

VirtuaLinux

Una soluzione open source per il clustering HPC

<http://sourceforge.net/projects/virtuallinux/>

M. Aldinucci, M. Torquati
Dipartimento di Informatica, Università di Pisa

Guida all'ascolto

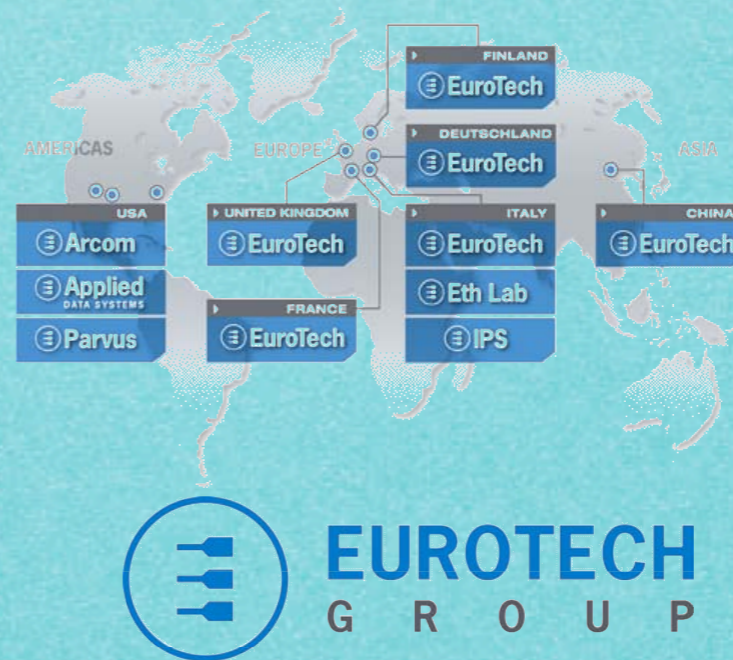
- ▶ **VirtuaLinux - concetti di base**
 - ▶ genesi e motivazioni
 - ▶ quali problemi affronta
 - ▶ come li risolve
 - ▶ architettura
- ▶ **VirtuaLinux - approfondimenti**
 - ▶ high-availability
 - ▶ boot si sistema senza master
 - ▶ virtualizzazione efficiente dello storage
- ▶ **Esperimenti e demo**
 - ▶ Versione “many-tier” in esclusiva per NSS 07
- ▶ **Riferimenti e conclusioni**

Il progetto Virtualinux Ricerca & Industria (Italiana!)

- ▶ Il progetto VirtuaLinux
 - ▶ finanziato da Eurotech S.p.A.
 - ▶ finalizzato ad aggredire problemi di ricerca industriale
 - ▶ che necessitano anche di ricerca di base
 - ▶ risultati scientifici pubblicati in conferenze internazionali
 - ▶ tutto il software rilasciato come open source GPL



*Dipartimento di Informatica
Università di Pisa*



Quali problemi affronta

▶ Cluster

- ▶ cioè un insieme di macchine indipendenti (eventualmente ad alta densità) connesse da una rete veloce
- ▶ sono fragili
 - ▶ il nodo master è un *single point of failure*
 - ▶ i dischi montati sui nodi sono una comune fonte di guasti
- ▶ sono complessi da installare e mantenere
 - ▶ l'installazione e la configurazione può richiedere giorni
 - ▶ sono necessari sistemisti esperti
 - ▶ la potenza dell'account di root è fonte di grossi problemi
- ▶ un singolo SO non è adeguato alle esigenze di tutti gli utenti
 - ▶ ... a me serve Ubuntu, a me piace CentOS, io *credo* in UindoZ ...

Virtualizzazione: una soluzione nuova nuova ...

Christopher Strachey published a paper titled *Time Sharing in Large Fast Computers* in the International Conference on Information Processing at UNESCO, New York, in June, 1959. Later on, in 1974, he clarified in an email to Donald Knuth that:

" ... [my paper] was mainly about multi-programming (to avoid waiting for peripherals) although it did envisage this going on at the same time as a programmer who was debugging his program at a console. I did not envisage the sort of console system which is now so confusingly called time sharing." Strachey admits, however, that "time sharing" as a phrase was very much in the air in the year 1960.

Robert P. Goldberg describes the then state of things in his 1974 paper titled *Survey of Virtual Machines Research*. He says: "Virtual machine systems were originally developed to correct some of the shortcomings of the typical third generation architectures and multi-programming operating systems - e.g., OS/360."

Christopher Strachey

New York, in June, 1959

Robert P. Goldberg

1974 paper titled *Survey of Virtual Machines Research*

Virtualizzazione: una soluzione nuova nuova ...

Christopher Strachey published a paper titled *Time Sharing in Large Fast Computers* in the International Conference on Information Processing at UNESCO, New York, in June, 1959. Later on, in 1974, he clarified in an email to Donald Knuth that:

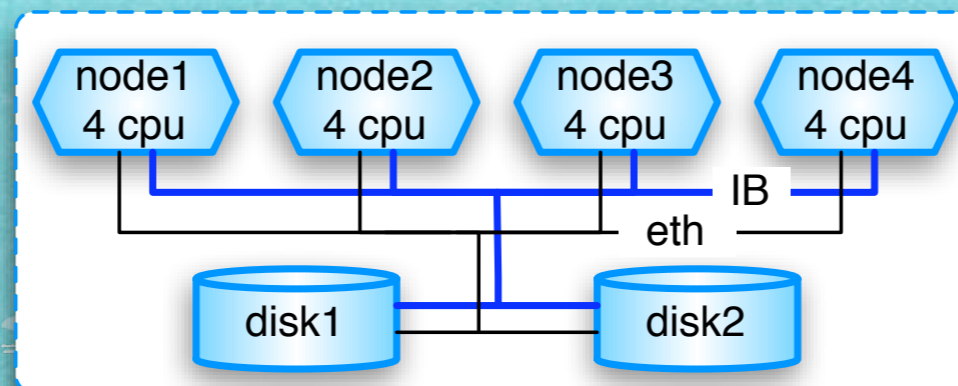
" ... [my paper] was mainly about multi-programming (to avoid waiting for peripherals) although it did envisage this going on at the same time as a programmer who was debugging his program at a console. I did not envisage the sort of console system which is now so confusingly called time sharing." Strachey admits, however, that "time sharing" as a phrase was very much in the air in the year 1960.

Robert P. Goldberg describes the then state of things in his 1974 paper titled *Survey of Virtual Machines Research*. He says: "Virtual machine systems were originally developed to correct some of the shortcomings of the typical third generation architectures and multi-programming operating systems - e.g., OS/360."

- ▶ Comunque, funziona (si tratta di *astrazione*, alla fine)
 - ▶ high-level (e.g. JVM)
 - ▶ medium-level (e.g. FreeBSD jails)
 - ▶ low-level (Simulazione [e.g. Cell], Binary translation [e.g. WMware, QEMU, ...], paravirtualization [XEN, KVM, ...])
- ▶ rende possibile
 - ▶ consolidare diversi SO sullo stesso HW
 - ▶ condividere risorse HW (e SW)
 - ▶ isolare classi di SO (user, admin, etc)

Cluster: La configurazione classica

- ▶ High-density blades + storage esterno (SAN o NAS)
 - ▶ le SAN-RAID sono efficienti e robuste
 - ▶ l'efficienza e la robustezza delle SAN sono garantite indipendentemente dal SO (spesso al livello HW)
 - ▶ talvolta richiesta per legge (e.g. legge USA Sarbenes-Oxley)
- ▶ Una pletora di distribuzioni Linux che affrontano il problema già esistono
 - ▶ ma richiedono la personalizzazione del sistema operativo
 - ▶ tipicamente la configurazione dei path e dei servizi
 - ▶ complessa e non standard specialmente per la fase di boot (initrd)



*Physical Cluster + external SAN
InfiniBand + Ethernet
4 Nodes x 4 CPUs
Cluster InfiniBand 192.0.0.0/24
Cluster Ethernet 192.0.1.0/24
Internet Gateway 131.1.7.6*

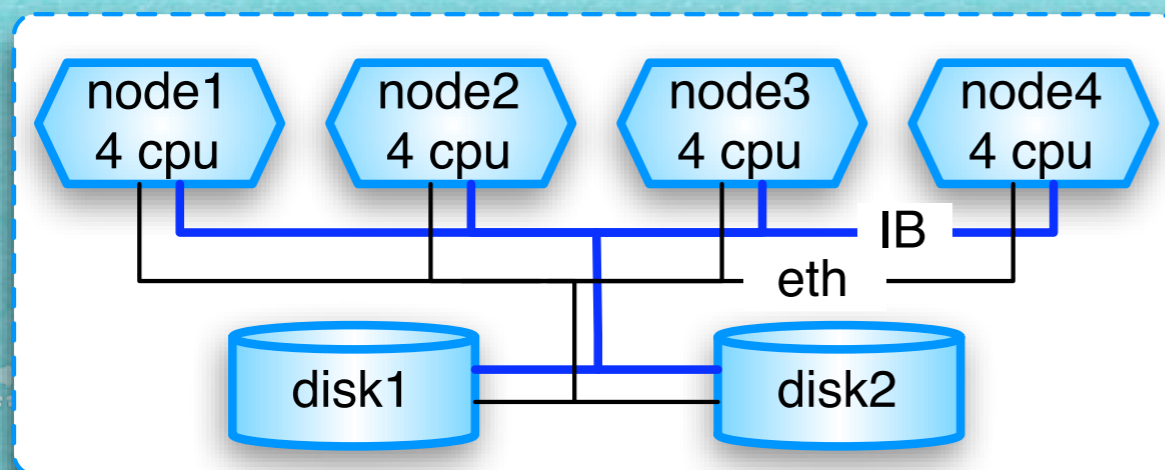
L'approccio VirtualLinux

- ▶ VirtualLinux è concettualmente una meta-distribuzione
 - ▶ scegli una distribuzione esistente e configurala in modo opportuno
- ▶ Master-less
 - ▶ il cluster non ha un nodo master (o meglio lo sono tutti)
- ▶ Disk-less
 - ▶ ogni nodo del cluster ha disposizione uno o più dischi privati (e condivisi) realizzati come volumi logici di una SAN (via iSCSI)
 - ▶ il SO non richiede modifiche sostanziali
- ▶ Supporta in modo trasparente Clusters Virtuali (VC)
 - ▶ fornisce un insieme di strumenti per la gestione dei VC
 - ▶ basati su paravirtualizzazione Xen
 - ▶ i VC sono completamente “isolati” uno dall'altro

Cosa è un Virtual Cluster

- ▶ Estensione “naturale” del concetto di macchina virtuale per multicomputer e cluster
- ▶ Un Virtual Cluster
 - ▶ un insieme di macchine virtuali complete
 - ▶ vCPU, vSTORAGE,
 - ▶ una rete virtuale (privata, proprietaria del VC)
 - ▶ uno storage condiviso (all'interno del VC)
 - ▶ il tipo di macchina virtuale è un “parametro”
 - ▶ diverse soluzioni possibili: VMware, Xen, QEMU,
 - ▶ tutto il lavoro è indipendente dalla VM
 - ▶ la qualità dei VC migliorana quando la VM migliora

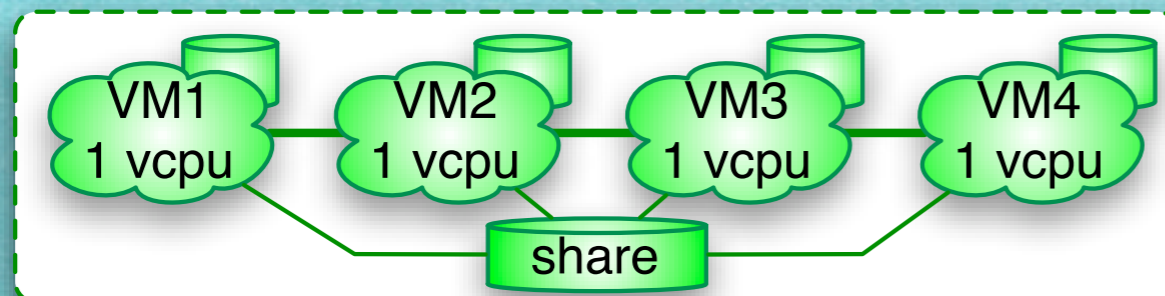
Cluster Virtuali (VC)



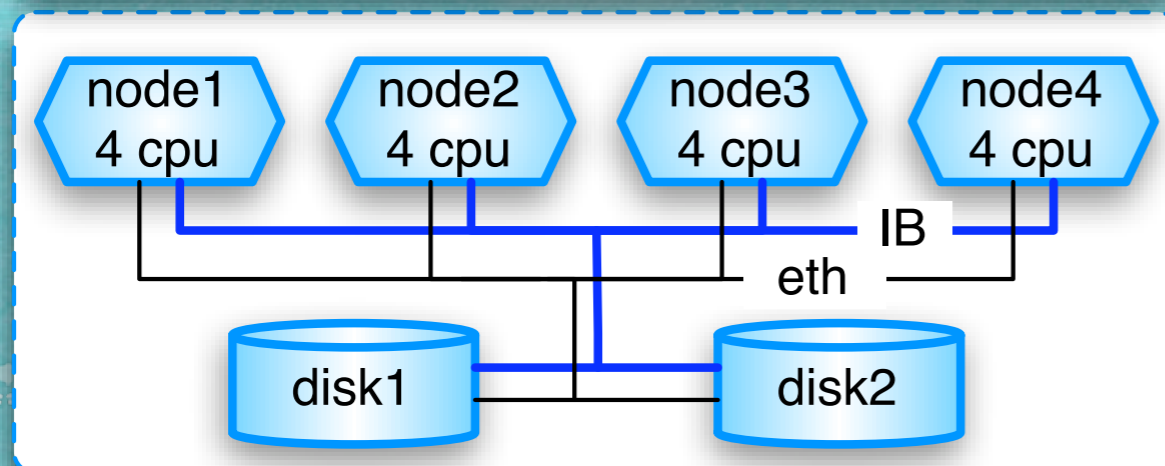
*Physical Cluster + external SAN
InfiniBand + Ethernet
4 Nodes x 4 CPUs
Cluster InfiniBand 192.0.0.0/24
Cluster Ethernet 192.0.1.0/24
Internet Gateway 131.1.7.6*

Cluster Virtuali (VC)

