500,000 Container Instances a Day?

Open Science Grid Singularity Infrastructure

Mats Rynge <rynge@isi.edu>
USC Information Sciences Institute
OSG Campus Research

Opening Questions

- What is the *Open Science Grid (OSG)*?
- What is *high throughput computing (HTC)*?
- Why have containers become an important part of the OSG infrastructure?

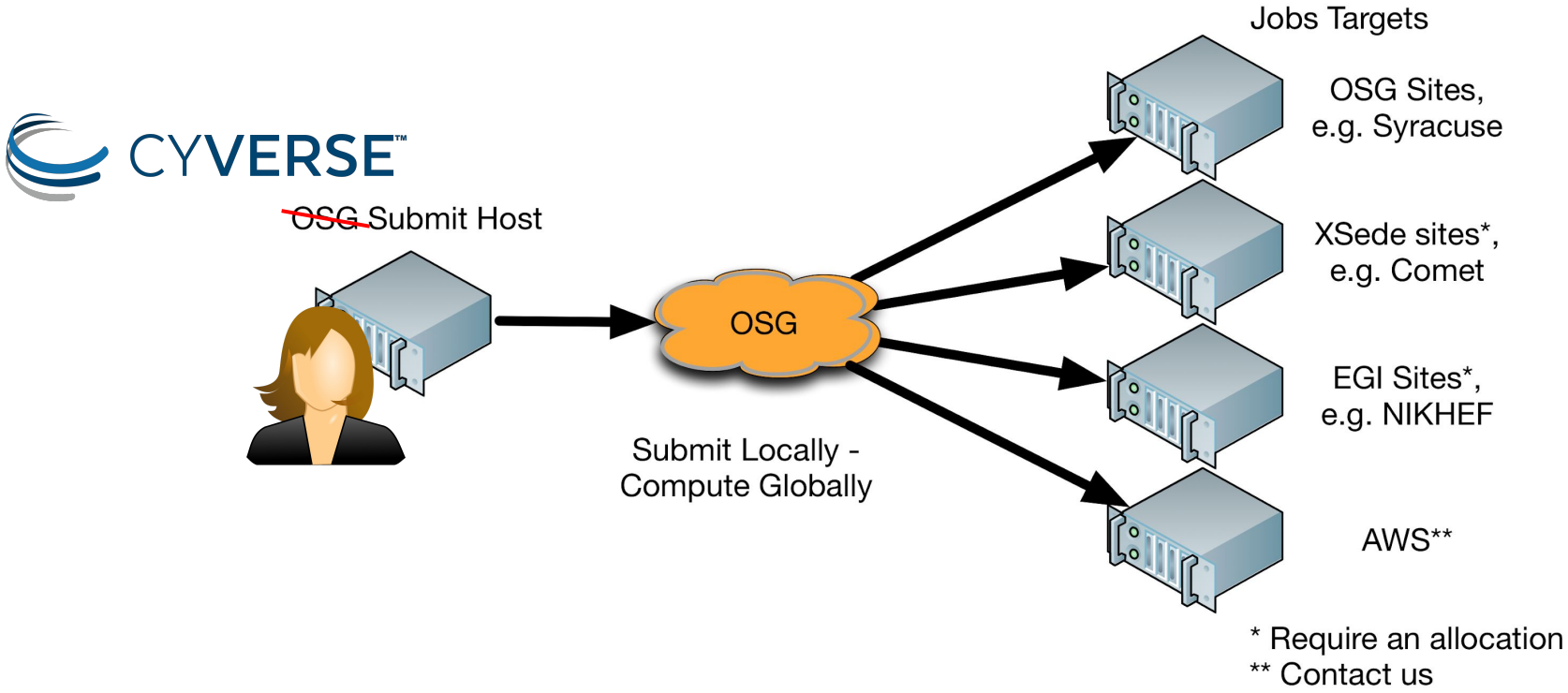
Open Science Grid

A framework for large scale distributed resource sharing addressing the technology, policy, and social requirements of sharing computing resources.

