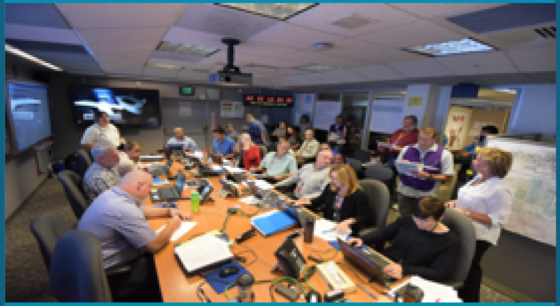
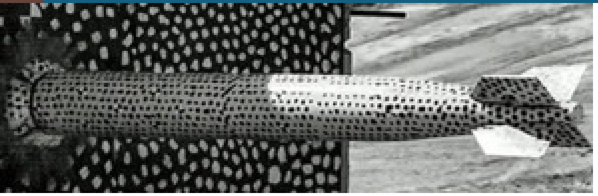
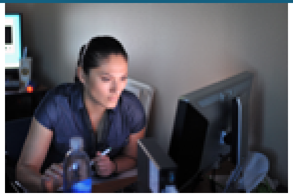
SAND2020-12022PE

Modern Container Runtimes for Exascale computing era







PRESENTED BY

Andrew J. Younge, PhD

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

High Performance Computing

- DOE has a long history of investment in HPC
  - Stockpile Stewardship
    - ASCI Red – first Teraflop Supercomputer
    - Red Storm - 100 Tflops and MPP
    - Sierra - 200 Pflops
  - Leadership-class science
    - Summit, Cori, Titan, …
  - Bulk synchronous parallel computing ~ HPC
    - 1 application spanning thousands of CPUs concurrently
    - Large-scale capability simulations
    - Also many capacity workloads
  - HPC represents the pinnacle of computing today

- HPC computational requirements demand scale
  - Tightly coupled BSP simulation codes typically use MPI for communication
  - Many workload ensembles quickly expanding to ML/DL/AI

HPC in the Cloud?

- Public cloud computing is often prohibitive
  - Cost – expensive to run millions of CPU hours
  - Security – Can we trust public clouds?
- However, HPC is not traditionally as flexible as "the cloud"
  - Shared resource models
  - Static software environments
  - No best fit for emerging apps and workflows
- What about Containers?
  - Can we support containers in HPC in the same way as clouds do?
  - Does this model fit for _both_ HPC and emerging workloads across DOE?
  - Can we adapt our programming environments into container images?

ECP Supercontainers

- Joint DOE effort - LANL, LBNL, LLNL, Sandia, U. of Oregon
- Ensure container runtimes will be scalable, interoperable, and well integrated across DOE
  - Enable container deployments from laptops to Exascale
  - Assist Exascale applications and facilities leverage containers most efficiently
- Three-fold approach
  - Scalable R&D activities
  - Collaboration with related ST and AD projects
  - Training, Education, and Support
- Activities conducted in the context of interoperability
  - Portable solutions
    - Optimized E4S container images for each machine type
    - Containerized ECP that runs on Astra, A21, El-Capitan, …
  - Work for multiple container implementations
    - Not picking a “winning” container runtime
  - Multiple DOE facilities at multiple scales

SUPERCONTAINERS

ECP
EXASCALE COMPUTING PROJECT

Astra: The First Petascale Arm Supercomputer

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY NNSA

SAND2020-10998 C

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525 Unclassified Unlimited Release .

