

# RAMCloud on an ATOM Server

June 5, 2014

Satoshi Matsushita

Stanford Univ. / NEC



# NEC Micro Modular Server

- Globally announced on May 20, 2014: ( Press release : [NEC raises the bar for high density IT solution platforms for the public and private cloud](#) )

Chassis: Redundancy (power supply, Networks, Fans) + Hot Swappable

- 2U in standard 19inch rack
- Up to 46 **Atom** server with  
32GB DRAM / 128GB SSD / 2x 2.5GbE
- 2x 230Gbps **switch** (FM5224), 4x 40Gbps uplinks
- Chassis Total: 1.4TB DRAM, 5.8TB SSD
- max. 2kW

16 chassis in a rack:

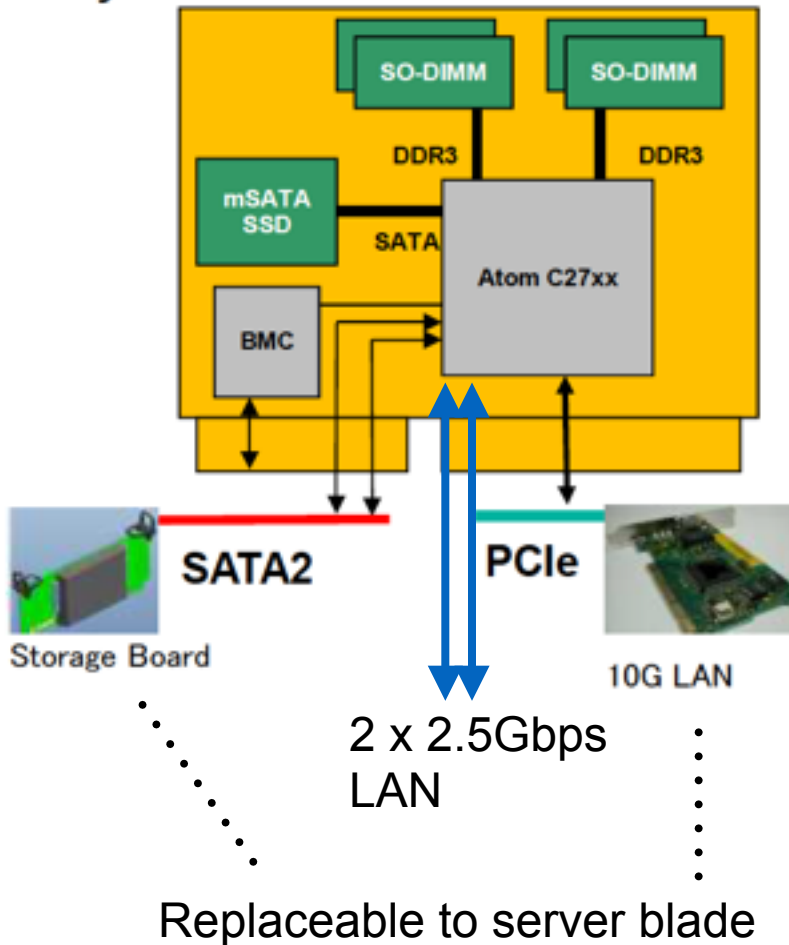
- 736 ATOM Servers : 5.9k core, 23TB DRAM, 92TB SSD:  
50TOp/s, 20TFlops, DRAM 368TB/s, SSD 647GB/s



# ATOM Server Blade

## Server Module

### Block Layout

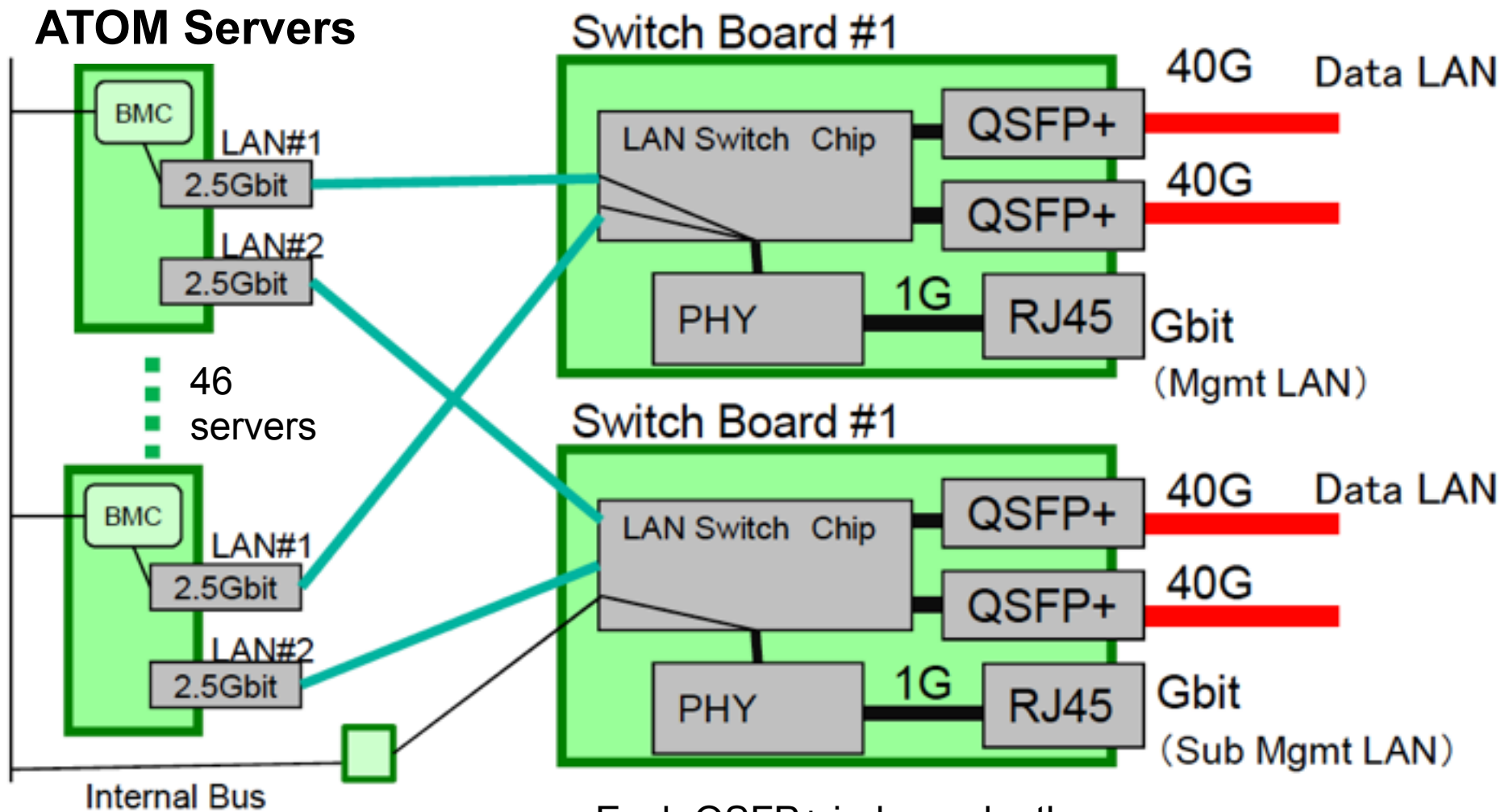


### 【SPECIFICATIONS】

- 1x CPU(Atom™ C27xx)
- 4x SO-DIMM (Max 32GB) w. ECC
- 1x mSATA SSD (128GB)
- 1x BMC
- 1x SATA3 (To mSATA SSD)
- 2x SATA2 (To storage board)
- 2x 2.5Gbit LAN

Processor	Cores	Frequency	Power
C2750	8C / 8T	2.4GHz	20W
C2730	8C / 8T	1.7GHz	12W

