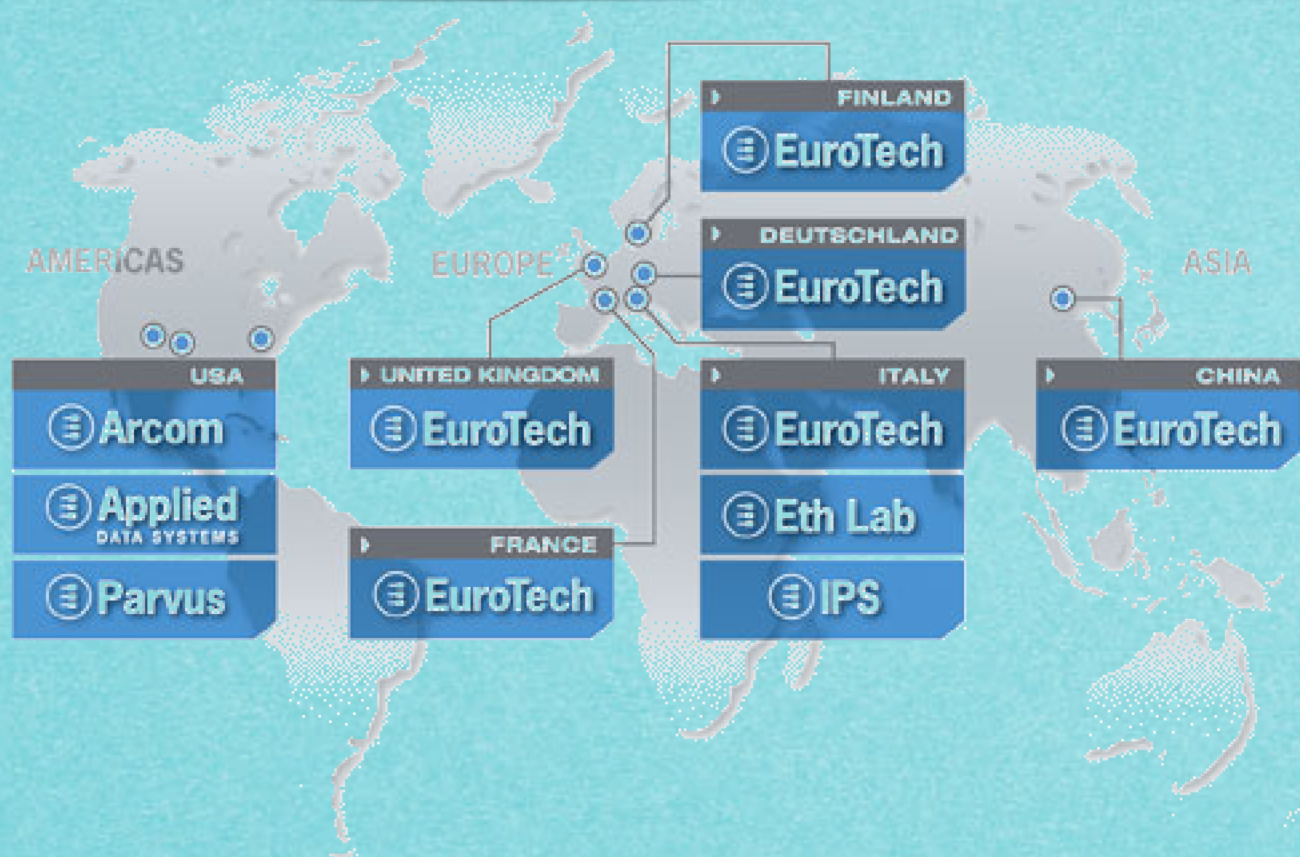
VirtuaLinux

An open source HPC clustering solution

http://sourceforge.net/projects/virtualinux/

M. Aldinucci, M. Torquati, et al. (Unipi)
P. Zuccato, A. Gervaso (Eurotech S.p.A.)

Eurotech HPC
http://www.eurotech.it



EUROTECH
GROUP

Pierfrancesco Zuccato
Responsabile di Eurotech HPC

Industry & research

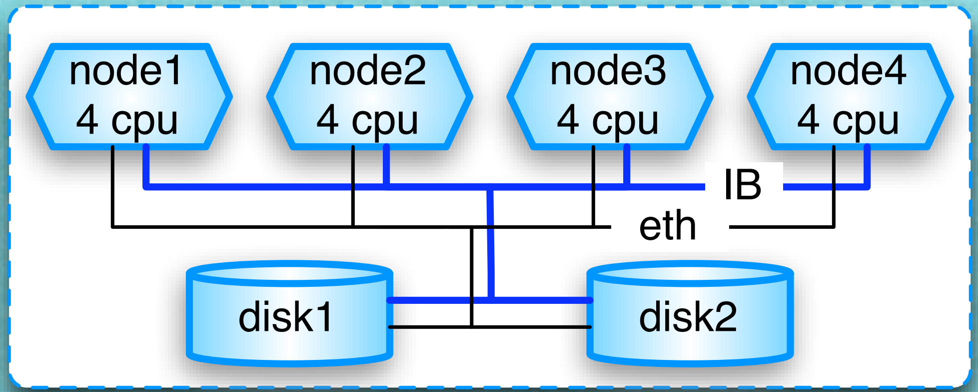
- The VirtuaLinux project
  - Entirely founded by Eurotech S.p.A.
  - Aims to solve industrial problems
    - They do need pure and applied research
  - Scientific goals published in A-class conferences
  - Developed software released as open source GPL

Problem statement

- Cluster
  - collection of (high-density) legacy independent machines connected by means of a LAN
  - are fragile
    - master node is a single point of failure
    - disks-on-blades are a common source of node failure
  - are hard to install and to maintain
    - installation requires days
    - skilled administrator are needed
      - root administrator problem
    - legacy OSes

Common industrial configuration

- High-density blades + external SAN (or NAS)
  - RAID SAN much more efficient & robust
    - Robustness and speed are separately addressed (often at HW level)
  - Sometimes legally required (e.g. USA Sarbenes-Oxley law)
- A number of linux distribution support this configuration
  - They require the customization of the standard distribution
    - typically path and configuration of services



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

Common hardware producer problem

- A single cluster configuration does not match all user expectation
  - I want CentOS, I need Ubuntu, I believe in Uindoz
    - the last one being much more a religion than an OS
- Cluster life-cycle in 5 easy steps
  1. the cluster is installed with factory distribution
  2. the cluster falls in the hands of site system administrators
  3. they mix-up user requirements
     - and they believe to be wizards, in reality they sometimes are sorcerer's apprentices
     - of course wizard exists, but they want to be paid, and this is forbidden by cluster owner religion
  4. after two days they destroy the cluster
  5. ask for the factory assistance, goto 1

Virtualization: a brand-new solution?

