Singularity

Containers In HPC

Michael Bauer

About Me

Michael Bauer

@bauerm97 on GitHub
bauerm@umich.edu

singularity.lbl.gov

Singularity

Singularity enables users to have full control of their environment. This means that a non-privileged user can “swap out” the operating system on the host for one they control. So if the host system is running RHEL6 but your application runs in Ubuntu, you can create an Ubuntu image, install your applications into that image, copy the image to another host, and run your application on that host in it’s native Ubuntu environment!

Register your Cluster | Add a Publication

Singularity also allows you to leverage the resources of whatever host you are on. This includes HPC interconnects, resource managers, file systems, GPUs and/or accelerators, etc. Singularity does this by enabling several key facets:

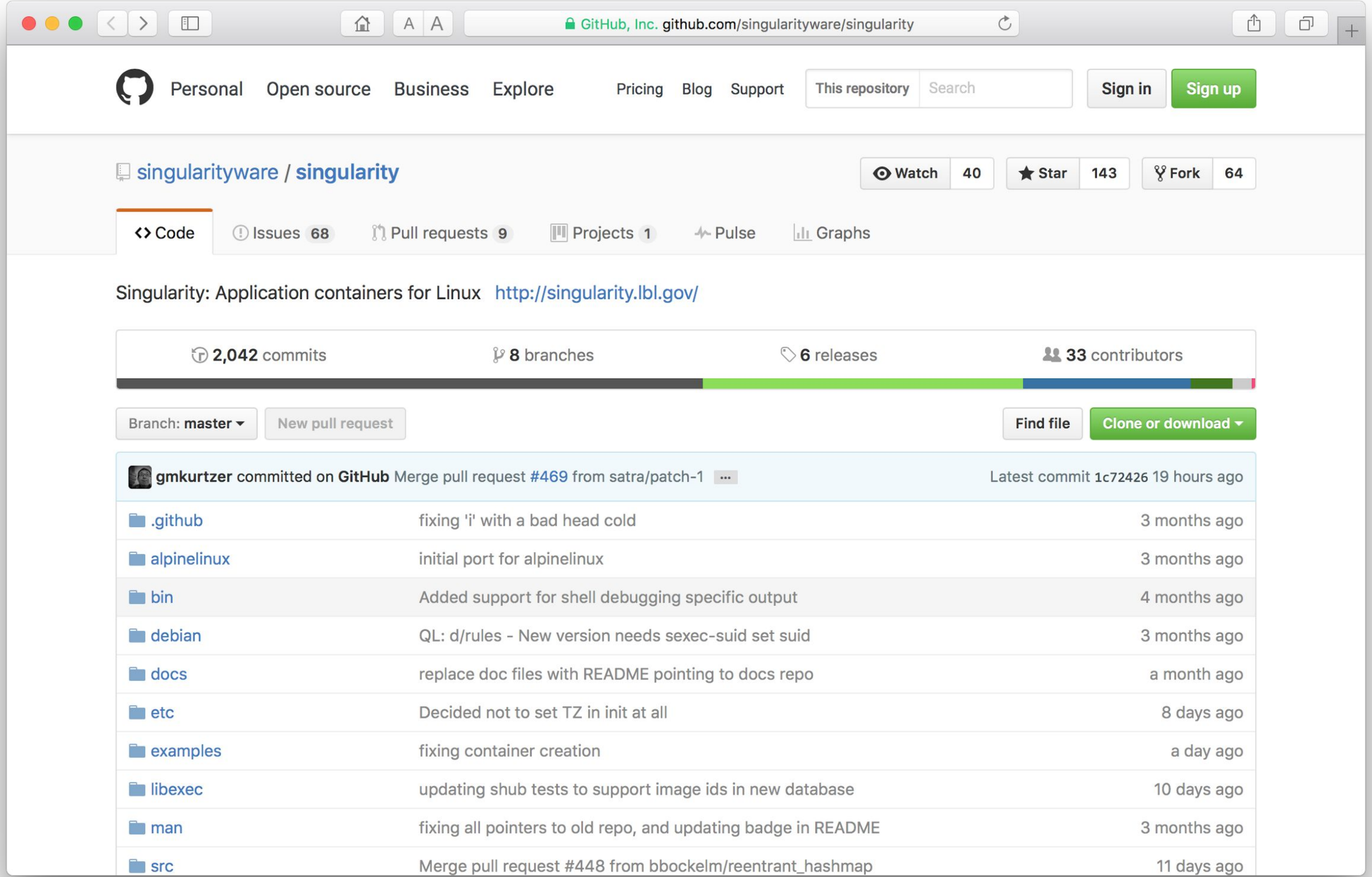
- Encapsulation of the environment
- Containers are image based
- No user contextual changes or root escalation allowed
- No root owned daemon processes

