VirtuaLinux: Virtualized High Density Clusters with no Single Point of Failure

M. Aldinucci, M. Danelutto, M. Torquati, F. Polzella, G. Spinatelli, M. Vanneschi, A. Gervaso, M. Cacitti, P. Zuccato

Parco 2007
RWTH Aachen, Forschungszentrum Jülich (Germany) 3rd-7th September 2007

CONTENTS

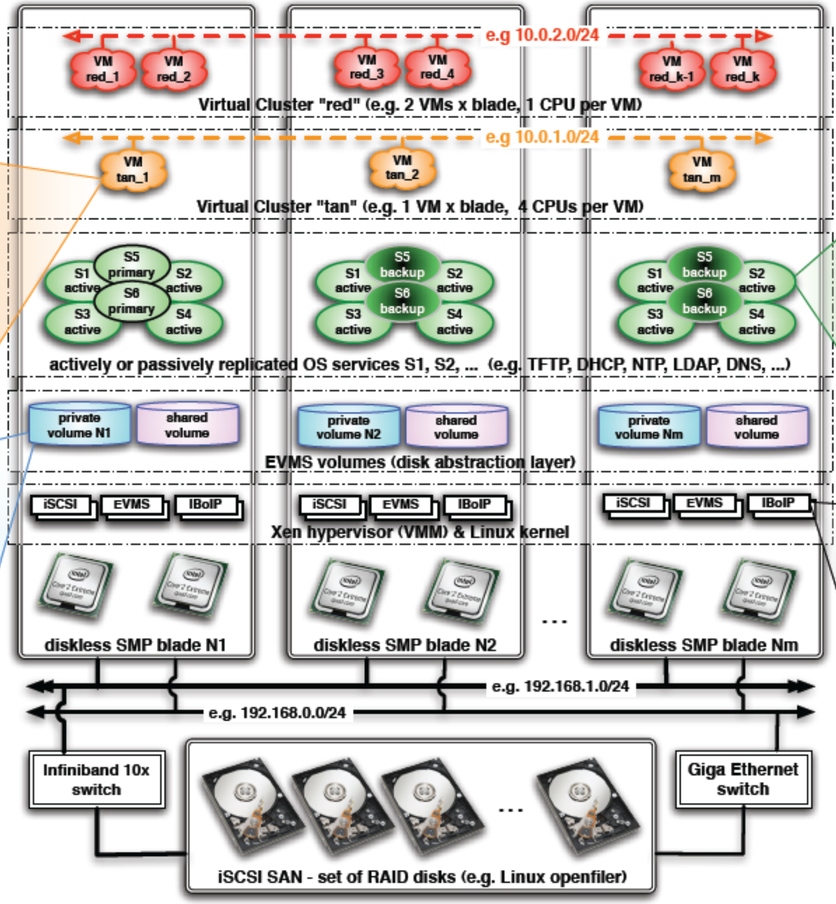
OVERVIEW: COMMON FLAWS

- Cluster configuration is hard to evolve as needs change (reinstallation)
- One cluster → one environment (how to cope with GRIDs, Beowulf, etc?)
- Strong coupling between physical and logical architectures
- Single point of failure (master node)
- Multiple points of failure (local hard disks)
- Resource waste (OS replication for homogeneous nodes)
- Configuration is not reusable (installation procedure must be redone for identical machines)

OVERVIEW: VirtuaLinux

Virtual clusters "VC"

