Xin Liu, Yu-tong Lu, Jie Yu, Peng-fei Wang, Jie-ting Wu, Ying Lu, 2017. ONFS: a hierarchical hybrid file system based on memory, SSD, and HDD for high performance computers. *Frontiers of Information Technology & Electronic Engineering*, **18**(12):1940-1971. http://dx.doi.org/10.1631/FITEE.1700626

ONFS: a hierarchical hybrid file system based on memory, SSD, and HDD for high performance computers

Key words: High performance computing; Hierarchical hybrid storage system; Distributed metadata management; Data migration

Corresponding author: Xin Liu
E-mail: xliu@cse.unl.edu

ORCID: http://orcid.org/0000-0003-3824-1726

Motivation

- HDD-based storage systems have huge storage space, but low bandwidth and high latency.
- With the rapid development of Exascale supercomputers and HPC applications, and an increasing number of CPU cores, HDD-based storage systems can hardly fulfill the requirements of high bandwidth and low latency.
- Most research to alleviate the I/O pressure can only work locally and optimize partial I/O.
- The advent and blossom of available storage devices makes hierarchical hybrid storage possible, which can provide huge storage, high bandwidth and low latency at the same time.

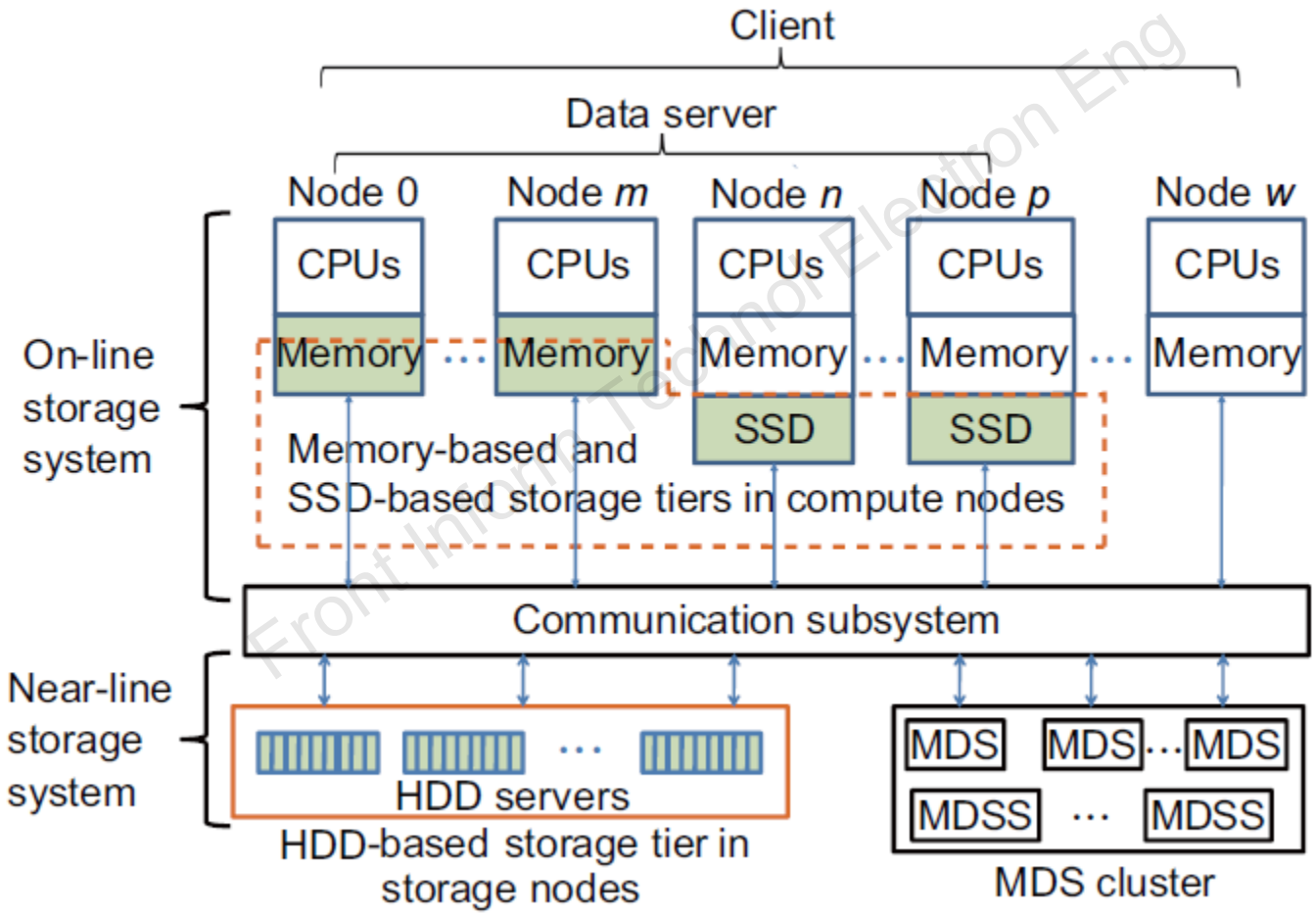
Main idea (1)

