CernVM – A Versatile Environment for High-Energy Physics Applications in the Cloud

Jakob Blomer

SLAC Computing Seminar
19th September 2014

1. Introduction
2. CernVM Base System
3. CernVM File System
4. Contextualization and CernVM Online
5. Use Cases

Virtualized and Cloud Resources

IaaS Clouds

- Complement batch scheduler
- CERN OpenStack: 22 K cores

High-level Trigger Farms

- ATLAS: 28 K cores
- CMS: 12 K cores

Volunteer Cloud

- 75 % computing resources and $>$ 1 trillion simulated events for CERN theory group
- Can be also seen as a "corridor grid"

Long-term data preservation

- Preserve data analysis environment
- Outreach & education
- Becomes relevant for LHC experiments

Using these resources requires virtual machine image

Virtualized and Cloud Resources

IaaS Clouds

- Complement batch scheduler
- CERN OpenStack: 22 K cores

Volunteer Cloud

- 75 % computing resources and > 1 trillion simulated events for CERN theory group
- Can be also seen as a "corridor grid"

High-level Trigger Farms

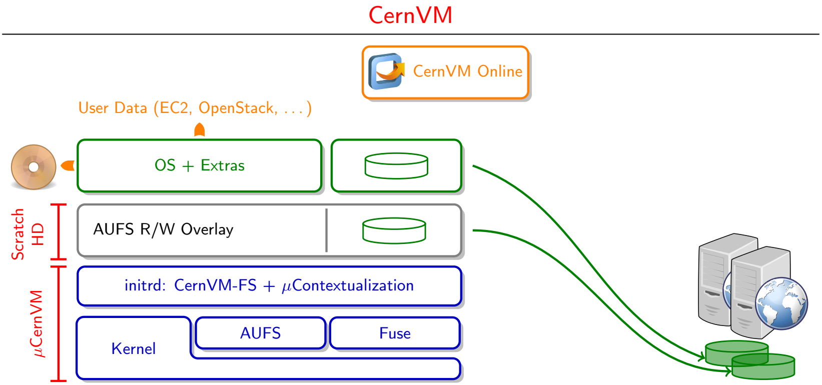
- ATLAS: 28 K cores
- CMS: 12 K cores

Long-term data preservation

- Preserve data analysis environment
- Outreach & education
- Becomes relevant for LHC experiments

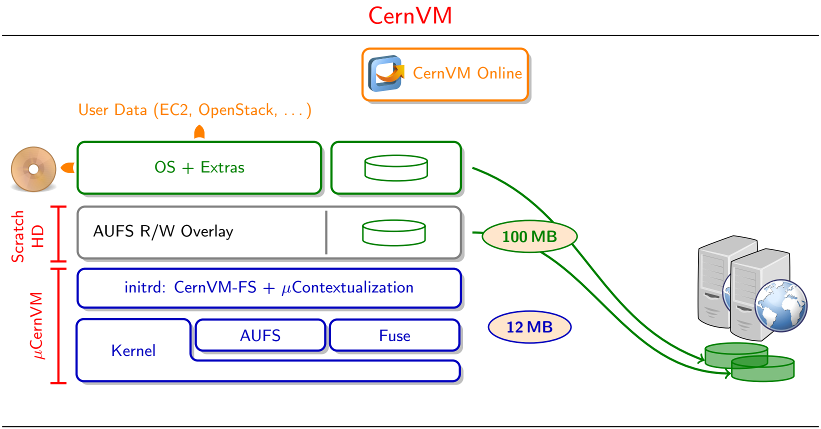
Using these resources requires virtual machine image

Building blocks of CernVM



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

Building blocks of CernVM



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

⇒ Drastic reduction in size

From "just enough operating system" to "operating system on demand"

400 MB image (compressed) ↦ 12 MB image + 100 MB cache

1. Introduction
2. CernVM Base System
3. CernVM File System
4. Contextualization and CernVM Online
5. Use Cases

μCernVM Boot Loader

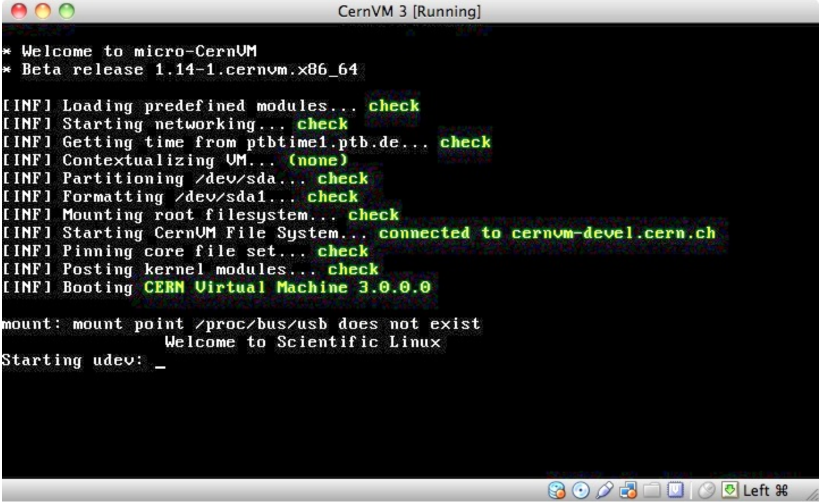
CernVM Kernel:	3.10 long-term kernel (2 year support)
Features:	KSM, zRam, THP, cgroups, X32-ABI
Extra modules:	AUFS, VMware drivers, VBox drivers, OpenAFS

"Virtualization-friendly", minimal set of device drivers:
100 modules / 8 MB as compared to 2 000 modules / 120 MB in SL6

---

1. Execute SYSLINUX boot loader
2. Decompress and load Linux kernel
3. Decompress init ramdisk, execute customized `/init`
    a) Start networking
    b) Contextualize (supports EC2, OpenStack, OpenNebula, vSphere)
    c) [Partition and] [format and] mount scratch space
    d) Mount CernVM-FS
    e) Mount AUFS root file system stack
    f) Change root file system and start operating system

Booting μCernVM



```
* Welcome to micro-CernVM
* Beta release 1.14-1.cernvm.x86_64

[INF] Loading predefined modules... check
[INF] Starting networking... check
[INF] Getting time from ptbtime1.ptb.de... check
[INF] Contextualizing VM... (none)
[INF] Partitioning /dev/sda... check
[INF] Formatting /dev/sda1... check
[INF] Mounting root filesystem... check
[INF] Starting CernVM File System... connected to cernvm-devel.cern.ch
[INF] Pinning core file set... check
[INF] Posting kernel modules... check
[INF] Booting CERN Virtual Machine 3.0.0.0

mount: mount point /proc/bus/usb does not exist
                Welcome to Scientific Linux
Starting udev: _
```

Hypervisor Support Status

Hypervisor / Cloud Controller	Status
VirtualBox	✓
VMware	✓
KVM	✓
Xen	✓
Microsoft Hyper-V	✓
Parallels	⚡[3]
Openstack	✓
OpenNebula	✓
Amazon EC2	✓[1]
Google Compute Engine	✓[2]
Microsoft Azure	?
Docker	?

[1] Only tested with ephemeral storage, not with EBS backed instances

[2] Only amiconfig contextualization

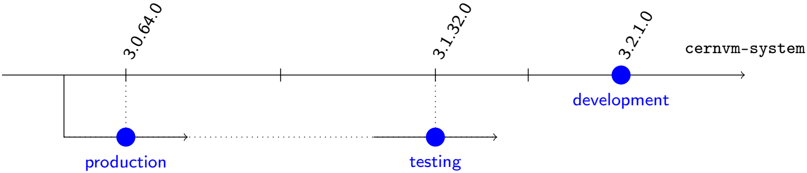
[3] Unclear license of the guest additions

Scientific Linux on CernVM-FS

General idea: Install packages with yum into a CernVM-FS chroot jail
Problem: Typical package repositories are not designed
to *preserve* an environment

The CernVM 3 build process **ensures strong versioning** on three levels

1. `cernvm-system` meta RPM
   fully versioned dependency closure
2. Named branches in the CernVM-FS repository
3. Versioned snapshots provided by CernVM-FS
   allow the very same image to instantiate any `cernvm-system` version
   helpful for **long-term data preservation**

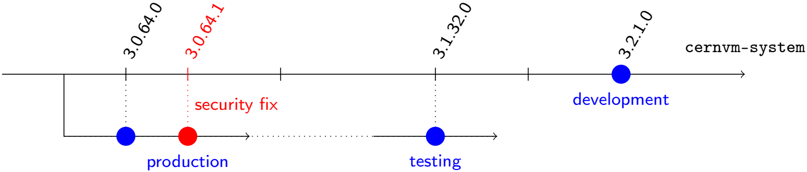


Scientific Linux on CernVM-FS

General idea: Install packages with yum into a CernVM-FS chroot jail
Problem: Typical package repositories are not designed
to *preserve* an environment

The CernVM 3 build process **ensures strong versioning** on three levels

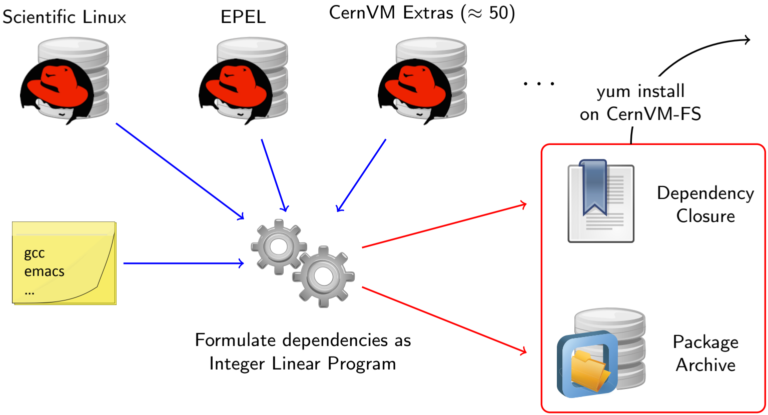
1. `cernvm-system` meta RPM
   fully versioned dependency closure
2. Named branches in the CernVM-FS repository
3. Versioned snapshots provided by CernVM-FS
   allow the very same image to instantiate any `cernvm-system` version
   helpful for **long-term data preservation**



