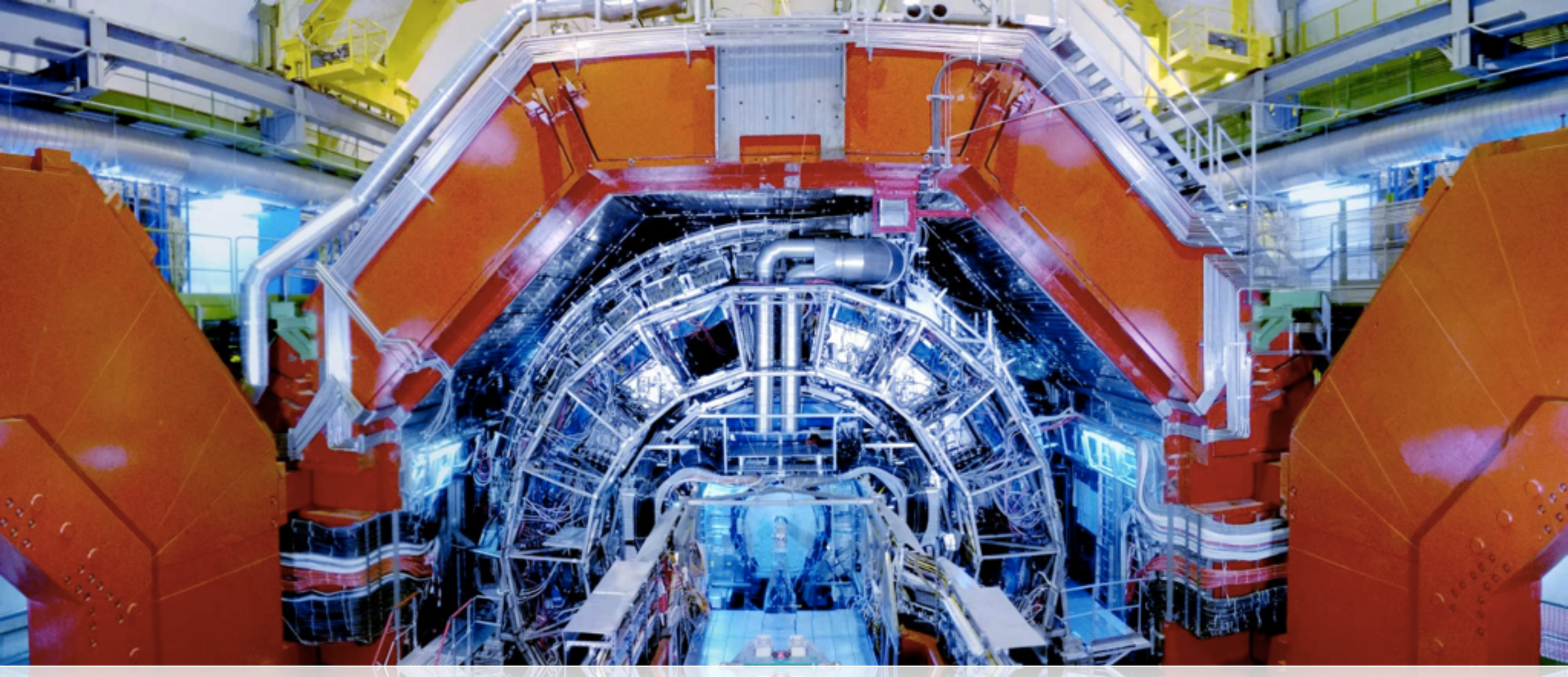


The new CAF

Dario Berzano

ALICE Offline - CERN

ALICE Offline Week - Jul 1-3, 2015

CernVM Elastic Clusters

CernVM elastic clusters on the cloud

- **Cluster:** the High Energy Physics model
  - A batch system with one head and many identical workers
- **Elastic:** number of workers is not fixed
  - Expand and shrink on demand
- **CernVM:** base image for virtual clusters
  - Lightweight: full root filesystem from CVMFS
- **Cloud:** generic and standalone
  - Works with any cloud through a standard API (EC2)
  - Fully self-contained: no external tools needed!

