Campaign Storage

Peter Braam 2017-04

Co-founder & CEO Campaign Storage

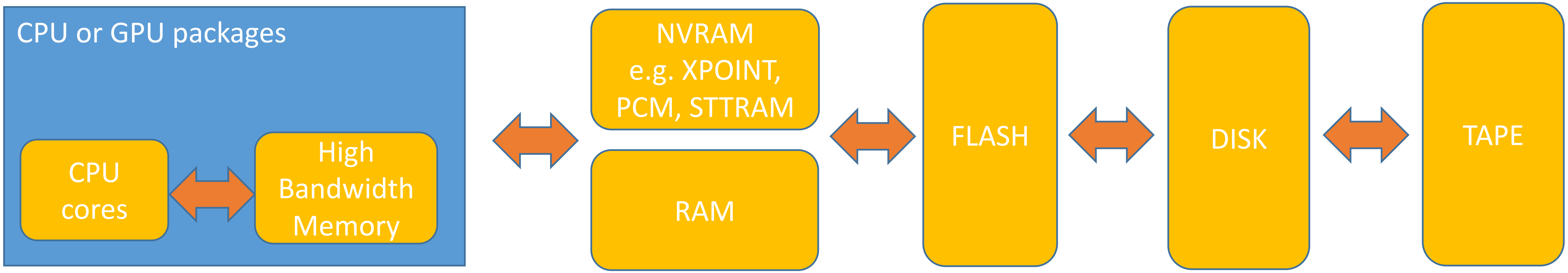
Contents

- Memory class storage & Campaign storage
- Object Storage
- Campaign Storage
- Search and Policy Management
- Data Movers & Servers
- Road Ahead

Campaign Storage

- Campaign Storage was invented at Los Alamos National Laboratory
  - 2014-
- Peter Braam & Nathan Thompson founded *Campaign Storage, LLC* in March 2016
  - deliver products in this space
  - Software Defined Storage – we will partner with integrators
- Other companies are addressing parts of Campaign Storage also

Storage Tiers & Campaign Storage



Burst Buffers –
DDN IME, Cray Data Warp

Node BW (GB/sec)	**1 TB/s**	**100 GB/s**	**20 GB/s**	**5 GB/s**	**350 MB/s**
Cluster BW (TB/sec)	**1 PB/s**	**100 TB/s**	**5 TB/s**	**100 GB/s**	**10's GB/s**
Software	**Language level**	**Language level, NVM libs** **HDF5 & DAOS**	**HDF5** **DAOS**	**Parallel FS &** **Campaign Storage**	**Archive &** **Campaign**
Key features	**transparent computation**	**transparent computation** **ultra-fast storage apps**	**name space** **scientific formats** **FS style container**	**bulk data movement** **- many files** **- subtrees of MD**	
BW Cost $/ (GB/s)	**$10 (CPU included!)**	**$10**	**$300**	**$2K**	**$30K**
Capacity Cost $/GB	**$**	**$8**	**$0.3**	**$0.05**	**$0.01**

Role of containers

Fundamentally unlikely:

different tiers perform data movement at similar granularity

Containers are a must-have

Tiers and NVRAM Considerations

Tiering

RAM tiers are for computation

➔ migrate **pointers, pages**

Flash storage is 5x faster with large IO

Disk similarly is very IO size sensitive:

➔Retrieve & store **containers** (distributed?)

➔Show internal structure on faster side

➔Stream and serialize data to slower side

Internal program data formats not re-usable

➔ computing format to **namespace**

Persistence

Distinguishing NVM feature is that data stays if power is off.

NVRAM will be the fastest storage device

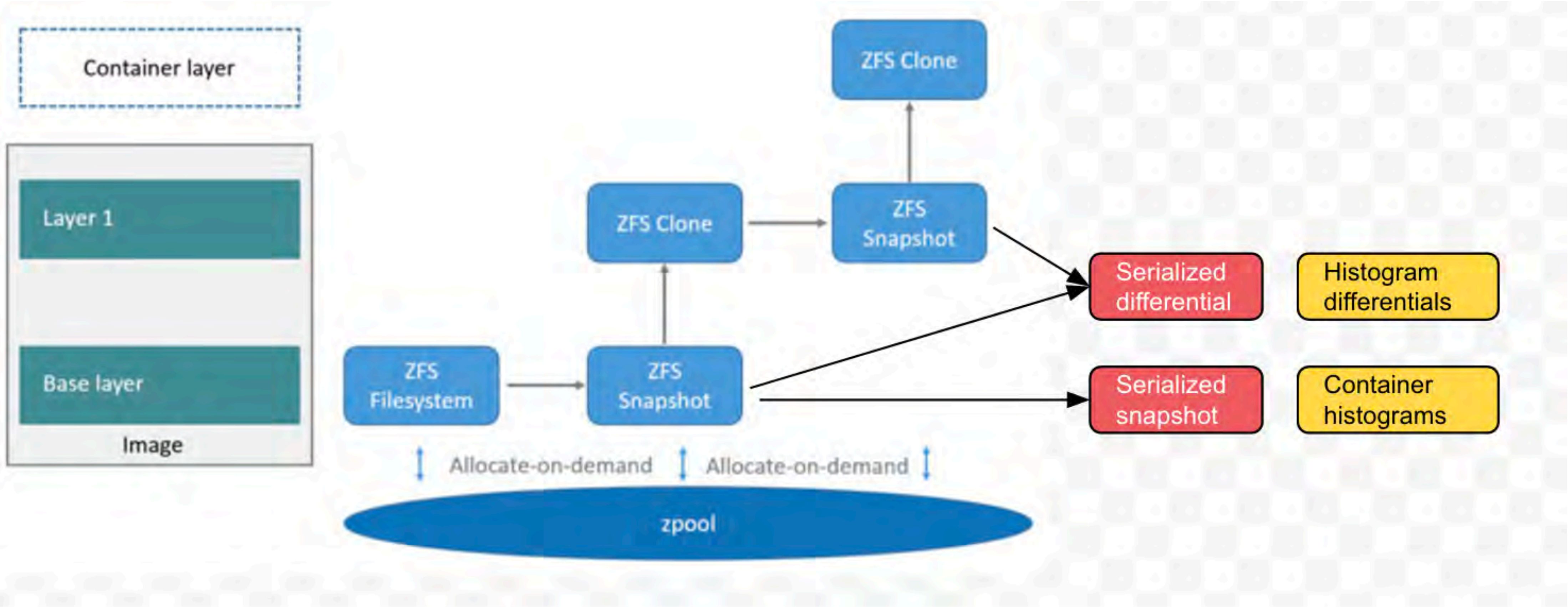
➔ for **most demanding storage applications**

