A Geo-Distributed Active Archive Tier for iRODS

Earle F. Philhower, III, Technical Marketing, WD
earle.philhower.iii@hgst.com

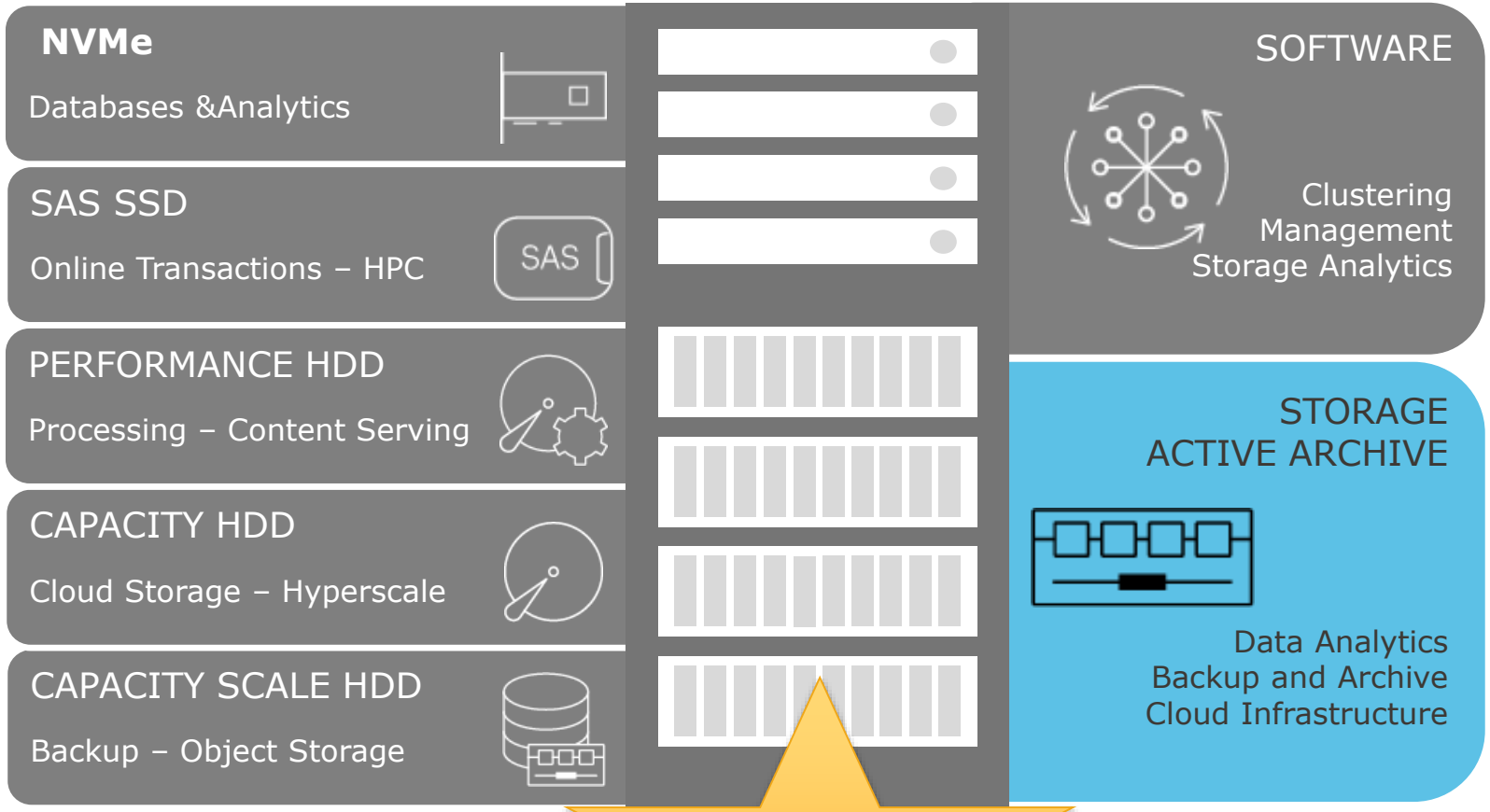
June 2016

Agenda

- Introductions
- The Big Picture
- Object Storage Quickie
  - HGST Active Archive Introduction
  - Geographic spreading of objects for DR, HA
- Using Active Archive with iRODS
  - Compound resources
  - New and improved S3 resource plugin
  - Sample architectures
- Performance Comparison
- Future Work