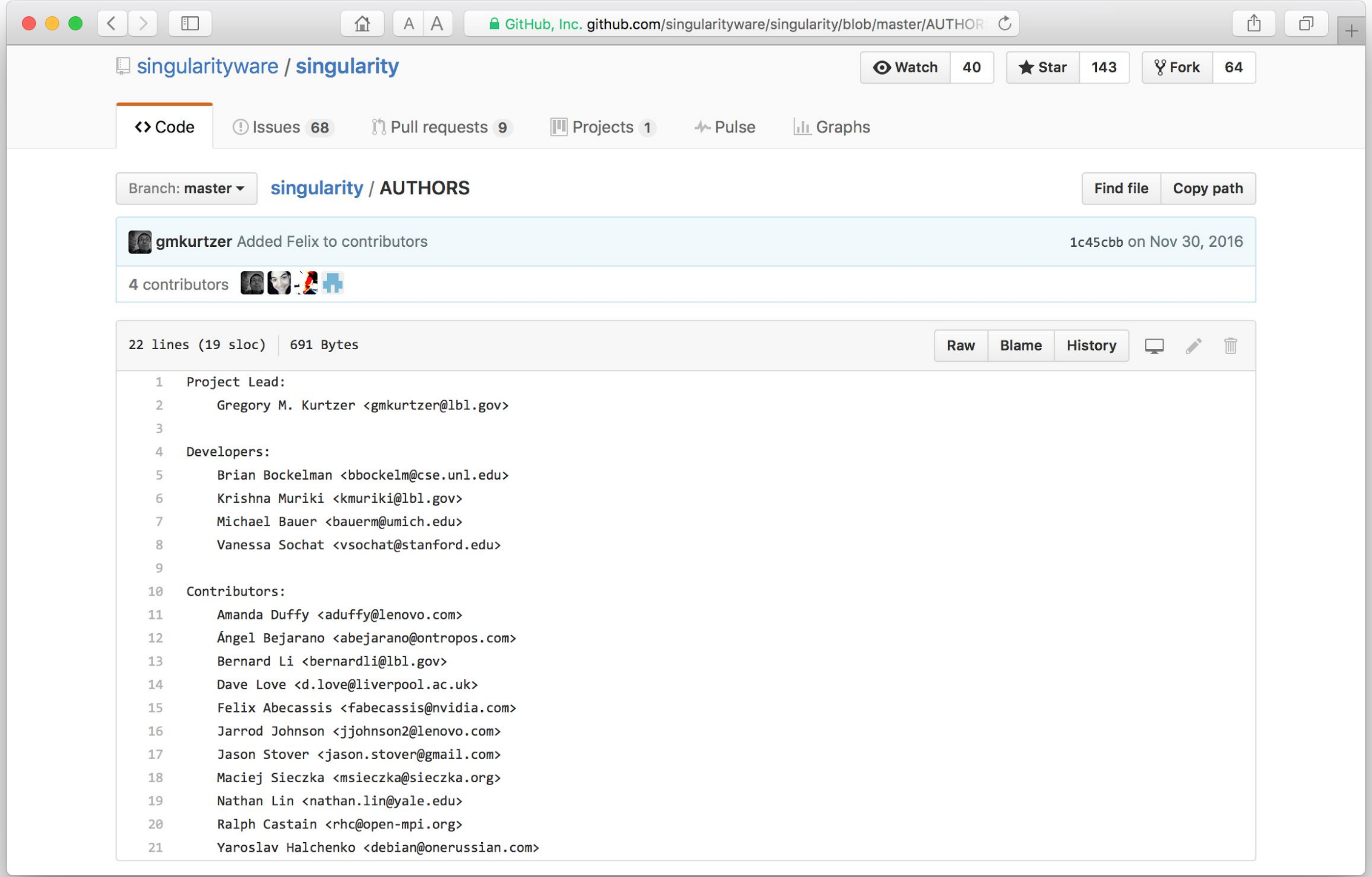
Getting started

Jump in and get started.

