Use of Lustre and ZFS on Hadoop

Anvika, Tanmay Rawat
Department Of Software Engineering
S.R.M. University

Introduction:

- Computer Science now witnesses innovation by combining the best of different components.
- It has transformed into a science that innovates by integrating different technology.
- This is what lies the basis for the proposed work- using ZFS in combination with Lustre and HDFS in Hadoop.

HDFS: An Introduction

- HDFS file system was designed to be a scalable, fault-tolerant, distributed storage system that works closely with MapReduce.

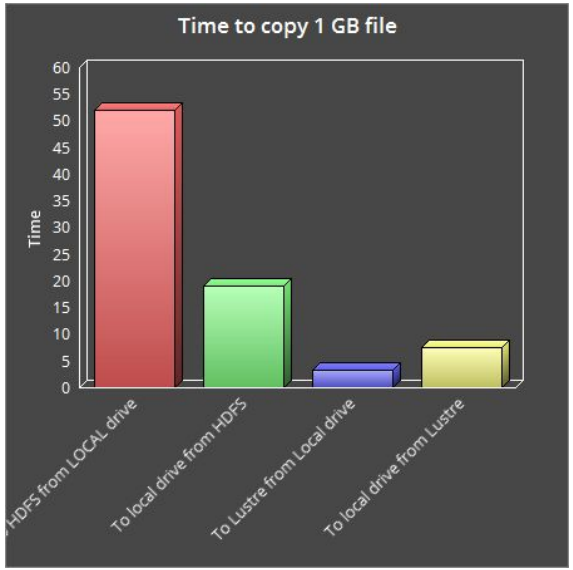
- **Features:**
  - Rack Awareness
  - Minimal data motion
  - Health check of the file system
  - Roll-back
  - Redundancy
  - Highly operable

Issues with the use of HDFS:

- HDFS is not a **POSIX-compliant file system,** and once data is written it is not modifiable.
- **Uncached data**: When the read load is random, seek-intensive or partial-block, the lack of data caching is a high cost performance deficit.
- **Slow file modifications** : Changing a small part of a file requires that all file data be copied.
- Data replication costs time
- Moving computation is cheaper than moving data
- Large block size implies fewer files.

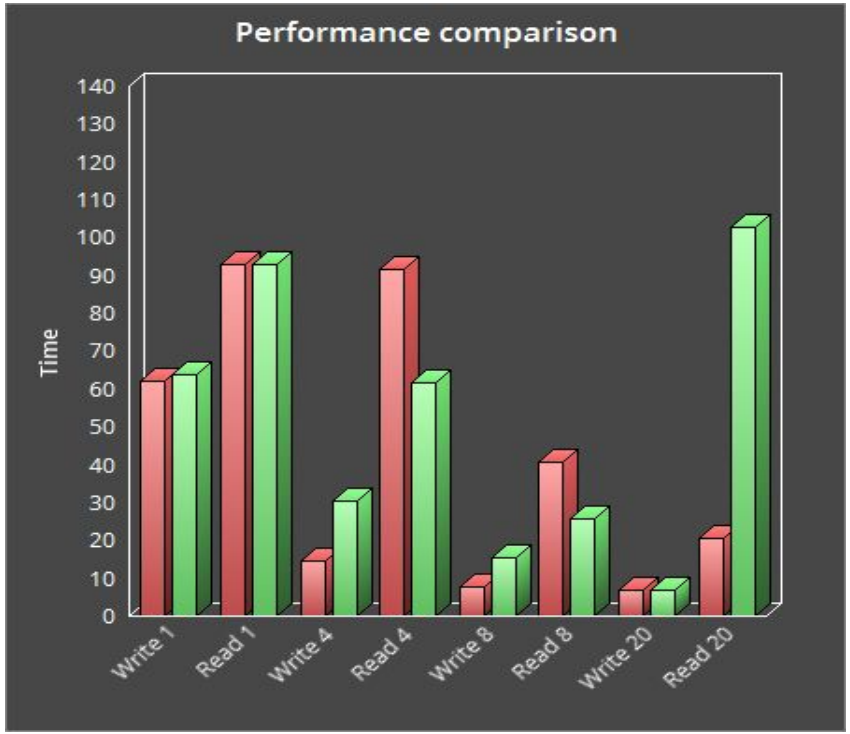
Cont’d

- It’s behaviour outside of Hadoop is not clearly predictable.
- HDFS is not a general-purpose file system.
- Before Hadoop jobs can run, the input data must be transferred into HDFS. These required transfers result in potential duplication of data as well as a time penalty for the additional data swaps.



