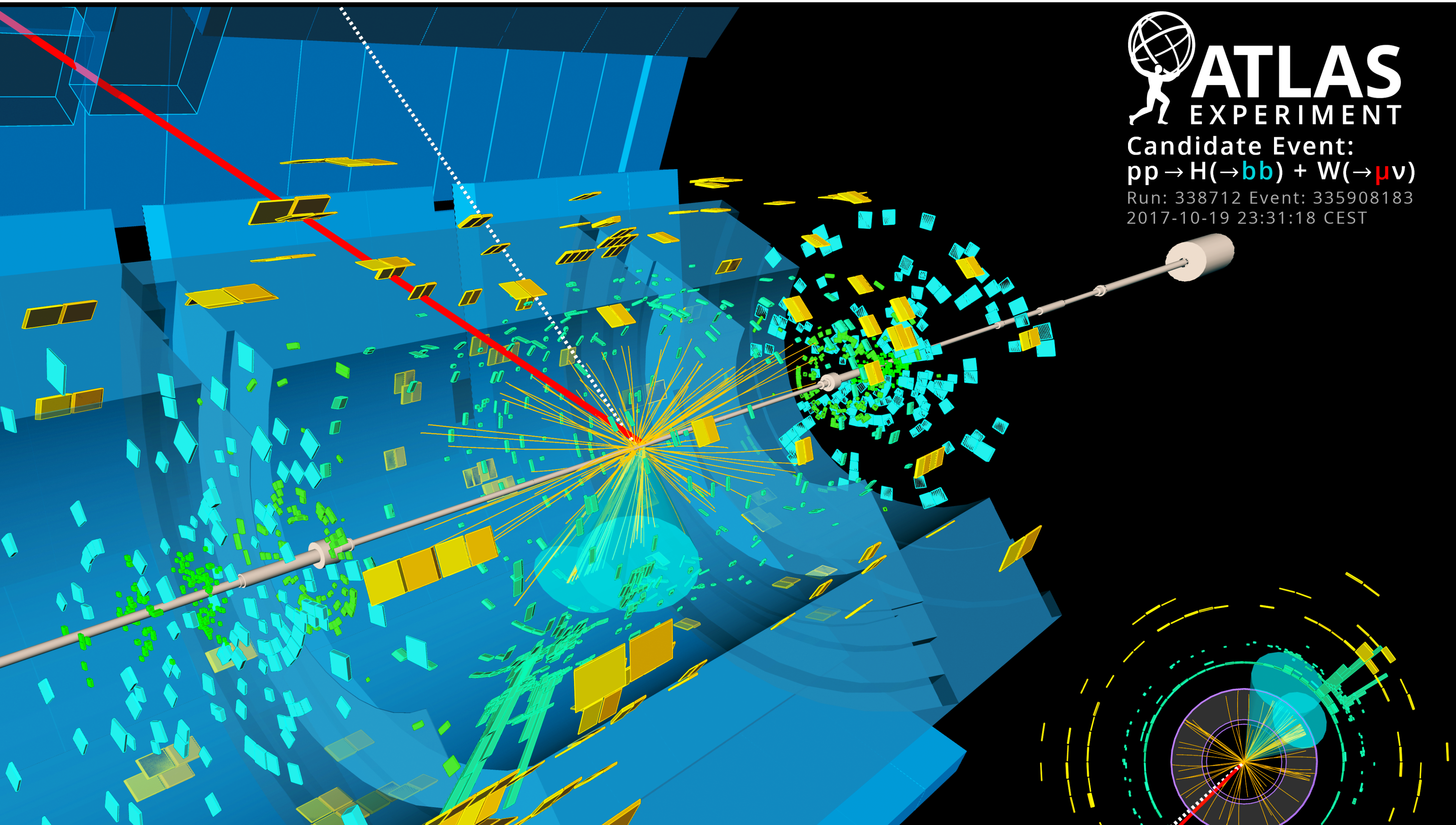
ATLAS Containers

Lukas Heinrich 2019/06/04 - CVMFS Workshop

HEP Computing — In a Nutshell

Foundational Principle: repeated experiment, i.e. proton collisions

- each event is independent of the other
- to zero-th order HEP computing is embarassingly parallel - great for distributed computing



HEP Computing — In a Nutshell

Idea: easier to **send code** to data than vice versa:

Worldwide LHC Computing Grid

necessarily heterogeneous

- small univ. clusters
- leadership class HPCs

Mission:
Keep it all working for all use-cases

- from well-oiled s/w
- to one-off users

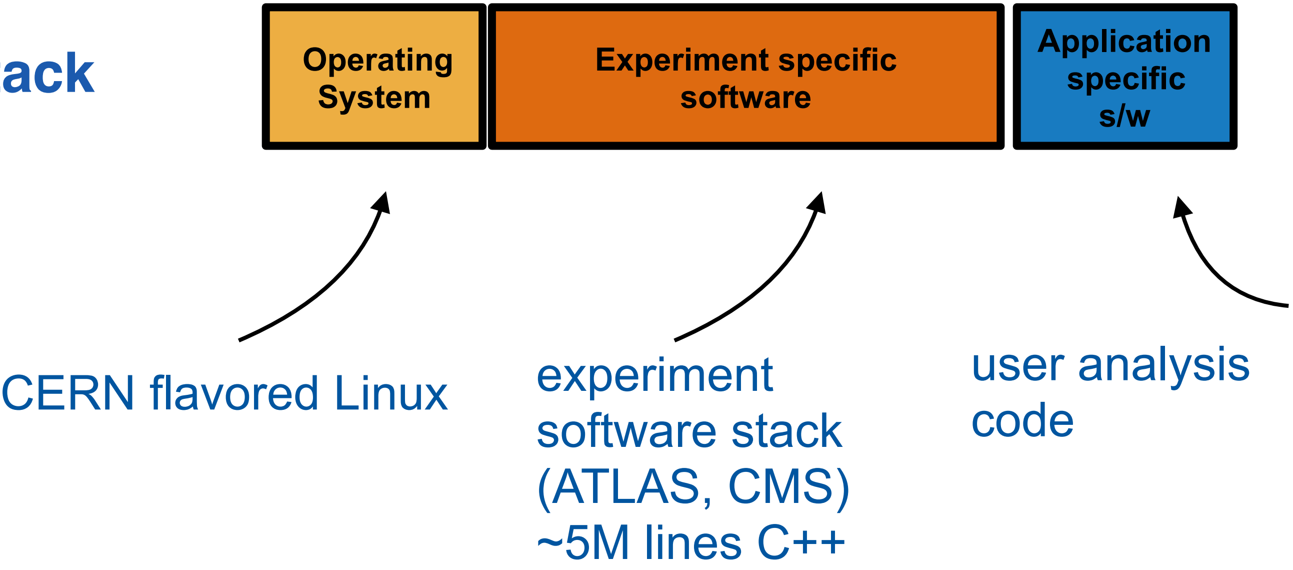
Tier 0
Tier 1

Distributed Computing ↔ Software Distribution

Idea: easier to **send code** to data than vice versa.

We need to materialize the software stack on the remote machines <u>somehow</u>.

