SAND2022-7113C

Sandia National Laboratories

Exceptional service in the national interest

The Design and Development of ATSE: Advanced Tri-lab Software Environment

May 2, 2022

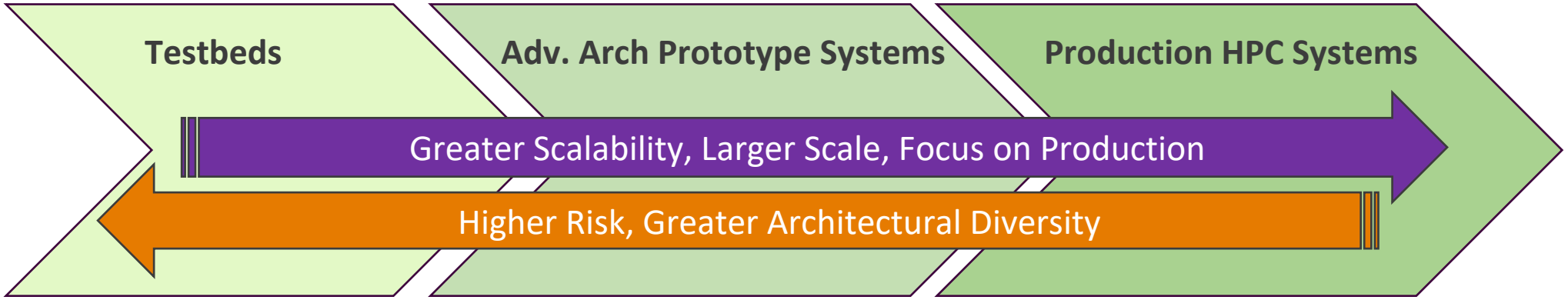
PI: Matthew Curry, Mike Aguilar, Si Hammond, James H. Laros III, Mike Levenhagen, Kevin Pedretti, Andrew Younge

Center for Computing Research
Sandia National Laboratories
Albuquerque NM
mlcurry@sandia.gov

SAND2022-5655 PE

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.

Spectrum of HPC Systems – Technology Maturation Path



Test Beds

- Small testbeds (~10-100 nodes)
- Breadth of architectures Key
- Brave users

Vanguard

- Larger-scale experimental systems
- Focused efforts to mature new technologies
- Broader user-base
- Not production, seek to increase technology and vendor choices
- **DOE/NNSA Tri-lab resource**

Production Platforms

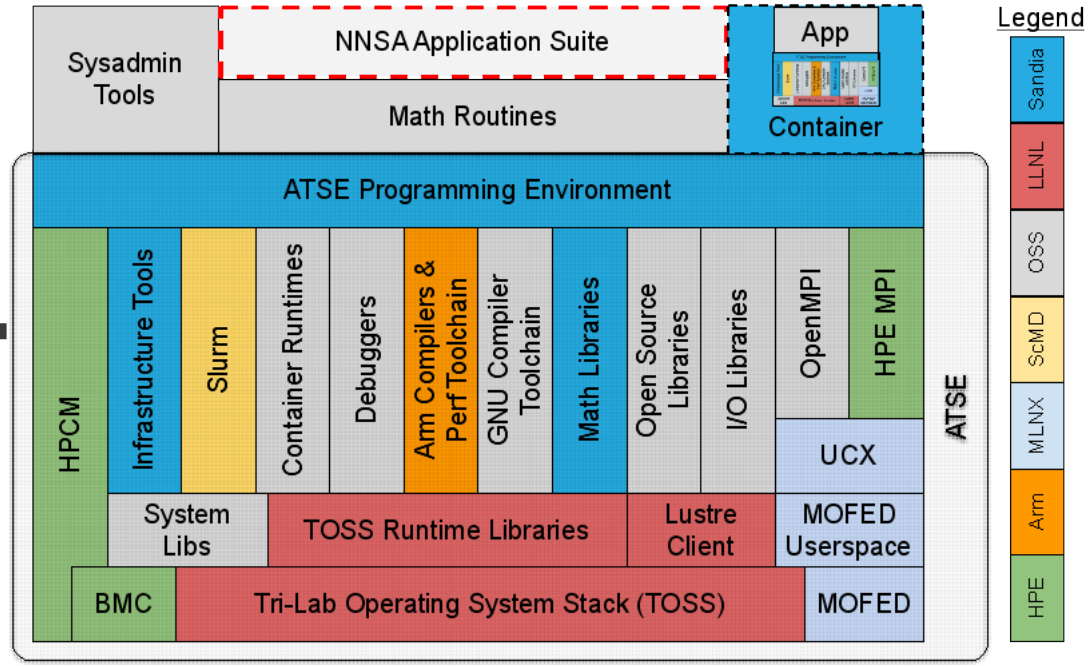
- Leadership-class systems (Petascale, Exascale, ...)
- Advanced technologies, sometimes first-of-kind
- Broad user-base
- Production use

