ISSN 1590-2595
Anno 2009_Numero 74

Tecniche di Alta Disponibilità per l'acquisizione di dati sismici in ambiente GNU/Linux: un'applicazione alla rete sismica di Stromboli

Quaderni di Geofisica

74

Istituto Nazionale di
Geofisica e Vulcanologia

Quaderni di Geofisica

Segreteria di Redazione
Francesca Di Stefano - coordinatore
Tel. +39 06 51860068
Fax +39 06 36915617

Rossella Celi
Tel. +39 06 51860055
Fax +39 06 36915617
redazionecen@ingv.it

ISSN 1590-2595
Anno 2009_Numero 74

Tecniche di Alta Disponibilità per l'acquisizione di dati sismici in ambiente GNU/Linux: un'applicazione alla rete sismica di Stromboli

High-Availabilty techniques for seismic data acquisition using GNU/Linux: an application to the Stromboli seismic network

Rosario Peluso, Ciro Buonocunto, Antonio Caputo, Walter De Cesare, Massimo Orazi, Giovanni Scarpato

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Napoli - Osservatorio Vesuviano)

in copertina Rack contenente macchine di acquisizione dati (particolare)

Tecniche di Alta Disponibilità per l'acquisizione di dati sismici in ambiente GNU/Linux: un'applicazione alla rete sismica di Stromboli

In questo articolo viene descritto l'utilizzo di tecniche di alta disponibilità per l'acquisizione di dati sismici nell'Isola di Stromboli. Tali tecniche consentono di eliminare o almeno ridurre i tempi di indisponibilità di un sistema informatico in caso di rottura totale o parziale di una sua parte. Applicate ad un sistema di acquisizione dati, esse consentono di evitare lunghi periodi di mancanza dei dati stessi prima di poter intervenire fisicamente sul sistema per le riparazioni del caso.

This article describes the implementation of high-availability tecniques for seismic data acquisition on the Stromboli Island. These tecniques help to eliminate or at least reduce time unavailability of a computing system in case of a global of partial breakage. When applied to a data acquisition system, they allow to avoid long periods of data leakage before any human intervention on the system for its repair.

Introduzione

L'alta disponibilità (*High-Availability*, HA) coinvolge tecniche volte ad assicurare un elevato grado di continuità operazionale di un sistema informatico. Un sistema non disponibile, che non possa essere utilizzato o a cui non sia possibile accedere è anche detto essere in *downtime*.

Occorre distinguere tra *downtime* programmati e *downtime* non programmati. I primi sono tipicamente il risultato di un'operazione di manutenzione che necessiti di un'interruzione delle operazioni del sistema e che non possa essere evitata nell'ambito del sistema installato. Un esempio di *downtime* programmato può essere rappresentato dalla necessità di modifiche *software* o di configurazione che richiedano un riavvio del sistema. Un *downtime* non programmato ha origine generalmente da un evento fisico, una *failure* dell'*hardware* o del *software* o un'anomalia ambientale. Mancanze di corrente elettrica, componenti (RAM, CPU, dischi, etc.) che si rompono, temperature elevate che impediscano il funzionamento, connessioni di rete interrotte a livello fisico o logico, violazioni catastrofiche della sicurezza oppure *failure* di una qualsiasi componente *software*, tutte rientrano nel novero dei *downtime* non programmati.

Molti fornitori di servizi internet excludono i *downtime* programmati dai calcoli di disponibilità assumendo, più o meno correttamente, che questi ultimi abbiano un impatto trascurabile sulla comunità di utenti dello specifico servizio. Escludendo questo tipo di *downtime*, molti sistemi possono sostenere di avere una disponibilità incredibilmente elevata. Sistemi che offrano una tale disponibilità continua sono però abbastanza rari e costosi e spesso implementano speciali soluzioni per eliminare ogni punto singolo di rottura (*single point of failure*, SPOF) e che consentano riparazioni, aggiornamenti e sostituzioni dell'*hardware*, della rete, del sistema operativo, del *software* e di quant'altro componga il sistema stesso pur lasciandolo in funzione.

Il tempo di ripristino di un sistema, cioè il tempo totale necessario a recuperare da una indisponibilità programmata o non programmata, è un concetto strettamente legato a quello di disponibilità. Tale tempo può al limite diventare infinito (cioè potrebbe non essere possibile il recupero completo) con certe combinazioni di sistemi e *failure*. Ad esempio un alluvione o un incendio che distruggano completamente un *data center* e i suoi sistemi quando questo non ne

abbia un altro secondario (*disaster recovery*).
Un altro concetto collegato è quello dell'affidabilità dei dati, cioè di quanto fedelmente i sistemi di archiviazione conservano e riportano i dati e le transazioni effettuate su di essi. In genere questo argomento viene trattato separatamente per determinare una perdita accettabile (o effettiva) a seconda dei vari eventi di *failure*. Talvolta può essere tollerata una interruzione del servizio ma non una perdita di dati.
Paradossalmente aggiungere componenti ad un sistema può in effetti diminuire la disponibilità del sistema stesso. Questo perché all'aumentare della complessità aumentano anche i punti ed i modi di rottura e, inoltre, un sistema complesso è intrinsecamente più difficile da implementare correttamente. I sistemi a più alta disponibilità hanno di solito uno schema di progetto semplice: un singolo sistema di alta qualità con ridondanza interna che implementa tutte le funzioni accoppiato ad un secondo sistema in un luogo separato.

1. L'acquisizione della rete sismica di Stromboli

La rete sismica dell'Isola di Stromboli è attualmente composta da 13 stazioni digitali [De Cesare et al., 2009] a larga banda equipaggiate con tre diversi tipi di digitalizzatori (*Gaia 1* e *2* in versione base [Salvaterra et al., 2008] e *Gilda* [Orazi et al., 2006, 2008]) e da 2 stazioni dilatometriche. I dati da esse prodotti vengono trasmessi in tempo reale verso i due centri di acquisizione situati a Stromboli presso il Centro Operativo Avanzato (COA) del Dipartimento di Protezione Civile (dPC) e a Lipari presso l'Osservatorio dell'Istituto Nazionale di Geofisica e Vulcanologia. La conformazione fisica dell'isola rende infatti impossibile la trasmissione diretta via radio verso un unico luogo.
Come mostrato in Figura 1 la trasmissione è così organizzata:

1. Quattro stazioni sul versante settentrionale (in viola in figura) trasmettono tramite *radiomodem* direttamente al COA.
2. Cinque stazioni sul versante meridionale (in rosso in figura) trasmettono sempre tramite *radiomodem* verso l'isola di Lipari.
3. Le restanti quattro stazioni (in blu in figura) utilizzano la rete senza fili installata sull'isola dalla Sezione di Napoli dell'INGV che le mette in comunicazione con il COA [De Cesare et al., 2009].

