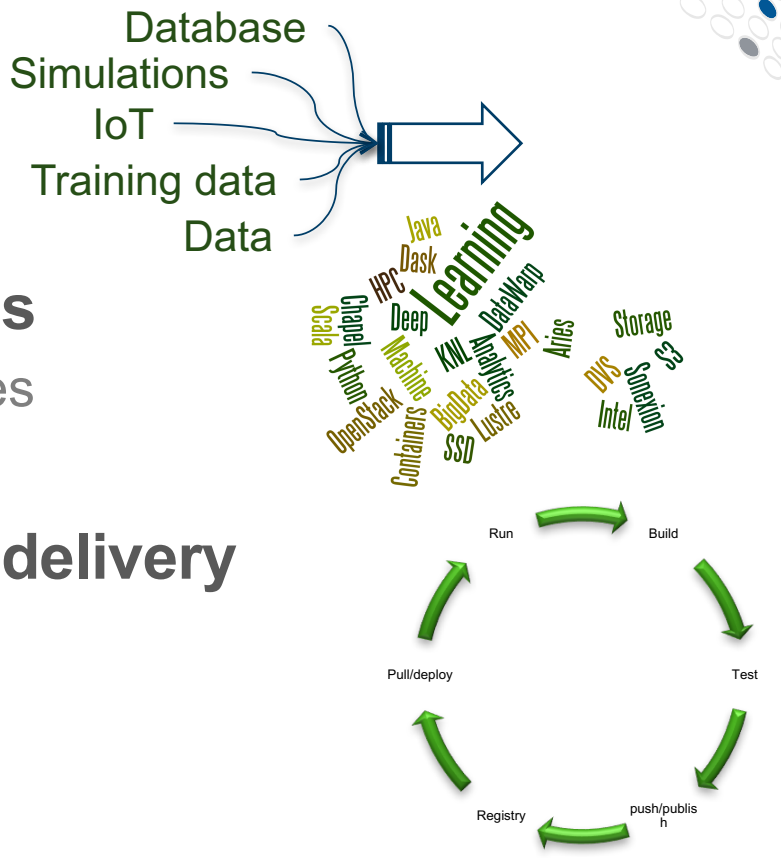
User-mode Containers: Keepin' it Real . . . and Contained

Larry Kaplan
Chief Software Architect

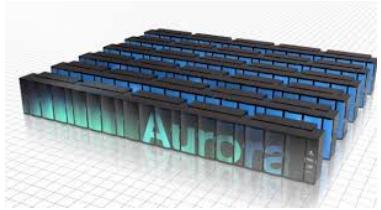
Disruptive Environment

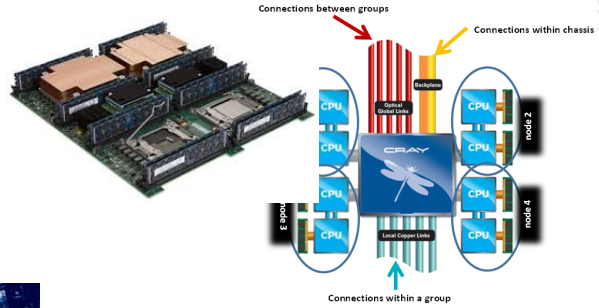
- Increasing volumes of data
- Addition of analytic workloads
  - New models/application/languages
- Desire for simple application delivery and orchestration

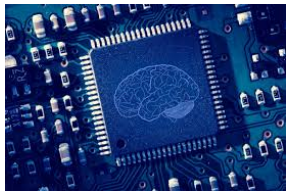


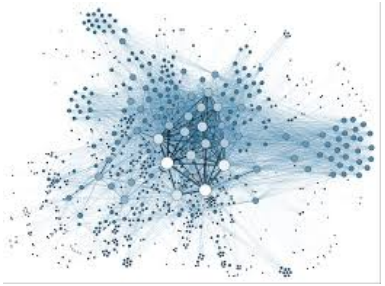
Converging HPC and Analytics platforms

- **HPC**
  - Large scale
  - Fast networks
  - Fast storage
  - Accelerators
- **Analytics**
  - Machine/Deep learning
  - Data manipulation/visualization
  - Analytic frameworks
- **Virtualization/Containers**





