SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

MarFS Multi-Component:

Scalable Long-Term Storage on Commodity Disk

Garrett Ransom

HPC-SYS, Infrastructure Team

LANL/US DOE

LA-UR-17-26375

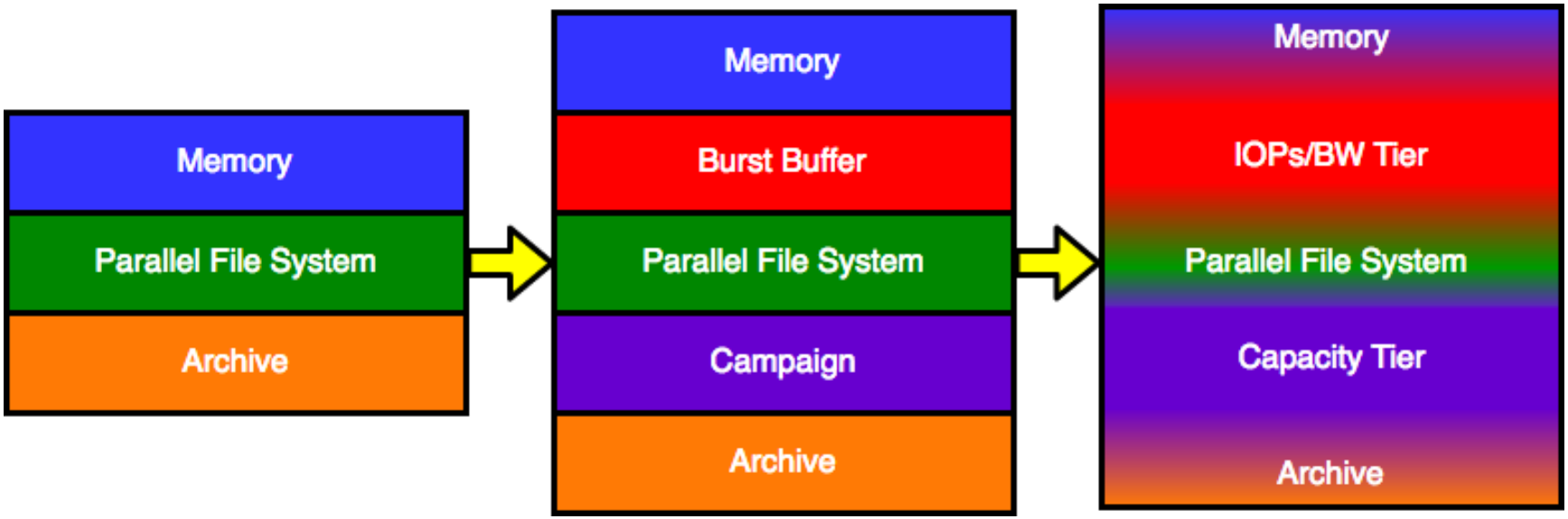
The Need

- Data sizes growing faster than long-term storage
  - Trinity: 2PB of memory / 4PB of flash
  - Crossroads: (maybe) 4PB mem / 10-100PB flash
  - HPSS Archive ~60PB Total, near capacity
- Bandwidth of archive is a limiting factor
  - HPSS ingests data much faster than it returns

What Can Do the Job?

- Parallel File Systems (i.e. Lustre)
  - Designed for high-bandwidth, short residency
  - Providing greater resiliency is expensive
- Tape Archive
  - Very reliable, very slow
  - Buying tape for bandwidth is expensive
- Both can suffer metadata scalability problems

Solution – More Tiers?