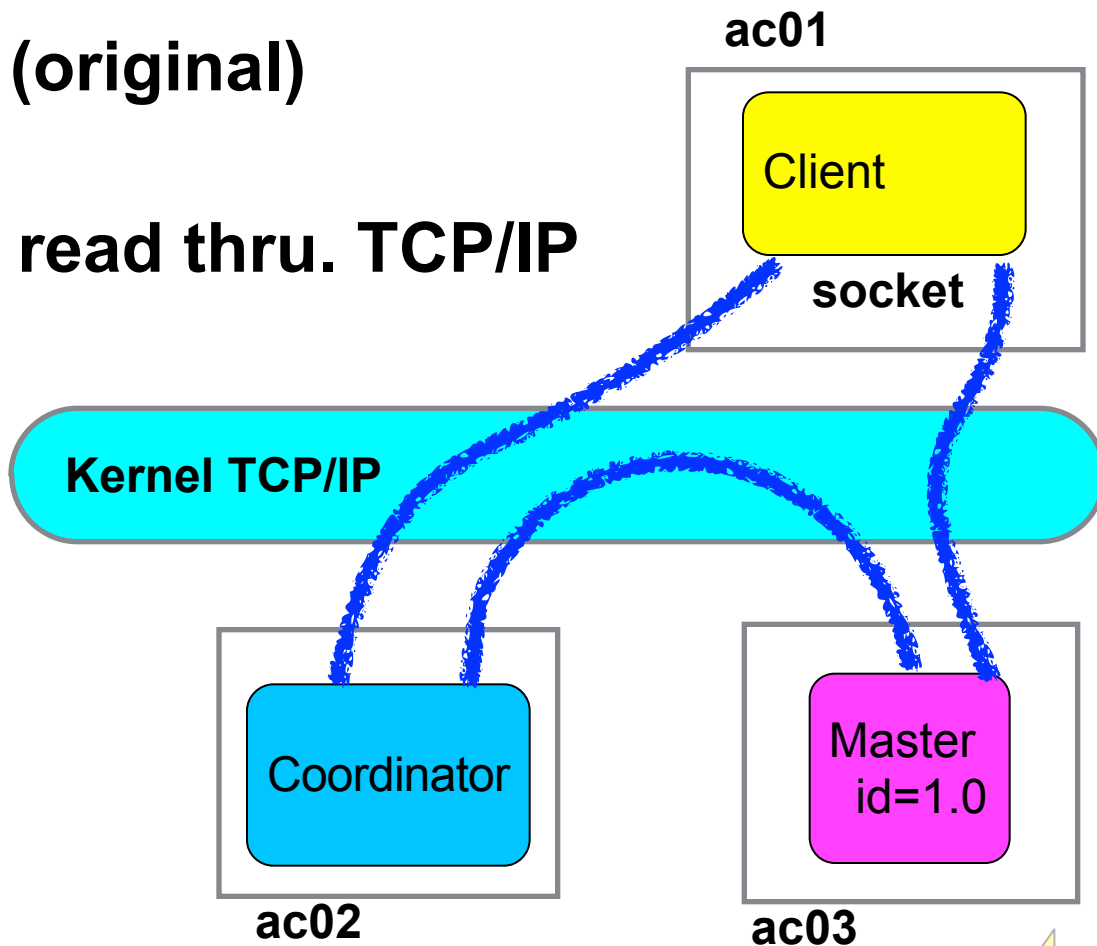
# Connection in a Chassis



Each QSFP+ independently configurable as ether 1x 40G or 4x 10G

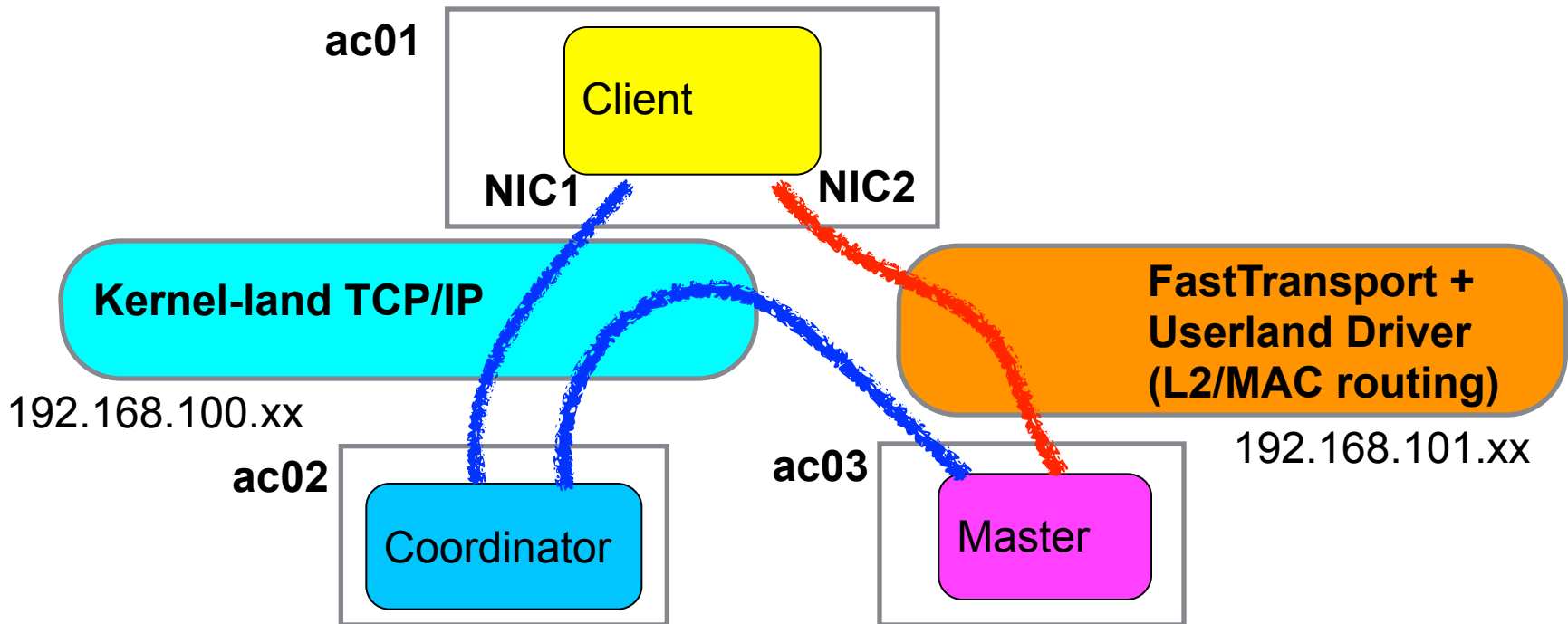
# Base Evaluation

- Disable replication (backup) and collocation of entity
- CentOS 6.5
  - Ping 120-150 us (original)
- Ported RAMCloud
  - 67.8 us for 100B read thru. TCP/IP (tuned)



# Improvement with Userland Driver

- Using userland driver only for critical Master-Client data path.  
No modification in RAMCloud code, changing startup parameter.
- Developed userland driver for NIC2 based  
on **Intel DPDK (Data Plane Development Kit)**



Command Line:

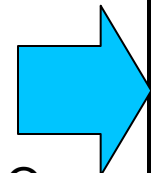
```
$ coordinator -C tcp:host=192.168.100.31,port=12246
```

```
$ server -C tcp:host=192.168.100.31,port=12246 -L fast+dpdk:host=192.168.101.29,mac=94:DE:80:AB:01:79,port=12247 -r 0
```

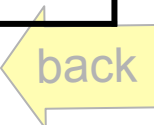
```
$ ClusterPerf -C tcp:host=192.168.100.31,port=12246 --numClients 1 basic
```

# Development Platforms for Userland Driver

	Summary	Performance	License	Comment
PACKET_MMAP	Implementation on the standard linux kernel.	At least one buffer copy needed because a device buffer cannot be mapped.	GPL	-
netmap			GPL/ BSD	Higher safety because userland code cannot access NIC registers directly
PF_RING / DNA (Direct NIC Access)	Possible to map device queue to user space.	Feasible to realize zero-copy in userland driver	GPL/ BSD	-
Intel DPDK (Data Plane Developer Kit)			<b><u>BSD</u></b> (GPL for kernel module)	Rather widely used



