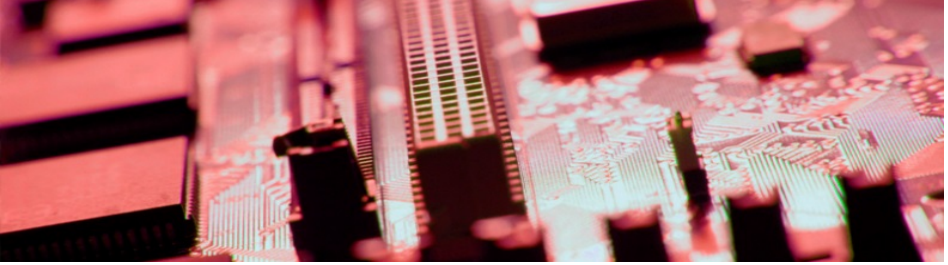

LCI workshop
on high performance clustered computing

Linux Clusters Institute: Survey of File Systems

Georgia Tech, August 15th – 18th 2017

J.D. Maloney | Storage Engineer
National Center for Supercomputing Applications (NCSA)
malone12@illinois.edu

NCSA

Baseline

- Look over slides from Pam Hill's talks at the beginner workshop in May 2017
  - http://www.linuxclustersinstitute.org/workshops/may2017/program.html
  - Segments of some following slides were adapted from her work
- Have an grasp on:
  - Definition of Parallel file system
  - Types of file systems (home, scratch, projects, archive)

Understanding Your Workloads

Finding Features You Need

- What features/traits do you need your file system to have?
- Some Examples:
  - Encryption at Rest
  - Tiered Storage/HSM
  - Simultaneous Multi-Fabric Support
  - High sequential throughput
  - High metadata performance
  - Open Source (aka no licenses)
  - Object Storage
  - Less FTE intensive to operate
  - Local node cache support
  - Very high client count

- Rank the order of importance, nothing has all the upsides with no downsides

Look at Current Metrics

- Similar to when exploring hardware, look at your current file system usage
  - Do you see a need for high metadata performance, sequential I/O performance?
  - Is there a need for growth to support a high number of clients (many 10,000s) over the lifetime of the system?
  - Does your data center require the file system to interface with multiple network fabrics and/or clusters?
- Look at your growth model to see how you're going to need to add capacity or performance later on

Talk to the Stakeholders

- New storage systems are a great opportunity to make changes and incorporate things users need/want
- Are there features upcoming projects will require that aren't already able to be serviced
- Do stakeholders envision new requirements on the horizon
- Storage Admins are stakeholders too,

  what do you need/want out of the file

  system to make your life easier

Popular HPC File Systems

Spectrum Scale (GPFS) Overview

- Product of IBM, gone through many name changes
- Licensed file system, based on server socket count and client count
- One of the two “prominent” file systems in used today by the world’s largest supercomputers
- Generally considered easier to administer due to product maturity and Enterprise level features
- For most part expects to be run on top of reliable disks presented through redundant RAID controllers

