CVserver

Un cluster di VServer

Micky Del Favero
micky@linux.it

BLUG - Belluno Linux User Group
Linux Day 2008 - Feltre 25 ottobre 2008

Motivazione

Legge di Murphy:
Se qualcosa può andare storto allora lo farà.

È necessario ridondare i servizi essenziali, al fine di ridurre al minimo la possibilità che un problema possa rallentare, danneggiare o bloccare le attività in atto.

Motivazione

Corollario:
Ogni soluzione genera nuovi problemi.

È necessario introdurre il minino numero possibile di modifiche al sistema da rendere affidabile per evitare di introdurre possibili nuovi problemi.

CVserver

CVserver è una soluzione che permette di costruire un cluster in alta affidabilità senza la necessità di apportare modifiche né di richiedere configurazioni diverse rispetto ad un sistema non in cluster.

CVserver è costituito da DRBD (volume), OCFS2 (filesystem), Linux-VServer (servizi) e heartbeat (gestione alta affidabilità).

Componenti fondamentali

- **DRBD**
  Un volume di memorizzazione distribuito per cluster basato su kernel Linux.
- **OCFS2**
  Un filesystem POSIX per volumi condivisi in un cluster ad alte prestazione e alta affidabilità.
- **Linux-VServer**
  Un sistema di virtualizzazione basato sui livelli di isolamento del kernel che garantiscono i *VPS* abbiano la necessaria sicurezza utilizzando nel contempo efficientemente le risorse disponibili.
- **heartbeat**
  Un sistema di gestione di un cluster che garantisce l'affidabiltà globale del sistema.

DRBD

Distributed Replicated Block Device è un sistema di archiviazione distribuito consistente in un modulo del kernel che si occupa di gestire oltre a tutto il flusso I/O attraverso il dispositivo (`/dev/drbdx`) da e verso i volumi locali ai nodi anche la comunicazione fra i nodi del cluster per garantire il sistema sia mantenuto sincrono e per gestire eventuali anomalie. È corredato da alcune applicazioni in spazio utente per gestire la configurazione e gestione del dispositivo.
Semplificando può essere visto come un volume RAID 1 distribuito su più macchine, il vantaggio rispetto ad una risorsa condivisa è l'assenza di SPOF nel sistema, garantendo quindi la sicurezza dei dati.

OCFS2

Oracle Cluster File System 2 è un filesystem *POSIX-compliant* adatto a volumi condivisi ad alte prestazioni e alta affidabilità.
Fornisce la semantica di un filesystem locale, applicazioni scritte appositamente per girare su un cluster possono beneficiare della possibilità di I/O parallelo su più nodi per incrementare le prestazioni, le altre applicazioni comunque beneficiano della possibilità di fail-over incrementando l'affidabilità del sistema.
Le caratteristiche più interessanti sono: dimensione dei blocchi variabile, allocazione flessibile, journaling, neutralità verso l'endianess, lock manager distribuito e supporto per I/O bufferizzato, diretto, asincrono, via splice e memory mapped.

