



# Singularity au CC

Vanessa HAMAR

- ▶ Pourquoi utiliser les containers ?
- ▶ Notre première expérience avec les containers dans la plateforme de calcul
- ▶ Singularity au CC
- ▶ Quelques tests des expériences

# Pourquoi les containers ?

- ▶ “Containers are a solution to the problem of how to get software to run reliably when moved from one computing environment to another. This could be from a developer's laptop to a test environment, from a staging environment into production and perhaps from a physical machine in a data center to a virtual machine in a private or public cloud.”\*

[\\*cio.com](http://*cio.com)



- ▶ Univa Grid Engine + Docker
  - Version Grid Engine 8.4.0 ~ May, 2016
  - Docker micro services
- ▶ Les pros
  - Configuration UGE -> Resource
  - Facile soumission
- ▶ Les contres
  - Le /var plein après quelques jobs exécutés (ferme de test)
  - Beaucoup de questions liées à la sécurité ...
  - Gérer le réseau



**Conclusion : nous continuons à chercher ...**

- ▶ <http://geekyap.blogspot.fr/2016/11/docker-vs-singularity-vs-shifter-in-hpc.html>
- ▶ ~ Nov 2016

## ▶ **Docker :**

- Micro services.
- Enterprise applications.
- Développeurs/DevOps



## ▶ **Shifter :**

- Utilise un grand nombre d'applications docker.
- Fournit un moyen de les exécuter dans HPC après un processus de conversion.
- Il supprime également toutes les exigences de root afin qu'ils soient exécutables sous l'identité des utilisateurs.
- Utilisateurs des applications scientifiques



## ▶ **Singularity :**

- Portabilité des applications (fichier image unique, contenant toutes les dépendances)
- Reproductibilité, exécution multiplateforme, prise en charge des systèmes hérités et des applications.
- Utilisateurs de l'applications scientifiques





Singularity est une solution de conteneurisation, créée pour répondre aux besoins des applications scientifiques

# Containers pour la Science !!!

Voilà !!! 

- ▶ Soumettre un job dans la ferme de test.

### A simple script:

```
mon_job_singularity.sh
```

```
#!/bin/bash
```

```
singularity exec $HOME/sl6 $HOME/my_script.sh
```

### Normal submission:

```
> qsub -q long@ccwtb2sge001 -l os=cl7 mon_job_singularity.sh
```

- ▶ Pas de changement dans la configuration du batch ✓
- ▶ Pas besoin de gérer le réseau ✓
- ▶ L'utilisateur qui soumet le job est l'utilisateur dans le container ✓



- ▶ Singularity est disponible depuis quelques mois sur l'ensemble des interactives et nœuds de calcul de nos fermes (CentOS 7)
  - Version **singularity-2.5.1-1.el7.x86\_64**
  - Déployé sous le mode setUID.
  - Module puppet pour gérer la version et la configuration.

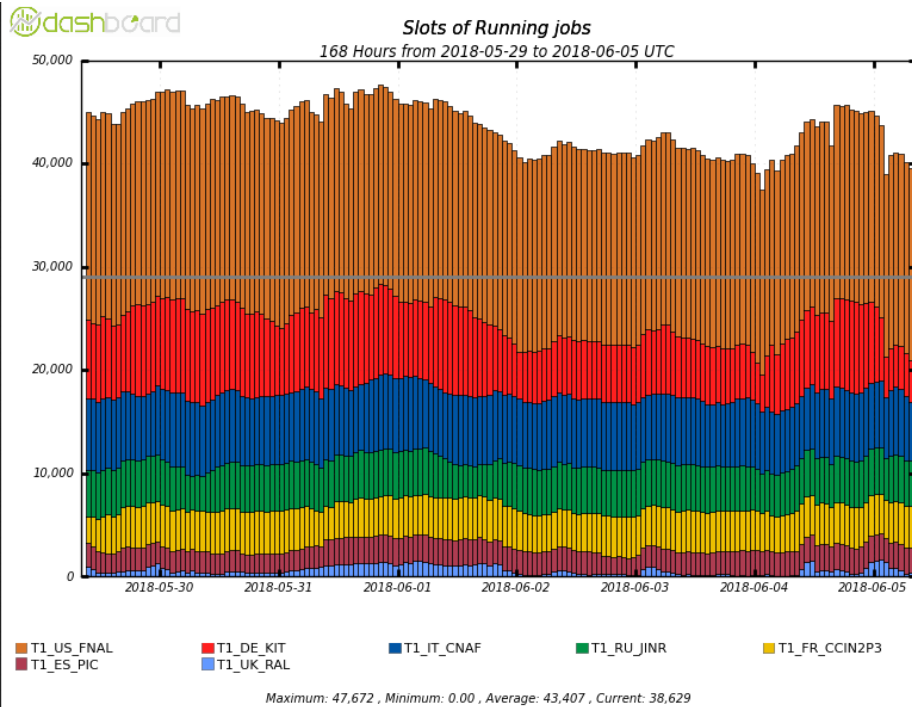
- ▶ Dans la [présentation](#) de Brian Bockelman, pre-GDB, July 2017 :
  - Given our heavy investment in CVMFS, it seems very natural to leverage it for image distribution.
  - Given CVMFS implementation details, images should be distributed as flat directories - Cache will work at the individual file level.

