Viaggio nei file system Linux di ultima generazione

Alla scoperta di nuove funzionalità dei file system, rispolverando alcune vecchie perle nascoste.

Motivazioni:

- curiosità, voglia di imparare, di ”smontare”
- gusto di ”sporcarsi le mani” con concetti e operazioni a basso livello

I file system sono uno dei fondamenti dei sistemi Linux/Unix

- ogni piccola scoperta/invenzione può innescare una cascata di novità nei livelli superiori

Viaggio nei file system

Obiettivi:

- appagare la curiosità e la voglia di imparare
- rispolverare le funzionalità classiche, riscoprendo alcune perle nascoste
- scoprire modi innovativi di combinare le funzionalità vecchie e nuove
- avere qualche strumento (potente) in più per risolvere i problemi

Viaggio nei file system

C'erano una volta i file systems Unix

- inode, blocchi, file sparsi...

Poi sono arrivate le funzionalità ”enterprise”

- loop devices
- file preallocati
- extents al posto dei blocchi
- journaling, device mapper, LVM

Andiamo a (ri)scoprirne qualcuna

File sparsi

Esistono da decenni.

In un file sparso alcuni blocchi NON sono allocati

- per convenzione, i blocchi non allocati contengono zeri
- permettono di risparmiare spazio
- permettono di creare rapidamente file enormi

Uso tipico

- immagini su file di CD, DVD o intere partizioni
  - assieme ai loop device

Chi ha mai usato un file sparso?

File sparsi

Domande

1. Si può creare un file sparso più grande dello spazio libero disponibile?
2. E più grande della partizione che lo contiene?
3. Cosa succede se, scrivendo su un file sparso, si esaurisce lo spazio nel file system che lo contiene?

File sparsi

Domande

1. Si può creare un file sparso più grande dello spazio libero disponibile?
   SI
2. E più grande della partizione che lo contiene?
3. Cosa succede se, scrivendo su un file sparso,
   si esaurisce lo spazio nel file system che lo contiene?

File sparsi

Domande

1. Si può creare un file sparso più grande dello spazio libero disponibile?
   SI
2. E più grande della partizione che lo contiene?
   SI
3. Cosa succede se, scrivendo su un file sparso,
   si esaurisce lo spazio nel file system che lo contiene?

File sparsi

Domande

1. Si può creare un file sparso più grande dello spazio libero disponibile?
   SI
2. E più grande della partizione che lo contiene?
   SI
3. Cosa succede se, scrivendo su un file sparso,
   si esaurisce lo spazio nel file system che lo contiene?
   La scrittura dà errore ENOSPC
   ”No space left on device”

Loop devices

Un loop device è un dispositivo a blocchi virtuale

- permette di usare un file (anche sparso) come se fosse un dispositivo a blocchi.

Il comando per gestire i loop device è **losetup**, e una volta che si ha un loop device si può:

- formattarlo con un qualunque file system
- montarlo e usarlo come una partizione
- eccetera…

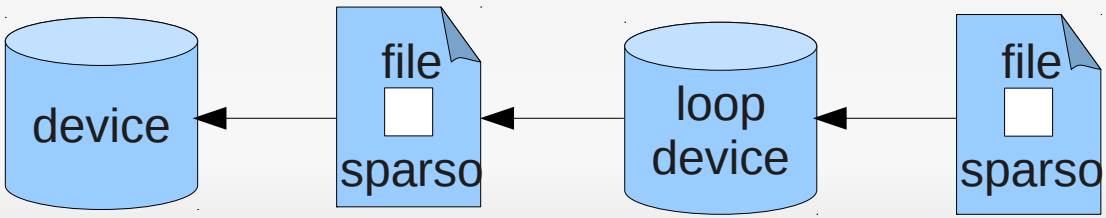
Comodo per usare le immagini di CD e DVD senza dover masterizzare niente.

Loop devices

Domande

1. si può creare un file sparso dentro un loop device che usa a sua volta un file sparso?

2. se sì, quante volte si può nidificare questo meccanismo?

