ORACLE®

Supporting SPDK in Oracle RDBMS

Presenters: Bang Nguyen, Akshay Shah
Contributors: Zahra Khatami, Avneesh Pant, Sumanta Chatterjee

Oracle Database Virtual OS
May 2018

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE

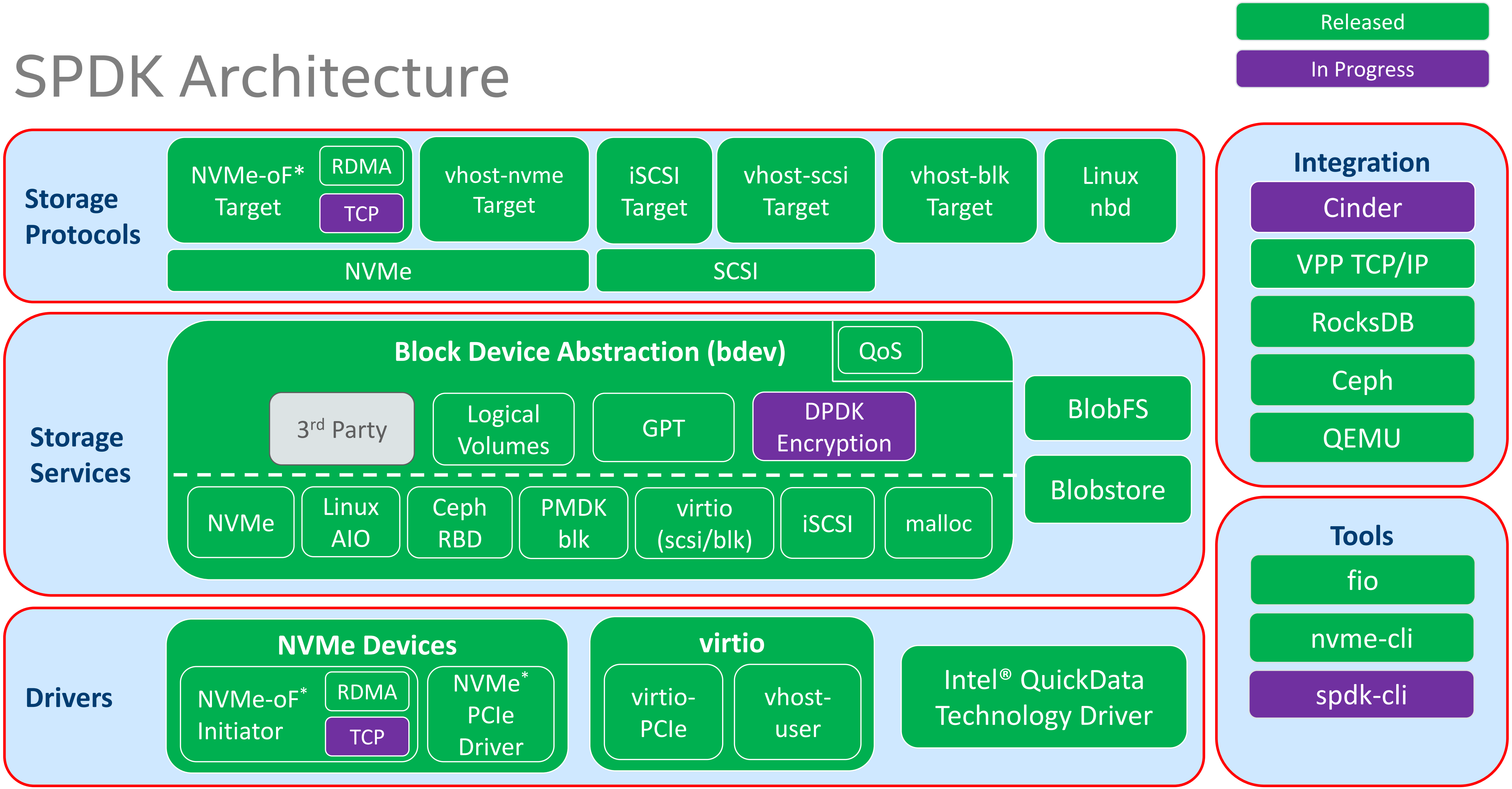
Program Agenda

1. SPDK
2. Challenges
3. Oracle Dispatcher
4. Memory Model
5. Future Work

ORACLE

ORACLE

SPDK Architecture



ORACLE

SPDK benefits

- Enables scalable storage applications.
- High-performing millions of I/Os per second.
- Direct access to local NVMe SSDs as well as access to remote storage targets using NVMeoF.
- Highly concurrent and asynchronous runtime with no locking in the I/O path.
- Directly polling the hardware queues for completions.

Significantly improves I/O performance for latency sensitive applications processing lots of concurrent disk I/O requests

Challenges

NVMe SSDs contain a limited number of hardware IO queues.