- We leverage memory and SSD in compute nodes, and HDD from storage servers to constitute a three-level storage system.
- To support increasing metadata access in supercomputers, and data migration among storage tiers, we propose a distributed metadata storage and management strategy based on User Group Sub-directory. We propose a peak-shaving MDS with a synchronous updating policy to support dynamic load balancing and the scale adjustment of metadata servers without introducing overhead of metadata migration.

Main idea (2)

- Since DRAM is a volatile storage device and the storage space is limited, we implement a dual-replicas with parallel updating and a recovery mechanism to achieve reliability, and gather multiple DRAM-based storage servers as a Group to provide larger storage space and higher parallel I/O bandwidth.
- To make as much I/O requests hit in DRAM-based storage tier, we calculate file coolness, migrate cold file to the lower storage tier for archive and prefetch files to be accessed to the upper storage tier guided by the the thresholds of the available storage space.

Architecture



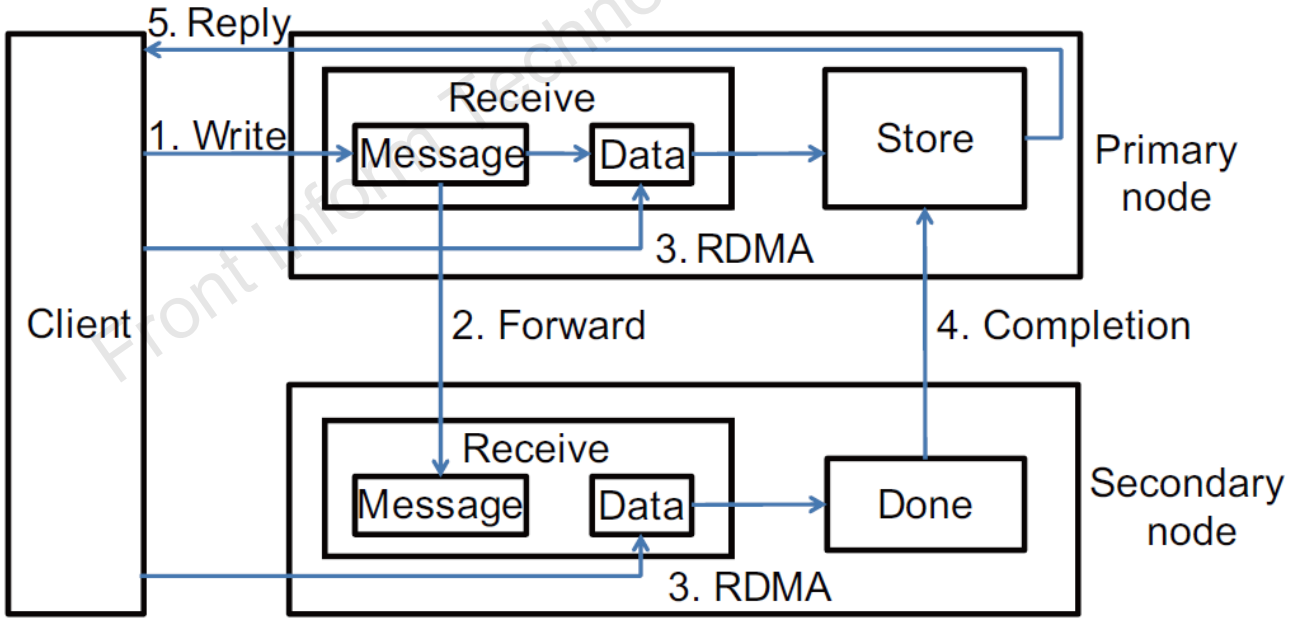
Method (1)

1. Distributed metadata storage and management based on User Group Sub-directory (Liu *et al*., 2017b).

2. Static and dynamic memory borrowing form compute nodes to utilize as much available memory; return memory resources to the compute nodes as soon as user programs require to ensure correct running of the programs.

