



**Settimana del Software Libero  
26 Aprile – 1 Maggio 2004**

# **Clustering su Linux: introduzione e sistemi in HA (alta disponibilità)**

**Massimiliano Filacchioni (CASPUR)**  
<http://www.mfila.it/>



- Cluster: concetti di base
- HA
- I cluster per l'HA
- Linux-HA Heartbeat
- Ringraziamenti
- Commenti e domande

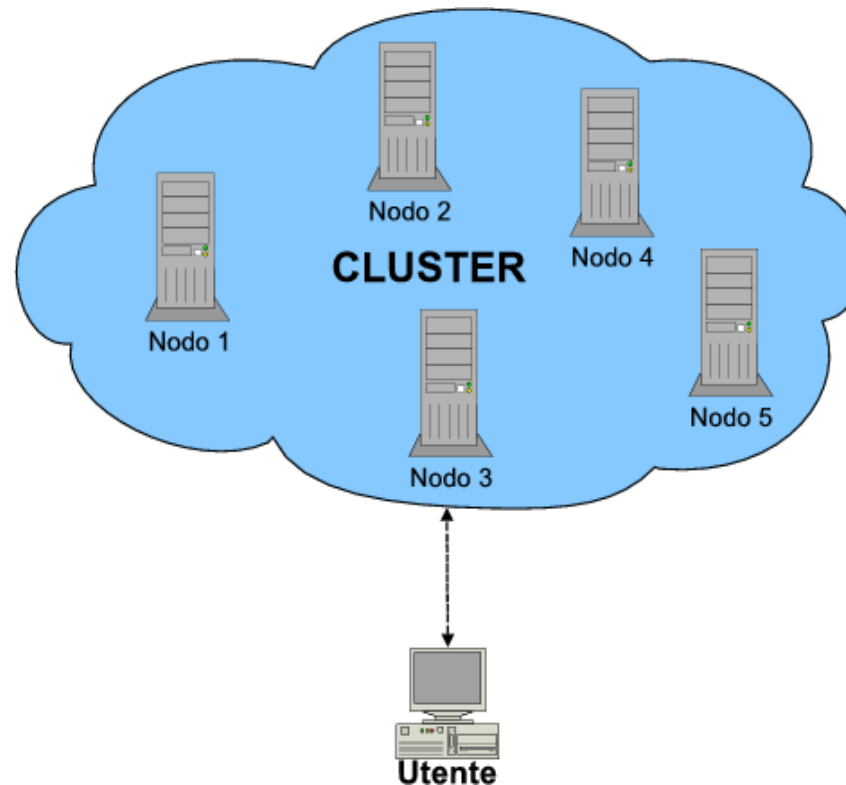


# Cluster: concetti di base



# Cos'è un cluster

- Un cluster è un insieme di computer tra loro interconnessi che vengono visti da chi li utilizza come un'unica entità
- I singoli computer che costituiscono un cluster prendono il nome di nodi





- In base alle funzionalità offerte si distinguono:
  - cluster per l'HA
  - cluster per il bilanciamento del carico
  - cluster per il calcolo intensivo

Le varie soluzioni di clustering disponibili possono fornire una o più delle suddette funzionalità

- In base all'architettura hw/sw dei nodi, invece, si distinguono:
  - cluster con architettura omogenea
  - cluster con architettura eterogenea



HA



# Definizione di HA (1 di 2)

- HA è l'acronimo di High Availability (alta disponibilità)
- Per HA si intende la capacità di un sistema informatico di essere costantemente utilizzabile dai propri utenti
- Un sistema in HA deve essere in grado di ridurre al minimo il tempo di disservizio (downtime) in seguito a:
  - guasti hardware
  - malfunzionamenti del Sistema Operativo
  - malfunzionamenti delle applicazioni
  - interventi di manutenzione (pianificati o meno)
  - errori umani
  - altri problemi



## Definizione di HA (2 di 2)

- Il fine ultimo dell'HA è quello di ottenere la disponibilità 24x7 (24 ore su 24, 7 giorni su 7) dei servizi offerti da un sistema informatico
- Tale fine ovviamente è estremamente difficile da raggiungere e nella maggior parte dei casi ci si accontenta di una sua buona approssimazione
- Per le applicazioni estremamente critiche (mission critical), comunque, è possibile ottenere, con ingenti investimenti economici, risultati molto prossimi alla disponibilità 24x7
- La dicitura 24x7 è un termine molto caro a chi si occupa di marketing e quindi spesso se ne fa un uso improprio! :-)



# La formula della disponibilità (1 di 2)

- Indicando con  $MTBF$  (Mean Time Between Failure) il tempo medio di funzionamento di un sistema prima che si verifichi un guasto, cioè la sua affidabilità (reliability)
- E con  $MTTR$  (Mean Time To Repair) il tempo medio di downtime a seguito di un qualsiasi malfunzionamento
- La disponibilità  $A$  del sistema può essere espressa mediante la seguente formula

$$A = \frac{MTBF}{MTBF + MTTR} \quad \text{con } A \in [0,1]$$

- Come è facile intuire l'HA si ottiene massimizzando  $MTBF$  e minimizzando  $MTTR$



# La formula della disponibilità (2 di 2)

- $A$  fornisce una misura del tempo di corretto funzionamento (uptime) di un sistema
- I valori limite di  $A$  sono:
  - 0, che indica un sistema mai funzionante
  - 1, che indica un sistema costantemente funzionante
- Valori più verosimili di  $A$  sono in genere maggiori di 0 e minori di 1



# Un'altra misura della disponibilità (1 di 2)

- Un'altra misura spesso utilizzata per esprimere il livello di disponibilità di un sistema è il così detto numero di 9
- Essa indica il numero di 9 presenti nella percentuale di uptime di un sistema nell'arco di 1 anno
  - non considerando valori inferiori al 90%
- Dato  $A$ , tale percentuale si ottiene moltiplicando  $A$  per 100
- Questa misura è un'altra passione di chi si occupa di marketing... di conseguenza spesso ne viene fatto un uso improprio! :-)



# Un'altra misura della disponibilità (2 di 2)

# di 9	% uptime	Tempo di downtime
1	90,00000%	37 giorni
2	99,00000%	3,7 giorni
3	99,90000%	8,8 ore
4	99,99000%	53 minuti
5	99,99900%	5,3 minuti
6	99,99990%	32 secondi
7	99,99999%	3 secondi

- È opinione abbastanza diffusa che l'HA richieda un numero di 9 non inferiore a 3



# Fattori che influenzano l'HA (1 di 2)

- Qualità dei componenti hw/sw dei sistemi
- Qualità delle infrastrutture utilizzate
  - locali
  - alimentazione elettrica
  - impianti di condizionamento
  - collegamenti di rete (interni e verso gli ISP)
  - ...
- Ridondanza
  - delle componenti critiche dei sistemi (alimentatori, schede di rete, ecc.)
  - dei sistemi nella loro interezza
  - delle componenti critiche delle infrastrutture