Mapping tasks finish at different times.

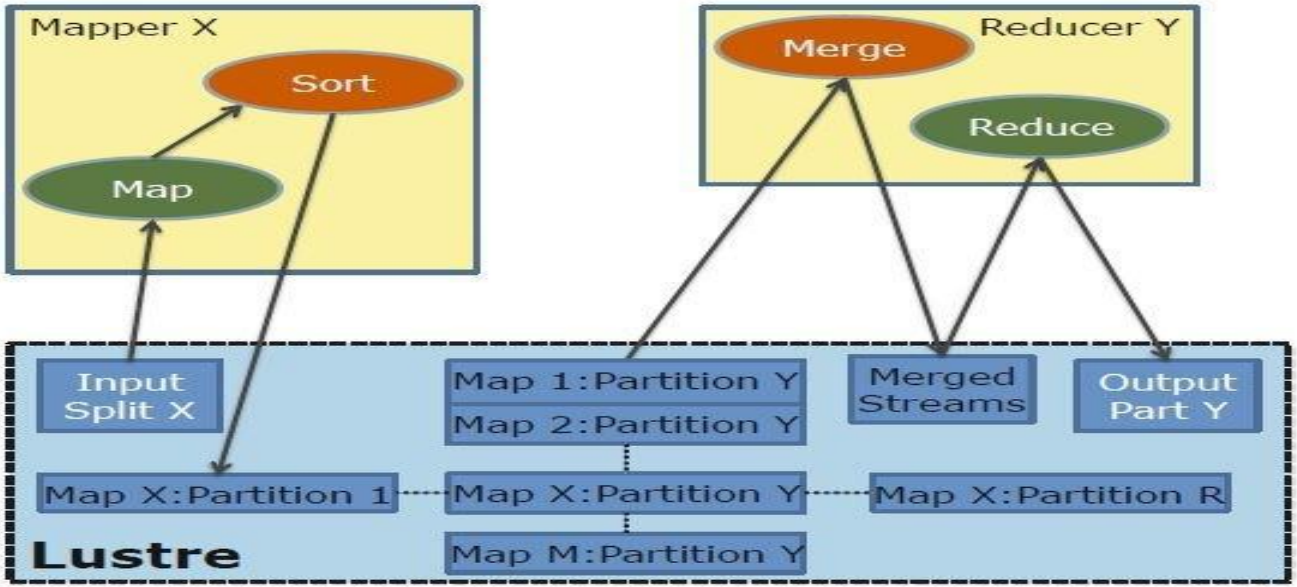
Replacing HDFS with Lustre as the underlying file system for Hadoop compute clusters in HPC environments can, result in significant improvements in system performance and cluster efficiency, or potentially offer a lower overall system cost.



The clusters are designed to achieve similar performance on both systems. In our test cases, we have seen that the performance of Map/Reduce on Lustre is approximately **three times faster** than Map/Reduce on HDFS.

Therefore, to achieve performance that equals Lustre, a HDFS cluster would require three times more compute nodes.

How Do they Work?



Advantages:

- Easier storage management (fewer storage nodes, more sophisticated management tools)
- Freedom from restrictions of data locality
- Easier loading/unloading of data to/from the system
- Flexibility to easily run Hadoop and other types of computing tasks in the same environment

What is ZFS ?

- It is a combined file system & logical volume manager designed by Sun Microsystems Inc. built with data integrity in mind.
- Implemented as Open Source Software .
- The ‘ZFS’ was originally called Zettabytes File System but as of today it stands for nothings.
- It is a 128 bit file system.
- It can store upto 256 Quadrillion Zettabytes (2^{70} x 10^{15} bytes).
- Maximum File Size is 16 Exabytes.

Why ZFS in Lustre ?

- The Lustre filesystem *ldiskfs* is limited to an 8 TB max file system size and no surety of data integrity
- ZFS is very helpful in providing data integrity to Lustre.
- All checksums are done in server memory, so catches errors like phantom writes, misdirected reads or writes, DMA parity errors, accidental overwrites.
- In Lustre, Checksumming is done by the Lustre client on the application node. It detects any data corruption in the network between the application node and the storage system.