Start VC
green



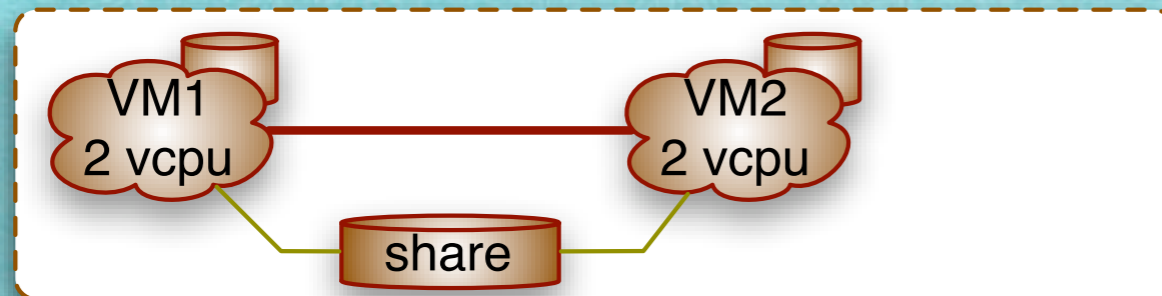
Virtual Cluster "green"
4VMs x 1VCPUs
10.0.3.0/24



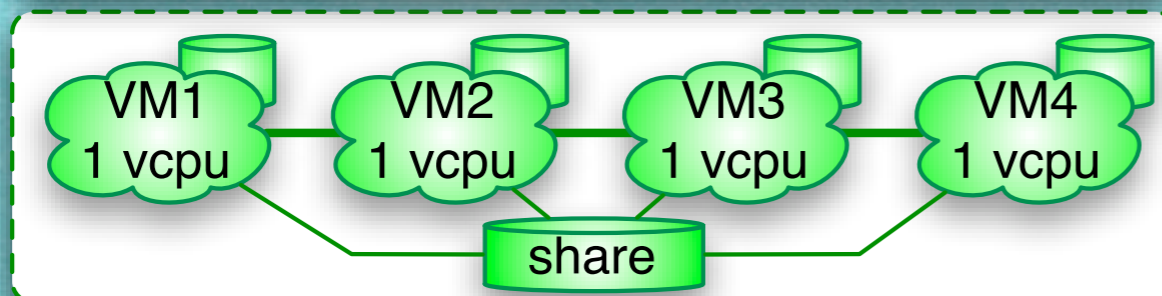
Physical Cluster + external SAN
InfiniBand + Ethernet
4 Nodes x 4 CPUs
Cluster InfiniBand 192.0.0.0/24
Cluster Ethernet 192.0.1.0/24
Internet Gateway 131.1.7.6

Cluster Virtuali (VC)

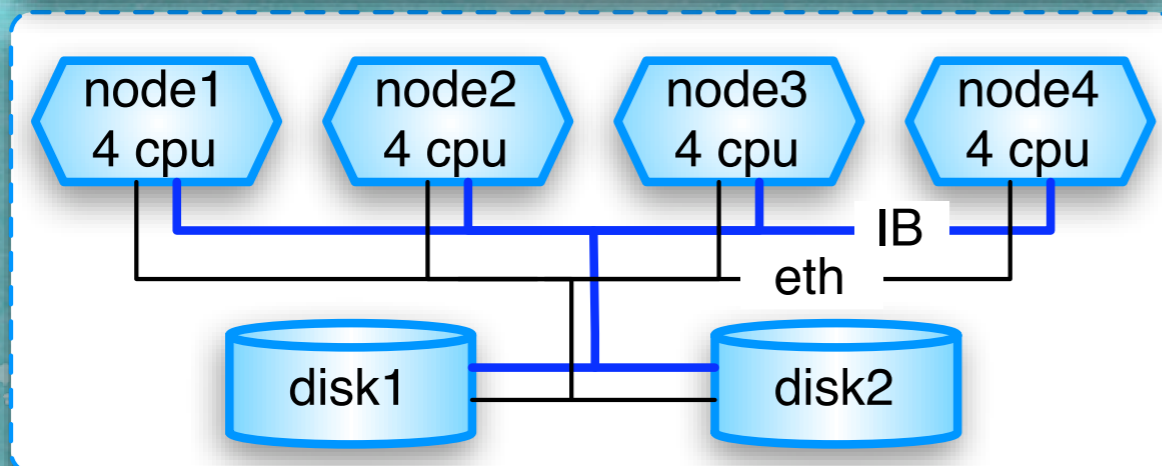
Start VC
tan



Virtual Cluster "tan"
2VMs x 2VCPUs
10.0.1.0/24



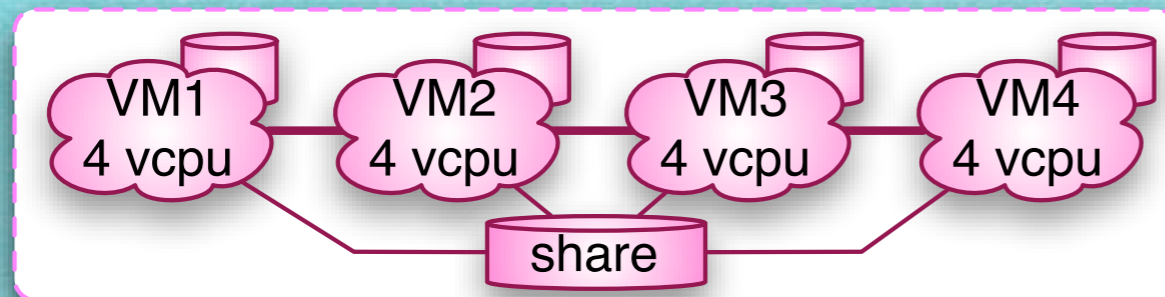
Virtual Cluster "green"
4VMs x 1VCPUs
10.0.3.0/24



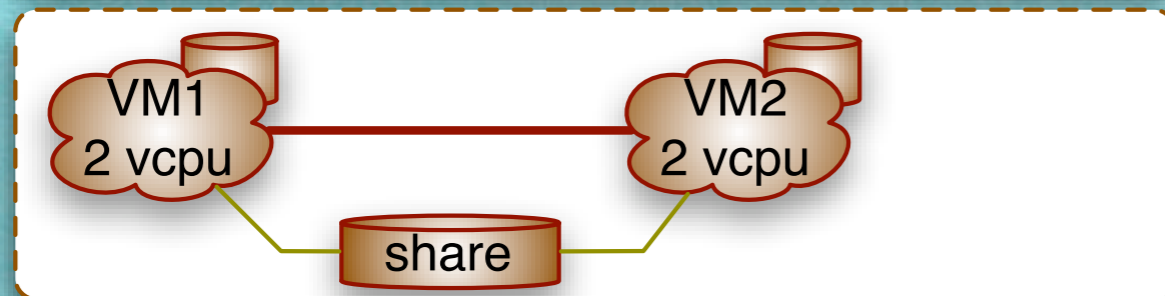
Physical Cluster + external SAN
InfiniBand + Ethernet
4 Nodes x 4 CPUs
Cluster InfiniBand 192.0.0.0/24
Cluster Ethernet 192.0.1.0/24
Internet Gateway 131.1.7.6

Cluster Virtuali (VC)

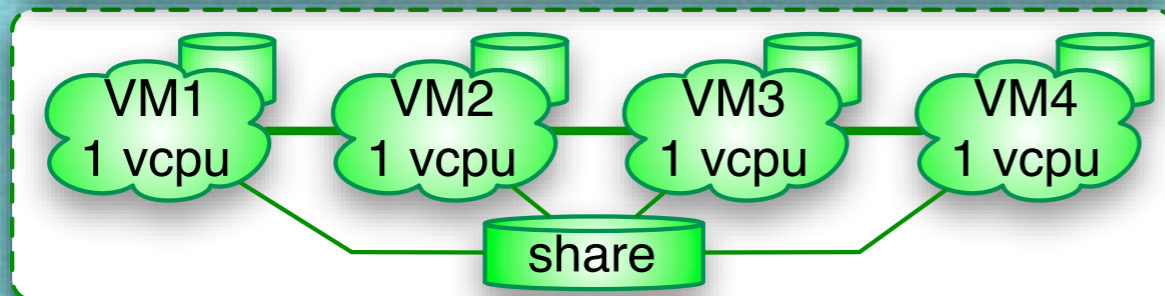
Start VC
pink



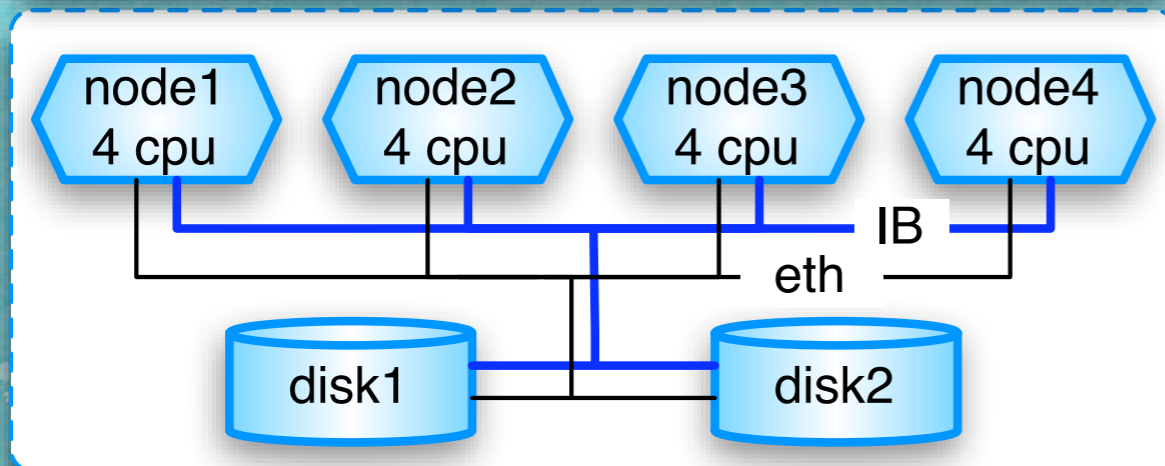
Virtual Cluster "pink"
4VMs x 4VCPUs
10.0.0.0/24



Virtual Cluster "tan"
2VMs x 2VCPUs
10.0.1.0/24



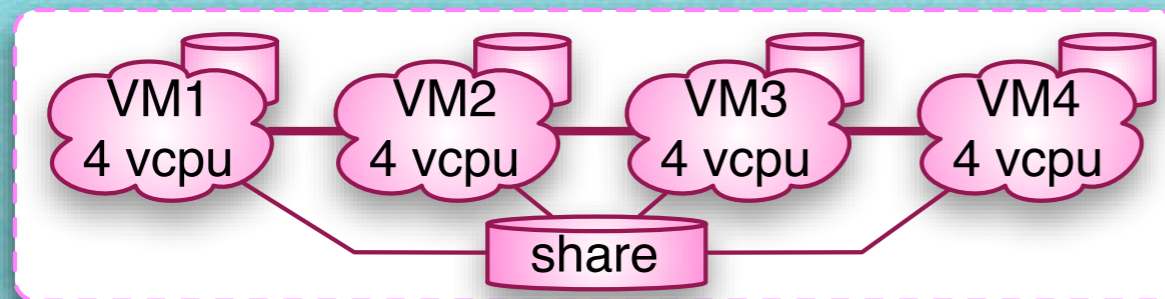
Virtual Cluster "green"
4VMs x 1VCPUs
10.0.3.0/24



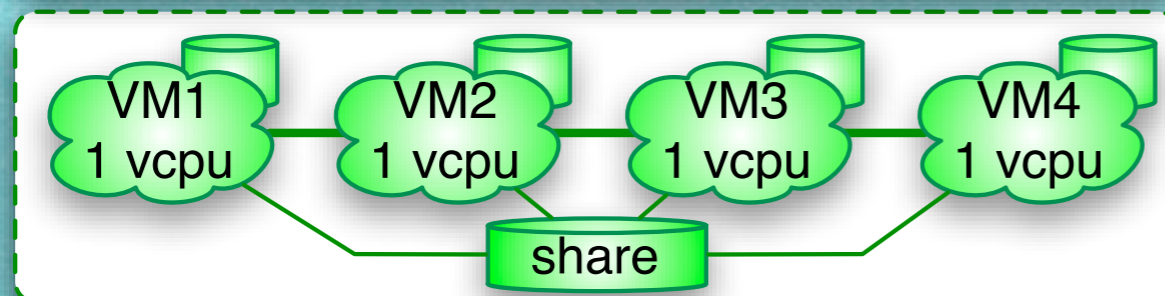
Physical Cluster + external SAN
InfiniBand + Ethernet
4 Nodes x 4 CPUs
Cluster InfiniBand 192.0.0.0/24
Cluster Ethernet 192.0.1.0/24
Internet Gateway 131.1.7.6

Cluster Virtuali (VC)

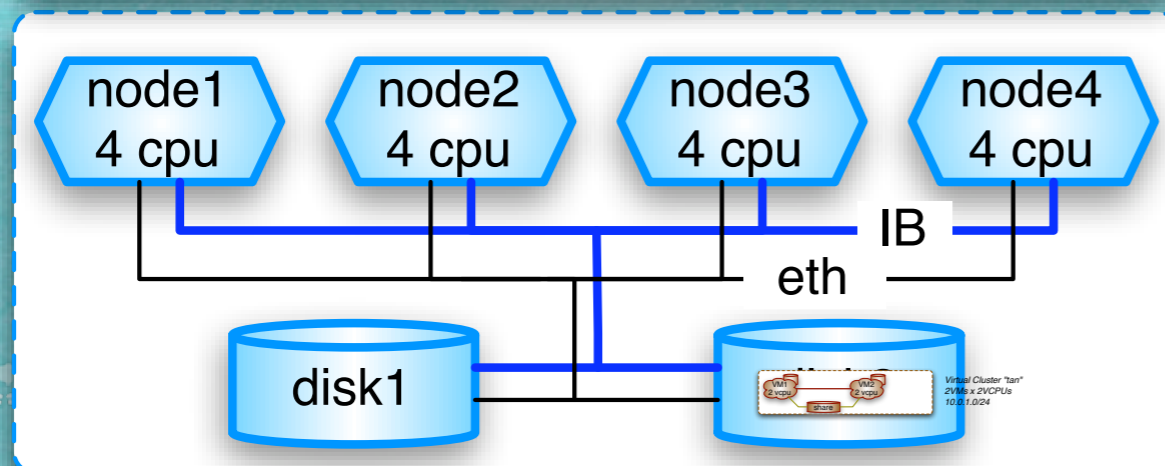
Suspend
tan



Virtual Cluster "pink"
4VMs x 4VCPUs
10.0.0.0/24



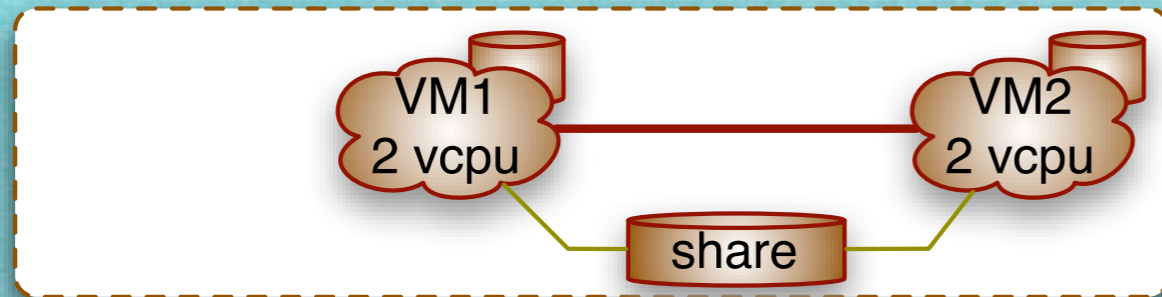
Virtual Cluster "green"
4VMs x 1VCPUs
10.0.3.0/24



