Utilizing Persistent Memory to Improve DB Performance & Reduce Costs

Karen Dorhamer, HPE

May 3, 2017
Agenda

- HPE Persistent Memory overview and usage
- Oracle Database use cases
- EnterpriseDB Postgres use case
- Future work: SQL Server on Linux
- Resources
Convergence of memory and storage

Persistent Memory = The speed of memory with the persistence of storage

- Load-Store Data
- Low Latency
- Volatile
- DRAM
- Block-Store Data
- Long Latency
- Non-Volatile
- Flash SDD
- Magnetic HDD

Memory

Persistent Memory

Storage

Speed

Persistence
HPE 8GB NVDIMM

Delivering the performance of memory with the persistence of storage

**Product:** HPE 8GB NVDIMM Module (782692-B21)
**List Price:** $899

**Features / Benefits**
- Breakthrough performance enabling faster business decisions
- Resilient technology designed for maximum uptime
- Complete hardware and software ecosystem for your business workloads

**Ideal for**
- Accelerating database and write caching

**HPE ProLiant Gen9 Servers Supported and OS Drivers**
- DL360 Gen9 and DL380 Gen9 E5-2600v4
- *NEW* HPE factory integration Configure-to-Order (CTO) support
- **Microsoft:** WS2012 R2 (HPE driver) and WS2016 (inbox driver)
- **Linux:** RHEL 7.3 and SLES 12 SP2
The Anatomy of an HPE NVDIMM

Industry-standard Innovation

- Type “NVDIMM-N” (JEDEC standard)
  - Combines DRAM and NAND Flash onto a single DIMM

- Flash used as persistent store
  - Characteristics of DRAM:
    - Capacity (10’s GB)
    - Performance (latency 10’s nanoseconds)
    - Endurance and reliability of DRAM

HPE Innovation

- HPE BIOS: detects and prevents system errors
- HPE byte-addressable Memory: standard interfaces with software partners
- NVDIMM Controller: moves data from DRAM to Flash upon power loss or other trigger
- HPE Smart Storage Battery: provides backup power to HPE NVDIMM-N’s
# HPE Persistent Memory – Gen9 View

<table>
<thead>
<tr>
<th>Software Apps</th>
<th>Block Storage (Existing Apps)</th>
<th>HPE working with industry to fundamentally change apps</th>
<th>Microsoft SQL Server 2016 (1st) SW Apps addressing PMEM technology in byte addressable manner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industry Standard SW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Systems</td>
<td>MSFT: WS2012 R2/WS2016 Linux: RHEL 7.3 SLES 12 SP2</td>
<td>1st Windows driver 1st NVDIMM supported by Linux distributions</td>
<td>VMware support Support for additional programming models</td>
</tr>
<tr>
<td>Persistent Memory</td>
<td>HPE 8GB NVDIMM</td>
<td>1st NVDIMM in the market designed around a server platform</td>
<td>Future Offerings with Increased Capacity and Performance</td>
</tr>
<tr>
<td>HPE Infrastructure</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## 2016-2017 | HPE Innovation | 2017 +
Application Programming Models to Persistent Memory

Existing applications unchanged – writes to special volume specified for certain operations

Applications partially changed - source code re-written to use new APIs for specific data

Application source code manipulates data structures directly in Persistent Memory

Conventional I/O Access

Filesystem APIs

Block I/O

OS Driver
(Block Device Emulation)

Middleware APIs / NVML

EXT4/XFS
Cached/UnCached
DAX (Linux)

NTFS/ReFS
Cached/UnCached
SCM
Block/DAX (Windows)

Native PM Access

Application source code manipulates data structures directly in Persistent Memory

Object Stores

New Apps

Data Analytics

Standard Open Interfaces

EXT4/XFS
AppDirect
DAX (Linux)

NTFS/ReFS
AppDirect
DAX (Windows)

Indirect I/O Access

Indirect PM Access

Direct PM Access

Application source code manipulates data structures directly in Persistent Memory

Object Stores

New Apps

Data Analytics

Standard Open Interfaces

EXT4/XFS
AppDirect
DAX (Linux)

