Virtualization for the LHCb Online system

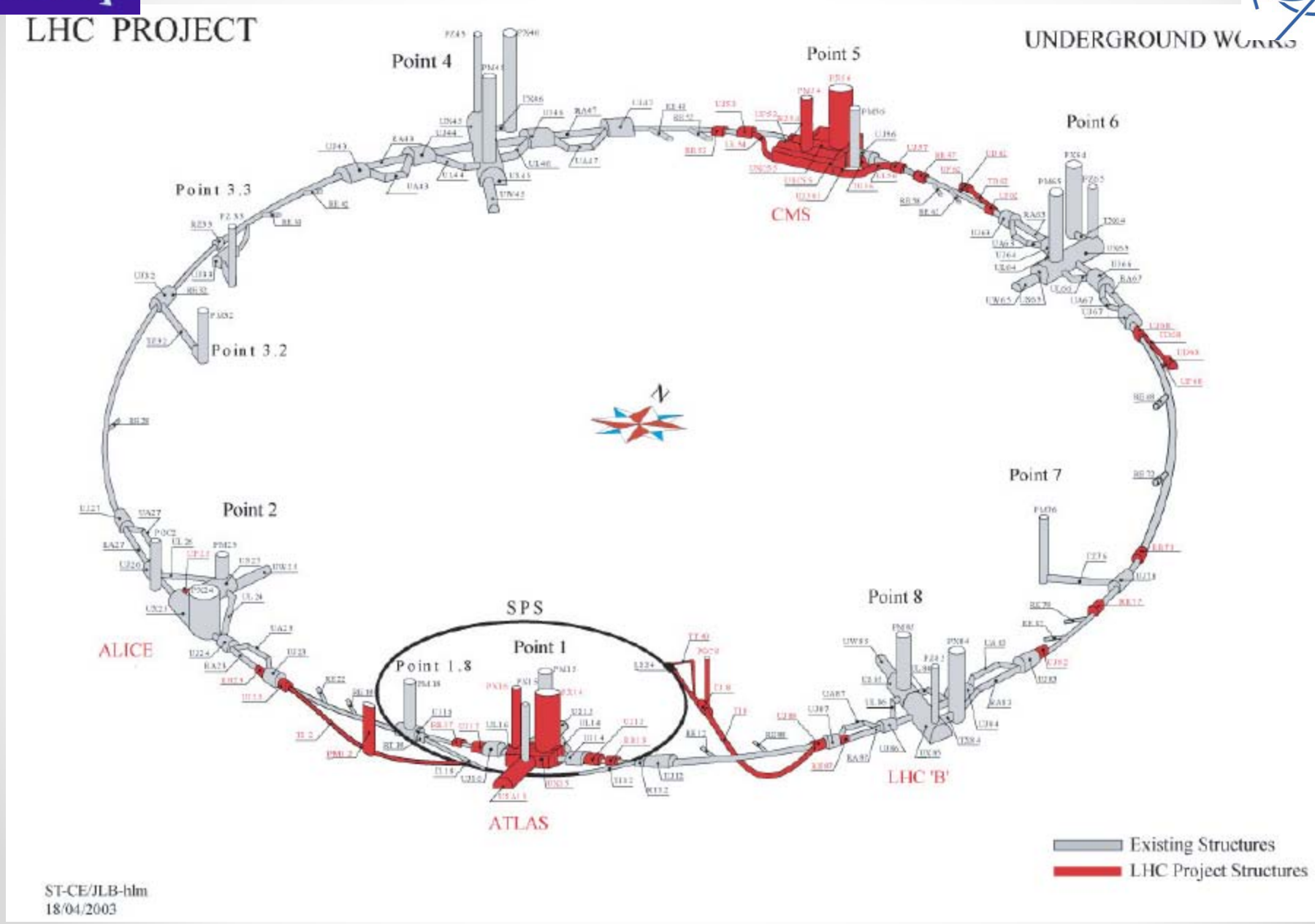
CHEP 2010 - Taipei

Dedicato a Zio Renato

Enrico Bonaccorsi, (CERN) enrico.bonaccorsi@cern.ch
Loic Brarda, (CERN) loic.brarda@cern.ch
Gary Moine, (CERN) gary.moine@cern.ch
Niko Neufeld, (CERN) niko.neufeld@cern.ch
Alexander Zvyagin , (CERN) alexander.zvyagin@cern.ch

