Linux Day 27 Novembre 2004

GULCh – Gruppo Utenti Linux Cagliari

High Performance Computing con openMosix

Matteo Dessalvi

High Performance Computing

Con questa espressione ci riferiamo ad un sistema dotato di risorse di calcolo dalle prestazioni decisamente elevate.

Tradizionalmente, molti sistemi HPC vengono implementati tramite singoli calcolatori, capaci di alloggiare numerose CPU e dotati di bus dati molto veloci ed efficenti.

Cray YMP

- Programmazione vettoriale.
- Hardware e software integrati.
- Elevate prestazioni.
- Sistema proprietario.

Un approccio differente

I sistemi HPC implementati attraverso un singolo calcolatore possono essere molto costosi.

Una soluzione alternativa, delineatasi nel corso degli anni, prende il nome di clustering ed ha permesso di creare supercomputer basati su componenti COTS (Commodity Off The Shelf).

Cosa è un cluster?

Un cluster è formato da un insieme di elaboratori, detti nodi, collegati in modo collaborativo. Vantaggi più evidenti:

- economicità: i componenti COTS sono facilmente reperibili e poco costosi.
- scalabilità: la potenza del cluster può aumentare al crescere del numero di nodi.
- fault tolerance: il guasto di un nodo non compromette il resto del cluster.

Tipologia di cluster

- HA (High Availability): vengono utilizzati per garantire la continuità di uno o più servizi per il maggior tempo possibile.
- Load balancig: sono utilizzati per ridurre il più possibile i tempi di sottoutilizzo di certe macchine.
- Calcolo parallelo: vengono utilizzati per effettuare calcoli che richiedano l'uso di algoritmi paralleli.

HPC con openMosix

openMosix ricade nella categoria dei cluster per il bilanciamento di carico.

Si tratta di una patch del kernel Linux che trasforma un insieme di macchine in un cluster SSI (Single System Image).

I sistemi di questo genere nascondono all'utente la struttura interna del cluster e danno l'impressione di lavorare con un unico grande computer.

Caratteristiche di openMosix

- I nodi sono paritetici e non in relazione master/slave.
- L'integrazione con il kernel Linux e l'uso del paradigma SSI consentono di operare con le proprie applicazioni senza che sia necessario modificarle.
- Il bilanciamento di carico è basato sul concetto di migrazione dei processi.

Uno sguardo all'interno

Le operazioni compiute da openMosix sono sostanzialmente di tre tipi:

- Valutazione delle risorse disponibili su ogni singolo nodo.

- Gestione delle risorse disponibili sul cluster.

- Migrazione dei processi.

Raccogliere informazioni

L'esatta valutazione sull'utilizzo reale delle risorse del cluster parte dai singoli nodi che lo compongono.

Su ognuno di essi openMosix raccoglie numerose informazioni: velocità della cpu, ammontare della RAM, le operazioni di I/O effettuate su disco o via rete, ecc.

Valutare le risorse del cluster

Le informazioni raccolte su ogni singolo nodo sono eterogenee, ovvero non sono confrontabili. E' quindi complicato avere un quadro chiaro delle risorse disponibili.

openMosix risolve questo problema convertendo queste quantità in un singolo parametro indicato come costo.

Migrazione dei processi

Una volta stabiliti i costi, i jobs lanciati dall'utente vengono assegnati ai nodi che offrono le proprie risorse ad un costo inferiore rispetto agli altri.

Migrare un processo è una operazione assolutamente trasparente all'utente e che non influisce sull'esecuzione del programma sul nodo originario.

