2015中国DPDK开发者大会
China DPDK Summit 2015

Presented By:

中国移动 China Mobile | 中国电信 CHINA TELECOM | HUAWEI | intel | ZTE

Practices for Building Core/Efficient Applications

Liang Cunming

2015.04.21

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm%20 Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Celeron, Intel, Intel logo, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel SpeedStep, Intel XScale, Itanium, Pentium, Pentium Inside, VTune, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel® Active Management Technology requires the platform to have an Intel® AMT-enabled chipset, network hardware and software, as well as connection with a power source and a corporate network connection. With regard to notebooks, Intel AMT may not be available or certain capabilities may be limited over a host OS-based VPN or when connecting wirelessly, on battery power, sleeping, hibernating or powered off. For more information, see http://www.intel.com/technology/iamt.

64-bit computing on Intel architecture requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel® 64 architecture. Performance will vary depending on your hardware and software configurations. Consult with your system vendor for more information.

No computer system can provide absolute security under all conditions. Intel® Trusted Execution Technology is a security technology under development by Intel and requires for operation a computer system with Intel® Virtualization Technology, an Intel Trusted Execution Technology-enabled processor, chipset, BIOS, Authenticated Code Modules, and an Intel or other compatible measured virtual machine monitor. In addition, Intel Trusted Execution Technology requires the system to contain a TPMv1.2 as defined by the Trusted Computing Group and specific software for some uses. See http://www.intel.com/technology/security/ for more information.

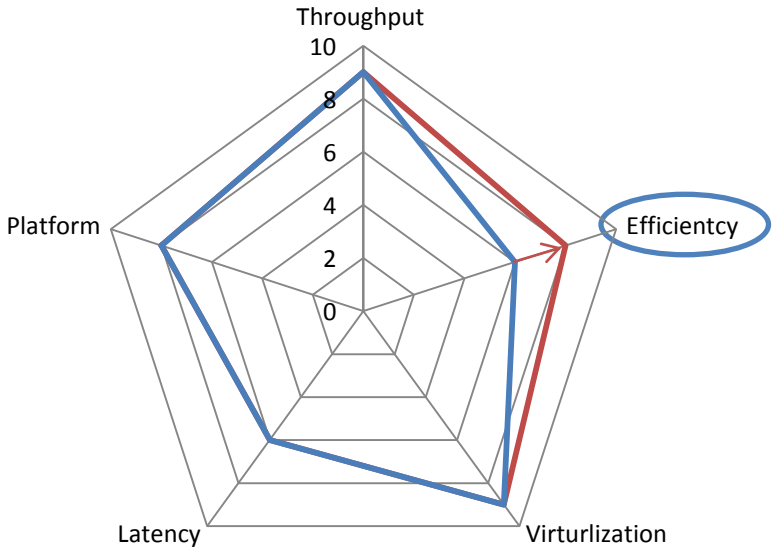
†Hyper-Threading Technology (HT Technology) requires a computer system with an Intel® Pentium® 4 Processor supporting HT Technology and an HT Technology-enabled chipset, BIOS, and operating system. Performance will vary depending on the specific hardware and software you use. See www.intel.com/products/ht/hyperthreading_more.htm for more information including details on which processors support HT Technology.

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM) and, for some uses, certain platform software enabled for it. Functionality, performance or other benefits will vary depending on hardware and software configurations and may require a BIOS update. Software applications may not be compatible with all operating systems. Please check with your application vendor.

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

Other vendors are listed by Intel as a convenience to Intel's general customer base, but Intel does not make any representations or warranties whatsoever regarding quality, reliability, functionality, or compatibility of these devices. This list and/or these devices may be subject to change without notice.

What's this talk is about



5 dimension assessment or user experience

The talk focus on efficiency

CPU Efficiency

1. Reducing stalled cycles
2. Managing idle loop
3. Leverage HW offload

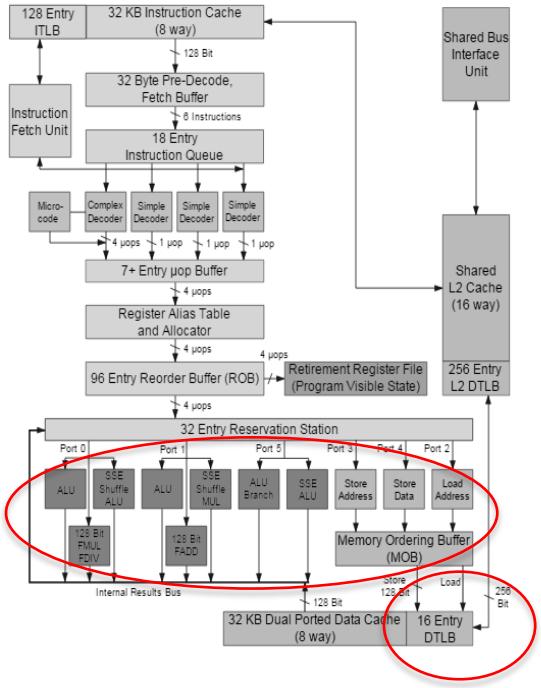
Practice Sharing

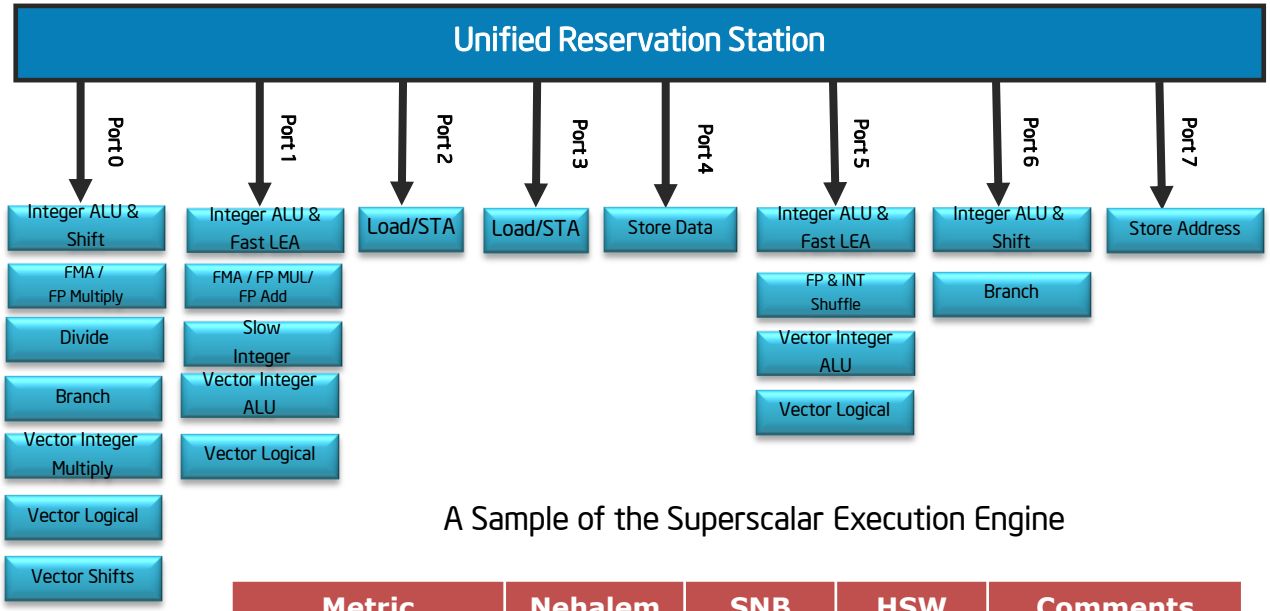
- SIMD in Packet IO
- AVX2 memcpy
- Dynamic frequency adaption
- Preemptive task switch
- Interrupt Packet IO
- Gain from csum offload

