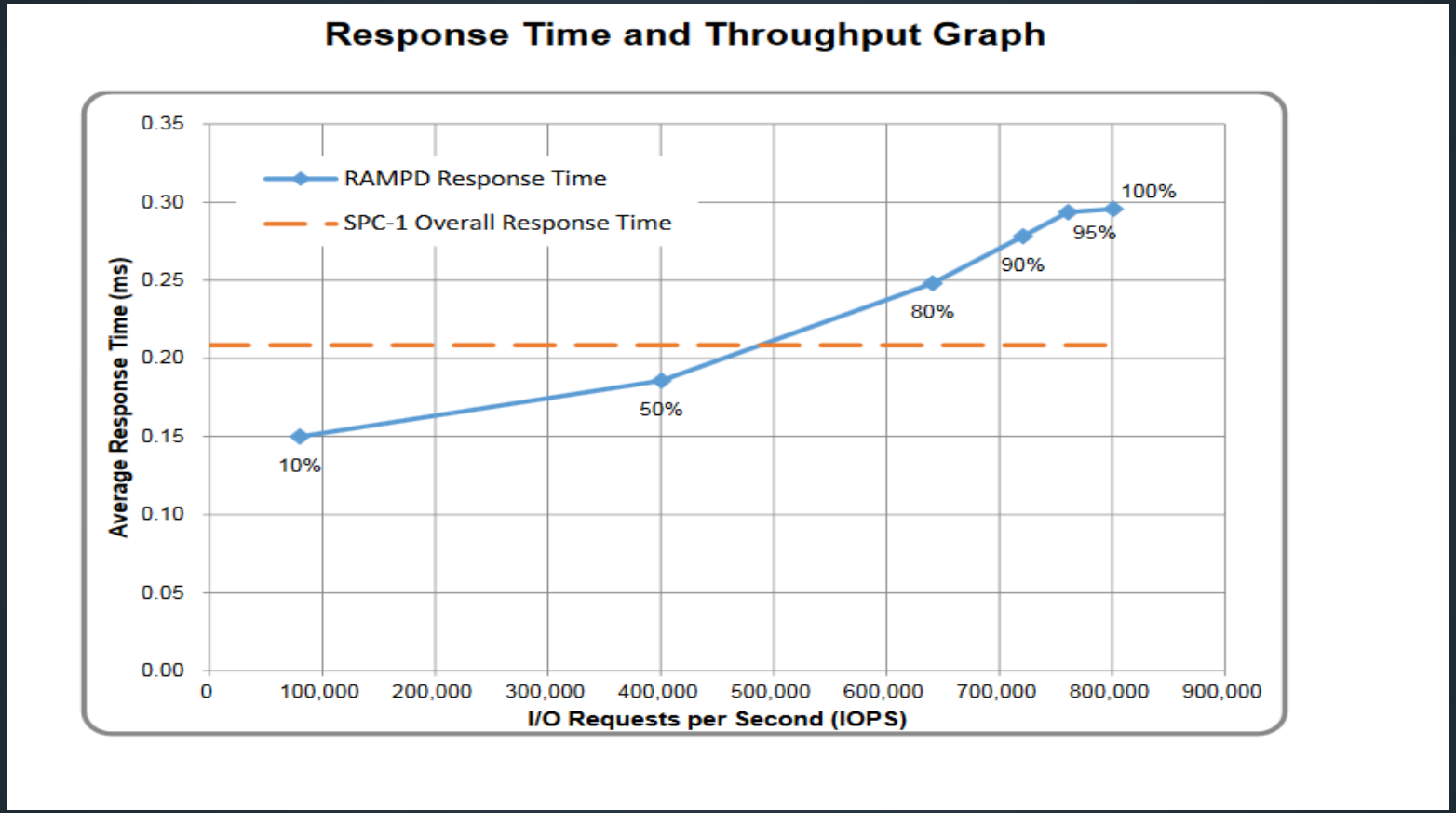
FusionStor

Performance Tuning of Distributed Block System

Wuqiang Qi
Architect of Distributed Storage

About Fusionstack

- The first HCI case in China（**FusionStack®**）
- SPC（Storage Performance Council）**1/4 Chinese Members**
- Top 1 SPC-1 by Price-Performance（**FusionStor®** ）



Fusionstor SPDK-based SPC-1 Result

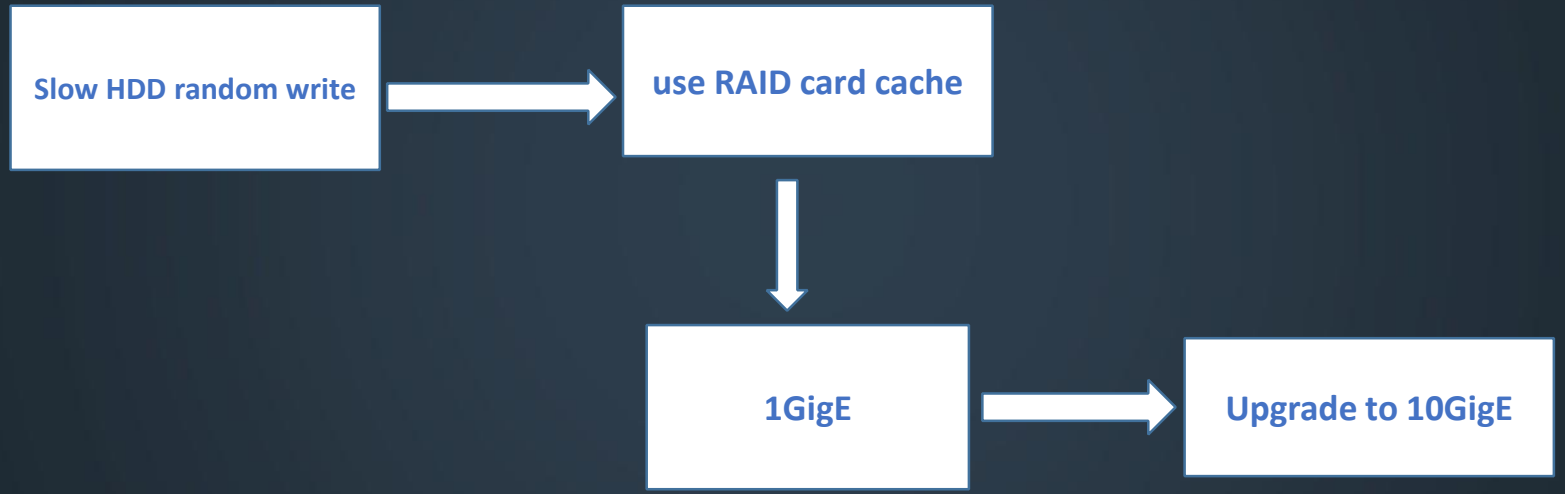
Criteria of Storage Product?