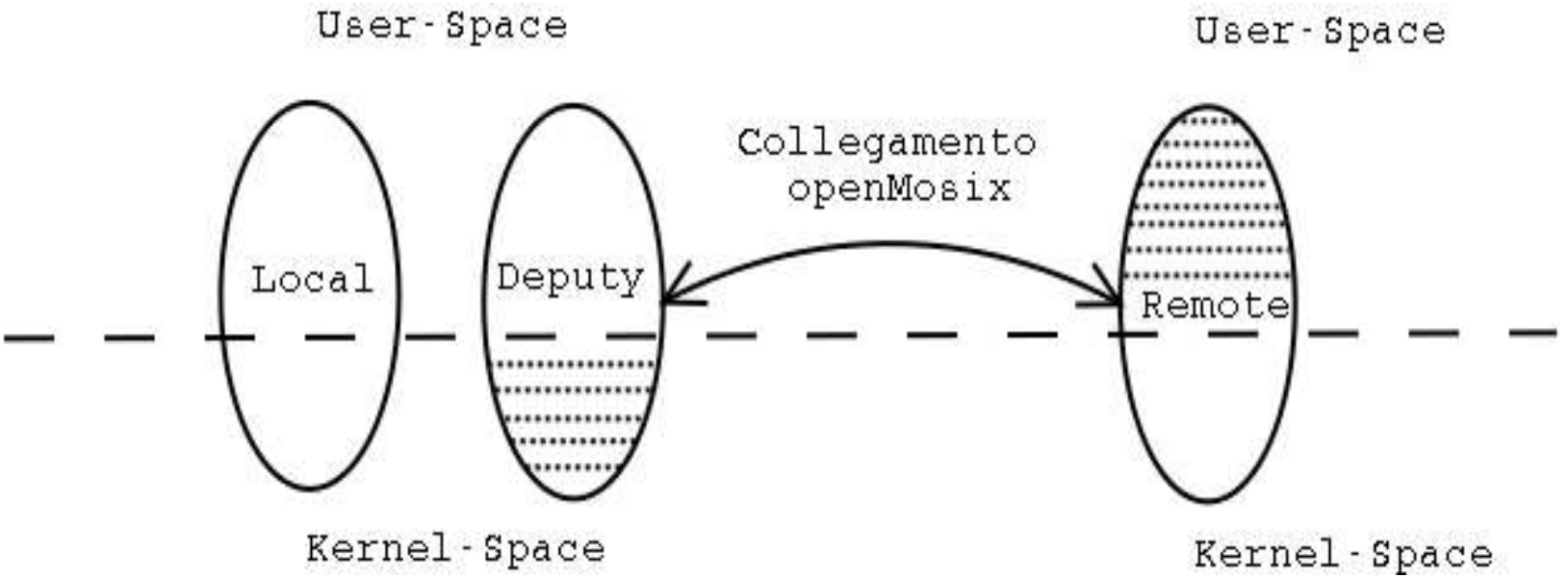
Diffondere le informazioni

Il nodo che offre le proprie risorse ad un costo minore potrebbe ritrovarsi sommerso dalle richieste degli altri nodi e questo porterebbe ad un collasso!

Le informazioni sui costi delle risorse vengono diffuse in modo random solo ad una parte dei nodi del cluster, in modo da evitare colli di bottiglia durante le elaborazioni.

Migrazioni

Ciò che viene effettivamente migrato su un nodo remoto è la parte in user space del processo, mentre la parte in kernel space rimane nel nodo di partenza.



In dettaglio

Una volta portato in memoria sul nodo remoto, il processo riprende il suo lavoro.

Le pagine di memoria virtuali necessarie vengono trasferite dal nodo di partenza a quello di destinazione ogni volta che il processo genera un page fault.

L'intero working set del processo verrà così ricostruito sul nodo di destinazione.

Filesystem di clustering

Le operazioni di un cluster SSI devono essere supportate da un filesystem che consenta una visione unificata dei files su ogni singolo nodo.

L'openMosix File System (oMFS) è lo strato software che consente al kernel di lanciare le operazioni sui files tramite il supporto dei comuni filesystem Linux: ext2, ext3, ReiserFS, JFS, ecc.

Caratteristiche di oMFS

- Un processo migrato potrà operare sui propri files come se si trovasse sul nodo di partenza.
- Vengono mantenuti coerenti i contenuti dei files in cache quando vengono modificati sui nodi remoti.
- Link e time stamp associati ad un file vengono mantenuti consistenti tra i vari nodi.

Applicazioni openMosix

- GNU Octave: calcoli numerici
- R: software per elaborazioni statistiche.
- Maple 8: Computer Algebra System.
- WRF(Weather Research and Forecast): simulazioni metereologiche.
- Make: supporto alla compilazione di progetti software.
- Postfix (Mail Transport Agent).

Costruire cluster openMosix

Abbiamo bisogno di un gruppo di PC, equipaggiati con schede di rete e lettori CDROM o DVD. Non è necessario che le CPU a bordo siano l'ultimo modello o molto recenti.