- Databases usually comprise 10s-1000s of processes.
- Each client process can allocate one or more IO queues for PCIe I/O to local NVMe devices.
- This can cause rapid exhaustion of available hardware queues.

Oracle's existing memory management infrastructure conflicts with DPDK.

- Allocating private/shared memory from same region.

ORACLE

1 SPDK

2 Challenges

3 Oracle Dispatcher

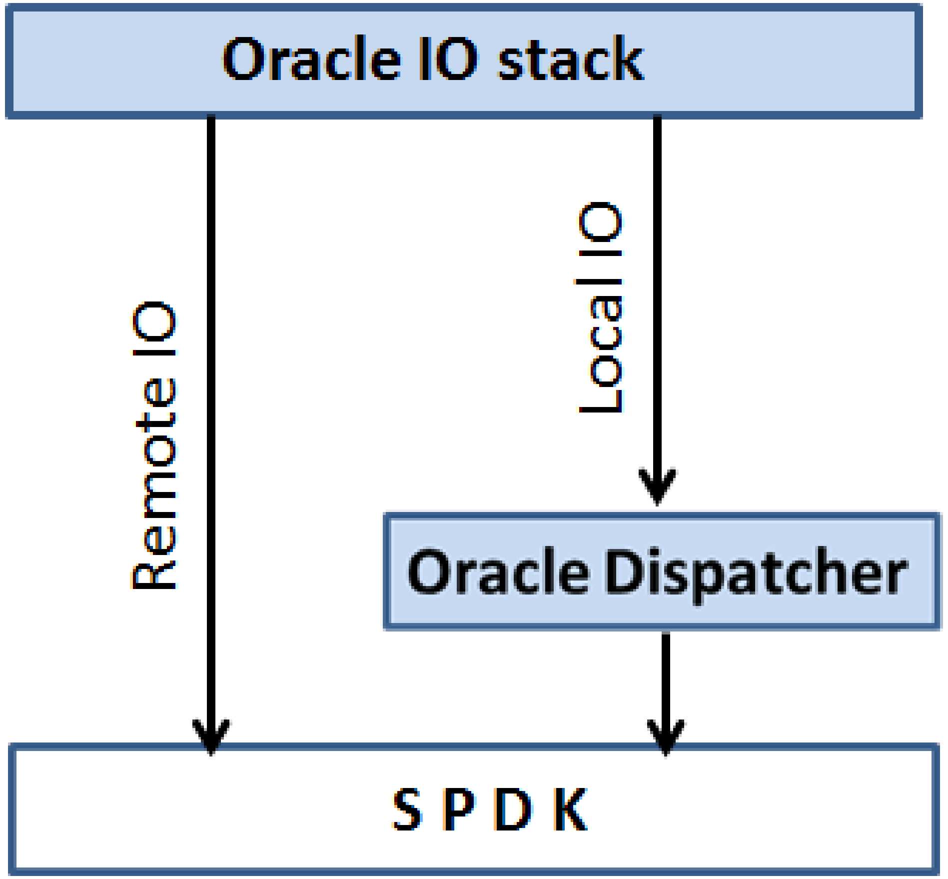
4 Memory Model

5 Future Work

ORACLE

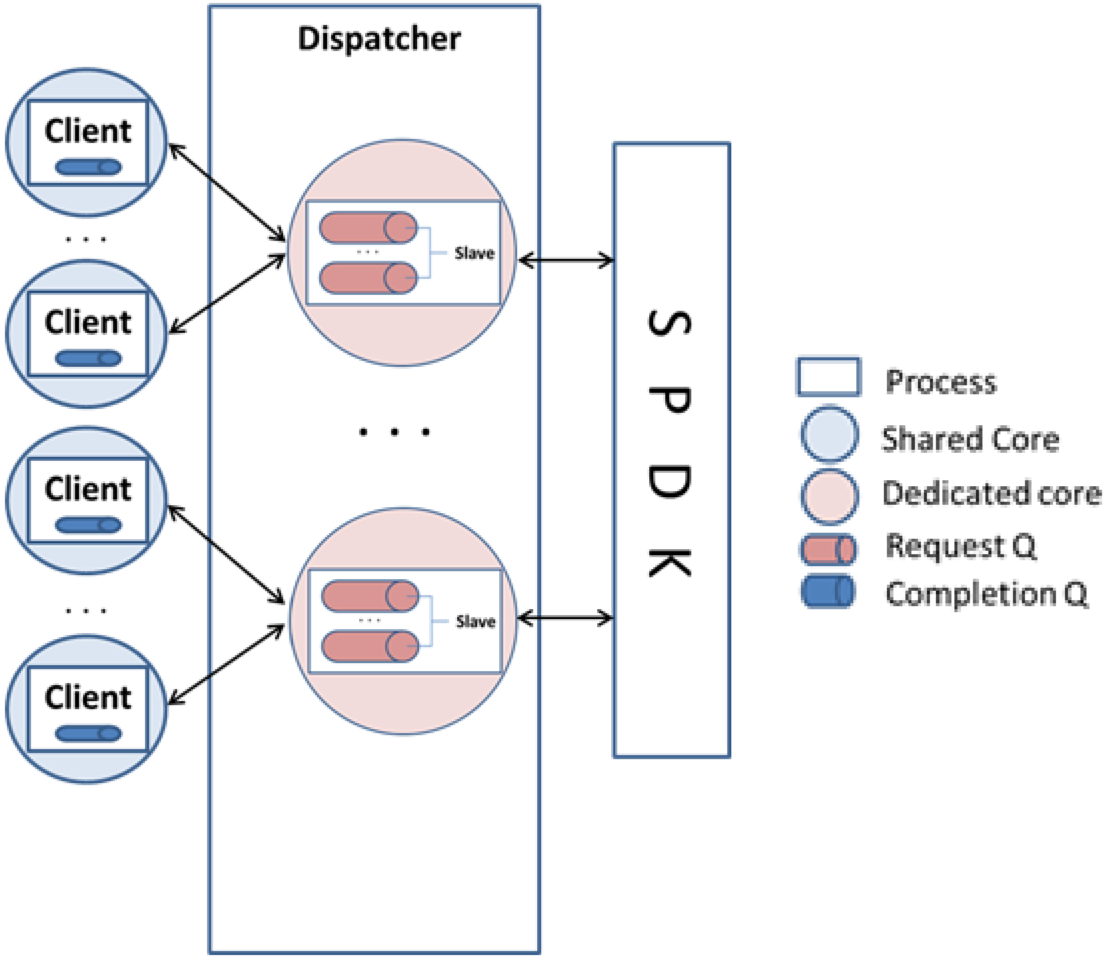
Oracle/ SPDK I/O stack

- 2 different I/O code paths into SPDK.
- Remote I/O: Submit/ Poll I/O directly to/ from SPDK.
- Local I/O: Submit/ Poll I/O directly to/ from Oracle dispatcher.
- Oracle dispatcher submits/ polls I/O to/ from SPDK.