Our traditional Stack



Operating System

Experiment specific software

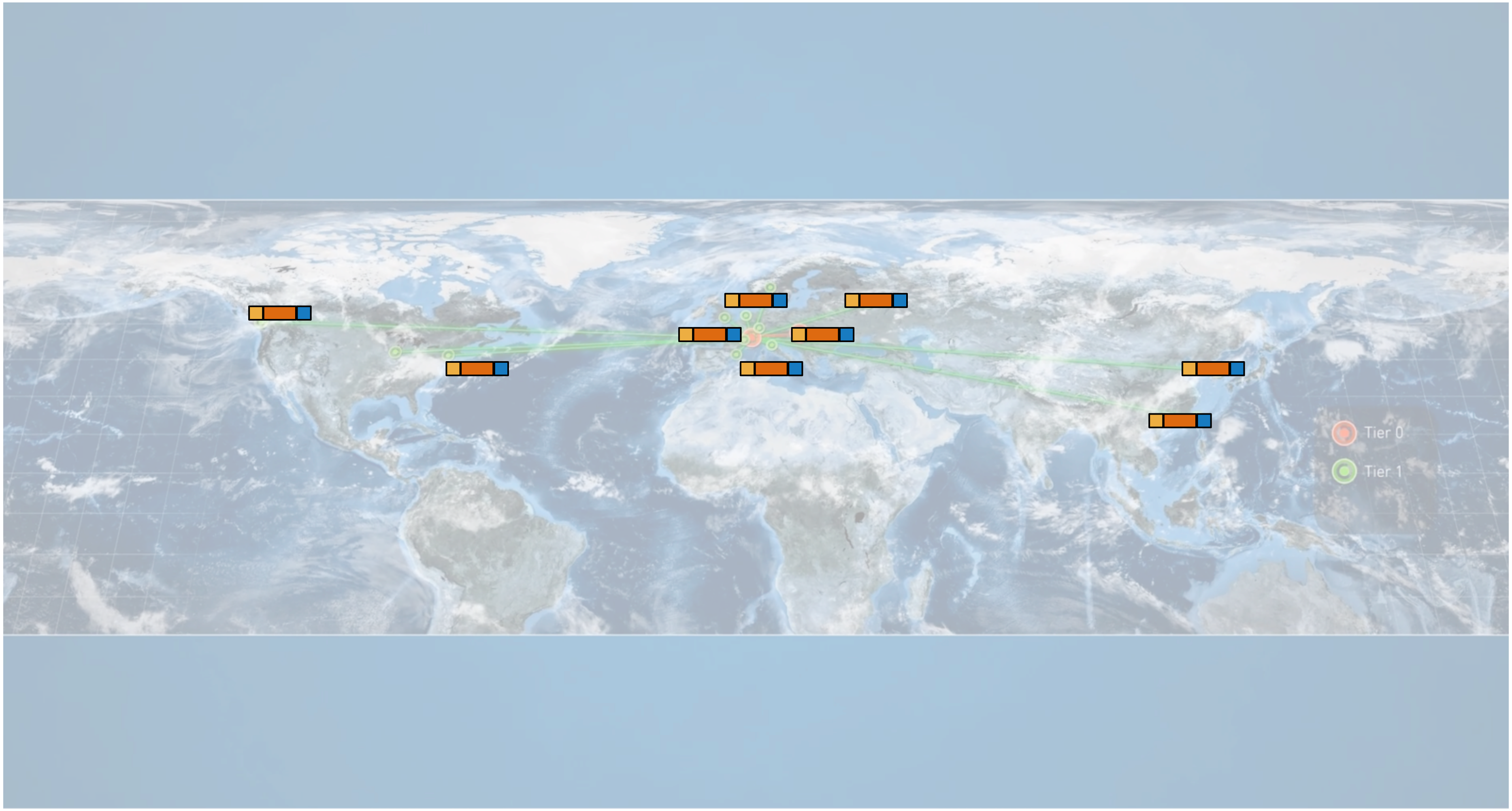
Application specific s/w

CERN flavored Linux

experiment software stack (ATLAS, CMS) ~5M lines C++

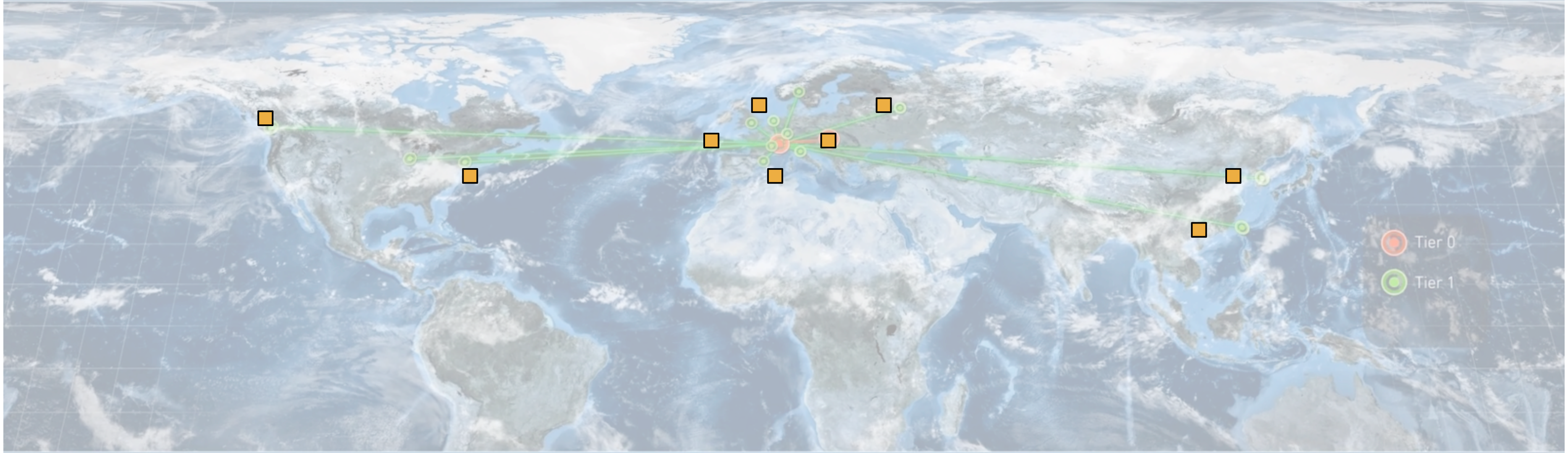
user analysis code

Software Distribution today



Software Distribution today

Operating System

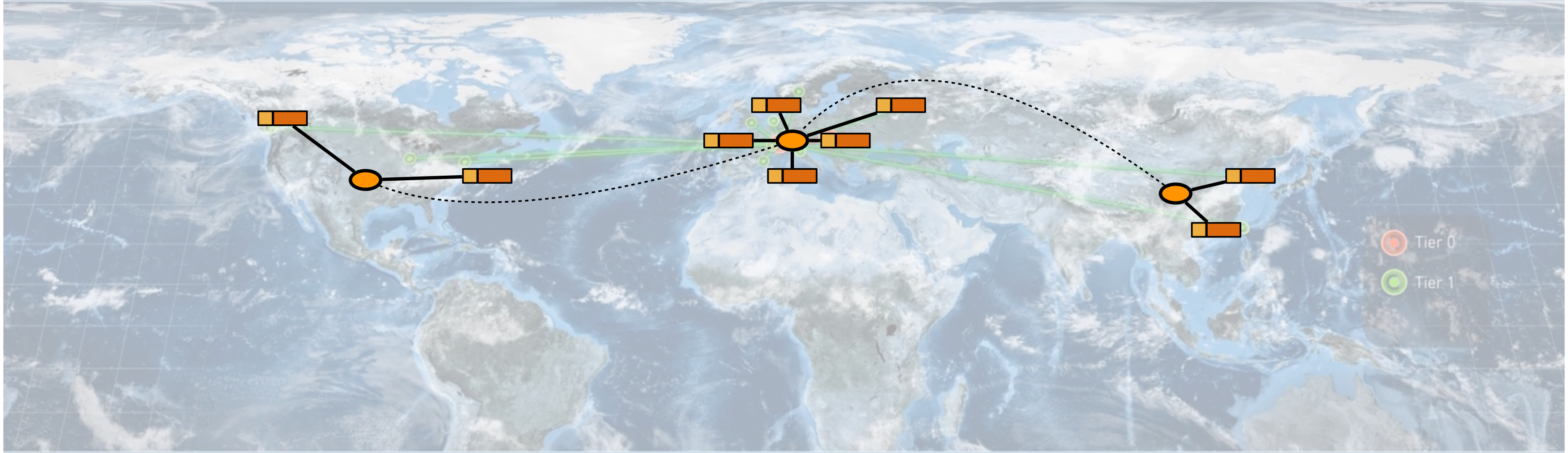


Provided by Site Admins - e.g. CERN Linux (SLC6, CC7)

Software Distribution today

Operating System

Experiment specific software



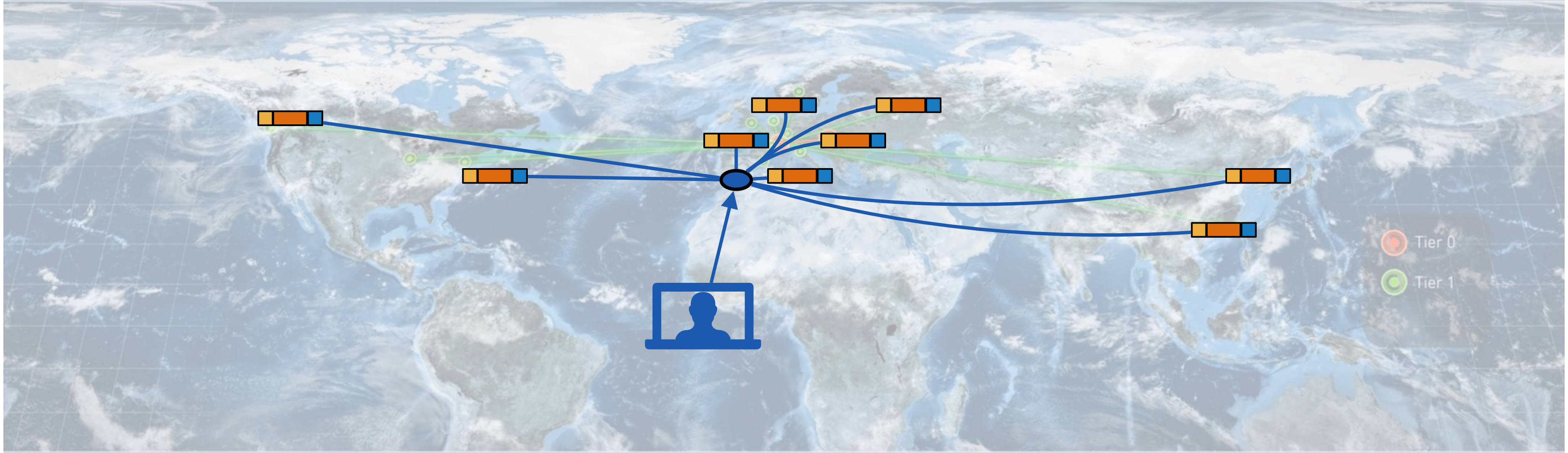
Provided by CVMFS (read only fs)
efficient hierarchy of caches

