HPC Containers in Use
Jonathan Sparks – Cray Inc.
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

Docker
  - runC
  - 206 contributors
  - 15 releases
  - 3,168 commits
Container Runtime Environments

- Selected two Enterprise, and two HPC container environments

- Docker
  - runC
  - 206 contributors
  - 15 releases
  - 3,168 commits

- CoreOS
  - rkt
  - 178 contributors
  - 59 releases
  - 5,186 commits
Container Runtime Environments

- Selected two Enterprise, and two HPC container environments

- Docker (runC)
  - 206 contributors
  - 15 releases
  - 3,168 commits

- CoreOS (rkt)
  - 178 contributors
  - 59 releases
  - 5,186 commits

- NERSC (Shifter)
  - 9 contributors
  - 2 releases
  - 1,408 commits
Container Runtime Environments

- Selected two Enterprise, and two HPC container environments

Enterprise:
- Docker
  - runC
  - 206 contributors
  - 15 releases
  - 3,168 commits
- CoreOS
  - rkt
  - 178 contributors
  - 59 releases
  - 5,186 commits

HPC:
- NERSC
  - Shifter
  - 9 contributors
  - 2 releases
  - 1,408 commits
- LBL
  - Singularity
  - 34 contributors
  - 7 releases
  - 2,048 commits

GitHub date: 4/16/17
Container Runtime Environment

- System integration
- Container runtime configuration management
- Deployment
$ aprun -n \textit{N} \ldots -b \textit{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 \n   --volume alps, kind=host, source=/var/opt/cray/alps/spool, readOnly=false \n   --mount volume=alps, target=/var/opt/cray/alps/spool \n   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`
Deployments

- **runc**: runC
- **Image GW**: Image GW
- **Parallel fs**: Parallel fs
- **Workload manager**: Workload manager
- **Docker**: Docker
- **singularity**: singularity
- **rkt**: rkt
- **tmpfs**: tmpfs
- **compute**: compute
- **user**: User
- **Workload manager**: Workload manager
- **User**: User
Performance Investigation

- **Launch times**
  - Time to setup and launch via container runtime

- **Application performance**
  - Hugepage optimization
  - Environment pass-through
Launch times

Container Execution Overhead
Execution time of /bin/true

Seconds

Number of nodes

2
4
8
16
32
64

CLE
Shifter
rkt
 Singularity
runC
Launch times & Image size

Container Execution Overhead
(Offset to CLE)

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

- Shifter:alpine
- Shifter:CLE
- Singularity:CLE

Node Count

Seconds

0 1 2 3 4 5 6 7

0 0.5 1 1.5 2 2.5

---

Copyright 2017 Cray Inc.
Launch times & Image size

Container Execution Overhead
(Offset to CLE)

- alpine: ~4.8 MB
- CLE: ~1.5 GB

Node Count

Seconds

Shifter:alpine
Shifter:CLE
Singularity:CLE

Linear (Shifter:alpine)
Linear (Shifter:CLE)
Linear (Singularity:CLE)
OSU Micro-Benchmarks

OSU One Sided MPI_GET latency Test v3.8

Latency (us)

Size

OSU One Sided MPI_GET Bandwidth Test v3.8

Bandwidth (MB/s)

Size

Copyright 2017 Cray Inc.
NPB Single node

NAS Parallel Benchmarks 3.3
Serial Single node CLASS=A

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>bt</td>
<td>CLE</td>
</tr>
<tr>
<td>cg</td>
<td>CLE</td>
</tr>
<tr>
<td>dc</td>
<td>CLE</td>
</tr>
<tr>
<td>ep</td>
<td>CLE</td>
</tr>
<tr>
<td>ft</td>
<td>CLE</td>
</tr>
<tr>
<td>is</td>
<td>CLE</td>
</tr>
<tr>
<td>lu</td>
<td>CLE</td>
</tr>
<tr>
<td>sp</td>
<td>CLE</td>
</tr>
<tr>
<td>ua</td>
<td>CLE</td>
</tr>
</tbody>
</table>

NPB - Single node
NPB Multi-node

NAS Parallel Benchmarks 3.3
NPROCS=256 CLASS=D

Time in seconds

 benchmark  
<table>
<thead>
<tr>
<th></th>
<th>CLE</th>
<th>Shifter</th>
<th>rkt</th>
<th>Singularity runC</th>
<th>CLE</th>
<th>Shifter</th>
<th>rkt</th>
<th>Singularity runC</th>
<th>CLE</th>
<th>Shifter</th>
<th>rkt</th>
<th>Singularity runC</th>
</tr>
</thead>
<tbody>
<tr>
<td>bt</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>cg</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>ep</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>ft</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>is</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>lu</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>mg</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>sp</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
</tr>
</tbody>
</table>

NPB
Quantum ESPRESSO

Quantum ESPRESSO 6.0 / Broadwell

Execution time (secs)

<table>
<thead>
<tr>
<th>Cores</th>
<th>CLE</th>
<th>Shifter</th>
<th>Singularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>144</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>216</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>252</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>288</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quantum ESPRESSO

Quantum ESPRESSO 6.0 / Broadwell hugepage

Execution time (secs)

<table>
<thead>
<tr>
<th>cores</th>
<th>CLE</th>
<th>Shifter</th>
<th>Singularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>260</td>
<td>240</td>
<td>220</td>
</tr>
<tr>
<td>72</td>
<td>210</td>
<td>190</td>
<td>170</td>
</tr>
<tr>
<td>108</td>
<td>160</td>
<td>140</td>
<td>120</td>
</tr>
<tr>
<td>144</td>
<td>110</td>
<td>90</td>
<td>70</td>
</tr>
<tr>
<td>180</td>
<td>60</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>216</td>
<td>30</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>252</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>288</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Radioss – Performance

Radioss : Offset to CLE
Shifter: Broadwell ppn:36

Elaspsed Time (sec)

Number of Cores

36  72  108  144

without hugepage support
Radioss – Performance

Radioss: Offset to CLE
Shifter: Broadwell ppn:36

Number of Cores

Elapsed Time (sec)

36 72 108 144

- without hugepage support
- hugepage filesystem support - Aries

CUG 2017
Radioss – Performance

Radioss : Offset to CLE
Shifter: Broadwell ppn:36

Elapsed Time (sec)

Number of Cores

without hugepage support
hugepage filesystem support - Aries
application hugepage support
Radioss – Performance

Radioss : Offset to CLE
Shifter: Broadwell ppn:36

Elapsed Time (sec)

Number of Cores

- without hugepage support
- hugepage filesystem support - Aries
- application hugepage support
- environment propagation
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.