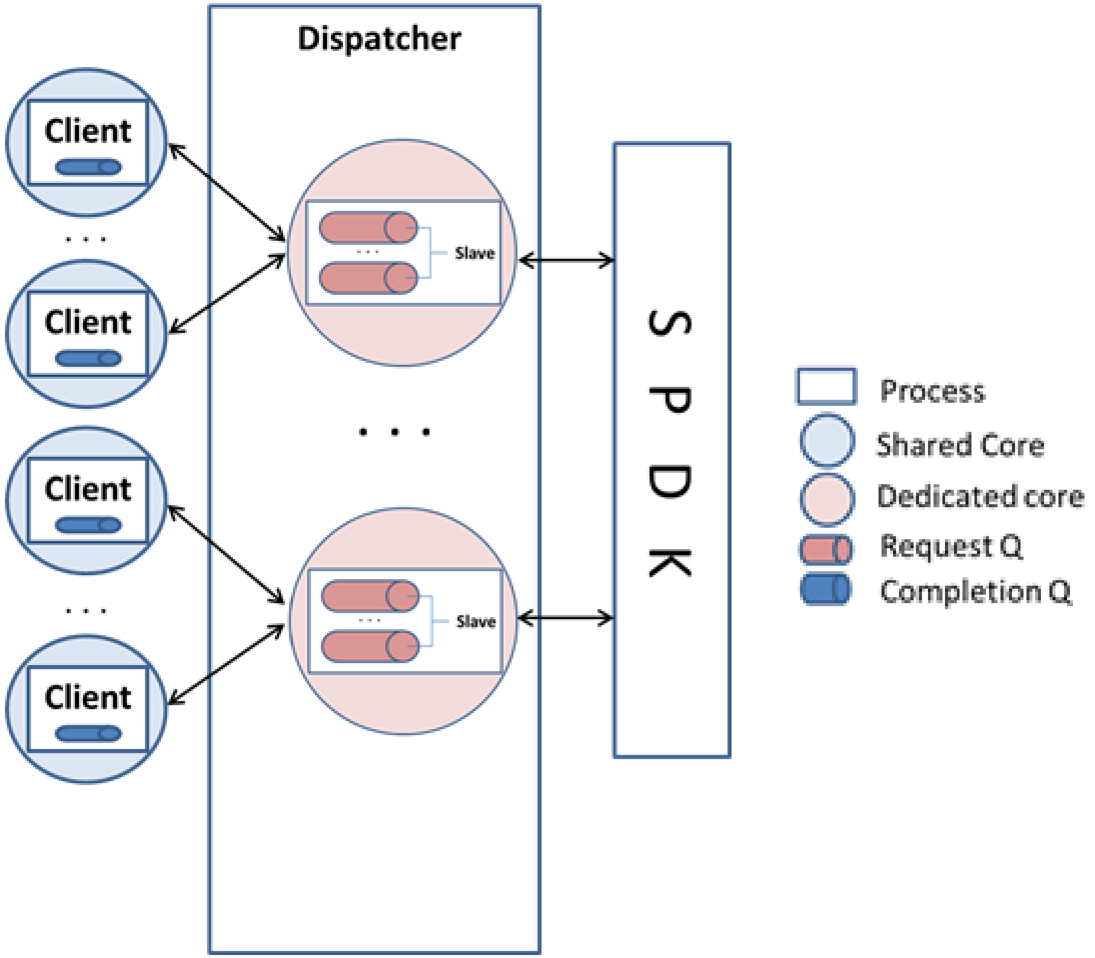
Oracle Dispatcher

- Local IO proxy that runs one or more slaves.
- Each Slave runs in its own core.
- Each Slave has one or more request queues.
- Request queue implemented as a lock-free ring.
- Slave processes IO requests from clients.
- Queues IO completions to client completion queues.
- Runs as secondary process to the target.



Oracle Client

- Bound to specific Dispatcher Slave/Request queue
- Clients bound to the same Request queue must run on separate cores.
- Clients bound to different Request queues can run on the same core.
- Each client has its own completion queue.
- Submits I/O requests to the Slave request queue.
- Polls I/O completions from its completion queue.



