Storage Virtualization

Andreas Joachim Peters – CERN IT-DSS

Outline

- What is storage virtualization ?
- Commercial and non-commercial tools/solutions
- Local and global storage virtualization

Scope

of this presentation

Layout of a Virtual Storage System

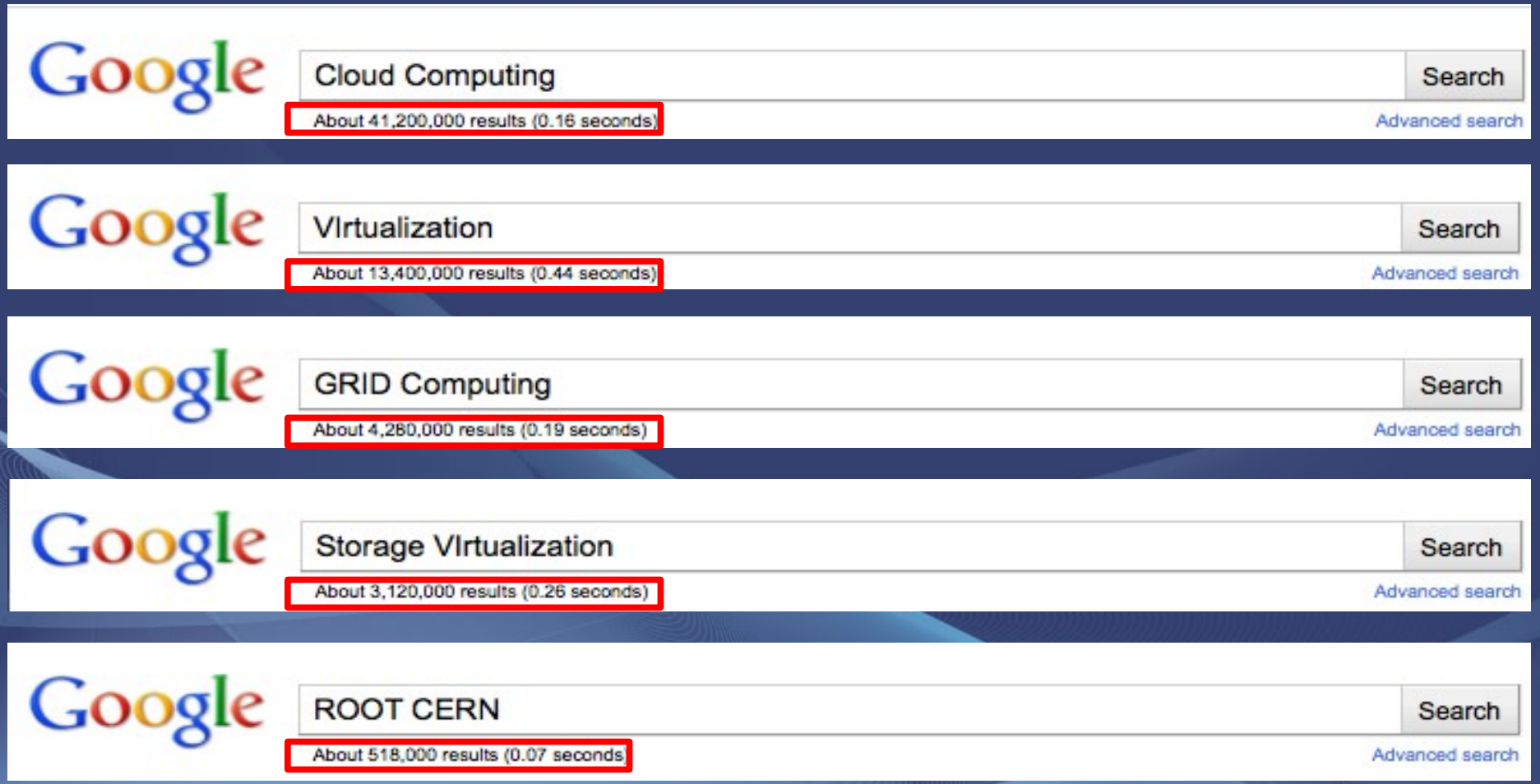
as volatile and persistent storage for experiment and user data!

Disclaimer:

This is a personal view!

Before we see what virtual storage means:

How 'interesting' is that?



It's not the mainstream topic (yet?) **!**

What is that (1) ?

- Storage virtualization is a concept in IT System Administration, referring to the **abstraction (separation) of logical storage from physical storage** so that it may be accessed without regard to physical storage or heterogeneous structure.

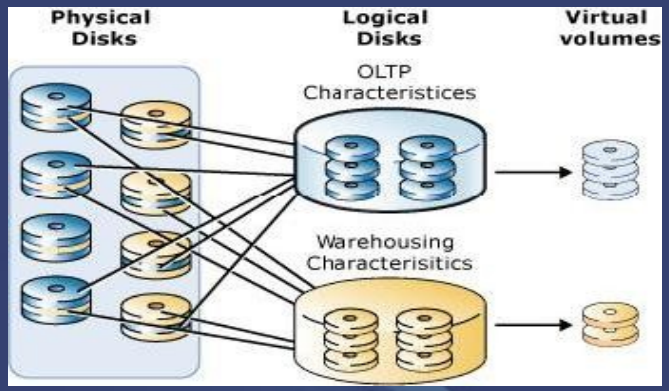
- This separation allows the Systems Admin increased flexibility in how they manage storage for end users.

