

# INSTALLAZIONE ED OTTIMIZZAZIONE LINUX SU UNITA' SSD

## PREMESSA

Le unità a stato solido SSD si basano su memoria flash di solito di tipo NAND per la memorizzazione dei dati, e per questo non richiedono parti meccaniche o magnetiche (dischi, motori e testine) portando quindi notevoli vantaggi soprattutto in merito alla sicurezza dei dati.

I principali vantaggi sono:

- rumorosità assente
- minore possibilità di rottura
- minori consumi
- minore produzione di calore
- tempo di accesso notevolmente ridotto

Purtroppo le SSD hanno anche degli svantaggi:

- una minore durata dell'unità stessa a causa del limite di riscritture delle memorie flash
- capacità (in GB) inferiore rispetto agli HDD, almeno per il momento poi in futuro...
- maggior costo per Gigabyte
- prestazioni che degradano con l'aumentare dello spazio occupato: con qualsiasi filesystem, l'occupazione massima di ogni partizione non deve superare il 75% del totale per garantirne un uso efficiente da parte del Kernel.

In questa guida sono indicati tutti gli step da seguire per installare ed ottimizzare Linux su SSD; inoltre sono indicate delle modifiche da apportare al Computer ed al sistema operativo per permettere ad una SSD di funzionare correttamente. Nessuna delle ottimizzazioni proposte è obbligatoria perché le unità SSD funzionano normalmente anche senza, ma hanno il solo scopo di ridurre i cicli di scrittura in modo tale da aumentare la vita stessa della SSD, ma anche per ottenere un ulteriore incremento delle performance.

La presente guida, anche se è stata testata solo su tre distribuzioni (Debian, LMDE versione 201204 e Crunchbang 11 Waldorf) è compatibile con qualsiasi distro Debian based.

L'unico problema rilevato è con LMDE versione 201303 in cui non è più possibile, come nelle precedenti versioni, impostare la tabella delle partizioni come GPT quindi è obbligatorio usare MBR (potrebbe essere un bug dell'installatore di LMDE o proprio la mancanza del supporto al GTP).

Per le altre distribuzioni sicuramente sarà necessario effettuare qualche piccola modifica ma in linea di massima i punti sono gli stessi.

## Capitolo 01 - PRIMI ACCORGIMENTI

La prima cosa da fare è assicurarsi che il Computer sia pronto ad ospitare una SSD, quindi il primo passo da fare è verificare che il Controller SATA sia settato in modalità AHCI.

L'AHCI, ossia Advanced Host Controller Interface, è un meccanismo hardware che permette di comunicare con le periferiche SATA ed è molto importante perché è in grado di offrire funzionalità avanzate come ad esempio l'NCQ (Native Command Queuing) che permette di aumentare la velocità con cui sono eseguiti i comandi di lettura. AHCI è pienamente supportato da Linux a partire dalla versione del kernel 2.6.19.

Per verificare che il Controller SATA sia impostato correttamente è necessario entrare nel BIOS o UEFI, ma non verranno spiegati i passi per farlo perché esistono una moltitudine di versioni di BIOS e quindi la guida diventerebbe troppo articolata.

Normalmente è possibile settare il Controller SATA in modalità IDE, AHCI o RAID, ed in teoria la scelta andrebbe fatta prima di installare il sistema operativo.

## Capitolo 02 - PARTIZIONAMENTO ED ALLINEAMENTO

Durante l'installazione del Sistema Operativo su SSD è vivamente raccomandato, ma non obbligatorio, prima del partizionamento impostare la tabella delle partizioni come GPT anziché MBR, inoltre è fondamentale effettuare un corretto allineamento delle partizioni create per ottenere prestazioni ottimali e longevità.

A differenza dei dischi con tabelle delle partizioni di tipo MBR, nei dischi GPT i dati fondamentali per il funzionamento del sistema si trovano nelle partizioni anziché in settori non partizionati o nascosti. Nei dischi GPT sono inoltre disponibili tabelle di partizioni primarie e di backup ridondanti per garantire una maggiore integrità della struttura dei dati della partizione. Per ottenere quindi l'allineamento, ogni singola partizione dovrebbe incominciare ad un valore di byte che è un multiplo dell'EBS (Erase Block Size).

L'EBS è per lo più dipendente dal costruttore; una ricerca su Google sul modello di interesse potrebbe essere un'ottima idea! Se non si conosce l'EBS del proprio SSD, è comunque possibile usare una dimensione di 512 KiB (o 1024 KiB se si vuole essere sicuri e non ci si preoccupa di perdere il primo MiB della propria SSD).

Per prima cosa avviare il PC con un LiveCD. Per partizionare la SSD è possibile usare per comodità il Gparted altrimenti è possibile farlo manualmente tramite “fdisk” se si vuole uno schema di partizione MBR, altrimenti è necessario utilizzare “gdisk” per lo stile GPT (infatti il comando “fdisk” non supporta il GPT).

In questa guida, avendo a disposizione una SSD da 240GB, ho deciso di partizionare secondo lo schema seguente:

- Partizione / di tipo ext4 da 20 GB
- Partizione /home di tipo ext4 da 80 GB di tipo ext4
- Partizione /Dati di tipo ext4 per il restante spazio libero (123.57 GB)

E' consigliato, con un quantitativo di RAM superiore a 2GB, di non creare una partizione di Swap in modo tale da ridurre ulteriormente eventuali scritture sulla SSD. E' da considerare comunque che i sistemi moderni che hanno generalmente più di 2GB di RAM raramente utilizzano lo spazio di Swap. Se invece si crea una partizione di Swap è necessario fare in modo che il sistema lo usi il meno possibile: il Kernel Linux dà la possibilità di controllare la frequenza di utilizzo dello Swap modificando la variabile di swappiness (spiegato successivamente nel Capitolo 05.B).

