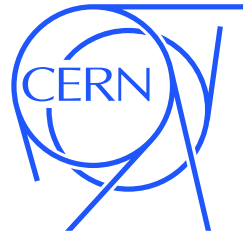


CernVM[FS] Technology for Emulation

Jakob Blomer for the CernVM Team

DPHEP Workshop on Full Costs of Curation

Base Technology: Virtual Machines

Virtual machines enable historic software environments
on today's infrastructure.

Base Technology: Virtual Machines

Virtual machines enable historic software environments
on today's infrastructure.

Add-On 1: CernVM File System

CernVM-FS is a *versioning* and *snapshotting* file system
used it to make the virtual machine's content accountable.

Overview

Base Technology: Virtual Machines

Virtual machines enable historic software environments
on today's infrastructure.

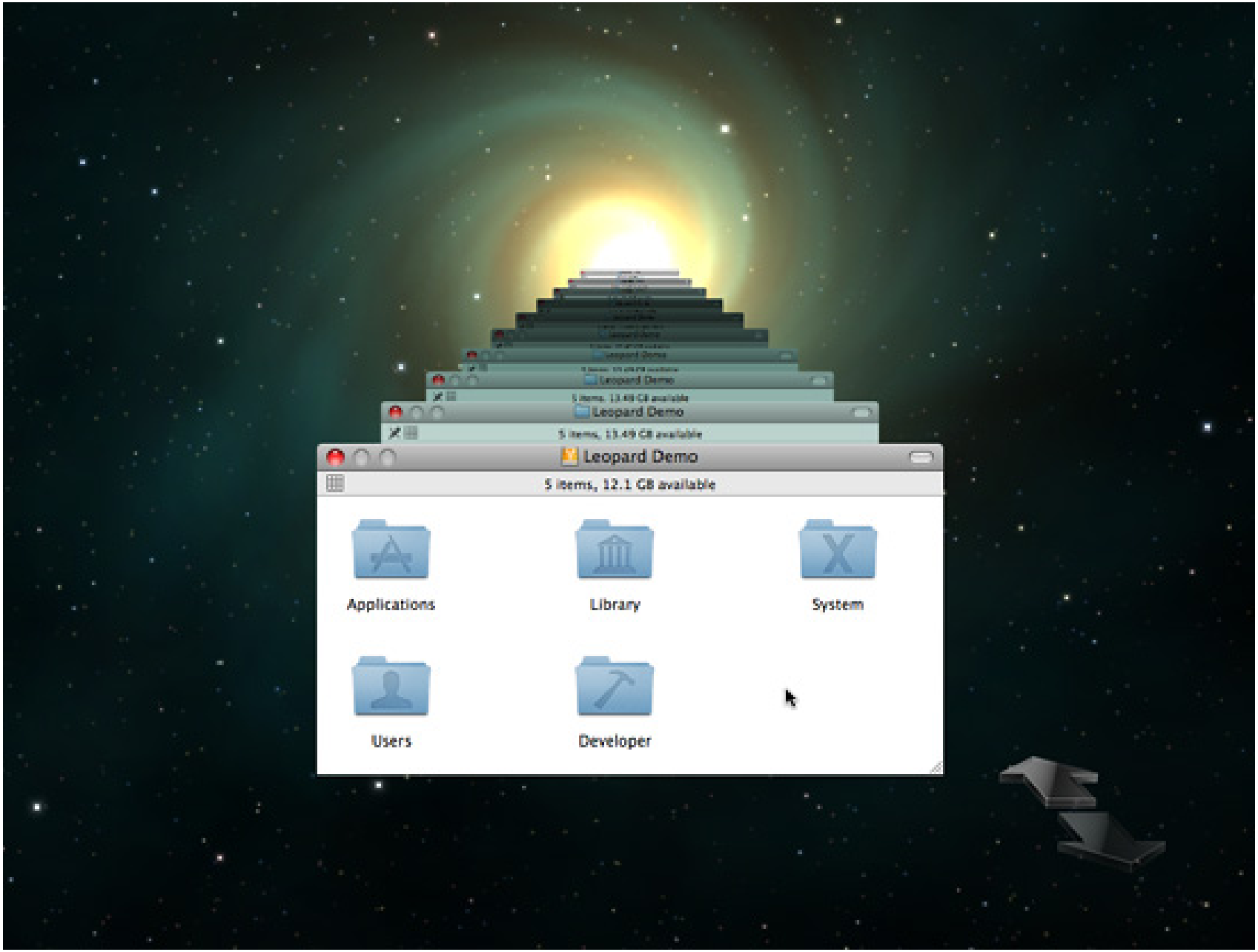
Add-On 1: CernVM File System

CernVM-FS is a *versioning* and *snapshotting* file system
used it to make the virtual machine's content accountable.

Add-On 2: CernVM Contextualization Agent

Supports textual specification for interacting CernVMs.
A *historic analysis cluster* is spawned from a single virtual machine image.

A Time Machine for the Analysis Environment



Example

- For LHCb software stored in CernVM-FS, we can go back to essentially every day until October 2010
- This capability becomes more powerful since we can associate meaningful tags with snapshots

Tag list for the CernVM internal repository:

cernvm@cernvm002:~/cernvm-rpm — ssh — 107×40

```
