

Cloud Computing

Pangfeng Liu

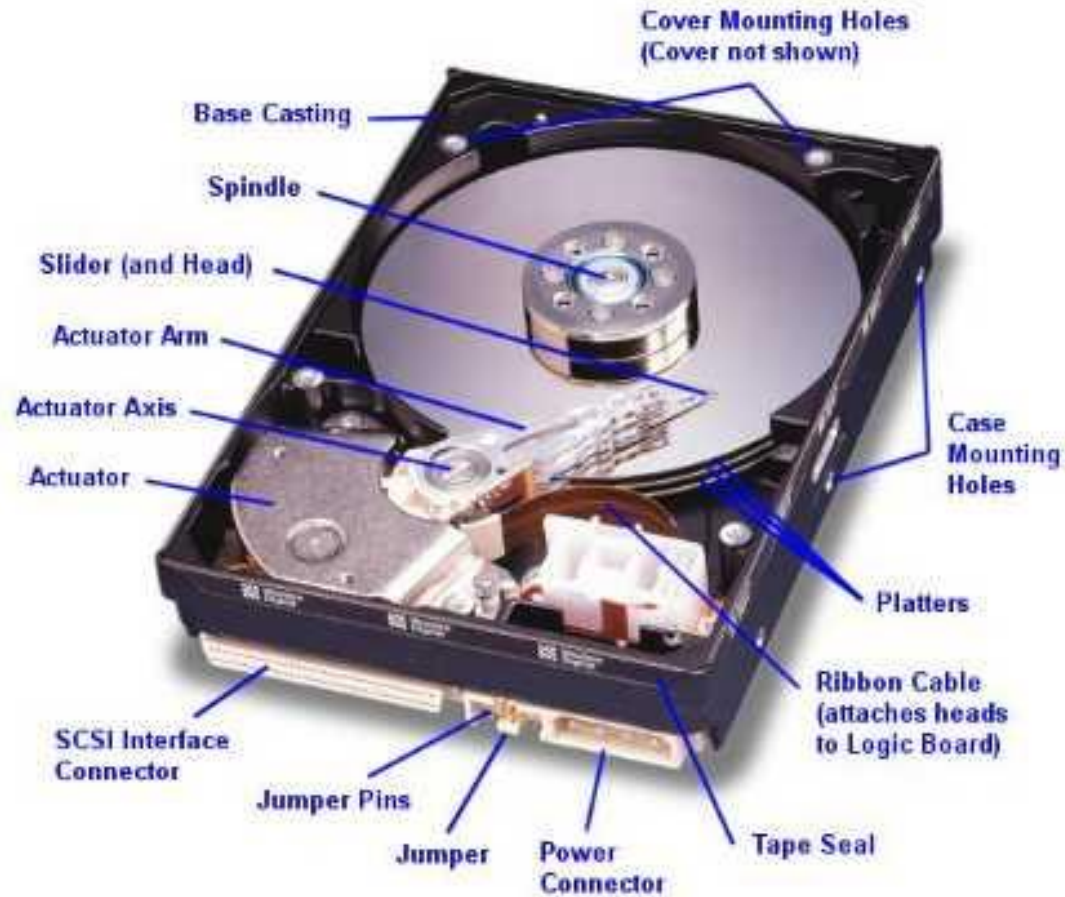
Distributed File System

See your files from anywhere.

File System

- ▶ A file system is a method of storing and organizing computer files and their data.

Disk

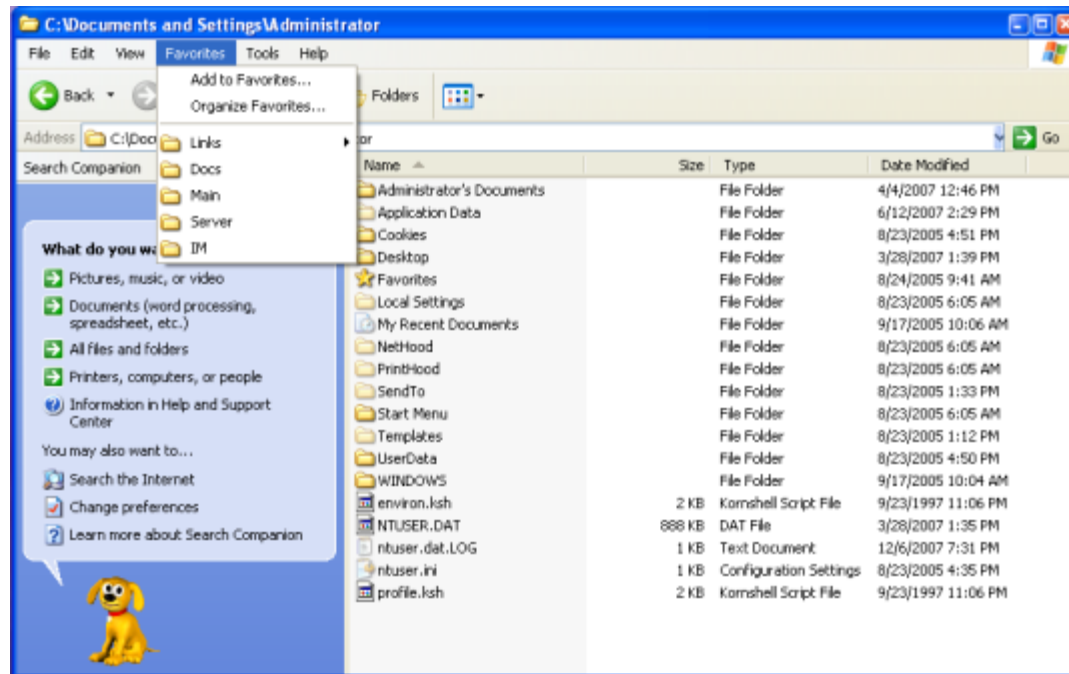


Disk Address

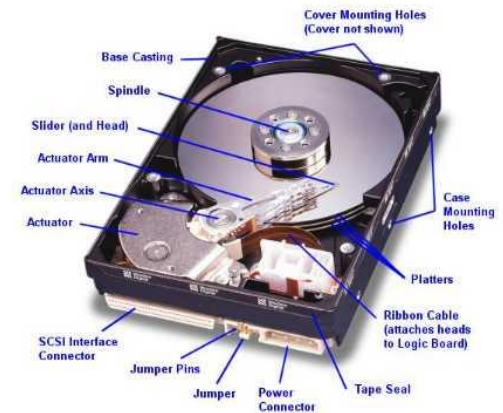
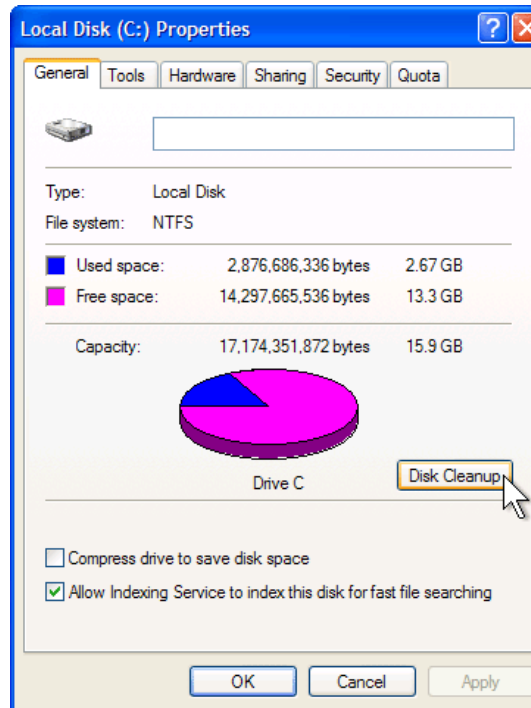
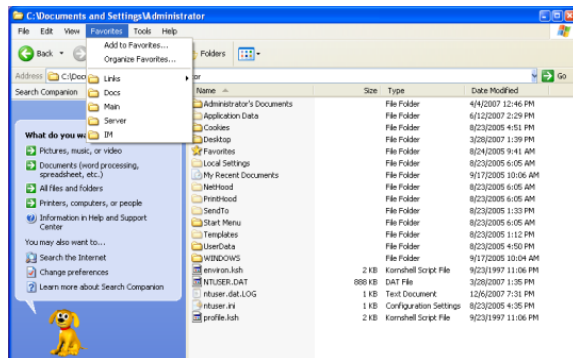
- ▶ How to locate a data?
 - Plate
 - Track
 - Sector
 - All these are combined into a linear address.
- ▶ It would be impossible to locate data if we address disk down to the sector level.

What Users Want

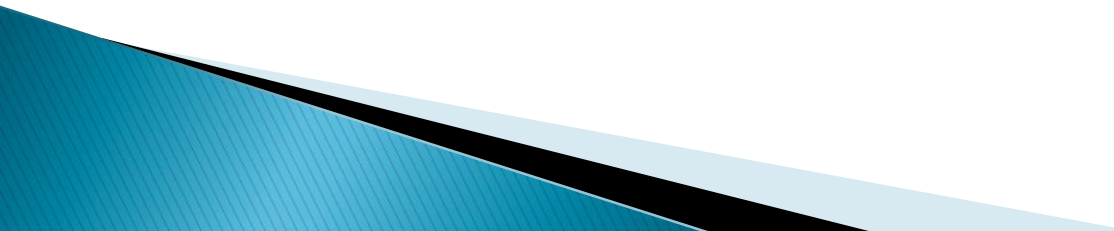
- ▶ No physical address, only logical names.



