PROOF on the Cloud

using CernVM and PROOF on Demand

Dario Berzano

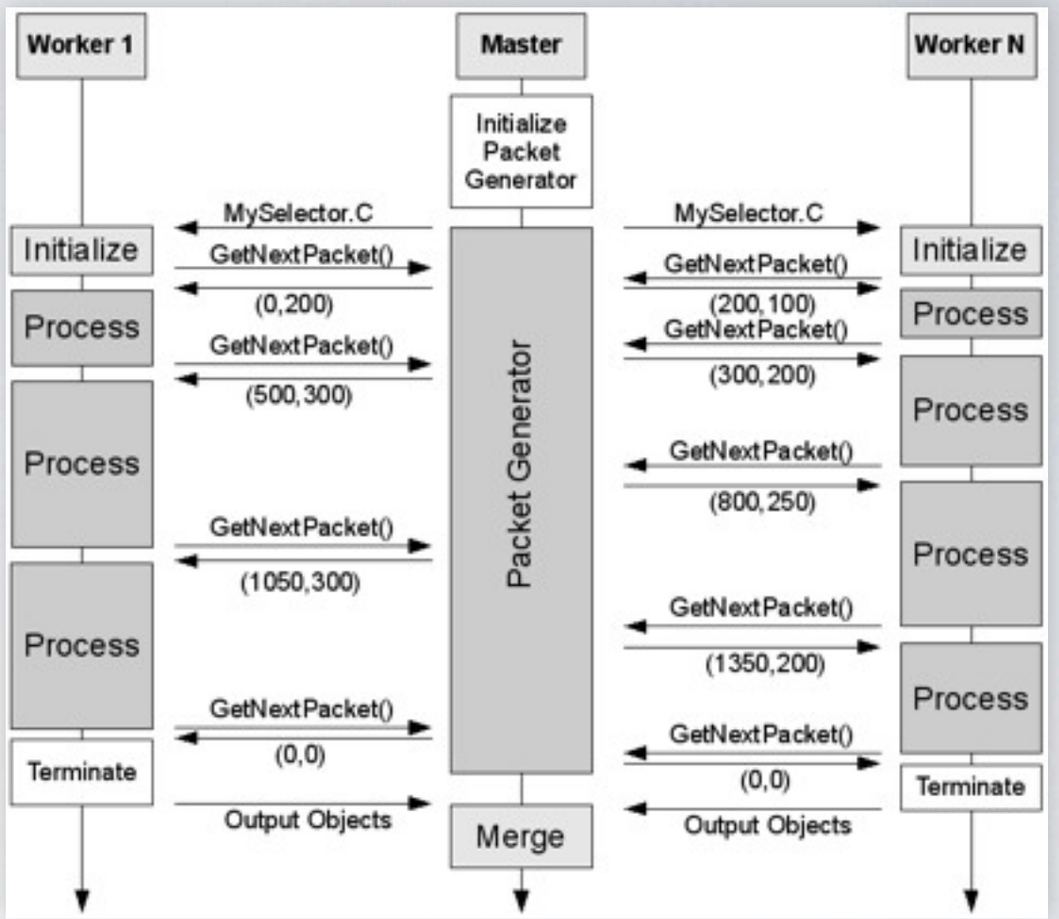
CERN PH-SFT

Workshop Commissione Calcolo e Reti INFN - Genova, 31 mag 2013

Introduction

Interactive analysis with PROOF

- Event-based parallelism
  - Process single events in parallel
  - Merge results eventually
- Interactive
  - All workers active at the same time
  - Workload assignment is dynamic
    → *Uneven work distribution leads to uniform completion time*



http://root.cern.ch/drupal/content/proof

Rationale: PROOF as a Service

Historical notable PROOF deployments

Dedicated clusters	PROOF on Demand
e.g. ALICE: standalone PROOF	*e.g. ATLAS: workers submission*

- Currently: diverse PROOF deployments
  → *issues are diverse as well: difficult to cope with everything*

