in the same manner as the point-in-time restores up to the limits allowed by the appliance.

Appendix E: Data changeset information

Building datasets for Windows and Linux VMs

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 applied changes for Phase 1 VMs. The following commands are for CentOS Linux, but the commands for Microsoft Windows are identical besides the specified folder paths.

Notes:

- $SEED varied from 1,000 to 2,000, 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 did not apply 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 through 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 SQL Server VMs

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]*$/#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/
```

Read the report at http://facts.pt/vgukZoo ▶

This project was commissioned by Dell Technologies.

Principled Technologies®

Facts matter.®