Red Hat Enterprise Linux 6

Panoramica sull'High Availability Add-On

Panoramica sull'High Availability Add-On per Red Hat Enterprise Linux

Edizione 6

Last Updated: 2017-11-23

Red Hat Enterprise Linux 6 Panoramica sull'High Availability Add-On

Panoramica sull'High Availability Add-On per Red Hat Enterprise Linux
Edizione 6

Nota Legale

Sommario

La Panoramica sull'High Availability Add-On fornisce un sommario delle informazioni per Red Hat Enterprise Linux 6.

Indice

INTRODUZIONE **3**
1. ABBIAMO BISOGNO DEI VOSTRI COMMENTI! 4

CAPITOLO 1. PANORAMICA SULL'HIGH AVAILABILITY ADD-ON **5**
1.1. CONCETTI DI BASE DEL CLUSTER 5
1.2. INTRODUZIONE ALL'HIGH AVAILABILITY ADD-ON 6
1.3. INFRASTRUTTURA DEL CLUSTER 7

CAPITOLO 2. GESTIONE DEL CLUSTER CON CMAN **8**
2.1. QUORUM DEL CLUSTER 8
2.1.1. Quorum Disk 9
2.1.2. Tie-breaker 9

CAPITOLO 3. RGMANAGER **11**
3.1. DOMINI DI FAILOVER 11
3.1.1. Esempi di comportamento 12
3.2. POLITICHE DEL SERVIZIO 13
3.2.1. Politica per l'avvio 13
3.2.2. Politica di ripristino 13
3.2.3. Estensioni della politica di riavvio 13
3.3. ALBERI DELLE RISORSE - DEFINIZIONI / DI BASE 14
3.3.1. Ordine d'avvio, dipendenze e rapporti Genitore / Figlio 14
3.4. STATI E OPERAZIONI DEL SERVIZIO 15
3.4.1. Funzioni del servizio 15
3.4.1.1. L'operazione freeze 15
3.4.1.1.1. Comportamenti del servizio quando sospeso 15
3.4.2. Stati del servizio 16
3.5. COMPORTAMENTI DELLA MACCHINA VIRTUALE 16
3.5.1. Operazioni normali 16
3.5.2. Migrazione 17
3.5.3. Funzioni RGManager per la macchina virtuale 17
3.5.3.1. Controllo della macchina virtuale 18
3.5.3.2. Supporto dominio transitorio 18
3.5.3.2.1. Funzioni per la gestione 18
3.5.4. Comportamenti non gestiti 18
3.6. AZIONI DELLE RISORSE 18
3.6.1. Valori restituiti 19

CAPITOLO 4. FENCING **20**

CAPITOLO 5. LOCK MANAGEMENT **25**
5.1. DLM LOCKING 25
5.2. STATI DI LOCK 26

CAPITOLO 6. STRUMENTI DI AMMINSITRAZIONE E CONFIGURAZIONE **27**
6.1. TOOL DI AMMINISTRAZIONE DEL CLUSTER 27

CAPITOLO 7. VIRTUALIZZAZIONE E HIGH AVAILABILITY **29**
7.1. VM COME SERVIZI/RISORSE AD ELEVATA DISPONIBILITÀ 29
7.1.1. Consigli generali 30
7.2. CLUSTER GUEST 31
7.2.1. Utilizzo storage condiviso iSCSI e fence_scsi 33
7.2.2. Consigli generali 33

APPENDICE A. DIARIO DELLE REVISIONI 35

INTRODUZIONE

Questo documento fornisce una panoramica dettagliata su High Availability Add-On per Red Hat Enterprise Linux 6.

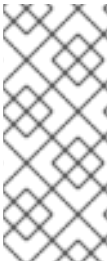
Anche se le informazioni presenti in questo documento sono generali, per la sua comprensione è necessario avere una conoscenza avanzata di Red Hat Enterprise Linux, e capire i concetti di computazione del server.

Per maggiori informazioni su come utilizzare Red Hat Enterprise Linux, consultate le seguenti risorse:

- *Red Hat Enterprise Linux Installation Guide*— Fornisce le informazioni relative all'installazione di Red Hat Enterprise Linux 6.
- *Red Hat Enterprise Linux Deployment Guide*— Fornisce le informazioni relative all'impiego, configurazione ed amministrazione di Red Hat Enterprise Linux 6.

Per maggiori informazioni su questo prodotto e sui prodotti relativi a Red Hat Enterprise Linux 6 consultate le seguenti risorse:

- *Configurazione e gestione di High Availability Add-On*Questa guida descrive le configurazione e la gestione di High Availability Add-On (conosciuta anche come Red Hat Cluster) per Red Hat Enterprise Linux 6.
- *Amministrazione del Logical Volume Manager*— Fornisce una descrizione del Logical Volume Manager (LVM), ed include le informazioni su come eseguire LVM in un ambiente clusterizzato.
- *Global File System 2: Configurazione e amministrazione*— Fornisce le informazioni sull'installazione, configurazione ed amministrazione del Red Hat GFS2 (Red Hat Global File System 2) incluso con il Resilient Storage Add-On.
- *DM Multipath* — Fornisce le informazioni relative all'utilizzo del Device-Mapper Multipath di Red Hat Enterprise Linux 6.
- *Amministrazione Load Balancer*— Fornisce le informazioni sulla configurazione dei servizi e sistemi ad elevate prestazioni con Red Hat Load Balancer Add-On (Precedentemente conosciuto come Linux Virtual Server [LVS]),
- *Note di rilascio*— Fornisce le informazioni sulla release corrente dei prodotti di Red Hat.



NOTA

Per maggiori informazioni sui metodi migliori disponibili per l'implementazione e l'aggiornamento dei cluster Red Hat Enterprise Linux usando High Availability Add-On e Red Hat Global File System 2 (GFS2) consultare "Red Hat Enterprise Linux Cluster, High Availability, e Metodi migliori per l'implementazione del GFS" disponibili sul portale clienti di Red Hat . https://access.redhat.com/kb/docs/DOC-40821.

Questo documento ed altre documentazioni di Red Hat sono disponibili in versione HTML, PDF, e RPM sul CD di documentazione di Red Hat Enterprise Linux ed online su http://access.redhat.com/documentation/docs.

1. ABBIAMO BISOGNO DEI VOSTRI COMMENTI!

Se individuate degli errori di battitura o se pensate di poter contribuire al miglioramento di questa guida, contattateci subito. Inviate un report in Bugzilla: http://bugzilla.redhat.com/ nei confronti di **Red Hat Enterprise Linux 6** sul componente *doc-High_Availability_Add-On_Overview* con un numero di versione: **6 . 6**.

Se avete dei suggerimenti per migliorare la documentazione cercate di essere il più specifici possibile. Se avete trovato un errore, vi preghiamo di includere il numero della sezione e alcune righe di testo, in modo da agevolare le ricerca dell'errore stesso.

CAPITOLO 1. PANORAMICA SULL'HIGH AVAILABILITY ADD-ON

High Availability Add-On è un sistema clusterizzato in grado di fornire affidabilità, scalabilità e disponibilità per i servizi di produzione critici. Le seguenti sezioni forniscono una descrizione dettagliata dei componenti e delle funzioni dell'High Availability Add-On:

- Sezione 1.1, «Concetti di base del cluster»
- Sezione 1.2, «Introduzione all'High Availability Add-On»
- Sezione 1.3, «Infrastruttura del cluster»

1.1. CONCETTI DI BASE DEL CLUSTER

Un cluster è costituito da due i più computer (chiamati *nodi* o *membri*), che operano insieme per eseguire un compito. Sono presenti quattro tipi principali di cluster:

- Storage
- High availability
- Bilanciamento del carico
- Elevate prestazioni

I cluster di archiviazione (storage) forniscono una immagine uniforme del file system sui server presenti in un cluster, permettendo ai server stessi di leggere e scrivere simultaneamente su di un file system condiviso singolo. Un cluster di archiviazione semplifica l'amministrazione dello storage, limitando l'installazione ed il patching di applicazioni su di un file system. Altresì, con un file system cluster-wide, un cluster di archiviazione elimina la necessità di copie ridondanti di dati dell'applicazione, semplificando il processo di backup e di disaster recovery. High Availability Add-On fornisce uno storage clustering insieme al Red Hat GFS. (parte del Resilient Storage Add-On).

I cluster High-availability forniscono una disponibilità continua dei servizi tramite l'eliminazione dei così detti single points of failure, e tramite l'esecuzione del failover dei servizi da un nodo del cluster ad un altro nel caso in cui il nodo diventi non operativo. Generalmente i servizi presenti in un cluster high-availability leggono e scrivono i dati (tramite un file system montato in modalità di lettura-scrittura). Per questo motivo un cluster high-availability deve essere in grado di garantire l'integrità dei dati, poichè un nodo del cluster può assumere il controllo di un servizio da un altro nodo. La presenza di errori in un cluster high-availability non risulta essere visibile da parte di client esterni al cluster. (I cluster high-availability sono talvolta indicati come cluster di failover.) High Availability Add-On fornisce un clustering high-availability attraverso il proprio componente High-availability Service Management, **`rgmanager`**.

I cluster a bilanciamento del carico 'cluster load-balancing' inviano le richieste del servizio di rete a nodi multipli del cluster, in modo da bilanciare il carico richiesto tra i nodi del cluster. Il bilanciamento del carico fornisce una scalabilità molto economica, poichè è possibile corrispondere il numero di nodi in base ai requisiti del carico. Se un nodo all'interno del cluster risulta essere non operativo, il software per il bilanciamento del carico rileva l'errore e ridireziona le richieste ad altri nodi del cluster. Il fallimento di un nodo nel cluster load-balancing non risulta essere visibile da parte dei client esterni al cluster. Il bilanciamento del carico è disponibile attraverso il Load Balancer Add-On.

I cluster High-performance utilizzano i nodi del cluster per eseguire processi di calcolo simultanei. Un cluster high-performance permette alle applicazioni di lavorare in parallelo aumentando così la

prestazione delle applicazioni. (I cluster High performance vengono anche identificati come cluster computational o grid computing.)

NOTA

I cluster sopra citati rappresentano le configurazioni di base; in base alle vostre esigenze potreste aver bisogno di una combinazione dei tipi di cluster appena descritti.

RHEL; High Availability Add-On rende disponibile il supporto per la configurazione e la gestione *solo* dei server ad elevata disponibilità. Esso *non* i cluster ad elevate-prestazioni.

1.2. INTRODUZIONE ALL'HIGH AVAILABILITY ADD-ON

High Availability Add-On è un set integrato di componenti software il quale può essere impiegato in una varietà di configurazioni idonee per far fronte alle vostre esigenze di prestazione, high-availability, di bilanciamento del carico, scalabilità, file sharing e di risparmio.

High Availability Add-On consiste nei seguenti componenti:

- Infrastruttura del cluster — Fornisce le funzioni fondamentali per i nodi in modo che gli stessi possano operare insieme come un cluster: gestione della configurazione-file, gestione appartenenza, lock management, e fencing.
- High-availability Service Management — Fornisce il failover dei servizi da un nodo del cluster ad un altro, in caso in cui il nodo non è più operativo.
- Tool di amministrazione del cluster — Tool di gestione e configurazione per l'impostazione, la configurazione e la gestione di High Availability Add-On. È possibile utilizzare i suddetti tool con i componenti dell'infrastruttura del cluster, e con componenti per la Gestione del servizio, High availability e storage.