Linux-VServer

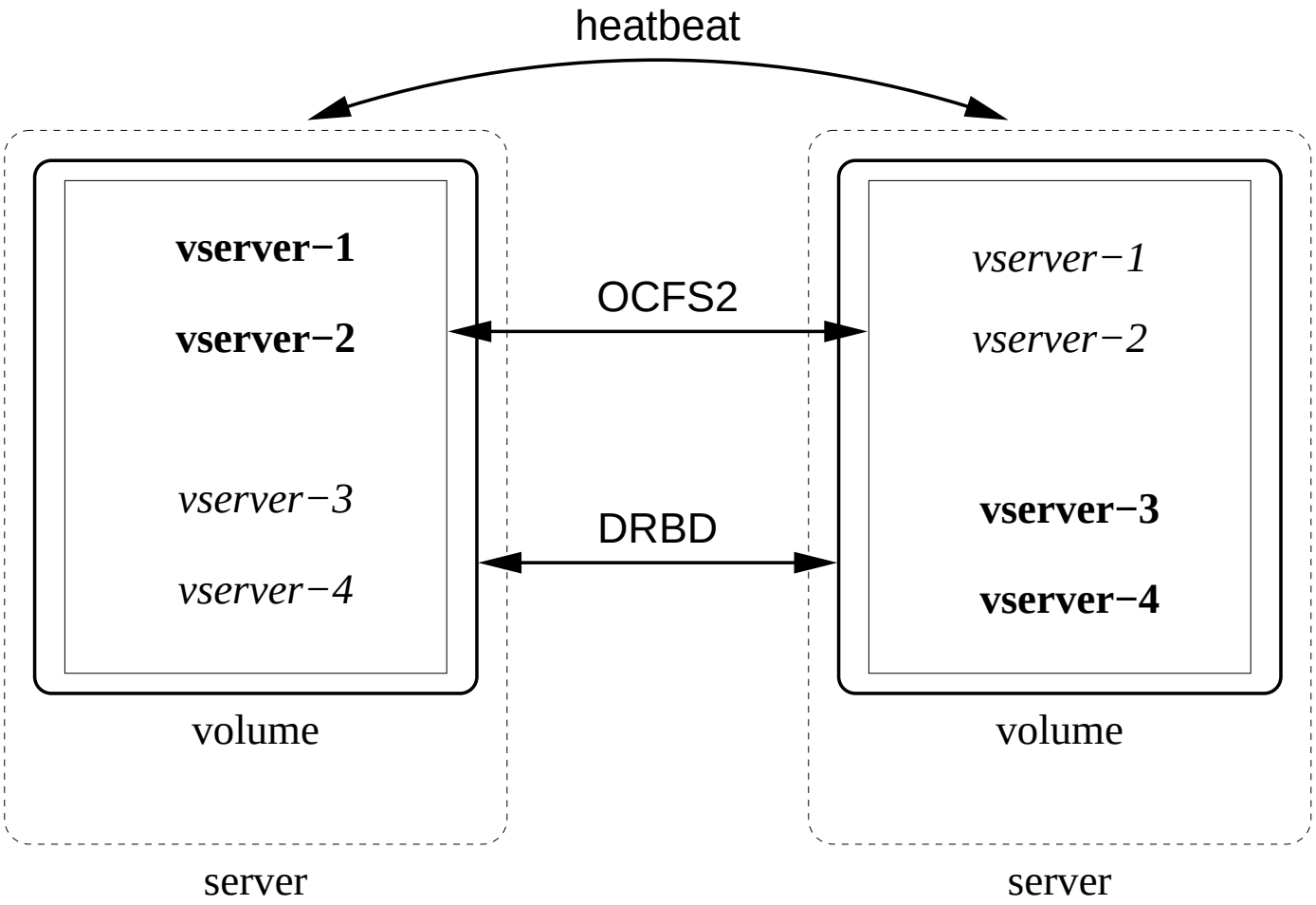
Linux-VServer è un'implementazione di *VPS* ottenuta aggiungendo le *capabilities* necessarie alla virtualizzazione al kernel di Linux.
È un sistema di *chroot* capace di partizionare le risorse della macchina (filesystem, tempo di CPU, indirizzi di rete e memoria) in modo tale che non sia possibile da un contesto inficiare gli altri. Caratteristica peculiare di Linux-VServer è il kernel condiviso e di conseguenza l'assenza di modifiche da apportare al codice di nessun processo debba essere lanciato in quanto le chiamate a sistema non vengono modificate.

heartbeat

heartbeat è un sistema di comando e controllo per sistemi in alta affidabilità. Su ogni nodo componente il cluster un processo heartbeat emette un segnale che viene ascoltato dai suoi omologhi sugli altri nodi, nel caso il segnale sparisca heartbeat esegue i comandi necessari a rispristinare i servizi che erano serviti dal nodo guasto sul nodo su cui gira.

Visione d'insieme

CVserver



Nucleo di comando e controllo

Il nucleo che garantisce l'affidabilità del sistema è uno script, lanciato da heartbeat, che controlla il montaggio del filesystem sui nodi e il lancio dei VServer che forniscono i servizi, può essere così schematizzato:

```
mpe=$(mount | grep ${mountpoint})
if [ ! "${mpe}" ] ; then
  ${logger} Mount ${drbddev} to ${mountpoint}
  mount -t ${fs} ${drbddev} ${mountpoint}
fi
/etc/init.d/util-vserver start
for vs in $(cat ${vslist}) ; do
  ${logger} Starting vserver ${vs}
  vserver ${vs} start
done
```

Niente e' facile come sembra

Corollario:
Niente e' facile come sembra

heartbeat costituisce il nucleo del sistema, ma oltre a questo sono necessari altri accorgimenti:

- garantire heartbeat sia in esecuzione.
- garantire assenza di *split-brain*.
- garantire la visibilità dei nodi.
- garantire all'interno dei vserver i servizi siano attivi.
- fare regolarmente **backup**.

Domande?

Grazie per l'attenzione.

Corollario: