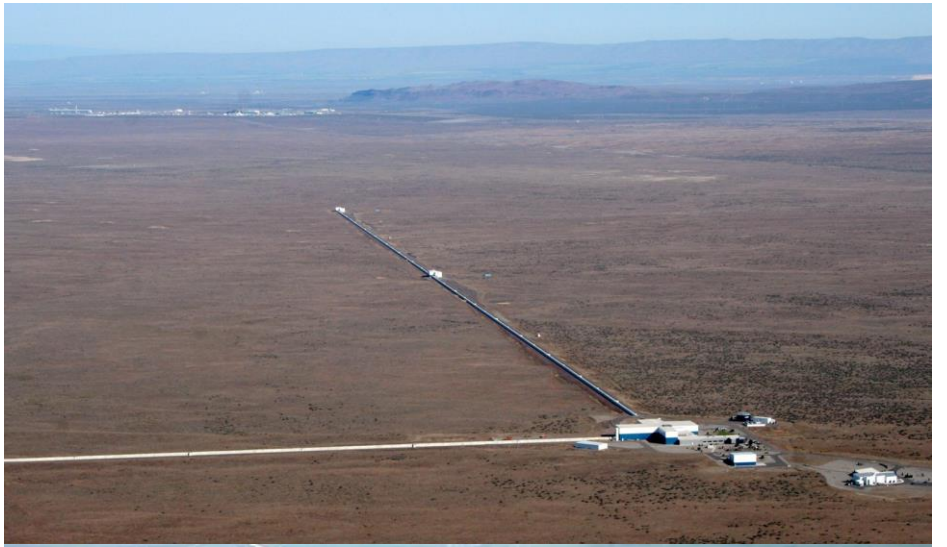
Automated Docker-CVMFS Conversion for LIGO and the Open Science Grid

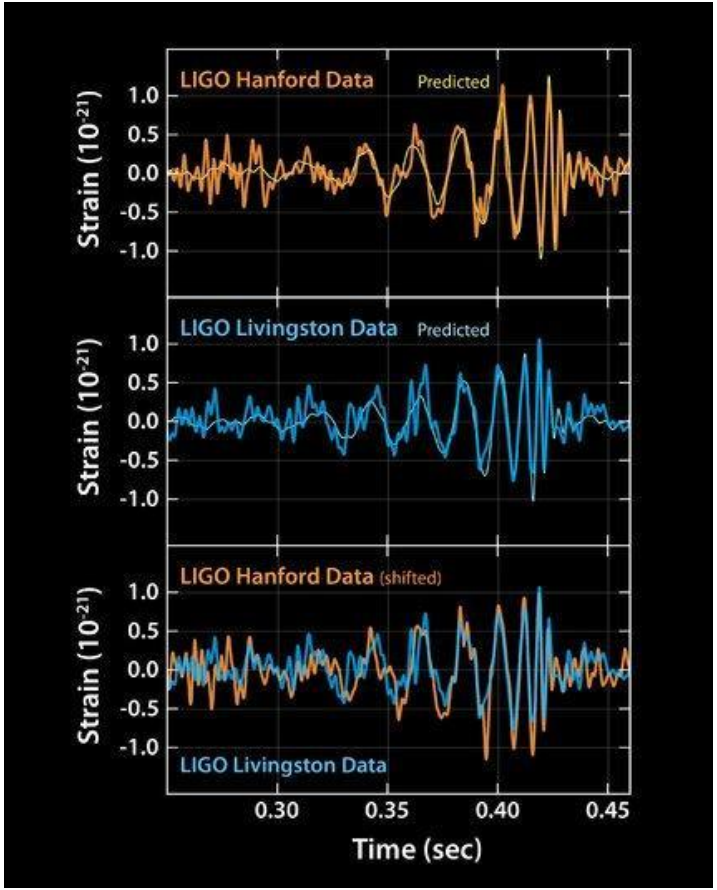
Thomas P Downes
Center for Gravitation, Cosmology & Astrophysics
University of Wisconsin-Milwaukee
LIGO Scientific Collaboration

LIGO-Virgo Advanced Detector Network

O1: September 2015 -- January 2016
O2: December 2016 -- August 2017
O3: ~1 year of observing TBA

Upper-right: LIGO Hanford, Washington State, USA
Lower-right: Virgo ca. Pisa, Italy
Unshown: LIGO Livingston, Louisiana, USA





Images courtesy LIGO Laboratory & Fisher Price

“Modeled” LIGO searches compare data to many simulations

Essentially all LIGO searches are embarrassingly parallel workflows on clusters running HTCondor. We are seeing growth in usage by

- LIGO researchers receiving computing resources from their institutions
- Open Science Grid resources (may also be a part of institutional resource)

These resources don't typically come with the same software or support/permission structure of a LIGO-managed site...

Enter CVMFS...

A greatly reduced list of our other problems

- We support two “RefOS” package repos
  - RHEL7 derivatives + Debian 8 (soon 9)
- But users work differently!
- Blurry line into infrastructure
  - Observatories calibrate / monitor data
- Actual jobs typically run via home directory + dependency chain into system
- Lots of replicated work across clusters
- Long e-mail chains across time zones
- Divine intervention required to replicate analyses in the future

Enter containers...