HGST Is Storage

and have been for a looooooong time…



In 1956, we invented the world's first hard drive. We didn't stop there…

Today, HGST innovates at every level of the storage stack – from the fastest solid-state drives to the densest storage systems on the planet.

The Big Picture

Why is this important to you?

- Add petabytes of storage capacity to existing iRODS
  - 672TB minimum, 30PB maximum raw capacity
  - Without the difficulty, cost, or support troubles of roll-your-own solutions
- Transparently migrate TB off of NAS
  - Whether the file is on an active archive or a filesystem resource hidden by iRODS
- World-class reliability, availability, durability, and ease of use
  - 15-9s, background data scrubbing, multiple failure tolerance, geo-redundancy

iRODS +

Object Storage in 60 seconds

Stuff you might not yet know but were afraid to ask

- Standard POSIX apps need not apply…
  - Immutable, no fseek/fwrite/append on objects. Objects are always either not present or fully present (i.e. partial file writes not possible)
  - No filesystem, everything is an object referenced with a GUID
  - RESTful – easy to use, well defined HTTP interface
- But there are benefits…
  - Erasure encoded (HGST AA, Ceph) or replicated (Amazon, others), not RAID
    - RAID on 10TB+ drives == :(
    - EC / replication provides data durability >> RAID
    - +++ much less space overhead than replication
  - Scalable to billion+ objects
    - Most filesystems fall over at these #s

Examples: Ceph, Swift, HGST Active Archive, etc.

