HPC Containers in Use

Jonathan Sparks – Cray Inc.

CRAY

CUG 2017 CAFFEINATED COMPUTING

Redmond, Washington May 7-11, 2017

Agenda

- Goals
- Container Environments
- Performance Characteristics
- Conclusion and Future Work

Goals

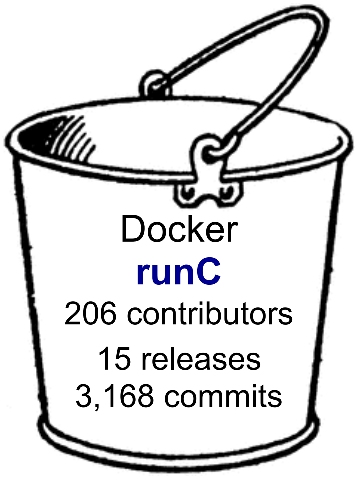
- Given the adoption rate of Containers in computing, investigate different container environments for use in HPC.
- Configuration management of container runtimes
- System integration
- Container performance comparison

Container Runtime Environments

- Selected two Enterprise, and two HPC container environments

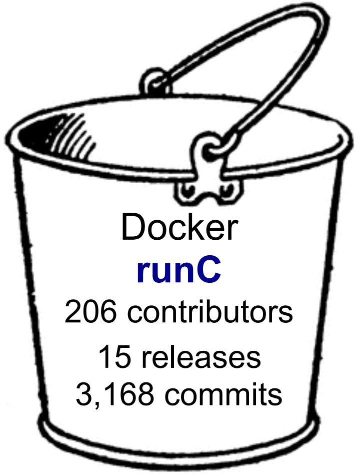
Container Runtime Environments

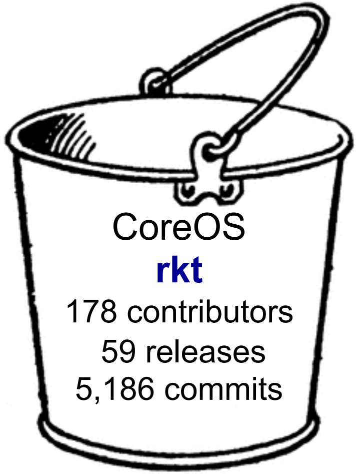
- Selected two Enterprise, and two HPC container environments



Container Runtime Environments

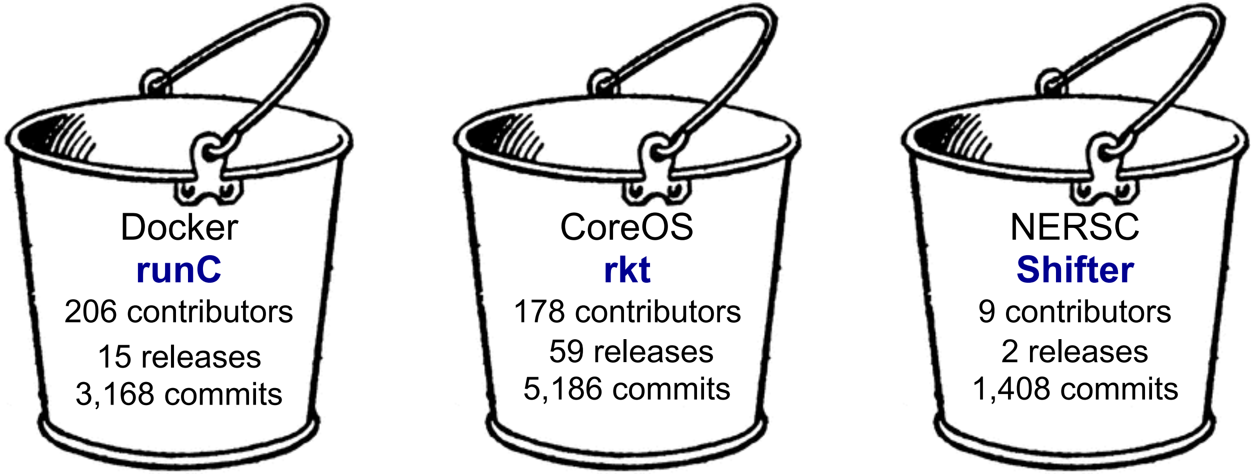
- **Selected two Enterprise, and two HPC container environments**





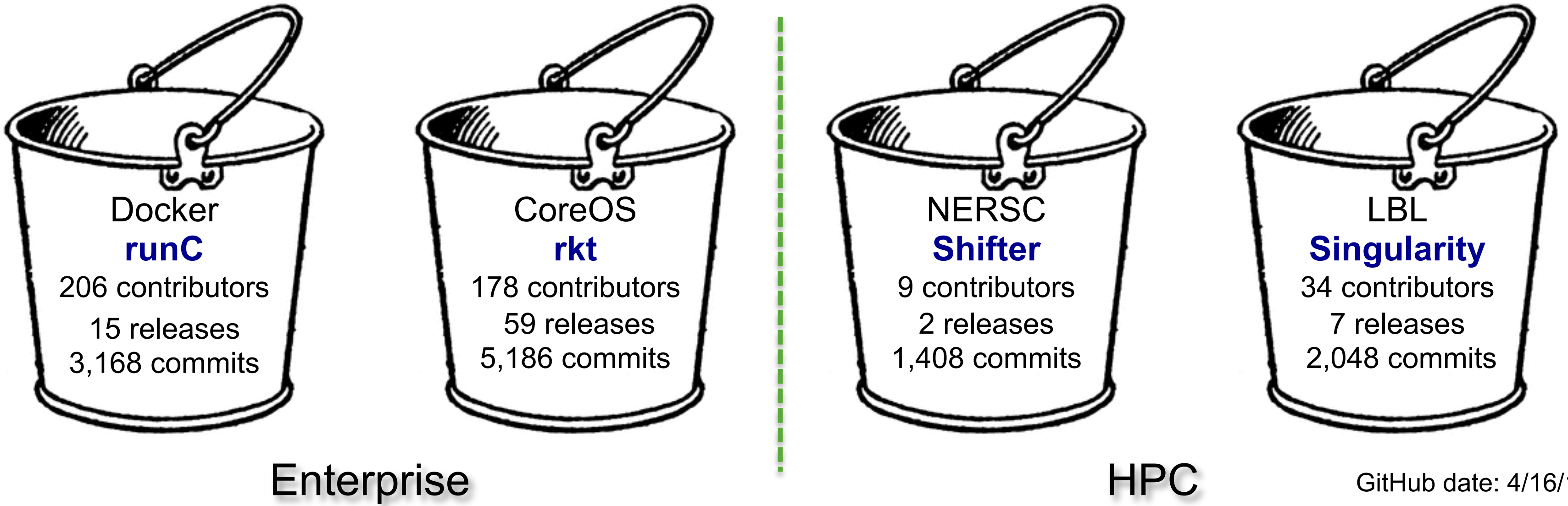
Container Runtime Environments

- **Selected two Enterprise, and two HPC container environments**



Container Runtime Environments

- Selected two Enterprise, and two HPC container environments



GitHub date: 4/16/17

Container Runtime Environment

- System integration
- Container runtime configuration management
- Deployment

System Integration

① $ aprun -n *N* … -b **shifter** --image cle:latest a.out

System Integration

① $ aprun -n *N* … -b **shifter** --image cle:latest a.out

② $ aprun -n *N* … -b **singularity** exec /global/cle.latest a.out

System Integration

① $ aprun -n *N* … -b **shifter** --image cle:latest a.out

② $ aprun -n *N* … -b **singularity** exec /global/cle.latest a.out

③ $ aprun -n *N* … -b **rkt** run \
--stage1-name=coreos.com/rkt/stage1-fly:1.21.0 \
--volume alps, kind=host, source=/var/opt/cray/alps/spool, readOnly=false \
--mount volume=alps, target=/var/opt/cray/alps/spool \
registry-1.docker.io/library/cle:latest --exec=/usr/bin/a.out

