



Ottimizzazione di un Cluster openMosix



Roberto Premoli

Introduzione

- ◆ **Cosa è un cluster?**

Un cluster è un insieme di computer (chiamati nodi) dedicati all'ottenimento di prestazioni (cluster di calcolo) o gradi di affidabilità (cluster ad alta affidabilità) non ottenibili dalle singole unità dell'insieme.

- ◆ **Cosa è openMosix?**

openMosix è una patch del kernel linux, che lo mette in grado di far migrare i processi da un PC ad un altro, migliorando così le prestazioni complessive del cluster.

Potenza equivalente (PE) TEORICA in MHZ

- ◆ Nel caso di N nodi identici, la $PE_{(cluster)}$ e':

$$PE_{(cluster)} = P_{(nodo)} * N$$

- ◆ Quindi, un cluster di 4 Pentium4 a 3Ghz, dovrebbe essere equivalente ad un unico Pentium4 a 12Ghz.... Purtroppo non e' cosi'.

Potenza equivalente (PE) Effettiva in MHZ

- ◆ Nel caso di N nodi identici, la $PE_{(cluster)}$ e'

$$PE_{(cluster)} = M * P_{(nodo)} * N$$

Dove M e' un coefficiente moltiplicativo.

Tanto piu' alto e' M, tanto piu' efficiente sara'
il Cluster. NOTA: M e' SEMPRE minore di 1.

Limiti alla scalabilità: i colli di bottiglia del cluster

- ◆ CPU (clock e cache)
- ◆ Clock bus PCI
- ◆ Ram (tipo, velocità e quantità)
- ◆ I/O Hard Disk (tipo, velocità e quantità)
- ◆ Velocità NIC
- ◆ Topologia connessione tra nodi
- ◆ Ritardi nella migrazione dei job
- ◆ Tipo di job da eseguire sul cluster

Limite invalicabile: il bus di sistema

- ◆ Non si può generare un traffico di rete o di I/O su disco che ecceda la capacità del bus.
- ◆ Il bus di sistema PCI, (in un prossimo futuro PCIX), è il vero ed invalicabile “collo di bottiglia” di un PC.
- ◆ La sequenza cronologica XT → ISA → VESA → PCI/AGP → PCIX è il tentativo di garantire un canale sempre più largo e veloce per le periferiche (CPU, scheda video, NIC, dischi etc)