Implementing a Capacity Tier

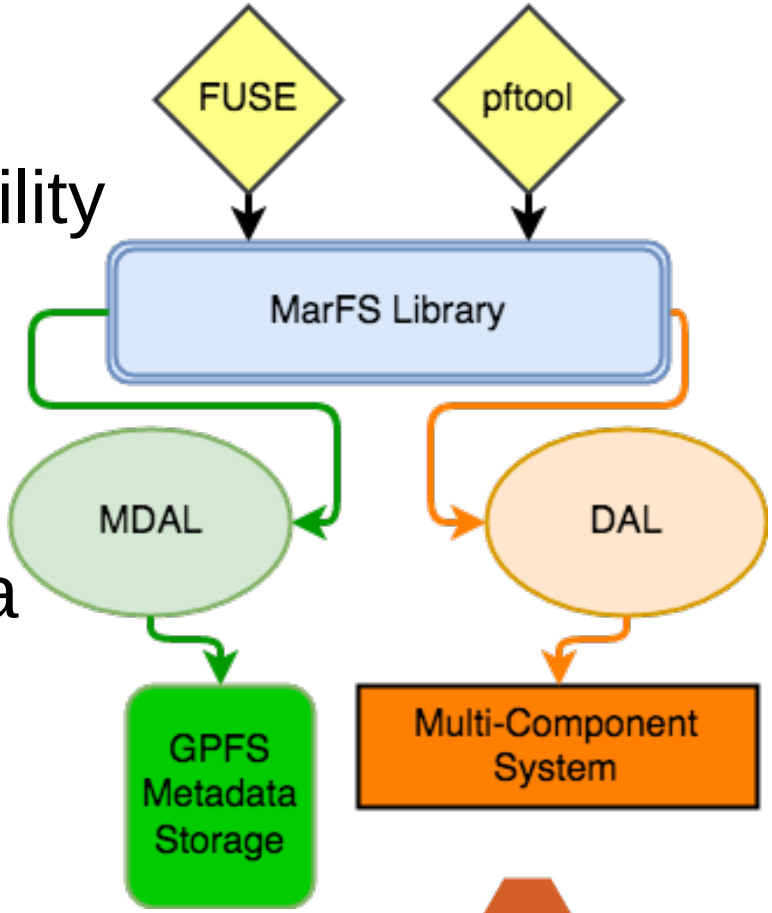
- Object Storage seems promising
  - Flat namespace allows easy scalability
  - Erasure coding allows for cheaper disk media
- Object Storage has limitations
  - Machines love object-IDs, people don’t
  - Potentially billions $ in applications expecting ‘POSIX-like’ file trees

MarFS

- A near-POSIX interface layered over distinct metadata and data implementations
  - Data stored as erasure coded objects
  - Metadata mirrored within a parallel file system (GPFS)
- Familiar semantics, fast metadata, stable objects

