opensolaris™

Zettabyte FileSystem

Antonio Prado
ICT Mentor

Sommario

- ZFS: cos'e' | SI prefixes
- FS tradizionali e limiti
- Storage pool VS Volume
- Data Integrity
- RAID
- Raid-Z
- Scalabilita'
- Performance
- Amministrazione
- Hints

ZFS: cos'e'

Zettabyte FileSystem sviluppato da SUN per Solaris Operating System

- 128-bit
- Storage pool
- Affidabilita'
- Scalabilita'

SI prefixes

Ki 2^10	K 10^3
Mi 2^20	M 10^6
Gi 2^30	G 10^9
Ti 2^40	T 1000^4
Pi 2^50	P 1000^5
Ei 2^60	E 1000^6
Zi 2^70	Z 1000^7
Yi 2^80	Y 1000^8

FS tradizionali e limiti

Noisy data corruption:

chiedo di fare, risponde non posso

Silent data corruption:

chiedo di fare, non fa e non risponde

Capacita’

non è in grado di aumentare con facilita’

Amministrazione

operazioni complicate

Storage pool Vs Volume 1/3

storage considerato come insieme:

si elimina la nozione di volumi e l'amministrare i dischi singolarmente.

Pool sta allo storage come la memoria virtuale sta alla memoria.

il filesystem si relaziona piu' con l'insieme di storage piuttosto che con i singoli dischi.

transactional operation mantiene sempre i dati consistenti sul disco, rimuove quasi tutti i problemi relativi all'I/O e permette di ottenere grandi prestazioni

Storage pool Vs Volume 2/3

VOLUME

ogni filesystem si occupa di un singolo disco

la necessita' di piu' spazio, velocita' e affidabilita' da una parte richiede un redesign del filesystem, dall'altra permette di utilizzare i volumi per raggruppare i dischi insieme.

una fiorente industria si e’ sviluppata attorno a queste caratteristiche:

programmi per gestire filesystems e volumi vengono venduti come prodotti a se' stanti

Storage pool Vs Volume 3/3

VOLUME

- l'astrazione viene gestita con dischi virtuali
- partizioni e volumi per ogni filesystem
- ingrandire e diminuire i volumi a mano
- limitazione della velocita' di ogni file system
- lo storage e' frammentato

ZFS

- l'astrazione viene gestita con l'allocazione di memoria
- non ci sono partizioni da gestire
- ingrandire e diminuire automaticamente
- e' sempre disponibile tutta la velocita'
- tutto lo storage nel pool e' condiviso

Data Integrity 1/4

i dati non vengono mai sovrascritti in tempo reale

lo stato sul disco e' sempre valido, non ci sono finestre di vulnerabilita'

non c'e' bisogno di fsck

tutto si basa su transazioni

i cambiamenti collegati o vanno a buon fine tutti oppure niente

non c'e' bisogno del journaling

per ogni dato viene fatto il checksum

dunque non c'e' la perdita di dati silenziosa

Data Integrity 2/4

transazioni copy-on-write

snapshot costanti: alla fine di ogni gruppo di operazioni viene fatto uno snapshot poiche' costa meno fare uno snapshot che non farlo

tradizionalmente il checksum viene archiviato con i blocchi di dati

zfs non archivia il checksum assieme ai dati ma nel parent block pointer. questo per avere una maggiore ridondanza e dunque differenziazione tra checksum e dati.

Data Integrity 3/4

mirroring

tradizionalmente il mirroring funziona cosi':

l'applicazione chiede una lettura; il mirror legge il primo disco che ad esempio ha un blocco corrotto e quindi non puo' rispondere all'ordine di lettura

il gestore di volume passa il blocco corrotto al filesystem. se e' un blocco di metadata il filesystem va in crash, altrimenti il filesystem restituisce il dato corrotto all'applicazione