Sarà possibile preparare il cluster anche se non si dispone di hard disk per tutte le macchine. Si può infatti utilizzare un sistema diskless.

Installazione

- Supponiamo di avere su ogni computer una RedHat, con installazione standard.
- Il kernel openMosix viene fornito anche in formato RPM. Una volta installato è necessario segnalarlo al nostro boot loader (Lilo o Grub).
- Modificare la mappa dei nodi oM.
- Modificare il file /etc/fstab aggiungendo le opzioni per l'oMFS.

openMosix.map

openMosix.map è un file di testo, posto nella directory /etc, nel quale vengono memorizzati i nodi del cluster openMosix.

OM	Node	Indirizzo IP	Num. processori
	1	192.168.10.1	1
	2	192.168.10.2	1
	3	192.168.10.3	1

.....

openMosix discovery daemon

Se il cluster dispone di numerosi nodi e la configurazione si modifica continuamente si può ricorrere al demone omdiscd. Lanciandolo da linea di comando si può controllare l'output sui files di log:

openMosix configuration changed:
openMosix #2780 is at IP address 192.168.x
openMosix #2638 is at IP address 192.168.x
.....

Visualizzare la map dei nodi

La openMosix map è visualizzabile con il comando showmap da cmd line:

My Node-Id: 0x0adc

Base Node-Id	Address	Count
0x0adc	192.168.10.1	1
0x0a4e	192.168.10.2	1
0x0a56	192.168.10.3	1
0x0a43	192.168.10.4	1
0x0a4a	192.168.10.5	1

Utilizzare oMFS

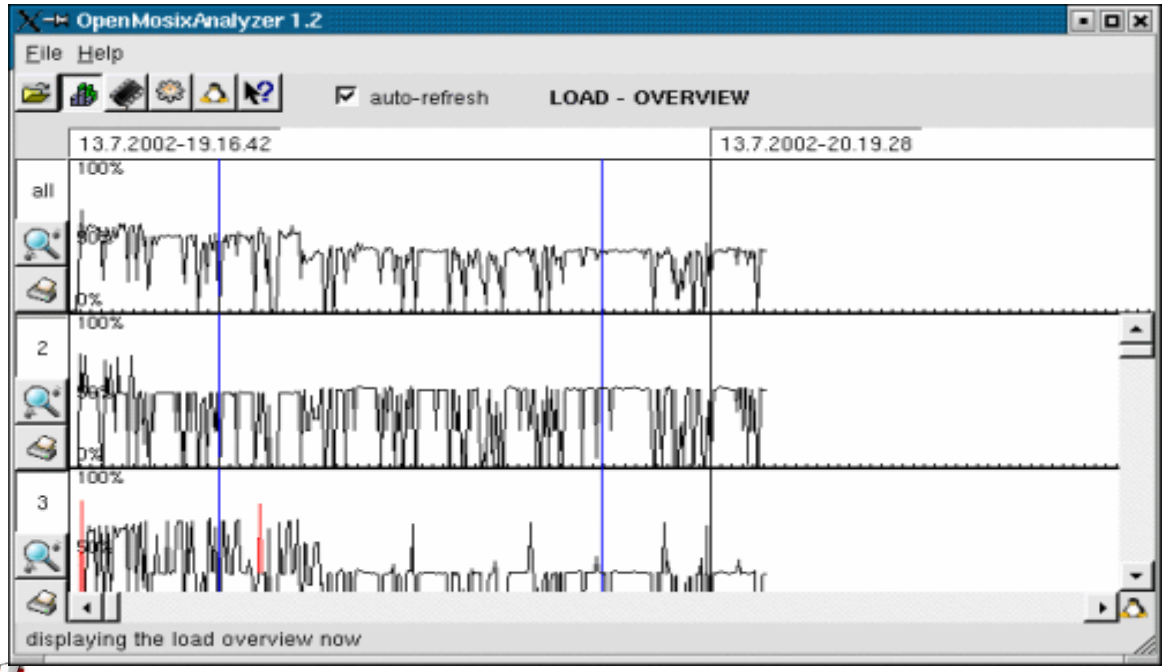
Su ogni nodo del cluster va creata una directory che funzionerà da mount point per il filesystem di clustering. Su fstab si deve inserire:

```
mfs_mnt    /mfs    mfs    dfsa=1      0 0
```

L'opzione dfsa (Direct FileSystem Access) segnala ad oM che le operazioni di I/O su un file possono avvenire anche al di fuori del nodo di partenza del processo.