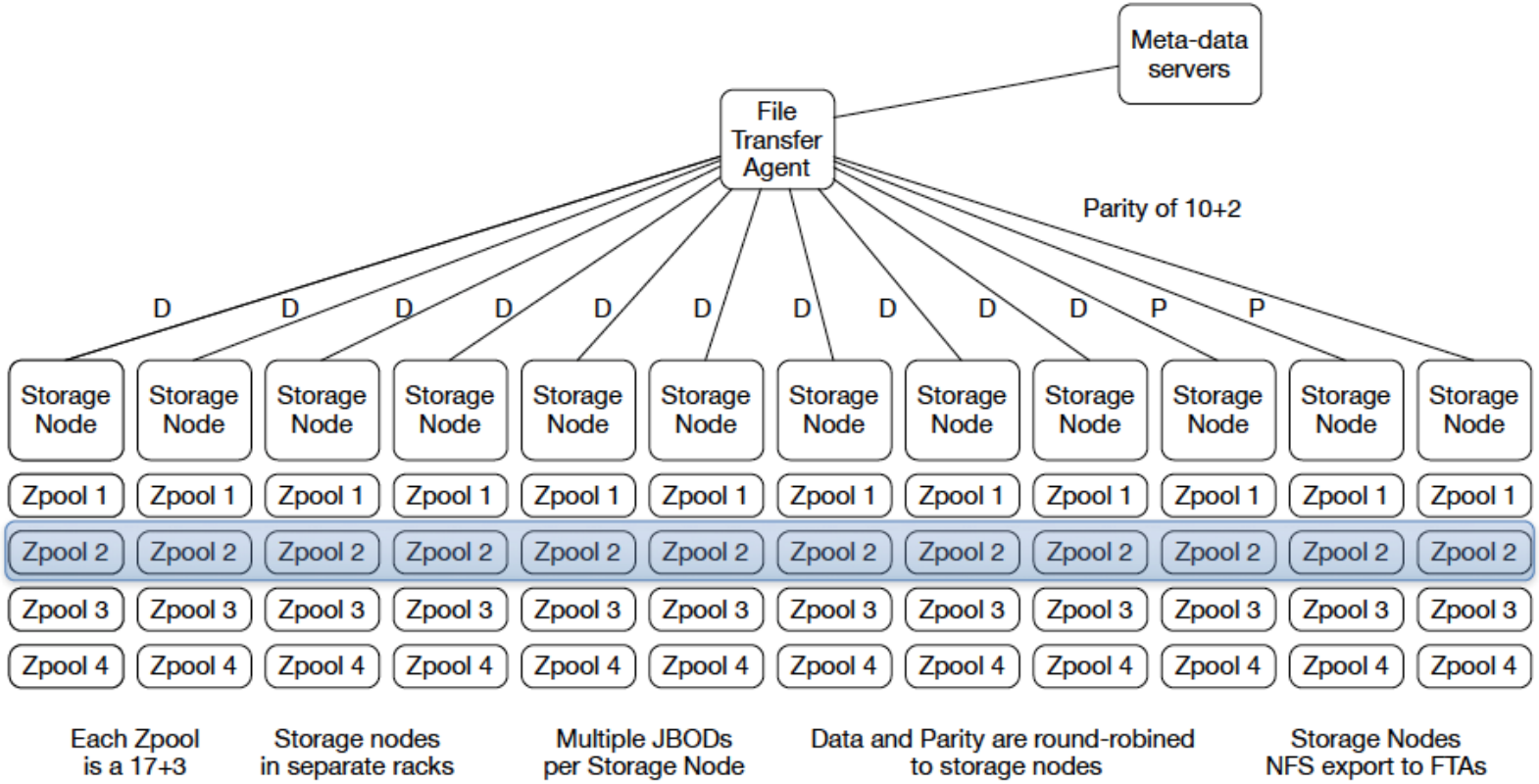
The Structure of MarFS

- Pftool – parallel file transfer utility
- FUSE mount – provides user view of metadata
- DAL/MDAL – Abstraction layers atop data and metadata respectively
  - Allow easy swapping of underlying storage



Multi-Component

- Cross-server erasure coding atop ZFS pools
  - Allows failure tolerance at both the disk and server level
    - More reliability allows the use of cheaper disk
    - Erasure coding performed through Intel's Storage Acceleration Library (isa-l)
  - Performance through parallelism
    - Threaded I/O to multiple servers