cernvm@cernvm002:~/cernvm-rpm [cernvm-devel]$ sudo cvmfs_server lstags cernvm-devel.cern.ch
NAME | HASH | SIZE | REVISION | TIMESTAMP | CHANNEL | DESCRIPTION
cernvm-system-3.0.0.0 | 6d193172cb140af2f8d44092487f9714d7be3535 | n/a | 170 | 9 Oct 2013 10:30:13 | 0 |
cernvm-system-3.0.1.0 | fcf6c2c74dfc9d0afc28d0640ae0fd8577cd2bf9 | n/a | 176 | 10 Oct 2013 17:57:18 | 0 |
cernvm-system-3.0.2.0 | efd6304fa538891e7d00dd0df3172846109b054a | n/a | 178 | 10 Oct 2013 18:36:17 | 0 |
cernvm-system-3.0.3.0 | f1ad2d041a28f8b6123c50fdc6c8e4da05914b4e | n/a | 180 | 10 Oct 2013 18:52:59 | 0 |
cernvm-system-3.0.4.0 | e4cdc00b286ea924e96406abf6e89f96fa1fc21a | n/a | 182 | 11 Oct 2013 17:59:00 | 0 |
cernvm-system-3.0.5.0 | 309fbd0fd3897774f7e22b08d19a2305559c9a7f | n/a | 185 | 12 Oct 2013 18:02:17 | 0 |
cernvm-system-3.0.6.0 | 811f8963b537f8a0af750b2c3bad0e92bc2fe531 | n/a | 187 | 21 Oct 2013 10:58:34 | 0 |
cernvm-system-3.0.7.0 | 54b19d80963b6a7709459570d4d0bc00096d1e17 | n/a | 189 | 22 Oct 2013 16:56:26 | 0 |
cernvm-system-3.0.8.0 | 4921730f65067b5ca7b8edea35ae40a6ce6a90c6 | n/a | 191 | 23 Oct 2013 15:26:39 | 0 |
cernvm-system-3.0.9.0 | 12f126ee6aa338b34024673184d098f3e109148c | n/a | 193 | 24 Oct 2013 19:08:49 | 0 |
cernvm-system-3.0.10.0 | eef6c40709e5b5bfef0da3decf47c90a99705ac8 | n/a | 195 | 30 Oct 2013 17:25:18 | 0 |
cernvm-system-3.0.11.0 | 87d0e8802e3e082bf16909c9858eb7fc03988a33 | n/a | 197 | 30 Oct 2013 21:47:48 | 0 |
```