Virtual storage layer



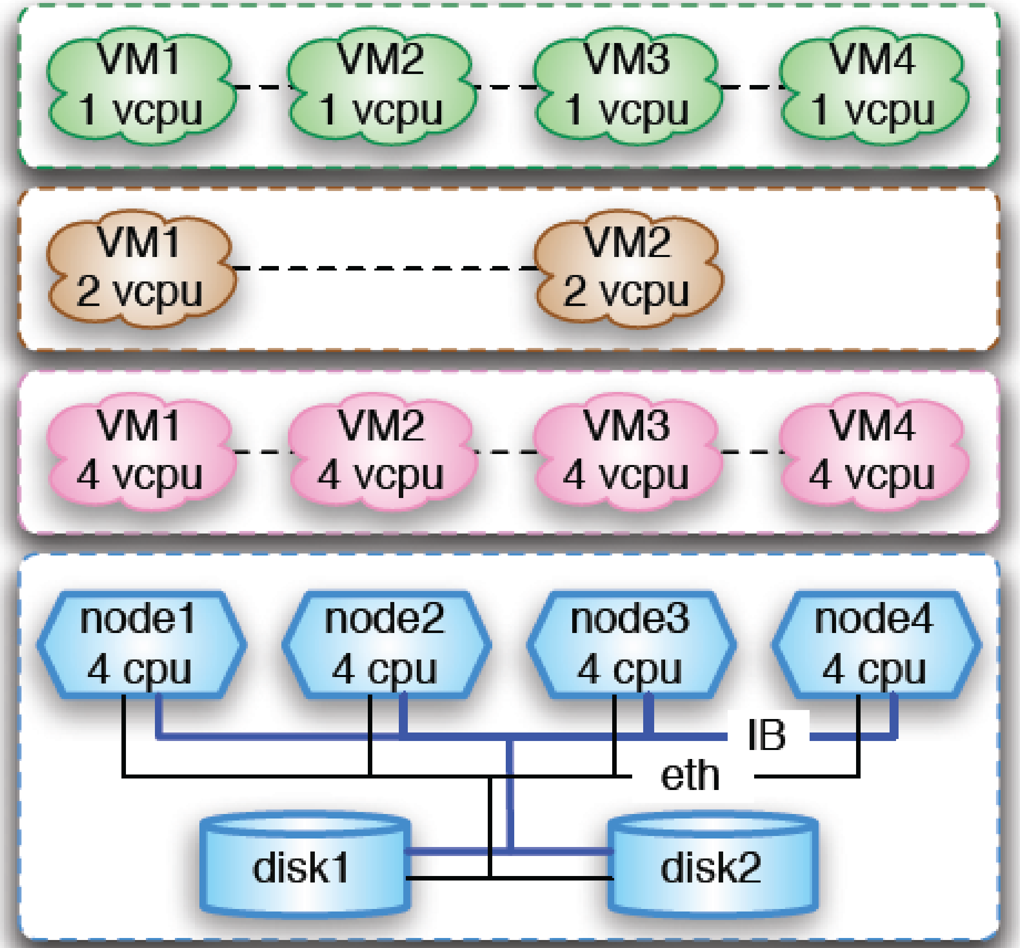
Service replication

Standard kernel

… and the tools to manage this

CONTENTS

VIRTUALIZATION: CLUSTER OS



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

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

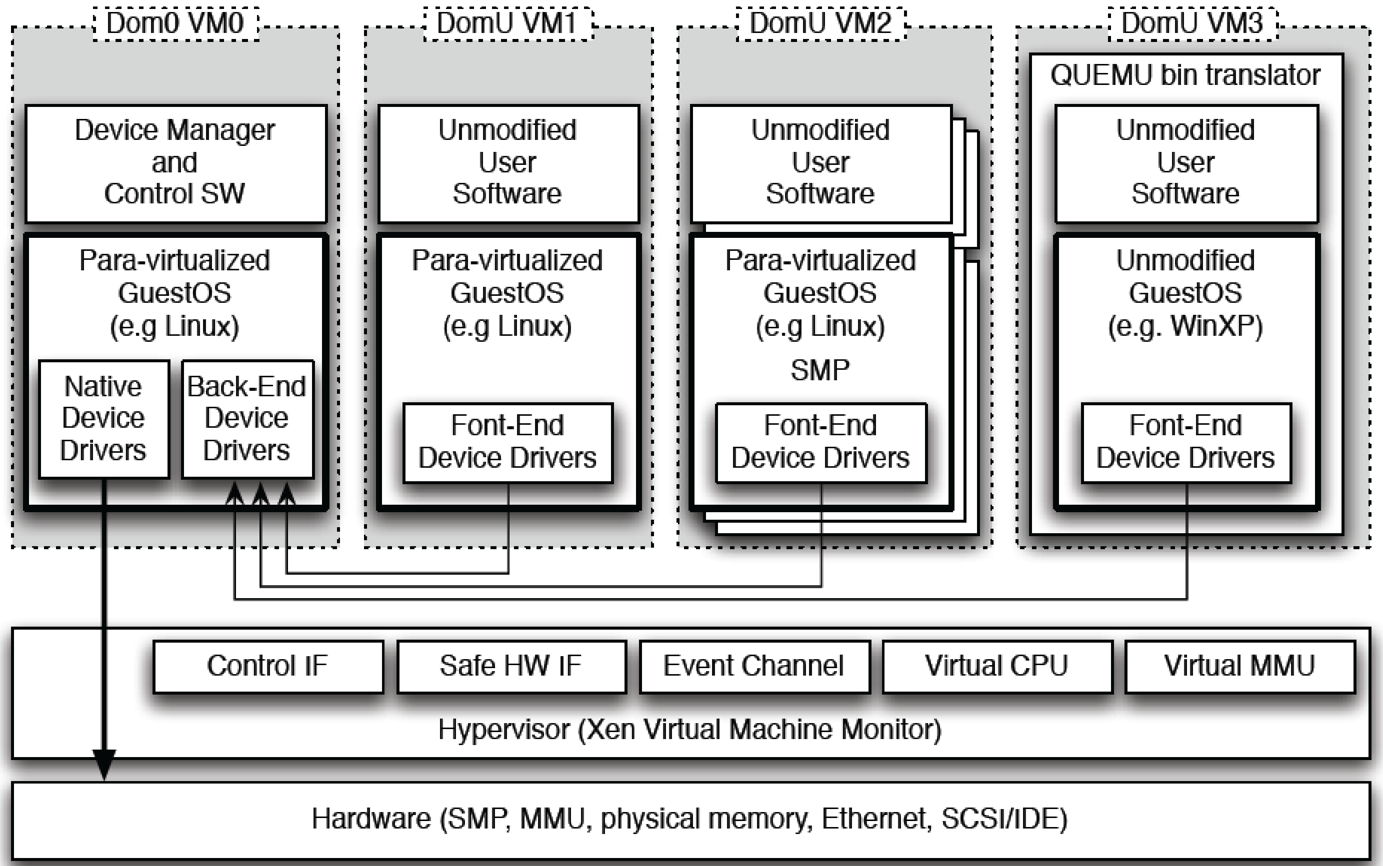
Virtual Cluster "pink"
4VMs x 4VCPUs
10.0.0.0/24