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



Operating System Updates

Options for updating: **stay**, **diverge**, **rebase**

Rebase high-level perspective:

1. On first boot, CernVM selects and pins newest available version
2. Automatic update notifications
3. Applying updates requires a reboot
   Most security critical updates require a reboot anyway

μCernVM Bootloader

- boot partition read-only, updates dropped on ephemeral storage
- 2 phase boot: start old kernel and ramdisk, **kexec** into updated version

CernVM-FS OS Repository

- Mount updated CernVM-FS snapshot
- Conflict resolution wrt. local changes
  1. keep local configuration
  2. map user/group ids
  3. merge rpm database

Operating System Updates

Options for updating: **stay**, **diverge**, **rebase**

Rebase high-level perspective:

1. On first boot, CernVM selects and pins newest available version
2. Automatic update notifications
3. Applying updates requires a reboot
   Most security critical updates require a reboot anyway

μCernVM Bootloader

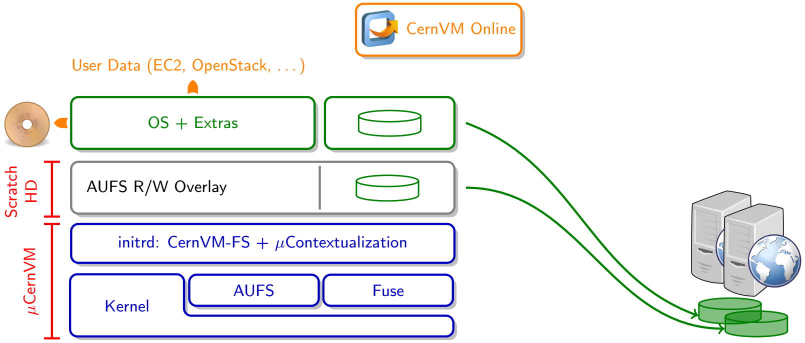
- boot partition read-only,
  updates dropped
  on ephemeral storage
- 2 phase boot:
  start old kernel and ramdisk,
  kexec into updated version

CernVM-FS OS Repository

- Mount updated CernVM-FS snapshot
- Conflict resolution wrt.
  local changes
  1. keep local configuration
  2. map user/group ids
  3. merge rpm database

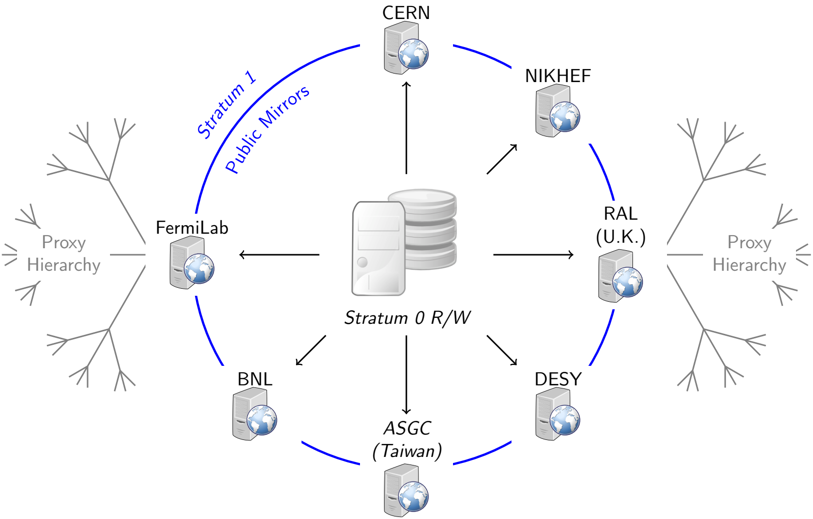
1. Introduction
2. CernVM Base System
3. **CernVM File System**
4. Contextualization and CernVM Online
5. Use Cases

CernVM File System


CernVM Online
User Data (EC2, OpenStack, . . .)
OS + Extras
Scratch HD
AUFS R/W Overlay
μCernVM
initrd: CernVM-FS + μContextualization
Kernel
AUFS
Fuse

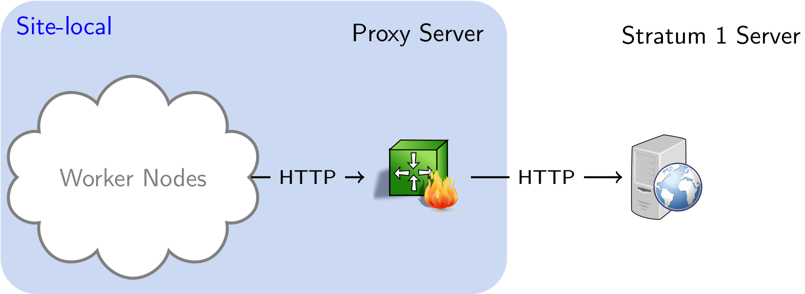
- Independent *repositories*, e. g. /cvmfs/atlas.cern.ch
- Single point of publishing
- HTTP Transport, access and caching on demand
- Has a life outside the CernVM virtual machine

CernVM-FS Content Distribution



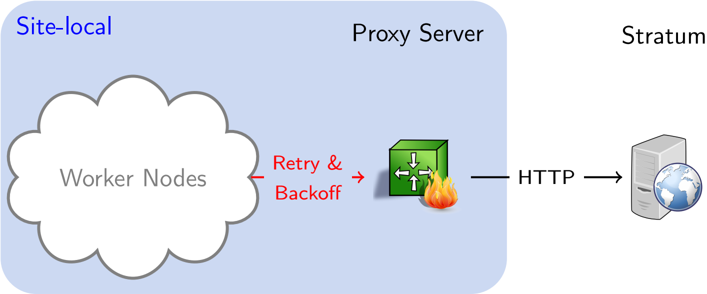
High-Availability by Horizontal Scaling

Server side: stateless services



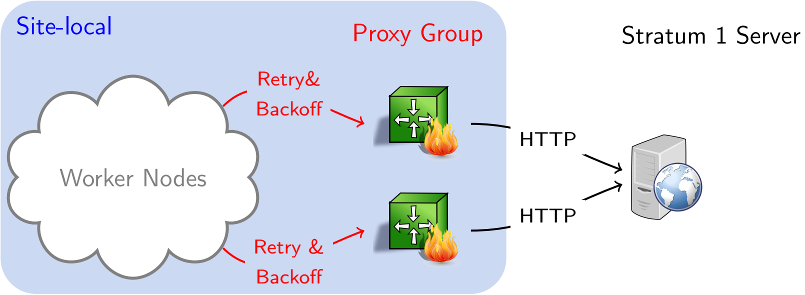
High-Availability by Horizontal Scaling

Server side: stateless services



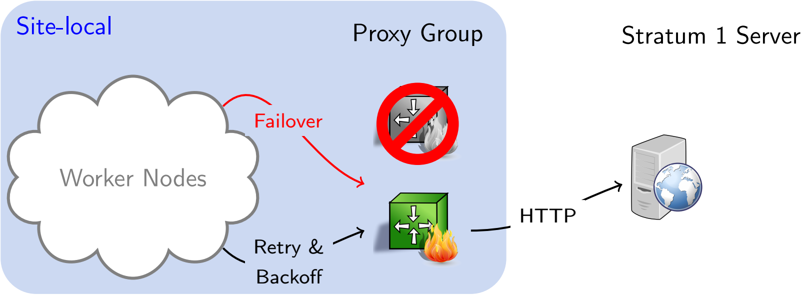
High-Availability by Horizontal Scaling

Server side: stateless services



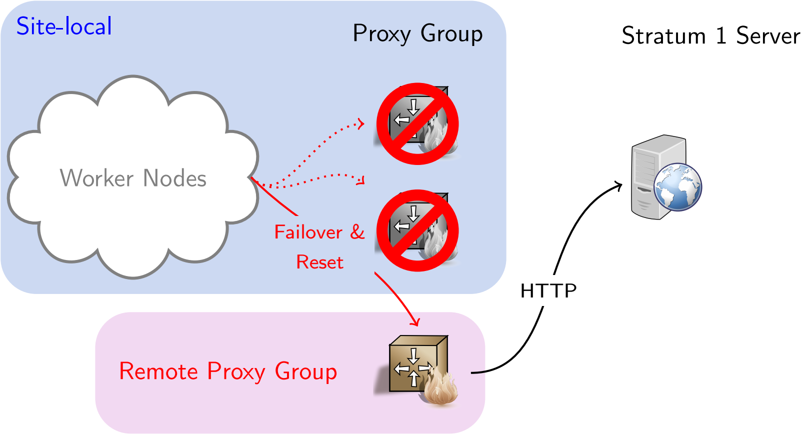
High-Availability by Horizontal Scaling

Server side: stateless services



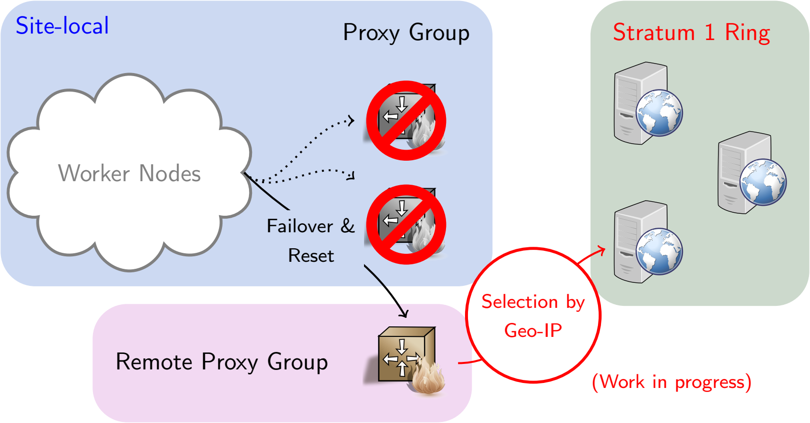
High-Availability by Horizontal Scaling

Server side: stateless services



High-Availability by Horizontal Scaling

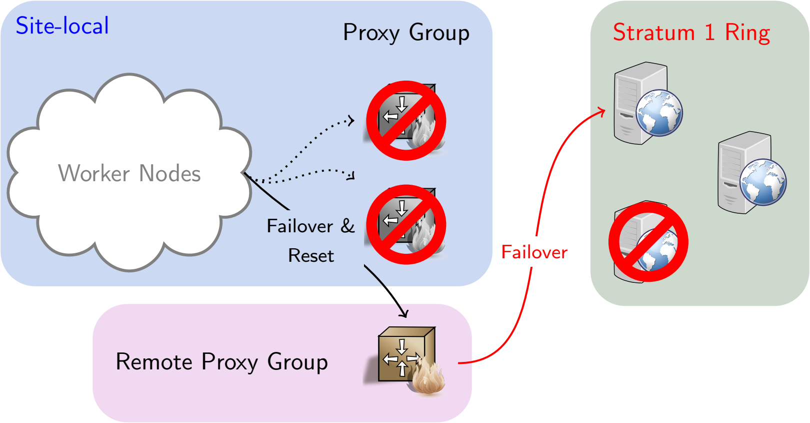
Server side: stateless services



(Work in progress)