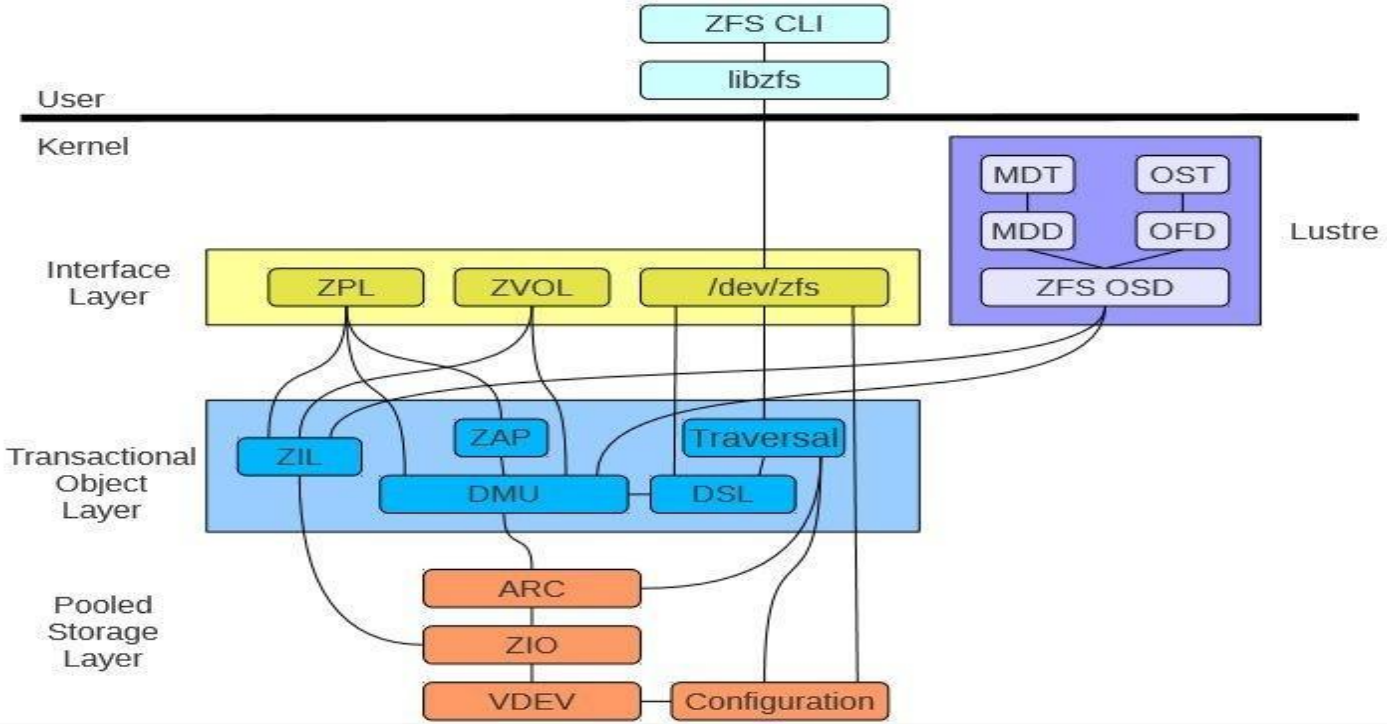
Cont’d

- Self Healing Capabilities
- Copy-on-write - Never overwrites existing data.
- Pooled Storage & Scalability
- Hybrid Pool – Supports addition of many devices like SSDs
- Improved administration -detects and reports data corruption.
- Snapshots & Compression - Transparent compression increases total usable capacity.

