



SPDK China Summit 2018

QOS RATE LIMITING ON SPDK BDEV & SPDK ECOSYSTEM RELATED TOOLS

Cao, Gang (gang.cao@intel.com)

Network Platforms Group, Intel Corporation

March 2018



Notices & Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit <http://www.intel.com/benchmarks>.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit <http://www.intel.com/benchmarks>.

Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at <http://www.intel.com/go/turbo>.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

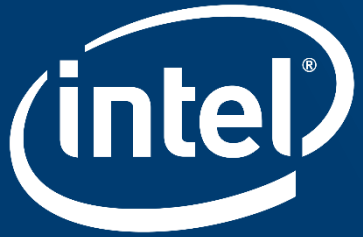
Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

© 2018 Intel Corporation.

Intel, the Intel logo, and Intel Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as property of others.



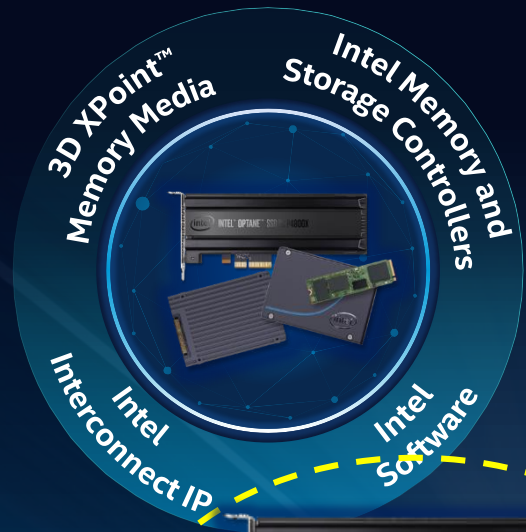


QOS RATE LIMITING ON SPDK BDEV



Avoid Noisy Neighbor

3D-Xpoint: Not Only Faster Flash or Cheaper Memory, But Also the Revolution of System Arch



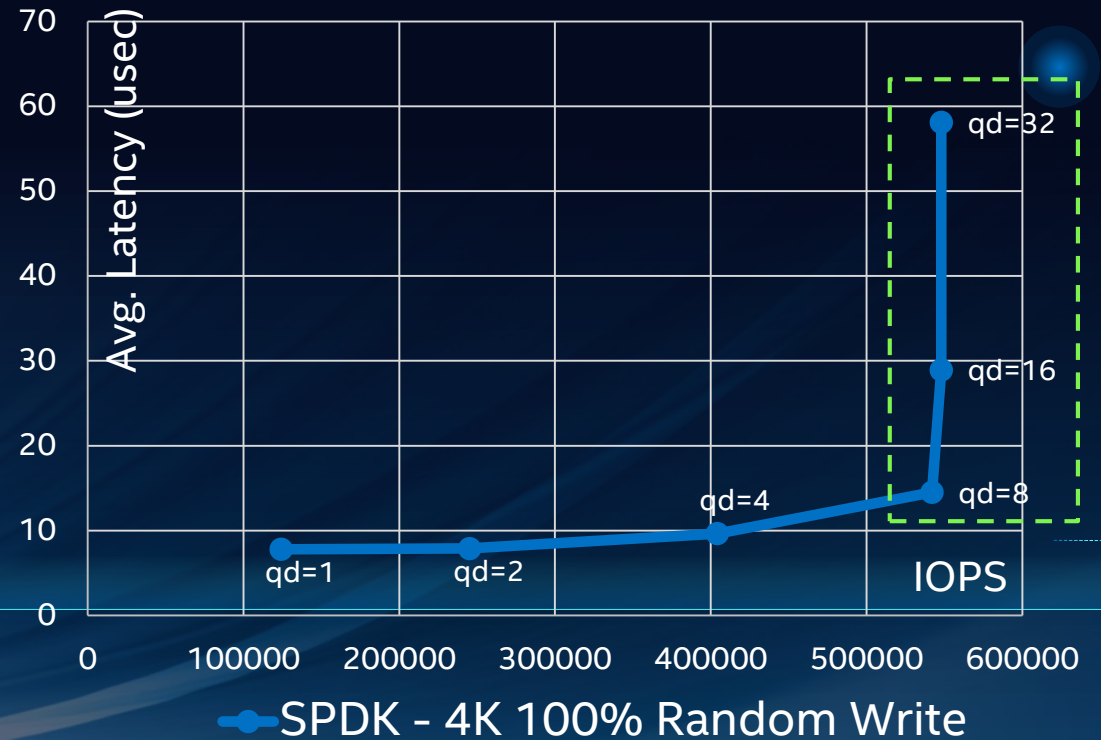
=

- ↑ Ultra-high Endurance
- ↓ Responsive Under Load
- ↑ Low Latency
- ↑ Predictably Fast Service
- ↑ QoS
- ↑ Breakthrough Performance
- ↑ IOPS



Share Fast Device

SPDK hit max IOPS at qd=8 on single core for single Optane



SPDK: Optane™ Local I/O Performance, Single Core
4KB 100% Writes, FIO-2.18, qd=1 to 32, numjobs=1, direct=1,

Expected QoS

- At a common layer
 - ✓ Below the application protocols (e.g., iSCSI, NVMe-oF, vhost...)
 - ✓ Above the real backend devices (e.g., NVMe SSD, AEP, Remote Connected Disk...)
- Strict Rate Limiting
 - ✓ IOPS
 - ✓ Bandwidth
- IO Prioritization
 - ✓ Data & Metadata
 - ✓ Read & Write