NVRAM: what other benefits to computing?

Current libraries – transactions, persistent heaps

(not so novel – see Camelot & RVM from 1980's)

Example Container Functionality - lower tiers



Tiers & Transparency

RAM

- Demote infrequently used pointers
- Promote frequently used pointers

If pointers are not first class objects

- Promote upon access
- Demote finding less used ones

Low level languages – HW or OS support

Storage

Same principle – transparency requires accessing data through a **handle**

One handle system with location database allows other objects to move

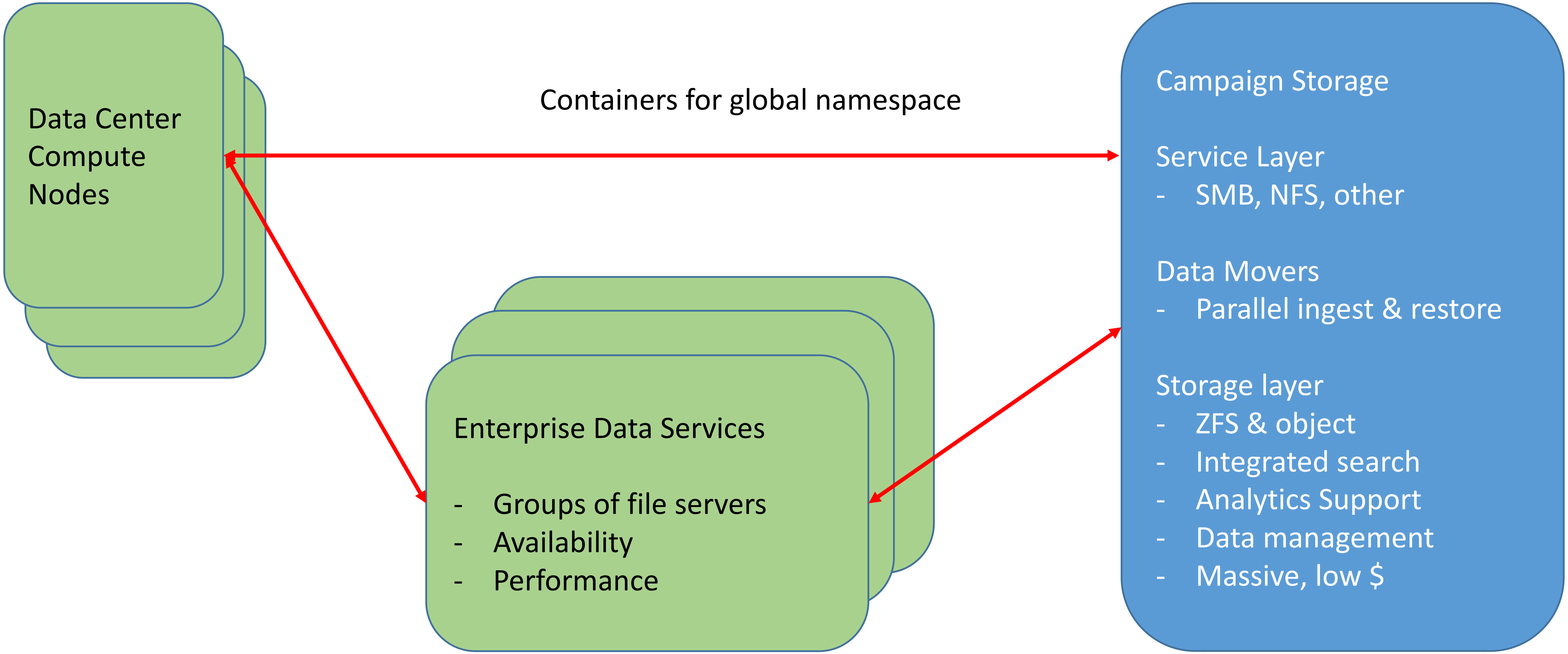
Expect distributed tiered KV store

- Key value lookup
- Callbacks for invalidation

Using Tiers

Data Center

Identity and namespace management with e.g. AD or LDAP



Containers for global namespace

Data Center Compute Nodes

Enterprise Data Services

- Groups of file servers
- Availability
- Performance

Campaign Storage

Service Layer

- SMB, NFS, other

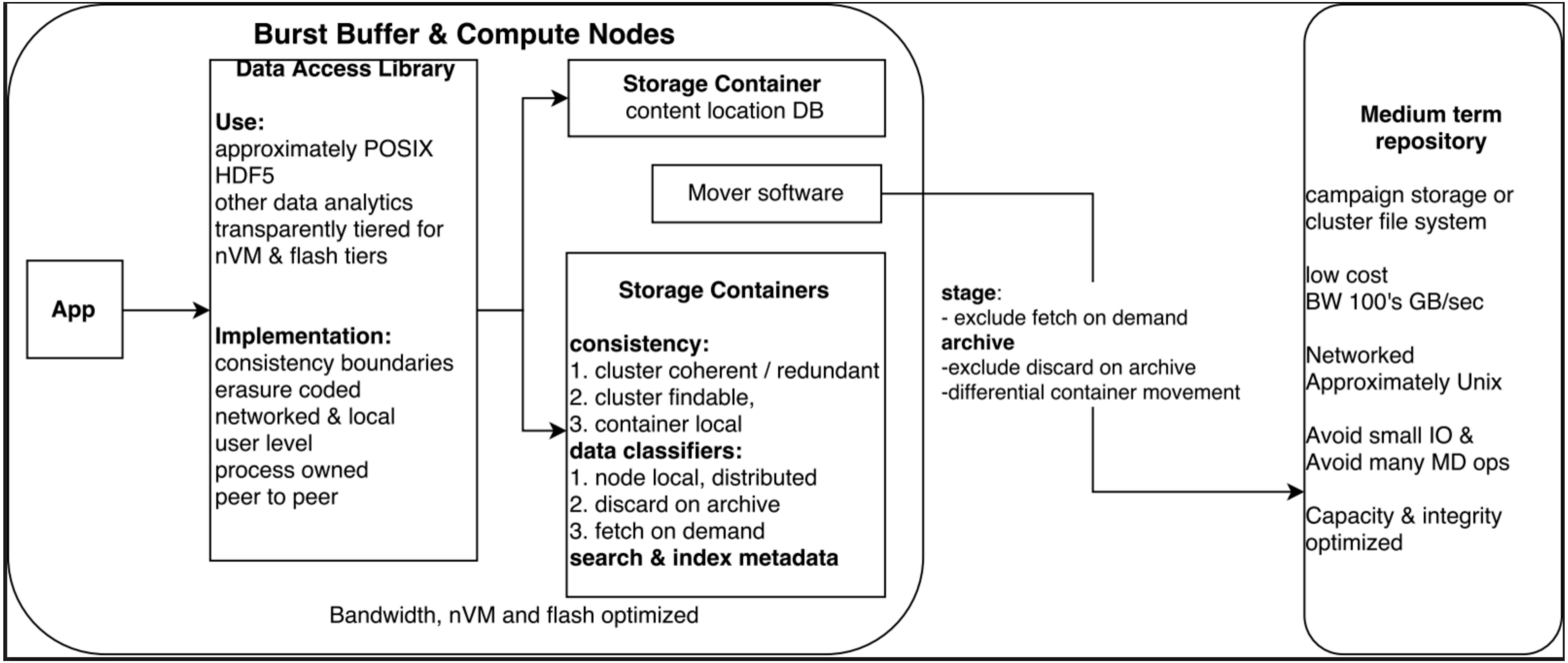
Data Movers

- Parallel ingest & restore

Storage layer

- ZFS & object
- Integrated search
- Analytics Support
- Data management
- Massive, low $

Future Exa-Scale Storage Architecture



Burst Buffer & Compute Nodes

App

Data Access Library

Use:
approximately POSIX
HDF5
other data analytics
transparently tiered for
nVM & flash tiers

Implementation:
consistency boundaries
erasure coded
networked & local