Reducing stalled cycles

- Instruction parallel
- Improving cache bandwidth
- Hide cache latency

Instruction Parallel and Cache BW





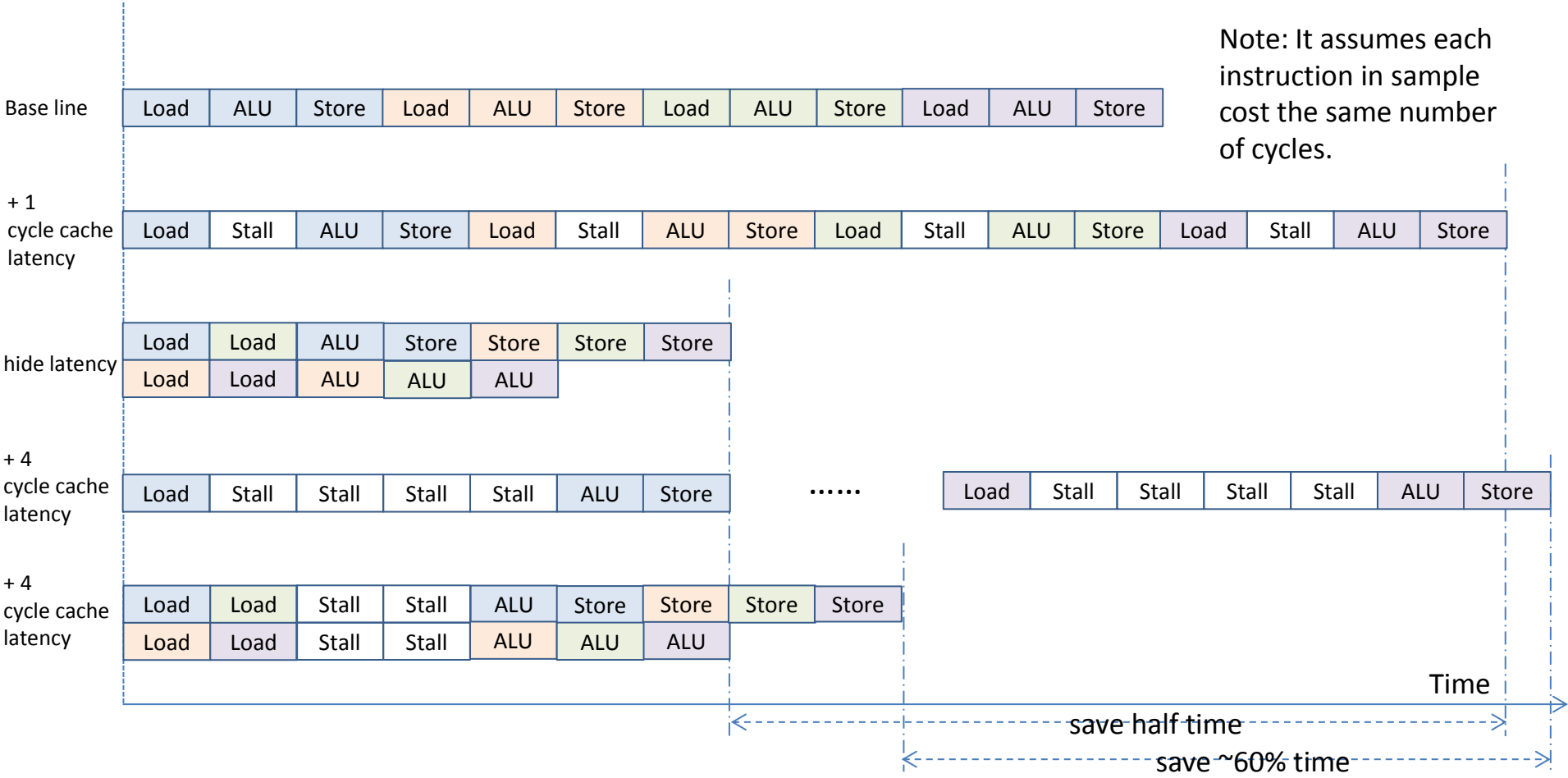
A Sample of the Superscalar Execution Engine

From Nehalem, SNB to HSW L1 Cache BW was greatly improved – How to expose this capability in DPDK ?

Metric	Nehalem	SNB	HSW	Comments
Instruction Cache	32K	32K	32K	
L1 Data Cache (DCU)	32K	32K	32K	
Hit Latency (cycle)	4/5/7	4/5/7	4/5/7	No index / nominal / non-flat seg
Bandwidth (bytes/cycle)	16+16	32+16	64+32	2 loads + 1 store
L2 Unified Cache (MLC)	256K	256K	256K	
Hit Latency (cycle)	10	12	12	Nominal load
BW (bytes/cycle)	32	32	64	HSW doubled MLC hit BW

https://www-ssl.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html

Hide Cache Latency



hide cache latency by bulk process

Sounds great in theory but how to realize this performance ?