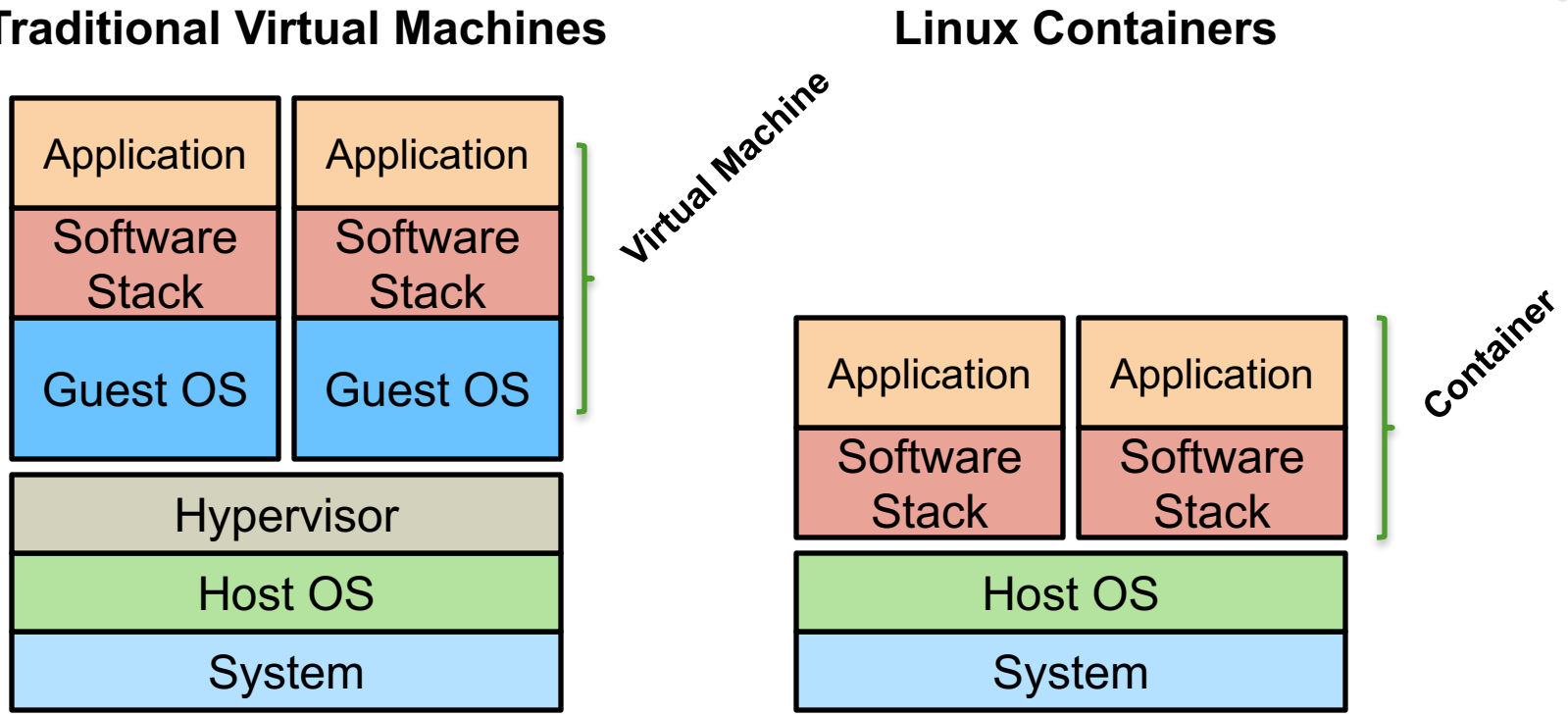




Delivery/Deployment

- **Traditional distribution**
  - iso/udf, rpm, tar
- **Management S/W**
  - Cray HSS/RSM, OpenStack
- **Application sources**
  - ISV's, DockerHub, online registries
  - Application marketplace
- **Cloud providers**
  - Cloud instances
  - Application tiers

Containers



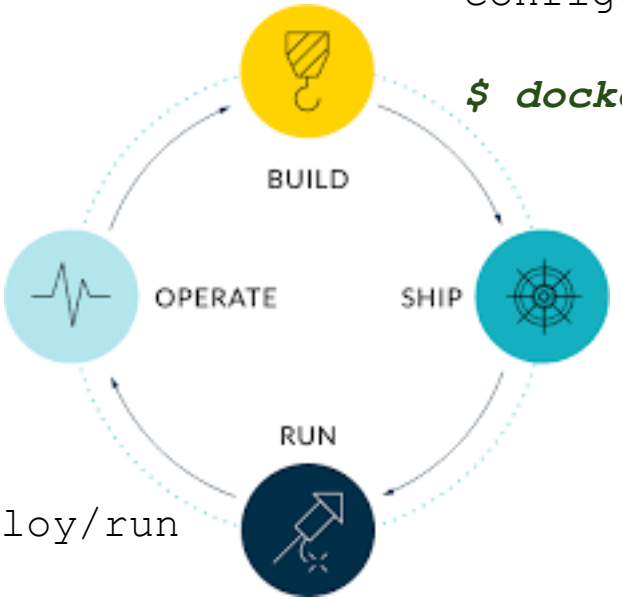
Docker Basics

Build image, application, dependencies, environment, configurations.

$ docker build …

Publish image to registry aka image repository.

$ docker push …

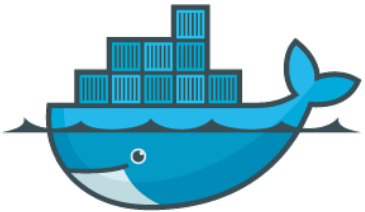


Monitor, upgrade services.

$ docker pull …

Pull image, and deploy/run on environment.