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

VIRTUALIZATION: CLUSTER OS

- Three layers: physical, host, guest clusters
- The host cluster layer hides and protects hardware details
- Guest clusters are clusters of Virtual Machines (VM) that are totally insulated (Virtual Clusters, VC)
- Key features of VCs:
  - Crashes and instabilities are confined
  - VCs can have different configurations and OS (Beowulf, Grid, RH, Ubuntu, Windows).
  - Guest administrators don’t have to be very skilled (limited damage)
  - VC and VMs can be preconfigured and downloaded from repositories
  - VMs within a VC can be moved between physical nodes
  - Possibly, superscalability (I/O bound and CPU bound VCs running on the same physical nodes)

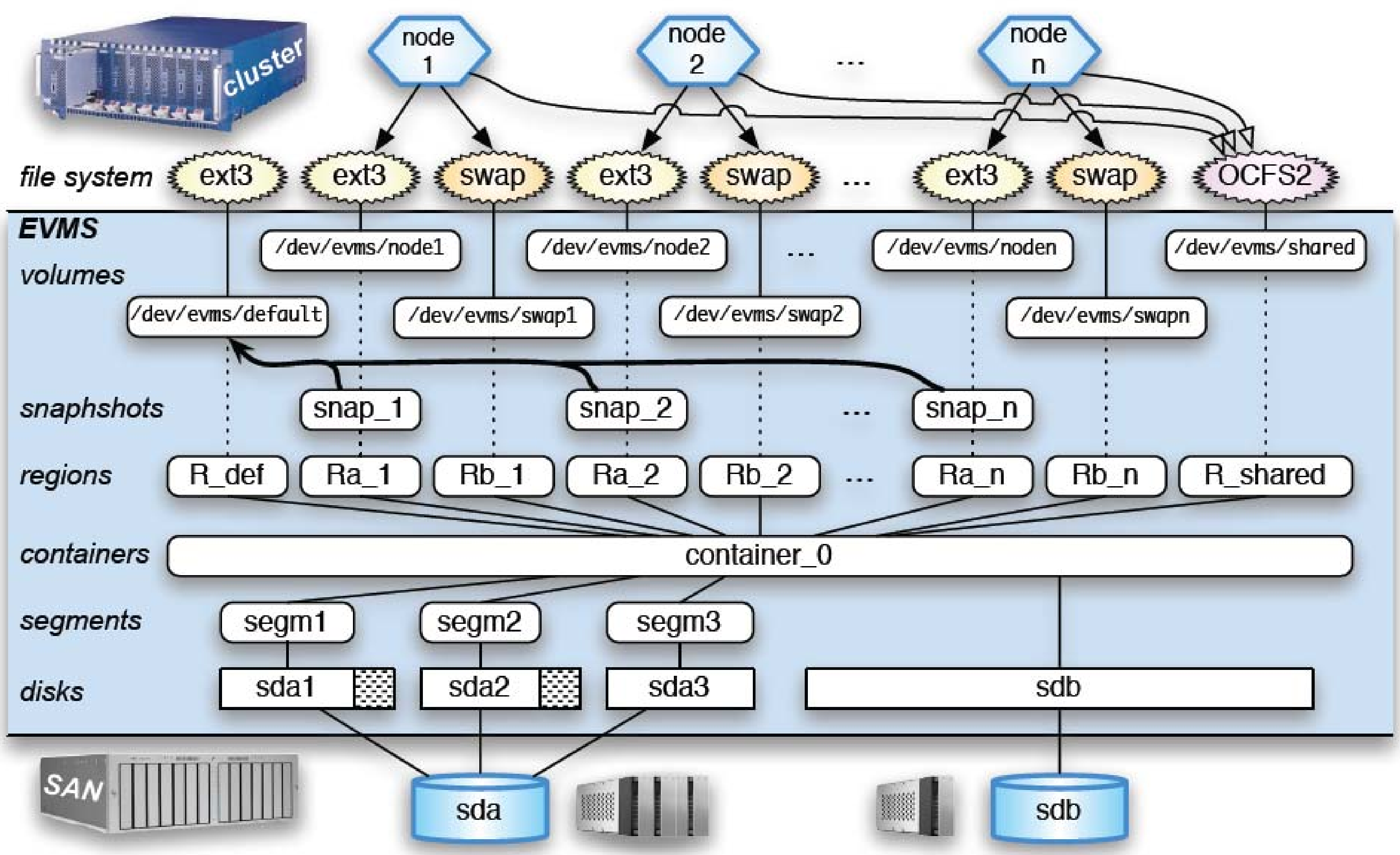
VIRTUALIZATION: CLUSTER OS



A physical node running 3 VMs (2 to 3 VCs nodes. One runs Windows through QUEMU). Xen provides the virtualization