- ▶ Certaines applications/expériences qui sont organisées au niveau mondial proposent leurs propres images via le système CVMFS.
  - C'est notamment le cas de l'expérience CMS.
  - Cette expérience exécute quotidiennement des centaines de tâches avec Singularity.
- ▶ Au CC nous avons un repo CVMFS avec un ensemble d'images "de bases" :
  - Debian
  - SL6
  - CentOS7
  - Ubuntu

- ▶ OSG-SEC-2018-03-08 Limiting Singularity image types
- ▶ OSG-SEC-2018-04-30 Multiple Singularity Vulnerabilities
- ▶ OSG-SEC-2018-04-18 [UPDATE] Critical Vulnerability in Singularity
- ▶ OSG-SEC-2018-04-02 Critical Vulnerability in Singularity
- ▶ EGI SVG 'ADVISORY' [TLP:AMBER] 'CRITICAL' risk. Local privilege escalation using singularity. (2018-03-29)
- ▶ OSG-SEC-2018-03-08 Limiting Singularity image types
- ▶ OSG-SEC-2018-03-05 Vulnerability in Singularity

Beaucoup d'alertes de sécurité en peu du temps ....

# Singularity tests



- ▶ CMS, une des quatres expériences LHC, utilise déjà Singularity pour exécuter sa production.
- ▶ CMS à demandé d'installer Singularity dans les worker-nodes pour Mars 2018.
- ▶ Test par l'expérience CMS de la disponibilité de Singularity
  - Ce test est jugé critique du coté applicatif.

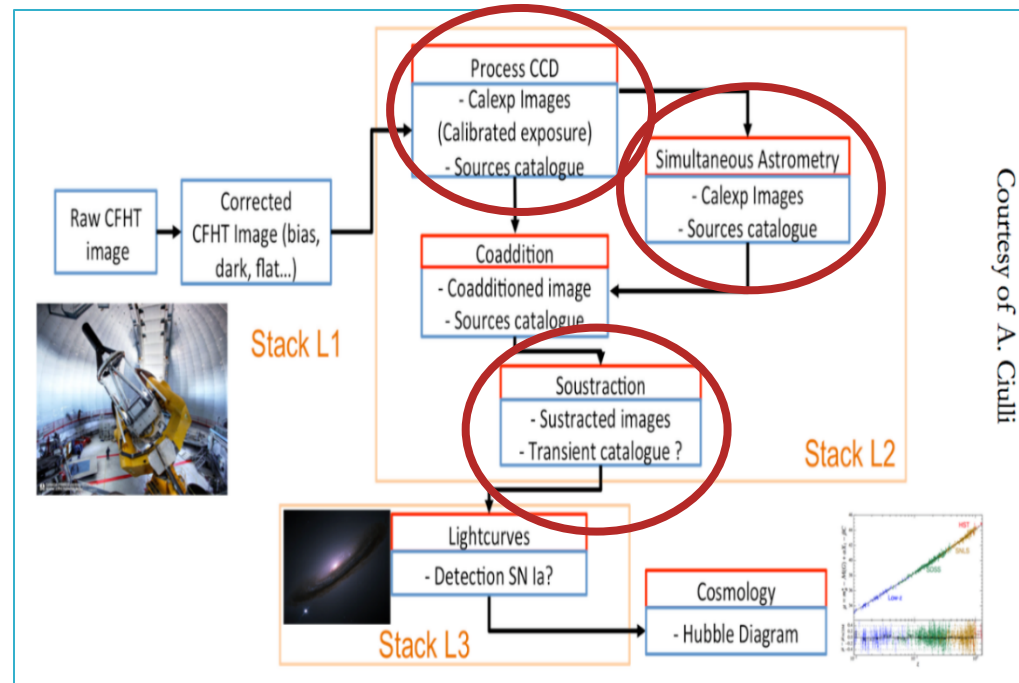
## Detailed output of Metric Result

Field	Value
Hostname	cccreamcel104.in2p3.fr
Metric	org.cms.WN-isolation
VOFQAN	/cms/Role=pilot
Service Flavour	CREAM-CE
Timestamp	2018-05-30T14:26:02Z
Status	OK
Summary	ccwsgel102: OK: OK
Details	<pre>/usr/bin/singularity exec --home /scratch/43018761.1.mc_long/tmp.YVSMhODsp4:/srv --bind /cvmfs --pwd /srv --contain --ipc --pid /cvmfs/singularity.opensciencegrid.org/bbockelm/cms:rhel6 echo Hello World OR grep Hello World WARNING: Container does not have an exec helper script, calling 'echo' directly Hello World</pre>