OSG is a **consortium** of software, service and resource providers and researchers, from universities, national laboratories and computing centers across the U.S., who together build and operate the OSG project. The project is funded by the NSF and DOE, and provides staff for managing various aspects of the OSG.

Integrates computing and storage resources from over **100 sites** in the U.S.

“Submit Locally, Run Globally”



Open Science Grid

In the last 24 Hours	
259,000	Jobs
3,660,000	CPU Hours
6,491,000	Transfers
1,332	TB Transfers
In the last 30 Days	
7,591,000	Jobs
132,380,000	CPU Hours
159,507,000	Transfers
28,623	TB Transfers
In the last 12 Months	
108,590,000	Jobs
1,650,606,000	CPU Hours
1,848,545,000	Transfers
246,000	TB Transfers

~ 3.6 million CPU hours delivered per day

High Throughput Computing

High Throughput Computing

Sustained computing over long periods of time. Usually serial codes, or low number of cores threaded.

vs. High Performance Computing

Great performance over relatively short periods of time. Large scale MPI.

***Distributed* HTC**

No shared file system

Users ship input files and (some) software packages with their jobs.

Opportunistic Use

Applications (esp. with long run times) can be *preempted* (or killed) by resource owner's jobs.

Applications should be relatively short or support being restarted.

HTC: An Analogy

Question: How do you bake the world's largest cake?

HTC: An Analogy

Answer: HTC-Style!

Many small cakes baked separately, joined together

Distributed High Throughput ("DHTC")

High Throughput Computing

Sustained computing over long periods of time. Usually ser
threaded/MPI.

vs. High Performance Computing

Great performance over relative short periods of time. Larg

***Distributed* HTC**

No shared file system

Users ship input files and (some) software packages with their jobs.

Opportunistic Use

Applications (esp. with long run times) can be *preempted* (or killed) by resource owner's jobs.
Applications should be relatively short or support being restarted.

DHTC Jobs

High Throughput Computing

Sustained computing over long periods of time. Usually serial codes, or low number of cores threaded/MPI.

vs. High Performance Computing

Great performance over relative short periods of time. Large scale MPI.

***Distributed* HTC**

No shared file system

Users ship input files and (some) software packages with their jobs.

Opportunistic Use

Applications (esp. with long run times) can be *preempted* (or killed) by resource owner's jobs.

Applications should be relatively short or support being restarted.

Container Motivations

- **Consistent environment (default images)** - If a user does not specify a specific image, a default one is used by the job. The image contains a decent base line of software, and because the same image is used across all the sites, the user sees a more consistent environment than if the job landed in the environments provided by the individual sites.

- **Custom software environment (user defined images)** - Users can create and use their custom images, which is useful when having very specific software requirements or software stacks which can be tricky to bring with a job. For example: Python or R modules with dependencies, TensorFlow

- **Enables special environment such as GPUs** - Special software environments to go hand in hand with the special hardware.

- **Process isolation** - Sandboxes the job environment so that a job can not peek at other jobs.

- **File isolation** - Sandboxes the job file system, so that a job can not peek at other jobs' data.