Singularity

- Information
- Download / Installation
- Contributing
- Getting Help
- Documentation


GitHub, Inc. github.com/singularityware/singularity
Personal
Open source
Business
Explore
Pricing
Blog
Support
This repository
Search
Sign in
Sign up
singularityware / singularity
Watch 40
Star 143
Fork 64
Code
Issues 68
Pull requests 9
Projects 1
Pulse
Graphs
Singularity: Application containers for Linux http://singularity.lbl.gov/
2,042 commits
8 branches
6 releases
33 contributors
Branch: master
New pull request
Find file
Clone or download
gmkurtzer committed on GitHub Merge pull request #469 from satra/patch-1 …
Latest commit 1c72426 19 hours ago
.github
fixing 'i' with a bad head cold
3 months ago
alpinelinux
initial port for alpinelinux
3 months ago
bin
Added support for shell debugging specific output
4 months ago
debian
QL: d/rules - New version needs sexec-suid set suid
3 months ago
docs
replace doc files with README pointing to docs repo
a month ago
etc
Decided not to set TZ in init at all
8 days ago
examples
fixing container creation
a day ago
libexec
updating shub tests to support image ids in new database
10 days ago
man
fixing all pointers to old repo, and updating badge in README
3 months ago
src
Merge pull request #448 from bbockelm/reentrant_hashmap
11 days ago



GitHub, Inc. github.com/singularityware/singularity/blob/master/AUTHOR
singularityware / singularity
Watch 40
Star 143
Fork 64
Code
Issues 68
Pull requests 9
Projects 1
Pulse
Graphs
Branch: master
singularity / AUTHORS
Find file
Copy path
gmkurtzer Added Felix to contributors
1c45cbb on Nov 30, 2016
4 contributors
22 lines (19 sloc) 691 Bytes
Raw
Blame
History
Project Lead:
Gregory M. Kurtzer <gmkurtzer@lbl.gov>
Developers:
Brian Bockelman <bbockelm@cse.unl.edu>
Krishna Muriki <kmuriki@lbl.gov>
Michael Bauer <bauerm@umich.edu>
Vanessa Sochat <vsochat@stanford.edu>
Contributors:
Amanda Duffy <aduffy@lenovo.com>
Ángel Bejarano <abejarano@ontropos.com>
Bernard Li <bernardli@lbl.gov>
Dave Love <d.love@liverpool.ac.uk>
Felix Abecassis <fabecassis@nvidia.com>
Jarrod Johnson <jjohnson2@lenovo.com>
Jason Stover <jason.stover@gmail.com>
Maciej Sieczka <msieczka@sieczka.org>
Nathan Lin <nathan.lin@yale.edu>
Ralph Castain <rhc@open-mpi.org>
Yaroslav Halchenko <debian@onerussian.com>

What are Containers?

What is a Virtual Machine?

“In computing, a virtual machine (VM) is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer.”

Examples:

Pros

- Run different OS on one set of hardware
- Save money (e.g. buy one laptop, have Windows, OSX, and Linux)
- Easy maintenance