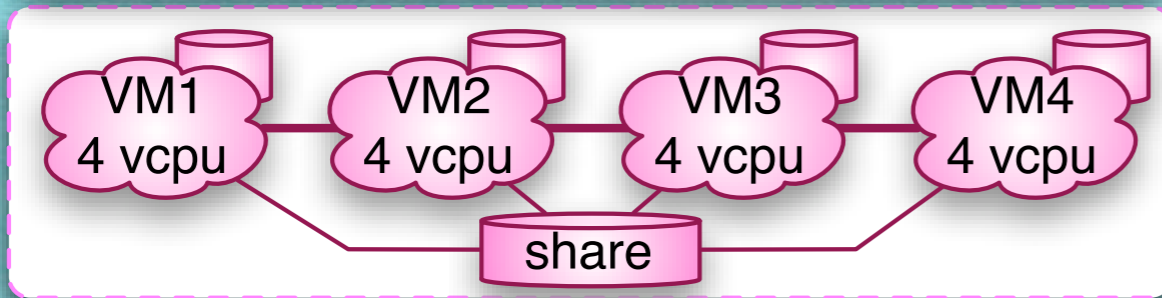
Physical Cluster + external SAN
InfiniBand + Ethernet
4 Nodes x 4 CPUs
Cluster InfiniBand 192.0.0.0/24
Cluster Ethernet 192.0.1.0/24
Internet Gateway 131.1.7.6

Cluster Virtuali (VC)

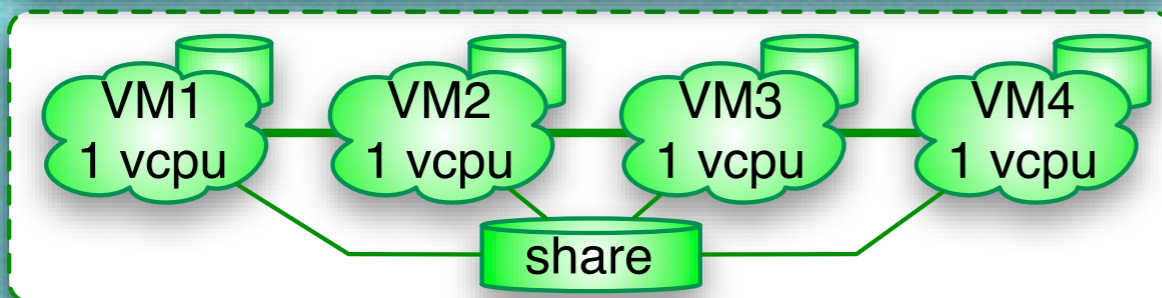
Restart & Remap tan



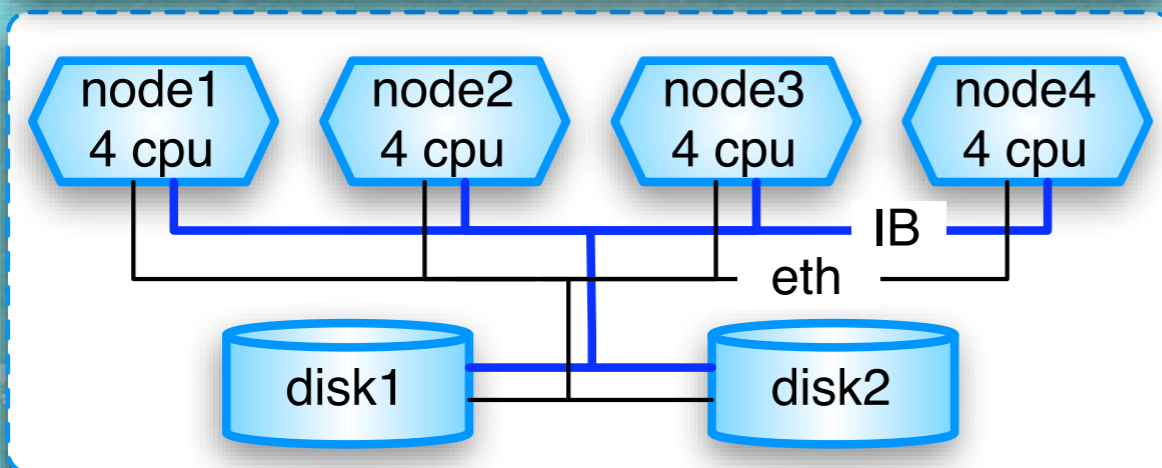
Virtual Cluster "tan"
2VMs x 2VCPUs
10.0.1.0/24



Virtual Cluster "pink"
4VMs x 4VCPUs
10.0.0.0/24

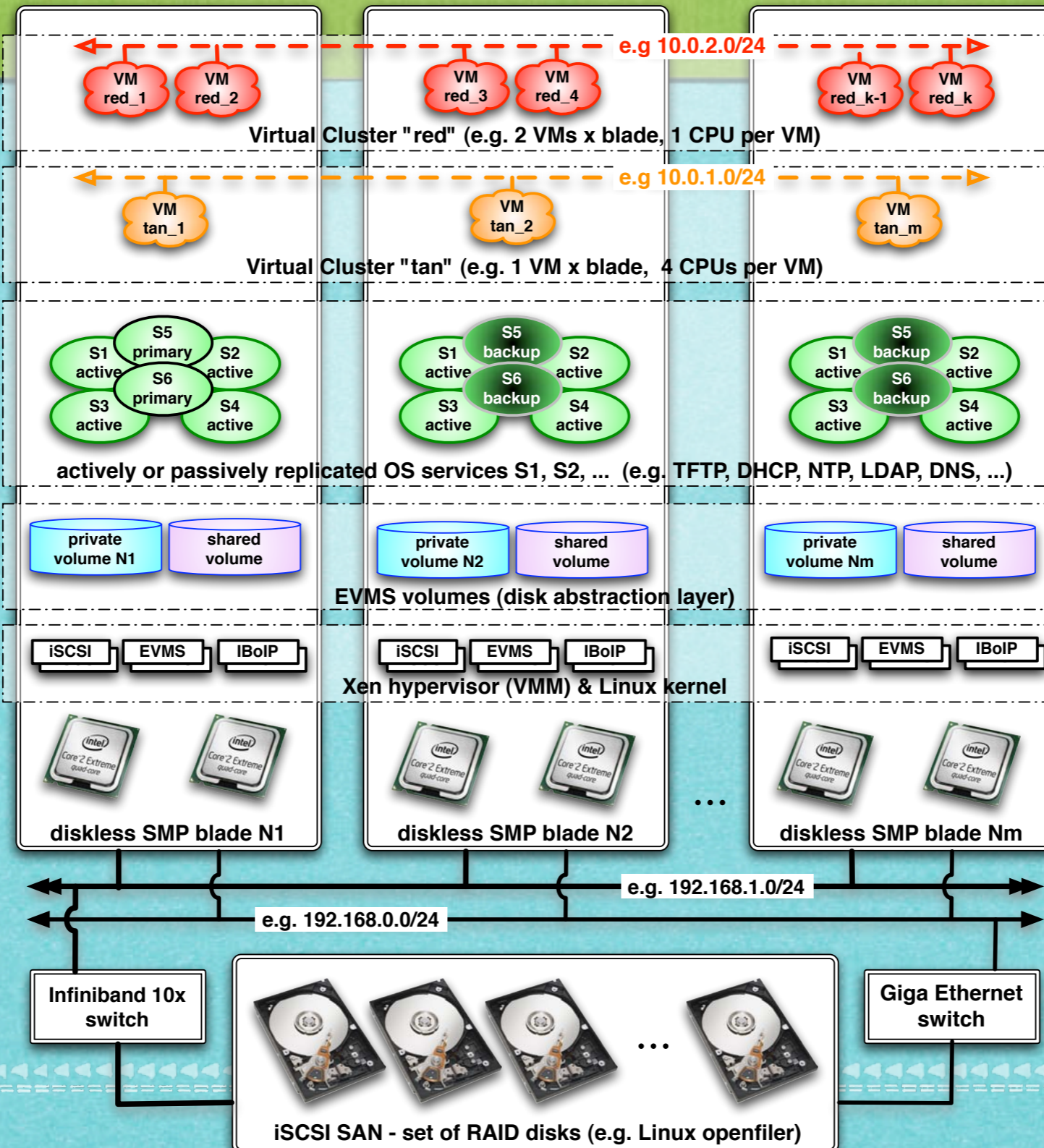


Virtual Cluster "green"
4VMs x 1VCPUs
10.0.3.0/24



Physical Cluster + external SAN
InfiniBand + Ethernet
4 Nodes x 4 CPUs
Cluster InfiniBand 192.0.0.0/24
Cluster Ethernet 192.0.1.0/24
Internet Gateway 131.1.7.6

Visione d'insieme





Alta disponibilità

(High Availability)

mediante replicazione attiva e passiva

Alta disponibilità

- ▶ 24/7 availability
- ▶ Hardware robusto
 - ▶ 5 alimentatori, 4 network switch indipendenti, ...
 - ▶ iSCSI-over-Infiniband o Fibre channel RAID
- ▶ Replicazione dei servizi
 - ▶ Tutti i nodi sono identici, non c'è un nodo master
 - ▶ I servizi essenziali del SO sono replicati su tutti i nodi
 - ▶ Ogni blade può essere spento in ogni momento o subire un guasto senza impatto sulla stabilità generale del sistema

Servizi replicati

Service	FT model	Notes
DHCP	active	Pre-defined map between IP and MAC
TFTP	active	All copies provide the same image
NTP	active	Pre-defined external NTPD fallback via GW
IB manager	active	Stateless service
DNS	active	Cache-only
LDAP	service-specific	Service-specific master redundancy
IP GW	passive	Heartbeat with IP takeover (via IP aliasing)
Mail	node-oriented	Local node and relays via DNS
SSH/SCP	node-oriented	Pre-defined keys
NFS	node-oriented	Pre-defined configuration
SMB/CIFS	node-oriented	Pre-defined configuration

```
root (hd0,0)
kernel /boot/vmlinuz-2.4.27-1-386 root=/dev/sda1 ro init=/bin/bash
initrd /boot/initrd.img-2.4.27-1-386
savedefault
boot
```

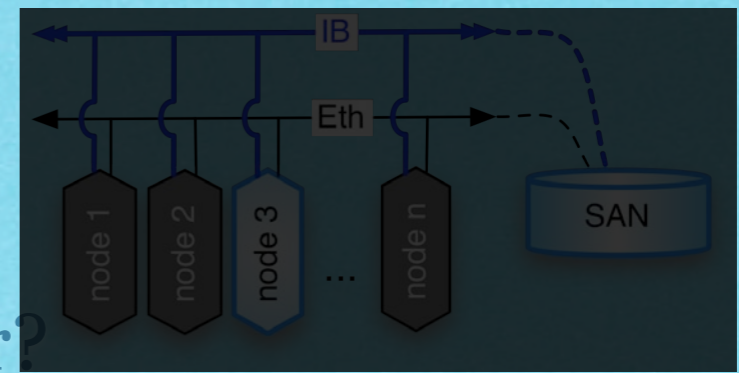
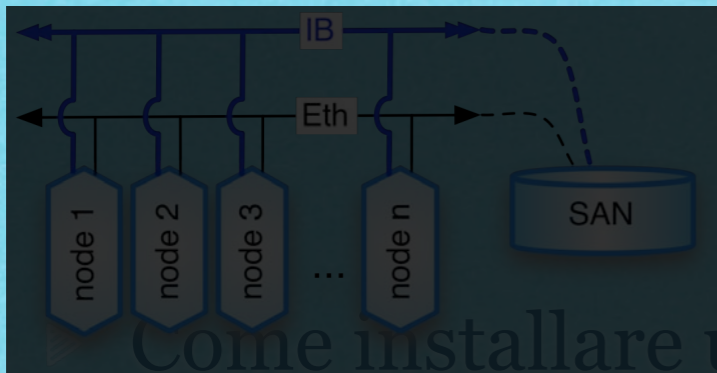
Use the ↑ and ↓ keys to select which entry is highlighted. Press 'b' to boot, 'e' to edit the selected command in the boot sequence, 'c' for a command-line, 'o' to open a new line after ('O' for before) the selected line, 'd' to remove the selected line, or escape to go back to the main menu.