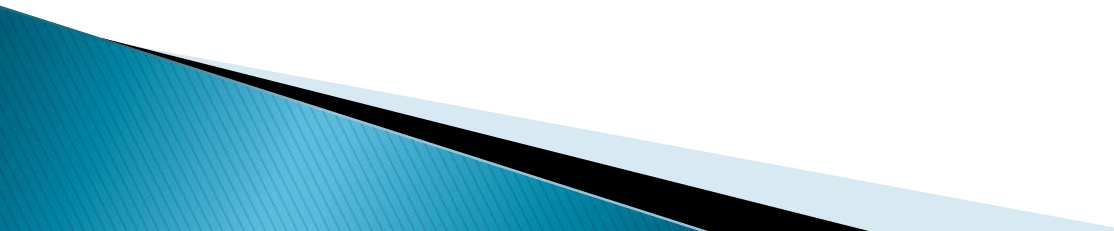
What We Need



File System

- ▶ Organizes data so that users can use logical concepts, e.g., file and directory, to locate the data they need.
 - ▶ Provides protection so that user files will not conflict with each other.
 - ▶ Provides permission checking so that users cannot access data they are not supposed to access.
- 

Metadata

- ▶ Data about data
 - ▶ In the context of files
 - The name of the file
 - The size of the file
 - Where are the pieces of file located
 - Who owns this file
 - Who can access this file
 - When was this file last modified
 - When was this file created
 - The type of file
- 

UNIX ls

```
Terminal — bash — 80x28

bash-3.2$ alias dirA="ls -lia"
bash-3.2$ dirA
total 56
 707856 drwxr-xr-x+ 20 Tech  staff    680 Oct 19 19:35 .
   31486 drwxr-xr-x   5 root   admin   170 Oct 12 15:38 ..
 707876 -rw-----   1 Tech  staff     3 Oct 12 14:50 .CFUserTextEncoding
 708168 -rw-r--r--@   1 Tech  staff 12292 Oct 22 22:29 .DS_Store
1093648 drwx-----   7 Tech  staff   238 Oct 23 00:13 .Trash
1626632 -rw-----   1 Tech  staff   110 Oct 19 19:35 .Xauthority
1094363 -rw-----   1 Tech  staff   655 Oct 22 23:42 .bash_history
1505318 drwx-----@   3 Tech  staff   102 Oct 13 21:26 .cups
1551192 drwxr-xr-x   5 Tech  staff   170 Oct 19 15:54 .fontconfig
1495680 drwx-----   3 Tech  staff   102 Oct 13 16:45 .ssh
 707877 drwx-----+ 10 Tech  staff   340 Oct 23 00:49 Desktop
 707879 drwx-----+ 18 Tech  staff   612 Oct 22 09:30 Documents
 707882 drwx-----+ 13 Tech  staff   442 Oct 22 01:58 Downloads
 707857 drwx-----+ 40 Tech  staff  1360 Oct 22 08:00 Library
 707923 drwx-----+  4 Tech  staff   136 Oct 22 22:30 Movies
 707925 drwx-----+  4 Tech  staff   136 Oct 12 17:20 Music
 707927 drwx-----+  4 Tech  staff   136 Oct 22 13:53 Pictures
1191756 drwxr-xr-x   7 Tech  staff   238 Oct 13 13:21 Programming
 707930 drwxr-xr-x+   5 Tech  staff   170 Oct 12 14:50 Public
 707870 drwxr-xr-x+  21 Tech  staff   714 Oct 20 10:12 Sites

bash-3.2$ dirA
bash: dirA: command not found
bash-3.2$
```

Windows



Windows

```
C:\Temp> dir
Volume in drive C is C
Volume Serial Number is 74F5-893C
```

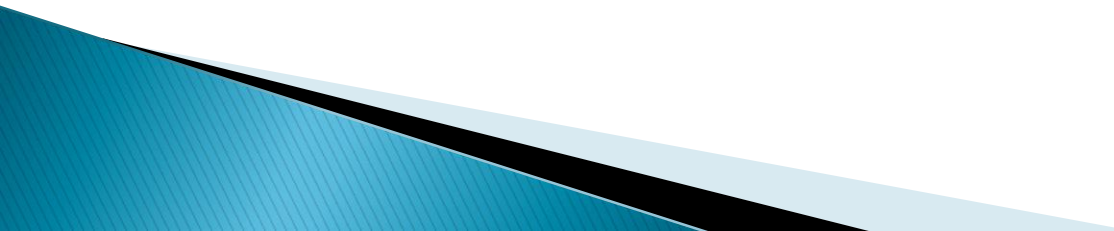
Directory of C:\Temp

```
2009-08-25 11:59 <DIR> .
2009-08-25 11:59 <DIR> ..
2007-03-01 11:37      2,321,600 AdobeUpdater12345.exe
2009-04-03 10:01      27,988 dd_depcheckdotnetfx30.txt
2009-04-03 10:01       764 dd_dotnetfx3error.txt
2009-04-03 10:01      32,572 dd_dotnetfx3install.txt
2009-06-09 11:46      35,145 GenProfile.log
2009-08-05 12:11       155 rs969856.log
2009-04-20 08:37       402 WS129e0b.LOG
2009-04-09 16:34     38,895 office1n11.log
2009-04-03 16:02 <DIR> officePatches
2009-07-14 14:30 <DIR> Offotfix
2009-08-25 10:52      16,384 PerfLib_Perfdata_c30.dat
2009-04-03 10:01       1,744 useventlog.txt
2009-08-25 11:42     50,245,632 WPV2F.tmp
2009-04-20 10:07       1,397 {AC768A86-7AD7-1033-7844-A81200000003}.ini
2009-04-20 10:13       617 {AC768A86-7AD7-1033-7844-A81200000003}.ini
      11 File(s)      52,723,295 bytes
      4 Dir(s)      83,570,208,768 bytes free
```

File System Components

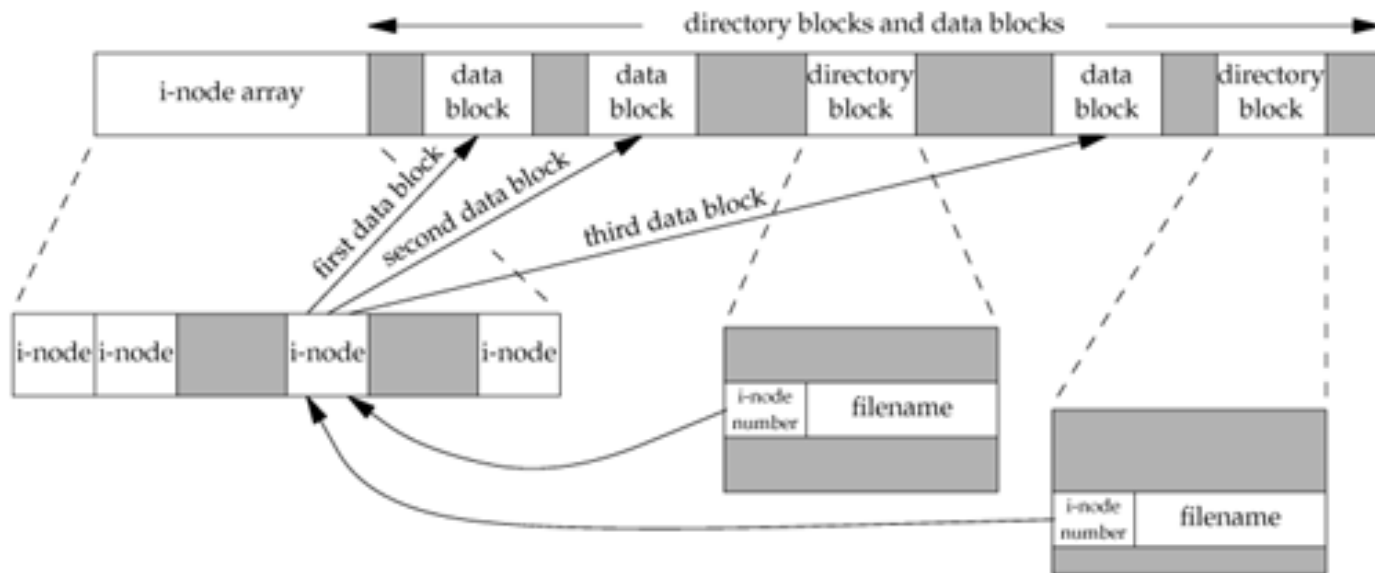
- ▶ Metadata
 - Data about data
- ▶ Data
 - Data itself

An Example: UNIX

- ▶ In UNIX everything is a file.
 - ▶ A regular file is a file that has data.
 - ▶ A directory file is a file containing information of its subdirectory and files.
 - ▶ A FIFO is a file for Inter-Process Communication (IPC)
 - ▶ A socket is a file for networking.
- 