Cons

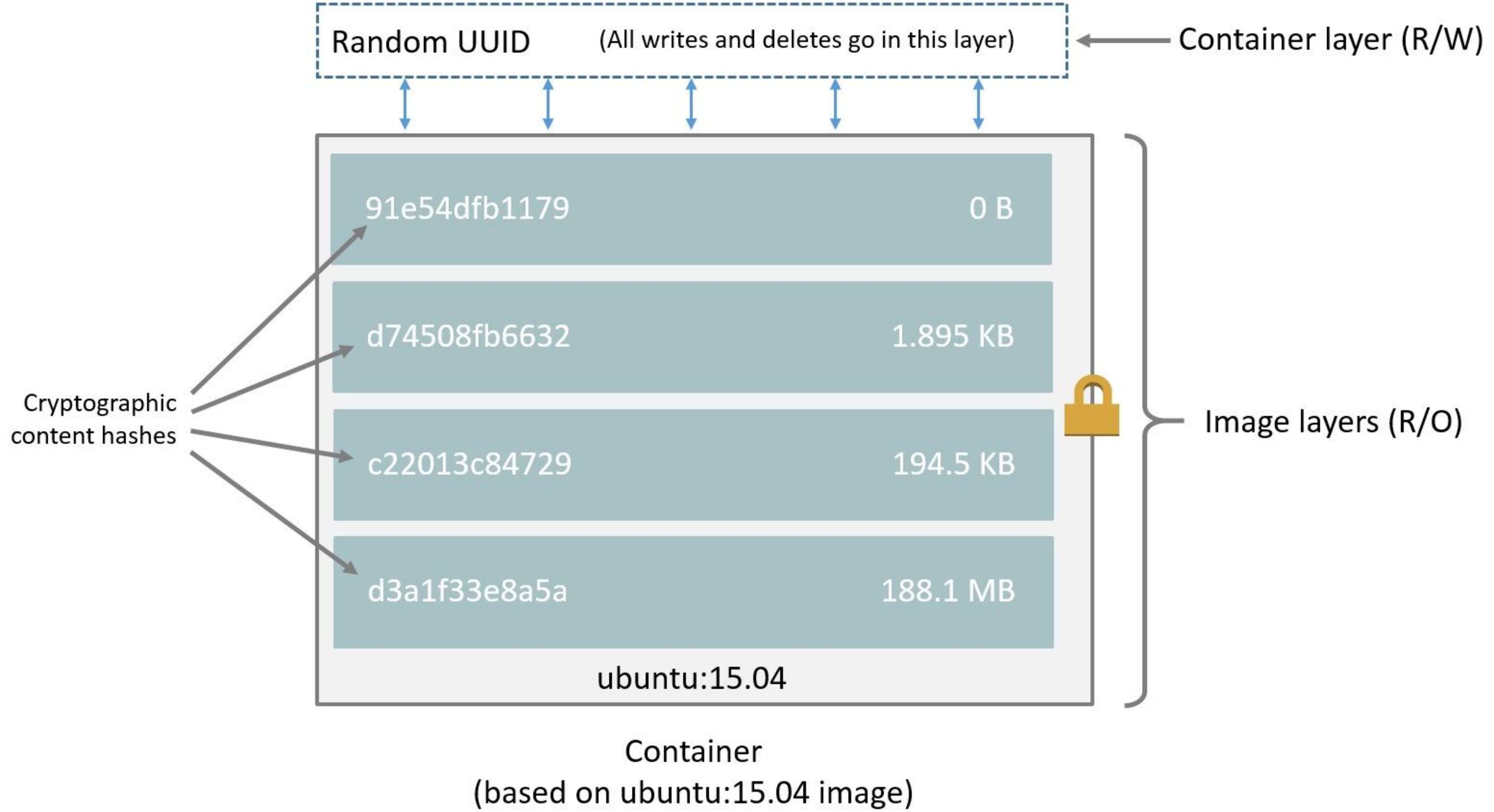
- Slower performance
- Memory/storage reqs.

What are containers?

- Similar goal as VMs
- No kernel emulation
- Not architecture level virtualization, but rather software level

What does that mean?

- Don’t waste extra ~5% performance doing emulation
- Smaller footprint (~500 MB vs ~20 GB VM)
- Very small startup time interval (~1 s vs ~1 min VM)
- Multiple instances can share one “container image”

