



Calcolo distribuito ad alte prestazioni

Esercitazione guidata n.1
Cluster High availability

Tutor: Damiano Perri
Contatti: damiano.perri@gmail.com

1.

Che problema affrontiamo?

Esercitazione n°1

Realizzazione di un cluster ad alta affidabilità



Un cluster ad alta affidabilità è un insieme di nodi interconnessi tra loro capaci di distribuirsi il carico di lavoro in maniera autonoma e capaci di garantire la fornitura dei servizi nel caso uno o più nodi si disconnettano.

Esercitazione n°1

Realizzazione di un cluster ad alta affidabilità



La resistenza ai guasti deve essere garantita dalla presenza di diversi nodi con lo stesso contenuto informativo all'interno della cartella pubblica (di norma in /var/www/html), in grado di subentrare nell'erogazione dei contenuti html non appena un nodo "active" presenti una problematica di connessione

2.

Quando un sistema di questo
tipo è utile?

Esercitazione n°1

Realizzazione di un cluster ad alta affidabilità



Tolleranza ai guasti

Ridondanza degli apparati

Blackout che colpisce alcuni nodi del cluster

Carichi di lavoro eccessivi per una sola macchina

E altri ancora...

3.

Punti a favore e a sfavore

Esercitazione n°1

Punti a favore



Economicità

Scalabilità grazie alle risorse distribuite

Facilità di aggiornamento e manutenzione

Disponibilità di software open source

Affidabilità

...

Esercitazione n°1

Punti a sfavore



Maggiore difficoltà nella manutenzione software

Maggiore spazio fisico occupato

Maggiore consumo di energia rispetto ad un singolo server

...

4.

Cosa è richiesto?

Entrando nello specifico

Lato software



Realizzare un cluster di 2 nodi nel quale si dovrà avere

- Pacemaker come CRM (Cluster Resource Manager)*
- Corosync come Cluster Engine*
- Apache per la gestione dei servizi web*
- DRBD per creare una risorsa drbd*
- Documentare dettagliatamente il lavoro fatto*

Entrando nello specifico

Lato software



Il cluster verrà realizzato tramite un ambiente virtualizzato

Si consiglia Virtualbox / Vmware come software di virtualizzazione

Il sistema operativo consigliato è Fedora

Entrando nello specifico

Lato hardware



Requisiti minimi di sistema:

*Per ogni macchina virtuale è consigliato assegnare
1.5GB di ram (3GB totali)*

*Il computer da utilizzare dovrebbe avere un
quantitativo di ram maggiore o uguale a 4GB di ram*

Consiglio

*se avete un SSD evitate di installare le VM su di esso,
in tal caso usate un disco meccanico (se possibile) o in
alternativa disabilitate lo swap in fase di installazione*

Un primo passo...



Una volta che Fedora è operativo procedere con l'installazione di un editor di testo basilare nano o gedit andranno benissimo

Es di comandi:

*su
dnf -y install nano*

E alla fine del lavoro



Produrre una relazione dettagliata di tutto quello che si è svolto.

Risulta apprezzato l'utilizzo di Latex per renderla anche gradevole alla vista.

Sono richiesti degli screenshots per spiegare meglio i passi che sono stati compiuti.

La relazione non deve essere un How-To per la risoluzione del problema, ma deve anche affrontare la tematica trattata in modo discorsivo

5

Gli applicativi fondamentali

Cosa bisogna utilizzare

Lato software



*I software installati nei nodi del cluster sono prevalentemente tre:
Pacemaker, Corosync e HeartBeat.*

Cosa bisogna utilizzare

HeartBeat 1/2



Heartbeat è un demone che consente la realizzazione logica del cluster.

Si occupa anche dello scambio di messaggi tra i nodi, dalla versione 2.0 comprende al suo interno un CRM, cioè un gestore delle risorse logiche.

Cosa bisogna utilizzare

HeartBeat 2/2



Heartbeat invia costantemente un messaggio ai nodi chiedendo se questi sono ancora vivi, nel caso in cui il nodo primario non risponda, uno dei nodi passivi prende in carico l'erogazione dei servizi forniti dal nodo primario gestendo l'indirizzo IP virtuale a cui il cluster fa riferimento.

Cosa bisogna utilizzare

Pacemaker



Pacemaker è responsabile del funzionamento delle risorse, permette di controllarne il loro stato, di avviarle o di stopparle, e gestisce il comportamento che queste devono avere nel caso si verifichino malfunzionamenti.

Cosa bisogna utilizzare

Corosync



Corosync è un altro demone che gestisce lo scambio di messaggi e l'appartenenza dei nodi all'interno dei gruppi. Rappresenta una evoluzione di OpenAIS, e promette di risolvere i problemi che una volta si osservavano con OpenAIS, Pacemaker e Apache Qpid.