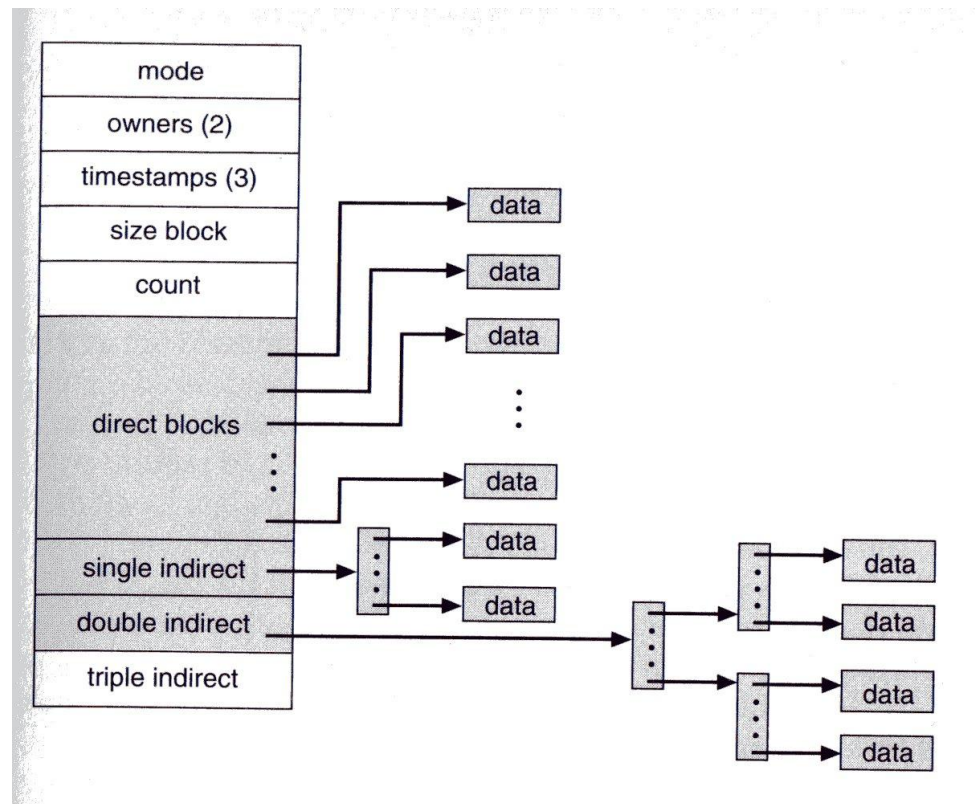
UNIX i-node

- ▶ Used for both data file and directory.
- ▶ File has data, directory has files and other directories.



Data Location

- ▶ i-node contains pointer to data (block)



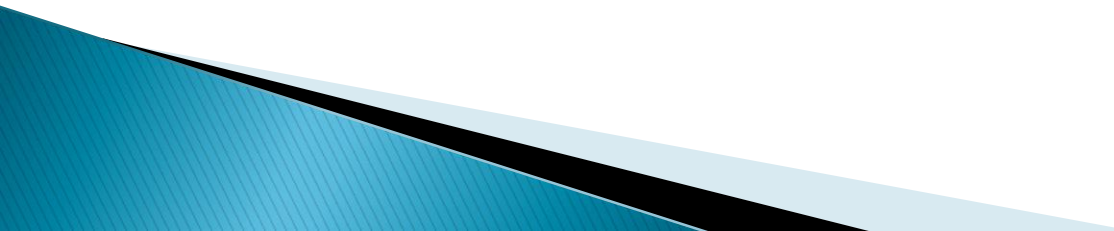
Block Size

- ▶ Data are managed as blocks.
- ▶ If the block size is too large, internal fragmentation will occur.
- ▶ If the block size is too small, the amount of meta data increases, which wastes storage.

File Systems for Disks

- ▶ http://en.wikipedia.org/wiki/List_of_file_systems
- ▶ Linux
 - ext2, ext3, ext4
 - Novell Storage Services
 - JFS, ReiserFS and btrfs.
- ▶ Windows
 - FAT
 - NTFS

File System Classification (Wiki)

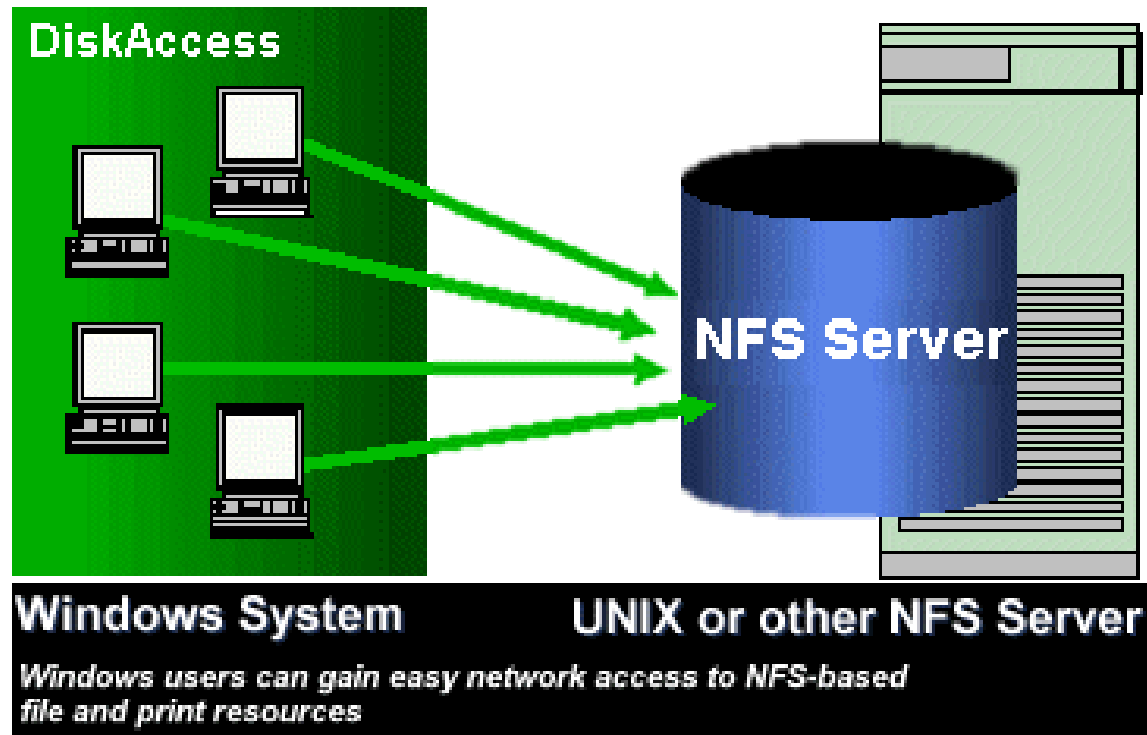
- ▶ Shared disk file systems
 - ▶ Distributed file systems
 - ▶ Distributed fault-tolerant file systems
 - ▶ Distributed parallel file systems
 - ▶ Distributed parallel fault-tolerant file systems
- 

Distributed File Systems

- ▶ In wiki, it means the clients are distributed.
 - Network File System (NFS)
 - A NFS server can serve a large number of distributed clients. Like we have a NFS server in 217 and all Linux boxes in this department can use files in the NFS server.
- ▶ Usually it means the data itself is distributed across participating servers.

NFS

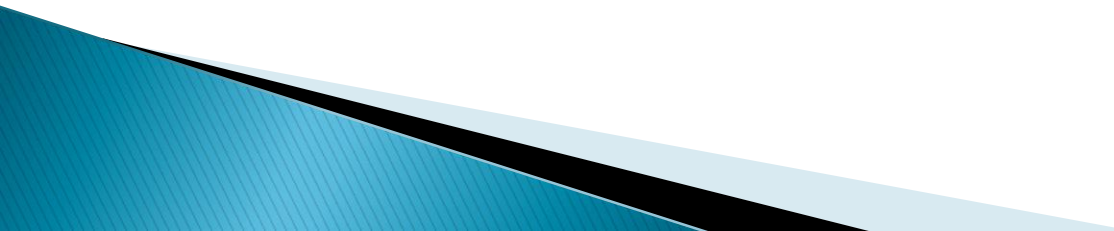
- ▶ All clients go to NFS server for data.



NFS

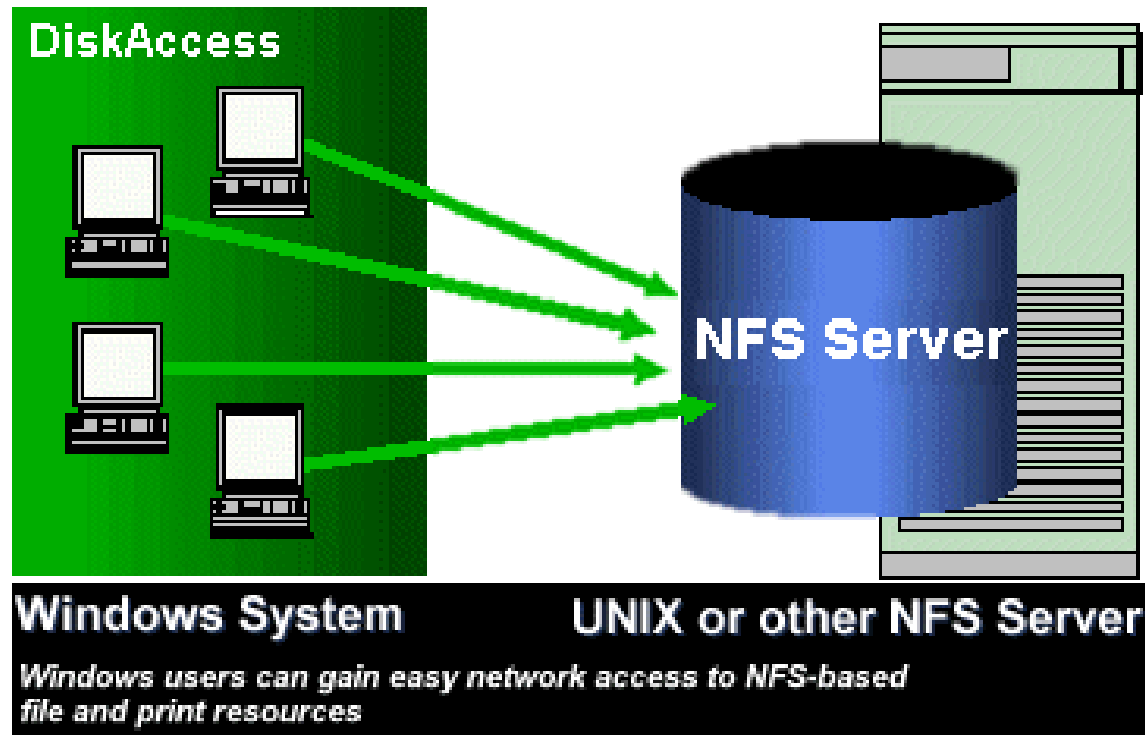
- ▶ Network File System (NFS) is a network file system protocol originally developed by Sun Microsystems in 1984, allowing a user on a client computer to access files over a network in a manner similar to how local storage is accessed.