High Availability

Functionality

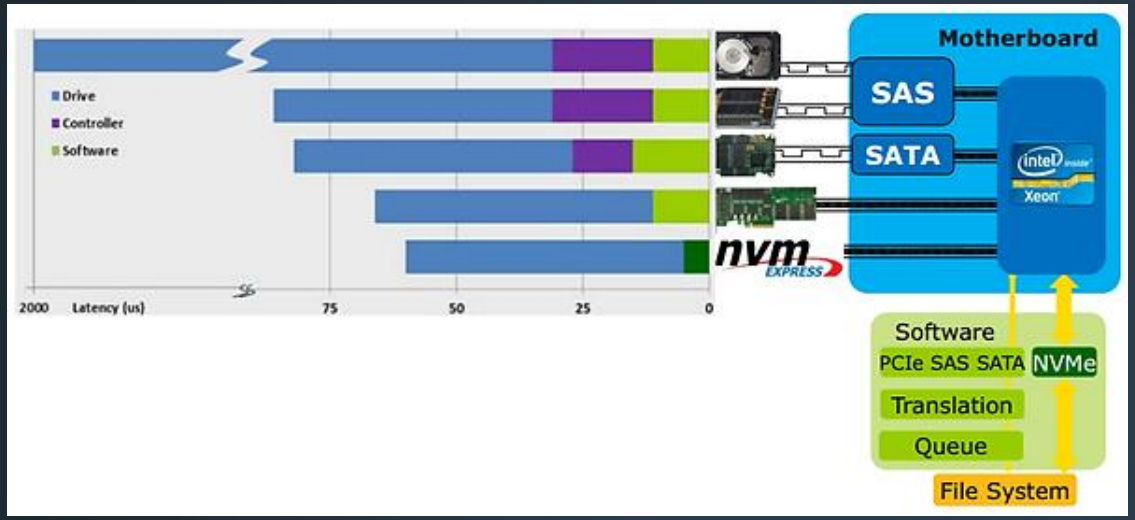
Performance

Performance Tuning in HDD age



What about using SSD to achieve High Performance?

NVMe SSD is much faster than SATA SSD



PCI-E has less latency

NVMe Shorten the cost of the protocol

How should software architecture adapt to NVMe SSD?

The Two Performance Killers correlating to kernel

Context switch

Memory copy

time cost of context switch

```
Context switching - times in microseconds - smaller is better
-------------------------------------------------------------------------
Host                 OS  2p/0K 2p/16K 2p/64K 8p/16K 8p/64K 16p/16K 16p/64K
                         ctxsw  ctxsw  ctxsw ctxsw  ctxsw   ctxsw   ctxsw
--------- ------------- ------ ------ ------ ------ ------ ------- -------
stor01    Linux 3.10.0- 2.1500 2.2300 2.0100 2.9100 4.2300 3.44000 3.46000
```