Our choice



# RAMCloud with Userland Driver

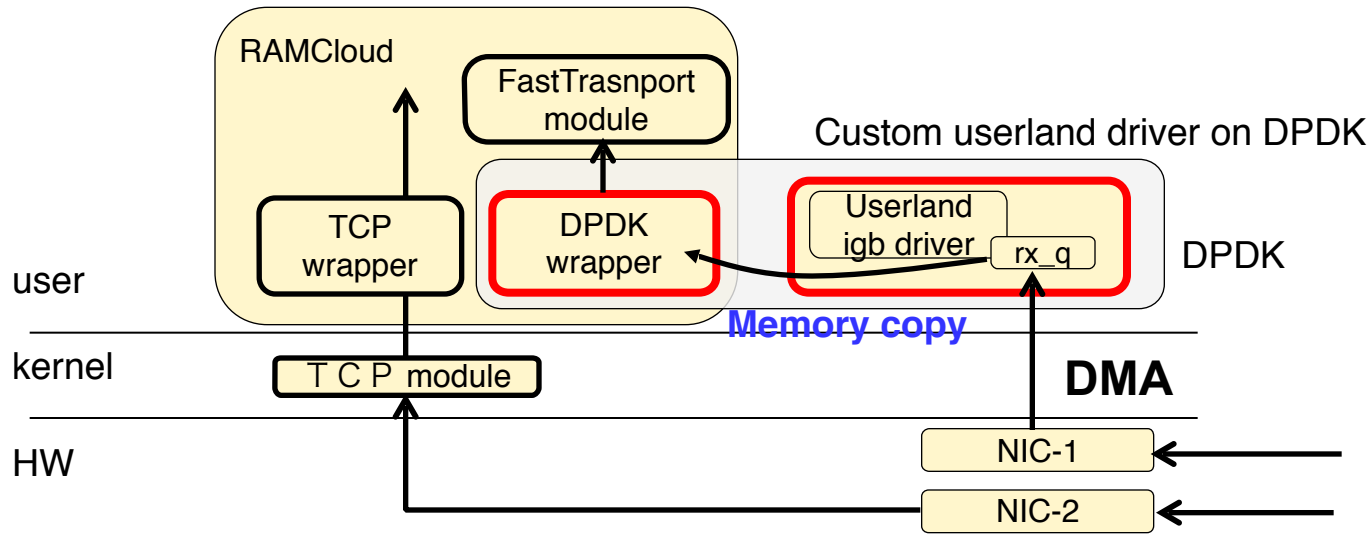


Figure. Customized transport for ATOM server (In-bound)  
(almost the same for out-bound)

- Limitation in current system:
  - L2 routing with 1500 B MTU
  - Non-shared: user space driver is exclusively used by a process
  - Asymmetric: User space driver on NIC-1, ordinal kernel driver on NIC-2
  - RAMCloud multithreading disabled



# Current Performance

- Clusterperf.py basic, 30B key, LAN switch mode: store-and-forward
- Average and min./max. in 100 ms period. (7,000 samples in 100B-read)
- Room for tuning: **long tail** (larger max.), slow write (even backup disabled).

	Atom Server: 1.7GHz + 2.5G Ether				rccluster ( 2.4GHz Xeon + 32Gbps Infiniband )	
Type	Ave.	Min.	Max.	Bandwidth	Ave.	Bandwidth
100B read	13.8 us	13.3	32.7	6.9 MB/s	5.1 us	18.7 MB/s
1KB read	20.7 us	20.0	37.7	46.1 MB/s	6.9 us	137.6 MB/s
10KB read	52.8 us	52.1	68.6	180.8 MB/s	10.4 us	914.1 MB/s
100KB read	373.2 us	371.3	379.0	255.5MB/s	47.2 us	2.0 GB/s
1MB read	3.9 ms	3.8	3.9	247.2 MB/s	420.8 us	2.2 GB/s
100B write	<u>18.2 us</u>	17.4	43.6	5.2 MB/s	15.7 us	6.1 MB/s
1KB write	25.6 us	24.7	64.1	37.2 MB/s	19.9 us	48.0 MB/s
10KB write	64.2 us	62.5	95.5	148.6 MB/s	38.5 us	247.7 MB/s
100KB write	431.4 us	423.2	463.0	221.0MB/s	235.3 us	405.3 MB/s
1MB write	4.7 ms	4.6	4.8	204.6MB/s	2.2 ms	436.0 MB/s

Backup Enabled



# Analysis

- Considerable tail latency and slower writes imply room for improvement

1. Latency breakdown

2. Analysis of low level (hardware) latency

- Comparison against ping with DPDK

- Switch mode effect:

store-and-forward vs. cut-through



# Latency Breakdown: 100B-read

Code Segment	Elapsed	Section	Components	Xeon+IB (rccluster)
Client Code	2.82 us	Co + Ci	Client code including	3.9~3.7 us
User Space Driver	8.35 us	Uo + Ui	Between DPDK driver outlets including NIC, LAN switch	
Server Code	3.02 us	S	Server code including	1.2~1.4 us

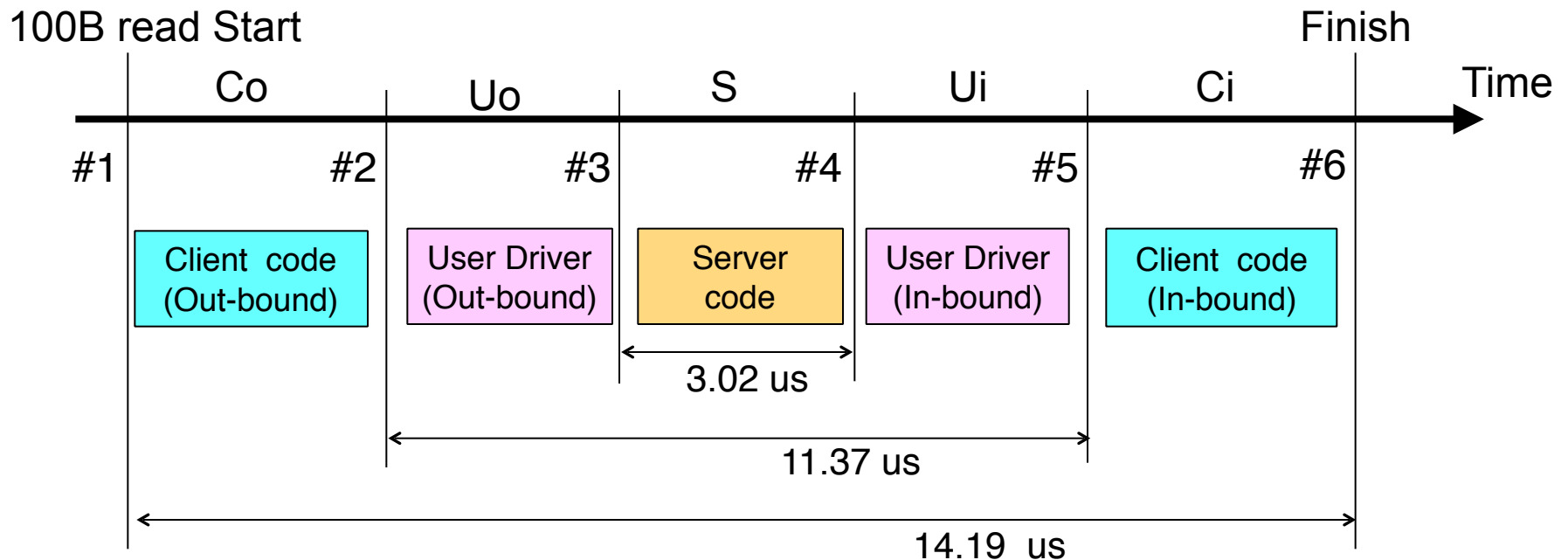


Figure. 100B read latency breakdown



# Latency Breakdown: Ping

Code Segment	Elapsed	Section	Components
Client Code	0 us	Co + Ci	None: IPMI-packet is DMA transferred by NIC (terminated in DPDK driver)
User Space Driver	7 us	Uo + Ui	Between DPDK driver outlets including NIC, LAN switch latency
Server Code	0 us	S	None : (same as Client code)