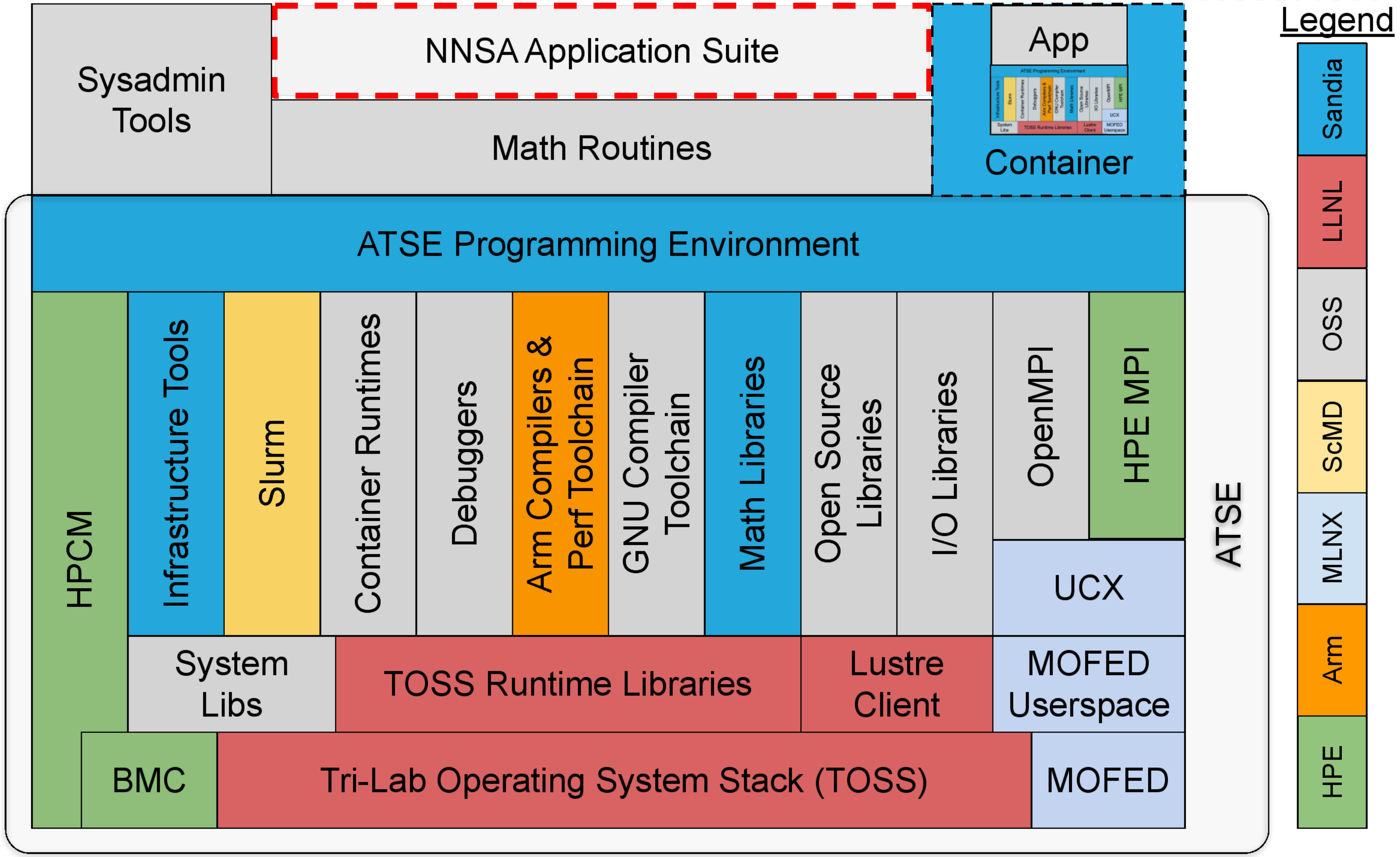
Vanguard Astra At a Glance

Hewlett Packard Enterprise

- 2,592 HPE Apollo 70 compute nodes
  - 5,184 CPUs, 145,152 cores, 2.3 PFLOPS (peak)
  - 36 compute racks arranged in two rows
  - Full system power ~ 1.2 MW
- Marvell ThunderX2 ARM SoC, 28 core, 2.0 GHz
  - 2 sockets per node
- Memory per node: 128 GB
  - 16 channels of 8 GB DDR4-2666 DIMMs
  - Aggregate capacity: 332 TB, 885 TB/s (peak)
    - 247 GB/s per node STREAM TRIAD
- Mellanox IB EDR, ConnectX-5
  - 112 36-port leaf, 3 648-port spine switches
  - SocketDirect capability
- ATSE software stack
  - Extensible user programming environment
  - TOSS Base Operating system
- HPE Apollo 4520 All–flash Lustre storage
  - Storage Capacity: 990 TB (usable)
  - Storage Bandwidth: 250 GB/s
    - 400 GB/s stunt mode, 432 GB/s peak

ATSE: Advanced Tri-lab Software Environment

- ATSE is a collaboration with HPE, OpenHPC, and ARM
- Many pieces to the software stack puzzle
- HPE's HPC Software Stack
  - HPE Cluster Manager
  - HPE MPI (+ XPMEM)
- Arm
  - Arm HPC Compilers
  - Arm Math Libraries
  - Arm Allinea Tools
- Open source tools - OpenHPC
  - Slurm, OpenMPI, etc.
- Mellanox-OFED & HPC-X
- RedHat 7.x for aarch64 – TOSS



ARM and Containers – will it blend?

- Little understanding of container mechanisms on Arm
  - Especially for HPC
  - It should ‘just work’ - but does it?

- Action Plan:
  - Draw from existing containers & virtualization R&D at Sandia
  - Leverage Astra and various Arm testbeds
  - Develop initial container workflow model
    - Build ARM-based containers?
    - How to map such containers to Astra?
    - Mechanisms to deploy mission applications?
  - Will it ~~blend~~ scale?

Podman for Un-privileged Container Builds

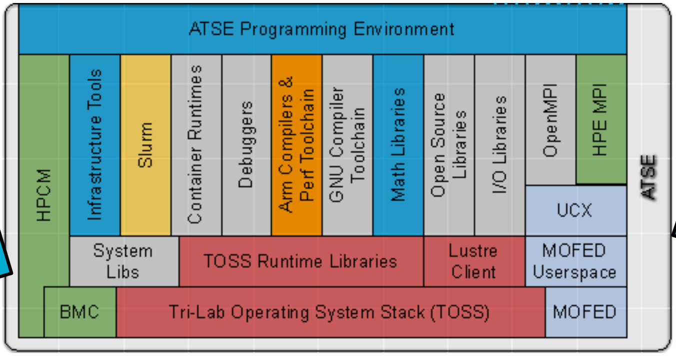
- Cannot build a container for Astra from my laptop
- Need to build containers directly on the supercomputer
  - Doing so requires root level privs
  - root in HPC is bad, Docker is root equivalent
- Leverage user namespaces for _building_ containers
- Podman and Buildah provide container builds while maintaining user-level permissions
  - Podman is CLI equivalent to Docker
  - User namespaces
  - Set uid/gid mappers
  - TBD Overlay & FUSE for mount
- Ongoing Collaboration with

  RedHat & Singularity folks

salloc -N 2048 && mpirun -np $NP singularity exec atse-astra-1.2.4.sif /app

singularity build atse-astra-1.2.4.sif docker://gitlab.doe.gov/atse/astra:1.2.4

podman push gitlab.doe.gov/atse/astra:1.2.4

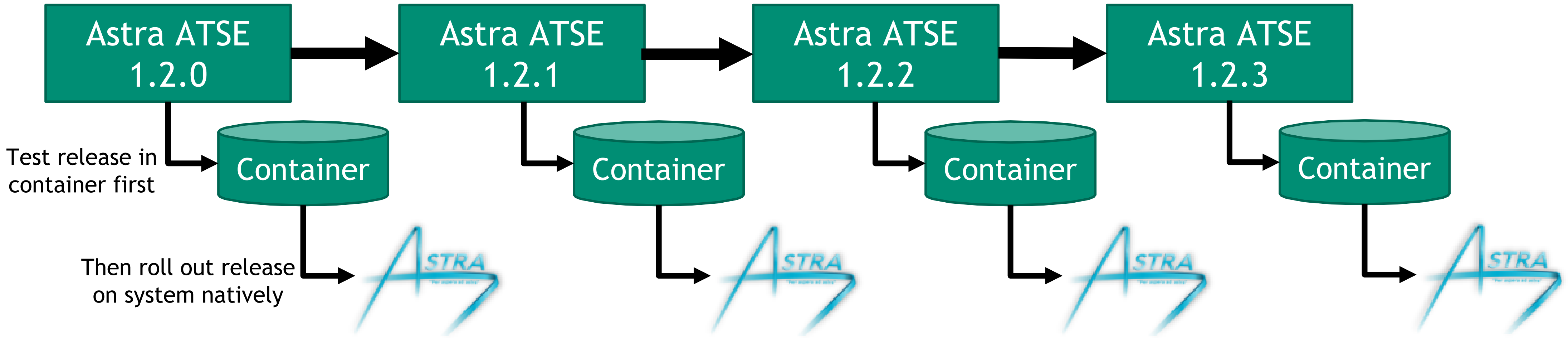


podman build -t “gitlab.doe.gov/atse/astra:1.2.4” .

Pedretti, Younge, et. al, *Chronicles of Astra: Challenges and Lessons from the First Petascale Arm Supercomputer* , to appear in SC20

Containers for Software Testing & Debugging

- Astra ATSE programming environment release consists of:
  - TOSS base operating system + Mellanox InfiniBand stack
  - {2 compilers} * {3 mpi implementations} * {~25 libraries} = 150 packages
  - Each release packaged as a container for testing and archival purposes



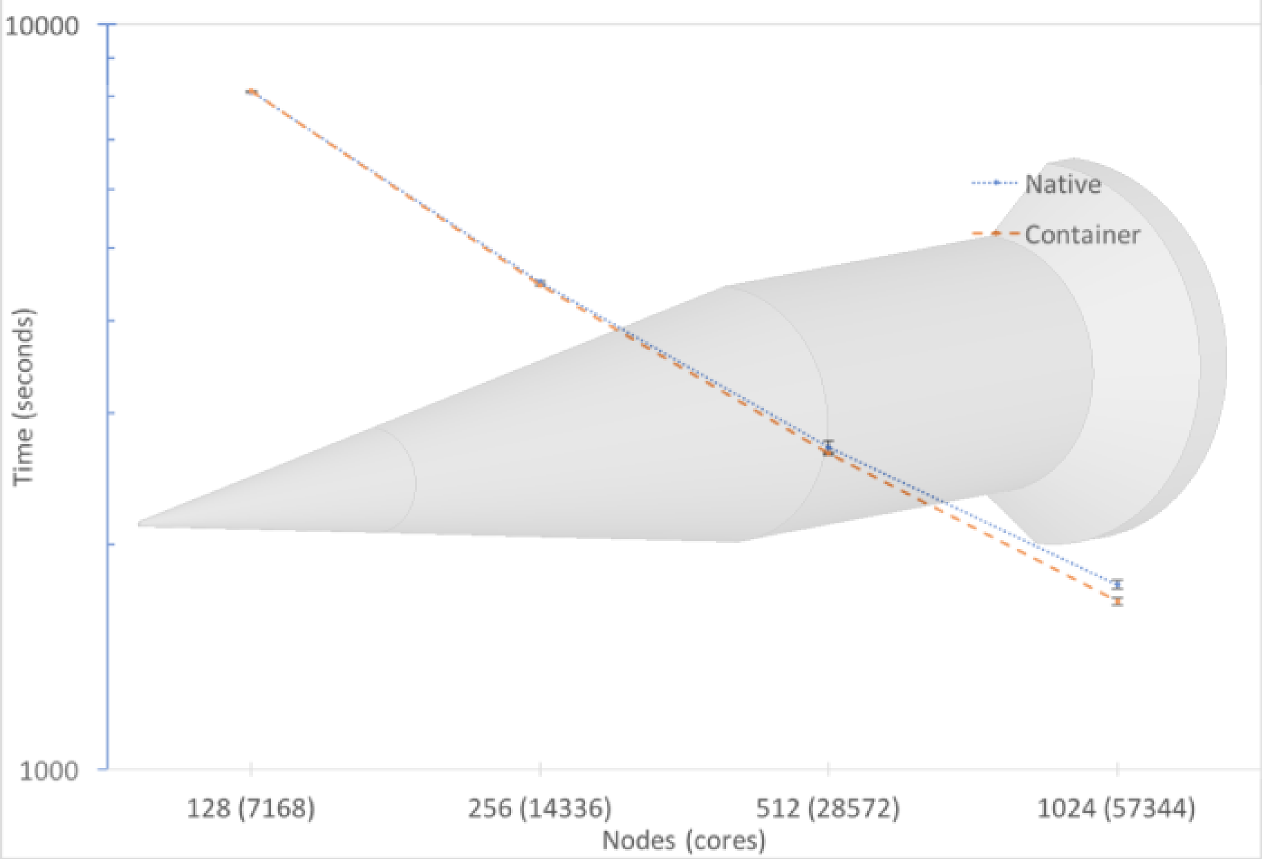
- ATSE Container use cases:
  - **Release testing:** Enables full applications to be built and run at scale (2048+ nodes) before rolling out natively
  - **Rollback debug:** If issues are identified, ability to easily go back to a prior software release and test
  - **Cross-system synchronization**: Move full user-level software environments between systems. In one instance, this allowed an Astra InfiniBand library bug to be debugged off platform on another Arm cluster