Data Integrity 4/4

mirroring

in zfs il dato si ripara autonomamente

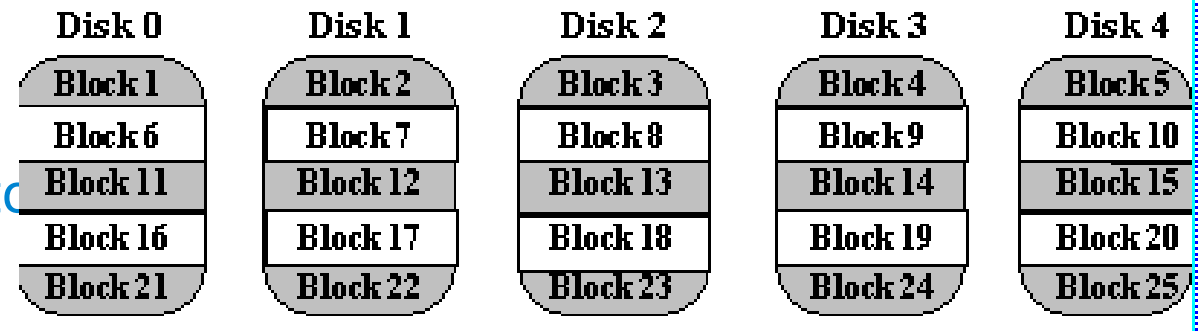
l'applicazione chiede una lettura, il mirror zfs prova ad usare il primo disco. il checksum rivela che il blocco sul disco e' corrotto.

allora zfs prova il secondo disco. il checksum dice che il blocco e' ok.

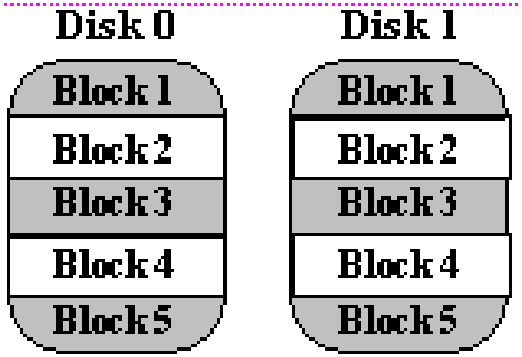
zfs restituisce il dato corretto all'applicazione e ripara il blocco danneggiato.

RAID redundant array of inexpensive disks 1/2

Raid 0 (non ridondato):

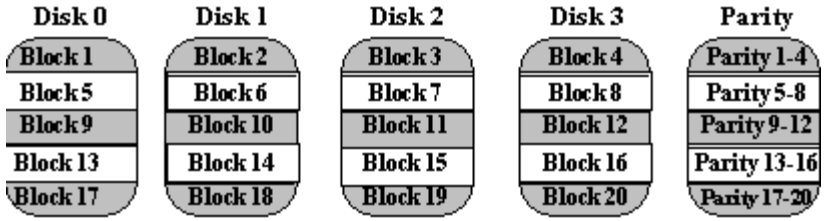


Raid 1 (mirror):

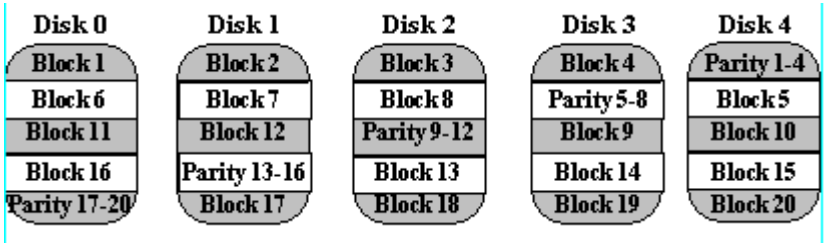


RAID redundant array of inexpensive disks 2/2

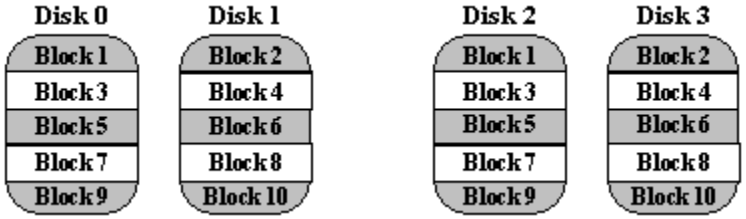
Raid 4:



Raid 5:



Raid 10:



RAID-Z 1/3

raid 4 e raid 5 tradizionali

molti dischi per dati e un disco per il controllo di parita'.

l'aggiornamento della parita' richiede un meccanismo di leggi-modifica-scrivi lento: leggi il vecchio dato e la vecchia parita' (due letture sincrone sul disco), calcola la nuova parita', scrivi il nuovo dato e la nuova parita'.

il punto debole e' un buco in scrittura: la perdita di corrente tra la scrittura del dato e quella della parita' corrompe i dati

incapace di accorgersi di silent data corruption e correggerla.