Corosync assicura che ogni server nel sistema mantenga una copia ridondante delle informazioni.

Questo approccio è chiamato "distributed state machine".

Cosa bisogna utilizzare

Stonith 1/3



In una situazione reale potrebbe essere possibile che un nodo slave si renda conto che l'attuale master non stia funzionando correttamente, ad esempio per un problema di rete. Se in questa situazione il nodo slave diventasse master, potrebbe creare inconsistenza nei dati in quando potrebbe essere possibile che l'anomalia che era stata analizzata fosse relativa alla rete e non al nodo stesso.

Cosa bisogna utilizzare

Stonith 2/3



Grazie a Stonith si renderebbe non operativo il master, evitando di creare inconsistenza nei dati. Ancora meglio sarebbe delegare questa funzionalità ad un componente hardware in grado di spegnere il nodo che presenta problemi di questo tipo.

Cosa bisogna utilizzare

Stonith 3/3



In questo progetto Stonith non verrà configurato in quanto non possono esserci situazioni di inconsistenza dei dati e i nodi non sono alimentati effettivamente da una presa di corrente in quanto virtualizzati all'interno della macchina Host.

Cosa bisogna utilizzare

Quorum



Quorum permette di disabilitare il cluster nel caso in cui meno della metà dei nodi sono inaccessibili e quindi disconnessi dalla rete.

Quorum dovrà essere disattivato perché con un esempio utilizzando solo 2 nodi rischieremmo che il cluster si disattivi nel caso in cui un solo nodo vada offline.

Cosa bisogna utilizzare

DRBD



Il DRDB (Distributed Replicated Block Device), viene utilizzato per replicare il contenuto informativo all'interno di un nodo master, sui vari nodi slave.

Verrà analizzato nel dettaglio tra poco..

6.

Installare il software base

I pacchetti fondamentali



```
sudo dnf -y install pacemaker corosync pcs  
sudo dnf -y install drbd-pacemaker drbd-udev  
sudo dnf -y install httpd  
sudo dnf -y install iptables-services
```

Per iniziare... 1/4



Assegnare IP statici alle macchine

Disabilitare IPV6

Riavviare l'interfaccia di rete

Assegnare un nome nel file hosts alla macchina

Configurare...o disabilitare il firewall :)

Successivamente... 2/4



Avviare pacemaker

Creare un utente per il cluster

Creare il cluster assegnandogli i nodi

Avviare il cluster

Avviare corosync

Fare un check per vedere che tutto vada con

Sudo pcs status

Successivamente... 3/4



Avviare stonith

Disabilitare quorum

“Quorum è un componente di pacemaker che disattiva il Cluster nel caso più della metà dei nodi risulta offline”

Assegnare un ip virtuale al cluster

Configurare apache `/etc/httpd/conf.d/status.conf`

E' necessario specificare che solo l'host locale deve accedere al servizio

Ci siamo quasi... 4/4



*Creare una pagina web per testare apache
Aggiungere la risorsa di tipo apache al cluster
e controllarne lo stato con “sudo pcs status”*

Attenzione:

*Se siete arrivati a questo punto il cluster dovrebbe già
sostenere la caduta di un nodo!*

7. DRBD

Ripassiamo a cosa serve

1/3



Il DRDB (Distributed Replicated Block Device), viene utilizzato per replicare il contenuto informativo all'interno di un nodo master, sui vari nodi slave. La funzionalità software di replicazione dei blocchi dell'hard disk del nodo master è completamente trasparente alle applicazioni che ne devono leggere il contenuto.

Ripassiamo a cosa serve

2/3



Se ad esempio inseriamo le pagine di un contenuto Web all'interno dello spazio governato dal DRBD, queste verranno replicate su tutti i nodi del cluster garantendo continuità di servizio e consistenza dei dati. Un applicativo software di questo tipo viene installato quando la soluzione che si sta realizzando è fortemente indirizzata all'affidabilità dei dati.

Ripassiamo a cosa serve

3/3



Se il nodo primario dovesse incorrere in problematiche di rete, heartbeat, avvierà la procedura che porta uno dei nodi secondari a prendere il posto del primario, garantendo la continuità del servizio offerto.

Realizzare DRBD



*Assegnare un disco da 1GB alla macchina virtuale
Partizionarlo e dargli un nome, es: “drbd”
All’interno delle configurazioni di drbd.d si
inserisce una nuova risorsa, chiamata a titolo di test
“wwwdata.res”
Aggiungere la nuova risorsa al cluster*

Grazie !
Domande?

Per eventuali contatti:
Damiano.perri@gmail.com