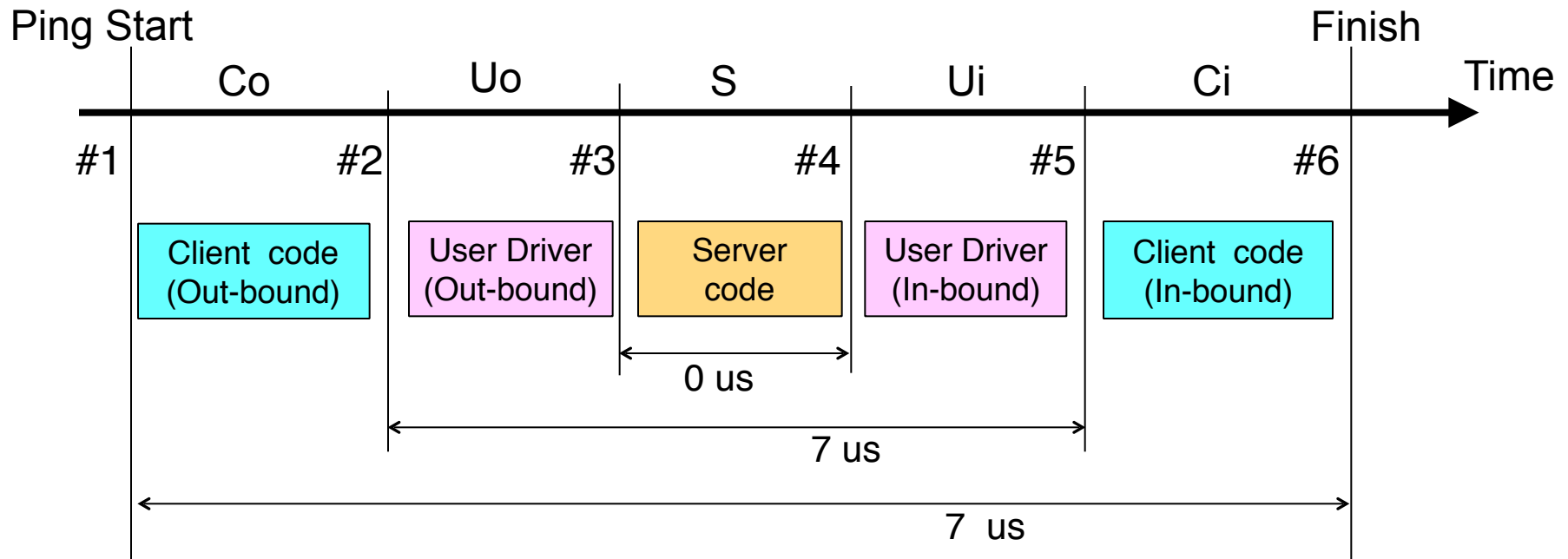


Figure. Ping latency breakdown



# Cut-through

- Slight improvement for larger object size (due to 1500B MTU)
- Clusterperf.py basic, 30B key
- Average and best/worst in 100 ms period. (7000 samples in 100B read)

Atom Server: 1.7GHz + 2.5G Ether								
LAN SW	Store-and-Forward				Cut-Through			
Type	Ave.	Min.	Max.	Bandwidth	Ave.	Min	Max	Bandwidth
100B read	13.8 us	13.3	32.7	6.9 MB/s	13.8 us	13.3	32.2	6.9 MB/s
1KB read	20.7 us	20.0	37.7	46.1 MB/s	17.9 us	17.3	29.0	53.4 MB/s
10KB read	52.8 us	52.1	68.6	180.8 MB/s	48.6 us	47.8	55.9	196.4 MB/s
100KB read	373.2 us	371.3	379.0	255.5MB/s	369.0 us	367.3	376.1	258.4 MB/s
1MB read	3.9 ms	3.8	3.9	247.2 MB/s	3.8 ms	3.8	3.8	251.4 MB/s
100B write	18.2 us	17.4	43.6	5.2 MB/s	18.1 us	17.4	35.2	5.3 MB/s
1KB write	25.6 us	24.7	64.1	37.2 MB/s	22.7 us	21.8	120.8	42.0 MB/s
10KB write	64.2 us	62.5	95.5	148.6 MB/s	60.1 us	58.2	100.3	158.6 MB/s
100KB	431.4 us	423.2	463.0	221.0MB/s	428.3 us	418.9	470.9	222.7 MB/s
1MB write	4.7 ms	4.6	4.8	204.6MB/s	4.6 ms	4.5	4.7	206.8 MB/s

improved

degraded

# Consideration

- Large latency in userland driver (DPDK):
  - 8.35 us for 100B-read, 7 us for ping
- Copy overhead would be negligible:
  - ~0.4us for 100B (~1Kbit) transfer at 2.5Gbps
  - Slight improvement with Cut-through mode
  - Negligible time for 100B memcpy  
(50 ns for 1KB copy on 2.4GHz Xeon)
- To tune the userland driver:
  - Further latency breakdown
  - DPDK parameter tuning
  - Cache footprint optimization??