# Fattori che influenzano l'HA (2 di 2)

- Procedure operative
  - backup
  - disaster recovery
  - monitoraggio/allarmistica
  - ...
- Dispositivi di accesso rapido ai sistemi
  - KVM switch (analogici e IP)
  - console server
- Formazione del personale addetto alla gestione dei sistemi, dei servizi e delle infrastrutture
- L'HA delle suddette risorse umane! :-)

**Tutto ciò può costare molto denaro!**



- La ridondanza dei sistemi (insieme all'utilizzo di componenti hw/sw di alta qualità) è generalmente la prima soluzione che si adotta per aumentarne la disponibilità
- Ciò soprattutto perché, allo stato attuale, è la soluzione più economica
- La ridondanza dei sistemi consente, quando uno di questi si guasta, di sostituirlo con una sua copia identica
- L'operazione di sostituzione di un sistema con una sua copia prende il nome di switchover



# Tipi di switchover (1 di 2)

- Cold switchover
  - effettuato manualmente
  - elevato downtime
  - una sola copia del sistema ridondato è in linea
  - in molti casi costituisce una buona soluzione iniziale per aumentare la disponibilità dei sistemi
- Warm switchover
  - effettuato automaticamente
  - downtime minimo (spesso non percepibile)
  - tutte le copie del sistema ridondato sono in linea, ma solo una è operativa



# Tipi di switchover (2 di 2)

- Hot switchover
  - effettuato automaticamente
  - downtime praticamente inesistente
  - tutti i sistemi sono contemporaneamente in linea e operativi
  - è il tipo di switchover più complesso da realizzare
- Lo switchover automatico in caso di fallimento di un sistema prende il nome di failover



# Bilanciamento del carico e HA (1 di 2)

- Per bilanciamento del carico si intende la ripartizione del carico operativo di un unico sistema tra più sistemi
- È facile rendersi conto che tale meccanismo, oltre ad aumentare il volume di carico a cui i sistemi possono essere sottoposti, ne aumenta anche la disponibilità
- Nel caso di un semplice bilanciamento di carico uniforme tra  $n$  sistemi, di cui  $m$  non siano disponibili, la probabilità  $p$  che un utente interagisca con uno di questi ultimi è data da

$$p = \frac{m}{n} \quad \text{con } p \in [0,1]$$



# Bilanciamento del carico e HA (2 di 2)

- La probabilità  $q$  che interagisca con un sistema disponibile, invece, è data da

$$q = 1 - p = 1 - \frac{m}{n} = \frac{n - m}{n} \quad \text{con } q \in [0, 1]$$

- Ad esempio nel caso di due sistemi di cui uno non sia disponibile, l'utente avrà il 50% di probabilità ( $p = 0.5$ ) di interagire con quello non disponibile ed il 50% di probabilità ( $q = 0.5$ ) di interagire con quello disponibile
- Per quanto esposto non è raro che il bilanciamento del carico e l'HA siano integrati in un'unica soluzione



# HA prevista dai servizi

- Alcuni servizi sono pensati per fornire meccanismi di HA a livello architetturale
- Tra questi è possibile individuare
  - Il routing delle mail mediante SMTP che, mediante i record MX del DNS, prevede che ogni dominio possa disporre di più server di destinazione, utilizzati dagli altri server in sequenza (secondo la loro priorità) fino a quando non ne viene individuato uno disponibile
  - Lo stesso DNS che fornisce un meccanismo analogo per i server che ne determinano il funzionamento, mediante i record NS
- In questi casi l'HA è praticamente gratis (essa può essere ovviamente migliorata, ma gli sforzi richiesti, a differenza di altri casi, sono minimi)



# I cluster per l'HA



- I cluster per l'HA
  - Rendono trasparente agli utenti la ridondanza dei sistemi (che trova una sua naturale incarnazione nei nodi che li compongono)
  - Si occupano dell'operazione di failover
  - Consentono agli amministratori uno switchover controllato (utile per le operazioni di manutenzione dei nodi)
  - Forniscono spesso funzionalità aggiuntive quali
    - Monitoraggio dei servizi
    - Replicazione dei dati
    - ...



- Le risorse gestibili mediante un cluster per l'HA includono:
  - Indirizzi IP
  - Dispositivi di storage
  - Filesystem
  - Servizi
  - ...
- Per consentirne il failover, tali risorse devono essere gestite unicamente dal cluster (quindi non dal Sistema Operativo dei nodi)
- Ogni cluster può gestire uno o più gruppi di risorse



# Cluster per l'HA con warm failover

- Sono i cluster per l'HA più diffusi in quanto più semplici da implementare

Le soluzioni semplici sono sempre preferite a quelle complesse non tanto perché più economiche, quanto perché la complessità ostacola l'HA

- Funzionano con applicazioni tradizionali
- Ogni gruppo di risorse è operativo su un solo nodo alla volta
- Lo switchover (automatico o controllato) in questo tipo di cluster viene spesso indicato con il termine migrazione
- D'ora in poi ci occuperemo solo di cluster per l'HA con warm failover

# Funzionamento di un cluster per l'HA (1 di 7)

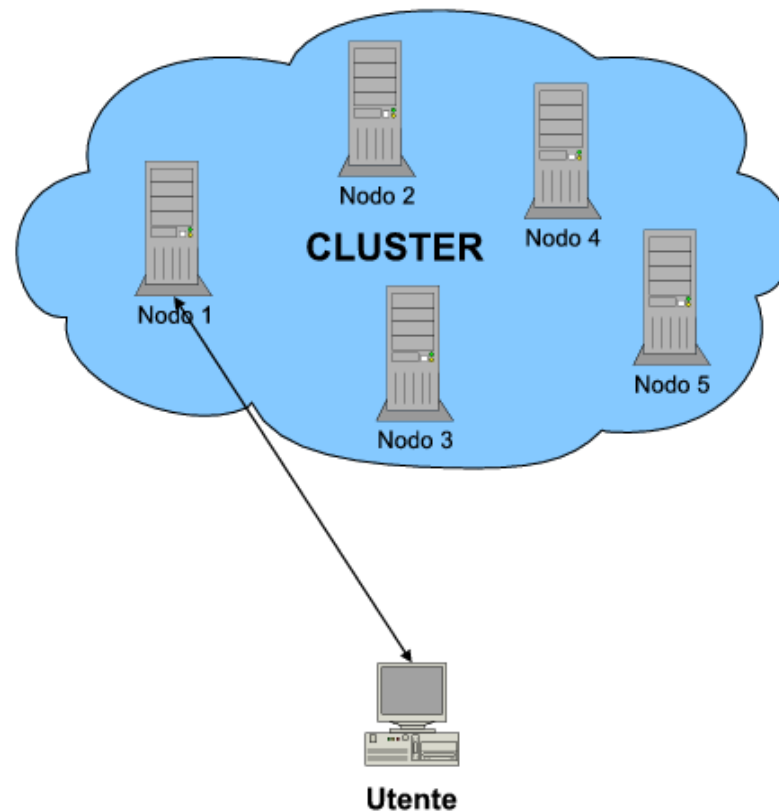
- L'unico nodo operativo rispetto ad un certo gruppo di risorse è denominato nodo attivo
- Gli altri nodi, rispetto al medesimo gruppo di risorse, sono denominati nodi in standby
- Il nodo designato come attivo rispetto ad un certo gruppo di risorse, quando tutti i nodi del cluster sono correttamente funzionanti, è denominato nodo primario
- Tutti gli altri, inizialmente in standby, nodi di backup
- I vari nodi si scambiano costantemente informazioni sul loro stato di salute (heartbeat)