- Cloud computing: system administrator ≠ service administrator
  → *"PROOF as a Service": boot a cluster of VMs to provide it*

- Elasticity
  → *PROOF "size" (num. of VMs) can be adapted on demand*

Goal: to provide a zero-conf "PROOF as a Service" cluster
a well-known reference deployment with no client requirements

The bricks that make the house

The Virtual Analysis Facility

- A cluster of original unmodified CernVM virtual machines
  → *all configured during contextualization*
- Cluster context: one head node + scalable num. of workers
  → *available on http://cernvm-online.cern.ch*
- Portability and usability
  → *both for users and system administrators*
- One PROOF deployment for all LHC experiments

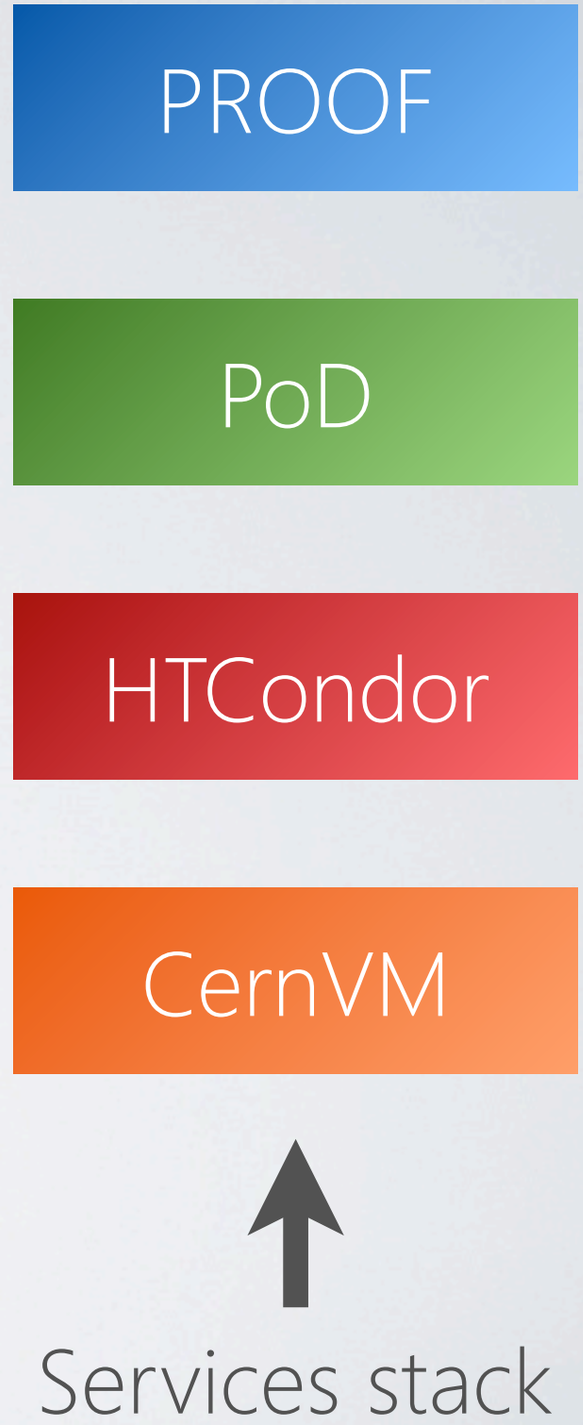
Overview of the VAF components

User interaction

- PoD to request and book workers
- PROOF to (re)use booked workers for analysis

Behind the scenes

- worker requests are scheduled by HTCondor*
  ** can be any resource management system*
- CernVM virtual machines are part of the HTCondor cluster



VAF components: PROOF and PoD

- PROOF on Demand is a scheduler and resource broker for PROOF on a general purpose RMS
- Our virtual cluster is a special dedicated RMS

PROOF benefits from PoD:

- Sandboxing
  → *no crash propagation to other users*
- Self-servicing
  → *users start/stop their personal PROOF cluster*
- No system-wide configuration
  → *config pushed by client, no privileged daemons*