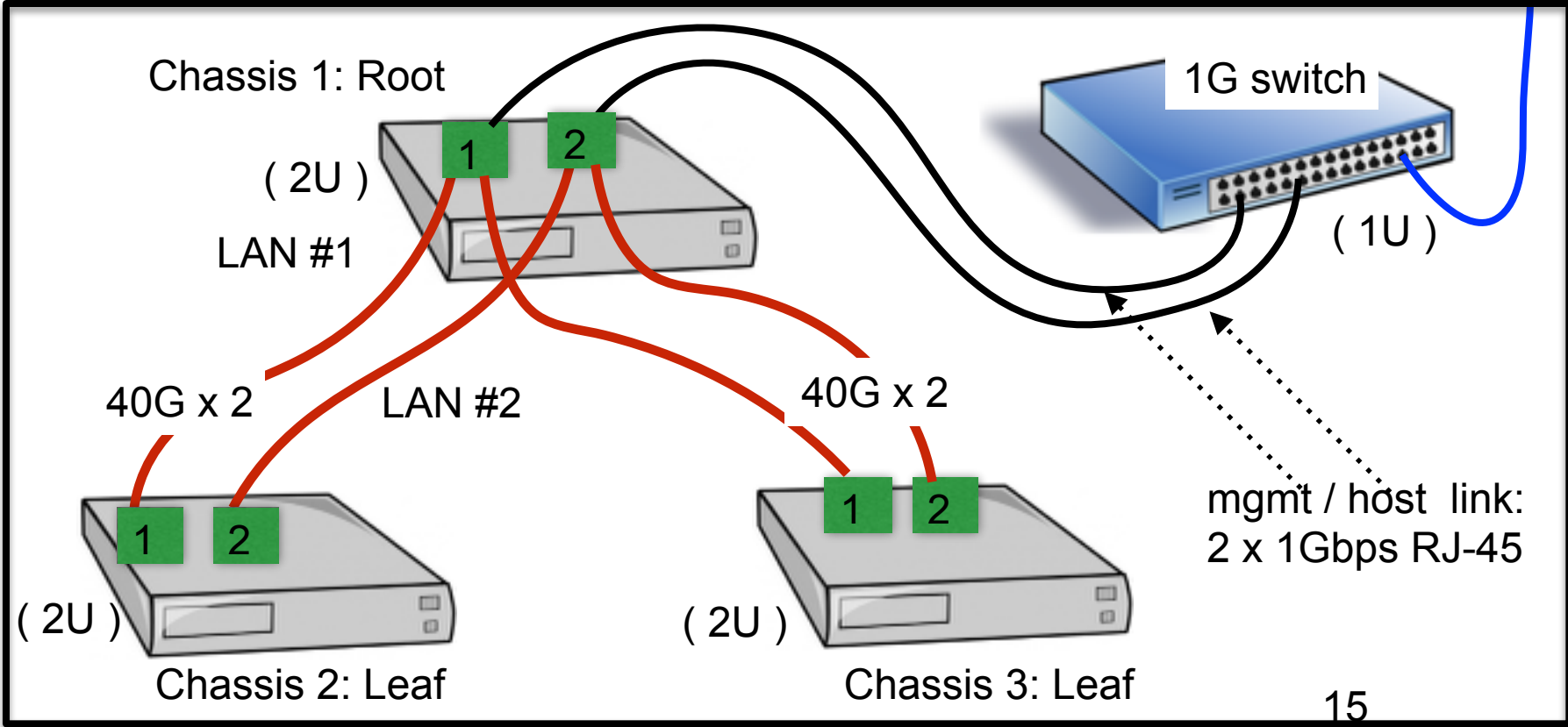


# Spine-switch-less cluster at Stanford

- 1. Connected to for large scale experiments, application development
- 2. Connected to rcmaster with 2x 1Gbps link
- 3. Smaller size, lower power : ~1/5 of Xeon server

## ATOM Cluster (NEC Modular Micro Server)

3 chassis: 132 Servers (1,056 cores), 4.1 TB DRAM, 16.5TB SSD



# Conclusion

- Initial performance evaluation:
  - 13.8 us for 100B-read with custom userland driver on ATOM server through chassis switch (1 hop)
  - Further analysis and tuning
- Functional enhancement:
  - Symmetric driver and link aggregation with two NICs
  - Providing a turn-key-solution
    - with job/network/storage/VM management tools
    - on a standardized hardware platform
- Further evaluation on a larger scale system
  - On a new ATOM cluster at Stanford
  - Application development and evaluation
- Very welcome for feedback to improve the Micro modular server and future systems