NOTA

Solo i cluster con un solo sito sono completamente supportati. Non sono supportati invece i cluster che si estendono su posizioni fisiche multiple. Per maggiori informazioni sui cluster con siti multipli contattare un rappresentante per il supporto o alle vendite di Red Hat.

È possibile altresì integrare i seguenti componenti all'High Availability Add-On:

- Red Hat GFS2 (Global File System 2) — È parte del Resilient Storage Add-On, esso rende disponibile un file system del cluster da usare con High Availability Add-On. GFS2 permette a nodi multipli di condividere lo storage ad un livello del blocco, come se lo storage fosse collegato localmente ad ogni nodo del cluster. Il file system del cluster GFS2 ha bisogno di una infrastruttura cluster.
- Cluster Logical Volume Manager (CLVM) — È parte del Resilient Storage Add-On e fornisce la gestione del volume del cluster storage. Il supporto CLVM ha bisogno anche di una infrastruttura cluster.
- Load Balancer Add-On — Software di instradamento che fornisce l'IP-Load-balancing. Load Balancer Add-On viene eseguito su di una coppia di server virtuali ridondanti, che distribuisce le richieste del client in modo omogeneo ai real server dietro i server virtuali.

1.3. INFRASTRUTTURA DEL CLUSTER

L'infrastruttura del cluster di High Availability Add-On fornisce le funzioni di base per un gruppo di computer (chiamati *nodi* o *membri*), in modo da poter operare insieme come un cluster. Una volta formato il cluster utilizzando l'infrastruttura del cluster stesso, è possibile utilizzare altri componenti per far fronte alle esigenze del proprio cluster (per esempio per l'impostazione di un cluster per la condivisione dei file su di un file system GFS2, oppure per l'impostazione del servizio di failover). L'infrastruttura del cluster esegue le seguenti funzioni:

- Cluster management
- Lock management
- Fencing
- Gestione configurazione del cluster

CAPITOLO 2. GESTIONE DEL CLUSTER CON CMAN

Una gestione del cluster permette di amministrare il quorum del cluster e la sua appartenenza. CMAN (abbreviazione di cluster manager) esegue una gestione del cluster con l'High Availability Add-On per Red Hat Enterprise Linux. CMAN è un cluster manager distribuito e viene eseguito su ogni nodo del cluster; la gestione del cluster viene distribuita su tutti i nodi nel cluster.

CMAN controlla l'appartenenza tramite il monitoraggio dei messaggi provenienti da altri nodi del cluster. Quando l'appartenenza del cluster cambia, il cluster manager invia una notifica ad altri componenti dell'infrastruttura, i quali a loro volta intraprendono l'azione appropriata. Per esempio, se un nodo del cluster non trasmette alcun messaggio entro un ammontare di tempo prestabilito, il cluster manager rimuove il nodo e comunica agli altri componenti dell'infrastruttura che il nodo in questione non risulta più essere un membro. Ancora, altri componenti dell'infrastruttura del cluster determinano le azioni da intraprendere, previa notifica, poichè il nodo non è più un membro del cluster. Per esempio, il fencing potrebbe scollegare il nodo non più membro.

CMAN controlla il quorm del cluster monitorando il conteggio dei nodi. Se più della metà dei nodi risultano attivi il cluster avrà un quorum. Se al contrario, la metà dei nodi (o un numero minore) risultano attivi, il cluster non avrà alcun quorum. In presenza di questo scenario l'attività presente al suo interno verrà arrestata. Il quorum impedisce il verificarsi di una condizione chiamata "split-brain" — una condizione nella quale due istanze sono in esecuzione nello stesso cluster. In una condizione split-brain ogni istanza del cluster potrà accedere alle risorse senza essere a conoscenza della presenza di una seconda istanza, in tale situazione si verificherà una corruzione dei dati.

2.1. QUORUM DEL CLUSTER

Il Quorum è un algoritmo di voto usato da CMAN.

Un cluster potrà operare correttamente solo in presenza di un accordo tra i membri relativo al loro stato. Quindi, un cluster risulta avere un quorum se la maggior parte dei nodi sono attivi, in comunicazione e conformi ai membri attivi del cluster. Per esempio, è possibile avere un quorum in un cluster a tredici nodi se sette o più nodi sono in comunicazione tra loro. Se il settimo nodo risulta inattivo il cluster perderà il suo quorum e non sarà più in funzione.

Per impedire problematiche di *split-brain* è necessario avere un quorum. In assenza del quorum si potrà verificare un errore di comunicazione sul cluster a tredici nodi, il quale potrà causare una situazione nella quale sei nodi svolgeranno le proprie funzioni sullo storage condiviso, e altri sei nodi che a loro volta svolgeranno le loro funzioni in modo indipendente. A causa di un errore di comunicazione i due cluster parziali sovrascriveranno aree del disco corrompendo il file system. Forzando l'implementazione delle regole del quorum, solo uno dei cluster parziali potrà usare lo storage condiviso, proteggendo così l'integrità dei dati.

Il quorum non è in grado di impedire situazioni di split-brain, ma è in grado di decidere qual è il membro dominante. Così facendo è possibile avere un funzionamento corretto nel cluster. In presenza di problematiche di split-brain il quorum impedisce qualsiasi attività ad un gruppo (o gruppi) del cluster.

Il quorum del cluster viene determinato per mezzo di una comunicazione tra i nodi tramite Ethernet. Facoltativamente il quorum può essere determinato usando una combinazione di messaggi per mezzo di Ethernet *e* un quorum disk. Per un quorum usando Ethernet, esso consiste in una maggioranza semplice (50% dei nodi + 1 extra). Nella configurazione di un quorum disk, il quorum consiste nelle condizioni specificate dall'utente.

NOTA

Per impostazione predefinita ogni nodo ha un voto per quorum. Facoltativamente è possibile configurare ogni nodo in modo da avere più di un voto.

2.1.1. Quorum Disk

Un quorum disk, o partizione, è una sezione di un disco impostata per un suo utilizzo con componenti del progetto cluster. Essa presenta due modalità d'uso le quali verranno affrontate tramite un esempio.

Supponiamo di avere i nodi A e B. Di questi il nodo A non è in grado di ricevere i pacchetti "heartbeat" del gestore del cluster provenienti dal nodo B. Il nodo A non è in grado di sapere il motivo per il quale non riceve alcun pacchetto, ma possiamo fare alcune ipotesi: la prima è che il nodo B potrebbe essere fallito, la seconda è che il nodo B è talmente impegnato da non poter mandare alcun pacchetto. La seconda ipotesi si può verificare se il cluster è troppo grande, i sistemi possono essere molto impegnati o la rete potrebbe non essere ottimale.

In tale situazione il nodo A non è sicuro se il problema è relativo a se stesso (nodo A) o al nodo B. Questa situazione è problematica soprattutto in presenza di un cluster a due nodi, poichè essendo entrambi non in grado di comunicare tra loro, essi potranno tentare di isolarsi a vicenda.

Prima di isolare un nodo potrebbe essere conveniente controllare lo stato attivo dello stesso, anche nel caso in cui non è possibile contattare il nodo in questione. Un quorum disk consente di fare quanto sopra indicato. Prima di isolare il suddetto nodo, il software del cluster è in grado di controllare se il nodo è ancora attivo. Per fare questo esso controllerà se il nodo ha scritto alcuni dati sulla partizione del quorum.

In presenza di un sistema con due nodi il quorum disk funge anche come tie-breaker. Se un nodo ha un accesso al quorum disk e alla rete, allora verranno presi in considerazione due voti.

Un nodo che perde qualsiasi contatto con la rete o quorum disk avrà perso un voto, e per questo motivo potrà essere isolato senza alcun problema.

Per informazioni sulla configurazione dei parametri del quorum disk consultare i capitoli di amministrazione **ccs** e Conga, nel manuale *Amministrazione del Cluster*.

2.1.2. Tie-breaker

I Tie-breaker sono valori euristici che permettono ad una partizione del cluster di decidere se è presente un quorum nell'evento di una suddivisione-equa prima di un fencing. Una struttura tipica di tie-breaker è un IP tie-breaker, chiamato anche nodo ping.

Con questo tipo di tie-breaker i nodi non solo controllano se stessi ma controllano anche il router presente sullo stesso percorso delle comunicazioni del cluster. Se i due nodi perdono contatto tra loro, il nodo che riuscirà a mantenere un contatto con il router sarà quello "vincente". Naturalmente sono presenti casi, come quello di uno switch-loop, dove i due nodi sono in grado di contattare il router ma non sono in grado di comunicare tra loro, questa situazione determina una condizione di split brain. Ecco perchè anche in presenza di tie-breakers è importante avere una configurazione corretta del fencing.

Altri tipi di tie-breaker includono anche la possibilità di una partizione condivisa, spesso chiamata quorum disk, in grado di fornire informazioni aggiuntive. clumanager 1.2.x (Red Hat Cluster Suite 3) presentava un disk tie-breaker in grado di abilitare un funzionamento normale anche quando la rete veniva interrotta, se entrambi i nodi erano ancora in comunicazione tra loro tramite la partizione condivisa.

Sono disponibili schemi di tie-breaker più complessi, come ad esempio QDisk (parte del cluster di linux). QDisk permette l'uso di valori euristici arbitrari. Questi valori permettono ad ogni nodo di determinare il proprio stato di partecipazione nel cluster. Esso viene spesso usato come un IP tie-breaker semplice. Per maggiori informazioni consultare la pagina man di qdisk(5).

CMAN non presenta alcun tie-breaker interno per vari motivi. Tuttavia è possibile implementare i tie-breaker usando un API. L'API permette una registrazione ed un aggiornamento del dispositivo quorum. Per un esempio consultate il codice sorgente di QDisk.

Sarà necessario usare un tie-breaker se:

- in presenza di una configurazione a due nodi con dispositivi di fencing su un percorso della rete diverso rispetto al percorso usato per le comunicazioni del cluster
- in presenza di una configurazione a due nodi dove il fencing è un livello fabric - in particolare per le prenotazioni SCSI

Tuttavia se siete in presenza di una configurazione per il fencing e di rete corretti, un tie-breaker apporterà un livello ulteriore di complessità (ad eccezione di alcuni casi).

CAPITOLO 3. RGMANAGER

RGManager è in grado di gestire e fornire le capacità di failover per la raccolta delle risorse del cluster, ad esempio servizi, gruppi di risorse o alberi di risorse. I gruppi di risorse presentano una struttura ad albero, ed hanno dipendenze genitore-figlio e rapporti di successione all'interno di ogni albero secondario.

RGManager permette agli amministratori di definire, configurare e monitorare i servizi del cluster. In pesenza di un failover di un nodo, RGManager riposizionerà il servizio clusterizzato su un altro nodo cercando di interrompere il meno possibile il servizio. È possibile altresì limitare i servizi su determinati nodi, come ad esempio limitare **`httpd`** ad un gruppo di nodi, mentre **`mysql`** può essere limitato ad un insieme separato di nodi.

Sono disponibili vari processi e agent per il funzionamento di RGManager. A tal proposito di seguito viene riportato un elenco.

- Domini di failover - Funzionamento del sistema del dominio di failover RGManager
- Politiche del servizio - Politiche per il ripristino e avvio del servizio di Rgmanager
- Alberi delle risorse - Funzionamento degli alberi delle risorse di rgmanager, inclusa la successione e gli ordini di avvio/arresto
- Comportamenti operativi del servizio - Significati dei vari stati e funzionamento operativo di rgmanager
- Comportamenti macchine virtuali - Comportamenti da considerare durante l'esecuzione delle VM in un cluster rgmanager
- ResourceActions - Le azioni usate da RGManager e la personalizzazione del file **`cluster.conf`**.
- Event Scripting - Se le politiche di ripristino e di failover di rgmanager non sono idonee per l'ambiente, sarà possibile eseguire una personalizzazione usando un sottosistema di script.

3.1. DOMINI DI FAILOVER

Un dominio di failover è un sottoinsieme ordinato di membri al quale è possibile assegnare un servizio. I domini di failover, anche se utili per la personalizzazione del cluster, non sono necessari per il normale funzionamento.

Di seguito viene riportato un elenco di semantiche relative alle opzioni in riferimento ai comportamenti di un dominio di failover in base alle diverse opzioni di configurazione.

- nodo o membro preferito: Il nodo preferito era il membro indicato per l'esecuzione di un servizio se il membro era offline. È possibile emulare questo comportamento specificando un dominio di failover non ordinato e non limitato di un membro.
- domini limitati: I servizi assegnati al dominio possono essere eseguiti solo sui membri del cluster appartenenti al dominio di failover. Se non è disponibile alcun membro del dominio di failover, il servizio verrà impostato con uno stato "stopped". In un cluster con diversi membri l'uso di un dominio di failover limitato potrà facilitare la configurazioni di un servizio (come ad esempio httpd), il quale necessita di una configurazione identica su tutti i membri che eseguono il servizio. Invece di impostare l'intero cluster per l'esecuzione del servizio, impostare solo i membri nel dominio di failover limitato da associare con il servizio del cluster.

- domini non limitati: Comportamento predefinito, i servizi assegnati a questo dominio possono essere eseguiti su tutti i membri del cluster. Essi possono anche essere eseguiti su un membro del dominio se disponibile. Ciò significa che se un servizio è in esecuzione esternamente al dominio, e un membro del dominio risulta essere online, il servizio eseguirà una migrazione su quel membro a meno che non sia stata impostata l'opzione nofailback.

- dominio ordinato: L'ordine specificato nella configurazione indica l'ordine delle preferenze dei membri all'interno di un dominio. Il membro con la posizione più alta del dominio eseguirà il servizio ogni qualvolta risulta essere online. Ciò significa che se il membro A ha una posizione più alta rispetto al membro B, il servizio eseguirà una migrazione sul membro A, (se in esecuzione sul membro B), solo se il membro A è passato da uno stato offline a uno online.

- dominio non ordinato: Comportamento predefinito, i membri del dominio non presentano alcun ordine di preferenze; qualsiasi membro è in grado di eseguire il servizio. I servizi eseguiranno sempre una migrazione sui membri del dominio di failover quando possibile, tuttavia, in un dominio non ordinato.

- failback: I servizi dei membri di un dominio di failover ordinato devono ritornare sul nodo responsbile per la loro esecuzione iniziale prima del suo fallimento, questa operazione è utile per nodi che falliscono frequentemente. Questa operazione impedisce una interruzione frequente del servizio tra il nodo fallito e quello di failover.

Ordering, restriction e nofailback sono flag e possono essere usati insieme in vari modi (es. ordered+restricted, unordered+unrestricted, ecc.). Queste combinazioni possono interessare sia l'avvio del servizio dopo la formazione del quorum iniziale, che i membri del cluster che assumono il controllo dei servizi in presenza di un fallimento.

3.1.1. Esempi di comportamento

Cluster composto da un insieme di membri: {A, B, C, D, E, F, G}.

Dominio di failover limitato, ordinato {A, B, C}

Con nofailback non impostato: Il servizio 'S' verrà sempre eseguito sul membro 'A' ogni qualvolta il membro 'A' risulta essere online e in presenza di quorum. Se tutti i membri di {A, B, C} risultano essere offline, il servizio non verrà eseguito. Se il servizio è in esecuzione su 'C', e 'A' passa ad uno stato online, il servizio verrà migrato su 'A'.

Con nofailback impostato: Il servizio 'S' verrà eseguito sul membro del cluster con la priorità più alta dopo aver formato un quorum. Se tutti i membri di {A, B, C} risultano essere offline, il servizio non verrà eseguito. Se il servizio è in esecuzione su 'C', e 'A' passa ad uno stato online, il servizio resterà su 'C' se 'C' non fallisce, in caso di fallimento verrà eseguito il failover sul membro 'A'.

Dominio di failover limitato non ordinato {A, B, C}

Il servizio 'S' verrà eseguito solo in presenza di un quorum e con almeno un membro {A, B, C} online. Se un altro membro del dominio passa online, il servizio non verrà riposizionato.

Dominio di failover non limitato, ordinato {A, B, C}

Con nofailback non impostato: Un servizio 'S' verrà eseguito ogni qualvolta è presente un quorum. Se un membro del dominio di failover è online, il servizio verrà eseguito sul membro con la priorità più alta, in caso contrario un membro del cluster verrà selezionato randomicamente per l'esecuzione del servizio. Per riassumere, il servizio verrà eseguito su 'A' ogni qualvolta 'A' è online, seguito da 'B'.

Con nofailback impostato: Un servizio 'S' verrà eseguito ogni qualvolta è presente un quorum. Se un membro del dominio di failover è online durante la formazione del quorum, il servizio verrà eseguito

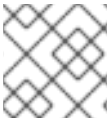
sul membro con la priorità più alta del dominio di failover. E quindi se 'B' è online ('A' offline), il servizio verrà eseguito su 'B'. Se in un secondo momento 'A' si unisce al cluster, il servizio non verrà migrato su 'A'.

Dominio di failover non limitato e non ordinato {A, B, C}

Chiamato anche come "Insieme di membri preferiti". Quando uno o più membri del dominio di failover risultano online, il servizio verrà eseguito su un membro online non specifico del dominio di failover. Se un altro membro del dominio di failover passa ad uno stato online, il servizio non verrà migrato.

3.2. POLITICHE DEL SERVIZIO

RGManager dispone di tre politiche per il ripristino del servizio personalizzabili dall'amministratore in base al servizio.



NOTA

Queste politiche vengono applicate anche alle risorse della macchina virtuale.

3.2.1. Politica per l'avvio

Per impostazione predefinita RGManager inizia tutti i servizi all'avvio di rgmanager e in presenza di un quorum. Gli amministratori possono modificare questo comportamento.

- autostart (default) - inizia il servizio all'avvio di rgmanager e in presenza di un quorum. Se impostato su '0', il cluster non inizierà il servizio e imposterà uno stato disabilitato.

3.2.2. Politica di ripristino

La politica di ripristino è l'azione predefinita intrapresa da rgmanager in presenza di un fallimento del servizio su un nodo particolare. Sono disponibili tre opzioni riportate nel seguente elenco.

- restart (default) - riavvia il servizio sullo stesso nodo. Se non viene specificata un'altra politica per il ripristino, allora verrà implementata questa opzione. Se l'azione di riavvio fallisce, rgmanager riposizionerà il servizio.
- relocate - Prova ad iniziare il servizio su un altro nodo presente nel cluster. Se nessun altro nodo è in grado di iniziare il servizio, quest'ultimo avrà uno stato "stopped".
- disable - Non fare niente. Posiziona il servizio in uno stato disabilitato (disabled).
- restart-disable - Cerca di riavviare il servizio e imposta uno stato disabilitato se il riavvio del servizio fallisce.

3.2.3. Estensioni della politica di riavvio

Utilizzando la politica di ripristino restart è possibile specificare un limite massimo di riavvii sullo stesso nodo per un periodo di tempo specifico. A tale scopo sono disponibili due parametri, max_restarts e restart_expire_time.

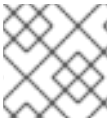
Il parametro max_restarts specifica il numero massimo di riavvii prima di una interruzione e di un riposizionamento del servizio su un altro host presente nel cluster.

Il parametro restart_expire_time indica a rgmanager il periodo entro il quale ricordare un processo di riavvio.

L'uso contemporaneo dei due parametri crea una finestra per il numero di riavvii tollerati in un periodo di tempo dato. Per esempio:

```
<service name="myservice" max_restarts="3" restart_expire_time="300" ...>
  ...
</service>
```

In questo caso la tolleranza del servizio è di 3 riavvii in 5 minuti. In presenza di un quarto fallimento in 300 secondi, rgmanager non riavvierà il servizio, al contrario eseguirà il riposizionamento dello stesso su un altro host disponibile nel cluster.



NOTA

È necessario specificare i parametri insieme; l'uso di un solo parametro non è definito.

3.3. ALBERI DELLE RISORSE - DEFINIZIONI / DI BASE

Quanto di seguito riportato mostra la struttura di un albero delle risorse con un elenco corrispondente il quale definisce ogni area.

```
<service name="foo" ...>
    <fs name="myfs" ...>
        <script name="script_child"/>
    </fs>
    <ip address="10.1.1.2" .../>
</service>
```

- Gli alberi delle risorse sono rappresentazioni XML di risorse, dei rispettivi attributi, dei rapporti tra elementi di pari livello e genitore/figlio. La "radice" di un albero è quasi sempre un tipo di risorsa speciale chiamata servizio. L'albero delle risorse, il gruppo e il servizio sono generalmente intercambiabili su questa wiki. Da una prospettiva di rgmanager, un albero delle risorse rappresenta una unità atomica. Tutti i componenti di un albero vengono iniziati sullo stesso nodo.
- fs:myfs e ip:10.1.1.2 sono imparentati
- fs:myfs è il genitore di script:script_child
- script:script_child è il figlio di fs:myfs

3.3.1. Ordine d'avvio, dipendenze e rapporti Genitore / Figlio

Le regole per i rapporti genitore/figlio in un albero sono molto semplici:

- I genitori vengono avviati prima dei figli
- Arrestare (correttamente) tutti i figli prima di poter arrestare un genitore
- È quindi possibile dire che le risorse figlio dipendono dalle risorse genitore

- Per considerare una risorsa in buono stato è necessario che tutte le risorse figlio siano in buono stato

3.4. STATI E OPERAZIONI DEL SERVIZIO

Le seguenti operazioni riguardano sia i servizi che le macchine virtuali, ad eccezione dell'operazione di migrazione che riguarda solo le macchine virtuali.

3.4.1. Funzioni del servizio

Le operazioni del servizio sono comandi utilizzabili da parte dell'utente per applicare una delle cinque azioni definite nel seguente elenco.

- enable — avvia il servizio, facoltativamente è possibile eseguire questa operazione su una destinazione preferita e in base alle regole del dominio di failover. In loro assenza l'host locale sul quale viene eseguito clusvcadm, avvierà il servizio. Se il processo d'avvio fallisce il servizio si comporta come se fosse stata richiesta una operazione di riposizionamento (consultare quanto di seguito riportato). Se l'operazione ha successo il servizio avrà uno stato started (avviato).

- disable — arresta il servizio e lo posiziona in uno stato disabled (disabilitato). Questa è l'unica operazione permessa quando un servizio è in uno stato failed (fallito).

- relocate — sposta il servizio su un altro nodo. Facoltativamente l'amministratore potrà specificare un nodo preferito per ricevere il servizio, ma l'impossibilità di eseguire il servizio sull'host (per esempio, se il servizio non viene avviato o se l'host è offline), non impedisce il riposizionamento e per questo motivo verrà scelto un nodo diverso. Rgmanager cerca di avviare il servizio su ogni nodo disponibile sul cluster. Se nessun nodo di destinazione presente nel cluster avvia con successo il servizio, il riposizionamento fallisce e verrà eseguito un tentativo di riavvio del servizio sul proprietario originario. Se il suddetto proprietario non è in grado di riavviare il servizio, quest'ultimo avrà uno stato di stopped (arrestato).

- stop — arresta il servizio e lo posiziona in uno stato "stopped".

- migrate — esegue la migrazione della macchina virtuale su un altro nodo. L'amministratore deve specificare un nodo di destinazione. In base all'errore, il fallimento del processo di migrazione potrebbe causare uno stato failed della macchina virtuale, o in uno stato started sul proprietario originario.

3.4.1.1. L'operazione `freeze`

RGManager è in grado di sospendere i servizi. Così facendo si permetterà agli utenti di aggiornare rgmanager, CMAN o qualsiasi altro software sul sistema, minimizzando il periodo di interruzione dei servizi gestiti da rgmanager.

Sarà possibile gestire parte dei servizi rgmanager. Per esempio se siete in possesso di un detebase e di un web server in un servizio rgmanager, sarà possibile sospendere il servizio rgmanager, arrestare il database, eseguire il mantenimento, riavviare il database ed eseguire l'operazione di unfreeze del servizio.

3.4.1.1.1. Comportamenti del servizio quando sospeso

- I controlli sullo stato sono disabilitati

- Le operazioni d'avvio sono disabilitate.

- Le operazioni d'arresto sono disabilitate.
- Il failover non verrà eseguito (anche se disabiliterete il proprietario del servizio)

IMPORTANTE

Se non si rispettano le suddette linee guida le risorse potrebbero essere assegnate su host multipli.

- Se il servizio è sospeso non arrestare tutte le istanze di rgmanager a meno che non pianificate di riavviare gli host prima di riavviare rgmanager.
- Non eseguite l'operazione di unfreeze di un servizio fino a quando il proprietario del servizio non si unisce nuovamente al cluster e riavvia rgmanager.

3.4.2. Stati del servizio

Il seguente elenco definisce gli stati dei servizi gestiti da RGManager.

- disabled — il servizio resterà disabilitato fino a quando un amministratore non eseguirà la riabilitazione, oppure se il cluster perderà il proprio quorum (a quel punto verrà valutato il parametro autostart). Un amministratore potrà eseguire l'abilitazione del servizio da questo stato.
- failed — Si presume che il servizio sia inattivo (dead). Questo stato si verifica ogni qualvolta una operazione d'arresto della risorsa fallisce. L'amministratore deve verificare che non siano state assegnate le risorse (file system montato ecc.) prima di inviare la richiesta. L'unica azione eseguibile da questo stato è "disable".
- stopped — In questo stato il servizio verrà preso in considerazione per l'avvio dopo la successiva transizione del nodo o del servizio. Questo è uno stato provvisorio. Da questo stato un amministratore sarà in grado di disabilitare o abilitare il servizio.
- recovering — Il cluster sta cercando di recuperare il servizio. Un amministratore è in grado di disabilitare il servizio per impedirne il recupero.
- started — Se il controllo dello stato fallisce, eseguire il ripristino seguendo la politica di recupero del servizio. Se l'host che esegue il sevizio fallisce, eseguire il ripristino seguendo le regole del servizio esclusive e il processo di failover del dominio. Un amministratore sarà in grado di riposizionare, arrestare, disabilitare e (all'interno delle macchine virtuali) migrare il servizio da questo stato.

NOTA

Altri stati, come ad esempio `starting` e `stopping` sono stati di transizione speciali di `started`.

3.5. COMPORTAMENTI DELLA MACCHINA VIRTUALE

RGManager gestisce le macchine virtuali in modo diverso da altri servizi non-VM.

3.5.1. Operazioni normali

Le VM gestite da rgmanager devono essere amministrate solo usando clusvcadm o un altro strumento compatibile con il cluster. Molti tipi di comportamento sono comuni con i servizi normali. Ciò include:

- Starting (enabling)
- Stopping (disabling)
- Monitoraggio dello stato
- Riposizionamento
- Ripristino

Per maggiori informazioni sui servizi virtuali ad elevata disponibilità consultare Capitolo 7, *Virtualizzazione e High Availability*.

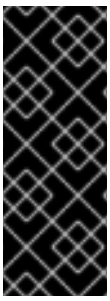
3.5.2. Migrazione

Oltre alle operazioni normali del servizio le macchine virtuali supportano anche un tipo di operazione non accettata da altri servizi: la migrazione. Il processo di migrazione minimizza il periodo di interruzione delle macchine virtuali, rimuovendo la necessità eseguire un avvio/arresto per la modifica della posizione di una macchina virtuale all'interno di un cluster.

Sono disponibili due tipi di migrazione supportati da rgmanager, essi sono selezionabili in base alla VM attraverso l'attributo specifico di migrazione:

- live (default) — la macchina virtuale continua l'esecuzione mentre la maggior parte dei contenuti in memoria vengono copiati sull'host di destinazione. Questa operazione minimizza l'inaccessibilità della VM (generalmente al di sotto di 1 secondo), a scapito delle prestazioni della VM durante la migrazione, e alla quantità totale di tempo necessario per completare il processo stesso.
- pause - la memoria di una macchina virtuale è sospesa mentre i suoi contenuti vengono copiati sull'host di destinazione. Questa operazione minimizza la quantità di tempo necessario per il completamento della migrazione di una macchina virtuale.

Il tipo di migrazione dipende dai requisiti di prestazione e disponibilità. Per esempio, una migrazione live potrebbe significare 29 secondi di prestazioni ridotte e 1 secondo di completa indisponibilità, al contrario una migrazione pause potrebbe significare 8 secondi di completa indisponibilità senza alcun deterioramento delle prestazioni.



IMPORTANTE

Una macchina virtuale può essere un componente del servizio, ma così facendo verranno disabilitate tutte le modalità di migrazione e la maggior parte delle funzioni di seguito riportate.

Altresì, l'uso del processo di migrazione con KVM richiede una configurazione corretta di ssh.

3.5.3. Funzioni RGManager per la macchina virtuale

La seguente sezione riporta i vari metodi utilizzati da RGManager per la gestione delle macchine virtuali.

3.5.3.1. Controllo della macchina virtuale

L'avvio di una macchina virtuale con **clusvcadm**, se VM è già in esecuzione, causerà una ricerca da parte di rgmanager del cluster per la VM, contrassegnando la VM come **started** ovunque si trovi.

L'amministratore che esegue accidentalmente una migrazione della VM tra i nodi del cluster senza utilizzare strumenti non-cluster come **virsh**, causerà una ricerca da parte di rgmanager del cluster per la VM, contrassegnando la VM come **started** ovunque si trovi.

NOTA

Se la VM è in esecuzione in posizioni multiple, RGManager non invierà alcun avviso.

3.5.3.2. Supporto dominio transitorio

Rgmanager supporta le macchine virtuali transitorie supportate da libvirt. Ciò permette a rgmanager di creare e rimuovere le macchine virtuali in tempo reale (on-the-fly), riducendo così la possibilità di un doppio-avvio accindentale se si utilizzano strumenti non-cluster.

Il supporto per le macchine virtuali transitorie permette di archiviare i file di descrizione XML di libvirt su un file system clusterizzato, così facendo non sarà necessario sincronizzare manualmente **/etc/libvirt/qemu** su tutto il cluster.

3.5.3.2.1. Funzioni per la gestione

L'aggiunta o la rimozione di una VM da cluster.conf non causerà l'avvio o l'arresto della VM stessa; con questo processo rgmanager potrà controllare, o meno, la VM

L'operazione di Failback (andare su un nodo preferito) viene eseguita usando il processo di migrazione per minimizzare il periodi di inattività.

3.5.4. Comportamenti non gestiti

Le seguenti condizioni e azioni non sono supportate da RGManager.

- Uso di uno strumento non-cluster-aware (come ad esempio virsh o xm) per la manipolazione dello stato delle macchine virtuali, o della configurazione durante la gestione della macchina virtuale da parte del cluster. È permesso il controllo dello stato di una macchina virtuale (es. virsh list, virsh dumpxml).
- Migrazione di una VM non gestita dal cluster su un nodo non-cluster, o un nodo presente nel cluster che non esegue rgmanager. Rgmanager riavvierà la VM nella posizione precedente, causando l'esecuzione di due istanze della VM e una corruzione del file system.

3.6. AZIONI DELLE RISORSE

RGManager prevede il ritorno dei seguenti valori dagli agent delle risorse:

- start - avvio della risorsa
- stop - arresto della risorsa
- status - controllo stato della risorsa
- metadata - riporto dei metadati OCF RA XML

3.6.1. Valori restituiti

OCF presenta una vasta gamma di valori per le operazioni di monitoraggio, poichè rgmanager invoca lo stato, esso si affida quasi esclusivamente ai codici SysV-style.

0 - riuscito

stop dopo uno stop, o stop quando non in esecuzione, deve tornare un valore di azione riuscita

start dopo start o start quando in esecuzione, deve tornare un valore di azione riuscita

nonzero - fallimento

Se l'operazione di stop ritorna un valore diverso da zero verrà conferito al servizio uno stato fallito (failed), eseguire un ripristino manuale del servizio.

CAPITOLO 4. FENCING

Il Fencing è quel processo in cui è possibile scollegare un nodo dallo storage condiviso del cluster. Tale processo interrompe l'I/O dallo storage condiviso, assicurando così l'integrità dei dati. L'infrastruttura del cluster esegue il fencing attraverso il demone, **fenced**.

Quando CMAN determina la presenza di un nodo fallito, esso lo comunica ad altri componenti dell'infrastruttura del cluster. **fenced**, una volta notificata la presenza di un errore, isola il nodo in questione. Successivamente gli altri componenti dell'infrastruttura del cluster determinano le azioni da intraprendere — essi eseguiranno qualsiasi processo necessario per il ripristino. Per esempio, subito dopo la notifica di un errore a DLM e GFS2, essi sospendono l'attività fino a quando non accerteranno il completamento del processo di fencing d parte di **fenced**. Previa conferma del completamento di tale operazione, DLM e GFS2 eseguono l'azione di ripristino. A questo punto DLM rilascia i blocchi del nodo fallito e GFS2 ripristina il jounal del suddetto nodo.