# Funzionamento di un cluster per l'HA (2 di 7)

- Quando si verifica un malfunzionamento del nodo attivo si ha un failover (uno dei nodi in standby ne prende quindi il posto acquisendone le risorse)
- Lo switchover dal punto di vista del nodo che acquisisce le risorse prende il nome di takeover
- Lo switchover che consente al nodo primario di riacquisire le risorse quando torna disponibile, invece, prende il nome di failback
- I cluster spesso consentono il failback automatico quando il nodo primario torna disponibile
- Cerchiamo ora di visualizzare quanto esposto con un esempio

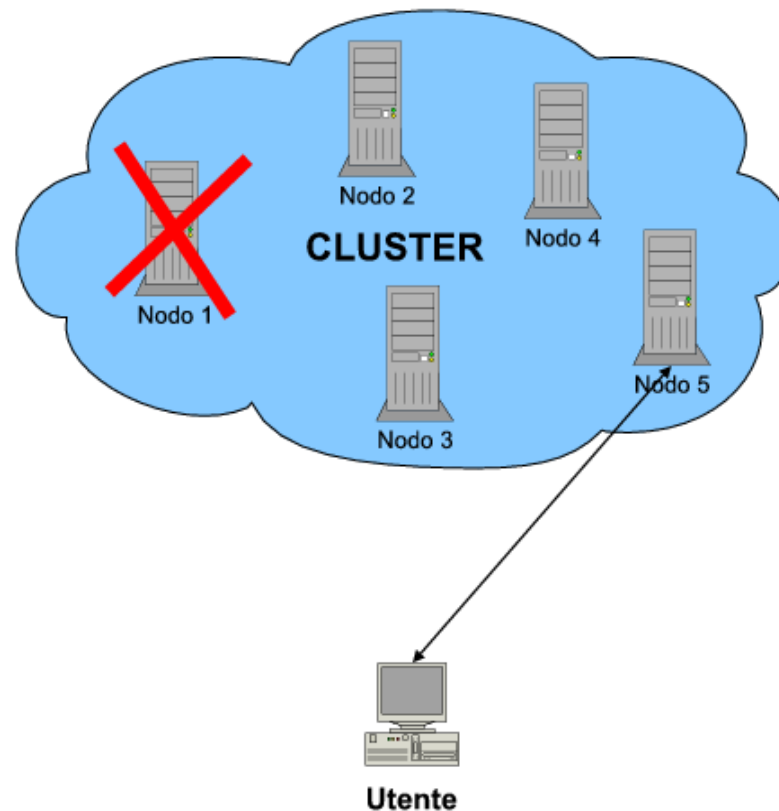
# Funzionamento di un cluster per l'HA (3 di 7)

- Quando tutti i nodi funzionano correttamente l'utente comunica unicamente col nodo primario (il nodo 1 in questo caso)



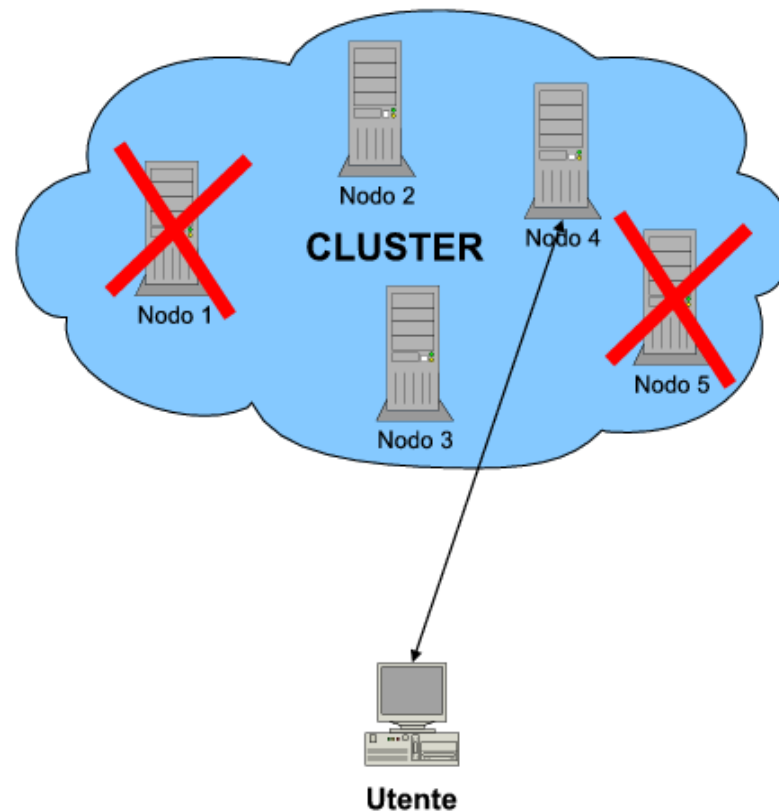
# Funzionamento di un cluster per l'HA (4 di 7)

- Nel momento in cui il nodo primario ha un problema di funzionamento, uno dei nodi in standby effettua il takeover prendendone il posto (il nodo 5 in questo caso)



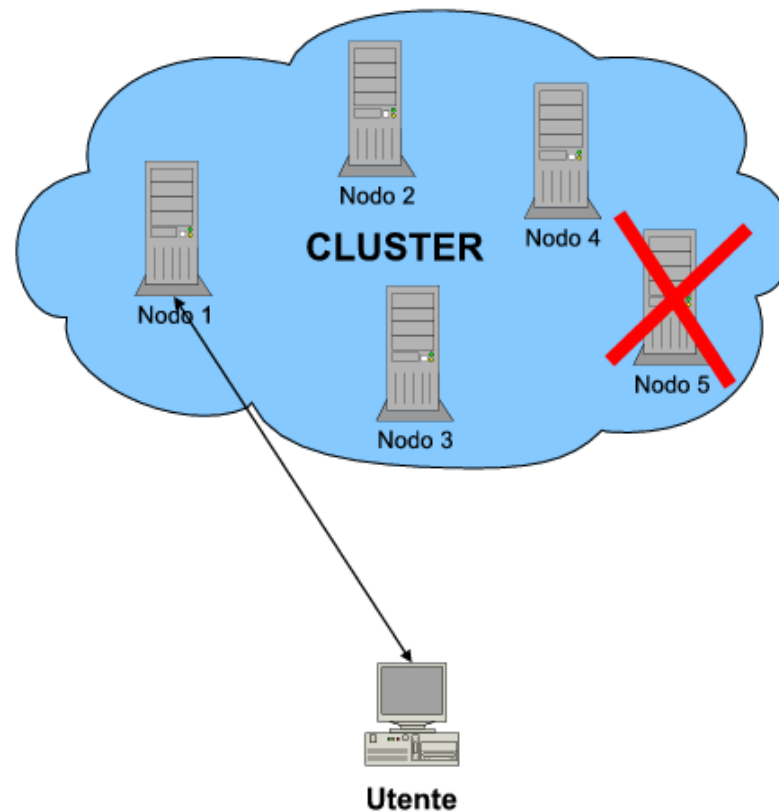
# Funzionamento di un cluster per l'HA (5 di 7)