Extensible & Controllable

ARCHITECTURE

Released

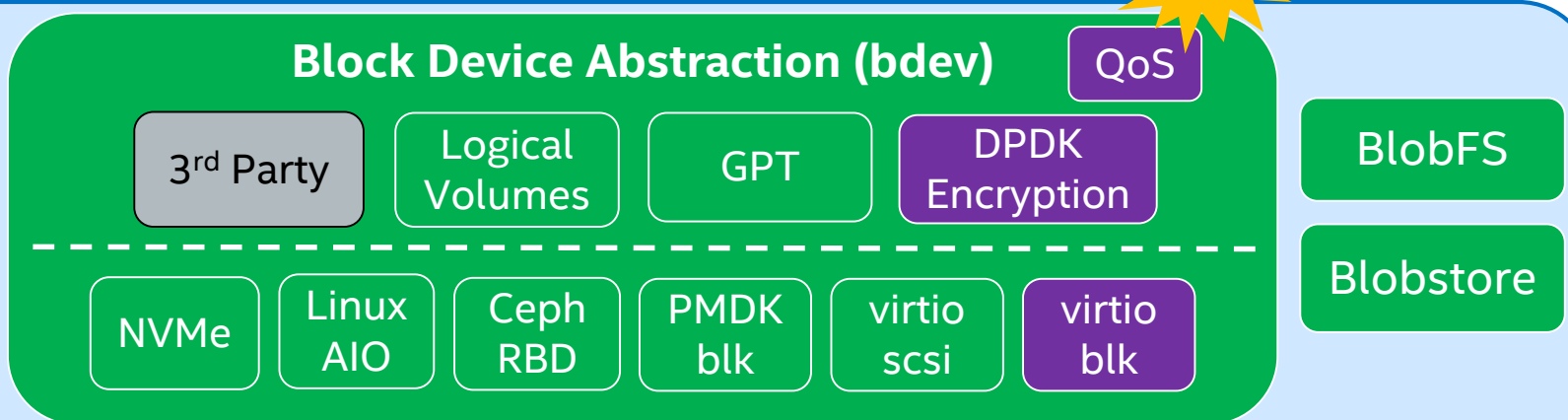
New release 18.01

1H'18

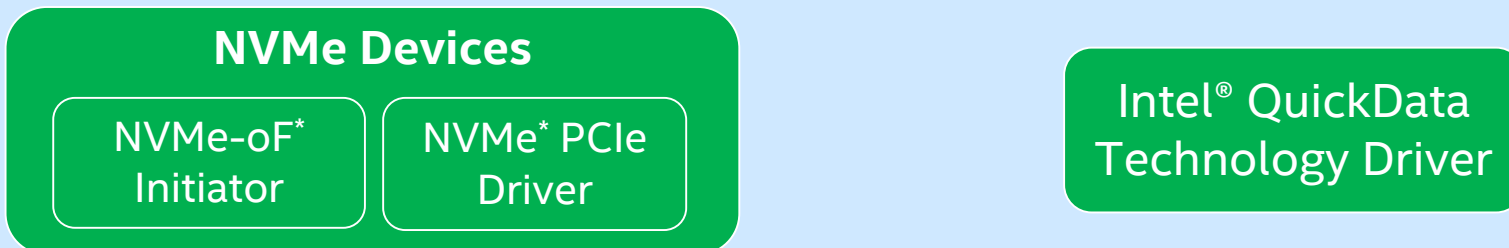
Storage Protocols



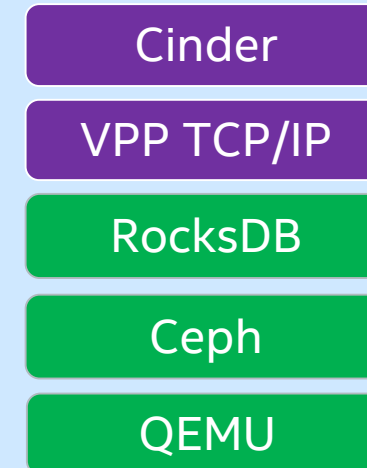
Storage Services



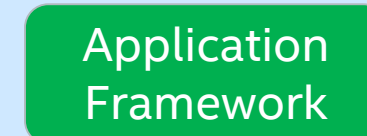
Drivers



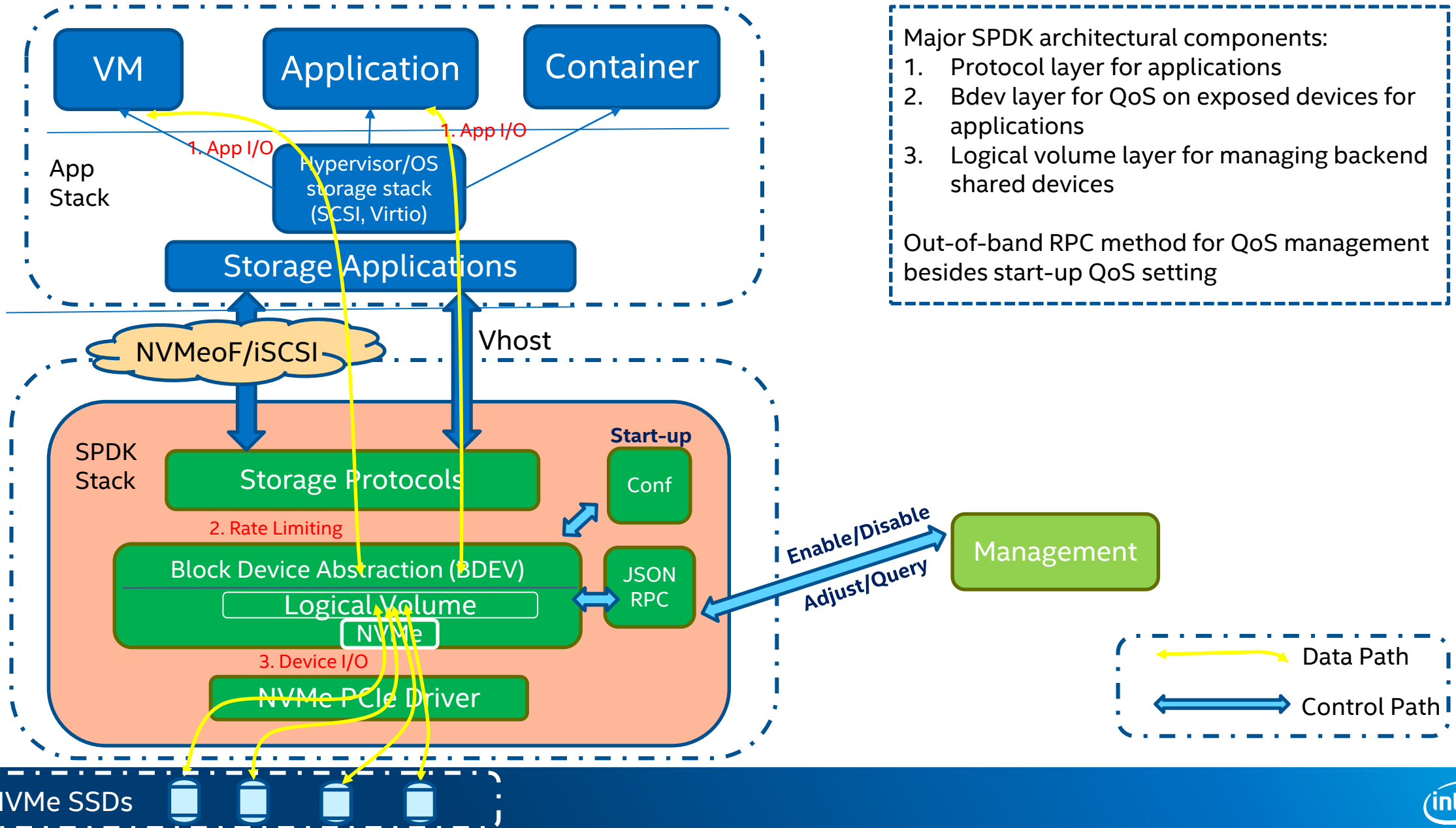
Integration



Core



