DDNの最新のLustreへの取り組みについて

DataDirect Networks, Inc.

Shuichi Ihara

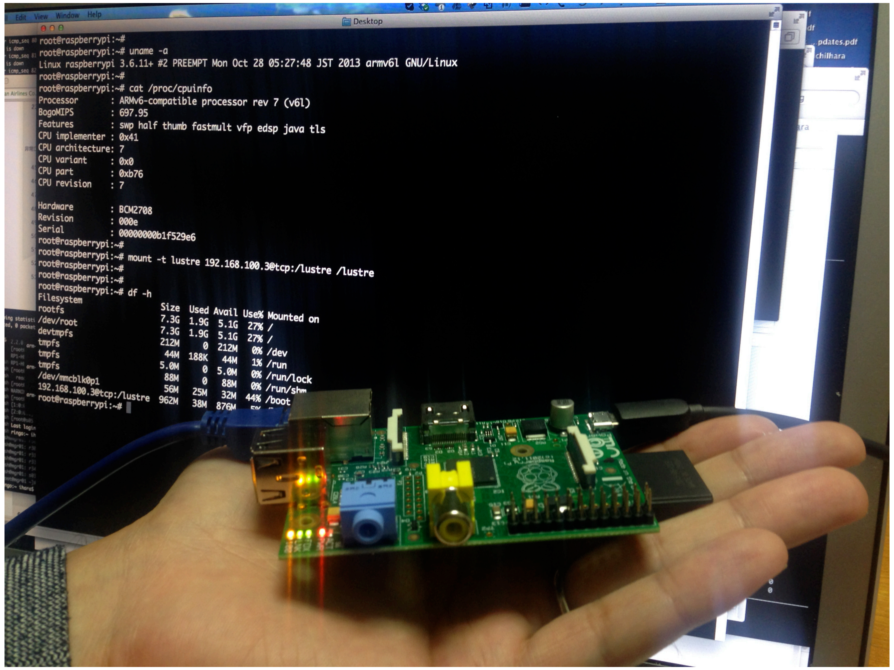
2017/11/02

DDN’s recent Lustre activities

- **DDN has been contributing Lustre community**
  - DDN is No.2 company of Lustre cods contribution.
  - DDN developed many new Lustre features and merged them into upstream Lustre.
- **Presenting at Lustre conference every year. Four presentations were selected at LAD17!**
  - Lustre on ARM servers
  - PCC(Lustre Persistent Client Cache) LU-10092
  - Lustre Audit with Changelogs LU-9727
  - Lustre/ldiskfs metadata performance boost LU-9796
- **Other activities**
  - Lustre QoS (Corroboration work with University of Mainz)
    - Selected paper at SC17 (Tuesday, November 14th11:30am - 12pm)
  - Data Archive Solution
  - Lustre Integrated Policy Engine
  - Lustre-ZFS
  - ... others

