RAMCloud on ATOM Server

June 5, 2014

Satoshi Matsushita
Stanford Univ. / NEC

back

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 Swap

- 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

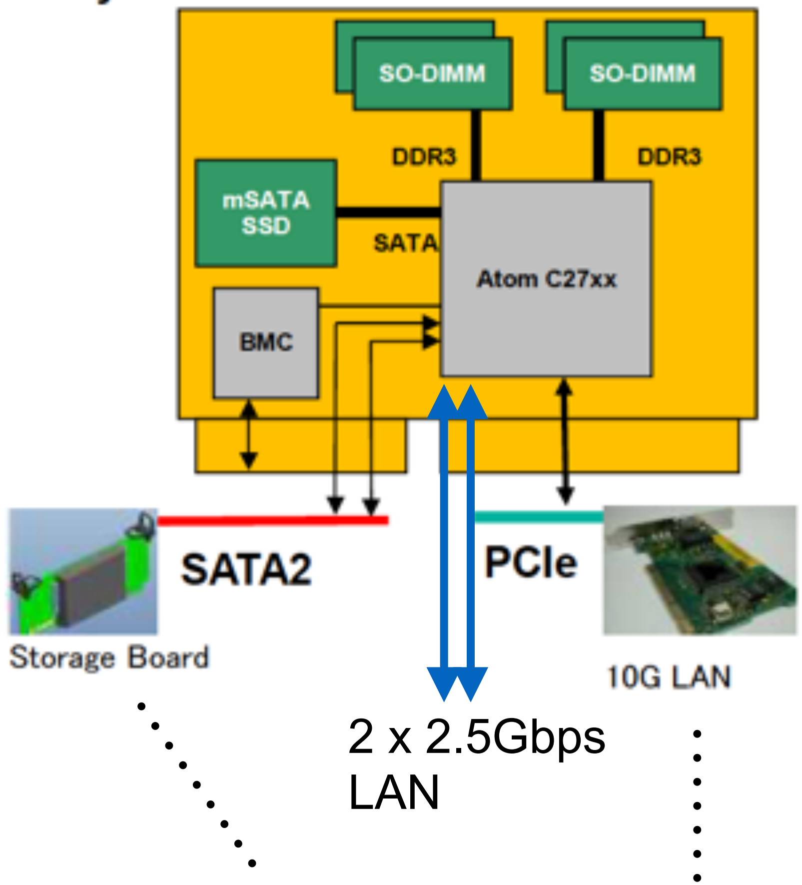
16 chassis in a rack:

- 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

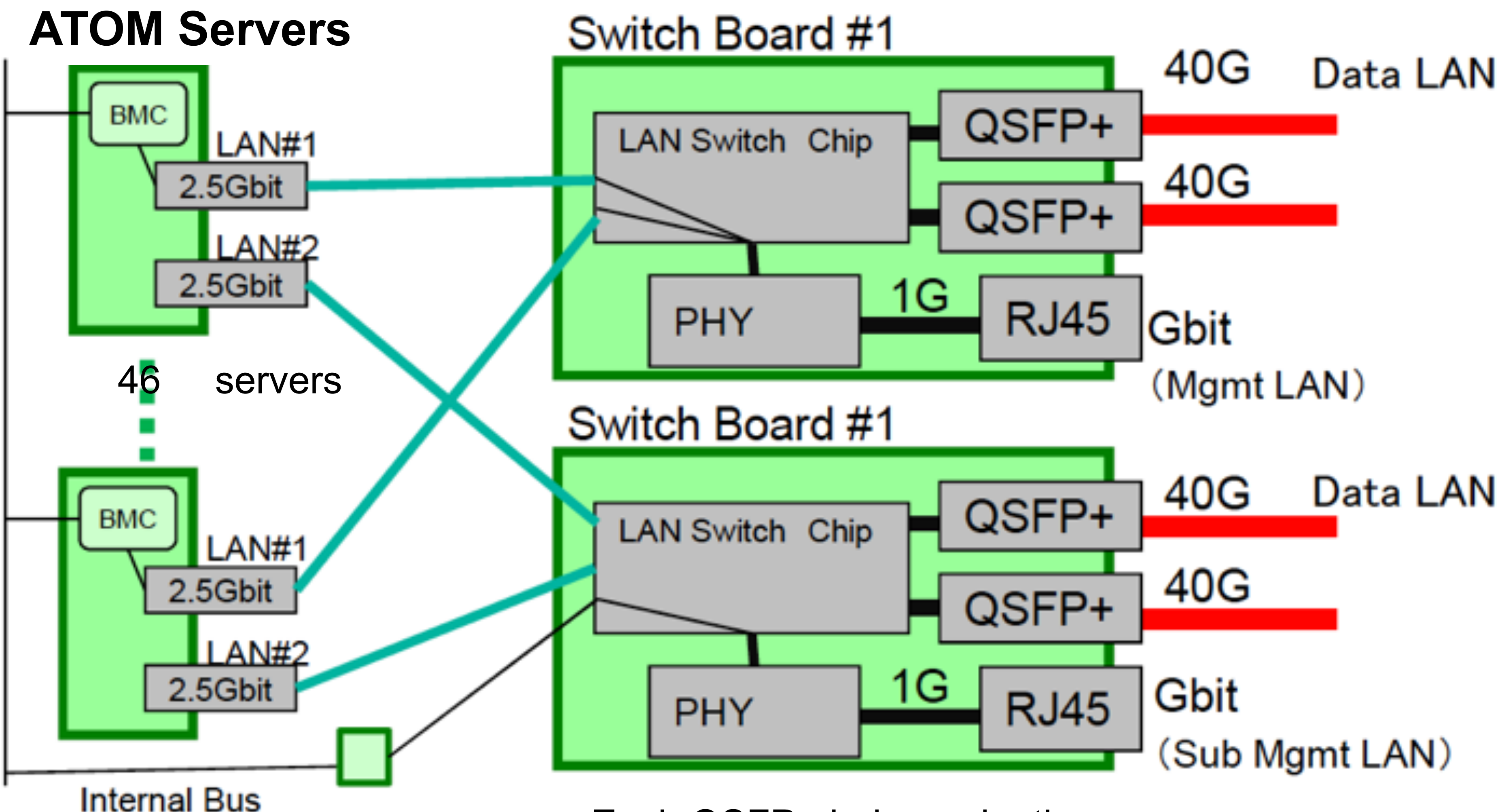


Replaceable to server blade

【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

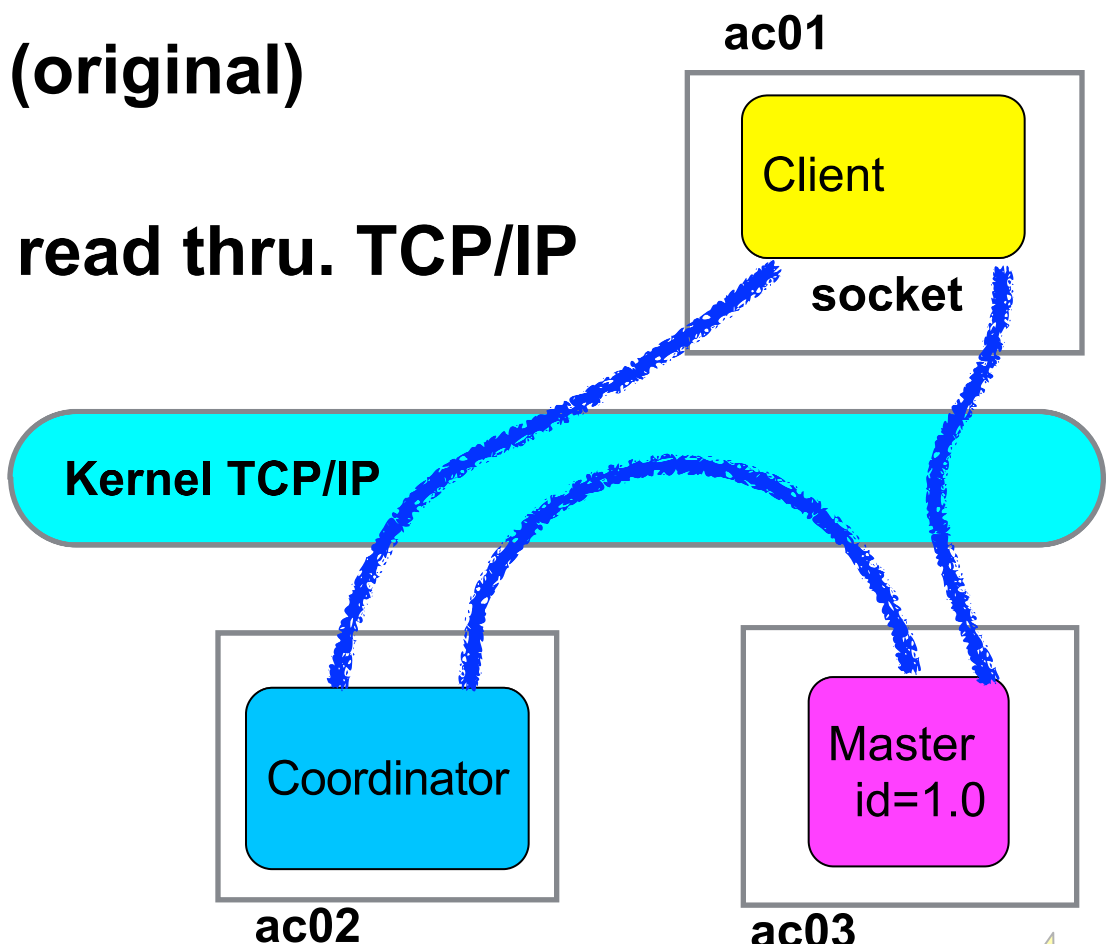


Each QSFP+ independently
configurable as ether 1 x 40G or 4 x 10G

back

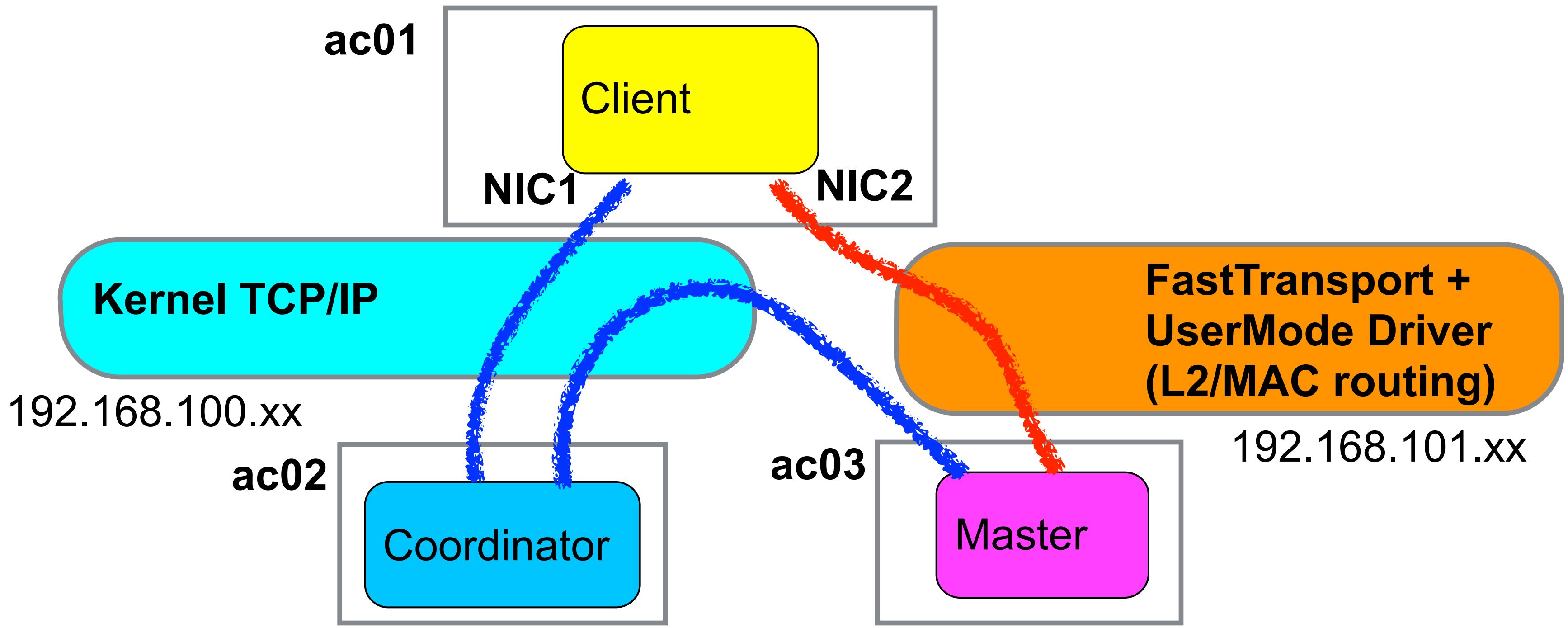
Base Evaluation

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



Improvement with User Space Driver

- User space driver only for critical path, ie. Master-Client data path
  No modification in RAMCloud code, changing startup parameter.
- Developed user mode 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

back

Development Platforms for User Space Driver

<table>
<tr><th></th><th>Summary</th><th>Performance</th><th>License</th><th>Comment</th></tr>
<tr><td>PACKET_MMAP</td><td>Implementation on the standard linux kernel</td><td>At least one buffer copy needed because a device buffer cannot be mapped</td><td>GPL</td><td>-</td></tr>
<tr><td>netmap</td><td rowspan="3">Possible to map device queue to user space</td><td rowspan="3">Feasible to realize zero-copy in user space driver</td><td>GPL/ BSD</td><td>Higher safety because user land code cannot access NIC registers directly</td></tr>
<tr><td>PF_RING / DNA (Direct NIC Access)</td><td>GPL/ BSD</td><td>-</td></tr>
<tr><td>Intel DPDK (Data Plane Developer Kit)</td><td><u>BSD</u> (GPL for kernel module)</td><td>Rather widely used</td></tr>
</table>