CONTENTS

VIRTUALIZATION: CLUSTER STORAGE



VirtuaLinux, EVMS and iSCSI

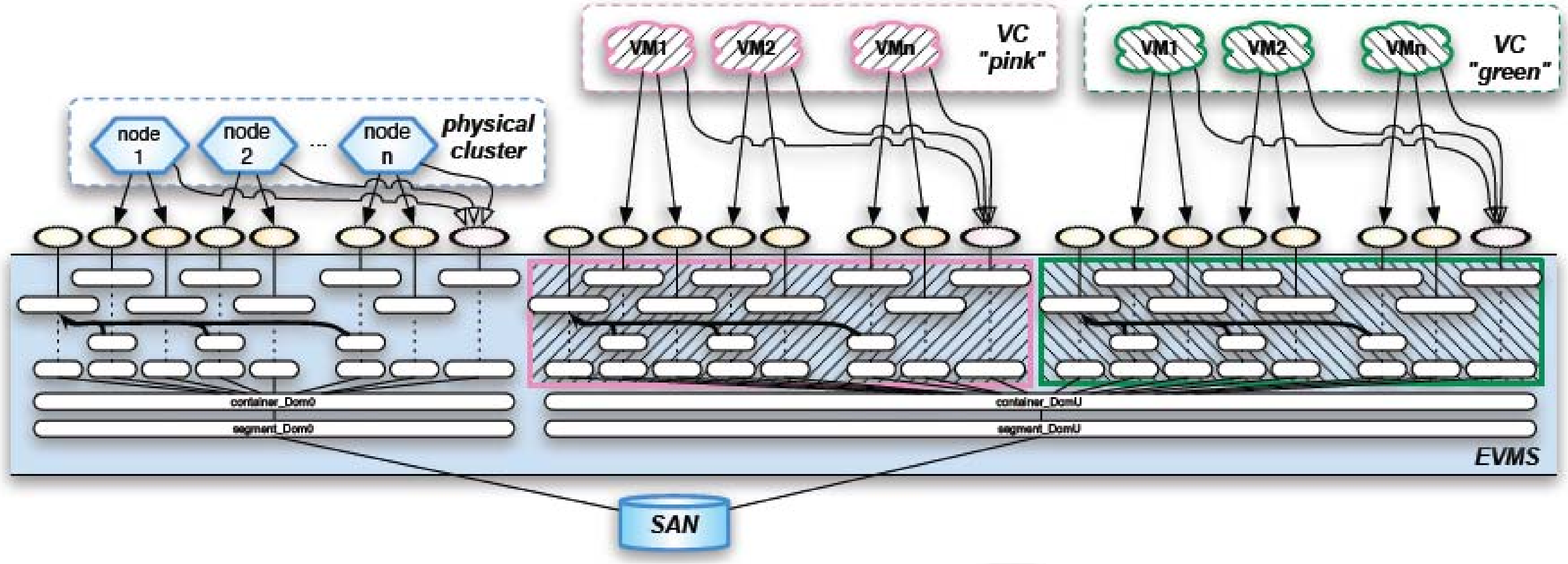
VIRTUALIZATION: CLUSTER STORAGE

- Storage virtualization in VirtuaLinux means EVMS+creative use of snapshots on an iSCSI SAN
- iSCSI provides standard, simple, cheap and cluster wide access to the remote, redundant disks, allowing diskless nodes
- iSCSI can reuse existing infrastructures (GigEth, Infiniband). No additional cost, cabling, adapters or drivers
- EVMS insulates the physical details
- EVMS provides a single, unified system for storage management
- EVMS logical volume naming avoids the use of raw device names (iSCSI devices connected to different nodes would have different names)

VIRTUALIZATION: CLUSTER STORAGE

- Generally, all VC virtual nodes share the same OS image, with minor differences
- 1000 virtual nodes -> 1000 replicas. Disk and time waste
- The same is true for classic clusters
- Snapshot is traditionally used to backup volumes, **but can be used to create replicas (volumes)**
- Modified snapshot technique:
  - Create the volume and install the OS image
  - Make N snapshots of the original (for N nodes)
  - The snapshots are created using ONLY metadata and pointers to the original copy
  - Creation time of N snapshots almost independent from N (seconds)