3

Lustre on ARM Servers



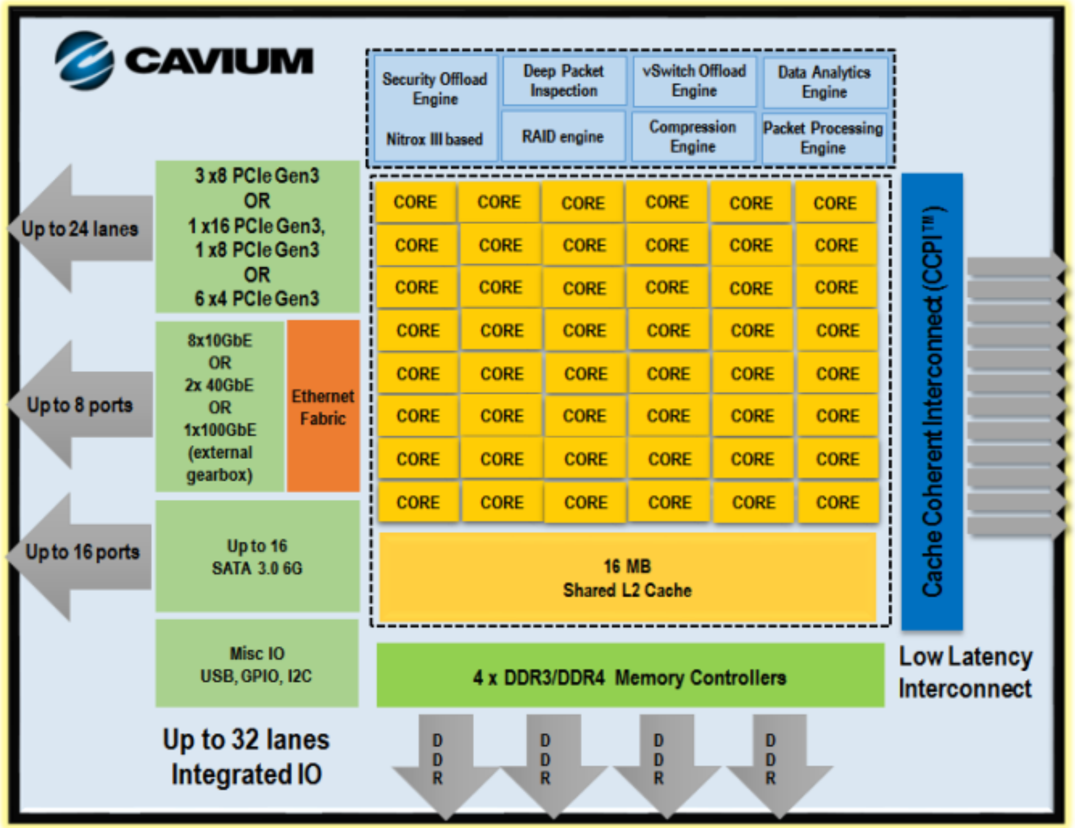
The Cavium ThunderX Architecture

▶ SoC architecture

- ISA: ARMV8

ARM

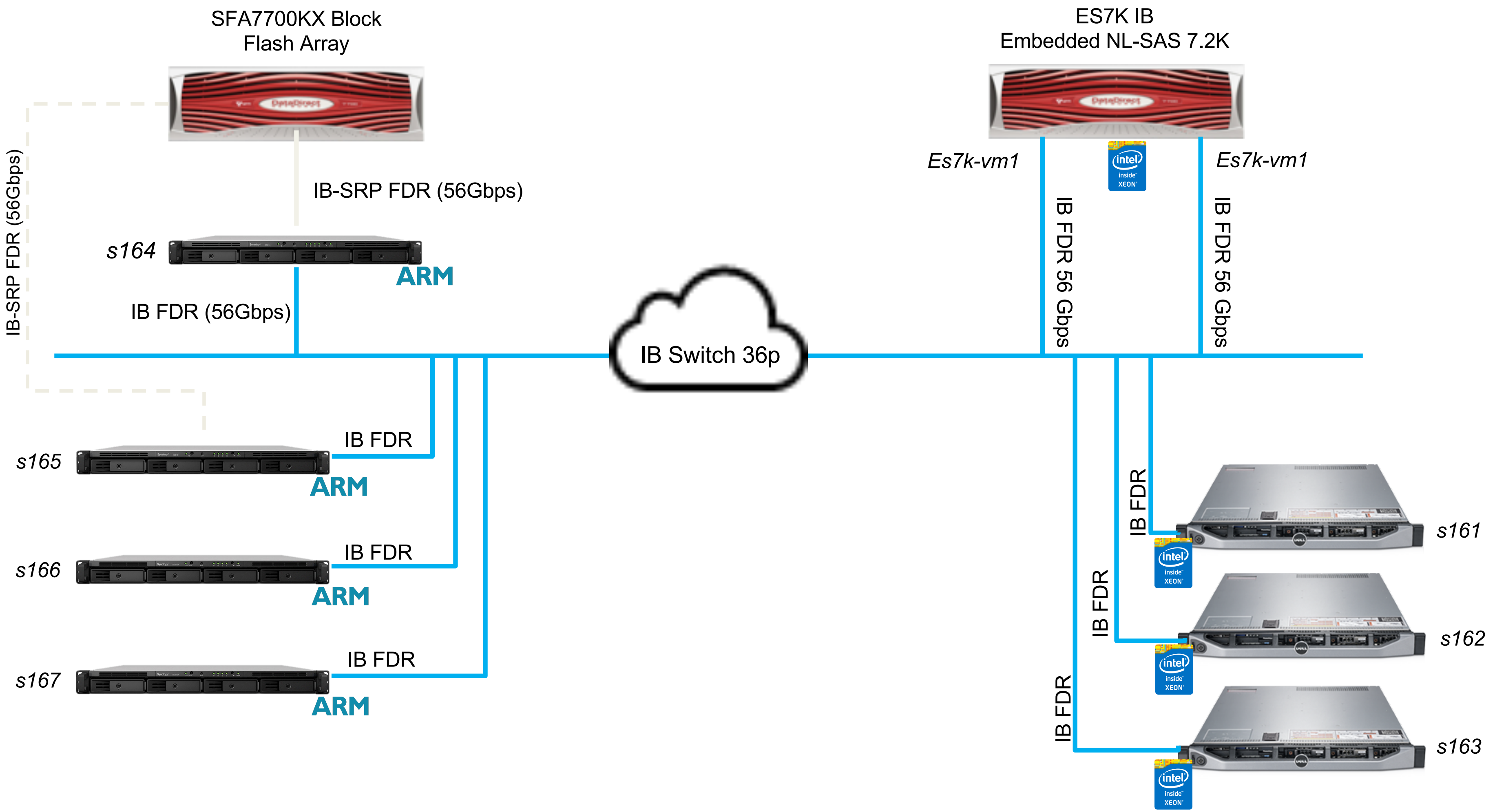
```
root@s167:/proc# lscpu
Architecture:          aarch64
Byte Order:             Little
                         Endian
CPU(s):                 96
On-line CPU(s) list:    0-95
Thread(s) per core:     1
Core(s) per socket:     48
Socket(s):              2
NUMA node(s):           2
L1d cache:              32K
L1i cache:              78KL2
cache:               16384K
NUMA node0 CPU(s):      0-47
NUMA node1 CPU(s):      48-95
```