- Tale comportamento continuerà a ripetersi in caso di problemi di funzionamento del nodo attivo (il takeover viene effettuato in questo caso dal nodo 4)



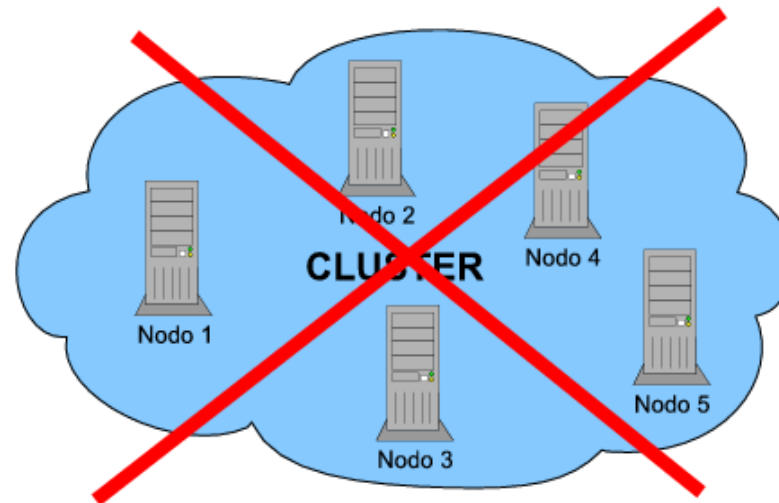
# Funzionamento di un cluster per l'HA (6 di 7)

- Se si utilizza il failback automatico, quando il nodo primario torna disponibile, riacquisisce le proprie risorse



# Funzionamento di un cluster per l'HA (7 di 7)

- Se si verificano problemi su tutti i nodi l'HA ovviamente non sussiste più (situazione che però dovrebbe essere evitata dall'intervento umano)





- Rappresenta le informazioni che i nodi di un cluster per l'HA si scambiano sul loro stato di salute
- È l'elemento chiave che consente il funzionamento di un tale cluster
- L'heartbeat può essere scambiato mediante vari canali di comunicazione, tipicamente collegamenti (link)
  - Seriali (nel caso di cluster di due nodi)
  - Di rete
- Tali canali sono generalmente dedicati per aumentarne l'affidabilità
- E ridondati (se ne utilizza cioè più d'uno) per aumentarne la disponibilità



# Le risorse IP (1 di 2)

- Queste risorse sono costituite dagli indirizzi IP mediante i quali i client utilizzati dagli utenti comunicano con i nodi attivi
- Tali indirizzi IP sono solitamente denominati IP virtuali
- Le reti a cui appartengono gli IP virtuali, invece, sono solitamente denominate reti di servizio
- Esistono varie tecniche che consentono il takeover degli IP virtuali, tra cui:
  - Indirizzi MAC multicast
  - Gratuitus ARP request/replay



## Le risorse IP (2 di 2)

- In alternativa è possibile effettuare il takeover dei nomi associati agli IP mediante la riconfigurazione dinamica del DNS



# Le risorse di storage (1 di 4)

- Queste risorse sono costituite dai dispositivi (dischi ed altri apparati) e dai filesystem, utilizzati per la memorizzazione dei dati condivisi dai nodi
- I dispositivi di storage possono essere
  - Locali ai singoli nodi con replicazione dei dati
    - In tempo reale
      - Sia a livello di filesystem, che a livello di blocchi
    - Ad intervalli regolari di tempo
      - Proponibile solo per dati relativamente statici
  - Fisicamente condivisi
  - Condivisi attraverso la rete
- Uno storage condiviso è ovviamente preferibile ad uno replicato



# Le risorse di storage (2 di 4)

- Come storage condiviso è possibile utilizzare
  - Dischi fisicamente condivisi (tipicamente da 2 nodi)
  - Soluzioni NAS (Network-Attached Storage)
  - Soluzioni SAN (Storage Area Network)
    - Fiber Channel
    - IP (basate su iSCSI o Internet SCSI e iFCP o Internet Fiber Channel Protocol)
    - Ethernet (basate su ATA-over-Ethernet)
- Le tecnologie più comunemente utilizzate per i dispositivi di storage sono
  - IDE (o ATA)
  - SCSI
  - Fiber Channel



## Le risorse di storage (3 di 4)

- Tali dispositivi sono solitamente utilizzati in configurazione ridondata per aumentarne la disponibilità (tipicamente RAID1 o RAID5)
- Per quanto riguarda i filesystem, invece, è possibile utilizzare:
  - Filesystem tradizionali (utilizzabili solo con dischi locali o condivisi da 2 nodi)
  - Filesystem di rete, ad esempio: NFS e CIFS (utilizzabili con dispositivi NAS)
  - Filesystem pensati specificamente per i cluster (denominati clustered filesystem), ad esempio: Veritas, GFS, StoreNext, ecc. (utilizzabili con dischi fisicamente condivisi e SAN)



# Le risorse di storage (4 di 4)

- Inoltre per semplificare la gestione dello storage spesso si utilizzano sistemi di gestione dei volumi (Volume Management System)
- Essi, infatti, facilitano: l'aggregazione, il ridimensionamento, la replicazione e il backup delle aree di storage (volumi)
- Tali sistemi sono integrati in alcuni clustered filesystem o disponibili come prodotti a se stanti



- I servizi che costituiscono le risorse di un cluster si distinguono in:
  - Servizi offerti all'utente (mediante appositi server)
    - Web
    - Mail
    - FTP
    - ...
  - Servizi di supporto
    - Caching dei nomi (associati ad indirizzi IP, agli utenti, ecc.)
    - Invio di notifiche all'amministratore del cluster in corrispondenza all'acquisizione o al rilascio di un gruppo di risorse
    - ...