System Integration

① $ aprun -n *N* … -b **shifter** --image cle:latest a.out

② $ aprun -n *N* … -b **singularity** exec /global/cle.latest a.out

③ $ aprun -n *N* … -b **rkt** run \
--stage1-name=coreos.com/rkt/stage1-fly:1.21.0 \
--volume alps, kind=host, source=/var/opt/cray/alps/spool, readOnly=false \
--mount volume=alps, target=/var/opt/cray/alps/spool \
registry-1.docker.io/library/cle:latest --exec=/usr/bin/a.out

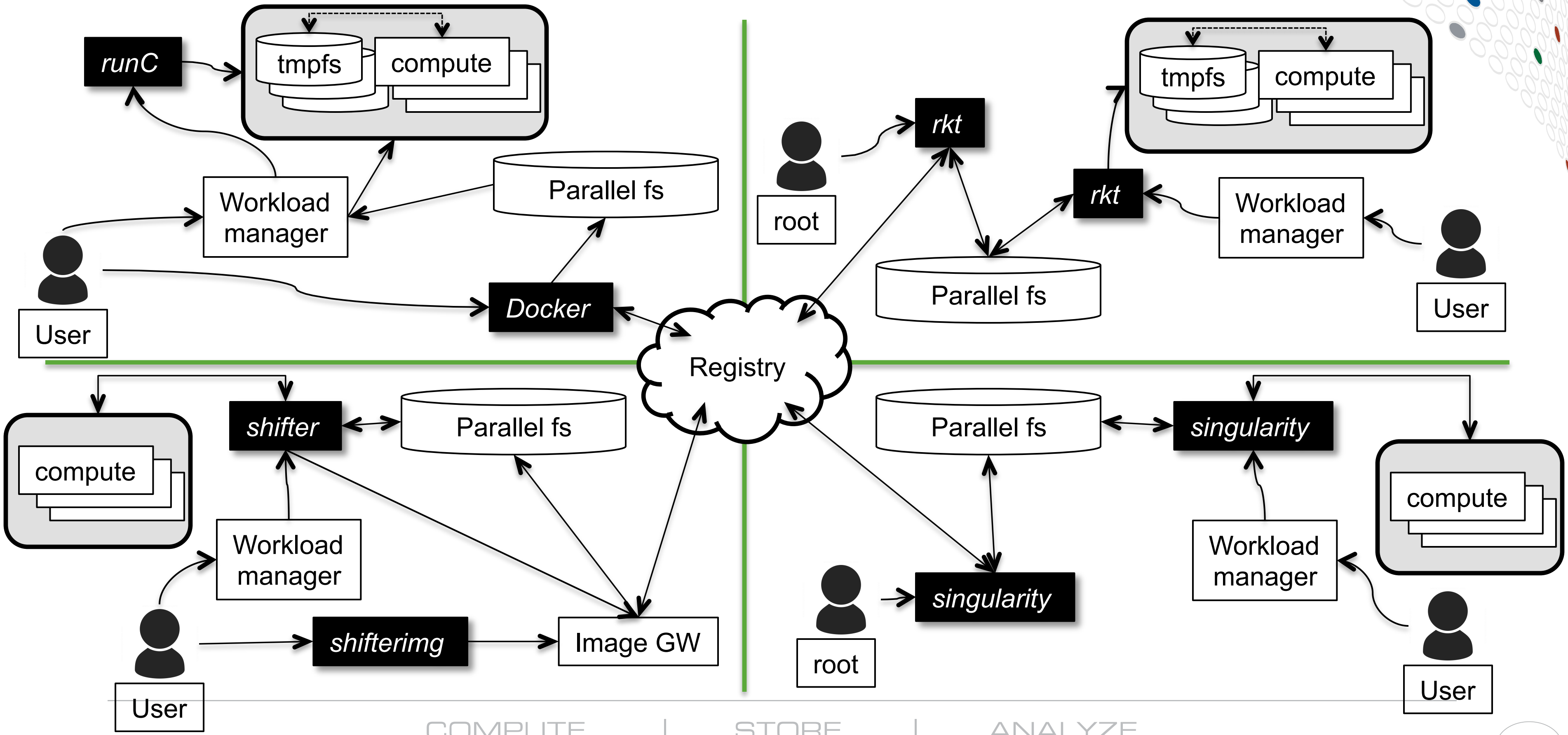
④ $ aprun -n *N* -b **runc** --bundle /tmp/cle.latest run $(date +%Y%m%d%H%M)

Container Runtime Configuration

- **rkt**
  - `/usr/lib/rkt`, `/etc/rkt`, and user-defined
    - Repository authentication policies, data and image locations
    - Command line can override system configurations
- **Shifter**
  - System configuration (`/etc/opt/cray/shifter`)
    - Authentication policies, data and image locations
- **Singularity**
  - System configuration `$SYSCONFDIR/singularity/singularity.conf`
    - Authentication policies, data and image locations
- **runC**
  - **Embedded** in the image definition (aka bundle): `config.json`