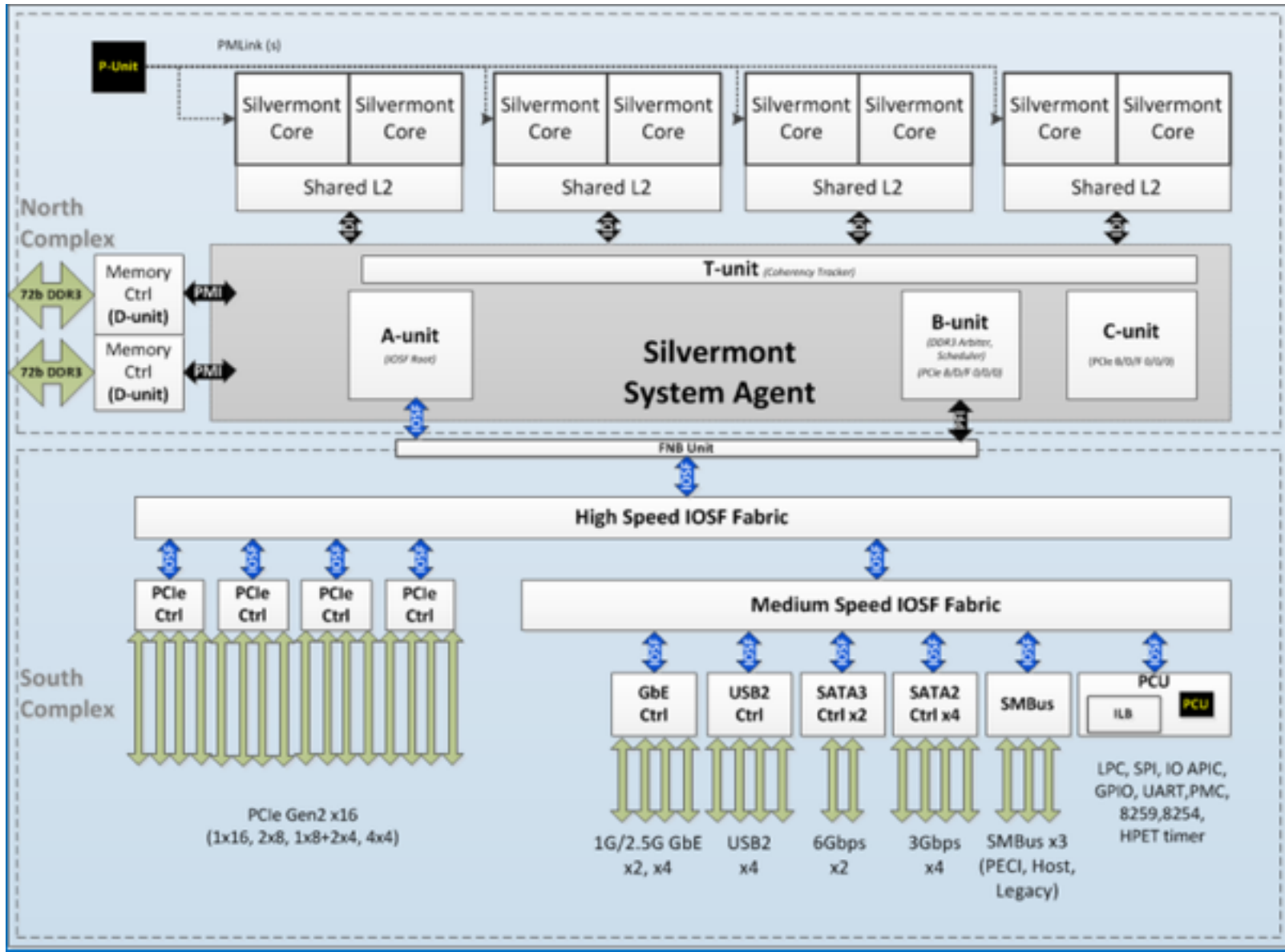




# Appendix

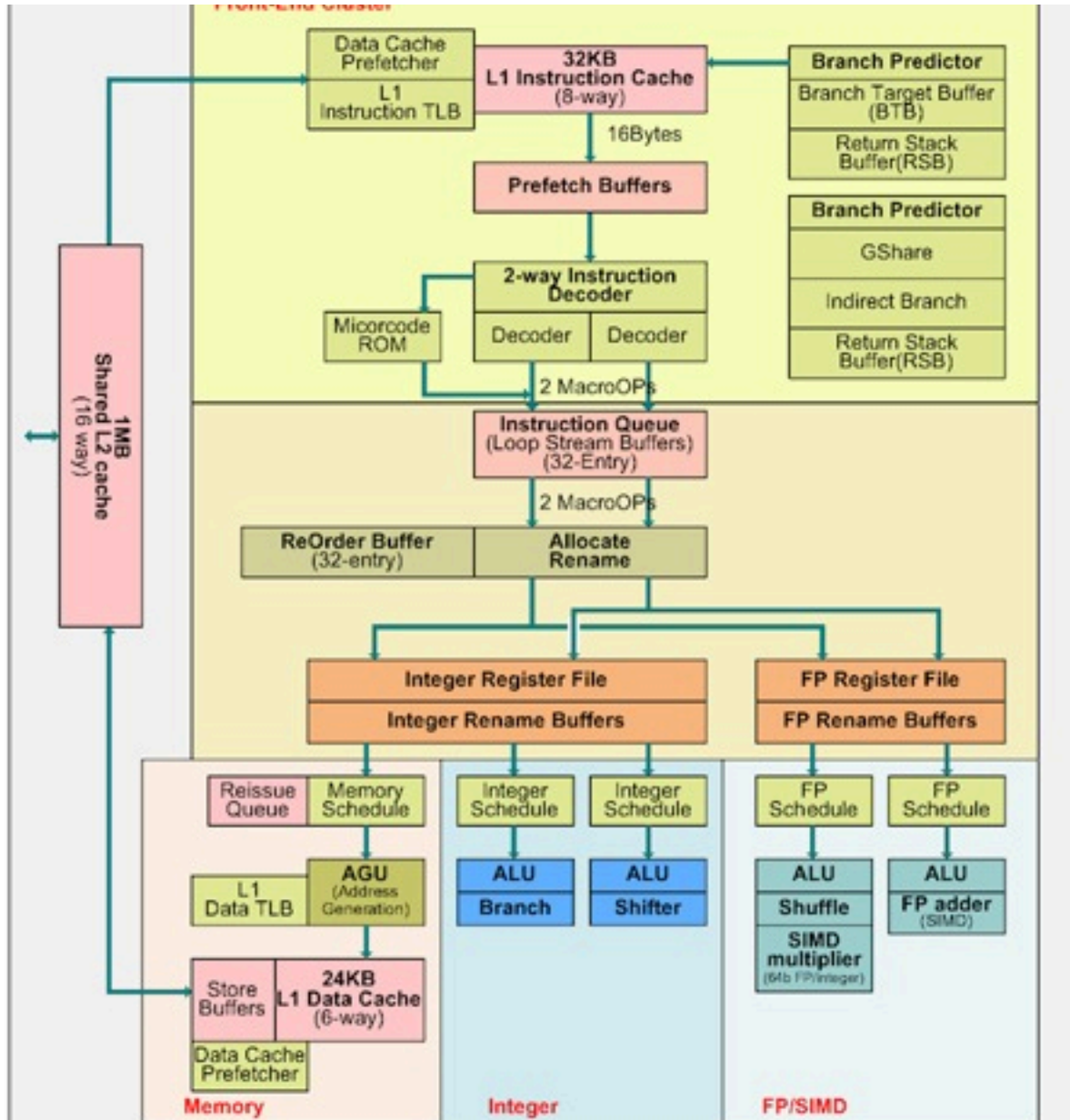


# ATOM Avoton SoC



8 core  
 4x1MB L2  
 2x72b  
 ECC DDR3  
 PCIe Gen2  
 x 16  
 I/O  
 2x 2.5GbE  
 USB2  
 SATA2/3  
 SMBus  
 PCU

# ATOM Silvermont Micro Architecture



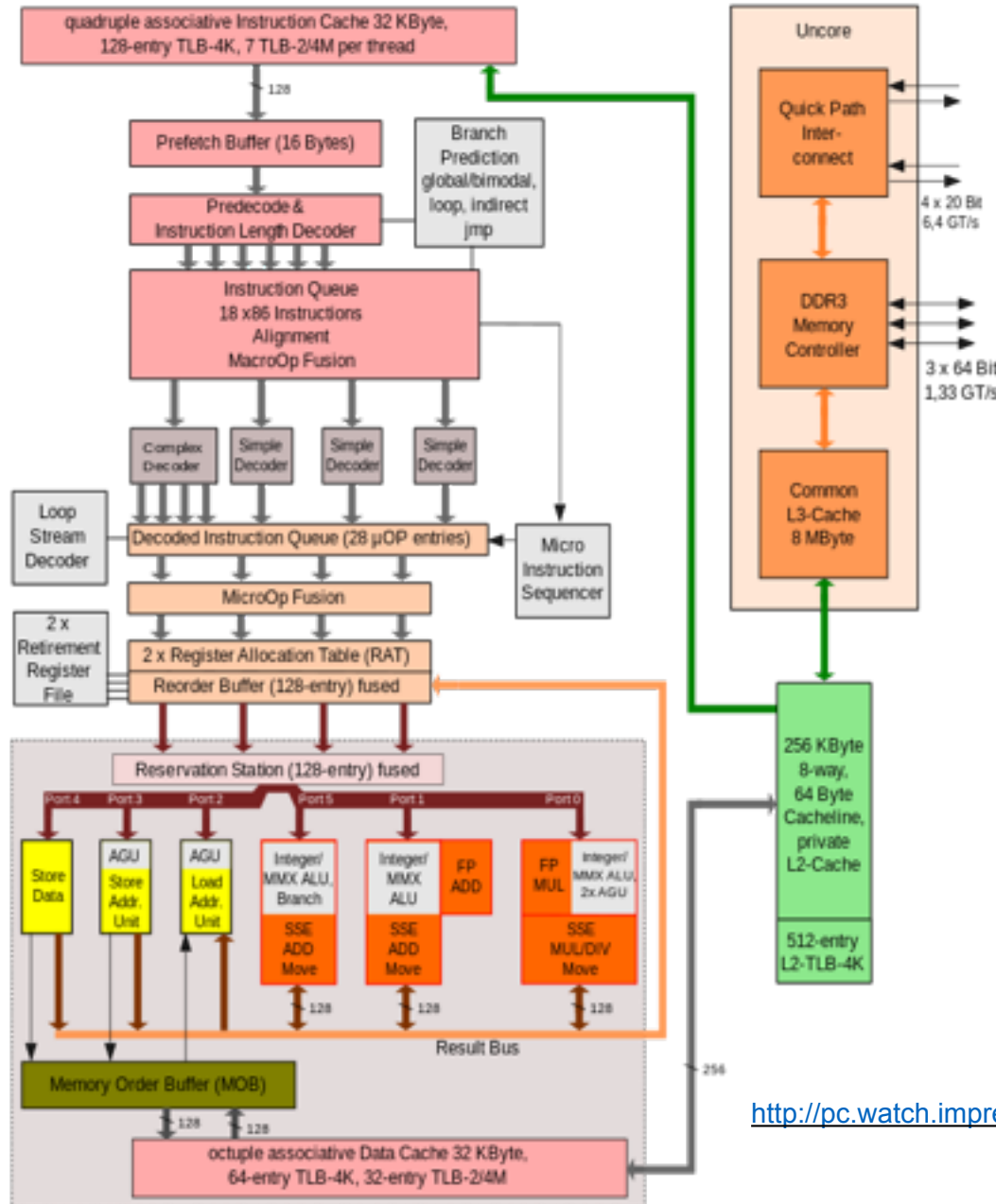
ATOM C27xx

1.7GHz/2.4GHz  
 I\$/D\$=32KB 8way  
 /24KB 6way  
 L2=1MB 16way

2 issue  
 32 Entry ROB

1 Memory Unit  
 2 ALU  
 2 FP

# Xeon (Nehalem) Micro Architecture



Current RAMCloud Cluster








Xeon E5620  
4core 2.4 GHz  
L3 :12MB

(Westmare 45nm  
- Nehalem Arch.)

