

Spark

 **hadoop**
MapReduce

APACHE
HBASE

 Flink

H₂O

 cloudera
IMPALA

FILESYSTEM(S) DISTRIBUTI

 S3

 GlusterFS

 **HDFS**

 openstack™

NFS

 ceph

Filesystem Distribuito (DFS)

È un particolare FS che permette la memorizzazione dei file e delle risorse in dispositivi di archiviazione distribuiti in una rete informatica (LAN,MAN,WAN).

- I dati non sono archiviati su un dispositivo locale ma, attraverso **un sistema C/S**, su dispositivi remoti collegati in modo trasparente alla propria gerarchia dei file.
- Un **DFS** deve gestire i file in modo **concorrente e trasparente** e può essere dotato di **autenticazione e crittazione**.

File Server

- È un host che ospita un DFS offrendo una serie di servizi ai client che lo sfruttano.
- Sui client è installata una **interfaccia** al **file server** che include delle operazioni che normalmente si fanno sui file locali.
 - Su **NFS** e **CIFS** questa interfaccia è **trasparente**: si utilizzano i comandi di sistema sui file come se fossero locali.
- Il **file server** controlla un insieme di dispositivi di memoria di massa su cui agisce in base alle richieste dei client.
- Può (DEVE) essere *replicato, ridondato* ecc...

Filesystem Distribuito – vista lato server

- In un ***filesystem distribuito*** i dispositivi di memorizzazione sono dislocati in una rete.
 - Le richieste e le risposte devono essere trasportate attraverso tale rete.
 - Invece di un dispositivo unico e centralizzato, il sistema ne può avere **molte** ed **indipendenti**.
- Le implementazioni di un **filesystem distribuito** possono variare:
 - In talune occasioni il **file server** viene eseguito su un'unica macchina dedicata.
 - In altre il sistema client accede simultaneamente a più **file server**.
 - La stessa macchina può ospitare sia un **file server** che un client.

Filesystem Distribuito – vista utente

- **Trasparenza:** idealmente un **filesystem distribuito** appare all'utente come un normale filesystem centralizzato: *la molteplicità e la dispersione dei server e dei dispositivi cui si riferisce **devono** essere celati.*
- L'interfaccia rivolta alle applicazioni che usa l'utente non dovrebbe poter distinguere tra file locali e file remoti: è *compito del filesystem distribuito individuare e trasportare i dati per mezzo della rete.*
- L'aspetto che maggiormente incide, lato utente, sulla **trasparenza** sono le **prestazioni** del filesystem distribuito.

Filesystem Distribuito - Prestazioni

- Il modo più conveniente per misurare le prestazioni di un **filesystem distribuito** è **quantificare l'ammontare di tempo impiegato per soddisfare una data richiesta**.
- Nei **sistemi convenzionali** questo tempo consiste nell'accesso al disco locale e in piccola quantità di elaborazione da parte della CPU.
- Nei **sistemi distribuiti** si somma (al precedente) il ritardo dovuto alle comunicazioni di rete (meno sono e meglio è).
 - Tale ritardo include il tempo necessario a sottoporre la richiesta al server, quello per ottenere la risposta attraverso la rete, che si sommano al tempo necessario alla CPU per manipolare il protocollo di comunicazione in ciascuna direzione.
- Le prestazioni di un **filesystem distribuito** incidono sul suo **livello di trasparenza**: idealmente, un sistema distribuito dovrebbe avere una velocità paragonabile a quella di un sistema convenzionale.

Filesystem Distribuito – accesso concorrente

- Un **filesystem distribuito** deve provvedere non solo all'accesso dei client ai files, ma anche alla loro modifica:
 - **gli aggiornamenti operati da un client non possono interferire con gli accessi e le modifiche fatte da altri client.**
- Sono quindi necessari meccanismi di controllo della **concorrenza** e di **locking** che possono essere:
 - Inclusi nella realizzazione del filesystem distribuito.
 -
 - Resi disponibili da un protocollo parallelo.