Rational of Preserving the Software Environment

1. Processing of legacy data
   - Software implicitly encodes knowledge about the correct interpretation of the data
   - After substantial upgrades and modifications of the detector, the new software might lose this legacy knowledge
   - After experiment decommission, porting and validation of software is likely to end
   - Porting and validation will at some point become prohibitively expensive

Rational of Preserving the Software Environment

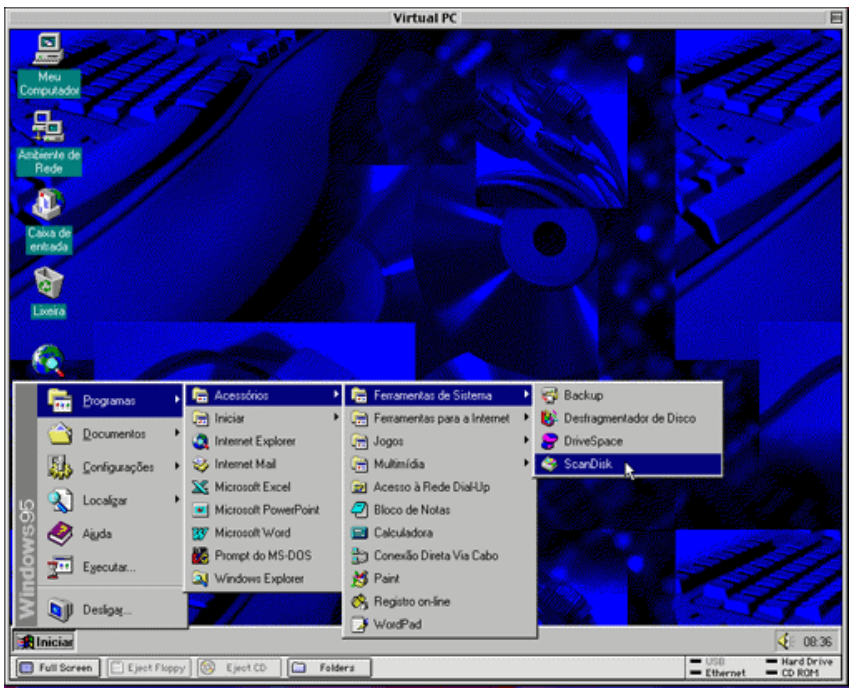
1. Processing of legacy data
   - Software implicitly encodes knowledge about the correct interpretation of the data
   - After substantial upgrades and modifications of the detector, the new software might lose this legacy knowledge
   - After experiment decommission, porting and validation of software is likely to end
   - Porting and validation will at some point become prohibitively expensive
2. Validation of new software versions
   - Otherwise, if the new software can process legacy data, comparison with historic version provides input for validation

Potential of Virtualization Technology

. . . with reasonable performance

- Very efficient on Intel architecture
- Blessed by almost *30 years* backwards compatibility
- Commodity hardware

Efficient virtualization across architectures:



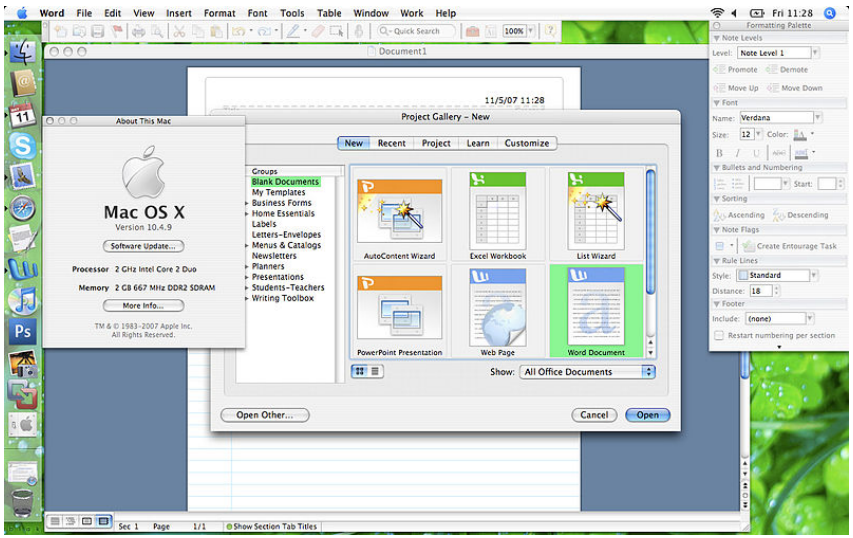
- Connectix Virtual PC ('90)
- Intel on PowerPC
- Windows, OS/2, Redhat Linux on Mac OS

