# Some Questions Relating to Future Archive and Backup Systems

Bruce Montague, VERITAS

May 2023

VERITAS

# Agenda

- Review of *Enterprise* Backup and Archive
- Experiments with a non-traditional archive device, the Seagate CLiFF
- Questions (and speculation) about future archive storage devices

# *Enterprise* Backup

- **Enterprise backup** usually differs from backup in other computer industry areas
  - Backup core *structured* data (databases) & *unstructured* data (app/servers) in large enterprise
    - (not laptops)
- Primary goal—***Restore*** a **consistent** backup as close to a desired point in the past as possible *(RPO)*, as fast as possible *(RTO)*.
  - RPO: Recovery Point Objective (when?); RTO: Recovery Time Objective (how long to get there?)
  - Need to be able to find things to be restored easily and rapidly (time-travel view is common)
- Typically based on fixed schedules: *daily, weekly, monthly, annual*, …
  - Fixed number of these backups kept; Backups often deleted after fixed **retention** period.
  - *Full, incremental, cumulative* backup
  - Most recent backups often most relevant
- Data lost due to: human error, system failures, bugs, hw failure, misconfigurations, malicious activity *(ransomware)*…
- Exact practice and infrastructure varies widely (old *3-2-1* rule)

# Backup (deeper dive)

- Output of a backup—a backup *image* (like a tarfile)
  - Typically produced by automatically scheduled backup *job*
- *Policy* (some sort of high-level instructions for an automated copy++ system)
    - Direct the centralized "brain" of the system: "who, what, when, where" across a backup *realm* or *domain*
  - Data *source*, output *target* (both usually specified as types, not as specific instances)
  - *D2D2T* (Disk to Disk to Tape). Complex image life-cycle copies and deletions.
- Backup *window*. (Few hours when backup has top priority; continuous operations means windows are no longer universal.)
- Backup infrastructure for entire organization (arbitrary security *domain*)
- Often knowledgeable of internal formats of files and databases
  - Must be knowledgeable about metadata—able to find things of interest
- Ability to *quiesce I/O* and generate *application-consistent* images
  - May require private APIs to flush all I/Os in application, OS, & hypervisor caches and I/O queues
  - Need consistent, immediately usable restore image, not a *crash-consistent* image

# Archive (aka, LTR—Long Term Retention)

- **Archive**—*not* the same as backup
- Primary goal—keep cold data as long as needed, without data loss, as cheaply as possible
  - Might be kept for legally mandated 5 or 7 years (after which data is to be securely deleted)
  - Might be forever (Library of Congress)
  - Cost is important
  - Need to be able to easily find archived data
- Attributes:
  - Cost: $/TiB
  - *Durability* (no data loss)
  - Usually archive/restore speed is not that important (which can reduce cost)
- Data might be archived only once, at the end of a project or period when it was generated and used
  - Data is often deleted from other storage after it is archived
  - It may be important to delete archived data at the end of its retention period
- Many archive devices: dedup hard-disks/flash, tape, optical-disks, arrays of idle disks, …