Thanks to Dave Bonnie/Kyle Lamb

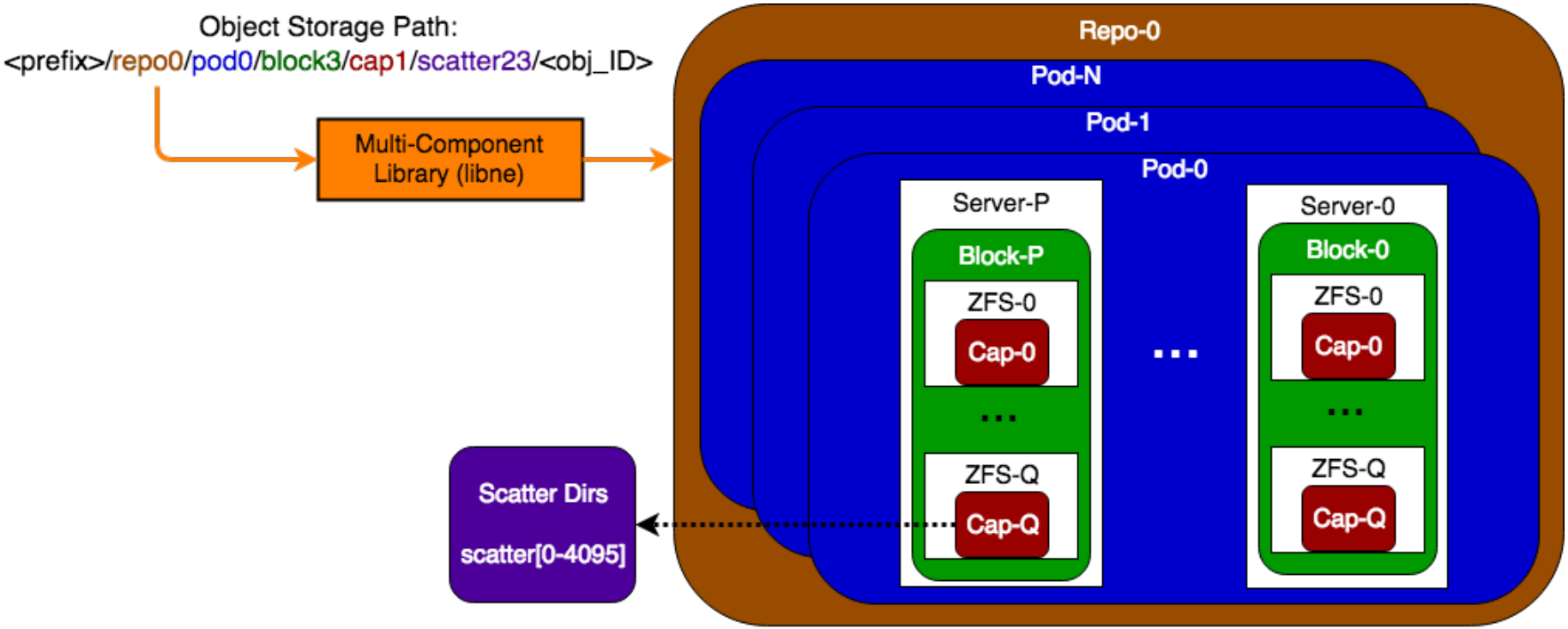
Multi-Component

- Load-Balancing
  - Objects hashed across available servers
- Highly Available
  - Accessible even after losing entire servers
- Transparent Architecture
  - Object stripes stored as files in ZFS