Software Distribution today

Operating System

Experiment specific software

Application specific s/w

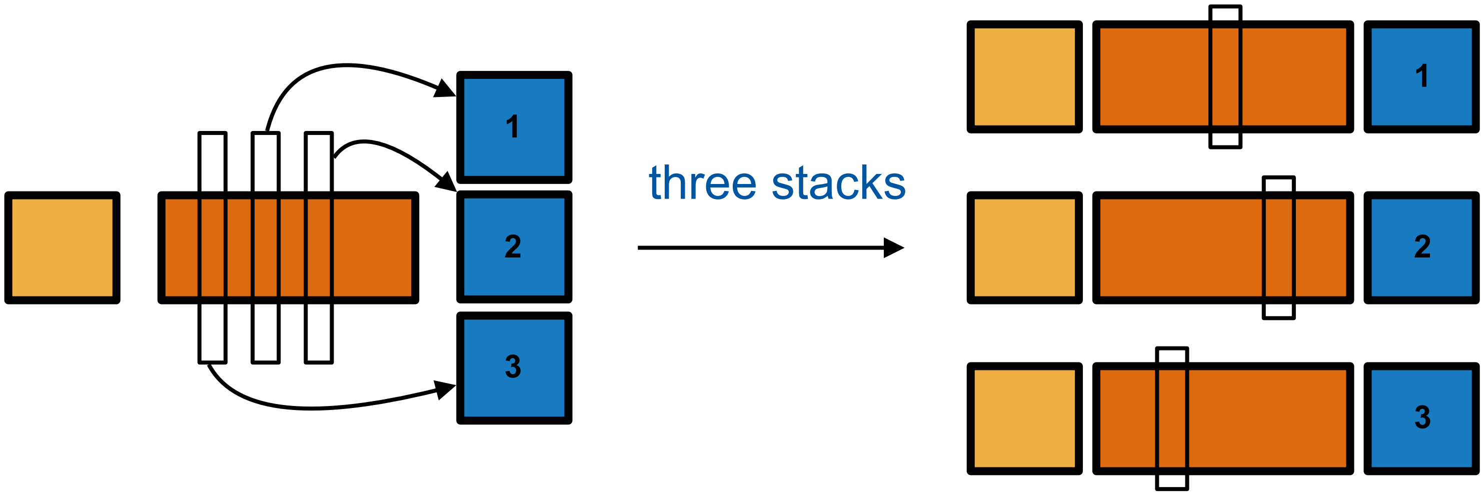


Provided by Users
distributed via HTTP download (.tgz)

Advantages

Software distribution is very efficient

- mainly through convention
  - we agree on a base OS
  - we publish all experiment sw/ on cvmfs.
    - agree not to delete, to responsibly manage global state
  - user app layer is small

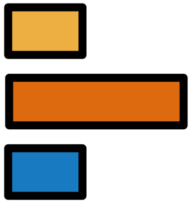


Challenges

The current system works very well for standard workloads (bulk reconstruction, vanilla analysis code)

But relies on separate parties to "materialize the stack"

- **site admins**
- **experiment experts**
- **users**



If any party breaks the stack, we have a problem.

inadvertent system update

faulty publishing by expts

user error

Goals

- We want reproducible software environments -- globally
  - dev - prod parity - easier testing
  - software archiving / computational reproducibility of results

- We want full control over our stack and loose coupling
  - global fs → global state, large dependency surface
    - hard to analysis precisely on *which* slice of cvmfs you depend on for a specific applicaiton

  - Out Stacks are changing and become more diverse
    - Machine Learning
    - Special Architectures
    - Long tail of data-science / analysis software (e.g. python eco-system)

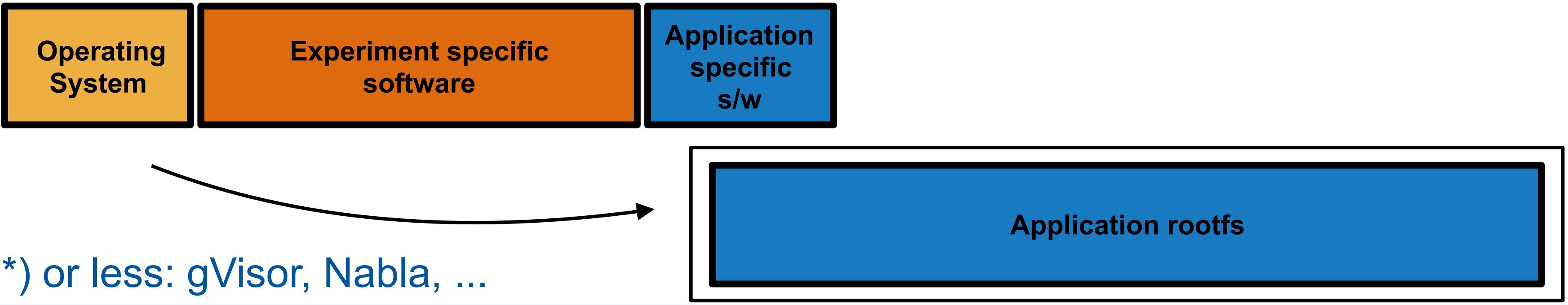
Distributed Computing ↔ Software Distribution

Industry found alternative way for reproducible, global sofware distribution

OCI Container Images

Give application developer full control (and responsibility) of defining their runtime environment.

Only expose Linux Kernel as interface (*)

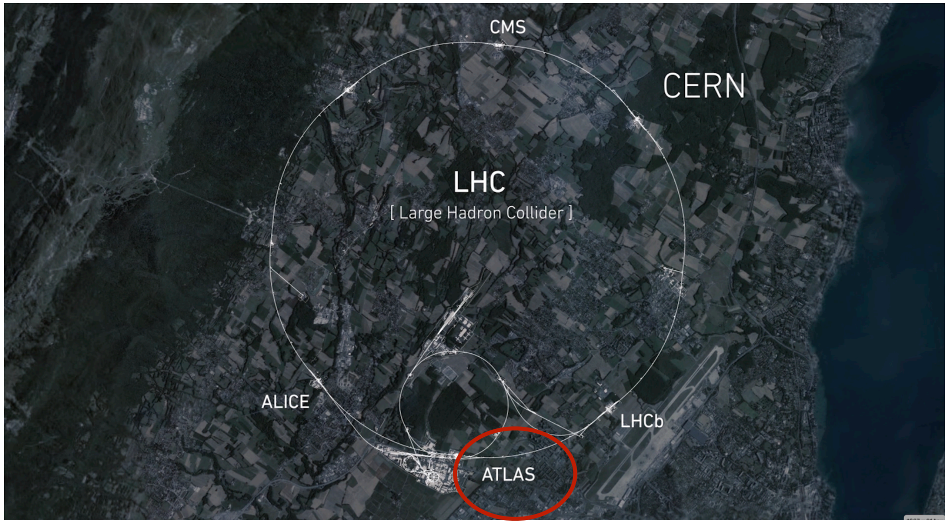


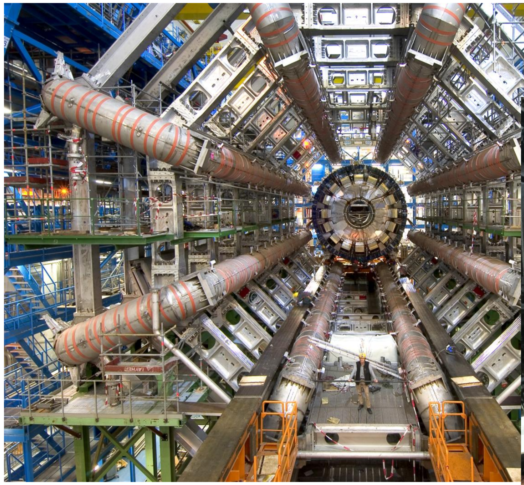
(*) or less: gVisor, Nabla, ...

Containers in ATLAS

ATLAS

- one of the 4 LHC experiments
- has been driving the use of containers in HEP
  - Continuous Integration
  - Analysis Preservation & Reuse
  - Machine Learning