  - Modifications are lightweight: a modification to a file just involves the copy and update of the single file
- Memory buffers in kernel limit the N. VirtuaLinux extends EVMS semantics (one line of C code) to allow any N

VIRTUALIZATION: CLUSTER STORAGE



As for the physical cluster, each VC node has a VC private virtual shared storage (OCFS2 distributed FS), a private virtual disk and a private virtual swap area.

Private virtual disks are functionally identical to traditional local disks

Virtual means that the volume is a snapshot

CONTENTS

FAULT TOLERANCE

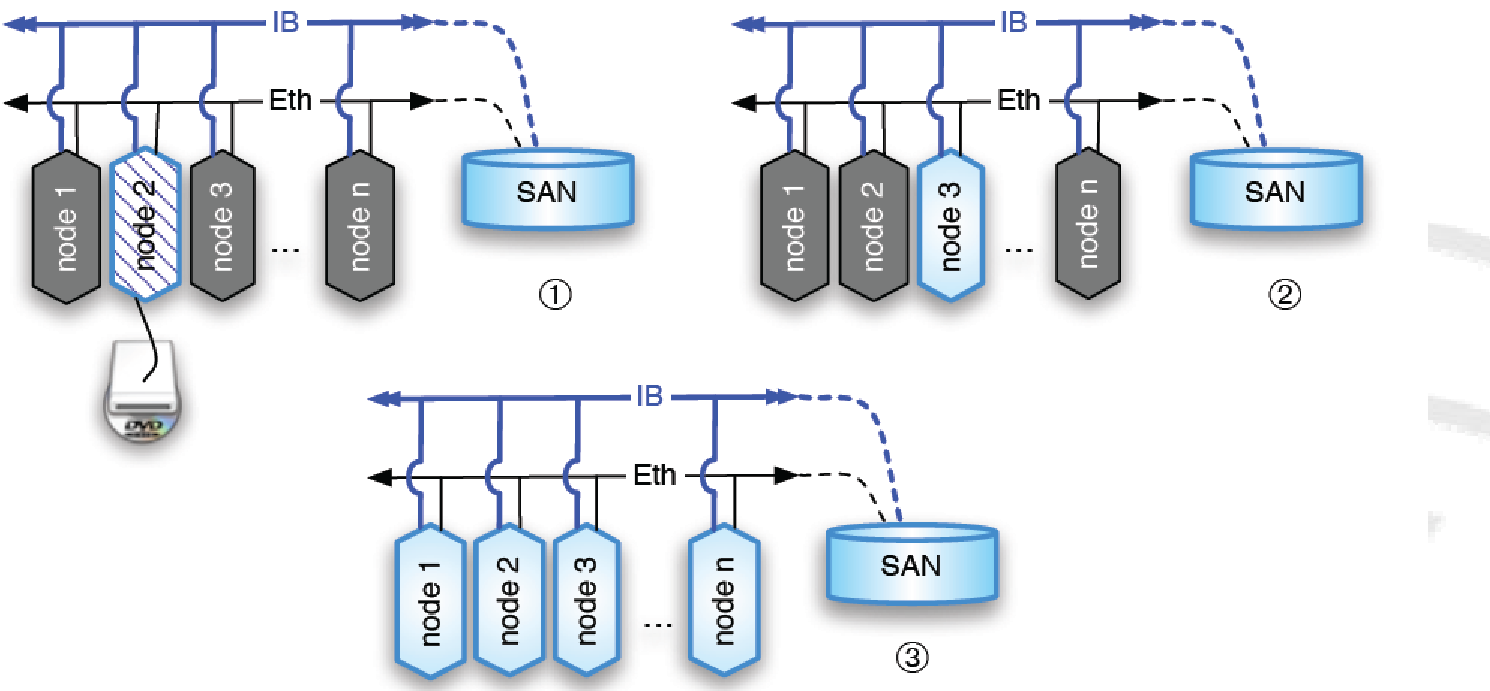
- Traditionally, clusters have a master node. Great risk!
- VirtuaLinux is a masterless architecture
- Active and passive replication (primary-backup) of services
- Service classification:
  - Stateless: Active replication (e.g IB manager). Clients automatically locate the server (broadcasting, etc). Most reactive server replies
  - Stateful: Passive replication (IP Gateway). Only one server exists and is statically known to clients (IP, MAC). Heartbeat with IP takeover between primary and backup servers
  - Node oriented: failure of node specific services is potentially catastrophic for the node only (local SSH, NFS). The rest of the cluster is unaffected. Outsite the scope of VirtuaLinux. Scheduler level?
  - Self-healing: fault tolerance is embedded in the service (NTP)