# Use Case LSST

Merci a Bertrand RIGAUD pour les slides

- ▶ Singularity est également utilisé au CC pour tester et valider des workflows applicatifs ou bien réaliser des challenges.
- ▶ LSST – Deep Learning Challenge
  - Ici, Singularity a permis de maîtriser l'environnement d'exécution.
  - Les frameworks nécessaires au code « machine learning » sont très dépendants des bibliothèques GPU. Singularity a permis de dé-corréler l'aspect implémentation des bibliothèques GPU des applicatifs (tensor flow)
- ▶ LSST : validation du workflow et impact sur les performance de stockage.



## Facts

- ▶ Tensorflow and Cuda / Cudnn versions increase rapidly (about a release per trimester)

## Benchmark

- ▶ Done with Tensorflow 1.3 (no problem)
- ▶ Want now to check again with Tensorflow 1.6 and see differences



## GPU Farm

- ▶ Cuda 8.0
- ▶ Cudnn 6.0

## Problem

- ▶ Tensorflow 1.6 requires Cuda 9.0 and Cudnn 7.0

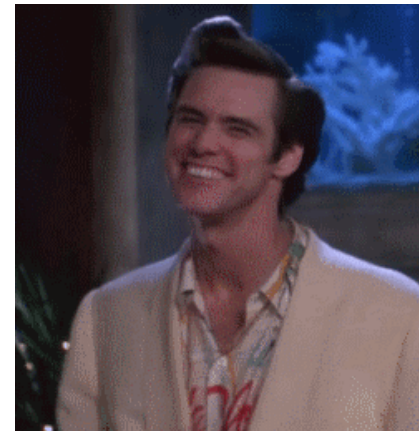
## Solution 1

- ▶ Ask SysAdmins to update farm



## Solution 2

- ▶ Use Singularity



## Moreover

- ▶ I want to benchmark with a compiled from sources version of Tensorflow so that it uses all hardware capabilities



## Step 1

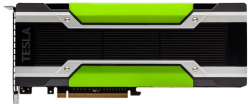
- ▶ Build a compiler image



- ▶ CentOS 7
- ▶ Cuda 9.0 / Cudnn 7.0 libraries
- ▶ Compiler environment (Bazel)
- ▶ Python environment (Miniconda)

## Step 2

- ▶ Execute compiler image from a gpu worker (interactive or qsub)



- ▶ CentOS 7
- ▶ Cuda 9.0 / Cudnn 7.0 libraries
- ▶ Compiler environment (Bazel)
- ▶ Python Environment (Miniconda)



- ▶ Git clone Tensorflow
- ▶ Compile



tensorflow-1.6.0-cp36-cp36m-linux\_x86\_64.whl

## Step 3

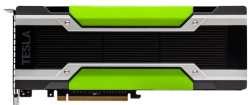
- ▶ Build a production image



- ▶ CentOS 7
- ▶ Cuda 9.0 / Cudnn 7.0 libraries
- ▶ Python Environment (Miniconda)
- ▶ Pip install tensorflow-1.6.0-cp36-cp36m-linux\_x86\_64.whl

## Step 4

- ▶ Execute production image from a gpu worker (interactive or qsub)



- ▶ CentOS 7
- ▶ Cuda 9.0 / Cudnn 7.0 libraries
- ▶ Compiler environment (Bazel)
- ▶ Python Environment (Miniconda)
- ▶ Tensorflow 1.6 (from sources)



- ▶ Benchmark



- ▶ Speed gain 15-20% between Tensorflow 1.6 and Tensorflow 1.6 (from sources)

## ▶ Links:

- [Singularity - User guide](#)
- [HPC Containers singularity](#)
- [Docker vs Singularity vs Shifter](#)

## ▶ Mailing lists:

- “wlcg-containers (WLCG container working group)”  
[wlcg-containers@cern.ch](mailto:wlcg-containers@cern.ch)
  - Wed, June 27th (ie. next week).
  - Wed July 11th
- “Singularity” [singularity@lbl.gov](mailto:singularity@lbl.gov)

## ▶ Groups

- [Conteneurs IN2P3](#)
- C3 – CC Containers

- ▶ A very nice idea 😊

## Pour les utilisateurs

- ▶ Facile à utiliser
- ▶ Bonne documentation
- ▶ Une mailing liste réactive

## Pour les administrateurs système

- ▶ Un nouveau projet qui avance vite
- ▶ Quelques problèmes de compatibilité avec les anciennes versions
- ▶ Attention aux alertes de sécurité