IBM
Spectrum
Scale

Spectrum Scale (GPFS) Architecture

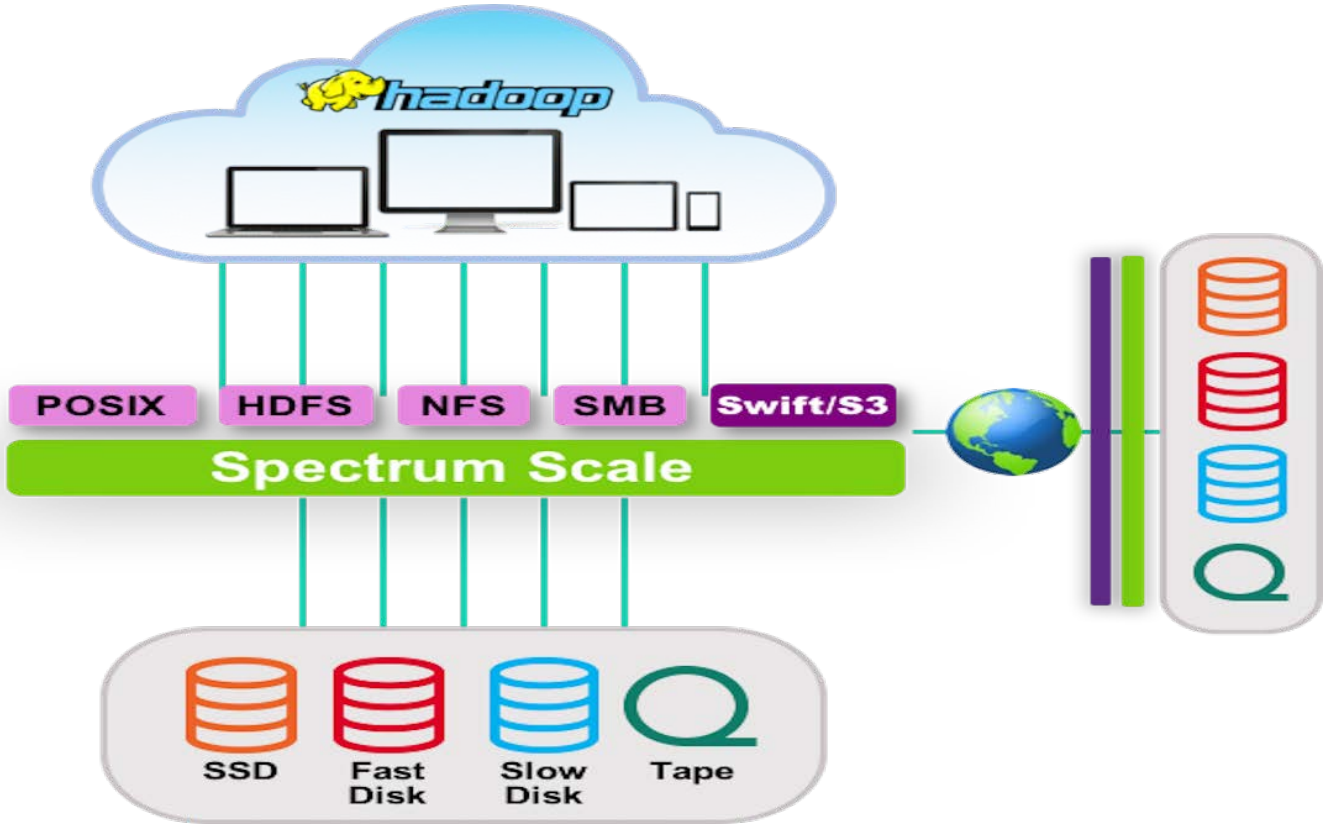

Image Credit: ibm.com

Spectrum Scale (GPFS) Architecture Notes

- Types of servers:
  - NSD Servers- Connected to disks, serve I/O to other nodes in the cluster
  - Manager Nodes- Cluster Manager, FS manager, Quorum Nodes
  - Clients- Mount file system for access, run GPFS daemon also
- Supports multiple storage pools and has HSM functionality
- Supports encryption at rest (Advanced License)
- Features Like:
  - AFM (Active File Management)
  - Built in Policy Engine (very powerful)
  - Native Cluster Export Services NFS, Samba/CIFS
  - Support for Object Storage via Cluster Export Services
  - Remote Cluster Mounts

Lustre Overview

- Open Source file system supported, developed by many companies and large institutions (Intel, Seagate, DDN, CEA, DOE)
- One of the two "prominent" file systems in used today by the world's largest supercomputers
- Known for its ability to scale sequential I/O performance as the storage system grows
- More complicated to administer, stricter operating environment (OFED stack, kernel, etc.)
- Can grow to greater numbers of clients