NTFS/ReFS
AppDirect
DAX (Windows)
Linux Distribution Support

– HPE-supported commercial distributions that are NVDIMM-enabled
  – RHEL7.3
    – Full support for block access, filesystem DAX is technology preview, no device DAX
    – Qemu 2.6 not included
    – Release notes specifically mention HPE NVDIMM-N
  – SLES12 SP2
    – Technology preview for block access, file system DAX and device DAX
    – Qemu 2.6 is included but not HPE-tested yet
    – Release notes specifically mention HPE NVDIMM-N

– Community distributions are also NVDIMM-enabled
  – Fedora 24 with 4.7 kernel and newer
  – OpenSUSE Tumbleweed with 4.7 and newer
File system support with DAX
Experimental with ext4 and xfs

– Create a file system on a pmem device
  
  # mkfs -t ext4 /dev/pmem0

– Mount the file system with –o dax option
  
  # mount -o dax /dev/pmem0 /mnt0

– Console/dmesg will display (RHEL7.3 example, xfs similar)
  
  EXT4-fs (pmem0): DAX enabled. Warning: EXPERIMENTAL, use at your own risk
  TECH PREVIEW: ext4 direct access (dax) may not be fully supported.
  Please review provided documentation for limitations.
  EXT4-fs (pmem0): mounted filesystem with ordered data mode. Opts: dax

– Using –o dax on a btt device (pmemXs) is not supported
  – ext4 will fail the mount
  – xfs will successfully mount but will turn off –o dax
    – Only notification is console/dmesg
Improving Oracle database performance with HPE persistent memory
Two Oracle scenarios with NVDIMM

- Oracle redo logs on RHEL file system, NVDIMM with DAX
- Oracle redo logs on Oracle ASM file system, NVDIMM block device
Hardware and software description
HPE ProLiant DL380 Gen9 server

Six HPE 8GB NVDIMM-Ns
- Balanced across the 2 sockets
- Interleaving (per socket) enabled
- Two memory pools presented to the OS (/dev/pmem[01])

Two regular RDIMMs
- One per socket

Red Hat Enterprise Linux 7
Oracle Database Enterprise Edition 12c
Single instance database using file system
Oracle OLTP workload with redo logs on file system on disk vs NVDIMM (with and without DAX mount option)

Workload generator: Swingbench with 26 users, 10 minute load

The higher the better

<table>
<thead>
<tr>
<th>Internal HD</th>
<th>NVDIMM no Dax</th>
<th>NVDIMM with Dax</th>
</tr>
</thead>
<tbody>
<tr>
<td>363,178</td>
<td>578,169</td>
<td>742,283</td>
</tr>
</tbody>
</table>

TPM
**Oracle AWR wait time statistics**

**Internal SAS Disk**

Top 10 Foreground Events by Total Wait Time

<table>
<thead>
<tr>
<th>Event</th>
<th>Waits Time (sec)</th>
<th>Avg(ms)</th>
<th>% DB Wait</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>log file sync</td>
<td>2,643,657</td>
<td>14.9K</td>
<td>5.62</td>
<td>73.4 Commit</td>
</tr>
<tr>
<td>DB CPU</td>
<td>4853.8</td>
<td>24.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>db file sequential read</td>
<td>15,081</td>
<td>286.5</td>
<td>18.04</td>
<td>1.4 User I/O</td>
</tr>
<tr>
<td>buffer exterminate</td>
<td>12,522</td>
<td>117.6</td>
<td>9.39</td>
<td>.6 Other</td>
</tr>
<tr>
<td>read by other session</td>
<td>432</td>
<td>75.5</td>
<td>174.81</td>
<td>.4 User I/O</td>
</tr>
</tbody>
</table>