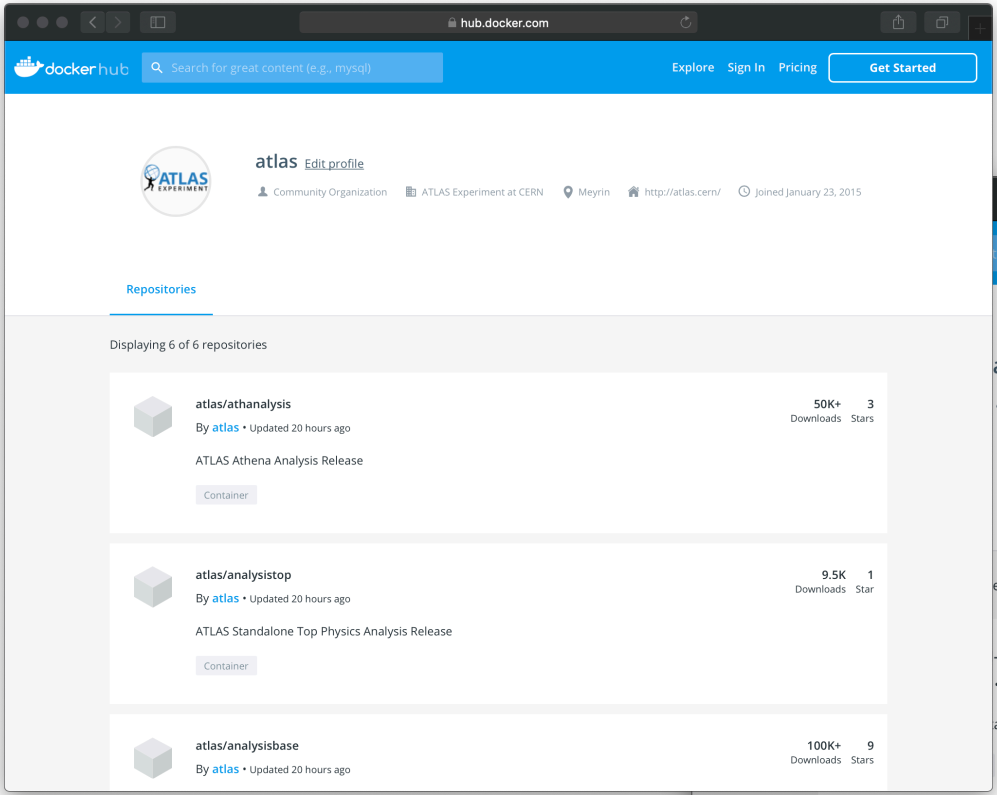




Me and my high school physics teacher today a.m.

Containers in ATLAS

- We provide to users curated base images with ATLAS software (single release images: ~2GB)



atlas Edit profile

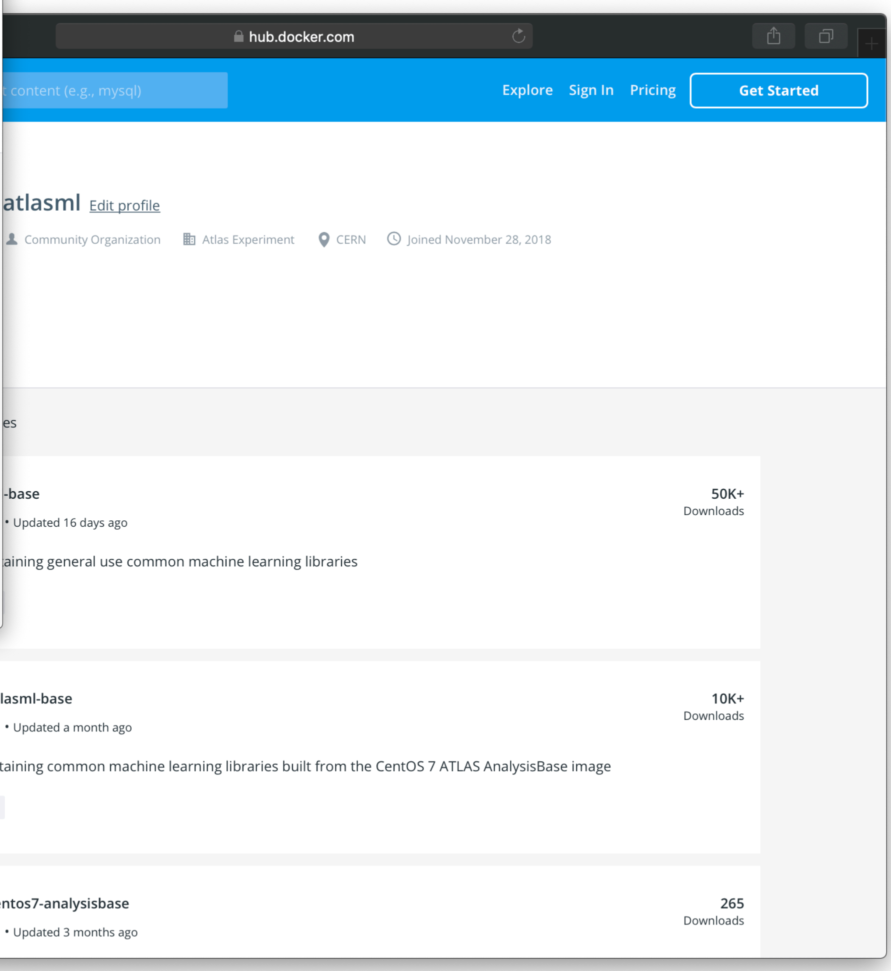
Community Organization · ATLAS Experiment at CERN · Meyrin · http://atlas.cern/ · Joined January 23, 2015

Displaying 6 of 6 repositories

atlas/athanalysis
By atlas • Updated 20 hours ago
ATLAS Athena Analysis Release
50K+ Downloads · 3 Stars

atlas/analysistop
By atlas • Updated 20 hours ago
ATLAS Standalone Top Physics Analysis Release
9.5K Downloads · 1 Star

atlas/analysisbase
By atlas • Updated 20 hours ago
100K+ Downloads · 9 Stars



atlasml Edit profile

Community Organization · Atlas Experiment · CERN · Joined November 28, 2018

es

-base
• Updated 16 days ago
aining general use common machine learning libraries
50K+ Downloads

atlasml/atlasml-base
By atlasml • Updated a month ago
Image containing common machine learning libraries built from the CentOS 7 ATLAS AnalysisBase image
10K+ Downloads

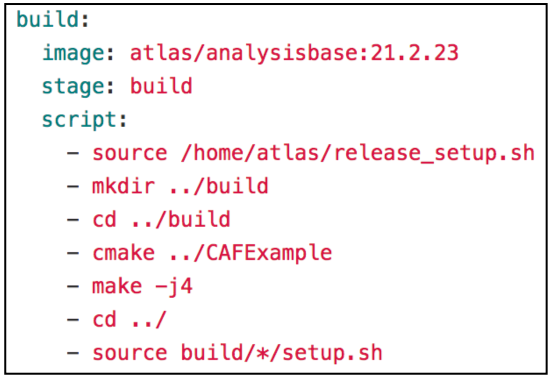
atlasml/centos7-analysisbase
By atlasml • Updated 3 months ago
265 Downloads

- Optimized ML images with e.g. Tensorflow Keras, Python 3, etc..

Containers in ATLAS

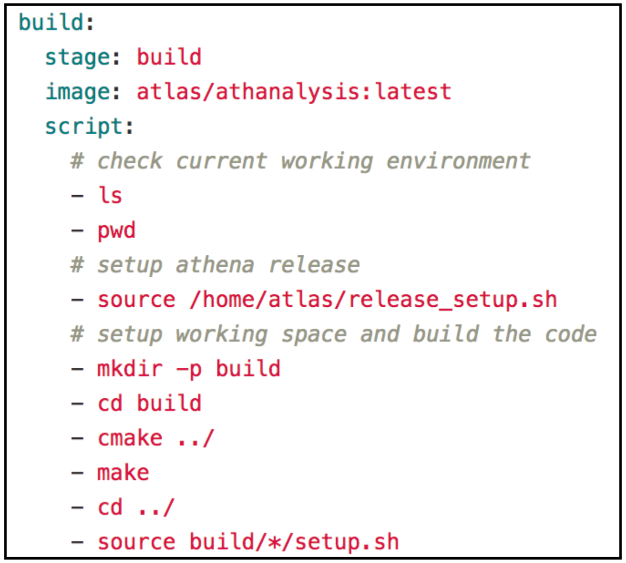
- **Usage in CI**
  - **works with <u>any</u> container-aware CI system (not only on-prem, e.g. Travis, CircleCI, etc..)**
  - **very natural workflow**

HWW Analysis (Higgs)



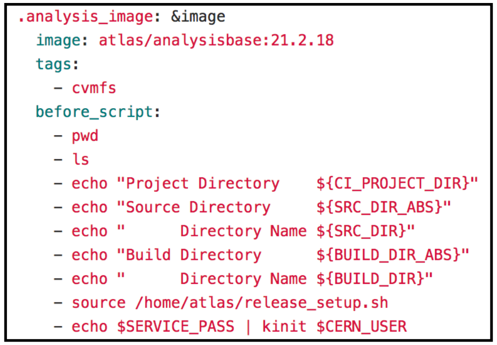
```
build:
  image: atlas/analysisbase:21.2.23
  stage: build
  script:
    - source /home/atlas/release_setup.sh
    - mkdir ../build
    - cd ../build
    - cmake ../CAFExample
    - make -j4
    - cd ../
    - source build/*/setup.sh
```

XAMMP monoH (Exotics)



```
build:
  stage: build
  image: atlas/athanalysis:latest
  script:
    # check current working environment
    - ls
    - pwd
    # setup athena release
    - source /home/atlas/release_setup.sh
    # setup working space and build the code
    - mkdir -p build
    - cd build
    - cmake ../
    - make
    - cd ../
    - source build/*/setup.sh
```