DFS – Implementazioni

- **NFS**: il primo **DFS** ad essere sviluppato (SUN 1985) diffusissimo e molto usato, ora alla versione 4.
- **CIFS/SMB**: filesystem distribuito di Microsoft.
- **AFS: Andrew File System**, sviluppato dalla **Carnegie Mellon University**, all'interno del **progetto Andrew**. Il nome è stato dato in onore di Andrew Carnegie e Andrew Mellon. Ha diversi vantaggi rispetto ai tradizionali file system distribuiti, in particolare riguardo alla sicurezza e alla scalabilità. Non è raro in ambito commerciale che questo file system supporti oltre 50.000 client. Utilizza **Kerberos** per eseguire le autenticazioni, e implementa liste per il controllo degli accessi alle singole cartelle per utenti e gruppi.

DFS – Implementazioni

- **Google FS:** è un filesystem distribuito **proprietario** e sviluppato da Google per proprio uso. Ottimizzato per l'immagazzinamento dei dati di Google. È chiamato anche "BigFiles", ed è stato una delle prime tecnologie sviluppate dai fondatori di Google Larry Page e Sergey Brin. Usato nei Cluster Computer (cloud) di Google.
- **Coda:** sviluppato dalla **Carnegie Mellon University** a partire da AFS, in particolare AFS2, dal 1987. Presenta caratteristiche interessanti, alcune derivate da AFS, altre proprie tra cui: elevate performance, replica, sicurezza, adattamento alle prestazioni della rete, scalabilità, ecc...
- **CephFS....**

Network File System (NFS)

- Sviluppato da **SUN**, presentando al mondo nella versione 2 nel 1985, ad oggi siamo alla versione 4.
- È un modello per integrare filesystem(s) locali differenti, basato sull'idea che **ogni sistema (host) fornisce una vista unificata del suo filesystem locale**.
- **NFS** può essere usato su gruppi eterogenei di computer (diverso SO, diverso HW, ecc).
- MS ha chiesto agli sviluppatori di implementarne una versione per Windows che è già esistente e funzionante da un bel po.
 - MS ha chiesto anche l'aggiornamento alla versione 4.

Network File System (NFS)

- Usa un **remote access model**:
 - I nodi client non sanno realmente dove sono i file.
 - I server esportano ai client una serie di operazioni sui file.
 - Si accede direttamente ai file e alle directory sui file server.
-

- Altri sistemi usano un **modello upload/download**:
 - Il nodo client scarica il file in una cache locale.
 - A modifiche avvenute il file caricato sul server.
 - Il server dovrebbe mantenere le versioni dei file.

Esempi sono i sistemi di versioning: **svn**, **git**, **cvs** ecc.

Altro esempio sono i **Roaming Profiles** di Windows Server.

Network File System (NFS)

- È indipendente dall'organizzazione del file system locale.
 - Per questo riesce ad integrare i vari filesystem di Unix, Linux, Windows, OS X.
- **Esporta all'utente una visione simile a quella dei filesystem Unix Like basato su files organizzati come sequenze di byte.**
- Utilizza come protocollo sottostante **RPC**.
- Dalla **versione 4** supporta server **non stateless**:
 - Fino alla versione 3 i client dovevano mantenere lo stato delle operazioni correnti su un fs remoto NFS.
 - Ora nella versione 4, **i server NFS mantengono lo stato delle operazioni**. Come?

Network File System (NFS) v4

- Nella versione **4**, **NFS** ha introdotto le **Compound Operations (CP)** che **comprendono più richieste di operazioni in una singola chiamata** riducendo così il numero di chiamate **RPC** **migliorando di fatto le prestazioni**.
- Le **CP** non vengono gestite come transazioni:
 - Se una operazione in una **CP** fallisce, le successive operazioni non vengono eseguite.
 - Viene ritornato un messaggio con le informazioni sulle operazioni eseguite e l'errore che si è verificato (no rollback).
 - Non conviene, quindi, inviare operazioni non correlate tramite una **CP**.
- La versione **4** ha anche portato un nuovo approccio più adatto ai **wide area distributed filesystem**, rinnovando e introducendo:
 - Sistemi di file locking.
 - Protocolli di cache consistency.
 - Procedure di Callback.

Network File System (NFS) v4

- Si basa sul protocollo **TCP** (in precedenza era **UDP**).
- Il server mantiene delle informazioni sullo stato delle operazioni sul **DFS** (**stateful**).
- Presenta un sistema per il **riconoscimento dell'utente che avviene tramite una stringa arbitraria** (ad esempio username). La traduzione tra queste stringhe e le informazioni necessarie a client e server avviene tramite un **id mapper** (demone `idmapd`). *Nelle versioni precedenti avveniva solo tramite IP del client.*
 - L'identità degli utenti può essere provata tramite sistemi di autenticazione esterni (**Kerberos**). In precedenza, basandosi solo sull'IP del client, l'unica protezione erano i permessi a livello di sistema operativo.
- Vengono introdotte nel protocollo delle possibili **migliorie delle prestazioni** (ad es. le deleghe) e **dell'affidabilità** (ad esempio il supporto per la replicazione, che però non è implementato direttamente nel protocollo).