Una volta create le partizioni, per verificare che siano allineate è necessario installare il pacchetto “gdisk” perché come detto il comando “fdisk” non supporta il GPT:

1. Aprire il Terminale, quindi digitare il comando per ottenere i permessi di root:  
**# sudo su**
2. Effettuare un update del repository e successivamente installare “gdisk”:  
**# apt-get update**  
**# apt-get install gdisk**
3. Quindi, per avere i dettagli delle partizioni, digitare:  
**# gdisk -l /dev/sda**

Il risultato del comando è:

```
Found valid GPT with protective MBR; using GPT.
Disk /dev/sda: 468862128 sectors, 223.6 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 90B445E7-6783-48EC-9104-A9933442DC92
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 468862094
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number Start (sector) End (sector) Size Code Name
  1      2048         41945087  20.0 GiB  8300 Linux filesystem
  2    41945088    209717247  80.0 GiB  8300 Linux filesystem
  3    209717248    468862094  123.6 GiB  8300 Linux filesystem
```

Come si può notare, i settori iniziali (seconda colonna, Start sector) sono tutti multipli di 512, ossia del “Logical sector size” (che nell'esempio è di 512 bytes).

Per chi ha optato per uno stile di partizionamento di tipo MBR deve utilizzare il comando:

**# fdisk -l /dev/sda**

Completato il partizionamento, è possibile proseguire con l'installazione. Non verranno spiegati i passi perché oramai quasi tutte le distribuzioni sono di facile installazione anche per i neofiti.

## Capitolo 03 - RIMOZIONE DEL JOURNALING

Finita l'installazione della distribuzione è possibile continuare con gli step successivi, ed il primo passo è disabilitare il journaling.

Il journaling è una tecnica utilizzata da molti file system moderni per preservare l'integrità dei dati da eventuali cadute di tensione. È una tecnologia derivata dal mondo dei Database infatti si basa sul concetto di transazione: ogni scrittura su disco è interpretata dal file system come una transazione. Quando un applicativo invia dei dati al file system per salvarli su disco questo memorizza prima le operazioni che intende fare su un file di log e solo successivamente provvede ad effettuare le scritture sul disco, quindi registra sul file di log le operazioni che sono state effettuate.

In caso di caduta di tensione durante la scrittura sul disco, al riavvio del sistema operativo il file system non dovrà far altro che analizzare il file di log per determinare quali sono le operazioni che non sono state portate a compimento e quindi sarà in grado di correggere gli errori presenti nella struttura del file system. Poiché nel file di log vengono memorizzate solo le informazioni che riguardano la struttura del disco (metadati), un'eventuale caduta di tensione elimina i dati che si stavano salvando, ma non rende incoerente il file system.

I più diffusi file system dotati di journaling sono: NTFS, ext3, ext4, ReiserFS, XFS, Journaled File System (JFS), VxFS, HFS+.

La rimozione del Journaling non è quindi una buona idea se si possiedono HDD classici ma nel caso delle SSD è consigliatissimo per limitare le scritture sul disco ma anche per velocizzare e rendere il sistema più scattante. Dopo questo accorgimento anche il boot viene ulteriormente velocizzato.

Potrebbe sorgere spontanea una domanda: “perché complicarci la vita? non sarebbe più facile e veloce utilizzare direttamente il file system ext2 che non ha il Journaling piuttosto che formattare in ext4 e poi smanettare per disabilitare il Journaling”? Sicuramente è più facile e veloce ma il file system ext2 non ha delle caratteristiche importanti come il supporto per l'HTree (ossia directory index) ma nemmeno gli EXTENTS. Inoltre, sembrerebbe da alcuni test trovati in rete, che l'ext4 senza Journaling risulti più performante dell'ext2.

Per chiarezza ecco alcune delucidazioni sui termini usati in precedenza:

- HTree: è una speciale struttura ad albero dei dati per l'indicizzazione delle directory
- EXTENTS: è un'area di memorizzazione contigua in un file system. Quando si inizia a scrivere un file, viene allocato un intero extent. Quando si scrive di nuovo sul file, i dati vengono scritti a partire dagli ultimi. Questo riduce la frammentazione dei file.

Detto questo, procediamo con la rimozione del Journaling:

1. Riavviare il PC facendolo partire con un LiveCD Linux
2. Assicurarsi che nessuna delle partizioni della SSD sia montata
3. E' necessario visualizzare lo schema delle partizioni ed annotarsi l'indicazione del disco ed il numero delle partizioni interessate attraverso il Gparted (oppure da terminale tramite `gdisk/fdisk`)
4. Verificare lo stato del Journaling sulla partizione desiderata attraverso il comando:  
**# sudo dumpe2fs -h /dev/sdXY**

dove X è l'identificativo del disco SSD e Y è il numero della partizione. Nel risultato del comando guardare la voce "Filesystem features": se compare l'opzione "has\_journal" e se alla fine del report del comando è presente una voce chiamata "Journal size", allora il file system ha il journaling attivato.

5. Per disabilitare il Journaling lanciare, per ogni partizione (tranne sullo Swap, se esistente), il comando:  
**# sudo tune2fs -O ^has\_journal /dev/sdXY**  
specificando l'identificativo del disco (X) ed il numero della partizione (Y) come nel punto 4.

