Under the Hood of an Exadata Transaction

How do we harness the power of Persistent Memory?

Jia Shi
Vice President
Exadata Development, Oracle
jia.shi@oracle.com
Meet an Exadata X8M

X8M-2 socket Xeon
- 48 cores per server
- 384 GB - 1.5 TB DRAM

X8M-8 socket Xeon
- 192 cores per server
- 3-6 TB DRAM

High Capacity
- 168 TB HDD
- 25.6 TB PCI NVMe Flash

Extreme Flash
- 51.2 TB PCI NVMe Flash

100 Gb/s RoCE RDMA over Converged Ethernet

1.5 TB Persistent Memory
## Exadata X8M: Scale-out with Elastic Expansion

<table>
<thead>
<tr>
<th>Eighth Rack</th>
<th>Quarter Rack</th>
<th>Elastic</th>
<th>Multi-Rack</th>
<th>X8M-8</th>
<th>Elastic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Database Servers</td>
<td>Storage Servers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2021, Oracle and/or its affiliates
Exadata
Extreme Performance and Availability at Lowest Cost

Ideal Database Hardware –
Scale-out, database optimized compute, networking, and storage

Database Aware System Software –
Unique algorithms vastly improve OLTP, Analytics, Consolidation

Automated Management –
Fully automated and optimized platform from end to end

Identical Across On-Prem, Cloud, Cloud@Customer
Thousands of Ultra-Critical Deployments Since 2008

87% of Fortune Global 100 Run Exadata

BEST FOR ALL WORKLOADS

Petabyte Warehouses

Ultra-critical Oltp – Financial Trading, Etc

Complex Business Applications - Sap, Oracle, Siebel

Massive Db Consolidation
Let’s go under the hood of OLTP - OnLine Transactional Processing
Meet a Super Critical OLTP Application

What does such an app do?

Ben wants to deposit $1000 to his bank account.
Alice wants to withdraw $500 from her bank account.
Bob wants to transfer $2000 from checking into savings.

....
Meet Ben’s Transaction
What constitutes a database transaction?

Ben wants to deposit $1000 to his bank account.
OLTP Challenge #1 -
What is the **IO cliff** for random data reads?
Challenge #1 – Random Data Read

1. Identifies the row of Ben’s account – where is the block? Miss in the buffer cache!
2. Issues the data read to storage

Buffer Cache (DRAM)

<table>
<thead>
<tr>
<th>User</th>
<th>Account Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Ben</td>
<td>$2000</td>
</tr>
</tbody>
</table>

Flash Cache

1. Finds the data block cached in Flash Cache
2. Issues local read to Flash

Database Server

Storage Server

3. Sends data block to DB

How long did the Super Critical OLTP App wait for this random block read?
How long does 8K random read take?

8K Local Read Latency @ < 100 usec

Network Round Trip
- User->Kernel & Kernel-> User context switches
- Network IO interrupt processing
- IO software stack

How long did the Super Critical OLTP App wait for this random block read?

~200 usec

~100 usec
OLTP Challenge #1 -
How to conquer the random read IO Cliff?
Do you recognize the dynamic duo?
Persistent Memory (PMEM)

- Persistent memory is a new silicon technology
  - Capacity, performance, and price are between DRAM and flash
- Intel® Optane™ Persistent Memory:
  - Reads at memory speed – much faster than flash
  - Writes survive power failure unlike DRAM
- Requires sophisticated algorithms to maintain integrity of data on PMEM during failures
  - Call special instructions to flush data from CPU cache to PMEM
  - Complete or backout sequence of writes interrupted by a crash
Do you recognize the dynamic duo?

PMEM

RDMA
Remote Direct Memory Access (RDMA)

Database Server

Memory Region

CPU

RDMA Write

Storage Server

Memory Region

CPU

RDMA Read
**Introducing RDMA over Converged Ethernet**

RDMA over Converged Ethernet is a protocol that runs InfiniBand RDMA software on top of Ethernet

- Same software at upper levels of network protocol stack
- InfiniBand packets sent as ethernet UDP packets at low level

Exadata RoCE provides **RDMA speed** and reliability on Ethernet fabric

- 100Gb/s link throughput
- Zero packet loss messaging
- Prioritization of critical database messages

Defined by an open consortium IBTA, developed in open-source Linux

- Supported by major network card and switch vendors
How do they conquer the IO Cliff?

- Parse the SQL
- Update SQL
- Traverse a b-tree index via primary key lookup
- Identifies the row of Ben’s account – where is the block?

Deposit $1000

Database Server

Storage Server

PMEM

RDMA

Copyright © 2021, Oracle and/or its affiliates
Drop in solution: Flash -> PMEM?

What happens to the IO Cliff?