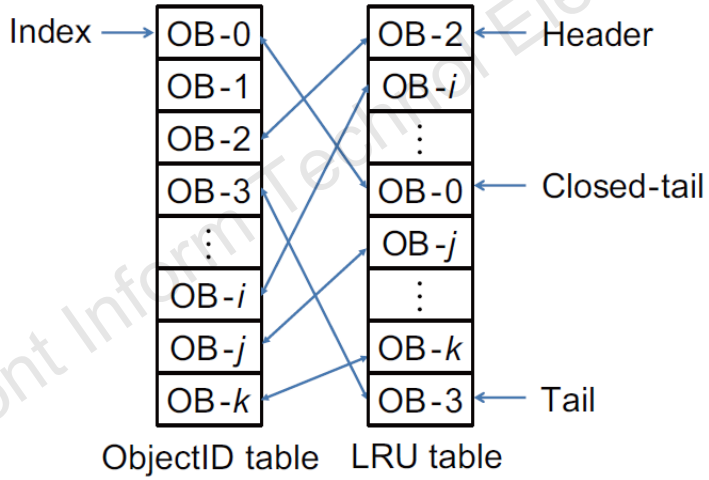
Method (2)

3. Parallel dual-replicas to avoid data transferring overhead, and recovery mechanisms for DRAM-based storage servers.



Method (3)

4. File coolness measurement for downward migration (FCM).



5. File migration policy based on the Quality of Service (QoS) method.

Major results (1)

- ONFS with three storage tiers perform much better than HDD-based Lustre.

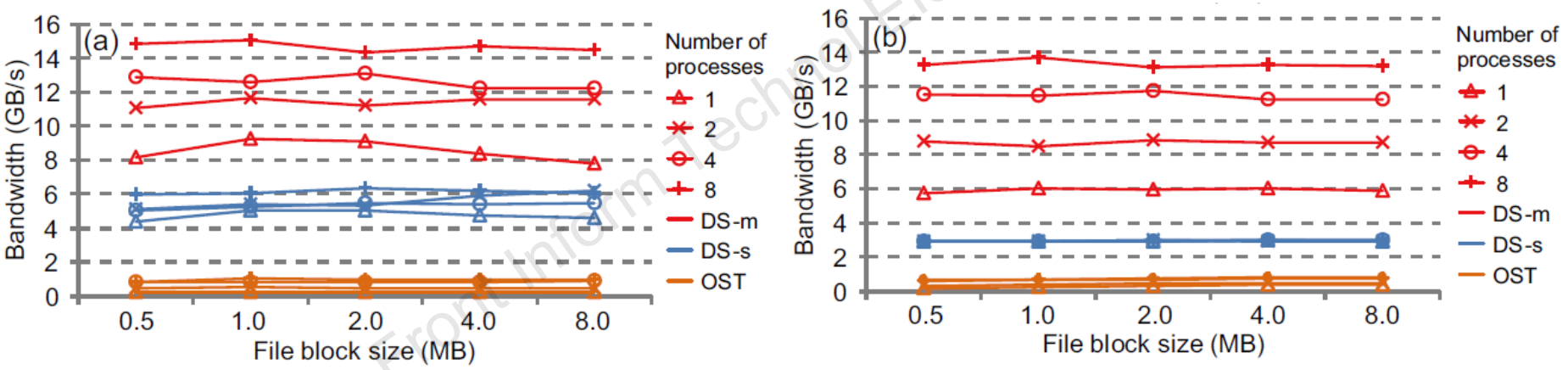


Fig. 16 Comparison of 'read' (a) and 'write' (b) performance of four data servers in each storage tier

Major results (2)

- Metadata performance increases linearly with the increase of MDS.