1.1 La precedente implementazione del sistema di acquisizione

In origine ognuna delle tre sottoreti veniva acquisita da un

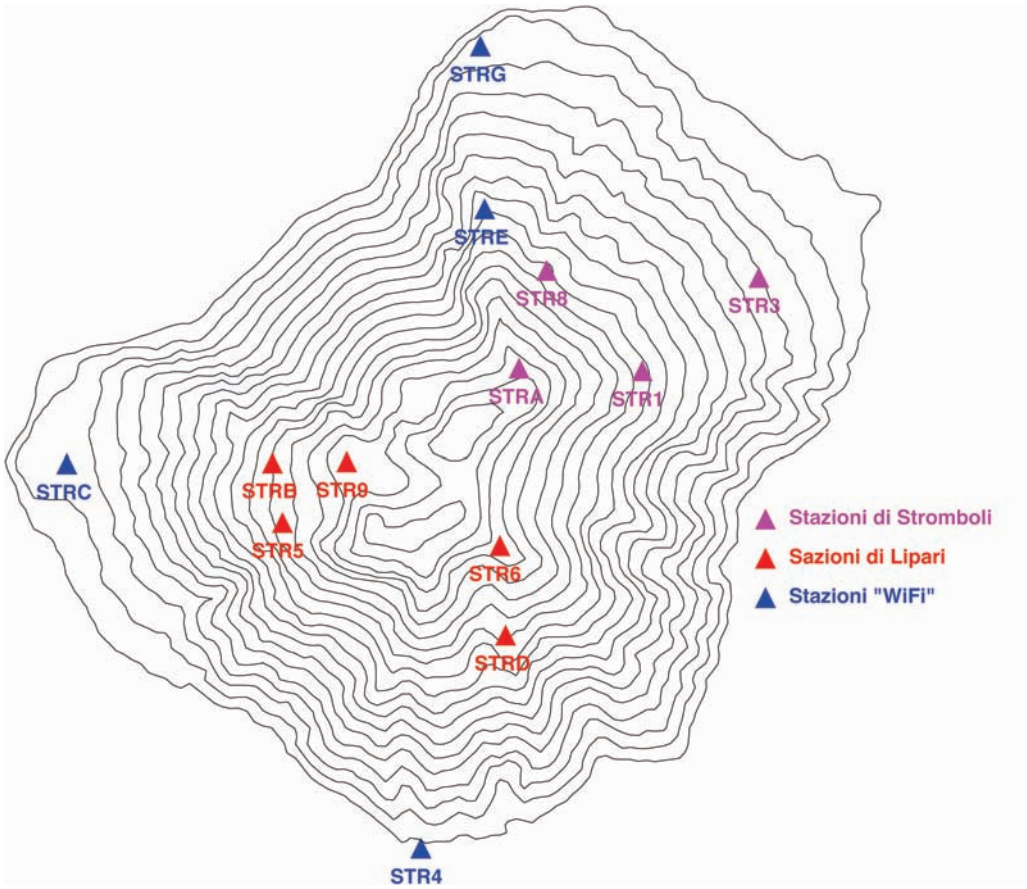

Figura 1 Mappa schematica della rete sismica di Stromboli.
Figure 1 Schematic map of the Stromboli seismic network.

calcolatore ad essa dedicata: uno posto a Lipari presso l'Osservatorio e gli altri due a Stromboli presso il COA. Il calcolatore di Lipari e uno di quelli di Stromboli utilizzavano una scheda multiseriale connessa ai *radiomodem* in modo da permettere la ricezione dei dati trasmessi via radio. Le stazioni della "rete wifi" (Figura 1) erano e sono tutt'ora equipaggiate con dei trasduttori seriale-ethernet in grado di trasformare il traffico seriale in traffico TCP/IP. In questo modo esse erano connesse direttamente all'altra macchina di acquisizione di Stromboli. I dati così acquisiti erano e sono trasmessi alle Sezioni di Napoli e Catania tramite il sistema *Earthworm*.

L'indipendenza dei sistemi di acquisizione rendeva il sistema abbastanza robusto: la rottura di uno dei calcolatori disconnetteva al massimo una delle tre sottoreti. I tre calcolatori non erano però intercambiabili e non era quindi possibile far ripartire l'acquisizione della sottorete mancante in maniera automatica. Una rottura rendeva quindi necessario l'intervento sull'isola di personale dell'Istituto lasciando nel frattempo l'acquisizione bloccata. Questo poteva diventare particolarmente difficoltoso durante i mesi invernali quando spesso le condizioni meteorologiche non permettono un accesso agevole all'isola.

Si è quindi pensato di rendere il sistema di acquisizione quanto più possibile resistente alle rotture e comunque in grado di reagire ad esse in maniera automatica. Sistemi di questo tipo rientrano nella categoria dei sitemi ad Alta Disponibilità di cui daremo una rapida descrizione nella prossima sezione.

2. L'alta disponibilità

Esistono in letteratura svariate tecniche in grado di garantire l'alta disponibilità di un sistema informatico. La prima e più semplice di esse consiste nell'avere un calcolatore costantemente ridondato da un gemello in grado di intervenire in qualunque momento per garantire la continuità del servizio. Per i nostri scopi eravamo interessati principalmente a due aspetti fondamentali: la continuità dell'acquisizione e la sicurezza e coerenza dei dati acquisiti. I sistemi operativi basati sul *kernel* Linux offrono due potenti soluzioni *open source* in grado di garantire questi due aspetti: *Heartbeat* ed il *Distributed Replicated Block Device* (DRBD). Anche se sviluppati da gruppi separati essi possono essere fortemente integrati e sono stati reputati una soluzione adeguata al problema in questione.

2.1 Heartbeat

Heartbeat[1] è composto da alcuni *daemon* Unix ed un insieme di programmi e *script* di supporto che permettono a due (o più) calcolatori di scambiarsi informazioni sul proprio stato e di prendere opportune contromisure nel caso uno di essi venga a mancare. La forma più utilizzata di sistema *Heartbeat* è quella di avere un calcolatore *master* sempre attivo ed uno *slave* in grado di intervenire quando necessario. Altre configurazioni più complesse sono comunque possibili ma non verranno trattate in questo rapporto in quanto non sono sembrate necessarie agli scopi in oggetto.

Due diversi eventi possono far intervenire lo *slave* per fargli prendere il posto del *master*:

1. Un malfunzionamento improvviso ed imprevedibile del *master*, come ad esempio una mancanza di alimentazione o la rottura di qualche componente *hardware* fondamentale,
2. Un avviso da parte del *master* che, a causa di qualche problema, non è più in grado di garantire il funzionamento.

Per comunicare di essere in vita, il *master* invia continuamente allo *slave* degli speciali pacchetti (detti, appunto, "heartbeat") che possono viaggiare tramite più mezzi di comunicazione. Un cavo seriale o parallelo, un cavo di rete dedicato o la LAN sono alcune di queste possibili vie. Questi pacchetti sono temporizzati e, se lo *slave* non ne riceve nessuno entro una soglia configurabile, decide che il *master* è morto e ne prende il posto. È bene evitare che lo *slave* prenda il controllo quando il *master* è ancora in funzione: questo evento crea in genere problemi tali da impedire comunque il funzionamento del sistema ("*split brain*"). Un guasto del canale di comunicazione potrebbe impedire allo *slave* di ricevere gli *heartbeat*, inducendolo così ad effettuare una partenza non richiesta. È quindi importante avere più di una via di comunicazione indipendente per gli *heartbeat*, in modo da ridurre la possibilità che un simile evento possa accadere.

Allo stesso tempo il sistema *Heartbeat* prevede l'utilizzo di sistemi di controllo (*Watchdog*) sia dell'*hardware* che del *software*: esso può essere quindi messo in grado di decidere se è possibile continuare a svolgere il compito previsto. Quando il *master* valuta che questo non è più possibile provvede a disattivarsi e passa di propria iniziativa il controllo allo *slave*. Il controllo dello stato è comunque attivo sulle due macchine contemporaneamente in modo da assicurare che esse siano entrambe in funzione: sarebbe quantomeno inopportuno che il *master* morisse e trovasse lo *slave* già fuori servizio.

Heartbeat nasce come sistema di alta disponibilità per servizi di rete (*server web, fileserver* e così via): una delle sue caratteristiche più interessanti è quella di essere in grado di condividere in modo automatico uno o più indirizzi di rete tra le macchine coinvolte. La macchina attiva in quel momento possiede questi indirizzi di rete "comuni": in questo modo è possibile garantire l'accesso ai servizi forniti anche senza conoscere quale dei due calcolatori sia attivo in quel momento.

