Oracle Database 12c Performance Characterization

Tegile IntelliFlash T4700 All-Flash Storage Array
Executive Summary

This white paper documents testing to characterize the performance of Oracle Database 12c on the Tegile IntelliFlash T4700 All-Flash Storage Array using the HammerDB tool with a TPC-C workload.

For these tests, an aggregate HammerDB score of **8,392,994 Oracle TPM** and **2,877,715 NOPM** was achieved using two Oracle instances connected to separate controllers on the Tegile storage array.

Node A score of 4,189,540 Oracle TPM and 1,435,474 NOPM.
Node B score of 4,203,454 Oracle TPM and 1,442,241 NOPM.

These are not official TPC-C results.

These results highlight the T4700 storage array as a high-performance solution for Oracle Database 12c deployments, and demonstrate the benefits of Tegile’s active/active controller technology.

Tegile all-flash arrays scale to meet the performance requirements of a wide range of database workloads, providing very low latency to dramatically reduce I/O wait times. Furthermore, all Tegile arrays are built with enterprise class SSDs with 10X more endurance, ensuring sustained performance over time.

These results demonstrate outstanding performance and provide proof that customers can deploy Oracle Database 12c on Tegile all-flash arrays with confidence.

Disclaimers

- The objective of this testing is to measure the performance of Oracle Database 12c on Tegile all-flash storage arrays for internal reference and informational purposes.
- These test results were NOT audited and approved by TPC council or HammerDB support.
- Some tuning parameters used to maximize the database performance in order to stress the T4700 storage array are not typically used in production environments. This is a common practice for all TPC testing.
- Comparative HammerDB results from other vendors on public websites may be removed or changed at any time.
Understanding HammerDB  (www.hammerdb.com/about)

HammerDB is a graphical open-source database load testing and benchmarking tool for Windows and Linux platforms. Originally developed for Oracle Database, HammerDB can now be run with many different databases. The HammerDB tool creates a test schema, loads it with data, and simulates a workload of multiple virtual uses against the database.

HammerDB includes a workload derived from the well-known TPC-C benchmark. HammerDB does not implement the full TPC-C benchmark. Official Audited TPC-C benchmarks are extremely costly, time consuming and complex to establish and maintain. HammerDB takes the essence of this benchmark and implements it in a way that can be run at a fraction of the cost. Unlike the TPC-C benchmark, HammerDB does not have a formal auditing procedure for publication. For this reason, HammerDB results should not be compared against published TPC-C benchmarks.

HammerDB produces two primary metrics called Transactions Per Minute (TPM) and New Orders Per Minute (NOPM).

Understanding TPC-C  (www.tpc.org/tpcc)

TPC-C is an industry-standard benchmark which generates an On-Line Transaction Processing (OLTP) workload that simulates a wholesale supplier with sales districts and associated warehouses. A warehouse covers 10 districts, and a district covers 3,000 customers. All warehouses maintain the entire stock of 100,000 items sold by the company.

The TPC-C business flow includes order entry, order processing, order delivery, payment recording, order status checking, and warehouse inventory management. Orders are processed from the local warehouse but can be processed from other warehouses in case of insufficient stock. Order cancellations and sales returns are possible. While the benchmark portrays the activity of a wholesale supplier, TPC-C is not limited to any particular business segment. Rather, it represents any industry that must manage, sell, or distribute a product or service.

From a database perspective, the TPC-C benchmark involves a mix of five concurrent transactions of different types and complexity. The database is comprised of nine types of tables with a wide range of record and population sizes.

