# SPDK Best Practice: Lessons Learned from Five Years Storage Evolution in Alibaba Cloud

Zhou, Yanbo

Alibaba Group

# What you will learn from this presentation?

Have you been confused about the following problems?

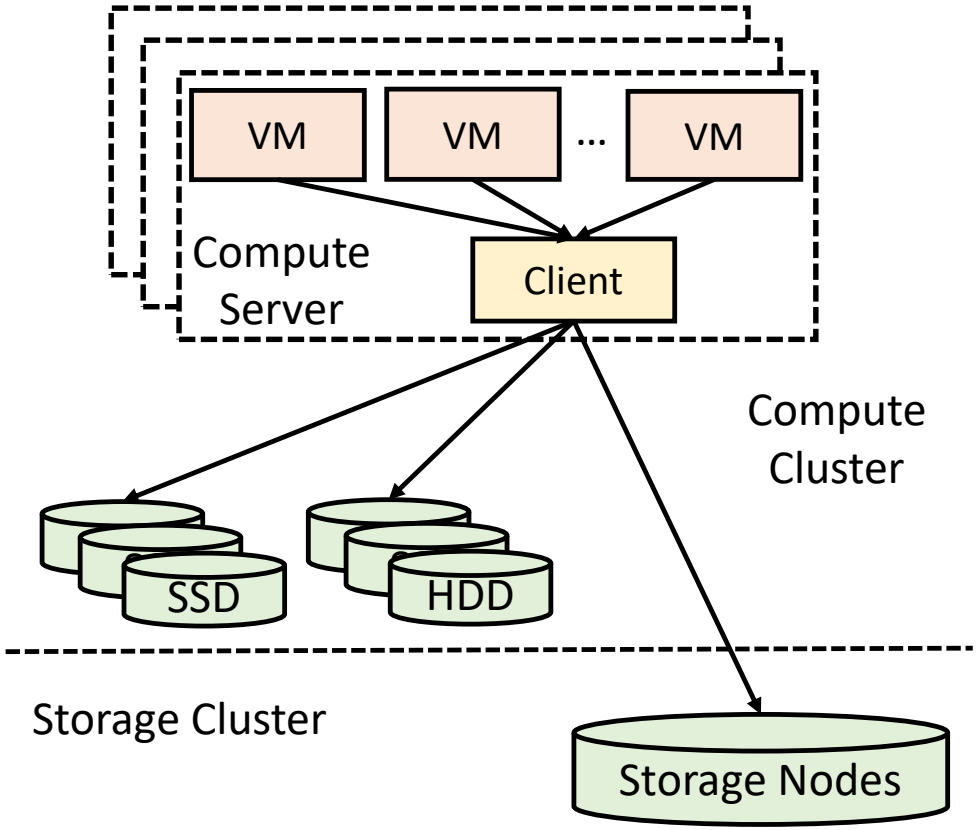
1. How to start and integrate SPDK into your system?
2. How to communicate with SPDK-managed storage devices?
3. How to resolve fast recovery and consistency challenges?
4. How to re-innovate new system based on SPDK?

Let’s go with me!

