



Things to Consider When Planning for File Services in a Hybrid Cloud Environment

Bernhard Behn
Avere Systems, Inc.



Objectives

- Identify the challenges that hybrid application and storage environments bring
- Discuss some of the infrastructure plumbing, networking, security, and most important: storage
- High-latency and limited-throughput will affect usable application performance
- Striking the tricky balance between local on-premises and cloud, compute and storage, all parts in motion

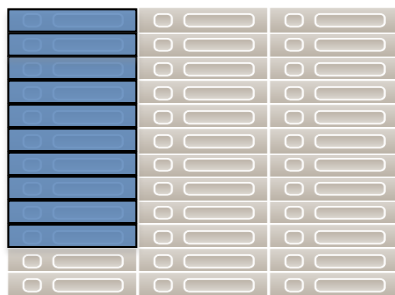
Challenges

- Historically, applications and their file storage resources were coupled in the datacenter
- Hybrid cloud is a dividing force between these coupled resources as compute and their data go their separate ways
- The ability to keep your apps' separation anxiety at bay is going to dictate quality of experience running in the Hybrid Cloud
- These apps use standard NAS protocols; coincidentally terrible over anything slower than a high-speed LAN
- Despite high-speed private links, latency to cloud will be high
- Application filesystem requirements and cloud API compatibility don't always match up, requiring gateways, translators, or server/agent sync&share to be brought into the mix

Hybrid Cloud Elements

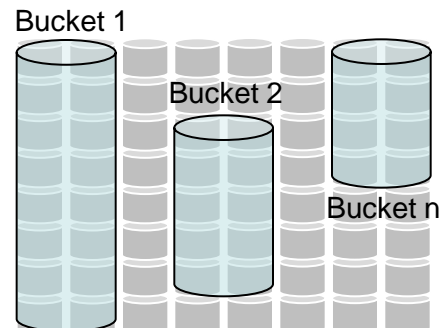
Public Compute Cloud

App servers



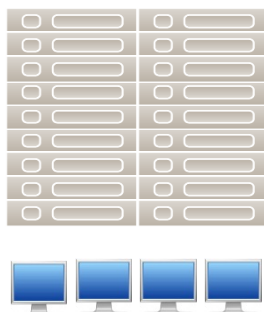
- Near infinite compute
- Cloud bursting
 - Permanent infrastructure

Public Storage Cloud

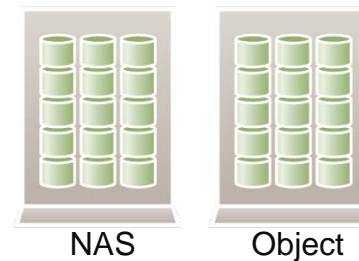


- Near infinite capacity
- Mostly backup and archive today

- App servers
- Compute farm
 - Desktops



On-Premises Compute



On-Premises Storage

- NAS and Object
- Multiple tiers of storage

- What does it take to deliver file services in a Hybrid Cloud? Networking, Storage & Security
- Network Setup
 - ◆ High-speed, low-latency, low-packet-loss
 - ◆ Either public Internet or private network links
 - ◆ IP routing with BGP/OSPF
- Security Setup
 - ◆ Directory Services management in the cloud
 - ◆ Encrypted VPN tunnel interconnect
- Storage
 - ◆ Where do we begin?

➤ File vs. Block?

- ◆ Let's get block quickly out of way/scope: it's generally consumed host:target at 1:1, and the cloud has largely solved this already

➤ Unstructured data and file sharing

- ◆ Your data used to be very close to your application, now the application is “roaming” or “remotely scaling”
- ◆ This is the challenge faced by Hybrid Cloud application deployments that share data across the cloudy enterprise

➤ Two practical approaches to unstructured file sharing

- ◆ Batched filesystem jobs: replicate + re-sync + conflict-merge
- ◆ Real-time remote access + proprietary optimizations