$ docker run …



Open Source Container Runtimes

- **Why not run Docker/CoreOS ?**
  - Requires local disk for image/container/metadata
  - Open security model – can be problematic in multitenant environments
  - Batch system integration challenges
  - Docker/CoreOS encompasses storage, networking, and resource management elements
  - Kernel requirements (namespace, cgroups, and virtualization components)
    - More than in standard CLE compute node OS

Other Container Runtimes

- **rkt**
  - Pronounced “rocket”, next-generation container manager for Linux clusters
    - Promoted by CoreOS

  5,082 commits 2 branches 56 releases 175 contributors

- **runC**
  - CLI tool for spawning and running containers according to the OCI specification
  - runC is the container runtime for Docker via the libContainer interface

  3,041 commits 4 branches 14 releases 195 contributors

- **Singularity**
  - Container solution for HPC and research environments, developed by LBNL

  2,033 commits 8 branches 6 releases 30 contributors

- **Shifter**

  1,254 commits 3 branches 2 releases 5 contributors

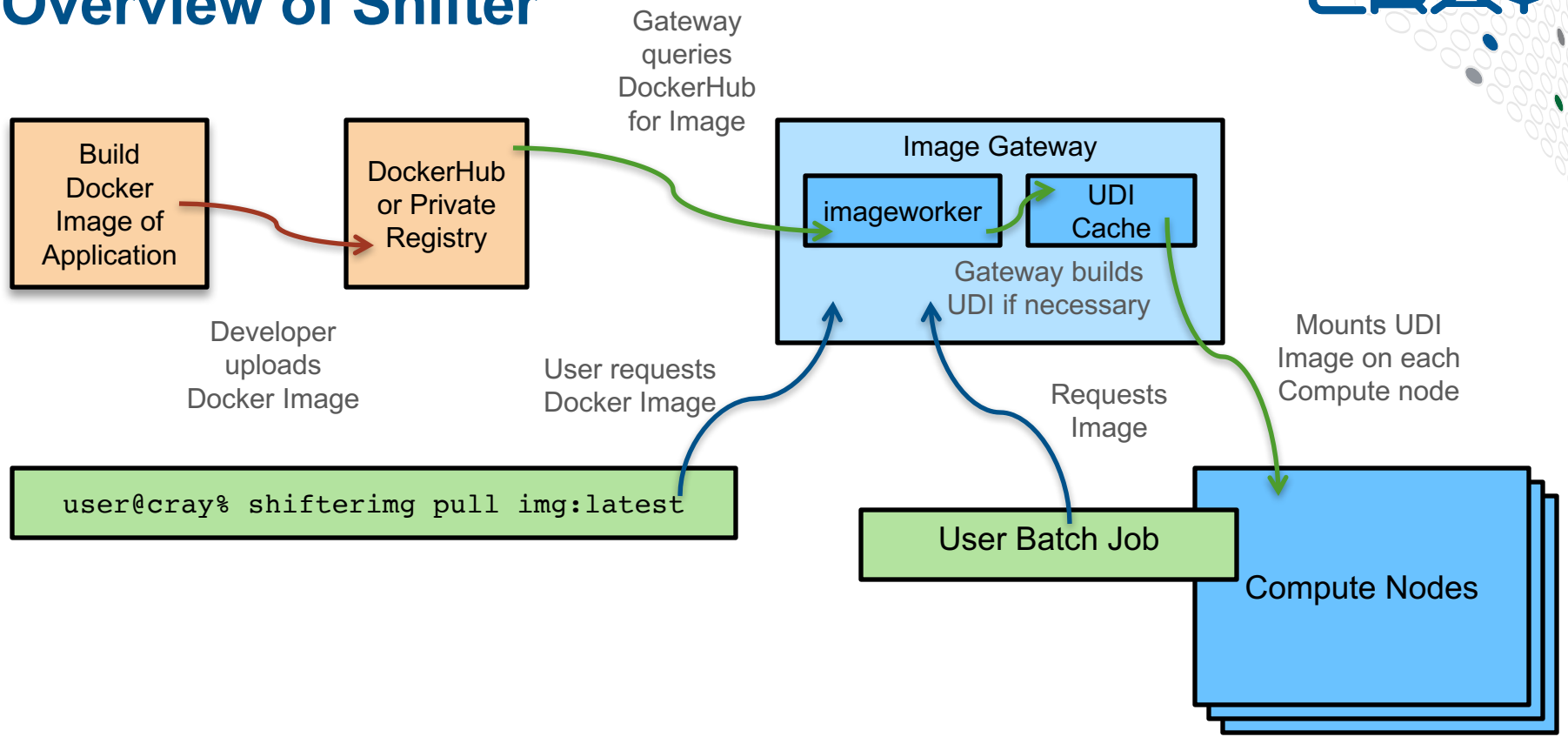
NERSC Shifter

- **Partnership with Cray to design a solution to run containers on an HPC platform**
- **Design Goals:**
  - User independence: require no administrator assistance to launch an application inside an image
  - Shared host resource availability (e.g., Lustre/DVS mounts and network interfaces)
  - Leverages or integrates with public/private image repos (i.e. DockerHub)
  - Seamless user experience
  - Robust and secure implementation