Il programma di fencing determina, dal file di configurazione del cluster, il metodo da utilizzare. Per la definizione del suddetto metodo è necessario prendere in considerazione due elementi principali: il dispositivo di fencing ed il fencing agent. Questo programma esegue una chiamata nei confronti di un fencing agent specificato nel file di configurazione del cluster. Il fencing agent a sua volta, isola il nodo tramite un dispositivo di fencing. Una volta completato, il programma esegue la notifica al cluster manager.

L'High Availability Add-On fornisce una varietà di metodi usati per il fencing:

- Power fencing — Metodo utilizzato da un controllore di alimentazione per disalimentare il nodo non utilizzabile.
- storage fencing — Un metodo di fencing che disabilita la porta del Fibre Channel che collega lo storage ad un nodo non utilizzabile.
- Altri tipi di fencing — Diversi metodi per il fencing che disabilitano l'I/O o l'alimentazione di un nodo non utilizzabile, incluso gli IBM Bladecenters, PAP, DRAC/MC, HP ILO, IPMI, IBM RSA II, ed altro ancora.

Figura 4.1, «Esempio Power fencing» mostra un esempio di power fencing. Nell'esempio, il programma di fencing nel nodo A causa la disattivazione del nodo D da parte del controller. Figura 4.2, «Esempio Storage fencing» mostra un esempio di storage fencing. Nell'esempio il programma di fencing nel nodo A causa la disabilitazione della porta per il nodo D da parte dell'interruttore del Fibre Channel, scollegando così il nodo D dallo storage.

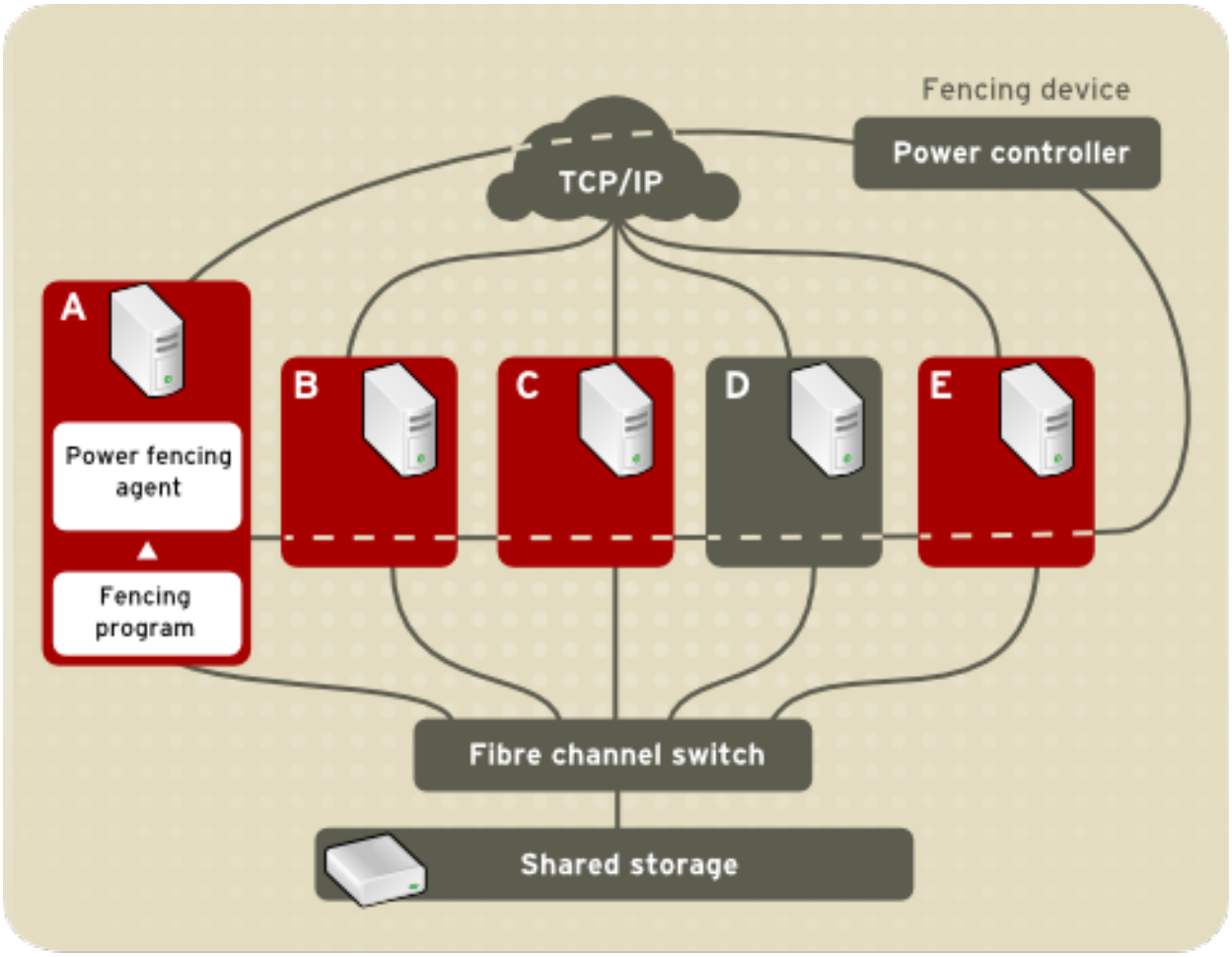


Figura 4.1. Esempio Power fencing

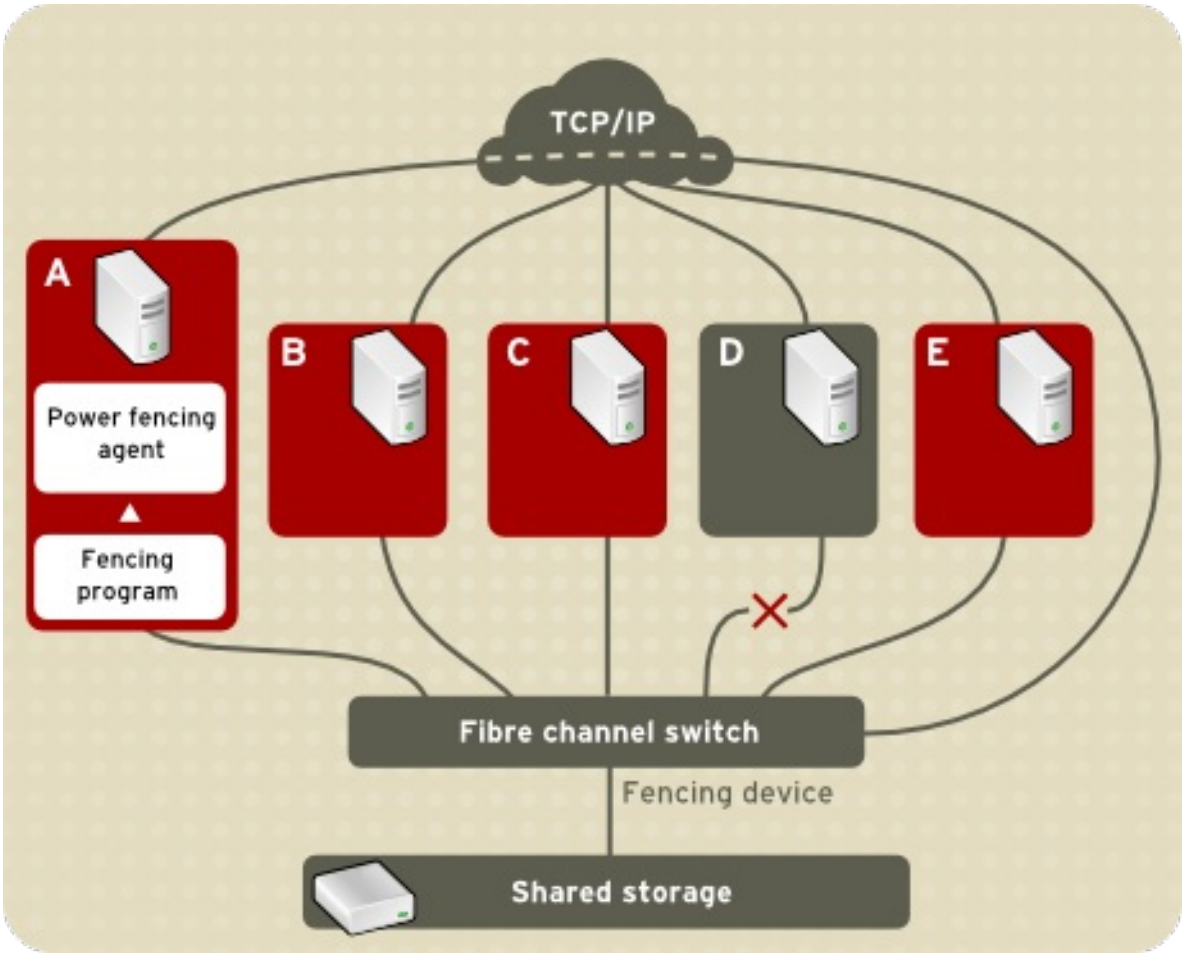


Figura 4.2. Esempio Storage fencing

Specificare un metodo significa modificare il file di configurazione del cluster in modo da assegnare un nome per il metodo di fencing desiderato, il fencing agent, ed il dispositivo di fencing per ogni nodo nel cluster.

Il modo attraverso il quale viene specificato il metodo di fencing dipende dalla presenza di una alimentazione doppia o di percorsi multipli per lo storage. In presenza di alimentazione doppia il metodo usato per il fencing deve specificare un minimo di due dispositivi — uno per il fencing per ogni sorgente di alimentazione (consultare la Figura 4.3, «Fencing di un nodo con alimentazione doppia»). In modo simile, se un nodo presenta percorsi multipli per lo storage del Fibre Channel, allora il metodo usato per il nodo deve specificare un dispositivo di fencing per ogni percorso dello storage del Fibre Channel, esso deve specificare anche due dispositivi da usare — uno per ogni percorso per lo storage del Fibre Channel (consultare la Figura 4.4, «Fencing di un nodo con connessioni doppie al Fibre Channel»).