TPC-C results are measured in transactions per minute (tpmC).
Benchmark Topology

The system under test has the following configuration:

Software Environment

Oracle Linux 7.3 x86_64 with the Unbreakable Enterprise Kernel 4.1.12-94.5.9.el6uek.x86_64
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 – 64 bit Production
Oracle Grid Infrastructure 12c Release 1 (12.2.0.1.0) for Linux x86-64
HammerDB Release 2.23 for Linux 64-bit

Hardware Environment

1 x Tegile IntelliFlash T4700 All-Flash Storage Array with
  4 x Intel® Xeon® E5-2680 v3 @ 2.50 GHz (12 cores per processor / 48 cores total)
  464 GB memory
  24 x 1 TB SSD
  IntelliFlash version 3.7.0.0.170808(GA)
2 x Dell PowerEdge R930 Server with
  4 x Intel® Xeon® E7-4830 v3 @ 2.10 GHz with multithreading enabled
  (12 cores per processor / 48 cores total / 96 threads)
  768 GB memory (48 x 16 GB DDR-4 dual-rank DIMMs @ 1333 MHz)
  2 x 278 GB internal SAS HDD
  1 x Integrated Dell PowerEdge RAID controller (PERC) H730P adapter (embedded)
  1 x Integrated Broadcom BRCM5720 GbE quad-port Ethernet controller (embedded)
  2 x QLogic QLE2662 16 Gb dual-port fibre channel adapter
  BIOS version 1.1.5
  Firmware version 2.21.21.21

1 x Brocade 6505 16 Gbps 24-port fibre channel switch

1 x Cisco Nexus 5020 10 Gbps Ethernet switch
BIOS Settings

In the Dell System BIOS Settings, Node Interleaving was disabled for non-uniform memory access (NUMA).

Figure 5 - System BIOS Memory Settings
The Performance System Profile was selected to maximize processor frequency and disable certain power-saving features such as C-states.

Figure 6 - System BIOS Profile Settings
Linux Kernel Settings

For optimal performance, Oracle Linux was configured with the following kernel settings in /etc/sysctl.conf as follows:

```
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.sem = 250 32000 100 128
kernel.shmctni = 4096
kernel.shmall = 1073741824
kernel.shmmax = 4398046511104
kernel.panic_on_oops = 1
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
net.ipv4.conf.all_rp_filter = 2
net.ipv4.conf.default_rp_filter = 2
net.ipv4.ip_local_port_range = 9000 65500
```
Storage Configuration

The T4700 storage array was configured with the following settings for all Storage Pools, Project, and LUNs:

Pool A  Controller Configuration: Active/Active
Disks: 12 x 1 TB SSD
RAID: 2-way mirror
Resource Group A

Pool B  Controller Configuration: Active/Active
Disks: 12 x 1 TB SSD
RAID: 2-way mirror
Resource Group B
<table>
<thead>
<tr>
<th>Projects</th>
<th>Pool Name</th>
<th>Data</th>
<th>Snapshot</th>
<th>Reserved</th>
<th>Free</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>Pool-A</td>
<td>270.3 GB</td>
<td>0 B</td>
<td>0 B</td>
<td>3.9 TB</td>
<td>89.3 %</td>
</tr>
<tr>
<td>Project B</td>
<td>Pool-B</td>
<td>269.9 GB</td>
<td>0 B</td>
<td>0 B</td>
<td>3.9 TB</td>
<td>89.3 %</td>
</tr>
</tbody>
</table>

**Figure 8 - Project Settings**

Project A
- Purpose: Generic
- Provisioning: Thin
- Compression: lz4
- Deduplication: Off
- Snapshots: None

Project B
- Purpose: Generic
- Provisioning: Thin
- Compression: lz4
- Deduplication: Off
- Snapshots: None
Node A LUNs  
Access: 16Gb Fibre Channel  
Purpose: Database  
Block Size: 16 KB  
8 x 150 GB LUNs for DATA tablespace  
8 x 130 GB LUNs for redo logs  
8 x 70 GB LUNs for ORDERS tablespace

Node B LUNs  
Access: 16Gb Fibre Channel  
Purpose: Database  
Block Size: 16 KB  
8 x 150 GB LUNs for DATA tablespace  
8 x 130 GB LUNs for redo logs  
8 x 70 GB LUNs for ORDERS tablespace
**Oracle ASM Configuration**

For optimal performance, a database on each node was created using Oracle ASM with the following configuration. Each disk group is comprised of eight LUNs for optimal performance.