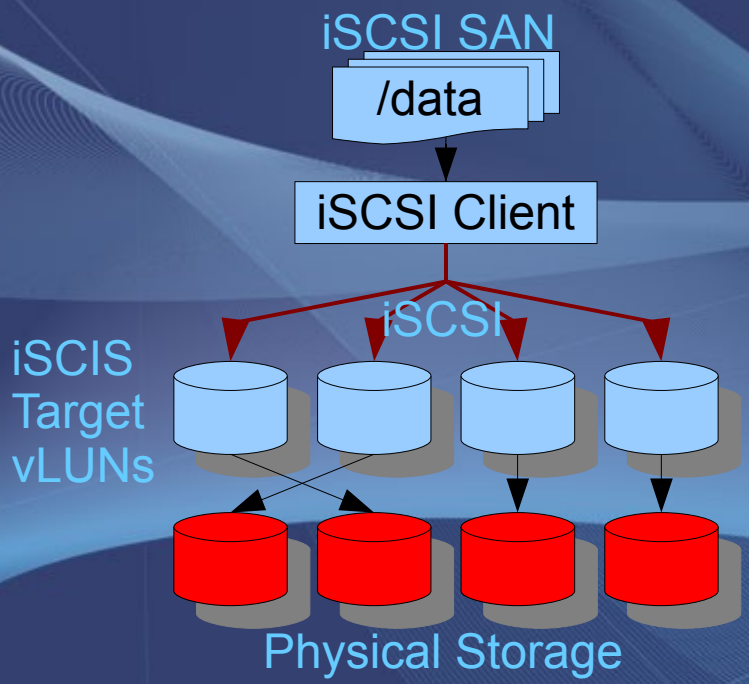
What is that (2) ?

http://www.virtualtechnologies.com/virtualization/whatisvirtualization.html

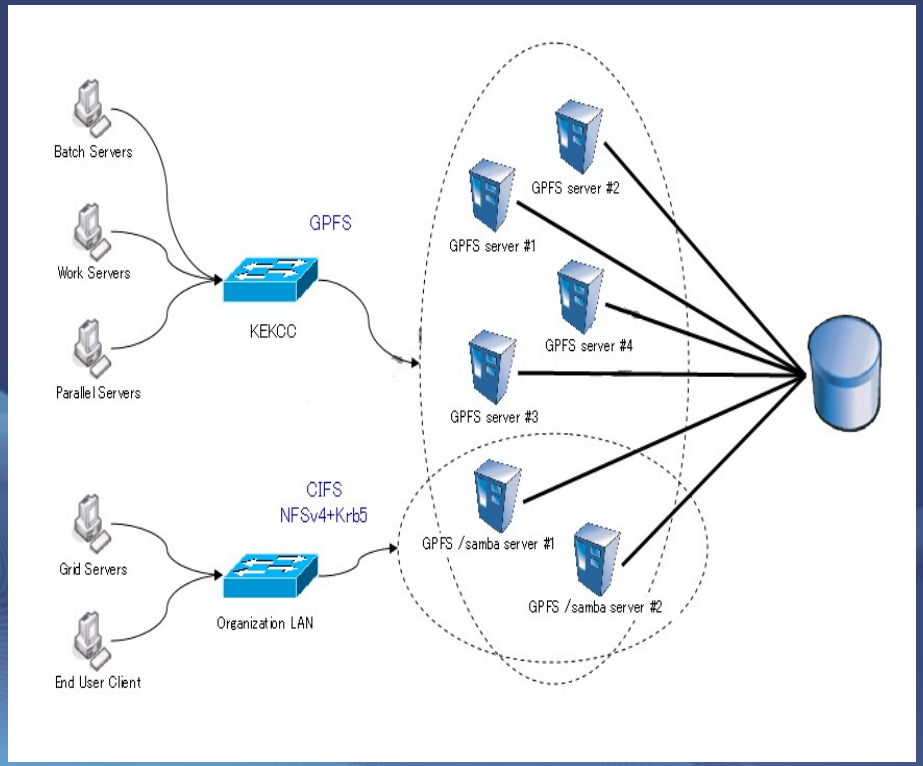
- **DEFINITION** - Virtualization is the pooling of physical storage from multiple network storage devices into what appears to be a single storage device that is managed from a central console. Storage virtualization is commonly used in a storage area network (SAN). The management of storage devices can be tedious and time-consuming. Storage virtualization helps the storage administrator perform the tasks of backup, archiving, and recovery more easily, and in less time, by disguising the actual complexity of the SAN.
- Users can implement virtualization with **software applications** or by using **hardware and software hybrid appliances**. The technology can be placed on different levels of a storage area network.

Few Examples Storage Virtualization





GPFS Cluster



Some tools used in virtualization ...