Astra was the first Vanguard Platform

The ATSE Software Stack

- Modular, extensible, and open HPC software stack
  - Provide operational independence from any single vendor, encourage vendors to add value
  - Focal point for collaboration activities to mature new technologies (HW + SW)
- Prototype software stack for prototype systems: *Adv Arch Prototype Systems (AAPS), Vanguard2, Testbeds, Arm+GPU, A64FX, etc.*

ATSE:
Advanced Tri-lab Software Environment



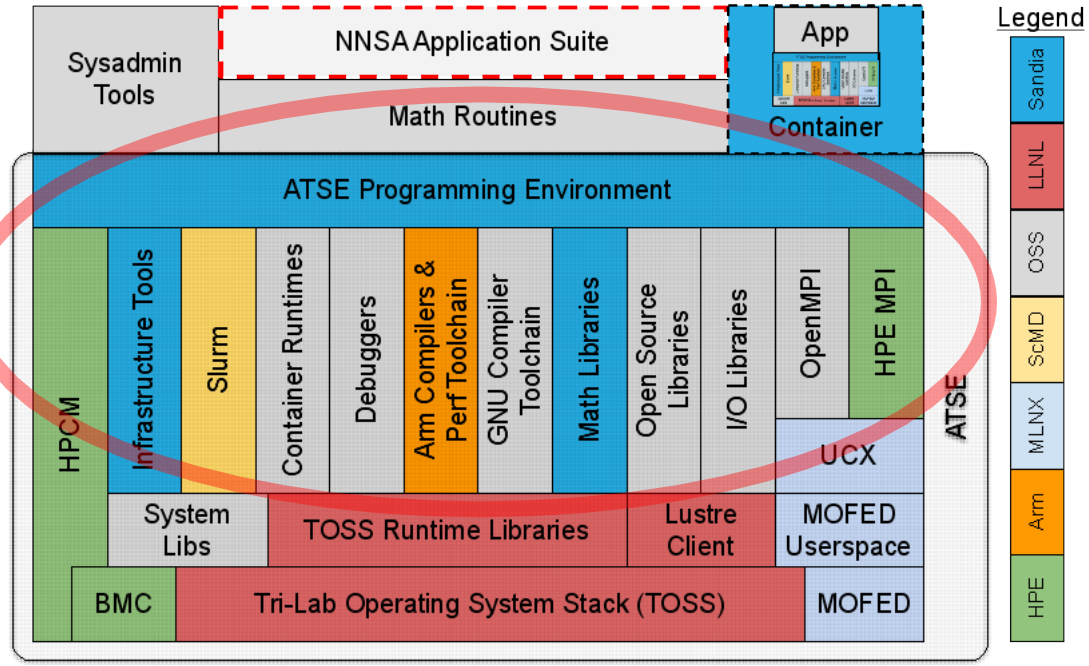
NNSA/ASC Astra Petascale Arm Supercomputer

More details in SC20 paper – Chronicles of Astra: Challenges and Lessons

Scalable Programming Environment

- Curated HPC software stack
  - Provides base set of compilers, MPI implementations, third-party libraries, tools, and other components known to work well together
  - Focused on needs of Sandia / DOE / NNSA / ASC codes
- Especially important for immature technologies
  - Many bugs, broken packages, and missing functionality
  - Need to do more to help users, avoid duplicated work
- Look and feel similar to OpenHPC, adapted for ASC:
  - Pin packages at specific versions, per code team requirements
  - Add missing packages (e.g., ParMETIS, CGNS)
  - Add microarchitecture and compiler optimizations
  - Add static library support, simplifies moving binaries between networks

ATSE:
Advanced Tri-lab Software Environment



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

ATSE Provides "Ready to Go" Programming Environment for ASC Codes

Building ATSE with Spack

- Now using Spack to build ATSE Prg. Env.
  - Developed automated workflow for generating reproducible builds with same look and feel
  - Combines curation of ATSE with power of Spack
  - Build time reduced to 3 hours (was > 24 hours)
  - Used successfully on Arm+GPU and A64FX

- ATSE contributions to Spack

Package version bumps	12
Variant additions	17
Package additions	2
Core bug fixes	1
Major feature additions	2 (pending)

Package install metrics (#14705)
Shared spack instances (#11871)

ATSE Shared Spack Instance Workflow

User issues `module load spack`

System Spack installation, provided by ATSE
"Batteries included" Spack installation
Install locations, mirrors, compilers, etc.