[1] http://www.linux-ha.com

2.2 Il Distributed Replicated Block Device

DRBD [Reisner and Ellenberg, 2005; Ellenberg, 2007] è un sistema in grado di replicare un dispositivo a blocchi di una macchina Linux da un calcolatore ad un altro. Funziona in pratica come un sistema RAID-1 tramite rete. Come nel caso di *Heartbeat*, è possibile distinguere due ruoli diversi per le varie macchine che vi partecipano che vengono generalmente chiamati "primario" e "secondario". Come per *Heartbeat*, è possibile avere configurazioni complesse con più di un primario e più di un secondario: esse non verranno trattate in quanto per i nostri scopi una configurazione semplice con un solo primario ed un solo secondario è già soddisfacente.
Esistono svariati sistemi per la condivisione di file tra più macchine in HA: molti di essi prevedono l'utilizzo di uno speciale *filesystem* distribuito o in qualche modo replicato oppure l'utilizzo di un dispositivo esterno cui le macchine possano accedere contemporaneamente. L'approccio del DRBD è invece diverso: viene creato un dispositivo a blocchi virtuale che è poi replicato tra le varie macchine, in questo modo è possibile utilizzare su di esso un qualunque *filesystem* come se fosse un normale disco fisico collegato alla macchina.
Il DRBD risiede su un dispositivo fisico dedicato (una partizione, un intero disco, un *array* di dischi) che lo ospita e che può avere o meno le stesse dimensioni su entrambe le macchine. Esso è completamente disponibile in lettura ed in scrittura dal primario, mentre il secondario non vi può accedere durante le normali operazioni. Quando il primario scrive un dato sul dispositivo virtuale esso viene scritto sul dispositivo fisico sottostante e, contemporaneamente, viene inviato (e scritto) anche al dispositivo fisico del secondario che ne conterrà in ogni istante una copia completa.
Chiaramente questo comportamento può degradare le prestazioni in scrittura del primario. In genere, però, l'utilizzo di una connessione di rete dedicata sufficientemente veloce (ad esempio un cavo *ethernet gigabit* incrociato) consente di ovviare in modo estremamente efficace a questo problema.
Inoltre il sistema prevede l'utilizzo di tre differenti protocolli con i quali è possibile specificare quando la scrittura sul dispositivo virtuale è da considerarsi finita. In questo modo è possibile arrivare ad un compromesso tra le prestazioni e la sicurezza dei dati scritti. Si riporta di seguito una breve descrizione dei tre protocolli [Ellenberg, 2007]:

Protocollo A: Replica asincrona. La scrittura sul primario viene considerata finita quando termina la scrittura sul dispositivo fisico del primario e il pacchetto di replica è stato scritto nel *buffer* TCP locale. È possibile che ci sia perdita di dati in caso di una failure improvvisa del primario. I dati sul secondario saranno comunque consistenti dopo la partenza, ma è possibile che le scritture più recenti possano andare perse.

Protocollo B: Replica semi sincrona. La scrittura sul primario viene considerata completa quando termina la scrittura sul dispositivo fisico e il pacchetto di replica ha sicuramente raggiunto il secondario. Non dovrebbero andar perse scritture se non nel caso, ad esempio, di uno spegnimento contemporaneo ed improvviso delle due macchine e contemporanea distruzione del supporto fisico del primario.

Protocollo C: Replica sincrona. La scrittura sul primario viene considerata finita quando termina la scrittura su entrambe le macchine. In questo modo è possibile garantire che non ci siano perdite di dati nel caso di guasto di una sola delle due macchine.

Il protocollo **C** è quello consigliato nonché quello di *default*. Attualmente sono in corso dei test a cura del gruppo di sviluppo del DRBD, per promuovere il protocollo **B** allo stato di consigliato, ma non esistono ancora risultati definitivi.
DRBD è in grado anche di gestire autonomamente il caso in cui il supporto fisico del primario si guasti durante il funzionamento. In tal caso il sistema può andare automaticamente in modalità cosiddetta *"diskless"*: il ruolo del primario e del secondario non vengono cambiati, ma il primario continua a funzionare effettuando le letture e le scritture direttamente dal supporto fisico del secondario.

2.3 L'integrazione del DRBD con Heartbeat

Come detto in precedenza, *Heartbeat* ed il DRBD possono lavorare in modo integrato. È possibile configurare le macchine in modo che *Heartbeat*, oltre ad occuparsi dello stato dei sistemi e far partire i servizi necessari sulle macchine, si occupi anche della promozione o della retrocessione tra i due stati di primario e secondario. Inoltre *Heartbeat* si può occupare del montaggio/smontaggio del dispositivo e dell'eventuale controllo di consistenza dello stato del *filesystem*.
Anche se il DRBD è indipendente dal *filesystem* che viene utilizzato su di esso è comunque consigliato utilizzare un *filesystem* "journaled"[2] in modo da minimizzarne quanto più possibile il danneggiamento in caso di *failure* del primario. In ogni caso è sempre possibile utilizzare un *filesystem* "non journaled" se si ha l'accortezza di farne gestire ad *Heartbeat* il controllo di consistenza che non deve però più essere fatto in automatico dal sistema operativo durante l'avvio.

3. Il nuovo sistema di acquisizione della rete sismica di Stromboli

Il primo passo effettuato nel processo di irrobustimento del sistema di acquisizione è stata l'installazione di un sistema ad alta disponibilità presso il COA di Stromboli. Come visto

[2] http://it.wikipedia.org/wiki/Journaling

nella sezione 1 in precedenza al COA c'erano due macchine che si occupavano separatamente dell'acquisizione delle due "sottoreti" illustrate in Figura 1. Una possibile scelta poteva essere quella di rindondare ognuna delle due macchine mantenendo la struttura delle sottoreti. Ciò avrebbe significato l'installazione di due sistemi separati ad alta disponibilità ognuno composto da due macchine. Questa ci è però sembrata una soluzione poco ottimizzata: un unico sistema ad alta disponibilità in grado di acquisire entrambe le sottoreti è altrettanto efficace essendo la macchina di acquisizione completamente ridondata.

3.1 L'hardware

3.1.1 Le macchine di acquisizione

La prima e più importante scelta per un sistema HA è quella delle macchine su cui far girare il *software* di acquisizione. Nel nostro caso, oltre alla robustezza e all'affidabilità un requisito necessario è la possibilità di avere una discreta quantità di spazio disco in modo da garantire la presenza di alcuni mesi di dati.

La nostra scelta è stata di utilizzare due server HP Proliant **DL380-G5**, macchine con montaggio a *rack* da 2U che rappresentano un buon compromesso tra i requisiti da noi richiesti di affidabilità, prestazioni, dimensioni e costi. La configurazione scelta, tra quelle minime possibili per questa serie, è la seguente:

- Doppio alimentatore indipendente.
- Singolo processore Intel Xeon da 2.0*GHz*.
- 2*GB* di RAM.
- Doppia scheda di rete *Gigabit*.
- *Controller* RAID multifunzione in grado di supportare sia dischi SATA che SAS (Serial Attached SCSI).
- Otto dischi SAS da 146.7*GB*.

Il *controller* RAID di queste macchine consente la gestione di più volumi logici differenti su insiemi differenti di dischi. In questo modo è stato possibile dividere completamente lo spazio riservato al sistema operativo da quello riservato ai dati. Due degli otto dischi sono stati configurati come un *array* RAID 1 e su di essi è stato installato il sistema operativo. I rimanenti sei dischi sono stati configurati come un *array* RAID 5+1 per una dimensione complessiva di ~547*GB*.

In questo modo si è introdotto un ulteriore livello di protezione fisico anche contro le rotture improvvise dei dischi medesimi[3]. L'*array* RAID 1 è in grado di sopravvivere alla rottura di uno dei due dischi. Allo stesso modo, il RAID 5+1, con un disco "*spare*" non utilizzato normalmente, può sopportare la rottura di due dischi e continuare a funzionare. Occorre però tener presente che dopo la rottura del secondo disco le prestazioni in lettura e scrittura risultano estremamente penalizzate.

Delle due schede di rete a bordo delle macchine, la prima è stata utilizzata per le normali funzioni, ed è anche quella su cui *Heartbeat* condivide gli indirizzi comuni del sistema di acquisizione. La seconda interfaccia connette direttamente le due macchine con un cavo *gigabit* incrociato e viene utilizzata come interfaccia ad alta velocità per la comunicazione del DRBD e come comunicazione primaria per *Heartbeat*. La prima scheda di entrambe le macchine è invece collegata ad un normale *switch* di rete che le mette in comunicazione con il resto della LAN. Questa interfaccia è stata anche scelta come mezzo di comunicazione secondario per i pacchetti di *Heartbeat*.

3.1.2 La macchina console

Per motivi che verranno spiegati nella sezione 3.2.1 le due macchine di acquisizione non dispongono di un'interfaccia grafica per gli utenti, ma solo testuale. Si è reso quindi necessario affiancare ai calcolatori di acquisizione un terzo a cui gli utenti possono collegarsi per controllare il flusso di dati o per fare delle analisi.

