STORAGE DEVELOPER CONFERENCE

SDC 21

BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

A SNIA Event

SPDK Schedulers

Realizing Power Savings in Polled Mode Applications

Tomek Zawadzki and Jim Harris

Intel Corporation

Introductions

Tomek Zawadzki

Jim Harris

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

Notices and Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing on certain dates using certain configurations and may not reflect all publicly available updates. Reach out to Intel for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

SPDK Overview

©2021 Storage Developer Conference ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE

SPDK Overview

- Storage Performance Development Kit
  - Framework for building high-performance storage applications
  - Set of drivers and libraries
  - Includes fully functional storage target applications
  - Userspace, polled-mode programming model
  - Open-source community
  - BSD licensed
  - https://spdk.io

Block Storage Protocols

Networking: NVMe-oF (RDMA, TCP, FC), iSCSI

Virtualization: vhost-scsi, vhost-blk, NVMe vfio-user

Block Storage Services

Partitioning: Logical Volumes, GPT

Caching: OCF

Host FTL: ZNS

Pooling: RAID-0

Transforms: Crypto, Compression

Block Storage Providers

NVMe, io_uring, Linux AIO, virtio, iSCSI, Ceph RBD

Drivers

NVMe (PCIe, RDMA, TCP), **virtio** (scsi, blk), **idxd**, ioat

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

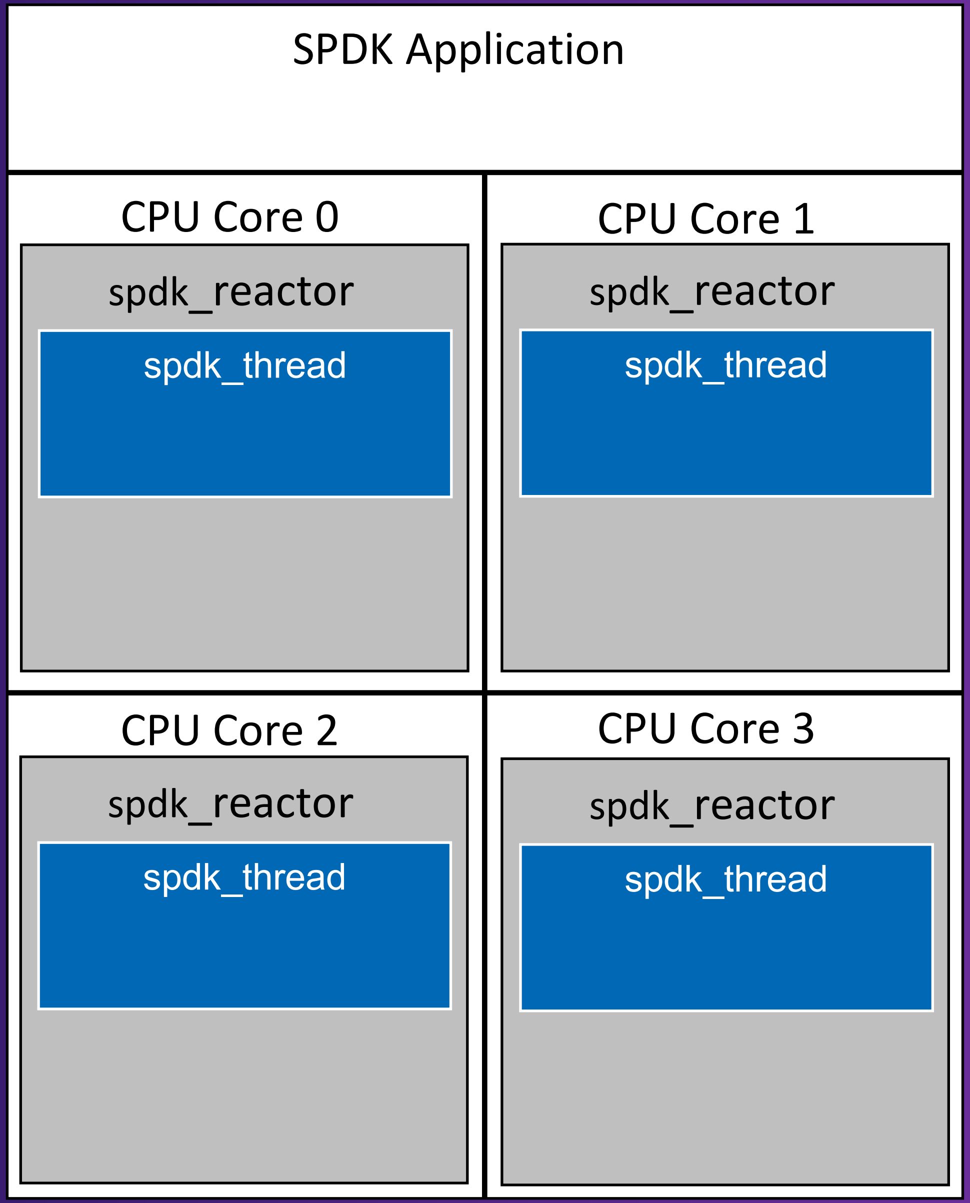
STORAGE DEVELOPER CONFERENCE SDC 21

SPDK Threading Model

©2021 Storage Developer Conference ©. Intel Corporation. All Rights Reserved.