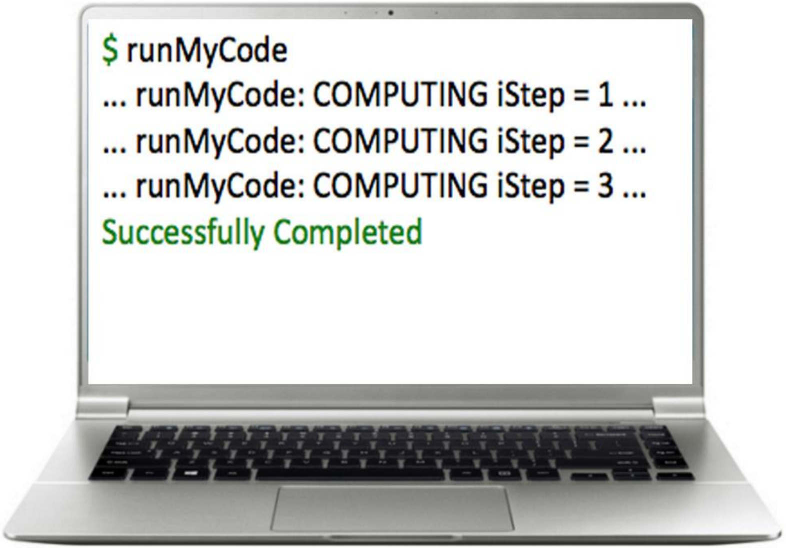

Random UUID
(All writes and deletes go in this layer)
Container layer (R/W)
91e54dfb1179
0 B
d74508fb6632
1.895 KB
c22013c84729
194.5 KB
d3a1f33e8a5a
188.1 MB
ubuntu:15.04
Cryptographic content hashes
Image layers (R/O)
Container
(based on ubuntu:15.04 image)

Containers for Scientific Computing

Why do we want containers in HPC?

- Escape “dependency hell”
- Local and remote code works identically every time
- One file contains everything and can be moved anywhere

Environment Matters



Needs for HPC containers

- Any user can run containers without special privileges (root)
- Integrate seamlessly into existing infrastructure
- Portability between many systems
- Users created and provided containers (no administrative oversight)

Singularity

Needs for HPC containers

- Any user can run containers without special privileges (root) ✓
- Integrate seamlessly into existing infrastructure ✓
- Portability between many systems ✓
- Users created and provided containers (no administrative oversight) ✓

Singularity

- Any container can be run by any user - same user inside container and on host
- No workflow changes necessary to use
- Single .img file contains everything necessary
- Safe to run any container without screening its contents

	Shifter	Charlie Cloud	Docker	Singularity
Privilege model	SUID	UserNS	Root Daemon	SUID/UserNS
Support current production Linux distros	Yes	No	No	Yes
Internal image build/boostrap	No*	No*	No**	Yes
No privileged or trusted daemons	Yes	Yes	No	Yes
No additional network configurations	Yes	Yes	No	Yes
No additional hardware	Maybe	Yes	Maybe	Yes
Access to host filesystem	Yes	Yes	Yes***	Yes
Native support for GPU	No	No	No	Yes
Native support for InfiniBand	Yes	Yes	No	Yes
Native support for MPI	Yes	Yes	No	Yes
Works with all schedulers	No	Yes	No	Yes
Designed for general scientific use cases	Yes	No	No	Yes
Contained environment has coorect perms	Yes	No	Yes	Yes
Containers are portable, unmodified by use	No	No	No	Yes
Trivial HPC install (one package, zero conf)	No	Yes	Yes	Yes
Admins can control and limit capabilities	Yes	No	No	Yes

* Relies on Docker

** Depends on upstream

*** With security implications

Site or Organization	System Name	Size (cores)	Purpose of the System
CSIRO	bragg-gpu	2048	broad base scientific
GSI Helmholtz Center	Greencube	300,000	Heavy Ion Physics
Holland Computing Center at UNL	Crane and Tusker	14,000	General purpose campus cluster
HPC-UGent	golett	2500	research across all scientific domains
Lunarc	Aurora	360	Research
Microway	Microway Research Cluster	192	Scientific benchmarking
MIT	openmind	1,176	Neuroscience
National Institute of Health HPC	Biowulf	54,000	General purpose biomedical research
Purdue University	Rice	11520	Campus HPC resource
Purdue University	Conte	78880	Campus HPC resource
Purdue University	Snyder	2220	Campus HPC resource
Purdue University	Hammer	3960	Campus HPC resource
Purdue University	Carter	10560	Campus HPC resource
R Systems NA, Inc.	Oak1	1024	Shared commercial/academic resource
R Systems NA, Inc.	Oak2	2048	Shared commercial/academic resource
R Systems NA, Inc.	HOU1	5376	Shared commercial/academic resource
San Diego Supercomputer Center	Gordon	16384	HPC cluster for XSEDE users
San Diego Supercomputer Center (SDSC)	Comet	47776	HPC Cluster for XSEDE users
Texas Advanced Computing Center	Stampede	102400	NSF key resource, all fields
UFIT Research Computing at the UF	HiPerGator	51,000	research computing cluster
Ulm University, Germany	JUSTUS	550	Computational Chemistry
University of Chicago	midway.rcc.uchicago.edu	24196	University cluster
University of Manitoba	Grex	3840	General purpose HPC cluster
Georgia State University	Orion	362	research
UNF	Stark	64	Functional MRI analysis of the Brain
Genentech, Inc.			Research
Rutgers University	sirius	32	scientific SMP machine
Stanford University	sherlock	12764	Compute for Stanford researchers
Stanford University	scg4	3920	Genomics at Stanford
The University of Leeds	MARC1	1236	Bioinformatics, data analytics
McGill HPC Centre/Calcul Québec	guillimin	22300	Compute Canada cluster
University of Arizona	Ocelote	10000	General Research
University of Arizona	ElGato	2300	GPU cluster
Washington University in St. Louis		2000	General purpose cluster