SPARC Demonstration with Containers

- SPARC containerized build & deployment
- Container image build with Podman
- Container on Astra with Singularity
  - Up to 2048 nodes
  - Largest non-x86 container deployment
  - Testing HIFiRE-1 Experiment (MacLean et al. 2008)
- Unable to determine any significant overhead by running Singularity containers
  - Confirm previous assertion from LANL (Torrez et al. 2019)
  - Can use multiple containers to compare performance in build variations

Nodes	Cores	Native (s)	Container (s)	Percent Diff
128	7168	8110	8119	+0.1%
256	14336	4501	4461	-0.9%
512	28672	2702	2651	-1.9%
1024	57344	1767	1705	-3.6%
2048	114688	1412	1429	+1.2%

Containers also useful for quickly testing PE changes



- Near-native performance using a container
- Performance difference within app variation at scale
- Above chart show container built identical to native but with new compiler optimizations
  - Container testing new optimizations for TX2 CPUs

Container Performance Portability Continuum

How do we strike the right balance?

- Portable container images can be moved form one resource deployment to another with ease
- Reproducibility is possible
  - Everything (minus kernel) is self-contained
  - Traceability is possible via build manuscripts
  - No image modifications
- Performance can suffer – no optimizations
  - Can't build for AVX512 and run on Haswell
  - Unable to leverage latest GPU drivers

- Performant container images can run at near-native performance compared to natively build applications
- Requires targeted builds for custom hardware
  - Specialized interconnect optimizations
  - Vendor-proprietary software
- Host libraries are mounted into containers
  - Load system MPI library
  - Match accelerator libs to host driver
- Not portable across multiple systems

Emerging workloads on HPC with Containers

- Extreme-scale Scientific Software Stack (**E4S**)
  - Container image contains everything and the kitchen-sink
    - Includes all ECP software activities
  - Lightweight base images now available
- Support merging AI/ML/DL frameworks on HPC
  - Containers may be useful to adapt ML software to HPC
  - Already supported and heavily utilized in industry
- Working with DOE app teams to deploy custom ML tools in containers
- Investigating scalability challenges and opportunities

Credit: Sameer Shende (U. Oregon)

Spack environments help with building containers

- We recently started providing base images with **Spack** preinstalled.
- **Very** easy to build a container with some Spack packages in it:

spack-docker-demo/
  Dockerfile
  spack.yaml

```
FROM spack/centos:7

WORKDIR /build
COPY spack.yaml .
RUN spack install
```

Base image with Spack in PATH

Copy in spack.yaml
Then run `spack install`

```
spack:
  specs:
    - hdf5 @1.8.16
    - openmpi fabrics=libfabric
    - nalu
```