Configurability

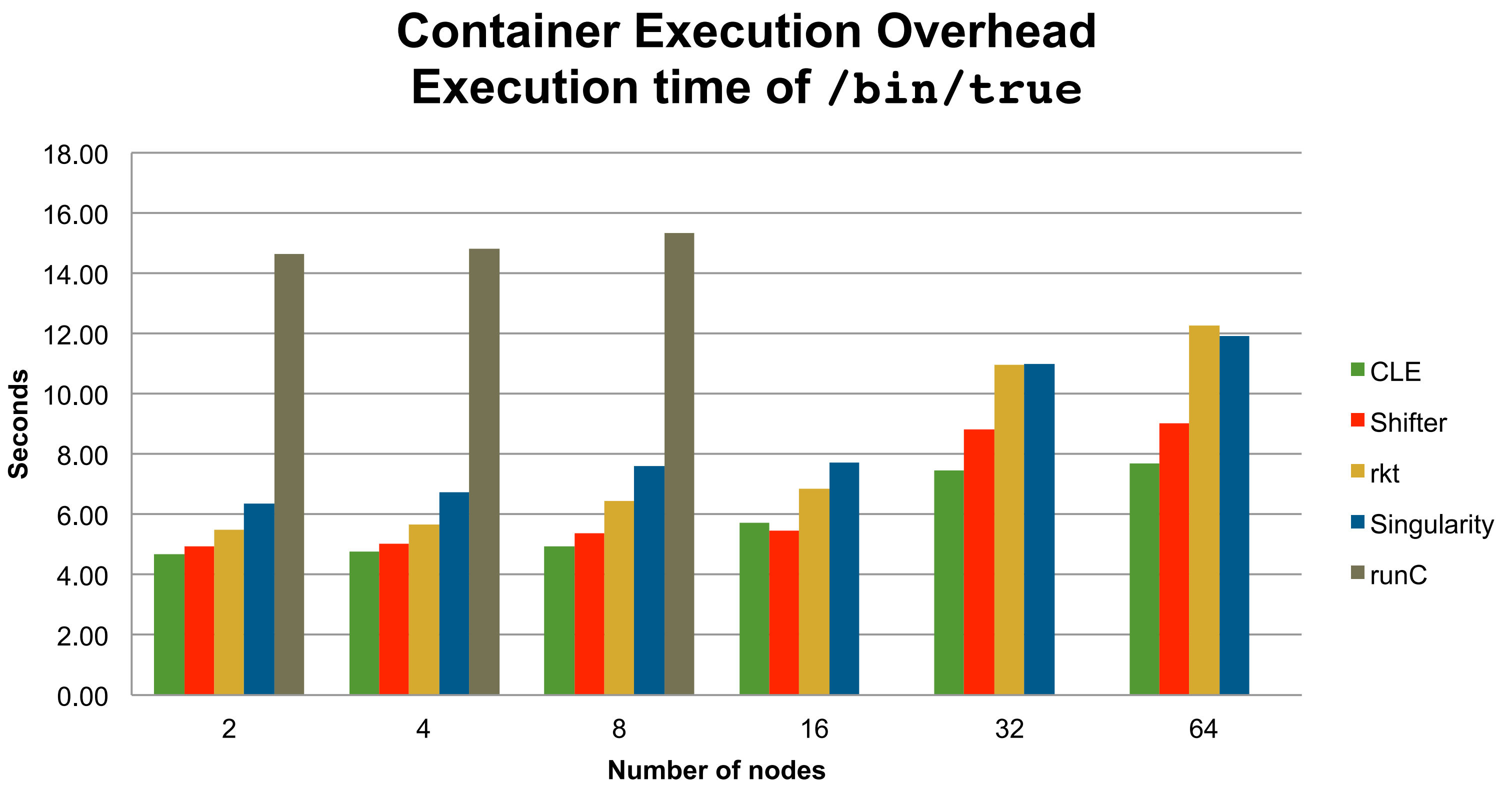
Deployments



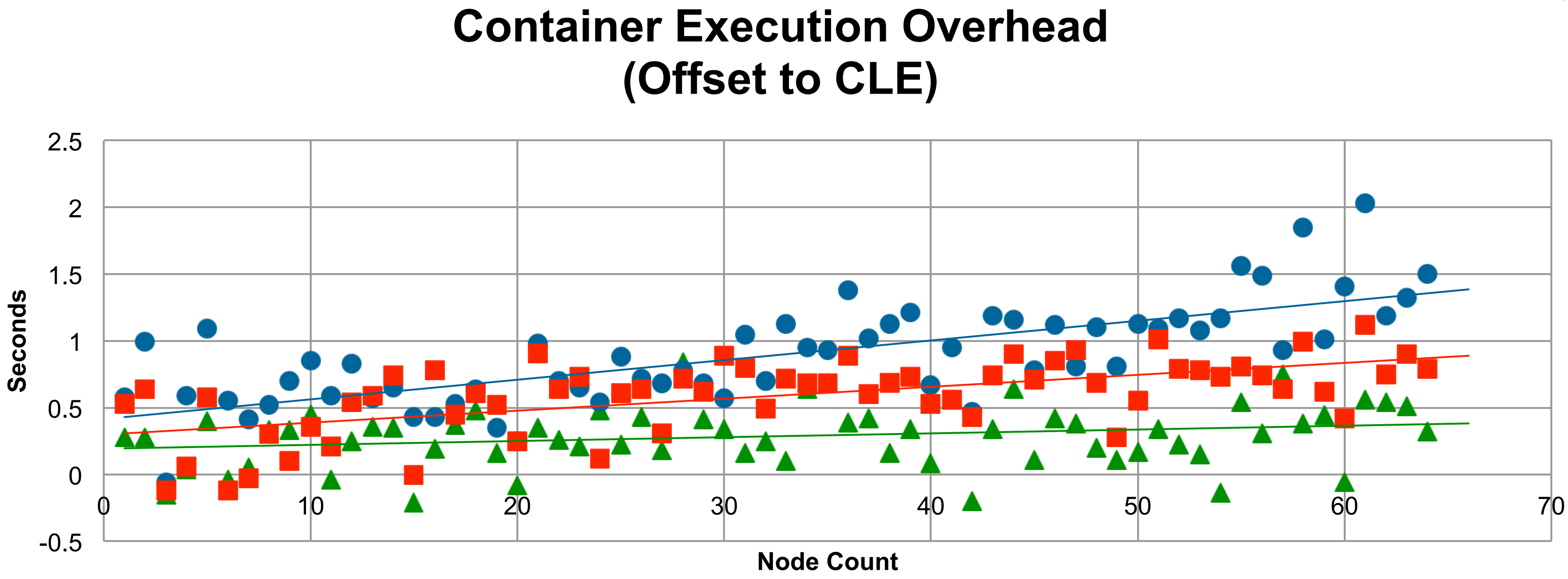
Performance Investigation

- **Launch times**
  - Time to setup and launch via container runtime
- **Application performance**
  - Hugepage optimization
  - Environment pass-through