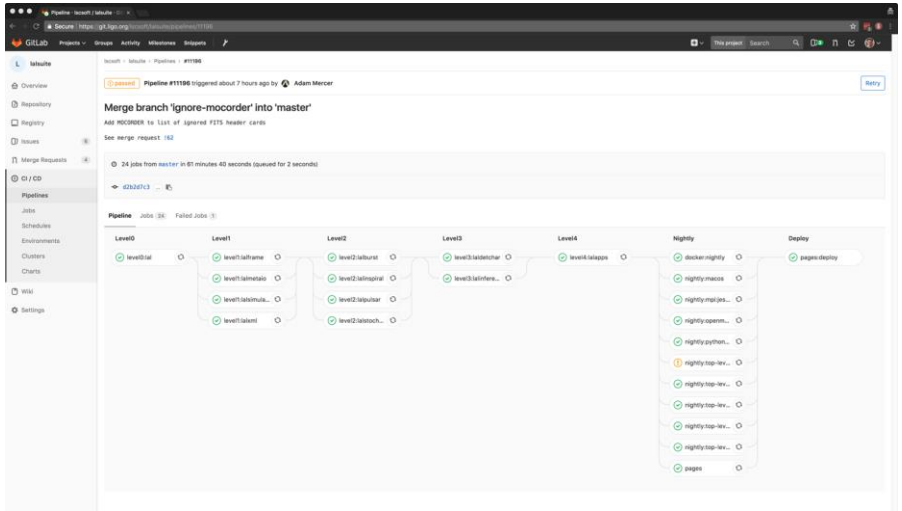
Users (and staff) get confused by “Inception” computing

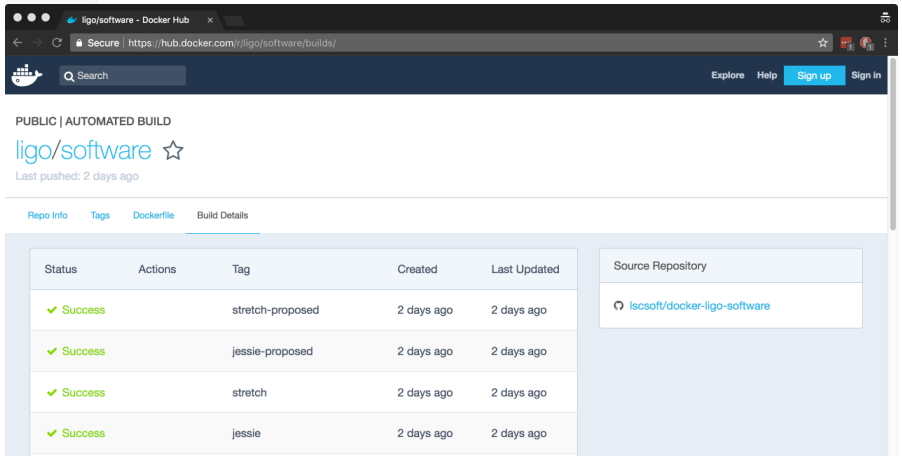
Automation helps

Above: Nightly build/public release of LIGO Algorithm Library container. Available via

docker pull containers.ligo.org/lscsoft/lalsuite:nightly

Below: API-triggered DockerHub rebuilds of our cluster login and job environment





DockerHub or GitLab Container Registry builds container and generates webhook
[DockerHub: +1 hour @ 5GB worker node]
[GitLab Container Registry: Θ(minutes)]

LIGO Webhook Relay validates and forwards event to CVMFS Publisher

CVMFS Publisher receives event and places it in job queue

Job queue pulls container images and publishes them 1-by-1
[+40 minutes @ 5GB]

Available to clients at
/cvmfs/ligo-containers.opensciencegrid.org

Automated publishing of Docker images to CVMFS

Thanks, CERN + Open Science Grid!

- CERN + OSG has greatly improved support for our Debian clusters and users
- OSG infrastructure serves as LIGO's Stratum 1 CVMFS Replicas
- Code to convert Docker images to CVMFS is a fork of OSG's nightly script developed by Brian Bockelman and Derek Weitzel
  - https://github.com/opensciencegrid/cvmfs-singularity-sync
  - *Still relies on shell calls to cvmfs*
- Feedback: Data/Auth needs to be/remain(?) First Class Citizens in CVMFS
- Feedback: CVMFS + MacOS (or Docker on MacOS) not easy / tough to sell!
- Feedback: Docs detailed, but hard to explain "big picture" to new admins.
  - Example: Difference between CVMFS docs themselves and specific configuration (and firewalling) necessary for use of external Stratum 1

These applications are distributed as fairly simple Docker Compose applications

- Webhook Relay: https://github.com/lscsoft/webhook-relay
  - Validates webhooks (to best of ability) and relays events it is configured to expect
- Webhook Queue: https://github.com/lscsoft/webhook-queue
  - Receives webhooks (from Relay or direct from service) and places event on a job queue
- Relay + Queue can easily be re-implemented (*e.g.* AWS API Gateway + Lambda + SQS)
  - 5M API calls per month for $3.50
- CVMFS-to-Docker worker: https://github.com/lscsoft/cvmfs-docker-worker
  - Processes job queue, gracefully moving to next job upon failure
  - Pulls image from remote registry (authentication supported) using docker-py
  - Adds OSG bind points for those without OverlayFS
  - Publishes image as directory structure in CVMFS

The infrastructure is freely available


THE BEATLES
HELP!

- Docker stores its images as separate directories stacked w/OverlayFS in a running container
- Fastest CVMFS operation would make use of this but...
- … CVMFS **publish** operation mounts current filesystem as **lowerdir** and interprets the **upperdir** as the contents of a **transaction**.
- But... **upperdir** cannot itself be an OverlayFS...
- … and symlinks to an **upperdir** fail to open during publishing (fd < 0)

Instead… use time-consuming stage of tarball via **docker save** and decompression into CVMFS filesystem in middle of transaction.

Is there an OverlayFS-based approach that can be found?

“cvmfs_server publish --diff /my/overlayfs/dir”

OverlayFS Inception