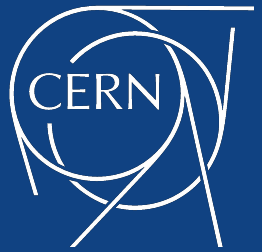


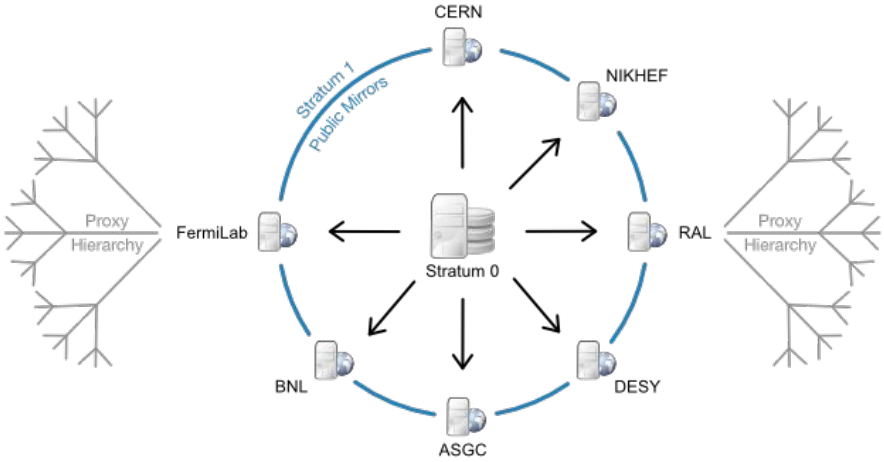
Lazy Image Pulling with Stargz - Spyros Trigazis, CERN

Second K8s-HEP Meetup 1-2 December 2020 https://indico.cern.ch/event/968726/

Lazy Image Pulling with Stargz

Spyros Trigazis, CERN

Software Distribution



CernVM-FS (CVMFS)

https://cernvm.cern.ch/fs/

Scalable software distribution for the Grid

POSIX read-only filesystem in user space

Aggressive caching, HTTP based

Question

Can we achieve the same efficiency for containerized workloads?

More Questions

Software packaged in container images

- How can we speed up container creation and startup?
- Images of 10s of gigabytes!

How can we reduce / optimize network usage?

Cluster auto scaling is a major topic

- How can we properly handle this with huge images?

Some History

FAST 16, Slacker: Fast Distribution with Lazy Docker Containers

https://www.usenix.org/conference/fast16/technical-sessions/presentation/harter

Docker CVMFS Graph Driver

https://github.com/cvmfs/docker-graphdriver

Lazy Pulling

Ongoing Work

Build on the existing CVMFS infrastructure for image distribution

https://github.com/cvmfs/containerd-remote-snapshotter

Today's presentation will focus on a more generic deployment

(e)stargz

Stargz Remote Snapshotter

Containerd Remote Snapshotter

- Based on seekable tar.gz (stargz)