Batched Filesystem Handling

➤ Workflow efficiencies

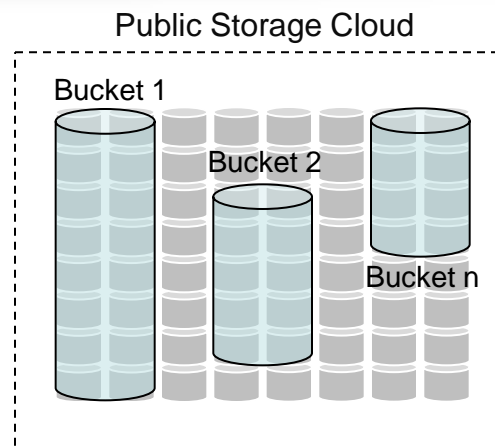
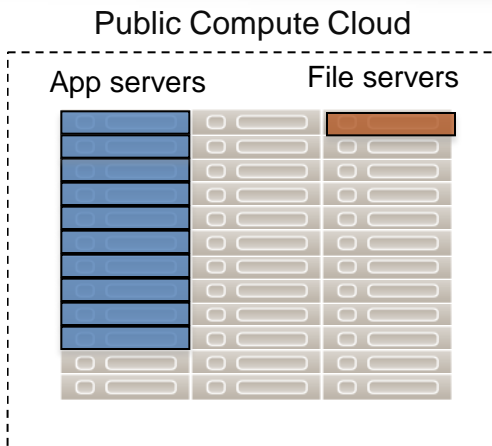
- ◆ Copying/distributing “golden” files, directories or volumes (tarballs, rsync, zfs send/receive, etc.)
- ◆ Can be automated, drag/drop, or 1-button replication operation
- ◆ Job-based granularity, scan & sync transfers changed blocks/file(s)

➤ Workflow inefficiencies

- ◆ Per-job replication passes are inherently asynchronous unless source is locked to read-only during scan & sync
- ◆ Remote storage required; cleaned-up remote copies will force full baseline scan & sync
- ◆ Determining changed files at remote site, merging back into master data, dealing with conflicting changes
- ◆ Job granularity management will become a scale issue (1:many)

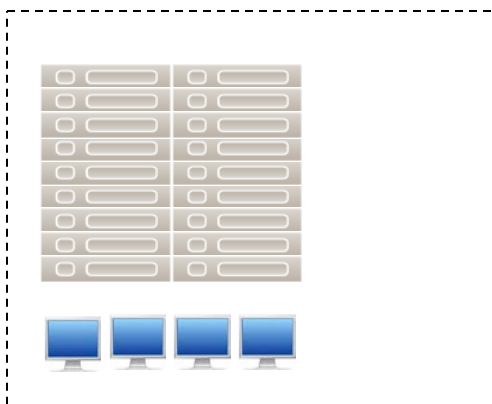
Batched Data Management

- File Server backed by block storage (\$\$\$)
- File server to receive target filesystem sync
- R/W changes being made local to replica

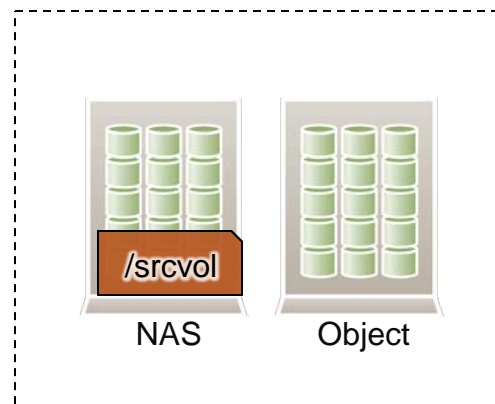


- The app may not be able to address this type of storage
- A gateway file server can store srcvol replica on object storage (\$)

- Consumers of srcvol data on local NAS
- Changes being made here R/W to source
- If merge conflicts are unresolvable, this site must be read-only



On-Premises Compute



On-Premises Storage

- /srcvol is the master data copy
- Replicate out to cloud file server or storage in other clouds

➤ Workflow efficiencies

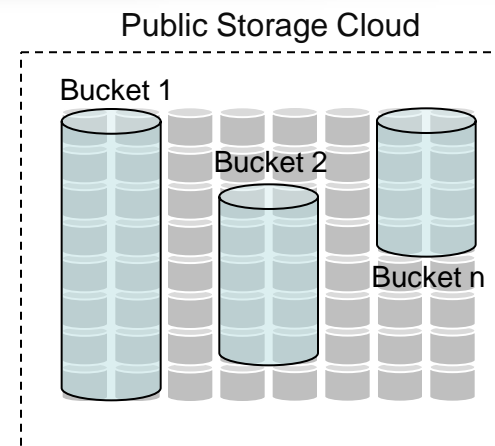
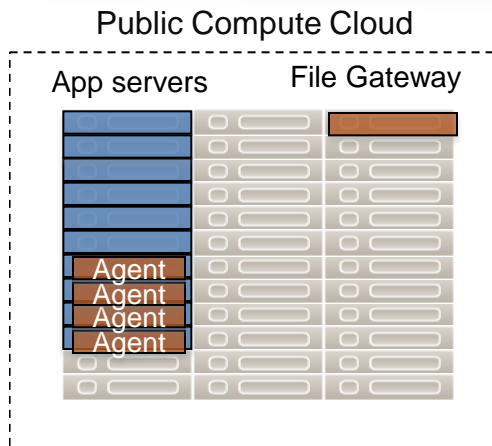
- ◆ Straight from the source; Go to server X for file Y
- ◆ Avoid storage growth by not replicating entire volumes
- ◆ Minimized bulk/wholesale storage transfer
- ◆ No versioning conflict resolution required