# Who we are?

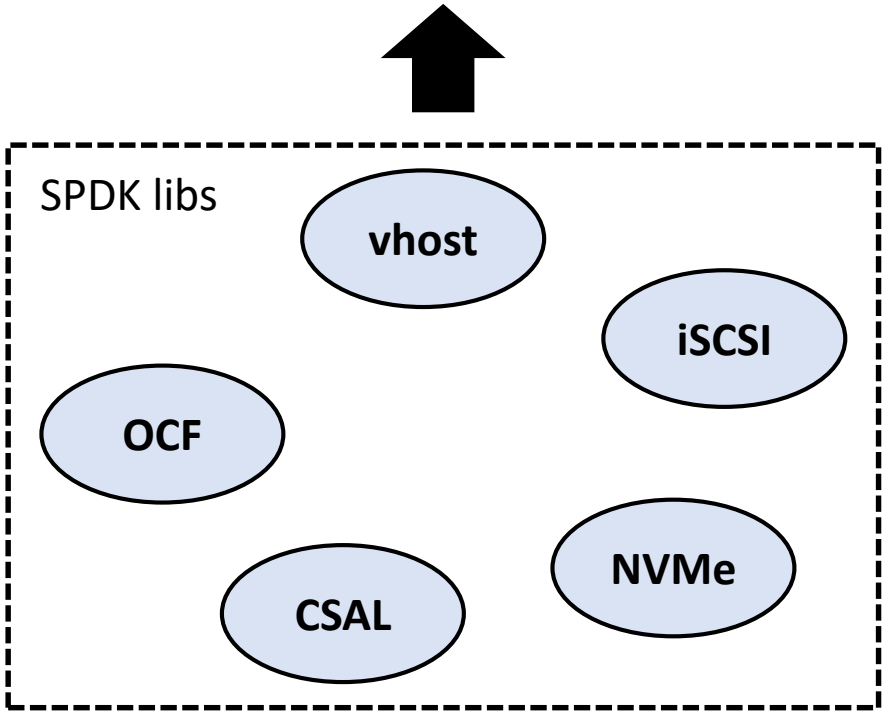
Elastic Block Storage (EBS) Team

- Provide block-level volumes for ECS
- EBS volume and local storage volume




SPDK in EBS:

- Local storage volume: I2/I4, D2/D3, etc.
- EBS volume: Ultra disk, enhanced SSD (ESSD)



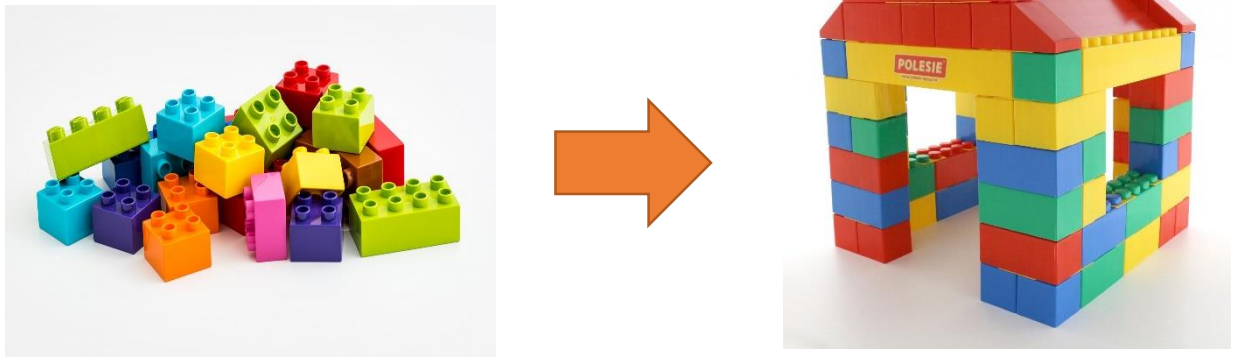
# Start and integrate SPDK into your system

SPDK provides:

- ✓ Basic libraries for storage software: NVMe, vhost, FTL, cache, etc.
- ✓ Application framework with polling-mode event handling.

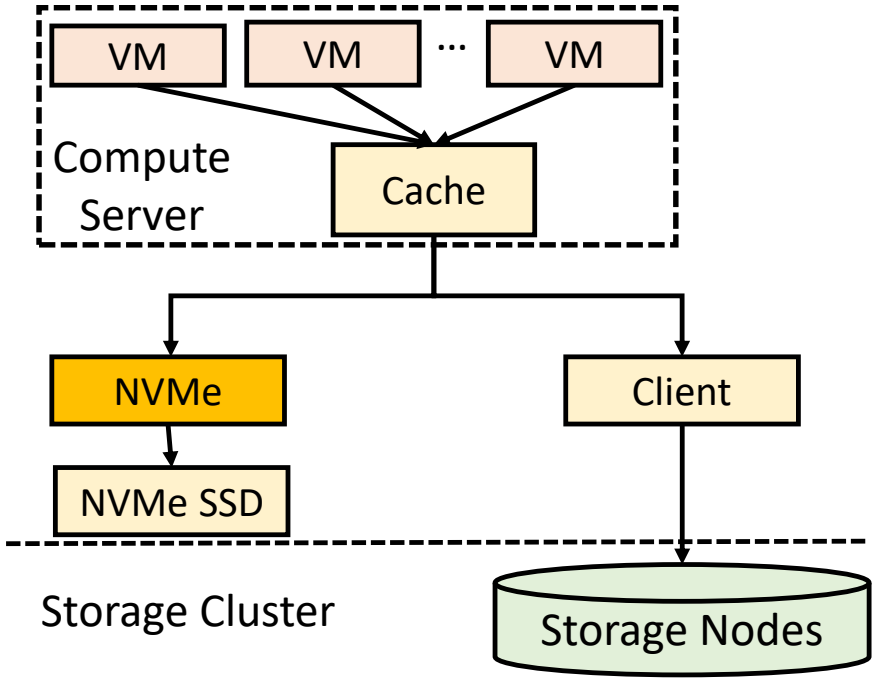
Start with basic libraries:

- Include SPDK headers you needed
- Put the *function call* in right place
- Compile your application with SPDK libraries




*Note: you should have your own thread and memory management*

*Example:*




```
gcc -o app app.c -I /path/to/headers
-L /path/to/libs -Wl,--whole-archive -lspdk_nvme
```

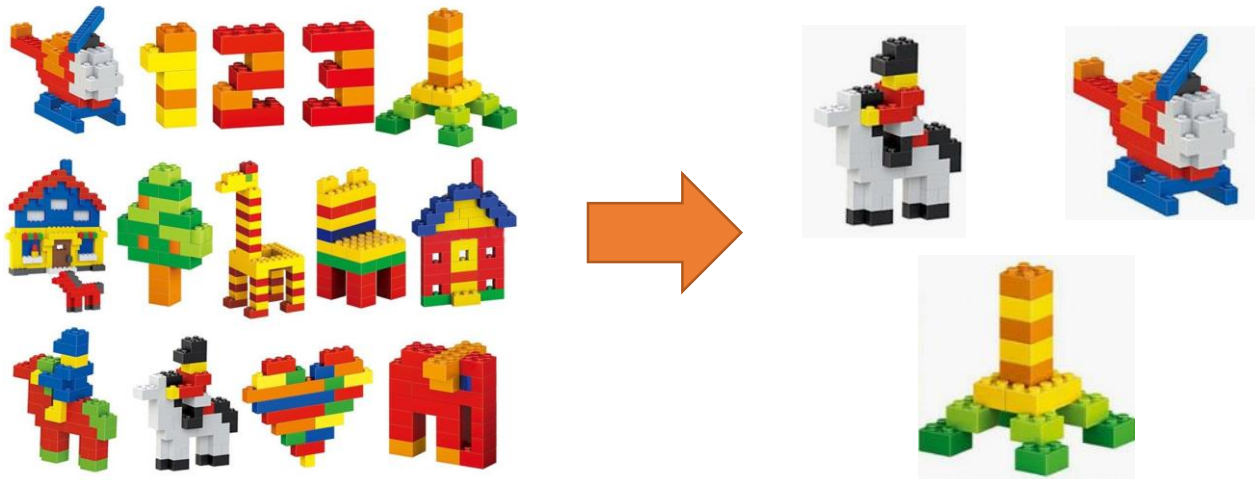
# Start and integrate SPDK into your system

SPDK provides:

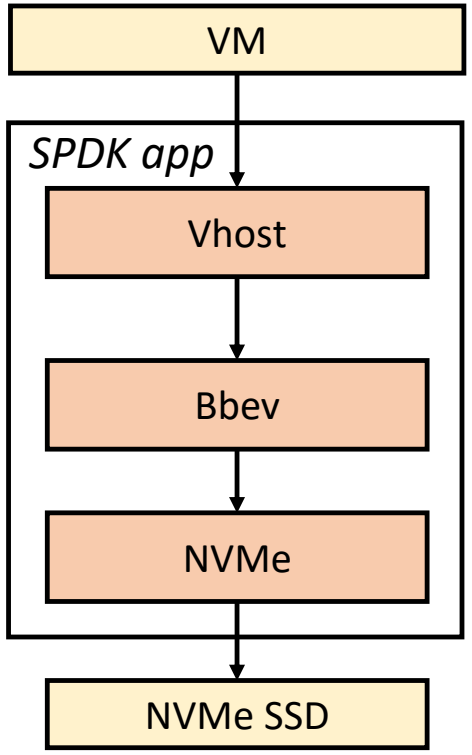
- ✓ Basic libraries for storage software: NVMe, vhost, FTL, cache, etc.
- ✓ Application framework with polling-mode event handling.