RAID-Z 2/3

ZFS

3 settori logici corrispondono a 3 blocchi di dati e 1 blocco di parita‘

tutte le scritture sono full-stripe: questo elimina il meccaniscmo leggi-modifica-scrivi e dunque e' veloce, elimina il buco di scrittura del raid5 quindi non c'e' bisogno di nvram.

si accorge e corregge la corruzione silenziosa dei dati.

non ha bisogno di hardware speciale, zfs ama i dischi economici.

RAID-Z 3/3

disk scrubbing

trova errori mentre sono ancora correggibili, verifica l'integrita' di tutti i dati: passa in rassegna i metadata del pool per leggere ciascuna copia di ogni blocco, verifica il checksum di ciascuna copia, nel frattempo si autoripara.

fornisce una riparazione veloce e affidabile.

mentre quella tradizionale prevede una copia dell'intero disco senza fare un controllo di correttezza, zfs copia in tempo reale e controlla il checksum di ogni cosa

Scalabilita’

la capacita' e' immensa, si tratta di un sistema a 128 bit

i metadati sono dinamici al 100%: non ci sono limitazioni sul numero di files, sul numero di directory.

Ogni cosa puo' essere eseguita in modo parallelo: lettura/scrittura, operazioni sulle directory ecc.

Performance 1/2

il design copy on write, fa si' che le scritture random diventino scritture in sequenza.

la grandezza dei blocchi non e' definita, ma viene scelta automaticamente per venire incontro al carico di lavoro.

Performance 2/2

Lo striping dinamico distribuisce il carico su tutti i dispositivi.

Le scritture avvengono su tutti i mirror, le letture ovunque il dato sia stato scritto. Il criterio di allocazione dei blocchi tiene in considerazione:

la capacita'; la velocita' (latenza e banda); lo stato di salute (degradato o no)

nel caso venga aggiunto un ulteriore mirror: le scritture vengono fatte su tutte i mirror, le letture ovunque il dato sia stato scritto, non c'e' bisogno di migrare i dati esistenti poiche': i vecchi dati sono scritti sui vecchi mirror, i nuovi su tutti i mirror e il meccaniscmo di copy on write si occupa di riallocare tutti i vecchi dati.

Amministrazione 1/2

grazie al pooled storage e al fattp che non si usino piu' i volumi, lo storage viene condiviso, non c'e' spazio che va perso, ne' c'e' spreco di banda

il filesystem e' gerarchico con proprieta' ereditabili.

i filesystems diventano dei punti di controllo amministrativo per l'applicazione di criteri relativi agli snapshot, alla compressione ai backup ai privilegi.

Amministrazione 2/2

e' possibile gestire filesystems logicamente legati come un gruppo

la possibilita' di ereditare i criteri fa si' che l'amministrazione su larga scala venga fatta in un attimo.

tutto viene fatto live, senza bisogno di mettere offline nulla.

Hints

zpool create miopool mirror c0t0d0 c1t0d0

zpool status

zpool list

zfs create pool1/fs.001

zfs set quota=24g pool1/fs.001

zpool add pool1 mirror c2t1d0 c4t1d0

zfs set reservation=10m miopool/filsystem1

zfs get reservation miopool/filsystem1

zfs set sharenfs=on miopool/filsystem1

zfs snapshot miopool/filsystem1@oggi

zfs rollback miopool/filsystem1@oggi

zfs destroy miopool/filsystem1@oggi

zfs set compression=on miopool/filsystem1

zfs get compression miopool/filsystem1

zfs clone miopool/filsystem1@oggi miopool/clone

zpool export miopool

zpool import miopool

zpool replace miopool c2t0d0 c3t0d0

zpool scrub miopool

zfs set sharenfs=on miopool/filsystem1