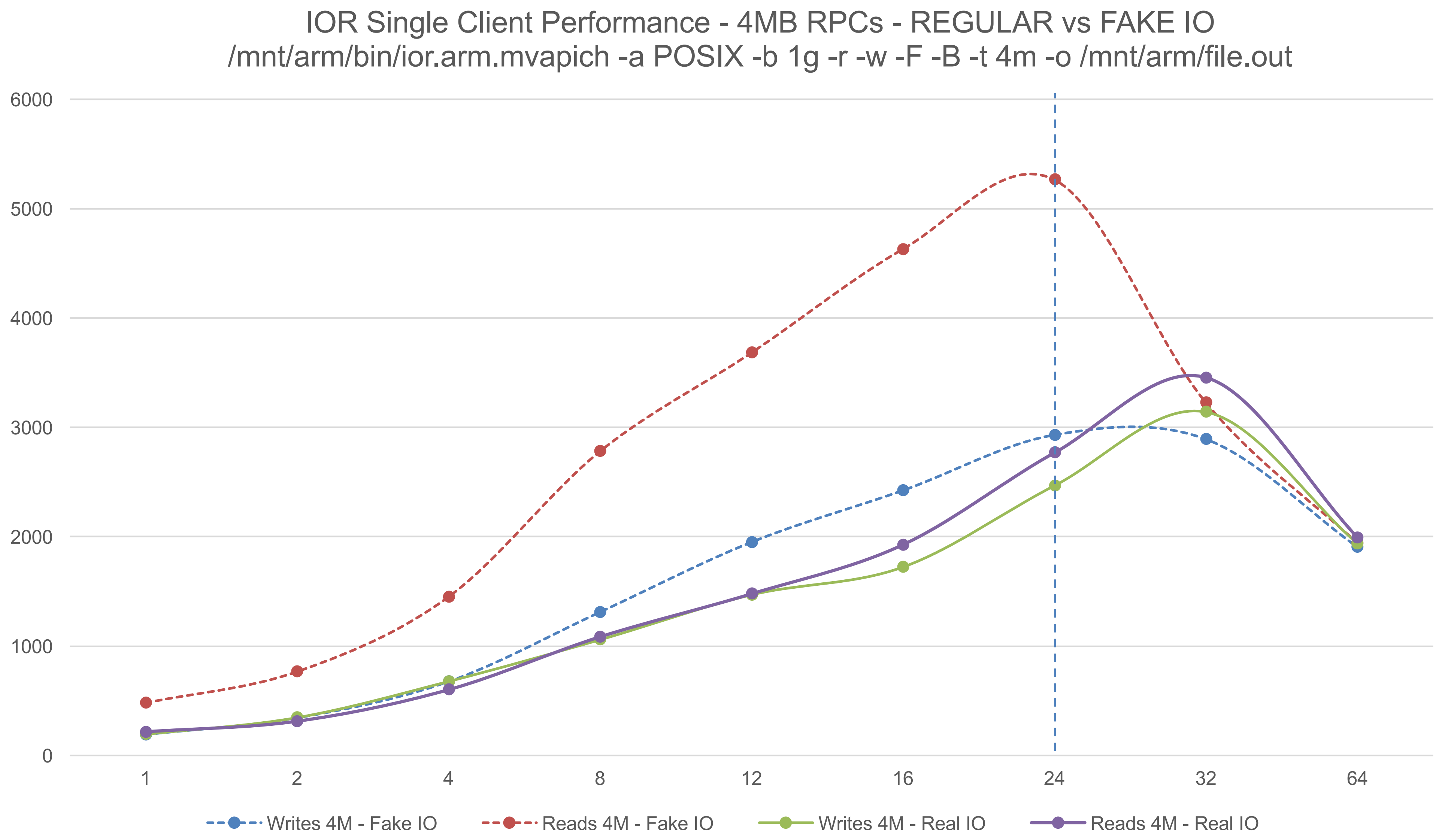
DDN Goals evaluating ARM

- Understand if it is a viable option for mid/long term future products
- Understand what's the effort necessary to make Lustre running optimally on ARM (client and server-side)
- Understand how Lustre and general I/O behaves on ARM SoC architecture
- Contribute to the community

Test Environment used for the study



IOR Single Client Performance – Multiple Threads



ARM and x86 Clients comparison IOR, multiple clients - Sequential

9

PCC(Lustre Persistent Client Cache)