Work Flow



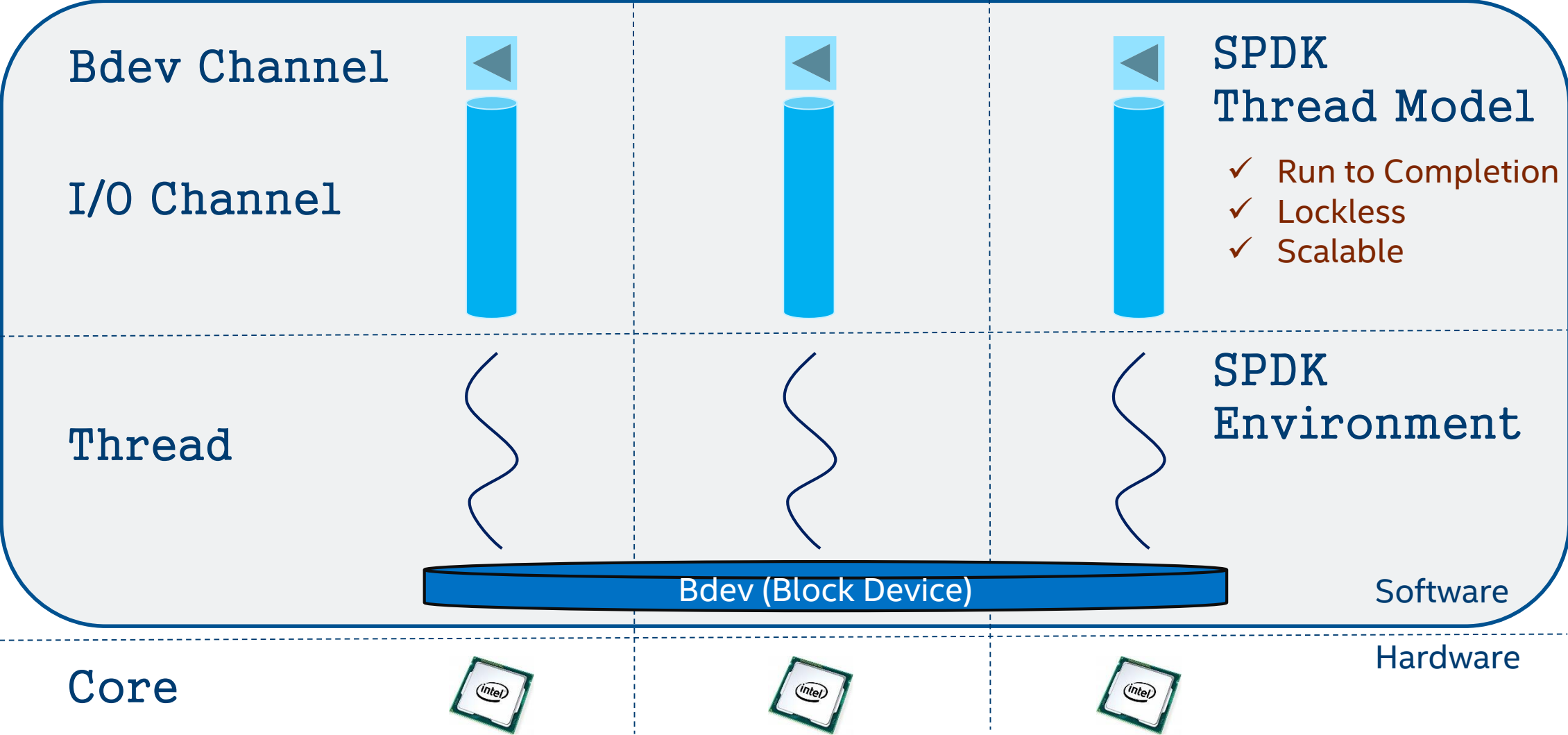
Key Points

1. Base on SPDK Application Framework
 - a. Asynchronous, Lockless, Event Driven

2. Base on Bdev Core Design
 - a. I/O Channel, Bdev Channel, Bdev Stats

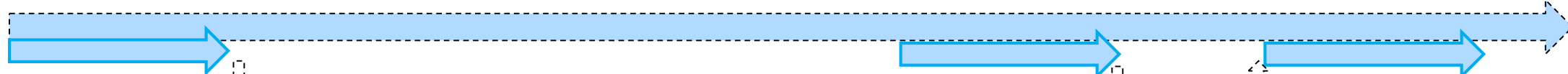
3. Algorithm Friendly
 - a. Clear workflow on I/O handling
 - b. Extensible for other algorithms