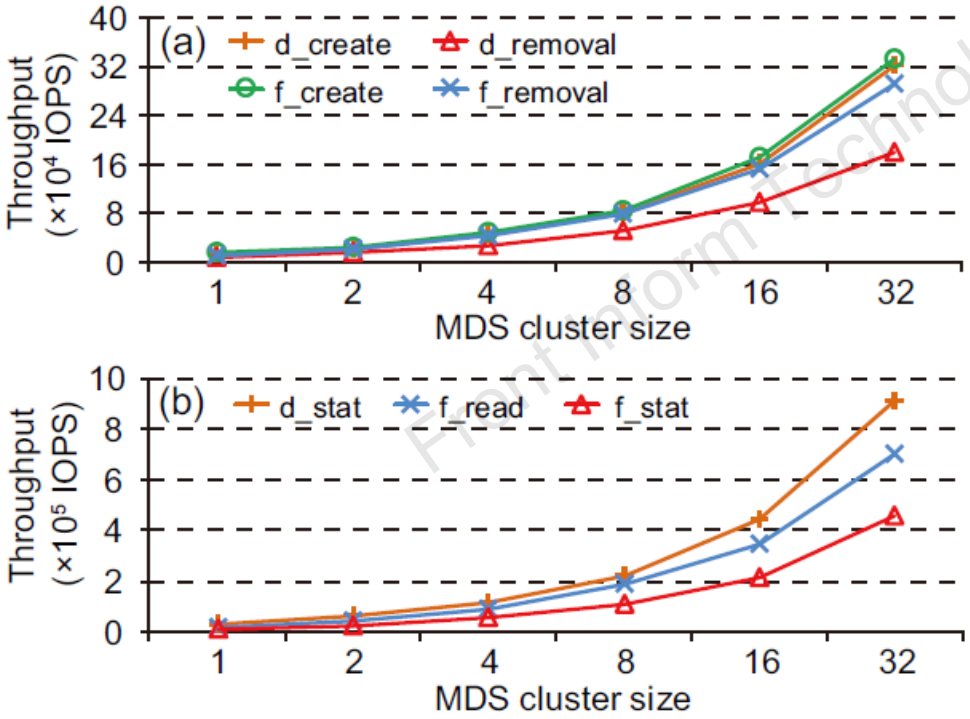


Fig. 19 Scalability of distributed metadata server clusters in on-line and near-line file system (ONFS): (a) metadata modifications; (b) metadata lookups

IOPS: input/output operations per second;
d_create: directory
create; d_removal: directory removal;
f_create: file
create; f_removal: file removal;
d_stat: directory status;
f_read: file read;
f_stat: file status

Major results (3)

- The I/O bandwidth of a typical data-intensive application (one-way wave depth migration) in ONFS is much better than that in Lustre

Table 10 Summary about OWDM applications

File system	Cache on/off	'Write' bandwidth (MB/s)	Run-time (s)	Percentage of run-time (%)
ONFS	On	1500	2927	86
	Off	1400	3552	24
Lustre	On	300	3416	
	Off	300	15 111	

Major results (4)

Table 11 Comparison of ONFS with the state-of-the-art storage systems

Storage system	Storage-media	Location	System features	Bandwidth (single node or OST) (GB/s)	Current system size	Scalability	Weakpoints (compared with ONFS)
ONFS	DRAM +SSD	Compute node	Hierarchical storage system, automatic migration	5.10	$O(N)$[3]	Scale to all compute nodes	NULL
FusionFS	DRAM	Compute node	Memory-based storage system	0.16[1]	$O(N)$[3]	Scale to all compute nodes	May impact job performance
DataWarp (CoriBBN)	SSD	Burst buffer node	Fast temporal storage system	5.90[2]	288 BBNs	Scale to limited burst buffer nodes	Manual migration
Gorden	SSD	I/O node	Localfile system in ION	2.60	64 IONs	Scale to limited I/O nodes	Limited scalability
Sonexion 3000 (Cori Lustre)	SSD +HDD	Storage-node	SSD as a cache of OST	1.40	248 OSTs	Scale to limited OSTs	Limited scalability

ONFS: on-line and near-line file system; DRAM: dynamic random access memory; SSD: solid state drive; HDD: hard disk drive; OST: object storage target

[1] The performance of FusionFS is tested with HDD in the paper. If tested with DRAM, it can perform much better with a speed larger than 0.16 GB/s

[2] The performance of DataWarp is the parallel bandwidth of four SSDs in the burst buffer node

[3] N represents the number of compute nodes in TH-1A and Intrepid

Conclusions

- We developed a hierarchical hybrid file system called ‘ONFS’, which manages three storage tiers in a unified namespace.
- We effectively borrow the underutilized memory in compute nodes to construct the DS-m storage tier that is located closer to applications.
- Experimental results reveal that benchmark tests on ONFS can achieve nearly 6-fold speedup and typical data-intensive applications on ONFS can gain around 6.35-fold speedup comparing with Lustre.