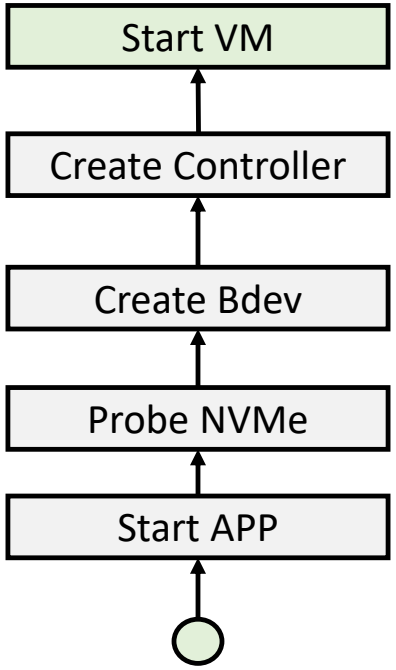
Start with app framework:

- Initialize DPDK *env* with hugepages and cores
- Start SPDK application
- Select and load modules you needed



*Example:*

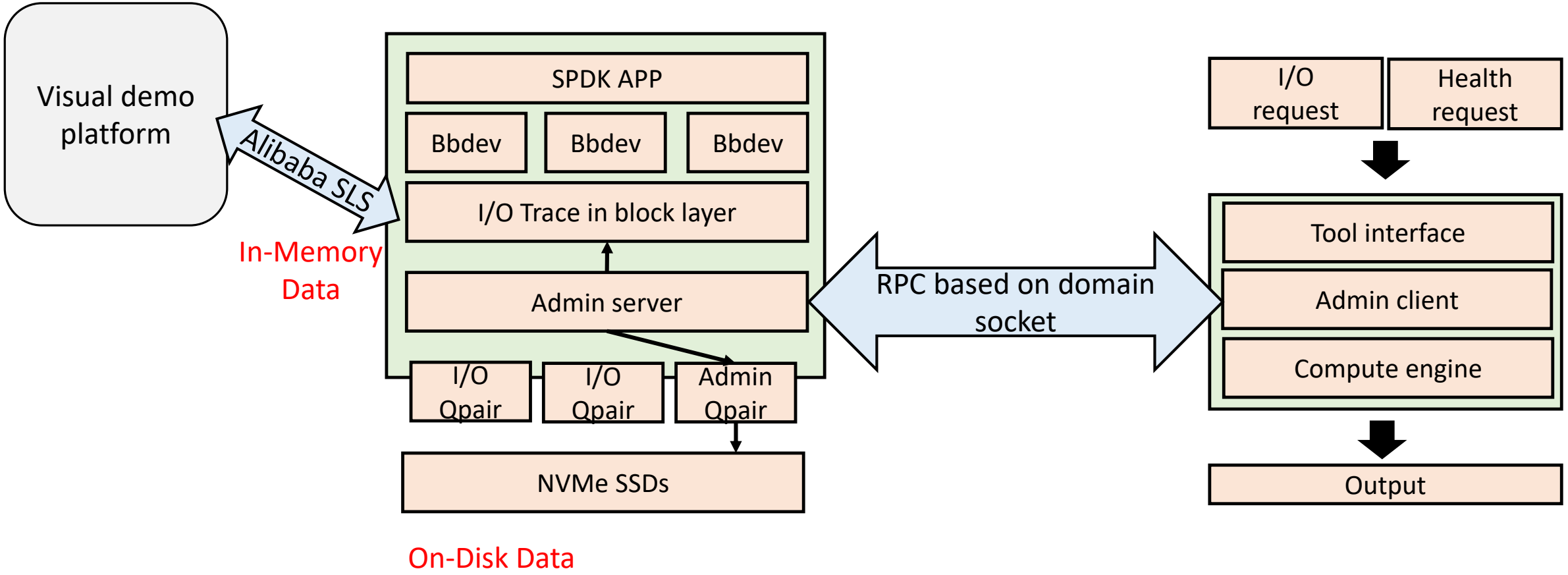


# Communicate with SPDK-managed storage devices

## Get general information – domain socket

- ✓ I/O information: blktrace, iostat, per-segment latency, FTL GC I/O, etc.
- ✓ Health information: SMART log, SMART extended log, private firmware log, etc.