2M task ,10 cpu core

Some cases of context switch

CPU time slice exhaustion

Preempted by other threads

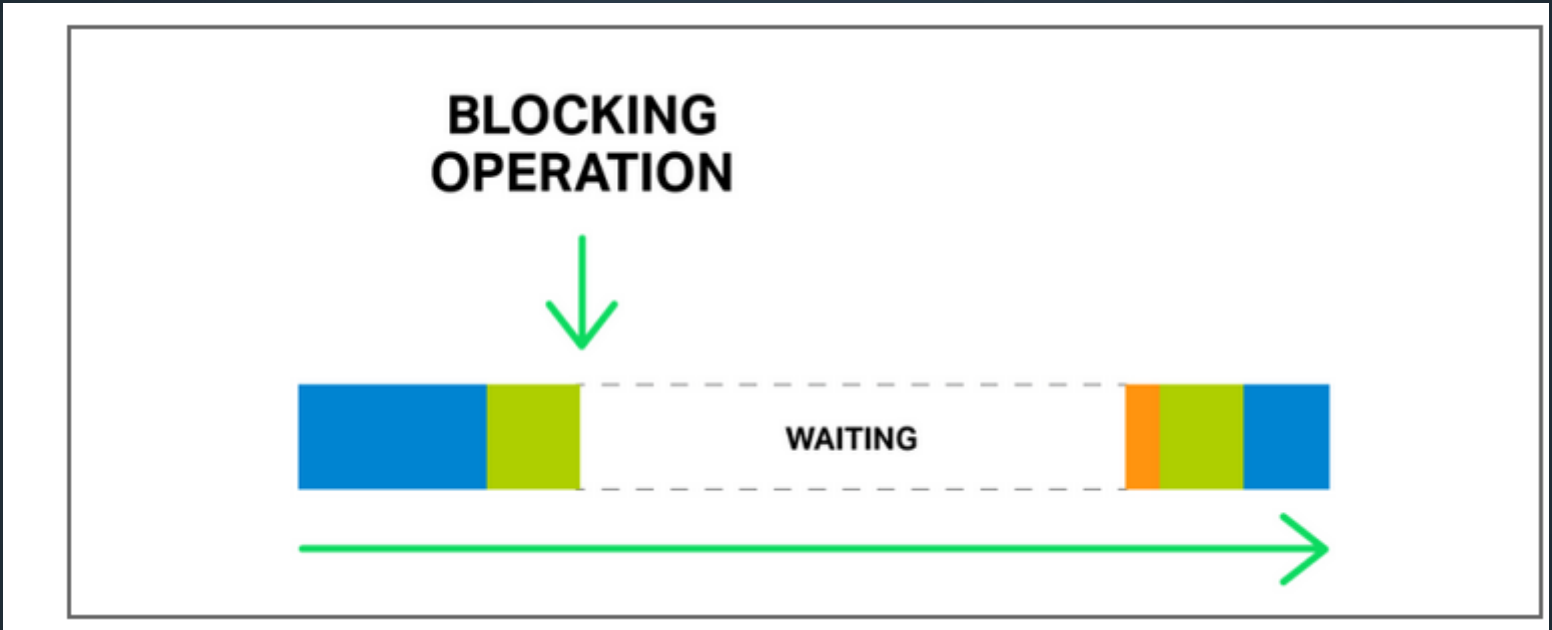
Blocking syscall

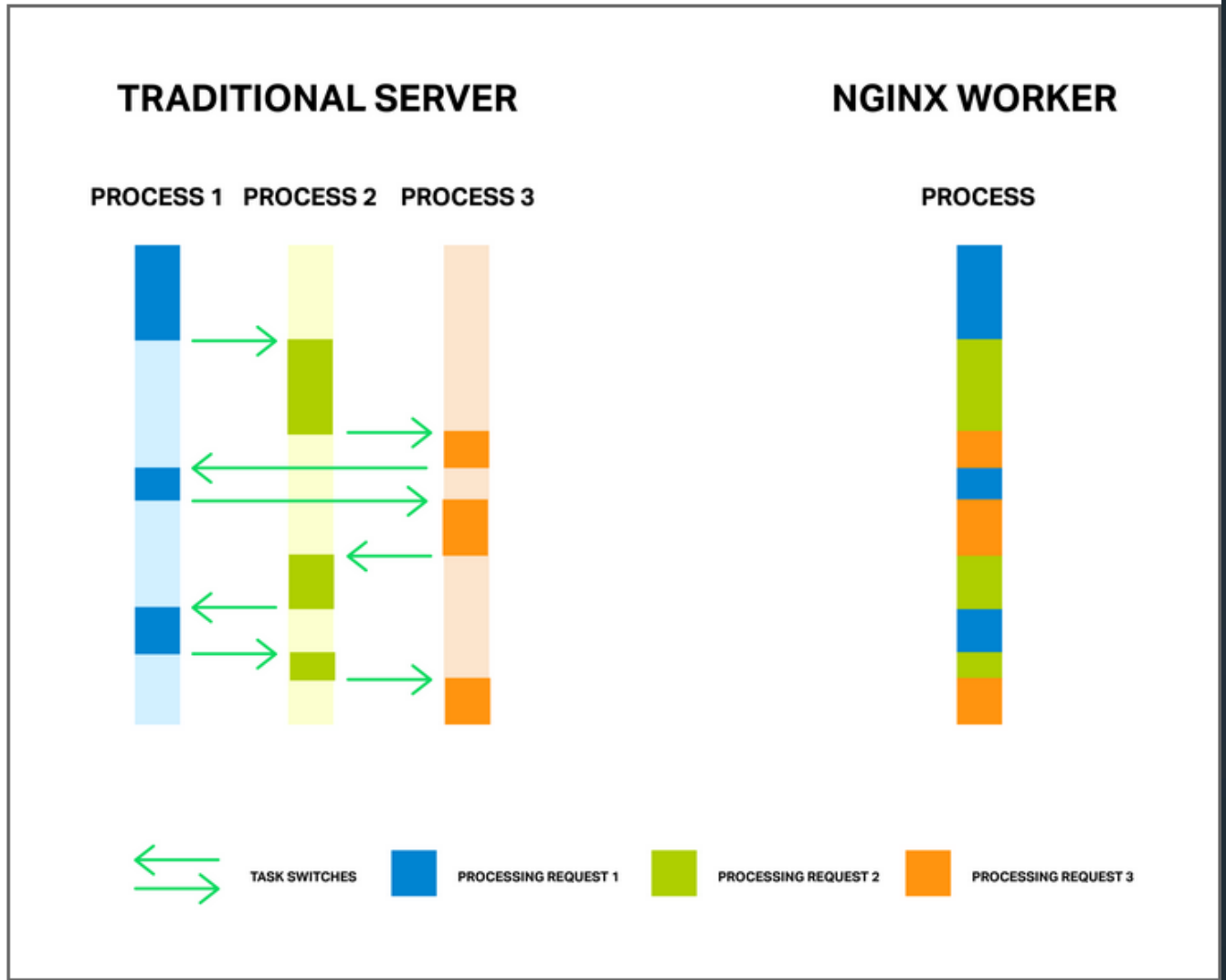
Yield to another thread

Blocked by lock

Hardware interrupts

Blocking Task

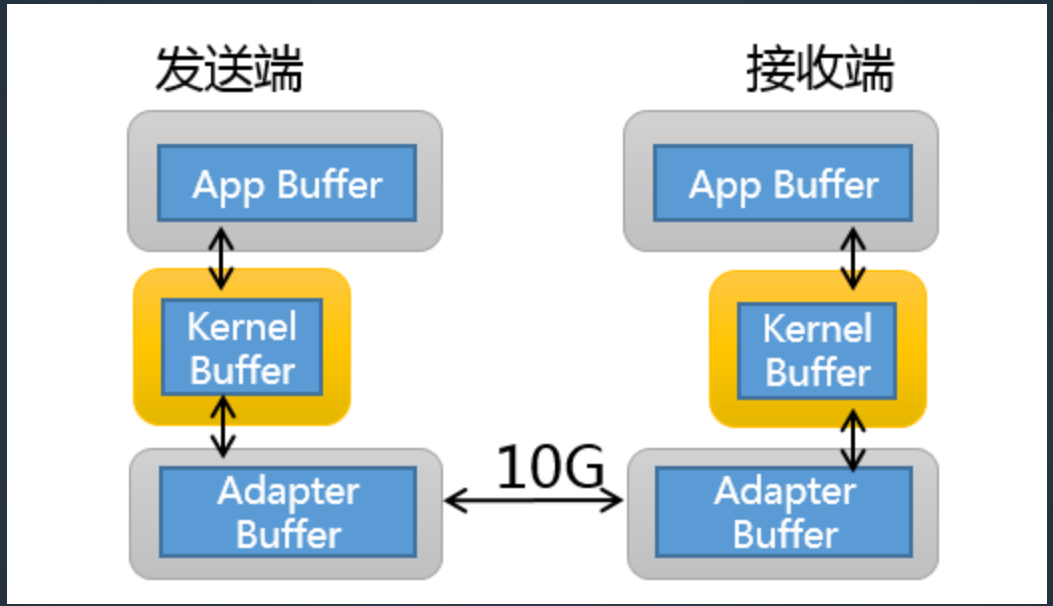



TRADITIONAL SERVER
NGINX WORKER
PROCESS 1
PROCESS 2
PROCESS 3
PROCESS
TASK SWITCHES
PROCESSING REQUEST 1
PROCESSING REQUEST 2
PROCESSING REQUEST 3