HGST Active Archive System

- Complete scale-up and scale-out object storage system

Breakthrough TCO

Linear Scale Performance

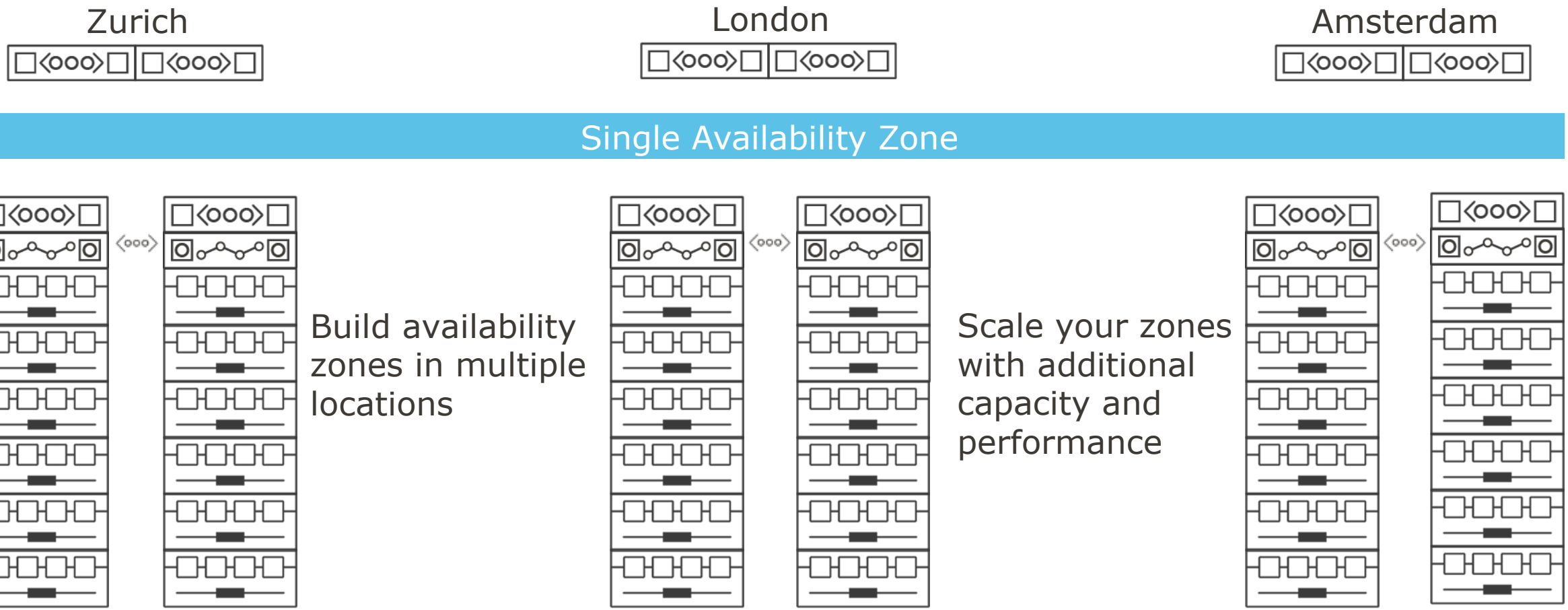
672TB-4.7PB Raw Capacity

Unbreakable Durability

Simplified Management

Geographic Spread for Disaster Recovery

Immediately consistent for reliability, durability, and sanity.