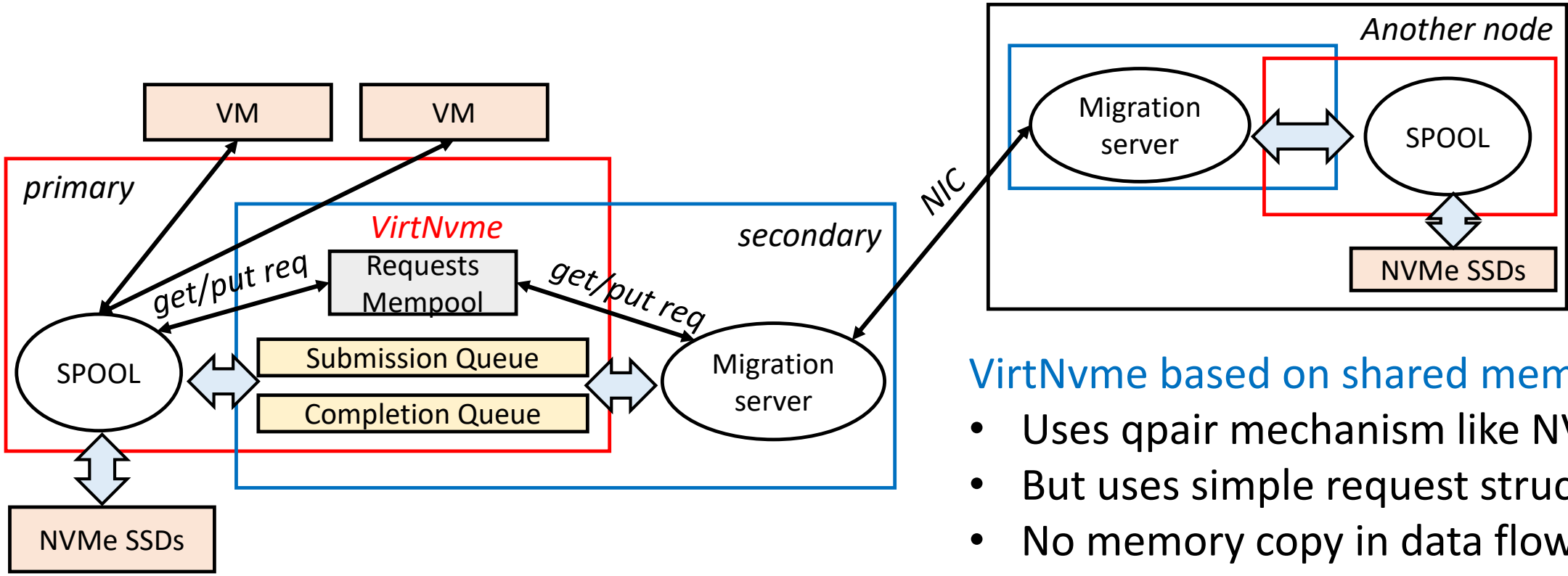


# Communicate with SPDK-managed storage devices

## Multi-process access single device – VirtNvme

✓ VM migration:

- Primary process is serving for VMs
- Secondary process will transform data between two servers using pre-copy




## VirtNvme based on shared memory:

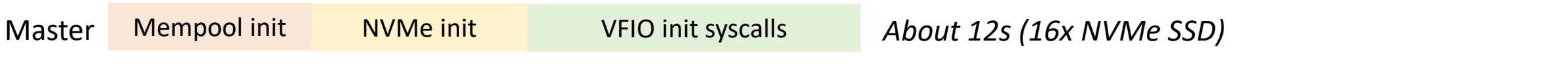
- Uses qpair mechanism like NVMe
- But uses simple request structure
- No memory copy in data flow