Our choice

back

RAMCloud w. User Space Driver

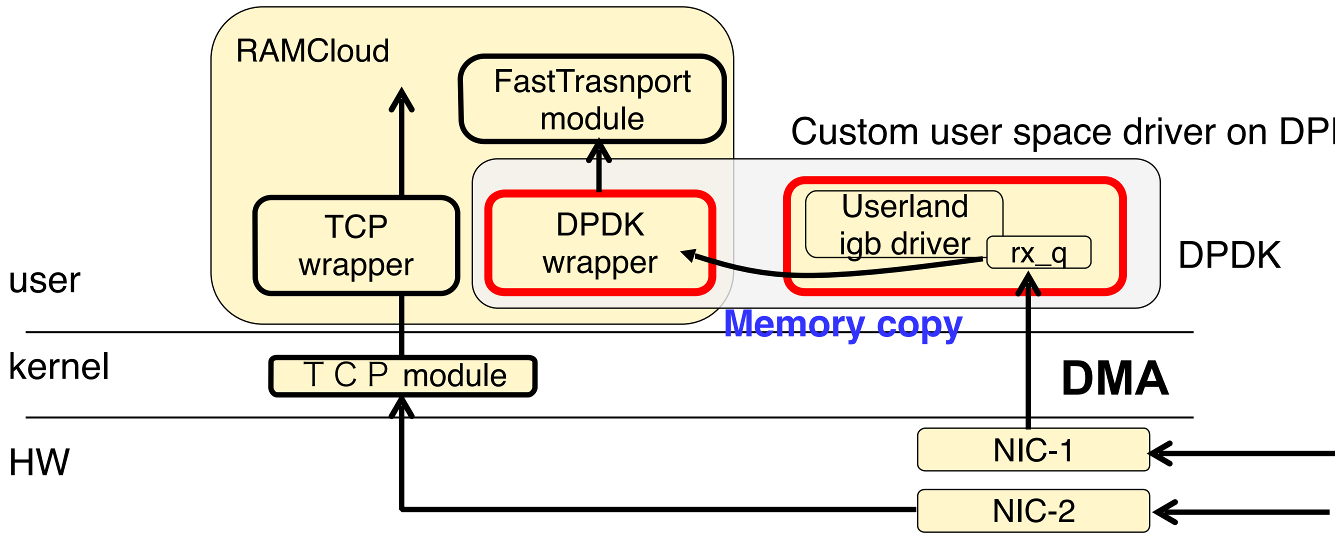


Figure. Customized transport for ATOM server (In-bound)
(almost the same for out-bound)

- 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, Store and forward LAN switch
- Average and best/worst in 100 ms period. (7000 samples in 100B read)
- Room for tuning: long tail (Max), slow write.

	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	18.2 us	17.4	43.6	5.2 MB/s	15.7 us	6.1 MB/s	Backup Enabled
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	

back

Analysis

- Considerable gap between min. and max. implies 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

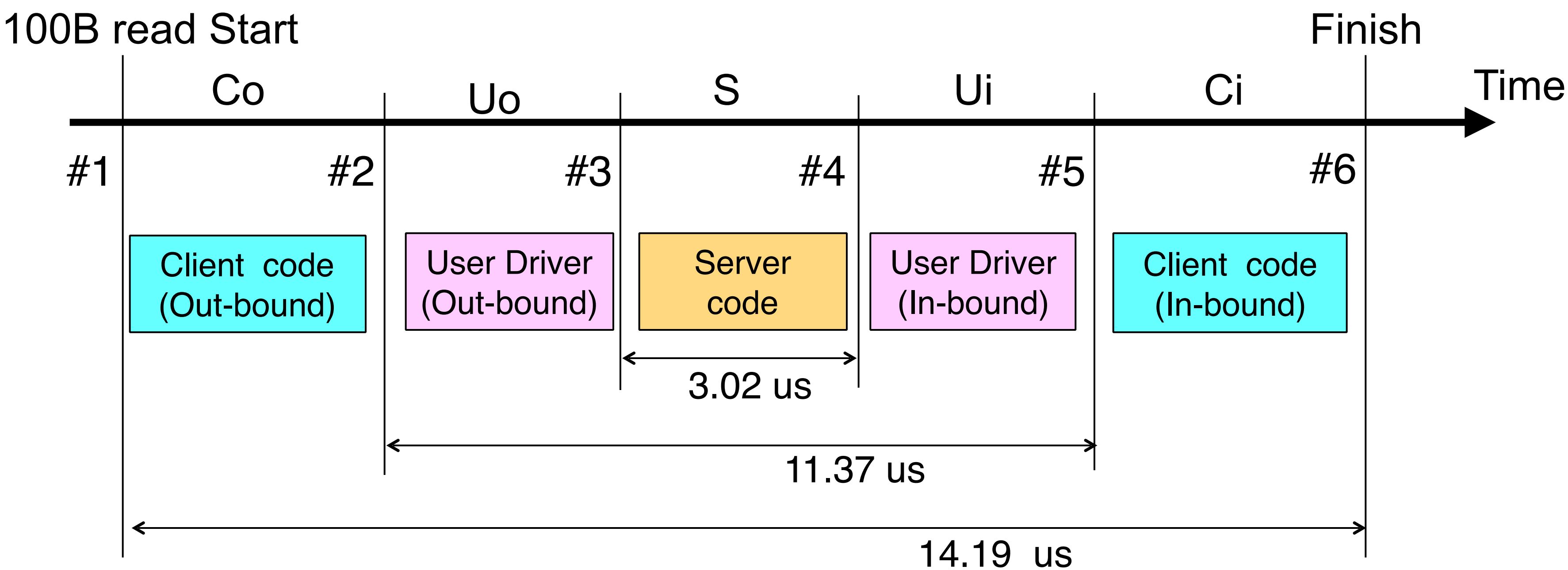


Figure. 100B read latency breakdown

Latency Breakdown: Ping

Code Segment	Elapsed	Section	Components
Client Code	0 us	Co + Ci	None: IPMI-packet is DMA transfered 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)

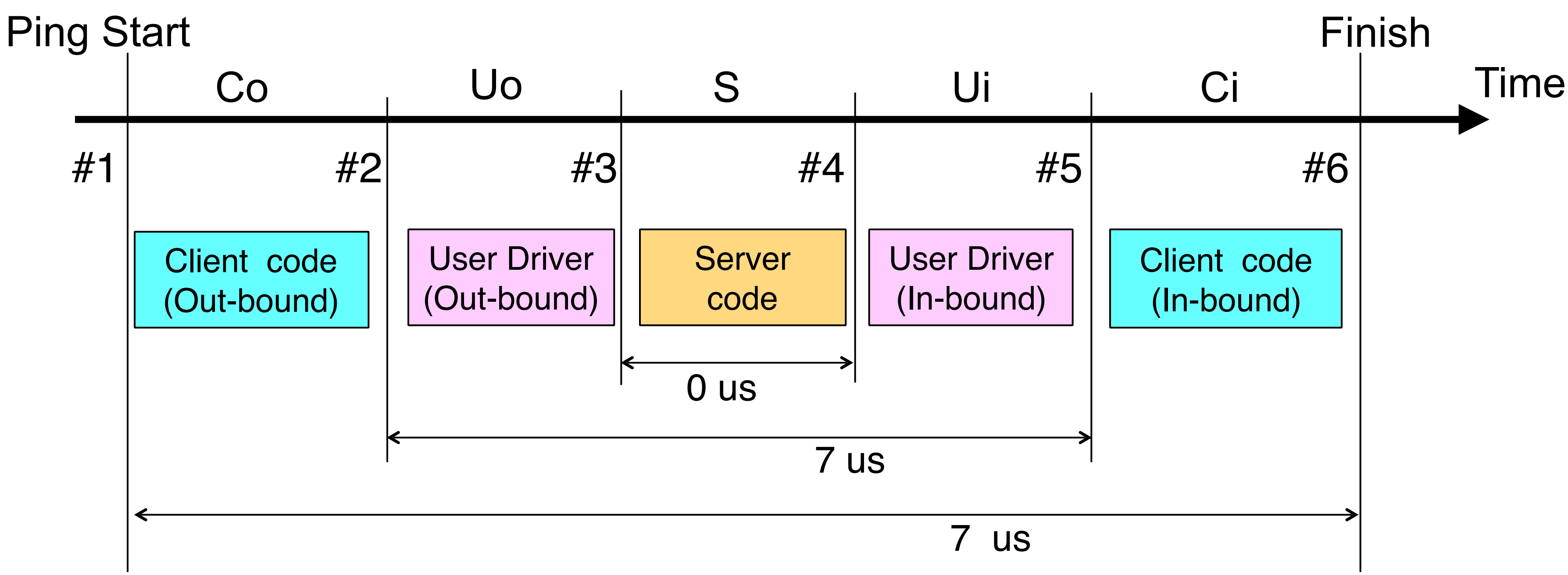


Figure. Ping latency breakdown

back

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

back

Consideration

- Large latency in user space driver (DPDK):
    - 8.35 us with 100B read, 7 us with 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 DPDK driver:
    - Further latency breakdown
    - DPDK parameter tuning
    - Cache footprint optimization??

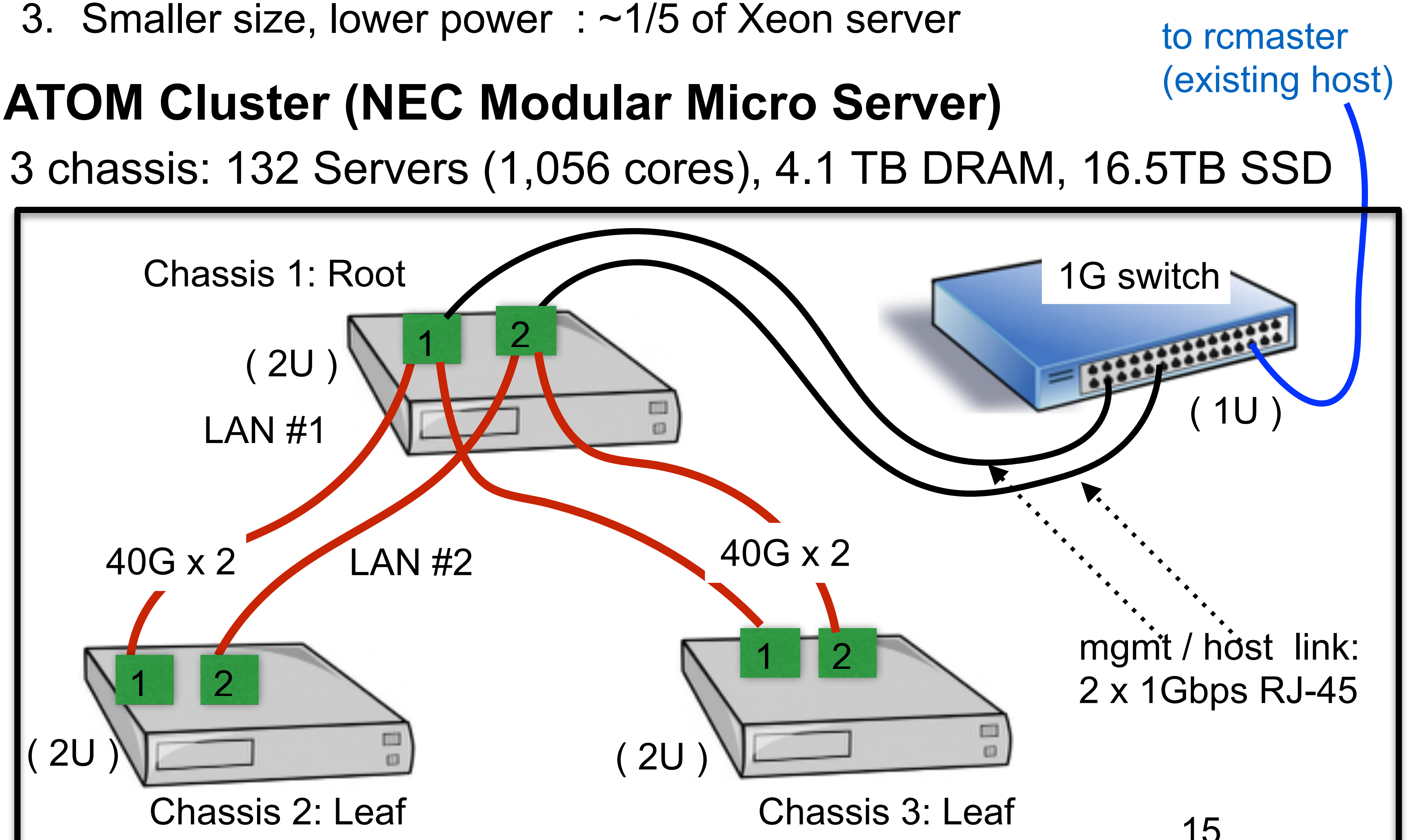
r2.07 Upd

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 user space 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