Using an Active Archive with iRODS

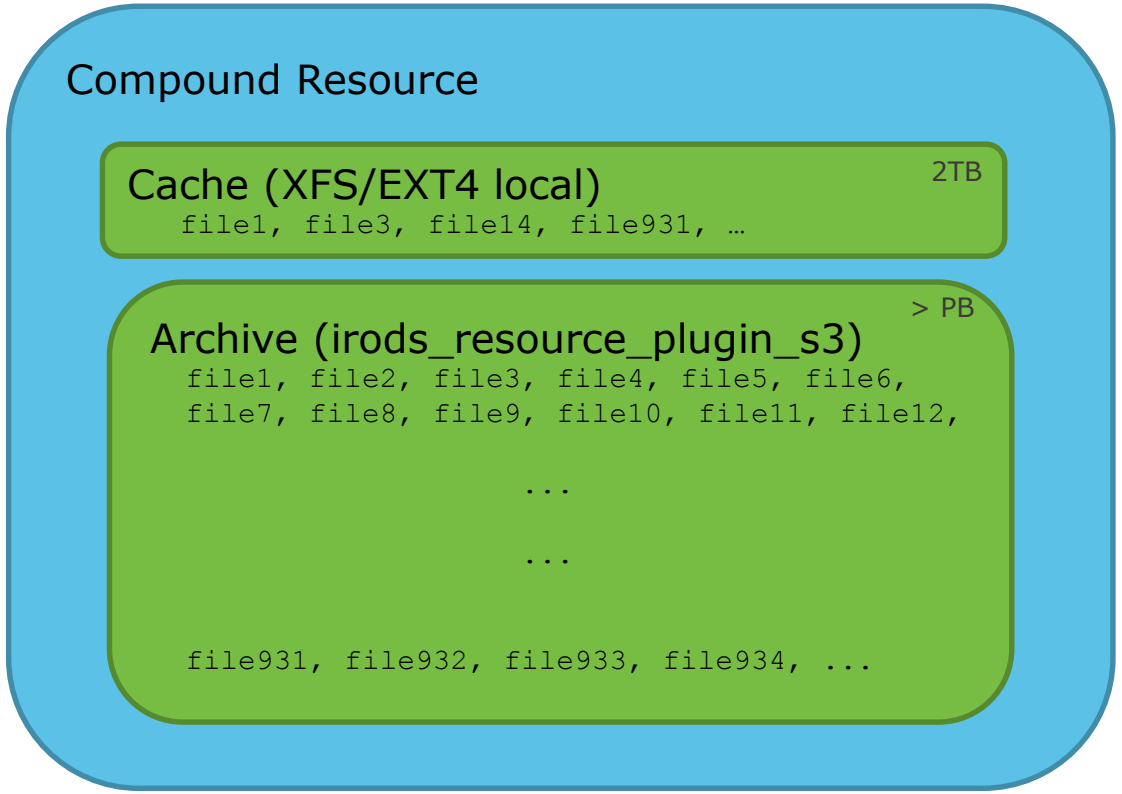
Compound resources to the rescue

- Compound resources convert Object => File in a Resource Server
  - Cache (POSIX ops happen here)
    - Local SSD, preferably NVM Express based
  - Archive (S3 Connector)
    - S3-interfaced backend (or other object protocol)
    - Permanent storage, sync to/from the cache
- iRODS replication allows seamless addressing of Archived files
  - iRule to place files on compound resource initially have 2 replicas, one on cache and one in Archive
  - Cached replica may be deleted to free space for new files
  - When files referenced again, a new replica from the Archive is generated
- Seamless integration with rest of iRODS infrastructure, S3 applications
  - Users don't know they're really talking to an Active Archive
  - S3 based applications can use archived file objects as-is (non-proprietary format)

Compound Resource

iRODS management of archive limitations

- Two replicas of all files
  - Cache – Transient but versatile
  - Archive – Permanent but limited
  - Auto-migrated by Compound resource
- Cache
  - All iRODS POSIX operations execute here
  - SSD / DRAM filesystem
    - NVM Express best (2++GB/s)
  - Manual/scripted/rules-based itrim'ing
- Archive
  - S3 Resource Plugin (or others)
  - stageToCache/syncToArch