Potential of Virtualization Technology

. . . with reasonable performance

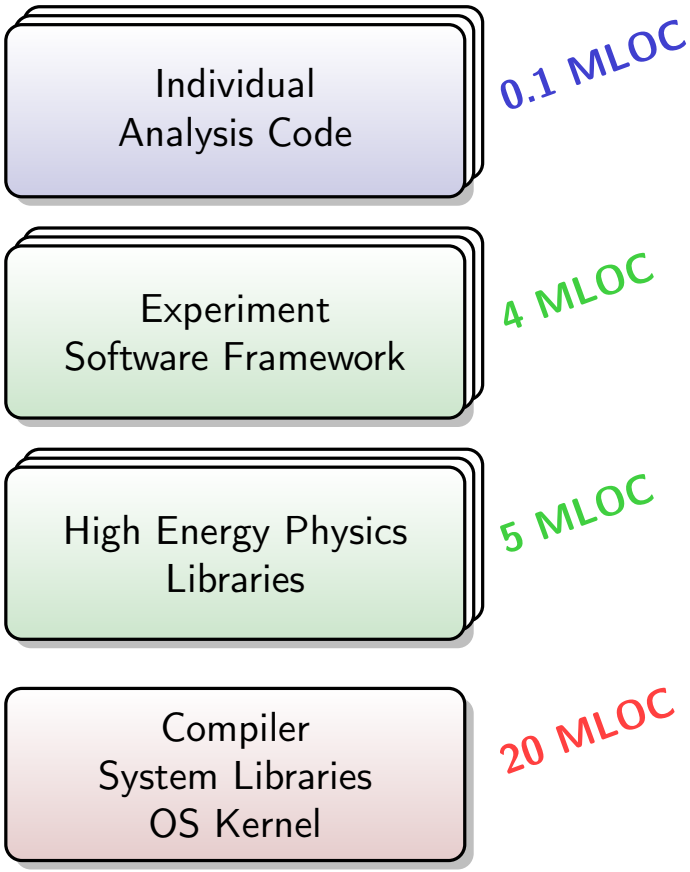
- Very efficient on Intel architecture
- Blessed by almost *30 years* backwards compatibility
- Commodity hardware

Efficient virtualization across architectures:



- Apple Rosetta (2006)
- PowerPC on Intel
- Speed: ≈50% of latest PowerPC

HEP Data Processing Environment



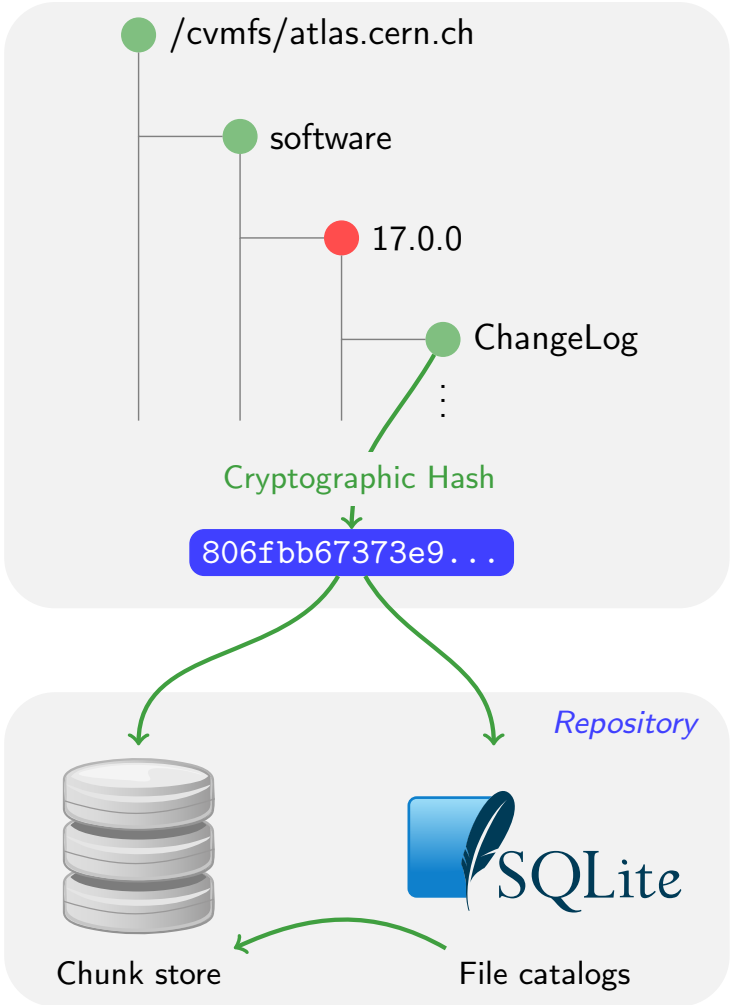
changing

stable

Amplifying

- Frequent Updates
- Not a single binary –
  a development environment
- Hundreds of libraries with
  partially untracked
  dependencies
- Not easily chunkable
- *Not easily packagable*