NFS - Installazione e configurazione

In Ubuntu/Debian:

```
# sudo apt install nfs-kernel-server ; sul server
# sudo apt install nfs-common ; su client e server
```

- Se non esiste, create la cartella da esportare, ad esempio:

```
# sudo mkdir /srv/nfs
# sudo chmod -R ugo+rwX /srv/nfs
```

- Per la configurazione editare `/etc/exports`, che ha la sintassi

```
/percorso/cartella/condivisa    indirizzoip(opzioni)
/percorso/cartella/condivisa    hostname(opzioni)
/percorso/cartella/condivisa    rete(opzioni)
```

**DOMANDA
DA ESAME**

Ad esempio:

```
/srv/nfs    <vostrarete>(rw,sync,no_subtree_check)
```

- Riavviate il servizio:

```
# sudo systemctl restart nfs-kernel-server
```

<vostrarete> somiglia a
192.168.100.0/24

NFS - Installazione e configurazione

- Provate a montare la vostra condivisione:

```
# sudo mkdir /mnt/nfs
```

<ipvostropc> è simile a
192.168.100.xx

```
# sudo mount <ipvostropc>:/srv/nfs /mnt/nfs
```

- Accedere e creare il file `prova.txt`:

```
# cd /mnt/nfs
```

```
# sudo touch prova.txt
```


NFS - Installazione e configurazione

Altri esempi li trovate in `/etc/exports` nei commenti:

```
# /etc/exports

# Example for NFSv2 and NFSv3:

# /srv/homes      hostname1(rw,sync,no_subtree_check) \
hostname2(ro,sync,no_subtree_check)

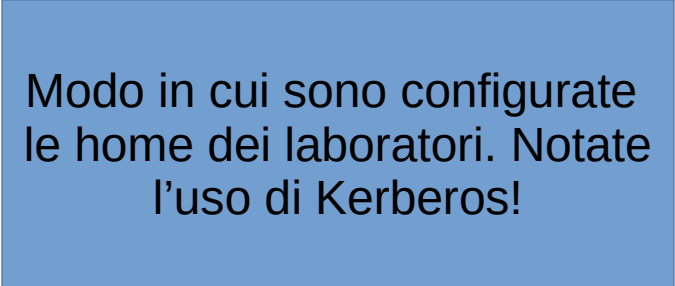
# Example for NFSv4:

# /srv/nfs4      #gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)

# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)

/home 157.138.22.21(rw, no_root_squash)

/home 157.138.0.0/16 (rw,async,
no_subtree_check,sec=krb5:krb5i:krb5p)
```



Modo in cui sono configurate le home dei laboratori. Notate l'uso di Kerberos!

NFS - Installazione e configurazione

- Per utilizzare un'area condivisa con **NFS** vi sono tre modalità:

1) Effettuare il **mount manuale**:

```
# sudo mount <ipvostropc>:/srv/nfs /mnt/nfs
```

2) Inserire il mount in **/etc/fstab** in modo che la condivisione sia montata in avvio (**mount statico**):

```
<ipvostropc>:/srv/nfs /mnt/nfs nfs rw,user,auto 0 0
```

3) Usare **automount**: soluzione migliore in caso di molti client e molti utenti che non hanno accesso diretto all'utente amministratore (`root`) e/o ai file di sistema (`/etc/fstab`).

- **NB:** 1,2 funzionano solo se agite come root!