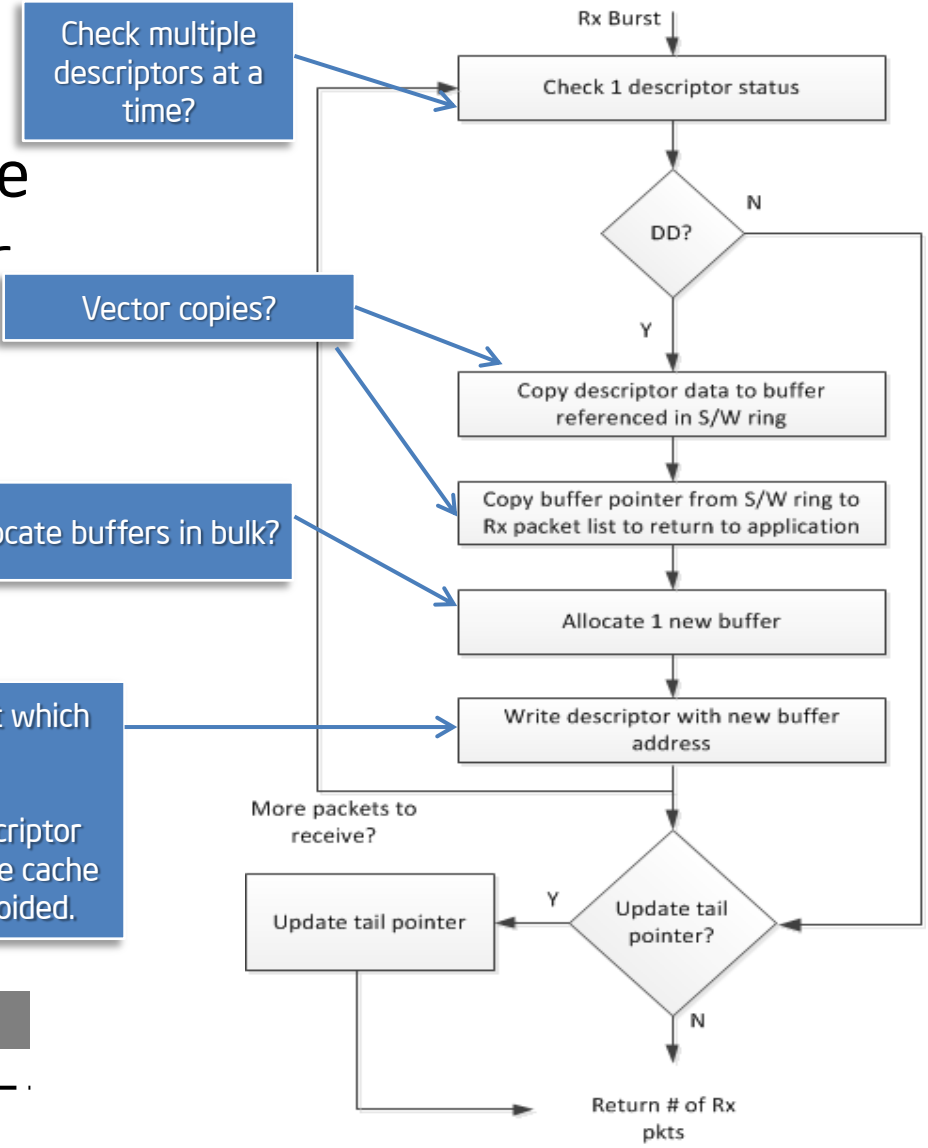
Practice: vector packet IO (1)

- One big while loop
- Things happen one at a time
  - Except amortizing tail pointer
- Straightforward implementation
- Linear execution

Check multiple descriptors at a time?

Vector copies?

Allocate buffers in bulk?

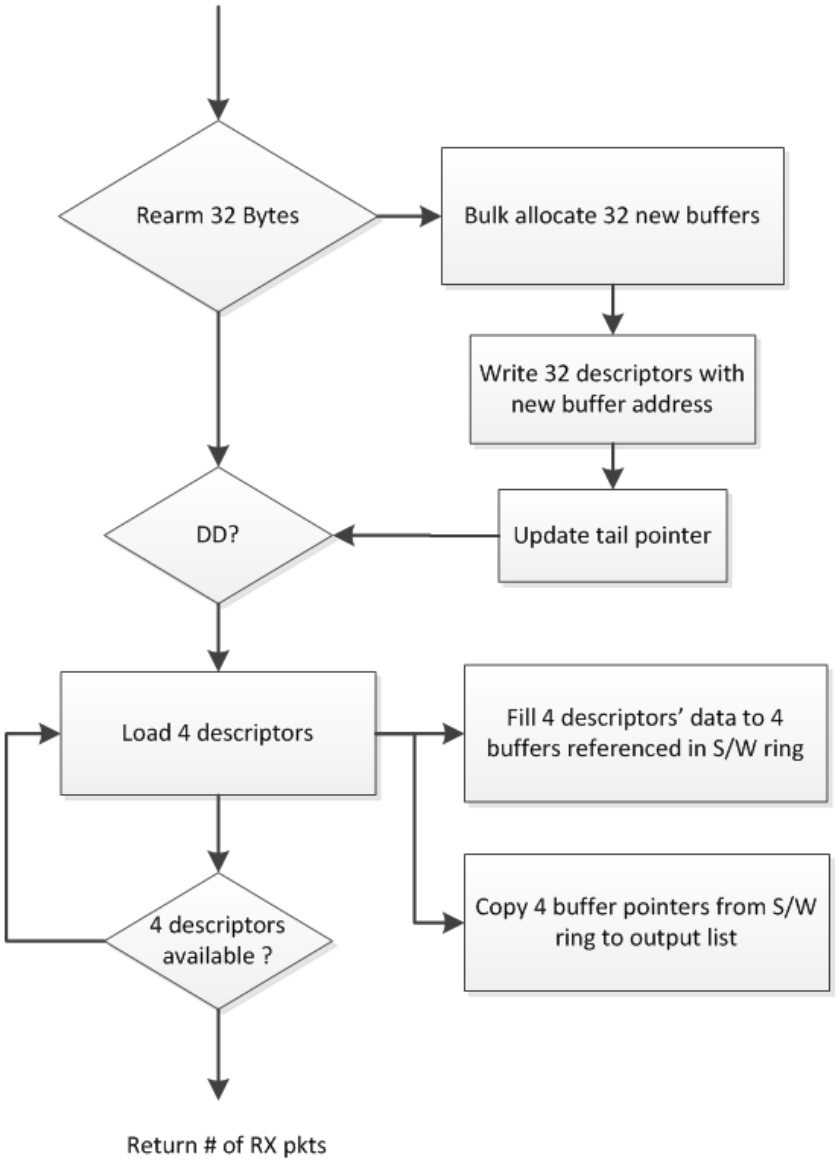


What happens when you poll much faster than the rate at which packets are coming in?

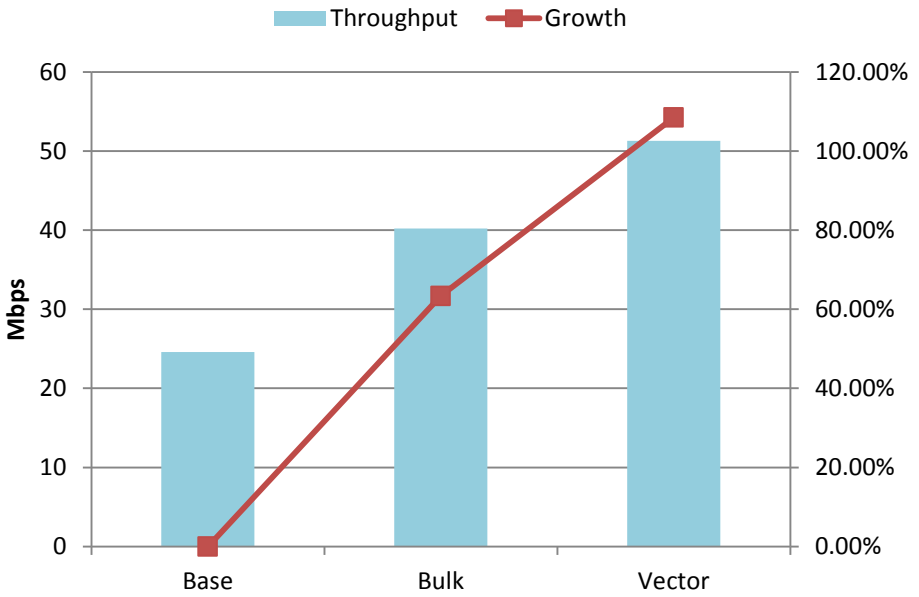
Every received packed will result in modification of a descriptor cache line (to write new buffer address)... likely in the same cache line that the NIC is reading. These conflicts should be avoided.

Practice: vector packet IO (2)

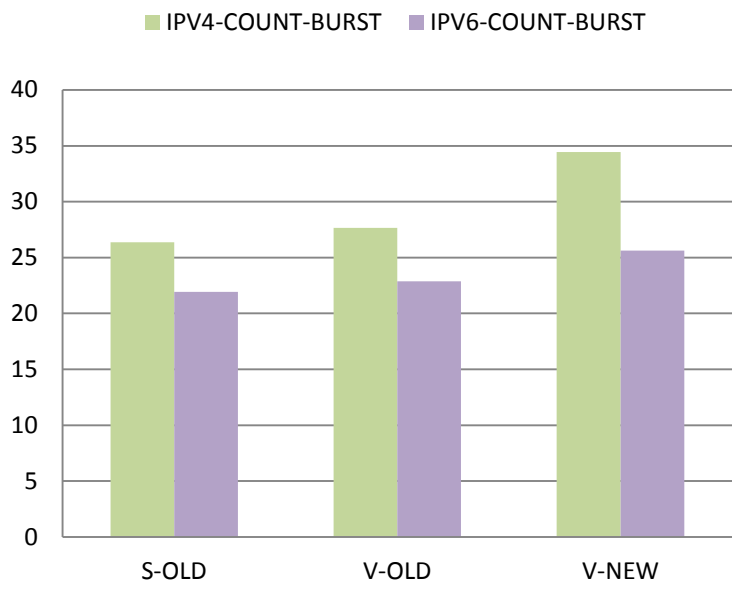
- Things happen in "chunks"
  - Loops unroll
  - Easy to do bulk copies
  - Easier to vectorize
- Defer descriptor writes until multiple have accumulated
  - Reduce probability of modifying a cache line being written by the NIC
- Remove linear dependency on DD bit check
  - Always copy 4 descriptors' data and 4 buffer pointers
  - Hide load latency and fully consuming the double cache load bandwidth, 16Bytes descriptor <-> 128bits XMM register <-> 2 buffer pointers.
  - 128bits descriptor shuffle to 16Bytes buffer header, issue shuffle in parallel
  - Using 'popcnt' to check the number of available descriptors



Practice: vector packet IO (3)



Packet IO throughput optimization



Vector L3fwd throughput

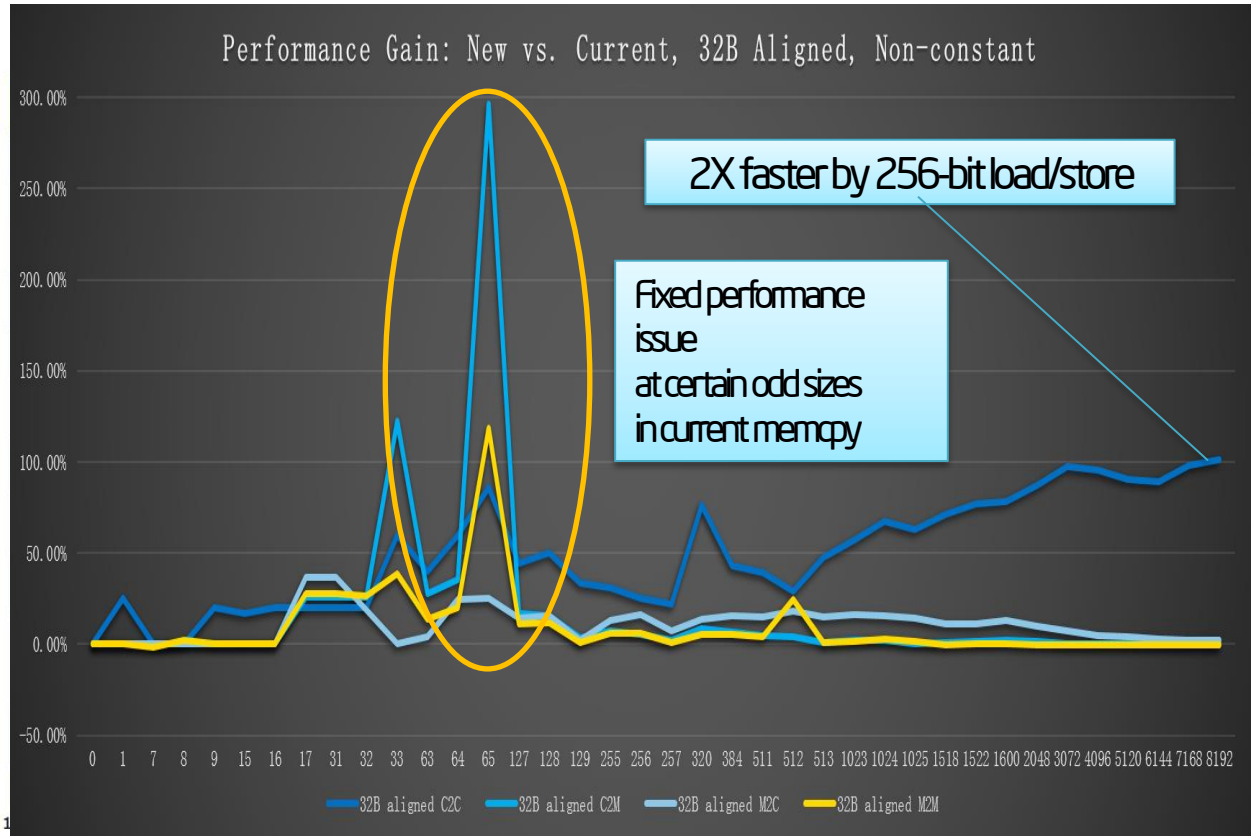
- SNB Server 2.7GHz
- No hyper-thread
- No turbo-burst
- 1 x Core
- 4 x Niantic card, one port/card