NSCC-Wuxi and the Sunway Machine Family

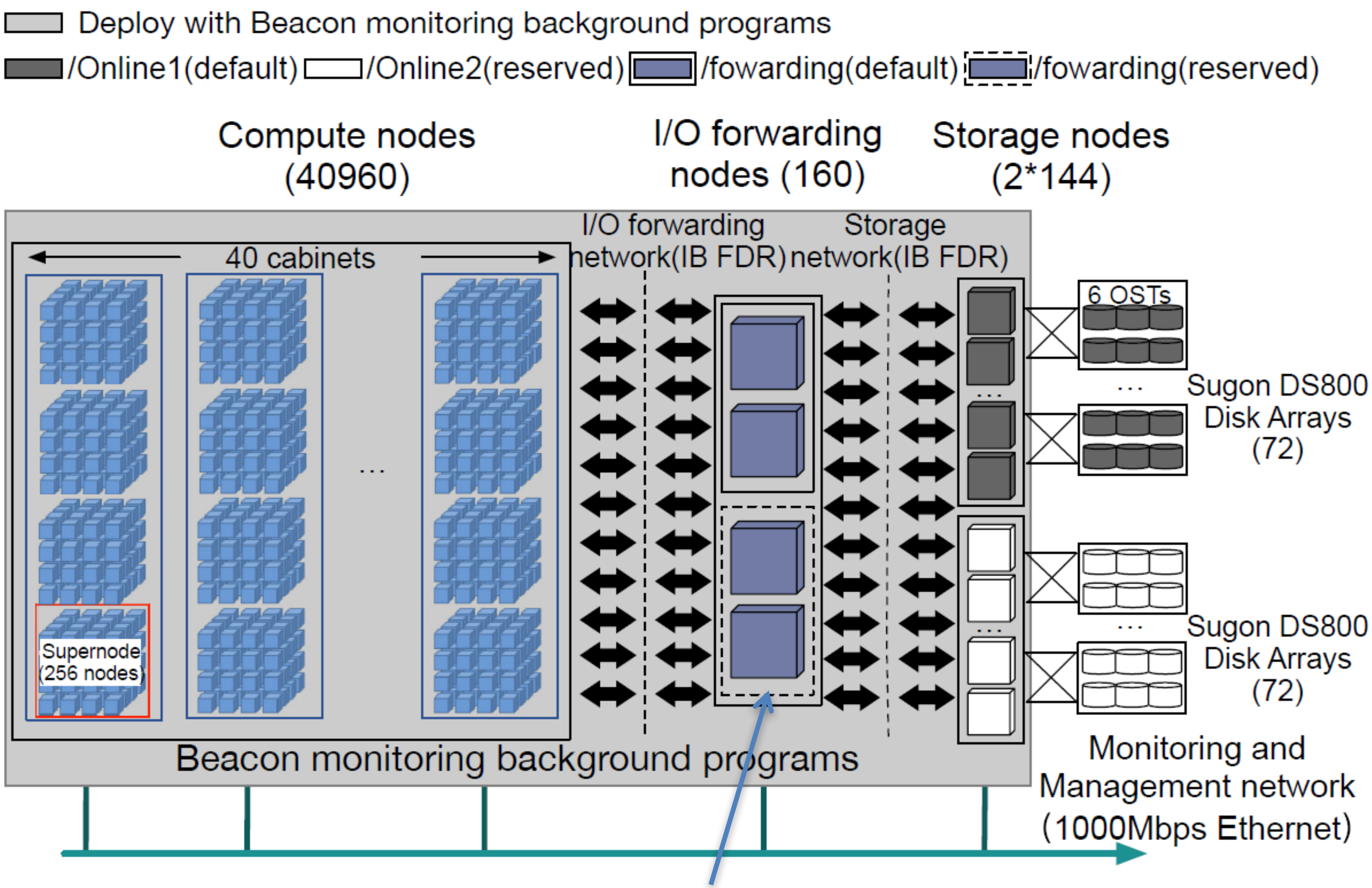
Sunway-I:
- CMA service, 1998
- commercial chip
- 0.384 Tflops
- 48th of TOP500

Sunway BlueLight:
- NSCC-Jinan, 2011
- 16-core processor
- 1 Pflops
- 14th of TOP500

Sunway TaihuLight:
- NSCC-Wuxi, 2016
- 260-core processor
- 125 Pflops
- 1st of TOP500