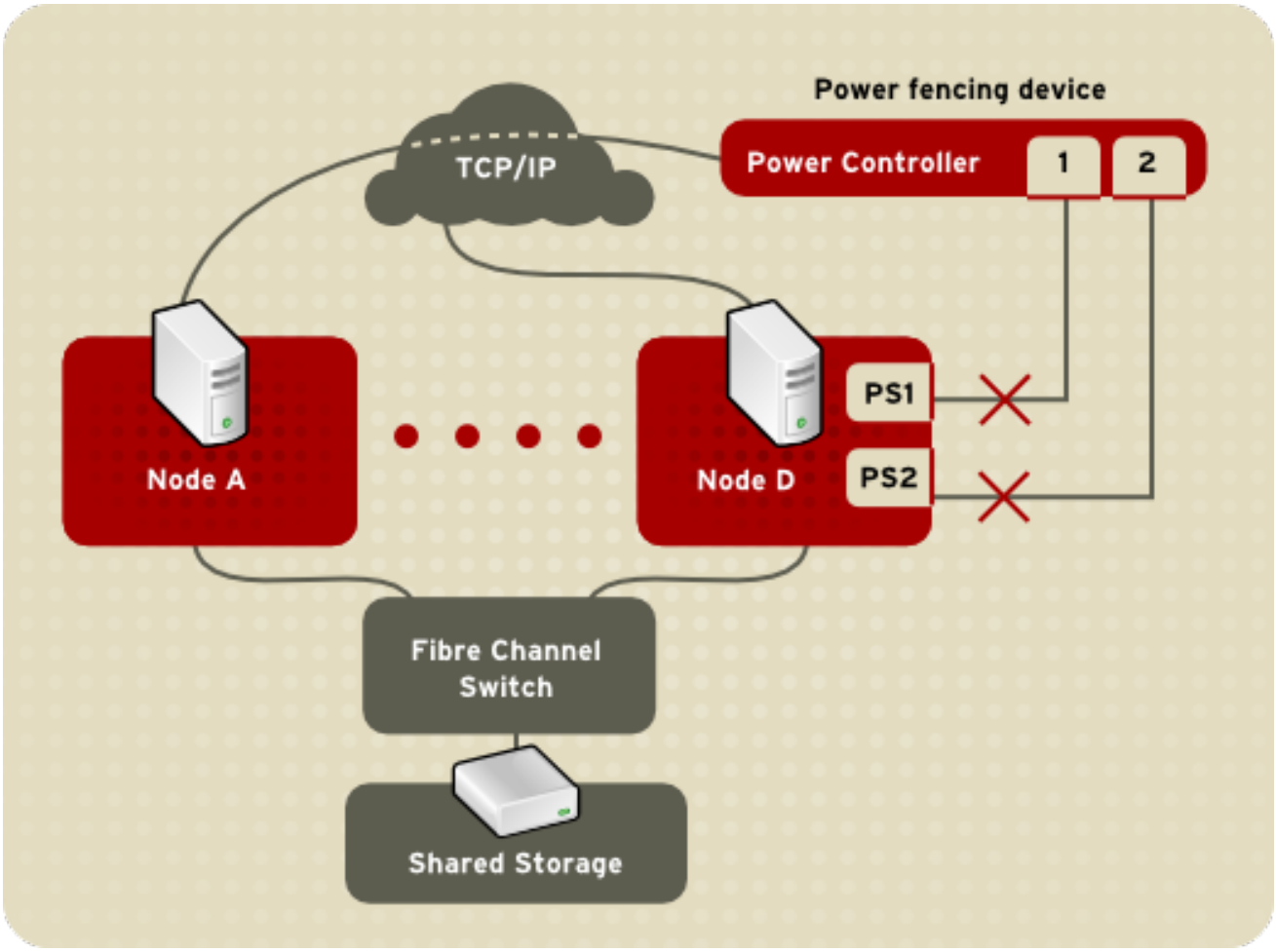


Figura 4.3. Fencing di un nodo con alimentazione doppia

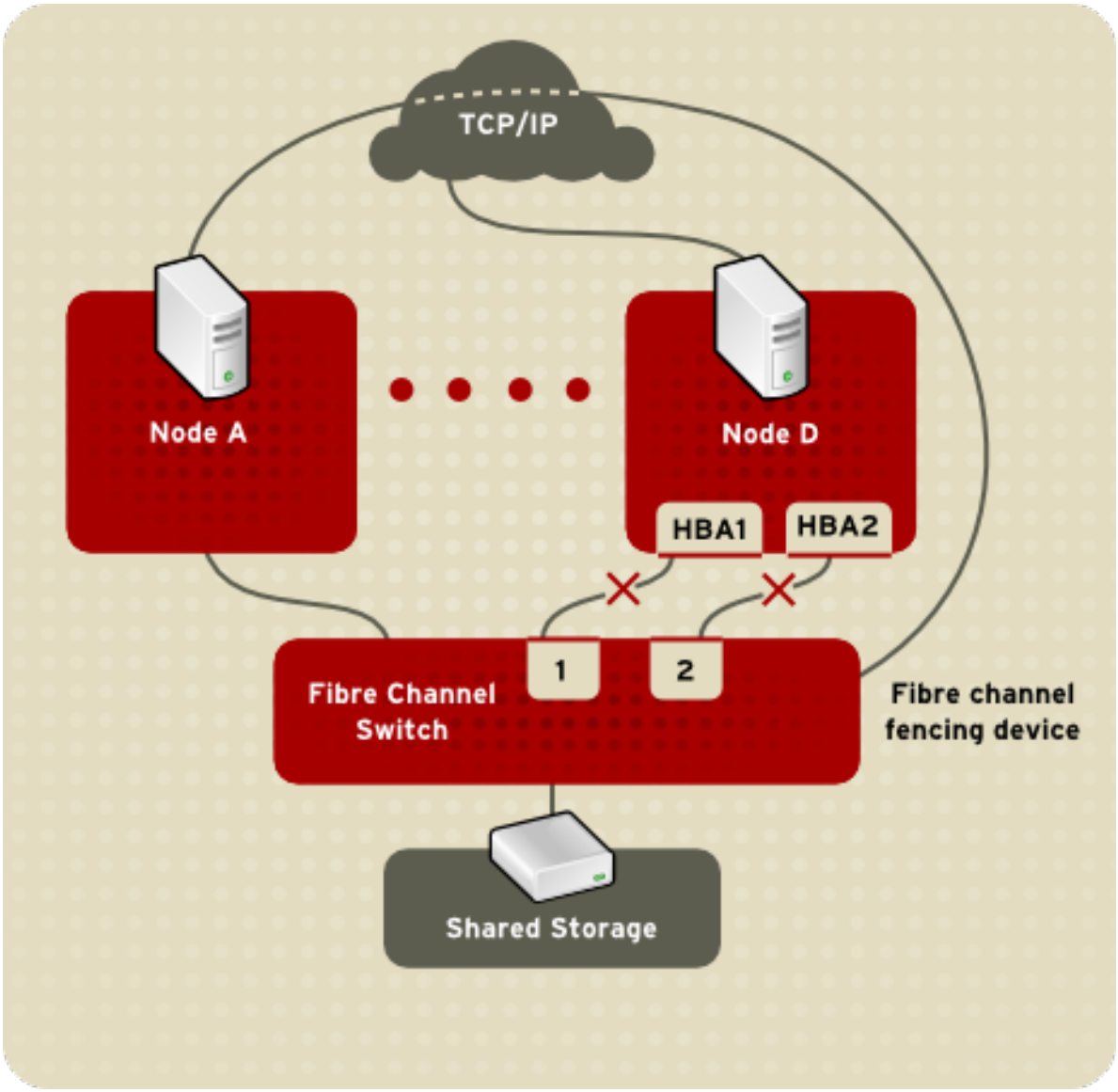


Figura 4.4. Fencing di un nodo con connessioni doppie al Fibre Channel

È possibile configurare un nodo con uno o più metodi di fencing. Quando configurate un nodo per un determinato metodo di fencing, tale metodo risulterà l'unico perseguibile per eseguire il fencing del nodo in questione. Se configurate invece un nodo con metodi di fencing multipli, i suddetti metodi seguiranno una determinata *sequenza* in base all'ordine riportato nel file di configurazione del cluster. Se un nodo fallisce, esso viene isolato utilizzando il primo metodo specificato nel file di configurazione del cluster. Se il primo metodo fallisce, verrà utilizzato il metodo successivo per quel nodo. Se nessun metodo è riuscito ad isolare il nodo, allora il processo di fencing inizierà nuovamente seguendo l'ordine appena descritto e specificato nel file di configurazione del cluster, fino a quando il nodo non verrà isolato con successo.

Per informazioni dettagliate sulla configurazione dei dispositivi di fencing consultate il capitolo corrispondente nel manuale *Amministrazione del Cluster*.

CAPITOLO 5. LOCK MANAGEMENT

Il Lock management è un servizio comune dell'infrastruttura-cluster in grado di fornire un meccanismo per altri componenti per la sincronizzazioe dell'accesso alle risorse condivise. In un cluster di Red Hat, DLM (Distributed Lock Manager) è il lock manager.

Un lock manager è un controllore del traffico in grado di controllare l'accesso alle risorse nel cluster, ad esempio l'accesso al file system GFS. Senza un lock manager non ci sarebbe alcun controllo dello storage condiviso, in tal caso potrebbe verificarsi una corruzione dei dati.

Come indicato dal nome DLM è un distributed lock manager il quale viene eseguito all'interno di ogni nodo del cluster; la gestione del lock viene distribuita su tutti i nodi presenti nel cluster. GFS2 e CLVM utilizzano i lock del lock manager. GFS2 utilizza i lock per sincronizzare l'accesso ai metadati del file system (su storage condiviso). CLVM invece utilizza i lock per sincronizzare gli aggiornamenti per i volumi LVM e gruppi di volumi (presenti sullo storage condiviso). In aggiunta, **rgmanager** utilizza DLM per sincronizzare gli stati dei servizi.

5.1. DLM LOCKING

Il modello DLM locking fornisce un insieme ricco di modalità per il locking e di esecuzioni sincrone e asincrone. Una applicazione è in grado di acquisire un blocco su una risorsa. Un rapporto del tipo one-to-many è presente tra le risorse e i lock: una risorsa può avere lock multipli ad essa associati.

Una risorsa può essere un oggetto, come ad esempio un file, una struttura dati, un database o una routine eseguibile, ma non deve corrispondere necessariamente ad uno di questi elementi. L'oggetto associato con la risorsa del lock determina la granularità del lock stesso. Per esempio, il locking di un intero database è considerato un locking con granularità grossolana. Al contrario, il locking di elementi singoli presenti in un database viene considerato come un locking con granularità più dettagliata.

Il DLM locking supporta:

- Sei modalità di locking che limitano l'accesso alle risorse
- L'avanzamento e il declassamento dei lock attraverso una conversione
- Completamento sincrono delle richieste di lock
- Completamento asincrono
- Dati globali attraverso i blocchi di valore del lock

DLM fornisce i propri meccanismi per il supporto delle funzioni di locking, ad esempio una comunicazione tra i nodi per la gestione del traffico, o i protocolli per riassumere il controllo del lock dopo il fallimento di un nodo o per la migrazione dei blocchi quando un nodo si unisce al cluster. Tuttavia DLM non fornisce alcun meccanismo per la gestione del cluster stesso. Infatti per questo motivo DLM deve operare in un cluster insieme con un secondo ambiente in grado di fornire i seguenti requisiti minimi:

- Il nodo è parte di un cluster.
- Tutti i nodi sono concordi sull'appartenenza del cluster ed è presente un quorum.
- Un indirizzo IP deve essere in grado di comunicare con DLM su un nodo. Normalmente DLM utilizza TCP/IP per una comunicazione tra i nodi, ciò impone un limite di un indirizzo IP per nodo (in tal caso è possibile renderlo più ridondante utilizzando un driver per il bonding). DLM

può essere configurato in modo da utilizzare SCTP per il trasporto tra nodi, abilitando così indirizzi IP multipli su ogni nodo.

DLM opera con qualsiasi ambiente dell'infrastruttura del cluster in grado di fornire i requisiti minimi sopra riportati. La scelta di un ambiente open source o closed source dipende dalle preferenze dell'utente. Tuttavia il limite principale di DLM è la quantità di prove eseguite con i diversi ambienti.