6. Rilanciare il comando (del punto 4) per verificare se il journaling è stato disattivato su ogni partizione esistente
7. Adesso è necessario modificare dei parametri del kernel tramite la modifica del grub. Essendo in LiveCD è necessario montare prima la partizione di root "/" della SSD, questo perché se si lancia il comando direttamente verrà aperto il file del grub relativo al sistema avviato in LiveCD, quindi:

**# sudo mount /dev/sda1 /mnt**

successivamente aprire il file del grub (tramite un editor di testo grafico o vi):

**# sudo vi /mnt/etc/default/grub**

8. Modificare la voce `GRUB_CMDLINE_LINUX_DEFAULT` aggiungendo dopo l'opzione "quiet" le seguenti altre opzioni:
  - **"rootfstype=ext4"** da inserire solo quando il journaling viene disabilitato. In pratica, etichetta la partizione root come ext4 nel caso venga, per qualsiasi motivo, identificata male
  - **"elevator=noop"** per velocizzare il boot ed il sistema in generale. Il kernel di Linux ha diversi modi per ottimizzare le operazioni di I/O. Uno di questi è l'uso dell'opzione "elevator" con cui ordina tutte le richieste di I/O in modo tale da gestire tali richieste in modo ordinato e sequenziale (cioè prima di far partire la lettura/scrittura sulla SSD pensa ad ordinare "i passaggi" così da non dover fare avanti-indietro). Più in particolare possiamo dire che sui dischi tradizionali, composti

da più piatti rigidi rotanti, il metodo di utilizzo è quello di far spostare le testine in modo ordinato e lineare facendo sì che si riduca al minimo il numero di salti da un piatto all'altro; per le unità SSD questo non è valido, quindi vanno ottimizzati gli "elevator" cioè quegli algoritmi che gestiscono le operazioni di scrittura/lettura in modo da rendere più fluido possibile tutti i processi di I/O. Linux è fornito di quattro tipi di elevator ed ognuno è ottimizzato per l'uso di un determinato hardware; di default l'elevator utilizzato è quello per i tradizionali dischi rigidi. Questo algoritmo non è utile per ottenere il massimo delle prestazioni per le unità SSD ed è per questo che conveniente attivare l'elevator chiamato "noop".

Quindi la stringa finale dovrebbe risultare questa:

**GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet rootfstype=ext4 elevator=noop"**

9. Successivamente è necessario effettuare un update del grub ma prima di farlo bisogna effettuare un chroot. Il chroot è un processo tipico degli ambienti Linux che significa "change root" ovvero "cambiare root"; è infatti un utilissimo tool che permette di cambiare, ad esempio, il root di un LiveCD con quello di un possibile sistema operativo installato sulla macchina (proprio come nel nostro caso).  
Dato che la partizione di root della SSD è già stata montata sotto "/mnt", continuare montando le altre partizioni necessarie per effettuare il chroot:

```
# sudo mount --bind /dev /mnt/dev
# sudo mount --bind /proc /mnt/proc
# sudo mount --bind /sys /mnt/sys
```

10. Effettuare un chroot sul proprio sistema, tramite il comando:

```
# sudo chroot /mnt
```

11. Lanciare quindi un update del grub (in questo caso Debian usa il grub2 ma altre distro potrebbero utilizzare il grub):

```
# sudo update-grub2
```

12. Uscire dal chroot eseguendo il comando:

```
# exit
```

13. Smontare tutti i dispositivi digitando i comandi:

```
# sudo umount /mnt/dev
# sudo umount /mnt/proc
# sudo umount /mnt/sys
# sudo umount /mnt/
```

14. Riavviare il sistema rimuovendo il LiveCD.

15. Dopo il riavvio verificare da terminale che nelle partizioni su SSD sia stato effettivamente

disattivato il journaling:

```
# sudo dumpe2fs -h /dev/sdXY
```

specificando l'identificativo del disco (X) ed il numero della partizione (Y) come indicato al punto 4.

## Capitolo 04 – LOG DI SISTEMA E CACHE IN RAM

Per rendere il sistema più scattante e per evitare parecchie riscritture, è possibile spostare i vari log di sistema nonché la cache nella RAM anziché tenerli sulle relative partizioni presenti sulla SSD. È un'operazione molto semplice ma si raccomanda di applicarla se si dispone di una quantità di RAM congrua. Per far ciò è necessario far uso del TMPFS (è possibile utilizzare questa opzione non solo per le SSD ma anche per i normali HDD).

TMPFS indica un tipo di memoria temporanea tipica dei sistemi operativi Unix-like. Questo file system viene “montato” come memoria volatile e non come memoria persistente. Tutto ciò che viene memorizzato in tmpfs è temporaneo, nel senso che nessun file verrà creato sulla SSD o HDD, tuttavia, in caso di esaurimento della memoria a disposizione verrà utilizzato lo spazio di swap, se presente ovviamente. Al riavvio, tutti i file scritti in tmpfs andranno persi, ed inoltre la memoria utilizzata da tmpfs è una memoria dinamica (cresce e si riduce per adattarsi ai file in essa contenuti). Bisogna notare che se si tiene il proprio sistema in esecuzione continuamente, per un lungo periodo, la dimensione di questa directory (e da qui il suo utilizzo di memoria) può crescere considerevolmente. Si raccomanda di fare il logout dal proprio DE, rimuovere il contenuto della directory, e fare nuovamente il login se si hanno problemi di scarsità di memoria. Molte distribuzioni Linux (ad esempio LMDE) utilizzano TMPFS per impostazione predefinita per la partizione /tmp e/o per la memoria condivisa.