PCC project is collaborated by NSCC-Wuxi and DDN

I/O Architecture of Sunway TaihuLight

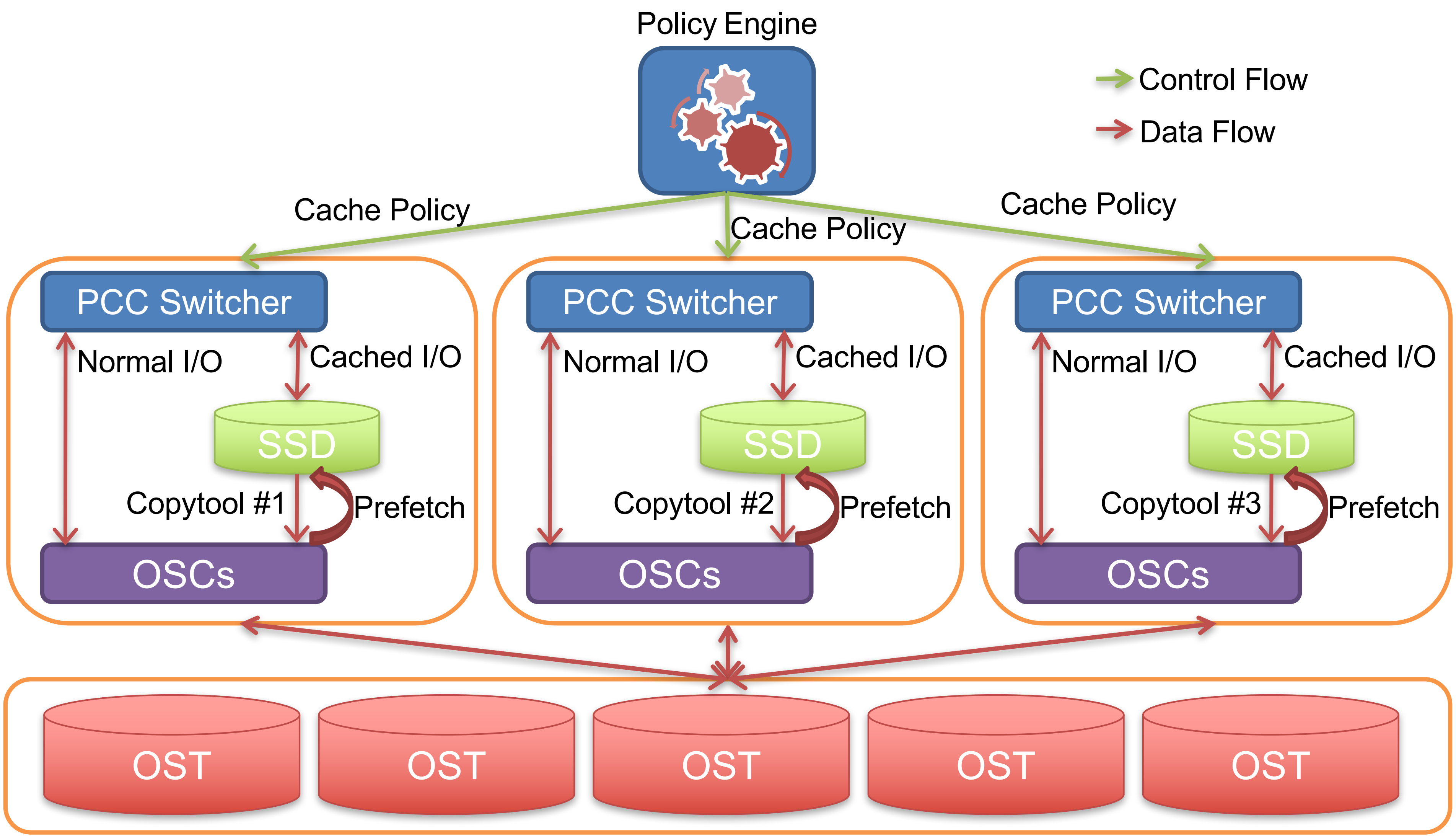


Cache on I/O forwarding nodes (Lustre clients) should be helpful

Why SSD cache on Lustre client?

- **Less overhead visible for applications**
  - No network latency
  - No LDLM lock and other Lustre overhead
- **Easier to be optimized for the best performance**
  - I/O stack is much simpler
  - No interference I/Os from other clients
- **Relatively easier than server side implementations**
  - Write support for SSD cache on server side is very difficult
  - Problems for write cache on server side:
    - Visibility when failover happens
    - Consistency when corruption happens
- **Less requirement on hardware**
    - Any kind of SSD can be used as the cache device
- **Reduces the pressure of OSTs**
    - Small or random I/Os are regularized to big sequential I/Os
    - Temporary files do not need to be flushed to OSTs

Architecture of PCC



Limitations

- **Not all applications are able to be accelerated by PCC**
  - Locality requirements of application I/Os
    - Applications shall not access the cached file through multiple clients
    - But no inconsistency will happen even the application writes the cached file on a remote client
  - Capacity of each local cache is limited
    - Size of a cached file is limited to the available space of the local cache
    - The total cached data on a single client is limited
- **Files can not be partly cached**
  - Partial cache can be implemented if HSM supports partial archive/restore
- **The total PCC clients are limited to 32 Today**
  - Only 32 different archive numbers are supported by Lustre
  - This upper limitation can be raised in the future

15

Lustre Audit with Changelogs

Need for audit in Lustre

- **Support of rich security features:**
  - authentication with Kerberos
  - mandatory access control with SELinux
  - isolation
  - etc.

⇒ **Audit as a proof of security in place**

- **Lustre outside of traditional HPC field**
- **e.g. Life science**
  - data privacy is crucial

⇒ **Audit as a regulation compliance**

Audit with SELinux

Pros	Cons
• integrated logging and auditing facility • proven	• on client side • need to consolidate

Audit with Changelogs

Pros	Cons
• integrated in Lustre • centralized • transactional	• lacks some info

Audit with Changelogs

- **Lustre activity as seen by MDS**
  - file system namespace
  - file metadata
- **Store in Changelog records**
  - internal Lustre files
- **Read from audit nodes**
  - dedicated clients

```
5 01CREAT 15:44:32.385864793 2017.07.18 0x0 t=[0x200000402:0x3:0x0]
          ef=0x1 p=[0x200000402:0x2:0x0] fileA
```

Lustre needs for proper audit

- **Identify subject of action**
  - uid/gid
  - NID
- **Record all actions**
  - open
  - close
  - xattr
  - denied accesses