user level
process owned
peer to peer

Storage Container
content location DB

Mover software

Storage Containers

consistency:
1. cluster coherent / redundant
2. cluster findable,
3. container local

data classifiers:
1. node local, distributed
2. discard on archive
3. fetch on demand

search & index metadata

Bandwidth, nVM and flash optimized

stage:
- exclude fetch on demand

archive
-exclude discard on archive
-differential container movement

Medium term repository

campaign storage or
cluster file system

low cost
BW 100's GB/sec

Networked
Approximately Unix

Avoid small IO &
Avoid many MD ops

Capacity & integrity
optimized

Object Storage

Cloud object stores – pros & cons

pro

- massive scalability
- very good storage management
- widely agreed S3 REST API
- runs on cheapest hardware

con

- data lacks organization
- API’s don’t allow distributed concurrent access or random writes
- performance can be disappointing
- difficult to re-use as a component of other storage systems

Too much choice?

- Caringo Swarm (formerly CAStor)
- Cleversafe dsNet
- Cloudian
- Data Direct Networks Web Object Scaler (WOS)
- EMC Atmos
- EMC Centera
- EMC Elastic Cloud Storage (ECS)
- HP StoreAll
- HGST Himalaya
- HGST Active Archive
- Hitachi Data Systems HCP
- NetApp StorageGrid Webscale
- Quantum Lattus
- Scality Ring
- SwiftStack Swift

To mention a few ……. (others S3, CEPH, SNIA T10, Seagate A200, DDN WOZ ….)