Home Directory

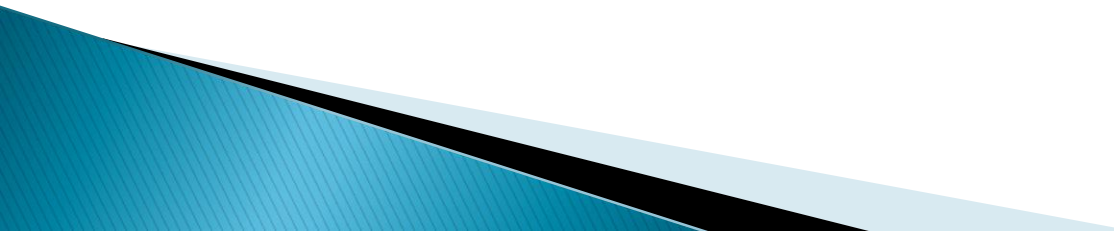
- ▶ We have many Linux machines in this department.
 - ▶ How to make sure that when a user logs in, he will see his home directory?
 - ▶ The solution is to have a NFS server hosts the disk that stores home directory for every user, then every Linux machine mount this disk as a part of his file system.
- 

NFS

- ▶ All clients go to NFS server for home directory.



New NFS

- ▶ NFSv4.1 adds the Parallel NFS pNFS capability, which enables data access parallelism.
 - ▶ The NFSv4.1 protocol defines a method of separating the filesystem meta-data from the location of the file data; it goes beyond the simple name/data separation by striping the data amongst a set of data servers.
 - ▶ This is different from the traditional NFS server which holds the names of files and their data under the single umbrella of the server.
- 

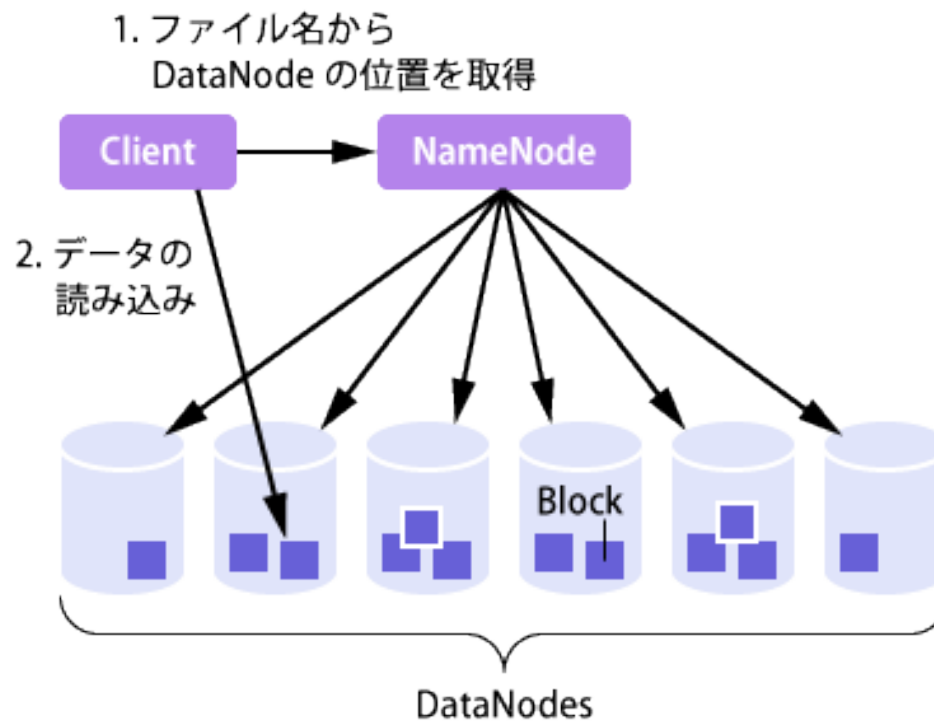
Limitations

- ▶ NFS is not scalable
 - We cannot have a large number of clients accessing data from the same NFS server.
- ▶ NFS is not fault-tolerant
 - NFS does not replicate data.

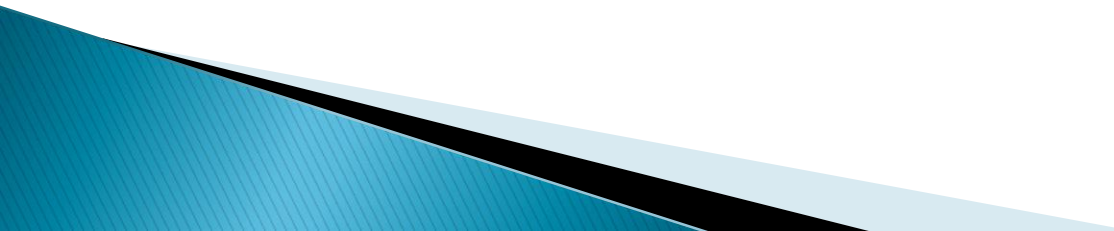
Distributed File Systems

- ▶ Data is distributed across participating data nodes.
- ▶ Clients first receive the data locations from the metadata server, then go to the datanode for data.

An Illustration



Characteristics

- ▶ Transparency
 - The clients use a distributed file system just as a single huge disk.
 - ▶ High performance
 - Data can be retrieved from multiple data nodes simultaneously.
 - Much like RAID-0.
 - ▶ Concurrent updates
 - Independent files can be updated concurrently.
- 

Fault-tolerant

- ▶ Hardware failure is inevitable.
 - When you have thousands of machines, it is evitable that some of them will fail.
- ▶ Data is replicated in different data nodes so that when one data node crashes, we can still retrieve the replica from other data nodes.

High Availability (HA)

- ▶ High availability is a system design approach and associated service implementation that ensures a prearranged level of operational performance will be met during a contractual measurement period.

糗！桃園機場出境行李系統當機 至少5航班因此延誤

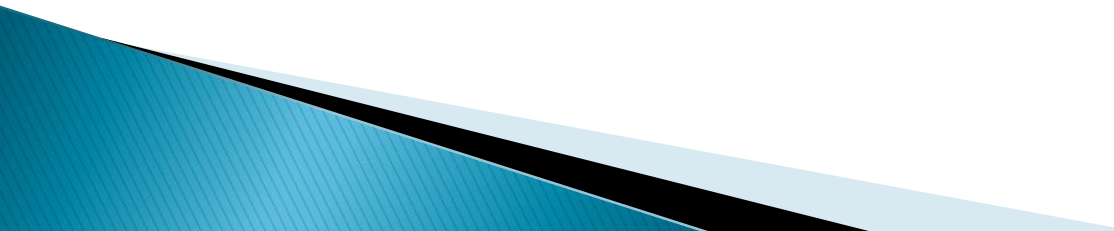
- ▶ 桃園機場二航廈出境的BHS輸送帶分檢系統，14日中午12時20分突然當機，造成出境的行李無法自動分檢到班機上，機場資訊人員查出是電腦硬碟當機，導致資料庫無法與主機連線，資訊人員雖趕工修復，並採取人工分檢行李，但已影響至少5個航班的起降。
- ▶ 行李分檢電腦系統大當機，導致資料硬碟無法運作，旅客行李輸送帶因而停擺，航空站緊急派遣150人，以人工方式協助行李分類輸送。桃園航空站發言人趙紹廉表示，電腦系統當機，造成資料硬碟無法運作後，使得多條輸送行李的輸送帶有過熱現象，才造成停擺，但在輸送帶冷卻後，已在14日下午3時起陸續恢復運作。
- ▶ 趙紹廉指出，機場資訊人員除進行後續維修，未來也會加強硬碟備份等工作。由於到了晚間仍無法解決問題，資訊人員在趕工修復之餘，已正在調度硬碟。而這起事件已造成華航飛香港、全日空飛成田等5個航班的起降受到延誤，還出現行李堆積如山的情況，旅客們忍不住說「真誇張」！

Availability Measurement

- ▶ Availability is usually expressed as a percentage of uptime in a given year.