FAULT TOLERANCE

Service	Fault-tolerance	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 DSN
LDAP	service-specific	Service-specific master redundancy
IP GW	passive	Heartbeat on 2 nodes with IP takeover (HA)
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

FAULT TOLERANCE

- How to install a masterless cluster? Chicken and egg problem!
- Solution: Metamaster! A temporary installation node that is then transformed in a standard node
- At the end of the installation, all nodes are identical and provide redundant services. Bye bye master node



CONTENTS

TOOLS

- VirtuaLinux is a set of scripts, independent of the Linux Distribution
- VirtuaLinux Virtual Cluster Management (VVCM) is a subset of scripts that allow the creation and management of VCs
- Main components are:
  - Database of the physical and virtual clusters (includes the mapping between physical and virtual clusters, virtual nodes, etc)
  - Command line library for the creation, activation and destruction of a VC (VC_Create, VC_Control, VC_Destroy)
  - Communication layer (for staging and executing VMs)
  - VC start time support for dynamic configuration of network topology an routing policies of the physical nodes

CONTENTS

DOES IT WORK?

Micro-benchmark	Unit	Ub-Dom0	Ub-DomU	CentOS
Simple syscall	usec	0.6305	0.6789	0.0822
Simple open/close	usec	5.0326	4.9424	3.7018
Select on 500 tcp fd's	usec	37.0604	37.0811	75.5373
Signal handler overhead	usec	2.5141	2.6822	1.1841
Protection fault	usec	1.0880	1.2352	0.3145
Pipe latency	usec	20.5622	12.5365	9.5663
Process fork+execve	usec	1211.4000	1092.2000	498.6364
float mul	nsec	1.8400	1.8400	1.8200
float div	nsec	8.0200	8.0300	9.6100
double mul	nsec	1.8400	1.8400	1.8300
double div	nsec	9.8800	9.8800	11.3300
RPC/udp latency localhost	usec	43.5155	29.9752	32.1614
RPC/tcp latency localhost	usec	55.0066	38.7324	40.8672
TCP/IP conn. to localhost	usec	73.7297	57.5417	55.9775
Pipe bandwidth	MB/s	592.3300	1448.7300	956.21

