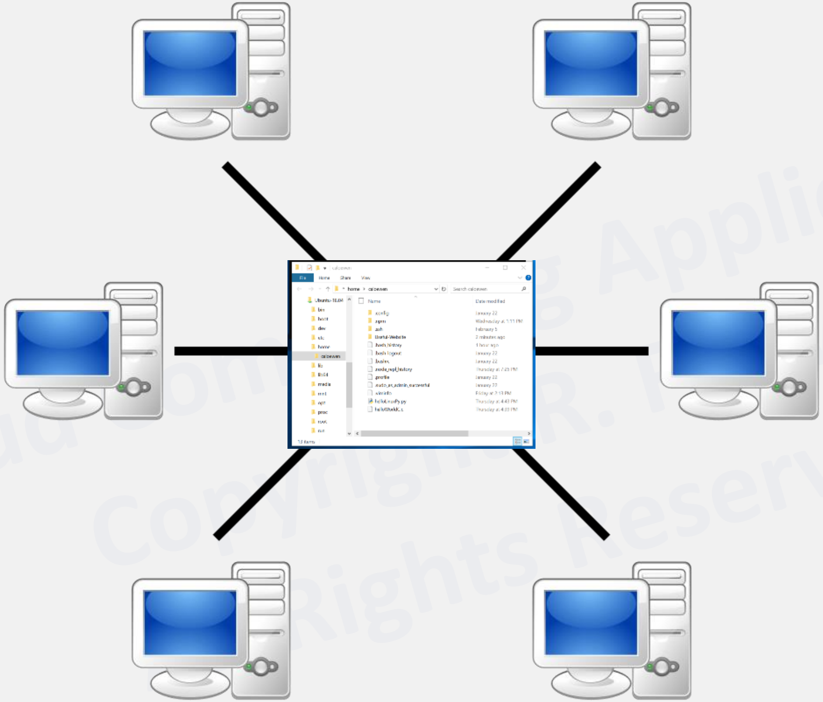
Cloud Storage: Managed File Systems

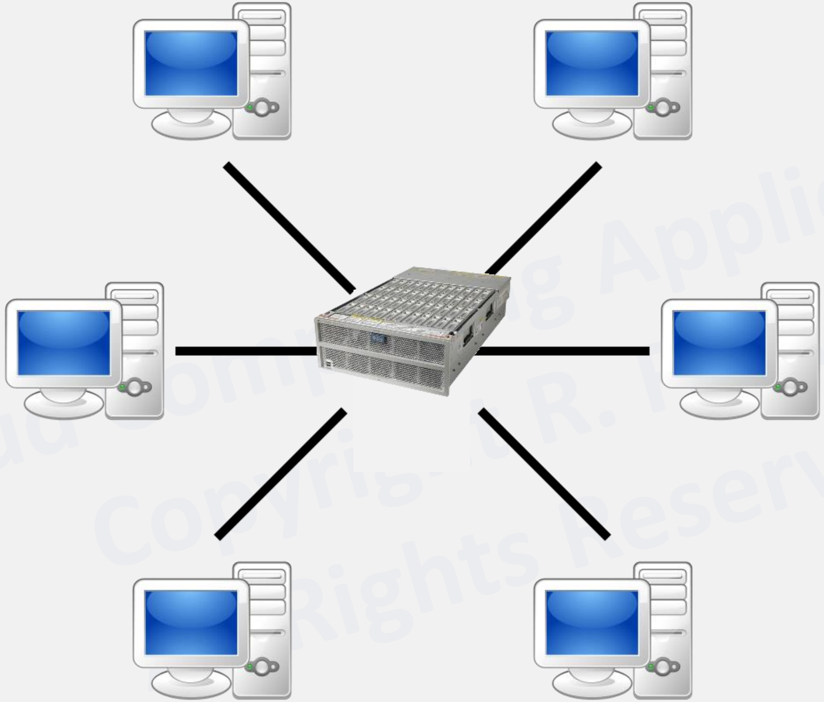
Prof. Reza Farivar

CLOUD COMPUTING APPLICATIONS

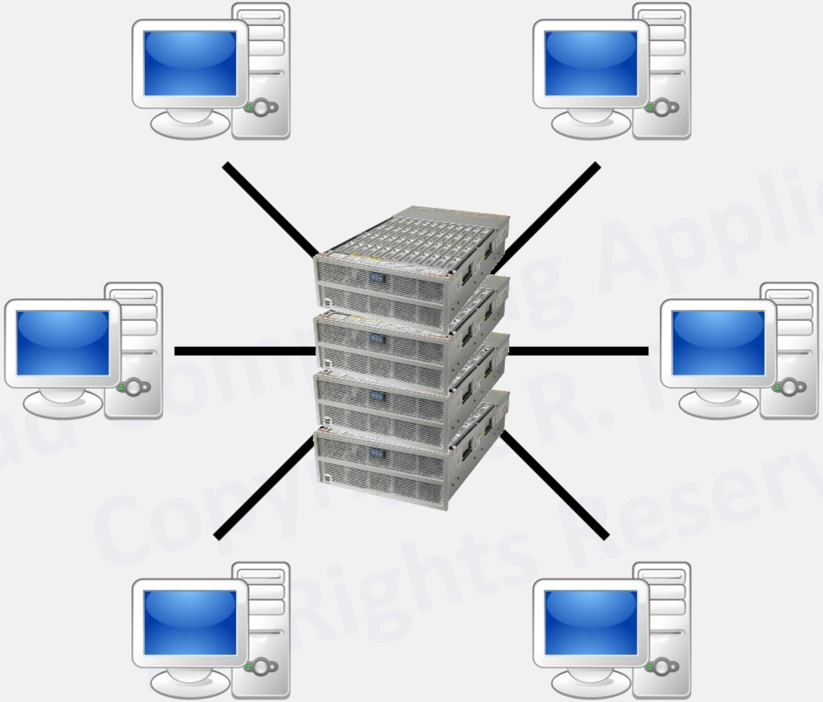
Logical View Networked File System



Network Attached Storage (NAS)



Network Attached Storage (NAS)



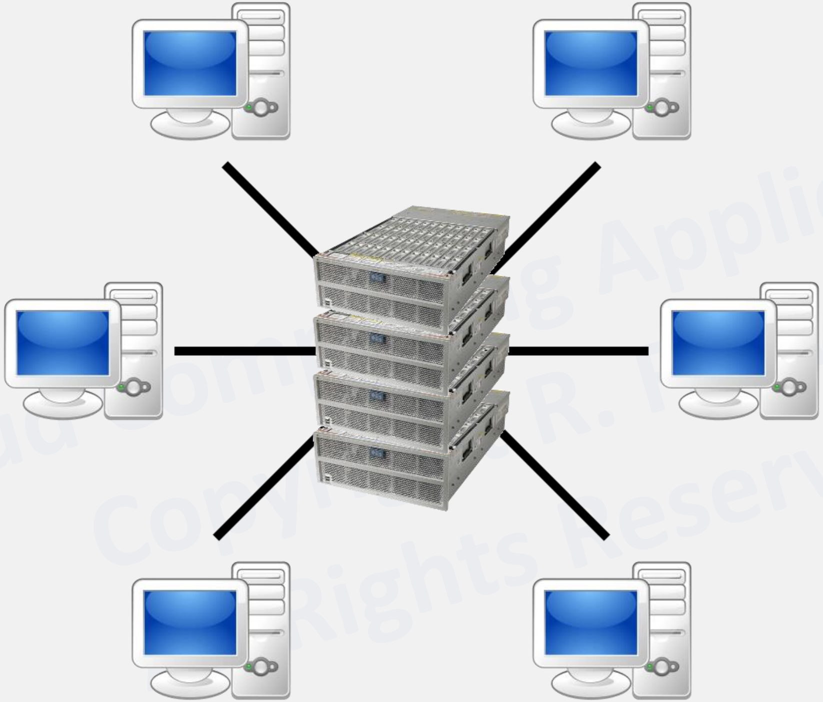
Clustered File Systems

- A clustered file system allows files to be accessed using the same interfaces and semantics as local files – for example, mounting/unmounting, listing directories, read/write at byte boundaries, system's native permission model.
  - Fencing
  - Concurrency
  - Consistency
- Examples
  - NFS
    - Unix
    - V4 , V4.1 (pNFS extension)
  - SMB
    - Windows
  - Lustre
    - HPC
  - Ceph
    - Many OpenStack implementations use Ceph as the storage substrate
  - Gluster
    - Classic file serving, second-tier storage, and deep archiving