Si è scelta una macchina molto più economica e meno potente delle precedenti, perché i compiti ad essa affidati sono semplici e non richiedono grande potenza di calcolo o di spazio disco.

La macchina scelta è un Koala Server prodotto dalla Koan Software[4]: una macchina da *rack* ad ingombro ridotto (solo 1U) e consumi estremamente bassi. Di seguito viene riportato un elenco delle sue caratteristiche:

- Scheda madre mini-ITX VIA, chipset CN700.
- Processore VIA C7 a 1.5*GHz*.
- Disco IDE da 3.5 pollici.
- Doppia scheda di rete 10/100.
- Scheda video integrata.
- Consumo totale, dischi esclusi: 25*W*.

3.1.3 Il trasduttore seriale-ethernet

Come visto nella sezione 1 l'acquisizione delle quattro stazioni del lato settentrionale dell'isola avveniva originariamente tramite una scheda multiseriale presente nel computer dedicato a quella sottorete e connessa direttamente con ognuno dei quattro *radiomodem* delle stazioni (Figura 2). Nel passaggio al sistema di alta disponibilità tale tipo di soluzione non è più utilizzabile in quanto non permette di spostare automaticamente la connessione seriale da una macchina all'altra.

Si è allora pensato di utilizzare una soluzione mutuata dall'acquisizione della "rete wifi": sostituire la scheda multiseriale con un trasduttore seriale-ethernet al quale collegare i *radiomodem* seriali ed a cui il programma di acquisizione possa connettersi per dialogare con le stazioni. La Figura 3 mostra uno schema di questo tipo di connessione. Il vantag-

[3] http://it.wikipedia.org/wiki/RAID

[4] http://www.koansoftware.com

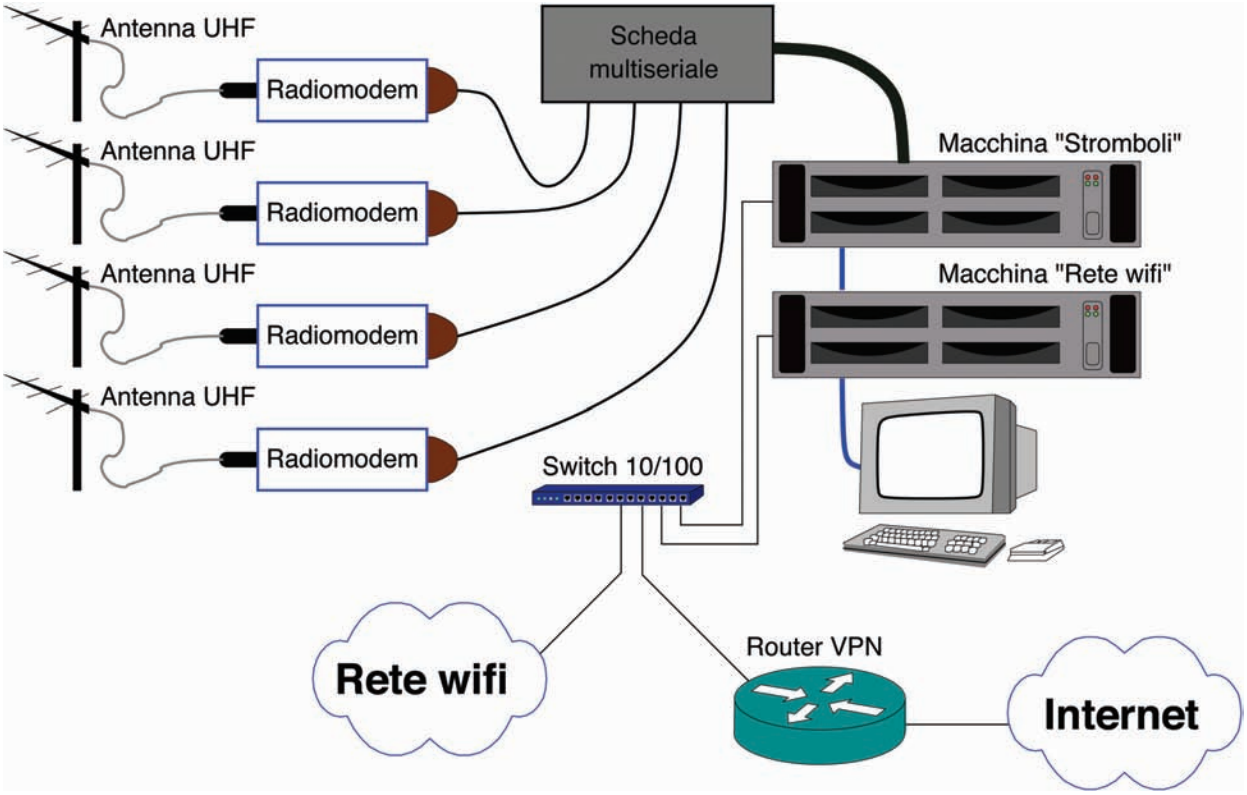

Figura 2 Schema del precedente sistema di acquisizione.
Figure 2 Schema of the old acquisition system.

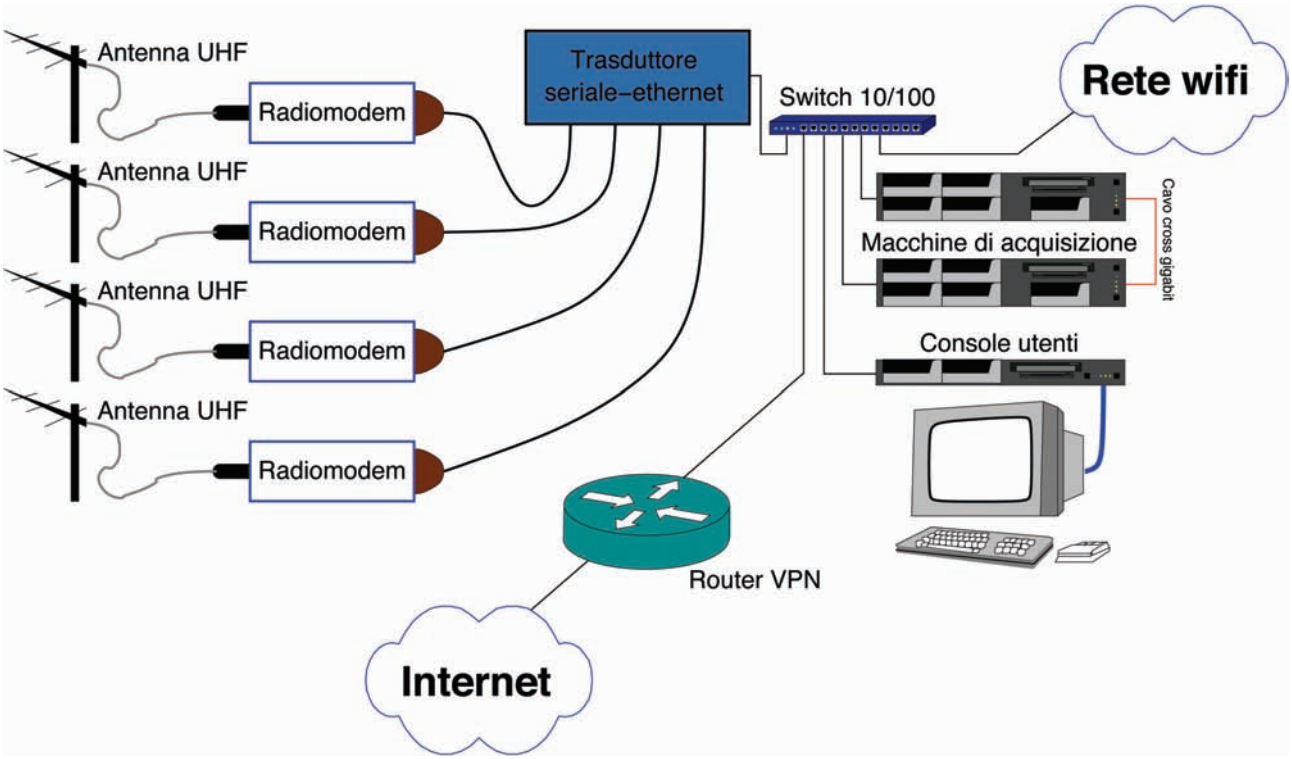

Figura 3 Schema dell'attuale sistema di acquisizione.
Figure 3 Schema of the new acquisition system.