- http://iscsitarget.sourceforge.net/

The iSCSI Enterprise Target Project

- http://www.openfiler.com/

Storage Deployment Models

Separation of Computing and Storage Nodes

VM COMPUTING NODES

SAN / NAS

STORAGE NODES

Every node contributes as Computing and Storage Node

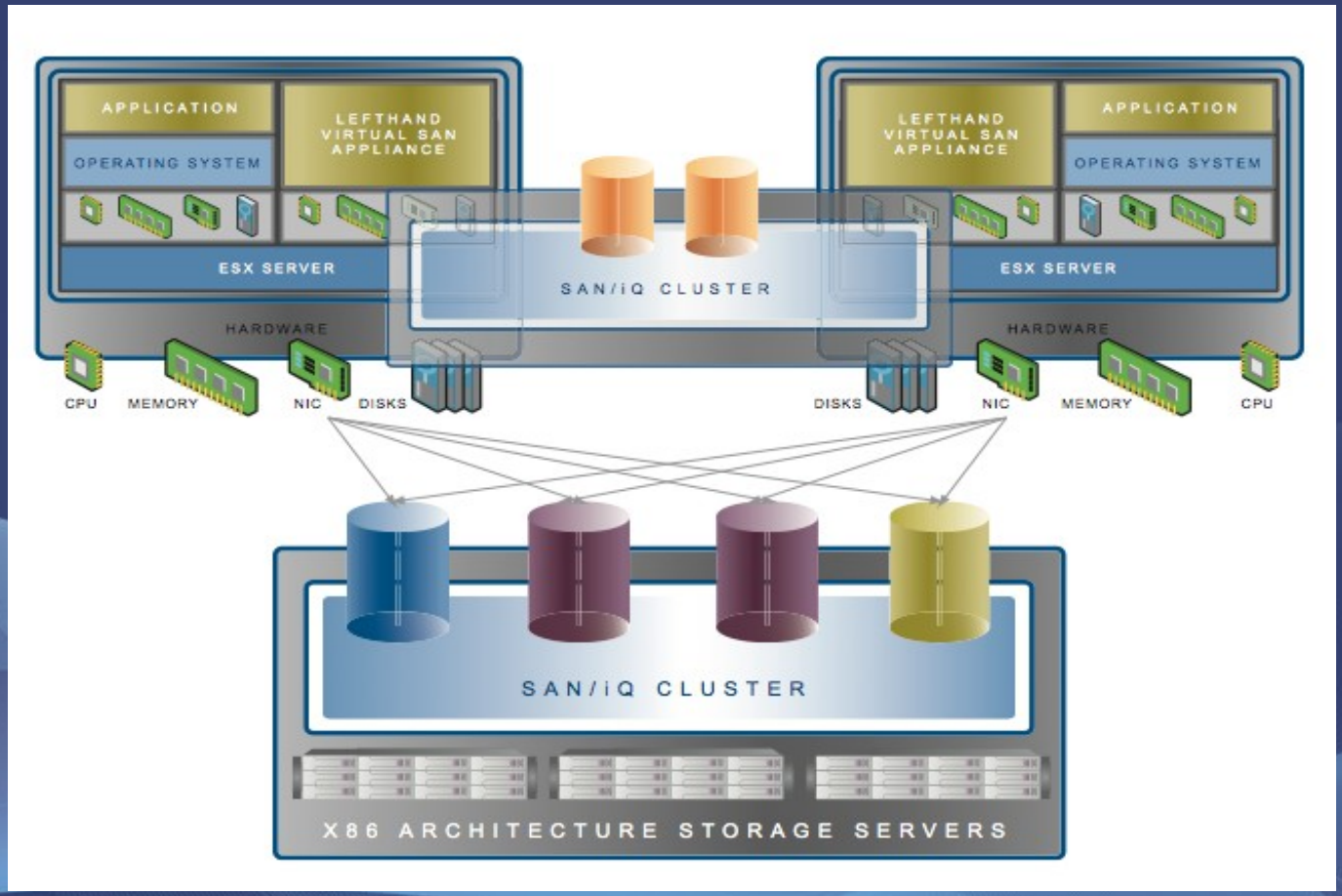
Cheaper
But CPU/DISK
Is somehow fixed

Which one is better depends on the use case and the budget!

TIER 0 STORAGE

ANALYSIS FARM

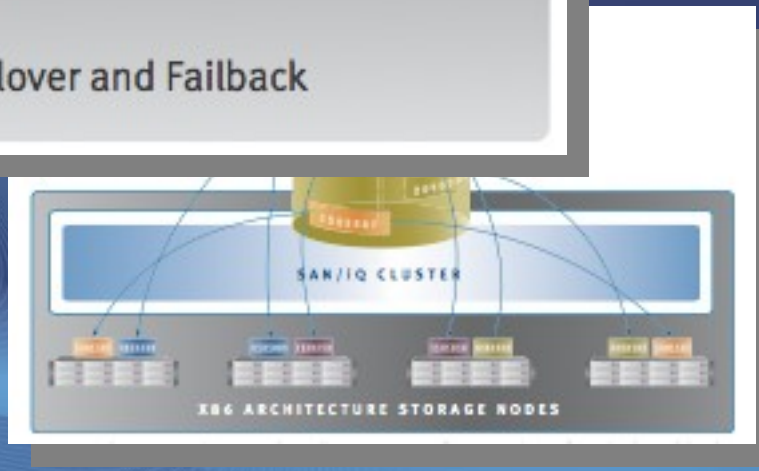
Virtual SAN Appliance VSAN Software



Commercial Virtual Storage Solution
HP LeftHand SAN

HP StorageWorks P4000 G2

SAN/iQ® Network RAID
Integrates Synchronous Replication with Automated Failover and Failback



Virtual SAN Appliance Software VSAN

Commercial Virtual Storage Solution
HP LeftHand SAN

All Inclusive Feature Set Enables Enterprise Functionality at an Affordable Price

