Lustre
Futures for HPC Storage

Peter Braam

peter@braam.io

2017-10

Contents

- Lustre 1998 - 2017
- Storage Tier Hardware Technology
- IO in big US HPC Systems
- Using IO in HPC
- New Software Developments
- Challenges and Conclusion

Speaker: introduced Lustre and other ideas. Presently independent researcher. Focus on future I/O, SKA telescope.

Lustre 1998 - 2017

A few thoughts & facts

Lustre has delivered:

1. interesting work to many 100’s of people
2. business to dozens of companies
3. a cornerstone to HPC infrastructure

The commitment of the user, business and developer community to Lustre has created this value. I hope everyone will experience a sense of purpose they have contributed.

This community has shaped my life – completely unexpectedly, beyond my wildest dreams. Thank you!

What wasn’t done?

The Lustre object servers could have shaped the cloud

The metadata approach was too traditional

However: this was a mandate to focus
this focus was key to Lustre’s success

Missed opportunities

A complete re-write:

- in a modern language
- with user space servers
- new abstractions (containers, write back caches)
- 100x lower maintenance

Good usability:

- dozens of nearly useless GUI's
- impossibly difficult configuration and tuning
- only highly skilled vendors deliver great Lustre systems

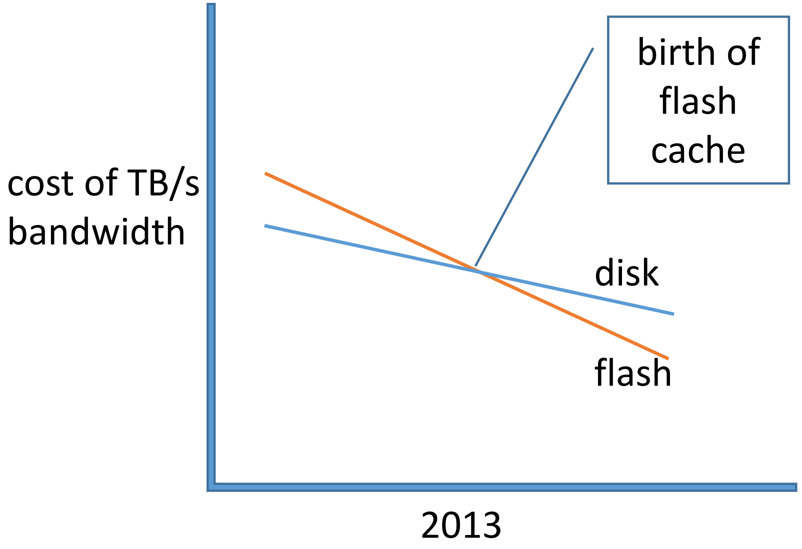
Future of HPC Storage

Storage Tiers

Tier Technologies and Parameters

	High Bandwidth Memory	RAM	NVRAM XPOINT / PCM / STTRAM	FLASH	DISK	TAPE
BW Cost $/ (GB/s)	$12	$10	>$10	$200	$2K	$20K
Capacity Cost $/GB	$9.60	$8	<$8	$0.3	$0.02	$0.01
Node BW (GB/sec)	1 TB/s	100 GB/s	<100GB/sec	20 GB/s	5 GB/s	
Cluster BW (TB/sec)	10-100 PB/s	100 TB/s	<100TB/sec	5 TB/s	100 GB/s	10’s GB/s
Software	HDF5		DAOS	DDN IME Cray Data Warp Lustre	Lustre / GPFS Campaign Storage	Archive & Campaign Storage