gio di questa soluzione rispetto a quella "classica" risiede essenzialmente nel fatto che non è necessario che il trasduttore sia connesso fisicamente al calcolatore di acquisizione, visto che lo scambio di informazioni viene fatto tramite protocollo TCP/IP. In questo modo il passaggio da una macchina di acquisizione all'altra è assolutamente trasparente e non richiede alcun intervento umano. Inoltre questo approccio permette di utilizzare lo stesso programma di acquisizione (lantronix2ring [Peluso and De Cesare, 2006]) usato per le stazioni della "rete wifi".
Il trasduttore scelto nella fattispecie è un *terminal server* Lantronix ad otto porte che permette la connessione di un dispositivo seriale su ognuna delle porte. Le prime quattro sono state allocate alle quattro stazioni, mentre le ultime due sono state utilizzate per accedere via rete alle console seriali delle macchine di acquisizione. Questo accorgimento serve a garantire un ulteriore punto di accesso remoto alle macchine stesse e quindi un maggior controllo. Le rimanenti due porte sono state lasciate libere per sviluppi futuri.

3.2 Il software

3.2.1 Il sistema operativo

Come sistema operativo si è scelto di utilizzare la versione 4.0 del sistema operativo Debian GNU-/Linux (nome in codice "Etch")[5] per le sue caratteristiche peculiari anche nell'ambito del *software open source*:

Stabilità: Ogni programma che viene inserito dal gruppo di sviluppo è attentamente testato e, eventualmente, ne vengono corretti gli errori conosciuti distribuendo periodicamente degli aggiornamenti.

Disponibilità: Il sistema operativo viene garantito essere completamente *open source* e viene distribuito gratuitamente attraverso la rete senza la necessità di pagare licenze.

Sicurezza: Nella versione cosiddetta "stabile", non vengono mai introdotte nuove versioni (e quindi nuovi eventuali bachi) dei programmi che compongono il sistema: per essi vengono prodotti solo degli aggiornamenti quando vengono trovati e corretti degli errori particolarmente gravi o che possano compromettere la sicurezza del sistema.

Flessibilità: Durante la fase di installazione è possibile scegliere con estrema granularità i *software* da installare. Si evita così di ritrovarsi sul sistema dei programmi inutilizzati o potenzialmente pericolosi.

Il sistema scelto, inoltre, dispone dei pacchetti precompilati sia per *Heartbeat* che per il DRBD in maniera già parzialmente preconfigurata ed integrata, rendendo particolarmente agevole l'installazione, la configurazione e l'utilizzo di detti *software*.
Per motivi di sicurezza è stata installata una versione estremamente minimale del sistema operativo. Avere solo i programmi indispensabili al funzionamento del sistema è un importante requisito per evitare che i calcolatori possano essere violati. Con questo obiettivo si è scelto di non installare alcun tipo di interfaccia grafica sulle due macchine principali in quanto essa non è strettamente necessaria per i compiti cui sono delegate: l'acquisizione e la conservazione dei dati.
Per facilitare i compiti dell'utente, il calcolatore "console" è stato di contro dotato di una interfaccia grafica e di un insieme più ampio di programmi che permettono sia il controllo dell'intero sistema che di effettuare delle semplici operazioni ed analisi sui dati acquisiti, è stato inoltre abilitato un accesso ai dati salvati dalle macchine di acquisizione tramite una condivisione *Samba*[6] del *filesystem* ospitato sul DRBD.
La scelta di *Samba* al posto del più classico (per ambienti *nix) NFS[7] è stata soprattutto dettata dal fatto che esso possiede un protocollo di comunicazione più moderno. Grazie a ciò esso risulta immune ad alcuni problemi di NFS che lo possono rendere scomodo da utilizzare in un ambiente HA come quello in uso. In particolare con NFS può capitare che, nel passaggio da una macchina all'altra, chi avesse montato il *filesystem* di rete, si ritroverebbe con l'accesso impossibilitato ai dati a meno di non smontare e rimontare la risorsa. Questa cosa si può invece rendere assolutamente trasparente con *Samba* configurandolo in maniera opportuna.

3.2.2 Earthworm

Il progetto *Earthworm*[8] è stato sviluppato da un gruppo di ricercatori dello *United States Geological Survey* (USGS) per la gestione di dati sismici. Il concetto alla base di *Earthworm* è quello di avere un insieme di piccoli programmi chiamati *moduli* che comunicano tra loro tramite uno o più *buffer* di memoria condivisa chiamato "ring".
Ogni modulo è ottimizzato per un singolo compito. Il sistema viene fornito con un insieme di moduli per svolgere diverse attività: esistono moduli per l'acquisizione o per la trasmissione di dati sismici, altri per l'analisi, altri ancora per la conservazione e così via. La libreria di *Earthworm* fornisce inoltre una API completa per gestire l'accesso concorrente al "ring" che permette di scrivere nuovi moduli in modo abbastanza semplice. In Figura 4 è riportato uno schema di funzionamento per una semplice configurazione di un sistema *Earthworm*.
Il modulo chiamato `startstop` si occupa dell'inizializzazione dell'intero sistema e di lanciare gli altri moduli necessari. Esso viene coadiuvato dallo "*Status Manager*" (`statmgr`) che tiene sotto controllo l'intero sistema e può richiedere che un modulo eventualmente morto venga lanciato nuovamente.

[5] http://www.debian.org
[6] http://www.samba.org
[7] http://it.wikipedia.org/wiki/Network_File_System
[8] http://folkworm.ceri.memphis.edu/ew-doc/

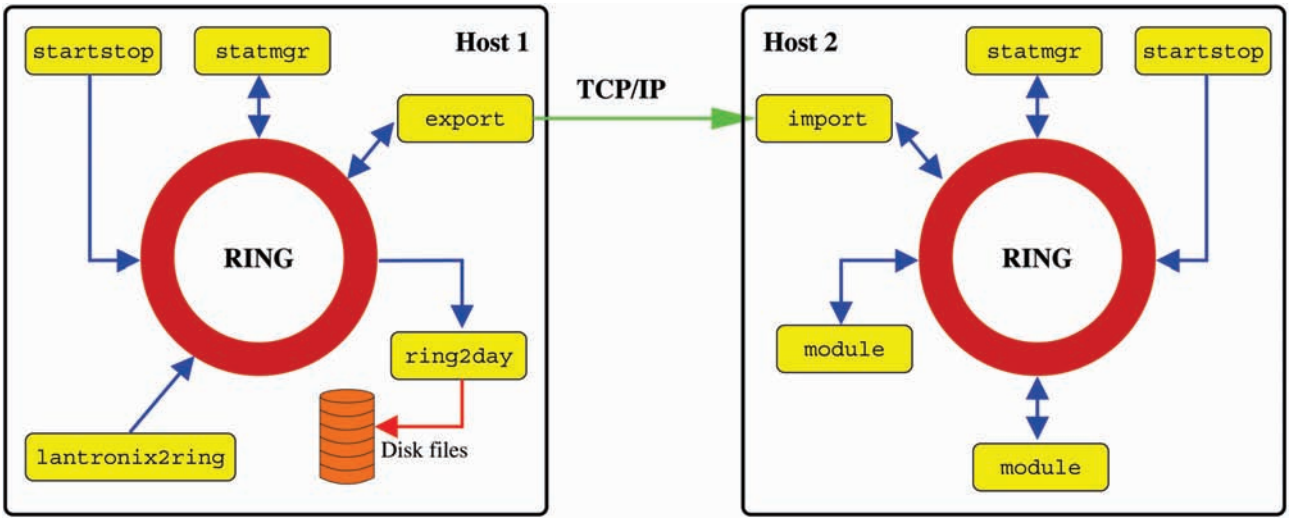

Figura 4 Un esempio di una semplice configurazione di un sistema Earthworm.
Figure 4 An example of a simple configuration of an Earthworm system.