# Tipologie di cluster per l'HA

- A seconda della configurazione dei nodi si distinguono le seguenti tipologie di cluster per l'HA
  - Nel caso di cluster di due nodi
    - Active/Standby (o asimmetrici)
    - Active/Active (o simmetrici)
  - N+1
  - N-to-1
  - N-to-M



# Cluster Active/Standby

- Prevedono, come suggerisce il nome, un nodo attivo e uno in standby
- Consentono di sfruttare solo uno dei due nodi disponibili
- Solitamente utilizzano uno storage fisicamente condiviso



# Cluster Active/Active

- Prevedono due nodi attivi (che risultano essere primari rispetto a diversi gruppi di risorse)
- Consentono di sfruttare entrambi i nodi
- Come i cluster Active/Standby, solitamente utilizzano uno storage fisicamente condiviso



- Prevedono N nodi attivi (che risultano essere primari rispetto a diversi gruppi di risorse) ed 1 nodo in standby (nodo di backup)
- Ognuno degli N nodi primari dispone di uno storage locale
- Tale storage è fisicamente condiviso con l'unico nodo di backup
- Lo svantaggio principale di tale tipo di cluster è costituito dall'elevato numero di connessioni (N) verso i dispositivi di storage di cui deve essere dotato il nodo di backup
- È ovvio, inoltre, che un cluster N+1 è in grado di far fronte al fallimento di uno solo degli N nodi primari



- Simili ai cluster N+1
- Prevedono N nodi primari rispetto a diversi gruppi di risorse ed 1 nodo di backup
- In questo tipo di cluster però lo storage è accessibile a tutti i nodi attraverso una SAN
- In questo modo viene eliminato lo svantaggio principale dei cluster N+1 (numero elevato di connessioni del nodo di backup allo storage)
- È ovvio, comunque, che anche un cluster N-to-1 è in grado di far fronte al fallimento di uno solo degli N nodi primari



- Rappresentano una generalizzazione dei cluster N-to-1
- Prevedono N nodi primari rispetto a vari gruppi di risorse e nessun nodo di backup
- In caso di fallimento di un nodo i suoi gruppi di risorse vengono presi in carico da un altro nodo
- Lo storage è accessibile a tutti i nodi attraverso una SAN
- Un cluster N-to-M è in grado di far fronte al fallimento di  $N - 1$  nodi



# L'integrità dei dati

- L'integrità dei dati è un aspetto importante in molte applicazioni, indipendentemente dall'HA
- A livello di filesystem il pericolo di danneggiamento dei dati può essere ridotto mediante
  - Il journaling (che consente di ripristinare una condizione consistente in seguito ad un crash di sistema)
  - Il locking (che consente di ridurre i rischi di inconsistenza legati agli accessi concorrenti ai dati)



- Il quorum è un meccanismo software che consente ad un nodo di utilizzare una risorsa condivisa solo se ottiene il voto favorevole dalla maggioranza dei nodi del cluster
- Tale meccanismo consente di garantire l'integrità dei dati quando tutti i nodi del cluster funzionano correttamente
- Esso è praticabile solo in cluster con un numero dispari di nodi (affinché sia possibile raggiungere una maggioranza di voti)
- Non può essere quindi utilizzato con soli due nodi
- A ciò si può comunque porre rimedio introducendo nel cluster un dispositivo utilizzato solo per le votazioni (una sorta di ulteriore nodo fittizio)



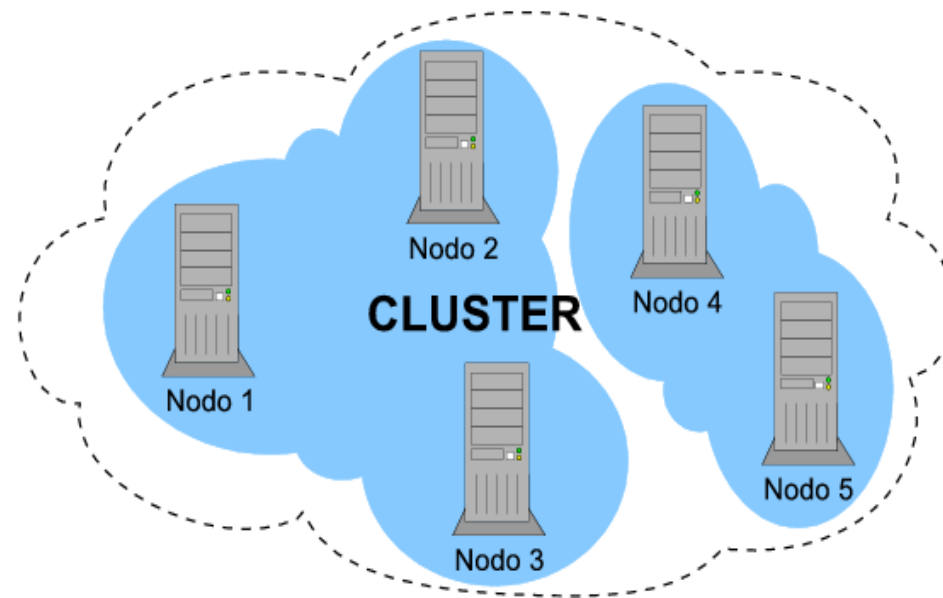
# Split-brain syndrome (1 di 2)

- La split-brain syndrome si verifica in un cluster per l'HA quando i suoi nodi, pur funzionando correttamente, non riescono più a comunicare tra loro (perdita dell'heartbeat)
- In tale situazione si vengono a creare più partizioni isolate di nodi tra loro comunicanti, ognuna delle quali “pensa” di essere l'unica operativa
- Questa situazione è molto pericolosa rispetto alla gestione delle risorse condivise (in particolare lo storage)
- In questa situazione, infatti, una sola scrittura su uno storage condiviso può essere catastrofica in termini di integrità dei dati



# Split-brain syndrome (2 di 2)

- L'incidenza della split-brain syndrome si riduce ridondando i canali utilizzati per l'heartbeat
- Essa comunque non è eliminabile del tutto
- Nella seguente figura viene mostrato un esempio di split-brain syndrome che da origine a due partizioni di nodi





# Soluzioni per la split-brain syndrome

- Il quorum, quando utilizzabile, non consente di risolvere il problema della split-brain syndrome in tutte le circostanze possibili
- Nelle circostanze in cui è in grado di risolverlo, inoltre, il raggiungimento di una maggioranza potrebbe richiedere troppo tempo (anche alcuni secondi)
- Tempo durante il quale si potrebbero verificare danni irreparabili dei dati
- Per ottenere una maggiore protezione dei dati si può far ricorso (eventualmente in aggiunta al quorum) al resource fencing



# Resource fencing

- Il resource fencing è un meccanismo hardware che consente ad un nodo, in caso di split-brain syndrome, di garantirsi l'uso esclusivo delle risorse condivise
- Esistono due tipi di resource fencing
  - A livello di dispositivo, ad esempio
    - I meccanismi di reserve/lease di SCSI
    - Gli switch Fiber Channel
  - A livello di sistema
    - STONITH



- STONITH è l'acronimo di Shoot The Other Node In The Head
- È il tipo di resource fencing più semplice ed efficace
- Consente ad un nodo, in caso di assenza dell'heartbeat, di togliere l'alimentazione ai nodi ritenuti isolati, in modo da garantirsi l'uso esclusivo alle risorse condivise
- Richiede l'impiego di speciali alimentatori controllabili mediante rete o linea seriale
- È un meccanismo piuttosto cruento, ma decisamente efficace! :-)



# Il pericolo associato all'uso di STONITH

- Vi è un unico pericolo associato all'uso di STONITH
- Che i nodi del cluster si “uccidano” a vicenda
- Tale situazione comunque si manifesta in rarissime circostanze
- L'incidenza di queste circostanze può essere ulteriormente ridotta mediante stratagemmi quali
  - Imporre ai nodi di attendere un tempo diverso prima di avviare la procedura
  - Utilizzare alimentatori che consentano ad un solo nodo alla volta di impartire comandi di controllo
- A favore dell'uso di STONITH va notato che spesso l'integrità dei dati è ritenuta più importante della stessa HA