High-Availability by Horizontal Scaling

Server side: stateless services



Site-local

Proxy Group

Worker Nodes

Failover & Reset

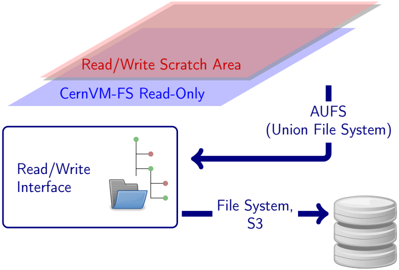
Remote Proxy Group

Stratum 1 Ring

Failover

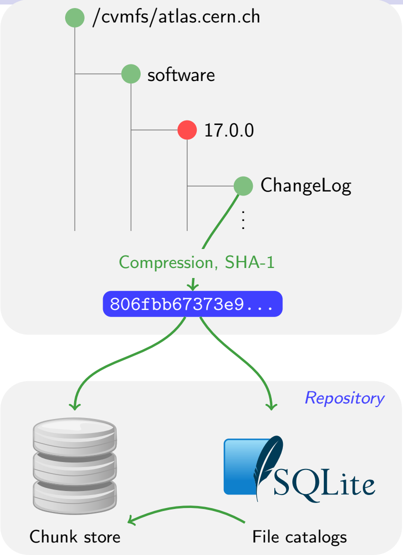
The Server: A Transactional Publish Interface

- Kernel-level Union File System AUFS
- $< 5\,\%$ performance loss (untar)



- Fully POSIX-compliant read-write file system
- Encapsulated change set in scratch area
- In contrast to file-wise write:
  batch publishing of consistent snapshots

Content-Addressable Storage in CernVM-FS



Data Store

- Compressed chunks (files)
- Eliminates duplicates

File Catalog

- Directory structure, symlinks
- Content hashes of regular files
- Digitally signed
  ⇒integrity, authenticity
- Time to live
- Partitioned / Merkle hashes
  (user assisted)

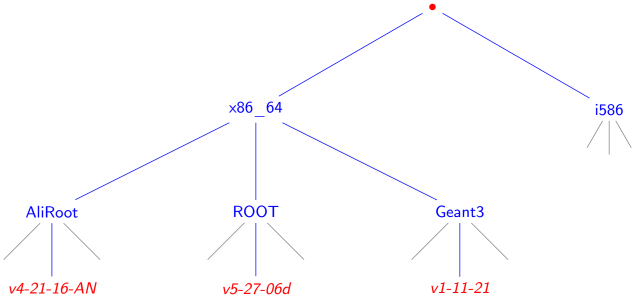
⇒ Immutable files, trivial to check for corruption, consistency by snapshots

≈ 6× reduction in number of files / volume

Partitioning of Meta-Data

Automatic Approaches

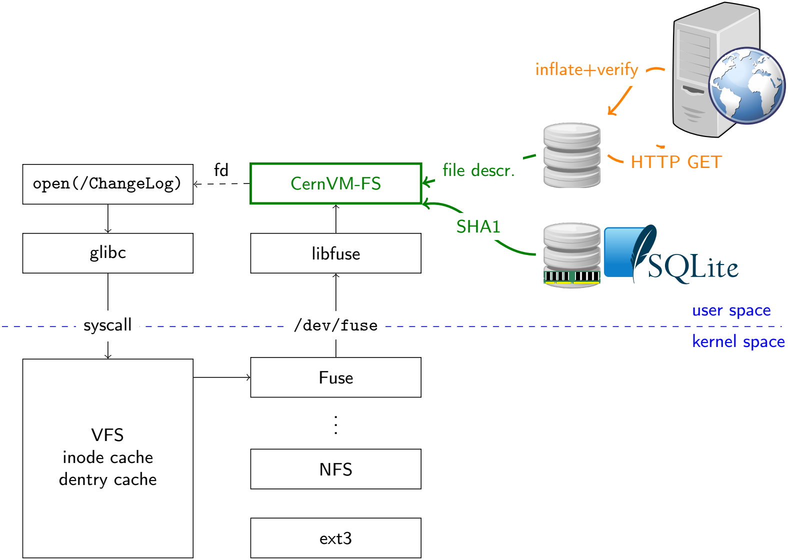
File based (Tolia et al. 2004), directory based (Kutzner 2008), global (Compostella et al. 2010)



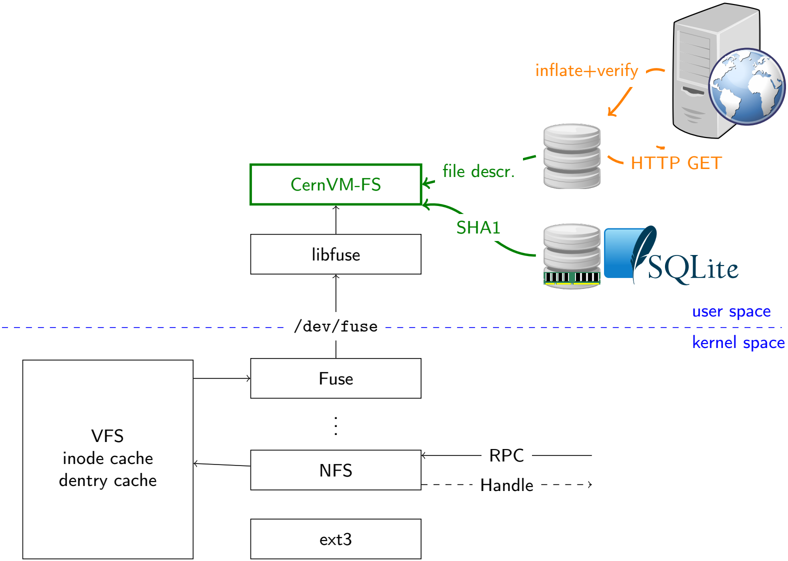
CernVM-FS: Semi-automatic, use of human knowledge about

- locality by software version
- locality by frequency of changes

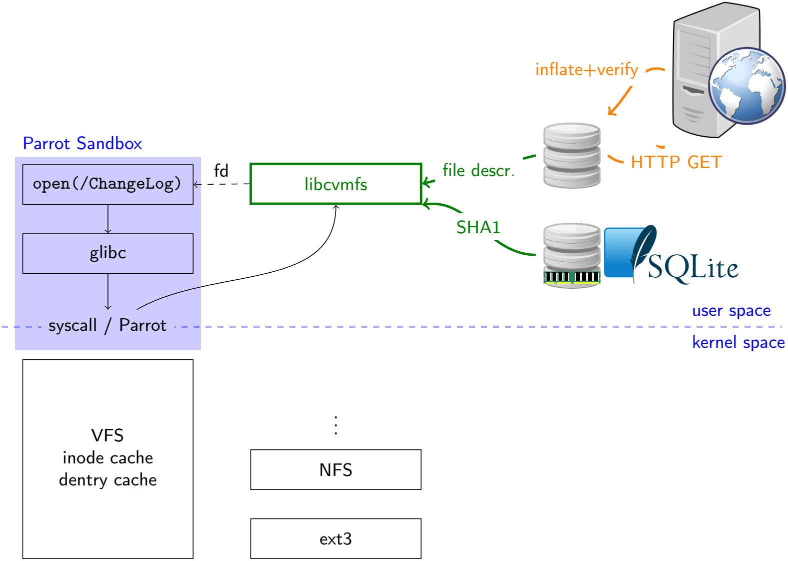
The Client: Fuse Module



The Client: NFS Exported Fuse Module



The Client: File System Option of Parrot



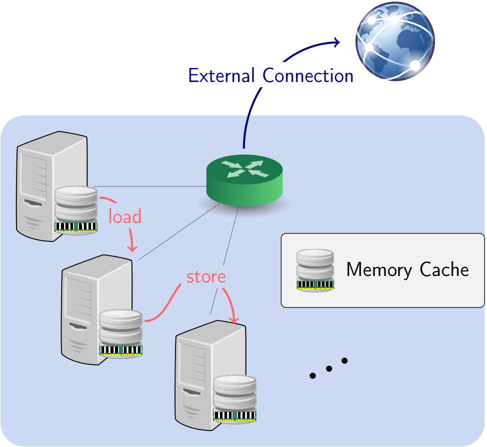
http://ccl.cse.nd.edu/software/parrot

Cooperative Cluster Cache

Early work in progress

Goal Reduce cache duplication, local hard disk consumption

Environment Super computers, diskless clusters



https://ramcloud.stanford.edu

RAMCloud: distributed key-value store in DRAM

Prototype based on RAMCloud

1. SQlite backend
2. Put/Get interface for CernVM-FS chunk storage

Deployment Configurations

1. As part of CernVM
   - Fuse module, auto-configured
2. As Fuse module on the worker node / workstation
   - Almost all of WLCG using it this way
   - Very little problems
   - "Private mounts" as normal user possible
3. NFS-exported Fuse module
   - Brings back infamous NFS bottleneck
   - Only solution for disk-less farms
   - DESY: 2 k nodes on CernVM-FS / NFS
4. As part of the Grid job
   - Using library interface + Parrot connector
   - Performance limitations, limited cache sharing
   - Runs as normal user