/etc/fstab – Esempio mount statico

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name
# devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>    <type>  <options>          <dump>
<pass>
proc          /proc          proc    nodev,noexec,nosuid 0
0
/dev/mapper/g-r /             ext4    errors=remount-ro 0
1
/dev/mapper/g-h /home          ext4    defaults          0      2
/dev/mapper/g-s none           swap    sw                0      0

<ipvostropc>:/srv/nfs /mnt/nfs  nfs rw,user,auto 0 0
```

/etc/fstab – Esempio come erano i laboratori... mount statico

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name
# devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>    <type>  <options>          <dump>
<pass>
/dev/sda5      /                ext4     errors=remount-ro 0          1
/dev/sda6 none           swap      sw              0          0
roar.dsi.unive.it:/s1      /stud/s1  nfsvers3      0          0
roar.dsi.unive.it:/s2      /stud/s2  nfsvers3      0          0
roar.dsi.unive.it:/s3      /stud/s3  nfsvers3      0          0
roar.dsi.unive.it:/s4      /stud/s4  nfsvers3      0          0
```

AUTOMOUNT

- *Non è consigliabile mantenere un **file system remoto** montato permanentemente su un sistema.*
 - In caso di fallimenti della rete, ad esempio, le operazioni che interessano dei file presenti nelle partizioni montate, potrebbero essere bloccate.
 - Inoltre, se si montano separatamente una gran quantità di posizioni (ad esempio le home directory degli utenti), in alcuni Unix si rischia di eccedere il numero massimo di mount.
- Per questo vari UNIX (tra cui Linux) implementano un meccanismo di **automount**: *quando un processo accede ad un file o ad una directory all'interno di un certo percorso gestito da un DFS, questo viene montato all'istante. Dopo un certo periodo di inattività, dopo che tutti i file sotto a quel mount point sono stati chiusi, il file system viene smontato.*

AUTOMOUNT

- In Linux, il sistema **automount** viene gestito dal **kernel** stesso, tramite l'*infrastruttura automount*, che si occupa:
 - Delle operazioni effettuate in un certo percorso (cancellazione, copia, creazione, modifica, ecc) dove è montato il **filesystem condiviso**.
 - Delle operazioni di mount/unmount di tale filesystem.
- La *tabella* di *corrispondenza tra percorso e locazione*, così come altre funzioni accessorie, sono mantenute da un programma in **user space** chiamato **autofs** (pacchetto omonimo).
 - In **autofs** le tabelle di corrispondenza possono sia risiedere sul filesystem, sia provenire da fonti esterne, come ad esempio un database **LDAP**.

AUTOMOUNT – installazione e configurazione

- In Ubuntu/Debian:

```
# sudo apt install autofs
```

- Per la configurazione basta editare il file `/etc/auto.master` da cui si possono ramificare altri file che gestiscono la configurazione per le directory specificate:

```
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
```

```
/misc    /etc/auto.misc -timeout=60
/smb     /etc/auto.smb
/misc    /etc/auto.misc
/net     /etc/auto.net
```

```
/mnt    /etc/auto.mnt
```

AUTOMOUNT- Esempio: montare share NFS

- Supponiamo di aver configurato il server con IP `192.168.100.xx` per esportare via **NFS** la directory `/srv/nfs`.
- Editiamo il file `/etc/auto.master` aggiungendo la riga:

```
/mnt                /etc/auto.mnt
```
- Creiamo `/etc/auto.mnt` aggiungendo:

```
- nfs -fstype=nfs4      192.168.100.xx:/srv/nfs
```


AUTOMOUNT Esempio: montare share NFS

- Riavviamo autofs:

```
# sudo service autofs restart
```

- Andiamo nella directory `/mnt/nfs`:

```
# cd /mnt/nfs
```

```
# ls
```

```
. .. prova.txt
```

- Così funzionano le home linux qui in laboratorio.
- E quelle Windows? Lo vediamo più avanti.

Ora in laboratorio....grazie ad automount....

```
# cat /etc/fstab
# <file system> <mount point>    <type>  <options>          <dump>  <pass>
# / was on /dev/sda5 during installation
UUID=..... /                ext4      errors=remount-ro 0          1
# swap was on /dev/sda6 during installation
UUID=..... none              swap      sw              0          0
```

```
# mount:
/etc/autos1.map on /stud/s1 type autofs
(rw,relatime,fd=6,pgrp=1382,timeout=300,minproto=5,maxproto=5,indirect)
/etc/autos2.map on /stud/s2 type autofs
(rw,relatime,fd=12,pgrp=1382,timeout=300,minproto=5,maxproto=5,indirect)
/etc/autos3.map on /stud/s3 type autofs
(rw,relatime,fd=18,pgrp=1382,timeout=300,minproto=5,maxproto=5,indirect)
/etc/autos4.map on /stud/s4 type autofs
(rw,relatime,fd=24,pgrp=1382,timeout=300,minproto=5,maxproto=5,indirect)
.....
roar.dsi.unive.it://s3/faromano on /stud/s3/faromano type nfs4 (...)
```