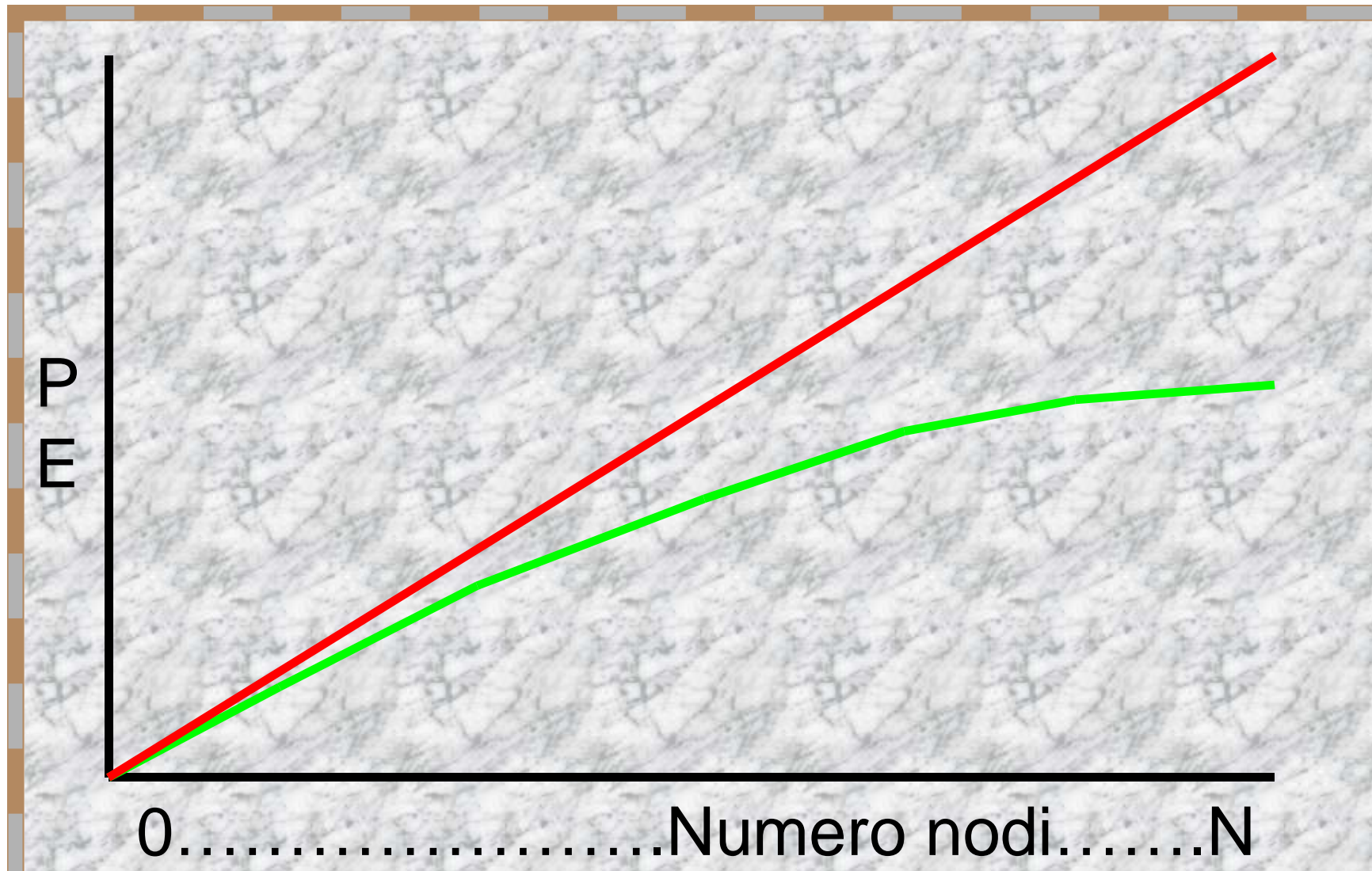
Il cluster ideale



Il cluster reale



Confronto IDEALE-REALE



Dove migliorare il cluster

- ◆ **Algoritmo di openMosix**
- ◆ **Hardware (CPU, RAM, etc)**
- ◆ **Gli applicativi**
- ◆ **Topologia di rete**

Dove migliorare il cluster (1)

- ◆ **Algoritmo di openMosix**
 - Se un nodo e' disponibile a ricevere processi, NON lo dice a tutti gli altri nodi, ma SOLO a qualche nodo a caso. In tal modo, il nodo libero non viene 'assalito' da tutti gli altri, ma solo da qualche nodo. Questo sistema garantisce un traffico di rete contenuto, assicurando al contempo che dopo un certo periodo il cluster sarà bilanciato, cioè ogni nodo avrà ricevuto una quantità di processi omogenea al carico degli altri nodi ed alla propria capacità computazionale.

Dove migliorare il cluster (2)

- ◆ **Hardware**

- Qui la soluzione e' banale:

- incrementi di RAM
 - upgrade di CPU
 - utilizzo di schede di rete piu' veloci
 - utilizzo di dischi SCSI

- ◆ sono tendenzialmente migliorativi per le capacità finali del cluster.

Dove migliorare il cluster (3)

◆ **Gli applicativi**

- Un cluster openMosix su cui gira UN solo processo e' INUTILE: la potenzialità di openMosix risulta evidente quando ci sono decine/centinaia di processi che impegnano al massimo la capacità dei nodi. Tanti piccoli processi (idealmente, uno per ogni nodo) verranno elaborati piu' velocemente di pochi grossi processi.
- lanciate molti istanze dello stesso processo su blocchi di dati piuttosto che lanciare un unico processo sull'intero insieme di dati da elaborare.

Dove migliorare il cluster (4)

- ◆ **Topologia di rete**

- La topologia di rete puo' rappresentare un collo di bottiglia, **ed e' qui che agiremo per migliorare il cluster**

Variabili del test

- ◆ Un cluster di **X** Pc identici a **Y** Mhz, in topologia **Z** equivalgono ad un unico Pc a **W** Mhz quando svolgono il compito **T**.
- ◆ La funzione diventa $W = \text{fn}(X, Y, Z, T)$ dove X, Y e T sono delle costanti e l'unica variabile e' Z, cioe' la topologia di connessione.

Variabili del test - dettaglio

- ◆ $X = 3$ (P2, 64Mram, NIC 10/100Mhz)
- ◆ $Y = 300\text{Mhz}$
- ◆ $T = 24$ job di conversione audio wav->mp3
- ◆ $Z =$ hub 10Mb/s, switch 100Mb/s, P2P

Job eseguiti

- ◆ Sono stati lanciati 24 job identici, al fine di rendere omogeneo il lavoro eseguito.
- ◆ I 24 job convertono 24 file audio da WAV in MP3, generando un traffico di rete totale di circa 2426MB: considerando che un terzo del job rimangono sul server, 1760MB vengono movimentati tra il server ed i due altri nodi.