Per verificare se nel sistema è già in uso il tmpfs e per quali partizioni, è necessario guardare come sono settate le opzioni RAMLOCK, RAMSHM e RAMTMP nel file “/etc/default/rcS”. Da terminale lanciare:

```
# cat /etc/default/rcS
```

Risultato del comando:

```
#  
# /etc/default/rcS  
#  
# Default settings for the scripts in /etc/rcS.d/  
#  
# For information about these variables see the rcS(5) manual page.  
#  
# This file belongs to the "initscripts" package.
```

```

# delete files in /tmp during boot older than x days.
# '0' means always, -1 or 'infinite' disables the feature
TMPTIME=0
# spawn sulogin during boot, continue normal boot if not used in 30 seconds
SULOGIN=no
# do not allow users to log in until the boot has completed
DELAYLOGIN=no
# assume that the BIOS clock is set to UTC time (recommended)
#UTC=yes # OBSOLETE; see /etc/adjtime and hwclock(8).
# be more verbose during the boot process
VERBOSE=no
# automatically repair filesystems with inconsistencies during boot
FSCKFIX=no
# mount /run/lock as a tmpfs (separately from /run)
RAMLOCK=yes
# mount /run/shm as a tmpfs (separately from /run)
RAMSHM=yes
# mount /tmp as a tmpfs
RAMTMP=yes

```

Come si può notare tutte e tre le opzioni sono già attive ! Per un ulteriore dettaglio è possibile lanciare il seguente comando:

**# cat /proc/mounts**

Risultato del comando:

```

rootfs / rootfs rw 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,relatime,size=10240k,nr_inodes=505703,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
/dev/disk/by-uuid/bea67f52-37bb-46fd-ad03-dc3f2d30f9b0 / ext4 rw,relatime,errors=remount-ro,user_xattr,barrier=1,data=ordered 0 0
tmpfs /var/run tmpfs rw,nosuid,noexec,relatime,size=406000k,mode=755 0 0
tmpfs /var/run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k 0 0
tmpfs /tmp tmpfs rw,nosuid,nodev,relatime,size=812000k 0 0
tmpfs /var/run/shm tmpfs rw,nosuid,nodev,relatime,size=812000k 0 0
/dev/sda2 /home ext4 rw,relatime,errors=remount-ro,user_xattr,barrier=1,data=ordered 0 0
/dev/sda3 /Dati ext4 rw,relatime,errors=remount-ro,user_xattr,barrier=1,data=ordered 0 0
fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0
binfmt_misc /proc/sys/fs/binfmt_misc binfmt_misc rw,nosuid,nodev,noexec,relatime 0 0
cgroupp /dev/cgroup/cpu cgroup rw,relatime,cpu,release_agent=/usr/local/sbin/cgroup_clean 0 0

```

In questo caso è stato fatto uso del tmpfs per le seguenti partizioni:



- /var/run
- /var/run/lock
- /var/run/shm
- /tmp

Possiamo aggiungere alcune che mancano, ad esempio:

- /var/tmp
- /var/cache/apt/archives

In realtà si potrebbe aggiungere anche “/var/log” ma ho avuto qualche problema dopo aver attivato questa partizione in ram, quindi per il momento tralasciamo.

Quindi, aprendo il file “/etc/fstab” con permessi di root e con l'editor preferito, aggiungere le seguenti righe alla fine del file:

```
tmpfs /var/tmp tmpfs defaults,noatime,mode=1777 0 0
tmpfs /var/cache/apt/archives tmpfs defaults,noatime,mode=0755 0 0
```

Volendo aggiungere anche la partizione “/var/log” la stringa da aggiungere è questa:

```
tmpfs /var/log tmpfs defaults,noatime,mode=0755 0 0
```

Per chiarezza ecco delle delucidazioni su alcune opzioni (per un maggiore dettaglio è possibile documentarsi semplicemente effettuando un man sul comando mount e fstab):

- **size=2m**  
serve per limitare lo spazio su alcune partizioni ed il suo valore è espresso in MB
- **noexec**  
non permette l'esecuzione dei file binari presenti nel filesystem
- **nodev**  
Un filesystem montato con l'opzione “nodev” non consente l'utilizzo o la creazione di file speciali "di dispositivo". In genere non c'è ragione di consentire ai filesystem di adottare l'interpretazione di particolari dispositivi di blocco o di carattere, che possono costituire potenziali rischi per la sicurezza
- **nosuid**  
Impedisce le operazioni di “setuid” e “setgid”, comunemente usate per permettere agli utenti di un sistema di eseguire programmi elevando temporaneamente i privilegi e permessi dell'utente
- **Mode=1777**  
serve ad impostare i permessi di lettura/scrittura/esecuzione per tutti gli utenti
- **mode=0755**  
imposta lettura/scrittura/esecuzione per il proprietario, e lettura/esecuzione per il gruppo ed altri

Bisogna considerare che esistono alcune opzioni attive sulle partizioni di cui però non c'è traccia nella configurazione di fstab. Tramite il comando lanciato in precedenza (ossia `cat /proc/mounts`) è possibile vedere tutte le opzioni attive, comprese appunto quelle che non compaiono in fstab: confrontando il risultato di questo comando con la configurazione di fstab si può notare che sono attive, sulle partizioni sda, le seguenti opzioni: ***relatime,errors=remount-ro,user\_xattr,barrier=1,data=ordered*** (di queste opzioni solo ***errors=remount-ro*** compare in fstab).