PLOS | SUBMISSION Submitted: December 19, 2016

Singularity: Scientific Containers for Mobility of Compute

Gregory M. Kurtzer[1], Vanessa Sochat[2*], Michael W. Bauer[3,4]

1 High Performance Computing Services, Lawrence Berkeley National Lab, Berkeley, CA, USA
2 Stanford Research Computing Center and School of Medicine, Stanford University, Stanford, CA, USA
3 Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA
4 Experimental Systems, GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt, Germany

Basic Usage of Singularity

Singularity Workflow

1. Create image file
   - $ sudo singularity create [image]
2. Bootstrap image
   - $ sudo singularity bootstrap [image] [definition.def]
3. Run image
   - $ singularity shell [image]
   - $ singularity exec [image] [/path/to/executable]
   - $ singularity run [image]
   - $ ./image

Singularity Workflow

https://asciinema.org/a/100297

Format	**Description**
directory	Standard Unix directories containing a root container image
tar.gz	Zlib compressed tar archives
tar.bz2	Bzip2 compressed tar archives
tar	Uncompressed tar archives
cpio.gz	Zlib compressed CPIO archives
cpio	Uncompressed CPIO archives

Docker Integration

https://asciinema.org/a/101984

SLURM Integration

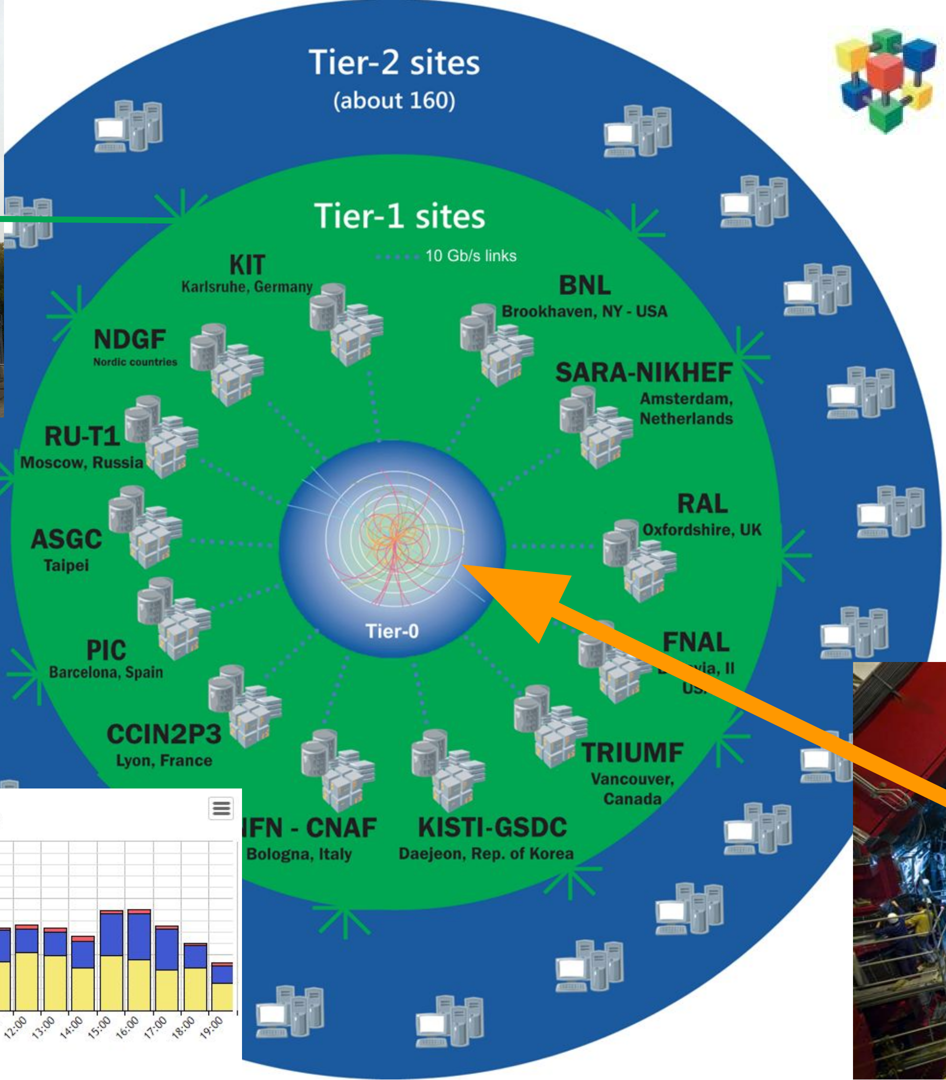
```
#!/bin/bash -l

#SBATCH --image=~/centos7/latest
#SBATCH -p debug
#SBATCH -N 64
#SBATCH -t 00:20:00
#SBATCH -J my_job
#SBATCH -L SCRATCH
#SBATCH -C haswell

srun -n 4096 ./mycode.exe
```

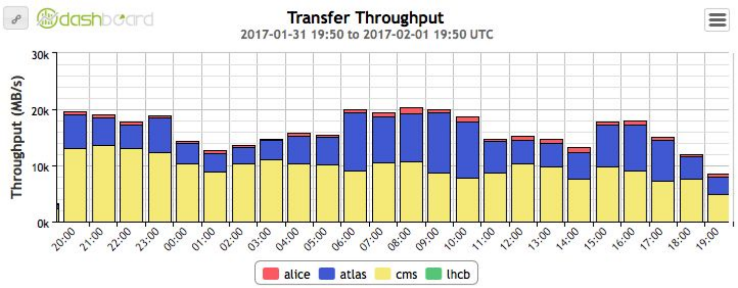
ALICE Tier 2 Use Case

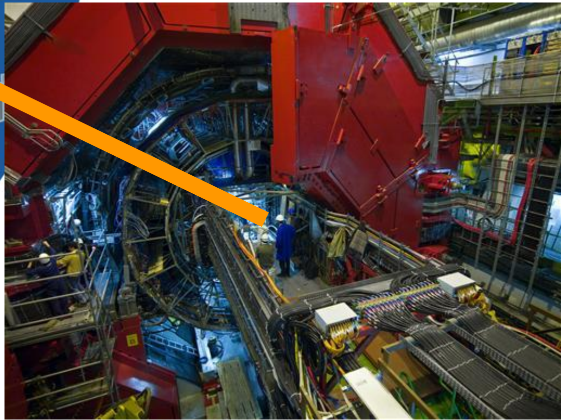
GSI Green Cube
Darmstadt
Germany



WLCG
Worldwide LHC Computing Grid

ALICE Detector LHC
Geneva
Switzerland





ALICE Tier 2: Problem

- Run ALICE jobs on ~2k jobs at any time
- Host machines run Debian 7.x kernel 3.16
- ALICE expects Scientific Linux 6 (SL6)
- Library incompatibilities cause frequent errors (much higher than expected)