Perspectives: a two actors model

User: runs elastic clusters

- A zero-configuration workload management system
- Exploit cloud resources in a familiar way: submit to batch
- No prior knowledge of needed resources: scale transparently

Admin: manages the cloud

- Self-contained analysis cluster not interfering with the rest
- No administration: resources relinquished when unused
- Different users with way different needs: multitenancy

- Users with a cloud account* can create their cluster in seconds
- Cloud layer invisible to the users: standard job submission
- *Every CERN user has one: https://openstack.cern.ch/

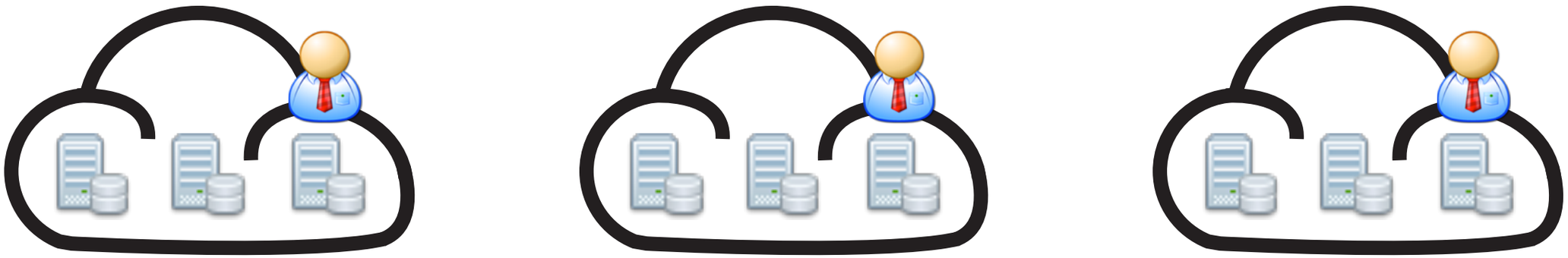
Perspectives: a three actors model

virtual infrastructure administrator

configures virtual machines: does not care about the underlying hardware

user

uses the services just like before, completely unaware of virtualization



administrators of distributed and independent clouds

manage the hardware, replace disks when broken, monitor resources usage, coordinate “local” and “remote” users

- New CAF model: we deploy it, virtualization transparent to users

Anatomy of an elastic cluster

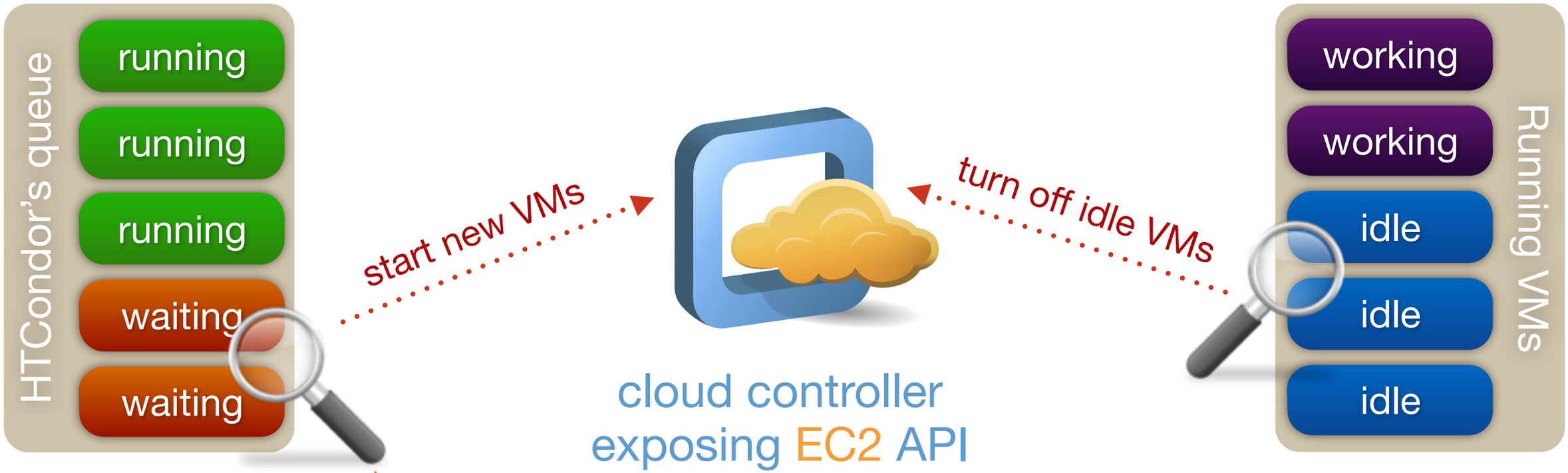
Your task | CernVM | HTCondor | elastiq

What is an Elastic Cluster?