Alcune di queste vanno sostituite con altre, altre vanno disattivate, in più è possibile aggiungerne altre. Ma vediamo nel dettaglio:

- Aggiungere **noatime** e rimuovere **relatime**  
con l'opzione “noatime” il kernel non aggiorna o scrive, nella tabella dei metadati di ogni file, le informazioni inerenti all’ora dell’ultimo accesso ai file, limitandolo di fatto a scrivere la data di accesso solo quando un file viene effettivamente modificato. Questo accorgimento può apportare notevoli aumenti di velocità e va applicato ad ogni partizione presente sulla SSD (tranne su un eventuale swap): aggiungendo questa opzione in automatico verrà rimosso “realtime” che invece scrive la data di accesso anche se il file viene solamente letto (quella del “realtime” è una pratica più sicura ma che comporta ovviamente più riscritture sul filesystem). È importante che tutte le opzioni siano attaccate l'una all'altra, separate da una virgola e che non ci siano spazi.
- Aggiungere **nouser\_xattr** e rimuovere **user\_xattr**  
con l'opzione “nouser\_xattr” vengono disabilitati gli attributi estesi (EA - Extended User Attributes) sul filesystem. Gli EA sono una serie di informazioni aggiuntive che possono essere associate ad un file. A differenza dei normali attributi dei file, gli attributi estesi possono essere di qualsiasi tipo e possono fornire informazioni aggiuntive ad una applicazione, al sistema operativo, al filesystem o all'utente. Gli attributi estesi non sono parte del file a cui vengono associati ma vengono salvati separatamente e gestiti dal filesystem. In particolare ogni attributo esteso viene identificato da una coppia (*nome*, *valore*) per permettere il reperimento delle informazioni salvate. Normalmente gli attributi estesi vanno abilitati sui server mentre sui PC di comune uso andrebbero disabilitati. Aggiungendo questa opzione in automatico verrà rimosso “user\_xattr” (che serve appunto per abilitare gli EA).
- Disattivare **barrier**  
questa opzione incrementa l'affidabilità del file system a costo di un po' di performance. Il file system, prima di scrivere la commit del journaling, deve assicurarsi che tutte le informazioni della transazione siano salvate nel journal. Quindi il file system deve esplicitamente richiedere al disco di scrivere concretamente tutti i dati del journal prima di scrivere il record di commit; se il record di commit fosse scritto per primo, il journal potrebbe risultare corrotto. Il sottosistema dell'I/O a blocchi rende possibile tutto ciò tramite l'uso delle barriere; in sostanza, una barriera impedisce la scrittura di blocchi dopo la barriera stessa fino a che tutti i blocchi scritti prima della barriera siano scritti concretamente sulla SSD. Usare le barriere permette al file system di garantirsi la consistenza della propria struttura in ogni momento. Detto questo si è capito che questa opzione serve solo nel caso in cui sia attivo il Journaling sulle partizioni.  
Anche questa opzione, attivata per default con valore a 1, dovrebbe essere quindi rimossa se

il journaling è stato rimosso. Per disattivarla è sufficiente aggiungere “barrier = 0”.

- Aggiungere **discard**  
con questa opzione viene abilitato il TRIM, opzione molto importante per il sistema operativo. Il comando TRIM permette di indicare i blocchi che non sono più in uso in una unità SSD, come ad esempio i blocchi liberati dopo l'eliminazione di uno o più file. Quando si effettua una cancellazione il Sistema Operativo contrassegna i blocchi data come “non in uso”. Il TRIM permette al Sistema Operativo di passare questa informazione al controller della SSD, il quale altrimenti non sarebbe in grado di capire quali blocchi eliminare. Il supporto nativo al TRIM è una caratteristica fondamentale sia per prolungare la vita delle unità SSD sia per ridurre la perdita di prestazioni in scrittura nel tempo. Il TRIM è stato aggiunto inizialmente al Kernel Linux versione 2.6.28, in seguito è stato pienamente supportato dalla versione 2.6.33, tuttavia nella stragrande maggioranza dei casi non è attivo di default. I vecchi SSD necessitano di un aggiornamento del firmware, altrimenti il comando verrà ignorato.
- Aggiungere **commit**  
questa opzione aumenta l'intervallo per cui il sistema fa una operazione di sincronizzazione tra i dati e i metadati: per default questo intervallo è di 5 secondi, aumentandolo ad esempio a 100, si riducono ulteriormente le scritture sulla SSD.
- Rimuovere **errors=remount-ro**

Guardando il risultato del comando “cat /proc/mounts” si può notare che l'opzione “data” è attiva ed impostata a “ordered” (gli altri valori possibili sono “writeback” e “journal”). Praticamente sono opzioni che permettono di scegliere il metodo di journaling quando si monta il filesystem. Con il modo data=writeback”, ext4 non fa nessun journaling dei dati, fornendo un journaling simile a quello trovato nei filesystem XFS, JFS e ReiserFS (solo metadati): questo potrebbe permettere che un file appena modificato si corrompa nel caso di un riavvio inaspettato o caduta di tensione. Nonostante questo svantaggio, il modo data=writeback offre la miglior prestazione in molte condizioni.

Con il metodo “data=ordered”, ext4 esegue il journaling dei metadati, ma raggruppa logicamente i blocchi di dati e di metadata in una singola unità denominata transazione. Il modo “data=ordered” effettivamente risolve i problemi di corruzione del modo “data=writeback” e degli altri metodi di journaling dei filesystem, senza richiedere il journaling completo di tutti i dati. In generale, “data=ordered” per un filesystem ext4 ha prestazioni leggermente più lenti di “data=writeback”, ma significativamente più veloce di un journaling eseguito su tutti i dati.

Il metodo “data=journal” fornisce il journaling dei dati e dei metadati: nel caso di un crash, il journal può essere rifatto, mantenendo sia i dati che i metadati in uno stato coerente. Teoricamente, il metodo “data=journal” è il più lento di tutti i metodi di journaling.

Quindi anche se l'opzione “data” è attiva e settata a “ordered” la si può ignorare perché è funzionale solo nel caso in cui sia attivo il Journaling sulle partizioni.

