Steering the Release Validation on the cloud

Dario Berzano

CERN

ALICE Offline Week - CERN, 25.06.2014

Part I.
Release Validation Cluster on the Cloud.

Introduction

Release Validation

- A series of batch jobs with dependencies
- Can run on every batch system: originally developed for GSI

- Trend: several computing resources nowadays available as cloud
  - At CERN: Agile Infrastructure
- The CernVM ecosystem features Elastic Clusters
  - Run a batch system on any cloud on virtual machines
  - One-click cluster deployment and scaling with no external tools

Run the validation on the cloud via CernVM ***Elastic Clusters***

CernVM Elastic Clusters

Your task | CernVM | HTCondor | elastiq

What is an Elastic Cluster?

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

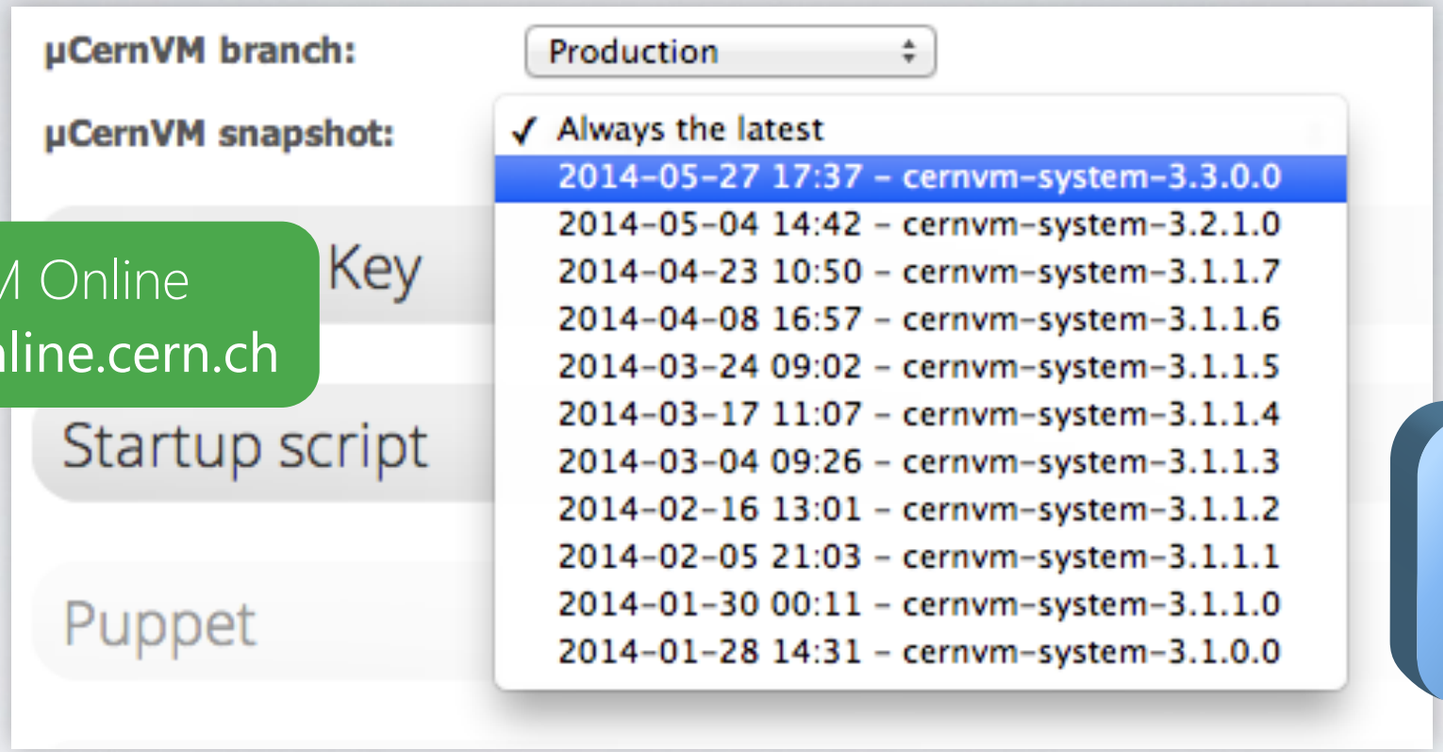
- Configured via a web interface: cernvm-online.cern.ch
- Entire cluster launched with a single command
- User interacts only by submitting jobs
- No external tools: embedded elasticity, ideal for opportunistic clouds