Multi-Component: Load-Balancing

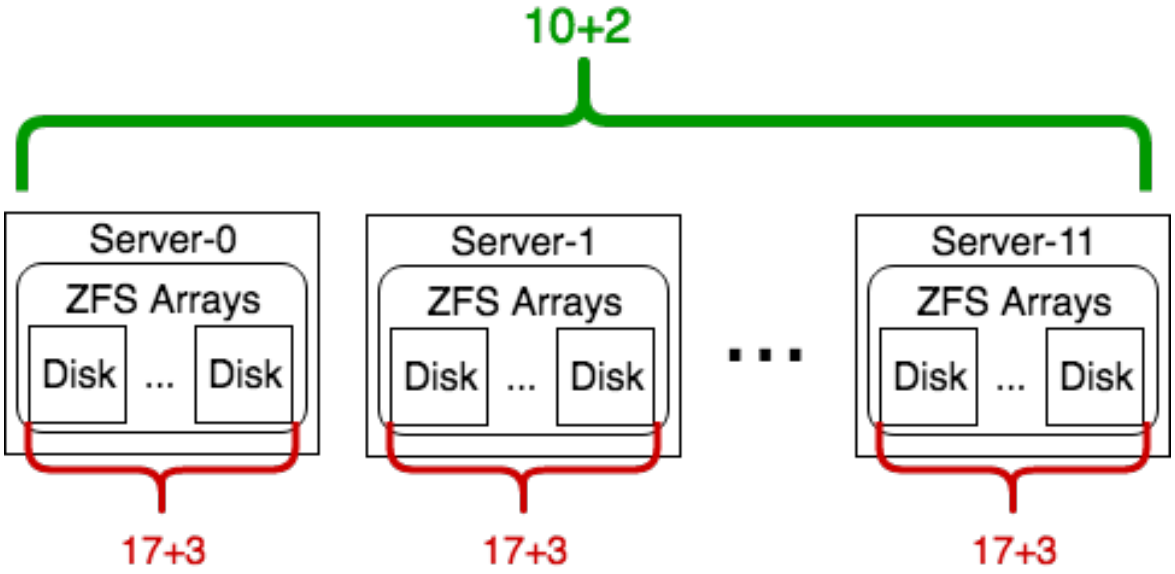
- Objects are balanced across a multi-level tree
- …/pod#/block#/cap#/scatter#/…
  - Pod: Corresponds to a server set (12, for 10+2)
  - Block: Corresponds to an individual server
  - Capacity-Unit (Cap): Corresponds to a ZFS Pool
  - Scatter Directory: Logical subdivision to avoid overloaded directories

Multi-Component: Load-Balancing



Multi-Component: Highly Available

- Multi-tier erasure
  - Multi-Component: 10+2 across servers
  - ZFS: 17+3 across disks



Multi-Component: Highly Available

- The Multi-Component Library (libne) automatically responds to read failures and makes use of erasure code if necessary
- ‘Degraded’ writes are permitted up to a configurable safety threshold (i.e. 10+2 can be written as 10+1 or 9+2 and later rebuilt to full width)