Versioning and Snapshots in CernVM-FS



Data Store

- Eliminates duplicates
- Never deletes, archiving

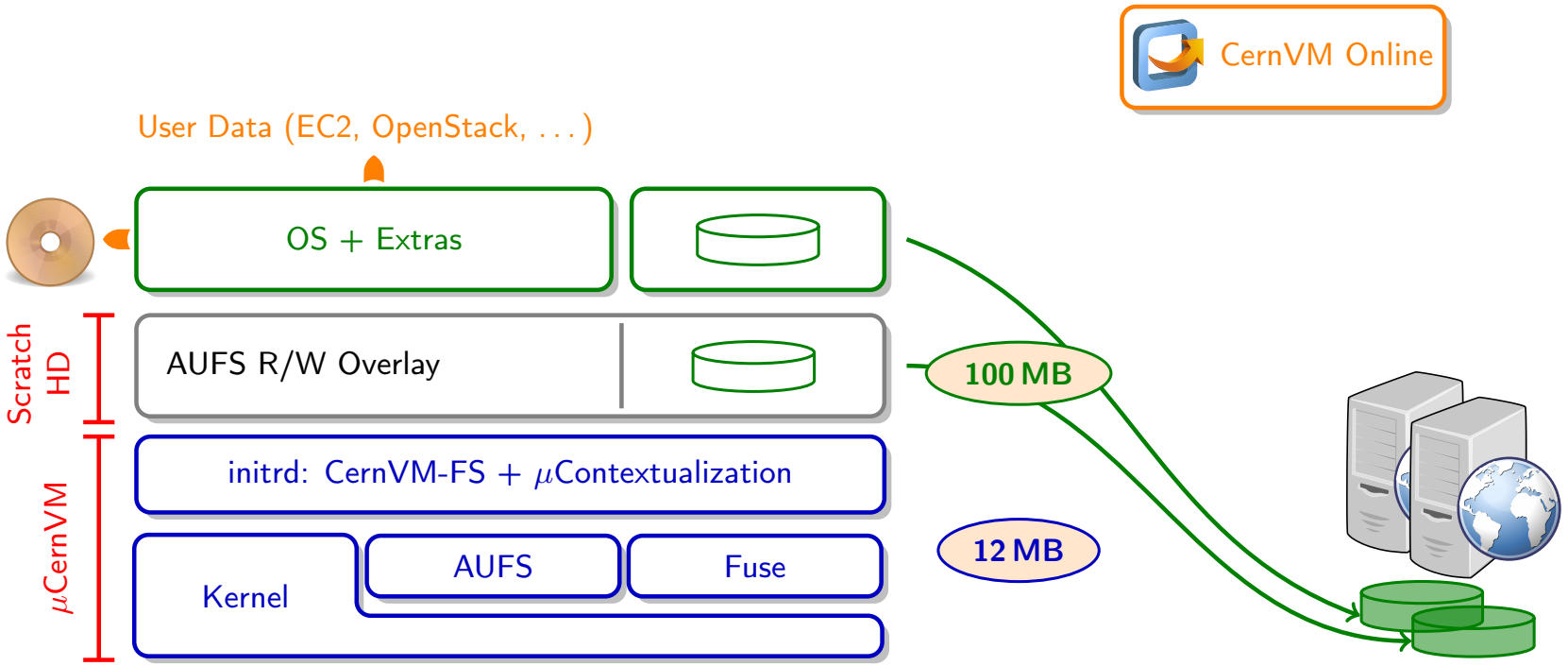
File Catalog

- Directory structure, symlinks
- Content hashes of regular files
- Digitally signed
- Plain files

The *root hash* (40 characters) defines a file system snapshot (similar to git)

Track record of LHC software and operating system

CernVM 3



Twofold system: μCernVM boot loader + OS delivered by CernVM-FS

- The very same image can be *contextualized* to run Scientific Linux 4 32bit as well as the latest Scientific Linux 6 64bit
- $\approx$ 10 years with a single image

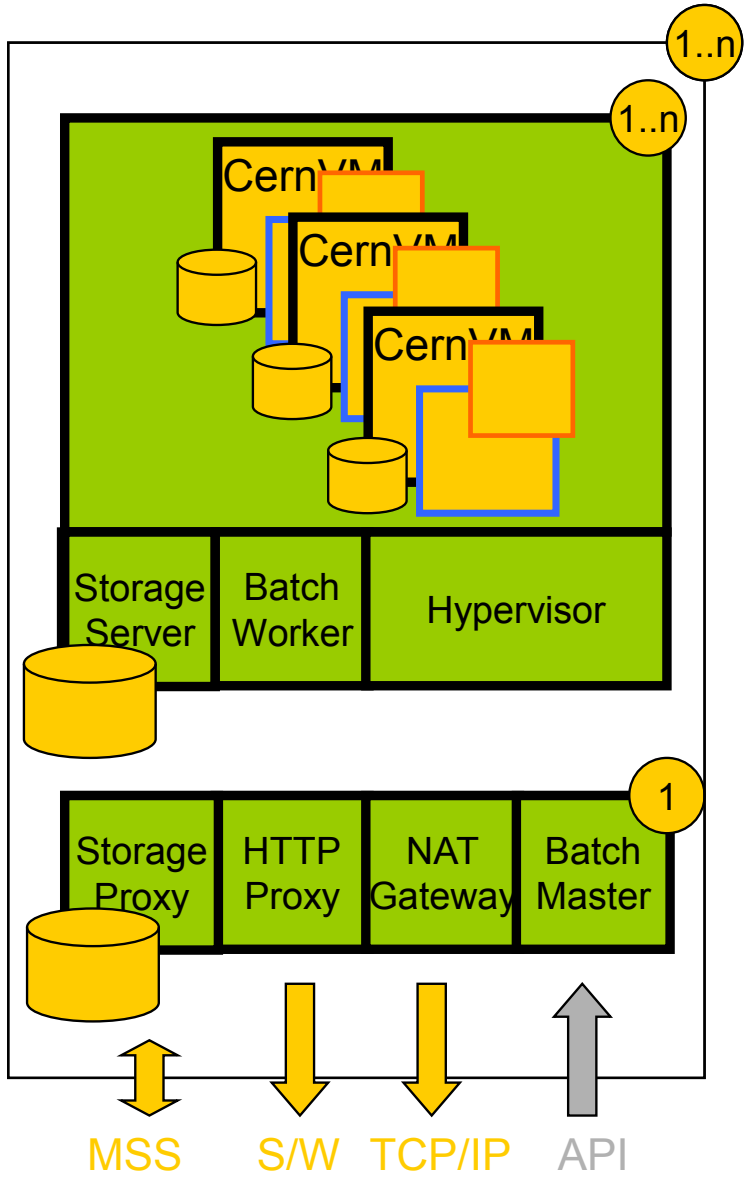
Cluster of CernVMs through Contextualization