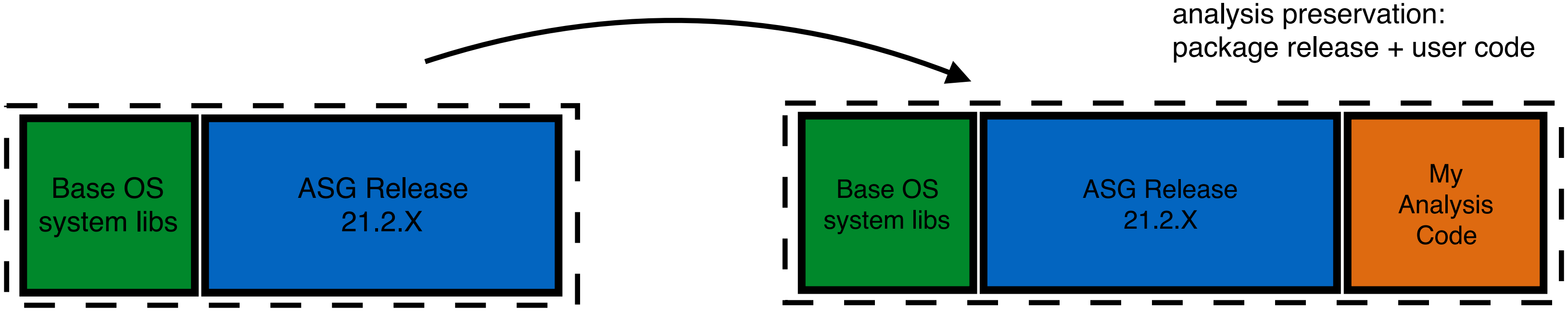
Multi-Bjet (SUSY Analysis)



```
.analysis_image: &image
  image: atlas/analysisbase:21.2.18
  tags:
    - cvmfs
  before_script:
    - pwd
    - ls
    - echo "Project Directory    ${CI_PROJECT_DIR}"
    - echo "Source Directory     ${SRC_DIR_ABS}"
    - echo "      Directory Name ${SRC_DIR}"
    - echo "Build Directory      ${BUILD_DIR_ABS}"
    - echo "      Directory Name ${BUILD_DIR}"
    - source /home/atlas/release_setup.sh
    - echo $SERVICE_PASS | kinit $CERN_USER
```

Containers in ATLAS

- Usage for Analysis Preservation
  - Build images as artifacts to be reused later by different teams



analysis preservation:
package release + user code

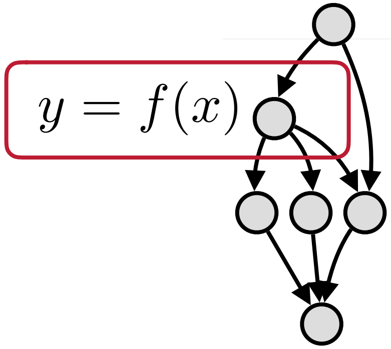
official base image

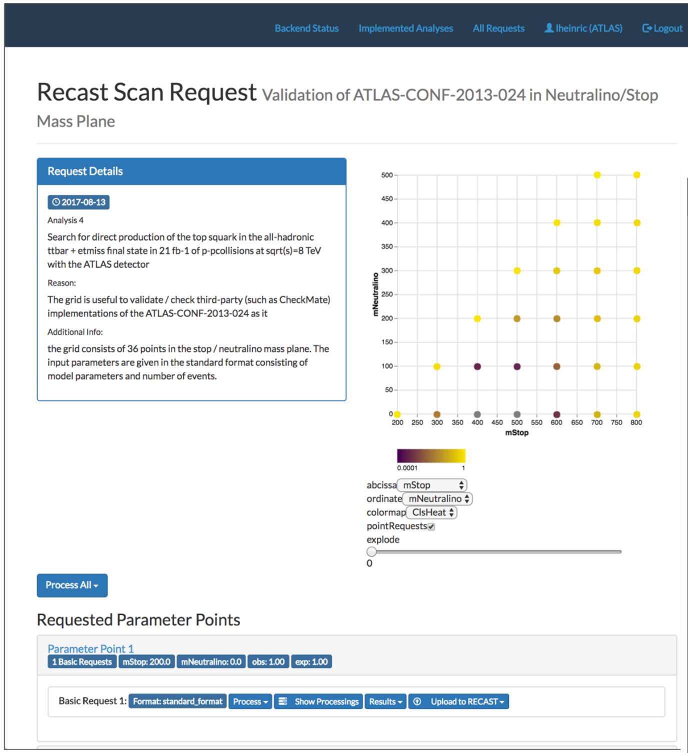
analysis-specific image
for analysis preservation

Containers in ATLAS - on the

Reproducible research data analysis platform

- RECAST: systematic reuse of past analyses
  - containerized, parametrized pipelines
  - better assess viability of physics theories in light of LHC data





Recast Scan Request Validation of ATLAS-CONF-2013-024 in Neutralino/Stop Mass Plane

Request Details

2017-08-13

Analysis 4

Search for direct production of the top squark in the all-hadronic ttbar + etmiss final state in 21 fb-1 of p-pcollisions at sqrt(s)=8 TeV with the ATLAS detector

Reason:

The grid is useful to validate / check third-party (such as CheckMate) implementations of the ATLAS-CONF-2013-024 as it

Additional Info:

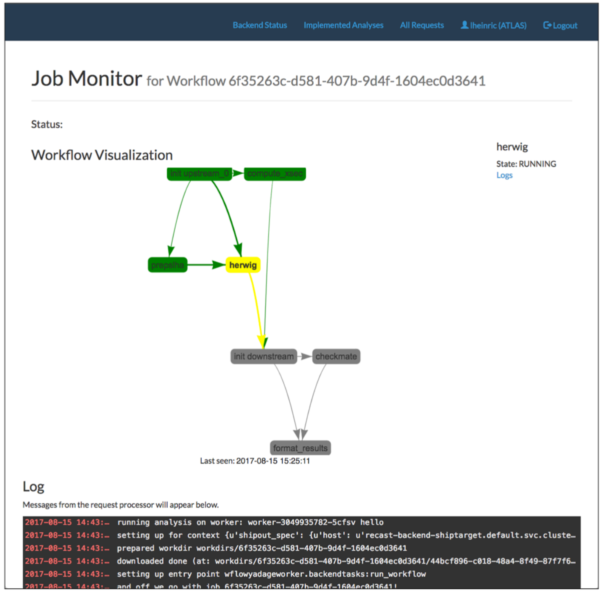
the grid consists of 36 points in the stop / neutralino mass plane. The input parameters are given in the standard format consisting of model parameters and number of events.

abcissa mStop
ordinate mNeutralino
colormap ClsHeat
pointRequests ☑
explode 0

Requested Parameter Points

Parameter Point 1
1 Basic Requests | mStop: 200.0 | mNeutralino: 0.0 | obs: 1.00 | exp: 1.00

Basic Request 1: Format: standard_format



Job Monitor for Workflow 6f35263c-d581-407b-9d4f-1604ec0d3641

Status:

Workflow Visualization

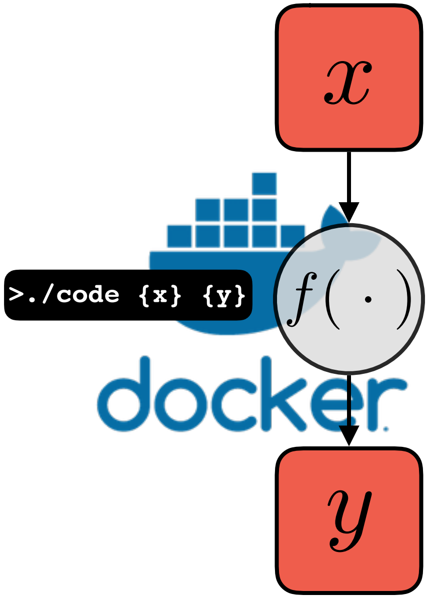
herwig
State: RUNNING

Last seen: 2017-08-15 15:25:11

Log

Messages from the request processor will appear below.

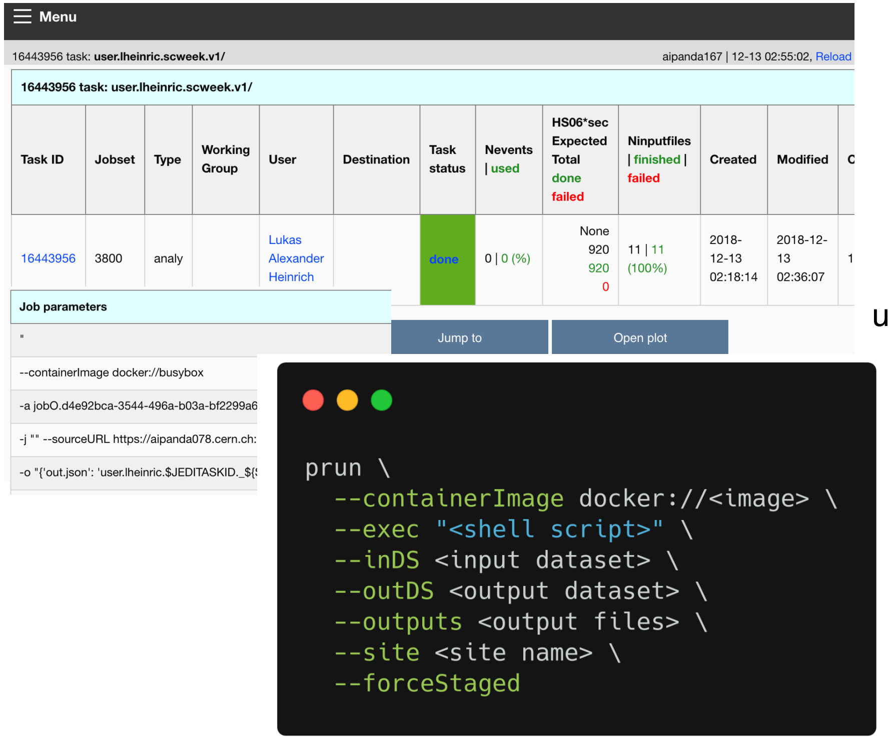
```
2017-08-15 14:43:…  running analysis on worker: worker-3049935782-5cfsv hello
2017-08-15 14:43:…  setting up for context {u'shipout_spec': {u'host': u'recast-backend-shiptarget.default.svc.cluste…
2017-08-15 14:43:…  prepared workdir workdirs/6f35263c-d581-407b-9d4f-1604ec0d3641
2017-08-15 14:43:…  downloaded done (at: workdirs/6f35263c-d581-407b-9d4f-1604ec0d3641/44bcf896-c018-48a4-8f49-87f7f6…
2017-08-15 14:43:…  setting up entry point wflowyadageworker.backendtasks:run_workflow
2017-08-15 14:43:…  and off we go with job 6f35263c-d581-407b-9d4f-1604ec0d3641!
```