More stability and less administration efforts

VAF components: HTCondor

HTCondor is a resource management system:

- enqueues and schedules jobs
- manages a distributed cluster

- Usually, HTCondor jobs are independent
  → *ours are communicating PROOF workers*

- Adds efficient users scheduling to PROOF
  → *user books resources that can be reused*

- Dynamic workers addition with no configuration
  → *new CernVM nodes promptly join the cluster*

Queue + dynamic config = elasticity
monitor the queue and start new VMs

VAF components: the CernVM ecosystem

The CernVM ecosystem provides our reference platform:

- CernVM Virtual Machine
  → *consistent running and devel environment*
- CernVM Filesystem
  → *sw downloaded transparently on demand*
- CernVM Online
  → *contextualize VMs and clusters via web*
- CernVM Gateway *(experimental)*
  → *talks with your clouds to instantiate VMs*

See Jakob Blomer's presentation

PROOF

PoD

HTCondor

CernVM

Services stack

Personal or multiuser?

Personal Analysis Facility
Infrastructure as a Service

- Users take care of deploying their cluster on the cloud
- User must have access to a cloud infrastructure

Shared Analysis Facility
Software as a Service

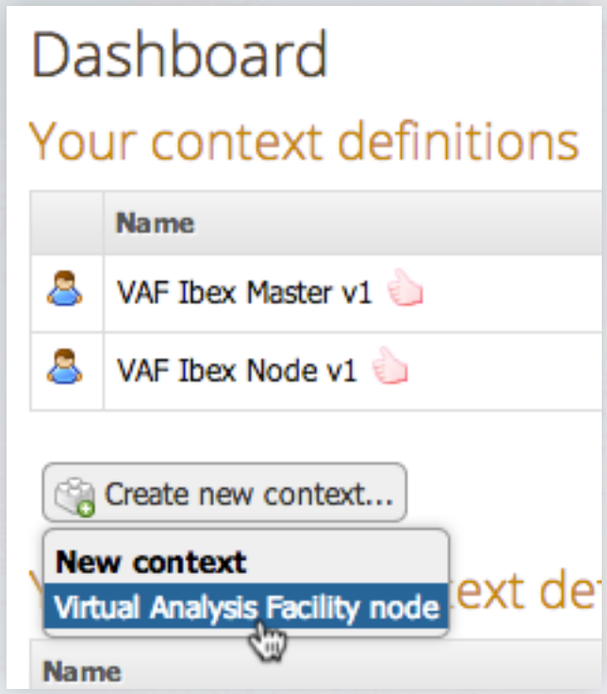
- A service administrator deploys the multi-user service
- End users are not responsible (nor aware) of the VMs

User analysis workflow is the same in both cases (always SaaS)