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."*

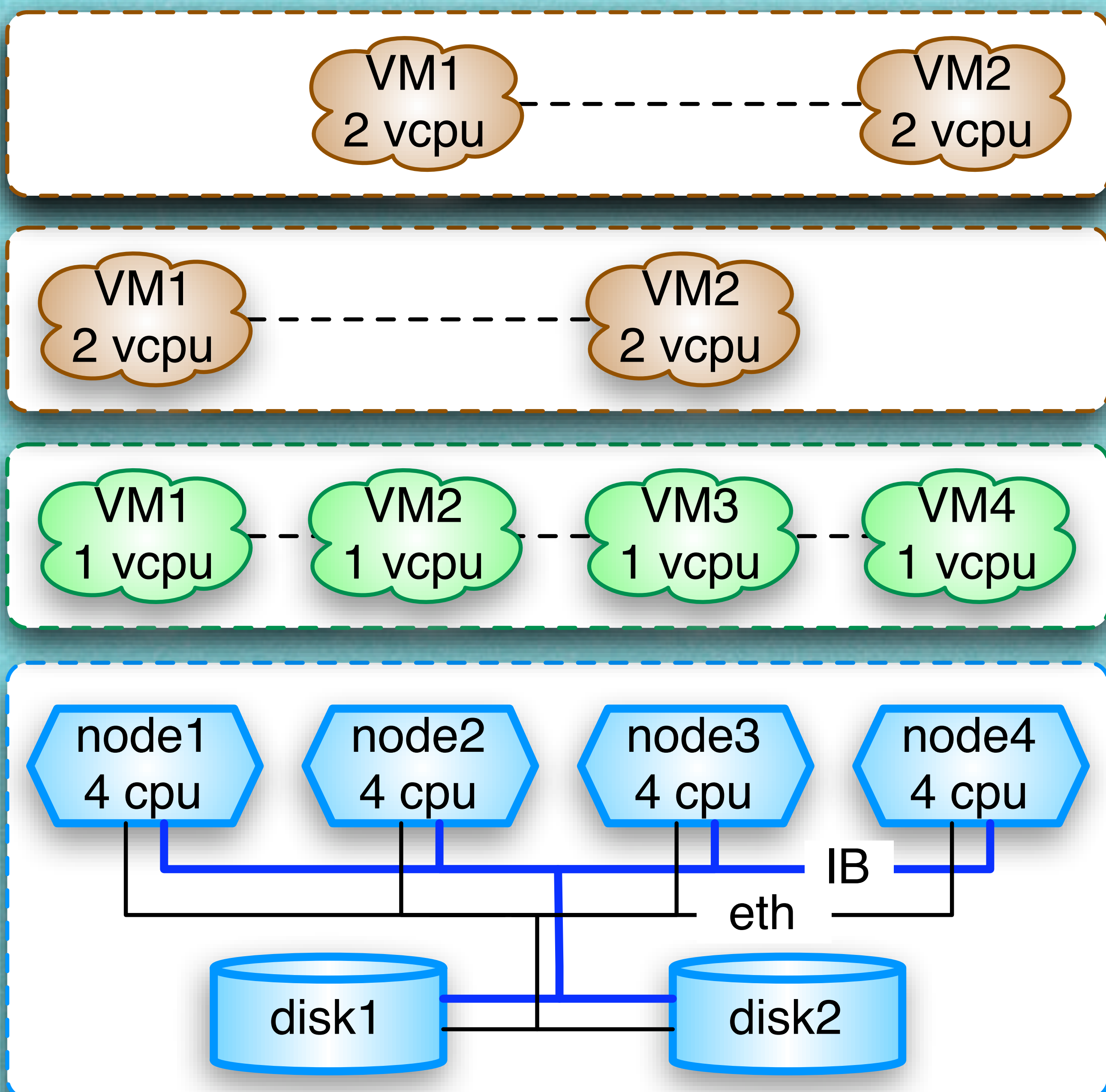
- Anyway, it works (it is abstraction, at the end)
  - High-level (e.g. JVM)
  - medium-level (e.g. FreeBDS jails)
  - low-level
    - Simulation (e.g. Cell), Binary translation (e.g. WMware, QEMU, ...), paravirtualization (XEN, KVM, ...)
- Make it possible to
  - consolidate several OSes
  - make it possible to share a physical resource
  - insulate user OSes from lower-level OS

VirtuaLinux approach

- A meta-distribution
  - get a Linux distribution and automatically configure it
- Master-less
  - Any master node; nodes are fully symmetrical
- Disk-less
  - Provide the cluster with fully independent virtual volumes starting from a network-attached SAN
  - No customization of the OS are required
- Transparently supporting Virtual Clusters
  - and the tools to manage them

Virtual clusters

Restart & Remap tan



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

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

Virtual Cluster "green"
4 VMs x 1 VCPUs
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

The Big Picture

Virtual Clusters (VC)

Each virtual node (VM) of a VC is a virtual machine that can be configured at creation time. It exploits a cluster-wide shared storage.

Each VC exploits a private network and can access the cluster external gateway.

VMs of a VC can be flexibly mapped onto the cluster nodes.

VCs can be dynamically created, destroyed, suspended on disk.

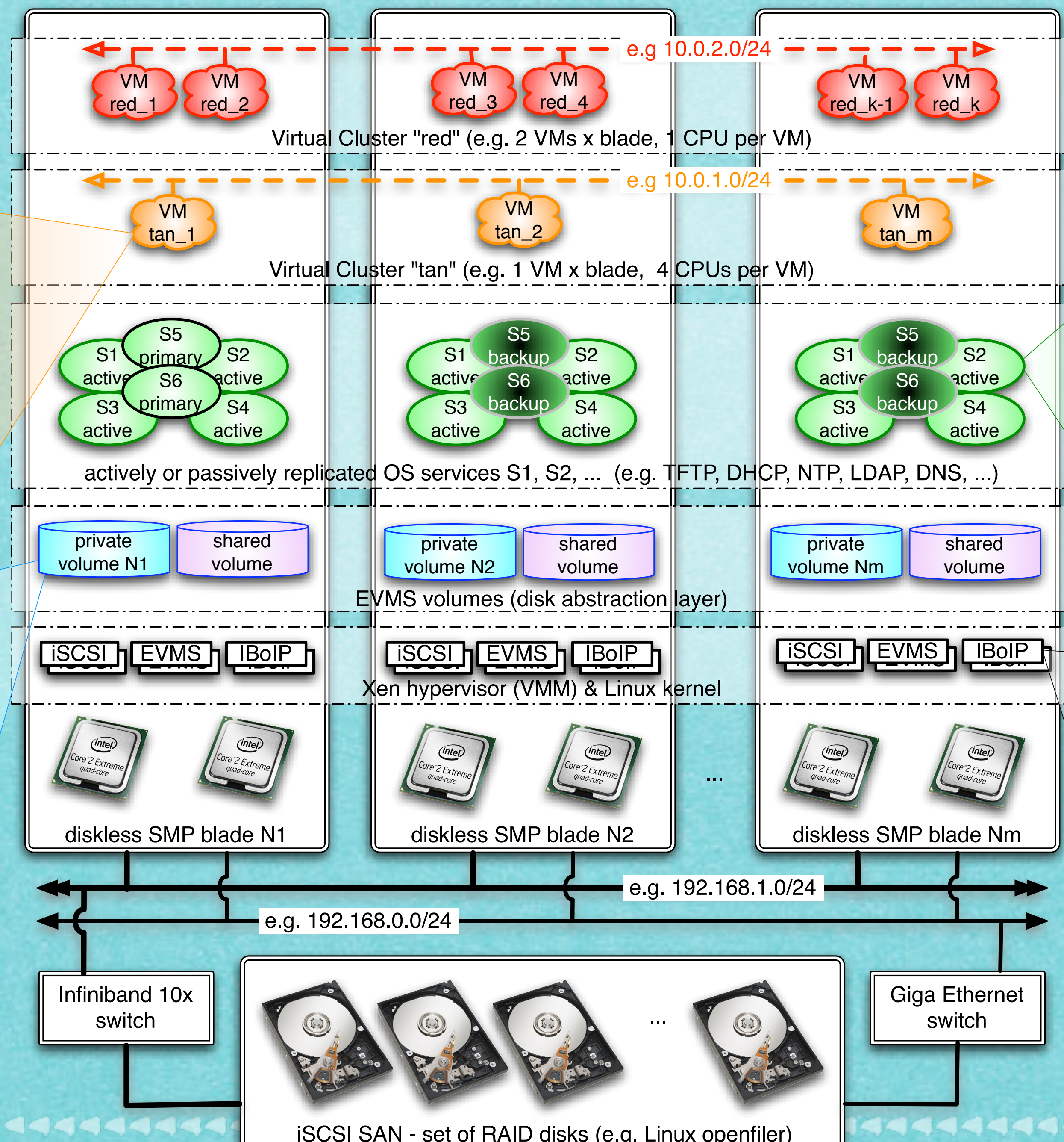
Disks Abstraction Layer

A set of private and shared EVMS volumes are mounted via iSCSI in each node of the cluster:

A private disk (/root) and a OCFS2/GFS cluster-wide shared SAN are mounted in each node.

EVMS snapshot technique is used for a time and space efficient creation of the private remote disk.

A novel plug-in of EVMS has been designed to implement this feature.



Linux OS services

All standard Linux services are made fault-tolerant via either active or passive replication:

Active: Services are started in all nodes; a suitable configuration enforces load balance on client requests. E.g. NTP, DNS, TFTP, DHCP.

Passive (primary-backup): Linux HA with heartbeat is used as fault detector. E.g. LDAP, IP gateway.

Kernel Basic Features

All standard Linux modules.

Xen hypervisor, supporting Linux paravirtualization, and Microsoft Windows via QEMU binary translation (experimental).

Network connectivity, including Infiniband userspace verbs and IP over Infiniband.

iSCSI remote storage access.