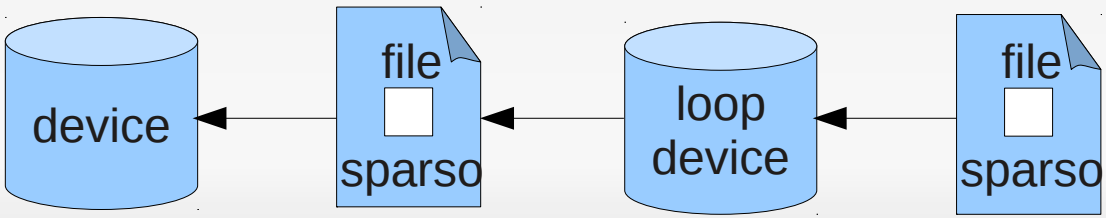


Loop devices

Domande

1. si può creare un file sparso dentro un loop device che usa a sua volta un file sparso?
   SI

2. se sì, quante volte si può nidificare questo meccanismo?

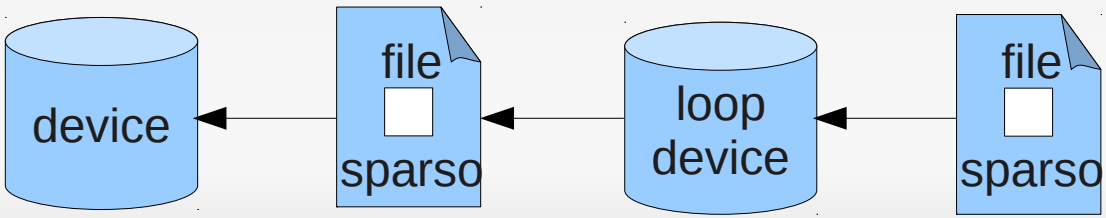


Loop devices

Domande

1. si può creare un file sparso dentro un loop device che usa a sua volta un file sparso?
   SI

2. se sì, quante volte si può nidificare questo meccanismo?
   L'unico limite è il numero di loop device (configurabile al caricamento del modulo)



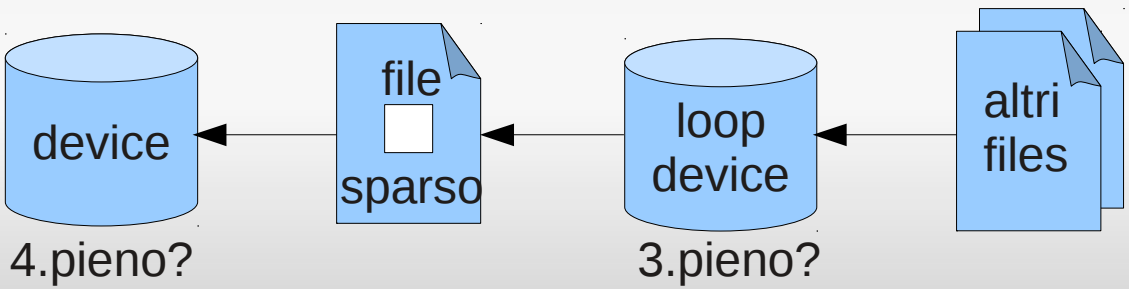
Loop devices

Supponiamo di creare un file sparso,
usarlo come loop device, formattarlo e montarlo.

Cosa succede se, scrivendoci dentro:

3. si esaurisce lo spazio nel loop device?

4. si esaurisce lo spazio nel device che contiene
   il file sparso?



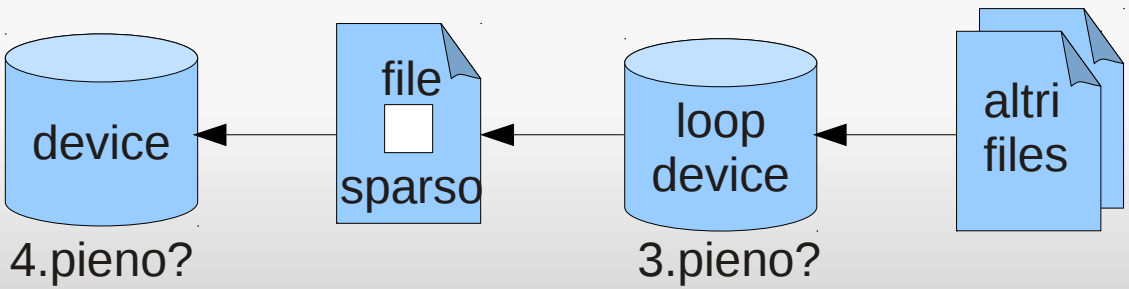
Loop devices

Supponiamo di creare un file sparso,
usarlo come loop device, formattarlo e montarlo.