Containers on the GRID

Containers are nice and well…
… but we need to integrate it tightly into our existing infra

E.g. native container-based jobs on WLCG grid



16443956 task: **user.lheinric.scweek.v1/** aipanda167 | 12-13 02:55:02, Reload

16443956 task: user.lheinric.scweek.v1/

Task ID	Jobset	Type	Working Group	User	Destination	Task status	Nevents | used	HS06*sec Expected Total done failed	Ninputfiles | finished | failed	Created	Modified	C
16443956	3800	analy		Lukas Alexander Heinrich		done	0 | 0 (%)	None 920 920 0	11 | 11 (100%)	2018-12-13 02:18:14	2018-12-13 02:36:07	1

Job parameters

"

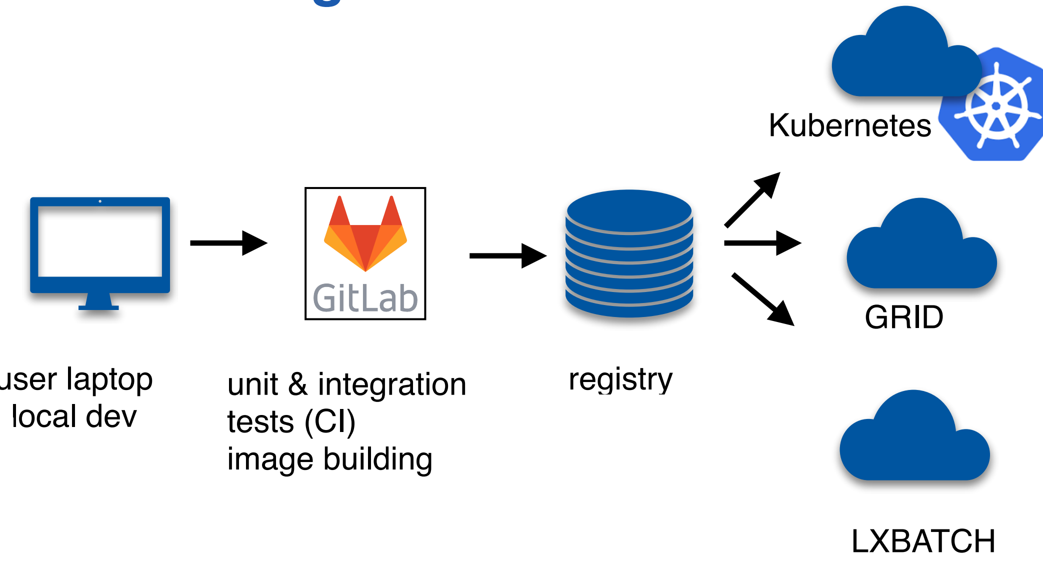
--containerImage docker://busybox

-a jobO.d4e92bca-3544-496a-b03a-bf2299a6

-j "" --sourceURL https://aipanda078.cern.ch:

-o "{'out.json': 'user.lheinric.$JEDITASKID._${

```
prun \
  --containerImage docker://<image> \
  --exec "<shell script>" \
  --inDS <input dataset> \
  --outDS <output dataset> \
  --outputs <output files> \
  --site <site name> \
  --forceStaged
```