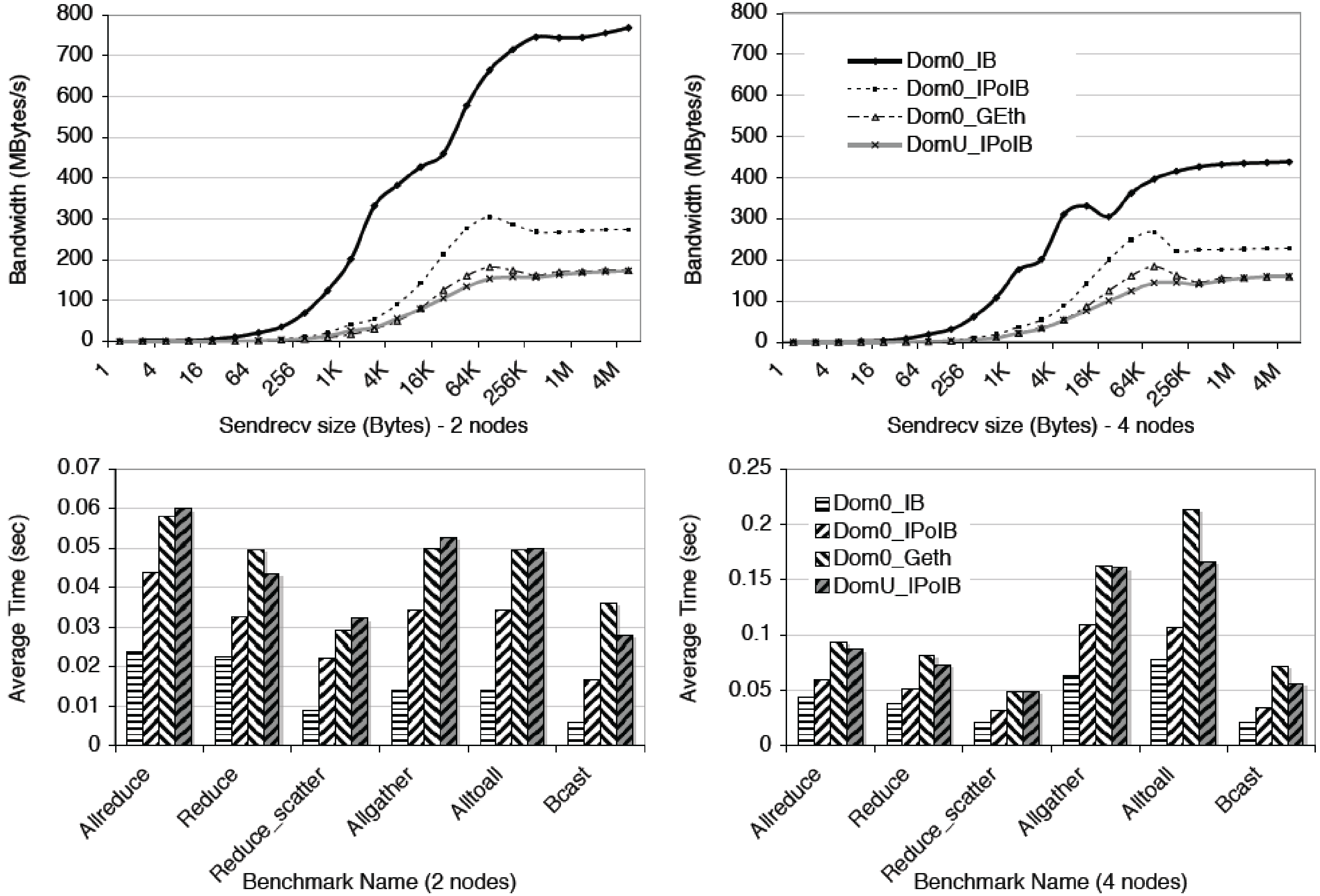
LM microbenchmark:
4 nodes, dual Opteron 280, 8GB ram, GigEth, SDR Infiniband
Environments: Host (Ub-Dom0), Guest (Ub-DomU), not virtualized (CentOS)

DOES IT WORK?