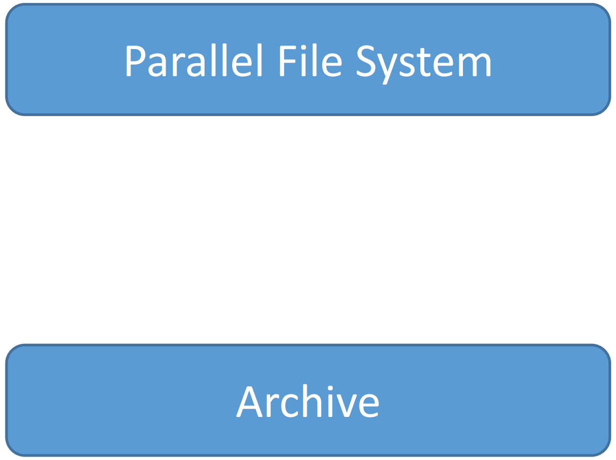
What is needed offers:

- Normal read/write IO per object
- Non overlapping IO from multiple clients
- 3 tier hierarchical redundancy (box, rack, data center)
- Transaction protocol to snapshot consistent state

Campaign Storage

Campaign Storage - a new tier

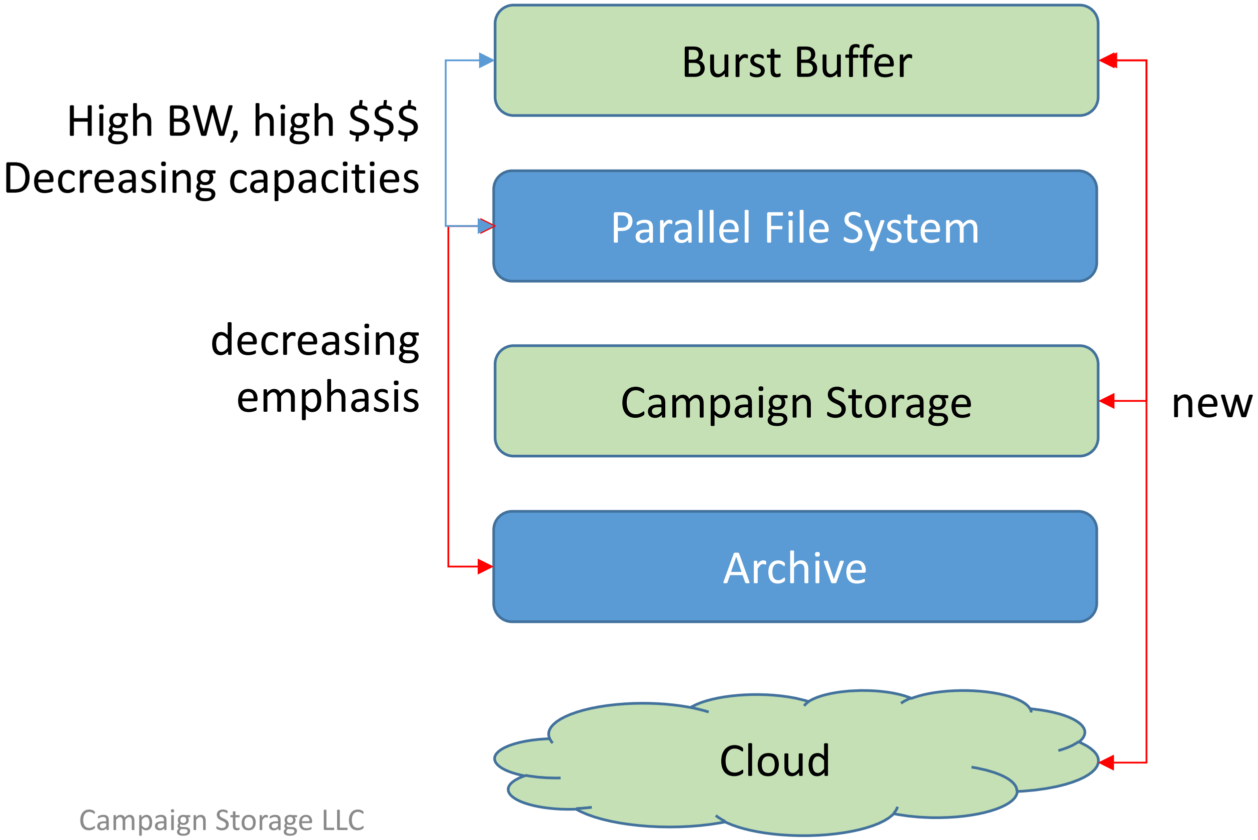
Old World



Parallel File System

Archive

New World



High BW, high $$$
Decreasing capacities

decreasing
emphasis

Burst Buffer

Parallel File System

Campaign Storage

Archive

Cloud

new

Campaign Storage

It is …

A file system

Focus: staging and archiving

Built from

- Industry standard object stores
- Existing metadata stores