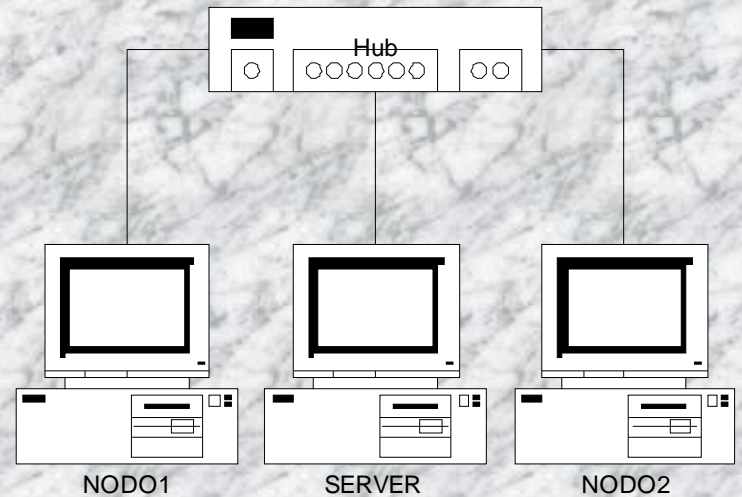
Topologie testate: Hub & Switch

Vantaggi

- **Economico**
- **Di immediato utilizzo**

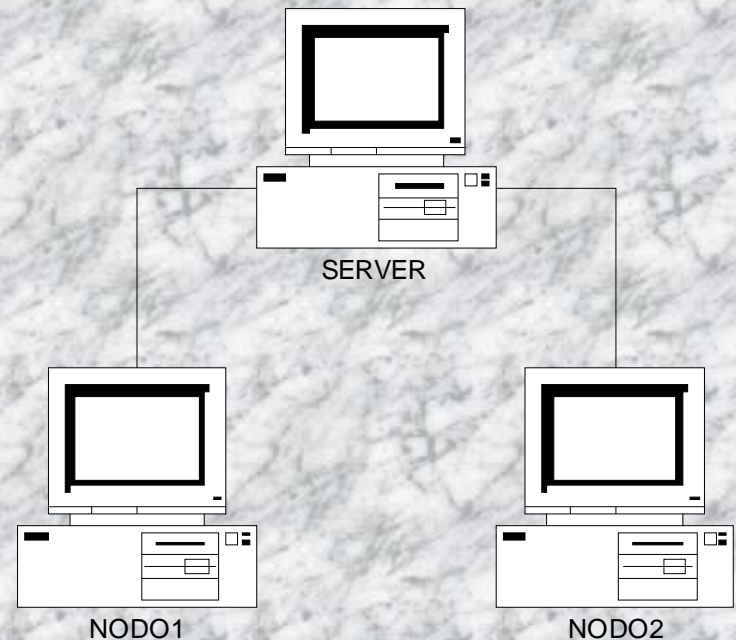
Svantaggi

- **Non reggono un elevato traffico di rete**



Topologie testate: P2P

- Vantaggi
 - **Elevata velocità di connessione**
- Svantaggi
 - **Il nodo master va configurato ad hoc**



P2P: svantaggi

- il master deve essere istruito su come instadare i pacchetti.
- C'e' un limite all'aumento del numero dei nodi dato dal numero di slot PCI presenti sul master da dedicare alle NIC aggiuntive.
- Il master e' **critico** per il funzionamento del cluster