SPDK Application Framework



Resource Management

I/O Thread



1. Create I/O Channel (associated bdev channel)

4. Destroy I/O Channel

5. Destroy Completion

QoS Configured

2. Create QoS bdev channel & assign QoS thread

I/O Handling

Cleanup I/O for that I/O Channel

6. Unregister Poller and Destroy QoS Bdev Channel

QoS Thread



3. Create & Register QoS Poller

When All I/O Channels Destroyed

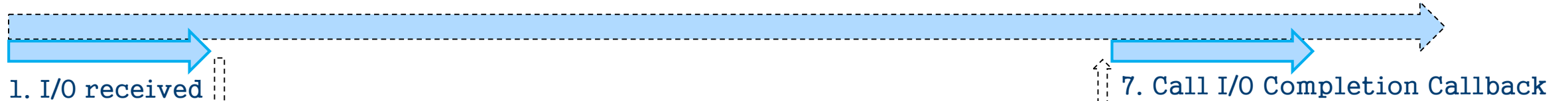
Cleanup I/O for all I/O Channels

RPC Disable QoS

RPC Enable QoS

Normal I/O Handling

I/O Thread



2. Sent I/O via Message

6. Sent I/O Completion via Message

Check Allowed I/O

QoS Thread

3. Queue I/O

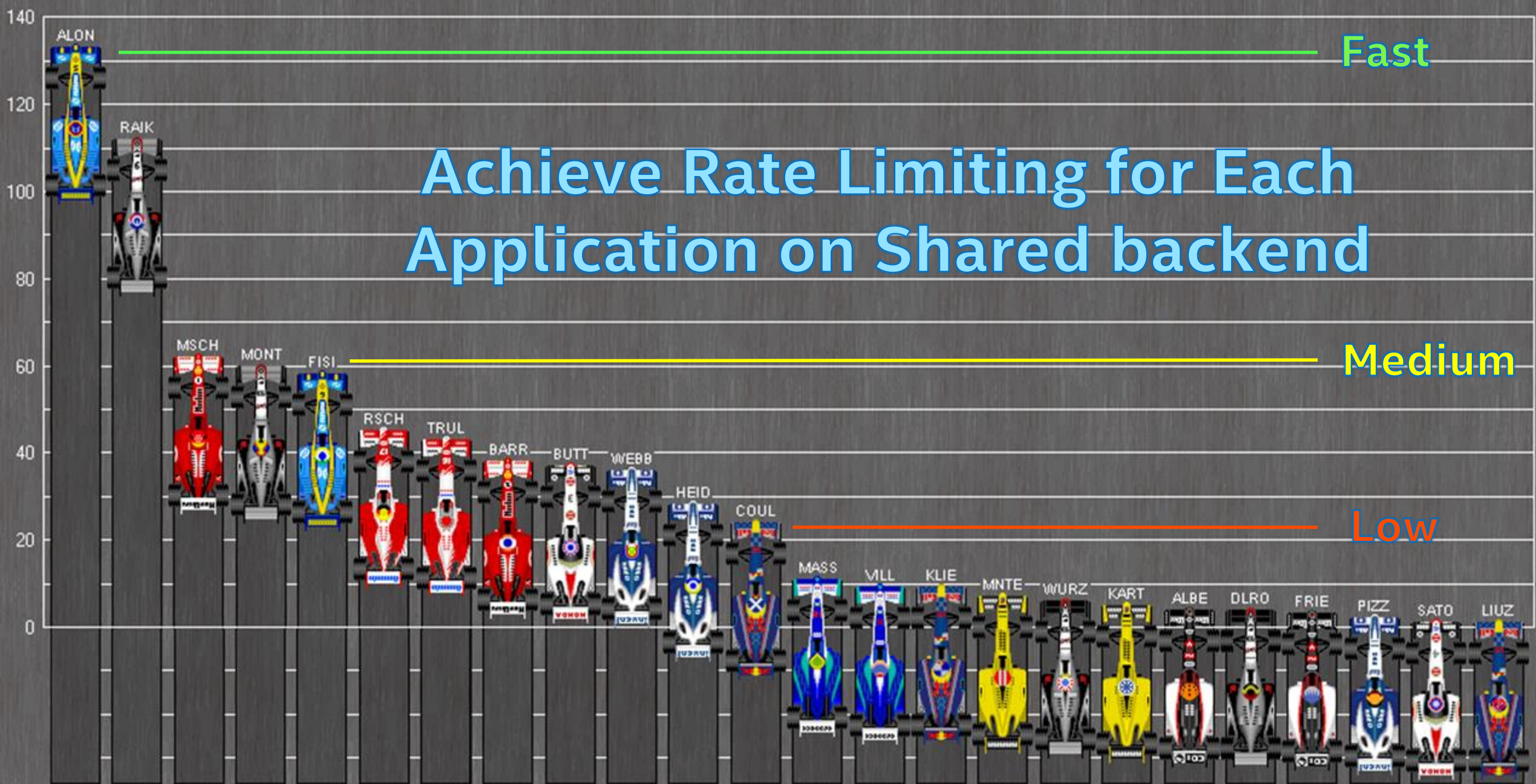
4. Send queued I/O down

5. Handle I/O completion



Periodically Run

Achieve Rate Limiting for Each Application on Shared backend



QoS Management

1. Startup configuration for application

✓ Example:

```
[QoS]
# QoS section defines limitation on performance metric like IOPS
#
# Format: Limit_IOPS Bdev_Name IOPS_Limit_Value
#
# Assign 20000 IOPS for the Malloc0 block device
Limit_IOPS Malloc0 20000
```

2. Runtime control through RPC (Enable / Disable / Adjust / Query / ...)

✓ Example:

```
#python ./scripts/rpc.py enable_bdev_qos Malloc0 -l 100000
```

Future Work

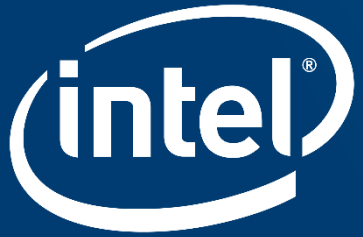
1. Rate Limiting

- ✓ Bandwidth
- ✓ ...

2. I/O Prioritization

- ✓ Data & Metadata
- ✓ Read & Write
- ✓ ...