ORACLE

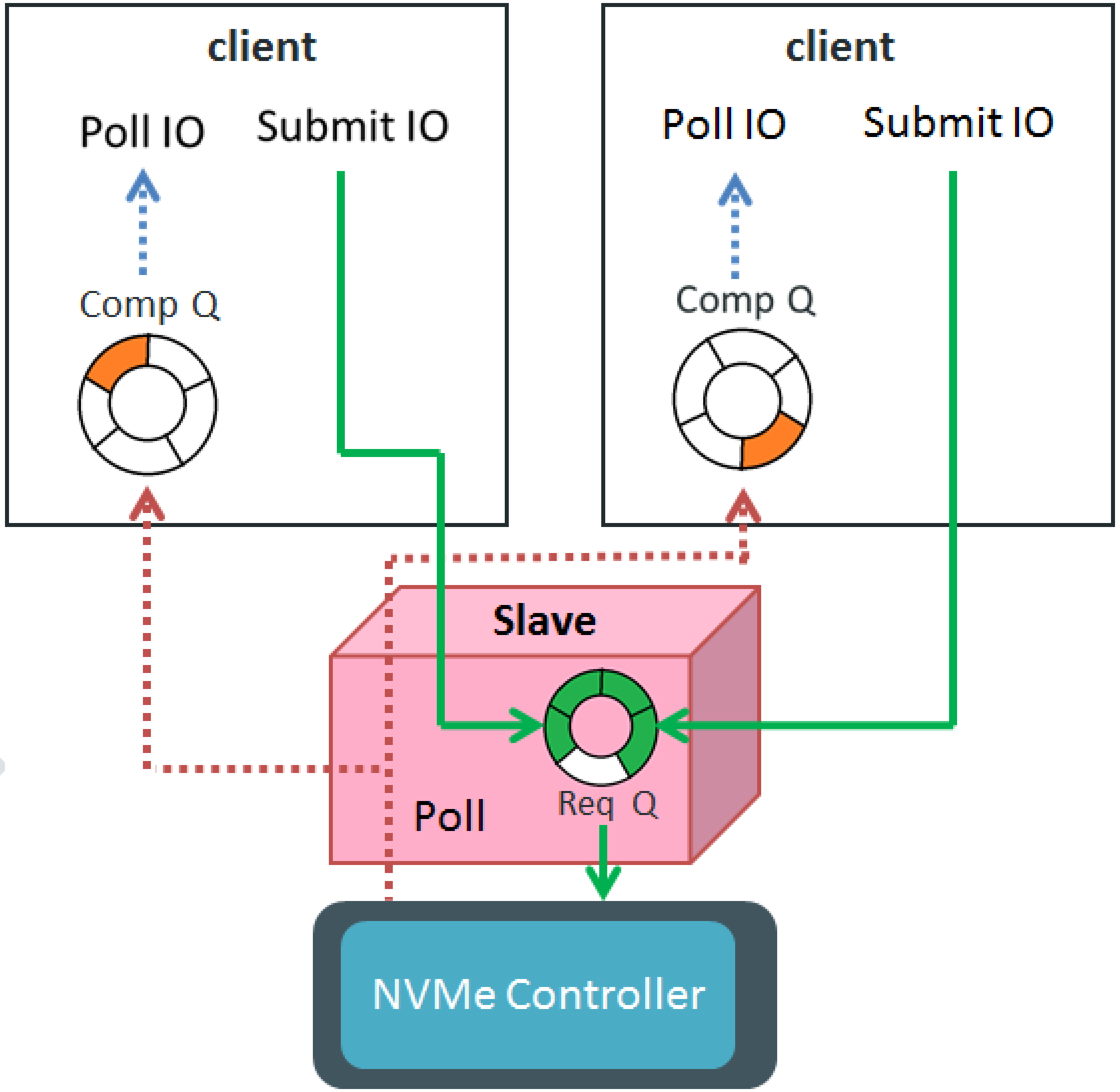
Lock-free Queues

Request Queue

- Multiple producers Single consumer queue.
- IO clients submit requests and Dispatcher Slave processes them.

Completion Queue

- Single producer Single consumer queue.
- Dispatcher Slave queue IO completions and Client reaps them.



ORACLE

DPDK Memory Model

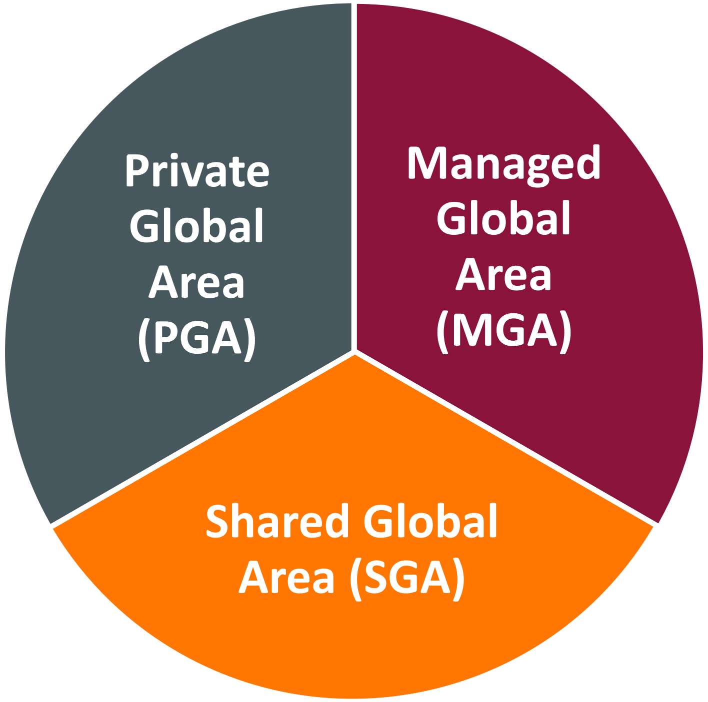
Decouple the storage libraries from DPDK

- DPDK toolkit is used for memory management:

1. Attaches to huge pages upfront.
2. Allocates private/shared memory from same area.
3. Shares memory map with both primary/secondary processes.