- Storage Clustering allows a customer to consolidate multiple storage nodes into pools of storage. All available capacity and performance is aggregated and available to every volume in the cluster. As storage needs increase the HP P4000 G2 can scale performance and capacity on-line.
- Network RAID stripes and protects multiple copies of data across a cluster of storage nodes, eliminating any single point of failure in the HP P4000 G2 SAN. Applications have continuous data availability in the event of a disk, controller, storage node, power, network, or site failure.
- Thin Provisioning allocates space only as data is actually written without requiring pre-allocation of storage. This raises the overall utilization and efficiency of the HP P4000 G2 SAN, reduces costs and ultimately increases the ROI.
- Snapshots create thinly provisioned, instant point-in-time copies of data on a per-volume basis. Administrators access snapshots to recover individual files from the volume, or rollback an entire volume. Built-in application integration provides automated quiescing for Microsoft VSS applications.
- Remote Copy replicates snapshots between P4000 G2 SANs at primary/remote locations. Copies are thinly provisioned with no space reservation required. Remote Copy enables centralized backup and disaster recovery on a per-volume basis and leverages application integrated snapshots for faster recovery.

Commercial Virtual Storage Solution
HP LeftHand SAN

- *Network RAID Level 0* stripes data across the cluster and stores one copy of each block. Useful for temporary files and backup data, this RAID level can provide continuous availability across many single-component failures. If a failure makes a node unavailable, the volume becomes unavailable.
- *Network RAID Level 2* stores two copies of each volume's block, providing continuous data availability across any single node failure. This is the most popular Network RAID level with customers.
- *Network RAID Level 3* stores three copies of each block for mission-critical data that needs to be available despite any double node failure.
- *Network RAID Level 4* is used in situations where a cluster is divided between two locations, and the data must be continuously available in the event of both a site failure and a node failure at the alternate site, as described below.

Also Network RAID 5 & 6 to save space (released 1.4.2010)

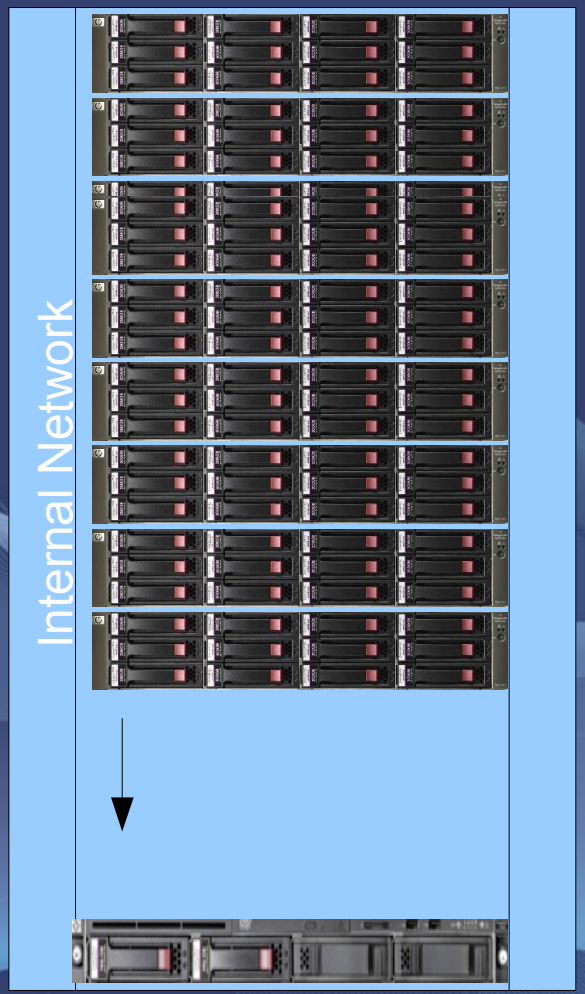
How would one (like to) use storage virtualization?

- Rack/Cluster **Local Storage Virtualization**
  - Dynamic Volume assignment and management in a rack/cluster
  - But, can we let all machines share the same namespace? Can we have one single mount point for all available storage?
- **Global Storage Virtualization**
  - Can we further virtualize the federation of all local clusters to have a single global mount point? A global file system?

VSAN alone cannot provide these!

•This subject is closely related to DM Jamboree in Amsterdam last week!

A virtualized one rack 'computer center' with virtualized computing and storage for analysis



- To create storage inside a single namespace one needs a cluster filesystem or a clustered file access system running server on the hypervisors e.g.
  - GPFS IBM (licensed)
  - Ibrix HP (licensed)
  - XtreemFS (open source)
  - Scalla/xrootd (open source)

  [primarily access via remote access protocol]

  ...
- VMs run the appropriate (FS) clients
  - Authentication is an issue
  - Storage HA/Failover required
  - Block Level or File Level Redundancy required