- S-OLD – original L3FWD with scalar PMD.
- V-OLD – original L3FWD with vector PMD.
- V-NEW – modified L3FWD with vector PMD

Vector based IO reduces cycle count @ 54 cycles/packet

Practice: AVX2 memory copy

- Utilized 256-bit load/store
- Forced 32-byte aligned store to improve performance
- Improved control flow to reduce copy bytes (Eliminate unnecessary MOVs)
  - Resolved performance issue at certain odd sizes



No weight applied in calculation

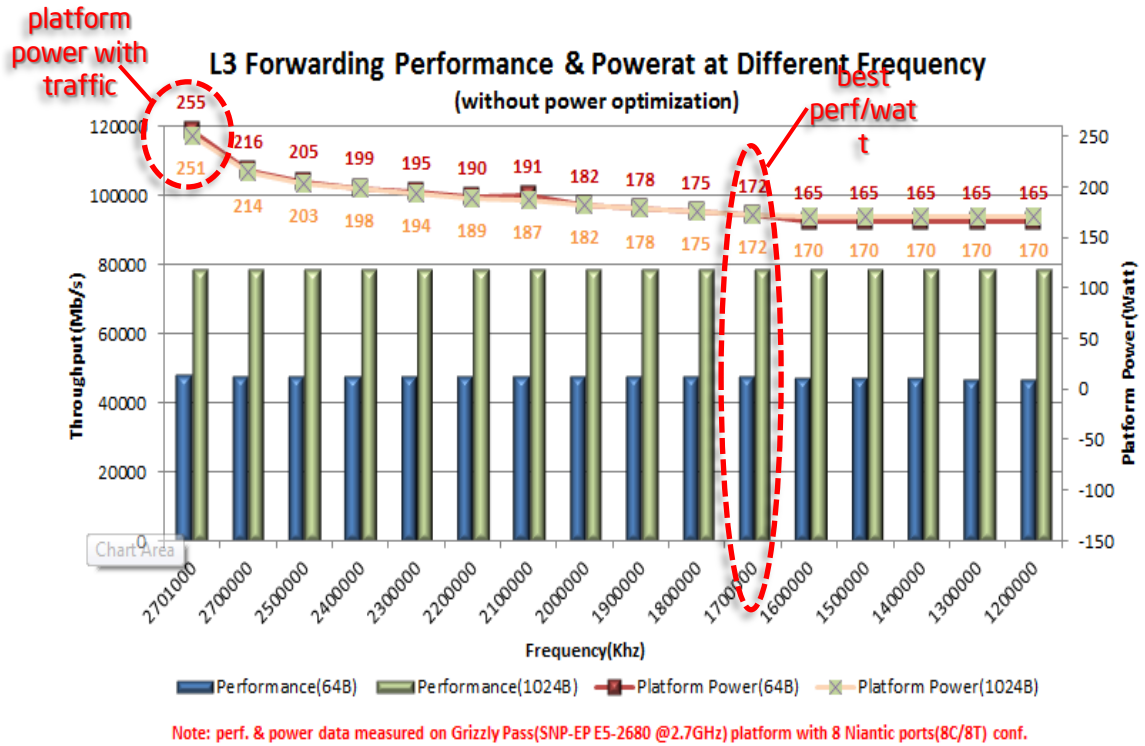
Average throughput speedup on selected sizes, Non-constant	32B aligned			
	C2C	C2M	M2C	M2M
new	1.85	4.57	1.26	2.62
current	1.24	4.06	1.14	2.41
glibc	1.00	1.00	1.00	1.00

Managing Idle Loop

- Problem Statement
  - "always dead loop even no packet comes in"
  - "can we do something else on the packet IO core"
- Effective Way
  - Frequency scale and turbo
  - Limit the IO thread in some quota
  - On-demand yield
  - Turn to sleep

Practice: Power Optimization(1)

	L3fwd
Platform Power (idle)	123W
Platform Power (L3fwd w/o traffic)	245W
CPU Utilization (L3fwd w/o traffic)	100%
Frequency (L3fwd w/o traffic)	2701000 KHz(Turbo Boost)



Note: perf. & power data measured on Grizzly Pass(SNP-EP E5-2680 @2.7GHz) platform with 8 Niantic ports(8C/8T) conf.

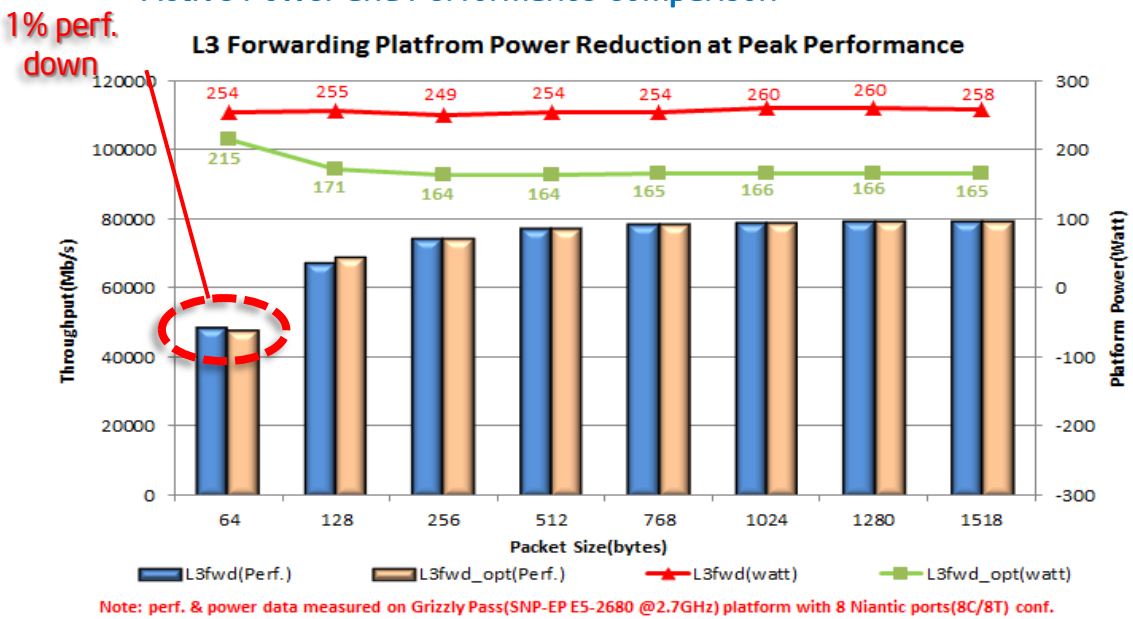
- **Idle Scenario**
  - L3fwd busy-wait loop consumes unnecessary cycles and power
  - Linux power saving mechanism totally not utilized!
- **Active Scenario**
  - Manually set P-state at different freq.
  - Freq. insensitive to I/O intensive DPDK' peak perf., but sensitive to power consumption
  - Negligible peak perf. degradation at lower freq.
  - On SNB, 1.7/1.8G freq. achieves best perf/watt(considering 1C/2T for 2 ports)