ORACLE

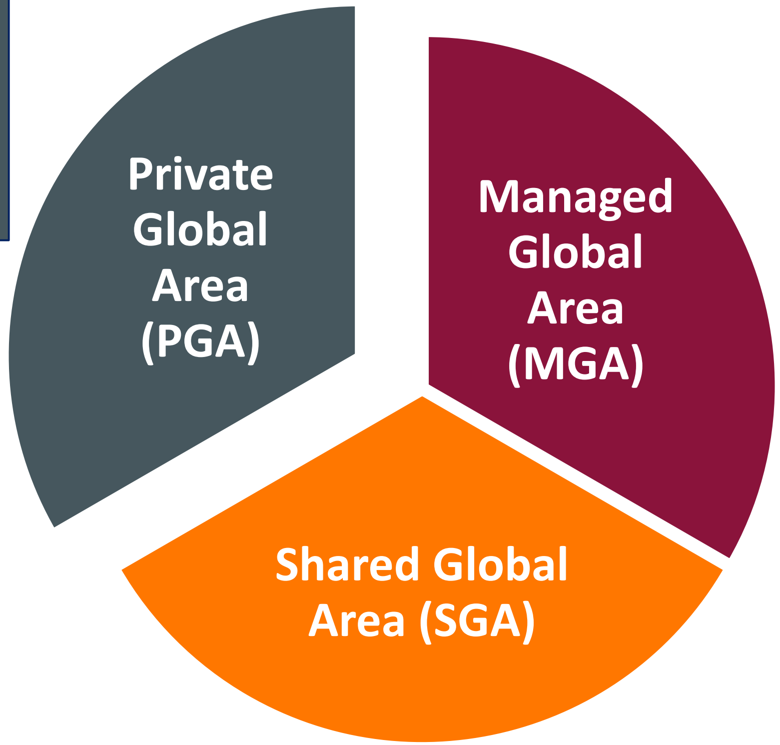
Oracle Memory Model



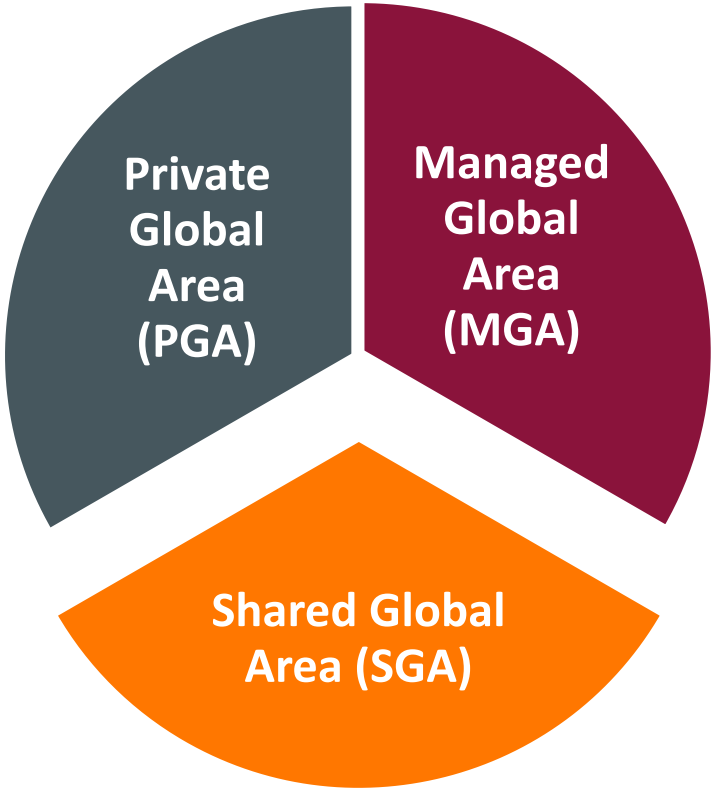
Oracle Memory Model

PGA: private and not-shared

Memory needed for the operation of one process.



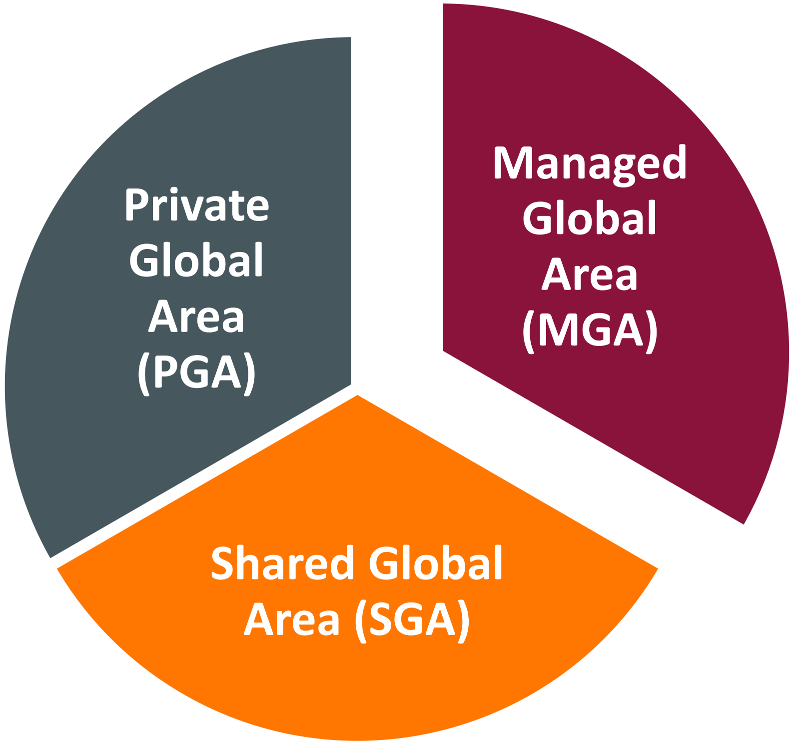
Oracle Memory Model



SGA: large physically shared memory.

Addressable by all processes within an instance.

Oracle Memory Model



MGA: Can be shared.

Uniquely identified by its name.

New segments can be added or existing segments be deleted.