User issues `spack install trilinos`

User Spack instance
Custom software selection installed per-user
/home/joe/.spack/

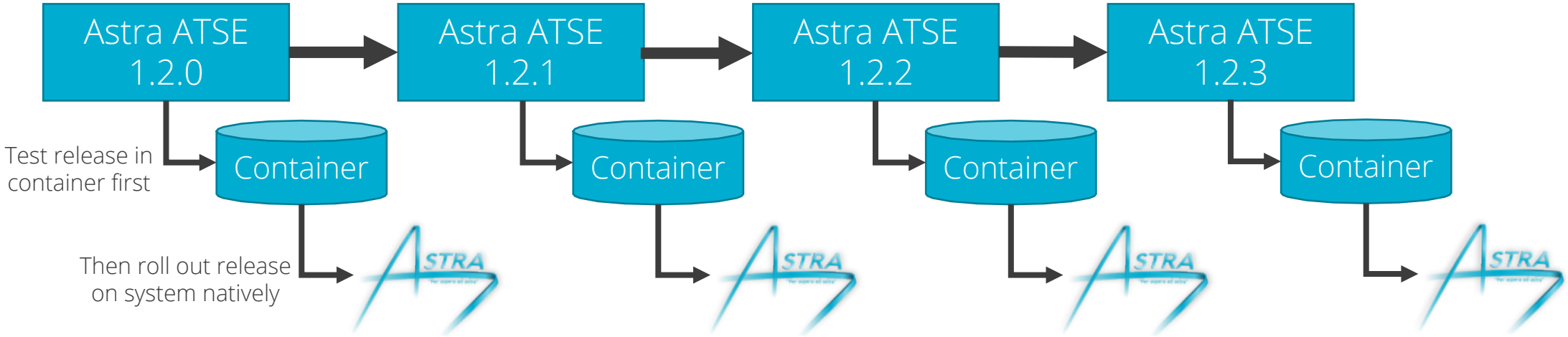
Trilinos depends on openblas, which is in ATSE

ATSE Spack instance
System-wide, optimized, supported
/opt/atse/openblas/0.3.4

ATSE is Leveraging and Contributing New Capabilities to Spack

How Containers were used for 1.2.x

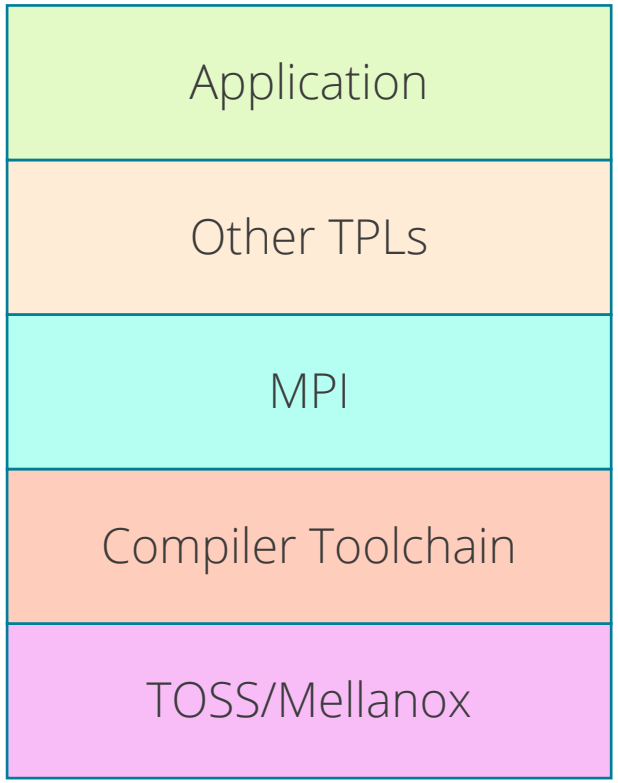
- Astra ATSE programming environment release consists of:
  - TOSS base operating system + Mellanox InfiniBand stack
  - {2 compilers} * {3 mpi implementations} * {~25 libraries} = 150 packages
  - Each release packaged as a container for testing and archival purposes



- ATSE Container use cases:
  - **Release testing:** Enables full apps build 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, it allowed an Astra InfiniBand library bug to be debugged off-platform on another Arm cluster.