Practice: Power Optimization(2)

- Idle Power Comparison

	L3fwd	L3fwd_opt
Platform Power (idle)	123W	123W
Platform Power (L3fwd w/o traffic)	245W	135W
CPU Utilization (L3fwd w/o traffic)	100%	0.3%
Frequency (L3fwd w/o traffic)	2701000 KHz (Turbo Boost)	1200000 KHz

- Active Power and Performance Comparison

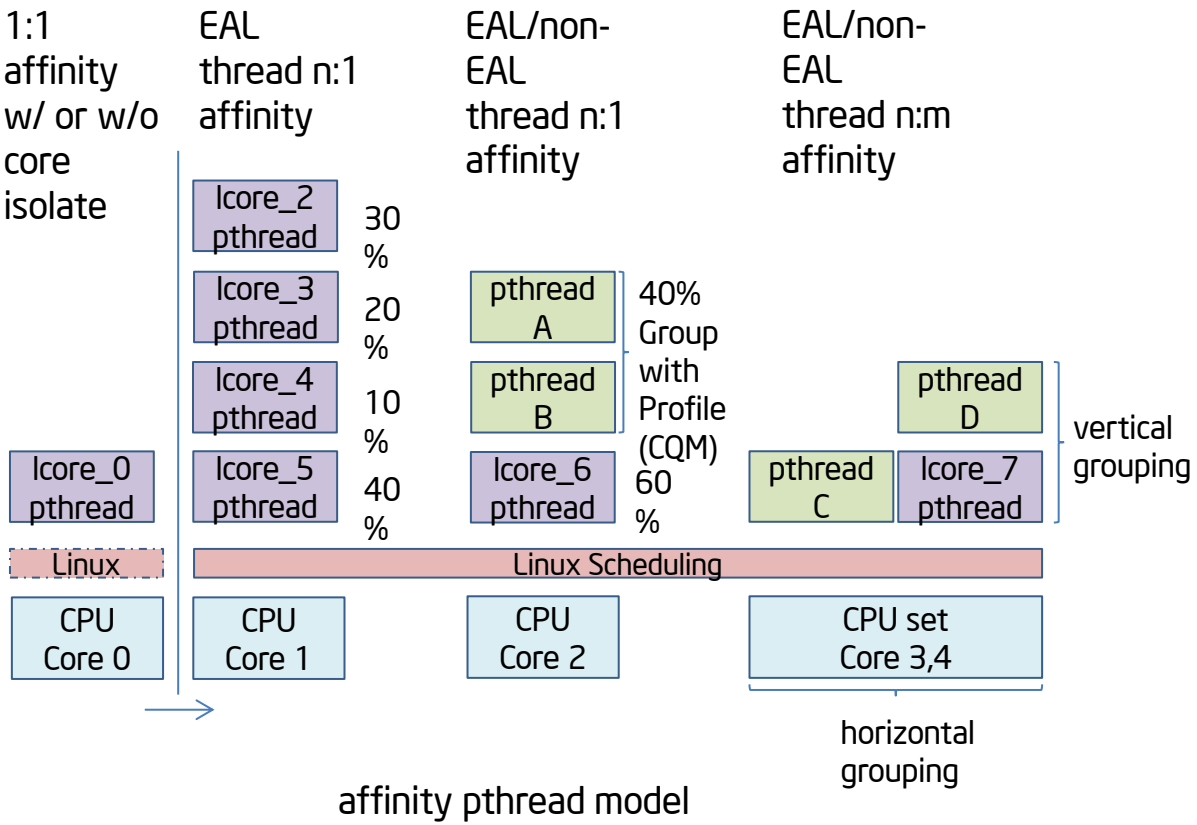


Note: perf. & power data measured on Grizzly Pass(SNP-EP E5-2680 @2.7GHz) platform with 8 Niantic ports(8C/8T) conf.

- Idle Scenario
  - Sleep till incoming traffic
  - Lowest core freq.
  - Power saving for tidal effect

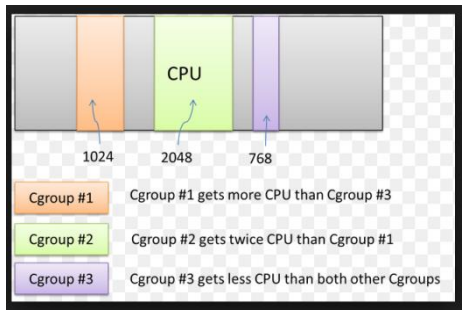
- Active Scenario
  - Peak perf. degradation for 64B only
  - ~90W platform power reduction for the most of cases(different packet sizes)

Practice: Multi-pThreads per Core(1)



affinity pthread model

- Many operations don't require 100% of a CPU so share it smartly
- *Cgroups allows Prioritization where* groups may get different shares of CPU resources
- Split thread model against packet IO



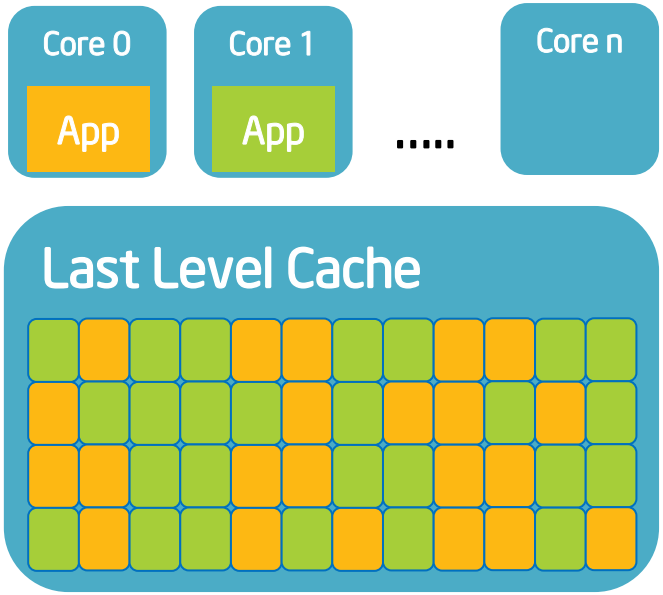
cgroup – Pre-emptive multitasking

Cgroup manages CPU cycle accounting efficiently but what about other resources ?