<http://pc.watch.impress.co.jp/img/pcw/docs/601/161/html/03.jpg.html>

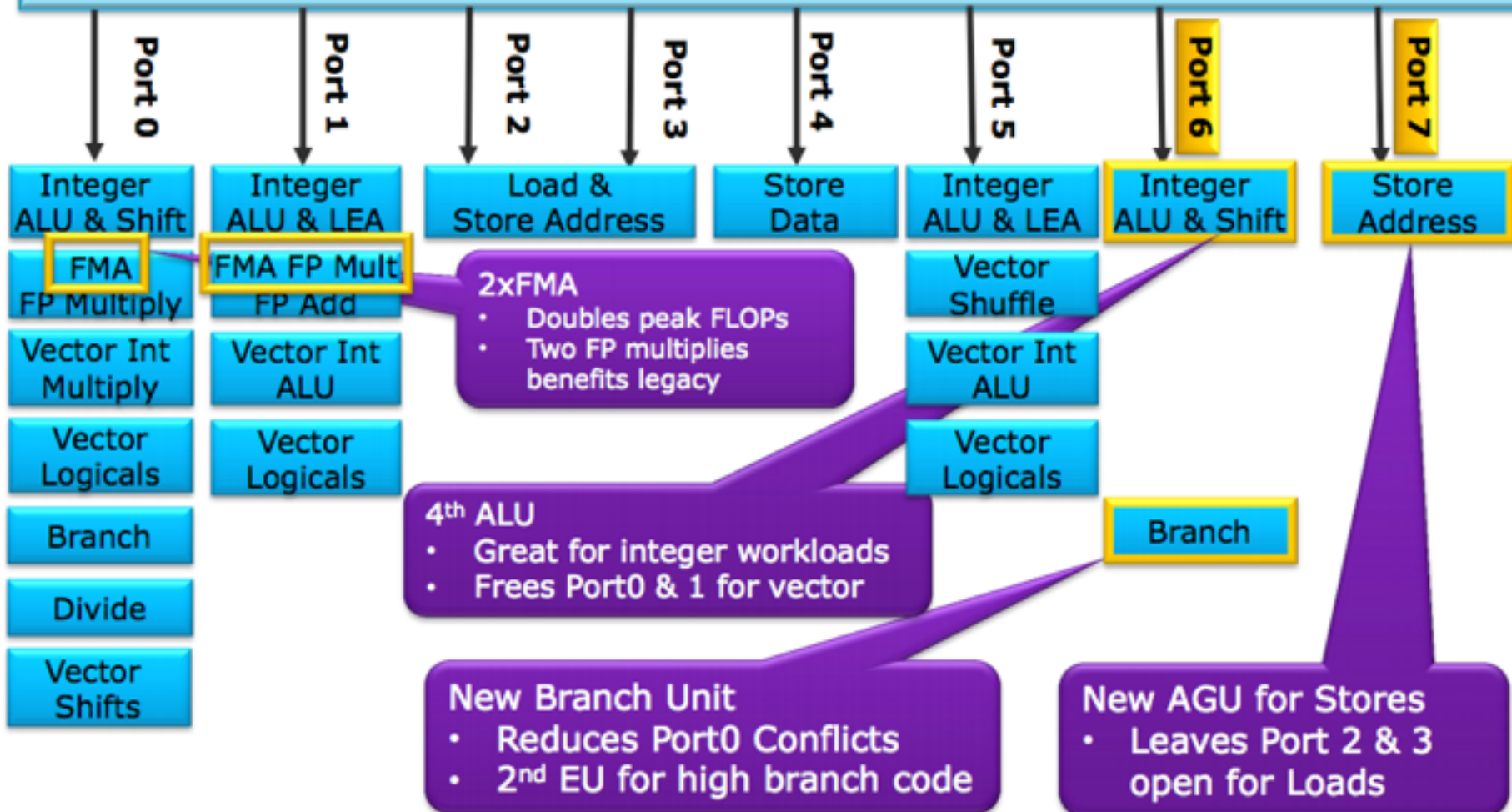
# Haswell Buffer Sizes

Extract more parallelism in every generation

	Nehalem	Sandy Bridge	Haswell	
Out-of-order Window	128	168	192	
In-flight Loads	48	64	72	
In-flight Stores	32	36	42	
Scheduler Entries	36	54	60	
Integer Register File	N/A	160	168	
FP Register File	N/A	144	168	
Allocation Queue	28/thread	28/thread	56	

# Haswell Execution Unit Overview

## Unified Reservation Station





# Core Cache Size/Latency/Bandwidth

Metric	Nehalem	Sandy Bridge	Haswell
L1 Instruction Cache	32K, 4-way	32K, 8-way	32K, 8-way
L1 Data Cache	32K, 8-way	32K, 8-way	32K, 8-way
Fastest Load-to-use	4 cycles	4 cycles	4 cycles
Load bandwidth	16 Bytes/cycle	32 Bytes/cycle (banked)	64 Bytes/cycle
Store bandwidth	16 Bytes/cycle	16 Bytes/cycle	32 Bytes/cycle
L2 Unified Cache	256K, 8-way	256K, 8-way	256K, 8-way
Fastest load-to-use	10 cycles	11 cycles	11 cycles
Bandwidth to L1	32 Bytes/cycle	32 Bytes/cycle	64 Bytes/cycle
L1 Instruction TLB	4K: 128, 4-way 2M/4M: 7/thread	4K: 128, 4-way 2M/4M: 8/thread	4K: 128, 4-way 2M/4M: 8/thread
L1 Data TLB	4K: 64, 4-way 2M/4M: 32, 4-way 1G: fractured	4K: 64, 4-way 2M/4M: 32, 4-way 1G: 4, 4-way	4K: 64, 4-way 2M/4M: 32, 4-way 1G: 4, 4-way
L2 Unified TLB	4K: 512, 4-way	4K: 512, 4-way	4K+2M shared: 1024, 8-way

All caches use 64-byte lines

# Intel® Ethernet Switch FM5224

## Microserver Switch Silicon

### Unmatched uServer density

- Up to 72 2.5G ports
- 8 10GbE or 2 40GbE uplinks

### Rapid Array shared memory

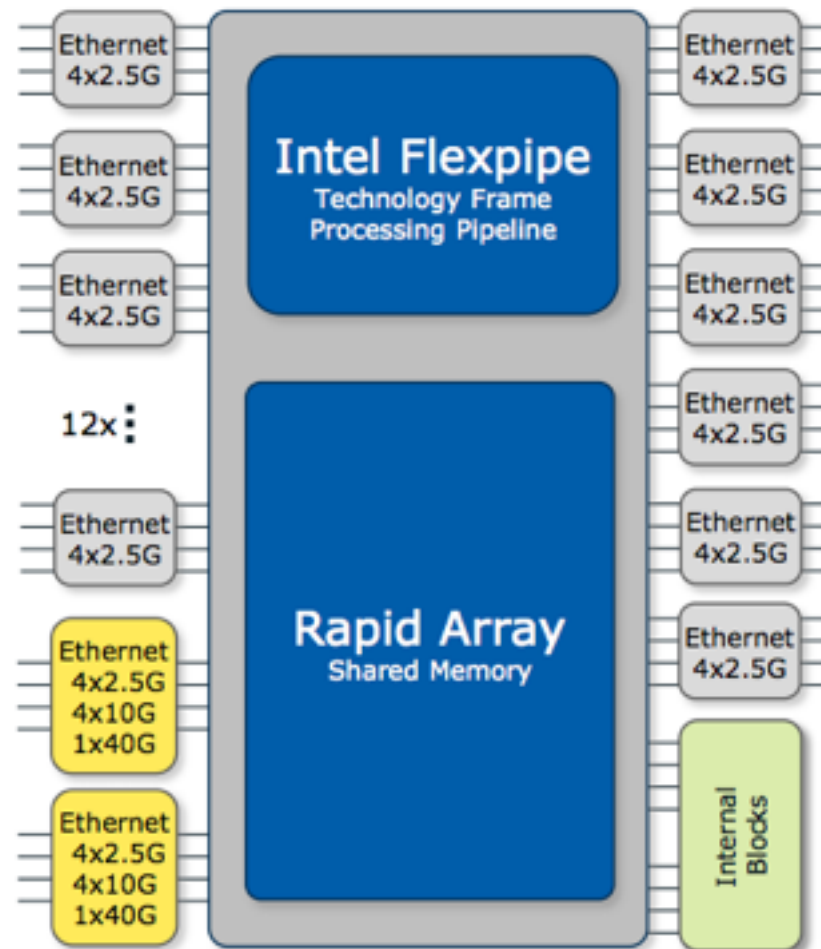
- 8MB shared memory
- 400nS cut-through latency

### Intel® Flexpipe™ Technology frame processing

- Intel Flexpipe Technology frame processing
- VXLAN and NVGRE support
- Advanced load balancing
- IPv4/v6 routing
- CEE/DCB with 8 traffic classes
- Server virtualization support