*Una originale sequenza di boot
che supporta il boot di sistemi senza master*

Installare senza master

- ▶ Come installare un cluster senza master?
 - ▶ un punto da cui iniziare deve pur esistere ...
- ▶ Soluzione: **Metamaster**
 - ▶ Un nodo che si comporta temporaneamente come master durante il boot e poi si trasforma in un nodo come gli altri
- ▶ Alla fine dell'installazione tutti i nodi sono identici e i servizi replicati

Installare senza master



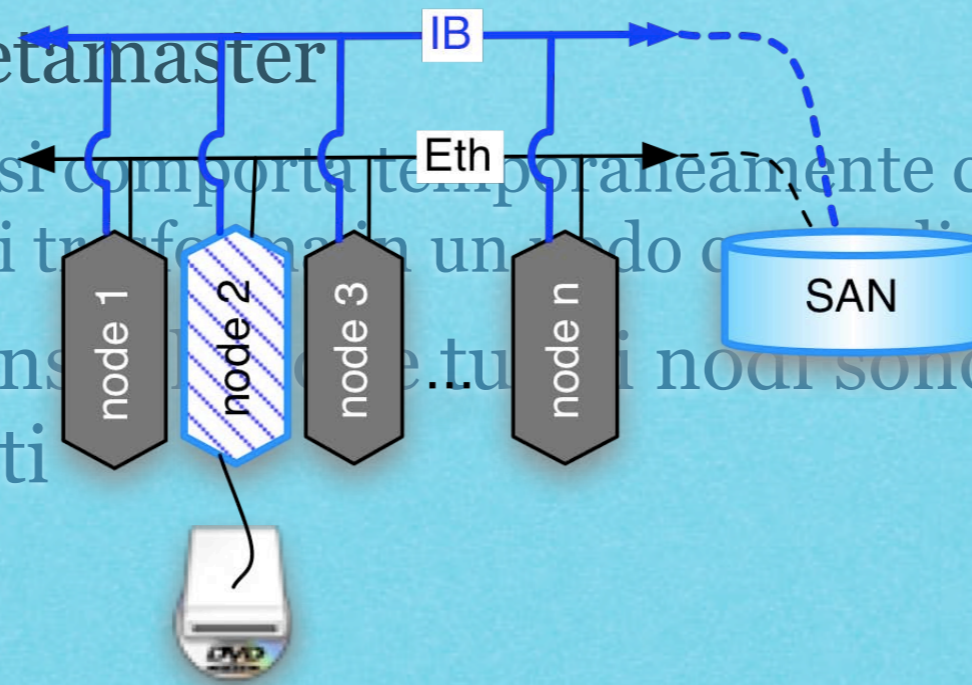
Come installare un cluster senza master?

▶ un punto da cui iniziare deve pur esistere ...

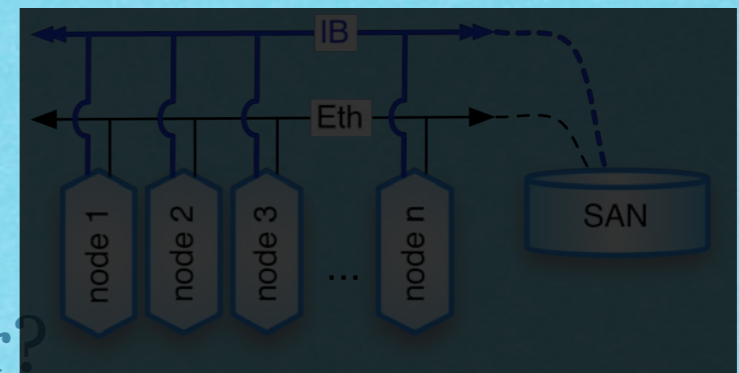
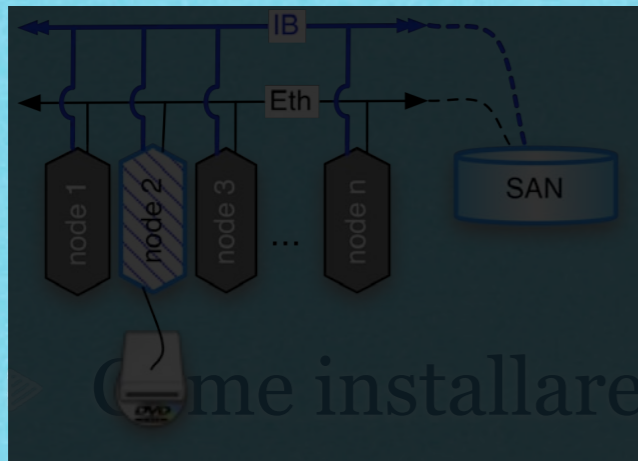
▶ Soluzione: Metamaster

▶ Un nodo che si comporta temporaneamente come master durante il boot e poi si trasferisce in un nodo o in un altro

▶ Alla fine dell'installazione tutti i nodi sono identici e i servizi replicati



Installare senza master



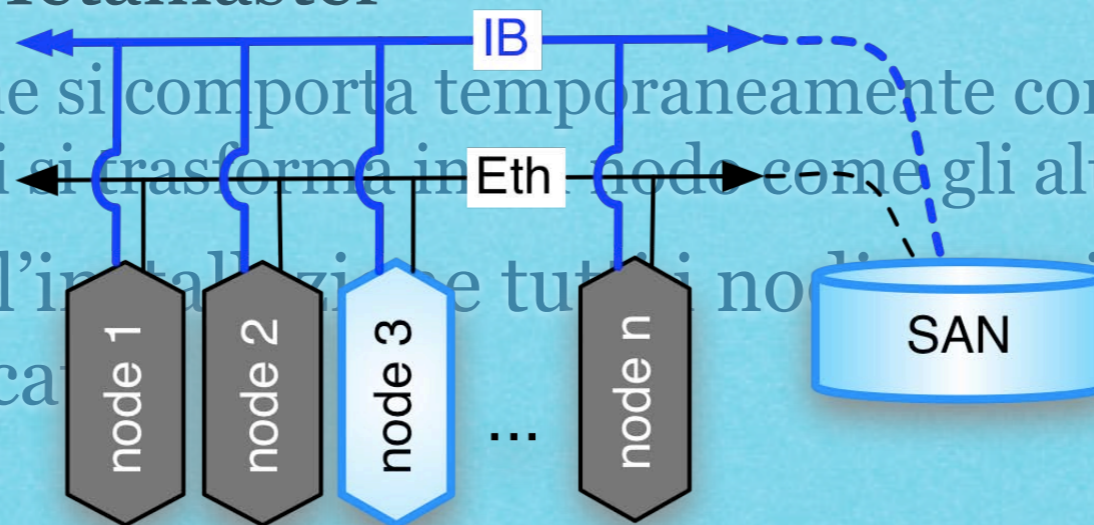
Come installare un cluster senza master?

▶ un punto da cui iniziare deve pur esistere ...

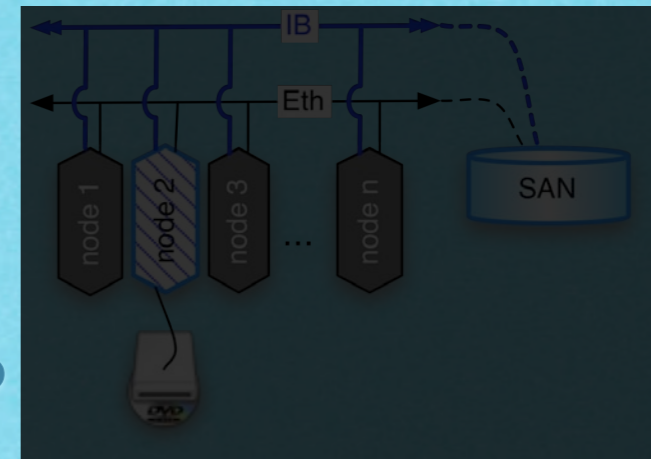
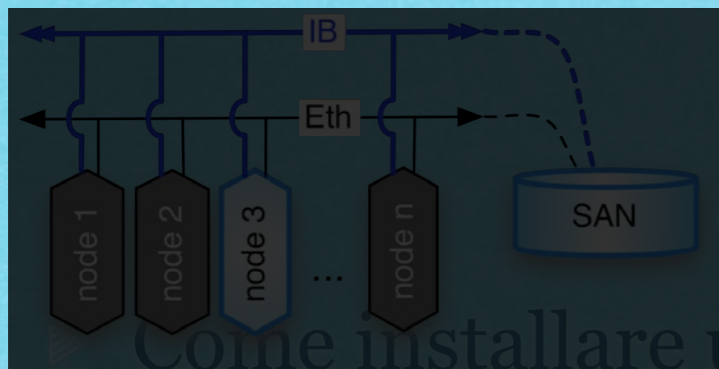
▶ Soluzione: Metamaster

▶ Un nodo che si comporta temporaneamente come master durante il boot e poi si trasforma in un nodo come gli altri

▶ Alla fine dell'installazione tutti i nodi sono identici e i servizi replicati



Installare senza master



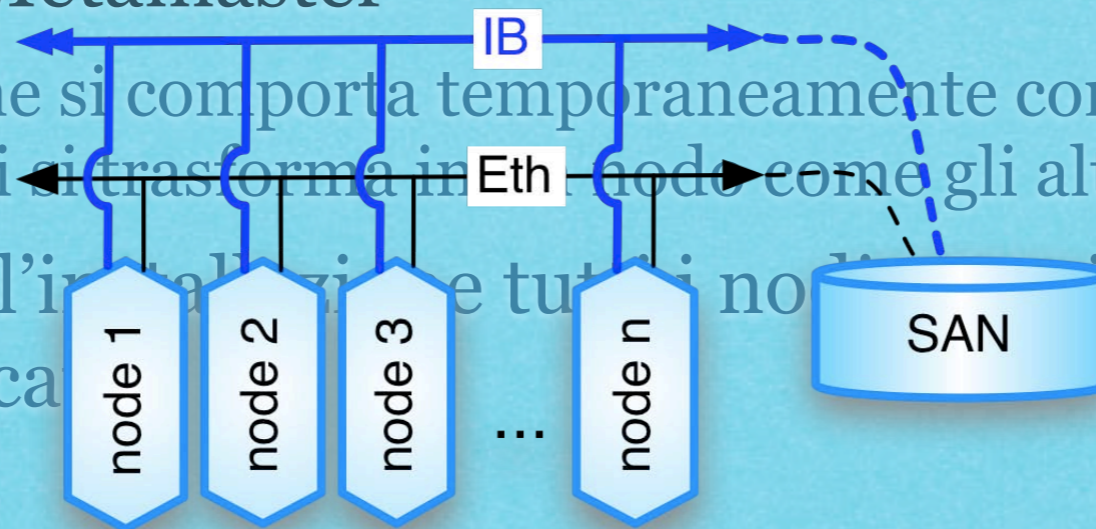
Come installare un cluster senza master?

▶ un punto da cui iniziare deve pur esistere ...

▶ Soluzione: Metamaster

▶ Un nodo che si comporta temporaneamente come master durante il boot e poi si trasforma in un nodo come gli altri