Availability %	Downtime per year	Downtime per month*	Downtime per week
90% ("one nine")	36.5 days	72 hours	16.8 hours
95%	18.25 days	36 hours	8.4 hours
98%	7.30 days	14.4 hours	3.36 hours
99% ("two nines")	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes
99.9% ("three nines")	8.76 hours	43.2 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99% ("four nines")	52.56 minutes	4.32 minutes	1.01 minutes
99.999% ("five nines")	5.26 minutes	25.9 seconds	6.05 seconds
99.9999% ("six nines")	31.5 seconds	2.59 seconds	0.605 seconds

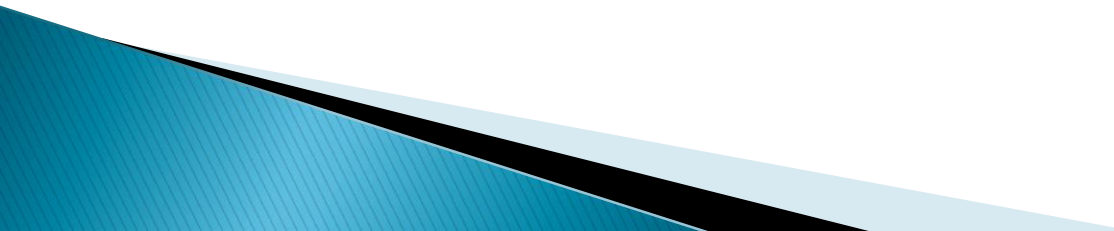
Techniques

- ▶ The most effective way to improve availability is to provide redundancy.
 - ▶ The data should be duplicated, or the data can be deduced from the remaining data.
 - RAID-1 (mirroring without parity or striping)
 - RAID-5 (block-level striping with distributed parity)
- 

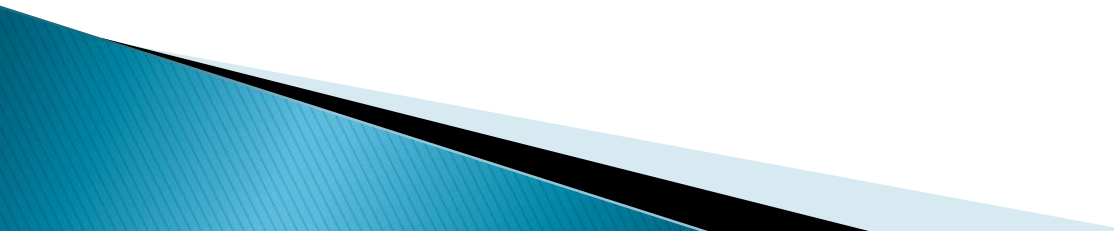
Parallel File System

- ▶ Distributed parallel file systems stripe data over multiple servers for high performance.
- ▶ They are normally used in high-performance computing (HPC).
 - RAID-0 like.

RAID

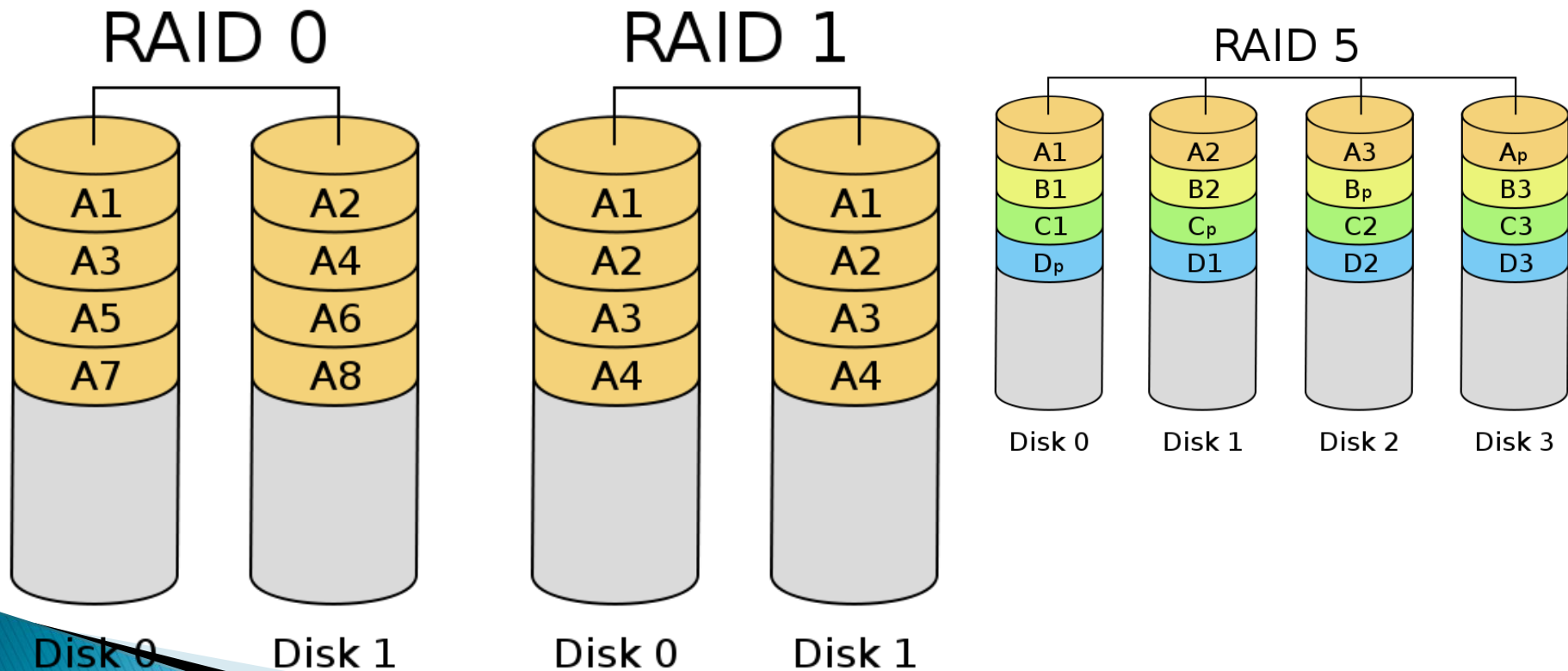
- ▶ Redundant Array of Independent Disks
 - ▶ A technology that provides increased storage reliability through redundancy, combining multiple relatively low-cost, less-reliable disk drives components into a logical unit where all drives in the array are interdependent.
- 

Levels

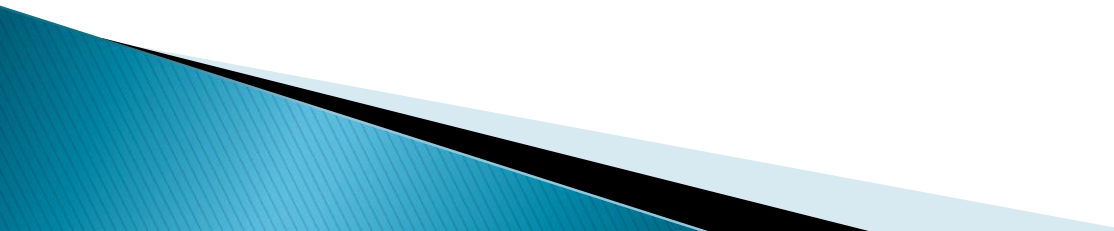
- ▶ RAID0 (block-level striping without parity or mirroring) provides improved performance and additional storage but no redundancy or fault tolerance.
 - ▶ RAID1 (mirroring without parity or striping), data is written identically to multiple disks (a "mirrored set").
 - ▶ RAID5 (block-level striping with distributed parity) distributes parity along with the data and requires all drives but one to be present to operate.
- 

Illustration

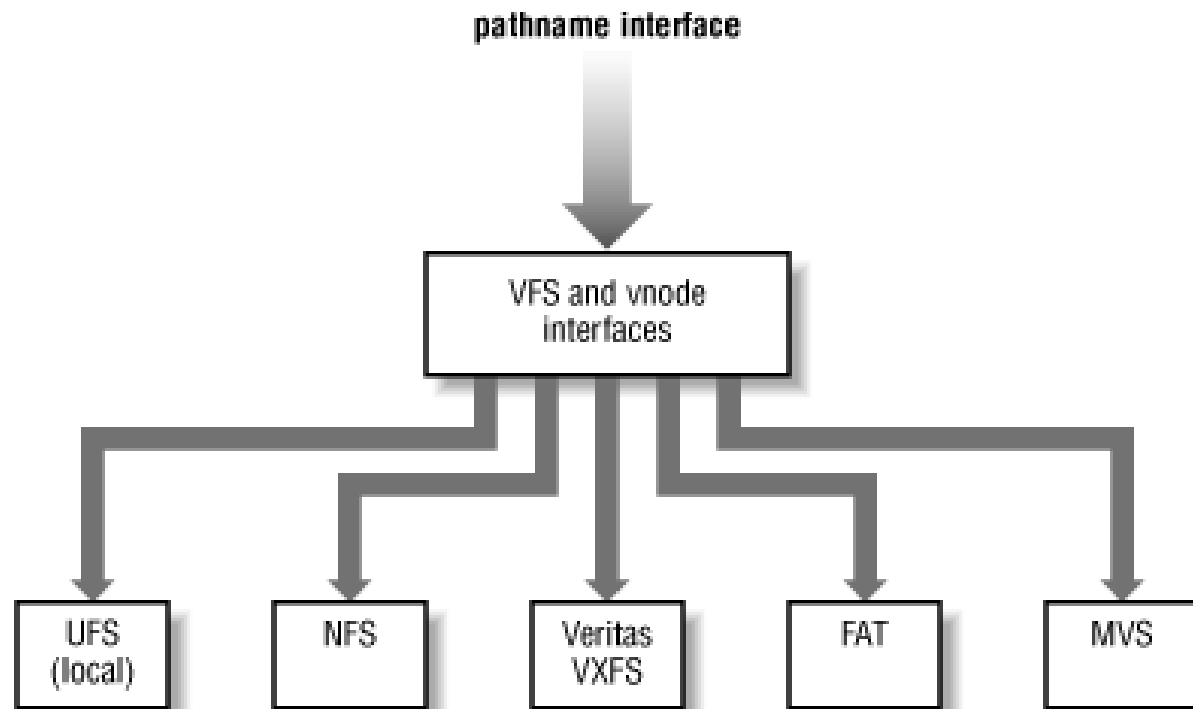
- ▶ <http://en.wikipedia.org/wiki/RAID>



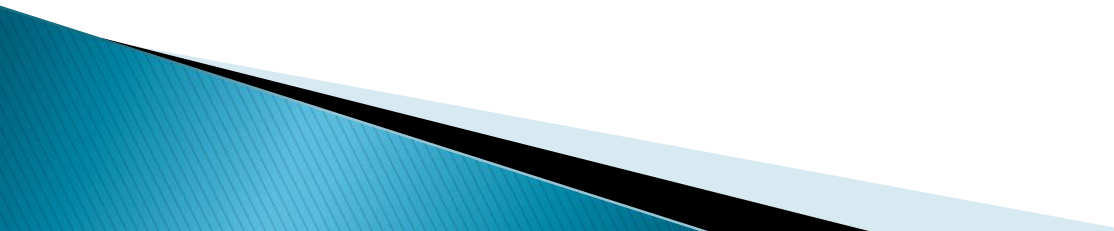
Virtual File System

- ▶ A virtual file system is an abstraction layer on top of a more concrete file system.
 - ▶ The purpose of a VFS is to allow client applications to access different types of concrete file systems in a uniform way.
 - ▶ A VFS can, for example, be used to access local and network storage devices transparently without the client application noticing the difference.
- 