- Cluster of CernVM virtual machines: one head node, many workers
- Running the HTCondor job scheduler
- Capable of growing and shrinking based on the usage with elastiq

- Automatic deployment: cluster launched with a single command
  - You launch the head node
  - Head node deploys all the workers thanks to elastiq
- User logs in and interacts only through job submission

Deploying and scaling the workers: elastiq



Jobs waiting too long will trigger a scale up

Supports minimum and maximum quota of VMs

You deploy only the master node: minimum quota immediately launches VMs automatically

Catches deployment errors and retries automatically

Very simple but very effective and stable

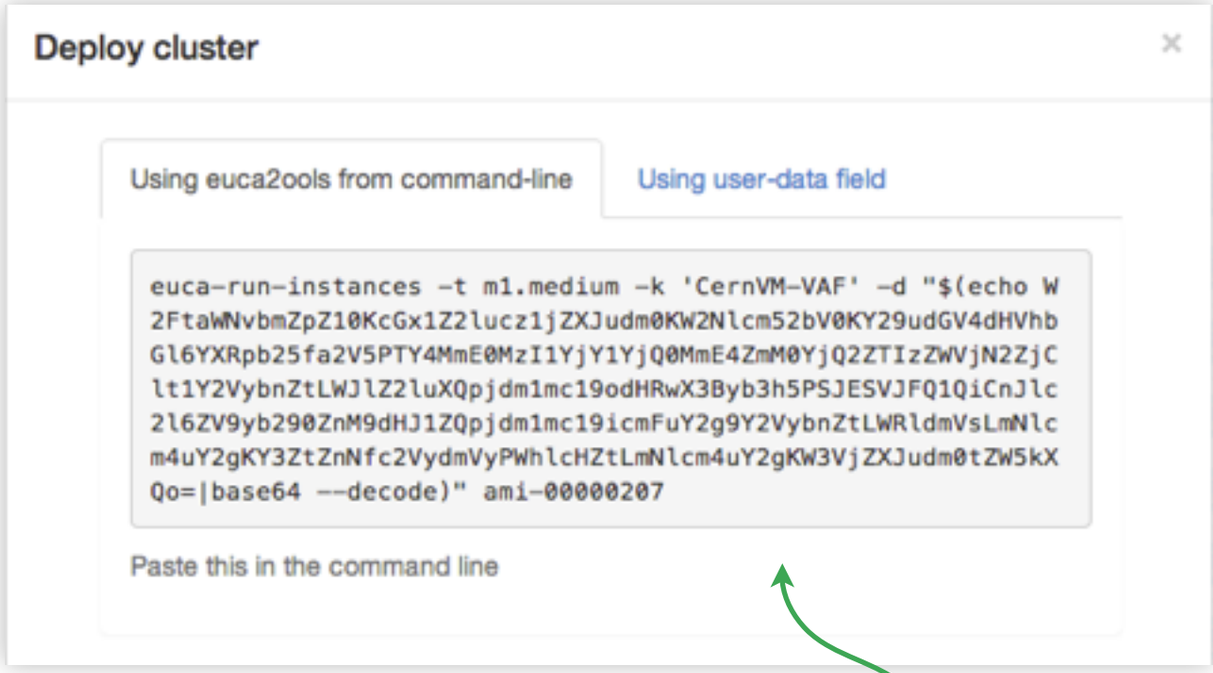
Integrated in CernVM

RPM + deb available
github.com/dberzano/elastiq/releases

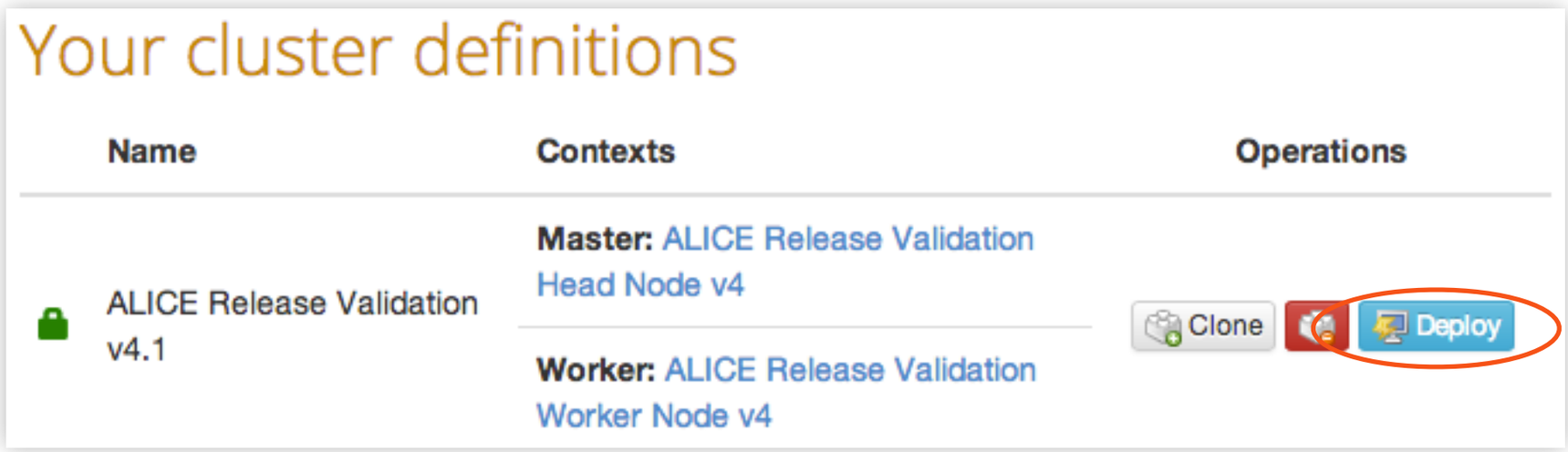
Create an elastic cluster yourself

A cluster in four steps

- Configure the head node
- Configure the worker nodes
- Put them together in a cluster
- Deploy the cluster



...then **copy and paste** the generated command



Just click on the **deploy** button...

- DIY simple procedure: cernvm.cern.ch/portal/elasticclusters

Running PROOF on the cloud

Virtual Analysis Facility: a brief history

- Original Virtual Analysis Facility conceived in Torino's Tier-2
  - From an idea of S. Bagnasco: making Grid worker nodes and PROOF coexist on the same infrastructure: "multitenancy"
  - Main motivation for moving Torino's Tier-2 to OpenNebula
- My Ph.D. thesis: from PROOF-specific VAF to elastic clusters