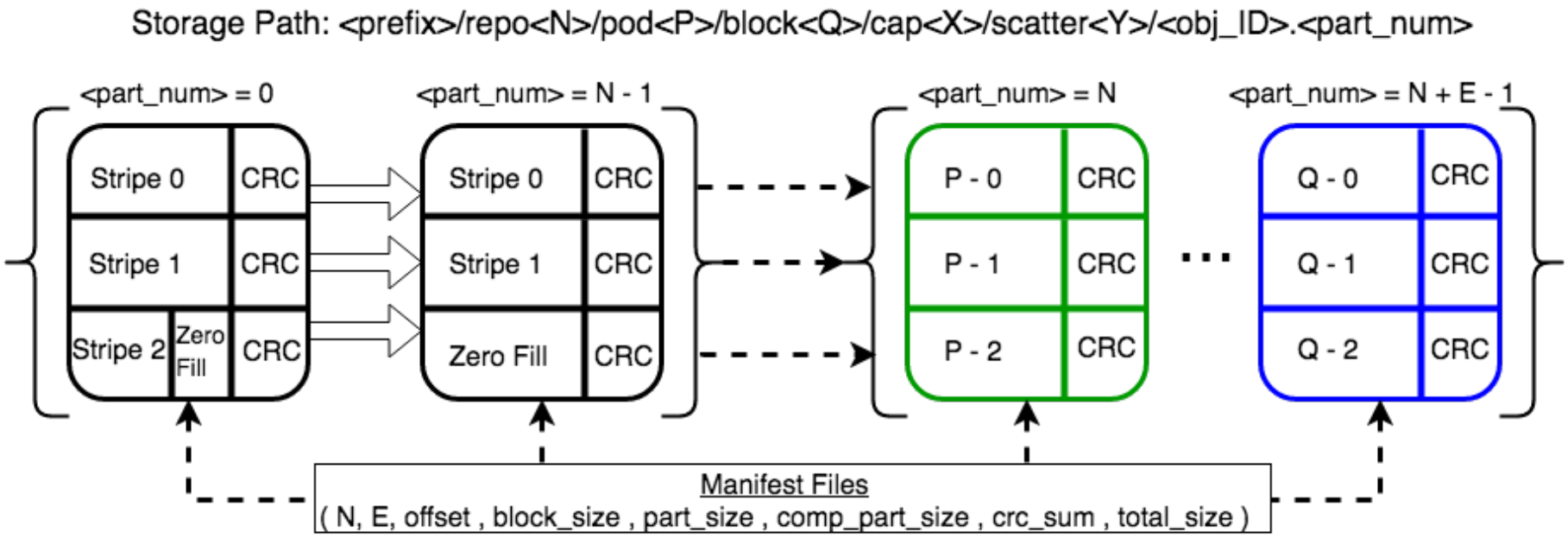
Multi-Component: Highly Available

- Any degraded objects detected during read/write are logged
- A utility crawls these logs and calls into the Multi-Component Library to rebuild objects
  - The rebuilder utility can also be run against entire capacity-units to assess overall health or recover from catastrophic failures

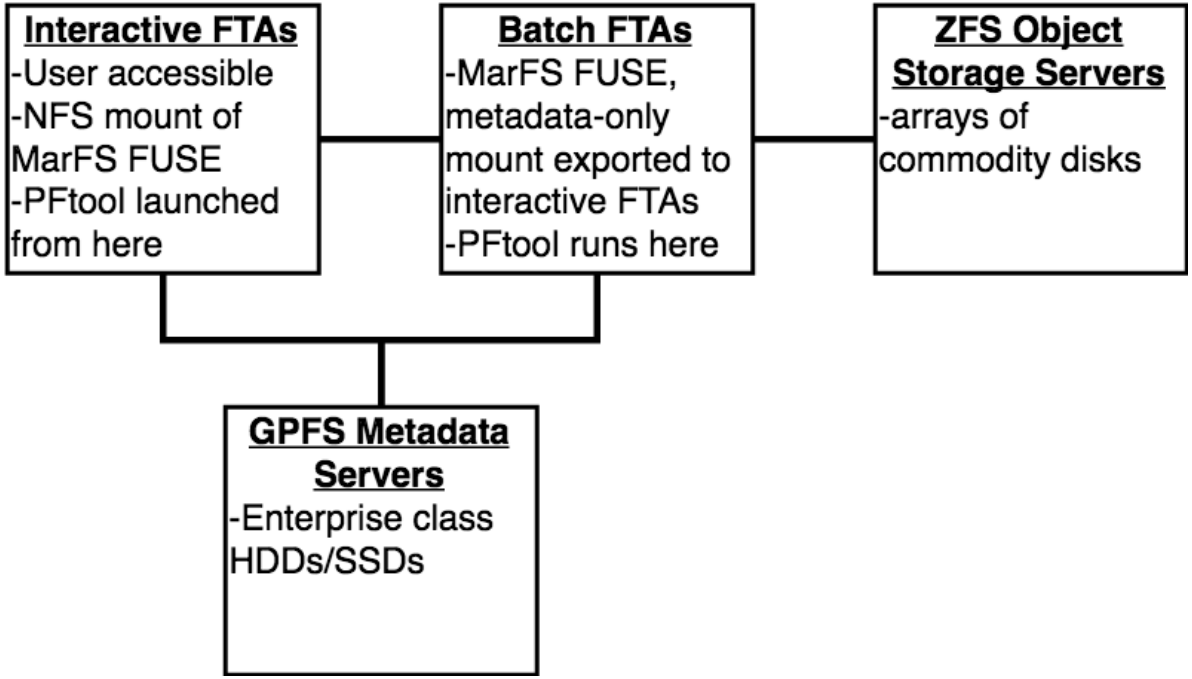
Multi-Component: Transparent Architecture