Lustre Architecture

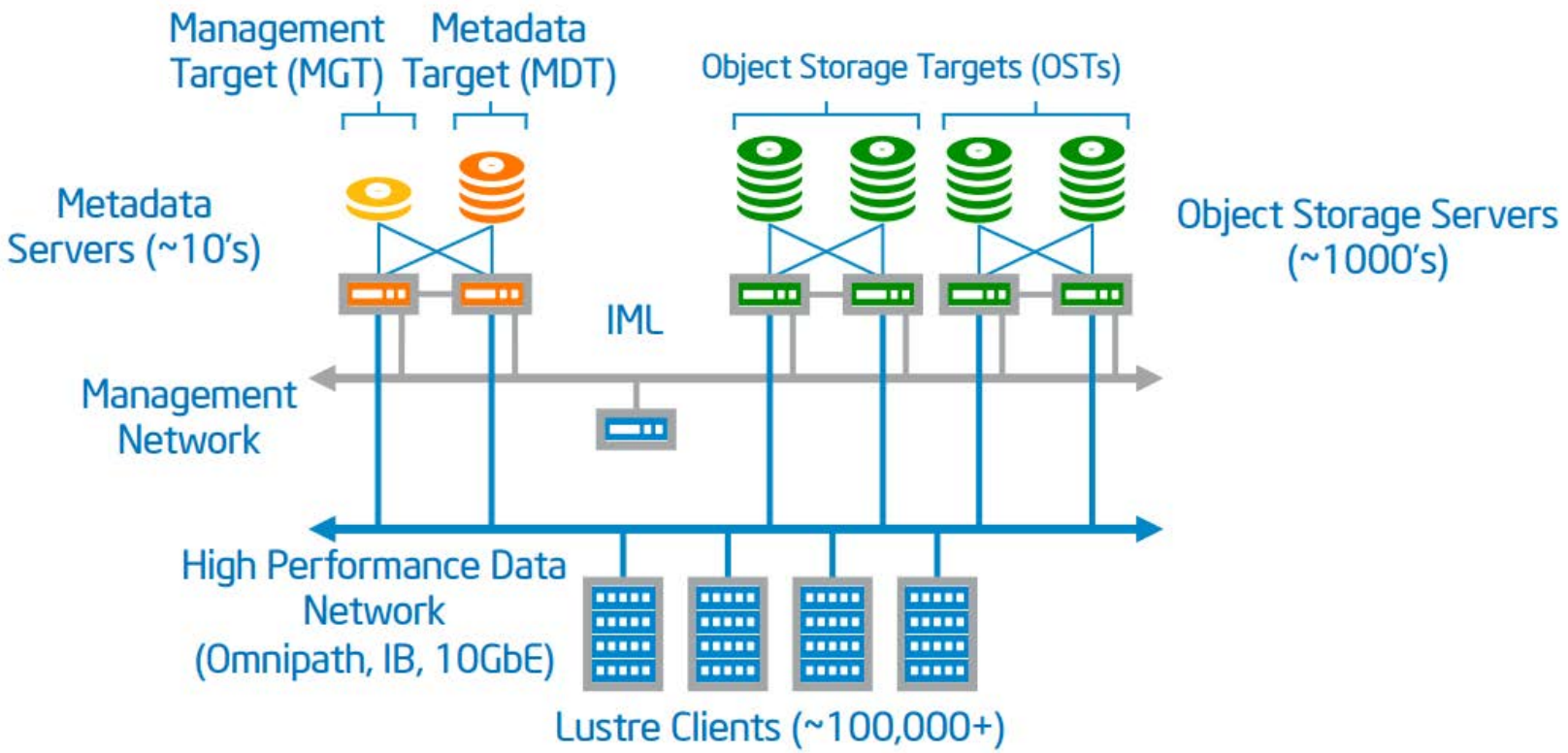

Image Credit: nextplatform.com

Lustre Architecture Notes

- Very saleable historically for OSS/OSTs (data storage/performance), but now also for MDS/MDTs (metadata storage/performance)
- Built in quotas, but no built in Policy Engine (external via Robinhood)
- Supports mixed fabrics within the same environment via LNET routing
- HSM support is built in, can be manipulated with external policy engine to migrate files between tiers
- Can run more ”commodity” storage, but expects reliable disk, HA maintained by failover

BeeGFS Overview

- File System formerly known as FgHS (Fronhoeffer File System)
- Maintained by the Fronhoeffer …..
- Most features and abilities are open source, support and some advance features can be purchased through Fronhoeffer
- Scaling of both data and metadata performance and capacity as hardware used grows
- More limited features set than other big file systems:
  - No encryption support
  - No HSM supported (yet)
  - No Policy Engine
- Gaining in popularity recently