Web proxies: Site-local, *Frontier* (CMS, ATLAS), Public Service

1. Introduction
2. CernVM Base System
3. CernVM File System
4. Contextualization and CernVM Online
5. Use Cases

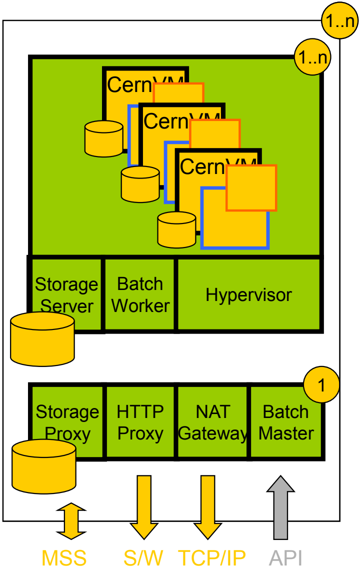
Contextualization

Transfer and Conventions

- Payload in attached device (CD, floppy)
- Payload on web server (fixed IP, gateway)
- Script or “user data” string
- Interpreted by VM agent (e. g. cloud-init, amiconfig)

Tasks

- Credentials (ssh, X.509)
- Condor head & batch services
- Squid server, XrootD proxy
- Network configuration & tuning



Source: Buncic

Sample Context

user-data.txt

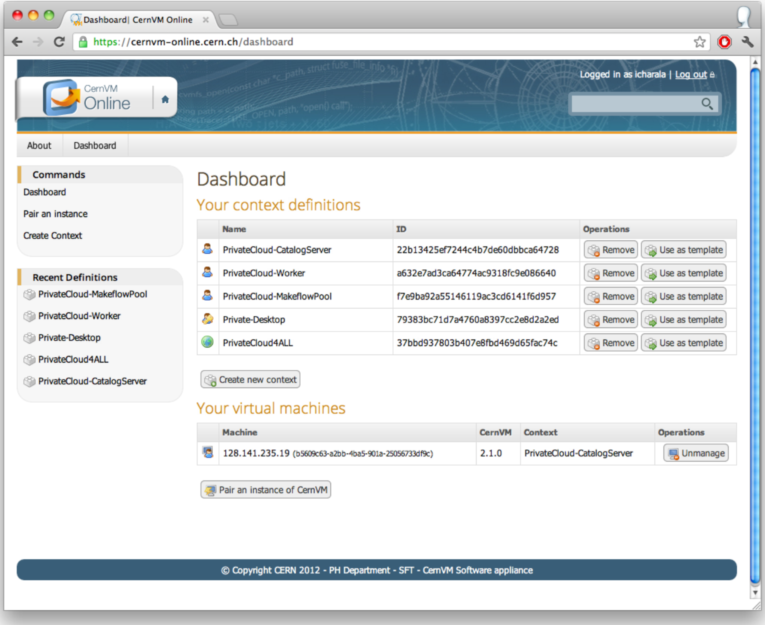
```
[cernvm]
organisations=ALICE
repositories=alice,alice-ocdb,sft
shell=/bin/bash
config_url=http://cernvm.cern.ch/config
users=alice:alice:ion
edition=Desktop
keyboard=us
startXDM=on
auto_login=on

[ucernvm-begin]
cvmfs_tag=cernvm-system-3.1.1.4
[ucernvm-end]
```

Boot on CERN OpenStack

```
nova boot AliceVM -image "cvm3" -flavor m1.small \
  -key-name ssh-key -user-data user-data.txt
```

CernVM Online: Context Bookkeeping & Pairing


Dashboard| CernVM Online
https://cernvm-online.cern.ch/dashboard
CernVM Online
Logged in as icharala | Log out
About
Dashboard

Commands

Dashboard

Pair an instance

Create Context

Recent Definitions

PrivateCloud-MakeflowPool

PrivateCloud-Worker

Private-Desktop

PrivateCloud4ALL

PrivateCloud-CatalogServer

Dashboard

Your context definitions

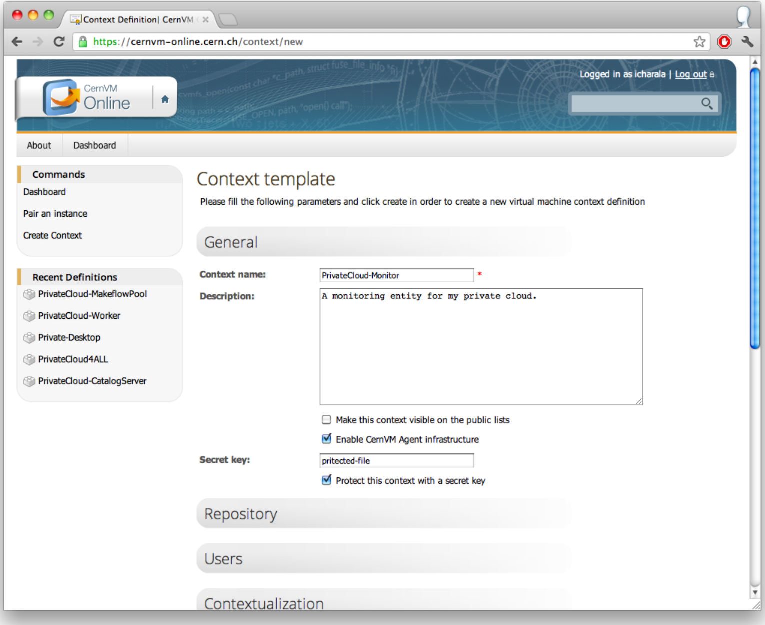
	Name	ID	Operations
	PrivateCloud-CatalogServer	22b13425ef7244c4b7de60dbbca64728	Remove Use as template
	PrivateCloud-Worker	a632e7ad3ca64774ac9318fc9e086640	Remove Use as template
	PrivateCloud-MakeflowPool	f7e9ba92a55146119ac3cd6141f6d957	Remove Use as template
	Private-Desktop	79383bc71d7a4760a8397cc2e8d2a2ed	Remove Use as template
	PrivateCloud4ALL	37bbd937803b407e8fbd469d65fac74c	Remove Use as template

Your virtual machines

	Machine	CernVM	Context	Operations
	128.141.235.19 (b5609c63-a2bb-4ba5-901a-25056733df9c)	2.1.0	PrivateCloud-CatalogServer	Unmanage

© Copyright CERN 2012 - PH Department - SFT - CernVM Software appliance

CernVM Online: Context Bookkeeping & Pairing



Context Definition| CernVM
https://cernvm-online.cern.ch/context/new
CernVM Online
Logged in as icharala | Log out
About
Dashboard

Commands

- Dashboard
- Pair an instance
- Create Context

Recent Definitions

- PrivateCloud-MakeflowPool
- PrivateCloud-Worker
- Private-Desktop
- PrivateCloud4ALL
- PrivateCloud-CatalogServer

Context template

Please fill the following parameters and click create in order to create a new virtual machine context definition

General

Context name: PrivateCloud-Monitor *

Description: A monitoring entity for my private cloud.

□ Make this context visible on the public lists

☑ Enable CernVM Agent infrastructure

Secret key: pritected-file

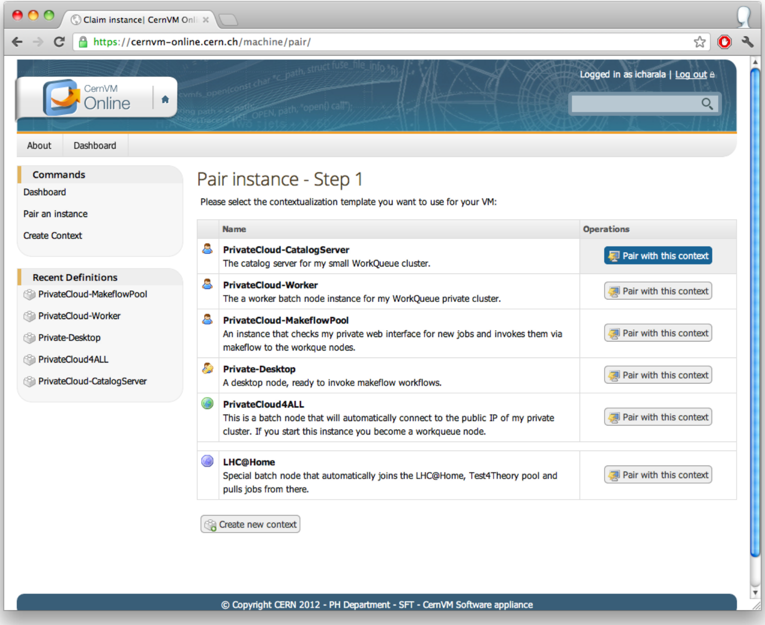
☑ Protect this context with a secret key

Repository

Users

Contextualization

CernVM Online: Context Bookkeeping & Pairing



Claim instance| CernVM Onl
https://cernvm-online.cern.ch/machine/pair/
CernVM Online
Logged in as icharala | Log out
About
Dashboard