"Context"

- Small ASCII text snippets
- Can be versioned

Connecting to Outside World

- Isolated network for security reasons
- Allows to identify links to external services
- POSIX file system interface for data access
  1. Data access protocol can evolve independently
  2. Newly created MC can be injected



Components of the CernVM Blue Print

Virtual Machine

- Linux distribution based on Scientific Linux.
- Supports all popular hypervisors.
- Minimal footprint, the VM *interface* is needed
- Flexible contextualization.

CernVM Filesystem

- Read-only, globally distributed file system optimized for software distribution.
- Based on plain files and HTTP
- Snapshotting and versioning file system
  - Already used in production by LHC experiments.

CernVM - based
data analysis environment
preservation

Bookkeeping

- CernVM-FS environment is defined by version strings. OS packages are defined by a versioned, closed package group (Meta-RPM)
- You need only the CernVM version string to rebuild CernVM image on demand.

Private Cloud

- Ensembles of CernVMs can recreate a virtual cluster for data processing.
- CernVM can be contextualized using a small subset of EC2 API that allows it to be deployed on public or private clouds

Summary

- Virtualization technology can **bridge tens of years**
- Virtualization helps us to **escape the dependency hell**
- CernVM technology provides access to a historic **data processing environment identified by a version string**
- Such virtual machines integrate well with today's cloud infrastructures
- Virtual machines are easy to deploy and can be given to "interested citizens"

