

Chapel in Containers

Sung-Eun Choi
Principal Engineer
Cray R&D

Chapel Implementers and Users Workshop
Lightning talk session
2 June 2017

Containers: the buzz word

- **Containerized environments provide isolation**
 - Lighter weight than full virtualization
 - Repeatable and self-consistent
 - Originally designed for micro services

- **Turns out to be handy for other things**
 - Software distribution
 - Reproducibility

- **Everyone's on the bandwagon**
 - *HPC Containers in Use*. Jonathan Sparks, CUG 2017

Containers: in HPC



- **Users are accustomed to “logging in”**
- **Obvious solution**
 - Install all tools in the container
 - Treat it like an interactive shell
- **This is the guidance for Chapel on DockerHub***

* Shamefully, at my insistence

Containers: for programming languages

- **Fire up the container to run the compiler, etc.**
 - Go's guidance on DockerHub

- **For Chapel, you'd want**
 - printchplenv
 - compiler
 - runtime
 - tools

Chapel in containers

- **Wrapper script for binaries and script (e.g. chpl, printchplenv)**
 - Invoked with current Chapel environment variables
 - Must handle include paths and other file-related info
- **Launcher invokes the container**
 - Container environment should be embedded in the launcher
- **Container overhead**
 - Insignificant for compilation and most programs
 - Noticeable for printchplenv, but better the second time

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.: ACE, APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, 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.