▶ Alla fine dell'installazione tutti i nodi sono identici e i servizi replica

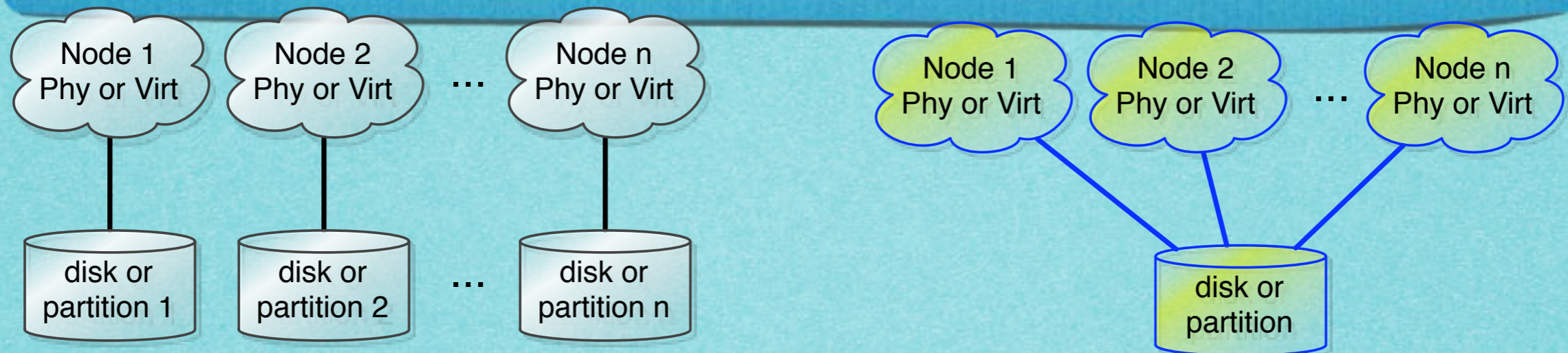




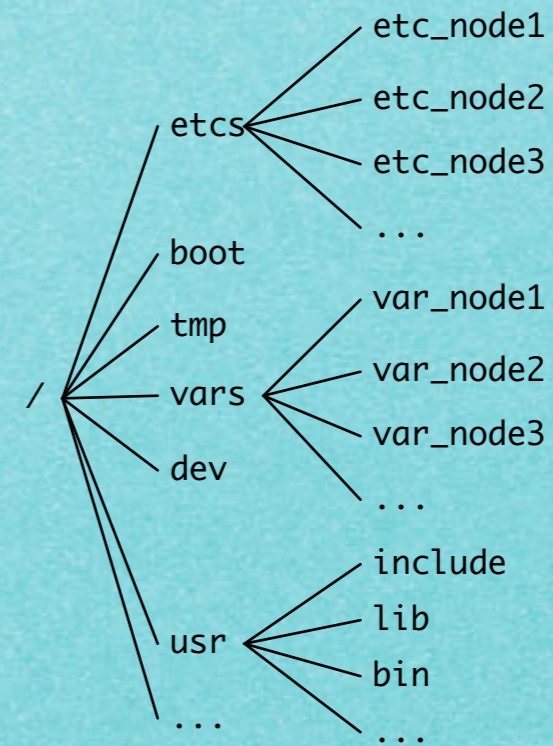
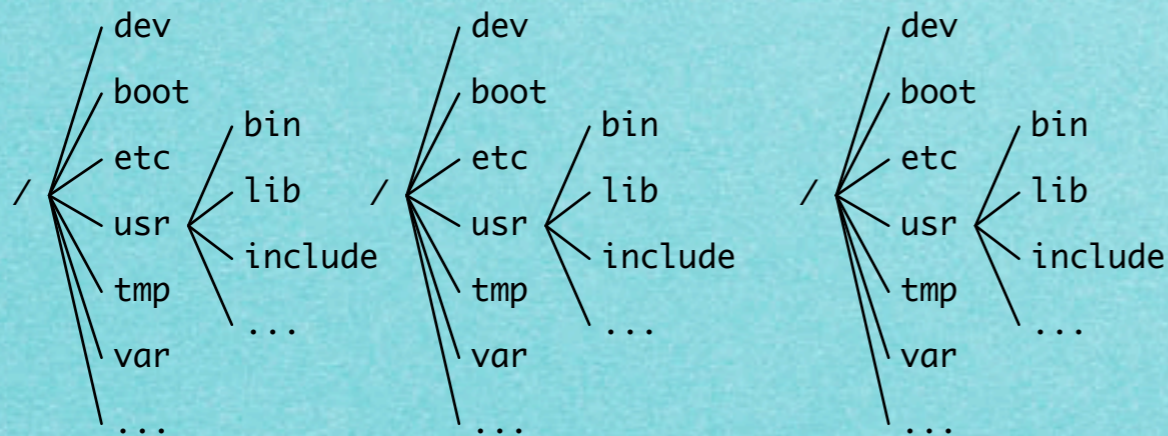
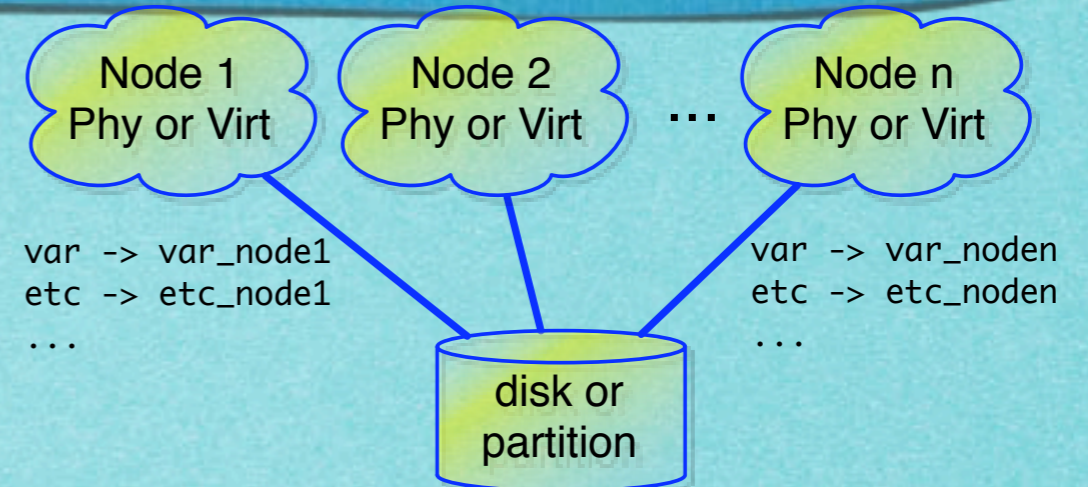
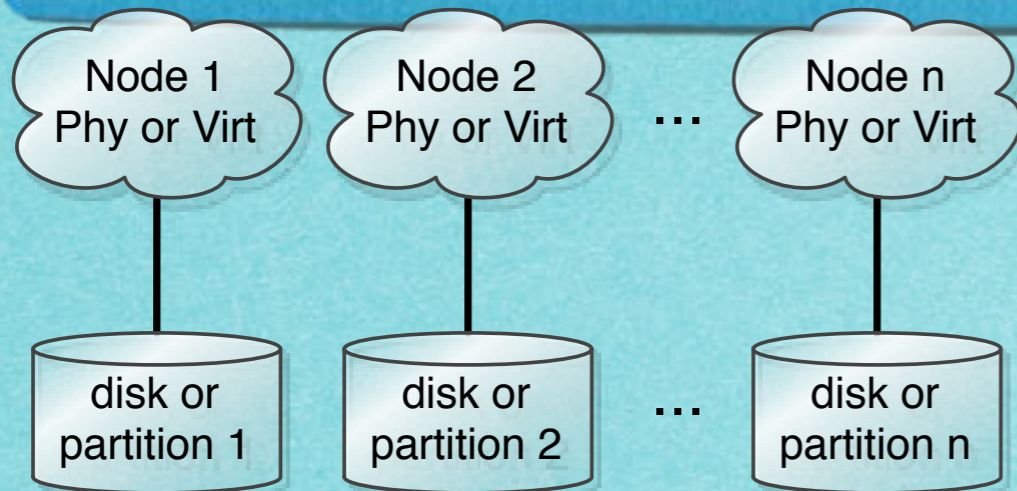
La virtualizzazione dello storage

*una soluzione originale per lo storage di n
volumi di root in tempo-spazio costante*

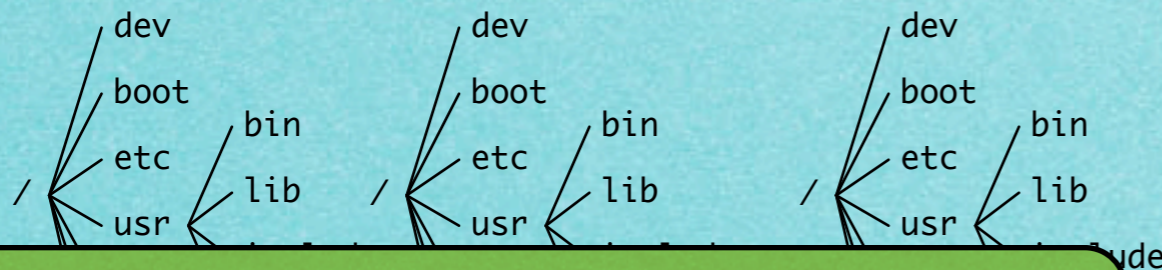
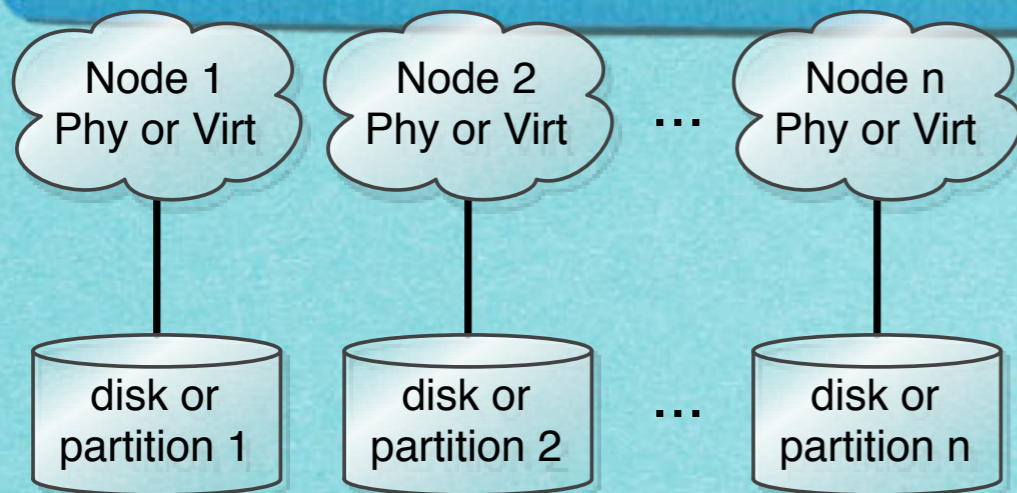
FS: Replicaione completa vs copia unica



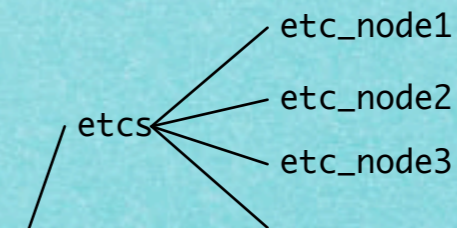
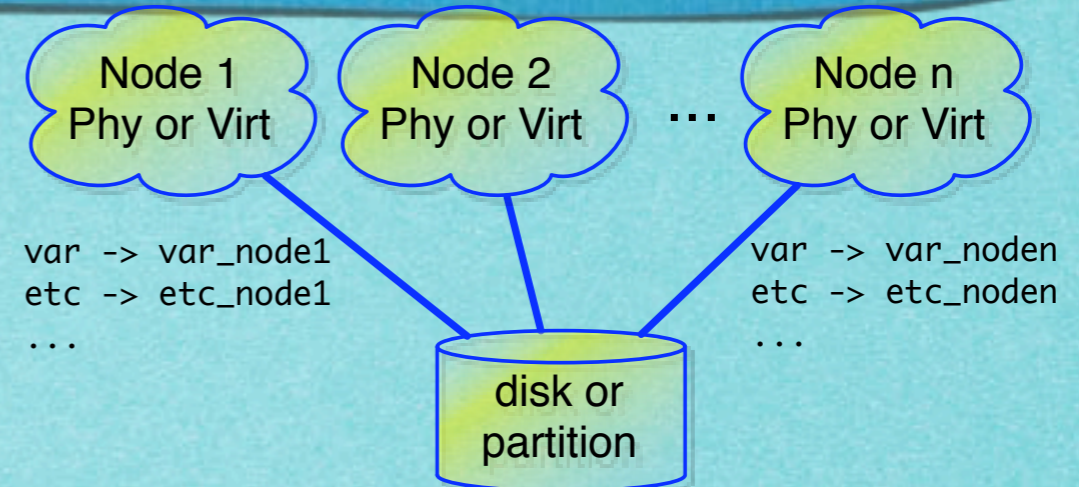
FS: Replicaione completa vs copia unica



FS: Replicaione completa vs copia unica



- ▶ Ogni nodo (fisico o virtuale) contiene una copia indipendente del SO
- ▶ Transparente, facile da installare/mantenere
- ▶ Il SO non necessita modifiche
- ▶ Inefficiente in tempo e spazio - $O(n \cdot \text{size})$
File identici del SO sono replicati



- ▶ Ogni nodo (fisico o virtuale) condivide una copia del SO (al livello di File Sytem)
- ▶ Non trasparente, difficile da installare/mantenere
- ▶ Il SO necessita modifiche
- ▶ Efficiente in tempo e spazio - $O(\text{size})$
File identici del SO non sono replicati

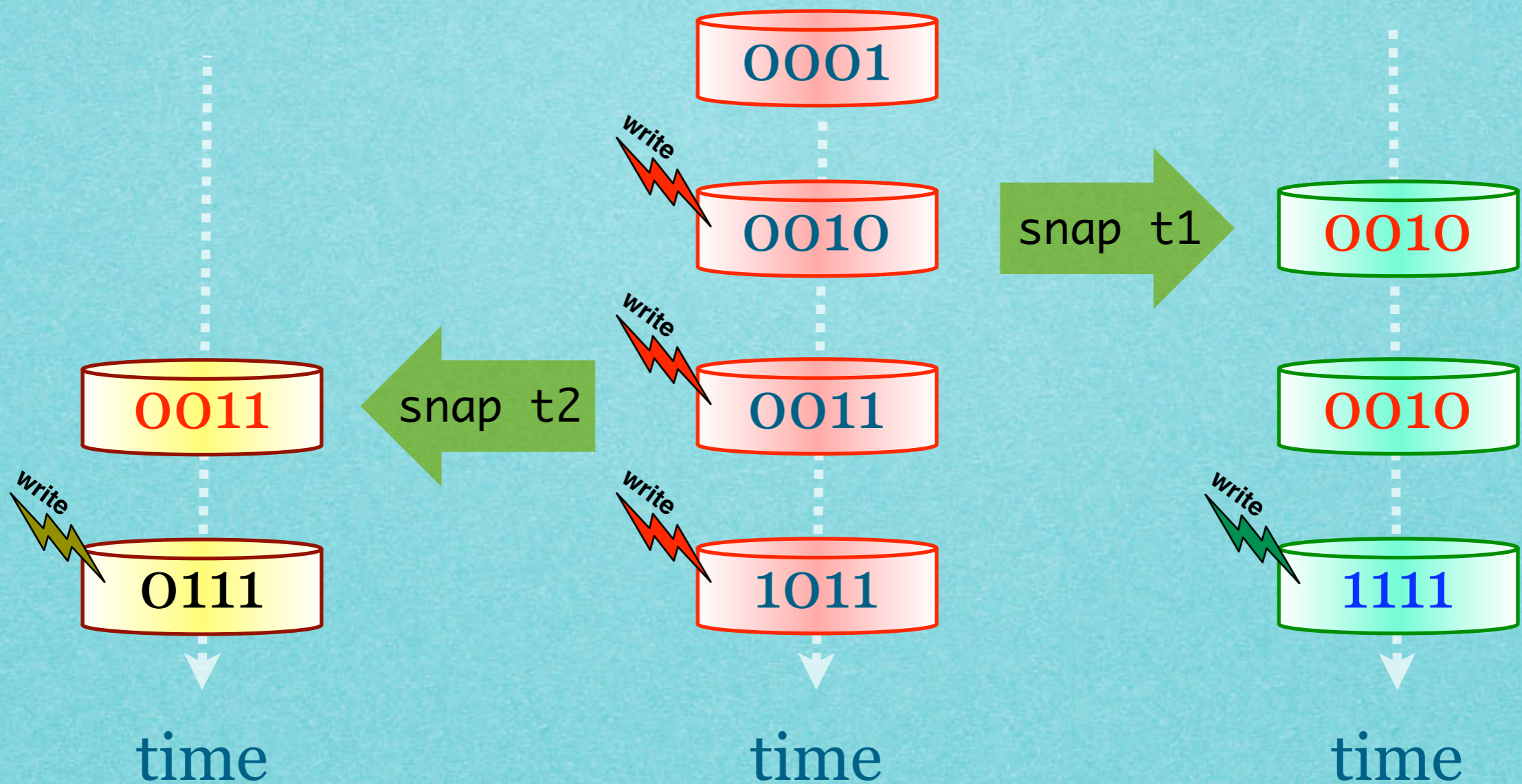
Requisiti per i VC

- ▶ Devono essere trasparenti ed efficienti (tempo e spazio)
 - ▶ devono essere indipendenti dal SO ospite
 - ▶ la creazione dei VC è lenta se affrontata in modo banale
 - ▶ e.g. 50 nodi x 10 GB x 100MB/s = ~ 2 hour
 - ▶ previsione ottimistica
 - ▶ distrugge la banda del disco, il sistema diventa subisce un sovracarico
 - ▶ i VC consumano molto spazio
 - ▶ e.g. 50 nodes x 5 GB = 100 GB
 - ▶ per ogni VC e solo per il SO, senza considerare i dati utente

Caratteristiche distintive dello storage dei VC

- ▶ I nodi di uno stesso VC sono omogenei (stesso SO)
 - ▶ 99% of OS-related files sono identici in tutte le VMs
 - ▶ non c'è ragione di avere VC eterogenei perchè si possono usare diversi VC
 - ▶ è virtualizzazione low-level
 - ▶ il virtualizzato è simile al reale
- ▶ Semplicemente, tenere questi file in copia unica
 - ▶ ma per essere trasparente dobbiamo “dirlo” alle VM, loro credono di essere completamente indipendente l'uno dall'altra
 - ▶ utilizzo nuovo della tecnica dello snapshot
 - ▶ finalizzato alla gestione della concorrenza negli accessi agli snapshot