3. Along with more functionalities from common SPDK bdev



SPDK China Summit 2018

SPDK ECOSYSTEM RELATED TOOLS

To build same experience with the kernel driver

- ✓ Nvme cli with vendor specific commands (e.g., Intel)
- ✓ FIO
- ✓ Iostat
- ✓ iSCSI / NVMe-oF initiator
- ✓ RPC
- ✓ VTune
- ✓

Support of nvme cli

<http://www.spdk.io/doc/nvme-cli.html>

1. Support of almost all existing nvme cli commands
2. Same usage experience with minor configuration change
3. Mostly used like “nvme list”, “nvme smart log”
4. Support Intel plugin like “internal-log”, “lat-stats” an so on

nvme intel internal-log

```
[root@node4 nvme-cli]# ./nvme intel internal-log 0000:84:00.0
Starting DPDK 17.11.0 initialization...
[ DPDK EAL parameters: nvme_cli -c 0x100 -m 512 --file-prefix=spdk1 --base-virtaddr=0x1000000000
--proc-type=auto ]
EAL: Detected 16 lcore(s)
EAL: Auto-detected process type: PRIMARY
EAL: Probing VFIO support...
EAL: PCI device 0000:08:00.0 on NUMA socket 0
EAL: probe driver: 8086:953 spdk_nvme
EAL: PCI device 0000:09:00.0 on NUMA socket 0
EAL: probe driver: 8086:2701 spdk_nvme
EAL: PCI device 0000:84:00.0 on NUMA socket 1
EAL: probe driver: 8086:953 spdk_nvme
Log major:1 minor:1 header:1024 size:39936
Successfully wrote log to Nlog_CVFT516600W51P6DGN.bin
```

nvme list

```
[root@node4 nvme-cli]# ./nvme list
Starting DPDK 17.11.0 initialization...
[ DPDK EAL parameters: nvme_cli -c 0x100 -m 512 --file-prefix=spdk1 --base-virtaddr=0x1000000000 --proc-type=auto ]
EAL: Detected 16 lcore(s)
EAL: Auto-detected process type: PRIMARY
EAL: Probing VFIO support...
EAL: PCI device 0000:08:00.0 on NUMA socket 0
EAL: probe driver: 8086:953 spdk_nvme
EAL: PCI device 0000:09:00.0 on NUMA socket 0
EAL: probe driver: 8086:2701 spdk_nvme
EAL: PCI device 0000:84:00.0 on NUMA socket 1
EAL: probe driver: 8086:953 spdk_nvme
```

Node	SN	Model	Namespace Usage	Format	FW Rev
0000:09:00.0 E2010101	FUMB6383004V140AGN	INTEL SSDPED1D140GA	1	140.04 GB / 140.04 GB	512 B + 0 B
0000:08:00.0 8DV10131	CVFT5341005D800CGN	INTEL SSDPEDMD800G4	1	800.17 GB / 800.17 GB	512 B + 0 B
0000:84:00.0 8DV10131	CVFT516600W51P6DGN	INTEL SSDPEDMD016T4	1	1.60 TB / 1.60 TB	512 B + 0 B

nvme smart-log

```
[root@node4 nvme-cli]# ./nvme smart-log 0000:84:00.0
Starting DPDK 17.11.0 initialization...
[ DPDK EAL parameters: nvme_cli -c 0x100 -m 512 --file-prefix=spdk1 --base-virtaddr=0x1000000000 --proc-type=auto ]
.....
Smart Log for NVME device:0000:84:00.0 namespace-id:ffffff
critical_warning      : 0
temperature          : 26 C
available_spare       : 100%
available_spare_threshold : 10%
percentage_used       : 0%
data_units_read       : 623,253
data_units_written    : 1,783,614
host_read_commands    : 555,946,160
host_write_commands   : 457,595,166
controller_busy_time  : 0
power_cycles          : 42
power_on_hours        : 3,876
unsafe_shutdowns      : 0
media_errors          : 126
num_err_log_entries   : 126
Warning Temperature Time : 0
Critical Composite Temperature Time : 0
Thermal Management T1 Trans Count : 0
Thermal Management T2 Trans Count : 0
Thermal Management T1 Total Time : 0
Thermal Management T2 Total Time : 0
```

FIO

https://github.com/spdk/spdk/blob/master/examples/nvme/fio_plugin/README.md

1. Leverage FIO framework for the IO testing
2. Also work for multiple Jobs (multiple threads)
3. A good comparison method for FIO IO testing on kernel driver

FIO plugin sample configuration

https://github.com/spdk/spdk/blob/master/examples/nvme/fio_plugin/example_config.fio

[global]

ioengine=spdk

thread=1

group_reporting=1

direct=1

verify=0

time_based=1

ramp_time=0

runtime=2

iodepth=128

rw=randrw

bs=4k

[test]

numjobs=1

```
/usr/src/fio/fio /home/sys_sgs/build_pool/agent/repo/examples/nvme/fio_plugin/example_config.fio
'--filename=trtype=PCIe traddr=0000.00.04.0 ns=1'
test: (g=0): rw=randrw, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=spdk, iodepth=128
fio-2.21-63-g658f-dirty
Starting 1 thread
Starting DPDK 17.11.0 initialization...
[ DPDK EAL parameters: fio -c 0x1 -m 512 --file-prefix=spdk_pid19188 ]
EAL: Detected 16 lcore(s)
EAL: WARNING: cpu flags constant_tsc=yes nonstop_tsc=no -> using unreliable clock cycles !
EAL: PCI device 0000:00:04.0 on NUMA socket -1
EAL:   Invalid NUMA socket, default to 0
EAL:   probe driver: 8086:5845 spdk_nvme

test: (groupid=0, jobs=1): err= 0: pid=19191: Fri Dec 15 02:04:18 2017
read: IOPS=17.6k, BW=68.7MiB/s (72.1MB/s) (138MiB/2001msec)
  slat (usec): min=3, max=123, avg=11.48, stdev= 6.08
  clat (usec): min=51, max=5551, avg=3608.29, stdev=459.43
    lat (usec): min=59, max=5567, avg=3619.77, stdev=461.33
  clat percentiles (usec):
    | 1.00th=[ 2868], 5.00th=[ 2933], 10.00th=[ 3163], 20.00th=[ 3261],
    | 30.00th=[ 3326], 40.00th=[ 3425], 50.00th=[ 3490], 60.00th=[ 3621],
    | 70.00th=[ 3818], 80.00th=[ 3982], 90.00th=[ 4228], 95.00th=[ 4490],
    | 99.00th=[ 5014], 99.50th=[ 5211], 99.90th=[ 5473], 99.95th=[ 5473],
    | 99.99th=[ 5538]
  bw (  KiB/s): min=64920, max=75032, per=0.10%, avg=69349.33, stdev=5171.20, samples= 3
  iops        : min=16230, max=18758, avg=17337.33, stdev=1292.80, samples= 3
```