Ramping Up Containers for Vanguard2

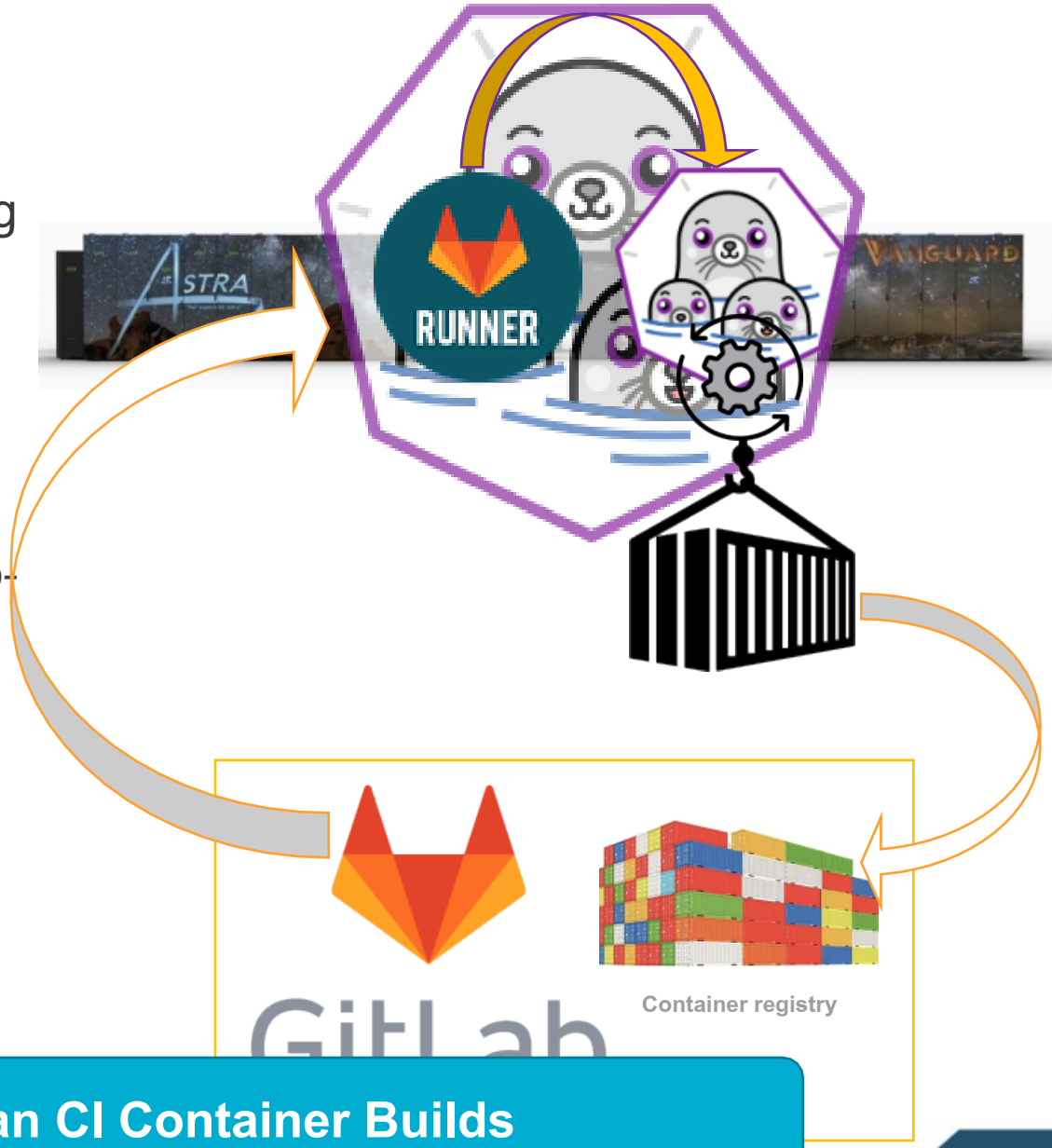
- Future Vanguard systems will be "containers first"
  - Each application will be able to bring in a tailored, lightweight version of ATSE with the application
- Layers will be used to encourage re-use and reduce container weight
  - Most containers will have a single compiler/MPI
  - A "supercontainer" containing everything is also available for development purposes
- Use reproducible build techniques to get the same* ATSE in all configurations



ATSE Treats Containers as First-Class Support Targets

Introduced Containerized CI

- Need to leverage Continuous Development and Continuous Integration capabilities
- Gitlab CI has git-lab runners, but expect long-running daemons with elevated privileges
  - Challenging to run on HPC systems, ATSE's desired target
- Solution: Podman-in-Podman
  - The gitlab runner built in a OCI container image
  - Run Rootless Podman on a compute node to have gitlab-runner think it has root privs
  - Automate building images and testing
- Solution appliable to SNL as well as greater DOE infrastructure
  - Future integration with DOE Jacamar runners