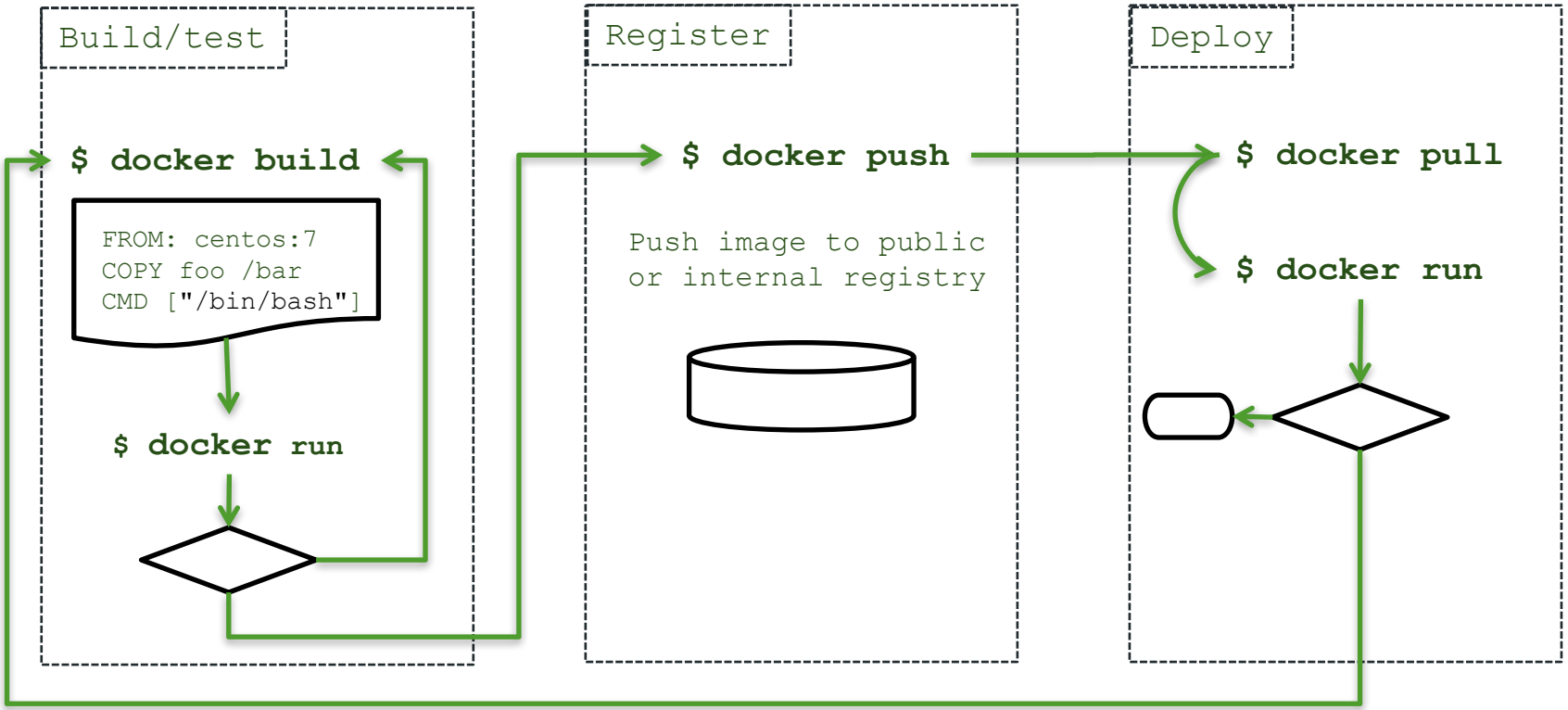
Shifter Components

- **Shifter Image Gateway**
  - Imports and converts images from DockerHub and Private Registries
- **Shifter Runtime**
  - Instantiates images securely on compute resources
- **Workload Manager Integration**
  - Integrates Shifter with WLMs (PBS, Moab/Torque, SLURM)