Commands

- Dashboard
- Pair an instance
- Create Context

Recent Definitions

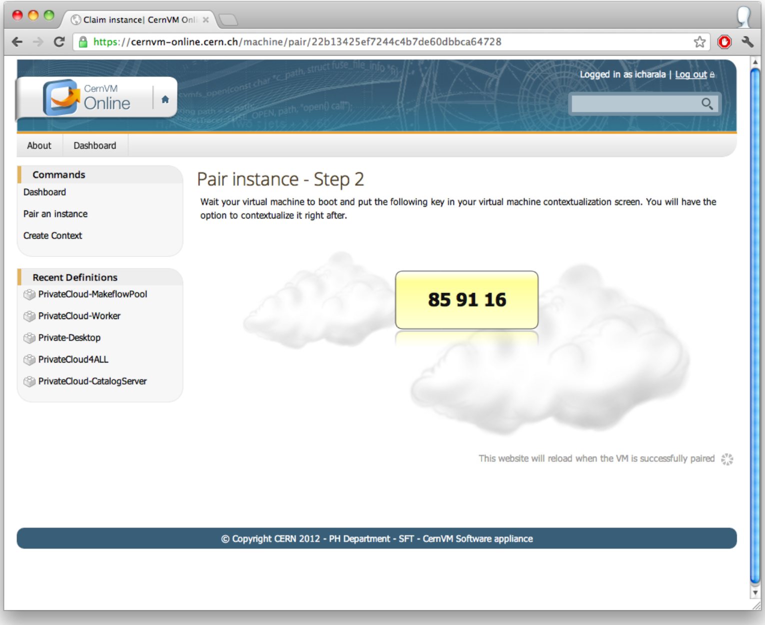
- PrivateCloud-MakeflowPool
- PrivateCloud-Worker
- Private-Desktop
- PrivateCloud4ALL
- PrivateCloud-CatalogServer

Pair instance - Step 1

Please select the contextualization template you want to use for your VM:

Name	Operations
PrivateCloud-CatalogServer The catalog server for my small WorkQueue cluster.	Pair with this context
PrivateCloud-Worker The a worker batch node instance for my WorkQueue private cluster.	Pair with this context
PrivateCloud-MakeflowPool An instance that checks my private web interface for new jobs and invokes them via makeflow to the workque nodes.	Pair with this context
Private-Desktop A desktop node, ready to invoke makeflow workflows.	Pair with this context
PrivateCloud4ALL This is a batch node that will automatically connect to the public IP of my private cluster. If you start this instance you become a workqueue node.	Pair with this context
LHC@Home Special batch node that automatically joins the LHC@Home, Test4Theory pool and pulls jobs from there.	Pair with this context

CernVM Online: Context Bookkeeping & Pairing



Claim instance| CernVM Onl
https://cernvm-online.cern.ch/machine/pair/22b13425ef7244c4b7de60dbbca64728
CernVM Online
Logged in as icharala | Log out
About
Dashboard

Commands

- Dashboard
- Pair an instance
- Create Context

Recent Definitions

- PrivateCloud-MakeflowPool
- PrivateCloud-Worker
- Private-Desktop
- PrivateCloud4ALL
- PrivateCloud-CatalogServer

Pair instance - Step 2

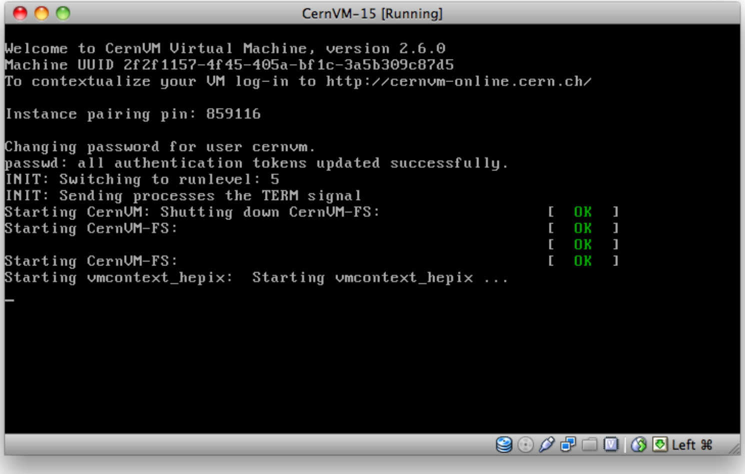
Wait your virtual machine to boot and put the following key in your virtual machine contextualization screen. You will have the option to contextualize it right after.

85 91 16

This website will reload when the VM is successfully paired

© Copyright CERN 2012 - PH Department - SFT - CernVM Software appliance

CernVM Online: Context Bookkeeping & Pairing



```
Welcome to CernVM Virtual Machine, version 2.6.0
Machine UUID 2f2f1157-4f45-405a-bf1c-3a5b309c87d5
To contextualize your VM log-in to http://cernvm-online.cern.ch/

Instance pairing pin: 859116

Changing password for user cernvm.
passwd: all authentication tokens updated successfully.
INIT: Switching to runlevel: 5
INIT: Sending processes the TERM signal
Starting CernVM: Shutting down CernVM-FS:                    [  OK  ]
Starting CernVM-FS:                                          [  OK  ]
                                                             [  OK  ]
Starting CernVM-FS:                                          [  OK  ]
Starting vmcontext_hepix:  Starting vmcontext_hepix ...
_
```

1. Introduction
2. CernVM Base System
3. CernVM File System
4. Contextualization and CernVM Online
5. **Use Cases**

Volunteer Computing

- Commonly known as …@Home projects
- Computing on *spare* resources of "interested citizens"
- Outreach program
- Opportunistic, volatile resources
- Big projects comparable to a TOP500 supercomputer
- LHC with 2 projects: LHC@Home & LHC@home 2.0

SETI@Home

- Search for extraterrestrial life
- 130 000 active participants
- 630 Tera-FLOPS

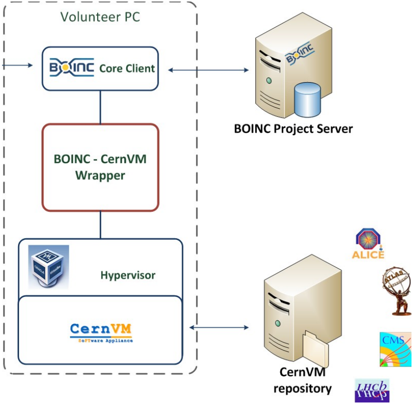
Einstein@Home

- Search for gravitational waves
- 34 000 active participants
- 470 Tera-FLOPS

Volunteer Computing: LHC@Home 2.0

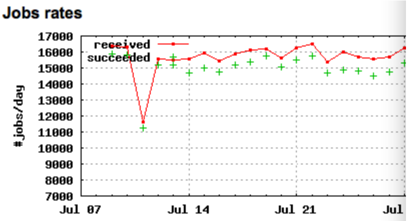
Monte-Carlo simulations, parameter tuning

First BOINC project using virtual machines



Numbers

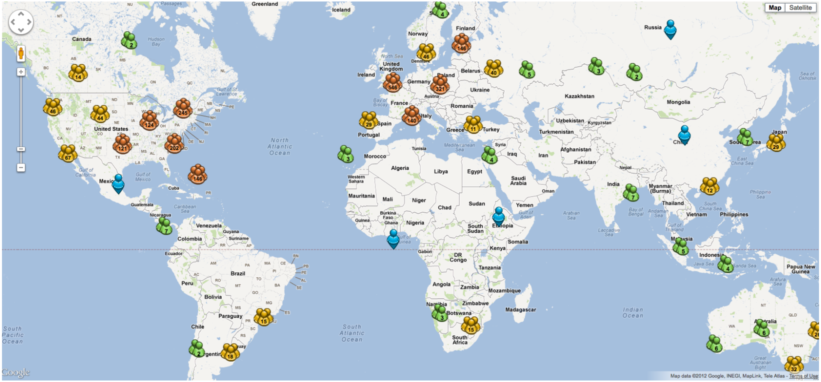
- At any point in time 600–700 VMs connected
- Overall: More than 1 trillion events created



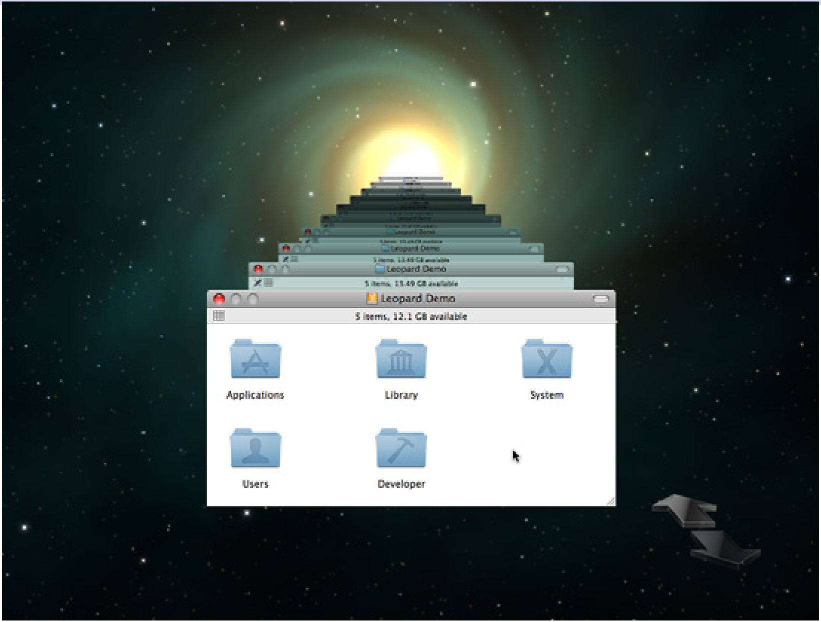
Source: Harutyunyan

Volunteer Computing: LHC@Home 2.0

1 day geographic distribution, 2500 distinct IPs



A Time Machine for the Analysis Environment

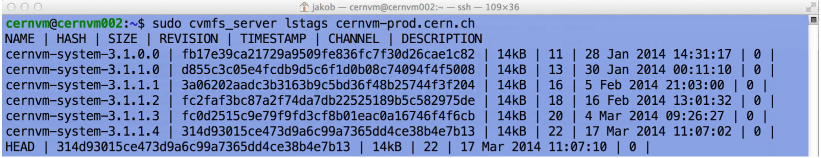


A Time Machine for the Analysis Environment

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 3 operating system repository:



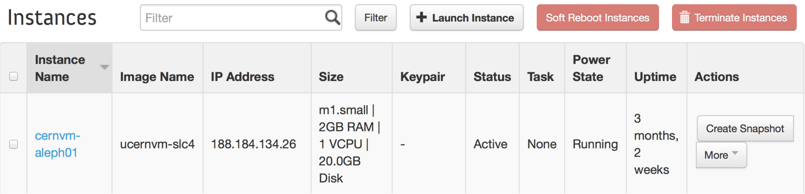
```
cernvm@cernvm002:~$ sudo cvmfs_server lstags cernvm-prod.cern.ch
NAME | HASH | SIZE | REVISION | TIMESTAMP | CHANNEL | DESCRIPTION
cernvm-system-3.1.0.0 | fb17e39ca21729a9509fe836fc7f30d26cae1c82 | 14kB | 11 | 28 Jan 2014 14:31:17 | 0 |
cernvm-system-3.1.1.0 | d855c3c05e4fcdb9d5c6f1d0b08c74094f4f5008 | 14kB | 13 | 30 Jan 2014 00:11:10 | 0 |
cernvm-system-3.1.1.1 | 3a06202aadc3b3163b9c5bd36f48b25744f3f204 | 14kB | 16 | 5 Feb 2014 21:03:00 | 0 |
cernvm-system-3.1.1.2 | fc2faf3bc87a2f74da7db22525189b5c582975de | 14kB | 18 | 16 Feb 2014 13:01:32 | 0 |
cernvm-system-3.1.1.3 | fc0d2515c9e79f9fd3cf8b01eac0a16746f4f6cb | 14kB | 20 | 4 Mar 2014 09:26:27 | 0 |
cernvm-system-3.1.1.4 | 314d93015ce473d9a6c99a7365dd4ce38b4e7b13 | 14kB | 22 | 17 Mar 2014 11:07:02 | 0 |
HEAD | 314d93015ce473d9a6c99a7365dd4ce38b4e7b13 | 14kB | 22 | 17 Mar 2014 11:07:10 | 0 |
```

ALEPH software on CernVM / SL4

Work in progress

Exercise: resurrecting the ALEPH environment

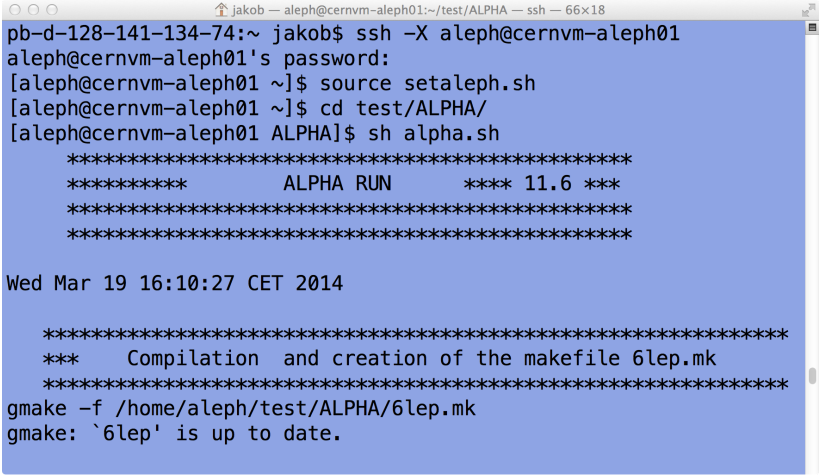
- Use CernVM on current CERN OpenStack infrastructure to do ALEPH physics?
- Backport of CernVM-FS to Scientific Linux 4
- Template installation of Scientific Linux 4 for use with μCernVM



□	Instance Name	Image Name	IP Address	Size	Keypair	Status	Task	Power State	Uptime	Actions
□	cernvm-aleph01	ucernvm-slc4	188.184.134.26	m1.small | 2GB RAM | 1 VCPU | 20.0GB Disk	-	Active	None	Running	3 months, 2 weeks	Create Snapshot More

ALEPH software on CernVM / SL4

Work in progress



```
pb-d-128-141-134-74:~ jakob$ ssh -X aleph@cernvm-aleph01
aleph@cernvm-aleph01's password:
[aleph@cernvm-aleph01 ~]$ source setaleph.sh
[aleph@cernvm-aleph01 ~]$ cd test/ALPHA/
[aleph@cernvm-aleph01 ALPHA]$ sh alpha.sh
     ***********************************************
     **********        ALPHA RUN      **** 11.6 ***
     ***********************************************
     ***********************************************

Wed Mar 19 16:10:27 CET 2014

   **************************************************************
   ***    Compilation  and creation of the makefile 6lep.mk
   **************************************************************
gmake -f /home/aleph/test/ALPHA/6lep.mk
gmake: `6lep' is up to date.
```

CMS Open Data Pilot on CernVM / SL5

Work in progress

Purpose: Provide an easy-to-use virtual machine with CMS computing environment for CMS Open Data

Data:

- Frozen data set
- Remote data access
  Initially through XrootD, eventually DPHEP portal

Software:

- Frozen CMS software framework (CMSSW.4.2.8.patch7)
- *Complete* analysis environment required (compile + run)
- Requires Scientific Linux 5 compatible virtual machine

Virtual machine, user interface:

- Graphical environment
- Easy-to-install and easy-to-use

CMS Open Data Pilot on CernVM / SL5

Work in progress

Deployment: as OVF/OVA bundle[1]

- Open specification for bundling VMs, stable since 2009
- OVA: tarball containing hard disk image and an XML specification

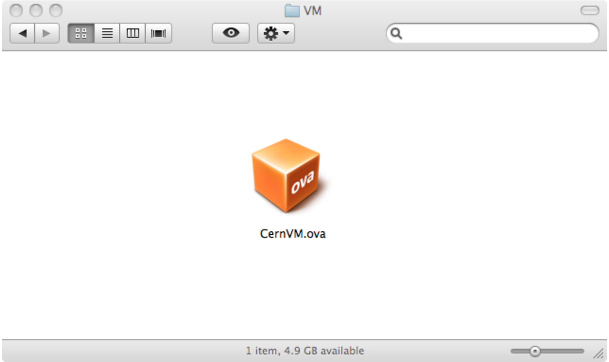
[1] Open Virtualization Format / Open Virtual Appliance, http://www.dmtf.org/standards/ovf

CMS Open Data Pilot on CernVM / SL5

Work in progress

Deployment: as OVF/OVA bundle[1]

- Open specification for bundling VMs, stable since 2009
- OVA: tarball containing hard disk image and an XML specification



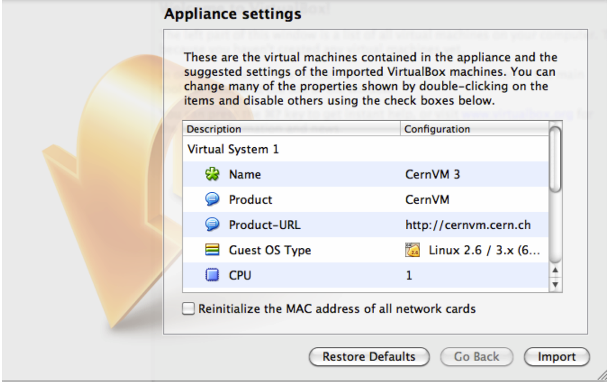
[1] Open Virtualization Format / Open Virtual Appliance, http://www.dmtf.org/standards/ovf

CMS Open Data Pilot on CernVM / SL5

Work in progress

Deployment: as OVF/OVA bundle[1]

- Open specification for bundling VMs, stable since 2009
- OVA: tarball containing hard disk image and an XML specification



Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Description	Configuration
Virtual System 1	
Name	CernVM 3
Product	CernVM
Product-URL	http://cernvm.cern.ch
Guest OS Type	Linux 2.6 / 3.x (6...
CPU	1

□ Reinitialize the MAC address of all network cards

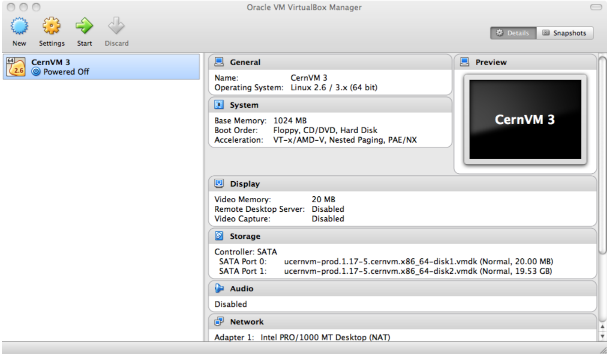
[1] Open Virtualization Format / Open Virtual Appliance, http://www.dmtf.org/standards/ovf

CMS Open Data Pilot on CernVM / SL5

Work in progress

Deployment: as OVF/OVA bundle[1]

- Open specification for bundling VMs, stable since 2009
- OVA: tarball containing hard disk image and an XML specification



CernVM 3
Powered Off

General

Name: CernVM 3
Operating System: Linux 2.6 / 3.x (64 bit)

System

Base Memory: 1024 MB
Boot Order: Floppy, CD/DVD, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX

Display

Video Memory: 20 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage

Controller: SATA
SATA Port 0: ucernvm-prod.1.17-5.cernvm.x86_64-disk1.vmdk (Normal, 20.00 MB)
SATA Port 1: ucernvm-prod.1.17-5.cernvm.x86_64-disk2.vmdk (Normal, 19.53 GB)

Audio

Disabled

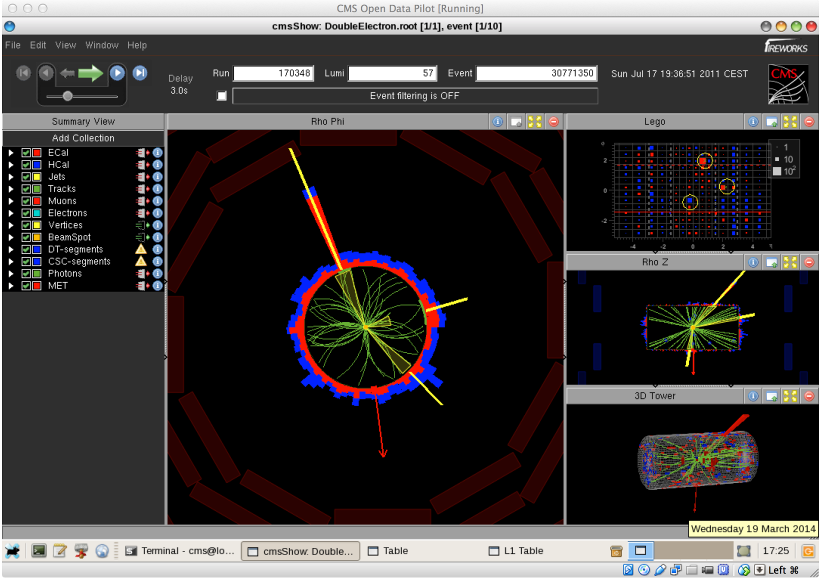
Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