Outline

- LHCb
  - What is LHCb
  - Online system & Experiment Control System
- Virtualization
  - What we virtualize
  - The choice of the hypervisor
  - Hardware used
- Architecture
  - General Hyper-V
  - LHCb Network & Security implementation
- Performance
  - Network
  - Hard disks
- Quattor integration
- Issues


LHC PROJECT
UNDERGROUND WORKS
Point 4
Point 5
Point 6
Point 3.3
Point 3.2
CMS
Point 7
Point 2
Point 8
SPS
Point 1
Point 1.8
ALICE
LHC 'B'
ATLAS
N
Existing Structures
LHC Project Structures
ST-CE/JLB-hlm
18/04/2003

What do we virtualize?

- Traditional Virtualization approach: Not Cloud Computing

- General log in services/ Terminal services
  - RDP windows remote desktops
  - SSH gateways
  - NX linux remote desktops
- Web services
  - 1 VM per Website
- Infrastructure services
  - DNS
  - Firewalls
  - Domain controllers

- Control PCs
  - Controlling detector hw, running PVSS(standard LHC SCADA System)
  - Running both on Linux and Windows
  - Some of them need special hardware to control the detector
    - SPECS (special dedicated PCI card)
    - CANBUS (USB)
    - Several more

Hypervisor

allow multiple operating system to run on a host computer

- 4 solutions with active community/support behind:
  - Xen
    - Currently available on Scientific Linux 5
    - Will be replaced by KVM for Scientific Linux 6
  - KVM
    - Necessary Kernel modifications for Scientific Linux 5
  - Vmware
    - Suitable, high price
  - Hyper-V core R2 (free edition)

Hardware & SAN

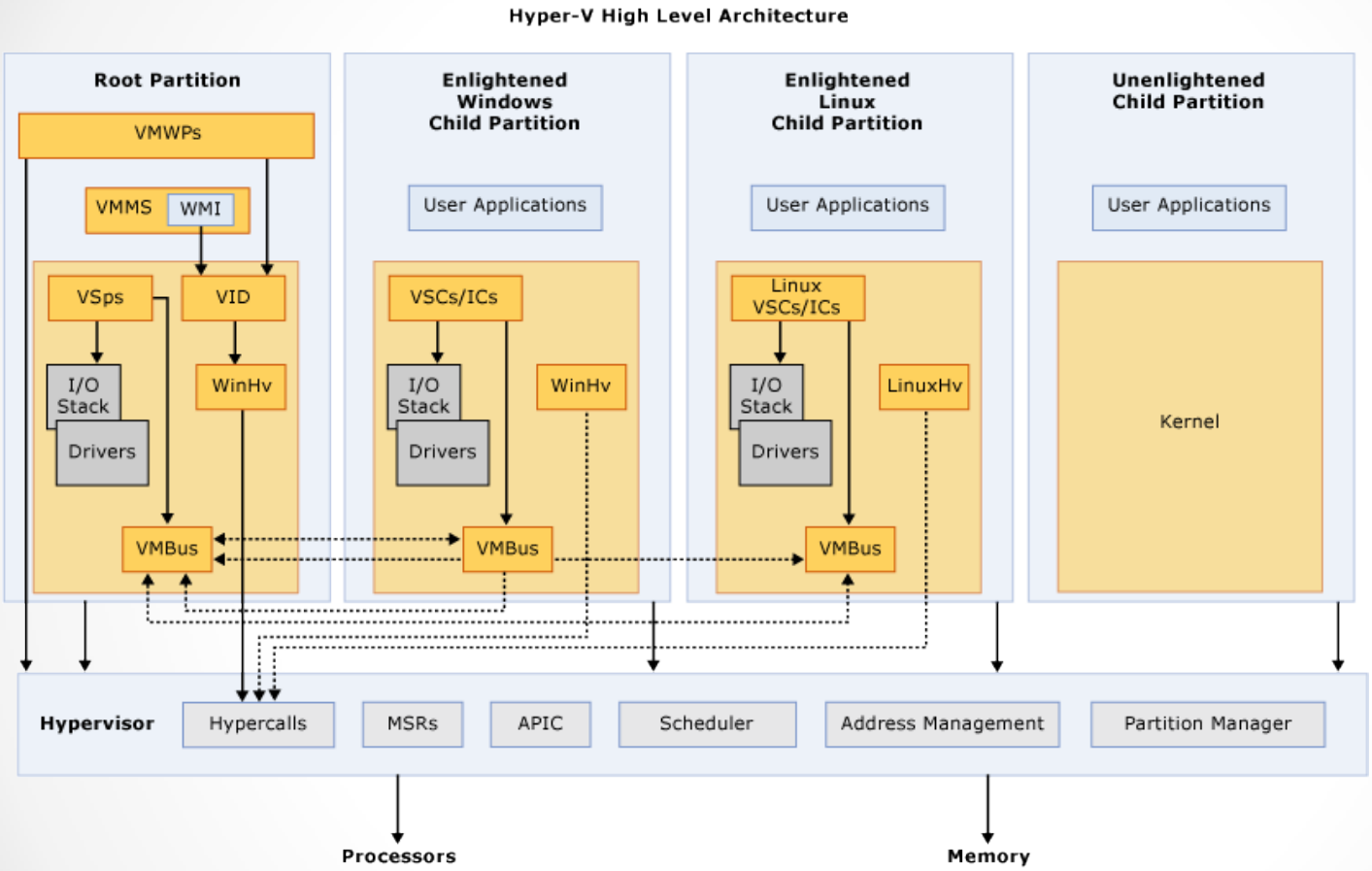
- 10 Blade Poweredge M610
  - 2 x E5530 @ 2.4GHz (8 real cores + Hyper Threading)
  - 3 x 8 GB = 24GB RAM
  - 2 x 10Gb network interfaces
  - 2 X 1Gb network interfaces
  - 2 X 8Gb fiber channel interfaces