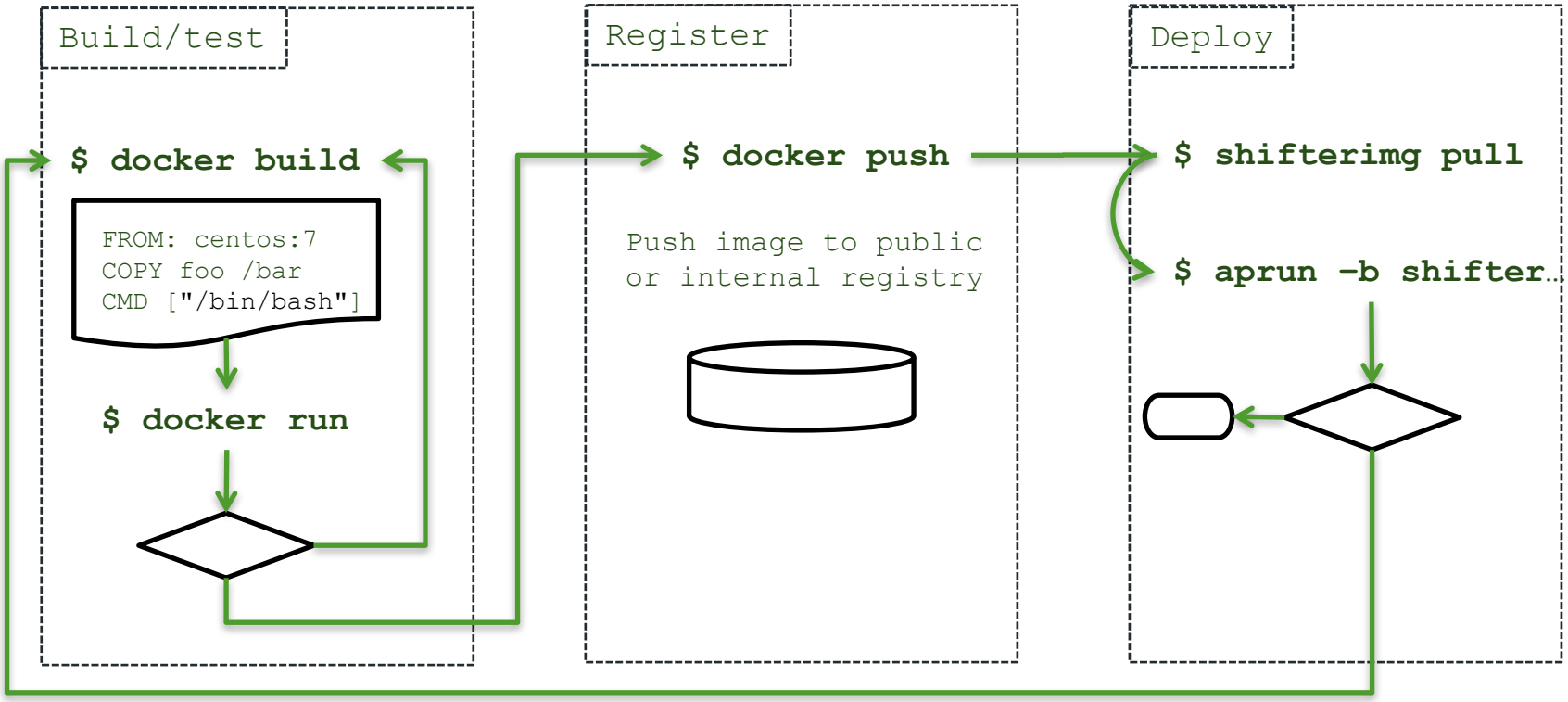
Overview of Shifter



Docker Workflow



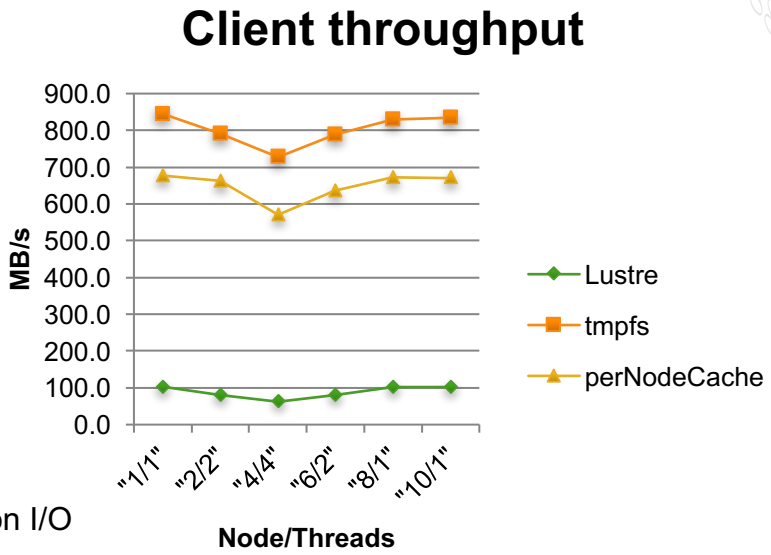
Shifter Workflow



Per-Node Write Cache[1]

Per-Node Write Cache provides local disk like functionality but is backed by the Parallel File System

Nodes/Threads	Lustre (MB/s)	Shifter (MB/s)	/tmp (tmpfs) (MB/s)
1/1	103.0	677.0	845.0
2/2	80.2	662.5	791.0
4/4	62.2	570.5	727.3
6/2	79.9	636.7	789.2
8/1	102.9	672.1	830.5
10/1	101.9	671.2	835.1