[1] Open Virtualization Format / Open Virtual Appliance, http://www.dmtf.org/standards/ovf

CMS Open Data Pilot on CernVM / SL5

Work in progress



CMS Open Data Pilot [Running]

cmsShow: DoubleElectron.root [1/1], event [1/10]

Delay 3.0s

Run 170348 Lumi 57 Event 30771350

Sun Jul 17 19:36:51 2011 CEST

□ Event filtering is OFF

- ☑ ECal
- ☑ HCal
- ☑ Jets
- ☑ Tracks
- ☑ Muons
- ☑ Electrons
- ☑ Vertices
- ☑ BeamSpot
- ☑ DT-segments
- ☑ CSC-segments
- ☑ Photons
- ☑ MET

Wednesday 19 March 2014

Summary

CernVM

- μCernVM + OS template on CernVM-FS + Contextualization
- 20 MB image that adapts
- Image for IaaS clouds, volunteer clouds, long-term data preservation, development environment
- Next steps: SL7, Docker

CernVM-FS

- Software Distribution
- HTTP transport, versioning, on-demand download & caching
- Limited applicability for data: common cluster working set required
- Next steps: better support for opportunistic resources, better support for small VOs

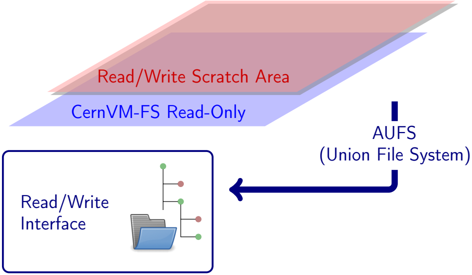
---

Source code: https://github.com/cernvm, https://github.com/cvmfs
Documentation and downloads: http://cernvm.cern.ch
Bug tracker: https://sft.its.cern.ch/jira/browse/CVM

Mailing lists: cernvm-talk@cern.ch, cvmfs-talk@cern.ch

6 Backup Slides

μCernVM Root File System Stack



CernVM-FS features targeted to loading the OS:

- Closing all read-write file descriptors in order to unravel file system stack on shutdown
- Redirect syslog messages
- In-flight change of DNS server
- GID / UID mappings

⇒ This file system stack requires special support from a read-only Fuse branch since it is started before the operating system.

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: $x_a + x_b \leq 1$.

$(\dots)$

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)

Use Cases for a Preserved 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 or just impossible
2. Validation of new software versions (see talk by S. Roiser)
   - Comparison with historic version provides input for validation
3. Stable environment for education (cf. CMS Open Data Pilot)
   - Stable operating system and experiment software version accompanies "open data" set and well-defined analysis tasks
   - Driver for data preservation:
     Opportunity to streamline data format and documentation
     Disentangle from grid environment

Use Cases for a Preserved 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 or just impossible
2. Validation of new software versions (see talk by S. Roiser)
    - Comparison with historic version provides input for validation
3. Stable environment for education (cf. CMS Open Data Pilot)
    - Stable operating system and experiment software version accompanies "open data" set and well-defined analysis tasks
    - Driver for data preservation:
      Opportunity to streamline data format and documentation
      Disentangle from grid environment

Use Cases for a Preserved 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 or just impossible
2. Validation of new software versions (see talk by S. Roiser)
    - Comparison with historic version provides input for validation
3. Stable environment for education (cf. CMS Open Data Pilot)
    - Stable operating system and experiment software version accompanies "open data" set and well-defined analysis tasks
    - Driver for data preservation:

      Opportunity to streamline data format and documentation

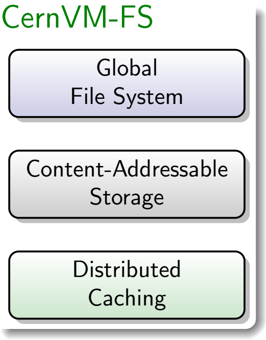
      Disentangle from grid environment

Software vs. Data

Based on ATLAS Figures 2012

Software	Data
POSIX Interface	put, get, seek, streaming
File dependencies	Independent files
10^7 objects	10^8 objects
10^{12} B volume	10^{16} B volume
Whole files	File chunks
Low latency	High throughput
Absolute paths	Any mount point
Open source	Confidential
WORM (“write-once-read-many”)	
Versioned	

Software vs. Data



Based on ATLAS Figures 2012

Software	**Data**
POSIX Interface	put, get, seek, streaming
File dependencies	Independent files
10^7 objects	10^8 objects
10^{12} B volume	10^{16} B volume
Whole files	File chunks
Variant symlinks	No Symlinks
Absolute paths	Any mountpoint
Open source	Confidential
WORM (“write-once-read-many”) Versioned	

CernVM-FS Client in Heterogeneous Environments

In order to fully benefit from CernVM-FS, the file system has to be available on all relevant computing resources.

Range of Environments:

Scientific Linux, Fedora, Ubuntu, SuSE, OS X
1 core to 48+ cores
Tens of mounted repositories
Possibly no Fuse, disk-less server farms

Portability:

- Portable C++ / POSIX code
- Library interface, connector to *Parrot* (by Dan Bradley)

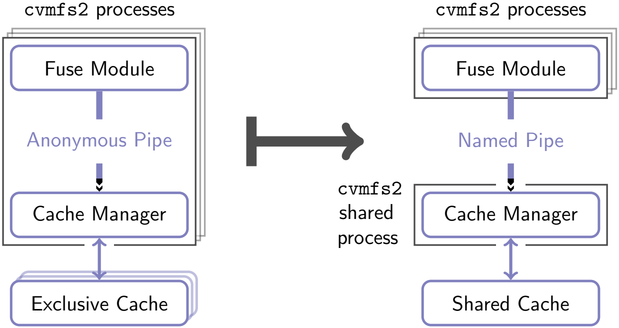
Scalability:

- Memory fragmentation
  open hash collision resolution ↦ linear probing
  path strings stored on the stack
- Concurrent file system access
  Fine-grained locking
  Asynchronous, parallel HTTP I/O
- Cache sharable among repositories
- Hotpatch functionality for Fuse client

Shared Local Hard Disk Cache

Issue: Enforce shared *quota*, coordinated bookkeeping required

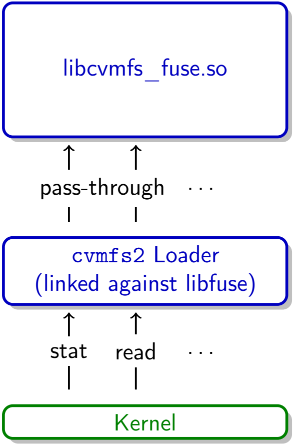
Idea: Turn the cache manager thread into a shared process



- No extra service: automatically spawned by first cvmfs mount point, automatically terminated by last unmount
- Named pipe can be turned into a network socket: Foundation for distributed shared memory cache

Hotpatching / Reloading

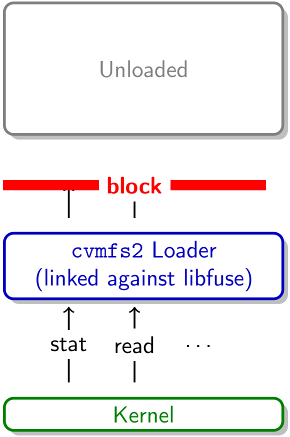
Addresses the issue of "worker node draining"
when there is a new version of CernVM-FS



- A minimal loader implements the Fuse interface
- The logic is part of an (unloadable) shared library
- Very little state across file system calls: open files and open directories
- Can be also seen as a reload of parameters (like SIGHUP)

Hotpatching / Reloading

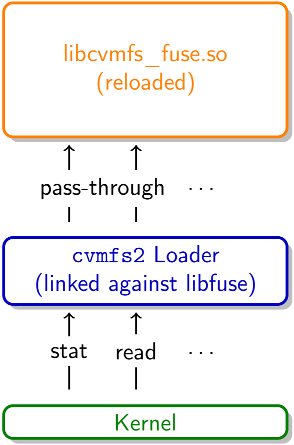
Addresses the issue of “worker node draining” when there is a new version of CernVM-FS



- A minimal loader implements the Fuse interface
- The logic is part of an (unloadable) shared library
- Very little state across file system calls: open files and open directories
- Can be also seen as a reload of parameters (like SIGHUP)

Hotpatching / Reloading

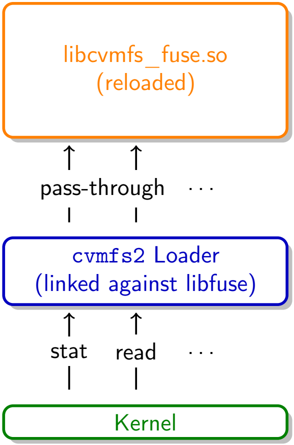
Addresses the issue of "worker node draining"
when there is a new version of CernVM-FS



- A minimal loader implements the Fuse interface
- The logic is part of an (unloadable) shared library
- Very little state across file system calls: open files and open directories
- Can be also seen as a reload of parameters (like SIGHUP)

Hotpatching / Reloading

Addresses the issue of "worker node draining"
when there is a new version of CernVM-FS



- A minimal loader implements the Fuse interface
- The logic is part of an (unloadable) shared library
- Very little state across file system calls: open files and open directories
- Can be also seen as a reload of parameters (like SIGHUP)

(Just released)