Nel sistema di acquisizione viene utilizzata la versione 6.2 di *Earthworm* con l'ausilio di alcuni moduli da noi sviluppati [Peluso and De Cesare, 2006] dedicati alle nostre particolari esigenze. Di seguito si riportano i principali:

`lantronix2ring`: È il modulo che si occupa dell'acquisizione dei dati connettendosi via TCP/IP alla porta dei trasduttori seriale-ethernet.

`ring2day`: Viene usato sulle due macchine di acquisizione per salvare i dati nel formato "day" nativo del programma *WinDrum* sviluppato presso la sezione di Napoli dell'INGV per la visualizzazione in tempo reale dei dati sismici. La macchina attiva delle due scrive un file di un'ora per ogni canale sul dispositivo virtuale condiviso tramite DRBD.

`ring2suds`: Viene utilizzato sulla macchina "console" per salvare i dati nel formato SUDS, in modo da poter utilizzare il programma *Kuds* (vedi oltre) per una rapida visualizzazione ed analisi dei dati acquisiti.

3.2.3 Il controllo e l'analisi dei dati

Per consentire delle semplici analisi e per visualizzare i dati acquisiti dalla rete sismica sono stati installati sulla "console" tre programmi grafici da noi sviluppati per GNU/LINUX utilizzando le librerie grafiche Qt[9]:

`kuds`: Programma inizialmente sviluppato per permettere la visualizzazione sulla macchine GNU/LINUX dei file SUDS utilizzati alla Sezione di Napoli per l'archiviazione dei dati sismici. Nel suo ultimo rilascio (versione 0.4.3, Figura 5) permette di effettuare in maniera semplice alcune operazioni:

1. Effettuare il picking delle fasi dei segnali sismici ed esportarle nei formati di input di `hypo71` [Lee and Lahr, 1975] e `NLLoc` [Lomax et al., 2000].
2. Visualizzare lo spettro dei segnali sismici.
3. Applicare filtri di vario tipo ai segnali.
4. Visualizzare il *particle motion* su un piano.

`waveviewer`: Consente di attaccarsi direttamente al WAVE_RING di *Earthworm* e di visualizzare in tempo reale 1 minuto di dati (Figura 6).

`dayviewer`: Permette di visualizzare file DAY di 24 ore organizzati in sei schermate di 4 ore ciascuna (fig. 7).

3.2.4 Il controllo dello stato delle macchine

Il controllo dello stato dei server di acquisizione è stato affidato ad alcuni programmi forniti dall'HP[10] per questa serie di calcolatori. Essi sono disponibili anche per la versione "stabile" del sistema operativo Debian GNU/LINUX. Due di essi si sono rivelati particolarmente utili per poter fornire un rapporto quanto più completo possibile:

`hpacucli`: (*HP Array Configuration Utility*) È il programma con cui è possibile gestire gli *array* di dischi del *controller* RAID a bordo delle macchine. Consente di ottenere informazioni dettagliate sullo stato degli *array* e dei singoli dischi che li compongono.

`hpasm`: (*HP System Health Application and Insight Management Agents*) È un insieme di programmi e moduli del *kernel* che gestiscono lo stato interno dei componenti delle macchine. Utilizzando un programma di interfaccia (`hpasmcli`) è possibile accedere ad informazioni dettagliate su tali stati. Sono disponibili, ad esempio,

[9] http://trolltech.com
[10] http://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=T8570AAE

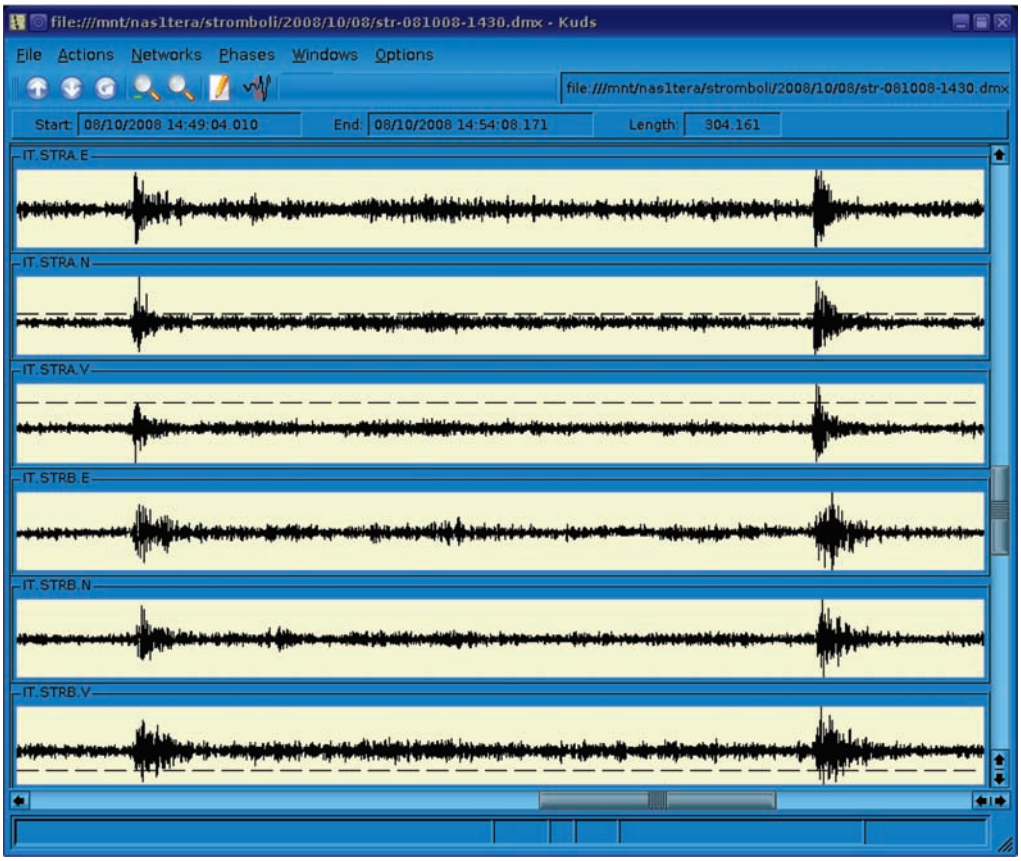

Figura 5 La schermata principale del programma kuds con le tracce si sei canali.
Figure 5 Main window of the kuds program with the tracks of six seismic channels.