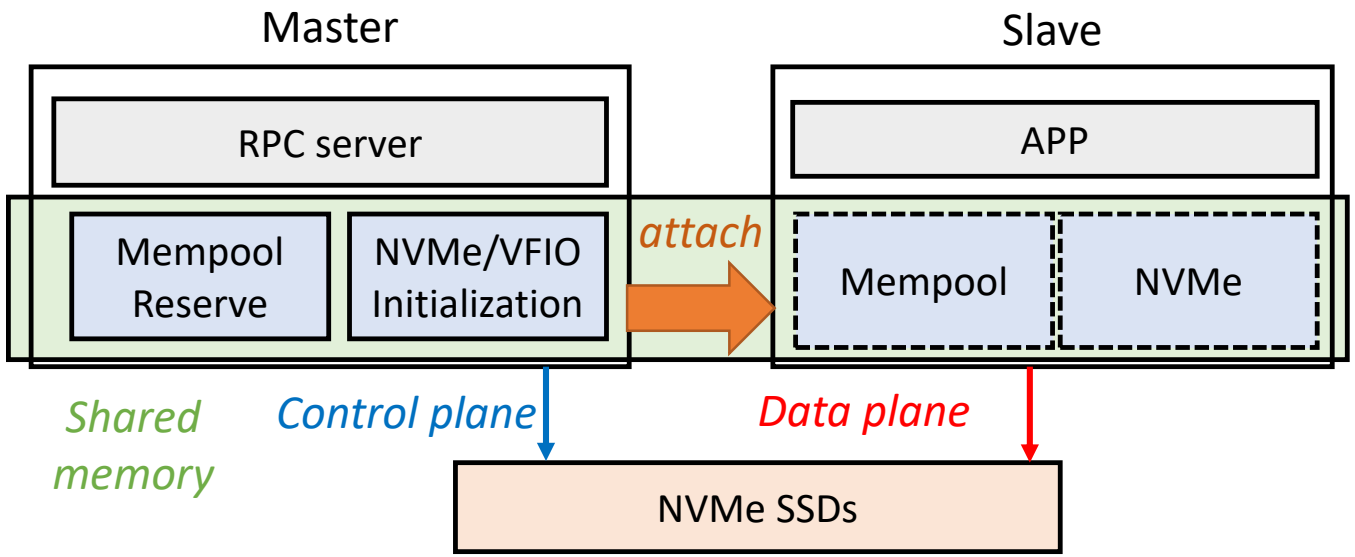
# Fast recovery and consistency

## Problem

✓ Unexpected crash and software upgrading happens everyday

- Impact user I/O downtime -> master-slave mode
- Impact data consistency




Master-Slave:

- Master initializes all initialization tasks;
- Keep master as simple as possible;
- Slave provide I/O services for applications;
- Control plane: *nvme probe/attach/detach;*
- RPC handles request for control plane.

# Fast recovery and consistency

## Problem

✓ Unexpected crash and software upgrading happens everyday

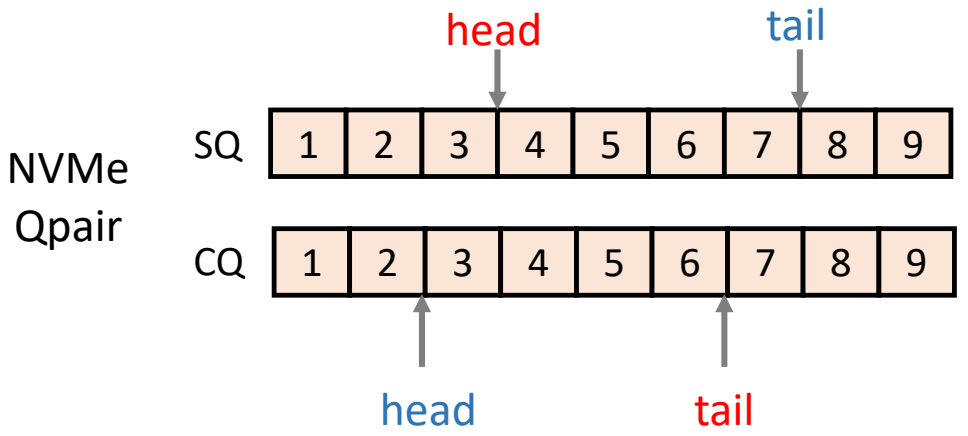
- Impact user I/O downtime -> master-slave mode
- Impact data consistency -> cross-process journal

Vhost NVMe shared qpair:

- Guest OS: SQ tail and CQ head
- Vhost: SQ head and CQ tail

If vhost crash:

- Lose SQ head and CQ tail
- Lose NVMe requests of inflight I/Os




*Note:*
*NVMe uses Out-of-Order processing (return order of NVMe commands are uncertain). When vhost fetches a NVMe command, the entry may be overwritten for next request by guest OS.*

# Fast recovery and consistency

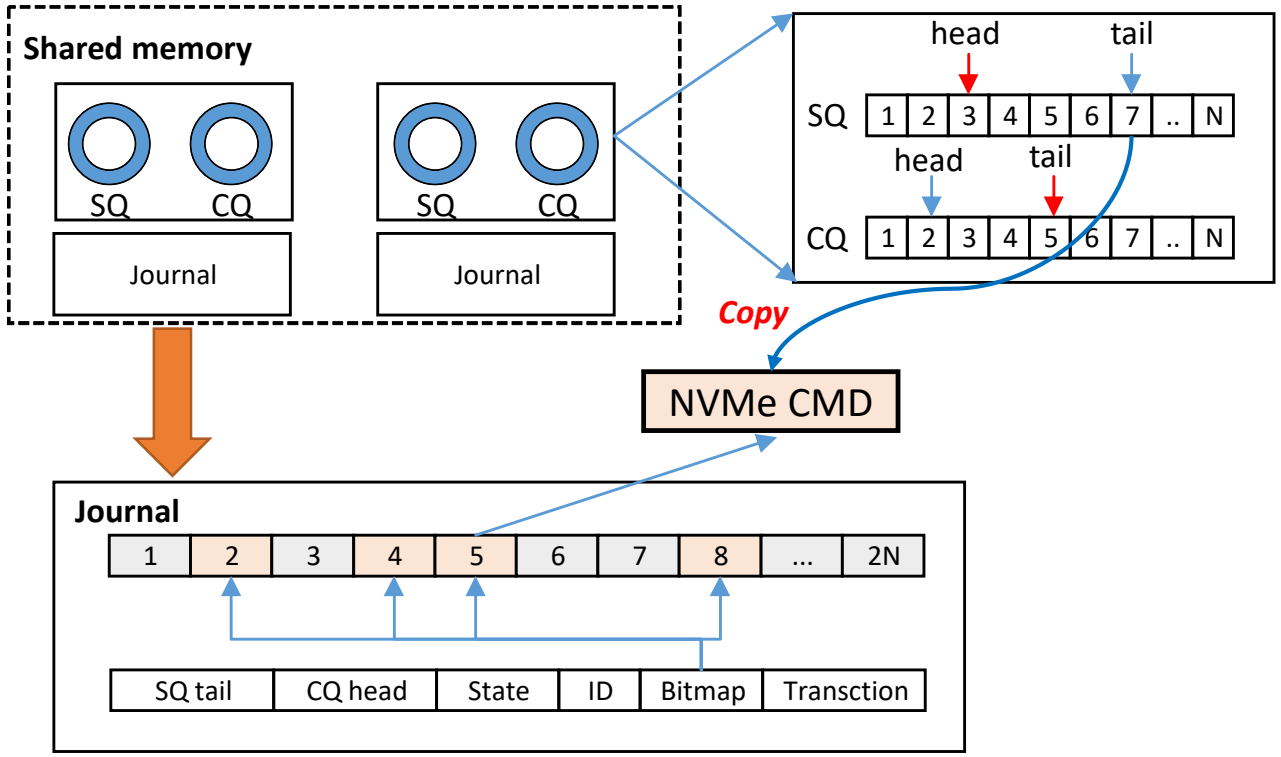
## Problem

✓ Unexpected crash and software upgrading happens everyday

- Impact user I/O downtime -> master-slave mode
- Impact data consistency -> cross-process journal

Cross-process journal

- Maintain SQ tail and CQ head
- Bitmap: record which one is inflight
- Replay inflight requests after crash
- Use same transaction mechanism as our ATC paper to update all members atomically.



# Re-innovate new system based SPDK

See following presentations for detailed introduction:

- ✓ Alibaba-Intel innovation - CSAL: cloud storage acceleration layer.
- ✓ Alibaba I4 instance: hardware/software co-design for NVMe virtualization.

Alibaba Group
阿里巴巴集团

# THANKS

--------- Q&A Section --------