Elapsed time of memory copy

```
*Local* Communication bandwidths in MB/s - bigger is better
-----------------------------------------------------------------------------
Host                OS  Pipe AF    TCP  File   Mmap  Bcopy  Bcopy  Mem   Mem
                             UNIX      reread reread (libc) (hand) read write
--------- ------------- ---- ---- ---- ------ ------ ------ ------ ---- -----
stor01    Linux 3.10.0- 3772 7802 7679 5885.7 8796.0 7048.5 4767.1 9380 5865.
```

2M iops ,4k blocksize , 10 cpu core



Memory copy in network layer

lock free model

nonblocking syscalls

using as few threads as possible (proportional to core number)

Less memory copy in I/O path（rdma/dpdk zero copy）

Less interrupt（rdma/dpdk）

The kernel isn’t the solution. The kernel is the problem

Programming Model：sync nonblock

- Scheduling：coroutine
- Event Handling：polling
- Multi-core sync： session based hash

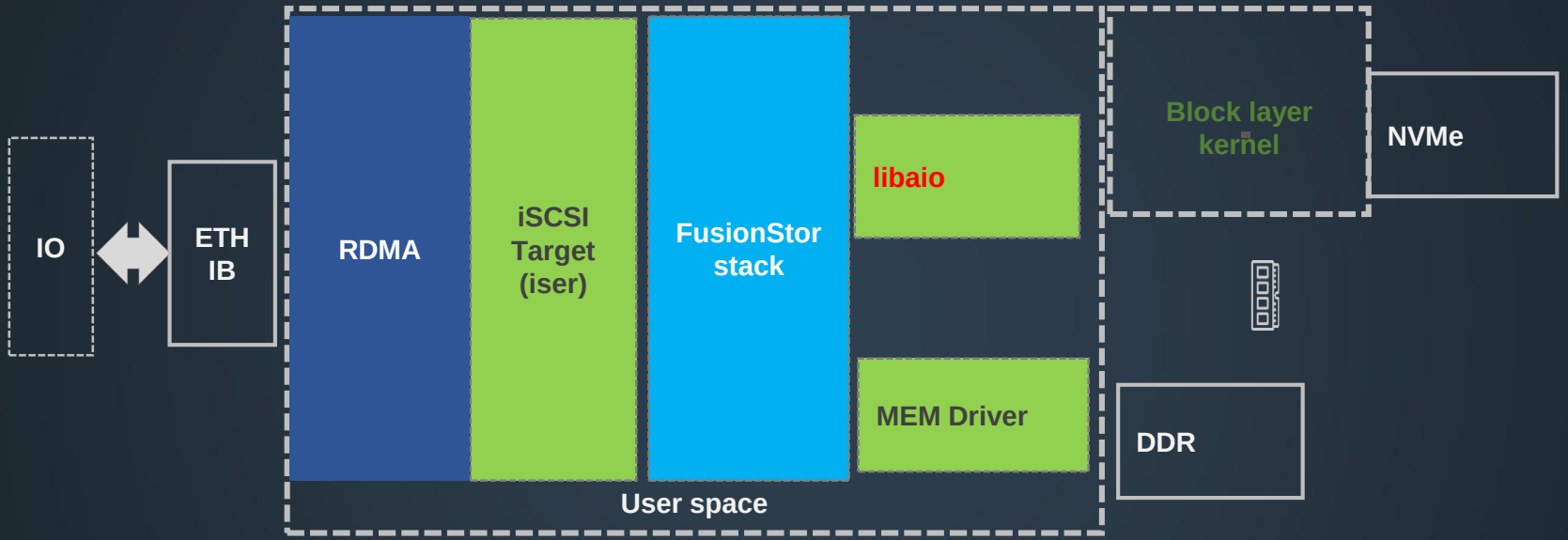
Data Path：kernel-bypass

- Network：RDMA / DPDK
- Flash: libaio
- Mem：hugepage

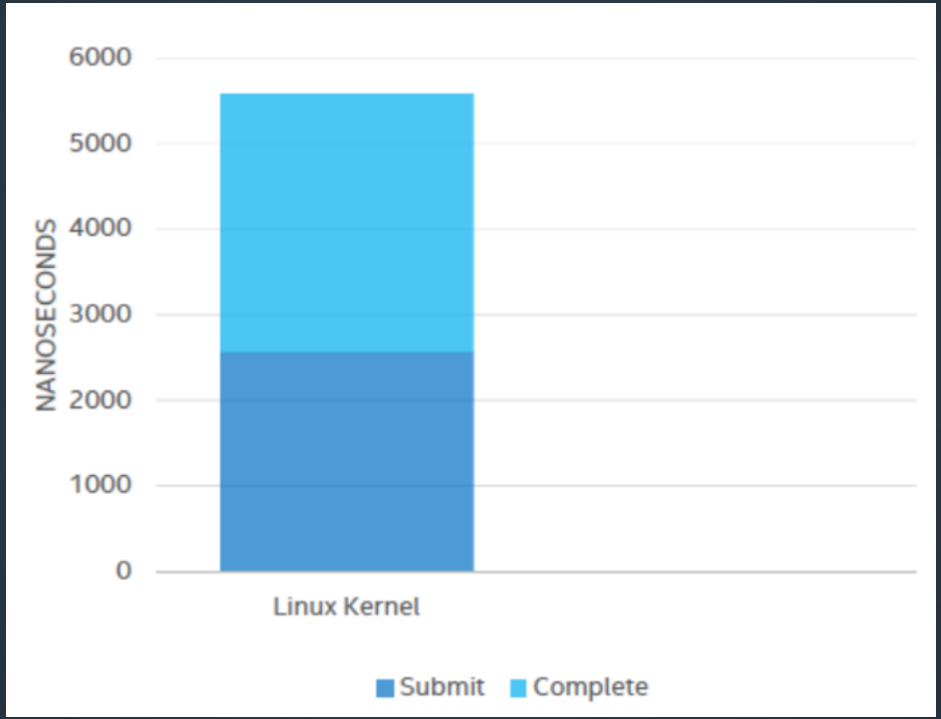
Other Performance Influencing Factors

- Software Algorithm
- Lock
- CPU (cache, TLB, branch prediction)
- Memory (page fault, NUMA)
- Compiler