Release Validation cluster

Fully disposable *Elastic Cluster for running the Release Validation*

- No need to carry the software with the VMs
  - AliRoot versions to validate on CernVM-FS
- The cluster *(incl. the head node)* can be thrown away after use
  - Worker VMs automatically wiped out when validation completes
  - Output and log files stored on shared storage (EOS)
- Procedure fully repeatable
  - Cluster can be rebuilt using a configuration file
  - The very same environment can be restored as it was *(see next)*

Long Term Data Preservation



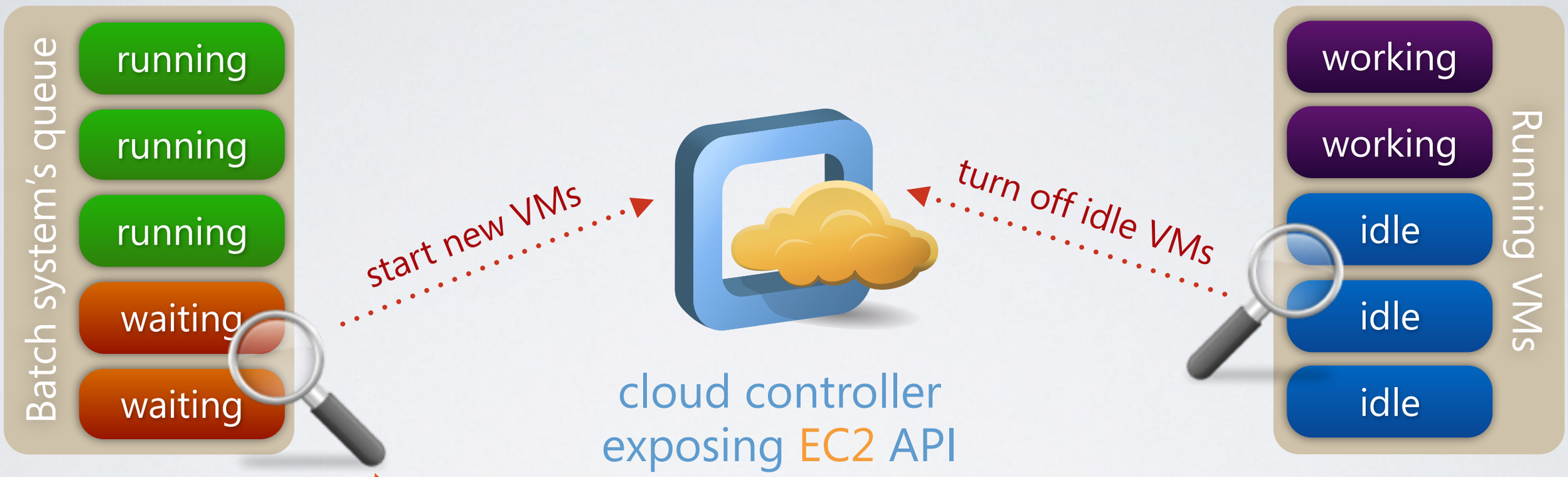
CernVM Online
cernvm-online.cern.ch

A time machine for selecting the exact version of the OS

- CernVM 3 allows for selecting any version of the OS from the past
- All versions will always be available via CVMFS which uses plain HTTP
- In 20 years from now we will be able to run an entire Release Validation Cluster with the same conditions of today

Automatic cluster deployment

elastiq: *app monitoring the queue to start/stop cluster VMs*



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

Very **robust** error handling

Can be used **outside CernVM** as well: RPMs: **github.com/dberzano/elastiq/releases**