Lowest cost HW

High capacity, ultra scalable

Not highest BW or lowest latency

- 10-100x higher than archives
- 10x lower than PFS

It is not …

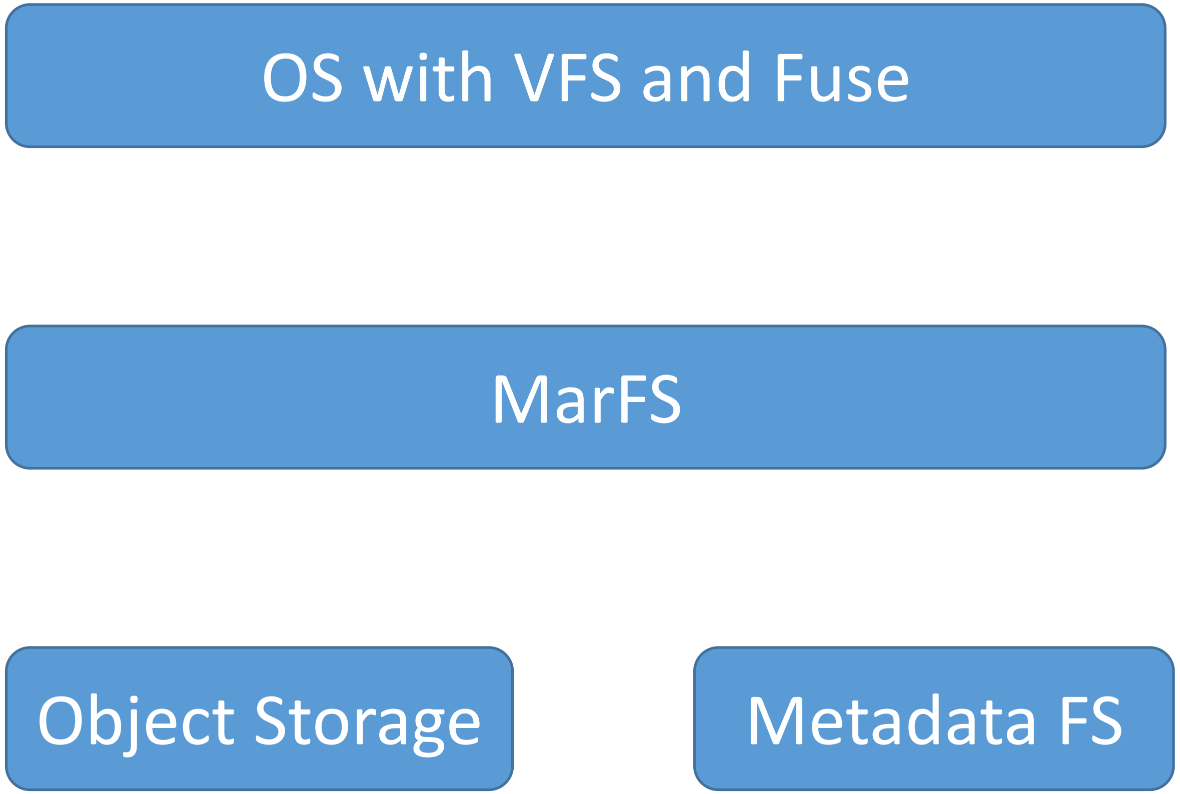
General purpose file system

- Wait … these don't exist actually

Using object stores has problems

- Limited set of data movers supported

Implementation



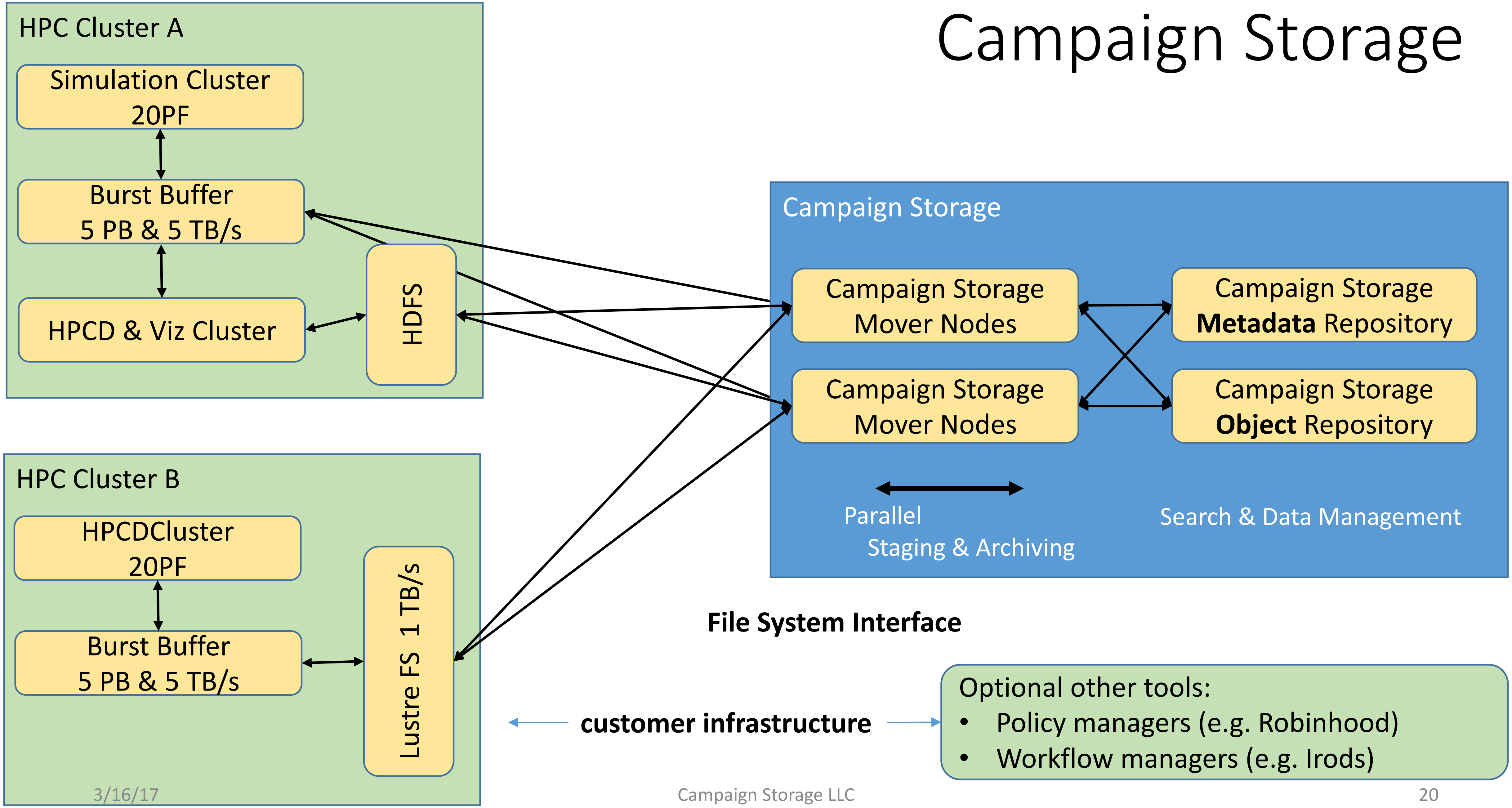
OS with VFS and Fuse

MarFS

Object Storage

Metadata FS

Campaign Storage



HPC Cluster A

Simulation Cluster 20PF

Burst Buffer 5 PB & 5 TB/s

HPCD & Viz Cluster

HDFS

HPC Cluster B

HPCDCluster 20PF

Burst Buffer 5 PB & 5 TB/s

Lustre FS 1 TB/s

Campaign Storage

Campaign Storage Mover Nodes