P2P: vantaggi

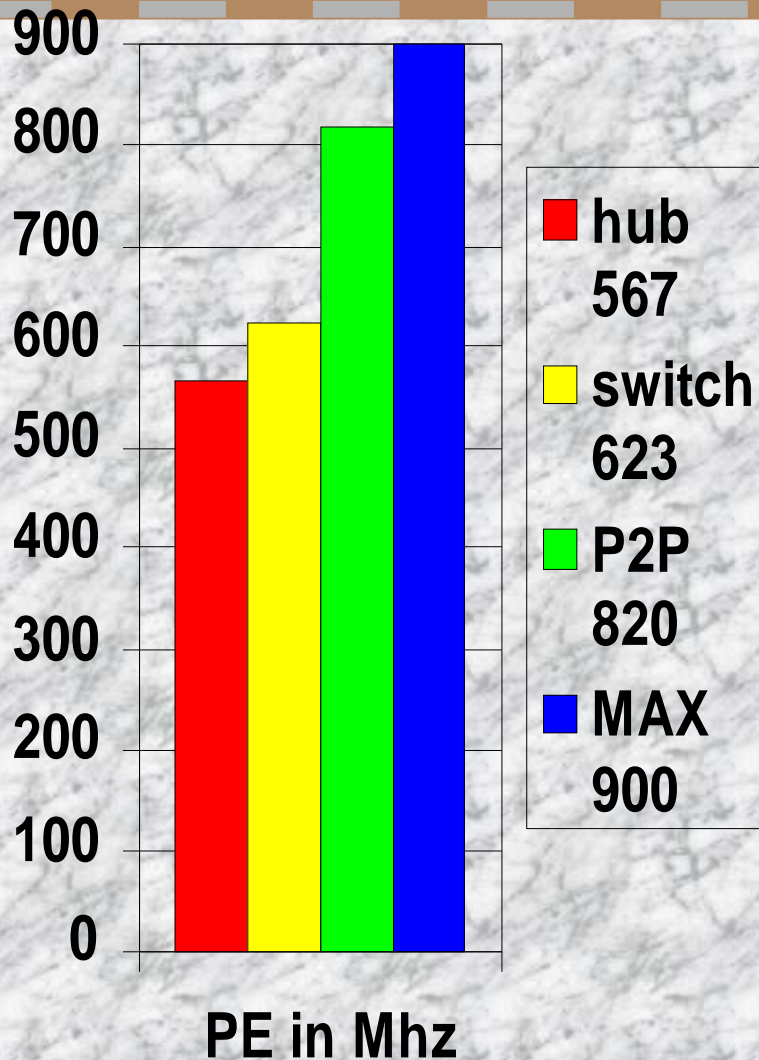
- ◆ Migliori prestazioni quando occorre processare grosse masse di dati e la rete viene saturata.
- ◆ Non e' piu' necessario l'uso di un hub o switch

Risultati Test - Numeri

- PE e Max PE sono espressi in Mhz

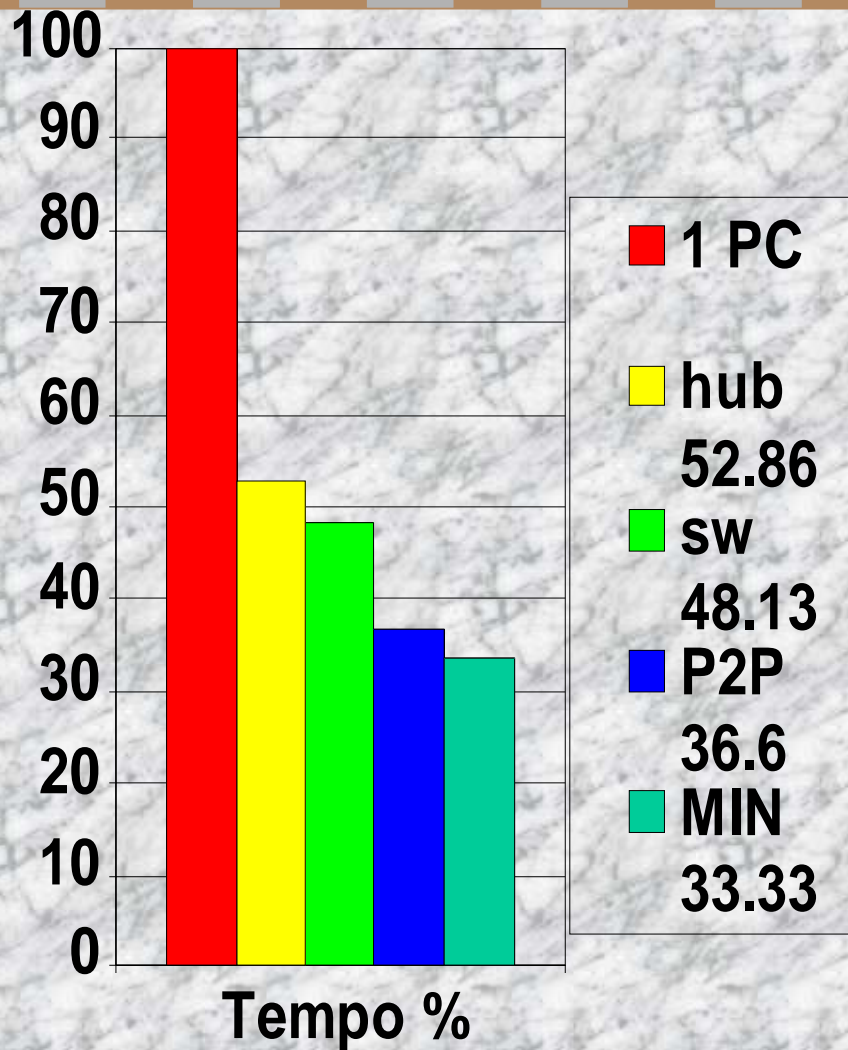
PC #	Topol.	PE (teor)	Max PE	Δ PE	M
3	Hub 10	900	567	-36.94%	0.63
3	Sw 100	900	623	-30.74%	0.69
3	P2P	900	820	-8.88%	0.91