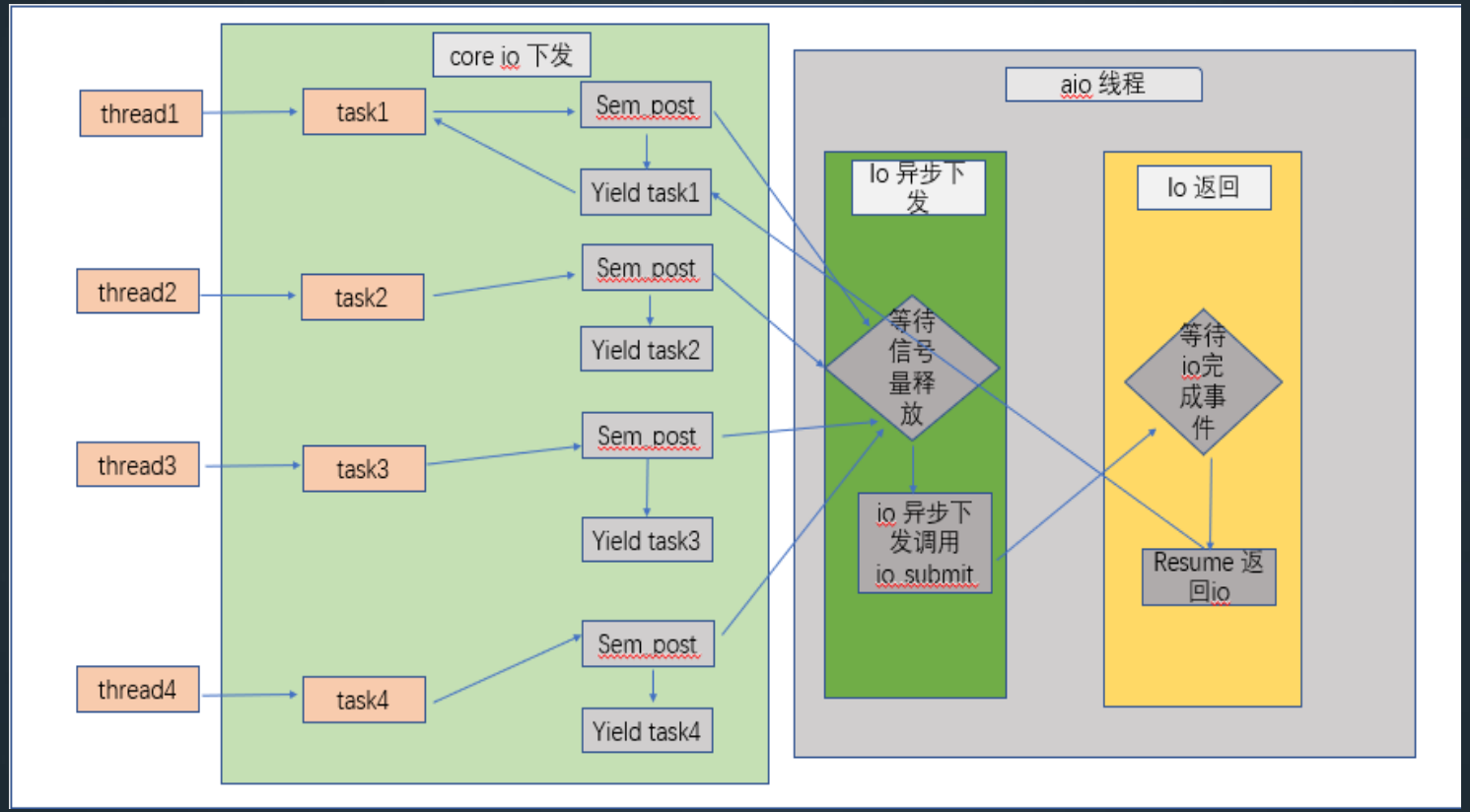
RDMA-based Storage Architecture



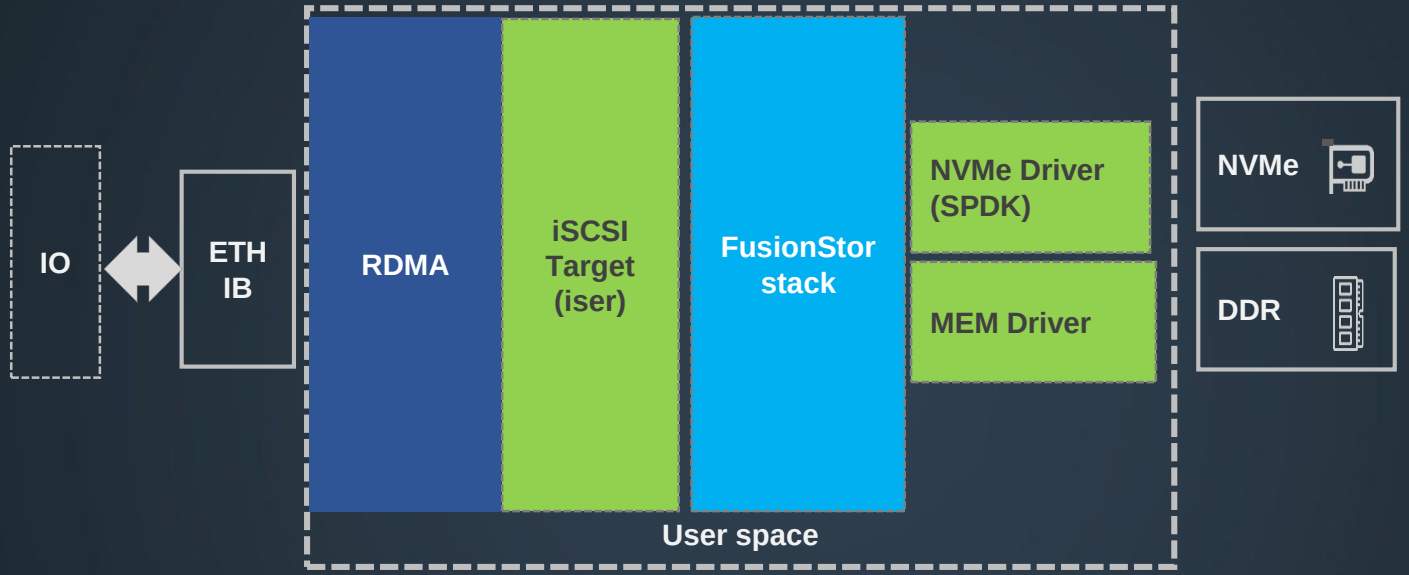
IO processing time in the kernel



the io path with libaio in fusionstor

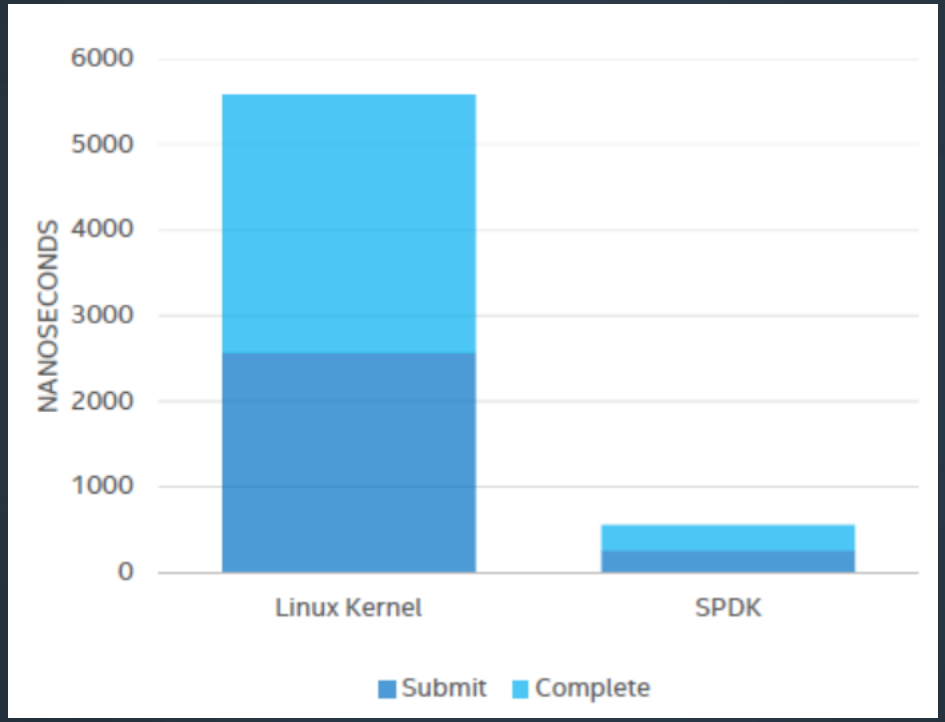


Maximize NVMe Performance using kernel-bypass architecture

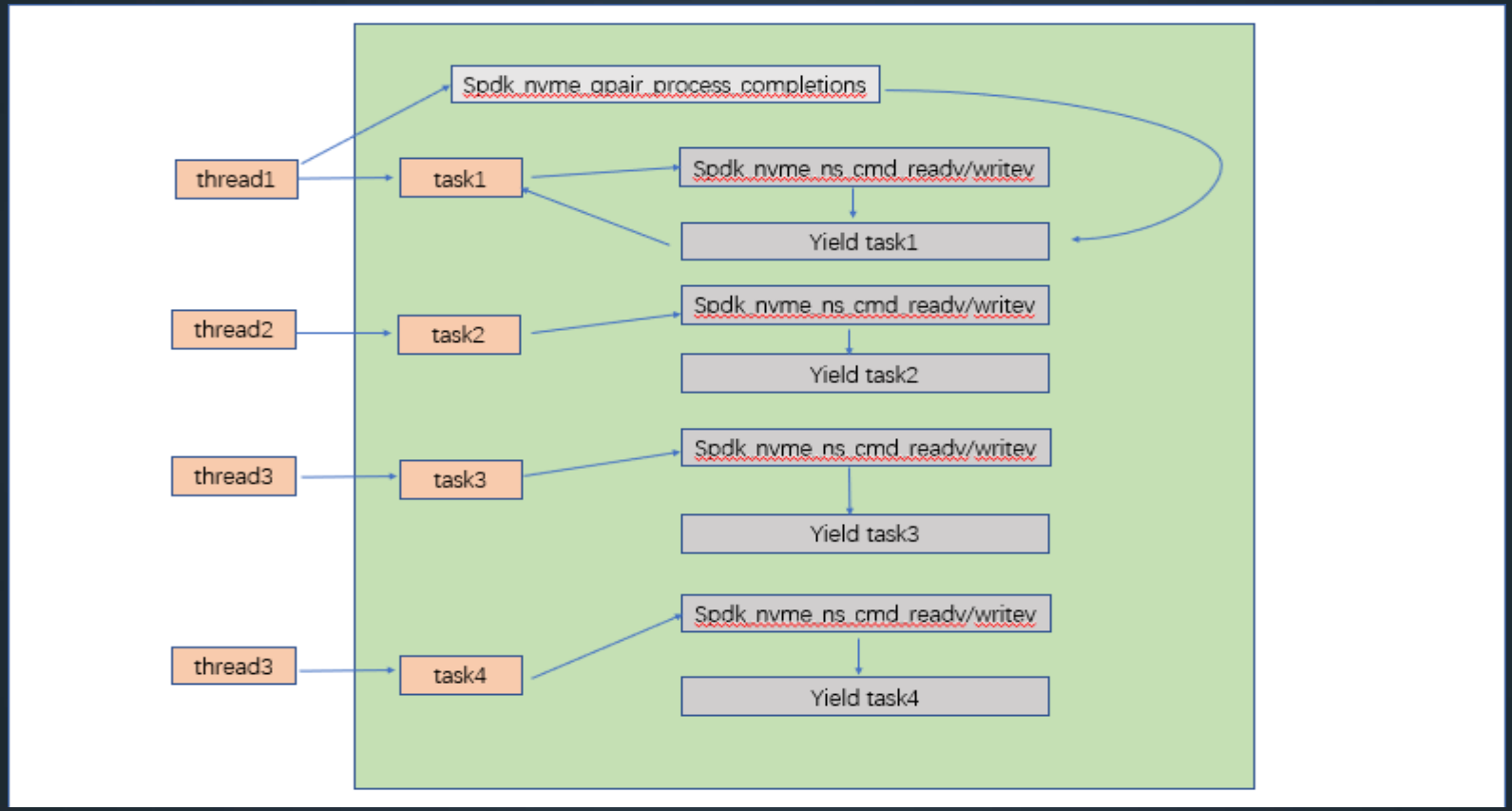


- ✓ Support RDMA、NVMe，Intel SPDK-based software architecture
- ✓ User space，zero copy
- ✓ Polling model，low interrupt cost，high io concurrency

IO processing time when used SPDK



the io path with SPDK in fusionstor



测试场景	持续时间	Vdbench性能均值											
		i/o rate	MB/sec 1024**2	bytes i/o	read pct	resp time	read resp	write resp	resp max	resp stddev	queue depth	cpu% sys+u	cpu% sys
7:3随机读写 性能测试,单主机压测，8个LUN 520Gx8(未启动SPDK)													
4K数据块;7:3随机读写;队列=16测试;IOPS不限制	30分钟	492023.71	1921.97	4096	70	0.258	0.275	0.22	728.122	0.302	127	25	19.6
4K数据块;7:3随机读写;队列=32测试;IOPS不限制	30分钟	557196.07	2176.55	4096	70	0.457	0.448	0.479	1358.381	0.689	254.8	34.9	28.7