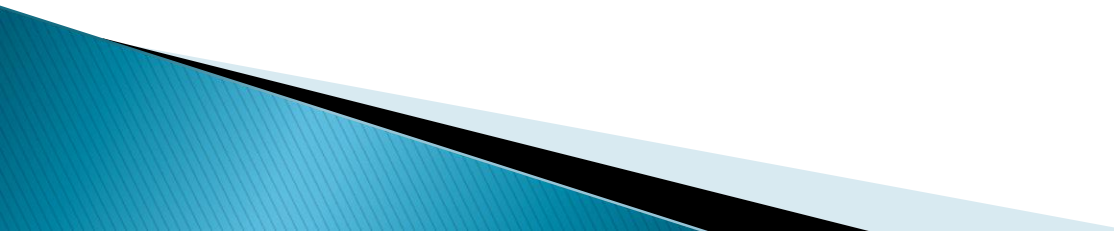
An Illustration



FUSE

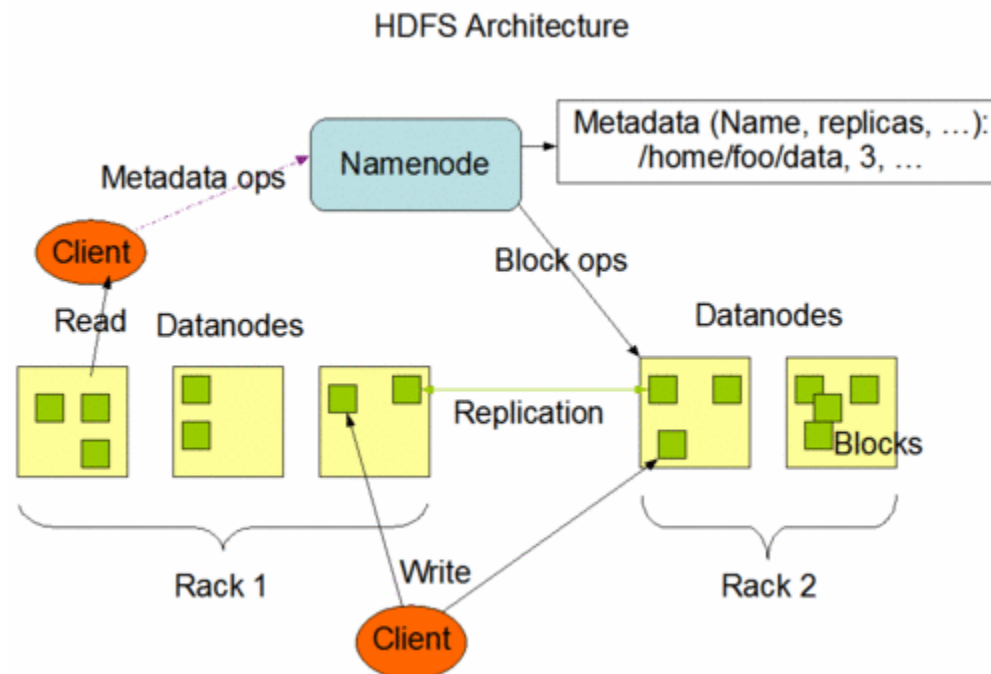
- ▶ Filesystem in Userspace (FUSE) is a loadable kernel module for Unix-like computer operating systems that lets non-privileged users create their own file systems without editing kernel code.
 - ▶ This is achieved by running file system code in user space while the FUSE module provides only a "bridge" to the actual kernel interfaces.
- 

What We Want

- ▶ Every VM can see every file, no matter where it is located.
 - We need a global logical view of storage.
 - ▶ When some data nodes failed, the system is still functional.
 - We need replication of data.
 - ▶ Can run on a thousand nodes.
 - We need scalability.
 - ▶ Can store tens of petabyte of data.
 - We need distributed file systems.
 - ▶ Can run on existing PC.
 - We need virtual file system in on of existing file systems.
 - ▶ Can adapt to changes quickly.
 - We need autonomous management.
- 

Hadoop File System

- ▶ The HDFS file system stores large files across multiple machines.



Google File System

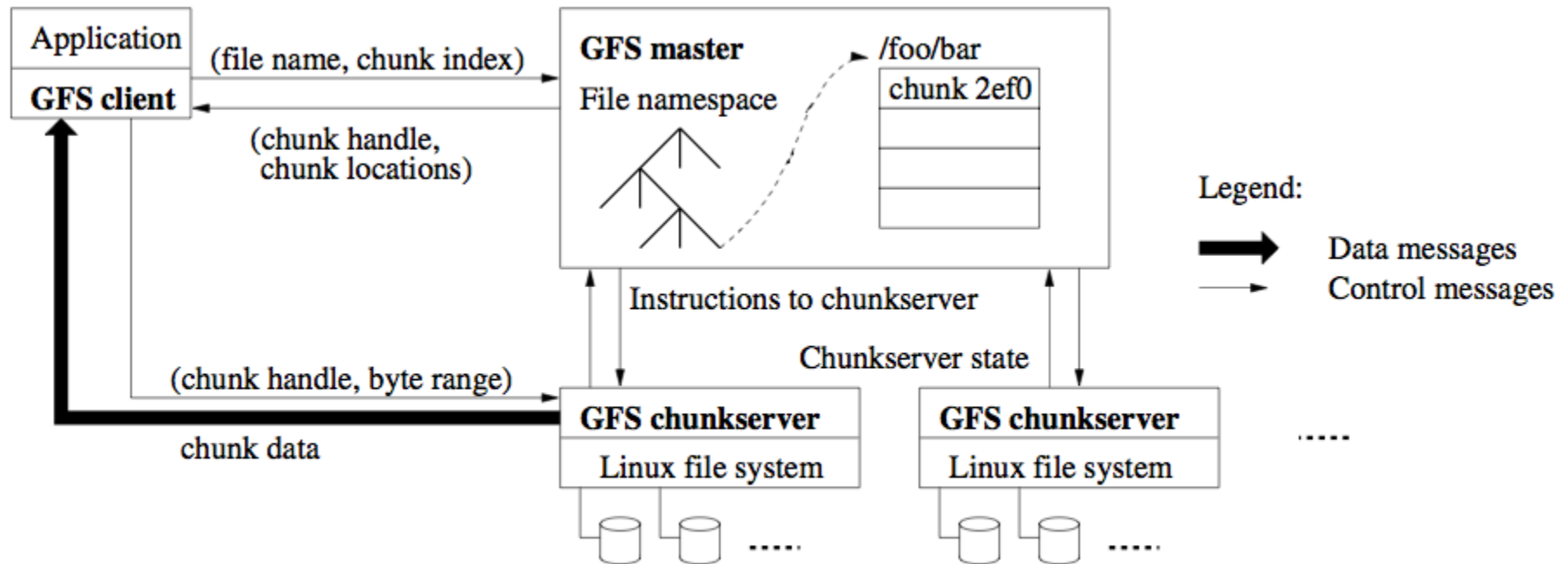


Figure 1: GFS Architecture

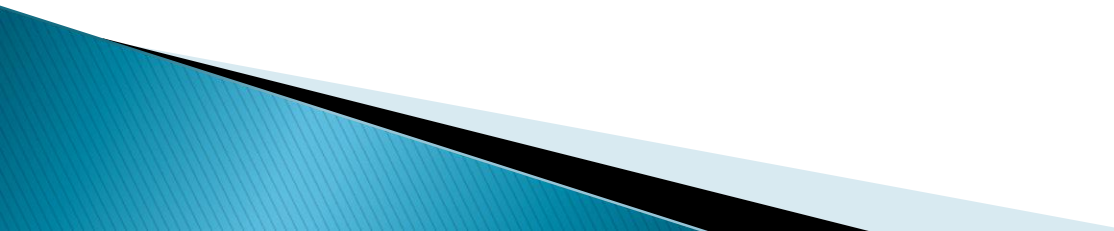
Architecture

- ▶ The system has one namenode and multiple datanodes.
 - In Google's term, master node and chunk servers.
- ▶ File is divided into blocks and stored in datanodes.
 - Namenode determines the locations of blocks. Data nodes store the blocks.

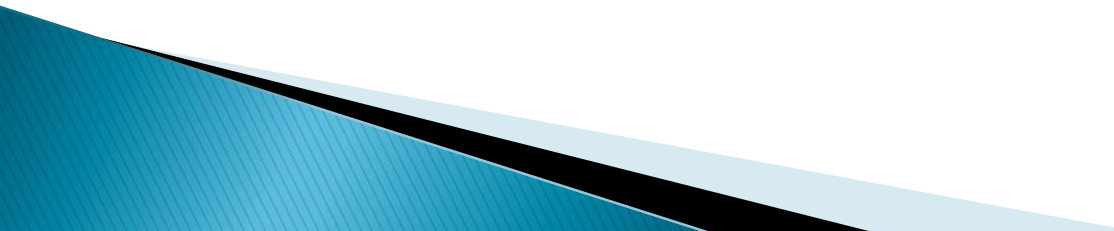
Namenode

- ▶ Only one Namenode
 - What will happen if you have multiple namenodes?
 - Single point of failure?
- ▶ Namenode Keeps track of metadata, including where to locate data blocks.
- ▶ Namenode is aware of tracks.
 - Replication should be done in different racks.