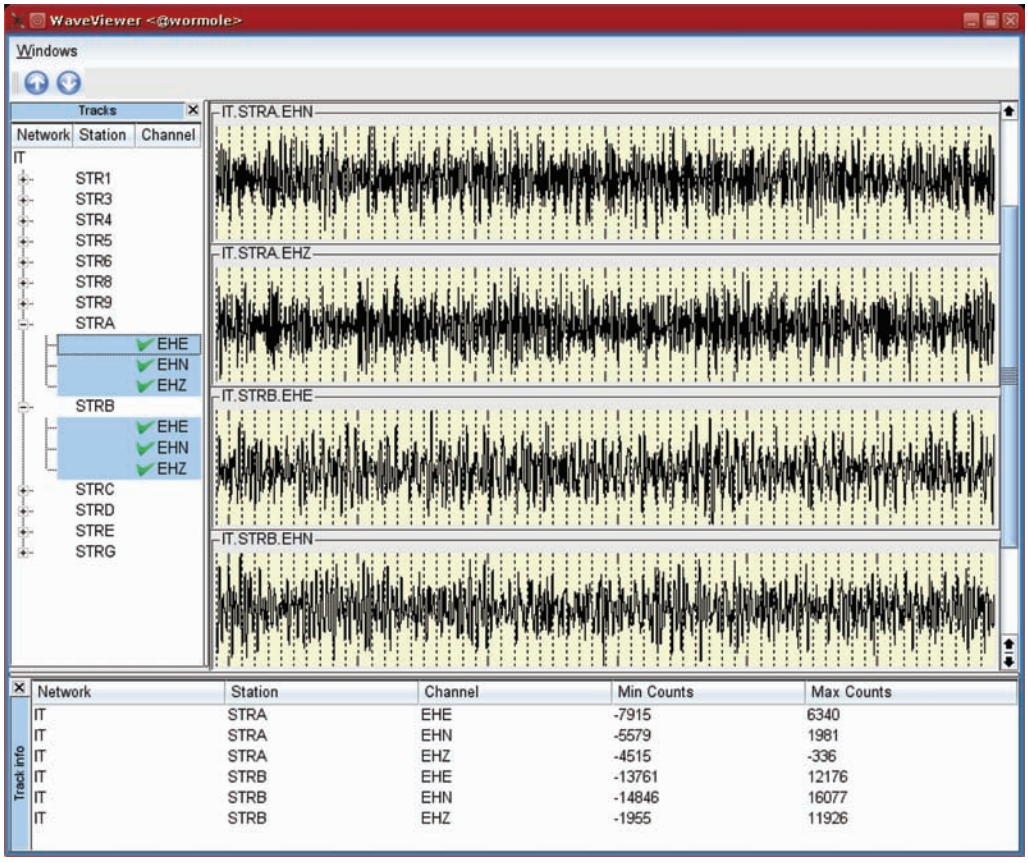

Figura 6 Una schermata del programma waveviewer.
Figure 6 A snapshot of the waveviewer program.

informazioni sullo stato degli alimentatori, delle ventole, sulla temperatura delle zone delicate ed altro.

Utilizzando questi programmi è stato creato uno *script* in grado di raccogliere queste informazioni. Un *daemon* standard Unix (*cron*[11]), in grado di eseguire dei comandi ad orari prefissati, si occupa di lanciare questo script ed inviare un messaggio di posta elettronica contenente dette informazioni. In questo modo è possibile tener sempre sotto controllo lo stato delle macchine ed accorgersi per tempo dell'insorgenza di qualche problema.

In Figura 8 è riportato un estratto di uno dei messaggi di posta elettronica inviati dal *master* in cui si può notare, nella sezione generata da `hpacucli`, come uno dei due dischi dell'*array* su cui risiede il sistema operativo sia in stato "Predictive Failure". Secondo HP ciò significa che il disco, sebbene ancora funzionante, potrebbe rompersi in tempi brevi ed è quindi consigliabile sostituirlo.

3.2.5 L'integrazione con Heartbeat

Utilizzando *Heartbeat* in ambiente GNU/LINUX è possibile integrare con esso un qualsiasi servizio *nix purché esso sia dotato di appositi *script* di partenza che seguano le indicazioni della *Linux Standard Base*[12]. La versione di *Samba* distribuita con Debian già segue questi consigli, risultando quindi immediatamente integrabile con il sistema *Heartbeat*.

Il pacchetto *Earthworm* non è invece pensato come un servizio *nix, ma come un sistema eseguibile separatamente ed indipendentemente da ogni utente di un calcolatore. Questo

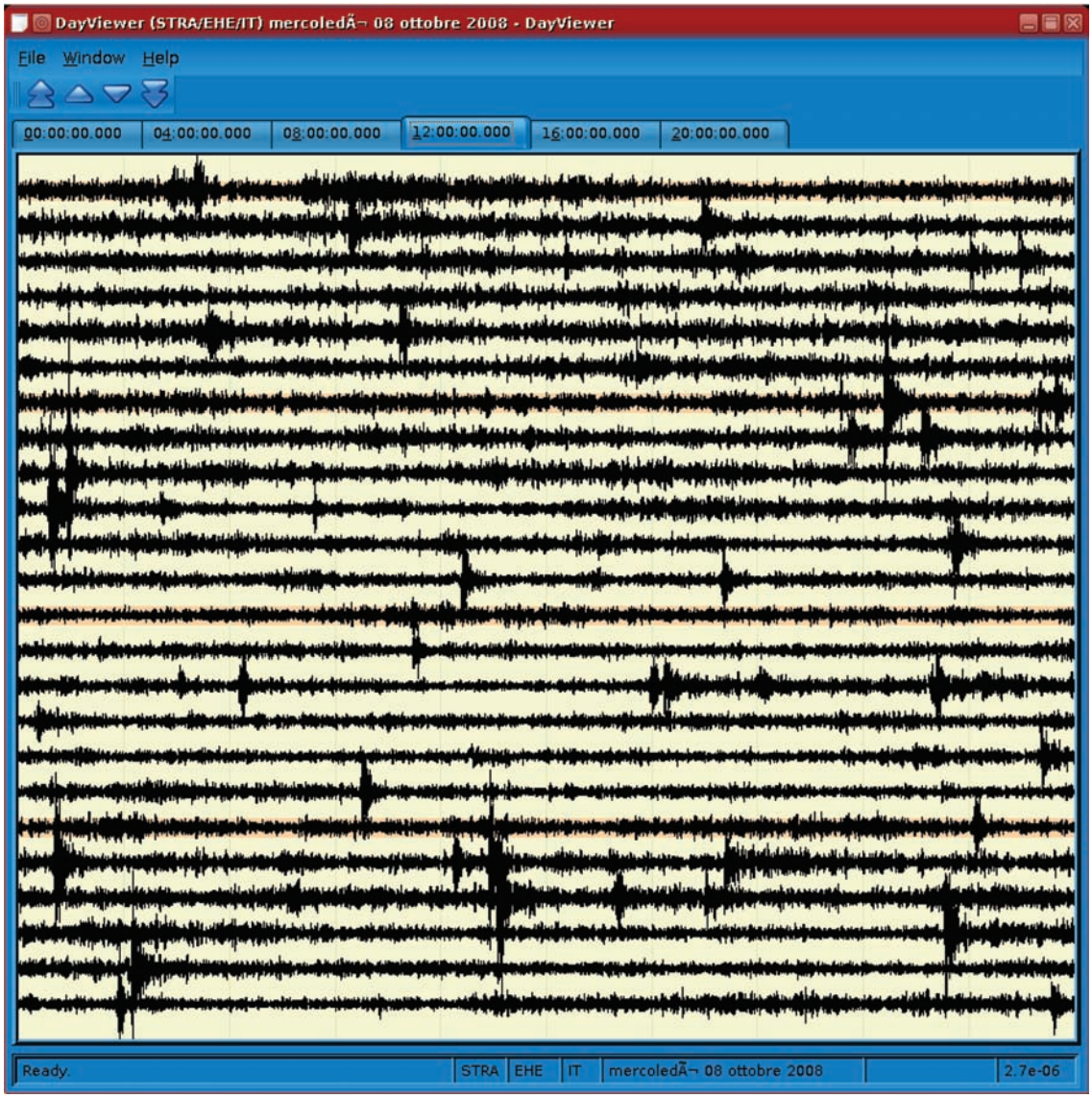

Figura 7 Una schermata del programma dayviewer.
Figure 7 A snapshot of the dayviewer program.

[12] http://www.linuxfoundation.org/en/LSB

```
HP Array Configuration Utility CLI 7.80-3.0
Detecting Controllers...Done.
Type help for a list of supported commands.
Type exit to close the console.

=>

Smart Array P400 in Slot 1 (sn: P61620G9VVS5GR)

    array A (SAS, Unused Space: 0 MB)

        logicaldrive 1 (136.7 GB, RAID 1+0, Interim Recovery Mode)

        physicaldrive 2I:1:1 (port 2I:box 1:bay 1, SAS, 146 GB, OK)
        physicaldrive 2I:1:2 (port 2I:box 1:bay 2, SAS, 146 GB, Predictive Failure)

    array B (SAS, Unused Space: 0 MB)

        logicaldrive 2 (546.8 GB, RAID 5, OK)

        physicaldrive 1I:1:5 (port 1I:box 1:bay 5, SAS, 146 GB, OK)
        physicaldrive 1I:1:6 (port 1I:box 1:bay 6, SAS, 146 GB, OK)
        physicaldrive 1I:1:7 (port 1I:box 1:bay 7, SAS, 146 GB, OK)
        physicaldrive 2I:1:3 (port 2I:box 1:bay 3, SAS, 146 GB, OK)
        physicaldrive 2I:1:4 (port 2I:box 1:bay 4, SAS, 146 GB, OK)
        physicaldrive 1I:1:8 (port 1I:box 1:bay 8, SAS, 146 GB, OK, spare)
[ ...]
```