Condivisione File Windows

- In generale in un **Dominio Active Directory** vi sono due tipologie di cartelle condivise strettamente legate agli utenti del sistema:
 - **I profili.**
 - **Le Home.**
- I profili possono risiedere:
 - **solo sulla macchina locale (client):** le impostazioni utente, i dati delle applicazioni, desktop e documenti risiedono solo su un client e non vengono trasportati su altri client nel caso l'utente cambi postazione.
 - **sul server: **Roaming profiles**,** i profili vengono scaricati su tutti i client a cui un determinato utente si collega, rendendo portabili le impostazioni utente, i dati delle applicazioni, desktop e documenti.

Condivisione File Windows

- **E` importante capire che i Roaming profiles NON coincidono con le Home esportate.**
- Le Home esportate sono, ad esempio qui in laboratorio, innestate nell'unità disco di rete **Z**, (condivisioni che dovrete usare per salvare tutti i vostri file importanti).
- Anche i *profili sono in una cartella condivisa ma vengono scaricati/caricati dal server ogni volta che fate login/logout. Non sono live come in NFS!!!*
 - **Spesso fanno fatica a sincronizzarsi.**
- In Windows la condivisione dei file è gestita dal protocollo **CIFS** di cui esiste una implementazione per Linux di nome **SAMBA**.

Common Internet File System (CIFS)

- È un **protocollo standard** che permette la **condivisione di file e risorse all'interno di una LAN.**
- È una versione migliorata del protocollo di **Microsoft SMB (Server Message Block).**
- Nasce con Windows 2000 come unico protocollo di condivisione risorse.
- Nonostante molte migliorie diventa inefficiente in reti di medie/grosse dimensioni, come tanti altri DFS.
- Ma è comunque uno dei migliori protocolli per la realizzazione di filesystem distribuiti.

CIFS - Caratteristiche

- **Integrità e concorrenza:** permette l'accesso multiplo allo stesso file evitando conflitti tramite **file locking**.
- **Ottimizzazione per connessioni lente (caching?).**
- **Sicurezza:** supporta sia trasferimento anonimo dei file sia accesso autenticato e crittografato.
- **Performance e scalabilità:** essendo integrato nel SO, le performance e la scalabilità migliorano con le versioni successive di Windows.
- **Nomi file unicode:** supporta varie codifiche dei caratteri.
- **Nomi file globali:** gli utenti non devono necessariamente montare il filesystem remoto ma possono riferirsi ai file usando path **UNC (Uniform Naming Convention)**:

`\\nas\bustepaga\pagamiseradisetteembre.pdf`

CIFS - Utilizzo

- Come si usa:
 - Spostatevi sul PC (ad esempio `contabilita`) con la cartella da condividere
 - Tasto destro su una cartella che volete condividere, ad esempio `"fatture"`.
 - Scegliete share o condividi.
 - Impostate quali utenti/gruppi possono accedere alla condivisione e con quali permessi.
 - Sul client aprite file explorer e sulla barra degli indirizzi digitate:

`\\contabilita\\fatture.`

SAMBA

- È un software che, girando su piattaforme Unix/Linux like, permette, una volta configurato, di interagire con i sistemi Microsoft Windows fornendo servizi di condivisione file e stampanti.
- È un'implementazione di **CIFS** per sistemi UNIX Like.
- Tramite **SAMBA** è possibile inserire una macchina Linux/Unix all'interno di un **dominio** Windows.
- Può svolgere alcune funzioni del servizio **Active Directory** di MS su piattaforme non Windows.

SAMBA

- **Samba** permette di configurare **share (condivisioni)**, via **SMB(CIFS)**, per le directory UNIX scelte (incluse le sottodirectory).
- Queste *appaiono agli utenti di Microsoft Windows come normali cartelle di Windows accessibili via rete.*
- Per accedere alle condivisioni gli utenti Unix possono:
 - 1) effettuare il mount delle directory condivise direttamente nell'albero del filesystem (`mount -t cifs ...`),
 - 2) usare una utility, `smbclient` installata con **samba** per leggere le directory condivise mediante un'interfaccia (simile a quella del programma `ftp`) a riga di comando,
 - 3) sfruttare una configurazione ad hoc in `/etc/fstab` (come visto per **NFS**).
 - 4) Utilizzare **automount** come visto per **NFS**.
- **NB: I punti 1,3 prevedono che l'utente sia anche amministratore.**