OCFS2 and GFS shared file systems

High Availability

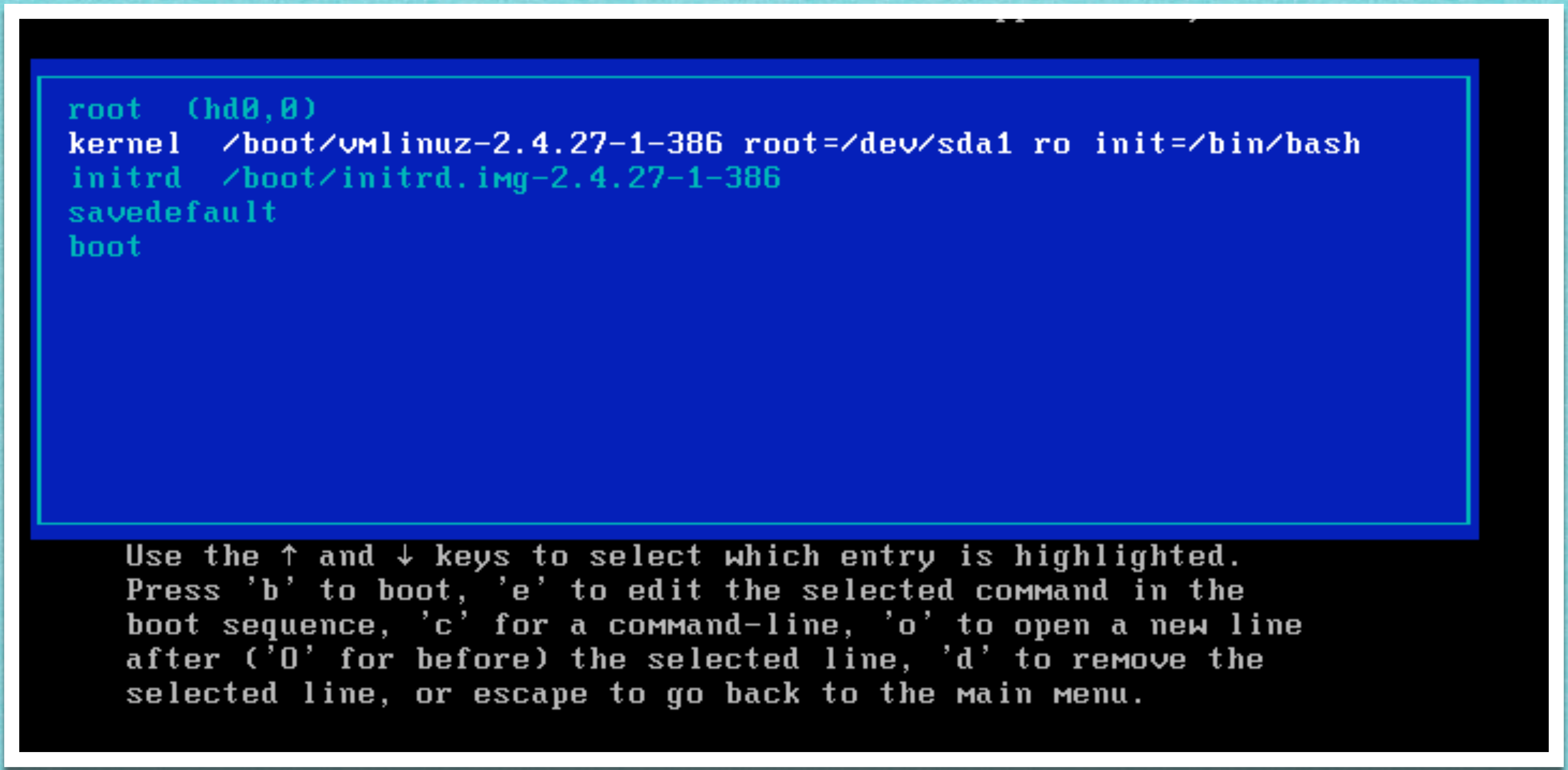
by way of the replication of services

High availability

- 24/7 cluster availability
- Redundant hardware
  - 5 Power supplies, 4 independent network switches, …
  - iSCSI-over-Infiniband or Fibre channel attached RAID
- Redundant services
  - All nodes are identical, and there is no master
  - All OS services are replicated on all nodes
    - Any blade can be detached, or crash at any point in time with no impact on system availability

Provide services without a (single) server

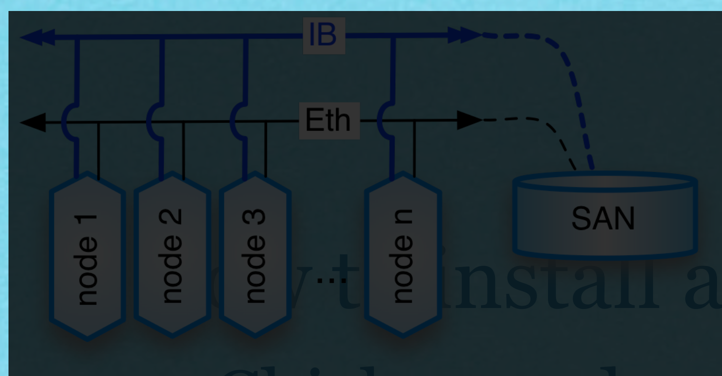
Service	FT model	Notes
DHCP	active	Pre-defined map between IP and MAC
TFTP	active	All copies provide the same image
NTP	active	Pre-defined external NTPD fallback via GW
IB manager	active	Stateless service
DNS	active	Cache-only
LDAP	service-specific	Service-specific master redundancy
IP GW	passive	Heartbeat 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

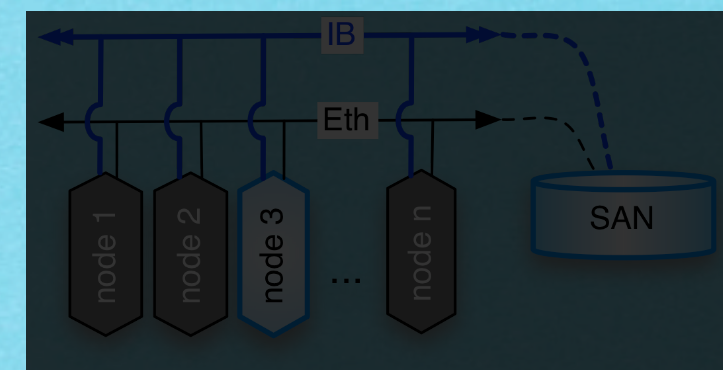


Novel boot sequence

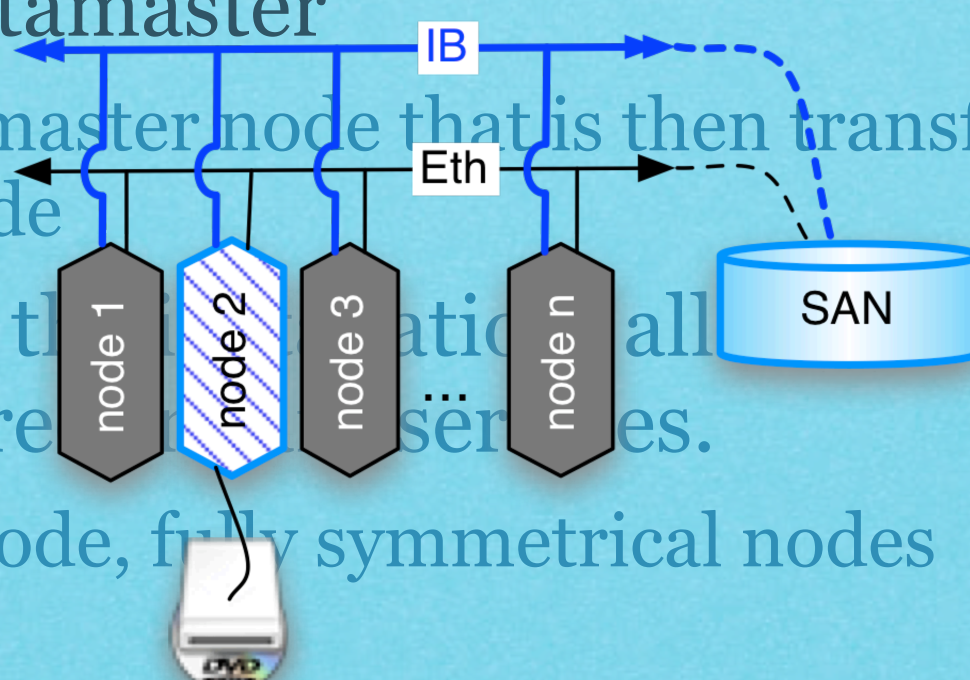
supporting the boot of master-less systems