BeeGFS®
developed by Fraunhofer

BeeGFS Architcture

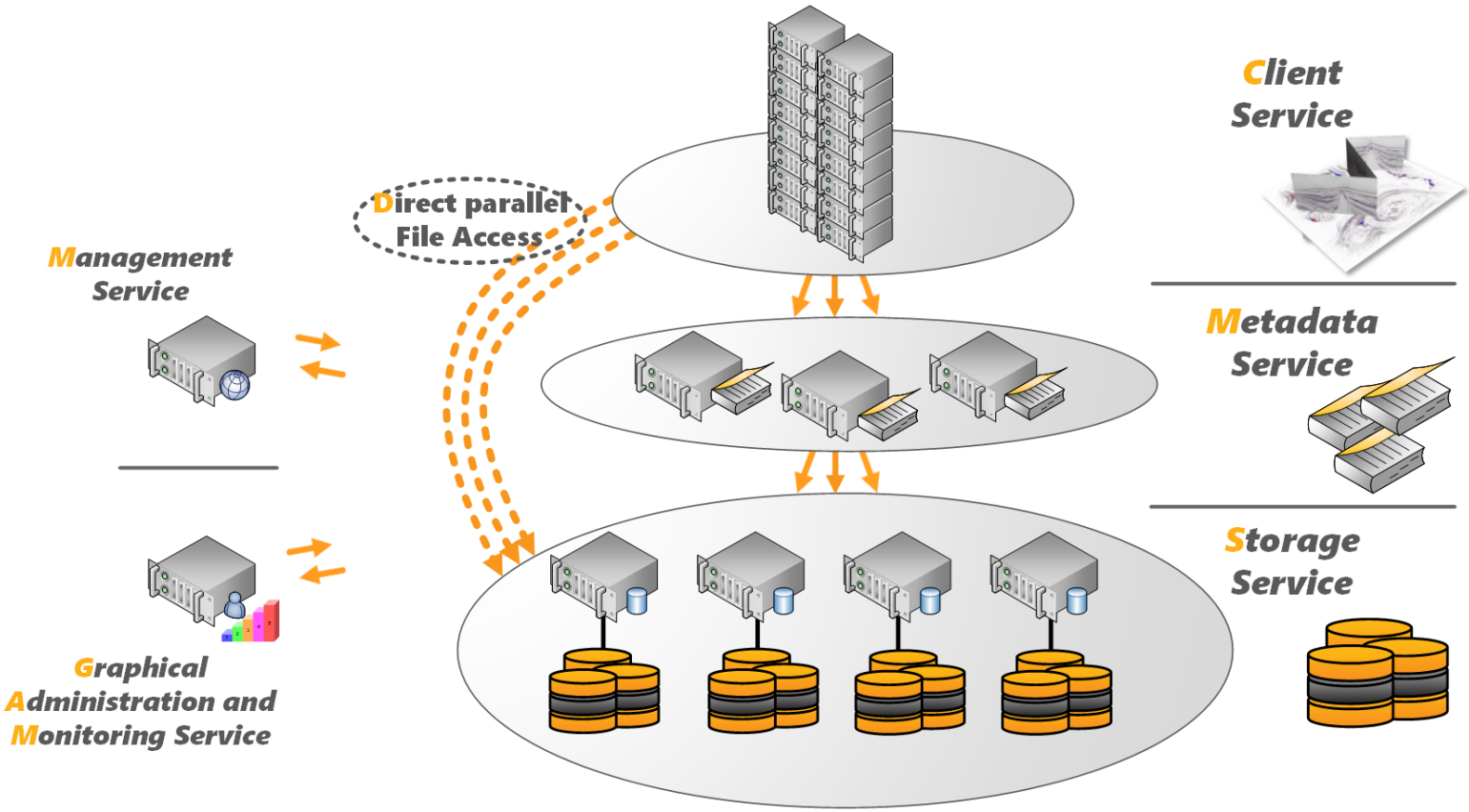

Image Credit: beegfs.io

BeeGFS Architecture Notes

- Scale metadata and data services independently
- Multiple services can be run from same node
- Host failure resistance by using ”Buddy Groups” to replicated data on a per directory granularity for high uptime
- Runs on “commodity” hardware
- Supportive of multiple fabric types (Ethernet, IB, OPA)
- Storage pools coming soon to allow grouping of LUNs by different types (SSD, HDD, etc.)