SAMBA

- Ogni directory può avere differenti privilegi di accesso sovrapposti ai normali privilegi UNIX.
- **Ad esempio:** le home directory potrebbero essere accessibili in lettura/scrittura a tutti gli utenti riconosciuti dal sistema permettendo ad ogni utente di accedere ai propri file, ma non agli utenti che accedono alle home tramite Windows che magari possono solo leggere tali file.
- La configurazione di **Samba** si ottiene modificando un singolo file (di solito presente come `/etc/smb.conf` o `/etc/samba/smb.conf`).

SAMBA 4

- Dalla release 4, **Samba** è in grado di svolgere quasi tutte le funzioni di un Domain Controller (DC), integrandosi anche con Active Directory (AD) di Windows Server.
- Infatti, pur potendo svolgere alcuni semplici compiti di tipo AD (esempio l'autenticazione di un client e la sua registrazione su dominio al login) non è attualmente in grado di implementare interamente la complessità funzionale dell'architettura AD di Microsoft.
- Rimane sostanzialmente uno strumento di interconnessione tra un dominio Microsoft e tecnologie non Microsoft offrendo servizi di condivisione di risorse di rete.

Esempio semplice di Samba su Ubuntu

- Supponiamo di avere un server con ip <ipvostravm> con accesso da amministratore (root o utente sudoer).
- Installiamo samba 4.x.x su ubuntu 22.04:

```
# sudo apt install -y samba samba-common system-config-samba
```

- Salviamo il file di configurazione di default e creiamone uno nuovo:

```
# sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.bak  
# sudo touch /etc/samba/smb.conf
```

Esempio semplice di Samba su Ubuntu

- Inseriamo con il nostro editor del cuore le seguenti righe nel file di configurazione `smb.conf` (`sudo vi /etc/samba/smb.conf`):

```
[global]
workgroup = LAS
server string = Las Samba Server %v
netbios name = las
security = user
map to guest = bad user
dns proxy = no
#===== Share Definitions =====
[condivisa]
path = /samba/condivisa
browsable =yes
writable = yes
guest ok = yes
read only = no
force user = nobody
```

Esempio semplice di Samba su Ubuntu

- Creiamo la cartella da condividere e assegnamo i permessi corretti:

```
# sudo mkdir -p /samba/condivisa  
# sudo chmod -R 0775 /samba/condivisa  
# sudo chown -R nobody:nogroup  
/samba/condivisa
```

- Infine riavviamo il servizio:

```
# sudo service smbd restart
```

- Da questo momento in poi potete accedere via Windows alla cartella condivisa digitando su una finestra di esplora risorse:

```
\\<ipvostropc>\condivisa
```

Problema

- In un ambiente multiutente, spesso vi è competizione tra gli utenti stessi per l'uso delle risorse.
- Una delle più importanti, poiché l'uso non è limitato al periodo in cui l'utente sta effettivamente operando sulla macchina, è lo **spazio di memorizzazione**.
- Questo spazio, per ragioni di prestazioni, affidabilità e disponibilità continua, è generalmente **molto più costoso** di quanto lo sia l'equivalente nell'ambito dei personal computer.
- È **necessario proteggere il sistema** e gli altri utenti, da chi vuole monopolizzare la risorsa sia volontariamente, che involontariamente (programmi errati che esauriscono lo spazio disco).
- **Quando lo spazio disco termina... nessun utente può lavorare.....**

Soluzione: Quota

- Il concetto di **quota disco** è un limite oltre al quale uno specifico utente non può più occupare spazio.
- La **quota** può essere usata anche su filesystem locali, ma trova la sua più naturale applicazione quando le home directory (o altri dati personali) degli utenti vengono condivise tramite un NAS o un file server in generale, quindi su un DFS.
- In Linux il **sistema di quota** viene gestito direttamente dal Kernel mentre le utility per la gestione sono presenti nel pacchetto `quota`.
- La **quota**, se presente, viene automaticamente usata anche per le operazioni sui filesystem esportati, tuttavia, se si vuole poter conoscere le informazioni sulla quota disponibile (comando `quota`) anche sui client, è necessario usare il demone `rpc.rquotad` (presente nel pacchetto `quota`).