  https://github.com/containerd/stargz-snapshotter https://github.com/google/crfs

- Proposed by Kohei Tokunaga, NTT

  https://kccnceu20.sched.com/event/ZepQ

Indexed files per image layer

Fuse mount per image layer

gRPC plugin for containerd

Stargz Remote Snapshotter

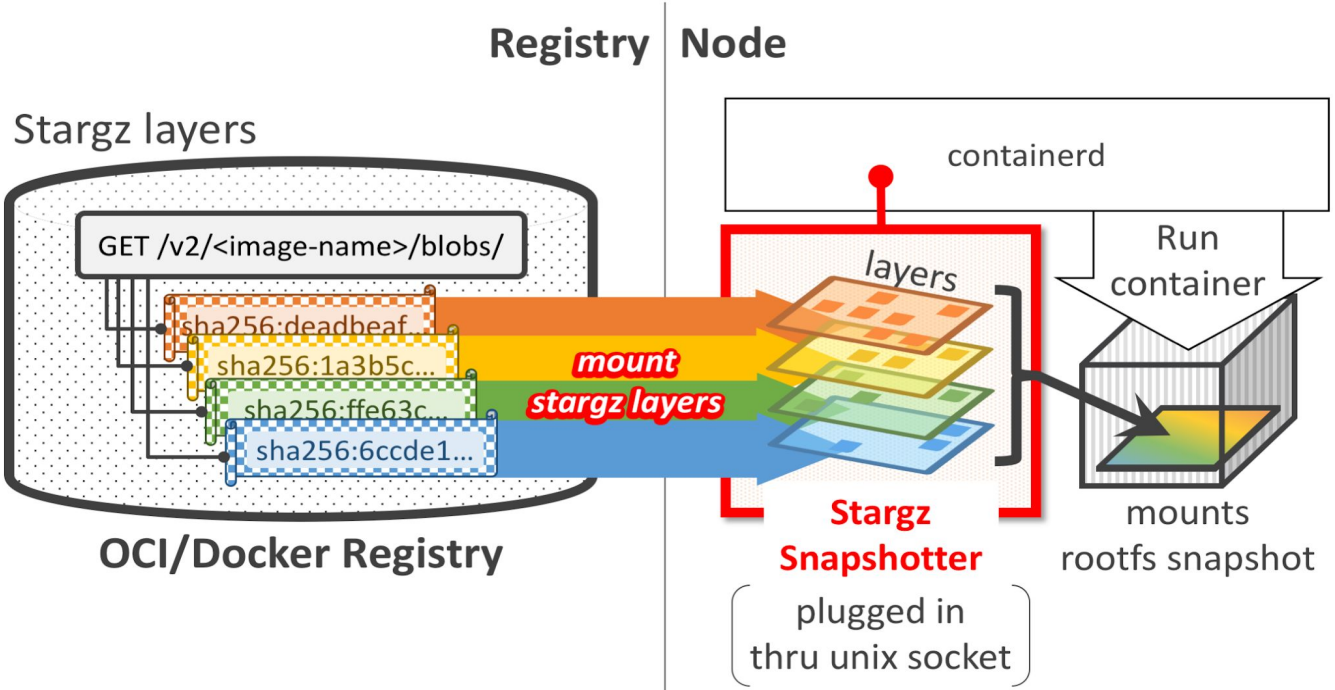


Image credit: The containerd Authors

Runtime statistics

Exec time, RAM, Network ingress

atlas/athena:21.0.15_100.0.2

17.2GB / 5.43GB

strigazi/athena:21.0.15_100.0.2-esgz-bash-version

17.2GB 5.56GB

mode	pulling time	RAM Containerd/ stanpshotter	Ingress on node	execution time workload
native	3m37s	257MB	5.84GB	7m15s
esgz	16s	1360MB	0.84GB	8m14s

- Fast startup time
- low network traffic (workload dependent)
- Memory consumption to be investigated
- 45m to convert to esgz

Demo

Status

Stargz remote snapshotter is already functional

- Super fast startup times

Reduced network usage

Low cpu overhead

Some registries do not support HTTP range queries RFC7233 [0][1]

- Gitlab Registry which we use extensively at CERN

https://docs.docker.com/registry/spec/api/#fetch-blob-part

https://tools.ietf.org/html/rfc7233#section-2.3

Improvements

Speed up image optimization, currently single core / serial

Allow (e)stargz builds with optimized base images

Mounted / external data during optimization step

Smaller Issues

- containerd fallback when remote snapshotter is down
- Further investigating needed for Harbor handling of large layers

Thanks

Akihiro Suda and Kohei Tokunaga, NTT

CVMFS Team at CERN

CERN Cloud Team

Participants of the May 2019 CERN Workshop on SW Distribution

https://indico.cern.ch/event/788994

References

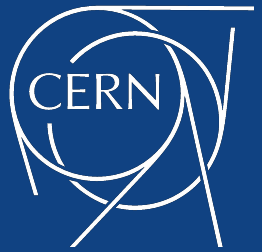
Speeding Up Analysis Pipelines with Remote Container Images

KubeCon NA 2020 https://sched.co/ekDj

OCI v2 images

https://groups.google.com/a/opencontainers.org/g/dev/c/Zk3yf45HIdA?pli=1

https://hackmd.io/@cyphar/ociv2-brainstorm


CERN