![ASM Configuration Assistant](image)

*Figure 10 - ASM Configuration Assistant*
**Oracle Listener Configuration**

The database was configured with four listener processes, each listening on a dedicated 10 Gb network IP connection with a separate port number for each listener. The listener.ora configuration follows:

```
$ cat listener.ora

LISTENER_TPCC_4 = (DESCRIPTION_LIST = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1525)))
LISTENER_TPCC_3 = (DESCRIPTION_LIST = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1524)))
LISTENER_TPCC_2 = (DESCRIPTION_LIST = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1523)))
LISTENER_TPCC_1 = (DESCRIPTION_LIST = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1522))
  (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1522))))
SID_LIST_LISTENER_TPCC_4 = (SID_LIST = (SID_DESC =
  (ORACLE_HOME = /oracle/12c/database) (SID_NAME = HDB))
SID_LIST_LISTENER_TPCC_3 = (SID_LIST = (SID_DESC =
  (ORACLE_HOME = /oracle/12c/database) (SID_NAME = HDB))
SID_LIST_LISTENER_TPCC_2 = (SID_LIST = (SID_DESC =
  (ORACLE_HOME = /oracle/12c/database) (SID_NAME = HDB))
SID_LIST_LISTENER_TPCC_1 = (SID_LIST = (SID_DESC =
  (ORACLE_HOME = /oracle/12c/database) (SID_NAME = HDB))
  (ORACLE_HOME = /oracle/12c/database) (SID_NAME = HDB))

ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_TPCC_1=ON
VALID_NODE_CHECKING_REGISTRATION_LISTENER_TPCC_1=SUBNET
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_TPCC_2=ON
VALID_NODE_CHECKING_REGISTRATION_LISTENER_TPCC_2=SUBNET
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_TPCC_3=ON
VALID_NODE_CHECKING_REGISTRATION_LISTENER_TPCC_3=SUBNET
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_TPCC_4=ON
VALID_NODE_CHECKING_REGISTRATION_LISTENER_TPCC_4=SUBNET
```
To equally distribute the workload across the four listener processes, load balancing was enabled using the tnsnames.ora configuration as follows:

```
$ cat tnsnames.ora
HDB = (DESCRIPTION =
  (LOAD_BALANCE=on)
  (FAILOVER=on)
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1522))
    (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1523))
    (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1524))
    (ADDRESS = (PROTOCOL = TCP) (HOST = db_10) (PORT = 1525))
  )
  (CONNECT_DATA = (SERVICE_NAME = HDB))
)
```
Oracle Database Creation

The Oracle Database was created with the following tablespaces:

```sql
$ cat createHDB.sql

CREATE DATABASE HDB
    CONTROLFILE REUSE
    USER sys IDENTIFIED BY tegile
    USER system IDENTIFIED BY tegile
    SET DEFAULT BIGFILE TABLESPACE
    LOGFILE GROUP 1 '+LOGS/log1' SIZE 500G REUSE,
                  GROUP 2 '+LOGS/log2' SIZE 500G REUSE
    MAXDATAFILES 200
    DATAFILE '+DATA/sys' SIZE 10G REUSE
    SYSAUX DATAFILE '+DATA/sysaux' SIZE 10G REUSE
    DEFAULT TABLESPACE data DATAFILE '+DATA/data' SIZE 500G REUSE
    DEFAULT TEMPORARY TABLESPACE temp TEMPFILE '+DATA/temp' SIZE 100G REUSE
    UNDO TABLESPACE undo DATAFILE '+DATA/undo' SIZE 500G REUSE;

CREATE TABLESPACE orders DATAFILE '+ORDERS/orders' SIZE 500G REUSE;

@/u01/app/oracle/product/12.1.0.2/db_1/rdbms/admin/catalog.sql
@/u01/app/oracle/product/12.1.0.2/db_1/rdbms/admin/catexp.sql
@/u01/app/oracle/product/12.1.0.2/db_1/rdbms/admin/catldr.sql
@/u01/app/oracle/product/12.1.0.2/db_1/rdbms/admin/catproc.sql
@/u01/app/oracle/product/12.1.0.2/db_1/rdbms/admin/catparr.sql
```
Oracle Database Tuning