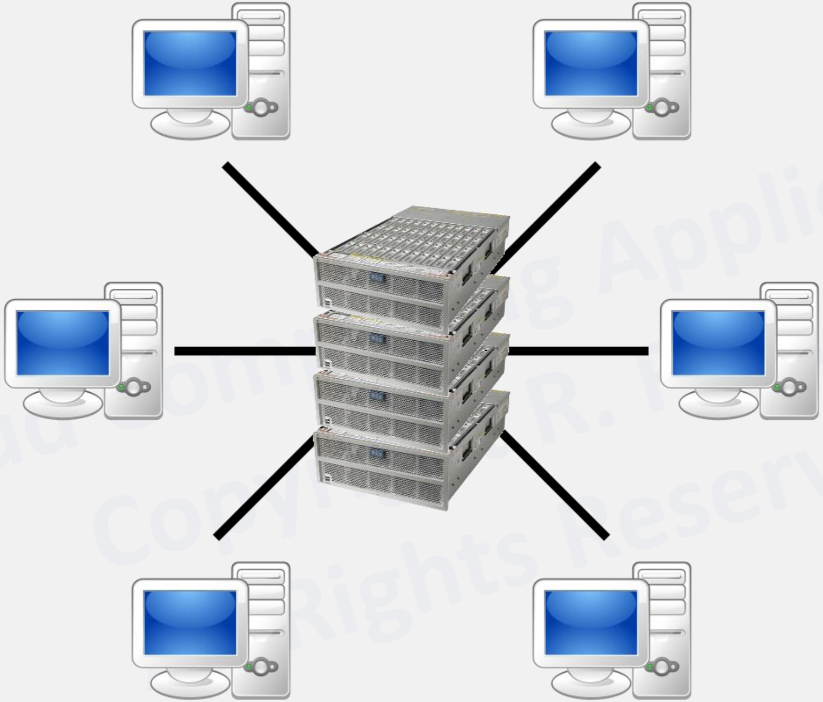
Clustered File System Consistency

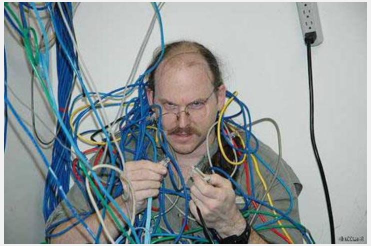
- Ideally, a parallel file system would completely hide the complexity of distributed storage and exhibit a behavior as it is specified by POSIX
  - Fencing
- Relatively easy on one Machine
- Much harder on a cluster of servers
- Maintaining CAP is extremely hard
  - **Consistency**: Every read receives the most recent write or an error
  - **Availability**: Every request receives a (non-error) response, without the guarantee that it contains the most recent write
  - **Partition tolerance**: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

Network Attached Storage (NAS)

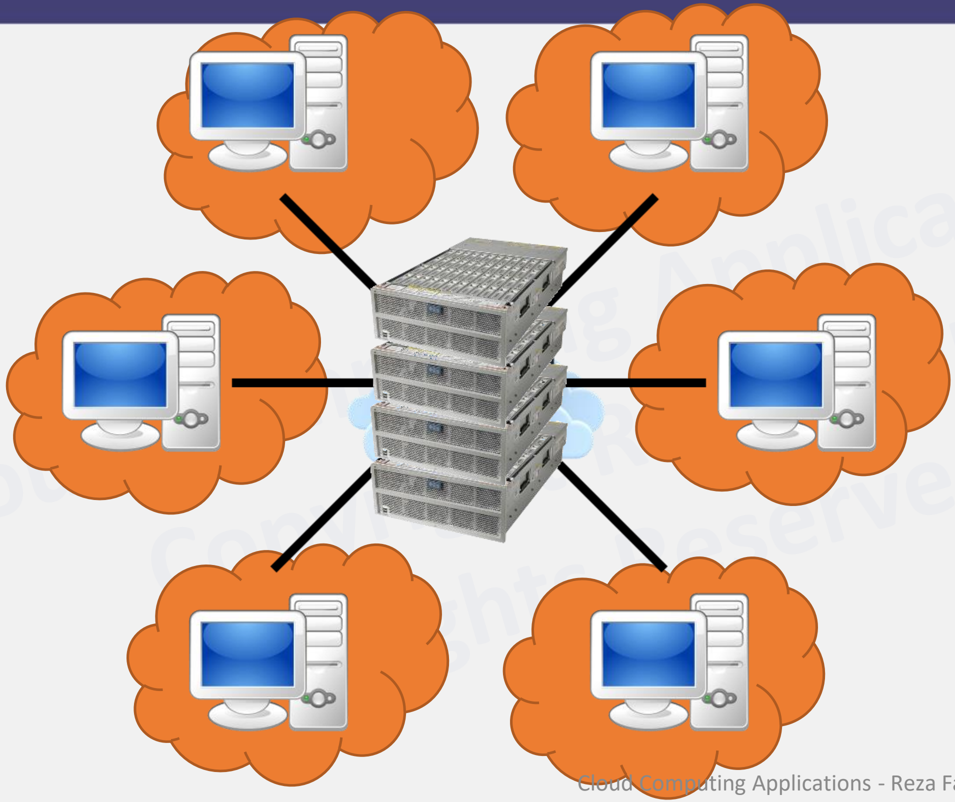


Network Attached Storage (NAS)