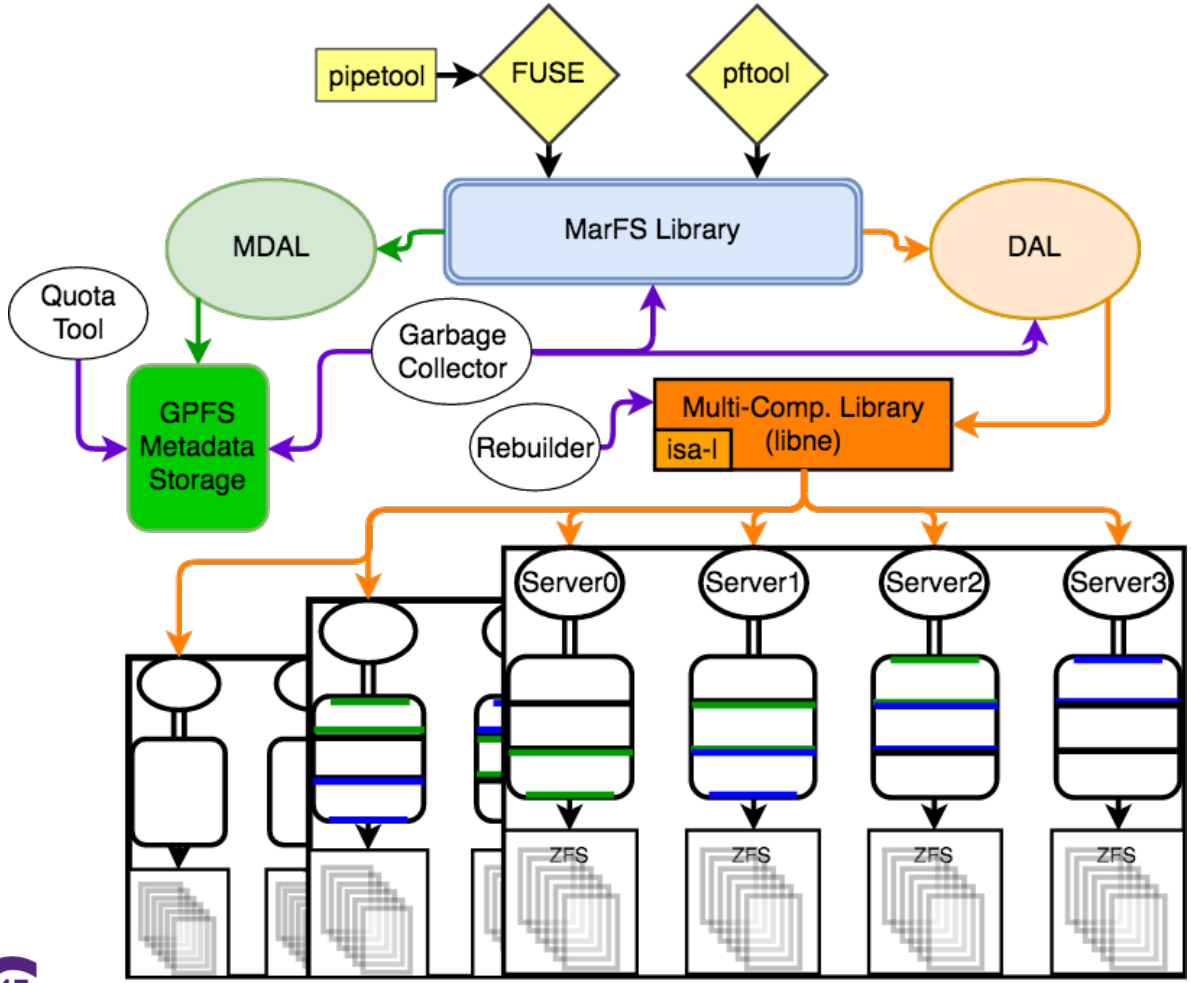
- Storing to ZFS systems means that objects are plainly visible to administrators
- Each object ‘part’ is paired with a manifest file, providing object and erasure structure info
- Multi-Component provides utilities for directly reading/writing objects and analyzing their health