Panasas Overview

- Proprietary file system that is sold as an appliance
- Locked into Panasas hardware
- Supports storage tiering through different levels of hardware
- File system is called PanFS
- Somewhat of a cross between NAS like storage and HPC storage
- Scales well with hardware growth

Panasas Archticture

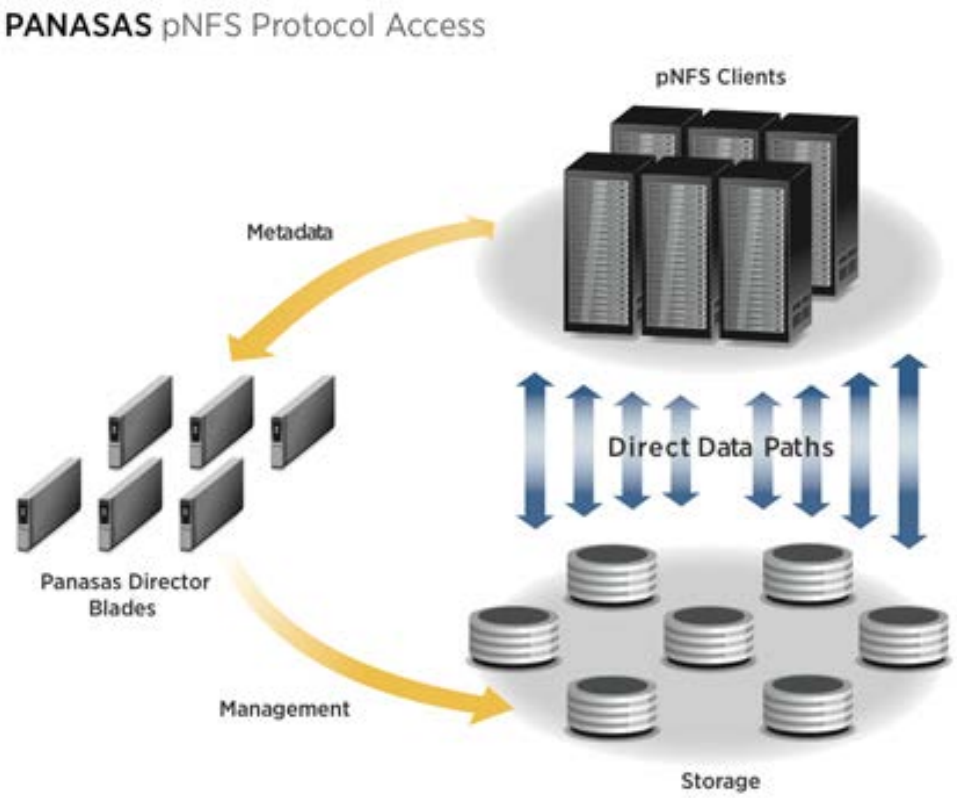

Image Credit: panasas.com

Panasas Architecture Notes

- Director blades handle file system metadata
- Data is stored in a object fashion on the capacity disks in the storage blades
- Object-based storage on the RAID allows for efficient rebuild of only lost blocks
- Automatic Tiering through Memory, Flash, and HDD levels of the file system
- Native Mount and NFS/CIFS support built into the file system
- Greater client support for Windows/Mac based clients

Popular “Cloud” File Systems

GlusterFS

- Distributed file system developed by Red Hat
- Popular file system to back cloud/virtualization resources
  - Integrated into Hypervisors such as Proxmox
  - Can be used to back Open Stack
- Open Source and free to use, support available via Red Hat
- Runs on ”commodity” hardware
- Includes features such as:
  - Erasure Coding
  - Geo-replication
  - Encryption at rest
  - Some HSM-like tiering options