A questo punto è necessario fermare momentaneamente il servizio di log e cancellare il contenuto di alcune partizioni (tmp e log). Quindi in sequenza lanciare da terminale i seguenti comandi:

```
# sudo /etc/init.d/rsyslog stop
# sudo rm -rf /tmp/*
# sudo rm -rf /var/log/*
# sudo rm -rf /var/tmp/*
# sudo mount -a
# sudo /etc/init.d/rsyslog start
```

Ovviamente, con i logs in RAM, ad ogni riavvio li perderemo quindi volendo per salvarli ogni tanto è possibile farne un backup lanciando il comando:

```
# rsync -av /var/log/* ~/Salvataggi_logfiles
```

Riavviare quindi per applicare le modifiche effettuate. Dopo il riavvio per visualizzare che tutte le partizioni siano state montate correttamente lanciare questo comando:

```
# cat /proc/mounts
```

Ricapitolando ecco il contenuto del nostro fstab prima e dopo le modifiche:

#### *Prima*

```
# /etc/fstab: static file system information.
```

```
#
```

```
# <file system> <mount point> <type> <options> <dump> <pass>
```

```
proc /proc proc defaults 0 0
```

```
# /dev/sda1
```

```
UUID=190a8... / ext4 rw,errors=remount-ro 0 1
```

```
# /dev/sda2
```

```
UUID=0708c... /home ext4 rw,errors=remount-ro 0 0
```

```
# /dev/sda3
```

```
UUID=f2ccf... /Dati ext4 rw,errors=remount-ro 0 0
```

#### *Dopo*

```
# /etc/fstab: static file system information.
```

```
#
```

```
# <file system> <mount point> <type> <options> <dump> <pass>
```

```
proc /proc proc defaults 0 0
```

```
# /dev/sda1
```

```
UUID=190a8... / ext4 rw,noatime,nouser_xattr,discard,commit=100,barrier=0 0 1
```

```
# /dev/sda2
```

```
UUID=0708c... /home ext4 rw,noatime,nouser_xattr,discard,commit=100,barrier=0 0 0
```

```
# /dev/sda3
```

```
UUID=f2ccf... /Dati ext4 rw,noatime,nouser_xattr,discard,commit=100,barrier=0 0 0
```

```
#
```

```
#
```

```
tmpfs /var/tmp tmpfs defaults,noatime,mode=1777 0 0
```

```
tmpfs /var/cache/apt/archives tmpfs defaults,noatime,mode=0755 0 0
```

## Capitolo 05 - OTTIMIZZARE IL KERNEL

### 05.A - Ottimizzare le Pagine Sporche

Quando un processo scrive dei dati che devono finire su un disco, il kernel li tiene in memoria per qualche tempo prima di inviarli al dispositivo. In questo modo può combinare più scritture insieme ed evitare di accedere ripetutamente al disco. Le pagine di memoria che contengono dati non ancora trascritti su disco si chiamano pagine sporche ("dirty").

Un thread chiamato "pdflush" gira in background osservando tutte le pagine sporche e mandando alla scrittura quelle scadute, cioè quelle il cui tempo di attesa supera il valore precisato tramite l'opzione "dirty\_expire\_centisecs".

L'attivazione del thread "pdflush" avviene sia a tempo, cioè ogni tot millisecondi impostati in un'opzione chiamata "dirty\_writeback\_centisecs", sia ogni volta che il totale delle pagine sporche supera una certa soglia, che può essere un valore assoluto impostato in "dirty\_background\_bytes" oppure una percentuale impostata in "dirty\_background\_ratio" della memoria totale (in realtà la percentuale è calcolata non sulla memoria totale, ma su una combinazione tra la memoria libera, la cache e le pagine mappate dai processi; il risultato è inferiore alla memoria totale).

Quando un processo genera pagine sporche più velocemente di quanto "pdflush" riesca a trascriverle su disco, questo meccanismo non è più sufficiente. In questi casi, il processo incriminato è obbligato a bloccarsi e iniziare a fare lui il lavoro di "pdflush", scrivendo pagine sporche su disco. Questo avviene ogni volta che il totale delle pagine sporche supera una seconda soglia, anch'essa espressa in valore assoluto tramite "dirty\_bytes", o come percentuale tramite "dirty\_ratio" della memoria totale.

In pratica:

- quando le pagine sporche sono poche, tra zero ed il valore di "dirty\_background\_bytes", la gestione è passiva: le pagine rimangono in memoria fino a che non scadono e un thread in background scrive su disco quelle scadute
- quando sono un po' di più, tra il valore di "dirty\_background\_bytes" ed il valore di "dirty\_bytes", lo stesso thread in background inizia a scrivere su disco anche quelle non ancora scadute, con l'obiettivo di ridurle
- quando sono decisamente troppe, cioè oltre il valore di "dirty\_bytes", anche i processi attivi intervengono a scriverle su disco

Sui portatili di solito si aggiunge un ulteriore meccanismo settato con una opzione chiamata "laptop mode". In pratica, ogni volta che il kernel risveglia un disco per leggere o scrivere dati, il thread "pdflush" interviene entro il tempo settato in "laptop\_mode" ed approfitta dello stato di veglia del disco per scriverci tutte le pagine sporche ancora in memoria, anche se non scadute. In questo modo si spera che il disco possa essere rimesso a dormire senza che debba essere risvegliato da lì a poco da "pdflush". Se "laptop\_mode" è settato con valore zero, questo meccanismo è disabilitato.