Install without a master





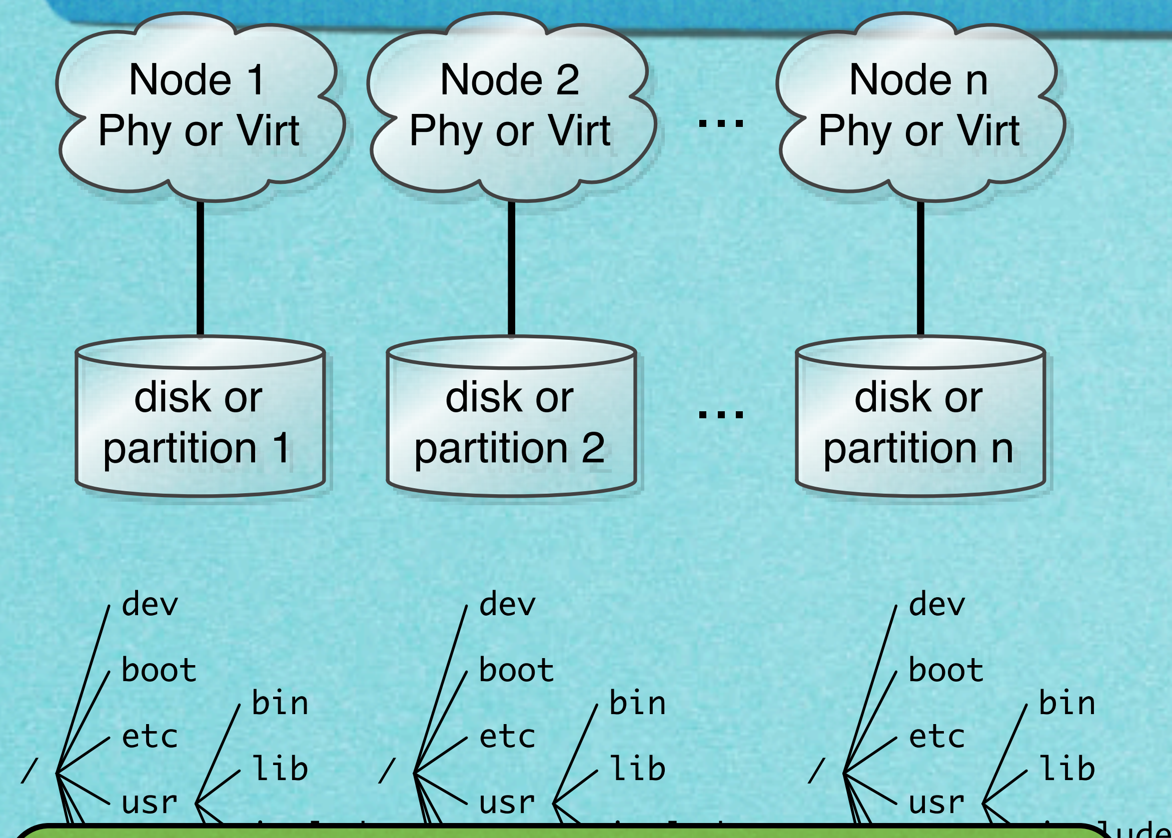
- … install a masterless cluster?
  - Chicken and egg problem!
- Solution: Metamaster
  - A transient master node that is then transformed in a standard node
- At the end of th… all … are identical and provide re… …es.
  - No master node, fully symmetrical nodes



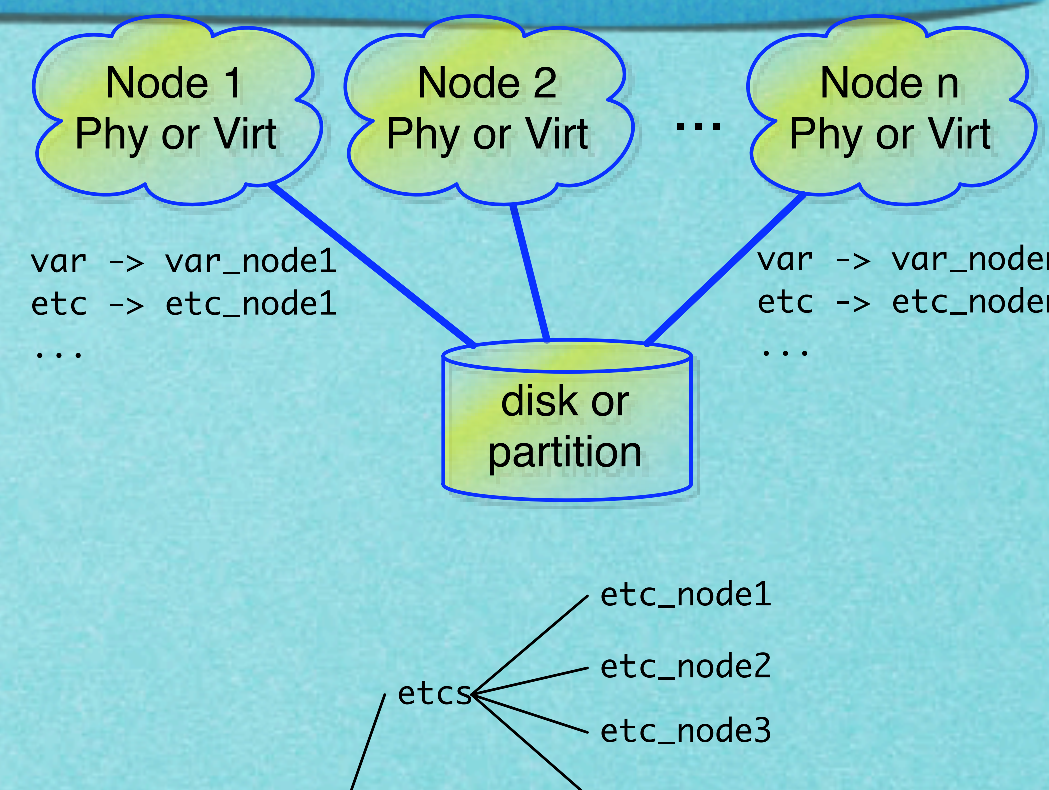
Storage Virtualization

a transparent constant time-space solution

Full vs no replication



- Each node (physical or virtual) has its own copy of the whole disk
- Transparent, easy to build and update
- OS does not need customisation
- Inefficient in time and space - O (n*size). Identical OS files are replicated



- Each node (physical or virtual) share a disk (a file system, actually)
- Not transparent, complex to build and update
- OS does need customisation
- Efficent in time and space - O(size). OS files are not replicated