GlusterFS Architecture

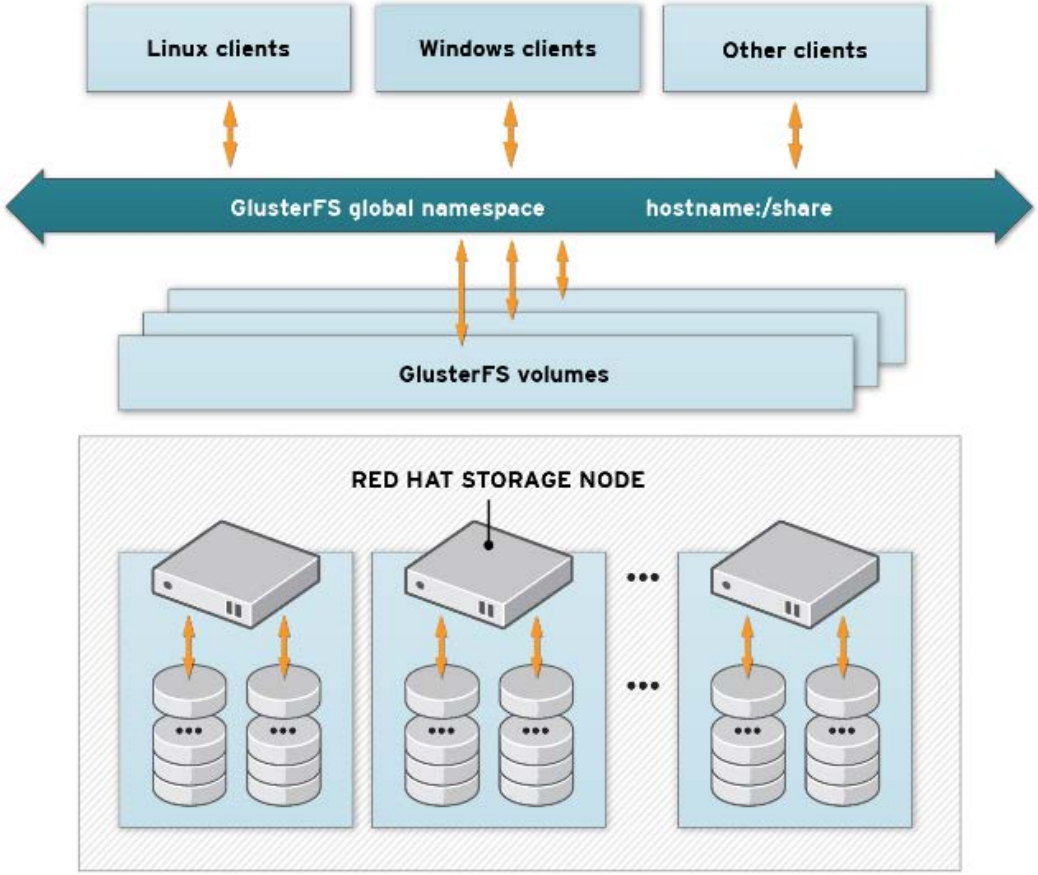

Image Credit: hmarcy.com

GlusterFS Architecture Notes

- Based on the “brick” principle, GlusterFS expects disks local to each machine to be grouped (via hardware or software RAID)
  - Disks don’t have to been by multiple hosts (unlike GPFS/Lustre)
  - Software RAID is usually used as it saves cost
  - Will work with individual drives, will just not like things once you scale up to hundreds or thousands of them
- Data and Metadata are stored together, no separate disk pools
- Can run on Ethernet or Infiniband
- Has native client as well as NFS/CIFS support for mounting across many different OS types
- Historic weakness is metadata operations (ls, rm, find) though performance is improving, set to be overhauled soon

Ceph Overview

- Object/Posix file system that is developed by Red Hat
- It along side GlusterFS are in Red Hat Storage Server product
- Open Source, free to use, support available through Red Hat
- Provides Object, Block, and Posix storage
- Very popular in the cloud/virtualization space
  - Can underpin things like Open Stack and Proxmox
- Runs on “commodity” hardware
- Can be deployed over Ethernet or IB
  - Ethernet is very popular with this FS