lostat

Same usage as the iostat tool with kernel driver

SPDK supported iostat based on the SPDK exposed devices (bdev)

```
[root@waikikibeach27 sysstat]# ./iostat
Starting DPDK 17.11.0 initialization...
[ DPDK EAL parameters: spdk_iostat -c 0x1 --file-prefix=spdk1 --base-virtaddr=0x100000000 --proc-type=auto ]
EAL: Detected 44 lcore(s)
EAL: Auto-detected process type: SECONDARY
EAL: Probing VFIO support...
EAL: WARNING: Master core has no memory on local socket!
Linux 3.10.0-514.26.2.el7.x86_64 (waikikibeach27)      12/16/2017      _x86_64_      (44 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.41    0.00    0.08    0.05    0.00   99.45

Device            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                 3.48         128.28         21.62    1159269    195382
Nvme0n1            1826.37      15988.08          0.00   144481912         0
```


iSCSI Initiator

```
iscsiadm -m discovery -t sendtargets -p 127.0.0.1:3260
```

```
iscsiadm -m node --login -p 127.0.0.1:3260
```

```
iscsiadm -m session -P 3
```

```
iscsiadm -m node --logout
```

NVMe-oF Initiator

```
nvme discover -t rdma -a 10.0.2.15 -s 4420
```

```
nvme connect -t rdma -n nqn.2016-06.io.spdk:cnode1 -a 10.0.2.15 -s 4420
```

```
nvme disconnect -n nqn.2016-06.io.spdk:cnode1
```

RPC

```
python /home/sys_sgsu/build_pool/agent/repo/scripts/rpc.py add_portal_group 1  
127.0.0.1:3260
```

```
python /home/sys_sgsu/build_pool/agent/repo/scripts/rpc.py add_initiator_group 2 ANY  
127.0.0.1/32
```

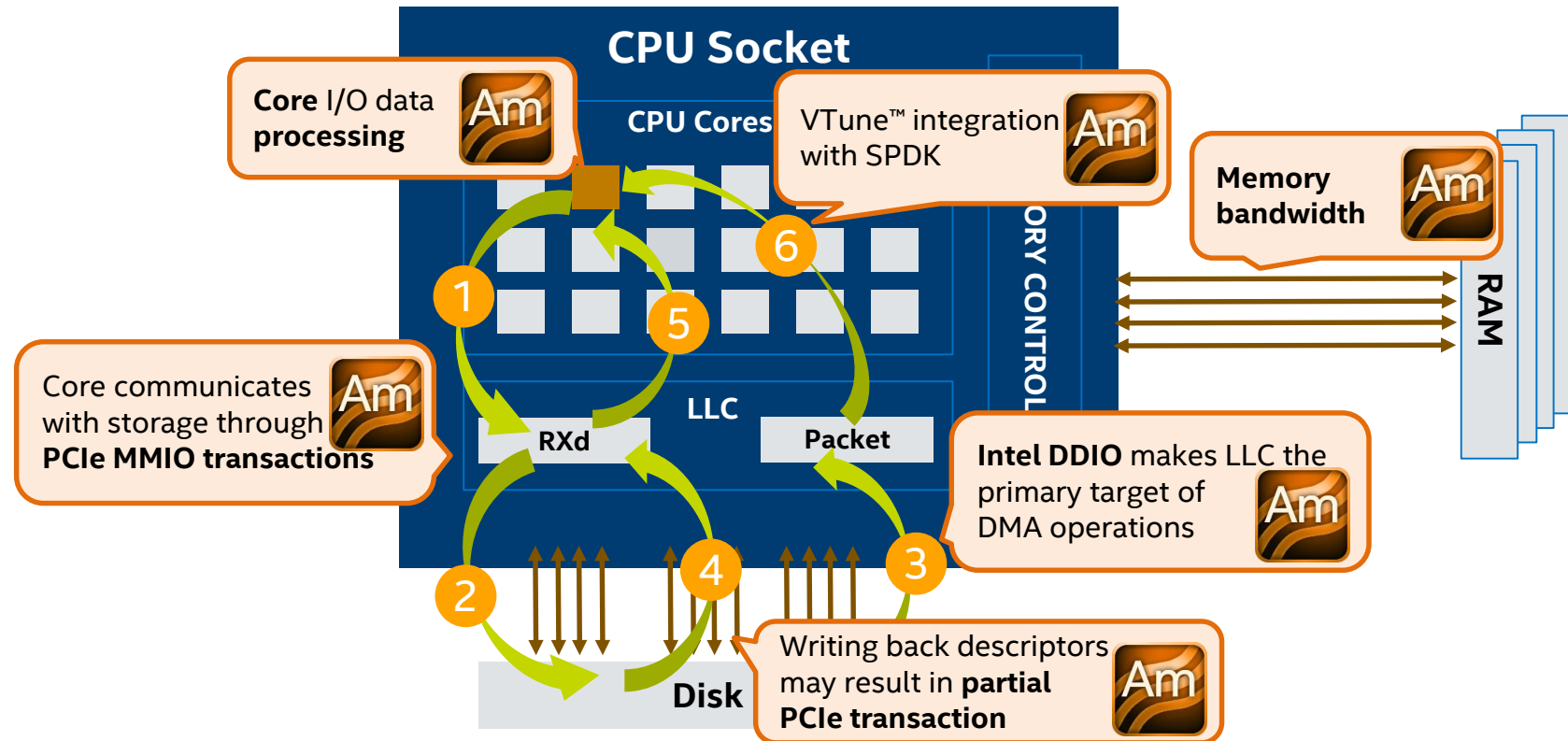
```
python /home/sys_sgsu/build_pool/agent/repo/scripts/rpc.py construct_malloc_bdev 64 4096
```

```
python /home/sys_sgsu/build_pool/agent/repo/scripts/rpc.py construct_target_node Target3
```

```
iscsiadm -m discovery -t sendtargets -p 127.0.0.1:3260
```