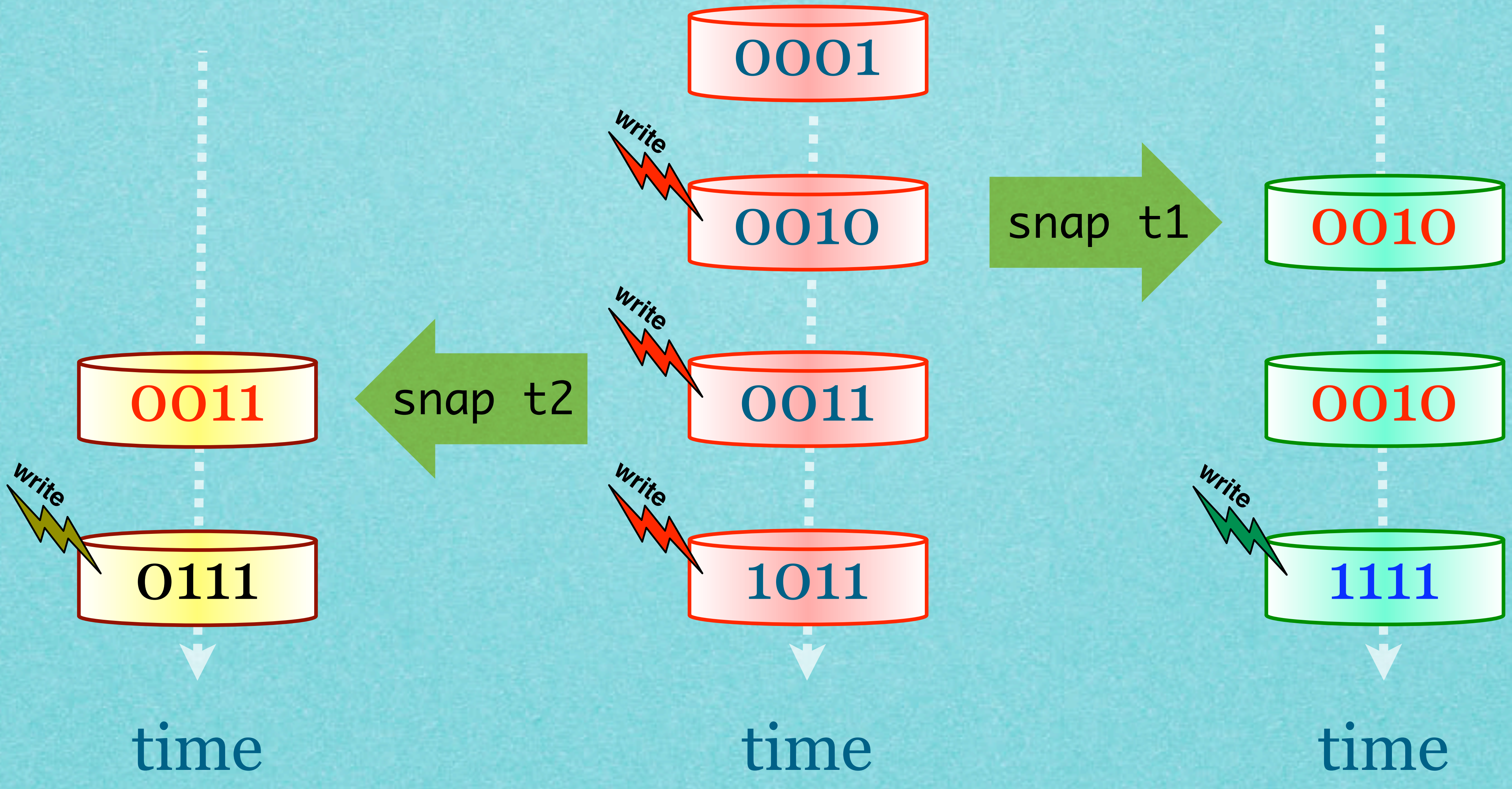
VC requirements

- Needs both transparency and time-space efficiency
  - Should be be independent from the OS
  - "Frequent" & very time consuming operation
    - e.g. 50 nodes x 10 GB x 100MB/s = ~ 2 hour
      - very optimistic forecast
      - just to start/suspend the VC (and the system become irresponsive)
  - Consumes lot of disk space
    - e.g. 50 nodes x 5 GB = 100 GB
      - for each VC and for the OS only - no user data here

VC storage properties

- Nodes of a VC are homogenous (same OS)
  - 99% of OS-related files are identical in all VMs
  - No reason to have heterogeneous VCs, just start more VCs of different kind
  - It is low-level virtualization
    - the virtualized is similar to the real
- Just keep them in a single copy
  - but don’t tell to the virtual nodes, they believe to be fully independent one each other
  - A novel usage of the snaphotting technique
    - copy-on-write

Snapshot tecnique



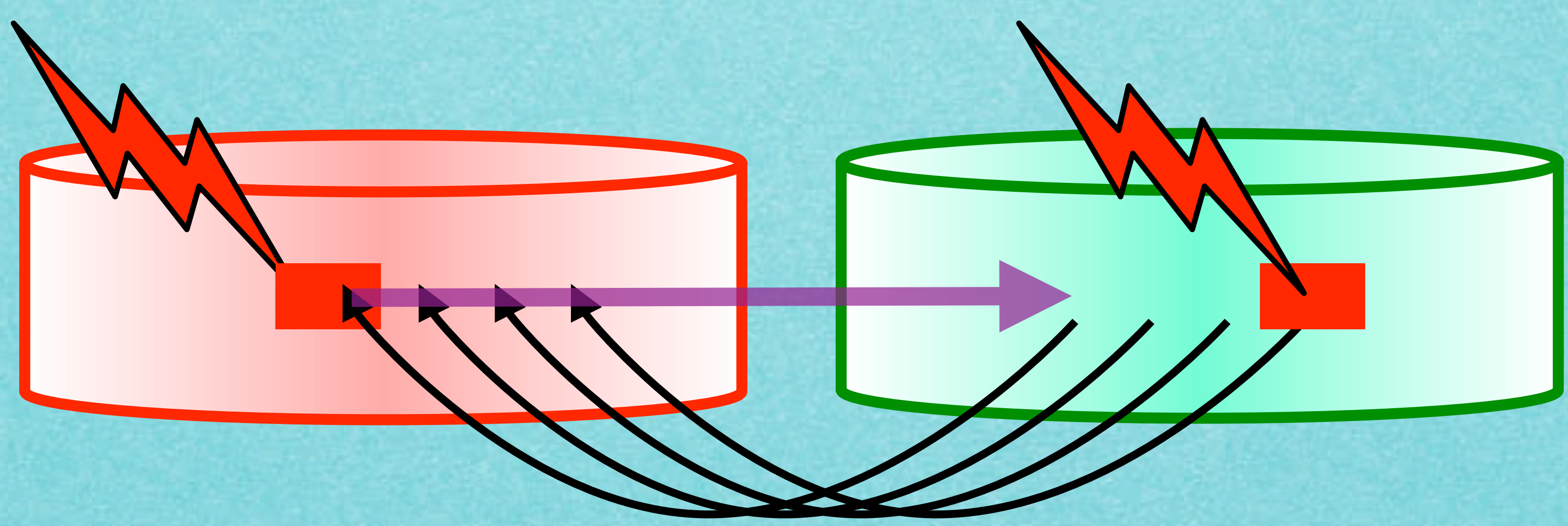
Snapshots

- Used to build online backups
  - Both original and snapshots can be written
- File system independent, transparent
- Supported by several tools, e.g. EVMS, LVM2
  - Implemented by linux standard kernel
  - dm_snapshot module (device mapper)
- Can be implemented in several ways
  - copy-on-write, redirect-on-write, split-mirror, ...