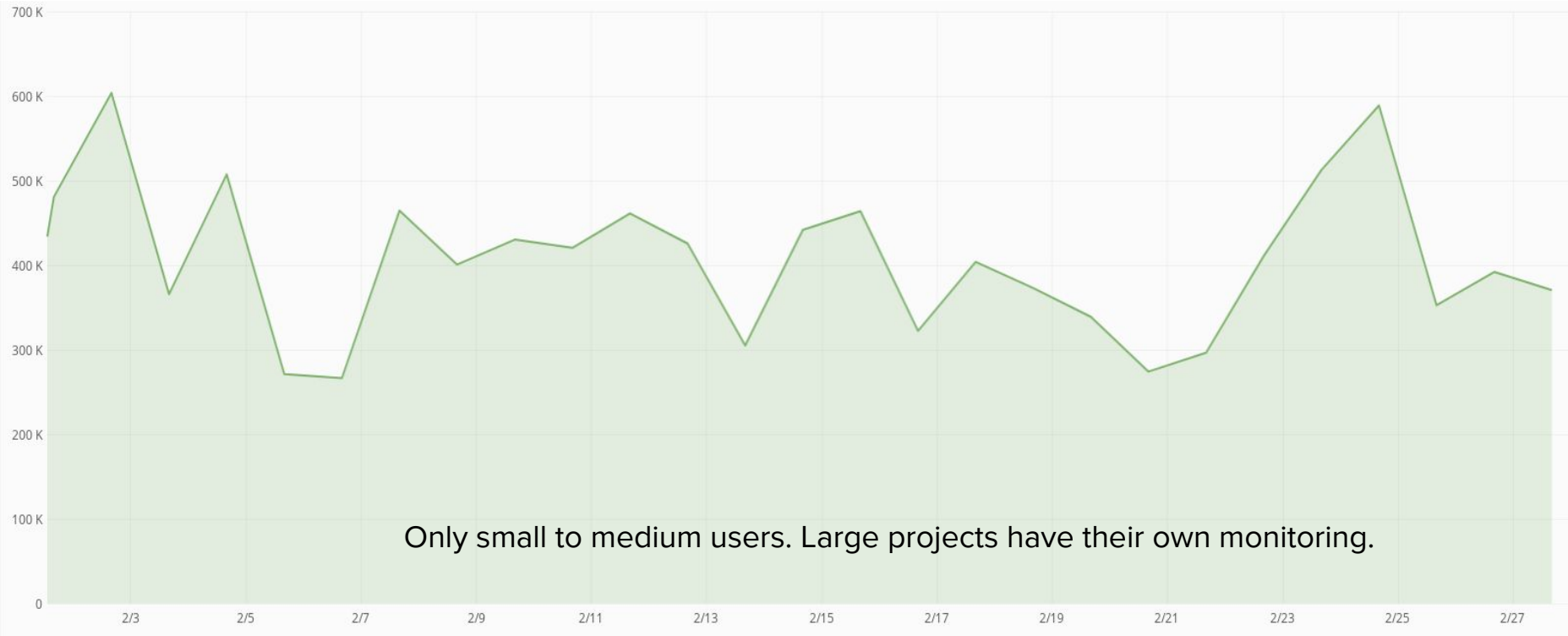
Container Lifecycle (Hint: ephemeral)

Each and every job is encapsulated in a separate container instance

Container instance dies when the job finishes

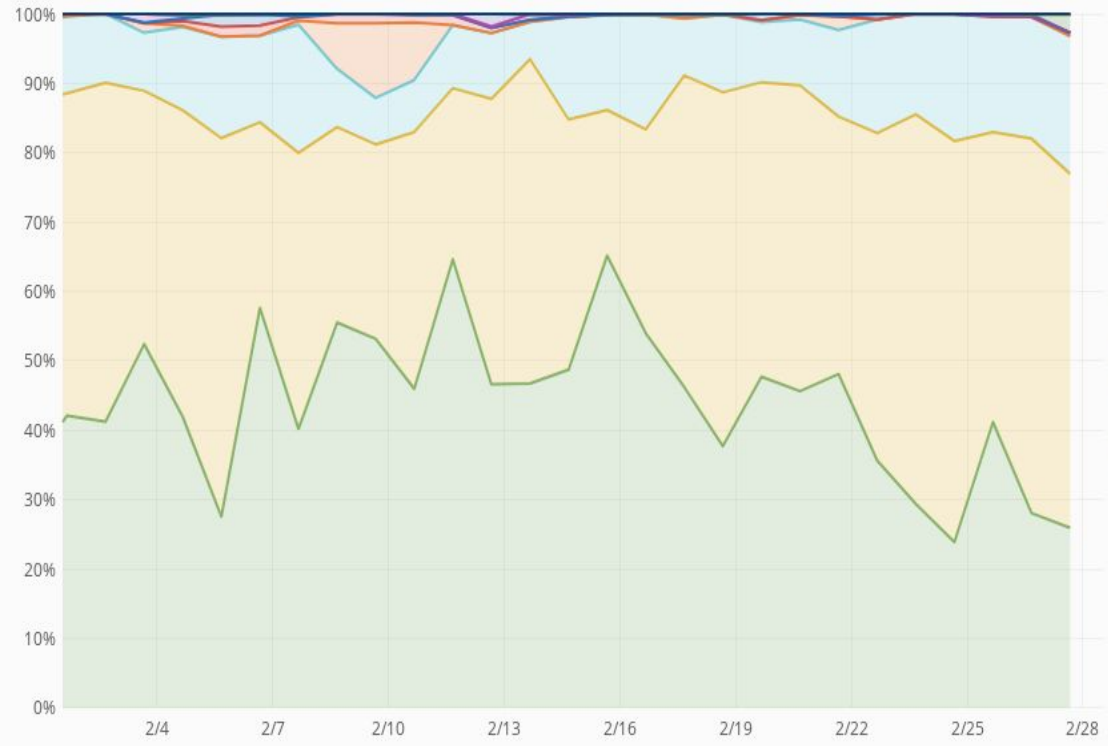
Lot’s of container image reuse, as workloads generally use one or a small number of images for a large number of jobs