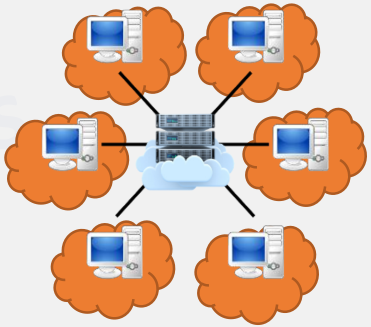


Network Attached Storage (NAS)



Cloud-Based File Systems

- Roll your own Clustered Distributed File System
  - Grab a number of “storage optimized” virtual machines, or metal machines
    - Each with large and fast instance block stores
      - A.k.a. SSDs / HDDs attached to the instance
    - Keep those instances on 24/7
    - Install a distributed file system on them
      - Lustre, MS DFS, NFS, Gluster, Ceph, Hadoop, etc.
- Managed filesystem deployed by the Cloud Provider



Cloud-managed file system

- Amazon
  - AWS FSx for Lustre
  - AWS FSx for Windows File Server
  - AWS EFS
- Azure
  - Azure Files
    - SMB access protocol
    - REST API
  - Azure DataLake Storage
    - Hadoop compatible file system
- Google
  - Cloud Filestore
    - NFS v3
    - Up to 64 TB

Google Filestore, IBM Cloud File Storage, AWS EFS

- Google Filestore

Simple commands to create a Filestore instance with gcloud.

```
gcloud filestore instances create nfs-server \
    --project=[PROJECT_ID] \
    --zone=us-central1-c \
    --tier=STANDARD \
    --file-share=name="vol1",capacity=1TB \
    --network=name="default"
```

Simple commands to install NFS, mount your file share, and set access permissions.

```
sudo apt-get -y update
sudo apt-get -y install nfs-common
sudo mkdir /mnt/test
sudo mount 10.0.0.2:/vol1 /mnt/test
sudo chmod go+rw /mnt/test
```

	Standard	Premium
Max read throughput	100 MB/s (1 TB), 180 MB/s (10+ TB)	1.2 GB/s
Max write throughput	100 MB/s (1 TB), 120 MB/s (10+ TB)	350 MB/s
Max IOPS	5,000	60,000
Max capacity per share	63.9 TB	63.9 TB
Typical customer availability	99.9%	99.9%
Protocol	NFSv3	NFSv3
Price	See Cloud Filestore pricing for more information	See Cloud Filestore pricing for more information

- IBM Cloud File Storage
  - Up to 12 TB
  - Up to 48,000 IOPS
- One beefy machine can handle them
  - AWS i3en.24xlarge: 8 x 7,500 NVMe SSD, 100 Gbps network
    - $60,405 / year per reserved instance
    - Cost of 64 TB standard storage EFS: $0.3*12*1024*64 = $235,929
    - Provisioned IO up to 1 GBps ~= 10 Gbps

Are Managed File Systems Distributed?

- FSx for Windows File Server
  - Max size: 64 TB
  - Can Utilize Microsoft DFS to unify data from many file servers for hundreds of petabytes
    - Shared namespace: Location transparency
    - Replication: Redundancy
- FSx for Lustre
  - Max size: 100 TB
  - Throughput: Read 50-200 MB/s per TB, can burst to 3,000 MB/s per TB
  - E.g. 50.4 TB runs on 22 file servers
    - Hundreds of GB/s of throughput
- AWS EFS
  - Maximum size: “Petabytes”
  - The throughput available to a file system scales as a file system grows
    - 50 MB/s per TB, can burst to 100 MB/s per TB
  - Supports NFS v4.1
- Azure
  - Azure Files
    - 100 TB
  - Data Lake Storage Gen 2
    - HDFS semantics
    - Built on top of Azure Blob Storage
    - Distributed file system
      - Can serve “many exabytes”
      - Throughput measured in gigabits per second (Gbps)

Distributed file Systems Design Goals

- *Access transparency*: clients are unaware that files are distributed and can access them in the same way as local files are accessed.
- *Location transparency*: a consistent namespace exists encompassing local as well as remote files. The name of a file does not give its location.
- *Concurrency transparency*: all clients have the same view of the state of the file system. This means that if one process is modifying a file, any other processes on the same system or remote systems that are accessing the files will see the modifications in a coherent manner.
- *Failure transparency*: the client and client programs should operate correctly after a server failure.
- *Heterogeneity*: file service should be provided across different hardware and operating system platforms.
- *Scalability*: the file system should work well in small environments (1 machine, a dozen machines) and also scale gracefully to bigger ones (hundreds through tens of thousands of systems).
- *Replication transparency*: Clients should be unaware of the file replication performed across multiple servers to support scalability.
- *Migration transparency*: files should be able to move between different servers without the client's knowledge.

Summary

- Clustered File Systems
- File Systems in the Cloud