# Effetto ping pong

- Con tale nome si indica il continuo switchover dei nodi di un cluster per l'HA che si può verificare in particolari circostanze operative
- A seconda delle circostanze, si possono utilizzare diverse strategie per ridurre (se non eliminare) quest'effetto



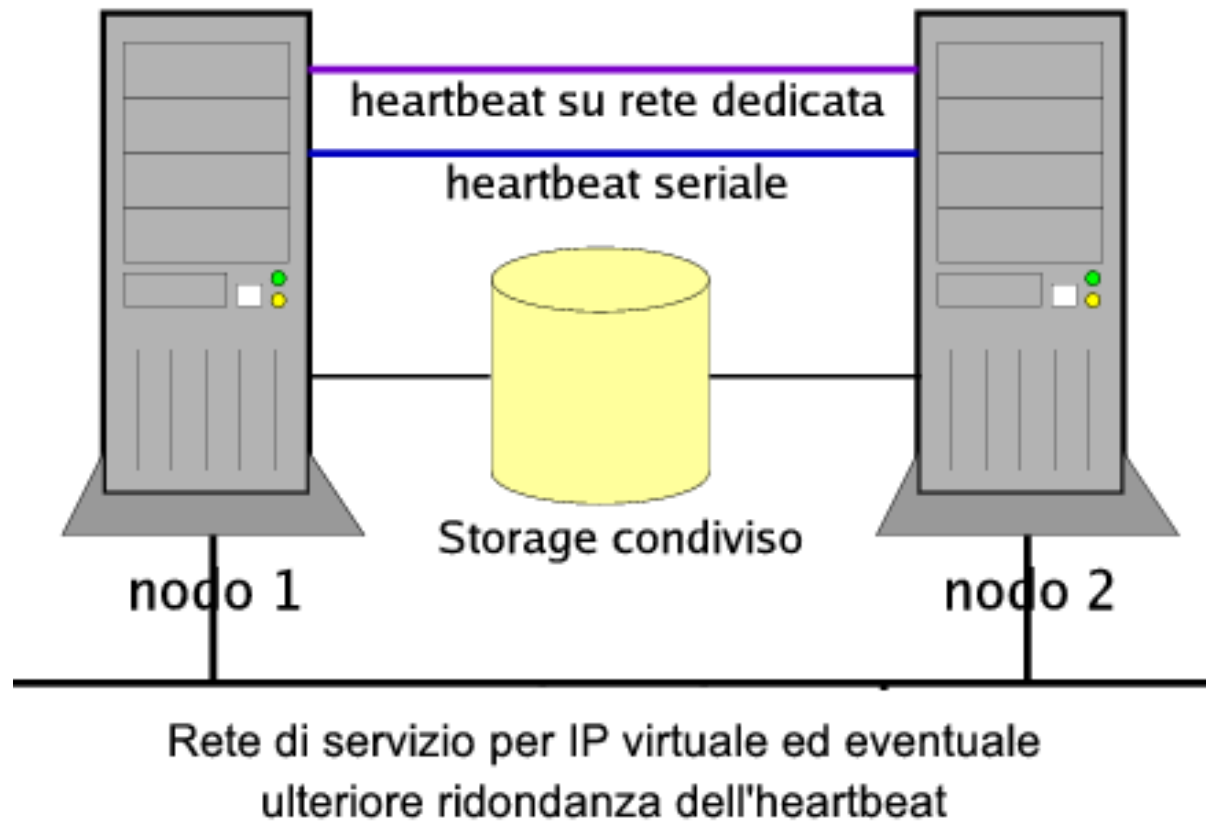
# Cluster di due nodi

- I cluster di due nodi sono i più diffusi in quanto più semplici da implementare
- Anche in questo caso vale quanto affermato relativamente al warm switchover e cioè

Le soluzioni semplici sono sempre preferite a quelle complesse non tanto perché più economiche, quanto perché la complessità ostacola l'HA



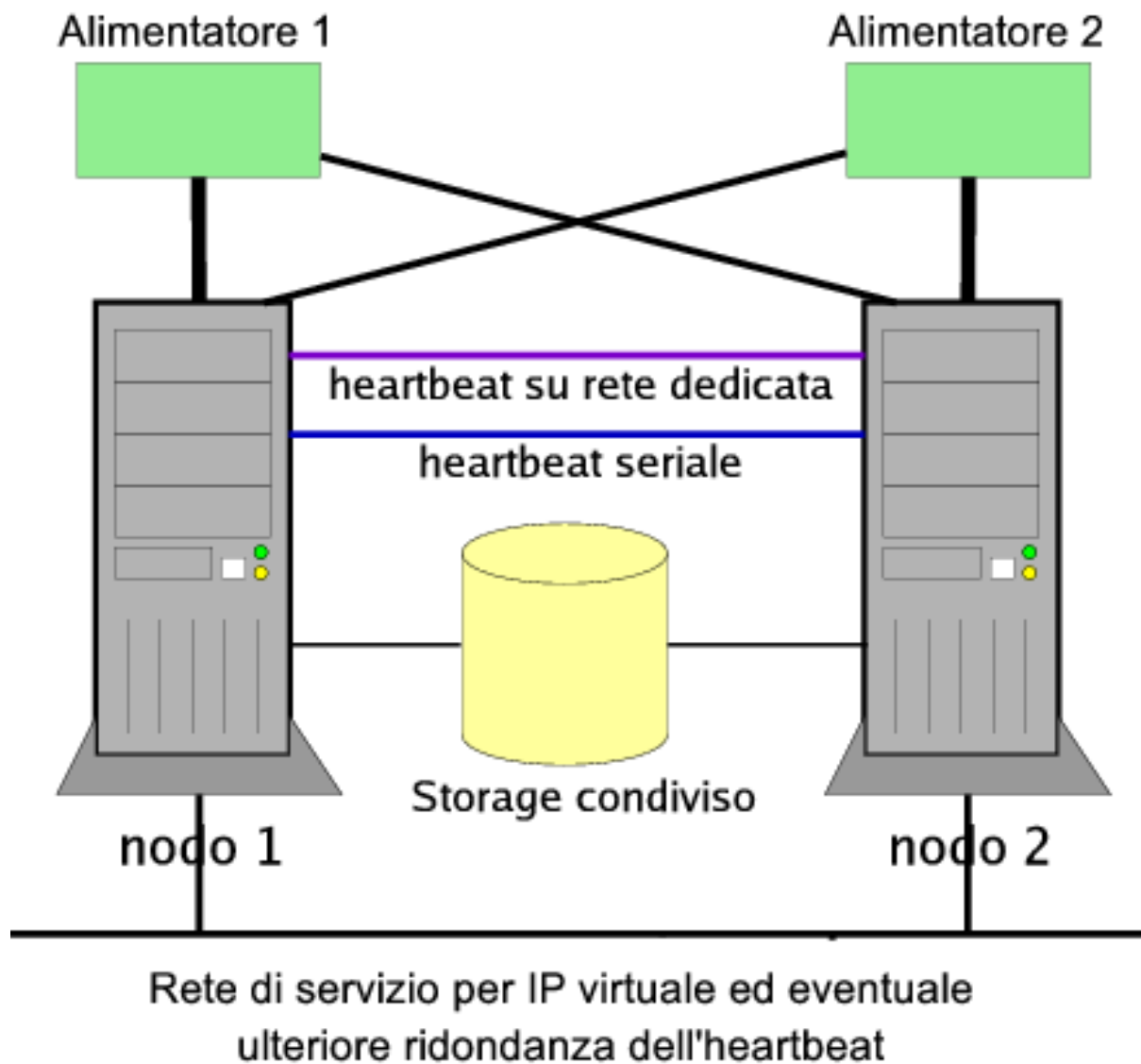
# Schema di un tipico cluster di 2 nodi



- Come collegamento di rete dedicato all'heartbeat si usa generalmente un cavo incrociato