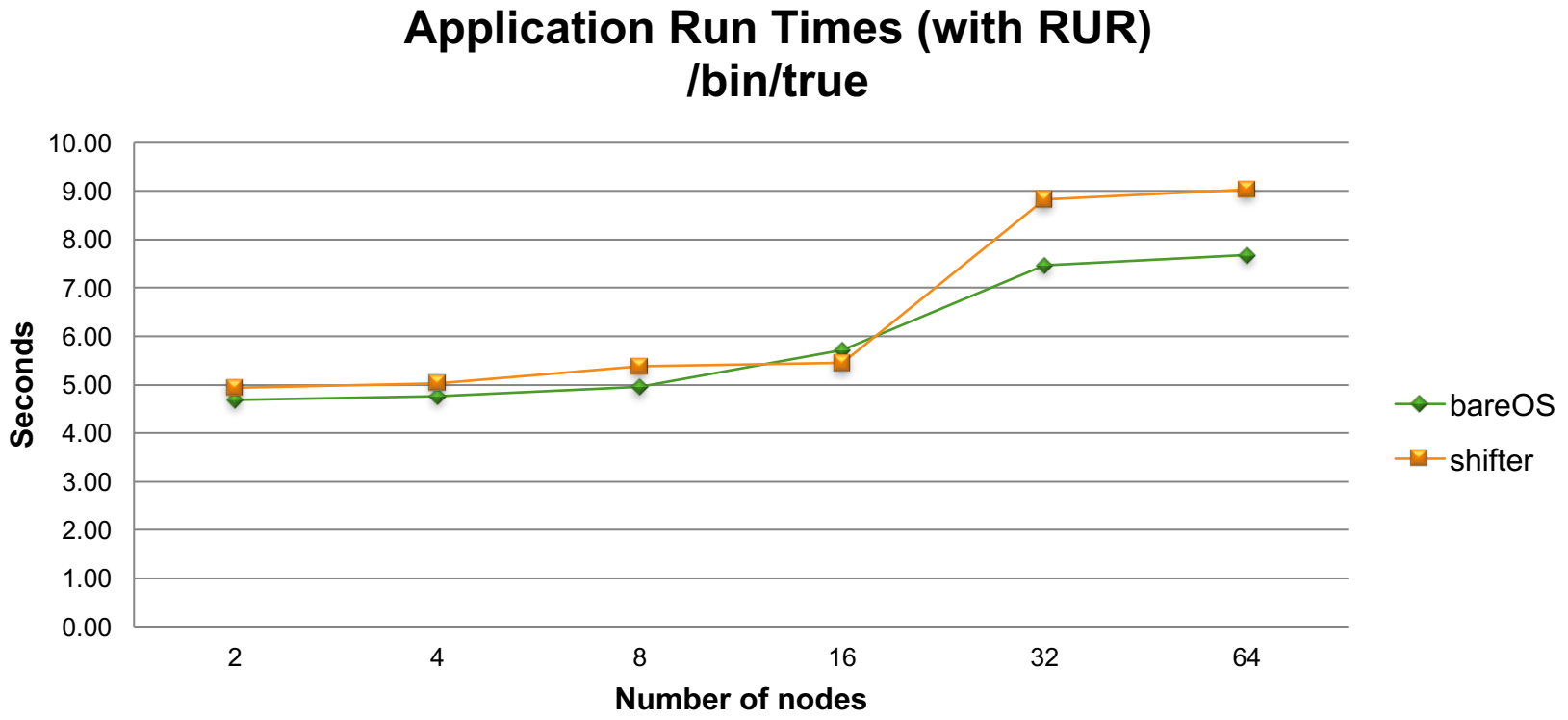
Results of a simple “dd” test to simulate writing ~5GB of small transaction I/O (`dd if=/dev/zero of=$TARGET bs=512 count=10M`)

[1]Cray Sonexion 3000 6 SSU.