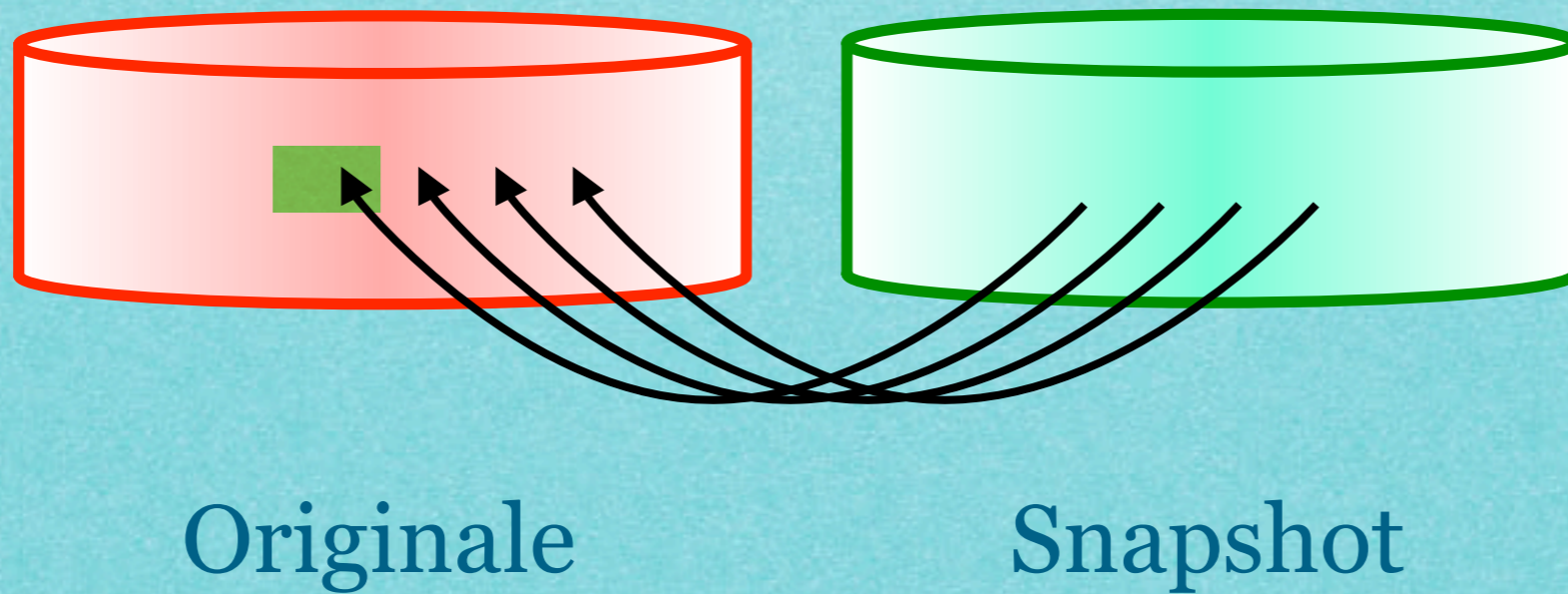
La tecnica dello snapshot



Snapshots

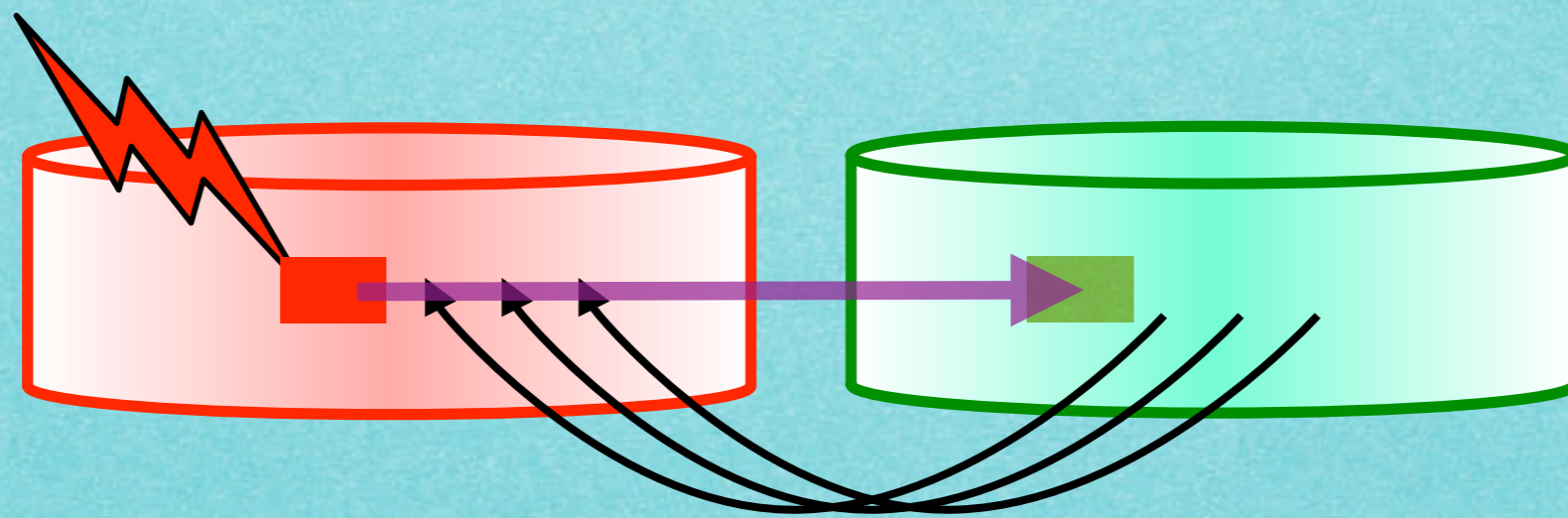
- ▶ Usati per realizzare backup online
 - ▶ Sia l'originale che lo snapshot possono essere modificati
- ▶ Indipendente dal file system, trasparente
- ▶ Supportata in diversi strumenti, e.g. EVMS, LVM2
 - ▶ Implementata dal kernel standard linux
 - ▶ dm_snapshot module (device mapper)
 - ▶ e in diversi altri SO, e.g. sistema di ripristino di Uindoz
- ▶ Può essere implementata in diversi modi
 - ▶ copy-on-write, redirect-on-write, split-mirror, ...

Copy-on-write



Copy-on-write

Scrivere
sull'originale



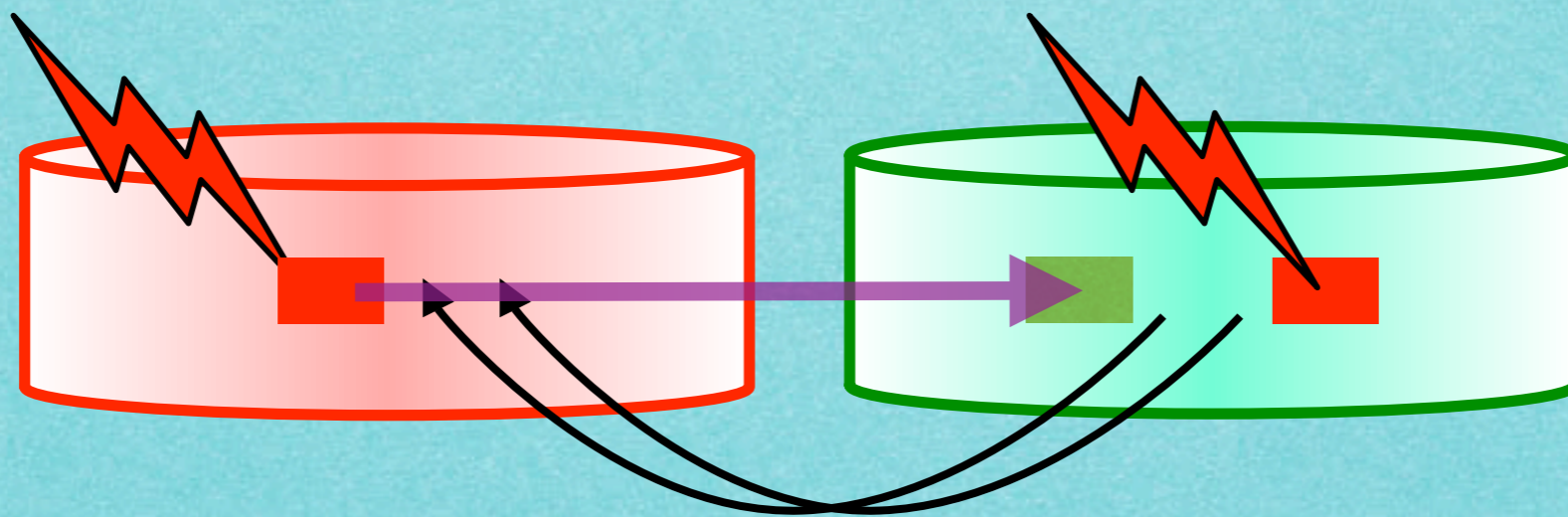
Originale

Snapshot

Copy-on-write

Scrivere
sull'originale

Scrivere
sullo snapshot



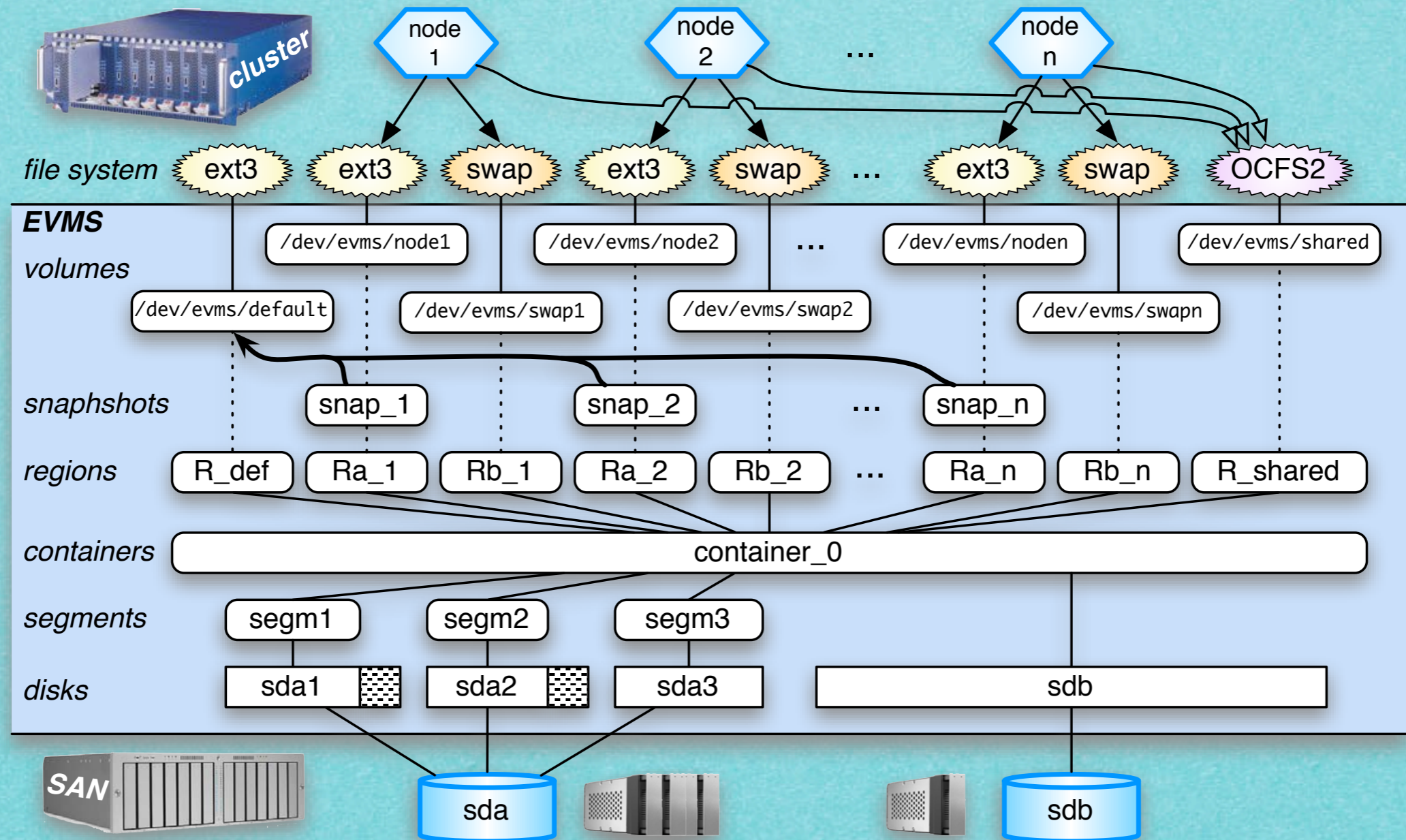
Originale

Snapshot

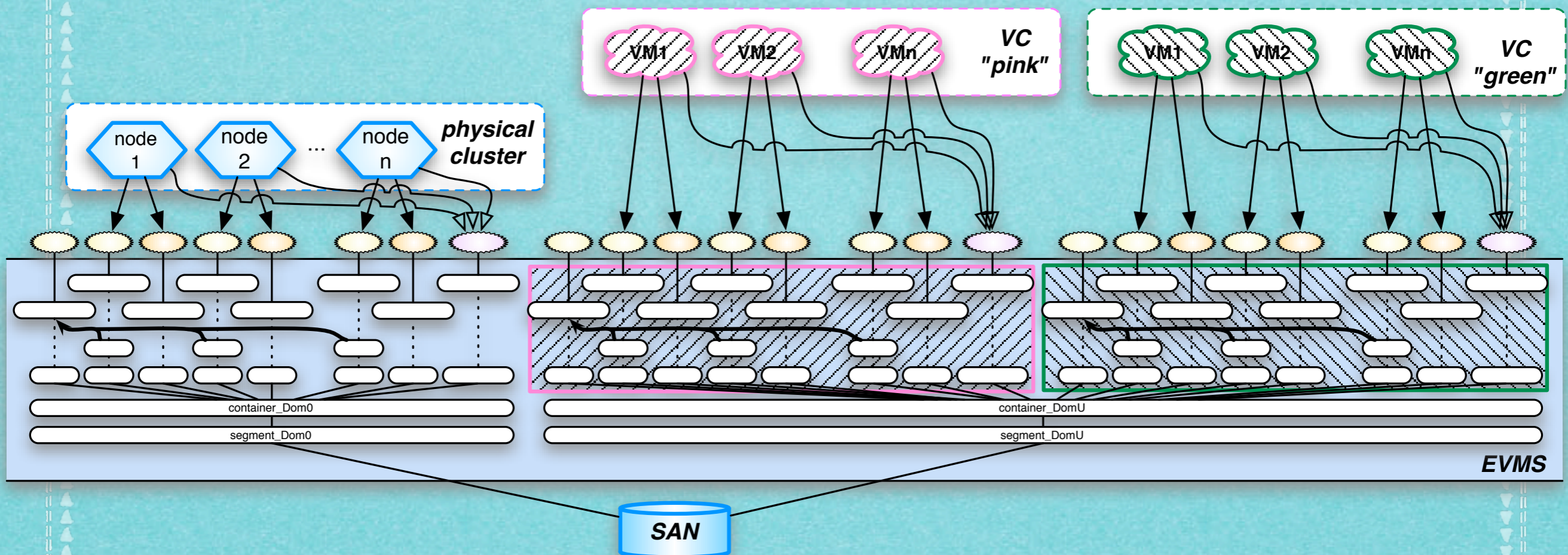
Snapshot concorrenti

- ▶ Gli snapshot non sono progettati per sistemi paralleli
 - ▶ se usati con la semantica standard, tutti gli snapshot devono rimanere sempre attivi su tutti i nodi
 - ▶ linux non può tenere attivi più di 8 snapshot per nodo
- ▶ Una nuova semantica per gli snapshots
 - ▶ rilassa la semantica standard mantenendo la correttezza
 - ▶ “marcare” come read-only parte dei blocchi dei volumi logici
 - ▶ permettere la disattivazione degli snapshot il cui originale rimane costante