~100 usec

• User->Kernel & Kernel-> User context switches
• Network IO interrupt processing
• IO software stack

Network Round Trip

~100 usec
How about a **radical** approach: RDMA to PMEM?

Database Server

Storage Server

PMEM Cache

**What happens to the IO Cliff?**

<19 usec

10x Faster Random Read!

---

**RDMA over 100Gb/s RoCE**

- User->Kernel & Kernel-> User context switches
- Network IO interrupt processing
- IO software stack

**PMEM**

- User->Kernel & Kernel-> User context switches
- Network IO interrupt processing
- IO software stack

~100 usec
**RDMA Read from PMEM Cache in Storage Server**

**Miss** in Buffer Cache: Need to fetch the data from storage

1. DB issues RDMA Read from PMEM Cache

8K Data Block returned to DB from PMEM - No software involved

<table>
<thead>
<tr>
<th>User</th>
<th>Account Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Ben</td>
<td>$2000</td>
</tr>
</tbody>
</table>

Copyright © 2021, Oracle and/or its affiliates
Ben wants to deposit $1000 to his bank account.

- Faster – 10x lower latency
- Cheaper – less CPU than context switch
Exadata X8M With Persistent Memory
World’s First and Only Shared Persistent Memory Optimized for Database

- Shared capacity across Databases
- Smart Capacity management for maximum performance
  - Primaries on persistent memory, secondaries on flash
  - Secondaries moved into persistent memory when they become active primaries
- Management server on storage server automatically manages persistent memory
  - Creates regions, namespaces, DAX devices etc as needed
  - Persistent memory replacement follows standard hardware replacement process
- End to End Security
  - Persistent memory is not visible to the OS on the database server
  - Protected end to end while serving RDMA
Meet Ben’s Transaction – time to commit?

Putting the D(urability) into Database Transaction ACID Properties

Ben wants to deposit $1000 to his bank account.

Ben’s Transaction Process:
1. Parse the SQL
2. Update index via primary key lookup
3. Update the row of Ben’s account
4. Add $1000 to Ben’s balance
5. Commit transaction

Database Server

Storage Server

User | Account Balance
--- | ---
... | ...
Ben | $2000 -> $3000

Update Ben’s row to add $1000

Falls off IO Cliff

How do we commit a transaction?
OLTP Challenge #2 -
Can lightning strike the same place twice?
What is the IO cliff for redo log writes?
Challenge #2 – Redo Log Writes

D = Durability
Write redo and commit records to durable storage media

1. Send redo writes to storage
2. Issues redo writes to flash and disk controller DRAM cache simultaneously
3. Upon 1st write completion, sends ack to DB
4. OLTP App notifies user of transaction commit

Database Server

Storage Server

OLTP APP

Flash Log

Disk Controller
Persistent DRAM Cache

1. Update Balance for Ben +$1000
2. Commit Transaction

How long did the Super Critical OLTP App wait for the log write?

~200 usec
Can PMEM + RDMA come to the rescue again?

Lightning strikes the same place twice!

Ben wants to deposit $1000 to his bank account.

- Parse the SQL
- Traverse a b-tree index via primary key lookup
- Identifies the row of Ben’s account – where is the block?
- RDMA to PMEM <19 usec
- Update Ben’s row to add $1000

<table>
<thead>
<tr>
<th>User</th>
<th>Account Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Ben</td>
<td>$2000 -&gt; $3000</td>
</tr>
</tbody>
</table>

Database Server

Storage Server

1. Update Balance for Ben +$1000
2. Commit Transaction

How do we not fall off the IO cliff?
Redo Log RDMA write to PMEM in Storage Server

1. Update Balance for Ben +$1000
2. Commit Transaction
3. OLTP App notifies user of transaction commit

D = Durability
Write redo and commit records to durable storage media

OLTP APP

Database Server

1. Log write via RDMA to PMEM
2. NIC to NIC Ack - No software involved

Storage Server

PMEM Log

PMEM Log Buffer

PMEM Log Buffer

PMEM Log Buffer

4. Destage redo to backing store
**Redo Log RDMA write to PMEM in Storage Server**

1. **Update Balance for Ben +$1000**
2. **Commit Transaction**
3. OLTP App notifies user of transaction commit

D = Durability
Write redo and commit records to **durable** storage media

Is my redo safe?
Which Begs the Question:
Are all writes to PMEM automatically persistent across power failures?

Intel Data Direct IO (DDIO) enabled by default
- Performs Write Allocate/Update into L3 Cache
- Places data outside Asynchronous DRAM Refresh (ADR) Safe Domain
  - Not Durable upon Power Fail

How to ensure redo log writes are persistent in PMEM?

Turn off DDIO for Network Card PCI Slot in BIOS
- Global Knob per PCI Device:
  - Affects DMA to both DRAM and PMEM
- Be aware of potential impacts to other network receive performance due to LLC misses
What happens if the network is congested?