➤ Workflow inefficiencies

- ◆ Chatty NAS protocols really suffer over long-haul networks
- ◆ I/O heavy bandwidth requirements can scale unreasonably
- ◆ Optimizations involve proprietary clients/drivers per OS
- ◆ Gateways and agents have management scalability concerns
- ◆ Hotspot management concerns

Real-time remote access + optimization

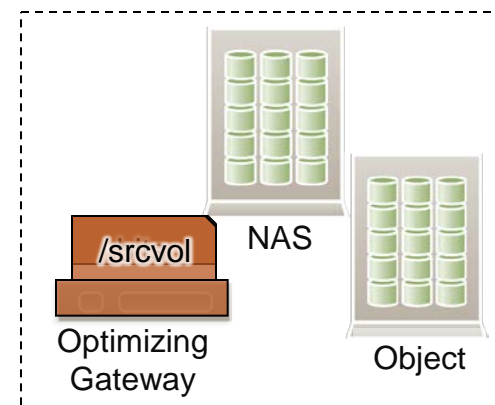
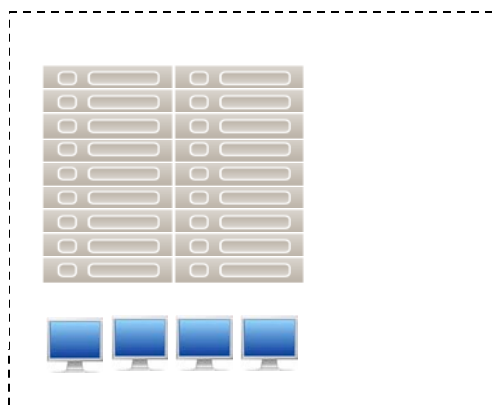
- Dedicated gateway to handle App requests (API/protocol?)
- OR-
- Agents installed on all App servers



- The gateway file server should cache /srcvol files on cloud object storage (\$)

Filesystem data is cached/managed by gateway/agent

- Consumers of srcvol data may need an agent installed
- Local gateway may be a bottleneck



- /srcvol is the master data represented by gateway
- Allow/disallow local RW changes?

What's killing my cloud app?

- Storage performance is the likely killer for otherwise healthy (no CPU/network bottleneck) applications running in the cloud
- Many tradeoffs to mitigate the storage performance impact
 - ◆ Response time, latency, throughput
 - › Bulk data synchronization or caching techniques help
 - › Consistency and scalability become more difficult
 - ◆ Consistency
 - › Centralized data repositories or agent-based solutions
 - › Application compatibility, and scalability are issues
 - ◆ Large-scale Multi-site Scalability
 - › Data-sync or caching techniques, multi-site scale-out clustering
 - › Complexity will increase as demands for consistency increase, costs can stack up
 - ◆ Application Compatibility
 - › Standard NAS protocols, kernel drivers + agents, POSIX filesystem semantics
 - › Agent management overhead, scalability and consistency can suffer here