Controlling the Release Validation

PWGPP/benchmark/benchmark.sh

- Author: Mikolaj Krzewicki
- Parameters from benchmark.config
  - fully documented
  - override with benchmark.config.d/*.config and command line
- List of input files in files.list

Generic script: currently used at **GSI** and **CERN**

- **Dependencies:** generates a deps file in Makeflow format *(see next)*
- **Jobs:** defines what each subjob does
- It controls the validation procedure **from the submission onwards**

Makeflow and Work Queue

- Tool to control and submit jobs with dependencies in the right order
- Submits on different systems: HTCondor, SGE, Work Queue…
- Work Queue workers themselves can run on top of batch systems
  - "Book" a certain number of job slots
  - Recycle a single job slot for many jobs

ccl.cse.nd.edu/software/{makeflow,workqueue}

- **At GSI:** Work Queue over SGE
- **On Elastic Cluster:** native HTCondor

Makeflow and Work Queue are preinstalled on **CernVM**

```
part1 part2 part3: input.data split.py
        ./split.py input.data

out1: part1 mysim.exe
        ./mysim.exe part1 >out1

out2: part2 mysim.exe
        ./mysim.exe part2 >out2

out3: part3 mysim.exe
        ./mysim.exe part3 >out3

result: out1 out2 out3 join.py
        ./join.py out1 out2 out3 > result
```

A Makeflow file resembles a Makefile

Part II.
Steering the validation on the cloud.

Before you begin

- Full documentation available here:
  https://dberzano.github.io/alice/release-validation
- Checklist covered by the documentation (to be done once):
  - Upload the CernVM image on your cloud
  - Obtain your EC2 API credentials
  - Embed a valid Grid Proxy *(needed for reading/writing on EOS)*
  - Create a Cluster profile on CernVM Online

Launch the validation

- Everything is under PWGPP/benchmark
- Default parameters in benchmark.config are OK in most cases

```
./alirelval --launch --aliroot vAN-20140623
```

What happens next?

- A session folder is created in ~/.alice-release-validation: it contains all needed files to repeat the same validation at a later time
- A SSH keypair is created to access the virtual machines
- The head node virtual machine is launched: it will launch the workers
- Necessary files are transferred to the VM via SSH
- Validation is launched via SSH in a "detached screen"

Check the validation progress

- Use the session ID for referring to the appropriate validation
- GNU screen is resilient: validation continues if SSH connection breaks

```
./alirelval [ --status | --attach | --shell ]
```

Operations

- You can choose the appropriate session ID interactively or with the switch **--session vAN-20140623_20140624-112941-utc**
- **--status:** tells if validation has finished and if it was successful
- **--attach:** attaches the validation terminal *(detach with Ctrl+A+D)*
- **--shell:** opens a shell to the virtual machine for debug
- It is not even needed to login to check the status

When validation has finished

- All virtual cluster workers are wiped out automatically
- Only the head node is left there
  - It can be reused for further validations
  - Useful for debug
- All output and log files are copied to EOS in a known directory
  - They can be downloaded from everywhere
  - "Paranoid copy" mode to work around temporary storage failures

Conclusions

- We have some dedicated cloud resources at CERN (two tenants):
  - ALICE Release Testing: 200 CPUs, 600 GB RAM
  - ALICE Cloud Tests: 400 CPUs, 800 GB RAM
- We have a working benchmark
- We have a tool to steer the validation on the cloud
- We have the possibility to "certify" AliRoot versions to CernVM snapshots for Long-Term Data Preservation

Thank you!

Links

- AliRoot Release Validation on the Cloud
  dberzano.github.io/alice/release-validation

- CERN Agile Infrastructure user guide
  information-technology.web.cern.ch/book/cern-private-cloud-user-guide

- Elastic Clusters with CernVM
  cernvm.cern.ch/portal/elasticclusters

- Configure a CernVM Virtual Machine and Cluster
  cernvm-online.cern.ch

- elastiq: the engine of Elastic Clusters
  github.com/dberzano/elastiq