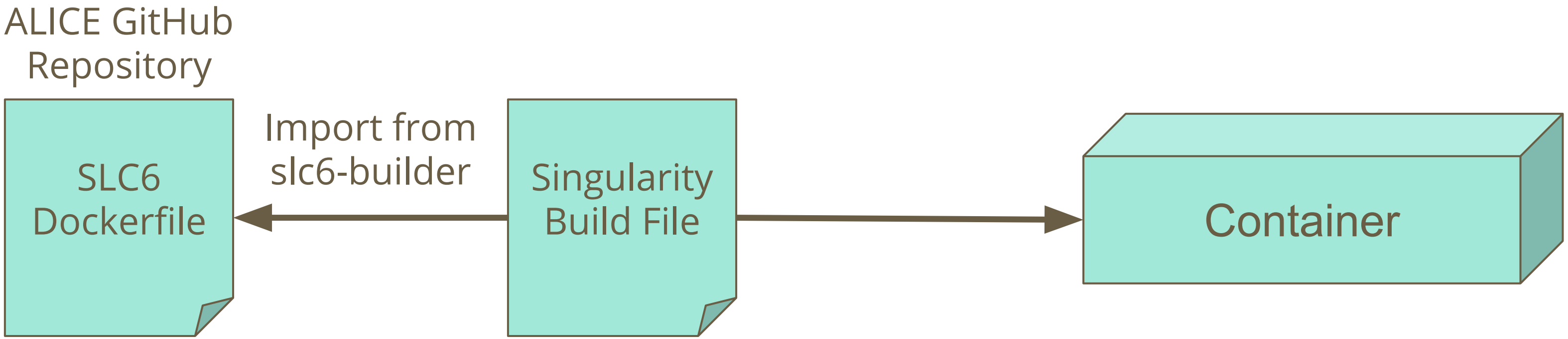
ALICE Tier 2: Pre-Singularity Solution

- Correct library versions mounted in Lustre
- SLURM job submission script alters $LD_LIBRARY_PATH to point to Lustre
- And maybe more?

Big Ugly Hack

ALICE Tier 2: Singularity Solution

- Package Scientific Linux 6 into container
- Modify SLURM submission script to run container
- No need to mount Lustre for access to library files
- Can test container locally before deploying to HPC



Global Options	
$-d - -debug$	Print debugging information
$-h - -help$	Display usage summary
$-q - -quiet$	Only print errors
$- - version$	Show application version
$-v - -verbose$	Increase verbosity +1
$-x - -sh - debug$	Print shell wrapper debugging information
General Commands	
help	Show additional help for a command
Container Usage Commands	
exec	Execute a command within container
run	Launch a runscript within container
shell	Run a Bourne shell within container
test	Execute any test code defined within container
Container Management Commands (requires root)	
bootstrap	Bootstrap a new Singularity image
copy	Copy files from your host into the container
create	Create a new container image
export	Export the contents of a container via a tar pipe
import	Import/add container contents via a tar pipe
mount	Mount a Singularity container image

GitHub, Inc. github.com/singularityware/singularity/blob/master/AUTHOR

singularityware / singularity

Watch 40 | Star 143 | Fork 64

Code | Issues 68 | Pull requests 9 | Projects 1 | Pulse | Graphs

Branch: master ▾ singularity / **AUTHORS**

Find file | Copy path

gmkurtzer Added Felix to contributors — 1c45cbb on Nov 30, 2016

4 contributors

22 lines (19 sloc) | 691 Bytes

Raw | Blame | History

```
Project Lead:
    Gregory M. Kurtzer <gmkurtzer@lbl.gov>

Developers:
    Brian Bockelman <bbockelm@cse.unl.edu>
    Krishna Muriki <kmuriki@lbl.gov>
    Michael Bauer <bauerm@umich.edu>
    Vanessa Sochat <vsochat@stanford.edu>

Contributors:
    Amanda Duffy <aduffy@lenovo.com>
    Ángel Bejarano <abejarano@ontropos.com>
    Bernard Li <bernardli@lbl.gov>
    Dave Love <d.love@liverpool.ac.uk>
    Felix Abecassis <fabecassis@nvidia.com>
    Jarrod Johnson <jjohnson2@lenovo.com>
    Jason Stover <jason.stover@gmail.com>
    Maciej Sieczka <msieczka@sieczka.org>
    Nathan Lin <nathan.lin@yale.edu>
    Ralph Castain <rhc@open-mpi.org>
    Yaroslav Halchenko <debian@onerussian.com>
```

We are always looking for more Collaborators!