# Tipico cluster di 2 nodi con STONITH





# Linux-HA Heartbeat

<http://linux-ha.org/>



- Heartbeat è un software open source (distribuito con licenza GPL) per la realizzazione di cluster per l'HA di 2 nodi in ambiente Linux
- Nasce nell'ambito del progetto Linux-HA
- È però utilizzabile anche in ambienti FreeBSD e Solaris
- È alla base di varie altre soluzioni per l'HA

# Distribuzioni Linux che includono Heartbeat





# Caratteristiche di Heartbeat (1 di 3)

- Heartbeat supporta
  - IP takeover mediante gratuitus ARP
  - Scambio dell'heartbeat mediante
    - Collegamento seriale
    - Collegamenti di rete (con protocollo UDP)
  - Configurazioni dei nodi in modalità
    - active/standby
    - active/active
  - Failback automatico
  - Failover scatenato dall'irraggiungibilità (mediante ping) di uno o più host remoti
    - Utile per individuare problemi a livello di schede di rete, cavi, switch ed altri apparati



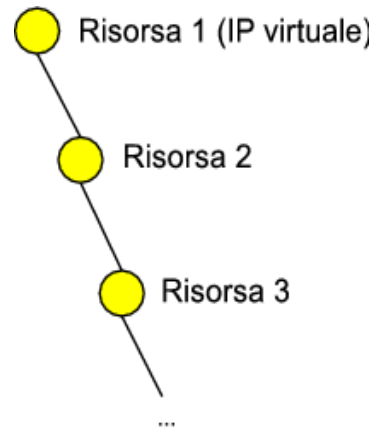
# Caratteristiche di Heartbeat (2 di 3)

- Gli host monitorati, comunque, devono essere il quanto più possibile “stabili” (tipicamente dei router)
- STONITH
  - È possibile utilizzare vari modelli di alimentatori
  - Ed una procedura software basata su SSH (ovviamente meno affidabile)
- API (Application Programming Interface) per la realizzazione di applicazioni in grado di interagire con il cluster (applicazioni cluster aware)
- Le singole risorse vengono controllate mediante script di tipo System V (i classici script presenti nella directory di sistema `/etc/init.d`)



# Caratteristiche di Heartbeat (3 di 3)

- Le risorse appartenenti ad ogni gruppo sono organizzate in una semplice gerarchia lineare



- Heartbeat non supporta il monitoraggio delle risorse (e di conseguenza il failover scatenato da problemi ad esse relativi)
  - Funzionalità comunque implementabile utilizzando altri prodotti
- È caratterizzato da un'architettura estremamente modulare ed efficiente



# Come controllare l'esecuzione di Heartbeat

- Dopo aver installato e configurato Heartbeat su entrambi i nodi, se ne può controllare l'esecuzione mediante lo script `/etc/init.d/heartbeat`
- Alcuni dei parametri riconosciuti da questo script sono
  - `start`
    - Che consente di avviare l'esecuzione di Heartbeat
  - `stop`
    - Che consente di fermarla
  - `status`
    - Che restituisce informazioni sullo stato di funzionamento di Heartbeat



# I file di configurazione di Heartbeat (1 di 2)

- `/etc/ha.d/ha.cf`
  - Contenente parametri di configurazione generali
- `/etc/ha.d/authkeys`
  - Contenente le direttive per l'autenticazione dell'heartbeat
- `/etc/ha.d/haresources`
  - Contenente i vari gruppi di risorse, ognuno associato al relativo nodo primario
  - Ogni gruppo di risorse è costituito dall'elenco delle risorse che lo compongono
  - Ogni risorsa, invece, è rappresentata dal nome del corrispondente script di controllo e gli eventuali parametri da esso richiesti



# I file di configurazione di Heartbeat (2 di 2)

- A ciò fa eccezione la risorsa IP virtuale (la prima di ogni gruppo) rappresentabile mediante l'IP stesso
- I vari script di controllo devono risiedere in una delle directory
  - `/etc/init.d`
  - `/etc/ha.d/resource.d`
- Un qualsiasi servizio di sistema può quindi essere una risorsa
- La seconda directory ospita gli script di controllo distribuito con Heartbeat
- I suddetti file di configurazione, a meno del valore di alcuni parametri, devono essere identici su entrambi i nodi



# Gestione delle risorse durante il failover

- Durante il failover Heartbeat effettua
  - Lo stop delle risorse sul nodo che le cede, nell'ordine inverso rispetto a quello in cui sono indicate in `/etc/ha.d/haresources`
  - Lo start delle risorse sul nodo che le acquisisce, nell'ordine in cui sono indicate in tale file
- Lo start è effettuato sempre dopo lo stop nel caso i nodi siano in grado di comunicare tra loro (cosa non possibile se il nodo attivo ha problemi di funzionamento seri)
- Dalla versione 1.2.1, Heartbeat forza il reboot del nodo che cede le risorse qualora lo stop fallisca (utile per riportare il nodo in uno stato coerente)

# Considerazioni sulla configurazione dei nodi

- Gli IP virtuali non devono essere assegnati direttamente alle interfacce di rete del sistema
- I servizi messi in HA non devono essere fatti partire automaticamente al boot del sistema
- Se si utilizza uno storage condiviso
  - il corrispondente FS non deve essere montato automaticamente dal sistema
  - È consigliabile memorizzare su tale FS non solo i dati utilizzati dai servizi messi in HA, ma anche (quando possibile) i relativi file di configurazione
- È opportuno associare gli indirizzi IP dei nodi e gli IP virtuali ai relativi nomi, anche mediante il file `/etc/hosts` (per non dipendere dal DNS per la risoluzione di tali nomi)



# Prodotti utilizzabili con Heartbeat (1 di 2)

- Mon
  - Per il monitoraggio delle risorse
  - <http://www.kernel.org/software/mon/>
- DRDB
  - Per la replicazione automatica dei blocchi di un dispositivo di storage attraverso la rete
  - <http://www.drbd.org/>
- rsync
  - Per la replicazione dei dati a livello di filesystem
  - <http://www.samba.org/rsync/>



# Prodotti utilizzabili con Heartbeat (2 di 2)

- LVS (Linux Virtual Server)
  - Per la gestione di cluster per il bilanciamento del carico
  - Heartbeat può essere utilizzato per l'HA del bilanciatore di LVS
  - <http://www.linuxvirtualserver.org/>
- Webmin
  - Interfaccia Web per l'amministrazione di sistema (contenente un modulo specifico per la configurazione di Heartbeat)
  - <http://www.webmin.com/>
- ...