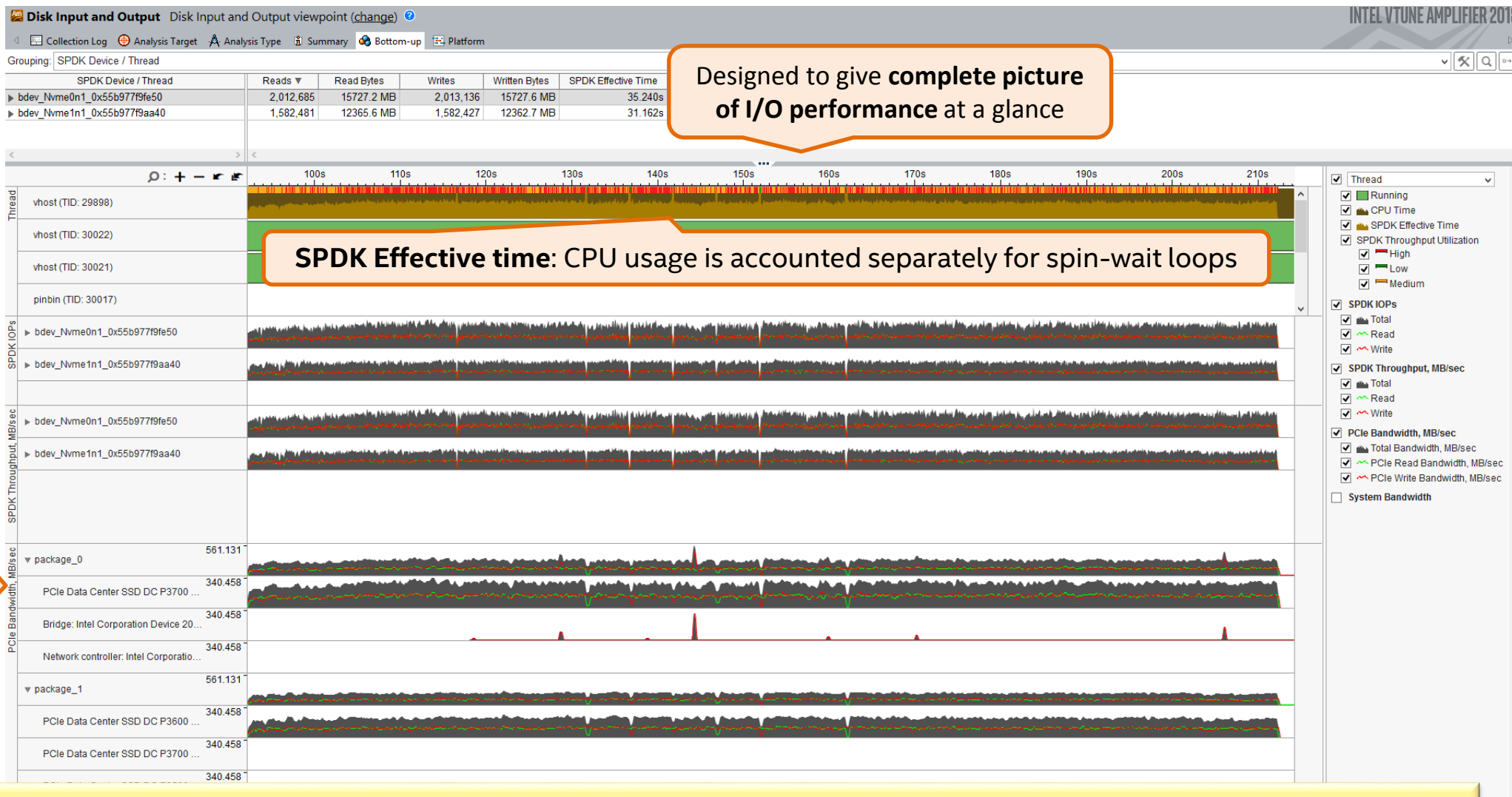
INTEL® VTUNE™ AMPLIFIER FOR PERFORMANCE PROFILING

Telemetry points



VTune™ enables performance visibility of the platform at all layers: hardware with all its key components and software acceleration engines