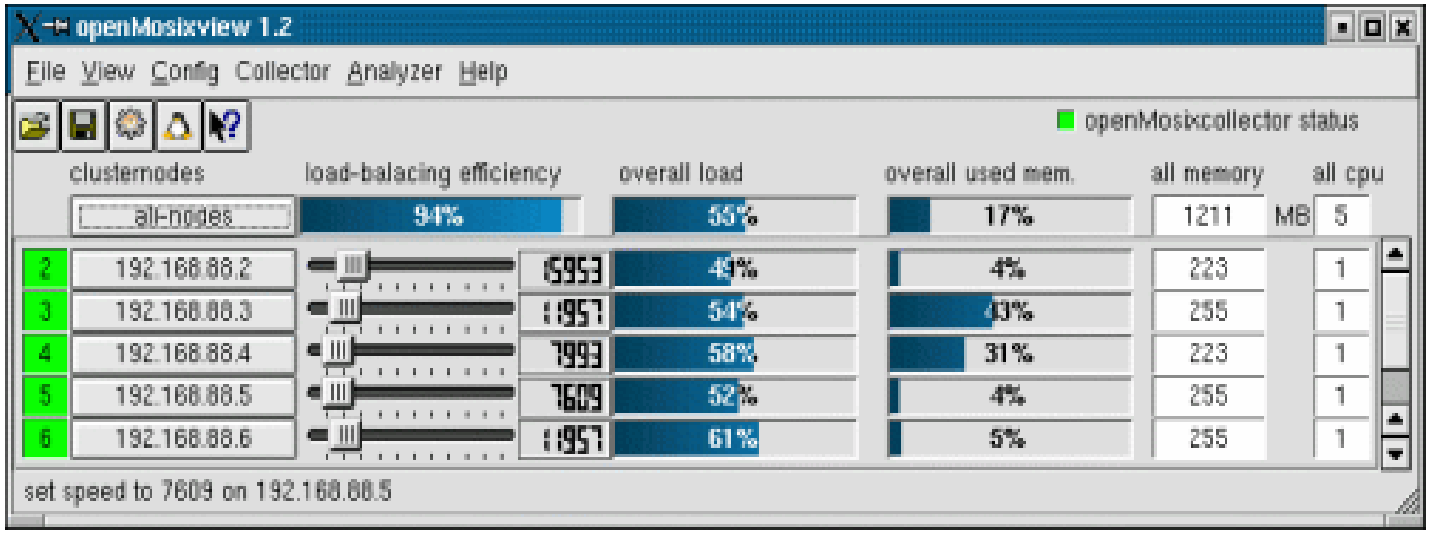
openMosix collector

Il collector di openMosix si occupa di raccogliere informazioni sul carico, sulla memoria e sui processi in esecuzione sul cluster. Queste informazioni vengono poi visualizzate graficamente:



openMosix view

Per monitorare un cluster openMosix sono disponibili alcune utility grafiche. Le openMosix View necessitano delle librerie Qt per poter funzionare.



Vantaggi di openMosix

- E' possibile sfruttare i propri programmi senza ricompilarli.
- I programmi che fanno uso delle librerie MPI/PVM si integrano senza problemi con openMosix.
- Disponibilità di diverse utility in user space per controllare ed interagire con il cluster.

...e svantaggi

- I kernel openMosix non sono in grado di sfruttare la migrazione su programmi che fanno uso di memoria condivisa o threads.
- E' possibile utilizzare unicamente macchine con architettura Intel.
- Non è possibile bloccare un processo e far riprendere l'elaborazione dal punto al quale si era fermato.

Perchè queste limitazioni?

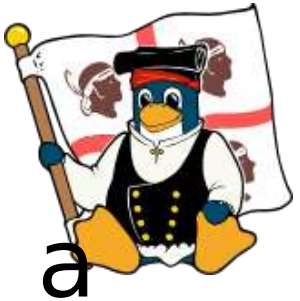
- Utilizzare la memoria condivisa significa non poter suddividere e ridistribuire lo spazio di indirizzamento su più nodi.
- Il modello di threading adottato da Linux (1:1 ad ogni thread in user space viene fatto corrispondere un thread in kernel space) impedisce la separazione di un thread dal suo pool di esecuzione.