SPDK Threading Model

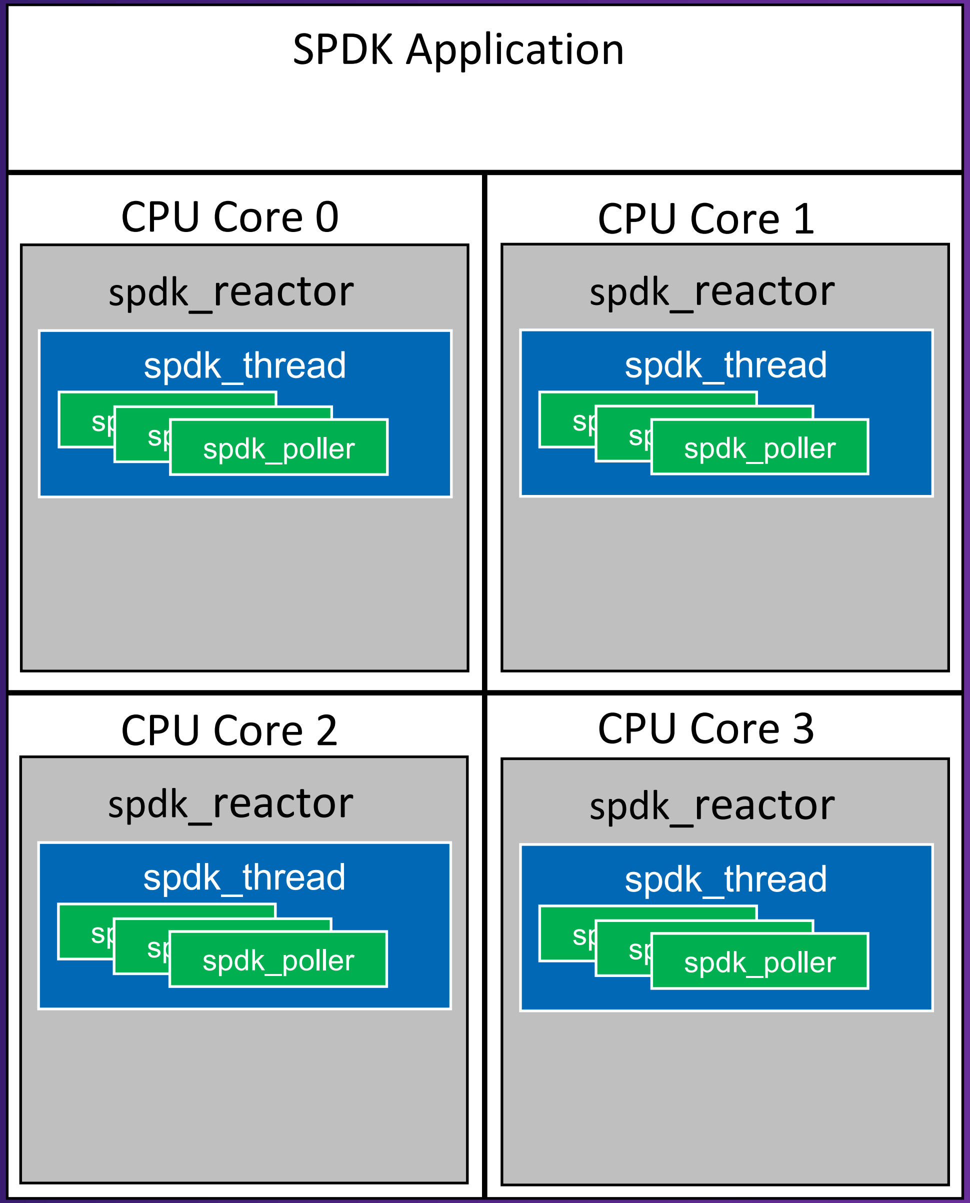
- spdk_reactor
  - One spdk_reactor per CPU core
  - Pinned POSIX thread
  - Created by SPDK application framework
- spdk_thread
  - Lightweight "thread" abstraction
  - By default, one spdk_thread per CPU core
    - Created by top-level block storage protocol library (nvmf, vhost, iscsi)
  - spdk_thread_poll() used by application framework to "run" an spdk_thread



 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

spdk_thread_poll()

- spdk_poller
  - Libraries register spdk_pollers to poll on something
    - NVMe qpair
    - epoll fd (group of TCP sockets or rbd eventfds)
    - RDMA completion queue
  - One call to spdk_thread_poll() runs every spdk_poller once
    - Except for timed pollers
- spdk_thread_send_msg()
  - Used for inter-thread communication



 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE SDC 21

Saving Power When Idle

©2021 Storage Developer Conference ©. Intel Corporation. All Rights Reserved.