**NVDIMMs (DAX)**

Top 10 Foreground Events by Total Wait Time

<table>
<thead>
<tr>
<th>Event</th>
<th>Waits Time (sec)</th>
<th>Avg(ms)</th>
<th>% DB Wait</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>log file sync</td>
<td>4,777,937</td>
<td>2172</td>
<td>0.45</td>
<td>14.6 Commit</td>
</tr>
<tr>
<td>DB CPU</td>
<td>10.7K</td>
<td>72.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>db file sequential read</td>
<td>89,088</td>
<td>1875.5</td>
<td>21.05</td>
<td>12.6 User I/O</td>
</tr>
<tr>
<td>library cache: mutex X</td>
<td>299,418</td>
<td>104.6</td>
<td>0.35</td>
<td>.7 Concurre</td>
</tr>
<tr>
<td>read by other session</td>
<td>1,026</td>
<td>103.1</td>
<td>100.51</td>
<td>.7 User I/O</td>
</tr>
</tbody>
</table>

The bottleneck on the redo logs was removed.
Persistent Memory test environment on ProLiant DL380 Gen9

**Hardware components:**
- HPE ProLiant DL380 Gen9 Server
- 256GB memory
- 16 x HPE 8GB NVDIMM modules (HPE Persistent Memory) for redo logs
- One RAID1 SSD OS disk
- One RAID5 SSD LUN for DB tablespaces, indexes and undo
- 8 x RAID1 SSDs or HDDs for redo logs

**Software components:**
- Red Hat Enterprise Linux 7
- Oracle 12c R1 Enterprise Edition
- Single instance database using Oracle ASM
Oracle DB throughput: HDD vs SSD vs NVDIMM

Oracle DB throughput
SAS vs SSD vs NVDIMM

Relative Throughput

Number of Oracle connections

Hewlett Packard Enterprise
redhat
Oracle redo log latency vs CPU utilization
HDD vs SSD vs NVDIMM
Reduce Oracle licensing costs while achieving higher throughput with NVDIMM as compared to HDD

Oracle DB throughput
SAS with 32 cores vs NVDIMM with 16 cores

NVDIMM with 16 cores performed better than SAS drives with 32 cores, reducing Oracle licenses by 50%
Summary: HPE Persistent Memory for Oracle databases

**Increase performance**

- Up to 2–4X increase in Oracle database throughput using HPE 8 GB NVDIMM for Oracle redo logs\(^1\)
- Much greater CPU utilization with NVDIMM than HDD drives
- Remove redo log bottleneck with fast write time to NVDIMM devices

**Reduce costs**

- Up to 50% reduction in Oracle licensing costs with 8 GB NVDIMM while achieving higher throughput as compared to 15K RPM SAS drives.\(^1\)
- Cost effective compared to SAS drives and SSDs
  - Up to 3X more cost effective using HPE 8 GB NVDIMM than an equivalent number of SSDs\(^1\)

---

\(^1\) Technical white paper, "Improving Oracle Database performance with HPE Persistent Memory on HPE ProLiant DL380 Gen9," August 2016.
Improving EnterpriseDB Performance with HPE Persistent Memory
EnterpriseDB Postgres solution with NVDIMMs

**Hardware**
- HPE ProLiant DL380 Gen9
- 2 x 12-core Intel Xeon E5-2650 v4 processors at 2.20 GHz
- 32 GB memory
- 3 x HPE 8GB NVDIMMs configured as single block device with ext4 filesystem
- DB transaction log, Write-Ahead Logging (WAL) on NVDIMM device
- 2 x 800GB SAS SSDs, RAID1 LUN for WAL for SSD comparison test
- 7 x 800GB SAS SSDs, RAID5 LUN for database tables, ext4 filesystem

**Software**
- Red Hat Enterprise Linux 7.3
- EDB Postgres Advanced Server 9.5
- HammerDB load test tool, 5000 warehouses, 1.1TB database
EDB Postgres transaction improvement with WAL on NVDIMM

### 8 GB WAL

- **5000 warehouses/1 TB database**
- **New Order Per Minute**
  - **SSD**: 227,585
  - **NVDIMM**: 282,058
  - **23.94% Improvement**

### 16 GB WAL

- **5000 warehouses/1 TB database**
- **New Order Per Minute**
  - **SSD**: 254,761
  - **NVDIMM**: 292,282
  - **14.73% Improvement**
Enterprise DB I/O throughput for WAL on NVDIMM vs SSD

8 GiB WAL, 12 GiB shared_buffer

Write throughput in MiB/s

Time in seconds

NVDIMM

SSD
EnterpriseDB CPU utilization with NVDIMM vs SSD
Future plans: SQL Server on Linux and HPE Persistent Memory
Key performance takeaways