Cgroup and Cache QoS

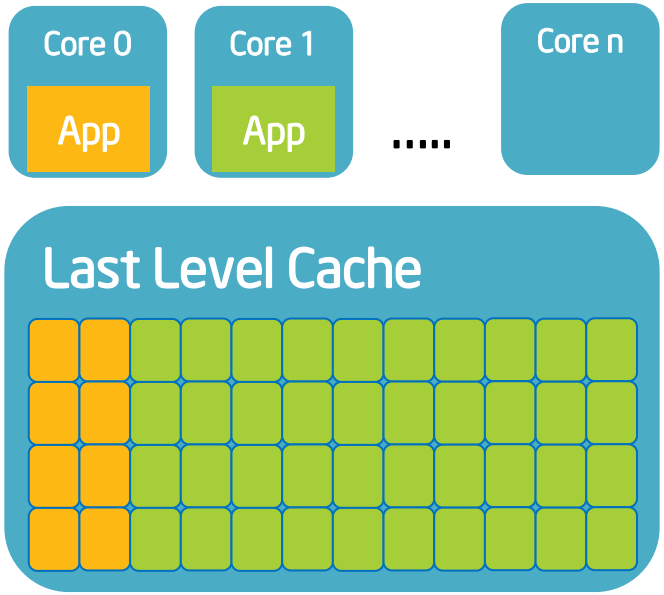
Cache Monitoring Technology (CMT)

- Identify misbehaving or cache-starved applications and reschedule according to priority
- Cache Occupancy reported on per Resource Monitoring ID (RMID) basis



Cache Allocation Technology (CAT)

- Available on Communications SKUs only
- Last Level Cache partitioning mechanism enabling the separation of applications, threads, VMs, etc.
- Misbehaving threads can be isolated to increase determinism



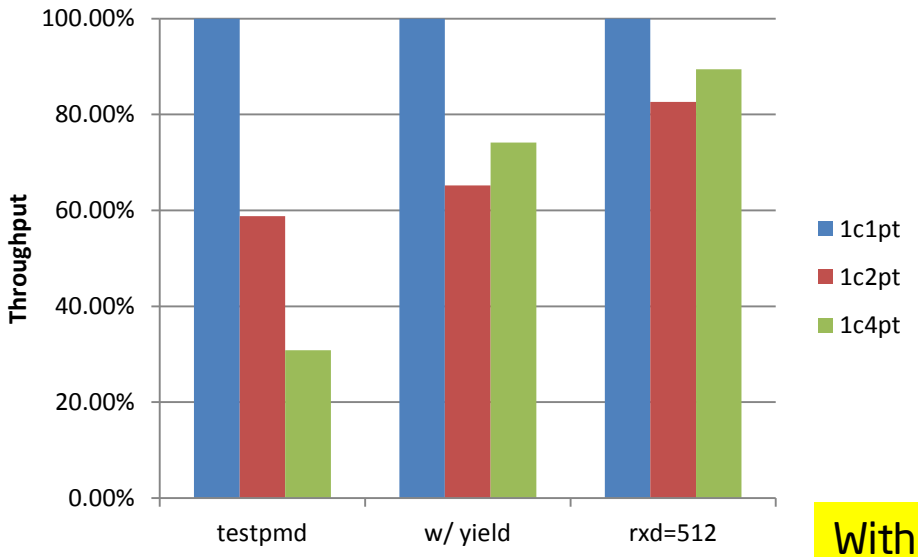
Cache Monitoring and Allocation Improve Visibility and Runtime Determinism

Cgroup can be used to control both of them.

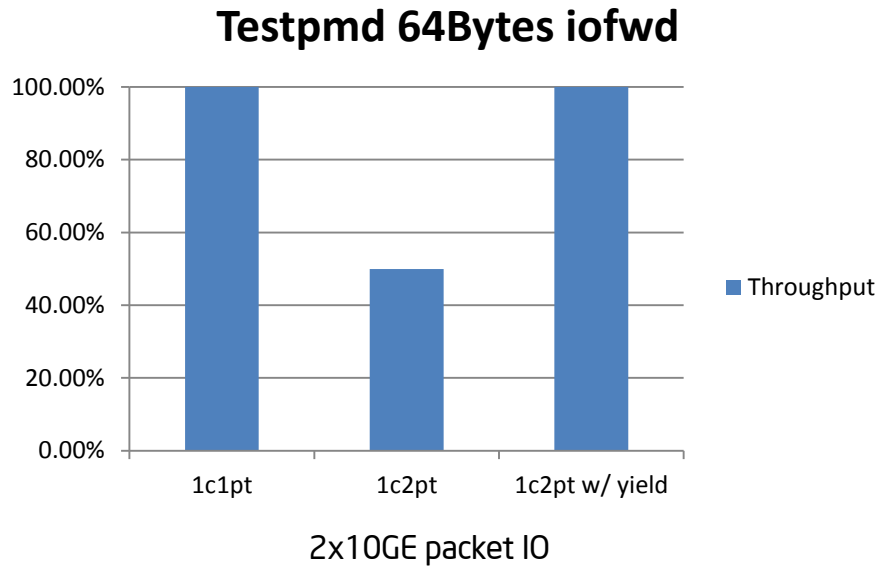
https://www-ssl.intel.com/content/www/us/en/communications/cache-monitoring-cache-allocation-

Practice: Multi-pThreads per Core(2)

- Scheduling task switch latency average 4~6us
- Impact & Penalty on IO throughput



SNB Server 2.7GHz, No hyper-thread, No turbo-burst, 1 x Core, 4 x Niantic card, one port/card