- The commercial solutions provide most of it

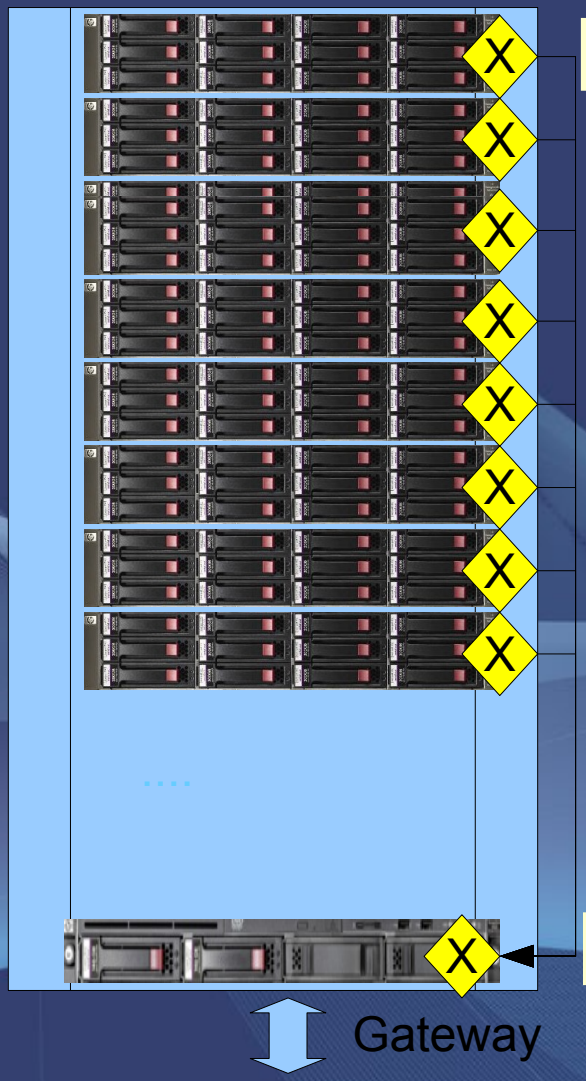
besides a single global shared namespace in a distributed GRID or Cloud environment ! [only AFS model with domain prefixes /afs/cern.ch/.... !]

XtreemFS not production ready/complete

xrootd misses redundancy and more but:

provides a global discovery network and is easy to extend!

Virtualizing Storage using xrootd



xroot server on hypervisor

- The full rack storage becomes visible to individual VMs via the xroot redirector
- The redirector can act as a proxy server
- These storage system is not (yet) virtual
  - No redundancy, no automatic life cycle management
- The application needs to support 'xroot' protocol
- There is no global namespace handling authorization meta data …
- There is no management console

xroot redirector/proxy on gateway

Virtualizing Storage using xrootd

- Which pieces are there?
- Which pieces are existing as prototypes
- Which pieces are missing? Can we get them?

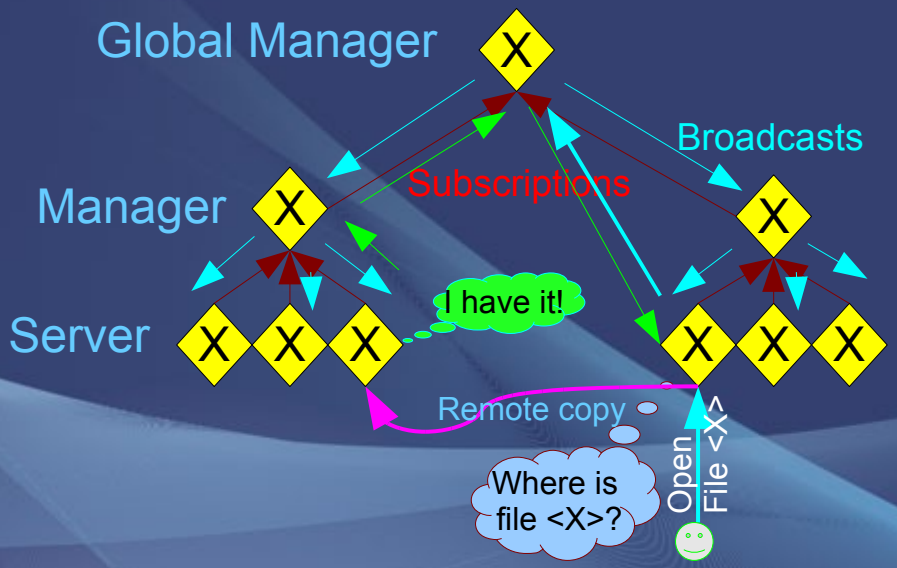
Virtualizing xrootd Storage Manager Network

- xrootd provides a global file discovery network which is arranged in a tree of manager servers caching file locations
- A manager broadcasts for missing files to the subscribed managers
- These broadcast requires a grace period in which the toplevel manager waits for responses of sub-managers (default 5s !)

Virtualizing xrootd Storage VMSS

√

Virtual Mass Storage System



- If a file is opened on a server and the file is missing
  - discover where it is
  - stage/copy it into the server
  - open the file for the client
- This requires still an external catalog knowing/defining where a file should be
  - without this policy definition movements and space usage become chaotic

Virtualizing xrootd Storage External Namespace

- The simplest way to complete xrootd as a global storage system is to add a central catalog with synchronous operations
  - ok if remote access protocol is
    used [one central WAN request per file open]
    this model is implemented by ALICE
  - difficult if namespace has to be mounted
    [high central WAN meta data request rate]

Virtualizing xrootd Storage FUSE plugin

- POSIX is understood by all applications
  - The most generic access 'protocol' to use
    - Every application can use it immedeatly
  - Have an up to date xroot client in production for LHC experiment frameworks has been difficult up-to now
  - Offers flexible extensions via extended attributes!
- FUSE prototype with write-back aggregation exists for xroot
  - excellent read performance (1GBit ov. 1GBit)
  - sufficient write performance (60 MB/s ov.1 Gbit)
    - Can also use direct IO for better write performance since used as data store and not for software executables!
  - Profits from kernel buffer cache
  - Supports krb5 & X509 authentication