## Un esempio (1 di 8)

- Di seguito viene riportato un esempio di configurazione di Heartbeat che consente di mettere in HA un server Web Apache
- Si supporrà che
  - Il nome dei due nodi sia `web1.example.com` (nodo primario) e `web2.example.com`
  - L'IP virtuale sia `10.10.10.10`
  - Sia utilizzato come storage un disco SCSI fisicamente connesso ad entrambi i nodi
  - Il relativo filesystem (nella partizione `/dev/sdb1`) venga montato sulla directory `/shared`
  - L'hearbeat sia scambiato mediante un collegamento seriale, uno di rete dedicato e la rete di servizio



## Un esempio (2 di 8)

```
/etc/ha.d/ha.cf ...
```

```
# parametro per syslog
Logfacility local7

# tempi inerenti l'heartbeat (in sec)
keepalive 1      # intervallo di scambio
warntime 2       # tempo di allarme
deadtime 10      # tempo di failover

# canali per l'heartbeat
bcast eth0 eth1  # eth0 nic di serv.
                  # eth1 nic dedicata
serial /dev/ttyS0 # link seriale

# attivazione del failback automatico
auto_failback on
```



```
... /etc/ha.d/ha.cf
```

```
# configurazione del failover per
# problemi sull'interfaccia di
# servizio
ping 10.10.10.1 # IP del router
respawn hacluster \
    /usr/lib/heartbeat/ipfail
apiauth ipfail uid=hacluster
```

- I nomi dei nodi devono essere quelli restituiti dal comando `uname -n`



```
/etc/ha.d/authkeys
```

```
Auth 1  
1 sha1 Segreto1234abcd
```

- Segreto1234abcd rappresenta la chiave utilizzata dal meccanismo di autenticazione dell'heartbeat selezionato (hash sha1)
- In aggiunta a questo, sono supportati altri due meccanismi di autenticazione basati su
  - Hash md5
  - CRC, che non richiede nessuna chiave (consigliabile solo se non si utilizza la rete di servizio come canale per lo scambio dell'heartbeat)



## Un esempio (5 di 8)

```
/etc/ha.d/haresources
```

```
web1.example.com 10.10.10.10 \  
Filesystem::/dev/sdb1::/shared::ext3 \  
httpd
```

- `Filesystem` è la risorsa rappresentante un FS, il relativo script di controllo viene distribuito con Heartbeat e consente il mount di tale FS
- `httpd` è la risorsa Apache, il relativo script di controllo viene fornito con praticamente tutte le distribuzioni Linux
- Anche in questo caso il nome del nodo primario deve essere quello restituito da `uname -n`



## Un esempio (6 di 8)

- A questo punto su entrambi i nodi
  - Fermare Heartbeat, se in esecuzione, mediante il comando

```
/etc/init.d/heartbeat stop
```

- Evitare che il sistema effettui il mount di `/shared` al boot, eliminando da `/etc/fstab` il riferimento a tale directory oppure utilizzando l'opzione di mount `noauto`
- Evitare che Apache sia gestito dal sistema mediante il comando

```
chkconfig --del httpd
```



## Un esempio (7 di 8)

- Inserire in `/etc/hosts` le informazioni relative all'IP virtuale aggiungendovi la seguente riga

```
10.10.10.10    web.example.com    web
```

- Effettuare il mount di `/shared` mediante il comando

```
mount /dev/sdb1 /shared
```

- Spostare la directory `/var/www` (contenente i dati utilizzati da Apache) in `/shared` e sostituirla con un link simbolico a `/shared/www` mediante i comandi

```
mv /var/www /shared  
ln -s /shared/www /var/www
```



## Un esempio (8 di 8)

- Effettuare l'unmount di `/shared` mediante il comando

```
umount /shared
```

- Avviare l'esecuzione di Heartbeat mediante il comando

```
/etc/init.d/heartbeat start
```

- Incrociare le dita!!! :-)



# Possibili evoluzioni dell'esempio

- Spostare in `/shared` anche i file di configurazione di Apache
- Aggiungere la risorsa `LinuxSCSI` prima di `Filesystem`, per far sì che i nodi vedano il disco condiviso solo quando necessario
- Aggiungere la risorsa `MailTo` per inviare via email le notifiche di failover all'amministratore del cluster
- Mettere in HA altri servizi utilizzabili con Apache (ad esempio MySQL)
- Utilizzare Mon per il monitoraggio delle risorse
- Configurare STONITH



# Alcuni comandi distribuiti con Heartbeat

- `hb-standby`
  - Che consente al nodo attivo di cedere le risorse da esso gestite
- `hb-takeover` (disponibile dalla versione 1.2.1)
  - Che consente al nodo in standby di acquisire le risorse gestite dal nodo attivo
- `stonith`
  - Che consente di controllare il meccanismo di STONITH

# Confronto fra Heartbeat e LifeKeeper (1 di 3)

- LifeKeeper è la soluzione commerciale per l'HA della società SteelEye Technology (disponibile in ambiente Linux, Windows e Solaris)
  - <http://www.steeleye.com/>
- LifeKeeper supporta
  - Condivisione delle risorse di storage tra due nodi
  - Configurazioni active/standby e active/active di tali nodi
  - Scenari di failover complessi mediante cluster di più nodi
  - Resource fencing dello storage mediante i meccanismi di reserve/lease di SCSI
  - Monitoraggio delle risorse

# Confronto fra Heartbeat e LifeKeeper (2 di 3)

- Una GUI Java per l'amministrazione del cluster
- Meccanismi di replicazione dei dati
- Ogni risorsa è gestita da un insieme di comandi (tipicamente script) denominati ARK (Application Recovery Kit)
  - I vari ARK sono disponibili a pagamento
  - Oppure possono essere sviluppati in proprio, mediante un kit di sviluppo software (SDK) open source

# Confronto fra Heartbeat e LifeKeeper (3 di 3)

- Nella mia attività lavorativa ho avuto modo di utilizzare sia LifeKeeper, che Heartbeat
- È mia opinione che, per l'implementazione di un cluster per l'HA di due nodi, con Heartbeat e l'ausilio di altri prodotti di supporto, si possano ottenere dei risultati del tutto analoghi a quelli ottenibili con LifeKeeper
- Disponendo ovviamente di conoscenze adeguate
- Conoscenze che, in scenari di una certa complessità, sono comunque necessarie indipendentemente dalla soluzione scelta



# Ringraziamenti

Ringrazio Virginia Calabritto e Leonardo Valcamonici per la revisione di questo materiale e molti altri per ragioni puramente personali! :-)



# Commenti e domande