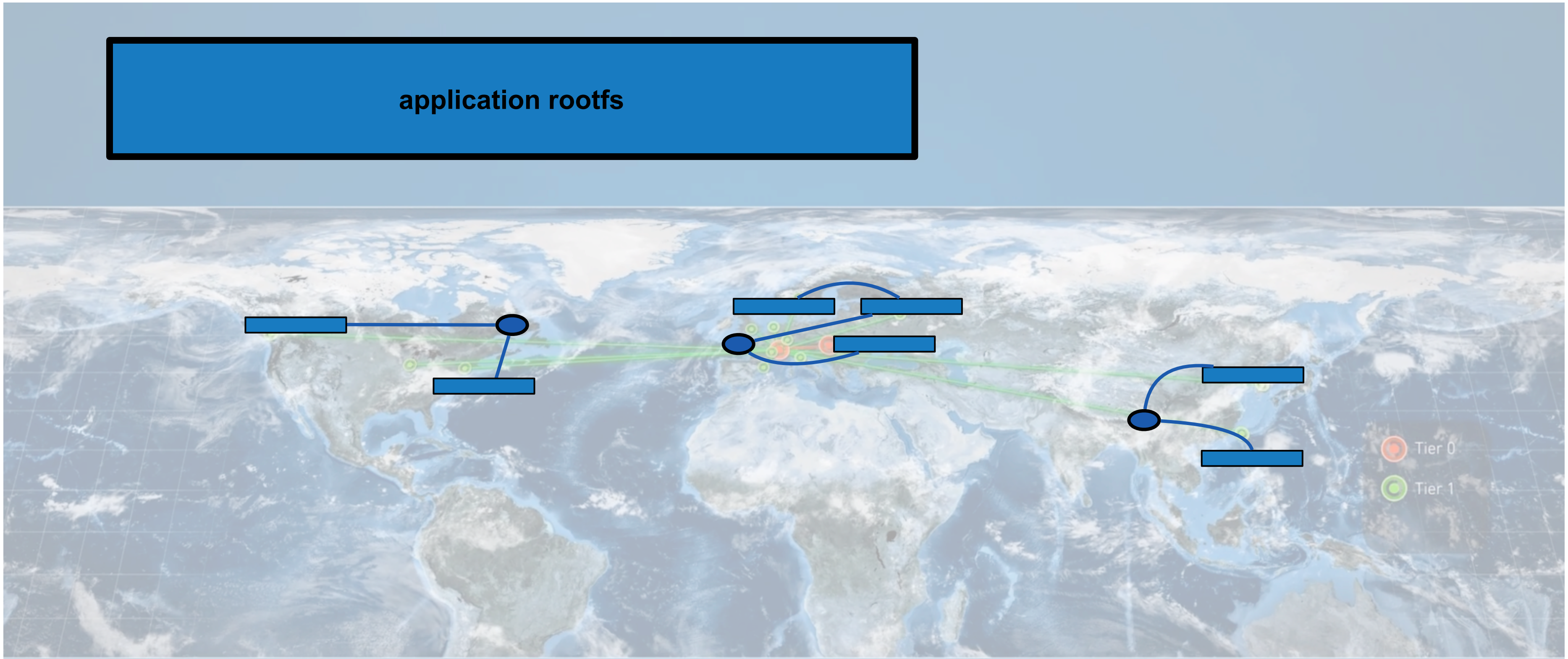


user laptop
local dev

unit & integration
tests (CI)
image building

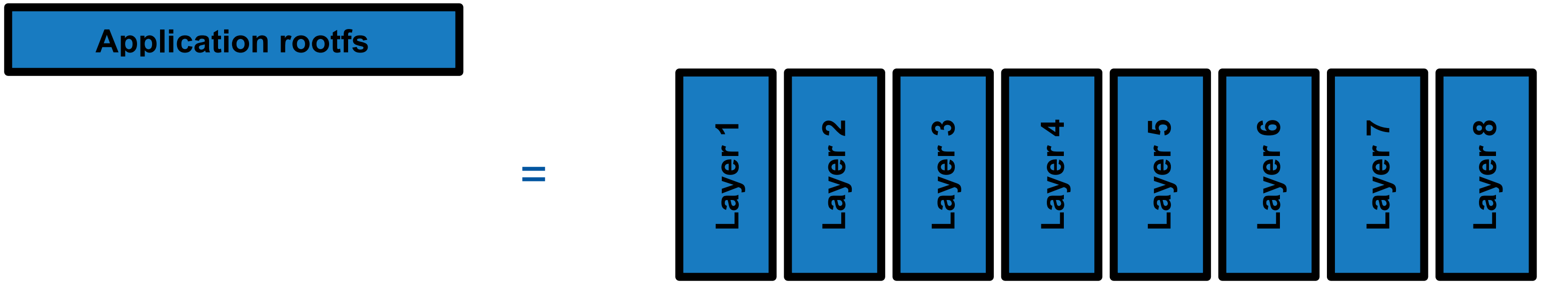
registry

Software Distribution today



Currently: distribution via HTTP of layer .tgz served by registry CDN

Image Distribution



Application rootfs

=

Layer 1

Layer 2

Layer 3

Layer 4

Layer 5

Layer 6

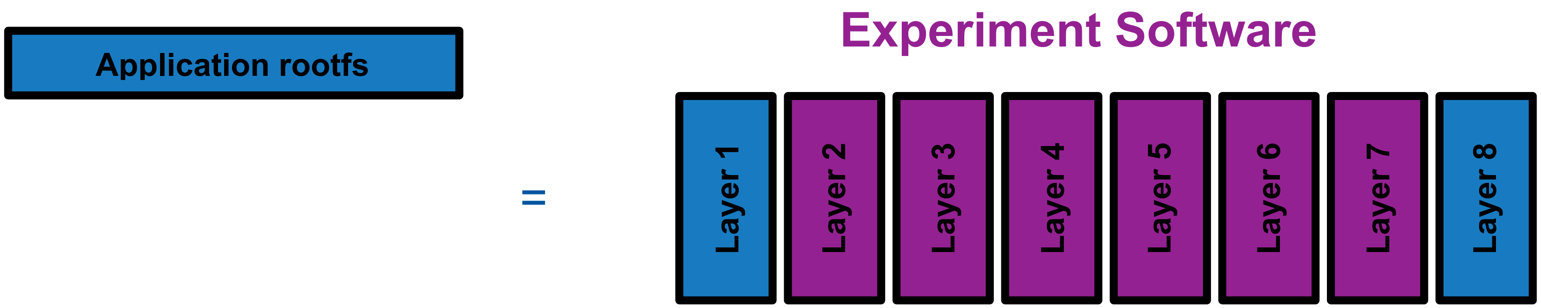
Layer 7

Layer 8

View the "image" not as a monolithic blob of layer data

- rather treat its manifest as a declaration of "intent" of what rootfs the user desires

Image Distribution



Application rootfs

=

Experiment Software

Layer 1

Layer 2

Layer 3

Layer 4

Layer 5

Layer 6

Layer 7

Layer 8

We know that images that users built will have significant overlap in the middle layers

- 90% of image size is in that middle part
- usually this layer is provided through a global read-only filesystem /cvmfs

- instead of exposing /cvmfs directly to users, can we distribute image files through /cvmfs?
  - best of both worlds: if /cvmfs available, use it as a CDN
  - if not available, pull full image

Image Distribution Using a Global Read-only FS

When constructing rootfs, container runtimes needs first acquire image data locally on the host and unpack

Idea: instead of downloading layer tarballs just use directories on global read-only filesystem

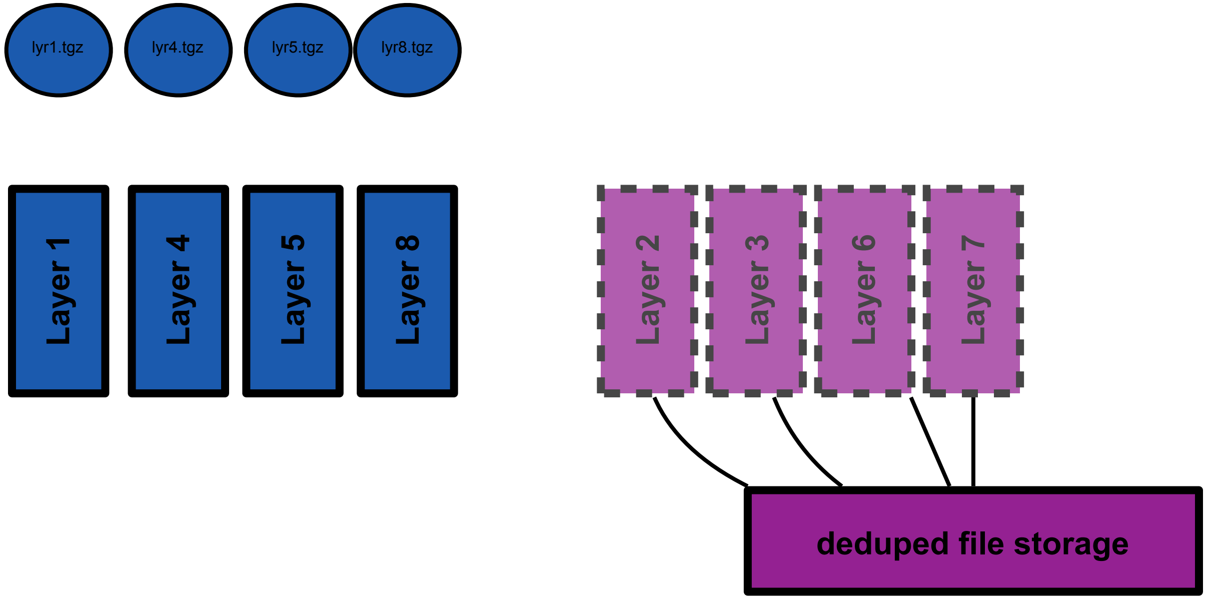
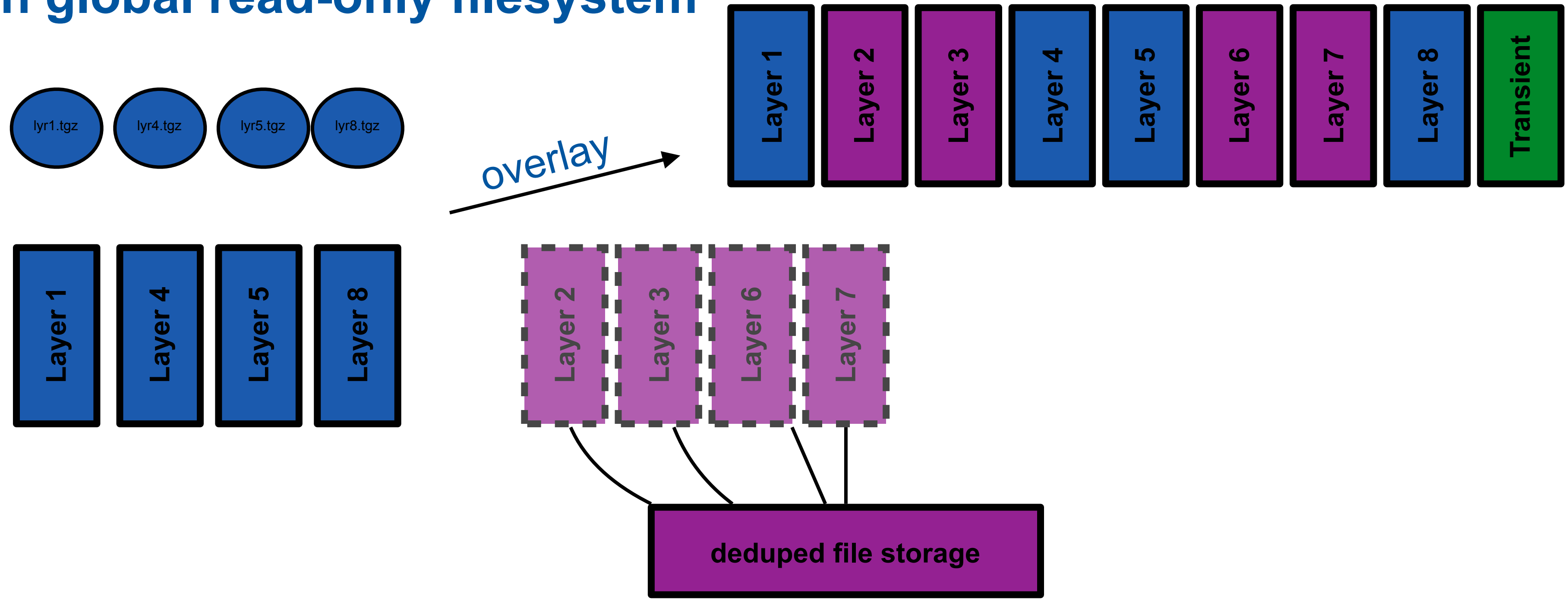
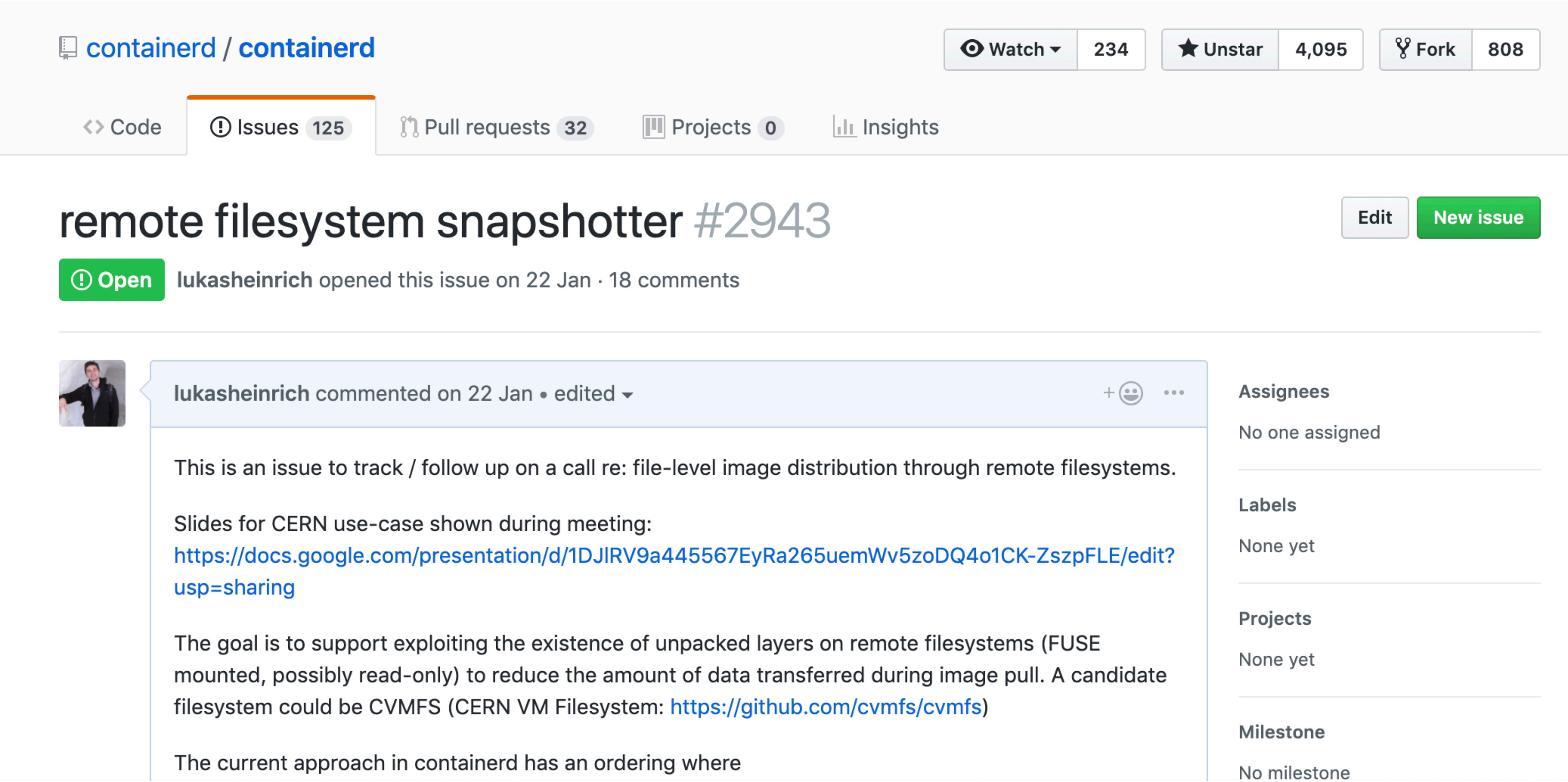