Exercise: resurrecting the ALEPH environment

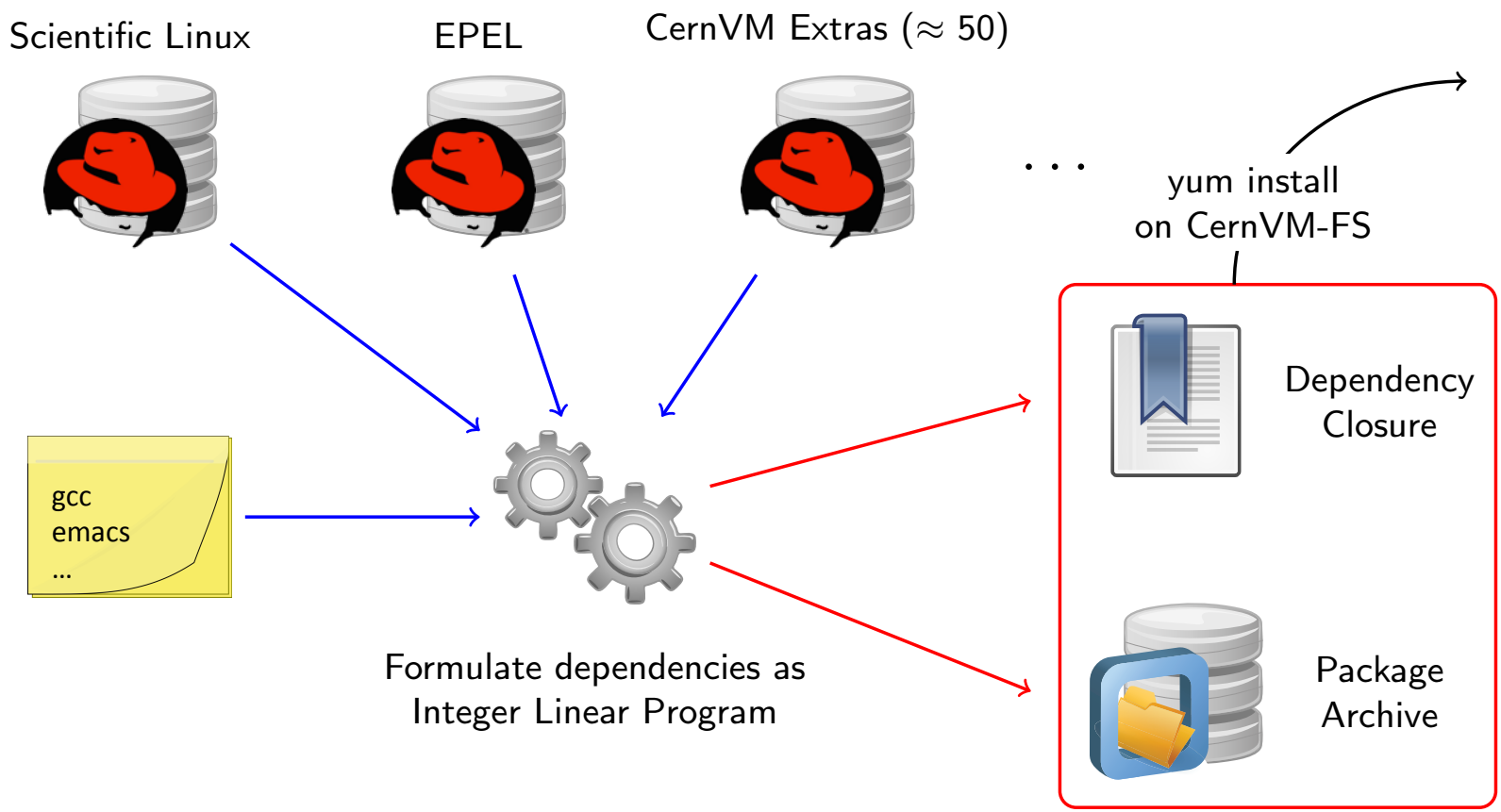
- Can we use CernVM on current CERN OpenStack infrastructure to do ALEPH physics?
- Set up Scientific Linux 4 on μCernVM $\approx$ 1 week of work
- Runs GALEPH and JULIA
- Ongoing work, potentially leads to a reference platform for LEP analysis environments

1. Backup Slides

Build Process: Scientific Linux on CernVM-FS

Maintenance of the repository **should not** become a Linux distributor's job
But: should be reproducible and well-documented

Idea: automatically generate a **fully versioned**, **closed** package list
from a "shopping list" of unversioned packages



Build Process: Package Dependency ILP

Normalized (Integer) Linear Program:

$$\text{Minimize } (c_1 \cdots c_n) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad \text{subject to } \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

Here: every available (package, version) is mapped to a $x_i \in \{0, 1\}$.
Cost vector: newer versions are cheaper than older versions.
(Obviously: less packages cheaper than more packages.)
Dependencies:
Package x_a requires x_b or x_c: $x_b + x_c - x_a \geq 0$.
Packages x_a and x_b conflict: $x_a + x_b \leq 1$.
(...)

Figures

≈17 000 available packages ($n = 17000$), 500 packages on "shopping list"
≈160 000 inequalities ($m = 160000$), solving time <10 s (`glpk`)
Meta RPM: ≈1 000 fully versioned packages, dependency closure

Idea: Mancinelli, Boender, di Cosmo, Vouillon, Durak (2006)

Hypervisor Support Status

Hypervisor / Cloud Controller	Status
VirtualBox	✓
VMware	✓
KVM	✓
Xen	✓
Microsoft HyperV	✓
Parallels	↯[4]
Openstack	✓
OpenNebula	✓[3]
Amazon EC2	✓[1]
Google Compute Engine	↯[2]

[1] Only tested with ephemeral storage, not with EBS backed instances
[2] Waiting for custom kernel support
[3] Only amiconfig contextualization
[4] Unclear license of the guest additions