All of this polling!

- Polled mode ideal for best performance and efficiency when CPU cores are busy
- But how can we save CPU cycles when we are not as busy?

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

Interrupt Mode

- SPDK does have some limited interrupt mode support
  - Restricted to very small subset of SPDK libraries (not including nvme driver or nvmf target)
  - Supporting libraries register fds with spdk_thread
  - spdk_reactor waits on epoll fd containing fds from all spdk_threads on that reactor
- Overly complex to implement efficiently
  - Avoid nested epoll fd groups
  - *Every* library must be modified to support interrupts

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

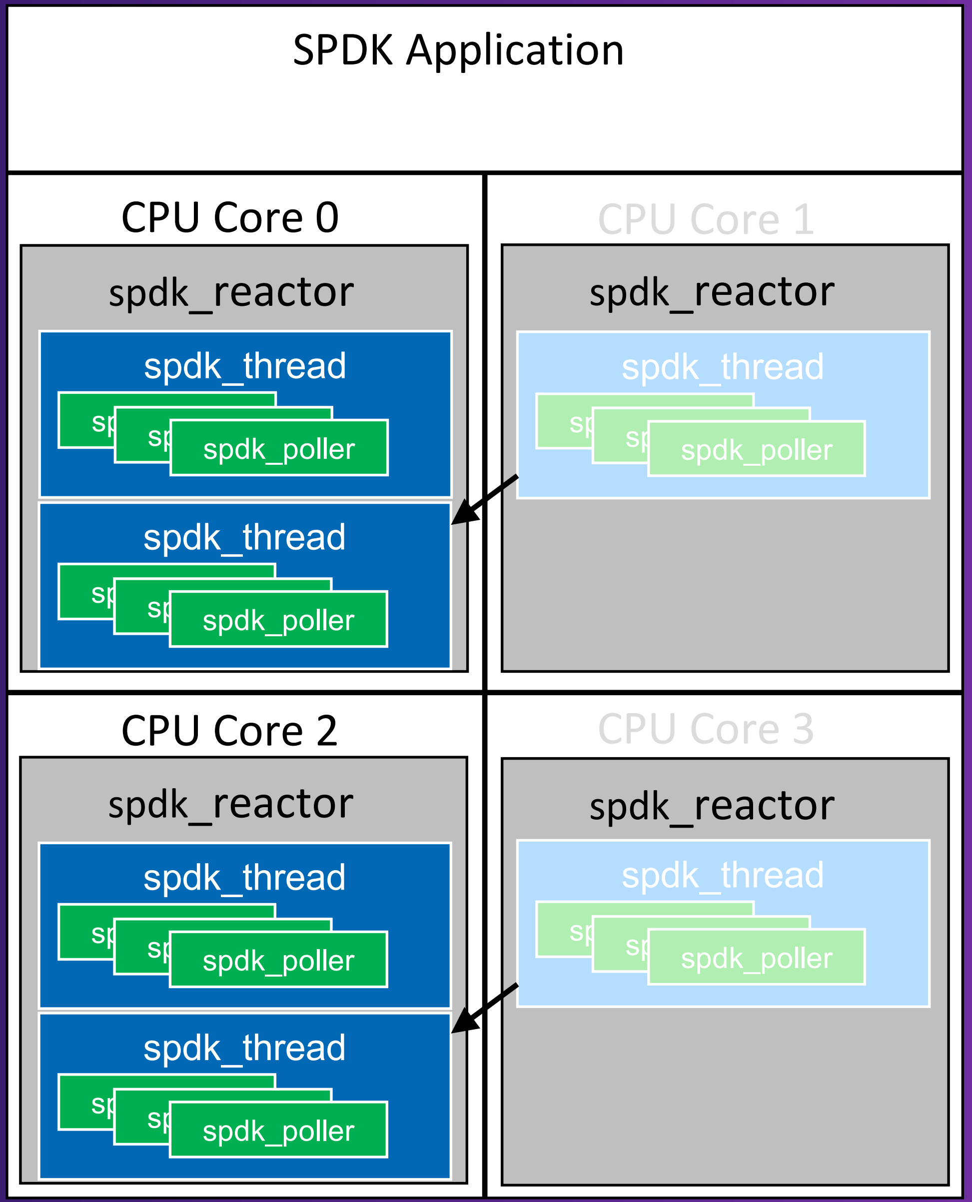
umonitor/umwait

- Newer x86 instructions to allow unprivileged monitor/mwait
- umwait – enables CPU to enter low-power state
  - Exits low-power state on observed write to memory range specified by umonitor
- Works well for one thread polling one HW queue
  - i.e. DPDK packet processing and userspace Ethernet PMDs
- Not suitable when polling many HW queues from one thread
  - Or when polling kernel TCP sockets!