VERITAS™

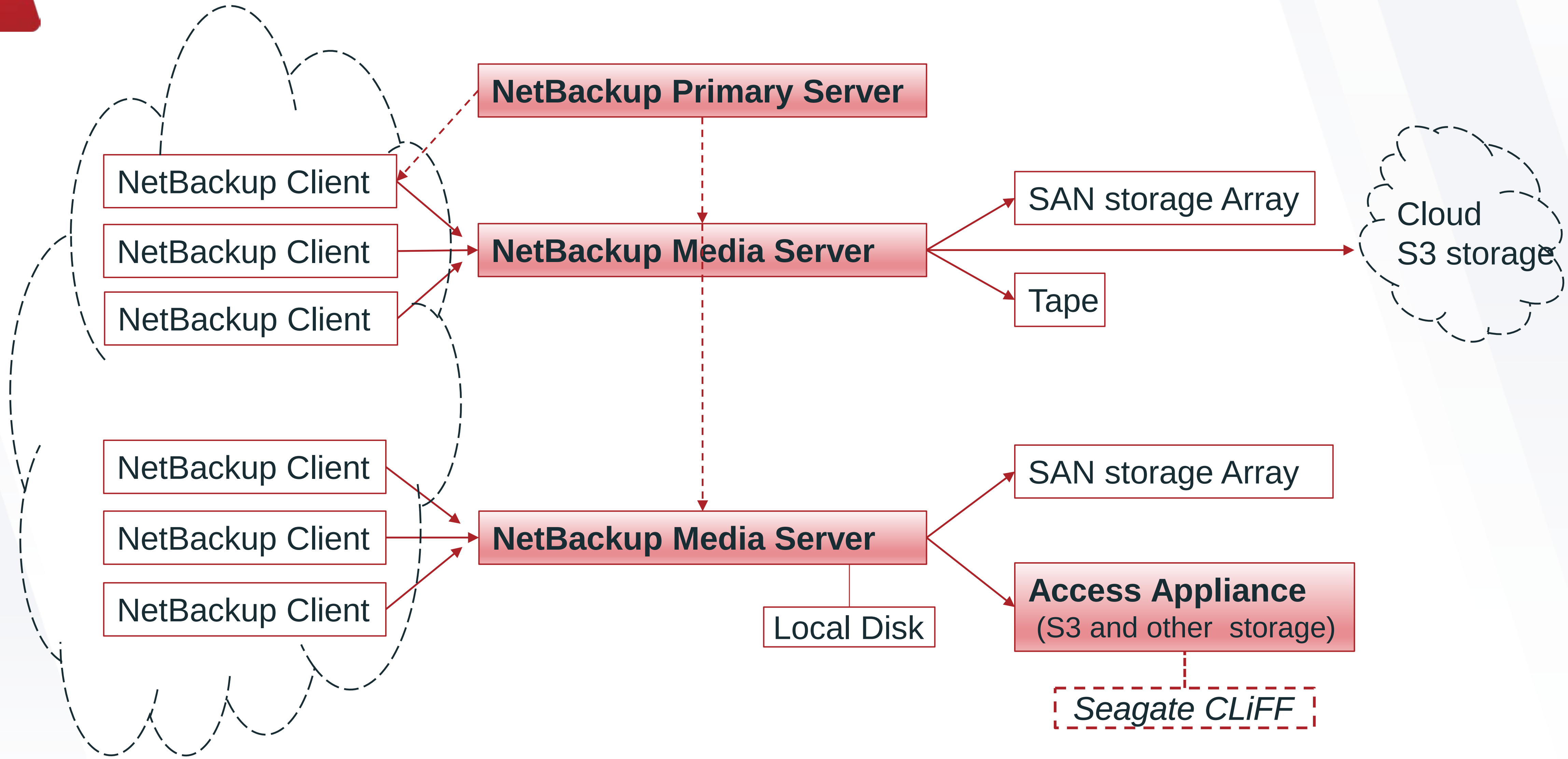
# Related Background (1)

- **Tape**. Not as dominant as it once was, but still common
  - Used with tape robots, carousels, …
  - Robots mechanically move tape cartridges from a library into drives, inventory the cartridges, …
  - *Offsite backups* are external to a datacenter. May not be in a datacenter at all.
  - *Vaulting* systems—backup/archiving jobs write to tape robots, manage the robot, print barcodes for tape cartridges, and track cartridges throughout the backup infrastructure (including offsite)
- **Dedup**. Backup data often changes little from the previous backup of the data (say, a directory)
  - Dedup only backs up what has changed… but need to keep metadata to track it all
- **DR** *(Disaster Recovery)*. Intercept all local I/O and re-play at remote location.
  - *Fail-over* to the remote location, *fail-back* to the primary
  - Often integrated with backup infrastructure
- **Enterprise Backup Infrastructure**
  - Have databases that track all backup requirements (policies), all the resources, current state, etc.
  - Do best-effort multi-dimensional scheduling, schedule work into available windows and resources…
  - Deal with entire backup image lifecycle (multiple copies, retention cycles, …)
  - Can resemble networking *middleware*—support VLAN trunking for dedicated backup networks, FC

# Related Background (2)

- **Cloud Backup**
  - In early cloud days application I/O could not be quiesced end-to-end on-the-fly. This led to a backup-style directed by the application (or code co-located with the app) that made explicit use of cloud REST APIs.
    - Today: distinguish backup-to-the-cloud and backup-in-the-cloud
  - Large distributed databases make their own backup images when requested. Backup systems collect these images.
  - *Snapshots* are a mechanism used by backup systems, but by themselves are not backup systems
- **High Availability *(HA)* Clusters**
  - *Availability*—works when needed
    - Often two classes of backup appliances: HA (for backup of key resources) and normal (reduce cost)
  - HA Clusters are different than parallel computation clusters, VM clusters, etc.
    - HA is not the same as Fault Tolerant (FT). FT masks failures.
    - HA restarts applications when they fail (if app or node fails, the cluster restarts the app on another node)
    - HA clusters historically used shared storage. All nodes had the same view of this shared storage.
      - Require storage management (volume managers, filesystems) that are cluster aware
      - 80% of HA clusters historically have likely been only two node clusters
  - Many HA clustering systems today descend somewhat from Cornell’s 1980s-era ISIS research system