4K数据块;7:3随机读写;队列=64测试;IOPS不限制	30分钟	559168.42	2184.25	4096	70	0.913	0.845	1.071	897.202	0.702	510.4	38.1	31.9
8K数据块;7:3随机读写;队列=8测试;IOPS不限制	30分钟	295836.34	2311.22	8192	70	0.215	0.259	0.112	551.854	0.348	63.5	11.7	8.6
8K数据块;7:3随机读写;队列=16测试;IOPS不限制	30分钟	433176.71	3384.19	8192	70	0.293	0.339	0.188	1044.549	0.41	127.1	21	16
8K数据块;7:3随机读写;队列=64测试;IOPS不限制	30分钟	511724.05	3997.84	8192	70	0.997	0.949	1.111	819.263	0.707	510.4	32.9	26.5
32K数据块;7:3随机读写;队列=8测试;IOPS不限制	30分钟	142374.6	4449.21	32768	70	0.447	0.565	0.174	1006.528	0.661	63.7	6.6	4.5
32K数据块;7:3随机读写;队列=16测试;IOPS不限制	30分钟	170767.7	5336.49	32768	70	0.747	0.8	0.623	1296.495	0.813	127.5	8.5	5.9
32K数据块;7:3随机读写;队列=64测试;IOPS不限制	30分钟	170001.66	5312.55	32768	70	3.008	2.66	3.82	1413.265	1.688	511.4	8.6	5.5
256K数据块;7:3随机读写;队列=4测试;IOPS不限制	30分钟	21053.28	5263.32	262144	70.01	1.511	1.744	0.968	1535.661	1.083	31.8	2.7	1.6
256K数据块;7:3随机读写;队列=8测试;IOPS不限制	30分钟	21525.95	5381.49	262144	70.01	2.963	2.905	3.1	875.016	1.273	63.8	2.9	1.7
7:3随机读写 性能测试,单主机压测，每个主机8个LUN 520Gx8(启动SPDK)													