For optimal performance, the following database tunings were applied:

```bash
$ cat initHDB.ora
  _ash_size=2013265920
  _disk_sector_size_override=TRUE
  _enable_reliable_latch_waits=TRUE
  _fast_cursor_reexecute=TRUE
  _fg_sync_sleep_usecs=500
  _undo_autotune=FALSE
  aq_tm_processes=0
  audit_file_dest='/home/oracle/database/audit'
  audit_trail='NONE'
  commit_logging='BATCH'
  commit_wait='NOWAIT'
  compatible='12.2.0.1.0'
  control_files='+DATA/control'
  db_block_checking=FALSE
  db_block_checksum=FALSE
  db_block_size=16384
  db_cache_advice=off
  db_cache_size=100G
  db_create_file_dest='+DATA'
  db_domain=''
  db_name='HDB'
  db_recovery_file_dest='+DATA'
  db_recovery_file_dest_size=4300000000000
  db_unrecoverable_scn_tracking=FALSE
  db_writer_processes=16
  ddl_lock_timeout=30
  deferred_segment_creation=FALSE
  diagnostic_dest='/home/oracle/database'
  disk_async_io=TRUE
  dispatchers=(PROTOCOL=TCP) (SERVICE=HDB)
  dml_locks=1000
```
filesystemio_options=setall
inmemory_size=0
lock_sga=TRUE
log_checkpoints_to_alert=TRUE
open_cursors=2400
optimizer_capture_sql_plan_baselines=FALSE
optimizer_dynamic_sampling=4
parallel_degree_policy='AUTO'
processes=8000
query_rewrite_enabled=FALSE
remote_login_passwordfile='EXCLUSIVE'
replication_dependency_tracking=FALSE
sga_max_size=250G
shared_pool_size=32G
statistics_level=BASIC
timed_statistics=FALSE
trace_enabled=FALSE
transactions=12000
transactions_per_rollback_segment=1
undo_retention=30
undo_tablespace='undo'
HammerDB Settings

HammerDB settings were configured as follows:

![HammerDB Oracle TPC-C Build Options](image)

Figure 11 - HammerDB Oracle TPC-C Build Options
Figure 12 - HammerDB Oracle TPC-C Driver Options
Figure 13 - HammerDB Virtual User Options

Figure 14 - HammerDB Autopilot Options
Figure 15 - HammerDB Oracle Transaction Counter Options

Figure 16 - HammerDB Connect to Agent Options
Results Summary

An aggregate HammerDB score of **8,392,994 Oracle TPM** and **2,877,715 NOPM** was achieved using two Oracle instances connected to separate controllers on the Tegile storage array.

Node A score of 4,189,540 Oracle TPM and 1,435,474 NOPM.
Node B score of 4,203,454 Oracle TPM and 1,442,241 NOPM.

These are not official TPC-C results.

![Figure 17 – Aggregate HammerDB Result](image-url)
Figure 18 – Results Summary for Node A
Figure 19 – Results Summary for Node B
Please note that the HammerDB tool has a GUI transaction counter that monitors progress during execution of the test. This counter generates fluctuating numbers over the course of the measurement window and therefore should not be used for reporting the final score.

Figure 20 - HammerDB Transaction Counter
The TPM number was calculated using the Transactions metric from the Oracle Automatic Workload Repository (AWR) snapshots collected at the beginning and end of the measurement window. The Transactions metric (number of transactions per second) recorded in the Workload Repository Report Summary is multiplied by 60 to arrive at Transactions Per Minute (TPM).