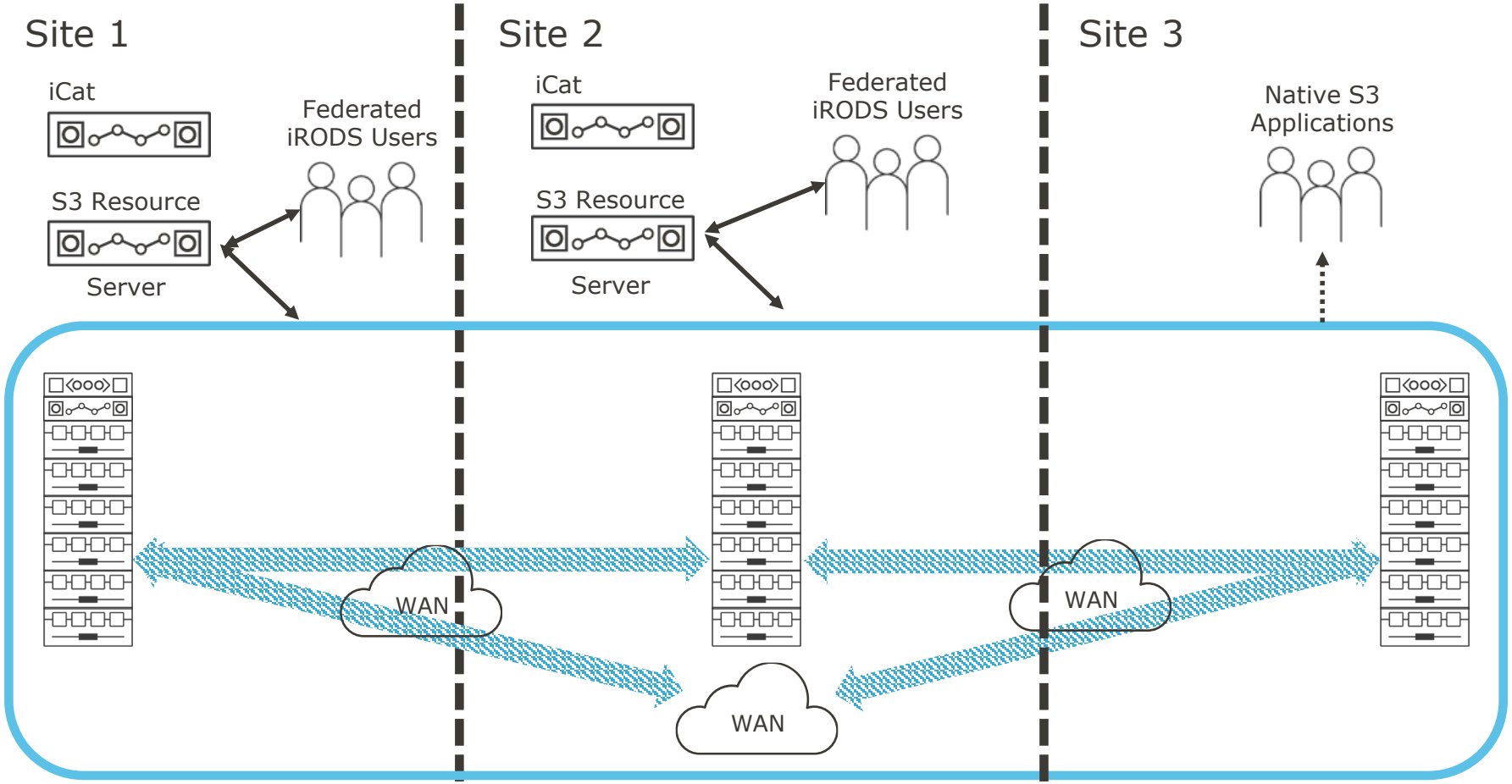
Updated S3 connector

Why and how it's been upgraded, where can it be used

- Existing S3 connector was mostly functionally correct, but…
  - Slow, single-threaded, large file issues, no checksum or encryption support
- S3 update (merged in iRODS 4.1.9 release)
  - Fully generic, work on **all S3 compliant Active Archives/web services**
- Speed
  - Multiple endpoints, parallel threads, multiple parts used for both iput and iget operations
  - Up to 2GB/s from a single resource server to a local HGST Active Archive
  - Cloud service providers should also see improvements (but limited to your uplink, of course)!
- Reliability
  - S3 protocol-based MD5 checksum to ensure integrity over the wire
  - 64-bit file operations support effectively unlimited file sizes
  - S3 server-side encryption specifiable for workloads that require it

Geo-Dispersed Architecture

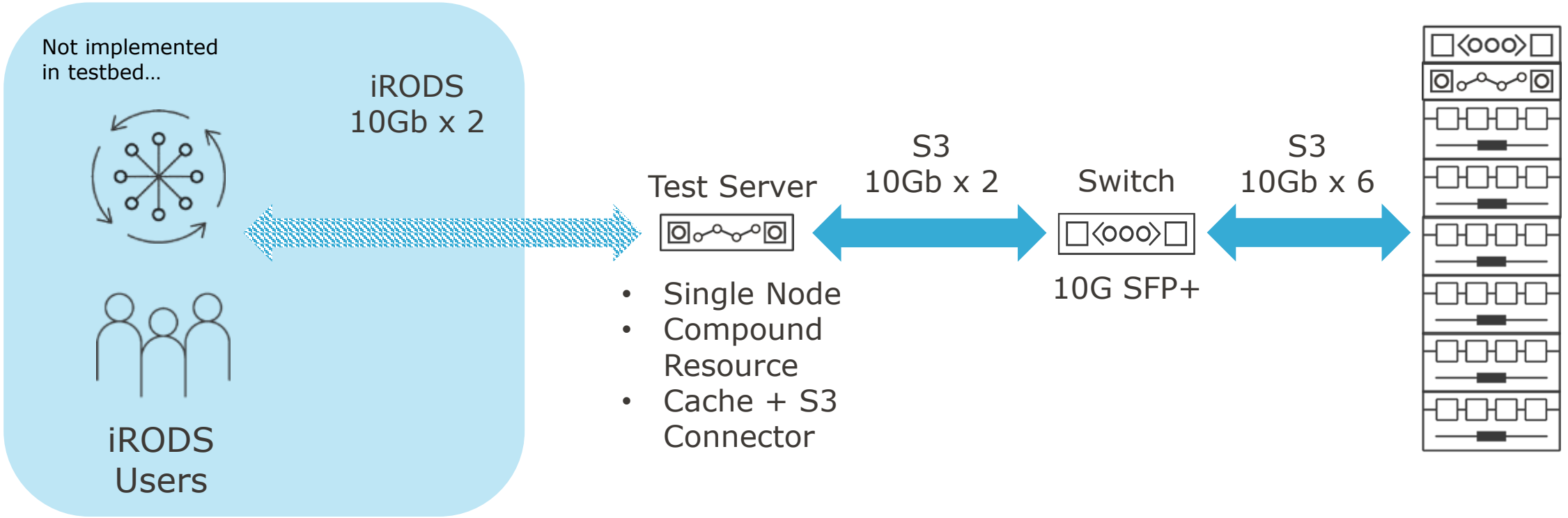
Multiple-campus availability, redundancy