Secondary Namenode

- ▶ Secondary Namenode regularly connects with the Primary Namenode and builds snapshots of the Primary Namenode's directory information, which is then saved to local/remote directories.
 - ▶ Checkpointed images can be used to restart a failed Primary Namenode without having to replay the entire journal of filesystem actions, the edit log to create an up-to-date directory structure.
- 

Google's Point of View

- ▶ Having only one master is a single point of failure and scalability bottleneck.
 - ▶ Google's solution is shadow masters and minimized master involvement.
 - Do not move data through master
 - Cache metadata at clients
 - Large chunk size
 - Master delegates authority to primary replicas in data update.
 - ▶ Simple and good enough
- 

Datanodes

- ▶ Store data for client to access.
- ▶ Datanodes can talk to each other to
 - Rebalance data
 - Move copies of data around
 - Keep the replication of data high.

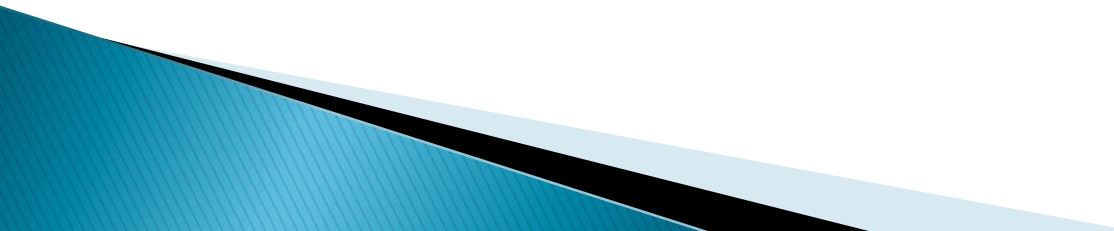
Clients

- ▶ Clients interact with namenode for metadata.
- ▶ Clients interact with datanode for data.
- ▶ For example, clients interact with namenode for locations of data, then interact with datanode directly for data.

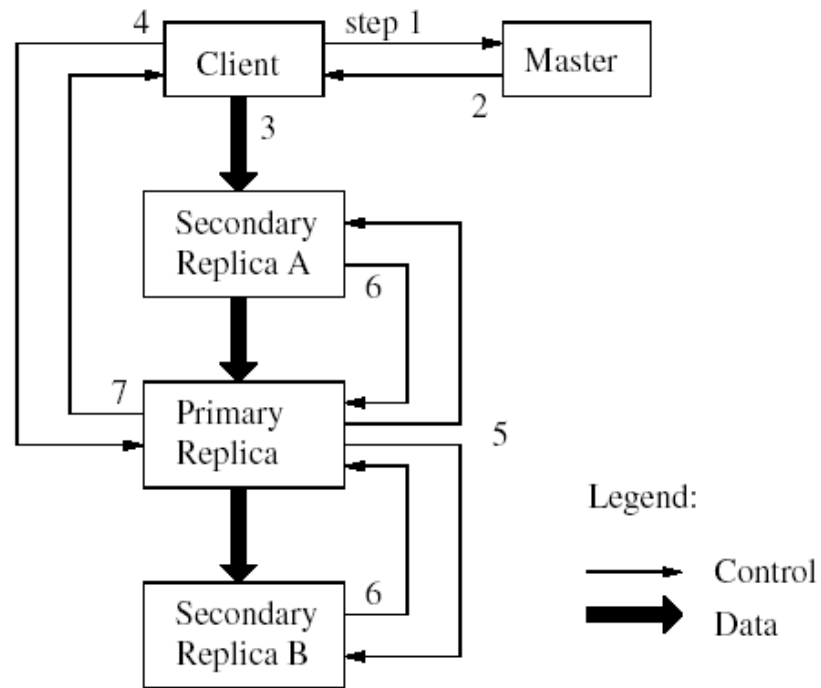
Replication

- ▶ HDFS achieves reliability by replicating the data across multiple hosts, and hence does not require RAID storage on hosts.
- ▶ With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack.
 - Why?

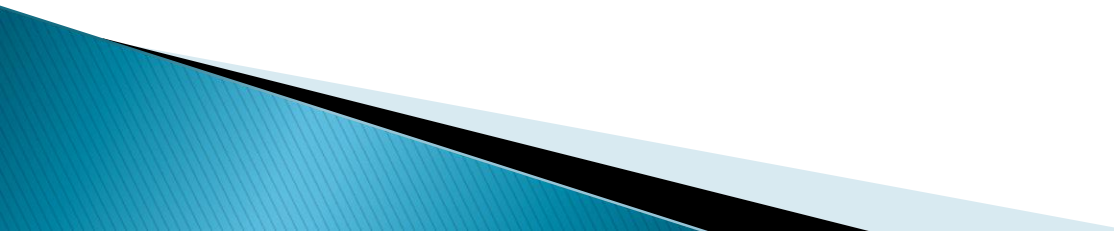
Update

- ▶ To minimize master involvement
 - The master picks one replica as the primary and gives it a “lease” for update.
 - The primary defines a serial order of updates.
 - All replicas follow this order.
 - ▶ Data flow decoupled from control flow.
- 

An Illustration



Access

- ▶ HDFS is built from a cluster of data nodes, each of which serves up blocks of data over the network using a block protocol specific to HDFS.
 - ▶ HDFS also serves data over HTTP, allowing access to all content from a web browser or other client.
 - ▶ HDFS will choose the closest replica.
- 

Everything in Memory

- ▶ HDFS follows the design of Google File System that all metadata is stored in memory.
 - Quick access to all metadata.
 - No need to synchronize metadata if only one node has it.
- ▶ The block size must be large.
 - Why?

Limitation

- ▶ Cannot be directly mounted by an existing operating system.
 - Getting data into and out of the HDFS file system can be inconvenient.
- ▶ A file system in user space (FUSE) virtual file system has been developed to address this problem, at least for Linux and some other Unix systems.

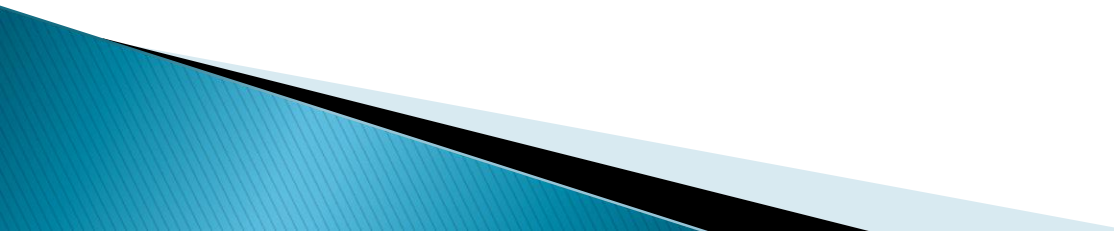
More Limitations

- ▶ No quota
- ▶ No links of any kind
- ▶ One writer at any time

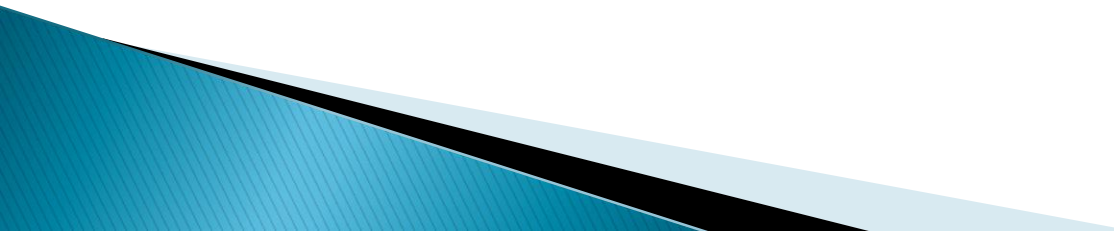
Hadoop Users

- ▶ [Yahoo!](#)
 - ▶ [A9.com](#)
 - ▶ [AOL](#)
 - ▶ [Booz Allen Hamilton](#)
 - ▶ [EHarmony](#)
 - ▶ [eBay](#)
 - ▶ [Facebook](#)
 - ▶ [Fox Interactive Media](#)
 - ▶ [Freebase](#)
 - ▶ [IBM](#)
 - ▶ [ImageShack](#)
 - ▶ [ISI](#)
 - ▶ [Joost](#)
 - ▶ [Last.fm](#)
 - ▶ [LinkedIn](#)
 - ▶ [Meebo](#)
 - ▶ [Metaweb](#)
 - ▶ [The New York Times](#)
 - ▶ [Ning](#)
 - ▶ [Powerset](#) (now part of Microsoft)
 - ▶ [Rackspace](#)
 - ▶ [StumbleUpon](#)
 - ▶ [Twitter](#)
 - ▶ [Veoh](#)
 - ▶ [Zoosk](#)
- 