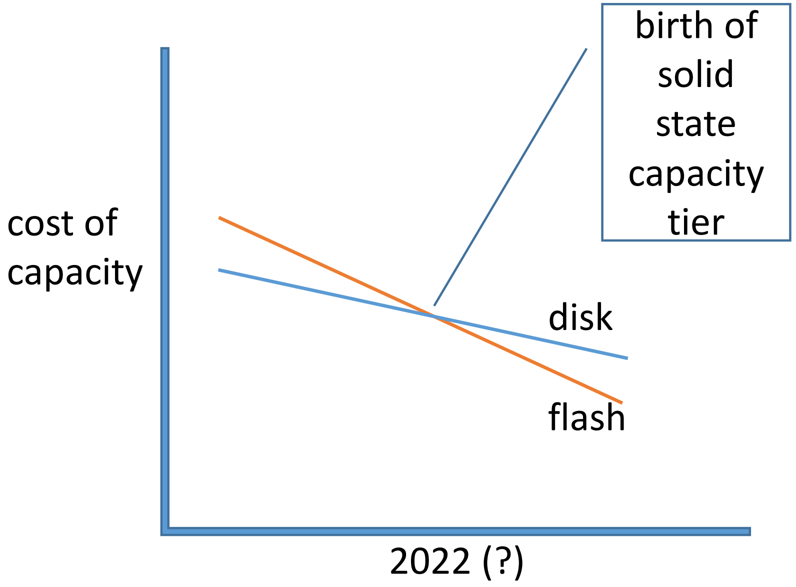
Economy



Modeling:

cluster wait time for IO = O (1 / bandwidth)
storage bandwidth cost = O (1/capacity cost)
cost variations: 10^2-10^3 x
perf variations: 10^2-10^3 x

many new models possible: tape - nvram



Large US Deployments

Large US deployments 2000 - 2023

Server Centric Storage

2000 – 2014

Racks with compute nodes
Disk enclosures
up to 1TB/sec, disk, 100x size of RAM
Lustre / GPFS

2014-2018

Racks with blades
Flash caches: 10x RAM, 5 TB/sec
Lustre / GPFS / DDN IME / Cray
Secondary disk FS: 100x RAM, 1TB/sec
Clients for many core Intel chips– not yet for GPUs

Client Centric Storage

2020 (Capability System at LANL)

Every compute node: Lustre ZFS flash server
100 TB/sec (10 GB/sec /node)
Secondary disk storage (100PB) – Campaign Storage

2023 (US Exascale)

1TB HBM / node -
Object store: 10,000 NVRAM server nodes
1PB / sec (100GB/sec / node)
secondary flash(?) storage(1EB)

Using IO in HPC

Cluster File System Performance Trouble

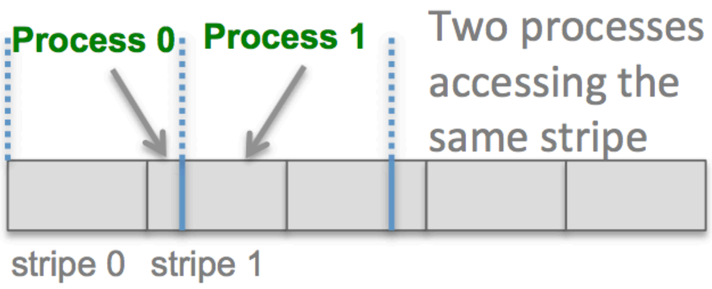
Massive exchange of small data

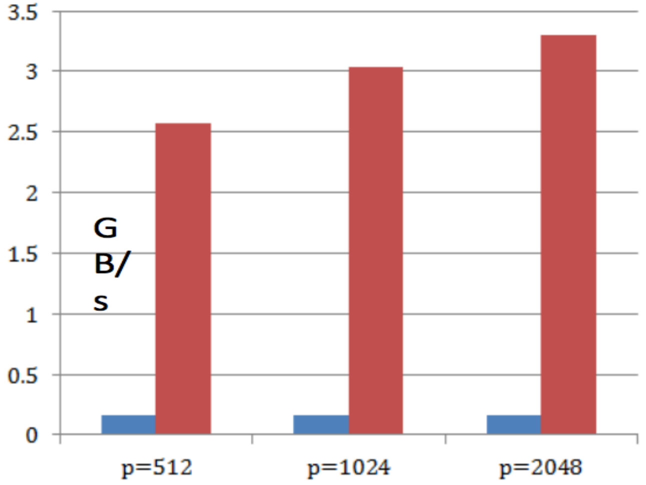
Not un-common

Root causes:

1. Concurrent resources required
2. Data layout must be carefully chosen
   - Ideally 1 process uses only 1 server
   - Reasonable stripe sizes
3. Complicated metadata data interactions

- 2007: ADIOS library addresses these issues





Tianhe-1A, ADIOS AMR .vs. Original IO

What does ADIOS really do?