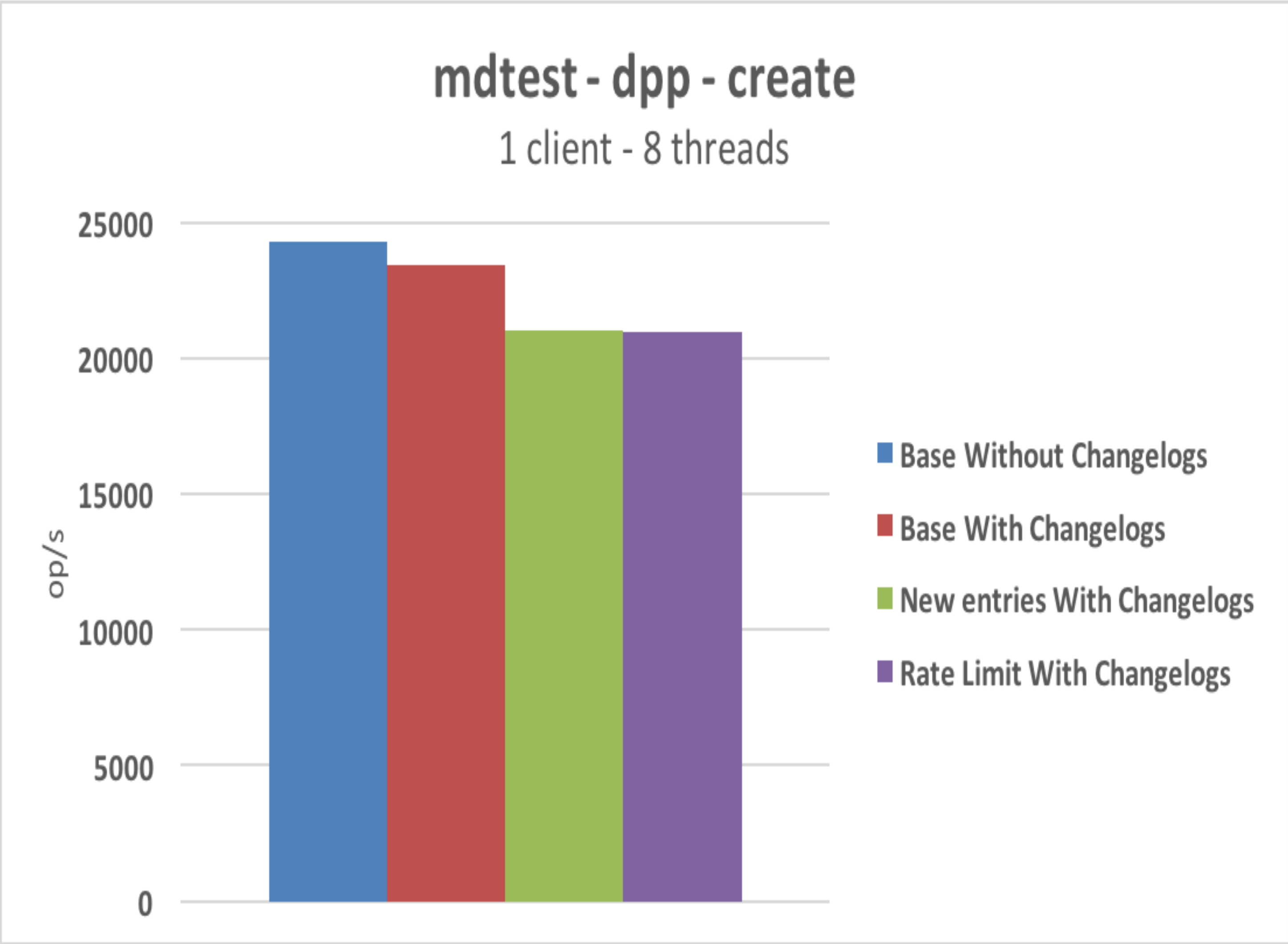
Audit with Changelogs: impact study

- **Changelogs space consumption evaluation**

	# changelog entries	changelog size
After 10 000 files created	30000	3755824
After 10 000 files read	50000	6096448
After 10 000 files removed	60000	7461440