5.2. STATI DI LOCK

Lo stato di un lock indica lo stato corrente di una richiesta. Un lock presenta sempre uno dei seguenti stati:

- Granted — La richiesta ha avuto successo e ha ottenuto la modalità richiesta.
- Converting — Un client ha provato a modificare la modalità di lock, la nuova modalità non è compatibile con il lock esistente.
- Blocked — La richiesta per un nuovo lock non è stata soddisfatta, il lock non è stato assegnato a causa della pesenza di lock in clonflitto tra loro.

Lo stato di un lock viene determinato dalla rispettiva modalità richiesta e dalle modalità dei lock presenti sulla stessa risorsa.

CAPITOLO 6. STRUMENTI DI AMMINSITRAZIONE E CONFIGURAZIONE

Il file di configurazione del cluster, **`/etc/cluster/cluster.conf`** specifica la configurazione dell'High Availability Add-On. Il file di configurazione è un file XML che descrive le seguenti caratteristiche del cluster:

- Nome del cluster — Specifica il nome del cluster, il livello di revisione del file di configurazione e le proprietà relative al timing per il fencing di base usate quando un nodo viene unito al cluster o isolato dallo stesso.
- Cluster — Specifica ogni nodo del cluster, specificandone il nome, l'ID ed il numero di voti del quorum del cluster insieme al metodo per il fencing corrispondente.
- Fence Device — Specifica i dispositivi per il fencing nel cluster. I parametri variano a seconda del tipo di dispositivo. Per esempio, per un controllore dell'alimentazione usato come un dispositivo per il fencing, la configurazione del cluster definisce il nome del controllore dell'alimentazione, l'indirizzo IP relativo, il login e la password.
- Risorse gestite — Specifica le risorse necessarie per creare i servizi del cluster. Le risorse gestite includono la definizione dei domini di failover, delle risorse (per esempio un indirizzo IP), e dei servizi. Insieme, le risorse gestite definiscono i servizi del cluster ed il comportamento del failover dei servizi del cluster.

La configurazione del cluster viene convalidata automaticamente in base allo schema del cluster su **`/usr/share/cluster/cluster.rng`** durante l'avvio ed al ricaricamento di una configurazione. È possibile convalidare una configurazione in qualsiasi momento usando il comando **`ccs_config_validate`**.

È disponibile uno schema per una revisione su **`/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html`** (per esempio **`/usr/share/doc/cman-3.0.12/cluster_conf.html`**).

La convalida della configurazione controlla la presenza dei seguenti errori:

- Validità XML — Controlla che il file di configurazione sia un file XML valido.
- Opzioni della configurazione — Controlla che le opzioni siano valide (elementi ed attributi XML).
- Valori delle opzioni — Controlla che le opzioni contengano i dati validi (limitati).

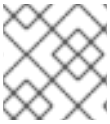
6.1. TOOL DI AMMINISTRAZIONE DEL CLUSTER

La gestione del software di Red Hat High Availability Add-On consiste nell'uso dei tool di configurazione per specificare il rapporto tra i componenti del cluster. I seguenti tool di configurazione del cluster sono disponibili con Red Hat High Availability Add-On:

- **Conga** — Questa è una interfaccia utente completa per l'installazione, configurazione e gestione dell'Add-On High Availability di Red Hat. Consultare la *Configurazione e gestione dell'High Availability Add-On* per informazioni sulla configurazione e gestione dell'Add-On High Availability con **Conga**.
    - Luci — Server delle applicazioni in grado di fornire l'interfaccia utente per Conga. Permette altresì agli utenti di gestire i servizi del cluster e rende disponibile le informazioni per l'assistenza e per la documentazione online quando necessario.

- Ricci — Demone del servizio in grado di gestire la distribuzione della configurazione del cluster. Gli utente sono in grado di inoltrare le informazioni usando l'interfaccia di Luci; la configurazione viene caricata in corosync per la distribuzione ai nodi del cluster.

- Con Red Hat Enterprise Linux 6.1 e versioni più recenti, Red Hat High Availability Add-On fornisce il supporto per il comando usato per la configurazione del cluster **ccs**. Questo comando permette la creazione, modifica e visualizzazione del file di configurazione del cluster cluster.conf da parte di un amministratore. Consultare il manuale *Amministrazione del cluster* per informazioni sulla configurazione e gestione dell'High Availability Add-On con il comando **ccs**.



NOTA

system-config-cluster non è disponibile in RHEL 6.

CAPITOLO 7. VIRTUALIZZAZIONE E HIGH AVAILABILITY

Varie piattaforme di virtualizzazione sono supportate insieme al Red Hat Enterprise Linux 6 usando l'High Availability e il Resilient Storage Add-On. Sono supportati due tipi di implementazioni insieme con il Red Hat Enterprise Linux High Availability Add-on.

Ciò si riferisce a RHEL Cluster/HA in esecuzione su host bare metal utilizzabili come piattaforme di virtualizzazione. Così facendo è possibile configurare il cluster resource manager (rgmanager) per la gestione di macchine virtuali (guest), come risorse ad elevata disponibilità.

- VM come servizi/risorse ad elevata disponibilità
- Cluster guest

7.1. VM COME SERVIZI/RISORSE AD ELEVATA DISPONIBILITÀ

Sia RHEL HA che RHEV forniscono i meccanismi utili per rendere disponibili le macchine virtuali HA (elevata disponibilità). A causa della loro funzionalità fare attenzione nella scelta del prodotto più idoneo a soddisfare le vostre esigenze. Di seguito sono riportate alcune linee guida per la scelta di RHEL HA o RHEV per HA delle VM.

Per il conteggio delle macchine virtuali e degli host fisici:

- Se un numero molto grande di VM risulta essere HA su numerosi host fisici, allora l'uso di RHEV potrebbe rappresentare una opzione migliore a causa della presenza di algoritmi più sofisticati per la gestione del posizionamento delle VM, considerando fattori come CPU, memoria e le informazioni sul carico.
- Se un numero ristretto di VM è HA su alcuni host fisici, l'uso di RHEL HA potrebbe rappresentare la soluzione migliore poichè essa implica un uso ridotto di infrastrutture supplementari. La soluzione più semplice di RHEL HA VM richiede due host fisici per un cluster con 2 nodi. Al contrario, la soluzione RHEV più semplice richiedere 4 nodi: 2 per fornire HA per il server RHEVM e 2 per fungere da host della VM.
- Non sono presenti linee guida specifiche sulla definizione di 'numerosi' host o VM. Ma ricordate che il numero massimo di host in un cluster RHEL HA è 16, e ogni cluster con 8 o più host avrà bisogno di una revisione dell'architettura da parte di Red Hat per determinare il tipo di supporto.

Utilizzo della macchina virtuale:

- Se le VM HA forniscono i servizi usati per le infrastrutture condivise, allora sarà possibile utilizzare RHEL HA o RHEV.
- Per l'HA di un insieme piccolo di servizi critici in esecuzione all'interno delle VM, sarà possibile utilizzare RHEL HA o RHEV.
- Per una infrastruttura idonea ad abilitare il provisioning rapido delle VM, utilizzare RHEV.
    - RHEV VM HA è stato ideato per essere dinamico. L'aggiunta di nuove VM al 'cluster' RHEV, può essere eseguito facilmente ed è completamente supportato.
    - RHEL VM HA non è stato ideato per essere un ambiente molto dinamico. Un cluster deve avere un numero fisso di VM, e per l'intero ciclo di vita è consigliato non aggiungere o rimuovere alcuna VM.

- Non utilizzare RHEL HA per la creazione di ambienti simili al cloud, a causa della natura statica della configurazione del cluster e per il conteggio basso di nodi fisici (16 nodi)

RHEL 5 supporta due piattaforme di virtualizzazione. Xen è supportato dalla release RHEL 5.0. KVM è stato introdotto con RHEL 5.4.

RHEL 6 supporta solo KVM come piattaforma di virtualizzazione.

RHEL 5 AP Cluster supporta sia KVM che Xen nelle macchine virtuali in esecuzione, gestite dall'infrastruttura cluster host.

RHEL 6 HA supporta KVM nelle macchine virtuali in esecuzione, gestite dall'infrastruttura cluster host.

Di seguito vengono riportati i diversi scenari attualmente supportati da Red Hat:

- RHEL 5.0+ supporta Xen insieme con RHEL AP Cluster
- RHEL 5.4 rende disponibile il supporto per le macchine virtuali KVM come risorse gestite in RHEL AP Cluster sotto forma di Anteprima di tecnologia
- RHEL 5.5+ rende disponibile un supporto completo per le macchine virtuali KVM.
- RHEL 6.0+ supporta le macchine virtuali KVM come risorse ad elevata disponibilità in RHEL 6 High Availability Add-On.
- RHEL 6.0+ non supporta le macchine virtuali Xen con RHEL 6 High Availability Add-On, poichè RHEL 6 non supporta più Xen.

NOTA

Per informazioni più aggiornate sulle note speciali relative agli scenari d'impiego supportati, consultare la seguente voce nel Red Hat Knowledgebase:

https://access.redhat.com/kb/docs/DOC-46375

Non ha importanza il tipo di macchine virtuali eseguite come risorse gestite. Qualsiasi guest supportato da Xen o KVM in RHEL, potrà essere utilizzato come guest ad elevata disponibilità. Ciò include le varianti di RHEL (RHEL3, RHEL4, RHEL5) e altre variati di Microsoft Windows. Consultare la documentazione di RHEL per l'ultimissimo elenco di sistemi operativi guest supportati con ogni hypervisor.

7.1.1. Consigli generali

- Con RHEL 5.3 e versioni meno recenti, rgmanager utilizza le interfacce Xen native per la gestione dei Xen domU's (guests). Con RHEL 5.4 questa impostazione è stata modificata in modo da utilizzare libvirt per gli hypervisor Xen e KVM, fornendo una unterfaccia omogenea tra i due tipi di hypervisor. Oltre a questa modifica con RHEL 5.4 e 5.4.z sono state rese disponibili alcune correzioni, e per questo motivo è consigliato aggiornare gli cluster host con i pacchetti di RHEL 5.5 più recenti prima di configurare i servizi gestiti di Xen.
- Per i servizi gestiti KVM è necessario eseguire un aggiornamento a RHEL 5.5 poichè questa è la prima versione di RHEL dove la funzionalità è completamente supportata.
- Controllare sempre l'ultimissima errata di RHEL prima di implementare un cluster, così facendo avrete sempre le ultimissime correzioni relative alle problematiche conosciute.

- L'uso di un insieme di diversi tipi di hypervisor non è supportato. Il cluster host deve essere basato su Xen o KVM.

- Eseguire il provisioning dell'hardware host in modo da poter accettare i guest riposizionati da altri host falliti, senza causare un overcommit della memoria o un overcommit delle CPU virtuali. In presenza di un numero molto elevato di errori tali da causare un overcommit della memoria o delle CPU virtuali, ciò potrebbe deteriorare le prestazioni causando un potenziale fallimento del cluster.

- L'uso diretto degli strumenti di xm o libvirt (virsh, virt-manager) per la gestione delle macchine virtuali (live migrate, stop. start) sotto il controllo di rgmanager, non è supportato o consigliato poichè ciò potrebbe baipassare lo stack di gestione del cluster.