Image Distribution Using a Global Read-only FS

When constructing rootfs, container runtimes needs first acquire image data locally on the host and unpack

Idea: instead of downloading layer tarballs just use directories on global read-only filesystem





remote filesystem snapshotter #2943

lukasheinrich opened this issue on 22 Jan · 18 comments

lukasheinrich commented on 22 Jan • edited

This is an issue to track / follow up on a call re: file-level image distribution through remote filesystems.

Slides for CERN use-case shown during meeting:
https://docs.google.com/presentation/d/1DJlRV9a445567EyRa265uemWv5zoDQ4o1CK-ZszpFLE/edit?usp=sharing

The goal is to support exploiting the existence of unpacked layers on remote filesystems (FUSE mounted, possibly read-only) to reduce the amount of data transferred during image pull. A candidate filesystem could be CVMFS (CERN VM Filesystem: https://github.com/cvmfs/cvmfs)

The current approach in containerd has an ordering where

Assignees

No one assigned

Labels

None yet

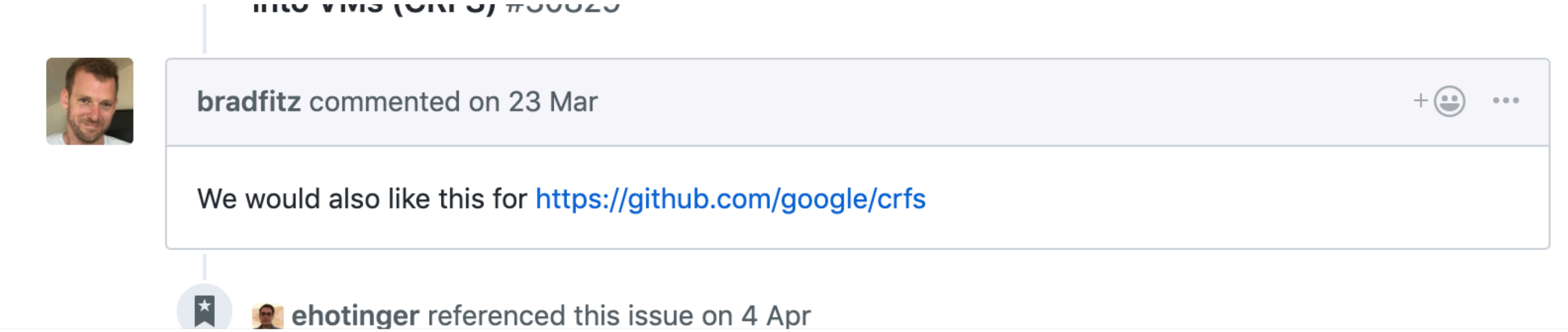
Projects

None yet

Milestone

No milestone

We're not alone

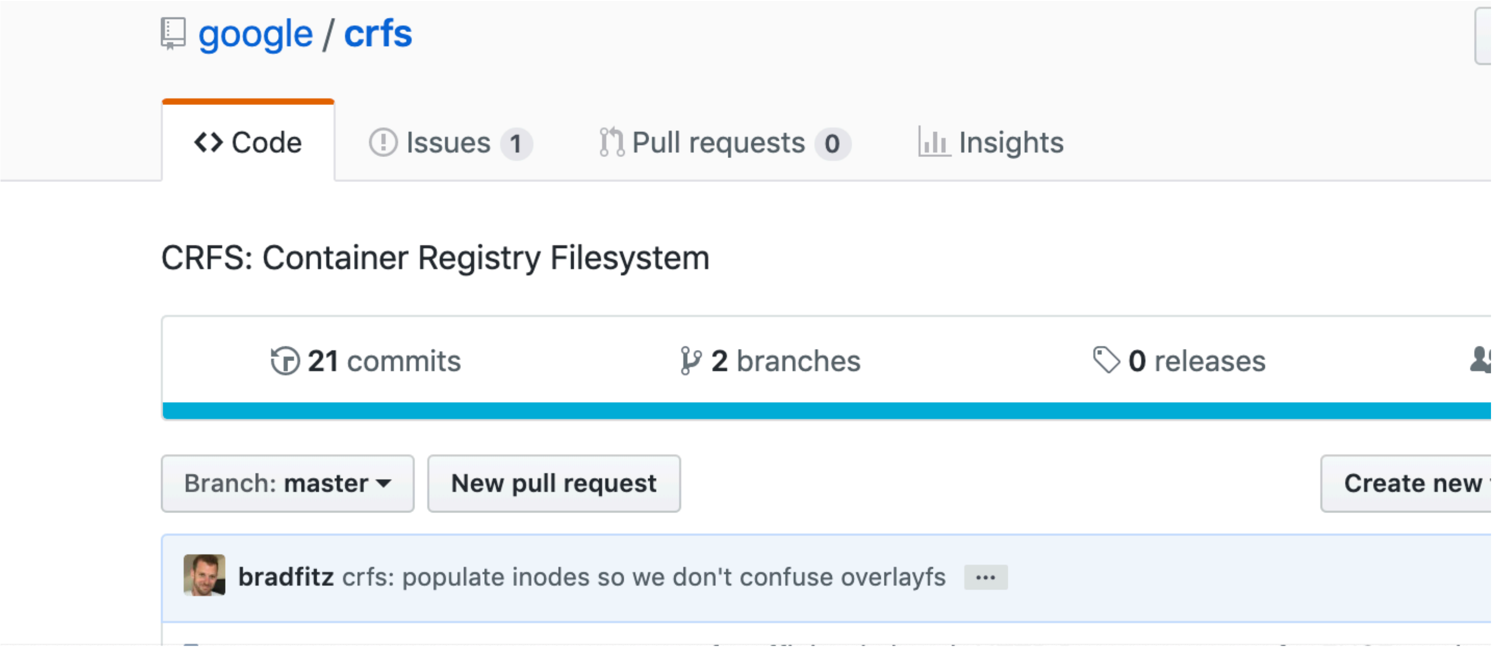


into VMs (CRFS) #30825

bradfitz commented on 23 Mar

We would also like this for https://github.com/google/crfs

ehotinger referenced this issue on 4 Apr



google / crfs

CRFS: Container Registry Filesystem

21 commits

2 branches

0 releases

bradfitz crfs: populate inodes so we don't confuse overlayfs

Conclusions

- Containers are a good abstraction. Overtook industry
- efficient distribution of Container images are an emerging problem
- Can use our long experience with read only global de-duplicated filesystems to serve container images efficiently
  - similar ideas in industry (google/crfs)
  - opportunity to work together