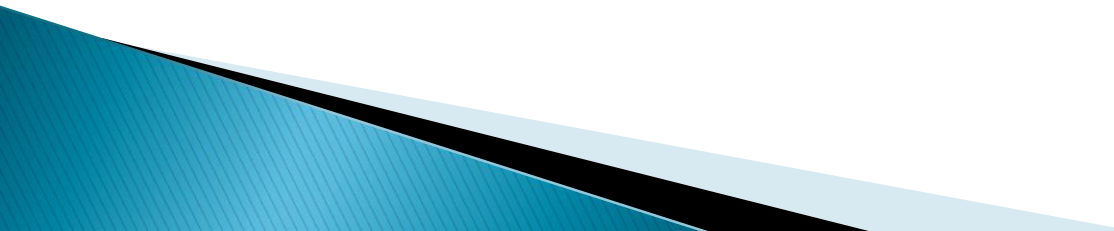
EC2 Features

- ▶ **Amazon Elastic Block Store**
 - ▶ Multiple Locations
 - ▶ Elastic IP Addresses
 - ▶ Amazon Virtual Private Cloud
 - ▶ Amazon CloudWatch
 - ▶ Auto Scaling
 - ▶ Elastic Load Balancing
 - ▶ High Performance Computing (HPC) Clusters
 - ▶ VM Import
- 

EBS

- ▶ Amazon Elastic Block Store (EBS) offers persistent storage for Amazon EC2 instances.
 - ▶ Amazon EBS volumes provide off-instance storage that persists independently from the life of an instance.
 - ▶ Amazon EBS volumes are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instance's boot partition or attached to a running Amazon EC2 instance as a standard block device.
- 

Duplication

- ▶ Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores, as Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone).
 - ▶ For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon S3, and automatically replicated across multiple Availability Zones.
- 

Cost

Balancer inside of the Amazon EC2 network, you'll pay Regional Data Transfer rates even if the instances are in the same Availability Zone. For data transfer within the same Availability Zone, you can easily avoid this charge (and get better network performance) by using your private IP whenever possible.

Amazon Elastic Block Store

Region:

Amazon EBS Volumes

- \$0.10 per GB-month of provisioned storage
- \$0.10 per 1 million I/O requests

Amazon EBS Snapshots to Amazon S3

- \$0.14 per GB-month of data stored

Elastic IP Addresses

Region:

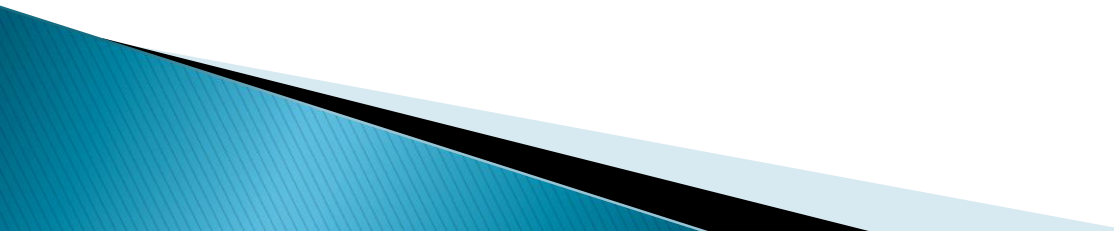
No cost for Elastic IP addresses while in use

- \$0.01 per non-attached Elastic IP address per complete hour
- \$0.00 per Elastic IP address remap – first 100 remaps / month
- \$0.10 per Elastic IP address remap – additional remap / month over 100

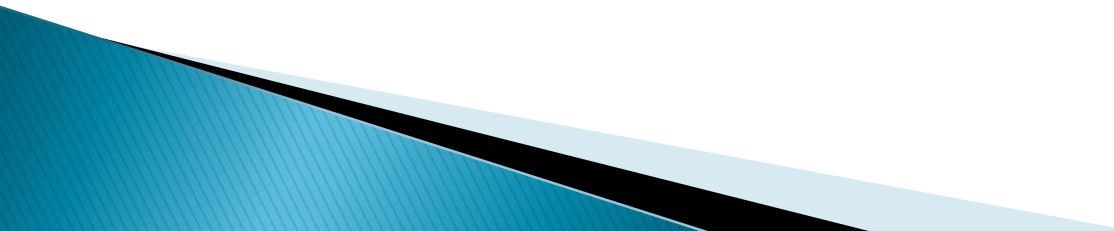
Amazon CloudWatch

Region:

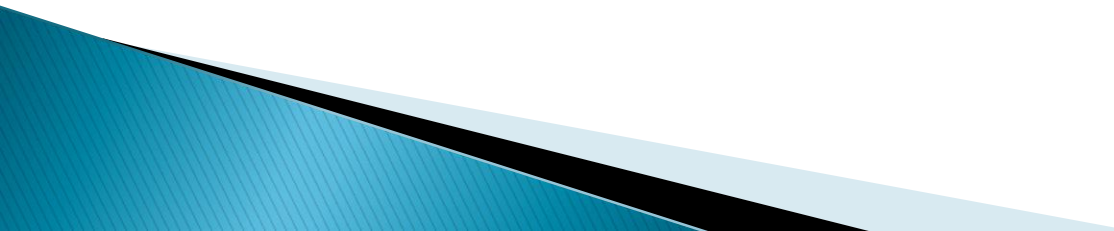
Amazon Simple Storage Service

- ▶ Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers.
 - ▶ Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network of web sites.
- 

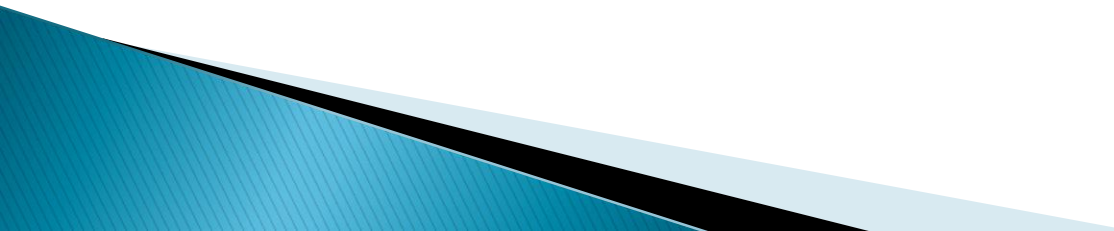
Functionality

- ▶ Write, read, and delete objects containing from 1 byte to 5 terabytes of data each. The number of objects you can store is unlimited.
 - ▶ Each object is stored in a bucket and retrieved via a unique, developer–assigned key.
- 

Regions

- ▶ A bucket can be stored in one of several Regions.
 - ▶ Amazon S3 is currently available in the US Standard, EU (Ireland), US West (Northern California), Asia Pacific (Singapore), Asia Pacific (Tokyo) and GovCloud (US) Regions.
 - ▶ The US Standard Region automatically routes requests to facilities in Northern Virginia or the Pacific Northwest using network maps.
- 

More Functionalities

- ▶ Authentication mechanisms are provided.
 - ▶ Options for secure data upload/download and encryption of data are provided.
 - ▶ Uses standard interfaces designed to work with any Internet-development toolkit.
 - ▶ Reliability backed.
- 

Pricing

Storage Pricing

Region: <input type="text" value="US Standard"/>		
	Standard Storage	Reduced Redundancy Storage
First 1 TB / month	\$0.140 per GB	\$0.093 per GB
Next 49 TB / month	\$0.125 per GB	\$0.083 per GB
Next 450 TB / month	\$0.110 per GB	\$0.073 per GB
Next 500 TB / month	\$0.095 per GB	\$0.063 per GB
Next 4000 TB / month	\$0.080 per GB	\$0.053 per GB
Over 5000 TB / month	\$0.055 per GB	\$0.037 per GB

Request Pricing

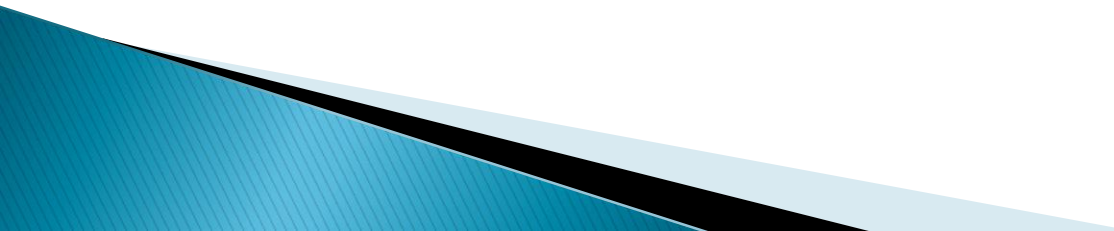
Region: <input type="text" value="US Standard"/>	
Pricing	
PUT, COPY, POST, or LIST Requests	\$0.01 per 1,000 requests
GET and all other Requests †	\$0.01 per 10,000 requests
† No charge for delete requests	

Pricing

Data Transfer Pricing

Region: <input type="text" value="US Standard"/>	
Pricing	
Data Transfer IN	
All data transfer in	\$0.000 per GB
Data Transfer OUT	
First 1 GB / month	\$0.000 per GB
Up to 10 TB / month	\$0.120 per GB
Next 40 TB / month	\$0.090 per GB
Next 100 TB / month	\$0.070 per GB
Next 350 TB / month	\$0.050 per GB
Next 524 TB / month	Contact Us
Next 4 PB / month	Contact Us
Greater than 5 PB / month	Contact Us

Steps

- ▶ Create a Bucket to store your data. You can choose a Region where your bucket and object(s) reside to optimize latency, minimize costs, or address regulatory requirements.
 - ▶ Upload Objects to your Bucket. Your data is durably stored and backed by the Amazon S3 Service Level Agreement.
 - ▶ Optionally, set access controls. You can grants others access to your data from anywhere in the world.
- 

Common Use Cases

- ▶ Content Storage and Distribution
- ▶ Storage for Data Analysis
- ▶ Backup, Archiving and Disaster Recovery

Features

- ▶ Secure
 - ▶ Reliable
 - ▶ Scalable
 - ▶ Fast
 - ▶ Inexpensive
 - ▶ Simple
- 