# Veritas NetBackup Infrastructure



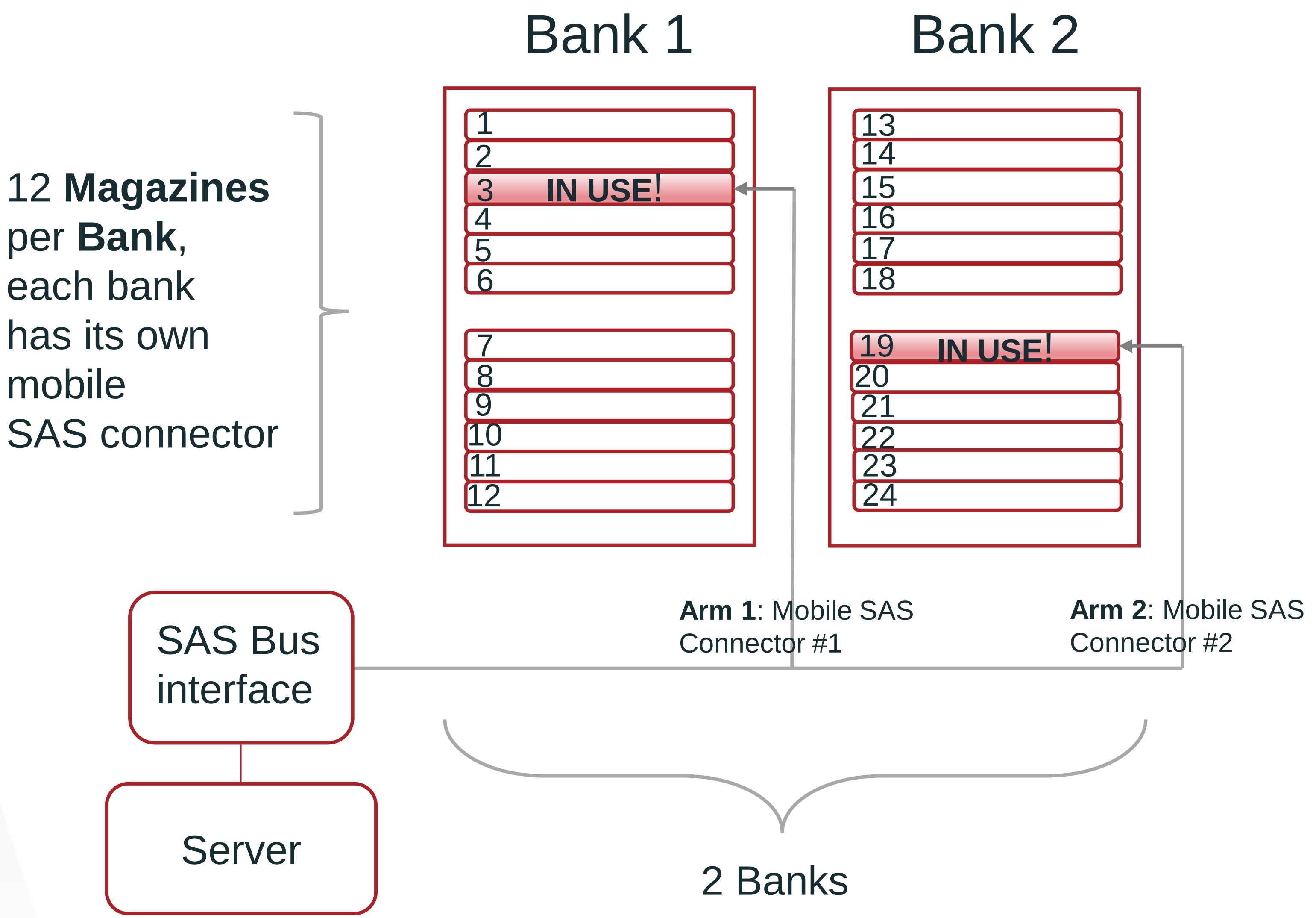



VERITAS

# Veritas InfoScale (an HA Cluster I/O Stack and HA Application Manager)

| Component | Name | Purpose |
|---|---|---|
| **VCS** | Cluster Server | HA cluster orchestration and management (including HA apps) |
| **CFS** | Cluster File System | Enables VxFS to provide a shared filesystem view across a cluster |
| **VxFS** | File System | File system optimized to run on top of VxVM and CVM |
| **CVM** | Cluster Volume Manager | Enables VxVM to share storage and volumes across a cluster |
| **VVR** | Volume Replication | Replicates volumes between InfoScale sites (WAN replication) |
| **VxVM** | Volume Manager | Combines disks into HA storage pools, creates logical volumes in pools |
| **DMP** | Device Multi-pathing | I/O driver, manages multiple HA connections to same storage devices |
| **DDL** | Device Discovery Layer | Finds and uniformly names storage devices across an HA cluster |

# The Seagate CLiFF—A prototype archive device




- Each magazine contains media for 5 hard disks (each disk 16TB)
- Some per-disk electronics is 'factored out' (not in the magazines)
- When a magazine is connected, all disks in the magazine power-up
- When a magazine is disconnected, all disks in the magazine power-down
- Only two magazines can be connected at the same time (only 10 disks)

# The Seagate CLiFF: Experimental Data—time to switch magazine

| Operation | Time | |
|---|---|---|
| **Mechanical Initialization:** | 4 to 30 seconds | |
| **CONNECT magazine:** | 15 to 20 seconds | |
| **DISCONNECT magazine:** | 24 seconds | |
| **DISCONNECT&CONNECT: (combined command)** | 42 seconds | Switch operation 1. |
| **CONNECT COMPLETE: (time for LUNs to appear)** | 25-34 seconds | Switch operation 2. |
| **Disk ready completion time:** | ~20 seconds | Switch operation 3. |
| **Total Magazine Switch time:** | ~1min 30 sec | |

VERITAS

# The Seagate CLiFF: Lessons Learned

- Mechanical engineering is hard! Mechanical connects were much less reliable than typical computer device electronics.
- The InfoScale DiskGroup (DG), implemented long ago to support FibreChannel fabrics (which can dynamically connect & disconnect FC disks), was a good match to the magazine concept. The DG potentially enables long-running storage operations to span magazine disconnection and reconnection.