Launch times

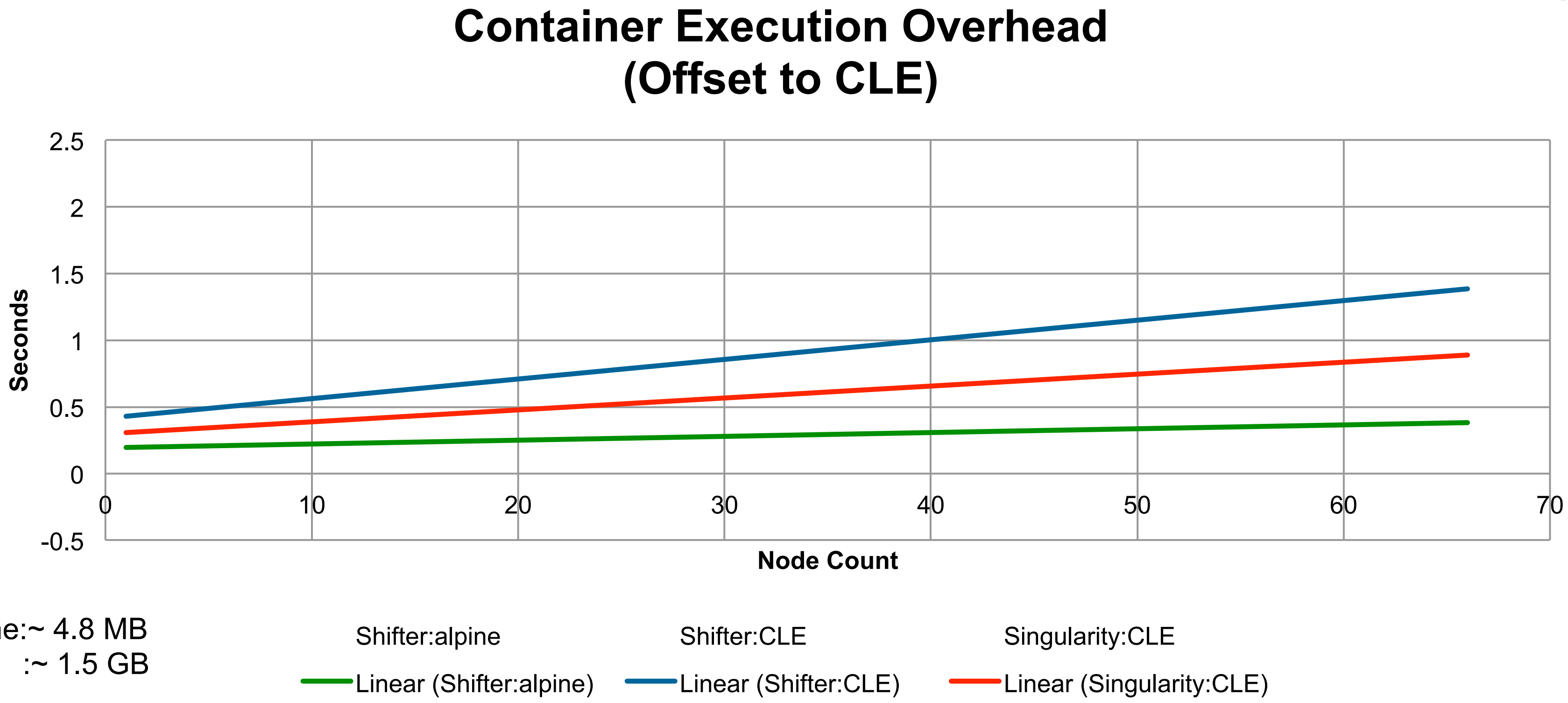


Launch times & Image size



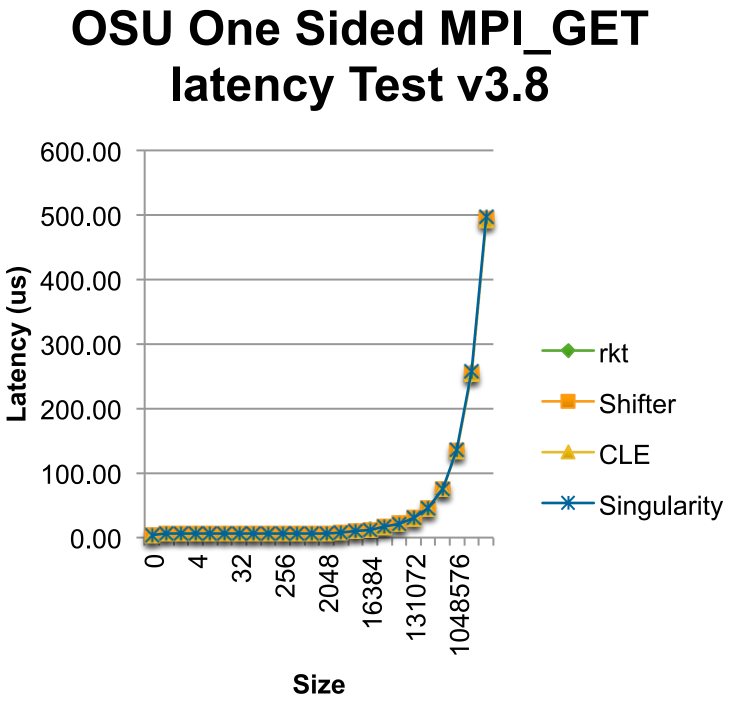
snx11010 is a 1600, running the 1.4 neo release, 2 SSUs