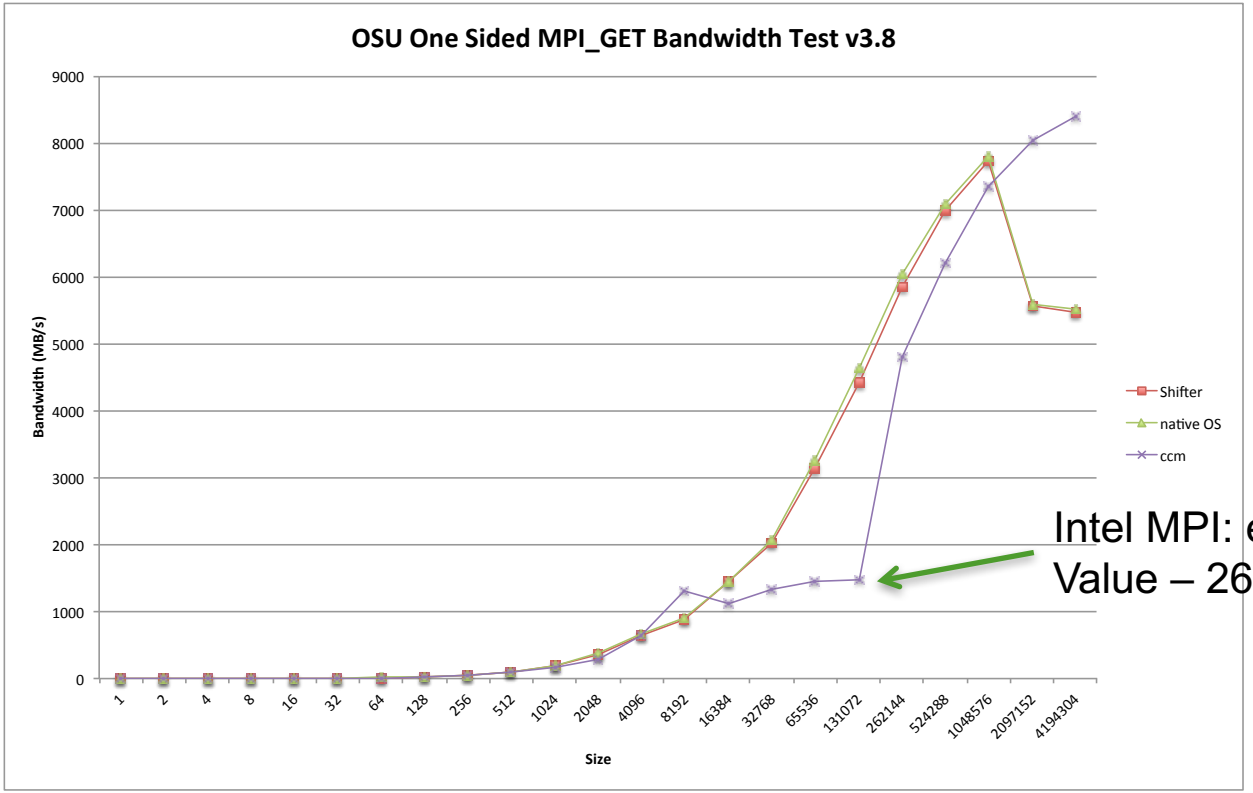
Shifter vs. Base OS Comparisons

- Launch time comparison
- OSU benchmarks
- NAS Parallel Benchmarks
- Radioss

Container Startup Times



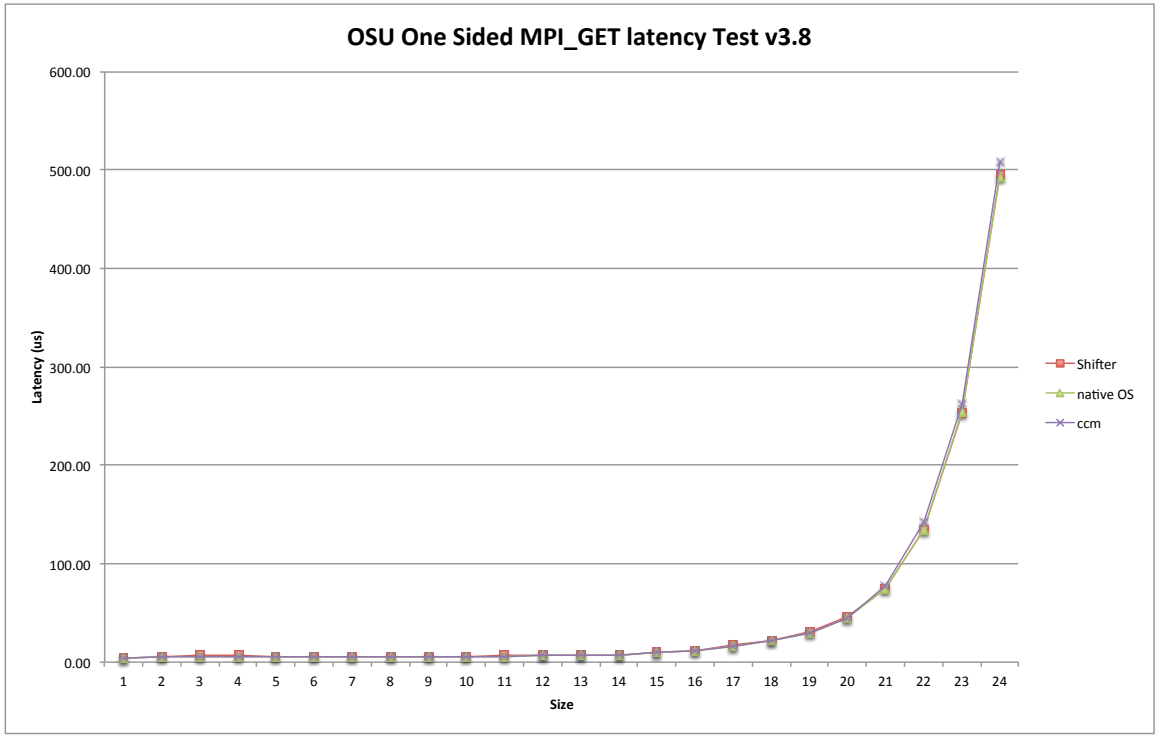
OSU One-sided Bandwidth – 2 Nodes



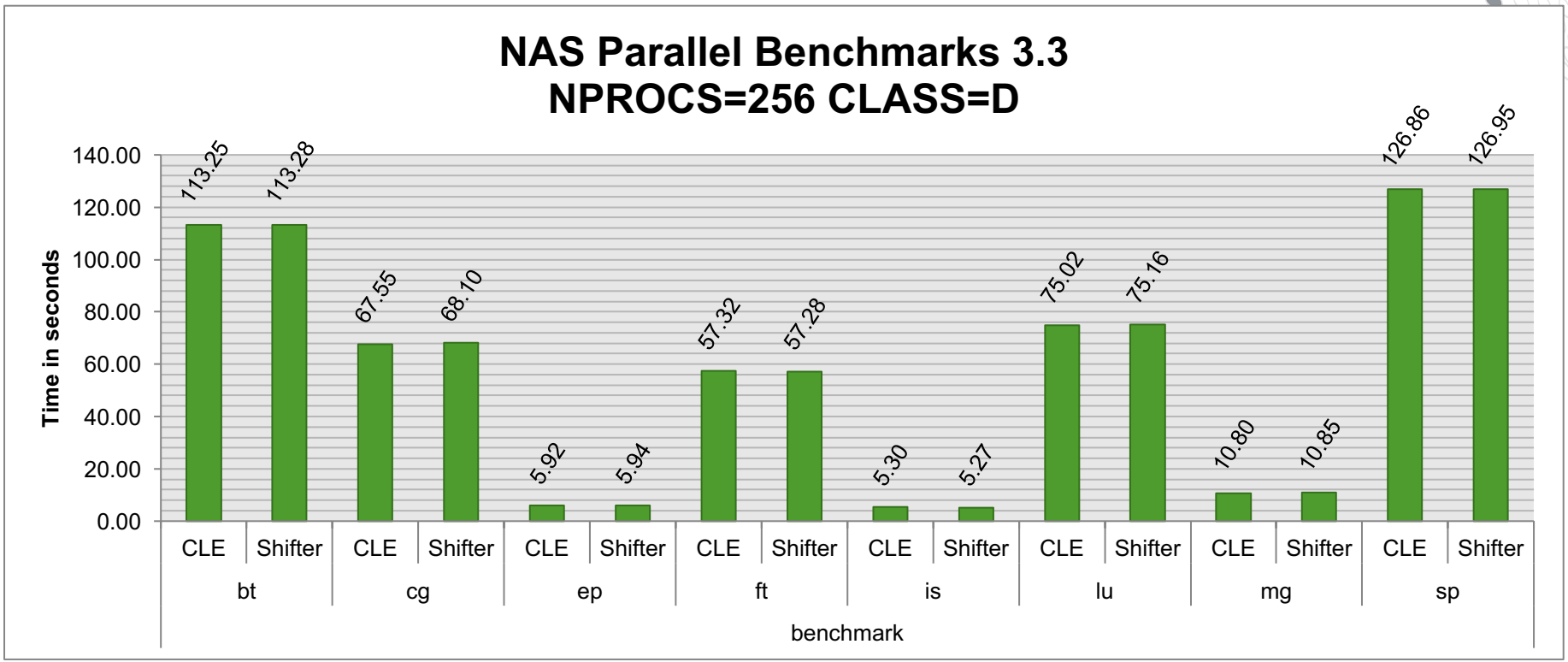
Preliminary Results

Intel MPI: eager/rendezvous Value – 262144

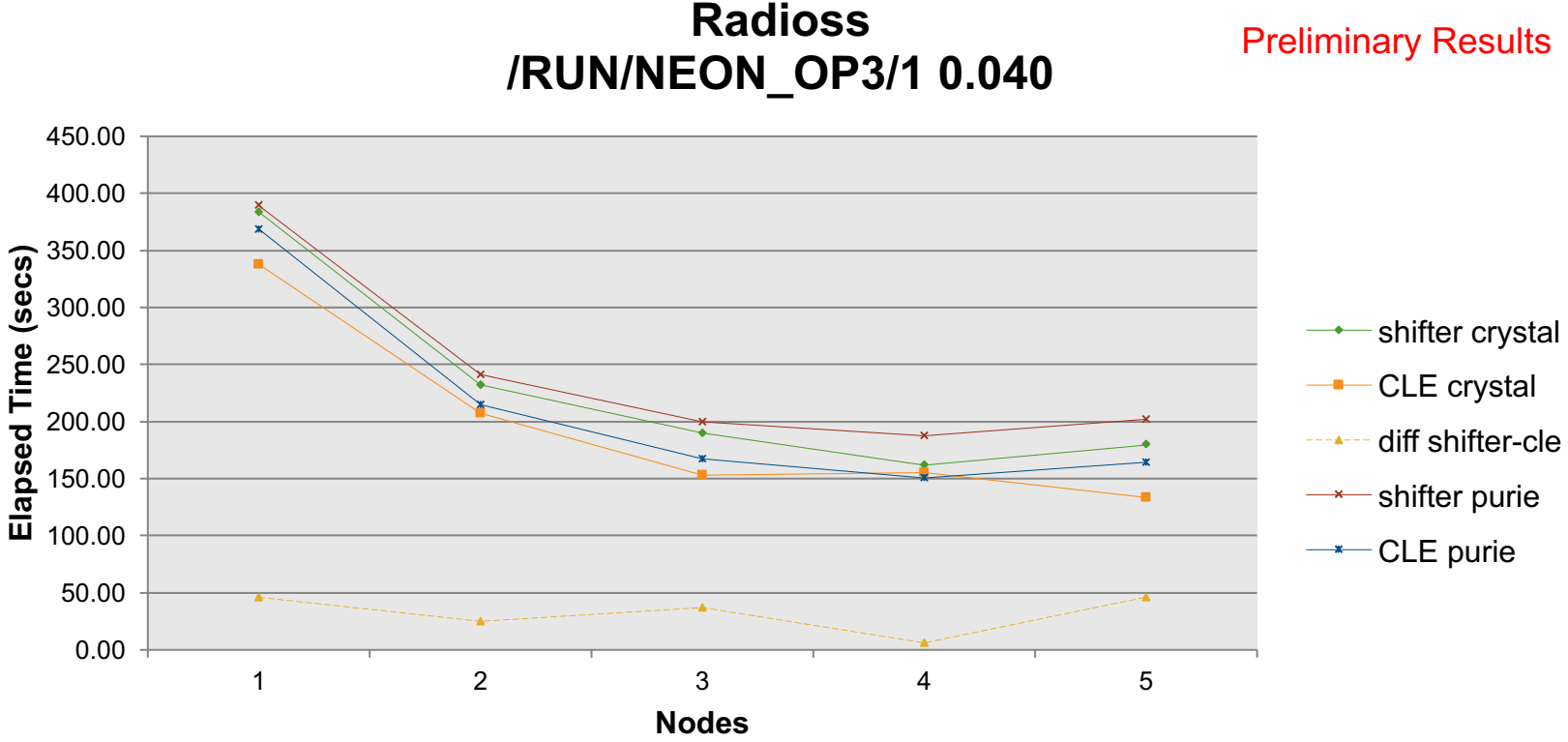
OSU One-sided Latency – 2 Nodes



NAS Parallel Benchmarks



Radioss



Conclusion

- Shifter is enabling and improving support for Data Intensive Workloads
- Integrated and supported in CLE

Shifter provides the flexibility of Docker without sacrificing security, scalability, or performance.

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, REVEAL, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.

Thank You!