Virtualizing xrootd Storage eos prototype

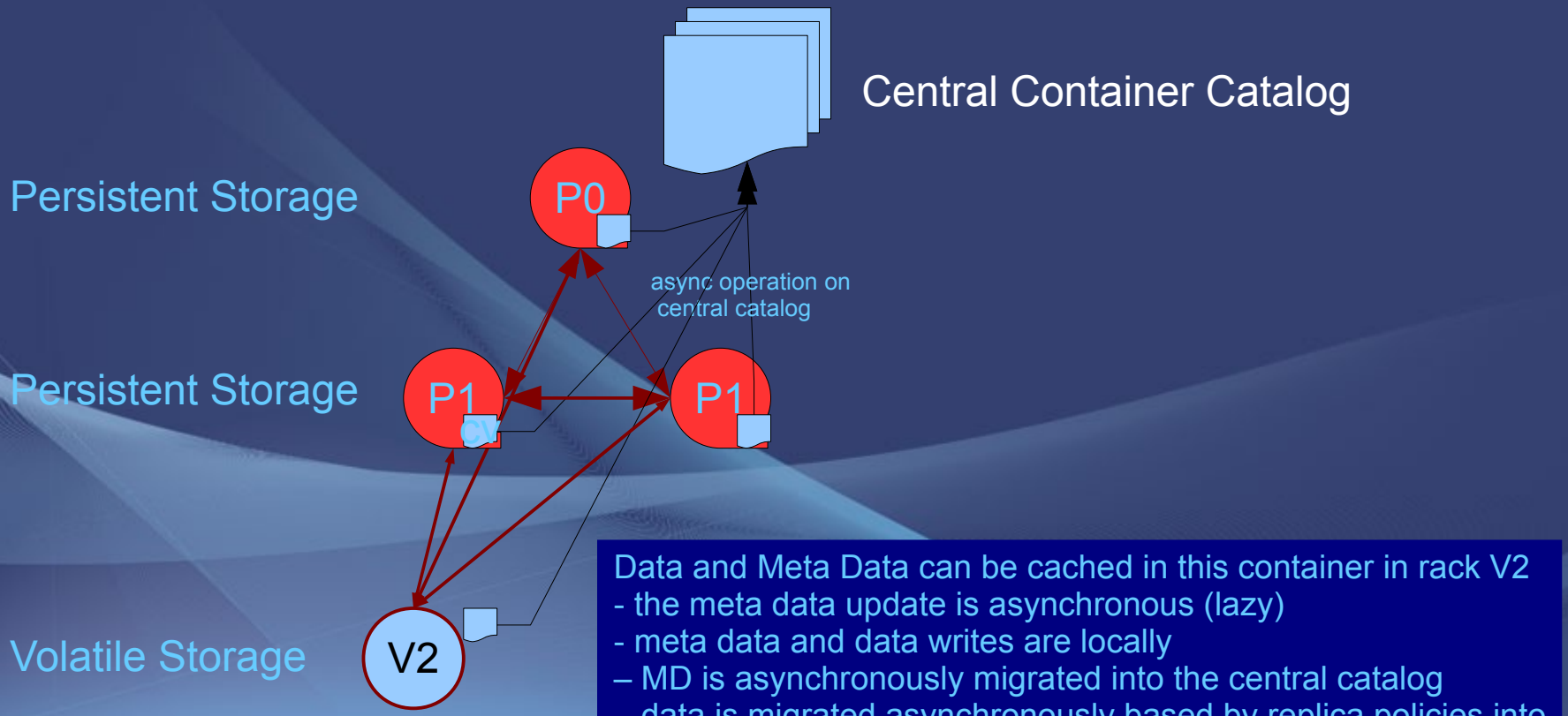
- Currently developing prototype in IT-DSS extending xroot functionality with
  - Hierarchical & non-hierarchical (in-memory cached) Namespace with virtual ID support
  - Logical Volumes providing HA & Quota
    - Network RAID 0,1,5,6 (synchronous replication)
    - Application transparent Life Cycle Management (filesystem drain, node replacement …)
  - Container based File Management
    - Define movements, placement and cache policies for file groups
  - Management Console ...

Virtualizing xrootd Storage Extending Meta Data Handling

- For a global FS view the container (directory) structure has to be kept centrally
- These can be classified as
  - read-only container
  - read-write container sync
  - read-write container async
- The container MD defines the authorization and the location policies
  - where the content is persistent
  - where the content is volatile (cacheable)