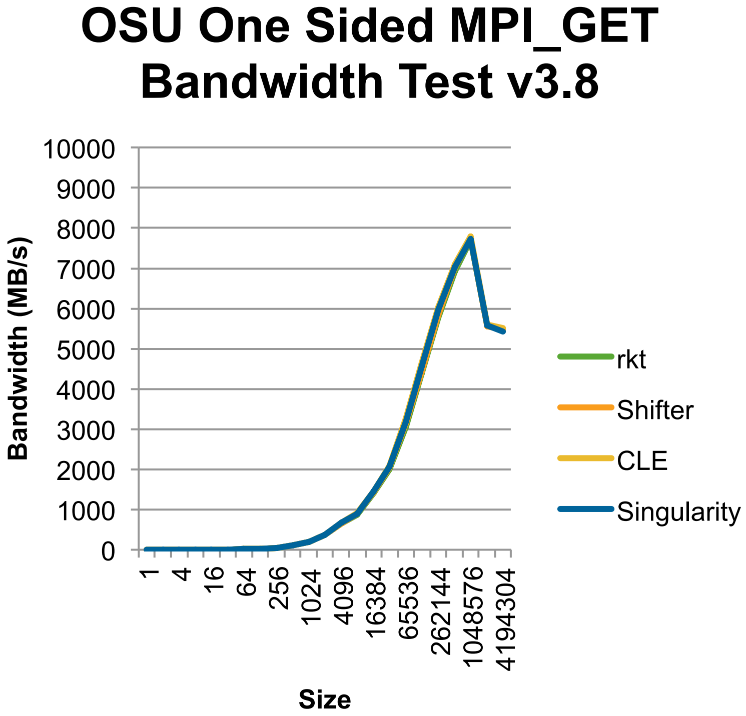
Launch times & Image size



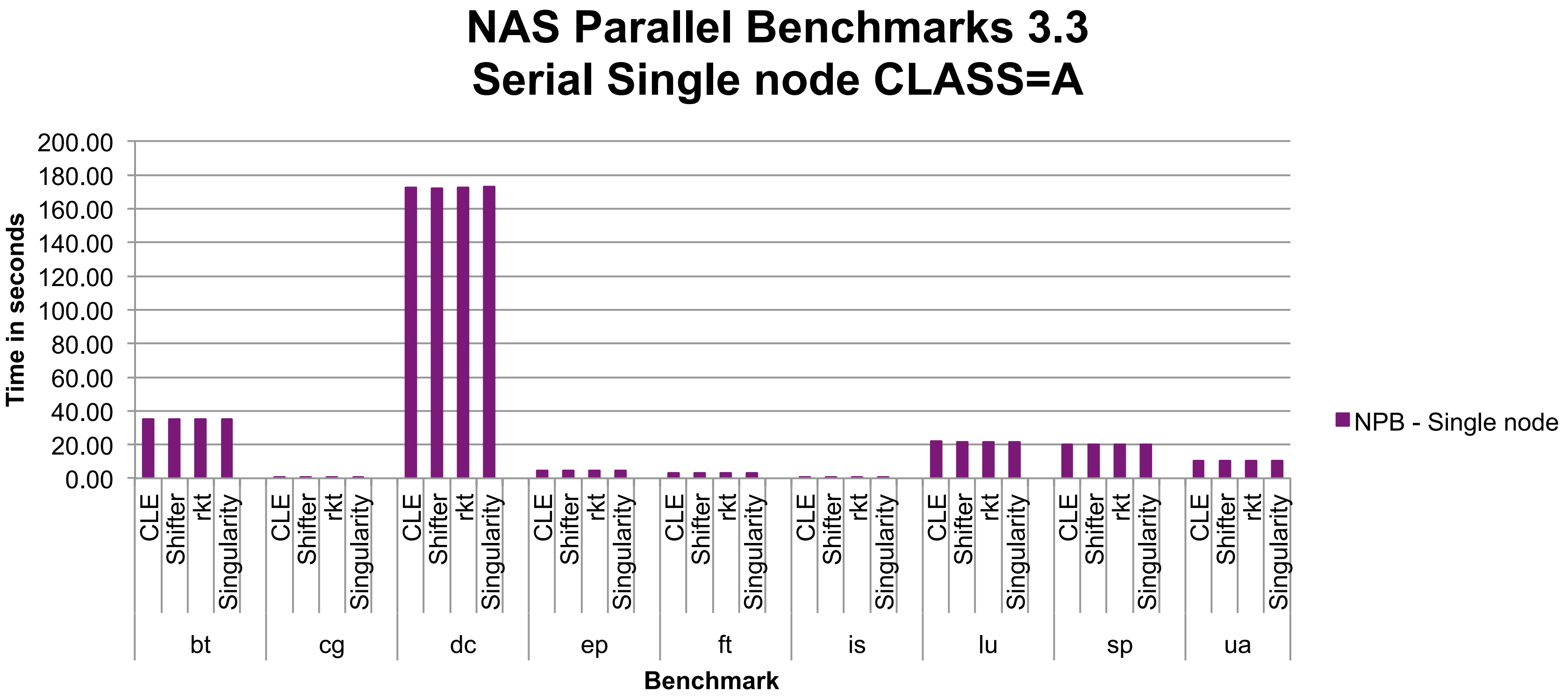
alpine:~ 4.8 MB
CLE :~ 1.5 GB

OSU Micro-Benchmarks

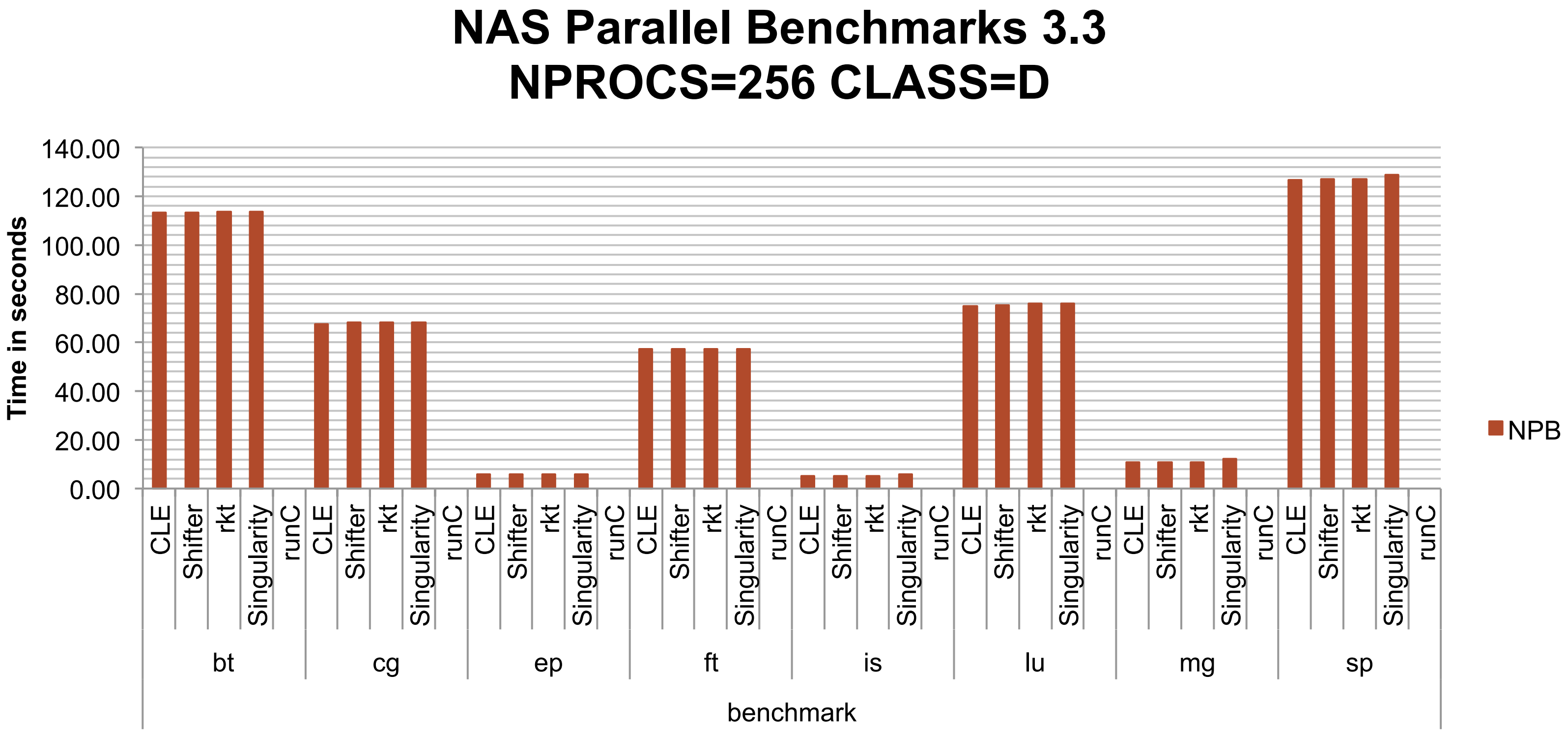




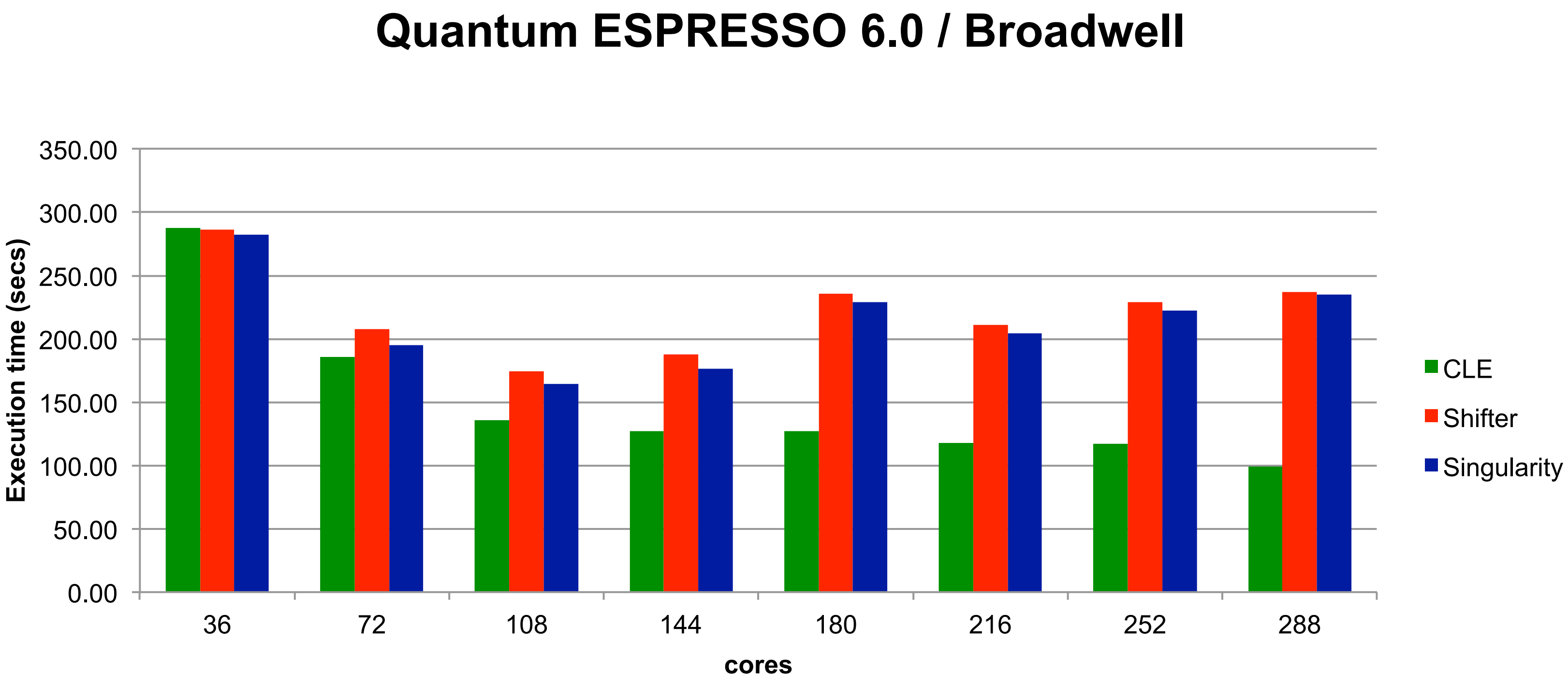
NPB Single node



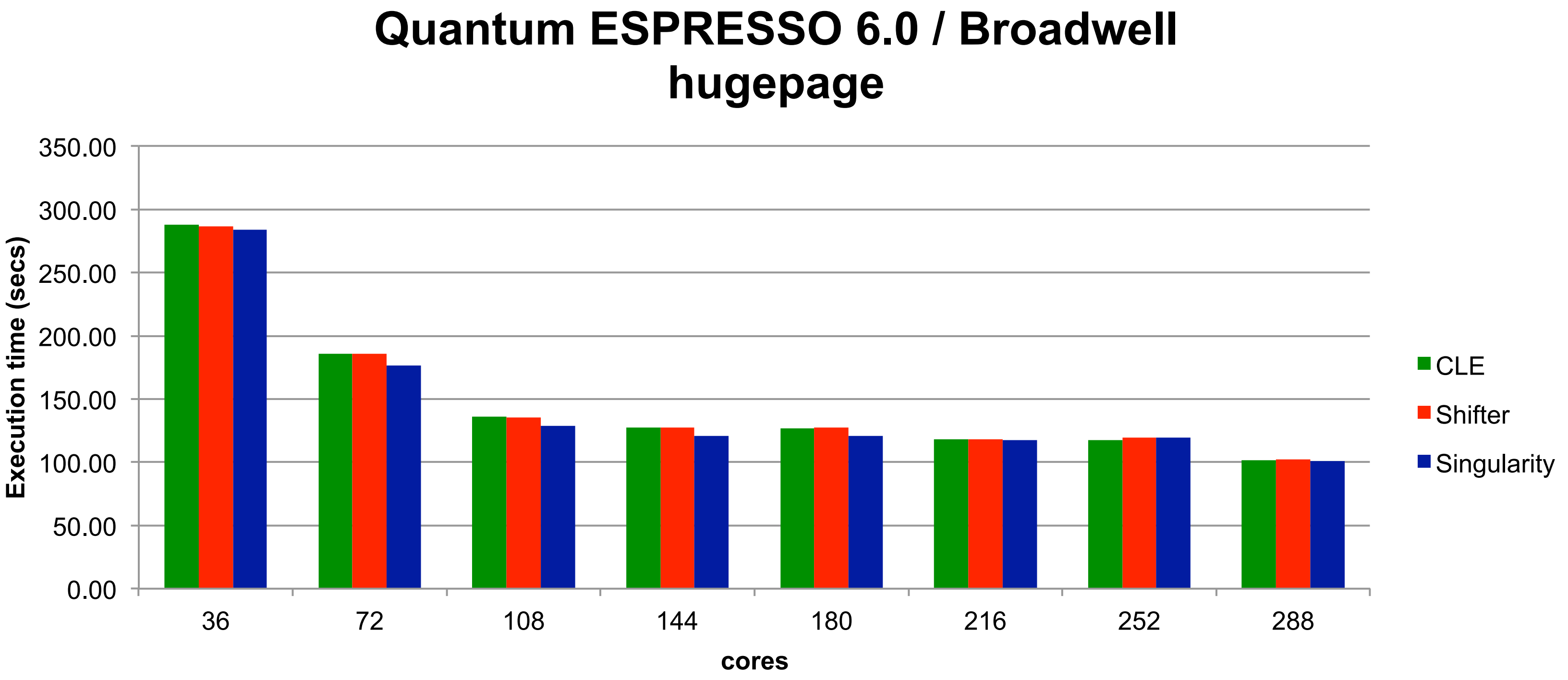
NPB Multi-node



Quantum ESPRESSO

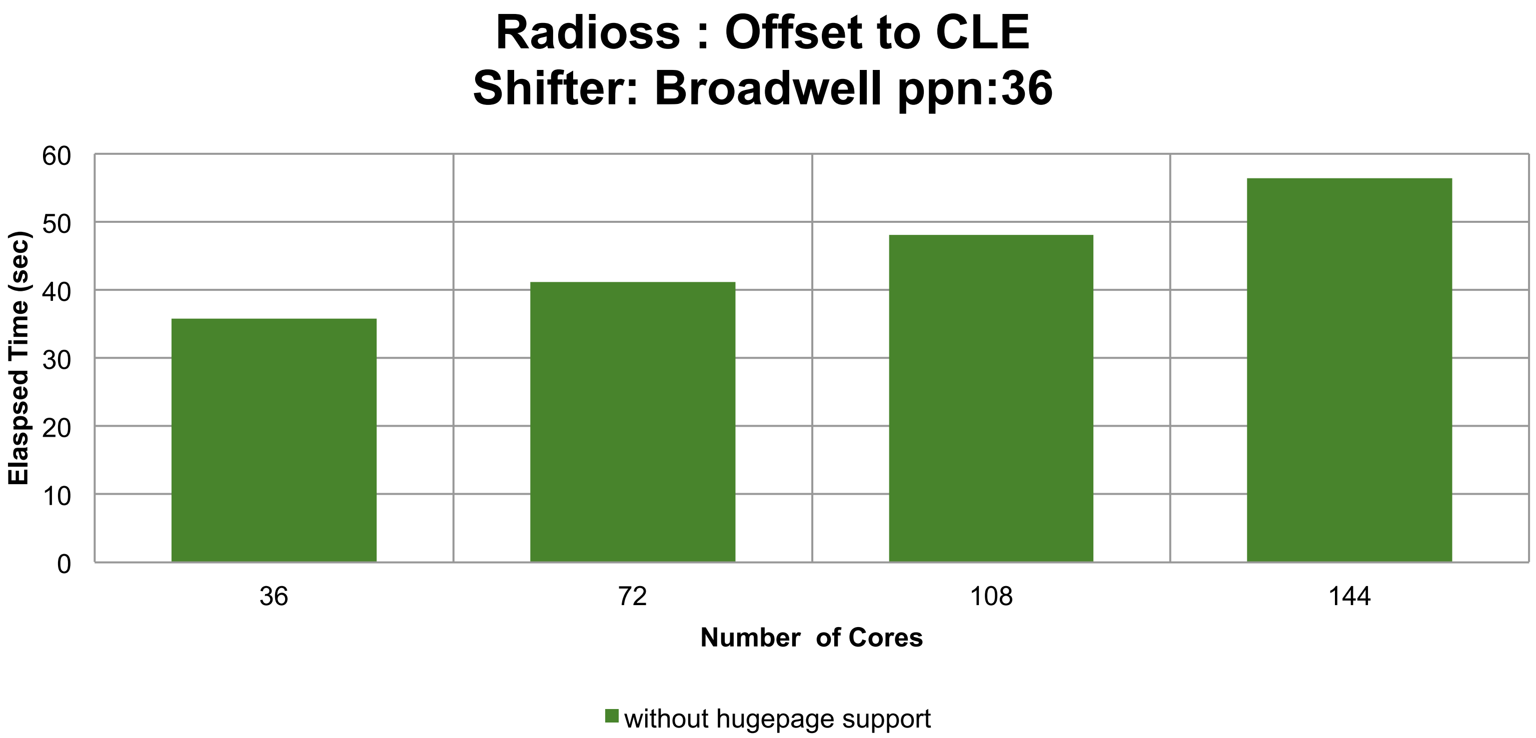


Quantum ESPRESSO

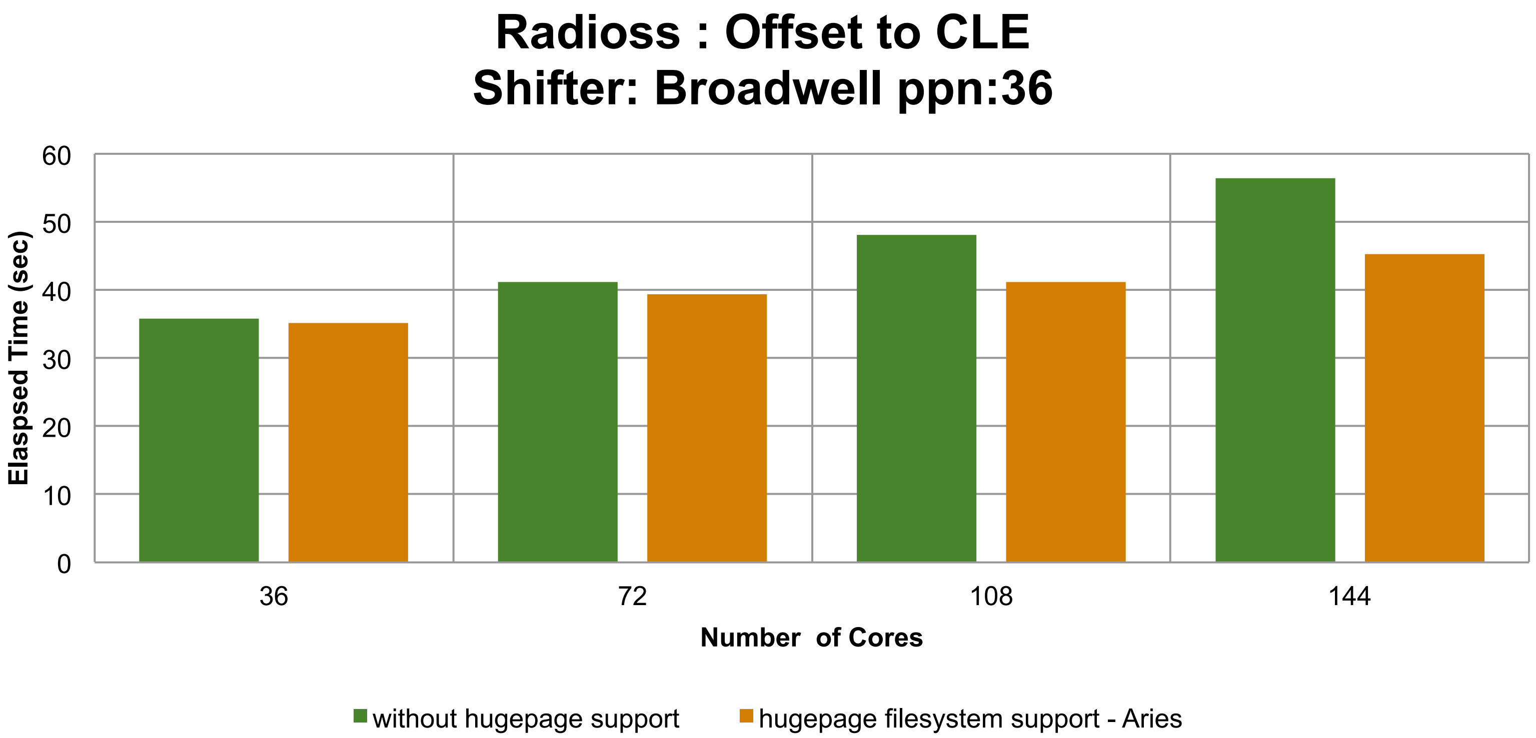


COMPUTE | STORE | ANALYZE