Risultati Test - PE



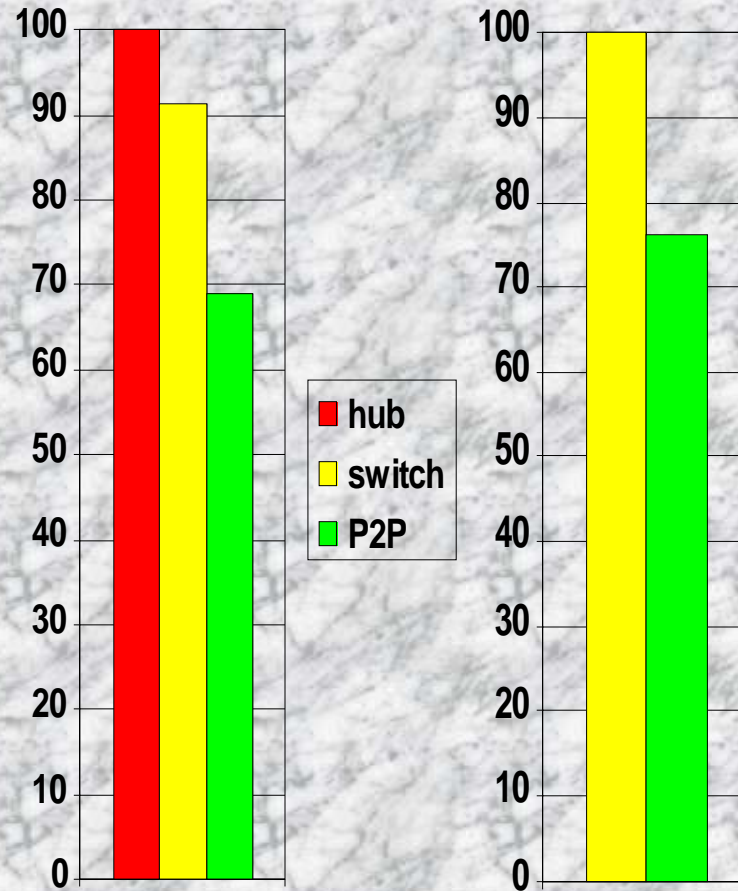
- La PE della topologia P2P e' 820Mhz, vicino ai 900Mhz teorici (e mai raggiungibili)

Risultati Test – Tempi assoluti



- La connessione P2P raggiunge il 36.6%, vicino al minimo 33.33% (teorico e mai raggiungibile)

Risultati Test – Tempi relativi



Hub = 100.00%
Switch = 91.05%
P2P = 69.24%

Switch = 100.00%
P2P = 76.04%

Risultati Test -Percentuali

- ◆ La connessione P2P guadagna in prestazioni il 30% rispetto all'hub ed il 24% rispetto allo switch. La banda di rete e' ben lontana dall'essere saturata, per cui l'aggiunta di altri nodi avrebbe dato un guadagno PE proporzionale o piu' che proporzionale.

Conclusioni

- ◆ E' stato dimostrato che un cluster quando esegue job richiedenti massicci scambi di dati tra i nodi **migliora le proprie prestazioni semplicemente modificando la topologia di rete**

Conseguenze

- ◆ Si puo' notare che a parita' di compito da svolgere, un cluster ottimizzato permette di avere gli stessi risultati
 - piu' velocemente, se si mantiene lo stesso numero di nodi
 - piu' economicamente, se si tolgano i nodi resi superflui dall'ottimizzazione (risparmio di spazio, corrente elettrica, manutenzione etc.)

Riferimenti



- ◆ <http://openmosix.sourceforge.net>
- ◆ roberto.premoli@tiscali.it
- ◆ www.scomodo.com/~roberto
- ◆ www.openbrains.org/roberto