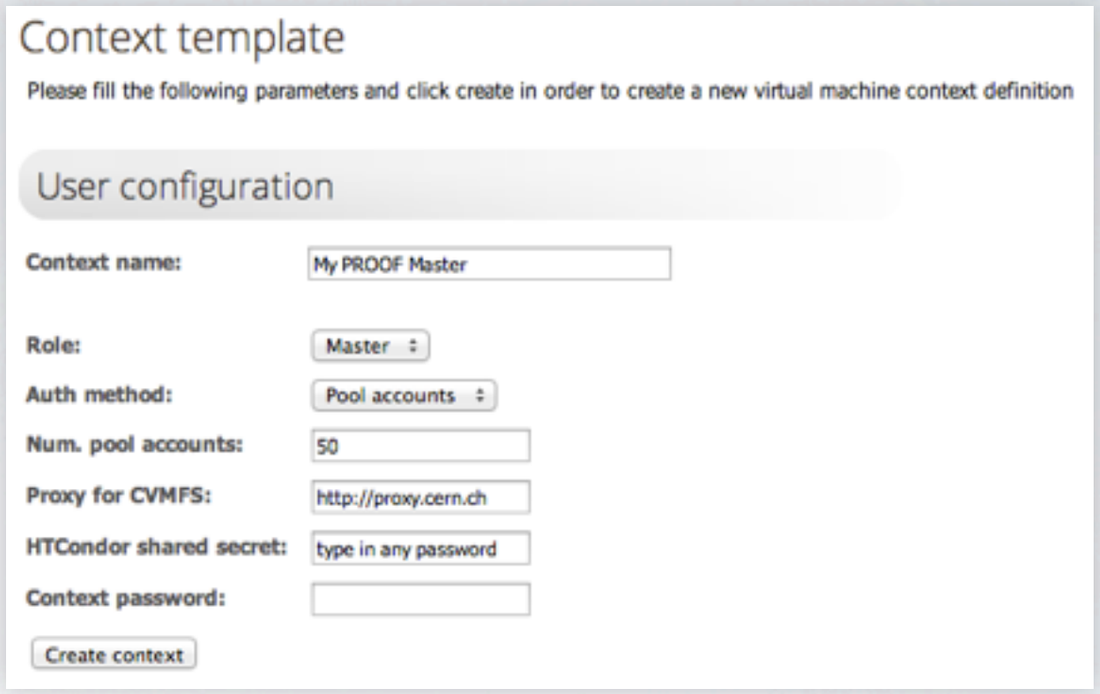
The Virtual Analysis Facility in practice

Instantiating the VAF

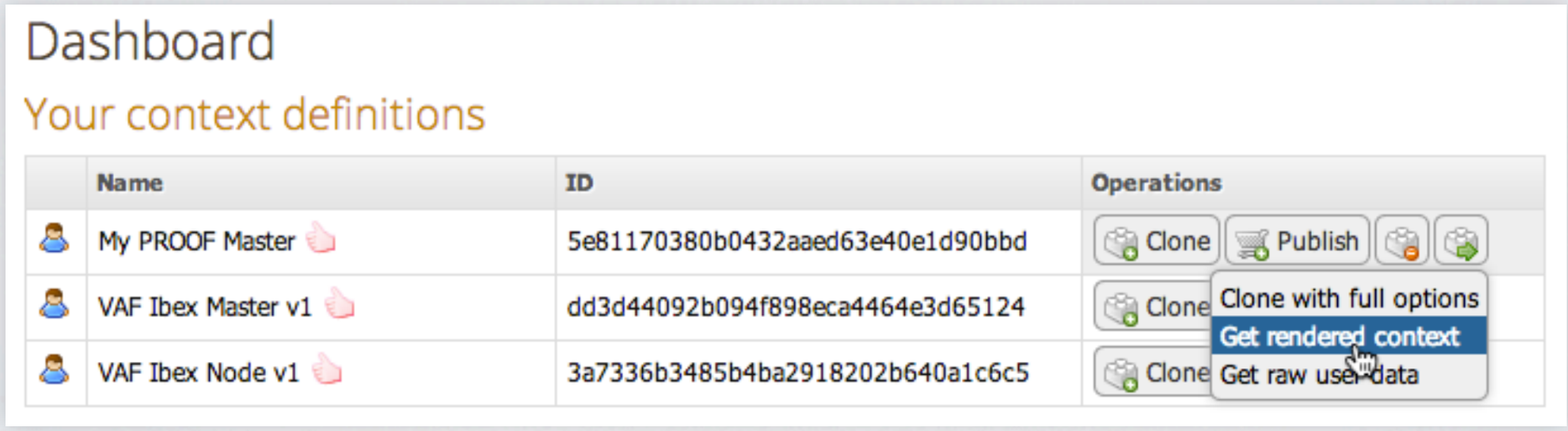
CernVM Online → *http://cernvm-online.cern.ch*