ATSE is Pioneering Podman-in-Podman CI Container Builds

Conclusion

ATSE provides a common set of well-supported packages to bleeding-edge platforms

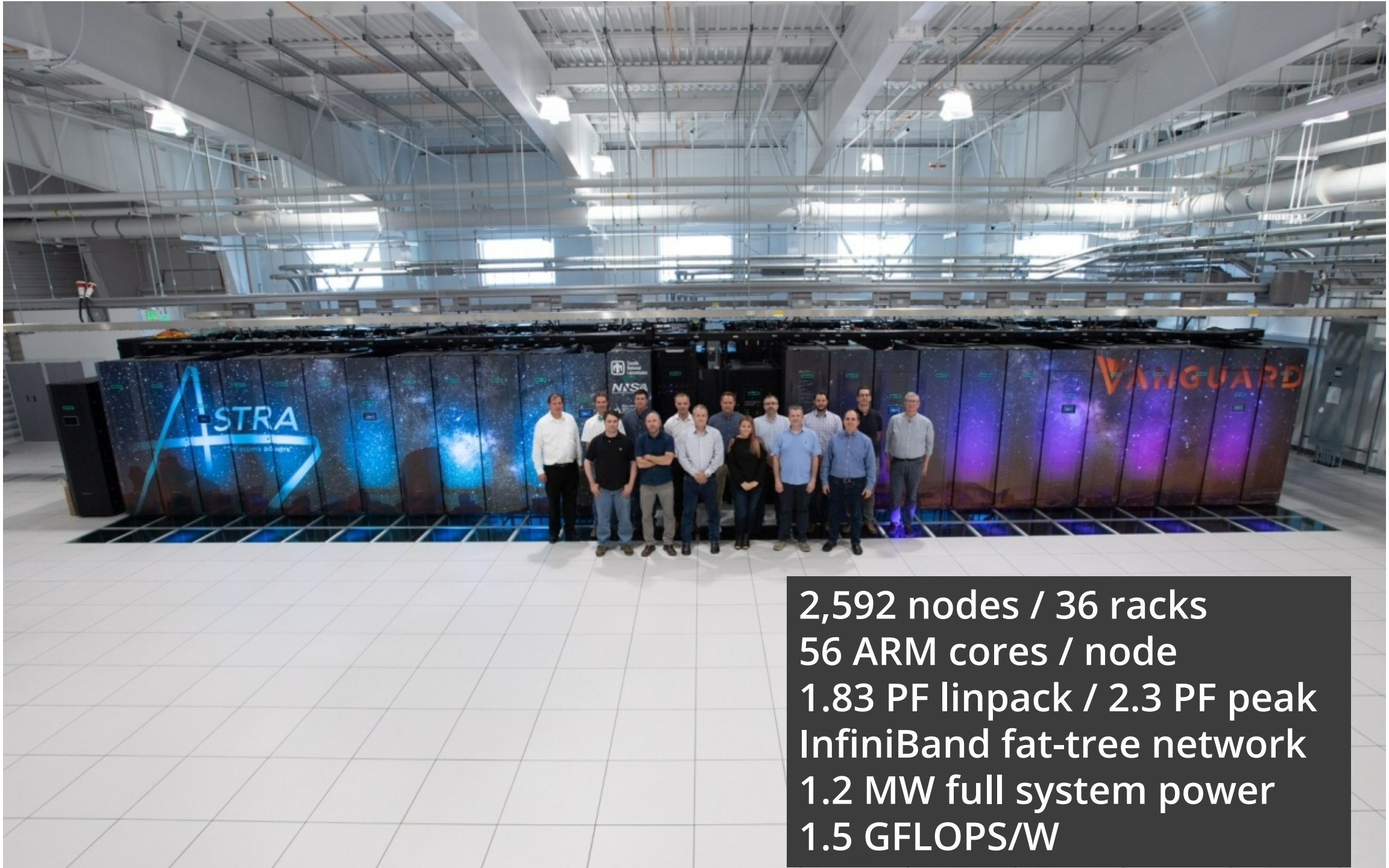
- Designed to be the first stack on a new system
- Agile and composable with vendor stacks
- A vehicle for co-design, innovation, and collaboration

ATSE is now primarily Spack-based

- Increased ability to replicate outside of Sandia
- Take advantage of others' developments
- Contribute back needed enhancements for new architectures

Doubling down on container strategies

- Containers first
- Containerized CI's



2,592 nodes / 36 racks
56 ARM cores / node
1.83 PF linpack / 2.3 PF peak
InfiniBand fat-tree network
1.2 MW full system power
1.5 GFLOPS/W