Environment Abstraction Layer (EAL)

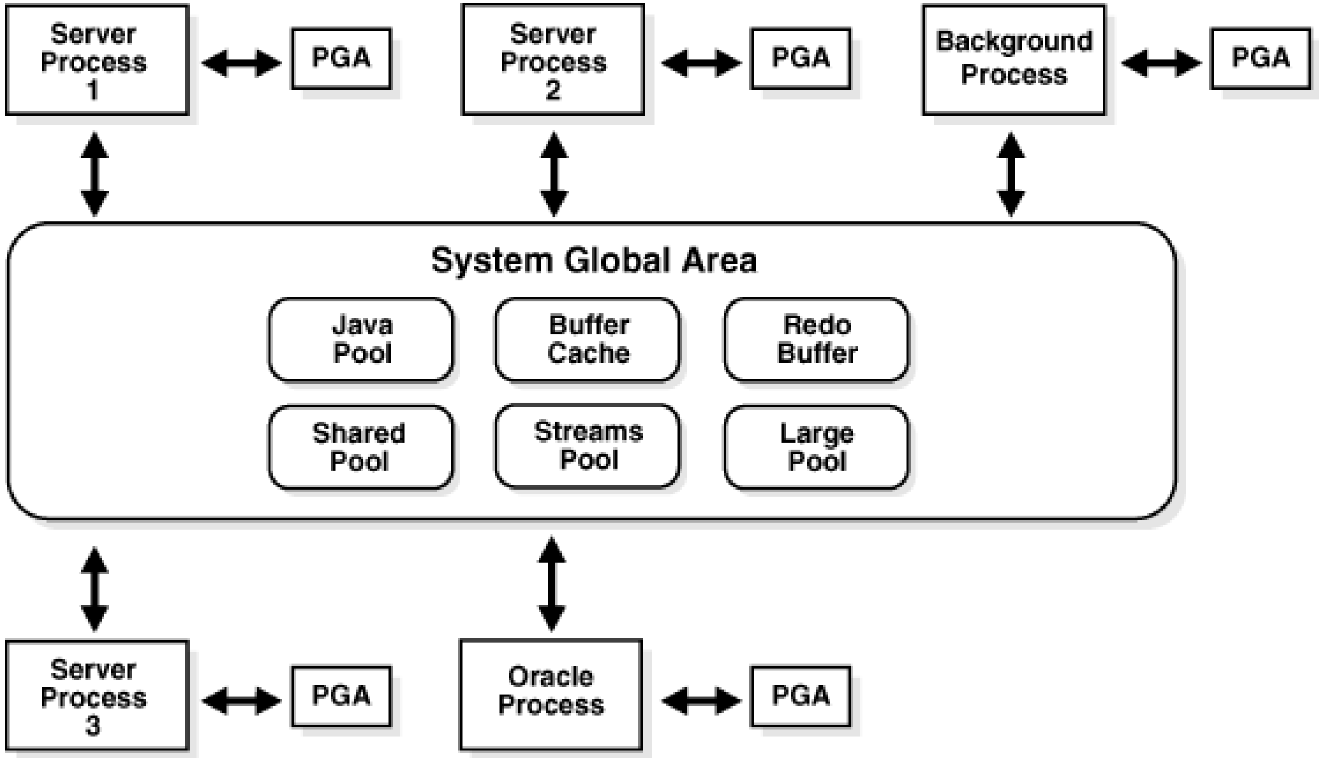
- Provides access to low-level resources such as hardware and memory space.
- Hides environment specifics from applications and libraries.
- Provides core assignment/ affinity, memory management, PCI enumeration, address translation, etc.
- DPDK is the default SPDK RTE.
- ORAENV is DPDK equivalent for Oracle database.

ORACLE

ORAENV

- Dynamic allocation of shared memory from SGA and MGA pools and private memory from PGA pools.
- Similar features to DPDK RTE environment using Oracle runtime services.
- RDMA data transfer optimizations for both local and remote IO.

ORAENV: Dynamic Memory Allocation



- Space is organized into heaps.
- Heap can be allocated in address space that is private to a particular process or shared by many processes.
- When client requests memory, chunk is allocated from a particular heap.
- A heap is composed of a set of contiguous chunks.
- Returns set of extents contained in the heap.

ORAENV: RDMA Data Transfer

Shared Protection Domain

Problem: Registering entire memory region in each process is not scalable

- Used for key lookup services.
- Access to shared data regions.
- Read and write access to the process.
- Allows a single mapping to be registered.
- Re-using memory mapping improves cache performance.
- PD is valid as long as one user process is attached.

ORACLE

ORAENV: Shared Protection Domain

local/remote
zero_copy
data transfer

Process A:

Allocate a shared protection domain and register memory using the PD in process A.