VirtualLinux: Virtualizzazione dello storage (cluster fisico)



VirtualLinux: Virtualizzazione dello storage (VCs)

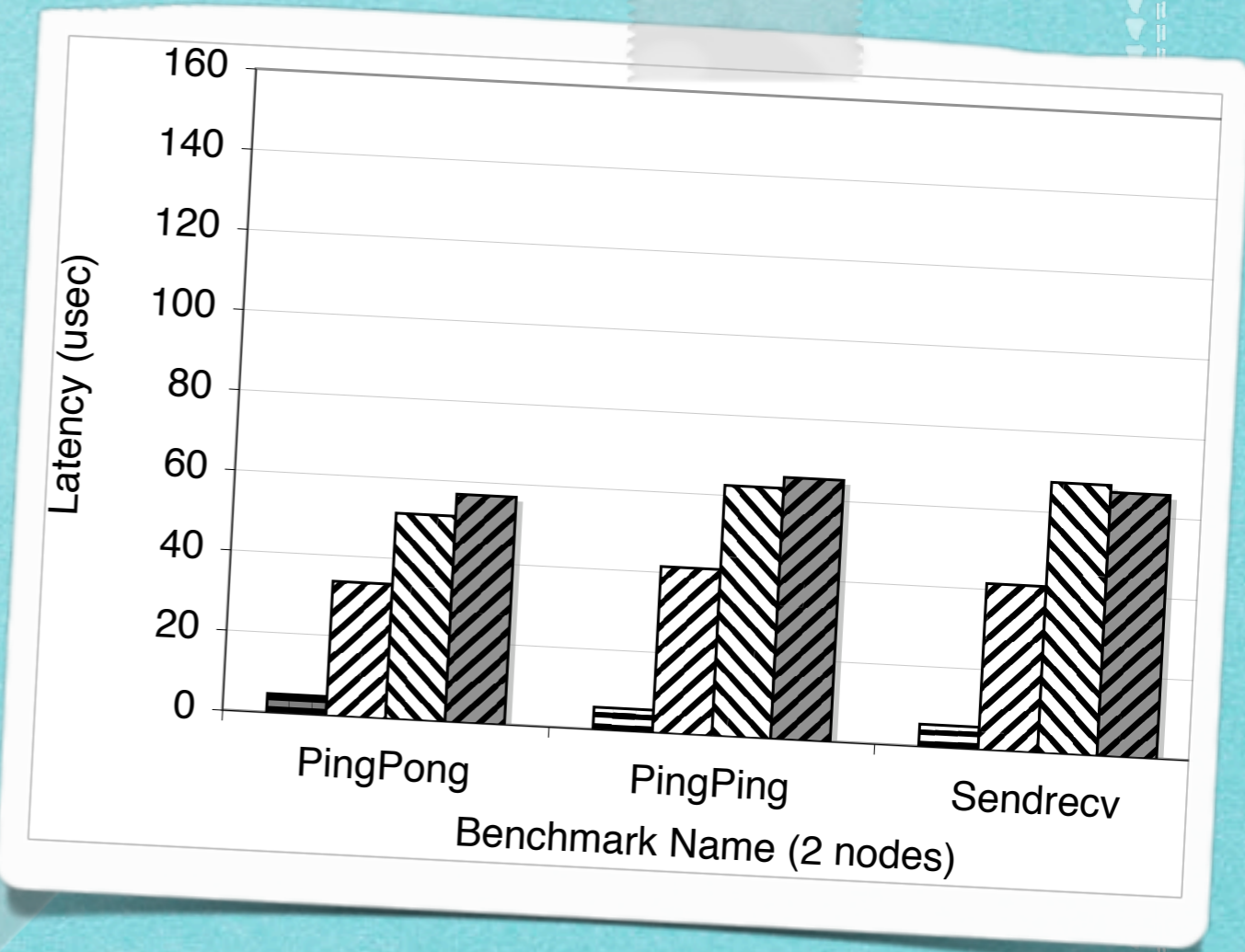
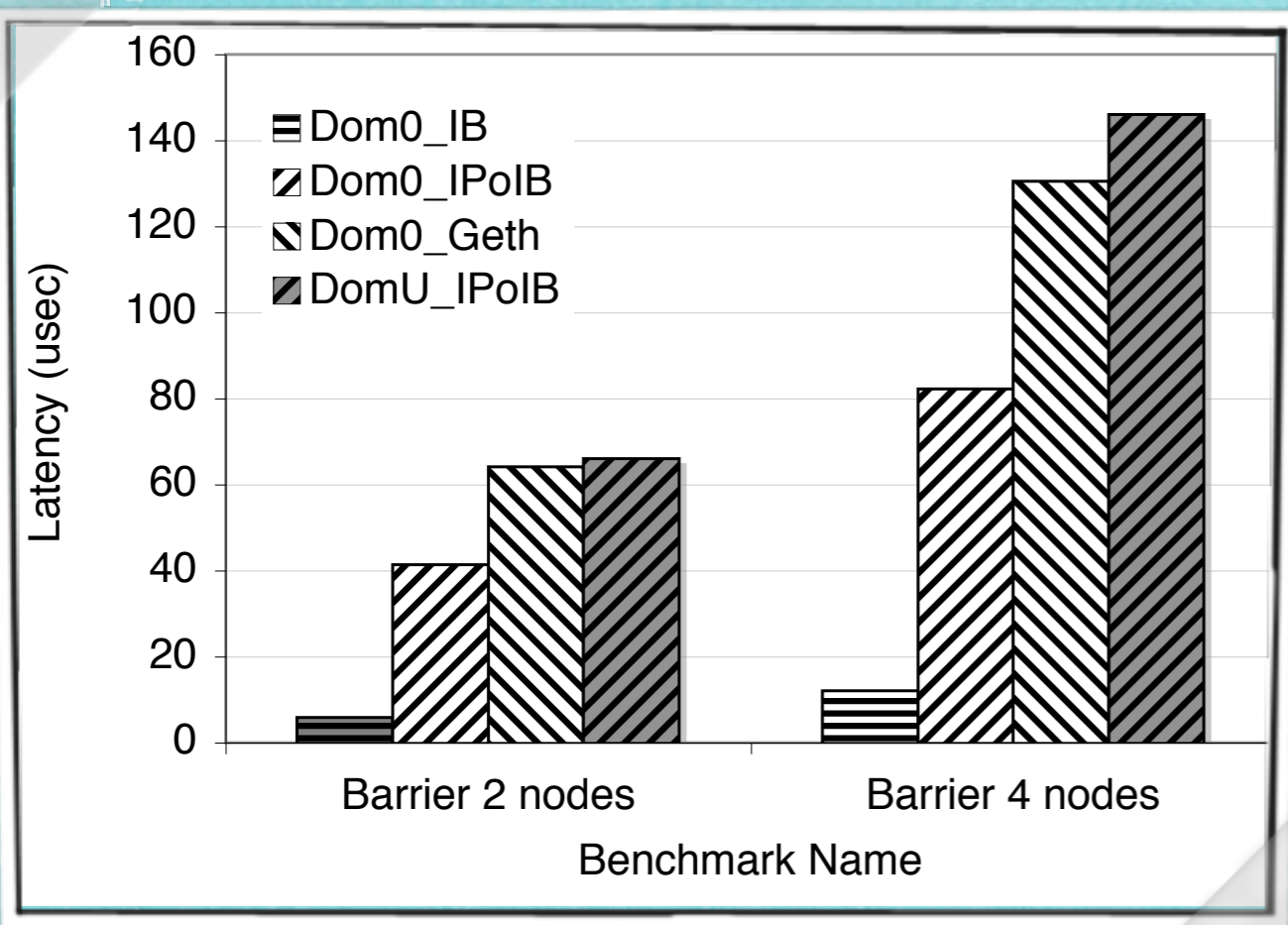




Esperimenti

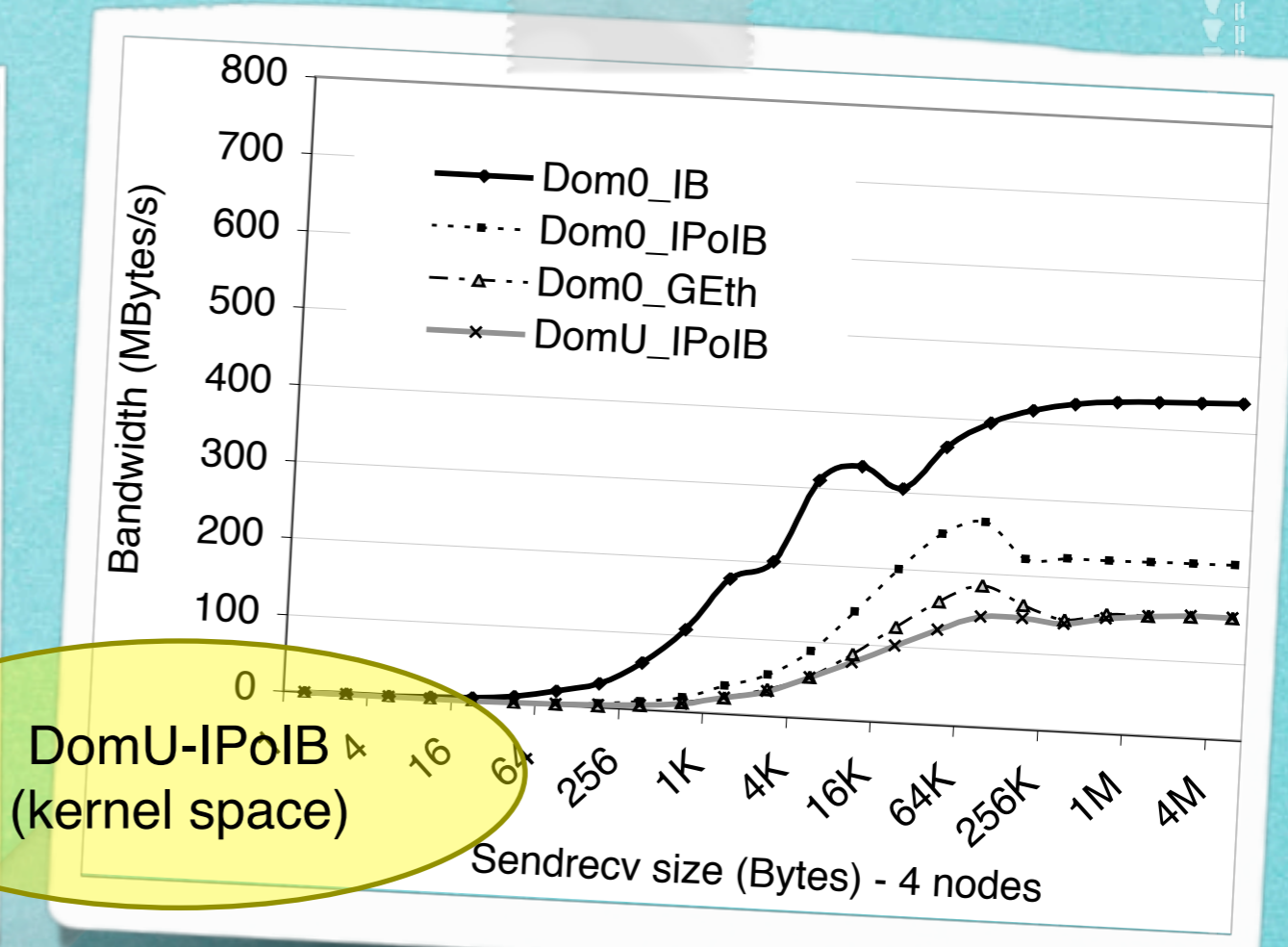
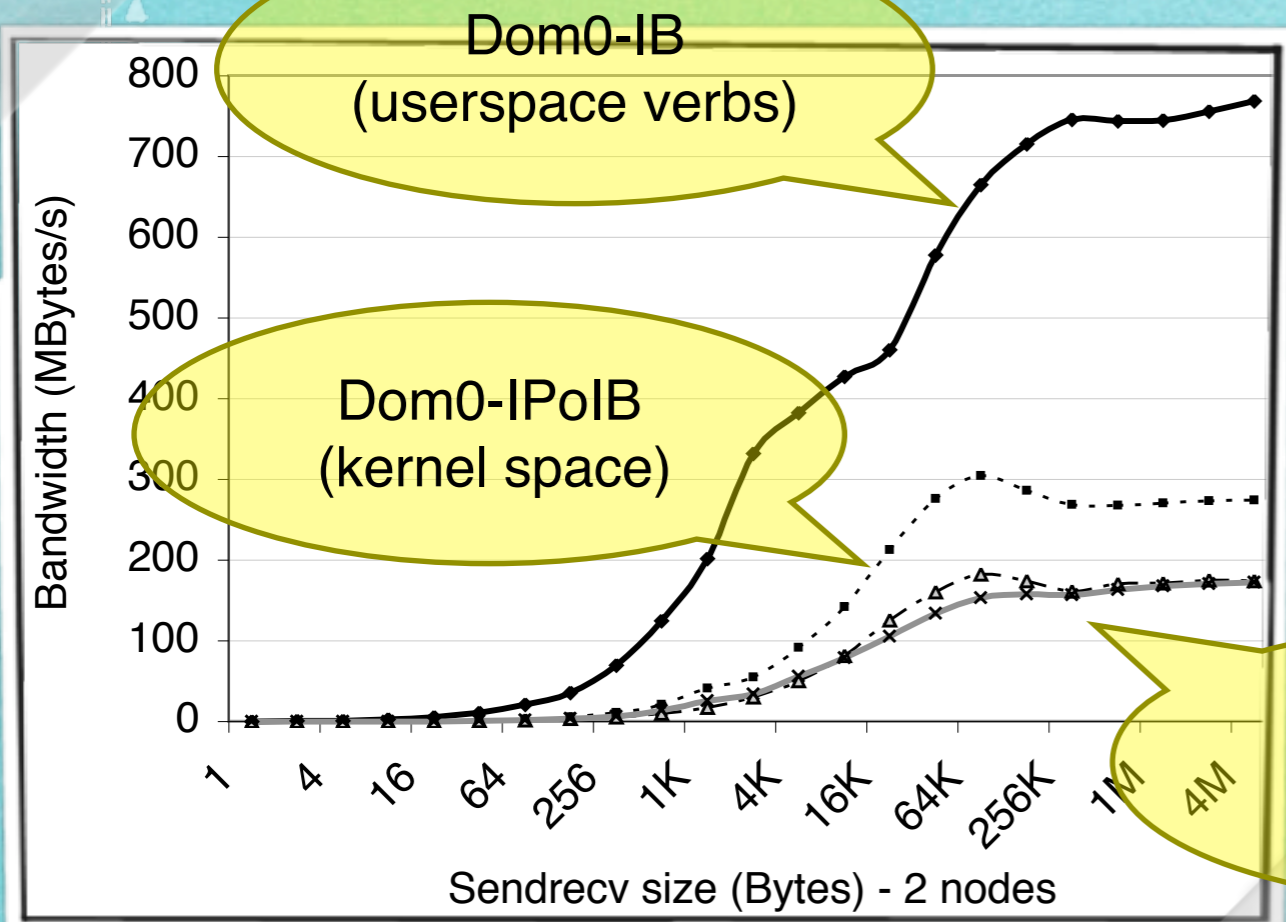
overhead dovuto alla virtualizzazione del processore, della rete e dello storage