Tutti i parametri citati si trovano sotto /proc/sys/vm/. Per vedere i loro valori è sufficiente effettuare un cat da terminale:

```
# cat /proc/sys/vm/dirty_expire_centisecs
# cat /proc/sys/vm/dirty_writeback_centisecs
# cat /proc/sys/vm/dirty_background_bytes
# cat /proc/sys/vm/dirty_background_ratio
# cat /proc/sys/vm/dirty_bytes
# cat /proc/sys/vm/dirty_ratio
# cat /proc/sys/vm/laptop_mode
```

praticamente i valori di default sono questi:

```
dirty_expire_centisecs = 3000
dirty_writeback_centisecs = 500
dirty_background_bytes = 0
dirty_background_ratio = 5
dirty_bytes = 0
dirty_ratio = 10
laptop_mode = 0
```

Per ottenere quindi una ulteriore diminuzione dell'accesso alla SSD è possibile modificare questi parametri aumentando i loro valori. Per modificarli è possibile farlo direttamente sui file stessi con un editor qualsiasi oppure tramite il comando echo (ad edempio: `echo 1500 > /proc/sys/vm/dirty_writeback_centisecs`): i nuovi valori vengono subito applicati ma al successivo riavvio vengono ripristinati i valori di default.

Per modificarli in maniera definitiva è necessario aprire il file `/etc/sysctl.conf` ed aggiungere le variabili alla fine del file. Quindi:

```
# sudo vi /etc/sysctl.conf
```

ed aggiungere alla fine del file:

```
vm.dirty_writeback_centisecs = 1500
```

```
vm.dirty_ratio = 20
```

```
vm.dirty_background_ratio = 10
```

nel caso di un portatile aggiungere anche

```
vm.laptop_mode = 5
```

### **05.B - Ottimizzare lo Swappiness / Sospensione / Ibernazione**

Come già detto in precedenza, con le SSD è sconsigliato creare una partizione di Swap: se proprio volete farlo è opportuno ottimizzare il kernel in modo tale da diminuire il più possibile il suo utilizzo per poter salvaguardare la SSD stessa. Inoltre, è consigliato disabilitare le features di sospensione/ibernazione soprattutto nel caso in cui NON sia presente una partizione di Swap, perché queste due features utilizzano proprio questa partizione per memorizzare i dati, anche se in realtà sarebbe possibile configurare il sistema per utilizzare la sospensione/ibernazione SENZA una partizione di Swap...

Per modificare la frequenza di utilizzo dello Swap è necessario modificare il valore di swappiness: dando alla swappiness un valore pari a zero significa che la scrittura sulla SSD sarà evitata a meno che non sia assolutamente necessaria, mentre con uno swappiness impostato a 100 significa che i dati saranno scritti quasi interamente sulla partizione di Swap.

Di default quasi tutte le distribuzioni hanno 60 come valore predefinito di swappiness, il che significa che lo Swap verrà utilizzato nel caso in cui venga occupata circa la metà della memoria RAM.

E' possibile controllare il valore della swappiness tramite il comando:

```
# cat /proc/sys/vm/swappiness
```

Nel nostro caso, dato che durante l'installazione non è stata configurata una partizione di Swap e le features di sospensione/ibernazione sono state disabilite, direi di impostare questo valore a 0.

Aprire ed editare il file sysctl.conf:

```
# vi /etc/sysctl.conf
```

aggiungere alla fine del file

```
vm.swappiness = 0
```

La modifica si rende attiva riavviando il sistema.

Se invece avete creato e configurato una partizione di Swap e non volete rinunciare alla sospensione/ibernazione, sarebbe opportuno impostare il valore di swappiness a 1, in modo tale da ridurre drasticamente l'uso dello Swap e per fare in modo che venga utilizzata quasi ed esclusivamente per l'ibernazione e/o sospensione del PC.

## Capitolo 06 - OTTIMIZZARE IL BROWSER (FIREFOX / CHROMIUM)

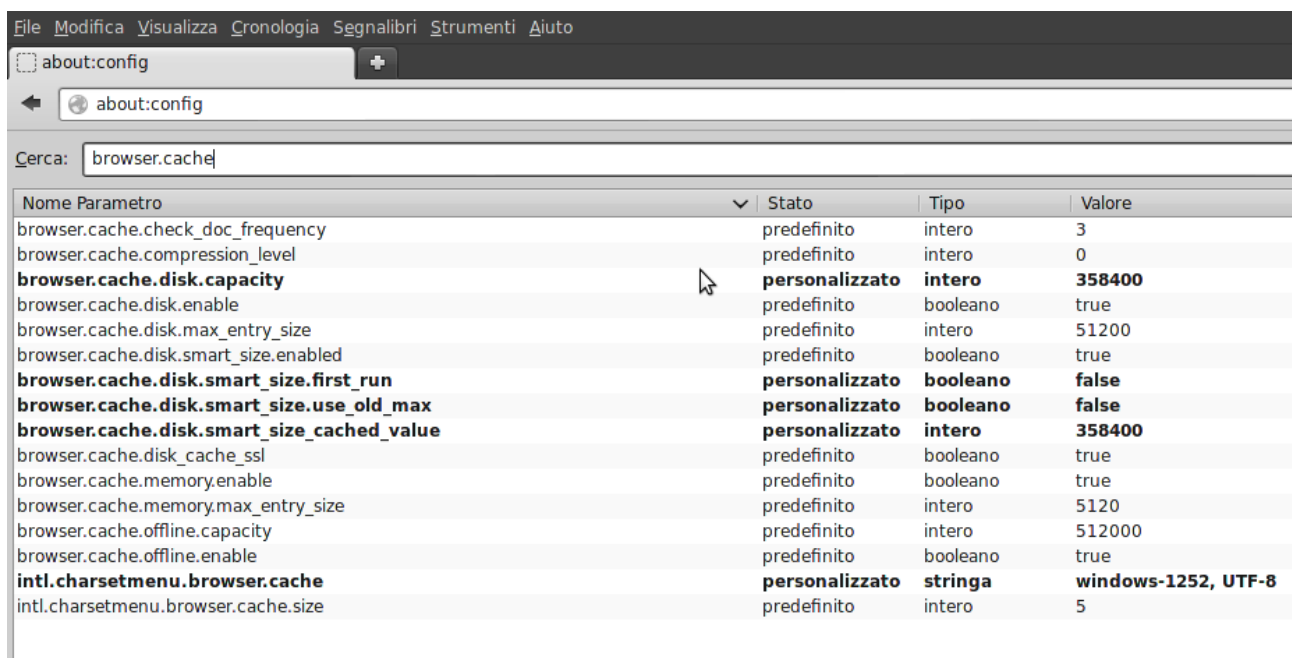
Possedendo almeno 4GB di RAM, che sono più che sufficienti soprattutto per un sistema Linux, è possibile spostare su di essa anche la cache del browser. In questo modo si sfrutta quello che difficilmente si utilizzerebbe ma soprattutto si evitano molte scritture sulla SSD, cosa che purtroppo accade anche in maniera eccessiva durante la navigazione Web.