Working Set as seen with CernVM

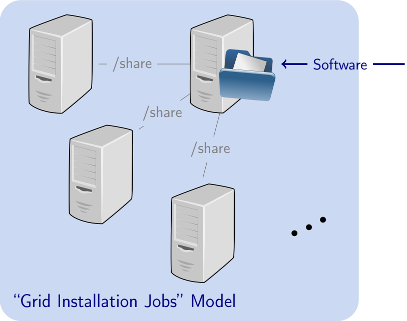
- $\approx$10 % of all available files are requested at runtime
- Median of file sizes: < 4 kB

Working Set as seen with CernVM

- $\approx$10 % of all available files are requested at runtime
- Median of file sizes: < 4 kB

Flash Crowd Effect

- Up to 500 kHz meta data request rate
- Up to 1 kHz file open request rate



"Grid Installation Jobs" Model

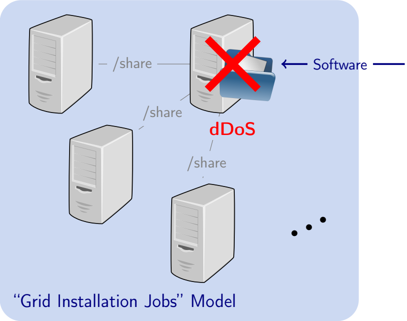
Runtime Behavior

Working Set as seen with CernVM

- ≈10 % of all available files are requested at runtime
- Median of file sizes: < 4 kB

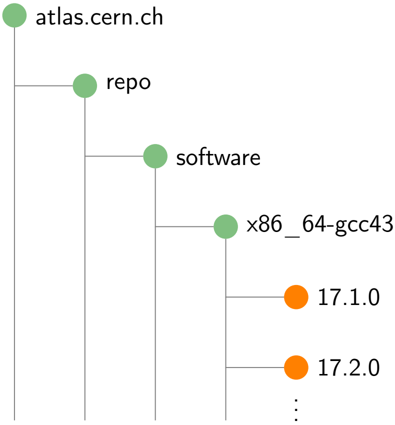
Flash Crowd Effect

- Up to 500 kHz meta data request rate
- Up to 1 kHz file open request rate



Experiment Software from a File System Viewpoint

Software Directory Tree



atlas.cern.ch

repo

software

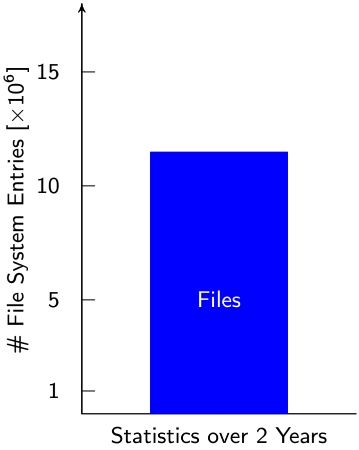
x86_64-gcc43

17.1.0

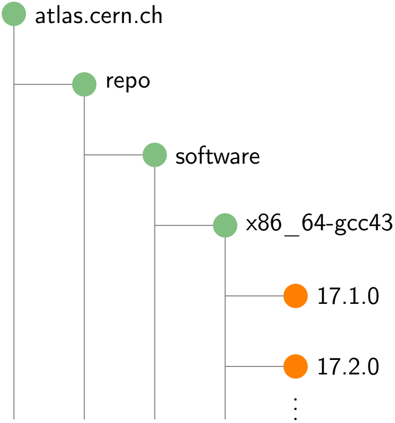
17.2.0

⋮

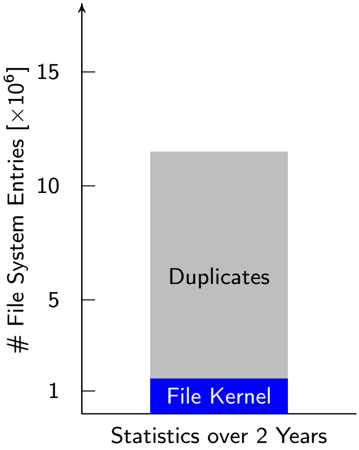
Experiment Software from a File System Viewpoint



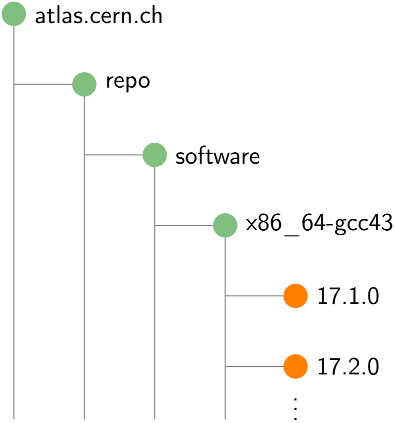
Software Directory Tree



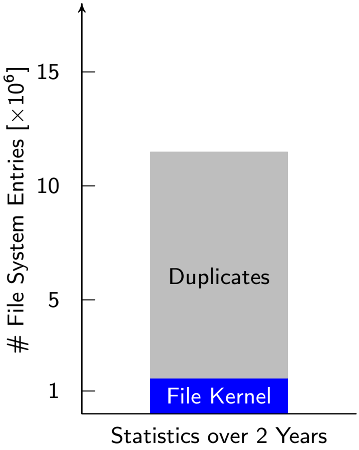
Experiment Software from a File System Viewpoint

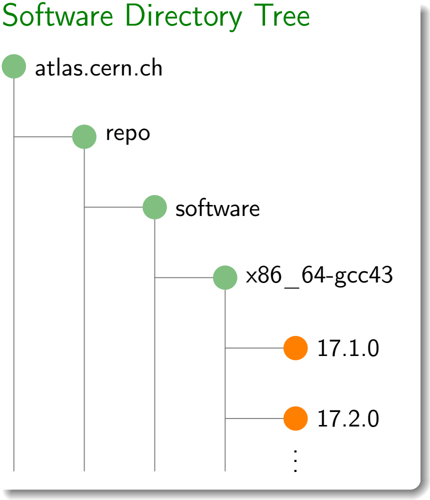


Software Directory Tree



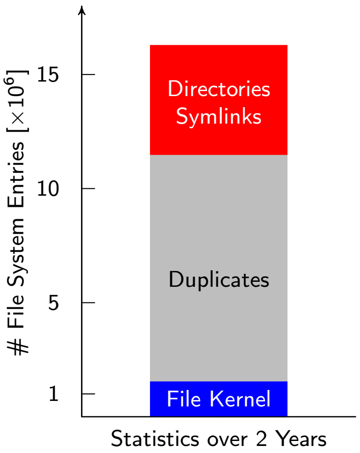
Experiment Software from a File System Viewpoint

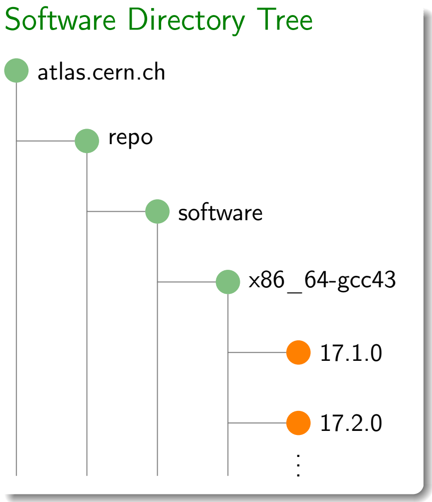




Between consecutive software versions: only ≈ 15 % new files

Experiment Software from a File System Viewpoint

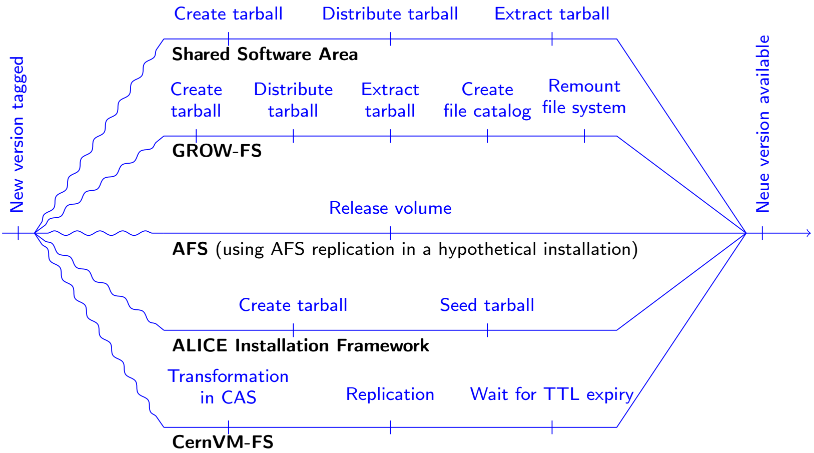




Fine-grained software structure (*Conway's law*)

Between consecutive software versions: only ≈ 15 % new files

Turn-Around for Software Distribution



Improvement: from days (Grid Installation Jobs) to hours

CernVM-FS Client Tools

Fuse Module

- Normal namespace: `/cvmfs/<repository>` e. g. `/cvmfs/atlas.cern.ch`
- Private mount as a user possible
- One process per fuse module + watchdog process
- Cache on local disk
- Cache LRU managed
- NFS Export Mode
- Hotpach functionality `cvmfs_config reload`

Mount helpers

- Setup environment (number of file descriptors, access rights, . . .)
- Used by autofs on /cvmfs
- Used by /etc/fstab or mount as root

```
mount -t cvmfs atlas.cern.ch
/cvmfs/atlas.cern.ch
```

Diagnostics

- Nagios check available
- `cvmfs_config probe`
- `cvmfs_config chksetup`
- `cvmfs_fsck`
- `cvmfs_talk`, connect to running instance

Considerations about volunteer computing

- Resources are unmanaged, e. g. there are no grid services
- Resources are untrusted, e. g. no grid certificates can be stored
  Results are typically verified by
  - Sanity checker
  - Credit based trust relationship
  - Quorum of results
- Resources are volatile; small or resumable work units required
- Resources are heterogenous (poor and powerful, Linux, Windows, Mac)
- Resources are donated; credits and community support required
- Resources have poor I/O connection;
  works for CPU dominated workloads only
- You must never run out of work!
- Technology can be re-used for "corridor grid"