Container Instances per Day



Only small to medium users. Large projects have their own monitoring.

Container Breakdown



	total
None	4819465
/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest	4602520
/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el6:latest	1362363
/cvmfs/singularity.opensciencegrid.org/agladstein/msprime:latest	124086
/cvmfs/singularity.opensciencegrid.org/markito3/gluex_docker_devel:latest	48284
/cvmfs/singularity.opensciencegrid.org/bbockelm/cms:rhel6	19991
/cvmfs/singularity.opensciencegrid.org/drtmfigy/herwig_hjets:latest	15576
/cvmfs/singularity.opensciencegrid.org/kai2019/osg-fsl:latest	10427
/cvmfs/singularity.opensciencegrid.org/rynge/osg-mcf10-mod:latest	10069
/cvmfs/singularity.opensciencegrid.org/pycbc/pycbc-el7:v1.13.1	1490
/cvmfs/ligo-containers.opensciencegrid.org/lscsoft/bayeswave/production	698
/cvmfs/ligo-containers.opensciencegrid.org/lscsoft/bayeswave/latest	61
/cvmfs/singularity.opensciencegrid.org/discoenv/osg-word-count:1.0.0	46
http://xd-login.opensciencegrid.org/scratch/eemt/singularity/eemt-current.img	38

500,000 containers x 5 GB (average size of container) =

2.5 PB / day

We need an efficient way to distribute containers!

CVMFS - CERN Virtual Machine File System

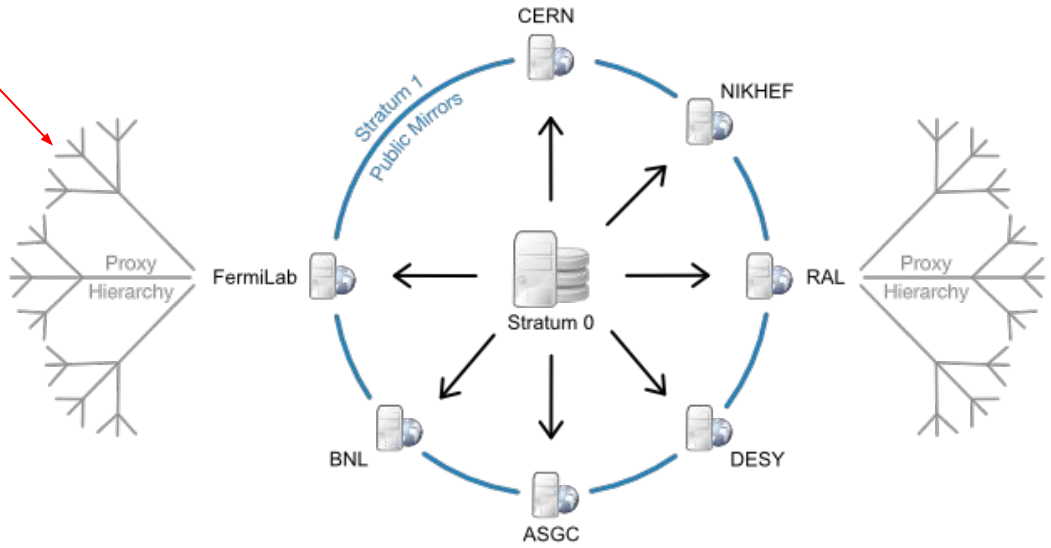
"The CernVM File System provides a scalable, reliable and low-maintenance software distribution service. It was developed to assist High Energy Physics (HEP) collaborations to deploy software on the worldwide-distributed computing infrastructure used to run data processing applications. CernVM-FS is implemented as a POSIX read-only file system in user space (a FUSE module). Files and directories are hosted on standard web servers and mounted in the universal namespace /cvmfs."