Copy-on-write

Writing on original

Writing on snapshot



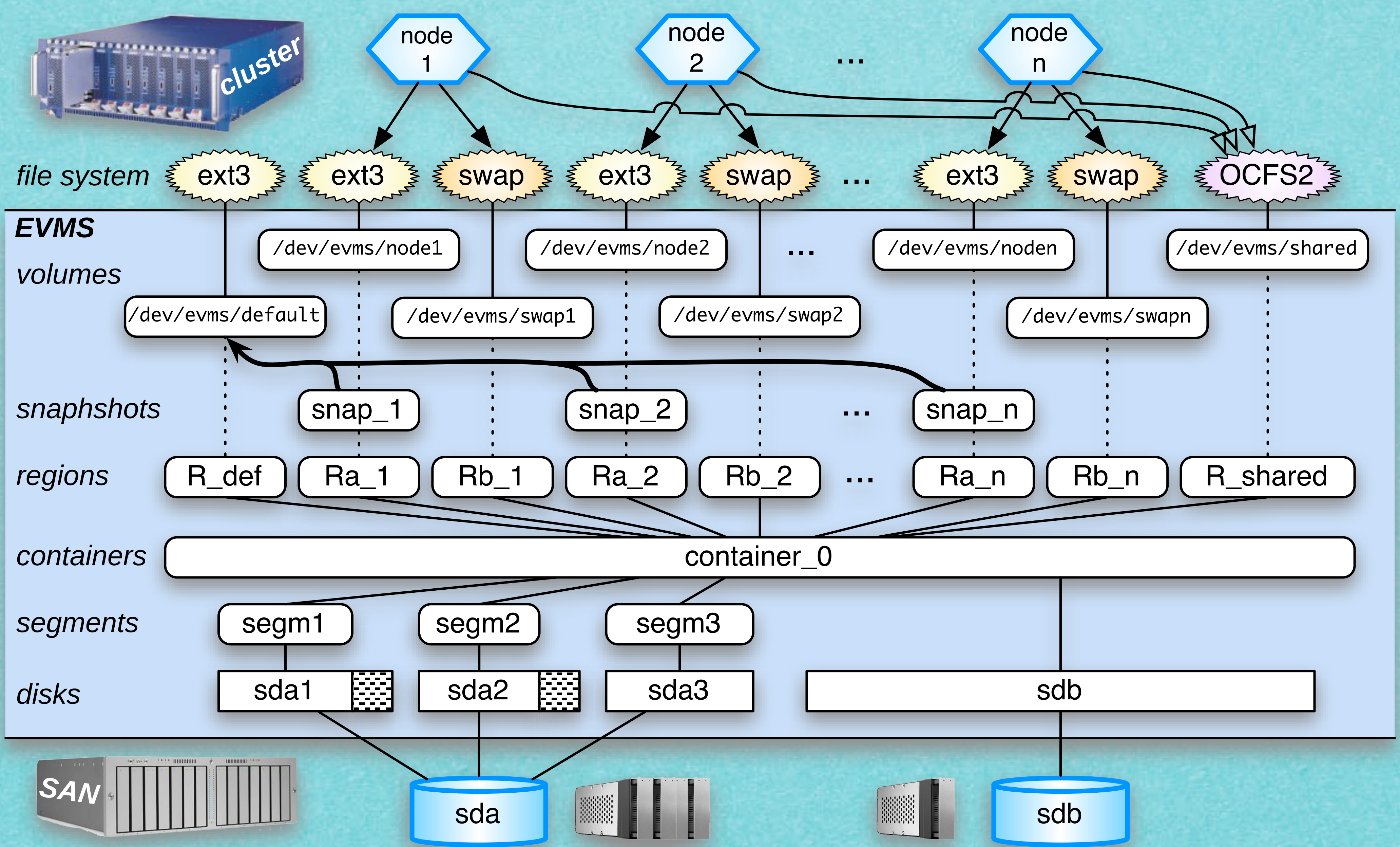
Originale

Snapshot

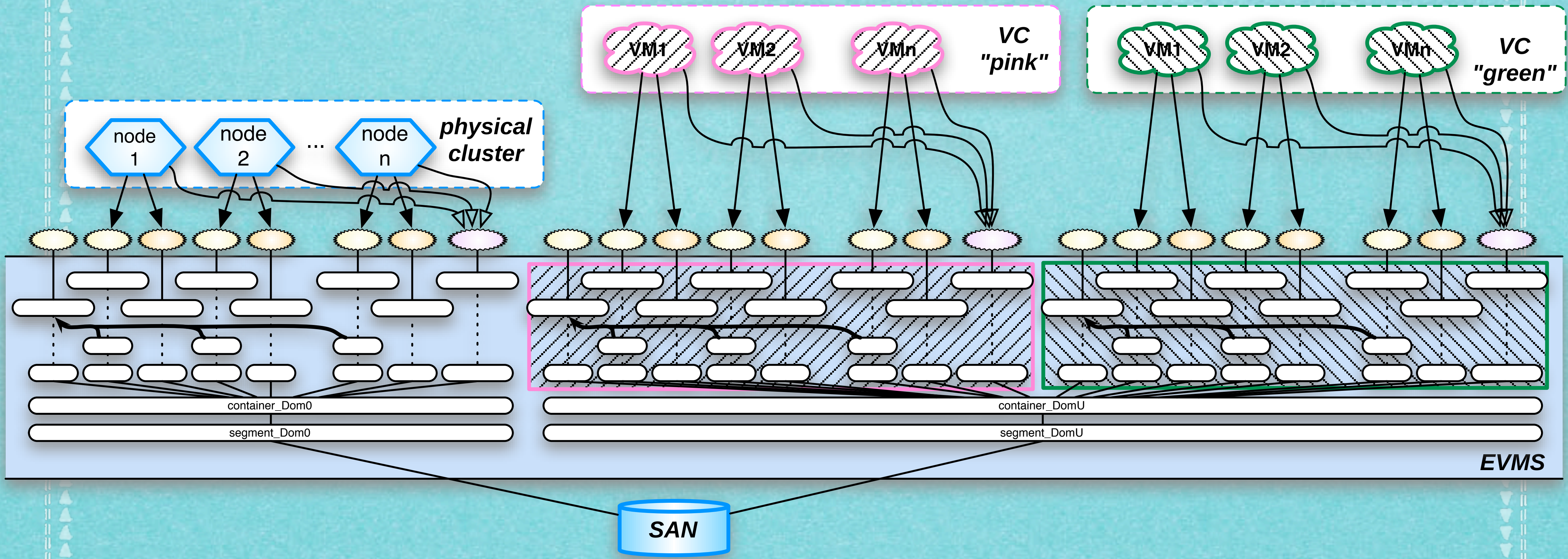
Concurrent snapshots

- Snapshots not designed for concurrency
  - All snapshot should be kept active in all nodes
  - Linux cannot keep active more than 8 snapshots per node
- Novel semantics for snapshots
  - Relax snapshots semantics while maintaining correctness
  - “Mark” as read-only parts on the blocks