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

Move spdk_threads?

- Would allow putting a CPU core to sleep!
  - While still ensuring the spdk_thread is continually polled (just on a new core)
- Supported by SPDK threading model
  - Since all resources allocated by an spdk_poller are spdk_thread local



 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

SPDK Scheduler Framework

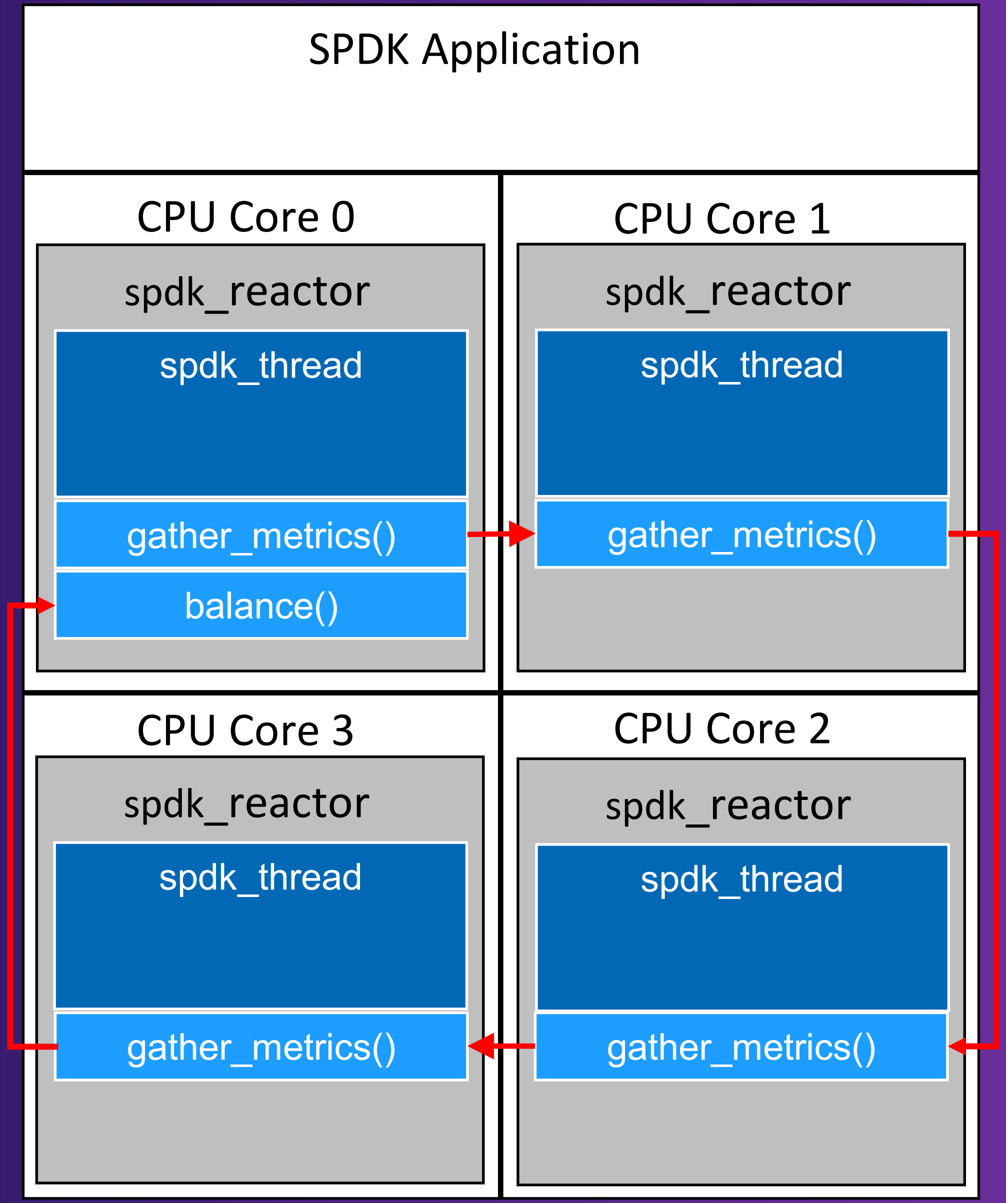
©2021 Storage Developer Conference ©. Intel Corporation. All Rights Reserved.