Immediately consistent, globally shared Active Archive object store.

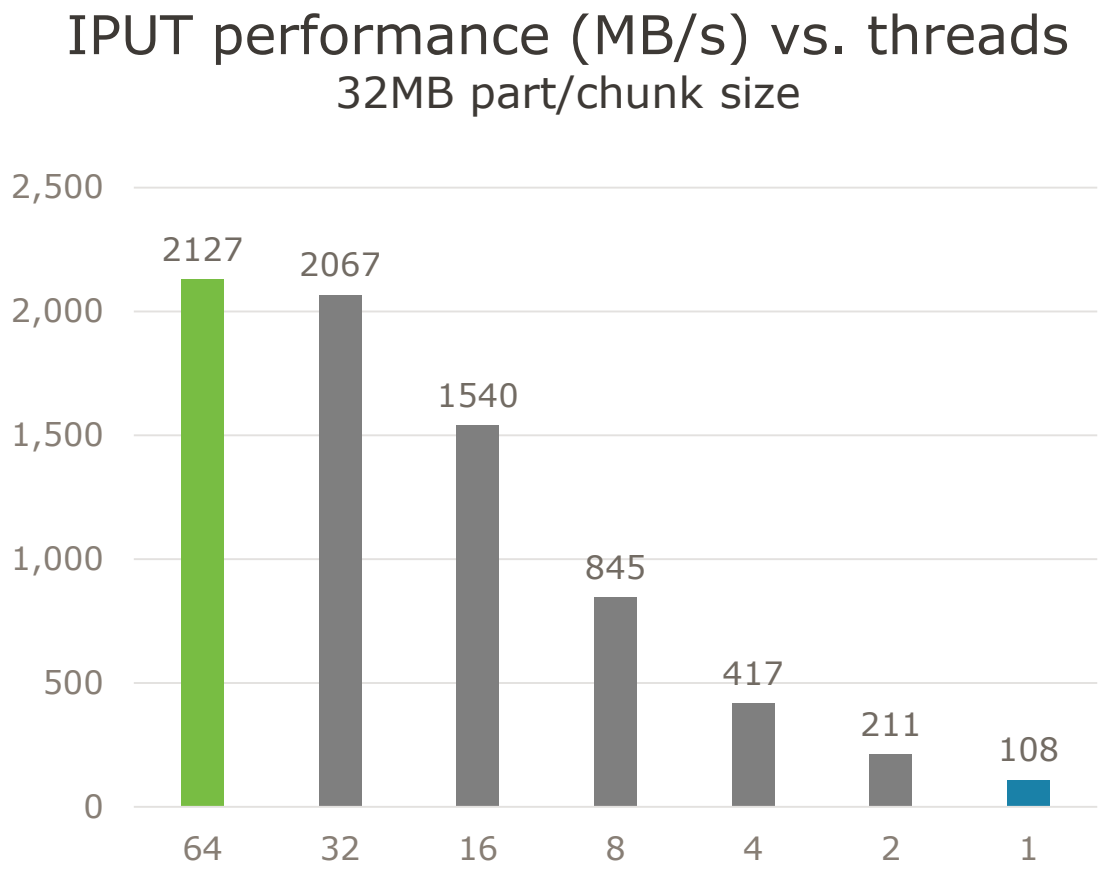
Sample 1-Site Tested Architecture

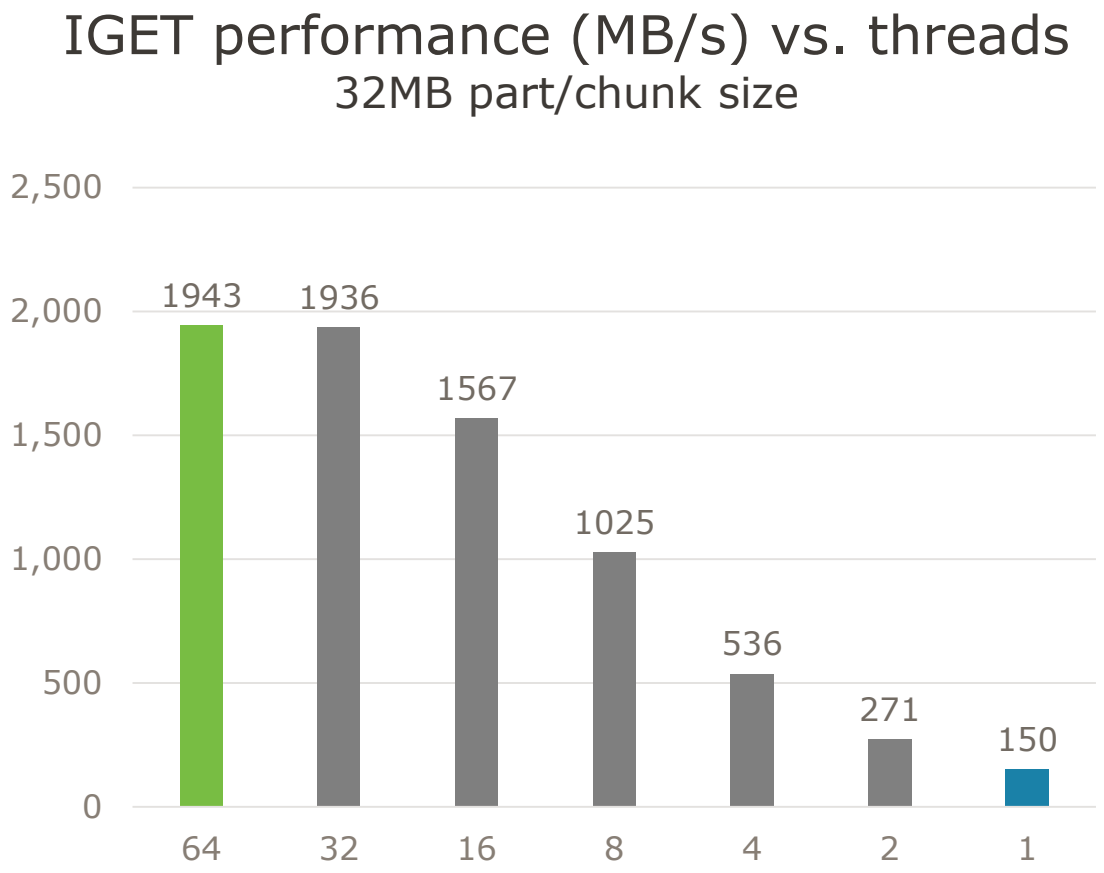
In-lab setup, simplified to isolate Archive performance



Test Results

Single iRODS resource server, 2x10G interfaces, 1 HGST AA





Future Work

Make it more compatible and easy to deploy

- Add V4 authentication for the iRODS S3 connector
  - Necessary for some Amazon availability zones, other S3-based Active Archives
  - Update LIBS3 to include V4 authentication?
    - Helps other open source projects using this simple framework, too!
  - Move to Amazon AWS C++11 SDK
    - Requires CLANG or very modern G++
    - May affect iRODS build environment substantially

- Generic cache and migration rule sets
  - Define generic rule sets for migrating existing data (maybe add ATIME to iRODS?)
  - Cache cleaning algorithm improvement (ATIME again would be helpful)

Questions?

Thanks!

Helping the World Harness the Power of Data with Smarter Storage Solutions.