MigSHM

E' una patch per openMosix che consente la migrazione di processi che facciano uso esplicito delle sys call: *shmget()*, *shmmat()*, *shmdt()*, *shmctl()* e *clone()*.

Nonostante il carattere sperimentale ha già dimostrato di poter migrare i processi del web server Apache.

ChPOX



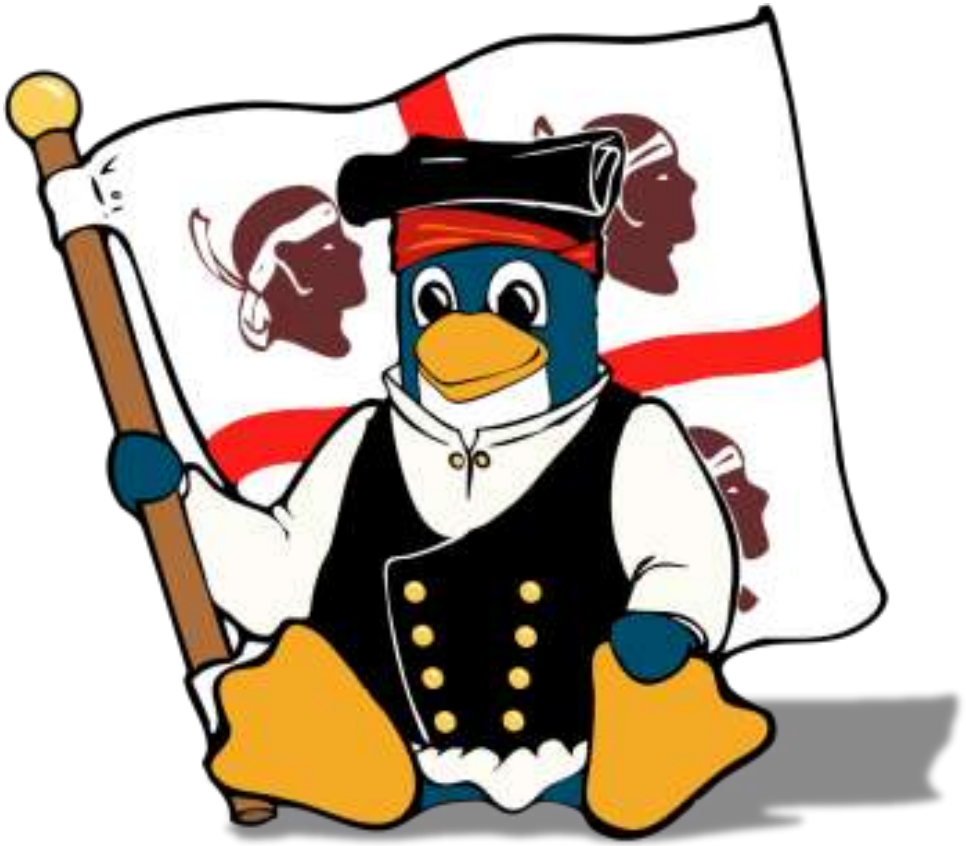
L'operazione di riavviare un processo, a partire dal punto nel quale l'elaborazione si è bloccata, è detta *checkpointing*.

ChPOX è un modulo per il kernel Linux che consente di salvare lo stato di un processo e dei suoi figli, per poi poterli ripristinare in caso di bisogno. Per poter operare al meglio è necessario che venga utilizzato più volte durante la vita di un processo.

Links utili

- openmosix.sourceforge.net (Main site)
- L'HOWTO di openMosix si trova su: http://howto.ipng.be/openMosix-HOWTO/
- Distro derivata dalla Knoppix per oM: bofh.be/clusterknoppix
- Sito di riferimento per ChPOX: www.cluster.kiev.ua/tasks/chpx_eng.html
- mcaserta.com/maask/ (MigSHM patch)
- http://www.openmosixview.com/

The end



- Grazie a tutti per l'attenzione.
- Appuntamento al LinuxDay 2005!
- Happy hacking!