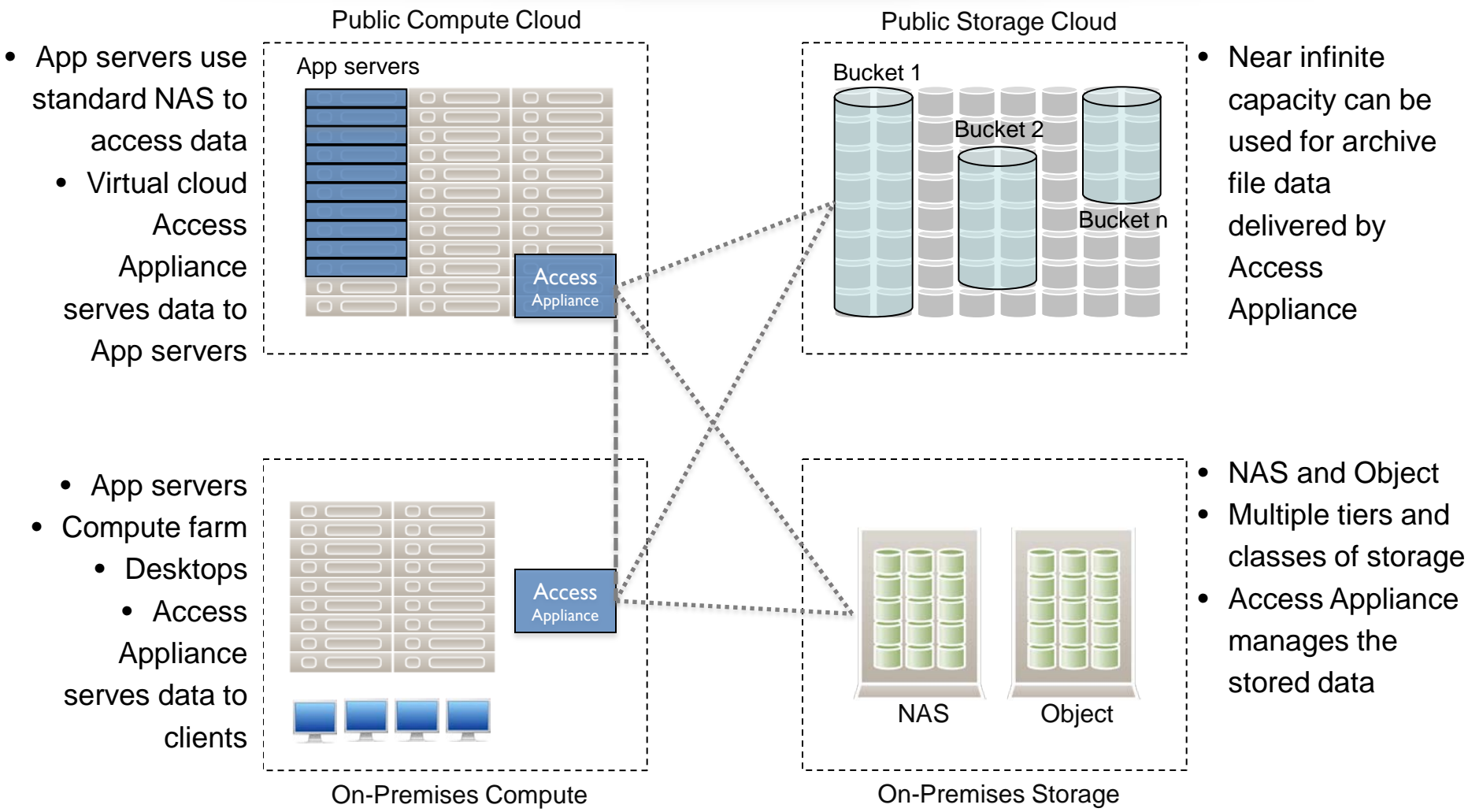
Current Cloud App Enablement

- **Batched replication, infrastructure/data management overhead**
 - ◆ Extra filers – replication targets to serve data to clients
 - ◆ Extra disk capacity – remote filesystem replicas can be large
 - ◆ Bigger network pipes – bandwidth intensive coarse replication
 - ◆ Re-sync/reconcile/resolve-conflict is complex and time consuming
- **Real-time remote-access methods deliver data-locality, but...**
 - ◆ Scalability – Gateway bottlenecks and file-agents sprawl
 - ◆ Compatibility – Application platform compatibility with proprietary agent
 - ◆ Complexity – Gateways, storage systems, full-mesh networks
 - ◆ Consistency – Latency and consistency are arch-enemies

Strike a Tricky Balance

- Centralized file-serving delivers shared consistency but suffers from poor throughput and response times; also a bottleneck and SPoF
- Sync/replicate workflows alleviate the throughput/latency problem, but not shared consistency; resource-efficiency suffers at large scale
- Caching is efficient and scalable, but not all workloads are cacheable, and shared consistency is a challenging add-on
- Agent-based file services are difficult to scale and manage on every client/server OS involved in the enterprise
- Migrating large amounts of data between on-premises and cloud storage can be expensive, time-consuming and impactful to the availability of data
- Applications running in the cloud may need real-time access to data normally residing on-premises

Hybrid Cloud



- App servers use standard NAS to access data
- Virtual cloud Access Appliance serves data to App servers

- Near infinite capacity can be used for archive file data delivered by Access Appliance

- App servers
- Compute farm
 - Desktops
 - Access Appliance serves data to clients

- NAS and Object
- Multiple tiers and classes of storage
- Access Appliance manages the stored data

Closing thoughts...

- ◆ Even after solving performance/scalability/management challenges with hybrid cloud, of course there is still more to worry about...
 - ◆ Cross-cloud and cross-enterprise Identity / Credential management
 - ◆ Access control and authentication of filesystem requests via supported protocols
 - ◆ Dynamic movement of data and workloads between various cloud service provider(s) and their regions
 - ◆ Protection of applications and data between on-prem and multiple cloud service providers
 - ◆ Retreating back on-premises if things go belly-up
- ◆ But let's walk before we run here...

Thank You!