Cosa succede se, scrivendoci dentro:

3. si esaurisce lo spazio nel loop device?
   La scrittura dà errore ENOSPC

4. si esaurisce lo spazio nel device che contiene
   il file sparso?



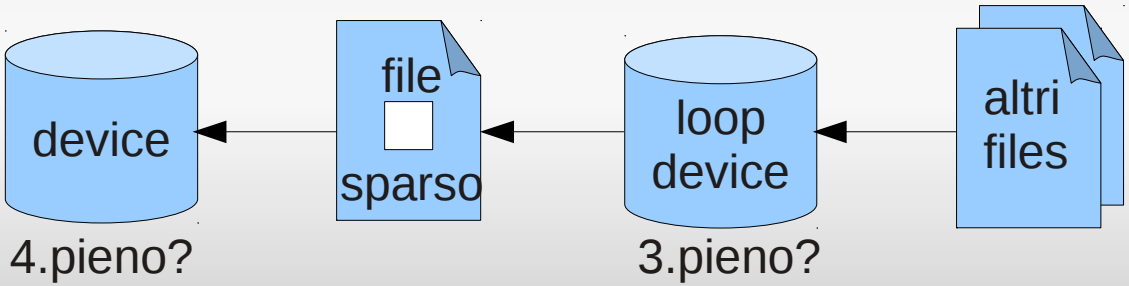
Loop devices

Supponiamo di creare un file sparso,
usarlo come loop device, formattarlo e montarlo.

Cosa succede se, scrivendoci dentro:

3. si esaurisce lo spazio nel loop device?
   La scrittura dà errore ENOSPC

4. si esaurisce lo spazio nel device che contiene
   il file sparso?
   Si PERDONO i dati!! Nessun errore segnalato,
   solo i log del kernel (dmesg) mostrano il guaio



Preallocazione file

2001: posix_fallocate() standard POSIX

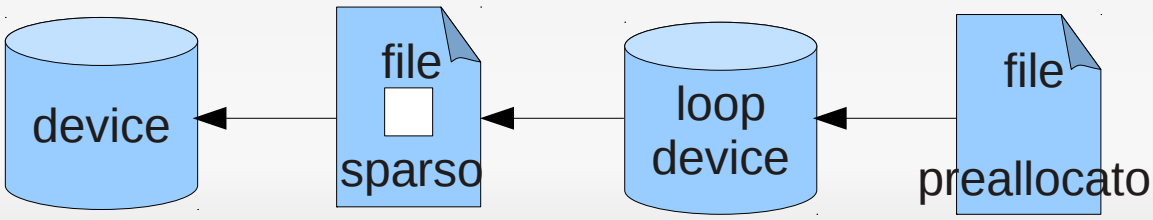
2010: fallocate() compare nel kernel 2.6.36
ad oggi solo per: btrfs, ext4, gfs, xfs

- chiede al file system di allocare i blocchi per un certo intervallo di un file
- il modo più rapido di creare un file NON sparso: NON scrive nei blocchi
- garantisce che scrivendo successivamente in quell'intervallo non si avranno errori ENOSPC