List of packages to install, with constraints

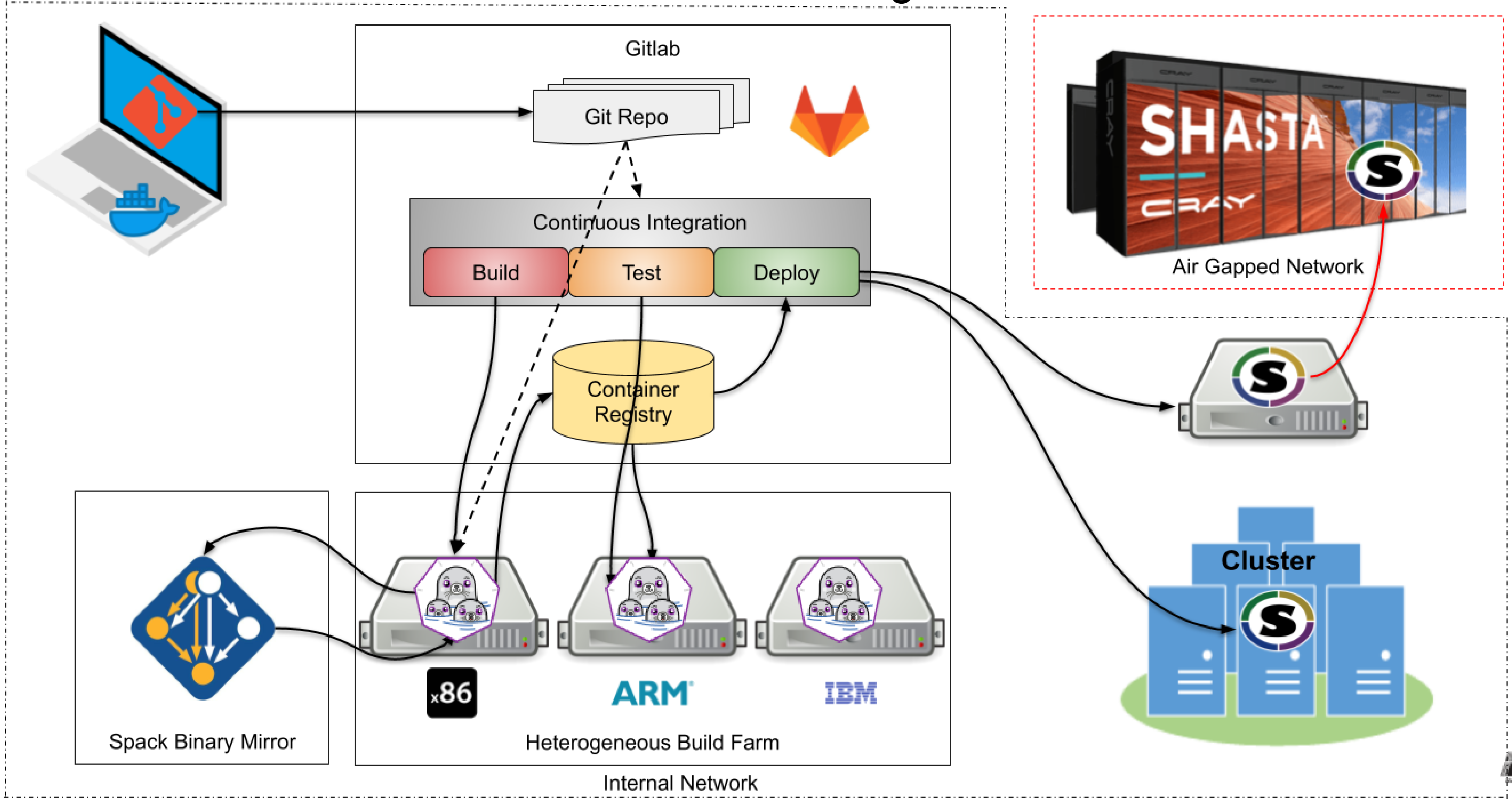
Build with `podman build .`

Run with Singularity (or some other tool)

Credit: Todd Gamblin (LLNL)

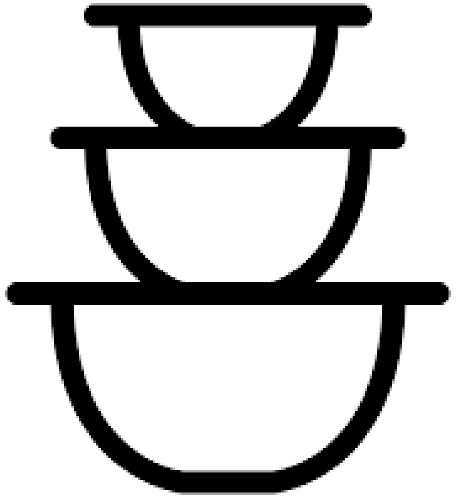
Containerized CI Pipeline

- As a *developer* I want to *generate container builds from code pull requests* so that *containers are used to test new code on target HPC machines.*



Container Takeaways (Tupperware?)