Campaign Storage **Metadata** Repository

Campaign Storage Mover Nodes

Campaign Storage **Object** Repository

Parallel Staging & Archiving

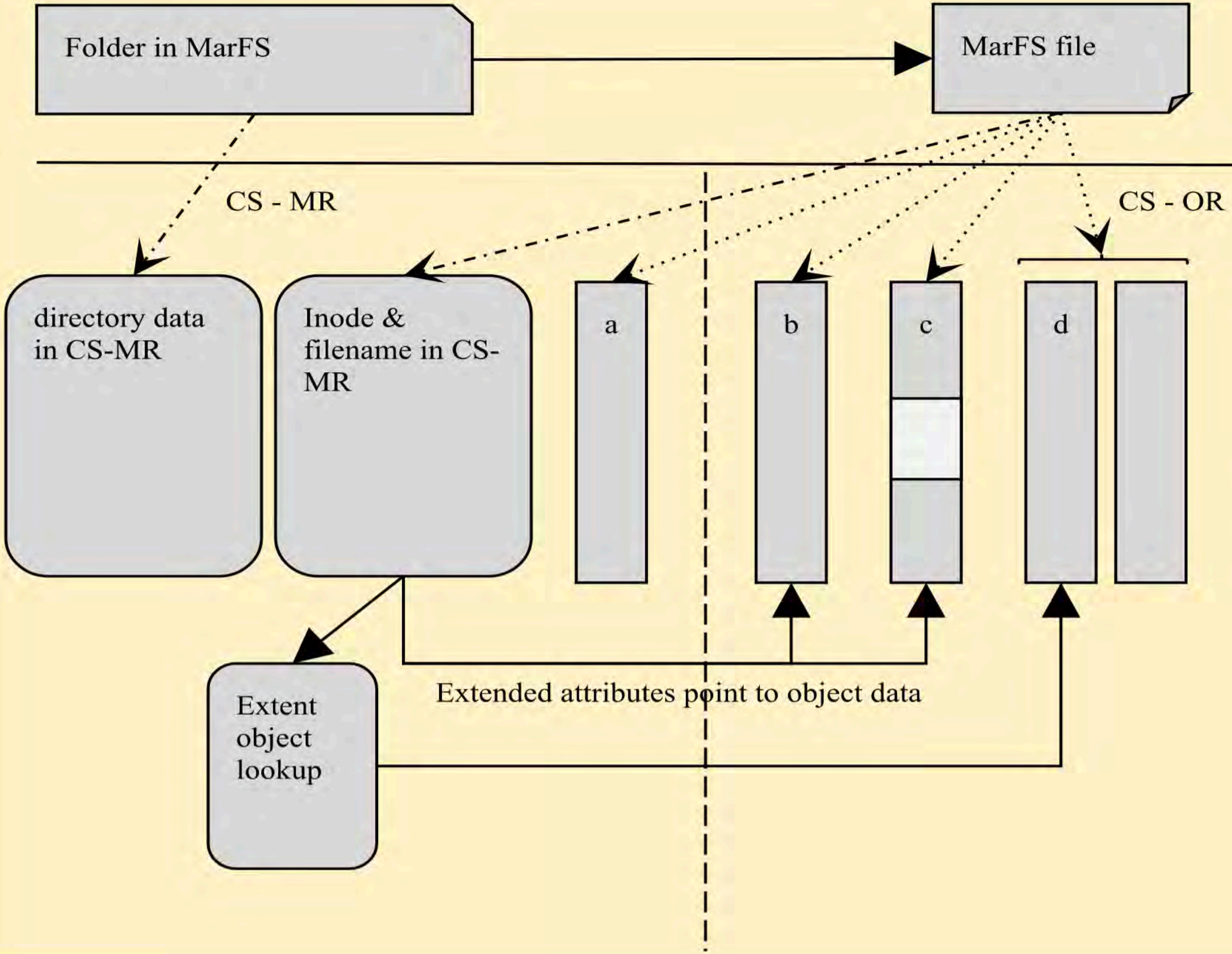
Search & Data Management

File System Interface

customer infrastructure

Optional other tools:

- Policy managers (e.g. Robinhood)
- Workflow managers (e.g. Irods)



Folder in MarFS

MarFS file

CS - MR

CS - OR

directory data in CS-MR

Inode & filename in CS-MR

a

b

c

d

Extent object lookup

Extended attributes point to object data

User visible data

Implementation of principal file system data

Store file data in:
a. CS-MR file system
b. In a single object
c. Packed data in a single object