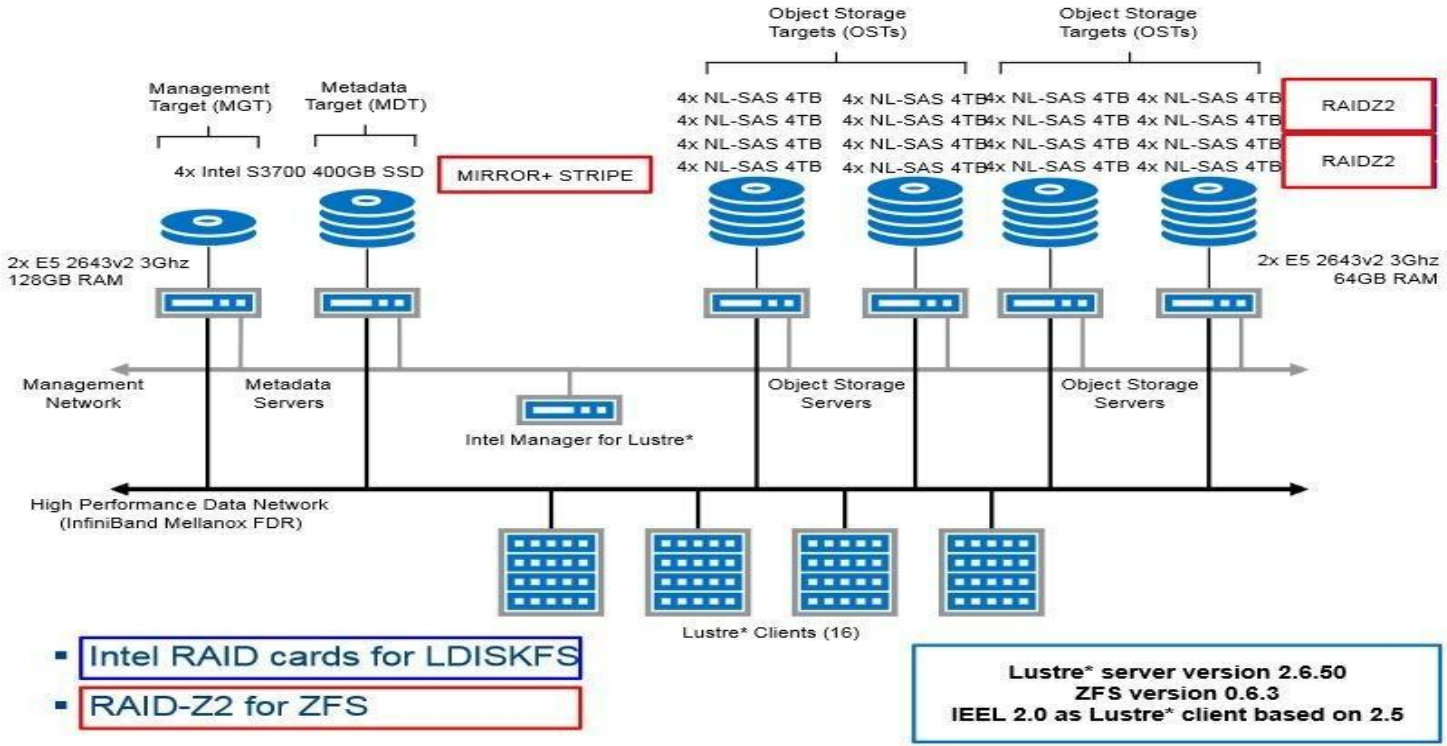
Lustre + ZFS Interaction:

- Lustre targets run on a local file system on Lustre servers.
- Object Storage Device (OSD) file system can be ZFS
- Lustre targets can be different types like ZFS, Idiskfs, hybrid of both
- Lustre clients are unaffected by the choice of OSD file system.
- Intel, Lawrence Livermore National University are already the testing the combination