1. Create context

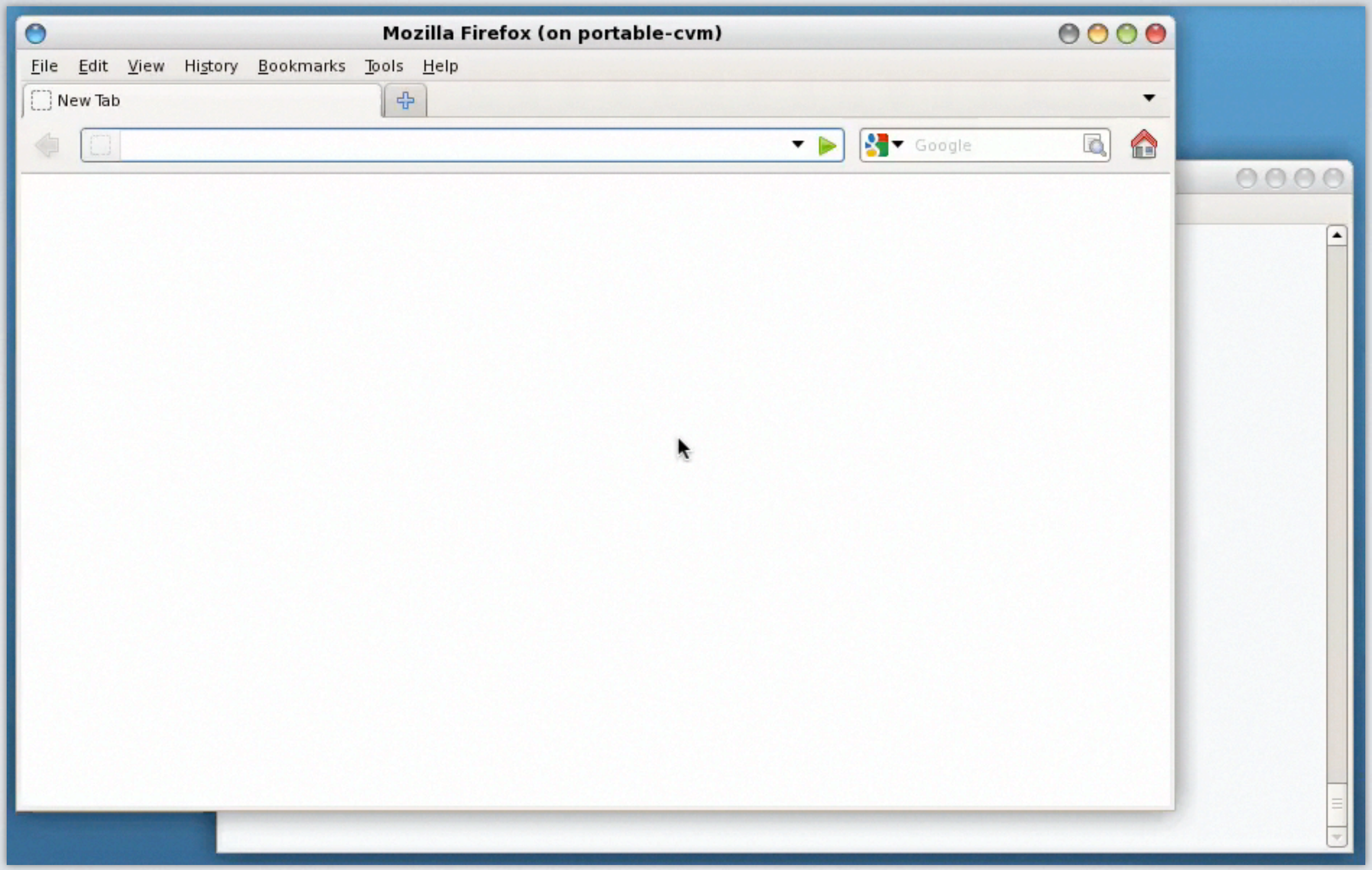


2. Customize options



3. Get configuration → *cloud controller's "user data"*

Using the VAF



Practical reference

Administrator's and User's manual
http://proof.web.cern.ch/

- Virtual Analysis Facility client *(only one script)*
  → *https://github.com/dberzano/virtual-analysis-facility*