- Storage
  - 2 X 8Gb Fiber channel switches
  - 10 Terabytes for Virtual Machines storage exported from 2 array controllers trough 2 independent fiber channel fabrics

- Network
  - 2 X 10Gb Ethernet switches
  - 2 X 1Gb Ethernet switches

- Limits:
  - Average of 20 VM per Server = ~200 Virtual Machines

Architecture

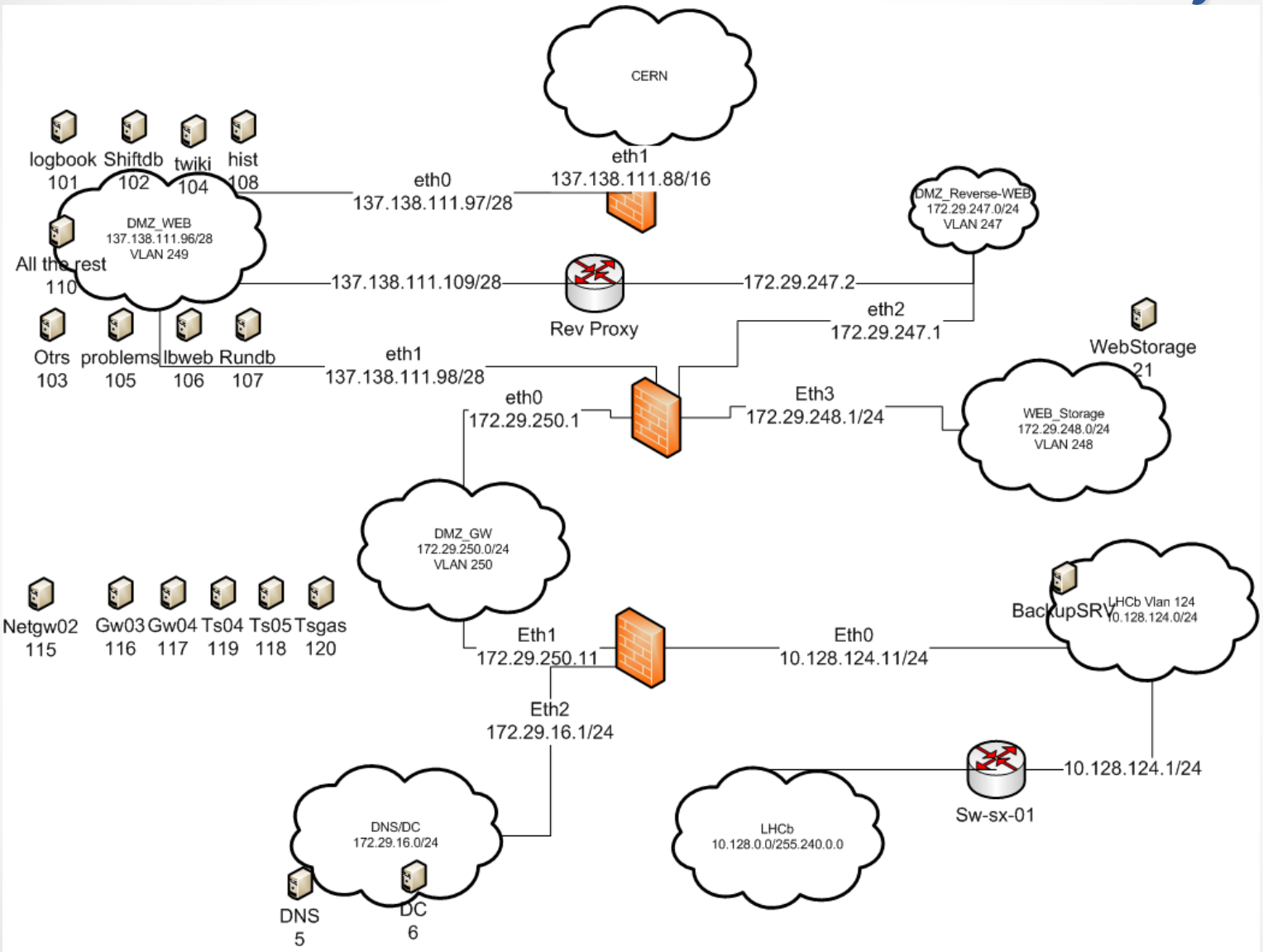


VMWP – Virtual Machine Worker Process

VSP – Virtualization Service Provider

VID – Virtualization Infrastructure Driver

Network architecture & Security



Virtual Machines performances

- Network (from VMs to real server inside LHCb Network)
  - Throughput: ~900Megabit/second
  - Latency: ~0.2 ms

- Disk
  - *(512 B blocks – our disk controller always read in 4k blocks)*
  - Reading: ~45MegaByte/sec
  - Writing: ~35MegaByte/sec

Virtual machines & Linux cluster management (Quattor)

- Server installation managed by Quattor using network boot/PXE
- Boot from network:
  - not supported by para-virtualized network interfaces
  - supported by emulated network interfaces (very slow)

- **Solution:**
  - **Do not install**
  - **Use cloning of virtual hard disks (virtual machine template)**
  - **Custom post boot script adjust main config file according to the PTR DNS record of the IP acquired by DHCP**
  - **Let quattor configure the linux virtual machine**

  New virtual machines ready to be used in less than 10 minutes

Issues

- General issues
  - Time, ntpd -> ntpdate
  - PCI cards -> N/A
  - Usb -> Usb over IP
  - Software licenses: hardware dependent(PVSS)
- Hyper-V issues
  - Ethernet -> multicast n/a, jumbo frames n/a
- Hardware issues
  - Intel 5500 Series / hyper-v Core / ACPI
  - Cluster filesystem sector size = 512B

Summary and outlook

- Virtualization of LHCb ECS
  - Aim at reduce hardware
  - Special attention to security
  - Many issues tackled and solved (or work around)
- Next phase:
  - USB/IP
  - iSCSI
  - Virtualize almost every control pc
  - Intrusion prevention system

Backup slides

Virtualization CPU overhead

- We run over virtual machines based on KVM what we call the «moore test»
- Moore: software for trigger decision

- Running directly on the real machine we measured:
- ~10% overhead

Sharing of VLAN

- Massive using of 802.1q
- VLAN exported to real servers using a dedicated trunked 10Gb link