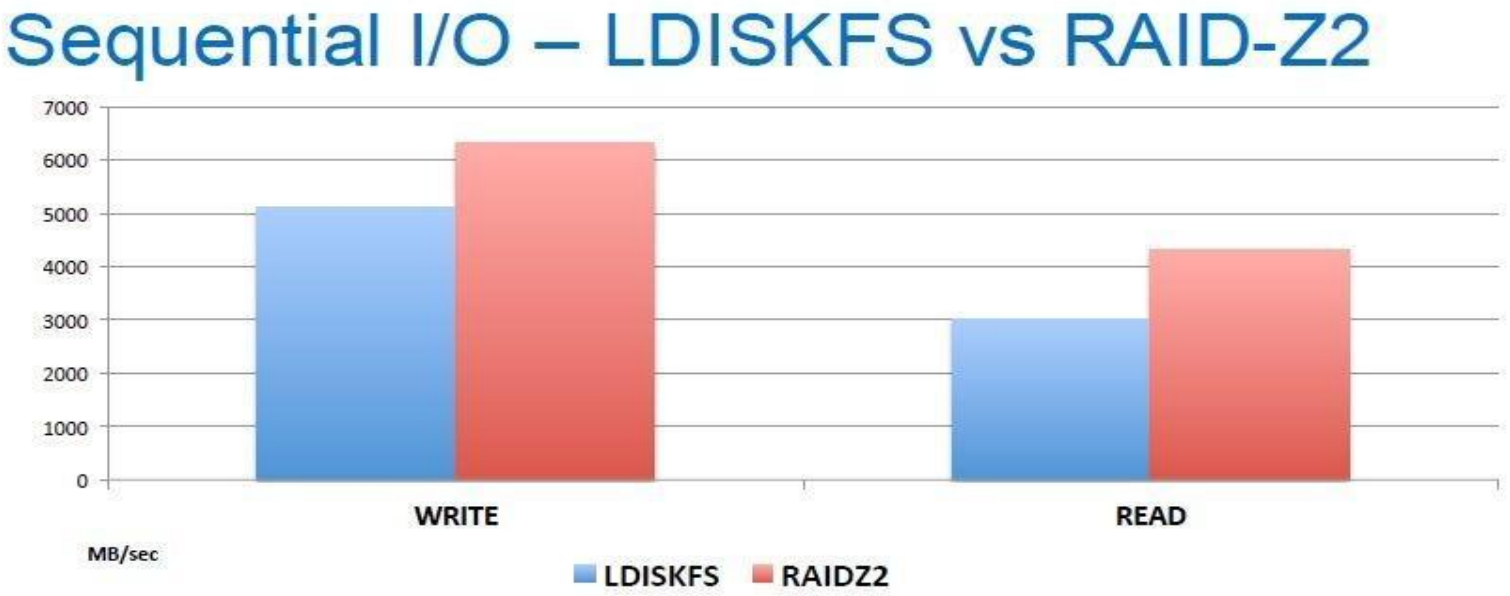
LLNL's method:



Intel's method:



Performance of ZFS:



Comparison between the file systems:

Features	HDFS	LUSTRE	ZFS
POSIX File Permissions	*No*	*Yes*	*Yes*
Time Stamps	*Yes*	*Partial(Not to clients)*	*Yes*
Access Control Lists	*Yes*	*Yes*	*Yes*
Checksums	*Yes*	*Partial(over networks)*	*Yes*
Snapshotting	*No*	*No*	*Yes*
Encryption	*No*	*No*	*Yes*
Copy-on-write	*No*	*No*	*Yes*
Volumes Resizable	*Yes*	*Yes*	*Partial(No Reduction)*
Variable file block size	*Yes*	*No*	*Yes*
Transparent Compression	*No*	*No*	*Yes*
OS Support	*Solaris, Windows, Mac OS/X, Linux, BSD*	*Linux, Mac OS/X, BSD*	*Linux, Mac OS/X, BSD, Solaris*

Working:

- There is DMU (Data Management Unit) between ZFS and disk(s) pool which does the transactions for ZFS.
- ZFS says to DMU - I want to perform list of these operations in order to rename the file. Do all of them, if you cannot do all of them then do none of them.
- DMU take list of all operations (steps) and create transactional group.
- DMU performs transactional group on pool, so operations from are done "all or nothing”.
- DMU also doesn't overwrite existing data, so FS is always consistent.

Cont’d

- DMU acts as a master thereby controlling all the disks and the operations they perform and hence they act as slaves.
- Lustre is used as an intermediary between the Map and Reduce phase to remove the overhead generated by HDFS.
- Lustre helps in the shuffle phase during which it transports data between the Map and Reduce phases.
- ZFS then stores the data after computation and through various techniques of scrubbing and checksum maintains high data integrity.
- ZFS prevents discarding of erroneous data by a process of resilvering.

Advantages:

- ZFS uses an Adaptive Replacement Cache (ARC), rather than a more traditional Least Recently Used (LRU) cache.
- L2ARC is the second level of the ZFS caching system which significantly increases read speeds for files
- ZIL accelerates synchronous transactions by using storage devices like SSDs.
- When data is overwritten on ZFS, the new data is written to a different block rather than overwriting the old data in place.
- Checksum
- Compression
- Scrubbing

Glossary:

- Adaptive Replacement Cache (ARC)
- Level 2 Adaptive Replacement Cache(L2ARC)
- HDFS: Hadoop distributed file system
- ZFS: Zettabyte File System
- LLNL: Lawrence Livermore National Laboratory
- ZIL: ZFS Intent Log

THANK YOU!