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%

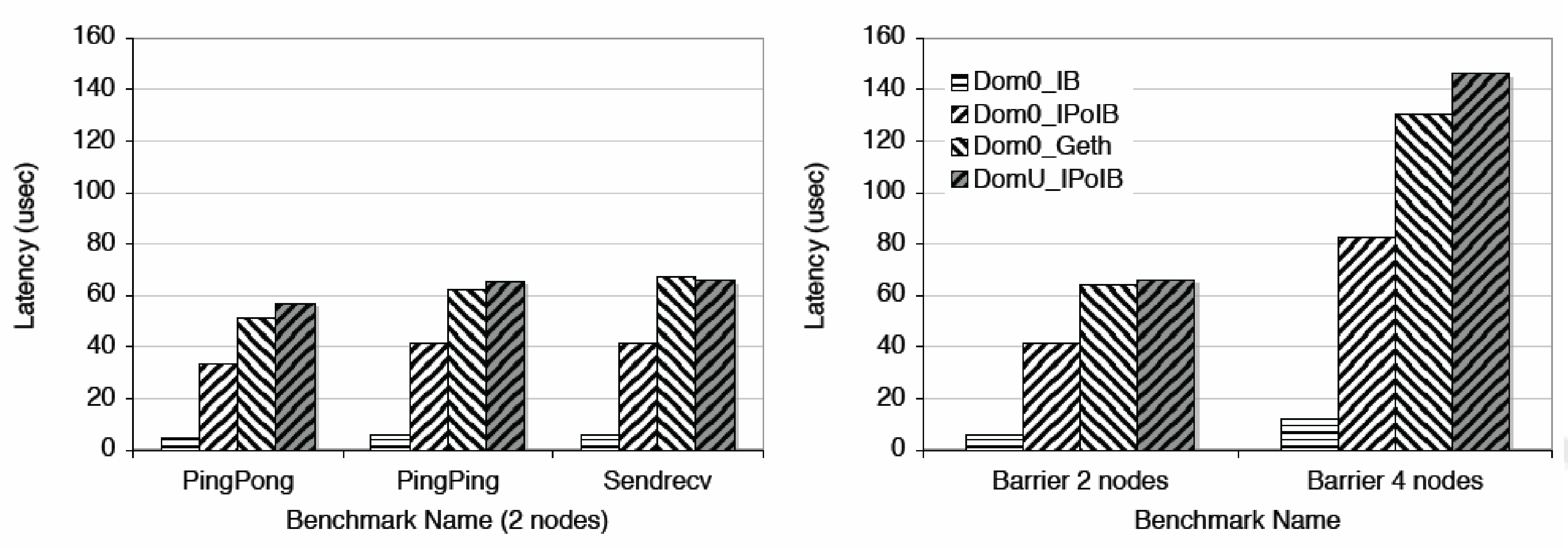
Virtualization impacts communications and system calls, not math

DOES IT WORK?



Intel MPI Benchmarks (VMAPICH): Bandwidth, collective communications

DOES IT WORK?



Intel MPI Benchmarks (VMAPICH): Latency

DOES IT WORK?

- (Para)Virtualization impact syscalls heavily, but the real cost for computationally intensive applications is not clear
- (Para)Virtualization has no cost for processor instructions (e.g. math)
- (Para)Virtualization significantly penalizes communication bandwith and latency, but:
  - Current native IB drivers (with user space verbs) cannot be used within VMs
  - TCP is used (IPoIB) within VMs. This is a major source of overhead
  - Most overhead will be eliminated when user-space IB verbs for Xen VM are available (currently under development, see J. Liu, W. Huang, B. Abali and D.K. Panda. High Performance VMM-Bypass I/O in Virtual Machines. USENIX Annual Technical Conference 2006, Boston, MA, May 2006.)

CONTENTS

CONCLUSIONS

- VirtuaLinux enables a virtualized, diskless, masterless cluster architecture
- Virtual clusters are insulated from the physical and host cluster, allowing multiple simultaneous VCs on the same hardware
- Storage virtualization coupled with a novel snapshot technique dramatically reduces installation time and permits a centralized management of the VCs
- Storage virtualization permits the backup and restore of entire VCs
- The absence of an individual master and local hard disks replaced by a SAN of redundant arrays avoids single points of failure
- **IB Verbs within VMs are absolutely required for real use of VirtuaLinux in a production environment, but they are coming…**

ACKNOWLEDGMENTS

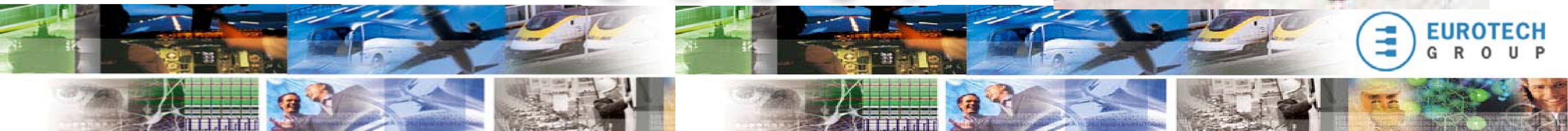
- VirtuaLinux is an ongoing experiment of Eurotech SpA and the HPC laboratory of the Computer Science Department of the University of Pisa

- VirtuaLinux is a open source software under GPL available at http://virtualinux.sourceforge.net/

- VirtuaLinux project has been supported by the initiatives of the LITBIO Consortium, founded within FIRB 2003 grant by MIUR, Italy.

- We are grateful to Peter Kilpatrick for his help in improving the presentation.

EUROTECH
GROUP



THANK YOU FOR YOUR ATTENTION

EUROTECH
G R O U P

"On résiste à l'invasion des armées; on ne résiste pas à l'invasion des idées.“

Victor Hugo, Paris 1877