SPDK PERFORMANCE ANALYSIS WITH INTEL® VTUNE™ AMPLIFIER



<https://software.intel.com/en-us/intel-vtune-amplifier-xe>

Analyzing results – Platform view



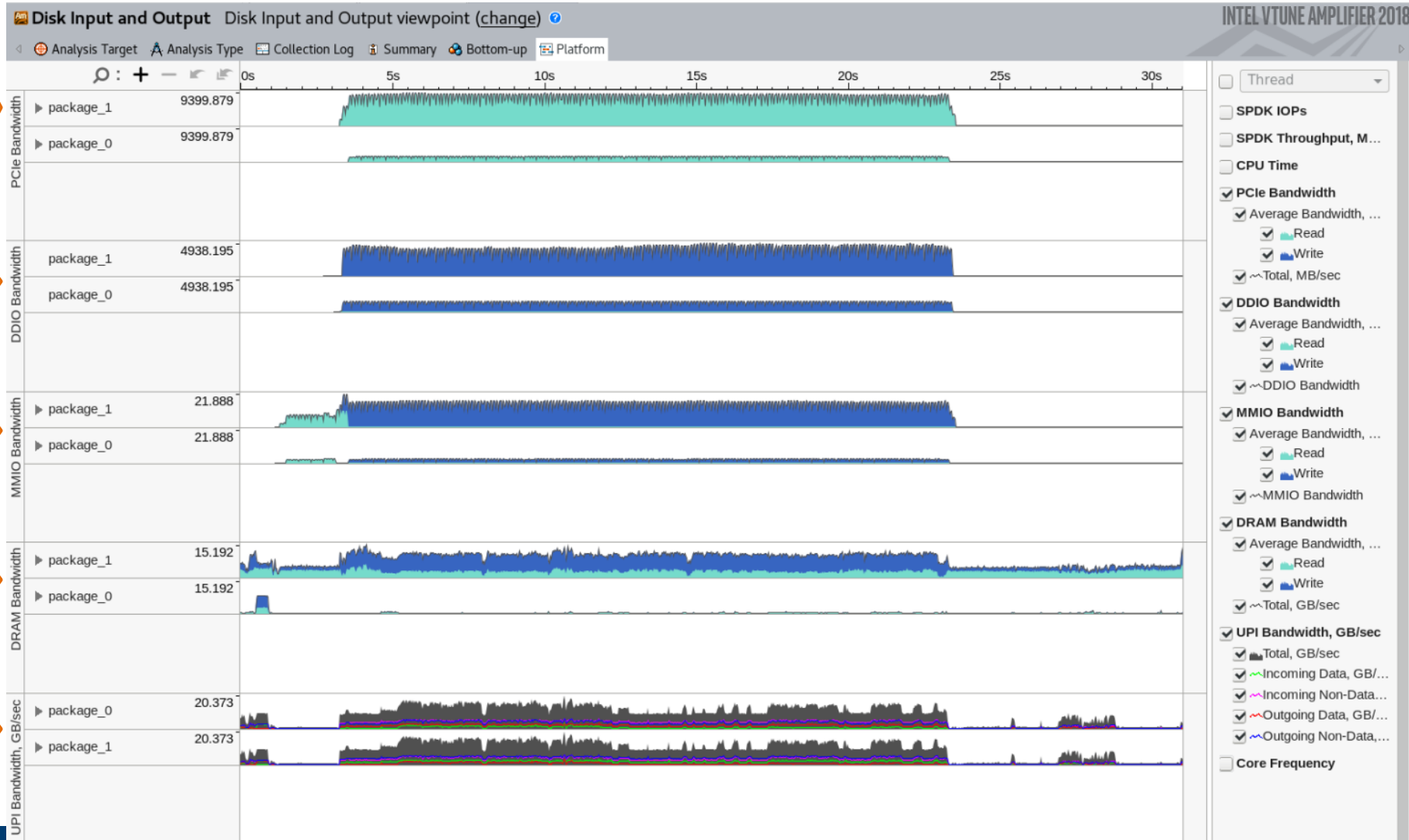
PCIe bandwidth with traffic breakdown per physical device

Intel DDIO misses resulted in write back to RAM

MMIO traffic. Avoid Reads and control Writes

DRAM bandwidth

Socket interconnect traffic



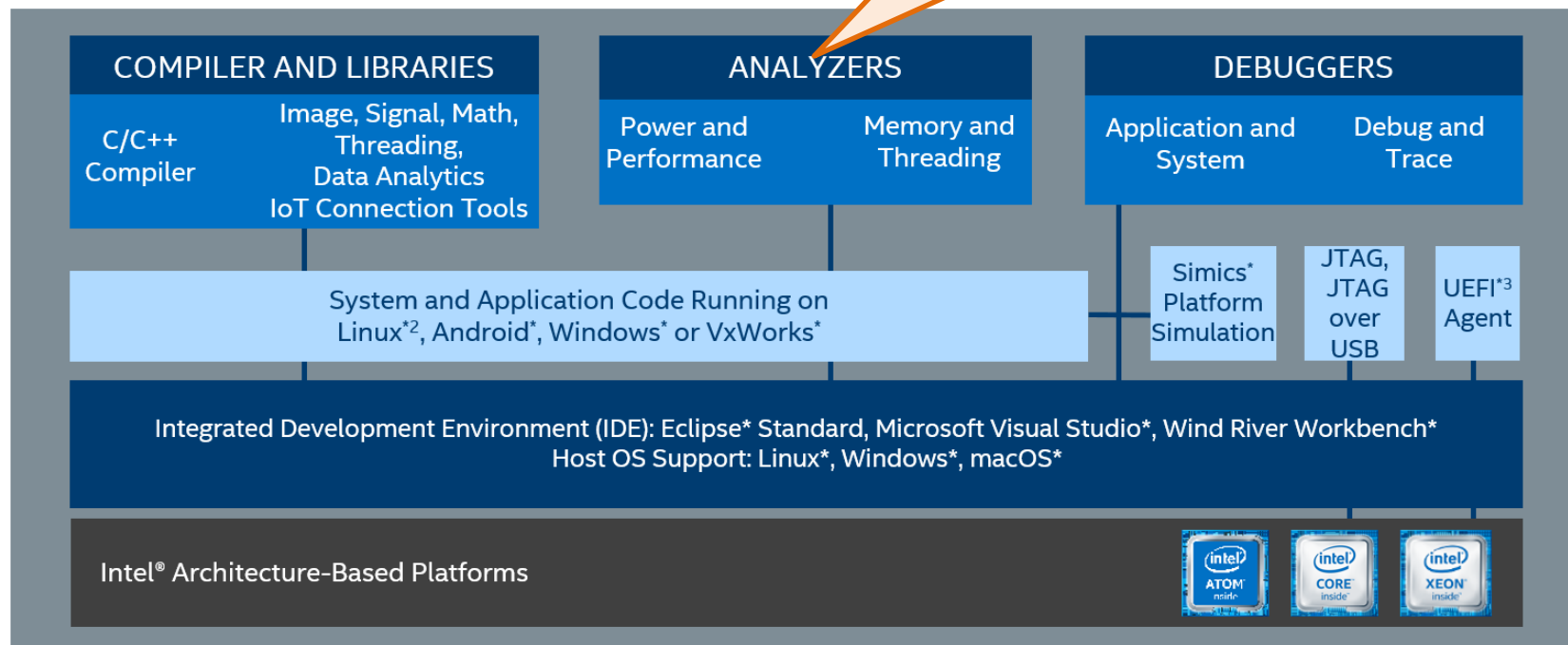
Intel® System Studio 2018

A cross-platform development tool suite

Intel® VTune™
Amplifier included

**Free Renewable
Commercial
License**

(Ultimate Edition)
includes public community
forum support



Registration & Download: <http://software.intel.com/en-us/system-studio>

Contact: Tong.Gu@intel.com Cellphone: 13910139956 **for Free Enabling License**