Latenza di comunicazione



Dom0_IB	Ubuntu Dom0, Infiniband user-space verbs (MPI-gen2)
Dom0_IPoIB	Ubuntu Dom0, Infiniband IPoverIB (MPI-TCP)
Dom0_Geth	Ubuntu Dom0, Giga-Ethernet (MPI-TCP)
DomU_IPoIB	Ubuntu DomU, virtual net on top of Infiniband IPoverIB (MPI-TCP)

Banda di comunicazione



Dom0_IB	Ubuntu Dom0, Infiniband user-space verbs (MPI-gen2)
Dom0_IPoIB	Ubuntu Dom0, Infiniband IPoverIB (MPI-TCP)
Dom0_Geth	Ubuntu Dom0, Giga-Ethernet (MPI-TCP)
DomU_IPoIB	Ubuntu DomU, virtual net on top of Infiniband IPoverIB (MPI-TCP)

Performance (CPU & SO)

Micro-benchmark	Ub-Dom0 vs CentOS	Ub-DomU vs CentOS	Ub-DomU vs Ub-Dom0
Simple syscall	+667%	+726%	+7%
Simple open/close	+36%	+34%	-2%
Select on 500 tcp fd's	+51%	+51%	0%
Signal handler overhead	+112%	+127%	+7%
Protection fault	+246%	+293%	+13%
Pipe latency	+115%	+31%	-40%
Process fork+execve	+143%	+119%	-10%
float mul	~0%	~0%	~0%
float div	~0%	~0%	~0%
double mul	~0%	~0%	~0%
double div	~0%	~0%	~0%
RPC/udp latency localhost	+35%	-7%	-31%
RPC/tcp latency localhost	+35%	-5%	-30%
TCP/IP conn. to localhost	+32%	+3%	-22%
Pipe bandwidth	-38%	+51%	+144%

Performance dello storage virtualizzato

Additional layer on top of iSCSI	read	write	rewrite
none (reference raw iSCSI access)	60	88	30
EVMS standard volume	66	89	32
EVMS snap, fresh files	63	88	31
EVMS snap, files existing on original	63	7	31

Table 1. Performances (MBytes/s) of the proposed storage abstraction layer. Results are referred to Bonnie block read/write/rewrite benchmarks on a iSCSI-attached SAN.



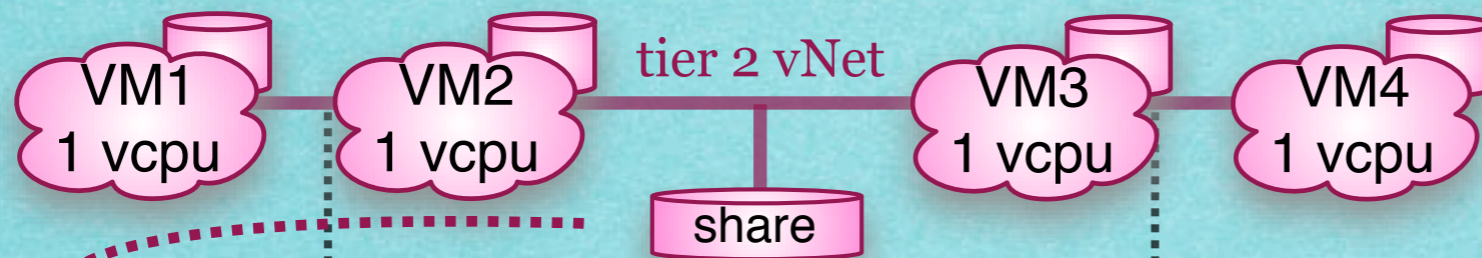
Demo-Linux

*VirtuaLinux versione 1.1 many-tier
(non ancora rilasciata, anteprima per NSS 2007)*

VirtualLinux 1.1 (many-tier)

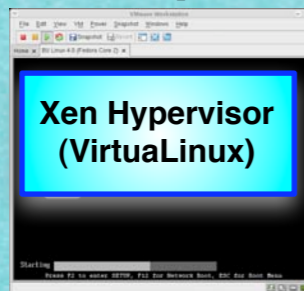
- ▶ Pre-release basata su Ubuntu, kernel 2.6.19-4 (gutsy)
- ▶ Pensato per facilitare lo sviluppo “home”
 - ▶ non tutti gli sviluppatori hanno a casa un cluster con almeno 3 nodi e una SAN iSCSI Infiniband
 - ▶ il cluster viene simulato mediante macchine virtuali (binary translator, e.g. VMware)
- ▶ tre livelli (due dei quali virtualizzati)
 - ▶ tier 0, standard linux - realizza la SAN (iscsi-target)
 - ▶ tier 1, macchine VMware - nodi del cluster “fisico”
 - ▶ tier 2, macchine Xen - nodi dei cluster virtuali iscsi-target
- ▶ Lento, ma permette lo sviluppo e l’esecuzione di un cluster con supporto per cluster virtuali con SAN su un solo laptop
- ▶ In realtà, è un sistema di demo per conferenze per app. //

Demo NSS07 (many-tier)



tier 2
VirtualLinux
con VC (Xen)

tier 1
vEth



tier 1 vNet



tier 1
cluster virtuale
di VM diskless
(VMware o QEMU)

iSCSI+EVMS
via tier 1 vNet



tier 0
laptop standard linux
+ iSCSI target

distance in which individuals are clustered

Dot sizes: ● = 1,000+ ● = 100 - 999 ● = 10 - 99 ◆ = 1 - 9 visits



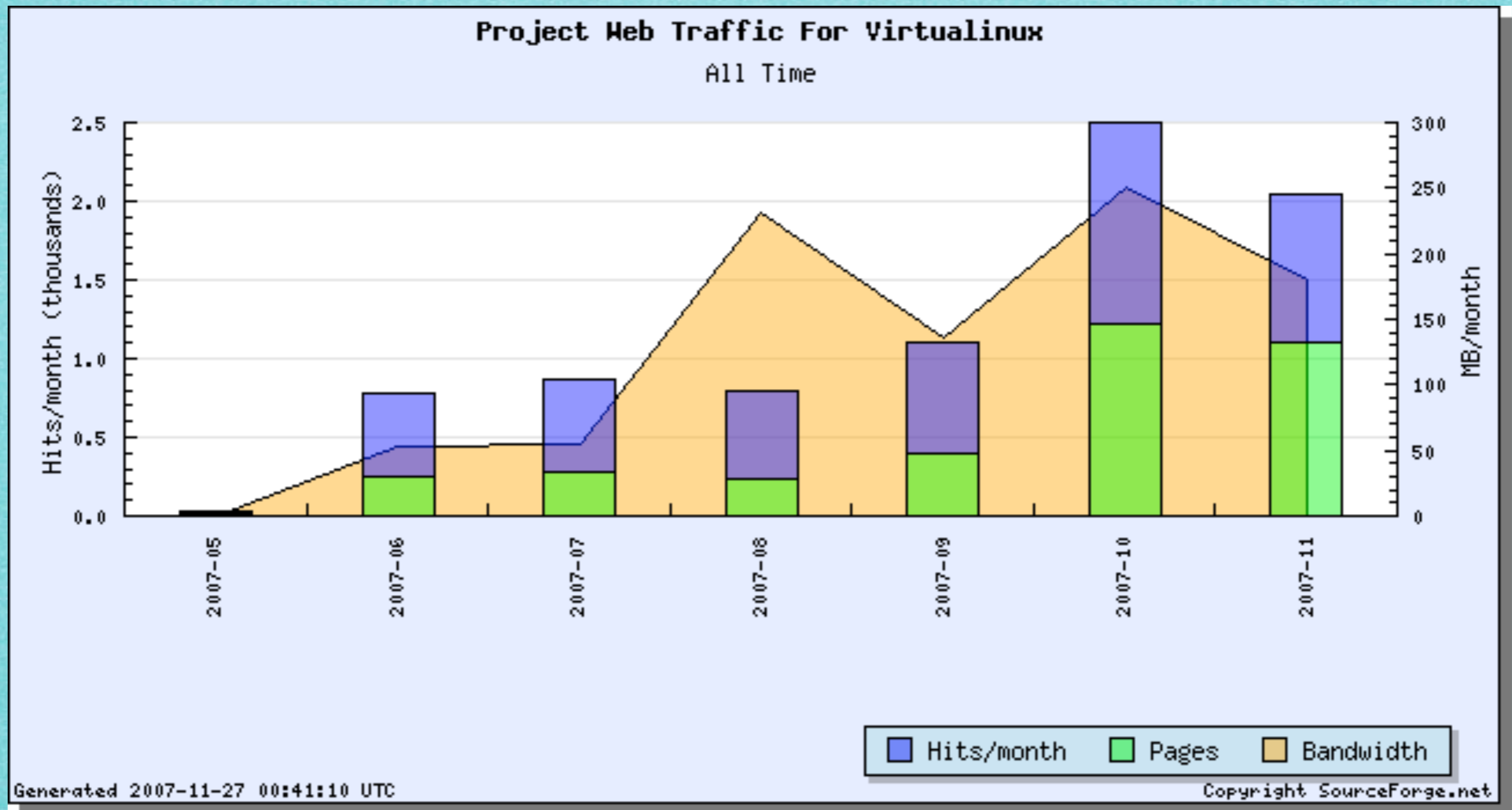
Riferimenti

mappa degli accessi (30 Lug - 10 Nov 2007)

Riferimenti

1. <http://sourceforge.net/projects/virtualinux>
2. M. Aldinucci, M. Torquati, M. Vanneschi, M. Cacitti, A. Gervaso, P. Zuccato.
VirtuaLinux design principles.
Technical Report TR-07-13, Computer Science Dept., University of Pisa, June 2007.
3. M. Aldinucci, P. Zuccato.
Virtual clusters with no single point of failure.
Intl. **Supercomputing** Conference (ISC2007), Poster session, Dresden, Germany, June 2007.
4. M. Aldinucci, M. Danelutto, M. Torquati, F. Polzella, G. Spinatelli, M. Vanneschi, A. Gervaso, M. Cacitti, and P. Zuccato.
VirtuaLinux: virtualized high-density clusters with no single point of failure.
In Proc. of the Intl. Conference ParCo 2007: **Parallel Computing**, Jülich, Germany, September 2007.
5. M. Aldinucci , M. Torquati , P. Zuccato, M. Vanneschi.
The VirtuaLinux Storage Abstraction Layer for Efficient Virtual Clustering.
Intl. Conference Euromicro PDP 2008, **Parallel Distributed and network-based Processing**, Toulouse, France, February 2008.
6. VirtuaLinux at **Linux Day** 2007 Meeting (invited talk, Trento, October 27th, 2007)

... on sourceforge



Conclusioni

- ▶ Affronta problemi clustering in ambito industriale
 - ▶ riduce costi di installazione e manutenzione
 - ▶ permette il consolidamento e lo sharing
 - ▶ progetta da interventi maldestri dopo il deployment
 - ▶ sottoinsiemi utilizzati come prodotti industriali da Eurotech SpA
- ▶ Risultati scientifici di interesse assoluto
 - ▶ virtualizzazione dello storage
 - ▶ unica soluzione simile è VMware Lab Manager (e non è open source)
 - ▶ **performance (talvolta) migliori della versione non virtualizzata**
 - ▶ Al momento testato con Ubuntu e CentOS

Open source (under GPL)

- ▶ Lo sviluppo continua
 - ▶ molto abbiamo fatto, molto rimane da fare
 - ▶ interfacce grafiche per l'installazione e management, ...
 - ▶ nuove distribuzioni, nuovi kernel, nuove VM (e.g. KVM), ...
- ▶ E' open source, cioè è vostro
 - ▶ morirà prima o poi come tutte le cose terrene ...
 - ▶ quando ... dipende da voi
 - ▶ si nutre di utilizzatori, beta-tester, sviluppatori, finanziatori, ... cioè una comunità di persone che lo sostengono
 - ▶ se volete giocarci, diventare sviluppatori, fare un tesi di laurea junior o senior ... scrivetemi ...
 - ▶ se poi pensate di avere buone idee , siamo pronti ad ascoltare ...
- ▶ **Marco Aldinucci - <http://www.di.unipi.it/~aldinuc>**

VirtuaLinux ringrazia ...

- ▶ Dipartimento di Informatica, Università di Pisa, Italy
- ▶ Eurotech SpA, Amaro (UD), Italy
 - ▶ finanziatore del progetto VirtuaLinux
 - ▶ incluso cluster con 16 nodi e reti infiniband per lo sviluppo
 - ▶ analisi dei problemi industriali, e delle possibili soluzioni
- ▶ Teknoservice, Pisa, Italy
 - ▶ supporto locale per hardware e strumenti di sviluppo
- ▶ Gli sviluppatori (9 attualmente ...)
 - ▶ aldinuc, califfo, gervystar, gobex, massimot, monica_d, patton73, pierfrancesco, spinatel

Xen architecture