Scheduling Phases

- Reactors are never halted
- 1) gather_metrics() collects info on core and threads status

```
struct spdk_scheduler_core_info {
        /* stats over a lifetime of a core */
        uint64_t total_idle_tsc;
        uint64_t total_busy_tsc;
        /* stats during the last scheduling period */
        uint64_t current_idle_tsc;
        uint64_t current_busy_tsc;

        uint32_t lcore;
        uint32_t threads_count;
        bool interrupt_mode;
        struct spdk_scheduler_thread_info *thread_infos;
};
```

```
struct spdk_scheduler_thread_info {
        uint32_t lcore;
        uint64_t thread_id;
        /* stats over a lifetime of a thread */
        struct spdk_thread_stats total_stats;
        /* stats during the last scheduling period */
        struct spdk_thread_stats current_stats;
};
```



 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

Balancing Threads

- 2) balance()
  - Change thread's core assignment
  - Put a core to sleep
  - Modify core frequency via governor

- Plug your own !

```
SPDK_SCHEDULER_REGISTER(scheduler_dynamic);
```

```
$ ./scripts/rpc.py framework_set_scheduler dynamic -p 1000000
```

```
struct spdk_scheduler {
	const char *name;

	/**
	* This function is called to initialize a scheduler.
	*
	* \return 0 on success or non-zero on failure.
	*/
	int (*init)(void);

	/**
	* This function is called to deinitialize a scheduler.
	*/
	void (*deinit)(void);

	/**
	* Function to balance threads across cores by modifying
	* the value of their lcore field.
	*
	* \param core_info Structure describing cores and threads on them.
	* \param count Size of the core_info array.
	*/
	void (*balance)(struct spdk_scheduler_core_info *core_info, uint32_t count);

	TAILQ_ENTRY(spdk_scheduler) link;
};
```

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

SPDK Governors

- Use of governors by scheduler is optional
- Dynamic scheduler uses dpdk_governor
  - rte_power library
- Plug your own !

```
SPDK_GOVERNOR_REGISTER(dpdk_governor);
```

```
rc = spdk_governor_set("dpdk_governor");
```

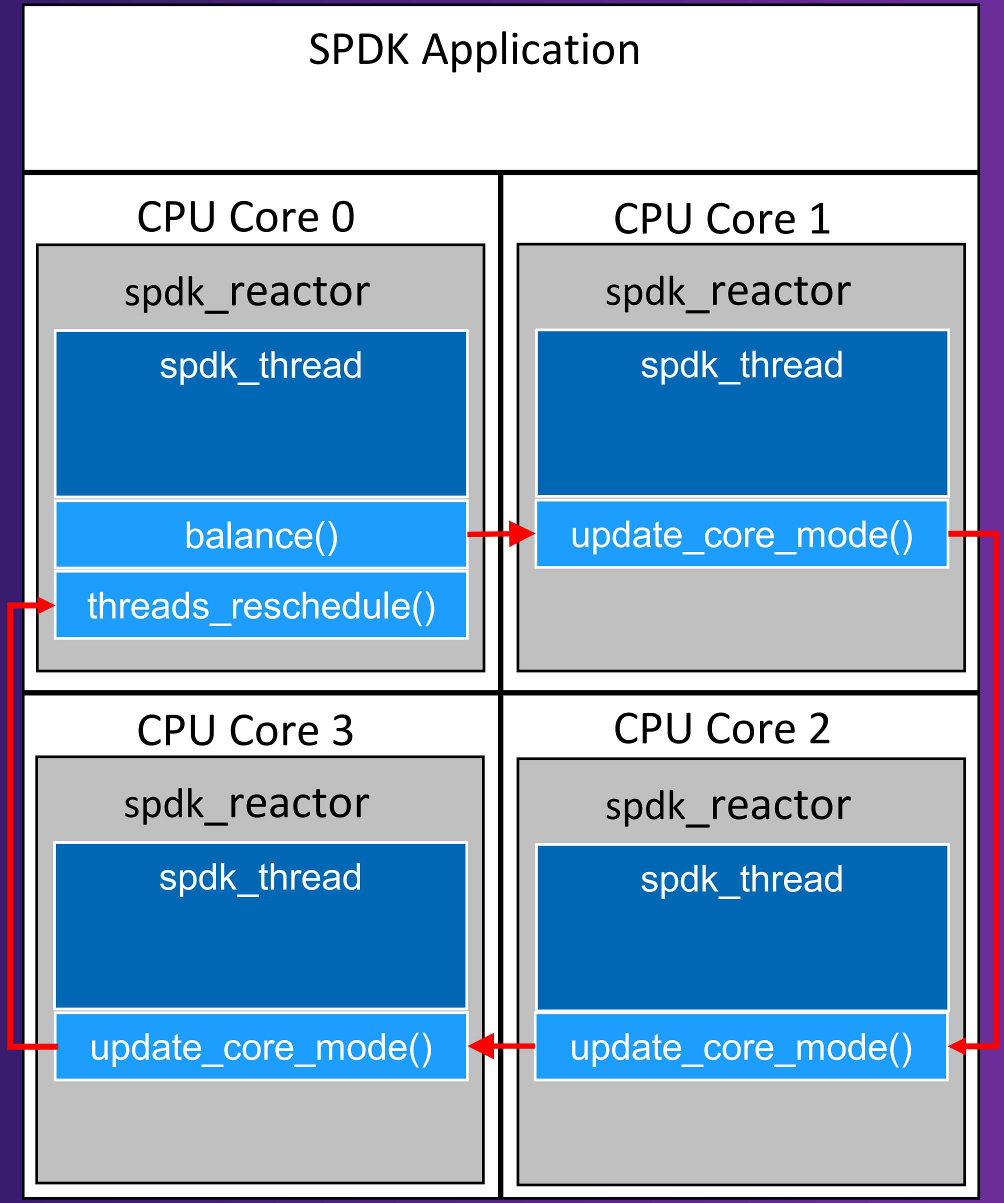
```
struct spdk_governor {
	const char *name;
	uint32_t (*get_core_curr_freq)(uint32_t lcore_id);
	int (*core_freq_up)(uint32_t lcore_id);
	int (*core_freq_down)(uint32_t lcore_id);
	int (*set_core_freq_max)(uint32_t lcore_id);
	int (*set_core_freq_min)(uint32_t lcore_id);
	int (*get_core_capabilities)(uint32_t lcore_id, struct spdk_governor_capabilities *capabilities);
	int (*init)(void);
	void (*deinit)(void);

	TAILQ_ENTRY(spdk_governor) link;
};
```