Quota

- La **quota** è espressa in 4 quantità:
 - La **soft quota sul numero di blocchi** (spazio disco) occupati. Dopo aver raggiunto questo limite si ottiene un warning.
 - La **hard quota sul numero di blocchi**. Al raggiungimento di questo limite, ogni ulteriore richiesta di allocazione viene immediatamente rifiutata
 - La **soft quota sugli inode(*) / numero di file**.
 - La **hard quota sugli inode(*) / numero di file**.
- **soft quota < hard quota**
- Per le **soft quota** è possibile impostare, un **grace period**. *Se, oltre tale periodo, l'utente si mantiene sopra la soglia della soft quota, ogni altra allocazione verrà rifiutata (praticamente si trasforma in hard quota).*

• (*) **inode**: è una struttura dati sul file system che archivia e descrive attributi base su file, directory ecc...



NON RIESCO A SCARICARE TORADORA :(

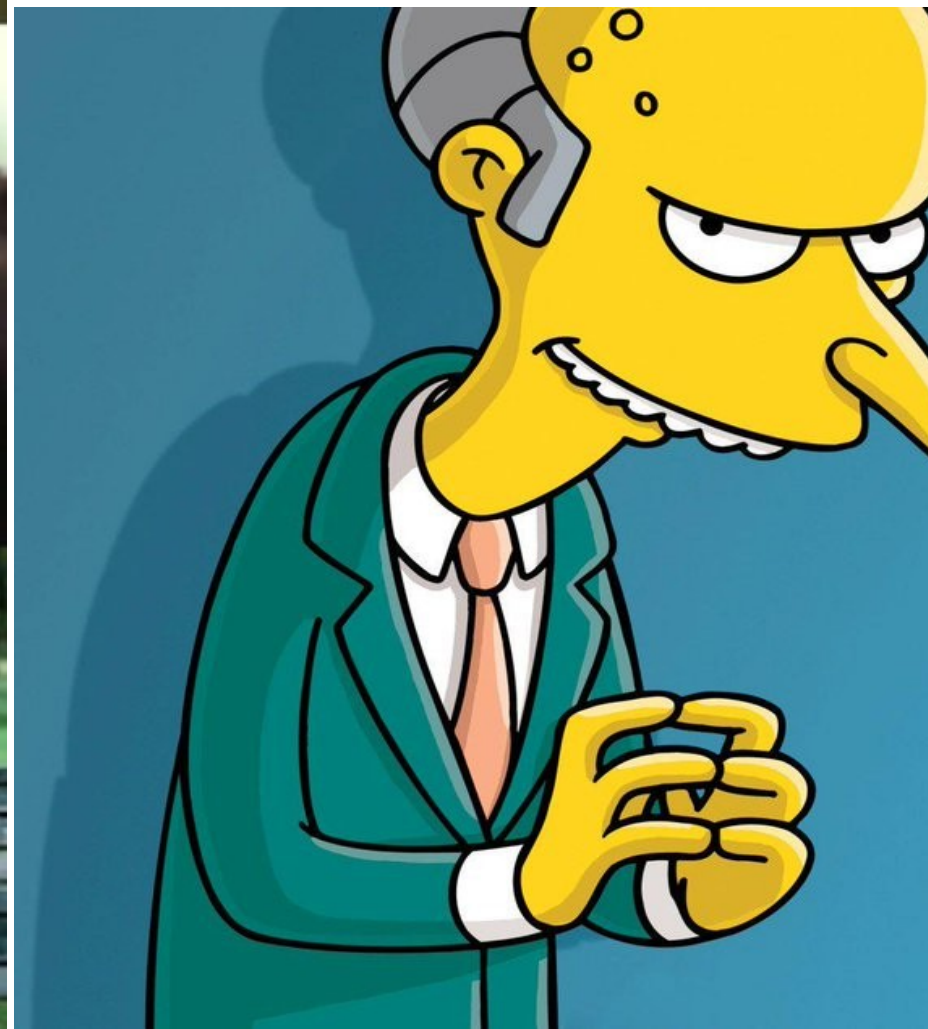


MA SE CI SONO 10 TB DI SPAZIO....