Preallocazione file

Domande

1. Se si crea un file preallocato dentro un loop device, a sua volta dentro un file sparso, occupa spazio o no?

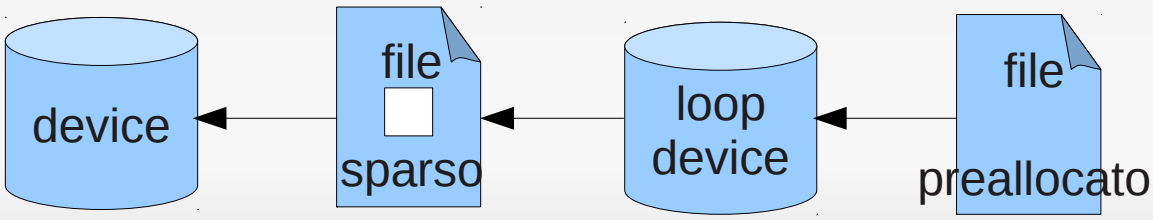


Preallocazione file

Domande

1. Se si crea un file preallocato dentro un loop device, a sua volta dentro un file sparso, occupa spazio o no?

   Occupa spazio dentro il loop device



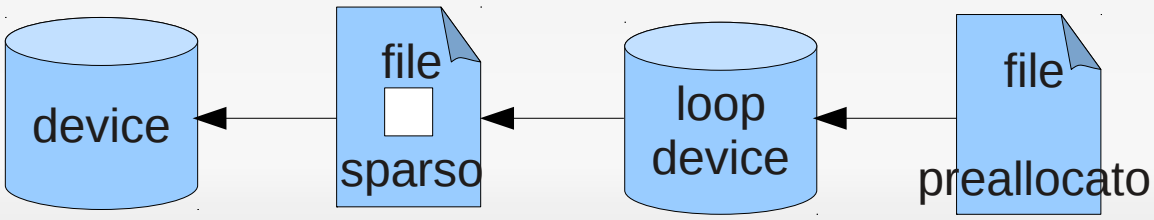
Preallocazione file

Domande

1. Se si crea un file preallocato dentro un loop device, a sua volta dentro un file sparso, occupa spazio o no?

   Occupa spazio dentro il loop device

   NON fa aumentare lo spazio occupato dal file sparso



Supporto per SSD

Alcuni file system sono ottimizzati per le memorie FLASH, es. SDD e chiavette USB

- NILFS2 – usa un device come unico journal circolare
  - scrive quasi sempre sequenzialmente
  - snapshots e checkpoints
  - garbage collecting per deallocare i blocchi obsoleti
- Btrfs – B-trees + copy-on-write
  - troppo da dire, servirebbe un talk dedicato...

Altri supportano solo le memorie NAND raw

- JFFS2, UBIFS, YAFFS

Supporto per SSD

Anche se si usano i normali file system, SSD e memorie USB funzionano meglio – e più a lungo – con alcuni accorgimenti

- disabilitare il journaling
- non usarle per /tmp, /var/tmp, /var/log e simili
- usare l'opzione di mount ”noatime” o ”relatime” di ext2/3/4
- lasciare un po' di spazio non partizionato
- per gli SSD, usare ”fstrim” - dice all'SSD di deallocare i blocchi corrispondenti allo spazio libero del file system

Conversione tra file system

2005: convertfs, abbandonato nel 2006

2008: btrfs-convert, solo ext3 → btrfs

2011: fstransform (Io ammetto, voglio farmi pubblicità)

- Trasformano un file system in un altro, mantenendo i dati e senza bisogno di backup
- RISCHIOSI: sono beta, se qualcosa va storto fareste bene ad avere un backup
- Possono essere di due tipi:
  - specifici per un file system, es. btrfs-convert
  - generali, es. convertfs e fstransform

Come funzionano? Lo vediamo tra poco...

Altre funzionalità

Extent: raggruppamento di blocchi consecutivi

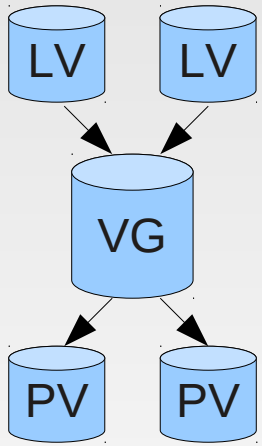
- risparmia spazio per le bitmap di allocazione dei blocchi
- più efficiente nella gestione dei file contigui

Journaling

- stranoto, non ha bisogno di parlarne ancora

Device Mapper

- innumerevoli possibilità, es. Cifratura



LVM, basato su Device Mapper

- unire più device (PV) per crearne uno più grande (VG)
- risuddividerlo in device (LV) anche a caldo
- aggiungere o togliere device (PV) a caldo...

Cosa possiamo farci?

Sono mattoni di base, i soli limiti sono la nostra capacità e fantasia

Burj Khalifa

Esempio: fstransform

Trasformare un file system in un altro, senza backup e preservando tutto il contenuto.

Come funziona?

- creazione di un file sparso grande quanto il device che lo contiene
- formattazione con il nuovo file system
- montato tramite loop device
- spostamento di tutti i file e directory uno alla volta
- spostamento dei blocchi del file sparso coincidere con il device originale (rimappatura)

Esempio: fstransform

Lo spostamento di tutti i file e directory uno alla volta è lento e rischioso, se il device originale si riempie sono guai: **si PERDONO i dati!**

Si può fare di meglio?

Anziché spostarli dentro il loop device, si possono preallocare! Attenzione ai file system con tail merging

La rimappatura diventa più complessa, ma possibile

Nella prossima versione (0.9.4)