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE SDC21

Scheduler Actions

- 3a) update_core_mode()
  - Puts a core into sleep
- 3b) threads_reschedule()
  - Marks spdk_thread for move



 | ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE SDC 21

Dynamic Scheduler

- Implementation of a scheduler

```
$ ./scripts/rpc.py framework_set_scheduler dynamic -p 1000000
```

- Prioritizes performance over power saving
  - Eager spdk_thread expansion
- Consolidates spdk_threads on minimal set of cores
- Puts unused cores to sleep
- Reduces CPU frequency of the main core on low use

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE SDC 21

Performance Data

©2021 Storage Developer Conference ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE

Test Setup

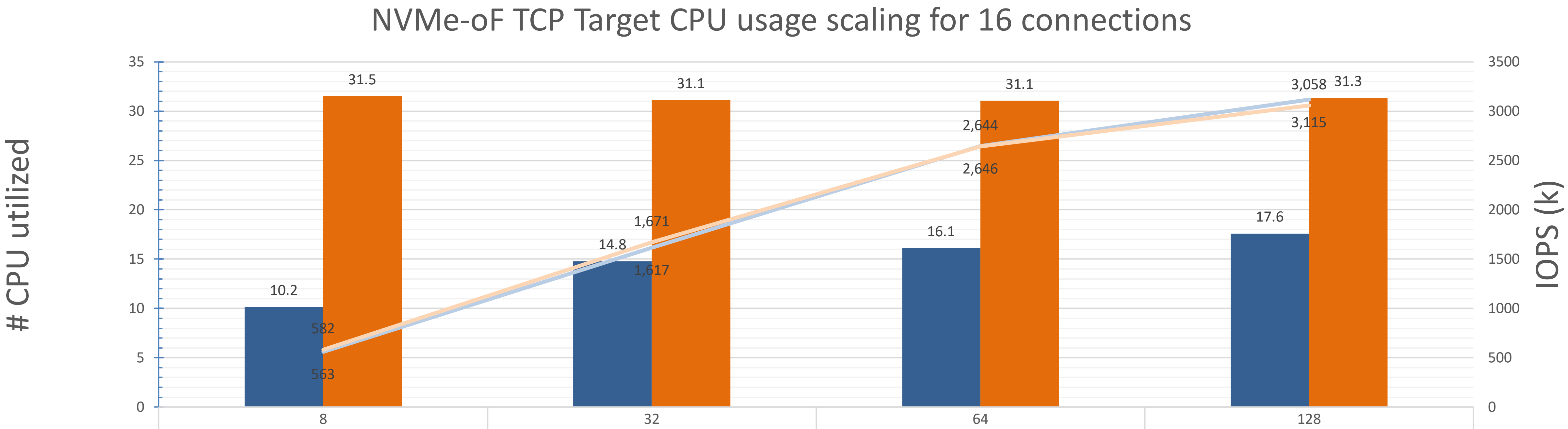
- SPDK NVMe-oF TCP Target
  - 30 CPU cores assigned for the whole application
- Two SPDK NVMe-oF TCP Initiators, each:
  - 4 CPU cores
  - 8 NVMe-oF subystems
- FIO 4k block size randread workload
  - Increasing Queue Depth
  - Increasing # of TCP connections with 'numjobs'