Virtualizing xrootd Storage Extending Meta Data Network

Example of an async r/w container

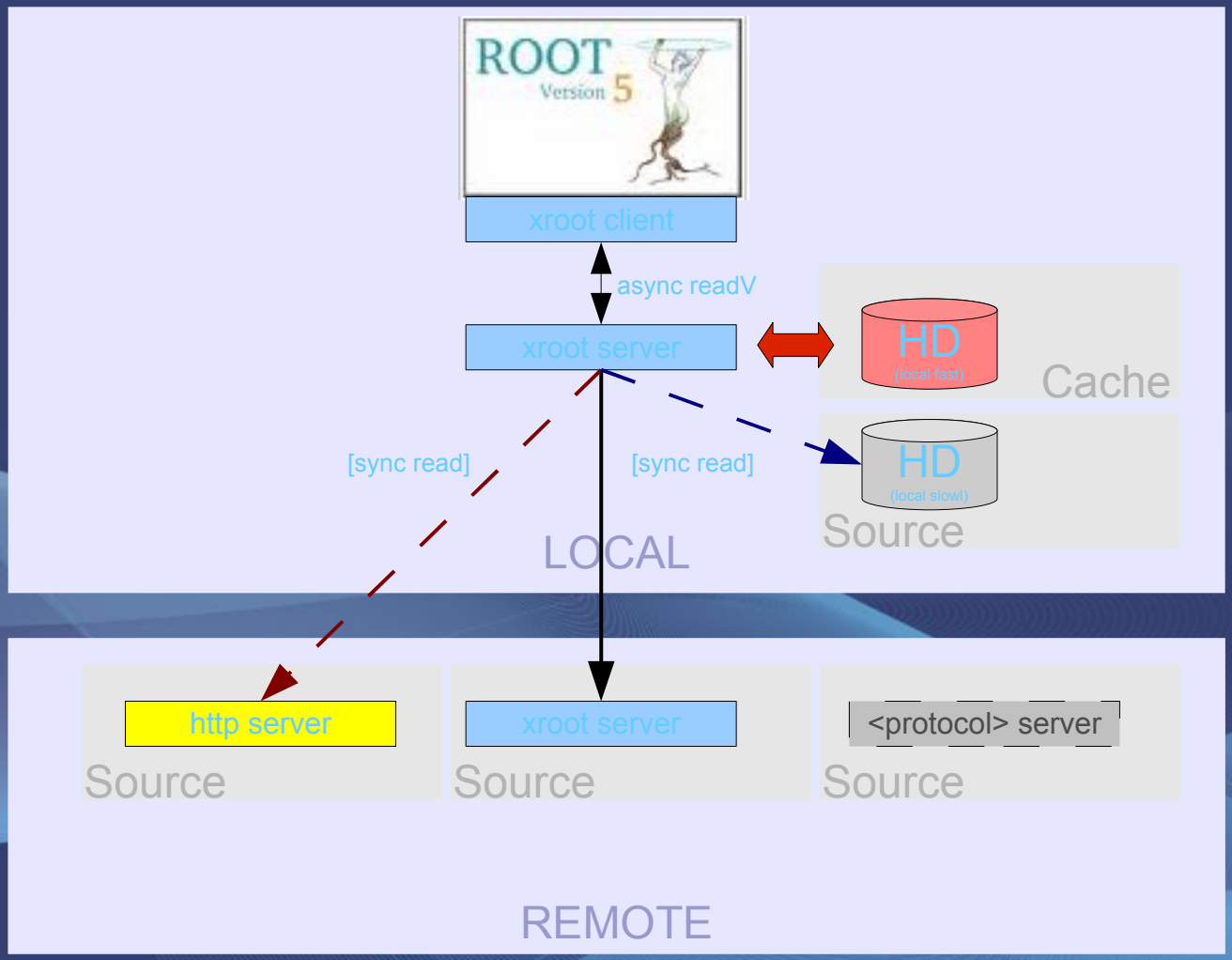


Data and Meta Data can be cached in this container in rack V2
- the meta data update is asynchronous (lazy)
- meta data and data writes are locally
– MD is asynchronously migrated into the central catalog
– data is migrated asynchronously based by replica policies into persistent storages P1 & P2
- One WAN operation to get/restore container MD
- LAN namespace operations for local mounts

Virtualizing xrootd Storage Efficient Caching

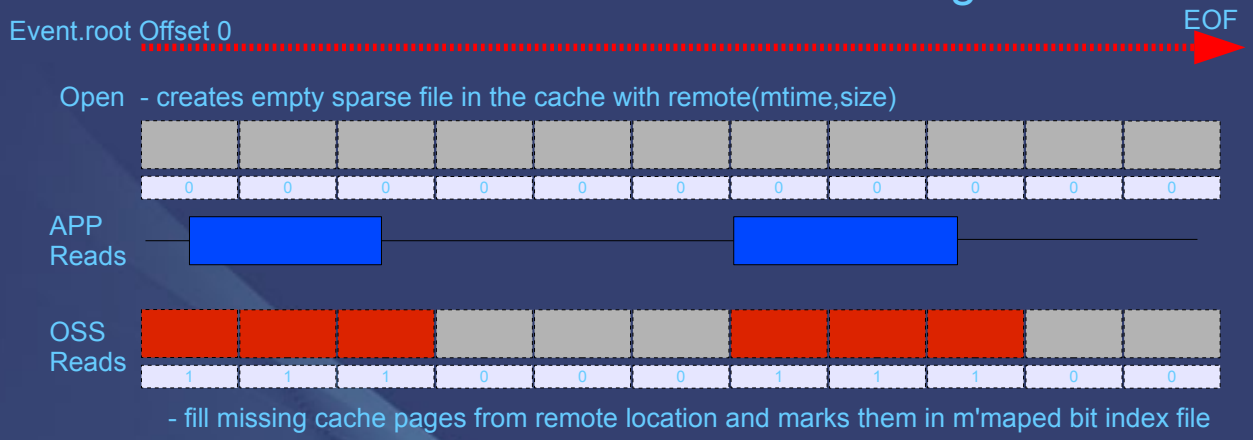
- Via FUSE kernel cache is not persistent and not shared within a storage cluster
- Like to have a shared cache built by all nodes with storage space which can cache partial files
- xroot OssCache extension