- End-to-end disk connection and spin-up time was not markedly better than tape access time. The device as-is is thus useful when random I/O is expected. As evidenced by USB, work on the software stack here would likely reduce this by eliminating the final 20 seconds (Disk Ready completion).



VERITAS™

# Some Problems in Archival Storage

- **Determining expected data loss of complete complex systems** (prior to building systems)
  - **Building accurate complex system models** (including rare event simulation)
  - **What is a good metric for complex system durability?** (Many components; Real-world system MTTDL?)
- **Determining real-world complex system data loss** (Legal issues: anonymous FAA-type incident reporting?)
  - **Testing large (PB-scale) storage systems** (that take months to reach interesting states; state injection?)
- **How to get basic device characteristics (manufacturer's AFR)** (bundle updateable doc in the device itself?)
- **Dealing with partial, intermittent, and performance (slowdown) device failure**
- **Trusting storage devices and, vice versa, enabling storage devices to trust storage systems**
- **Upgrading both complex systems and devices (including device documentation, models, and firmware)**
- **Migrating storage, perhaps by physically migrating device cartridges**
- **Converging cloud-storage and on-prem storage solutions**
- **Usability—improving the administrator-to-server ratio of storage-related systems**
- **Offsite debugging real-world field problems—is current log analysis practice all there is?**



VERITAS

# The “Perfect” Archive/Backup Storage Device?

- Perhaps in the future we might see:
  - A removable fabric-based device with:
    - A large amount of storage and random I/O
    - Low cost, low power, can readily be powered on/off, long-lasting in powered-off state
    - Weight, physical size, and portability akin to tape cartridges or laptops
  - Redfish/Swordfish, etc., compatibility
  - A *standard ancillary interface* (perhaps tunneled) providing:
    - Datasheet and other device documentation ***(self-documenting)***
    - Code for basic models of the device itself, bundled in a standard vendor-neutral format ***(self-modeling)***
      - Contains availability, durability, performance, power, and economic models
    - Current status of the device, relating to the bundled models
    - Vendor-neutral IoT-device-type logging, with some in-device analysis ***(self—log-analysis)***
    - A firmware upgrade standard, providing update of bundled firmware, models, and documentation
    - A versioned interface that, in-principle, is backwards compatible forever

VERITAS