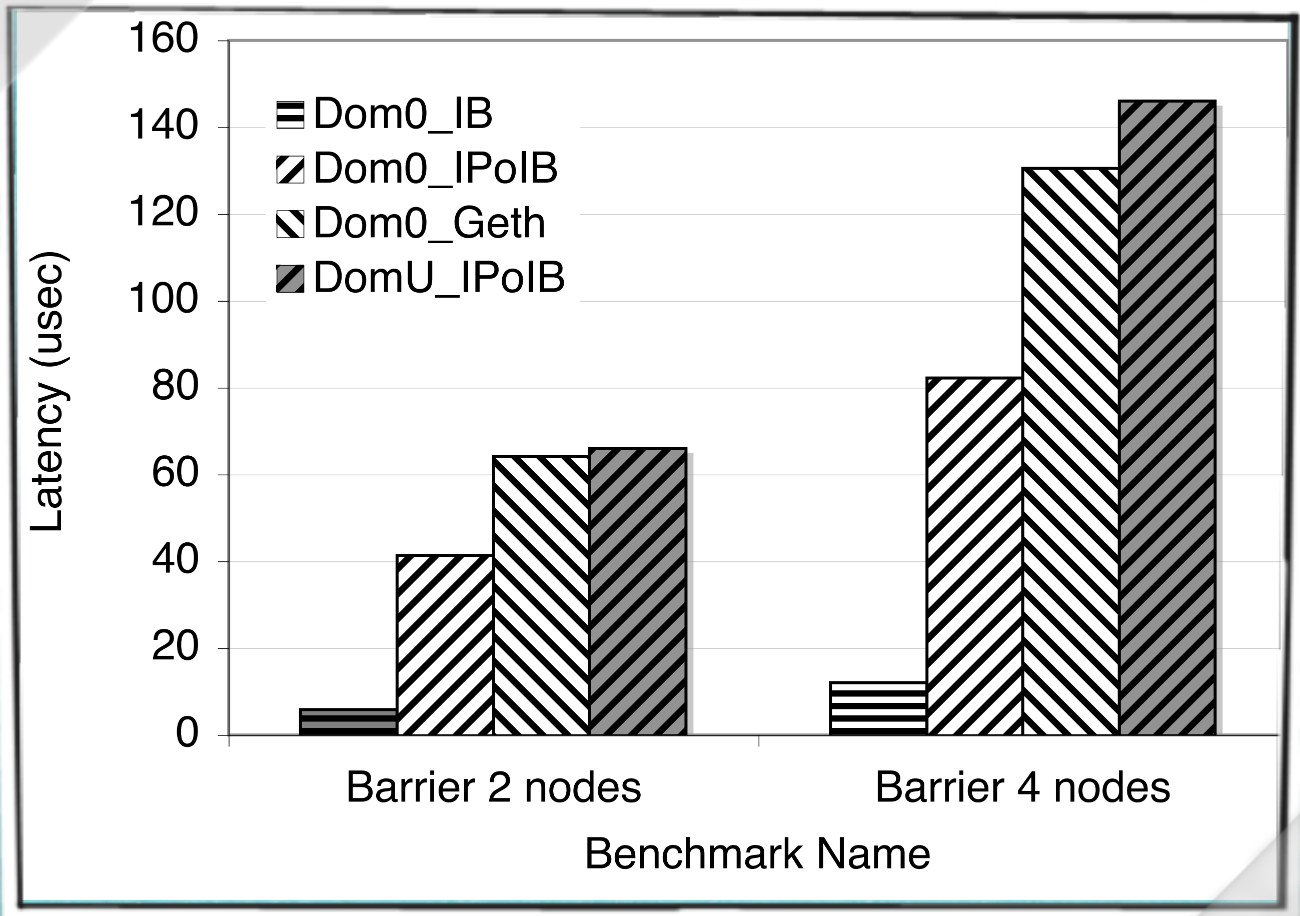
Storage virtualization

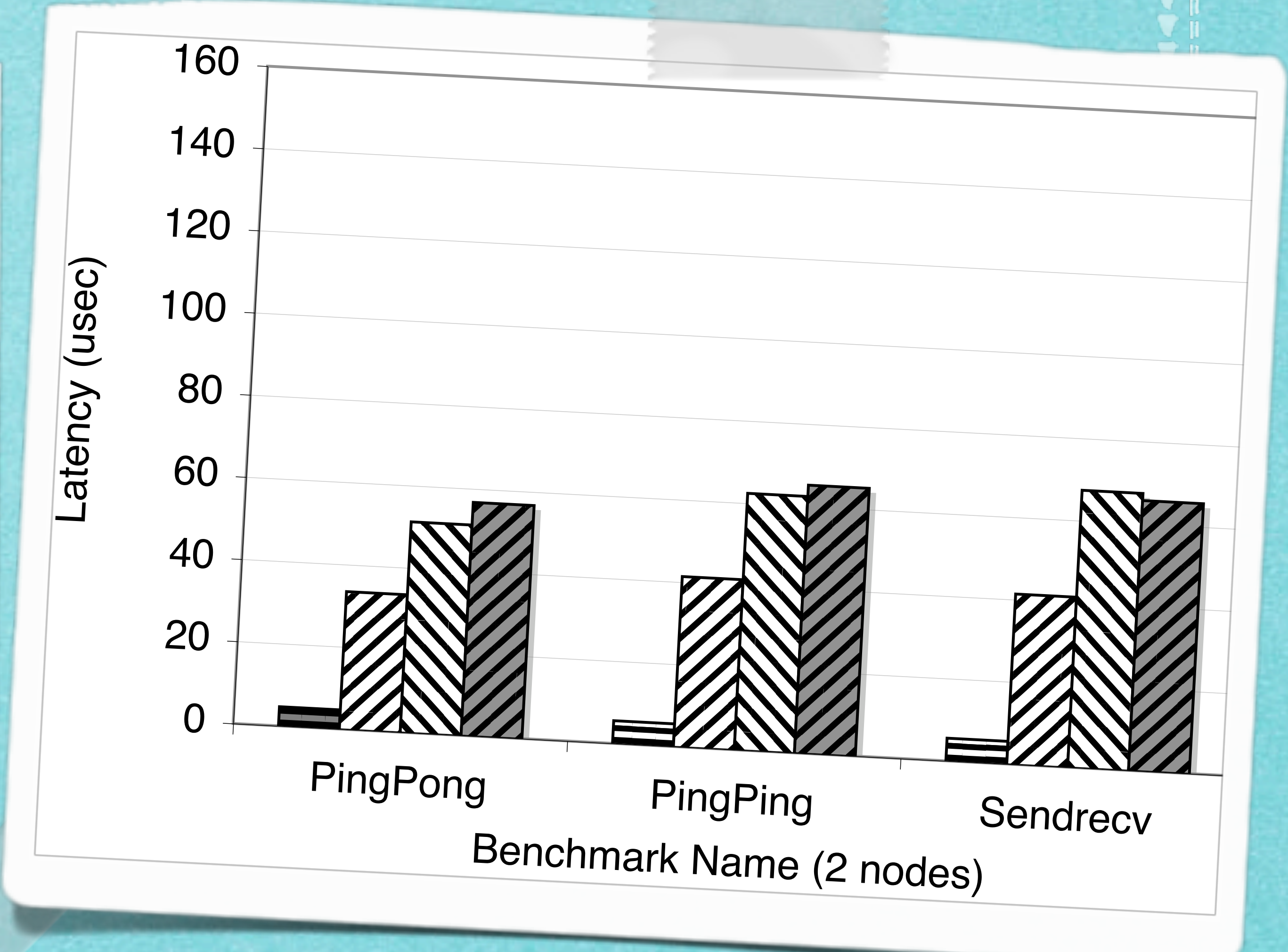


Storage virtualization (VCs)