- **… MDT**

Audit with Changelogs: impact study



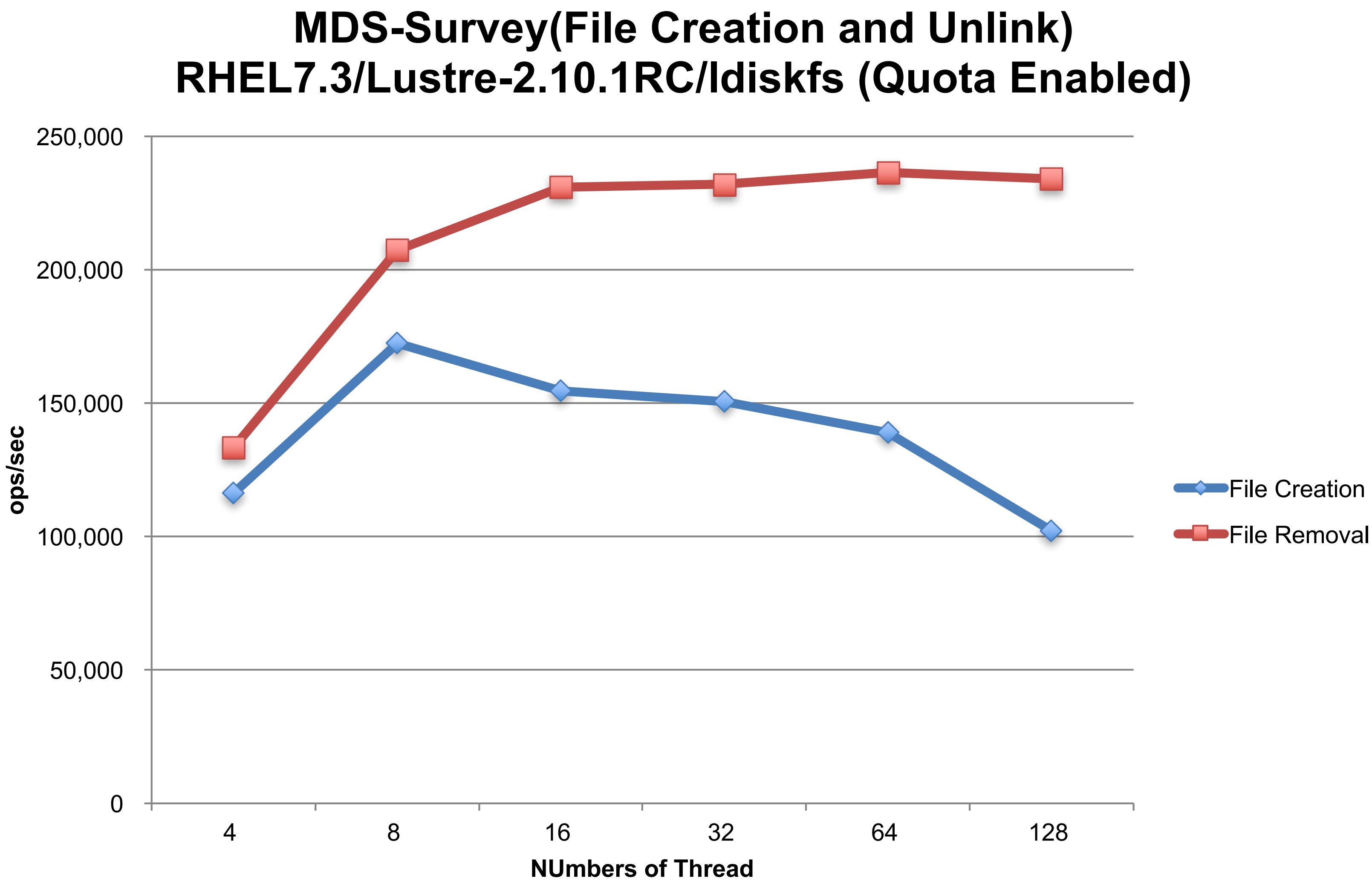
23

Lustre Metadata Performance improvement

Why is metadata performance important?

- **Lustre is general purpose filesystem for Big data**
  - 1 Million files per job are quite common with life science application
  - AI/Machine learning type of workload requires small file access with low latency. Metadata performance is one of key factors of it.
  - Lustre metadata performance has been performing well.
- **Vertical and Horizontal scale**
  - 28 (and 32) CPU cores/socket is available Today.
  - DNE helps Horizontal scale out Metadata, but needs to understand your single MDS metadata performance first.

MDS-Survey RHEL7.3/Lustre-2.10/ldiskfs



A problem on File creation under concurrency

Profiled with perf-tools during mdtest to ldiskfs/ext4

- Collected CPU costs for all functions in ext4 and jbd2
- Found heavy lock contentions on group spinlock

FUNC	TOTAL_TIME(us)	COUNT	AVG(us)
ext4_create	1707443399	1440000	1185.72
_raw_spin_lock	**1317641501**	**180899929**	**7.28**
jbd2__journal_start	287821030	1453950	197.96
jbd2_journal_get_write_access	33441470	73077185	0.46
ext4_add_nondir	29435963	1440000	20.44
ext4_add_entry	26015166	1440049	18.07
ext4_dx_add_entry	25729337	1432814	17.96
ext4_mark_inode_dirty	12302433	5774407	2.13

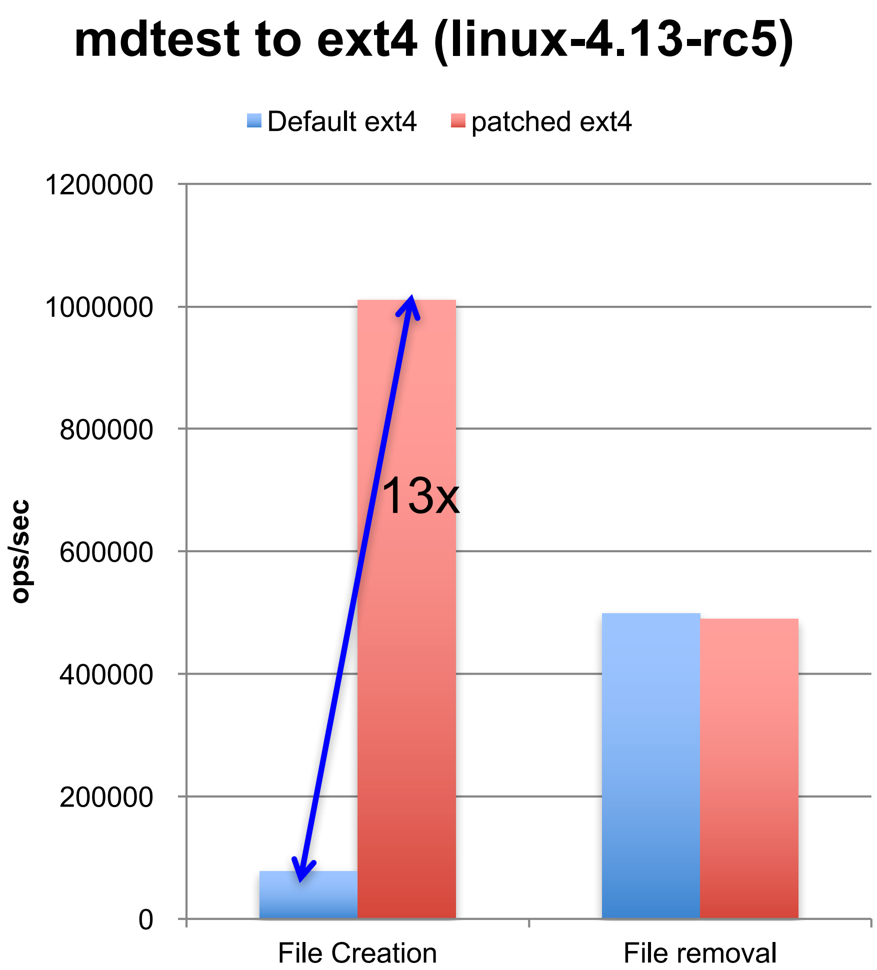
- Same contentions exist in the upstream kernel