Spostare la cache del browser su RAM velocizza inoltre anche il caricamento delle pagine però comporta alcune problematiche. La più significativa riguarda l'impossibilità di effettuare un ripristino della sessione del browser nel caso in cui quest'ultimo si chiuda in seguito ad un problema.

### 06.A - Firefox

Aprire Firefox e digitare sulla barra degli indirizzi “about:config” (senza i doppi apici); verrà visualizzato un avviso riguardante una eventuale compromissione della stabilità, della sicurezza e delle prestazioni: cliccare sul tasto “Farò attenzione, prometto” (!?) per proseguire...

Scrivere “browser.cache” nel campo del filtro e premere invio per effettuare la ricerca:



The screenshot shows the Firefox 'about:config' page with a search filter 'browser.cache' applied. The results are displayed in a table with columns: Nome Parametro, Stato, Tipo, and Valore. The table lists various cache-related parameters, with 'browser.cache.disk.capacity' highlighted as 'personalizzato' (customized) with a value of 358400.

Nome Parametro	Stato	Tipo	Valore
browser.cache.check_doc_frequency	predefinito	intero	3
browser.cache.compression_level	predefinito	intero	0
<b>browser.cache.disk.capacity</b>	<b>personalizzato</b>	<b>intero</b>	<b>358400</b>
browser.cache.disk.enable	predefinito	booleano	true
browser.cache.disk.max_entry_size	predefinito	intero	51200
browser.cache.disk.smart_size.enabled	predefinito	booleano	true
<b>browser.cache.disk.smart_size.first_run</b>	<b>personalizzato</b>	<b>booleano</b>	<b>false</b>
<b>browser.cache.disk.smart_size.use_old_max</b>	<b>personalizzato</b>	<b>booleano</b>	<b>false</b>
<b>browser.cache.disk.smart_size_cached_value</b>	<b>personalizzato</b>	<b>intero</b>	<b>358400</b>
browser.cache.disk_cache_ssl	predefinito	booleano	true
browser.cache.memory.enable	predefinito	booleano	true
browser.cache.memory.max_entry_size	predefinito	intero	5120
browser.cache.offline.capacity	predefinito	intero	512000
browser.cache.offline.enable	predefinito	booleano	true
<b>intl.charsetmenu.browser.cache</b>	<b>personalizzato</b>	<b>stringa</b>	<b>windows-1252, UTF-8</b>
intl.charsetmenu.browser.cache.size	predefinito	intero	5

Effettuare le seguenti modifiche:

- modificare la voce **browser.cache.disk.enable** impostandola a “false” (basta effettuare un semplice doppio click sulla voce stessa)
- modificare la voce **browser.cache.memory.enable** impostandola a “true” (potrebbe già essere ipostata a “true” a seconda della versione di Firefox)
- aggiungere una nuova voce chiamata “browser.cache.memory.capacity” di tipo intero cliccando con il tasto destro del mouse sullo spazio bianco e selezionando “Nuovo” quindi



“Intero”: nel valore inserire -1 se volete lasciare a Firefox il compito di allocare dinamicamente lo spazio in base alla RAM posseduta oppure inserire un valore espresso in Kbyte.

Chiudere la finestra delle impostazioni quindi cliccare sul menù Modifica ==> Preferenze ==> Avanzate ==> Rete e selezionare l'opzione “Non utilizzare la gestione automatica della cache” e nella voce “Limita la cache a” impostare il valore 0, quindi cliccare sul tasto “Cancella adesso”. Per finire riavviare Firefox.

## **06.B - Chromium**

Per questo browser spostare la cache in RAM è decisamente più veloce e semplice, infatti è sufficiente modificare il comando “Exec” nel suo lanciatore. Aprire il file chromium.desktop con l'editor preferito (nel mio caso è mousepad):

**# mousepad /usr/share/applications/chromium.desktop**

quindi modificare la voce

***Exec=/usr/bin/chromium %U***

in

***Exec=/usr/bin/chromium %U --disk-cache-dir="/var/tmp"***

Con questa modifica viene detto a Chromium di usare, per la cache, la partizione /var/tmp che abbiamo spostato nella RAM (nel capito 4). E' ovvio che se Chromium viene lanciato non dal lanciatore ma ad esempio da terminale, bisogna ricordarsi di precisare le opzioni dopo il comando (chromium %U --disk-cache-dir="/var/tmp").