Smart network prioritization of redo log writes!

Network prioritization for latency-sensitive DB algorithms

- Ensures that messages requiring low latency are not slowed by high throughput messages
  - Low latency: transaction commit, redo log writes
  - Medium priority: OLTP data reads
  - High throughput: backup, reporting, batch workloads

RoCE Class of Service (CoS)

- Allows packets to be sent on multiple classes of service, each with separate network buffers for independence

Exadata uniquely chooses the best Class of Service for each database message
Remember Ben’s Transaction?

Ben wants to deposit $1000 to his bank account.

- Parse the Update SQL
- Traverse a b-tree index via primary key lookup
- Identify the row of Ben’s account – where is the block?
- RDMA Read from PMEM Cache < 19 usec

What happens to the IO Cliff?

8x Faster Log Writes!

1. Update Balance for Ben +$1000
2. Commit Transaction

Update Ben’s row to add $1000

RDMA Write to PMEM Log
10’s usec

Write the redo log and commit the transaction

<table>
<thead>
<tr>
<th>User</th>
<th>Account Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Ben</td>
<td>$2000 -&gt; $3000</td>
</tr>
</tbody>
</table>
Are we done?
Yes and No

Ben wants to deposit $1000 to his bank account – YES!

1. Update Balance for Ben +$1000
2. Commit Transaction

Not yet – Database needs to checkpoint updated data to storage!
OLTP Challenge #3 - What is so difficult about writing data back to PMEM Cache in Storage Server?
What is splinch?
No torn blocks!

PMEM has 8-byte Atomic Write Guarantee
• Database blocks are much larger – 8K being the typical size
• No power failure should result in torn blocks!

Is RDMA to PMEM a good choice here? NO!!!
How do we prevent splinch?

3 D’s - Destination, Determination, and Deliberation

Destination:
DB sends the 8K data block to storage via regular messaging

Determination:
Storage software recognizes that the data needs to be updated on PMEM Cache

Deliberation:
Storage software guarantees atomic database block writes to PMEM via staging writes

What happens during a power fail?
• Staging buffer recovery!

How do we prevent splinch?

3 D’s - Destination, Determination, and Deliberation

Destination:
DB sends the 8K data block to storage via regular messaging

Determination:
Storage software recognizes that the data needs to be updated on PMEM Cache

Deliberation:
Storage software guarantees atomic database block writes to PMEM via staging writes

What happens during a power fail?
• Staging buffer recovery!

Copyright © 2021, Oracle and/or its affiliates
How do we prevent splinch?

3 D’s - Destination, Determination, and Deliberation

**Destination:**
DB sends the 8K data block to storage via regular messaging

**Determination:**
Storage software recognizes that the data needs to be updated on PMEM Cache

**Deliberation:**
Storage software guarantees atomic database block writes to PMEM via staging writes

What happens during a power fail?
- Staging buffer recovery!

**User | Account Balance**
--- | ---
... | ...
Ben | $2000 -> $3000

**PMEM Cache Line**

Re-apply the 8K staging buffer to PMEM Cache
How do we ensure persistence across power failure?

Stores in CPU Cache are NOT Persistent

- Use Non-Temporal Stores to Bypass CPU Cache
  - `_mm512_stream_si512` or `_mm_stream_si64` + `sfence`
- Use CLWB to Flush into ADR Domain
  - `_mm_clwb_` + `sfence`

We use **NT stores** for PMEM Cache:
- Faster for sequential 8K data writes
- Avoid polluting CPU Cache
How do we achieve better PMEM write scaling?

- Previously, write scales poorly with concurrent writes
  - Especially with writes from remote core

- IO Directory Cache contention
  - Specific IODC setting reduces contention
  - Still room for future improvements

- Each thread slows down writes from another thread

- Best performance with fewer threads doing IOs

![Graph showing effective bandwidth vs. number of threads doing random 8 kiB IOs.](image)
Take Away – Do not fall off the IO Cliff!

Forget about Ben’s silly transaction!
What is a truly Super Critical OLTP Application in the Real World?

- Fraud tracking
- Real-time analytics upon a click on the screen
- …
OLTP on Exadata
How do we harness the power of PMEM?

How to have a cake and eat it too?
- >99% of PMEM used for PMEM Cache – 1.5TB per server
- <1% of PMEM used for PMEM Log – 10G per server

New Leading-Edge RDMA fabric
- Based on 100 Gb/sec Converged Ethernet

RDMA to PMEM in Storage
- 10X better transaction processing IO latency @ <19 usec
- 8X faster log writes for faster commit processing
- 16 Million read IOPS on a full rack of ultimate OLTP database machine

PMEM automatically tiered and shared across Databases
Our mission is to help people see data in new ways, discover insights, unlock endless possibilities.