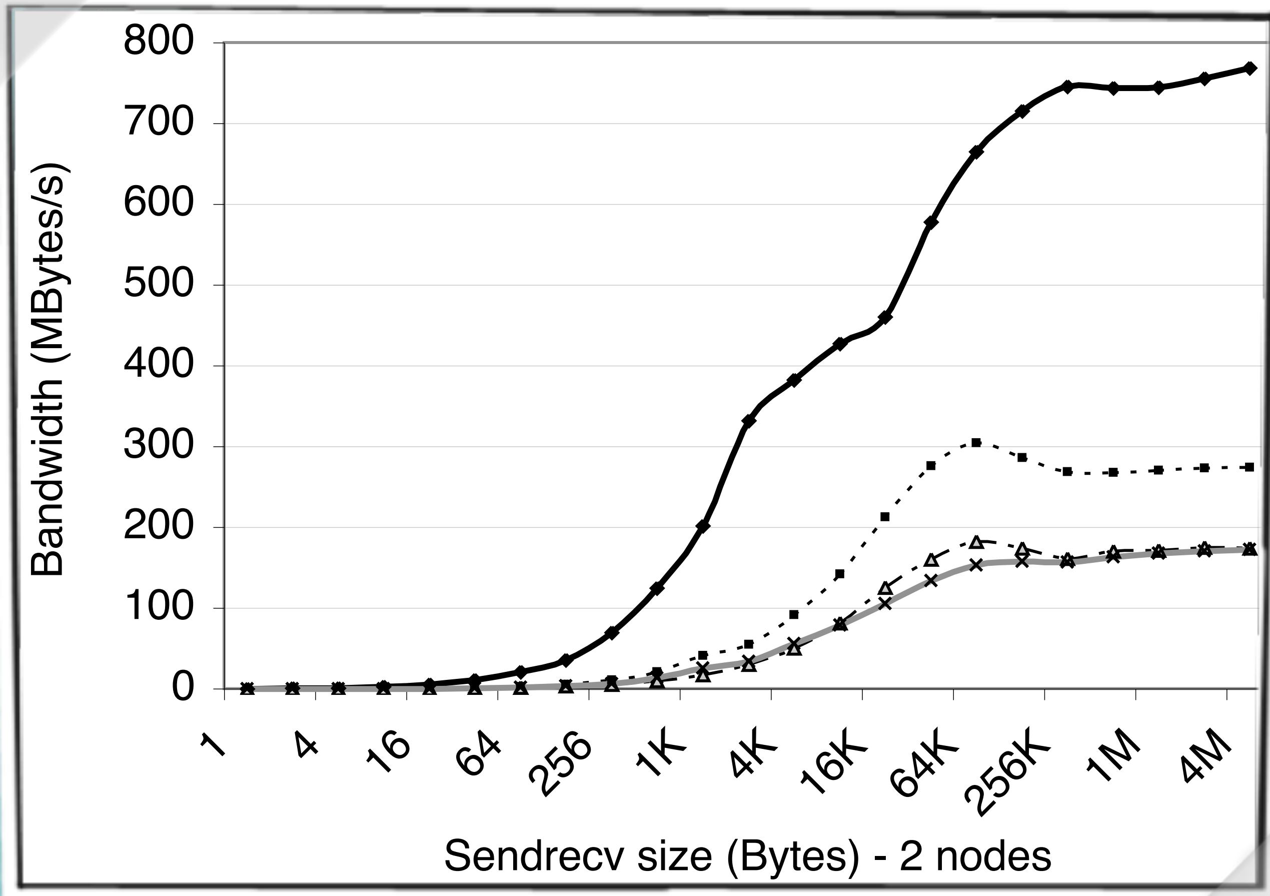
Latency

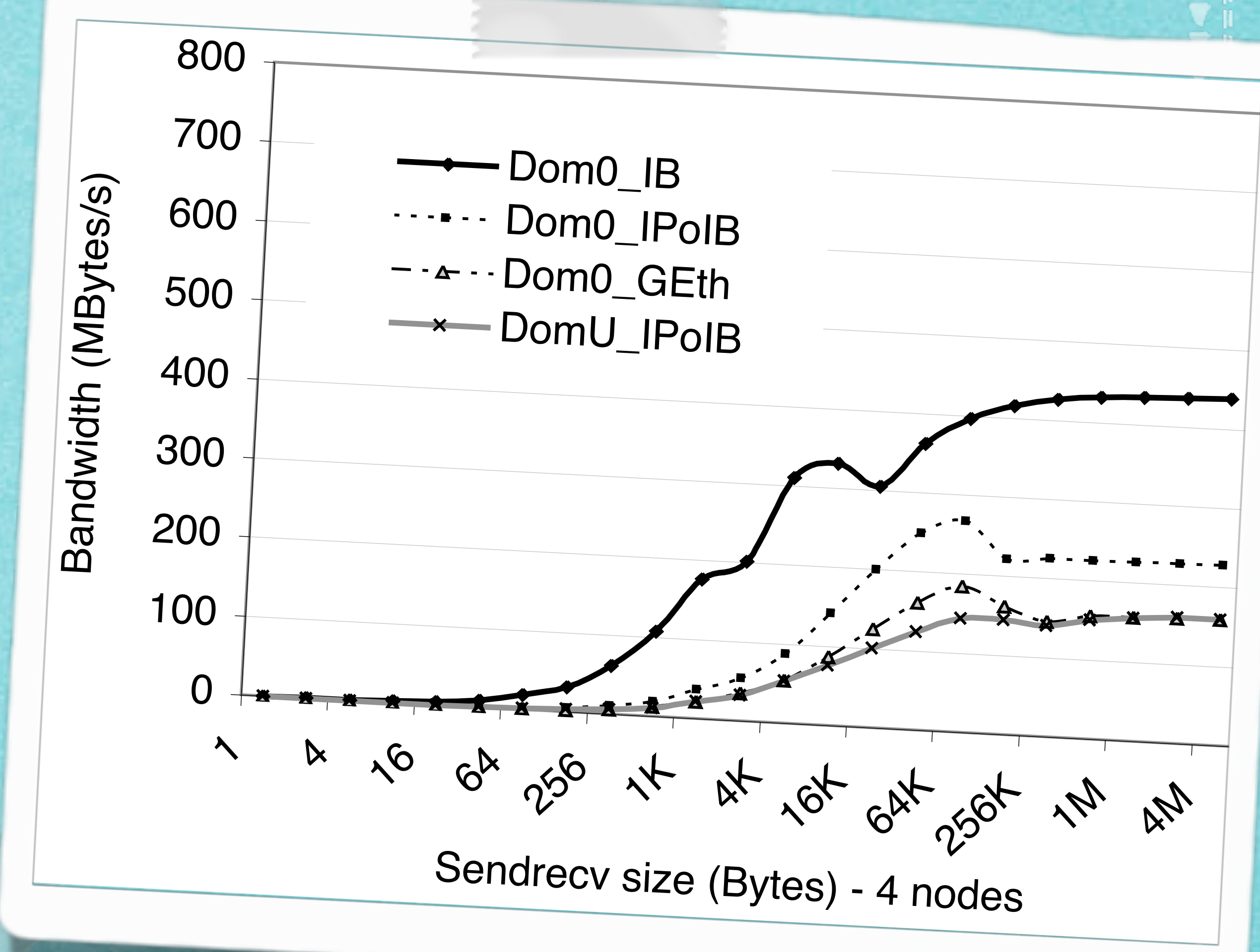




Dom0_IB	Ubuntu Dom0, Infiniband user-space verbs (MPI-gen2)
Dom0_IPoIB	Ubuntu Dom0, Infiniband IPoverIB (MPI-TCP)
Dom0_Geth	Ubuntu Dom0, Giga-Ethernet (MPI-TCP)
DomU_IPoIB	Ubuntu DomU, virtual net on top of Infiniband IPoverIB (MPI-TCP)

Bandwidth





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)

CPU & OS Performances

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%

Storage Virtualization Performances

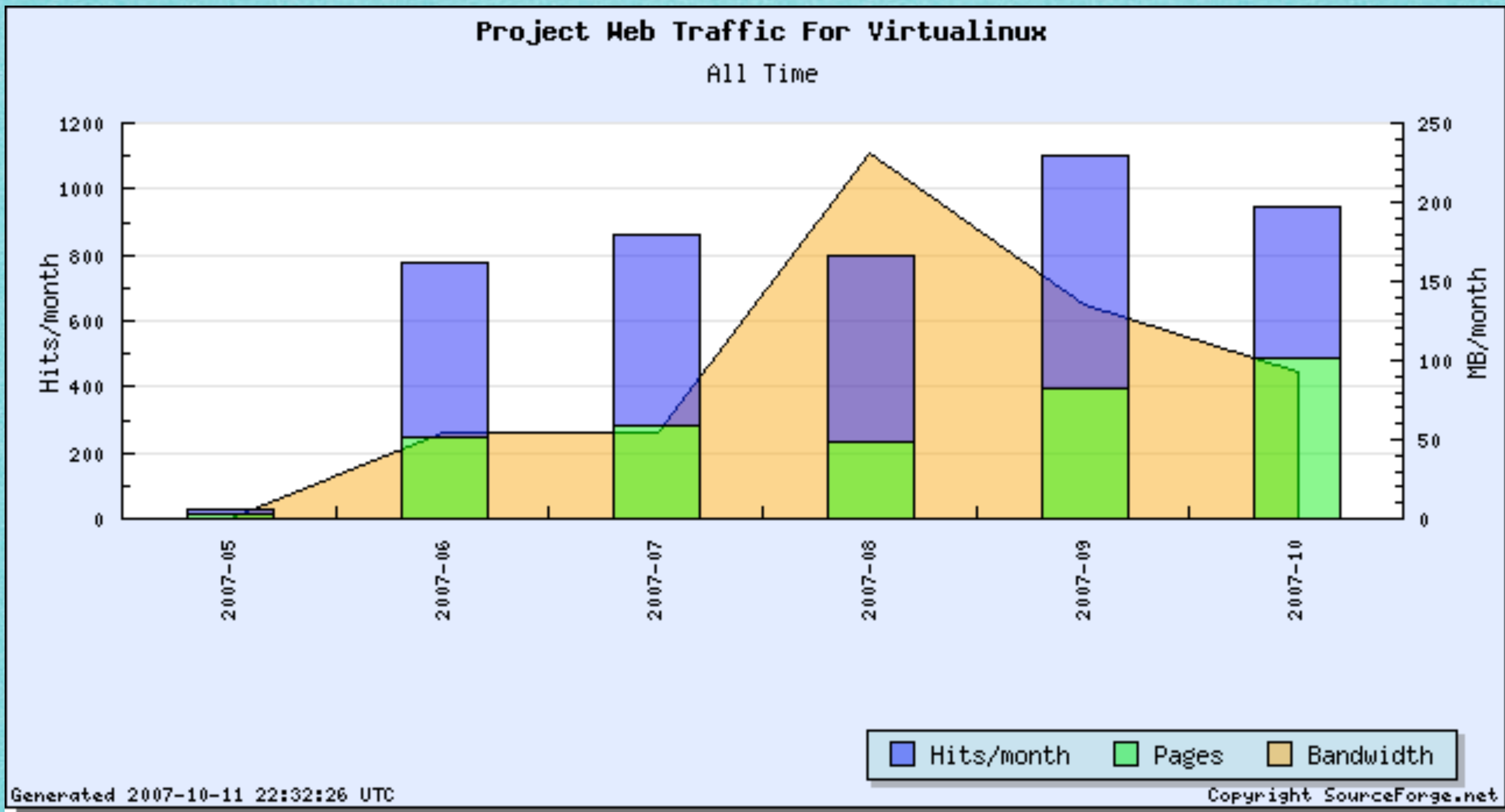
Additional layer on top of iSCSI	read	write	rewrite
none (reference raw iSCSI access)	60	88	30
EVMS standard volume	66	89	32
EVMS snap, fresh files	63	88	31
EVMS snap, files existing on original	**63**	**7**	**31**

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.

Scientific results

1. 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.
2. M. Aldinucci, P. Zuccato.
   Virtual clusters with no single point of failure.
   Intl. Supercomputing Conference (ISC2007), Poster session, Dresden, Germany, June 2007.
3. 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.
4. 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
   Accepted, to appear in February 2008.
5. VirtuaLinux at Linux Day 2007 Meeting (invited talk, October 27th, 2007)
6. VirtuaLinux at NSS07 Conference (invited talk, November 27th, 2007,

OpenSource



Project Web Traffic For Virtualinux
All Time
Hits/month
MB/month
1200
1000
800
600
400
200
0
250
200
150
100
50
0
2007-05
2007-06
2007-07
2007-08
2007-09
2007-10
Hits/month
Pages
Bandwidth
Generated 2007-10-11 22:32:26 UTC
Copyright SourceForge.net

Xen architecture