- Builtin authentication uses your Grid certificate and key for SSH
  → *https://github.com/dberzano/sshcertauth*

Sample user's configuration file for CMS

```
# Version of CMSSW (as reported by "scram list")
export VafCmsswVersion='CMSSW_5_3_9_sherpa2beta2'
```

Development directions

Elasticity using CernVM components

request 8 workers

request VMs

fwd req

- CernVM Agent and Gateway are experimental
- CernVM components enable automatic elasticity

PROOF development directions

Dynamic addition of workers

- Currently: new workers can be added to PROOF, but become available only on the next analysis
- Make possible for new workers to join an ongoing analysis
- Perfect for PoD: analysis could be started when only one requested worker is available, others will join automatically

Improve object sending and merging strategy

- Evaluate a non-locking master collector for output data
- Reduce master memory usage during merging

PoD and VAF in LHC experiments

ATLAS

- Future plans to do "cloud computing" (i.e., "VM submission")
  → *PROOF is run mostly using PoD on local queues (PBS, LSF…)*
- ATLAS software is officially on cvmfs
- Past attempts to exploit the Grid for PROOF using PoD gLite-WMS
- PoD Panda plugin is almost ready for testing
  → *Currently solving some Tier connectivity troubles*
- Data access: mostly D3PD and skimmed D4PD via federated xrootd
  → *As soon as Panda is ready: xrootd access tests*
- For ntuples output native Panda merging can be used
  → *Interactive analysis with PROOF and deferred merging*

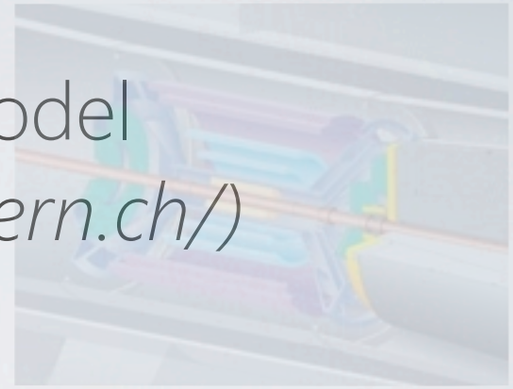
Thanks to A. De Salvo for the contribution

CMS Padova

- VAF-like setup (yet on physical hardware) in Padova
  → *it uses PROOF on Demand and the VAF client*
- SL5 nodes configured like Grid nodes: no extra effort
  - all software from cvmfs, all data from Lustre, auth from LDAP
  - integrated in current local LSF: separated PoD queue used
- A few WNs and UIs currently enabled for PROOF *(tests ongoing)*
  → *switched to PROOF when needed by changing LSF configuration*
- Turned out that lack of documentation was a showstopper…
  → *current documentation (http://proof.web.cern.ch/) is OK*

Thanks to M. Sgaravatto for this contribution (and feedback!)

ALICE



- PROOF is officially part of the computing model
  → *AAF: ALICE Analysis Facilities (http://aaf.cern.ch/)*
- PoD not used for now
  → *native PROOF is used*
- Current analysis facilities are mostly static and on physical hardware
  → *notable exception: TAF in Torino, ancestor of the current VAF*
- TAF (Torino Analysis Facility) is currently 100% VAF
  → *CernVM + PoD + cvmfs*
- Plans to gradually move CAF @ CERN to VAF
  → *on CERN OpenStack infrastructure*

Final considerations

- Every VAF layer is "cloud-aware" or "elastic":
  - HTCondor plays optimally with nodes added/removed on the go
    → *"cattle computing", not "puppies" (http://bit.ly/15cMdrR)*
  - PROOF has a pull-based dynamic workload assignment
    → *drop the assumption that same-sized VMs perform equally*
- Single components are optional and non-obtrusive
  → *e.g. elasticity is achieved by silently observing a queue*
- VAF works out of the box with zero configuration
  → *only the batch scheduler might need fine tuning for multiusers*
- VAF reuses and integrates solid components
  → *don't reinvent the wheel (http://bit.ly/134U4He)*

Thank you!