Multi-Component: Transparent Architecture



Deployment





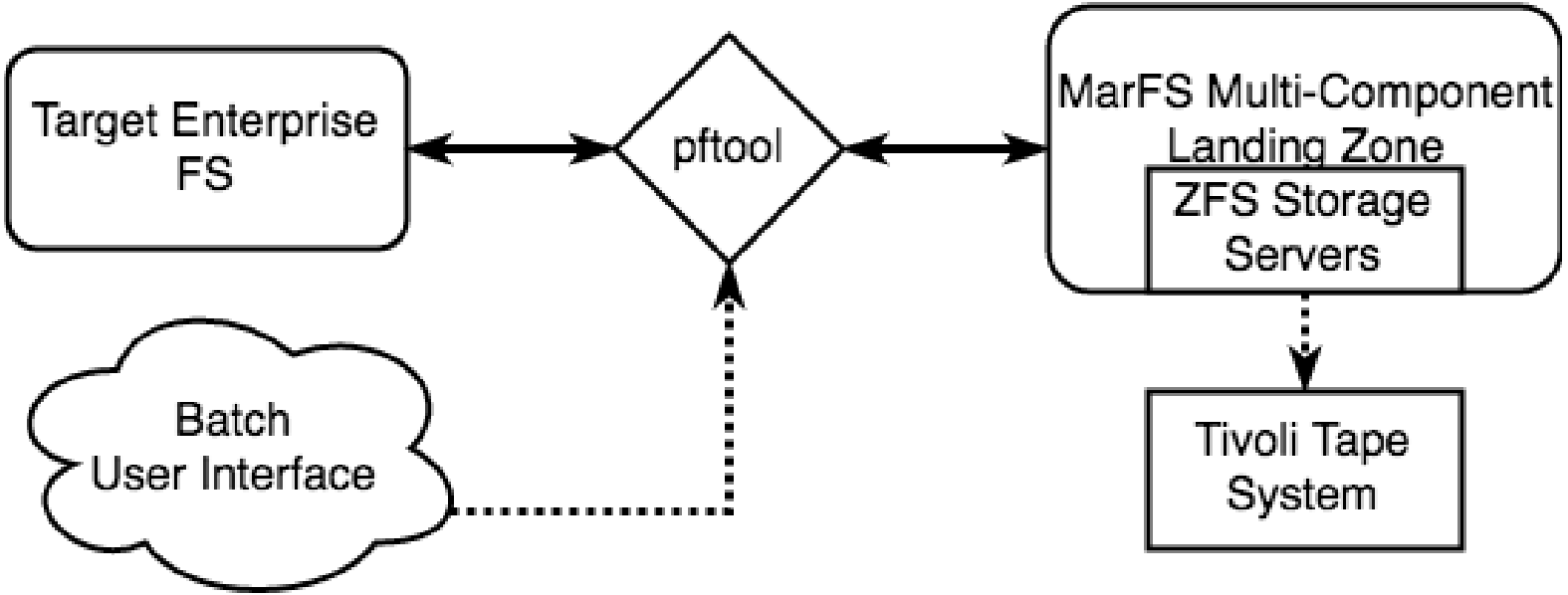
Current Production System

- 5 interactive and 25 batch FTAs
- 60PB of total storage
- Roughly 25GB/sec aggregate bandwidth for both read and write
- NFS seems to be the bottleneck

Future Work

- Infiniband RDMA support for Multi-Component
- Better administrator controls
- Capacity-unit migration
- Easier (and better documented) system setup/configuration
- Could a MarFS system, backed by tape, act as next-generation archive?

Potential Archive Implementation



LA-UR-17-26375