Your job is here!

Used for software and data!

Heavily cached, read-only

Available across OSG, EGI, some XSEDE resources



CVMS Repositories

```
/cvmfs/
        ams.cern.ch
        atlas.cern.ch
        cms.cern.ch
        connect.opensciencegrid.org
        gwosc.osgstorage.org
        icecube.opensciencegrid.org
        ligo-containers.opensciencegrid.org          <- large project with their own containers
        nexo.opensciencegrid.org
        oasis.opensciencegrid.org                    <- “modules” software
        singularity.opensciencegrid.org              <- general containers (next few slide)
        snoplus.egi.eu
        spt.opensciencegrid.org                <- project from talk yesterday (South Pole Telescope)
        stash.osgstorage.org                         <- ~1PB of user published data
        veritas.opensciencegrid.org
        xenon.opensciencegrid.org
```

Available Containers

	Image Location	Definti on	Description
EL 6	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el6:latest	GitHub	A basic Enterprise Linux (CentOS) 6 based image.
EL 7	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest	GitHub	A basic Enterprise Linux (CentOS) 7 based image.
Ubuntu Xenial	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-ubuntu-xenial:latest	GitHub	A good image if you prefer Ubuntu over EL flavors
Ubuntu 18.04 (Bionic)	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-ubuntu-18.04:latest	GitHub	A good image if you prefer Ubuntu over EL flavors
TensorFlow	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/tensorflow:latest	GitHub	Base on the TensorFlow base image, with a few OSG package added
TensorFlow GPU	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/tensorflow-gpu:latest	GitHub	Used for running TensorFlow jobs on OSG GPU resources

~150 images, consisting of pre-defined ones by OSG staff, base images from Docker (different OSes, Python, r-base, ...) and custom images by our users

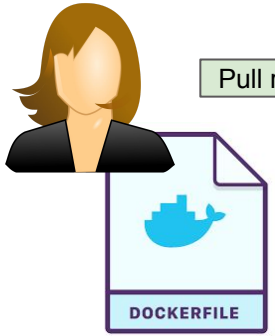
cvmfs-singularity-sync

- Containers are defined using Docker
  - Public Docker Hub

- ... and executed with Singularity
  - No direct access to the Singularity command line - that is controlled by the infrastructure

- https://github.com/opensciencegrid/cvmfs-singularity-sync (next slide)

User-defined Container Workflow



Pull request or ticket to register container (one time)

GitHub

docker_images.txt
cvmfs-singularity-sync

docker build + push

git push

GitHub

Automatic builds on changes

docker Hub

Automatic sync on changes

/cvmfs

Extracted Images

OSG stores container images on CVMFS in extracted form. That is, we take the Docker image layers or the Singularity img/simg files and export them onto CVMFS. For example, ls on one of the containers looks similar to ls / on any Linux machine:

```
$ ls /cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest/
cvmfs   host-libs   proc   sys   anaconda-post.log     lib64
dev     media       root   tmp   bin                   sbin
etc     mnt         run    usr   image-build-info.txt  singularity
home    opt         srv    var   lib
```

Result: Most container instances only use **a small part** of the container image **(50-150 MB)** and that part is **cached** in CVMFS!

Summary

Containers have enabled OSG to provide a **consistent environment** across a large number of contributed compute resources, as well as provide a mechanisms for users to bring their own **custom environments**.

More information:
https://support.opensciencegrid.org/support/solutions/articles/12000024676-docker-and-singularity-containers

(short url: https://goo.gl/Yq9CYH)