  - Elastic clusters are generic: PROOF is a possible application
  - Work done within the PH-SFT group at CERN
- Italian HEP community: government-funded project
  - Very important in spreading and sharing the competences on cloud computing in HEP-related Italian research institutions

PROOF and ALICE

- PROOF: the Parallel ROOT Facility. Features:
  - Automatic merging and immediate display of results
  - Use parallel resources from your ROOT prompt
  - Fine-grained dynamic scheduling
- PROOF in ALICE: AAFs (M. Vala, A. Hayrapetyan). Use cases:
  - Debugging and fine tuning of analysis parameters
  - Fast calibrations: appreciated during data taking
- PROOF users in ALICE are, historically, mostly power users:
  - A preferential lane for the most important computing tasks
  - Provide a quick response even when the Grid is flooded

PROOF on Demand

- PROOF on Demand by A. Manafov (pod.gsi.de)
  - No PROOF deployment needed: run it over any batch resource
  - Use batch resources in an interactive fashion
  - Book batch resources once, reuse them many times
    - VAF = elastic cluster + PoD
- Reduced management efforts and improved quality of service:
  - AAF: “CAF down” occurred frequently, manual intervention from admins required
  - VAF: PROOF daemons are user sandboxes: self-service recovery and crash isolation

PoD and environment consistency

- AAF: static PROOF deployment
  - PROOF daemons used a certain ROOT version
  - Your PROOF session could use another ROOT version (you could choose it before connecting to PROOF)
  - This inconsistency caused hiccups from time to time