- Ogni nome della VM deve essere unico su tutto il cluster, incluso le VM local-only / non-cluster. Libvirtd applica solo i nomi unici in base all'host. Se eseguite una clonazione manuale di una VM, modificarne il nome nel file di configurazione.

7.2. CLUSTER GUEST

Questo si riferisce a RHEL Cluster/HA in esecuzione nei guest virtualizzati su una varietà di piattaforme virtualizzate. In questo caso RHEL Clustering/HA viene utilizzato principalmente per l'esecuzione delle applicazioni all'interno dei guest ad elevata disponibilità. Questo scenario è simile all'impiego tradizionale di RHEL Clustering/HA negli host bare-metal. La differenza è che il Clustering viene eseguito all'interno dei guest.

Di seguito viene riportato un elenco di piattaforme di virtualizzazione e il livello di supporto attualmente disponibile per l'esecuzione dei cluster guest utilizzando RHEL Cluster/HA. Nell'elenco i guest di RHEL 6 racchiudono sia High Availability (core clustering) che il Resilient Storage Add-Ons (GFS2, clvmd e cmirror).

- Gli host Xen RHEL 5.3+ supportano i cluster guest in esecuzione, dove i sistemi operativi guest hanno una versione RHEL 5.3 o più recente.
    - I cluster guest Xen possono utilizzare fence_xvm o fence_scsi per il fencing del guest.
    - Per utilizzare fence_xvm/fence_xvmd è necessario eseguire un cluster host per il supporto di fence_xvmd, e fence_xvm deve essere utilizzato come un guest fencing agent su tutti i guest clusterizzati.
    - Lo storage condiviso può essere reso disponibile tramite i dispositivi a blocchi condivisi Xen o iSCSI, supportati da uno storage a blocchi dell'host o da un file backed storage (immagini raw).
- Gli host KVM di RHEL 5.5+ non supportano i cluster guest in esecuzione.
- Gli host KVM di RHEL 6.1+ supportano i cluster guest in esecuzione, dove i sistemi operativi guest presentano una versione RHEL 6.1+ o RHEL 5.6+. I guest RHEL 4 non sono supportati.
    - È permesso usare un mix di nodi del cluster bare metal e nodi virtualizzati.
    - I cluster guest RHEL 5.6+ possono utilizzare fence_xvm o fence_scsi per il fencing del guest.
    - I cluster guest RHEL 6.1+ possono utilizzare fence_xvm (nel pacchetto **`fence-virt`**) o fence_scsi per il fencing del guest.

- Gli host KVM di RHEL 6.1+ devono utilizzare fence_virtd se il cluster guest utilizza fence_virt o fence_xvm come fence agent. Se il cluster guest utilizza fence_scsi, allora non sarà necessario utilizzare fence_virtd sugli host.
- fence_virtd può operare in tre modi:
  - In modalità standalone dove la mappatura host su guest ha una codifica fissa, e la migrazione live dei guest non è consentita
  - Uso del servizio Openais Checkpoint per controllare le migrazioni live dei guest clusterizzati. Per questa operazione eseguire il cluster host.
  - Utilizzo di Qpid Management Framework (QMF) reso disponibile dal pacchetto libvirt-qpid. Uso di QMF per controllare la migrazione dei guest senza la presenza di un cluster host completo.
- Lo storage condiviso può essere reso disponibile tramite i dispositivi a blocchi condivisi KVM o iSCSI, supportati da uno storage a blocchi dell'host o da un file backed storage (immagini raw).

- Red Hat Enterprise Virtualization Management (RHEV-M) versioni 2.2+ e 3.0 attualmente supportano guest clusterizzati RHEL 6.1+ e RHEL 5.6+.
  - I cluster guest devono essere omogenei (tutti guest RHEL 5.6+ o tutti guest RHEL 6.1+).
  - È permesso usare un mix di nodi del cluster bare metal e nodi virtualizzati.
  - Il fencing viene reso disponibili da fence_scsi in RHEV-M 2.2+ e da fence_scsi e fence_rhevm in RHEV-M 3.0, ed è supportato usando fence_scsi come di seguito descritto:
    - L'uso di fence_scsi con lo storage iSCSI è limitato ai server iSCSI che supportano SCSI 3 Persistent Reservation con il comando preempt-and-abort. Non tutti i server iSCSI supportano questa funzionalità. Consultare il rivenditore dello storage per controllare se il server è conforme al supporto SCSI 3 Persistent Reservation. Da notare che il server iSCSI disponibile con Red Hat Enterprise Linux, attualmente non supporta SCSI 3 Persistent Reservation e per questo motivo non è idoneo all'uso con fence_scsi.

- VMware vSphere 4.1, VMware vCenter 4.1, VMware ESX e ESXi 4.1 supportano i cluster guest in esecuzione, dove i sistemi operativi del guest sono RHEL 5.7+ o RHEL 6.2+. Anche la versione 5.0 di VMware vSphere, vCenter, ESX e ESXi è supportata; tuttavia, a causa di uno schema WDSL non completo fornito nella release iniziale di Vmware vSphere 5.0, l'utilità fence_vmware_soap non funziona con l'installazione predefinita. Consultare il Red Hat Knowledgebase https://access.redhat.com/knowledge/ per le procedure aggiornate per correggere questo problema.
  - I cluster guest devono essere omogenei (tutti guest RHEL 5.7+ o tutti guest RHEL 6.1+).
  - È permesso usare un mix di nodi del cluster bare metal e nodi virtualizzati.
  - L'agent fence_vmware_soap ha bisogno di VMware perl API di terze parti. Questo pacchetto software deve essere scaricato dal sito web di VMware e installato sui guest clusterizzati RHEL.
  - Alternativamente fence_scsi può essere usato per fornire il fencing come di seguito riportato.

- Lo storage condiviso può essere reso disponibile tramite i dispositivi a blocchi condivisi raw iSCSI o VMware.
- L'uso dei cluster guest VMware ESX è supportato tramite l'uso di fence_vmware_so_ap o fence_scsi.

- Attualmente non è supportato l'uso dei cluster guest Hyper-V.

7.2.1. Utilizzo storage condiviso iSCSI e fence_scsi

- In tutti gli ambienti di virtualizzazione sopra riportati lo storage iSCSI e fence_scsi possono essere utilizzati al posto dello storage nativo condiviso e dei dispositivi di fencing nativi.

- fence_scsi può essere usato per fornire il fencing I/O per lo storage condiviso disponibile attraverso iSCSI, se il target iSCSI supporta correttamente SCSI 3 persistent reservation e i comandi abort e preempt. Controllare con il rivenditore dello storage per determinare se la soluzione iSCSI desiderata è in grado di supportare le funzionalità sopra indicate.

- Il software del server iSCSI disponibile con RHEL non supporta SCSI 3 persistent reservation, per questo motivo non è possibile un suo utilizzo con fence_scsi. Al contrario, è possibile una sua implementazione come soluzione di archiviazione condivisa insieme ad altri dispositivi di fencing come fence_vmware o fence_rhevm.

- Se utilizzate fence_scsi su tutti i guest allara non sarà necessario utilizzare un cluster host (nei casi in cui viene utilizzato RHEL 5 Xen/KVM e RHEL 6 KVM Host)

- Utilizzando fence_scsi come fence agent, tutto lo storage condiviso deve essere composto da iSCSI. Non è permesso utilizzare un mix tra storage condiviso nativo e iSCSI.

7.2.2. Consigli generali

- Come precedentemente indicato è consigliato aggiornare sia gli host che i guest con gli ultimissimi pacchetti RHEL prima di utilizzare le funzionalità della virtualizzazione, poichè saranno disponibili numerose correzioni e miglioramenti.

- Non è supportato l'uso di un mix di piattaforme di virtualizzazione (hypervisor) sottostanti ai cluster guest. Tutti gli host sottostanti devono utilizzare la stessa tecnologia di virtualizzazione.

- Non è supportata l'esecuzione di tutti i guest in un cluster guest su un host fisico, poichè con questa impostazione non sarà possibile avere una elevata disponibilità in presenza di un errore dell'host. Tuttavia, è possibile utilizzare questa configurazione a scopo di sviluppo o per prototipi.

- Le procedure migliori includono:
  - Non è necessario avere un singolo host per guest anche se questa configurazione è in grado di fornre il livello più alto di disponibilità, poichè il fallimento di un host interesserà solo un singolo nodo nel cluster. In una mappatura 2 a 1 (due guest in un cluster per host fisico), il fallimento di un solo host risulterà nel fallimento di due guest. Per questo motivo è consigliato avere una mappatura con un rapporto vicinissimo a 1 a 1.
  - L'uso di un mix di cluster guest indipendenti sullo stesso insieme di host fisici non è al momento supportato se si utilizzano fence_xvm/fence_xvmd o fence_virt/fence_virtd.

- L'uso di un mix di cluster guest indipendenti con lo stesso insieme di host fisici potrà funzionare solo se utilizzate fence_scsi + iSCSI storage o fence_vmware + VMware (ESX/ESXi and vCenter).

- L'esecuzione come cluster guest di guest non-clusterizzati sullo stesso insieme di host fisici è supportata, tuttavia poichè gli host si isoleranno fisicamente a vicenda se è presente un cluster host, gli altri guest verranno terminati durante una operazione di fencing.

- Eseguire il provisioning dell'hardware dell'host in modo da evitare l'overcommit della memoria o della CPU virtuale. L'overcommit della memoria o della CPU virtuale abbasserà il livello delle prestazioni. Se le prestazioni raggiungono un livello critico, interessando anche l'heartbeat del cluster, si potrebbe verificare il fallimento del cluster.

APPENDICE A. DIARIO DELLE REVISIONI

Revisione 1-15.1 **Wed Feb 18 2015** **Francesco Valente**
Translation files synchronised with XML sources 1-15

Revisione 1-15 **Tue Dec 16 2014** **Steven Levine**
Aggiornato per l'implementazione di sort_order nella pagina di apertura di RHEL 6.

Revisione 1-13 **Wed Oct 8 2014** **Steven Levine**
Versione GA per Red Hat Enterprise Linux 6.6.

Revisione 1-12 **Thu Aug 7 2014** **Steven Levine**
Versione Beta per Red Hat Enterprise Linux 6.6

Revisione 1-11 **Fri Aug 1 2014** **Steven Levine**
Risolve: #852720
Piccoli problemi editoriali

Revisione 1-10 **Fri Jun 6 2014** **Steven Levine**
Bozza per Red Hat Enterprise Linux 6.6

Revisione 1-7 **Wed Nov 20 2013** **John Ha**
Release per il GA di Red Hat Enterprise Linux 6.5.

Revisione 1-4 **Mon Feb 18 2013** **John Ha**
Release per il GA di Red Hat Enterprise Linux 6.4.

Revisione 1-3 **Mon Jun 18 2012** **John Ha**
Release per il GA di Red Hat Enterprise Linux 6.3.

Revisione 1-2 **Fri Aug 26 2011** **John Ha**
Aggiornamento per la versione 6.2

Revisione 1-1 **Wed Nov 10 2010** **Paul Kennedy**
Release iniziale