Fix lock contentions in upstream kernel

- **Fixed and merged upstream kernel (4.14)**

  Wang Shilong (2):
  ext4: cleanup goto next group
  ext4: reduce lock contention in __ext4_new_inode

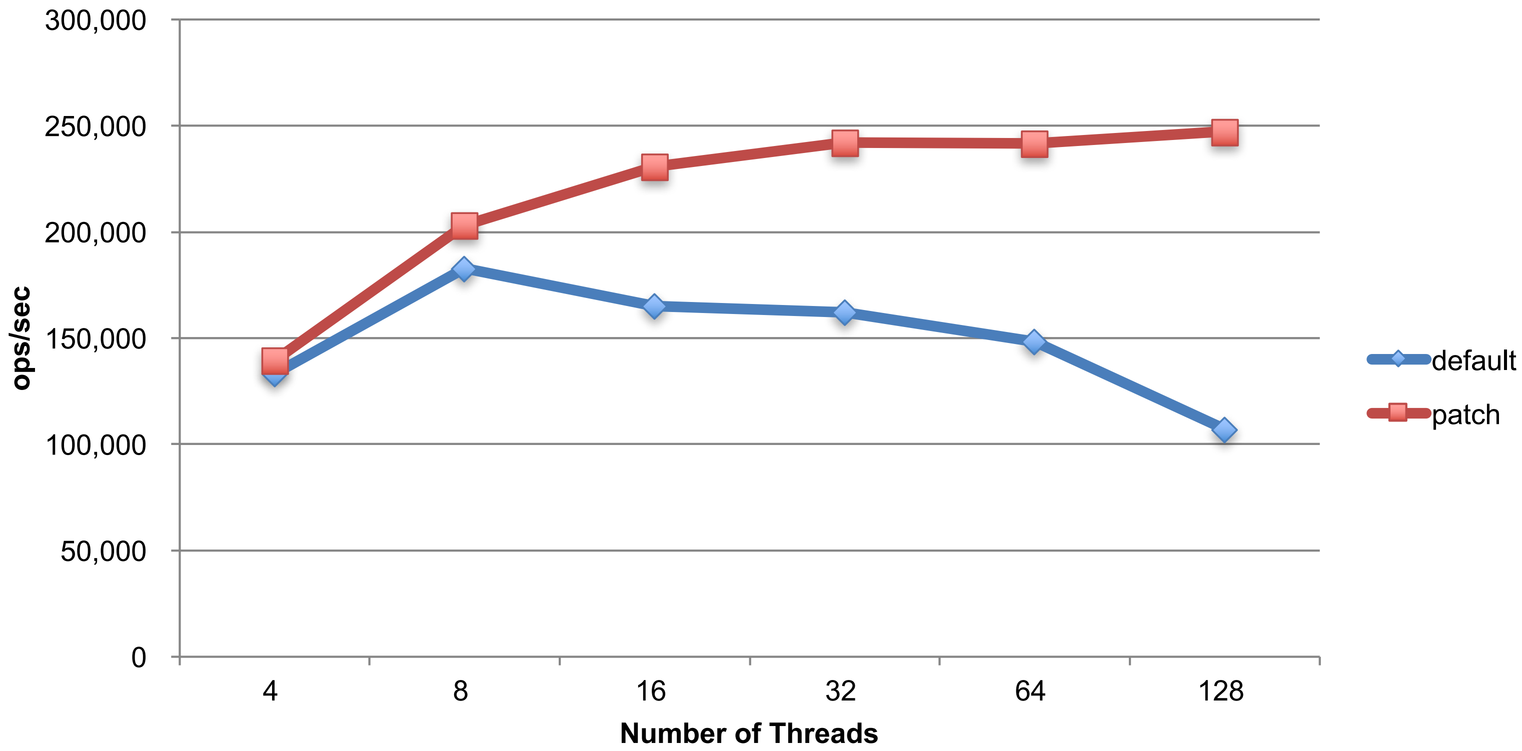
- **13x performance improvement on file creation**
  - Run mdtest to ext4 directly
  - Unique directory operations
  - Quota disabled



mds-survey on patched ldiskfs

- LU-9796: speedup file creation under heavy concurrency
- Ported patches to ldiskfs for RHEL7 kernel

File Creation :mds-survey on ldiskfs
1 x MDS and 1 x MDT(2 x RAID1 SSD)



Conclusion

- **DDN keeps investment to Lustre and contributions to Lustre community**
  - DDN Lustre R&D in Japan and China
  - Our most of developed new features comes from valuable customer feedbacks!
- **Deliver adaption and optimizations for new hardware and new technology in advance**
  - Performance lab is located in Tokyo
  - Various early testing, performance optimization are ongoing
- **Welcome Co-research and Collaboration**
  - Not only co-research, but also Alpha/Beta testing and feedback are much appreciate!