- VAF: same ROOT version on client, daemons and sessions
  - No ROOT on your laptop: SSH to a login node
  - Env from CVMFS using alienv: set AliRoot/AliPhysics only, dependencies are automatic
  - Much less error prone: no chance to get deps wrong

The Virtual CAF: what's new

Runs at CERN OpenStack

- ALICE has resources on CERN OpenStack (openstack.cern.ch):
  - Physically running at the Wigner datacenter in Hungary
  - It still “appears” to be in the CERN network
- Currently allotted resources (we can have more):
  - 50 VMs, 200 CPUs, 500 GB of RAM
  - Each VM has 4 CPUs and 8 GB of RAM
- Thanks to the CERN IT for providing us with the OpenStack project and support!

Administrative domains in practice

- CERN IT: manages the OpenStack cluster and all the associated hardware and software services
- ALICE Offline: we manage the sole VAF service
- “Outsourcing”: since we do not control the hardware, it is easier for us to concentrate our efforts on the service
- We simply redeploy the cluster in case of a major disaster, and your data is not lost *(see later on)*

CERN Virtual CAF topology

- The CERN Virtual CAF has a slightly different topology compared to the vanilla VAF
- It scales in two dimensions:
  - Scale worker nodes: they run PROOF workers and are visible from all login nodes *(same as vanilla VAF)*
  - Scale login nodes: they run PROOF masters (sometimes memory intensive operations), we needed to distribute them on many machines
- Deployment of worker nodes is automatic: elastiq even takes care of deployment errors and resubmits VMs if they fail
- Deployment of additional login nodes is manual

CERN Virtual CAF topology: HTCondor

HTCondor

Collector, Negotiator

central manager: not a login node

single node

Schedd | Schedd | Schedd

login nodes: they have PROOF masters

we can scale them manually

tartd | Startd | Startd | Startd | Startd | Startd | Startd | Start

HTCondor workers: they run PROOF workers

they can scale automatically

- We have designed the new CERN VAF to scale it easily

CERN-specific services

- The new CERN VAF has multiple login nodes:
  - Login via Kerberos using your CERN username and password
  - Login restricted to ALICE users (plus a short list of externals)
  - Different from vanilla VAF and works only inside CERN network

- Your home directory is on AFS:
  - Same thing you have from lxplus
  - Shared between all login nodes
  - AFS token transparently created at login (just like lxplus)

CernVM and contextualization

- CernVM “generic” elastic clusters
  - Simple online configuration from cernvm-online.cern.ch
- CERN Virtual CAF
  - Less trivial contextualization with cloud-init
  - Initially developed using a vanilla CentOS 6 image, but works seamlessly on CernVM
  - Tested with both, but we prefer CernVM: we don’t need to maintain our own image

Dealing with failures

- Two persons are responsible for managing the service:
  - Myself and Maarten Litmaath
  - We have a documented procedure for spawning the cluster: dberzano.github.io/alice/vcaf/admin
- No important data is stored on the VAF nodes: they are fully disposable
  - Input data is on the Grid
  - Your home directory is on AFS (external shared storage)
- We can bring up the full cluster (currently 50 nodes) from scratch in less than 20 minutes

Data management: local vs. remote

- Old CAF: data distributed on the physical nodes
  - Very fast access but frequent manual interventions to fix datasets and staging
  - Node down? Data from that node not accessible!
    - Cannot do on VAF: nodes are virtual and disposable
- Current Virtual CAF takes data directly from “the Grid”
  - Fast enough for the moment: most data stored at CERN
  - But we want to improve that
- Using a dedicated EOS cache is foreseen *(see later)*

Data management: datasets

- Old CAF used PROOF datasets
  - Basically: ROOT files containing list of files
  - A staging daemon read the lists to copy data locally
- We have separated computing from data management
  - ROOT is about computing, it should not manage data
- No more static “list of files”
  - Our data is on AliEn: we query the catalog via a PROOF-to-AliEn interface using the “AliEn find” syntax
  - AliEn “find” queries are cached for faster retrievals
  - We will not restore static datasets, ever: everything can be done by querying the AliEn catalog

Using the new CAF

The CAF is dead. Long live the CAF!