Ceph Architecture

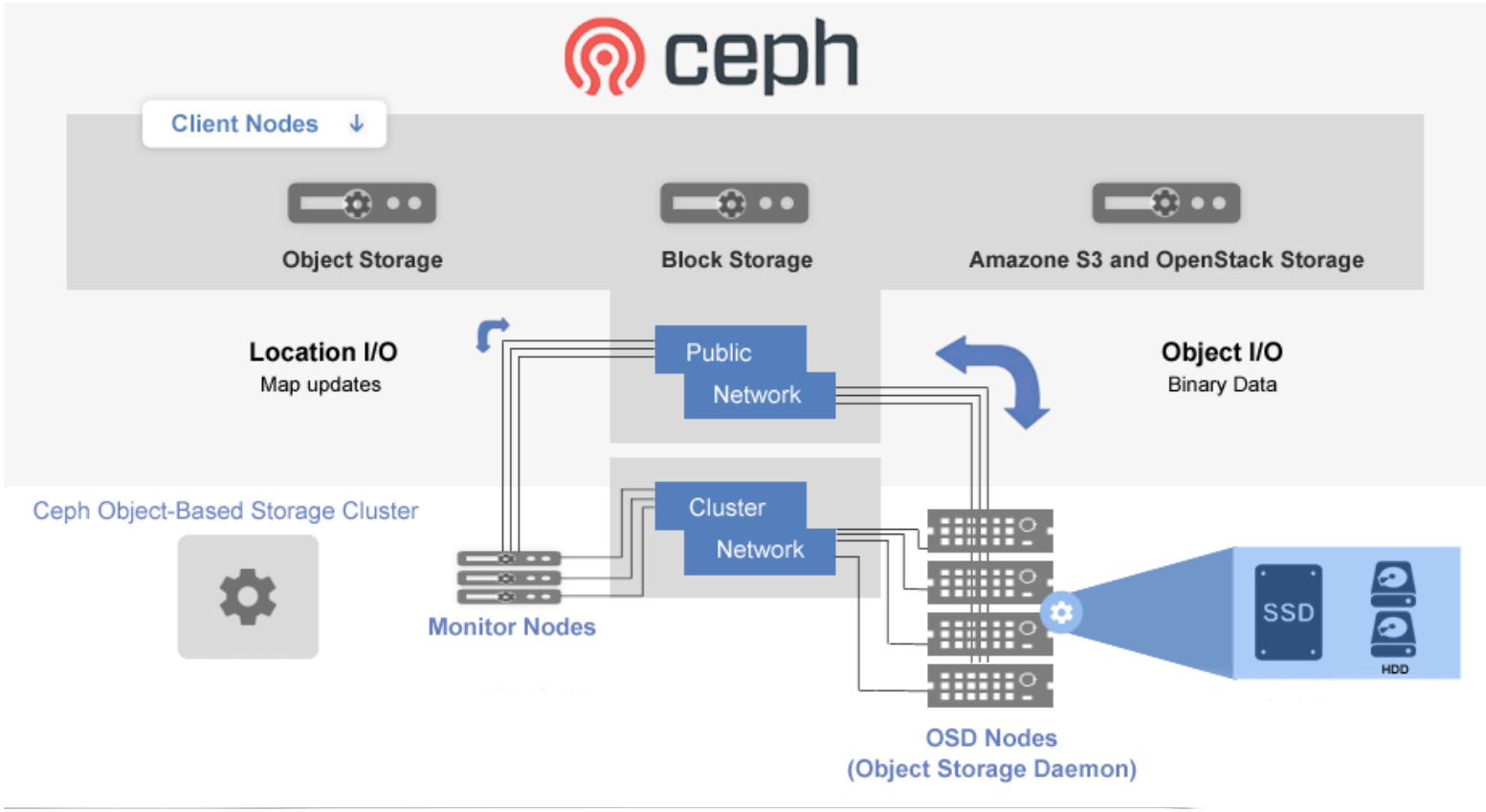

Image Credit: innotta.com.au

Ceph Architecture Notes

- Uses the CRUSH algorithm to handle file storage and retrieval across the cluster
- Disks are handled individually, each disk has a ”journal” device that allows for quick writes for small files
  - Multiple disks can share the same journal device, tuning the ratio and drive types allows one to increase or decrease performance
- Supports both replication or erasure coding for redundancy
  - Does NOT expect any devices to be RAID’d together before being presented
- Current implementation uses disks that are formatted XFS, but upcoming release will use the new “Bluestore” format on the drives to increase performance
- Multi metadata server support (for posix component) coming soon as well

Others

- There are many other file systems out there to choose from:
  - Swift
  - XtreemFS
  - LizardFS
  - OrangeFS
  - HDFS
  - List Goes On
- Always good to keep an eye on these as everything has to start somewhere, also helps you understand which way things are headed, where development is

Wrap Up

- Many of file systems mentioned in this presentation are open source and available to play with for free
  - Set some of the appealing ones to you up and test them out in VMs or on older hardware
  - Keep up on new releases as features get added or weak points get addressed
  - Tons of options out there to choose from
- No one right solution, many factors go into making the decision
  - Balance the trade-offs, your environment will have different constraints than someone else's
- Reach out to others in the community, attend User Group meetings

Acknowledgements

- Thanks to Pam Hill from NCAR for content assistance and laying the groundwork for this workshop
- Members of the SET group at NCSA for slide review
- Members of the steering committee for slide review

Questions