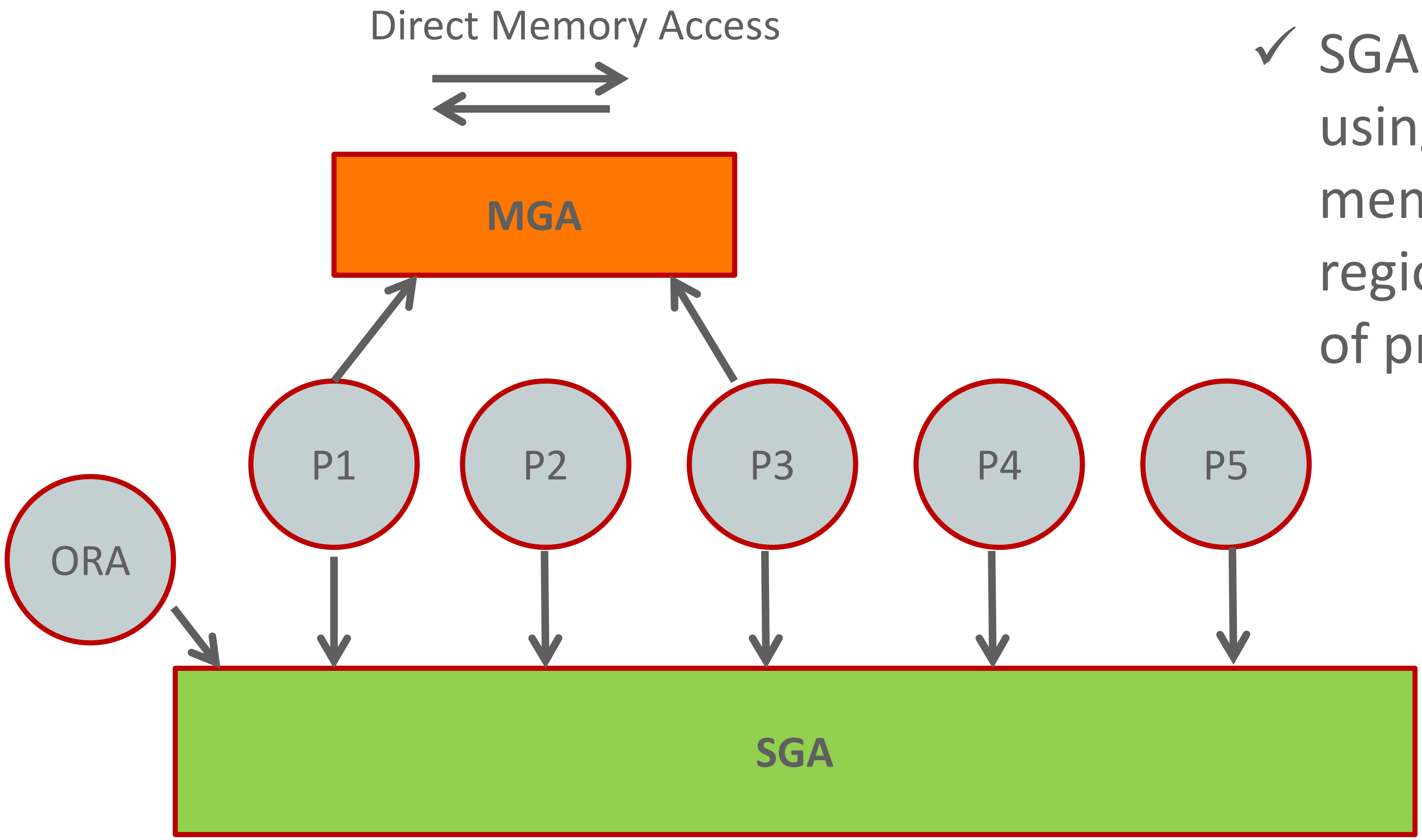
All other processes:

Map/attach to the allocated memory using the same shared PD.

Shared memory registration enables zero copy data transfer.

ORACLE

ORAENV



- ✓ SGA and MGA are both registered using shared PD so only a single memory registration for the region can be used across 1000s of processes.

I/O Resource Management with SPDK

- Ability to rate limit IOPS and throughput for database workloads.
- Database workloads can be standalone databases or pluggable databases in a multi-tenant container.
- Prioritize high priority I/Os such as redo log writes over other I/Os.
- Prevent low priority tasks such as database backups from impacting other workloads.

ORACLE

Security Management for NVMeoF

- All devices in an NVM subsystem are accessible from all hosts and databases.
- Need ability to isolate access to namespaces for different database tenants.
- Implement per-connection authentication and access control checks to restrict visibility and access to namespaces.
- Add IPSec support for encrypted data transfer over the network.

NVMeoF Transport

- Current implementation is based on NVMe over RDMA.
- Implement support for TCP as it is more commonly available in data centers.
- Awaiting TCP/IP standardization from NVMe technical working group.
- Awaiting TCP/IP support in SPDK.

Conclusion

- SPDK enables scalable I/O performance for Oracle database.
- Oracle dispatcher reduces I/O latency for local NVMe devices.
- ORAENV integrates Oracle's existing memory model with SPDK.
- Provides support for memory management, registration and address translation.

ORACLE

THANK YOU!