d. Striped or chunked across multiple objects

Search & Policy Management

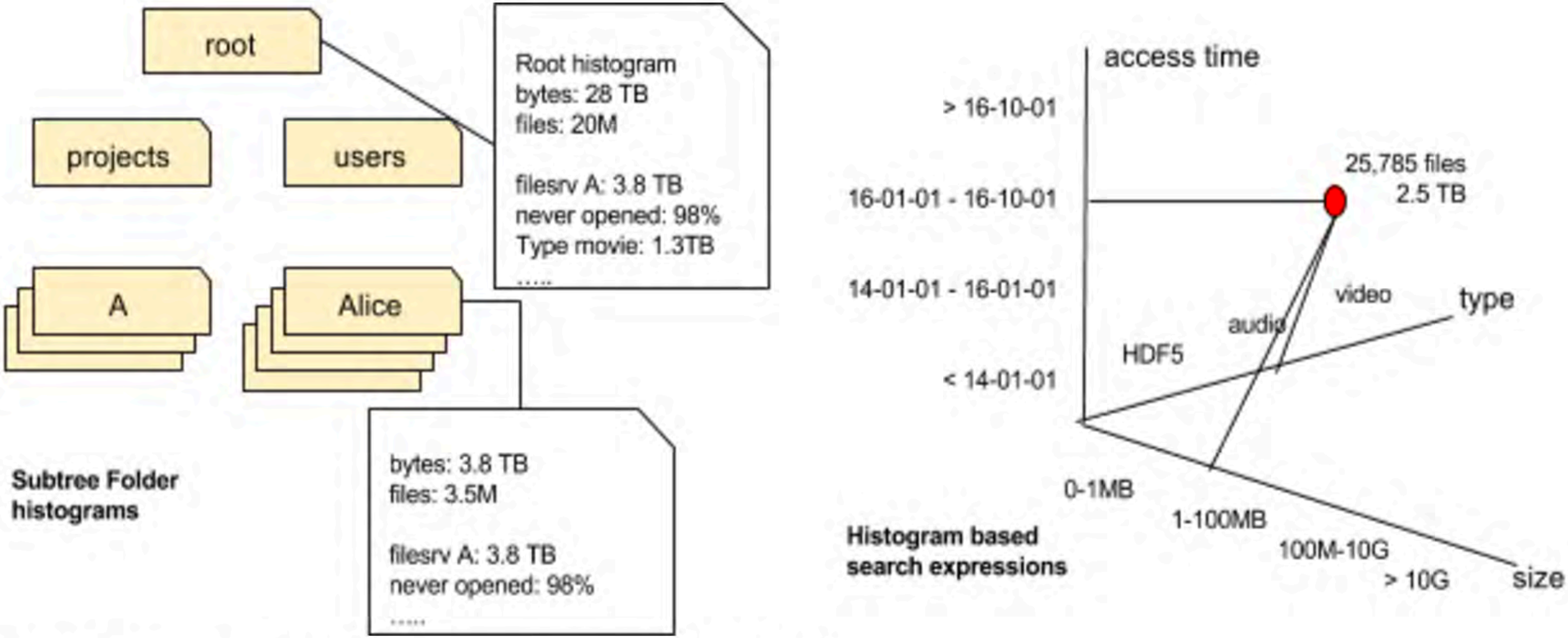


Figure 3: Subtree histogram indexes and search algebra

Histograms for subtree search

Every directory has histogram DB recording properties of its ***subtree***:

- i.e. #files, #bytes in the subtree have a property?
- Limited granularity, limited relational algebra
- Store perhaps ~100,000 properties in multiple histograms

Examples:

- Quota in subtree?
- What fileservers contain files?
- Geospatial information in file?
- (file type, size, access time) tuples
  - Allows limited relational algebra
- User database for subtree – eliminates reliance on external identity management