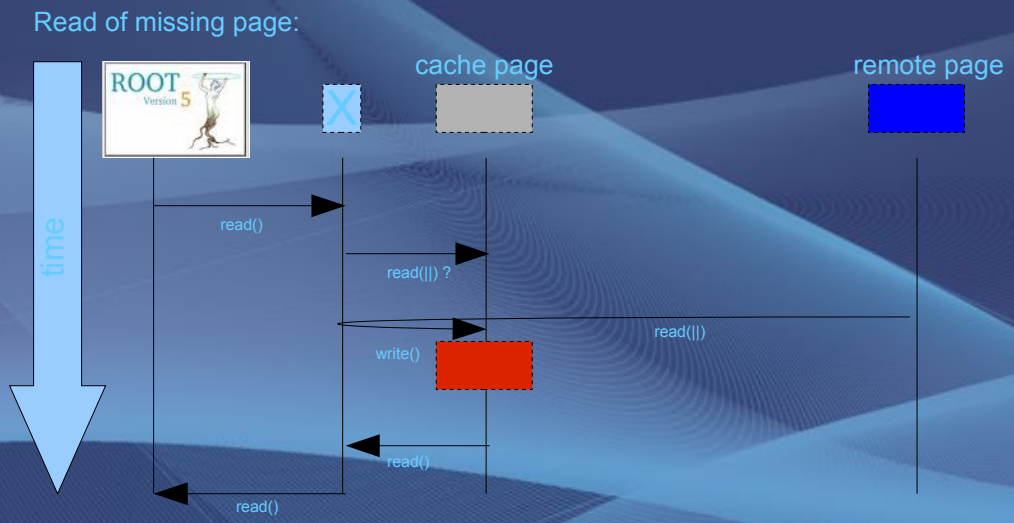
Virtualizing xrootd Storage Volatile Page Cache - OssCache



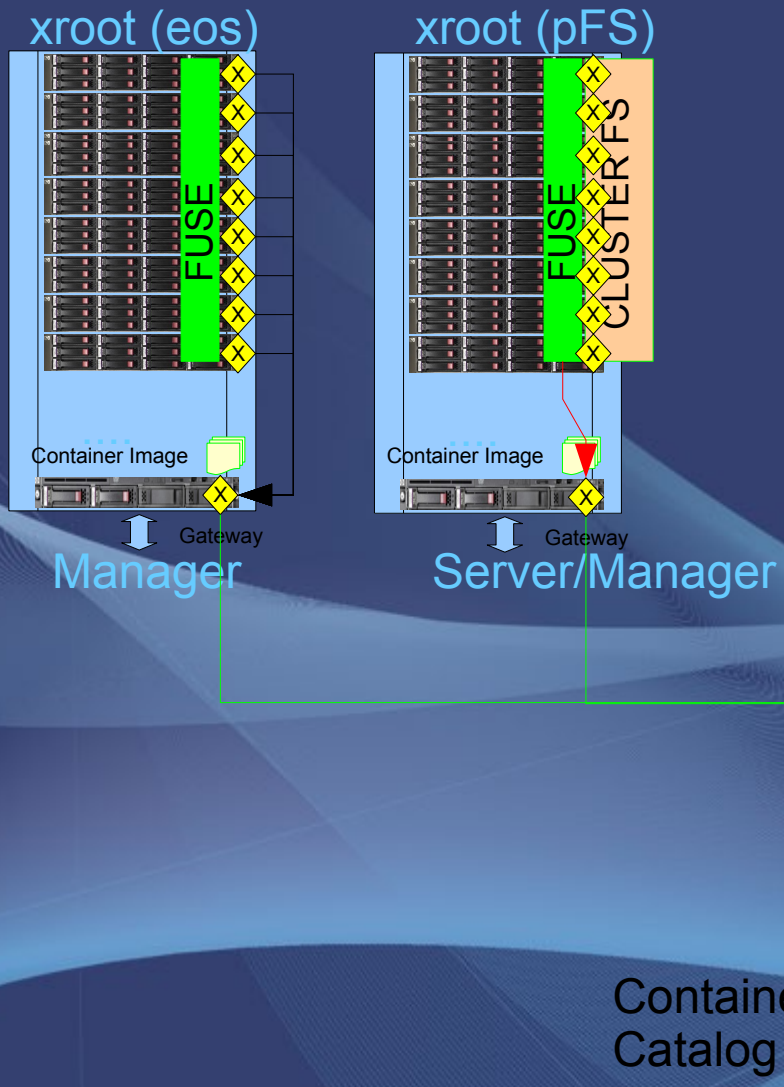
Virtualizing xrootd Storage Volatile Page Cache - OssCache

Local Client File Caching





Complete picture of a fully virtualized storage system



FUSE(xroot) layer:

- Provides global namespace via cached or global container
- Uses xroot pool or cluster FS like an object store
- Implements a persistent passive (on-deman) page cache using container policies
- Takes care of authorization as defined for each container
- Takes care of file placement and replication policies

Conclusions

- Local Storage Virtualization is commercially available (VSAN, Cluster Filesystems)
  - there are also attempts for 'globalization'
  - CERN DSS prototype for a non-commercial solution providing a virtual storage system based on xrootd (eos)
- Storage virtualization can be extended to a global scale
  - Global virtualization can be done using xrootd
    - requires to adjust few customized pieces (container catalog, VM Fuse layer)
- Proposed virtualized storage system allows
  - simple management of analysis clusters via a persistent cache layer and cache policies (no active data movements like FTS …)
  - to use any mountable storage system as local data store
  - to use standard POSIX access via global mount point
    - FUSE can be easily deployed within a virtual machine ;-) (however without mount point complete system would be simpler!)
- Subject touches LHC DM and can help to simplify DM in general!

Thank you for your attention!

Question or Comments?