Test by Intel as of 9/15/2021.
Target Node: 1-node, 2x Intel® Xeon® Gold 6230 Processor, 20 cores HT On Turbo ON Total Memory 384GB (12 slots/32GB/2933 MHz), BIOS: 3.4 (ucode:0x5003003), Fedora 33, Linux Kernel 5.8.15-300, gcc 9.3.1 compiler, fio 3.19, SPDK 21.07 with 28ab38a, Storage: 16x Intel® SSD DC P4610 1.6TB, Network: 2x 100 GbE Mellanox ConnectX-5.
Host Nodes: 2-nodes, 2x Intel® Xeon® Gold 6252 Processor, 24 cores HT On Turbo ON Total Memory 192GB (6 slots/32GB/2933 MHz), BIOS: 3.4(ucode:0x5003003), Fedora 33, Linux Kernel 5.8.15-300, gcc 9.3.1 compiler, fio 3.19, SPDK 21.07 with 28ab38a, Network: 1x 100 GbE Mellanox ConnectX-5

https://ci.spdk.io/download/performance-reports/SPDK_tcp_perf_report_2104.pdf

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

Dynamic vs Static Scheduler



	8	32	64	128
# CPU Dynamic	10.2	14.8	16.1	17.6
# CPU Static	31.5	31.1	31.1	31.3
IOPS (k) Dynamic	562.6	1617.3	2646.1	3115.3
IOPS (k) Static	582.3	1671.2	2644.2	3058.1

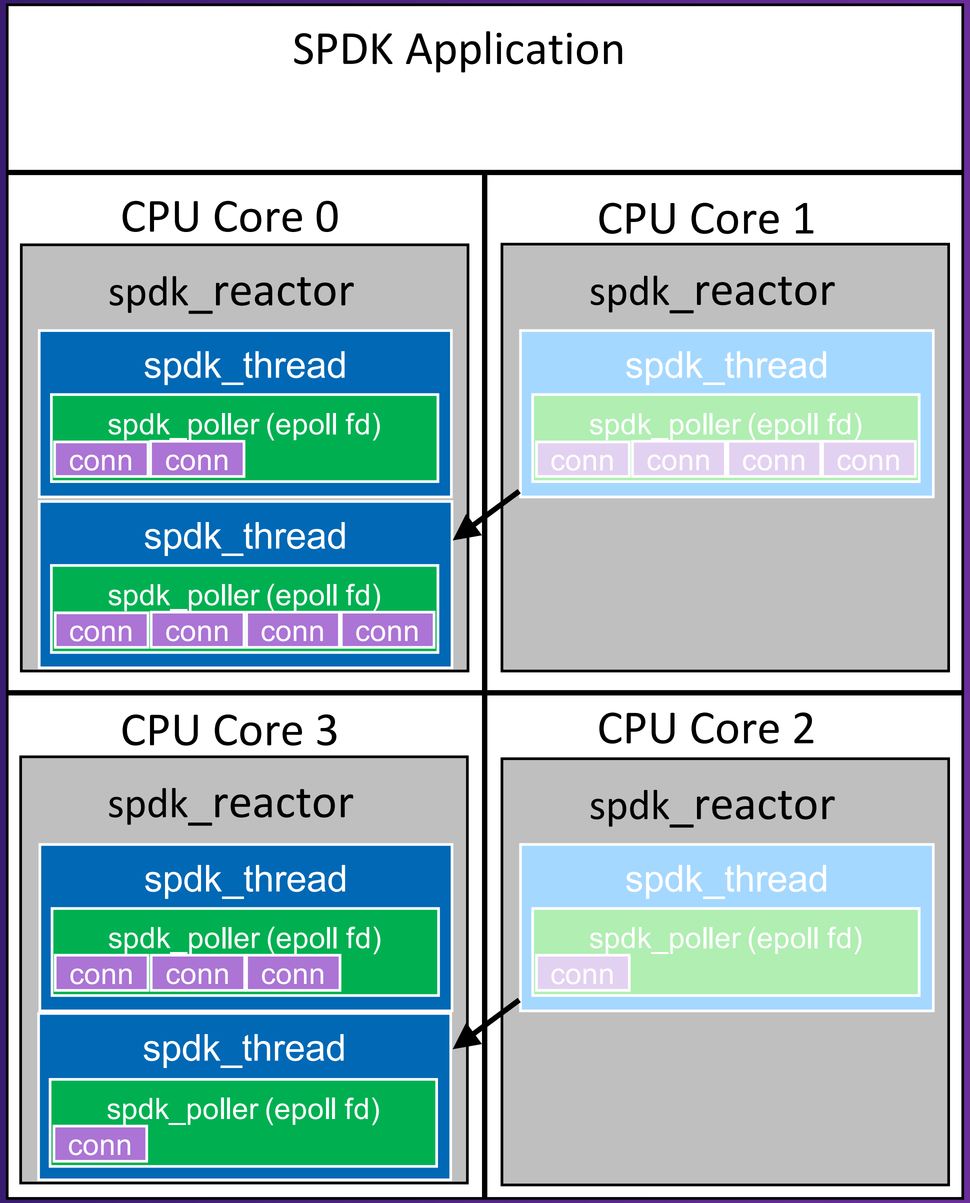
- 16 NVMe-oF subsystems
- 1 connection each