Figura 8 Un estratto da uno dei messaggi di posta elettronica inviati dal master: si può notare come uno dei due dischi dell'array del sistema operativo si trovi nello stato "Predictive Failure".
Figure 8 An excerpt of one of the e-mail messages sent by the master: it is possible to note how one of the two disks of the array of the operating system is in the "Predictive Failure" status.

comporta che anche le versioni distribuite per sistemi operativi Unix o unix-like (Solaris e GNU/LINUX) non dispongano di siffatti *script* di partenza.
L'integrazione di *Earthworm* con *Heartbeat* è stata quindi realizzata scrivendo un apposito script di partenza conforme alle direttive *Linux Standard Base* e che fosse in grado di far partire il sistema come un servizio *nix. Il programma `startstop` (sezione 3.2.2) è pensato per funzionare in maniera interattiva collegato ad un terminale utente. Per poterlo utilizzare come servizio è stato quindi necessario farlo girare in una console che fosse possibile distaccare dal terminale e riattaccabile a richiesta. Per far ciò è stato usato `screen`, un programma abbastanza complesso che, tra le altre cose, è in grado di fare esattamente quanto richiesto.
Lo *script* che lancia *Earthworm* è stato dunque organizzato in modo che il programma `startstop` venga fatto partire all'interno di una sessione di `screen` distaccata e che, all'arresto, esso venga terminato tramite i comandi standard di *Earthworm* stesso. In questo modo è possibile, in qualsiasi momento, attaccarsi alla console in cui viene fatto girare `startstop` e controllare così l'intero sistema *Earthworm*.

3.3 L'archiviazione dei dati

I file DAY ottenuti dal sistema di acquisizione vengono archiviati ogni giorno su un NAS (*Network Attached Storage*) installato nella Sala Macchine dell'Osservatorio Vesuviano. La copia avviene durante la notte alle 01:00 GMT ovvero in un orario in cui si presume ci sia meno traffico in entrata/uscita dal COA. Come nel caso del messaggio di controllo descritto nella sezione 3.2.4, questo servizio viene garantito con l'utilizzo del *daemon cron*.
Per minimizzare l'occupazione di banda durante la copia dei *file* si effettuano delle copie incrementali, vengono cioè copiati solo i file ancora non presenti sulla macchina di archiviazione. Per questa funzione si utilizza il programma *rsync*[13] che è in grado appunto di effettuare questa operazione.

4. Primi riscontri e sviluppi futuri

Il sistema descritto è attualmente installato e funzionante al centro di acquisizione di Stromboli (Figura 9). Al

[13] http://samba.anu.edu.au/rsync/

momento della stesura di questo rapporto, esso è in funzione da circa un anno ed ha accusato come unico problema il disco prossimo alla rottura di cui si è detto nella sezione 3.2.3 e nella Figura 8. La probabilità di rottura entro i primi 6 mesi di vita di un disco nuovo è di circa il 5% [Pinheiro et al., 2007]: era dunque prevedibile che uno dei 16 dischi potesse danneggiarsi.

Non abbiamo ancora, quindi, prove di resistenza del sistema in produzione in caso di gravi malfunzionamenti. Sono stati però effettuati alcuni test sia in fase di realizzazione del sistema che dopo la sua effettiva installazione simulando in vari modi la morte del *master*. Questi test hanno mostrato che, nel caso peggiore, lo *slave* riesce a prendere il controllo della situazione in circa due minuti dal momento in cui il *master* fallisce.

Questo tempo è il minimo possibile che si è riusciti ad ottenere nel peggiore dei casi agendo sulla configurazione dei *timeout* del sistema *Heartbeat*. È importante notare come durante questo tempo l'acquisizione dei dati sia ovviamente interrotta e, poiché i digitalizzatori con cui sono equipaggate le stazioni (vedi sezione 1) non sono in grado di ritrasmettere un dato eventualmente perso, non è comunque possibile garantire la completa continuità del dato acquisito. Una interruzione di pochi minuti (o meno) del flusso di dati è in ogni caso tollerabile per gli scopi della rete sismica.

Un ulteriore beneficio di questa architettura si è resa evidente durante una delle ultime missioni sull'Isola. A causa delle notevoli quantità di cenere vulcanica presente nell'atmosfera di Stromboli, le ventole dei sistemi vengono rapidamente intasate dalla polvere. Questo richiede lo spegnimento delle macchine per poterle ripulire. Con il precedente sistema di acquisizione questo significava tenere ferma la stessa per un certo tempo. Questo non è più un problema visto che è sempre possibile effettuare una qualunque manutenzione *hardware* ad solo un calcolatore per volta, lasciando, quindi, sempre attiva l'acquisizione.

Un sistema identico verrà a breve installato anche all'Osservatorio di Lipari in modo da rendere per quanto possibile completamente ridondato ed autosufficiente l'intero sistema di acquisizione della rete sismica di Stromboli.

Figura 9 Fotografia del rack contenente le macchine di acquisizione. Dall'alto verso il basso si possono riconoscere: la macchina console, il master, lo slave ed uno dei vecchi assemblati che si occupavano precedentemente dell'acquisizione.
Figure 9 Photo of the rack containing the acquisition machines. From top to bottom it is possible to view: the console machine, the master, the slave and one of the old assembled computers once doing the acquisition.

Bibliografia

De Cesare W., Orazi M., Peluso R., Scarpato G., Caputo A., D'Auria L., Giudicepietro F., Martini M., Buonocunto C., Capello M. and Esposito A.M., (2009). *The broadband seismic network of Stromboli volcano, Italy*. Seismological Research Letters, 80(3):435–439, May/June 2009. doi: 10.1785/gssrl.80.3.435.

Ellenberg L., (2007). *Drbd 8.0.x and beyond shared-disk semantics on a shared-nothing cluster*. In LinuxConf Europe 2007, August 10th 2007, Cambridge.

Lee W. H. K. and Lahr J. C., (1975). *Hypo71 (revised). A computer program for determining hypocenter, magnitude and first motion pattern of local earthquakes*. U.S. Geological Survey Open-File Report, 75(311):116.

Lomax A., Virieux J., Volant P. and Berge C., (2000). *Probabilistic earthquake location in 3d and layered models: Introduction of a metropolis-gibbs method and comparison with linear locations*. In Advances in Seismic Event Location (C. H. Thurber and N. Rabinowitz, eds.), pp. 101–134. Kluwer, Amsterdam.

Orazi M., Martini M. and Peluso R., (2006). *Data acquisition for volcano monitoring*. EOS, 87(38), 19 september 2006.

Orazi M., Peluso R., Caputo A., Capello M., Buonocunto C. and Martini M., (2008). *A multiparametric low power digitizer: project and results*. In Conception, verification and application of innovative techniques to study active volcanoes (W. Marzocchi and A. Zollo eds.), pp. 435–460. Copyright © (2008) Istituto Nazionale di Geofisica e Vulcanologia.

Peluso R. and De Cesare W., (2006). *Acquisizione dati da stazioni sismiche digitali tramite earthworm in ambiente gnu/linux*. Technical Report 8, INGV Osservatorio Vesuviano, Napoli.

Pinheiro E., Weber W. D. and Barroso L. A., (2007). *Failure trends in a large disk drive population*. In Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST 2007), February 2007, San Jose, CA.

Reisner P. and Ellenberg L., (2005). *Drbd v8 replicated storage with shared disk semantics*. In Proceedings of the 12th International Linux System Technology Conference, September 18th 2005, Hamburg.

Salvaterra L., Pintore S. and Badiali L., (2008). *Rete sismologica basata su stazioni gaia*. Rapporti Tecnici INGV, 68, Roma.

Indice

Coordinamento editoriale e impaginazione
Centro Editoriale Nazionale | INGV

Progetto grafico e redazionale
Daniela Riposati | Laboratorio Grafica e Immagini | INGV

Via di Vigna Murata, 605
00143 Roma
Tel. +39 06518601 Fax +39 065041181

http://www.ingv.it

Istituto Nazionale di Geofisica e Vulcanologia