What needs to be written

- New API – not POSIX, very simple
- Form group of processes
- Declare what items and how many need to be read / written
- Do IO asynchronously

How will it be written?

- External specification of file
- Use software plugin to drive the right storage infrastructure
- Describe the desired layout of data

Support for storing structured data

HDF5

- HPC standard for arrays, KV store and more
- Surprisingly small overlap with similar custom data layout for cloud

- Other formats (e.g. NetCDF) starting to leverage HDF5
- HDF5 beginning to use sophisticated lower layers (e.g. ADIOS)

Desired for Lustre: Very best HDF5 integration.

New IO Software

DAOS – distributed async object store

A USA DOE – Intel – HDF5 group collaboration

2012 – 2015: initial prototype based on Lustre / ZFS
2015 - : 2nd pre-production NVM implementation
Open Source

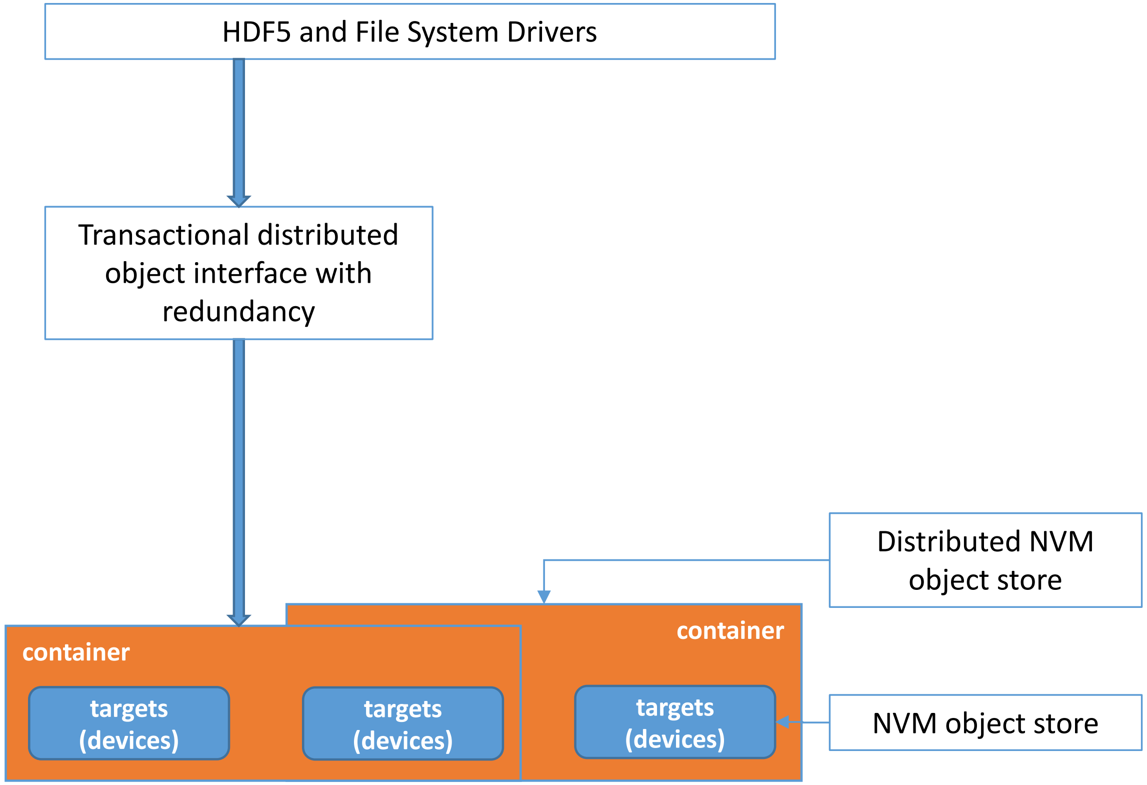
Key capabilities:

- Low level NVM transactional, versioned storage
- Distributed groups of processes collaborate on IO
- Scales to 100K's servers, 1B client processes
- Redundancy

Applications

- Underpinning for HDF5 and legacy file system

Probably not so easy to use directly


HDF5 and File System Drivers
Transactional distributed object interface with redundancy
Distributed NVM object store
container
container
targets (devices)
targets (devices)
targets (devices)
NVM object store

Role of containers

Fundamental issue: fast side vs slow side in hierarchy

Hence:

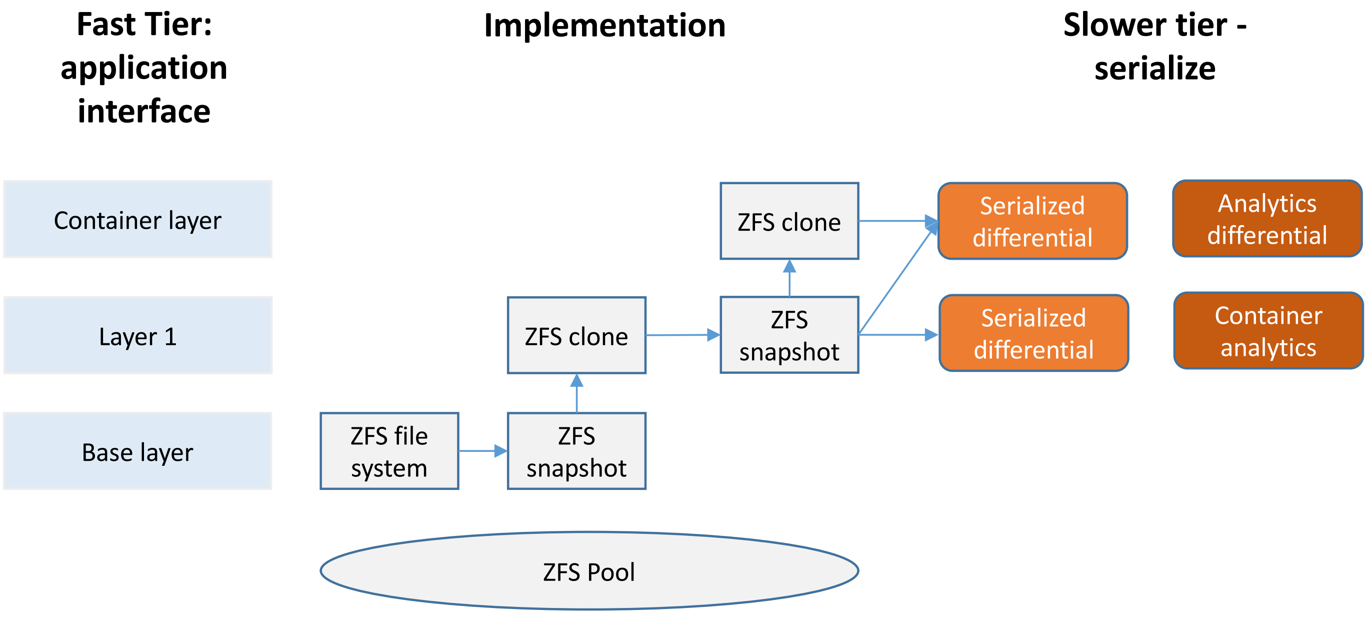
- Create fine grained data
- Move coarse grained data

Container implementations

- Can be based on ZFS
- Respect slower and faster interfaces

Other approaches:

- DDN IME
- Cray Data Warp



Challenges & Conclusions

Challenges

API introduction

- Cluster File Systems leveraged well established API: POSIX
- New systems must create and establish API. “All” applications must come along.

Deployment contrast

- HPC must become more cloud compatible
- Cloud Data Storage presently has fundamentally different qualities

Conclusions

Beauty and Simplicity

- Simple, convincing systems are emerging:
  - DAOS, ADIOS, Containers, HDF5, Campaign Storage
- Exciting challenges exist
- Hardware developments have been fantastic

Thank you