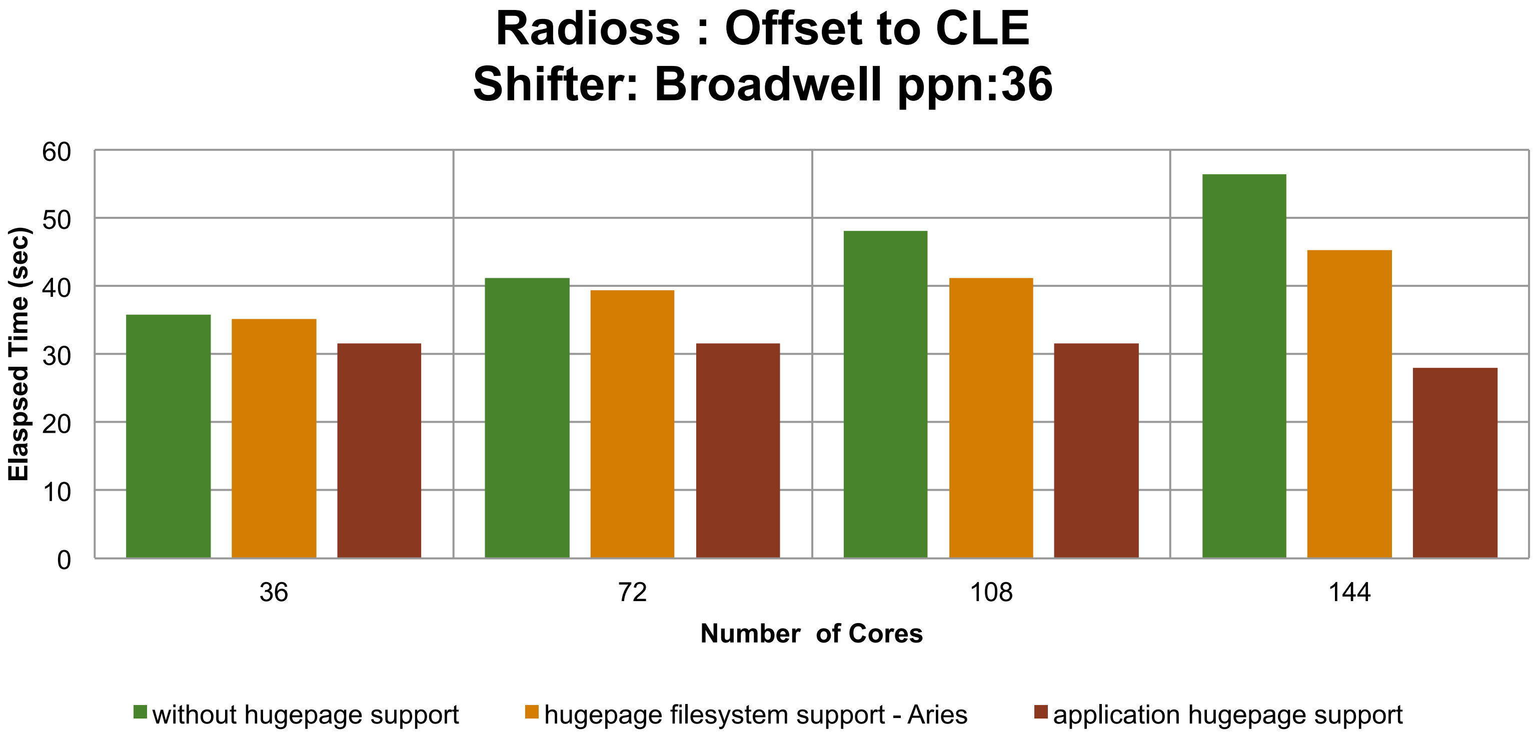
Radioss – Performance



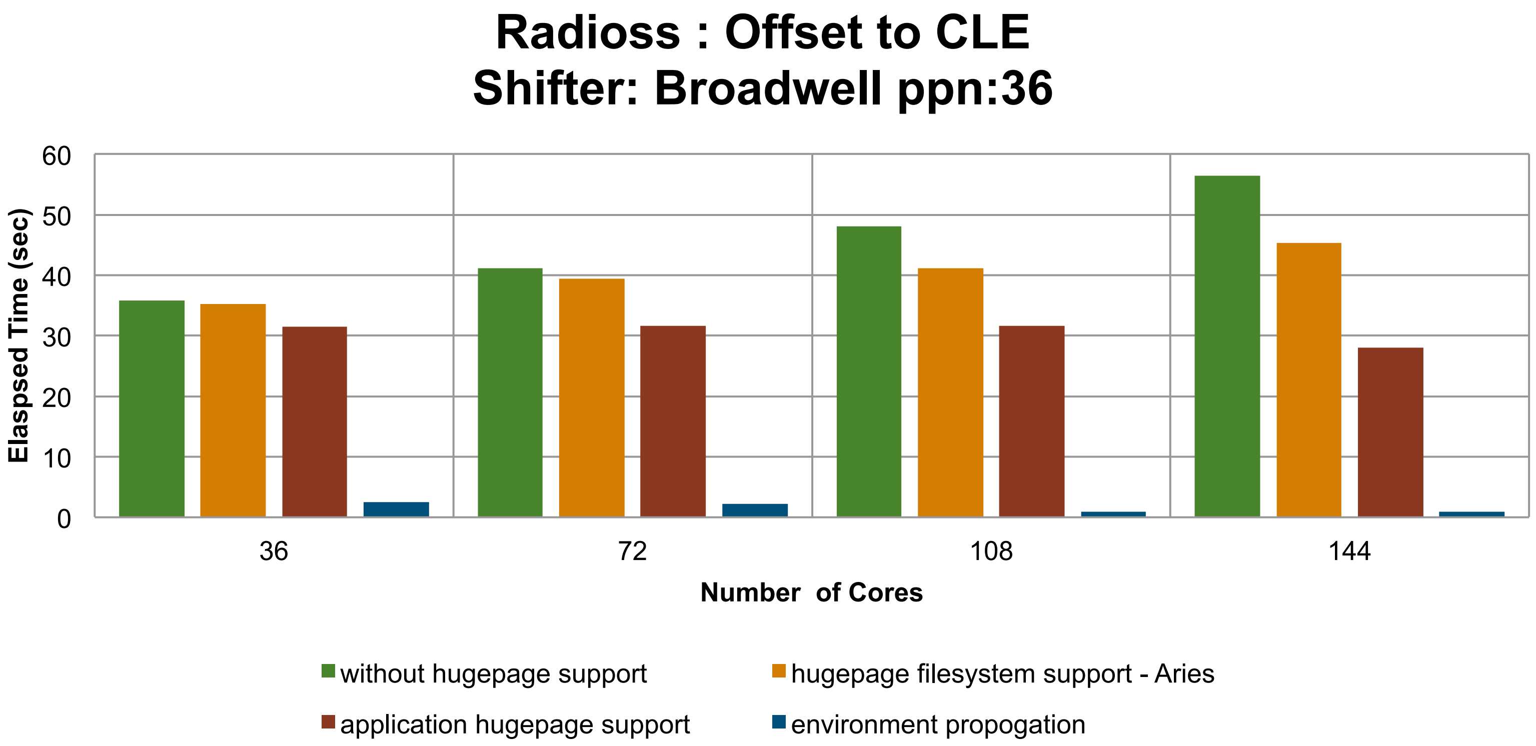
Radioss – Performance



Radioss – Performance



Radioss – Performance



Conclusions

- **Container runtimes**
  - Enterprise frameworks can be used for HPC applications

Conclusions

- **Container runtimes**
  - Enterprise frameworks “can” be used for HPC applications
- **Performance**
  - Native application performance can be achieved, requires host-level access to resources (network, file system)
  - Environment pass-through. Cray PE dependent on environment variables
  - Launch time dependent on container infrastructure and image size

Future Work

Future Work

- **Scaling investigation of open container frameworks**
  - Shared image across nodes (ro)
  - Container file system (rw)

Future Work

- **Scaling investigation of open container frameworks**
  - Shared image across nodes (ro)
  - Container file system (rw)
- **Tools**
  - Framework to support multiple container runtimes.

Future Work

- **Scaling investigation of open container frameworks**
  - Shared image across nodes (ro)
  - Container file system (rw)
- **Tools**
  - Framework to support multiple container runtimes.
  - Analysis tools
    - Inspection (static/runtime/content)
    - Performance characterization

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.

Q&A

Jonathan Sparks
jsparks@cray.com