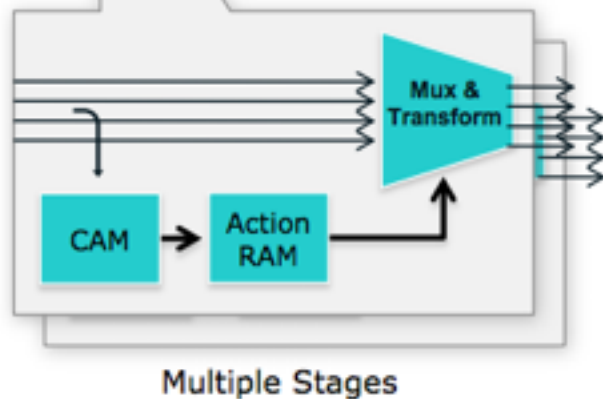
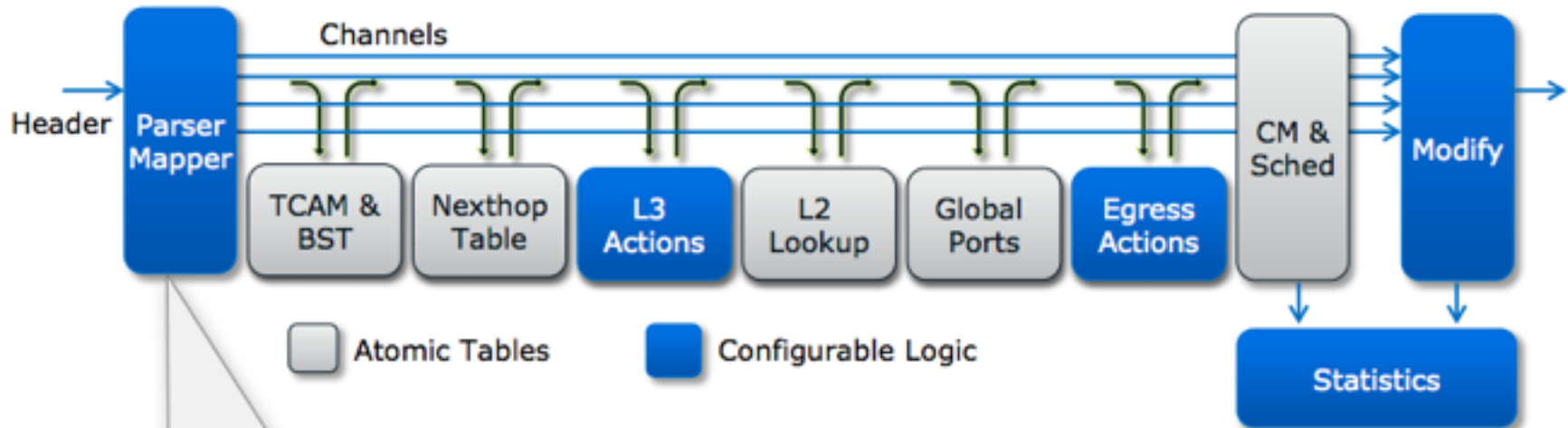
### Compact, flexible port logic

- Integrated SFI, KR PHY
- All ports can also operate at 10/100/1000/2500





# Intel® Flexpipe™ Technology Frame Processing Pipeline



## Sample Programmable Protocols

Tunneling	TRILL, MPLS, NAT
Network Overlays	VxLAN, NVGRE
Virtualization	EVB, VEPA, VEPA+, VN-Tag
Proprietary	Customer defined headers

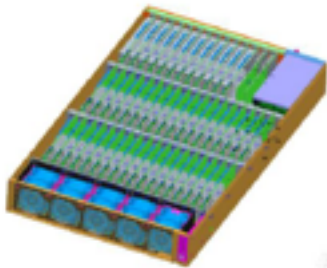
**Programmable and deterministic up to 960Mpps**

Ref: [https://intel.activeevents.com/sf13/connect/fileDownload/session/A02B7458AF93EB0153BB728308E30F99/SF13\\_CLDS006\\_101.pdf](https://intel.activeevents.com/sf13/connect/fileDownload/session/A02B7458AF93EB0153BB728308E30F99/SF13_CLDS006_101.pdf)

# Machine Specification

- Three Chassis arrived in April 2014.
- Total: 6U, 150kg
- 200V 6kW power (2 redundant C-13 socket @chassis)
- 4.8kW maximum at normal operation
- less than 1/3 of max. with CPU idle
- Can turn off each server blade and observe chassis temperature with IPMItool.

## Server Chassis

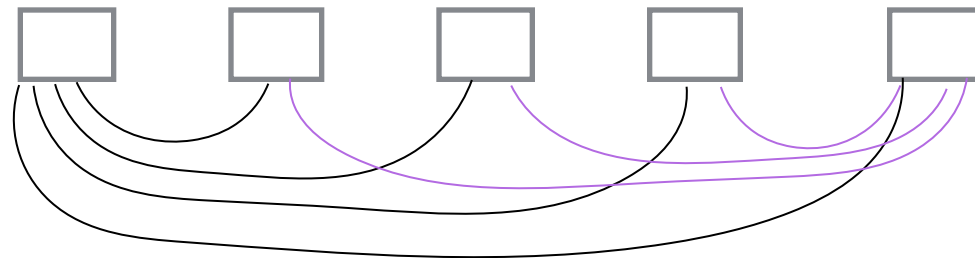


		Chassis
Chassis Dimension (Width x Depth x Height)	x x	2U Chassis Width : 19Inch Depth: 800mm
Weight		Up to 47 kg (system)
Power		Two AC 200V 1.6kw power supply IEC C-14 connector Power consumption : Up to 1.9kw per chassis
Temperature		10 degree C ~ 40 degree C (Ambient air temp.)

# More Chassis - Various Spineless Topology

- Extension for loop tolerable routing needed
- Up to 5 chassis (220 servers): full connection with shortest 1 hop - connect other four servers with four links in a chassis

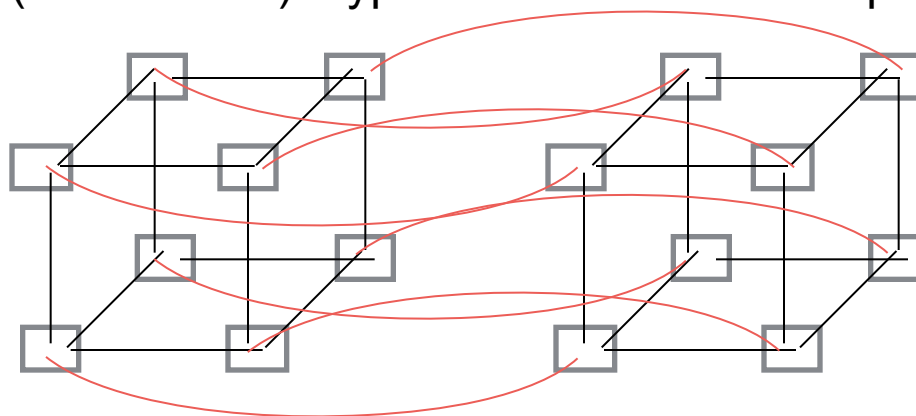
n link/chassis:  
n+1 chassis  
within 1 hop



Note)  
Number of  
TOR SW to  
go through  
is hop + 1

- Up to 16 chassis (704 servers): hyper cube with max 4 hops

$2^n$  chassis  
within n hop



If n=16:  
65,536 chassis  
(2.9M servers)  
within 16 hop!!!  
(Hard to connect  
though...)

- Split fiber increase ports from a chassis to 16 x 10G which enable much larger configuration. (or combination of 10G & 40G)

- 4D torus, Fat tree, etc : more chassis with more hops or hot spots

# Intel® Data Plane Development Kit (Intel® DPDK)

Intel® DPDK embeds optimizations for the IA platform:

## - Data Plane Libraries and Optimized NIC Drivers in Linux

### User Space

*Queue & Buffer Management, Packet Flow Classification, Poll-Mode NIC Drivers (1/10GbE), and more!*

*Simple API Interface, Uses standard tool chain (gcc/icc, gdb, profiling tools)*

## - Run-time Environment

*Low overhead, run-to-completion model optimized for fastest possible data plane performance*

## - Environment Abstraction Layer and Boot Code

*Primarily platform-specific boot guidelines and initialization code, eases application porting effort*

## - BSD-licensed & source downloadable from Intel and leading ecopartners

*Provided under a very flexible BSD licensing model*

*Offered as a free, unsupported standalone solution by Intel or as part of commercial solutions and offerings from leading ecopartners*