- SQL Server 2017 on Red Hat Enterprise Linux surpasses the previous #1 TPC-H@1000GB result achieved with SQL Server 2016
- 6% higher performance
- 5% better price/performance
- The first and only result with Microsoft SQL Server 2017 Enterprise Edition
- Results achieved on similarly configured servers with two Intel® Xeon® E5-2699 v4 processors

#1 performance and price/performance on non-clustered TPC-H@1000GB

HPE, Microsoft, and Red Hat deliver first-ever result with SQL Server 2017 Enterprise Edition

Winning partnerships!

HPE ProLiant DL380 Gen9

SQL Server 2017 Enterprise Edition

Red Hat Enterprise Linux 7

Higher performance

678,492

717,101

$0.64

$0.61

$1.00

$0.50

$0.00

$ USD/QphH

QphH

HPE ProLiant DL380 Gen9 w/SQL Server 2016 on Microsoft Windows

HPE ProLiant DL380 Gen9 w/SQL Server 2017 on Red Hat Enterprise Linux

Better price/performance

HPE ProLiant DL380 Gen9 w/SQL Server 2016 on Microsoft Windows

HPE ProLiant DL380 Gen9 w/SQL Server 2017 on Red Hat Enterprise Linux

Read the performance brief at hpe.com/servers/benchmarks.

© Copyright 2017 Hewlett Packard Enterprise Development LP. Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation. Red Hat, Red Hat Enterprise Linux, and the Shadowman logo are registered trademarks of Red Hat, Inc. Linux is a registered trademark of Linus Torvalds. Intel and Xeon are trademarks of Intel Corporation in the U.S. and other countries. TPC and TPC-H are trademarks of the Transaction Processing Performance Council. TPC-H results show the HPE ProLiant DL380 Gen9 with a result of 717,101 QphH @ 1000GB and $0.61 USD/QphH @ 1000GB with system availability as of 10-19-2017 (results published 04-19-2017; see tpc.org/3327); the HPE ProLiant DL380 Gen9 with a result of 678,492 QphH @1000GB and $0.64 USD/QphH @ 1000GB with system availability as of 07-31-2016 (results published 03-24-2016; see tpc.org/3320). The TPC believes that comparisons of TPC-H results published with different scale factors are misleading and discourages such comparisons. Please see tpc.org for up-to-date information. Competitive claims valid as of 04-19-2017.
SQL Server 2016 Tail of Log

Server configuration:
- 1x HPE ProLiant DL380 Gen9 (both sockets populated)
- 1x NVDIMM-N (8 GB) – for the tail of the log
- 2x SATA SSD (400 GB) – as the store for database files
- 1x NVMe SSD (400 GB) – as the store for both logs
- 128 GB memory

Software:
- Windows Server 2016 TP5
- SQL Server 2016 RC3
  - SQL tables are stored on 2x SATA SSDs that are striped (Simple Space)
  - SQL Tail of the Log enabled
  - Table size configured to match data and log storage capacities
  - Threads: 1 per Windows logical processor
  - SQL queries: Create, Insert, Update
  - SQL PerfCollectors: None
  - Batch size: 1
  - Row size: 32B

Executed tests and results:
- 05/19/2016: 2x with a HPE write-intensive NVMe SSD
- 05/06/2016: 3x with a mixed (vs. write-intensive) type NVMe SSD
- June 2016: 4x with a SAS SSD
HPE Persistent Memory Resources

Website
- Persistent Memory web page
- Persistent Memory software
- Persistent Memory wiki on kernel.org

Videos and Blogs
- Persistent Memory 3D Product Demo
- Persistent Memory Overview Video
- NVDIMM-N as Byte-Addressable Storage in Windows Server 2016
- NVDIMM-N as Block Storage in Windows Server 2016
- Persistent Memory blogs

Technical Papers
- Persistent Memory technical white paper
- Persistent Memory on SQL Server 2016
- Persistent Memory on Windows Server 2012 R2
- Reducing Oracle licensing and improving performance
- Accelerate EDB Postgres Advanced Server
Thank you