- Use ~~Docker~~ Podman to build manifests of full apps
    - Developers specify base OS, configuration, TPLs, compiler installs, etc
    - Use base or intermediate container images (eg: TOSS RPMs in a container)
  - Leverage container registry services for storing images
  - Import/flatten OCI images into Singularity & run on HPC resources
    - Also works for Charliecloud and Shifter
    - Can Podman also be used for scalable launch in the future?
- Containers have demonstrated minimal overhead for HPC apps
- Enabling On-prem unprivileged containers builds
  - More to come with Podman & Buildah for HPC
- HPC Container Advantages
  - Simplify deployment to analysts (just run this container image)
  - Simplify new developer uptake (just develop FROM my base container image)
  - Decouple development from software release cycle issues
  - Reproducibility has a new hope?
- Supercontainers for Exascale
  - Preparing to enable containers at Exascale
  - Simplify HPC application deployment via modern DevOps
  - Support next generation AI & ML apps



EXASCALE COMPUTING PROJECT

SUPERCONTAINERS

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories

Thanks!

ajyoung@sandia.gov

ATSE Provided a Focal Point for Development

- Curated HPC software stack
  - Provides base set of compilers and MPI implementations known to work well together
  - Didn’t want users to have to build entire third-party library (TPL) stack themselves for Arm
- Labor intensive to assemble and test
  - Leveraged OpenHPC recipes to speed development, extremely helpful
  - Complicated by specific version requirements, μarch optimizations, static library support
- ATSE continuing to evolve
  - Moving to Spack-based build process to improve development speed + flexibility
  - Improving container packaging and deployment
  - Shifting towards Vanguard-II
- Effort to standardize programming environment components

ATSE 1.2.5 Recipes Available @ https://doi.org/10.5281/zenodo.4006668

Conclusion

- Astra the first Petascale supercomputer based on 64-bit Arm processors
  - First Vanguard Advanced Prototype platform for DOE/NNSA
  - Now running production applications for NNSA mission
- Several technology gaps were identified and addressed
  - Software stack enablement, Linux bugs at scale, thermal management considerations, power management capabilities, parallel filesystem support, and enabling advanced container support.
  - Pushed fixes back into community
  - Focus on Arm compiler maturity
- Several lessons learned applicable to any first-of-a-kind Supercomputer
  - First DOE Exascale platforms
  - Future Vanguard II+ platforms
  - RISC-V and other custom designs

Acknowledgements:
Computer Systems & Technology Integration group (9320)
Extreme Scale Computing group (1420)
IT Infrastructure Services group (10770)
NNSA Tri-lab collaborators (LANL and LLNL)
Industry Partners: HPE, Mellanox, Red Hat, Arm, and Marvell

Arm is now a viable production HPC technology for the largest-scale supercomputers

See Fugaku - #1 Top500

Position 2: … and so is the Cloud

- The hyperscalers are finally paying attention to HPC
  - *“The physical network topology does affect performance; particularly important is the performance of MPI Allreduce, grouped by splitting the mesh by a subset of the dimensions, which can be very efficient [5] [6] if each such group is physically connected.”* – Shazeer et al Google Brain, Mesh-TensorFlow: Deep Learning for Supercomputers.
  - As learning techniques grow in scale, HPC becomes more important.

- HPC cannot compete with the hyperscalers
  - Let’s stop trying and start *integrating*
    - *That doesn’t mean adopting Cloud as-is*
    - *That doesn’t dissolving HPC either*
  - The closer HPC and cloud paradigms get, the better we all are
    - Encourage open source infrastructure
    - Collaborative partnerships
  - Avoid boutique solutions without sacrificing performance