See configuration details – slide 20

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

NVMe-oF Poll Group

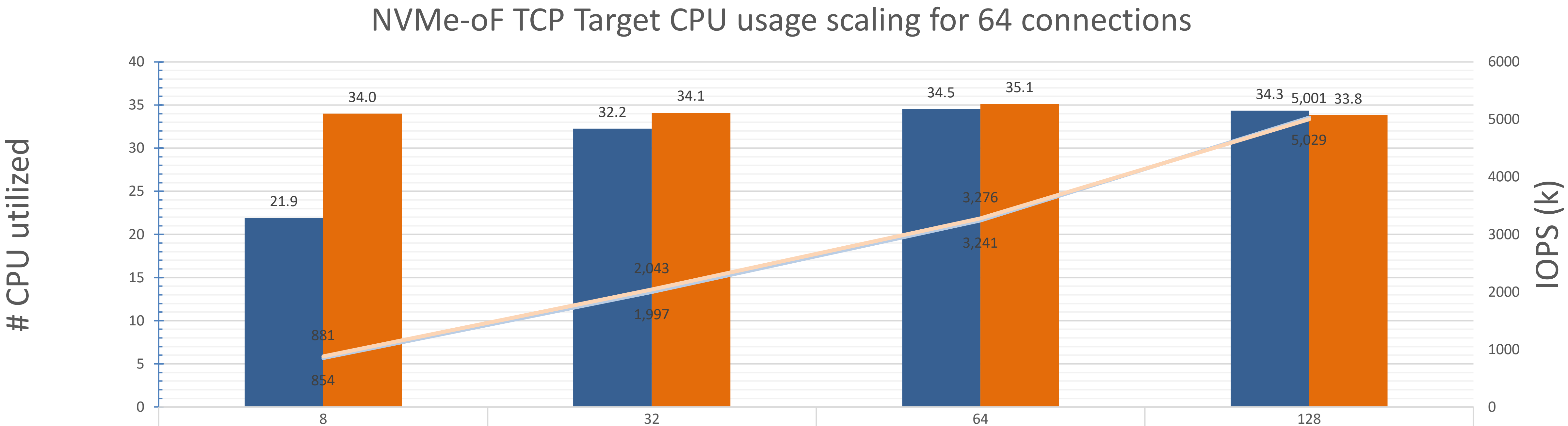
- spdk_poller polls an epoll fd
  - Group multiple TCP sockets
  - Round robin assignment of NVMe-oF qpairs
- No guarantees on balance across spdk_threads
  - Mix of active and idle qpairs
  - Qpairs can disconnect
- Initiator spreads load across qpairs



 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE SDC 21

Cost of NVMe-oF TCP Poll Groups



	8	32	64	128
# CPU Dynamic	21.9	32.2	34.5	34.3
# CPU Static	34.0	34.1	35.1	33.8
IOPS (k) Dynamic	854.1	1997.3	3240.7	5029.1
IOPS (k) Static	880.7	2043.3	3276.5	5000.7

Queue depth

- 16 NVMe-oF subsystems
- 4 connections each

See configuration details – slide 20

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

Summary and Next Steps

©2021 Storage Developer Conference ©. Intel Corporation. All Rights Reserved.

Summary

- Poll mode applications require special handling to save power and CPU cycles when idle
- SPDK event framework allows moving idle spdk_threads to put cores to sleep thus saving power
- Plugable scheduler framework is provided to define when spdk_threads should be moved
- Dynamic scheduler consolidates spdk_threads on minimal set of cores and puts remaining cores to sleep

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE SDC 21

Next steps

- Further improve the logic dynamic scheduler for spdk_thread placement
  - Give tweakable values to the user
- Address the cost of multiple poll groups on single core
- Scale CPU frequency of all cores
- Prioritize cores
  - Based on NUMA, hyperthreading and high frequency cores

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

Please take a moment to rate this session.

Your feedback is important to us.

 ©2021 Storage Networking Industry Association ©. Intel Corporation. All Rights Reserved.

STORAGE DEVELOPER CONFERENCE
SDC 21