The TPM number was calculated using the Transactions metric from the Oracle Automatic Workload Repository (AWR) snapshots collected at the beginning and end of the measurement window. The Transactions metric (number of transactions per second) recorded in the Workload Repository Report Summary is multiplied by 60 to arrive at Transactions Per Minute (TPM).

### Report Summary

**Load Profile**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Per Second</th>
<th>Per Transaction</th>
<th>Per Exec</th>
<th>Per Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Time(s):</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>DB CPU(s):</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Background CPU(s):</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Redo size (bytes):</td>
<td>369,349,705.0</td>
<td>5,289.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical read (blocks):</td>
<td>3,976,037.9</td>
<td>56.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block changes:</td>
<td>2,177,236.4</td>
<td>31.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical read (blocks):</td>
<td>1,719.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical write (blocks):</td>
<td>771.2</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read IO requests:</td>
<td>968.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write IO requests:</td>
<td>115.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read IO (MB):</td>
<td>26.9</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write IO (MB):</td>
<td>12.1</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IM scan rows:</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session Logical Read IM:</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User calls:</td>
<td>53,512.9</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parses (SQL):</td>
<td>32,307.9</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hard parses (SQL):</td>
<td>0.8</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Work Area (MB):</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logins:</td>
<td>0.1</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Executes (SQL):</td>
<td>1,440,724.5</td>
<td>20.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rollbacks:</td>
<td>117.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transactions:</td>
<td>69,625.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 21 – Workload Repository Report Summary for Node A
## Report Summary

### Load Profile

<table>
<thead>
<tr>
<th>Metric</th>
<th>Per Second</th>
<th>Per Transaction</th>
<th>Per Exec</th>
<th>Per Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Time(s)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>DB CPU(s)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Background CPU(s)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Redo size (bytes)</td>
<td>370,626,636.8</td>
<td>5,290.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical read (blocks)</td>
<td>3,391,668.8</td>
<td>57.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block changes</td>
<td>2,185,417.1</td>
<td>31.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical read (blocks)</td>
<td>1,750.7</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical write (blocks)</td>
<td>8.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read IO requests</td>
<td>1,014.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write IO requests</td>
<td>1.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read IO (MB)</td>
<td>27.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write IO (MB)</td>
<td>0.1</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IM scan rows</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session Logical Read IM</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User calls</td>
<td>53,725.4</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parses (SQL)</td>
<td>32,426.5</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hard parses (SQL)</td>
<td>0.7</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Work Area (MB)</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logons</td>
<td>0.1</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Executes (SQL)</td>
<td>1,445,467.2</td>
<td>20.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rollbacks</td>
<td>110.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transactions</td>
<td>70,057.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 22 – Workload Repository Report Summary for Node B*
Storage Array Performance

The Tegile T4700 storage array delivered excellent performance for the HammerDB workload.

Browse the publications at hammerdb.com/benchmarks.html to learn about the performance levels achieved using HammerDB.

Please note that publications on public websites may be removed or changed at any time.
About Tegile Systems, Inc.

Tegile Systems is pioneering a new generation of flash-driven enterprise storage arrays that balance performance, capacity, features and price for virtualization, file services and database applications. With Tegile’s line of all-flash and hybrid storage arrays, the company is redefining the traditional approach to storage by providing a family of arrays that accelerate business critical enterprise applications and allow customers to significantly consolidate mixed workloads in virtualized environments.

Tegile’s patented IntelliFlash™ technology accelerates performance and enables inline deduplication and compression of data so each array has a usable capacity far greater than its raw capacity. Tegile’s award-winning solutions enable customers to better address the requirements of server virtualization, virtual desktop integration and database integration than any other offerings. Featuring both NAS and SAN connectivity, Tegile arrays are easy-to-use, fully redundant, and highly scalable. They come complete with built-in snapshot, remote-replication, near-instant recovery, onsite or offsite failover, and VM-aware features. Follow us on Twitter @tegile or visit us at www.tegile.com