Not a new idea. Can be added to ZFS & Lustre

Data Movers & Services

Data Movers

Data Movement

Today

- LANL “parallel rsync” – pftool
- Lustre HSM mover
- Packing small files & striping big files

Candidates

- DMAPI HSM mover
- Gridftp
- Full POSIX interface

Metadata Movement

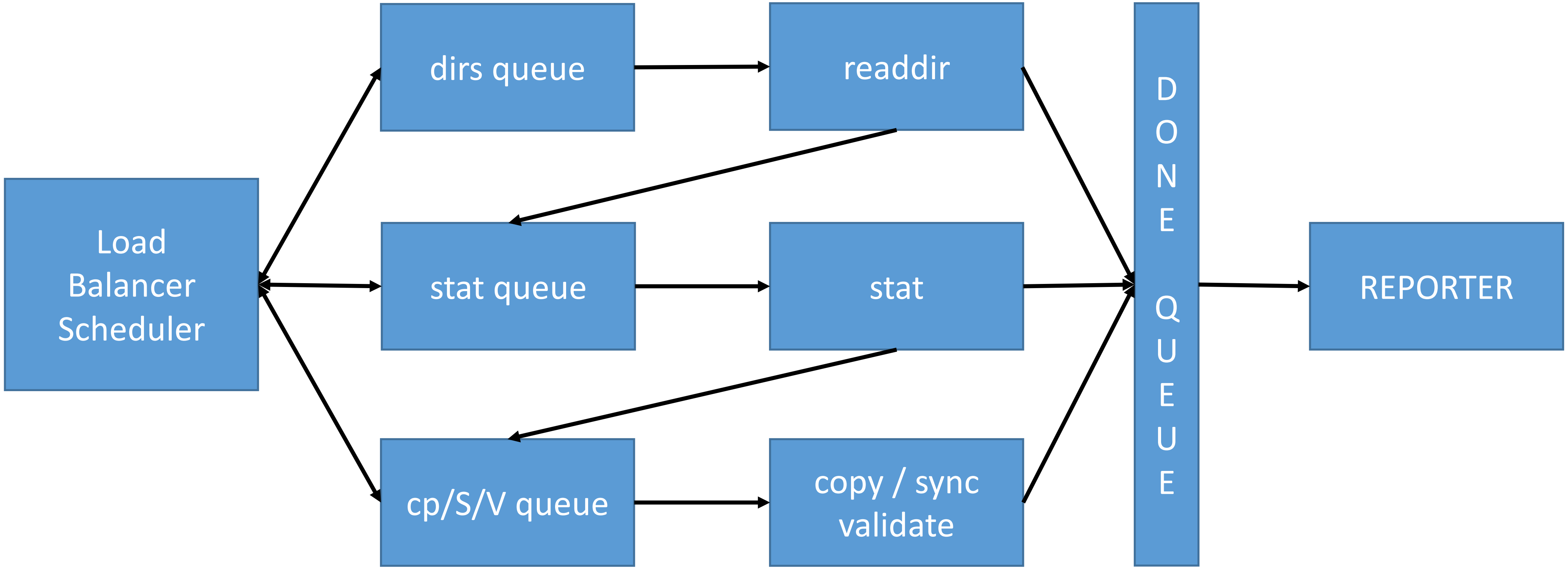
Today

- Traditional metadata API
- Multiple namespaces

Coming

- Bulk integration of containers
- Accompanying metadata

pftool internals



Features of DS3 archival data mover

- Object store moves batches of files
- New concept: file level I/O vectorization
  - Includes server driven ordering
  - Packing small files into one object

```
int copy_file_range_fv(copy_range *r, uint count, int flags)

struct copy_range {
        int source_fd;
        int dest_fd;
        off_t source_offset;
        off_t dest_offset;
        size_t length;
}
```

Services

Campaign Storage always exports the MarFS file system

Enterprise services as further exported protocols:

- SMB, NFS, HTTP
- Data movement can be out of band

Integration of namespaces, user databases, other plugins

Campaign Storage Use Cases

Workflows - HPC

Staging & De-staging

- Schedule migration with pftool

HSM

- Copy metadata first
- Use subtree search index
- Execute policies
- Specialized data movers
  - For transparent retrieval & attributes

Single project extraction

- Use ZFS namespace and object bucket per project

Hot vs cold Campaign Locations

- Select destination object stores
- Migration on campaign storage

Multi site

- Leverage object bucket replication
- Leverage ZFS pool replication

Cloud

- Migrate pool and buckets to S3
  - Use Snowball?

Workflows – Data Center

Staging & archive

- Schedule migration with pftool

Service offload to Campaign

- Data available without staging

Single project extraction

- Use ZFS namespace and object bucket per project

Hot vs cold locations

- Select destination object stores
- Migration within campaign storage
- Automatic movement when services need the data

Multi site

- Leverage object bucket replication
- Leverage ZFS pool replication

Cloud

- Migrate pool and buckets to S3
  - Use Snowball?

Road Forward

Unique opportunity to innovate data management

LANL and Campaign Storage created an “Industry Steering Group”

Seek agreement on

- Data layout handling
- Attributes used in connection with long term storage
- Interfaces for workflows

Conclusions

Conclusions

Hardware diversification ➔ Software Specialization

Expect a rich high speed exa-scale I/O platform to use containers

Similar containers will organize enterprise tiers of storage

Campaign Storage: bulk data store, archive & data movement

Thank you