4K数据块;7:3随机读写;队列=16测试;IOPS不限制	30分钟	770306.81	3009.02	4096	70	0.16	0.2	0.06	23.19	0.29	126.4	39.2	31.6
4K数据块;7:3随机读写;队列=32测试;IOPS不限制	30分钟	873889.2	3413.63	4096	70	0.29	0.31	0.25	24.84	0.32	253.7	54.8	46.8
4K数据块;7:3随机读写;队列=64测试;IOPS不限制	30分钟	859147.85	3356.05	4096	70	0.59	0.57	0.64	26.83	0.4	508.7	59.1	51.8
8K数据块;7:3随机读写;队列=8测试;IOPS不限制	30分钟	324842.1	2537.83	8192	70	0.2	0.25	0.06	21.28	0.37	63.5	12.4	9.1
8K数据块;7:3随机读写;队列=16测试;IOPS不限制	30分钟	504113.12	3938.38	8192	70	0.25	0.33	0.08	21.58	0.45	127	23	17.3
8K数据块;7:3随机读写;队列=64测试;IOPS不限制	30分钟	612571.34	4785.71	8192	70	0.83	0.83	0.85	23.45	0.58	509.9	41.1	33.8
32K数据块;7:3随机读写;队列=8测试;IOPS不限制	30分钟	134259.02	4195.59	32768	70	0.48	0.63	0.11	13.29	0.67	63.7	6.1	4.1
32K数据块;7:3随机读写;队列=16测试;IOPS不限制	30分钟	170655.97	5333	32768	70	0.75	0.87	0.47	19.67	0.67	127.6	8.4	5.8
256K数据块;7:3随机读写;队列=4测试;IOPS不限制	30分钟	20256.87	5064.22	262144	70	1.57	1.93	0.72	12.61	1.04	31.8	2.6	1.5
256K数据块;7:3随机读写;队列=8测试;IOPS不限制	30分钟	21529.83	5382.46	262144	70	2.96	3.04	2.78	17.6	1.16	63.8	3	1.7
大变化IO size，随机读写 性能测试,2个主机并发压测，每个主机8个LUN 520x8													