IL SYSADMIN HA MESSO LA QUOTA!

imgflip.com



Quota - Esempio

- Supponiamo per semplicità di avere un server multiutente in cui gli utenti hanno delle pagine web in `/var/www`, filesystem montato a parte.
- Creiamo `/var/www`:

```
# sudo mkdir /var/www
```
- Installiamo il pacchetto quota:

```
# sudo apt install quota
```
- Editiamo il file `/etc/fstab`, abilitando la quota sulla riga corrispondente al filesystem `/var/www`:

```
UUID=.. /var/www      ext4    usrquota,grpquota 0 1
```
- Rimontiamo `/var/www` abilitando le quote:

```
# sudo mount -o remount /var/www
```

Quota - Esempio

- Il seguente comando crea un nuovo file delle quote nella directory principale del file system.
- *Questo è un file indicizzato utilizzato dallo strumento per la gestione della quota per tenere traccia dello spazio disco usato dall'utente.*
- *Esso contiene anche i limiti per gli utenti e le opzioni configurate.*

```
# sudo quotacheck -cug /var/www  
/* c:crea, u:user, (g:group) */
```

- Abilitiamo la quota:

```
# sudo quotaon /var/www
```

Quota - Esempio

- A questo punto avete un sistema di quota sulla partizione `/var/www` che potete gestire con i comandi: `edquota`, `setquota` e `repquota`.

- Ad esempio “quotiamo” l’utente las:

```
# setquota -u las 10000 20000 0 0 /var/www
```

- Creiamo una directory per il sito dell’utente las:

```
# sudo mkdir /var/www/lassite
```

```
# sudo chown -R las:las /var/www/lassite
```

```
# touch /var/www/lassite/index.html
```

- Verifichiamo:

```
# quota las
```

```
Disk quotas for user las (uid 1000):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sda5	4	10000	20000		1	0	0	

Quota - Esempio

- Proviamo il comando `repquota`:

```
# repquota -u /var/www
```

```
*** Report for user quotas on device /dev/sda5
```

```
Block grace time: 7days; Inode grace time: 7days
```

		Block limits				File limits			
User		used	soft	hard	grace	used	soft	hard	grace
root	--	20	0	0		2	0	0	
las	--	4	10000	20000		1	0	0	

Ceph

- **Ceph** è una soluzione di archiviazione software-defined open source progettata per soddisfare le esigenze di archiviazione distribuita.
- **Ceph** fornisce uno storage affidabile e scalabile mantenendo costi contenuti allineati con quelli dell'hardware sottostante.
- **Ceph** consente di “separare” i dati dall'hardware di storage fisico utilizzando livelli di astrazione software, che forniscono capacità di scalabilità e gestione dei guasti senza pari.

Ceph

- **Ceph** è quindi ideale per cloud, Openstack, Kubernetes e altri carichi di lavoro basati su microservizi e container, in quanto può soddisfare efficacemente le esigenze di archiviazione di grandi volumi di dati.
- Il vantaggio principale di **Ceph** è che fornisce interfacce per più tipi di storage all'interno di un singolo cluster, eliminando la necessità di più soluzioni di storage o qualsiasi hardware specializzato, riducendo così i costi generali di gestione.
- **Ceph** può essere usato nell'infrastruttura cloud privata, nell'analisi dei big data, o come alternativa allo storage su cloud pubblico.
- Refer: <https://ubuntu.com/ceph/what-is-ceph>

CephFS

- **Ceph File System (CephFS)** è un file system distribuito robusto e completo con funzionalità di snapshot, quote e mirroring multi-cluster.
- I file gestiti da **CephFS** vengono distribuiti tra gli oggetti archiviati da **Ceph** per prestazioni e scalabilità estreme.
- I file **CephFS** vengono mappati agli oggetti che Ceph archivia nel **Ceph Storage Cluster**.
- I client Ceph montano un filesystem CephFS come oggetto kernel o come filesystem in User Space (**FUSE**).
- I sistemi Linux possono montare i filesystem CephFS in modo nativo, tramite un client basato su **FUSE** o tramite un gateway NFSv4.
- Refer: <https://docs.ceph.com/en/latest/>

Progetto

- Realizzare un sistema con filesystem distribuito NFS e quota.
- Provare ad analizzare il sistema delle quote in Windows Server.
- Realizzare un sistema multiutente con home in locale e gestione della quota.
- Realizzare una condivisione tra una macchina windows e una linux usando samba.
- Realizzare un server di dominio active directory usando samba.

Avete un NAS con FS Distribuito!