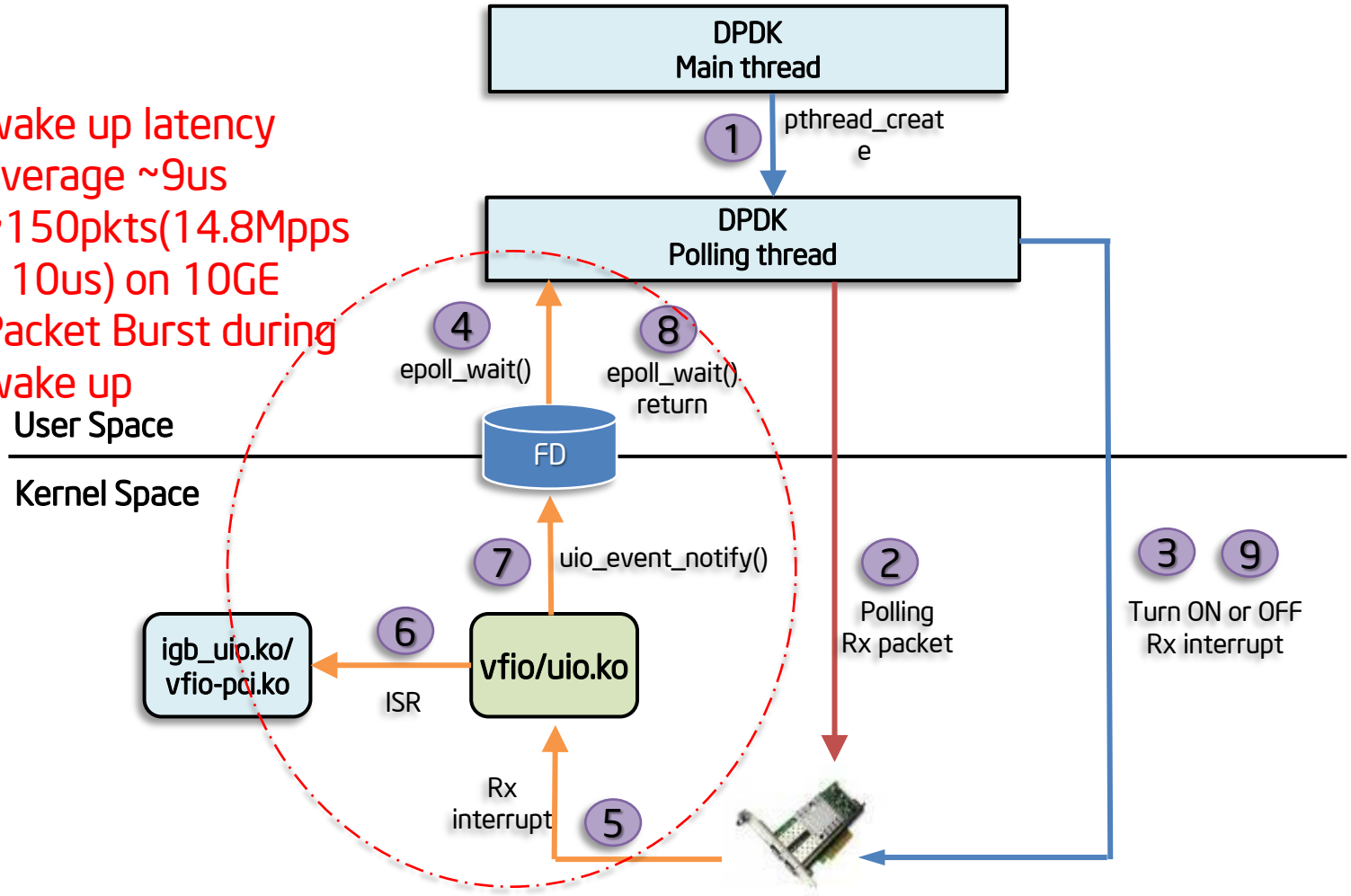


- Make use of RX idle
- On-demand yield
- Queuing more descriptor

With rxd 512 and 1c 4 pthreads achieves ~90% line rate

Practice: Interrupt Mode Packet IO

wake up latency
average ~9us
~150pkts(14.8Mpps
* 10us) on 10GE
Packet Burst during
wake up



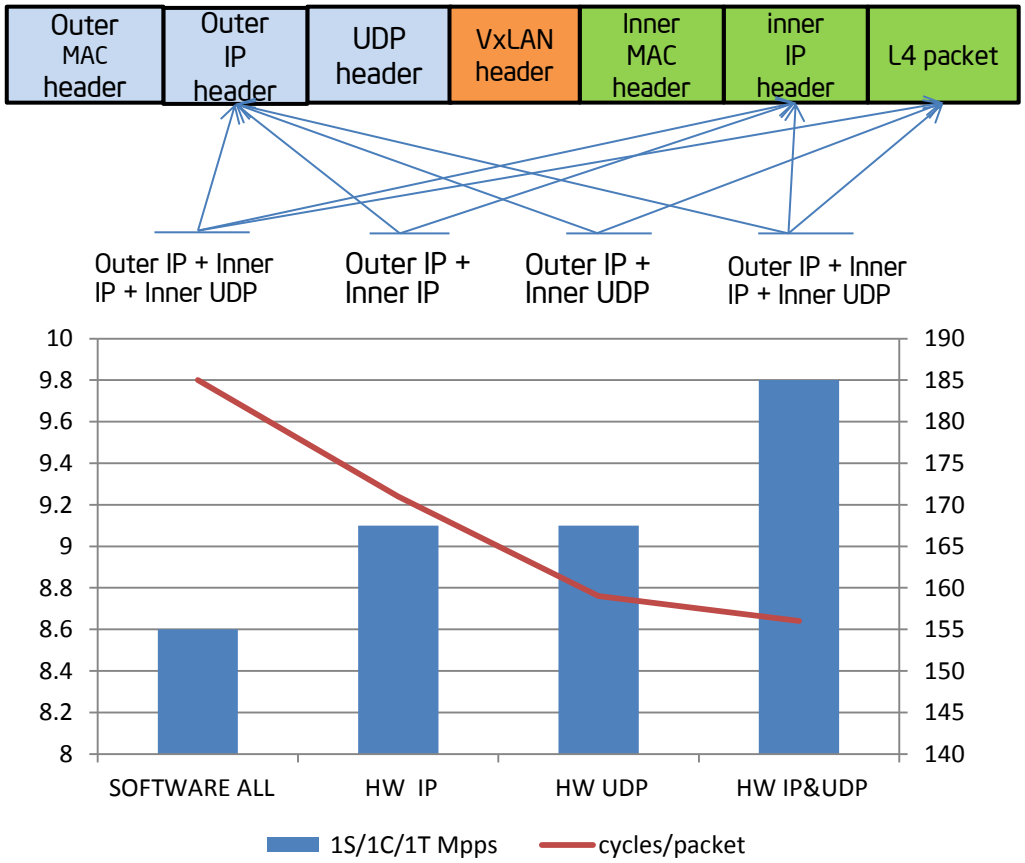
SNB Server 2.7GHz, 1x Niantic port

RX -> interrupt -> Process -> TX -> sleep

Leverage HW offload

- Reduce CPU utilization by HW
- Well known offload capability
  - RSS
  - FDIR
  - CSUM offload
  - Tunnel Encap/Decap
  - TSO

Practice: CSUM offload on VXLAN



VXLAN Inner CSUM offload
HSW 2.3GHz, Turbo Burst Enabled

- 1x40GE FVL
- 128Bytes packet size
- Tunneling packet, VxLAN as sample
- Offload do helps to reduce CPU cycles

Envision/Future

- Light weight thread (Co-operative multitask)
- AVX2 vector packet IO
- Interrupt mode packet IO on virtual ethdev (virtio/vmxnet3)
- Interrupt latency optimization

Thanks