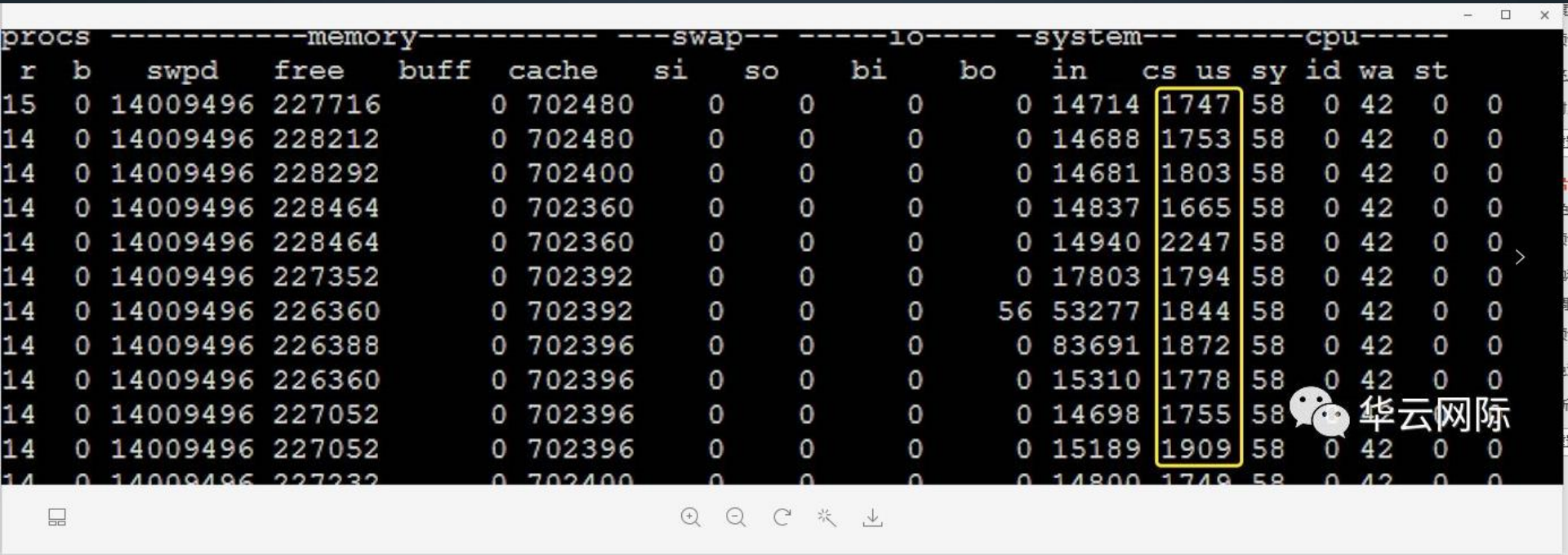
Using SPDK: Yes VS No

Programming Model：Sync nonblock

- Scheduling：coroutine
- Event Handling：polling Model
- Multi-core sync： session based hash

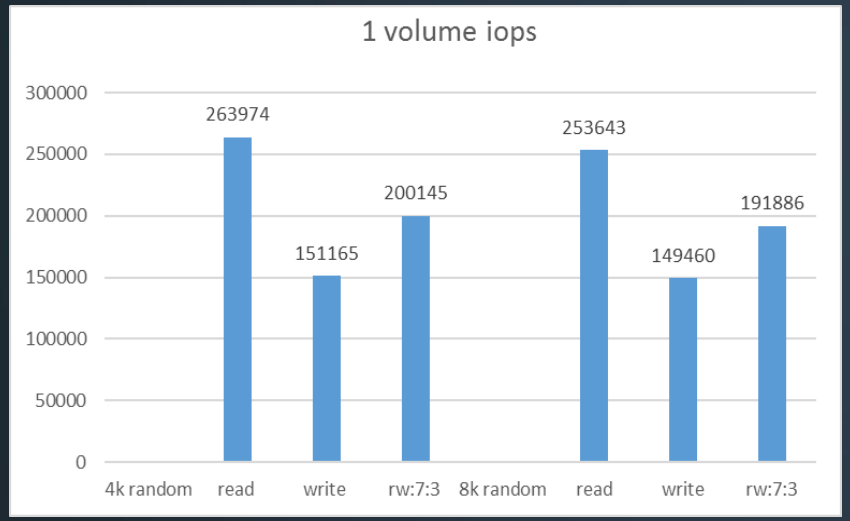
I/O Path：kernel-bypass

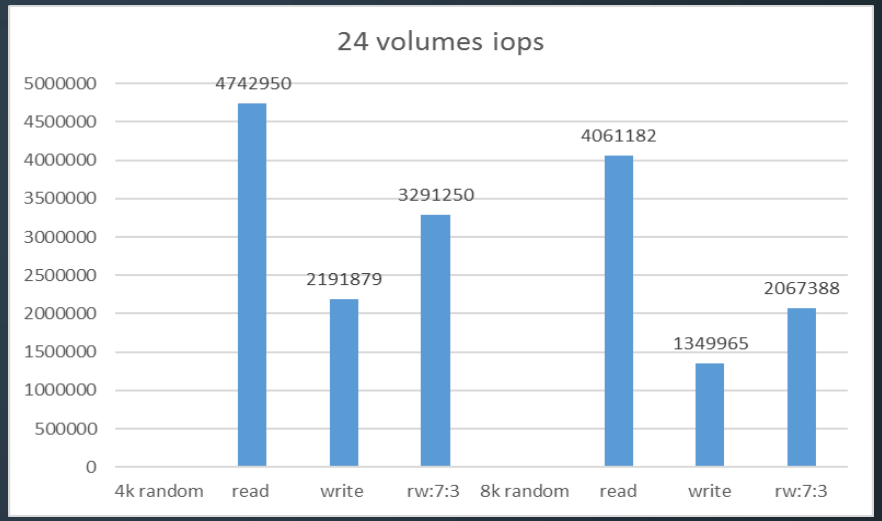
- Network：RDMA / DPDK
- Flash：SPDK
- Mem：hugepage



eliminate context switch

Performance result of using kernel-bypass





THANK YOU