- Old CAF has been retired for good
  - Persons building it no longer at CERN: support was difficult
  - Hardware was off warranty and nodes/disks started dying
  - Ultimate deadline for retirement was May 30, but it was effective on Jun 25
- We provide the new Virtual CAF as a replacement
  - Already tested in various implementations in Torino since 2012: nothing new under the sun (seen many times at Offline Weeks)
  - It is mature: we can finally give it to a broader audience

Documentation and support

- Everything you need: dberzano.github.io/alice/vcaf/usersguide
  - Working examples are provided
  - Documentation constantly updated
- Current feedback is positive:
  - More stable than previous CAF
  - Easier to set environment: sw deps are automatic (no errors)
  - Previous Analysis Tasks needn’t any modification
- Open JIRA tickets for requesting support (on the AAF project):
  - alice.its.cern.ch

Workflow: connecting and environment

- Login via SSH to a node:
  - cernuser@alivaf-{001,002…}.cern.ch
- Select AliRoot/AliPhysics from conf file:
  - export VafAliPhysicsVersion="vAN-20150630"
- Enter the VAF environment:
  - vaf-enter
- From this point on you have root/aliroot/etc. already set with the correct dependencies, and your AliEn token ready as well

Workflow: start and use PROOF

- Start your personal PROOF cluster:
  - vafctl start
- Request a certain number of PROOF workers (repeat to add):
  - vafreq 80
- Wait for some workers to be ready. Monitor with:
  - vafcount
- When you are satisfied, start your PROOF analysis:
  - root MyPoDMacro.C
- Do it many times. When done, free resources:
  - vafctl stop

Some more notes on the workflow

- Resources are also freed automatically when unused in case you forget to “stop”
- If there is a major problem, you can restart and reiterate the process (no need to contact the admins):
  - vafctl stop ; vafctl start ; vafreq 80 *# …and so on*
- Quotas are in place, so you are probably not going to get all the workers you ask for
- Assigned workers are CPUs dedicated to you, it could not be the case with old CAF
- You need to manually choose one of the available login nodes for now: we are working to make it automatic (active list in doc)

Outlook and TODOs

Dedicated storage pool with EOS

- Some data not at CERN: we don't want this to be a bottleneck
- Ideally: use an EOS cache to access data
  - xrootd interface or even FUSE mounted filesystem
  - Only link data already at CERN
  - Effectively mirror data outside CERN at first access
- Note: in principle the old CAF was designed to work like this…
  - Using xrootd's vMSS to cache data at first access
  - Execute nodes and storage nodes were the same: issues
  - Asynchronous data stager created: online access was unstable

Monitoring

- We are working on adding resources monitoring in the new CAF
- Some work has been done in Torino by S. Vallero
  - Using “ELK” (Elasticsearch + Logstash + Kibana)
    - We can maybe discuss on how to integrate it
- We could also use MonALISA (as for the old CAF):
  - Since worker nodes are “dynamic” and they have randomly assigned hostnames we could have some confusion in the logs
  - Old CAF monitoring not usable as-is: it requires some work

DDS

- The Dynamic Deployment System by A. Manafov and A. Lebedev
  - github.com/FairRootGroup/DDS
  - Under development: can cover PoD as a special case
  - “Successor” of PoD (but it aims to do much more)
- Immediate advantage: no more multiple login nodes, as the PROOF master can be distributed anywhere on the cluster
  - Only one scaling dimension (workers): topology simplified
- DDS will be used by FAIR and ALICE O^2
- It should be easily adapted to support PROOF
  - Real life test: happy to test it on the VAF before O^2 comes

Conclusions

The ALICE reference environment for analysis

- The Virtual CAF is our reference environment for non-Grid analysis
  - A well-defined execution environment
  - No compatibility problem: CernVM
  - No software distribution problem: CVMFS
    - Can be used without PROOF: HTCondor batch submission
- Self-consistency = perfect reproducibility in the future
  - Long term data preservation
  - CernVM snapshots: recover the exact same environment

Links

- CernVM elastic clusters
  - cernvm.cern.ch/portal/elasticclusters

- Virtual CAF at CERN: user and admin documentation
  - dberzano.github.io/alice/vcaf

- Issues
  - alice.its.cern.ch *(create them on the AAF project)*