

Chapel: Hands-on

Morning Hands-on Session

Goals:

- Get everyone up and running with Chapel
- Try out base language and data parallel features

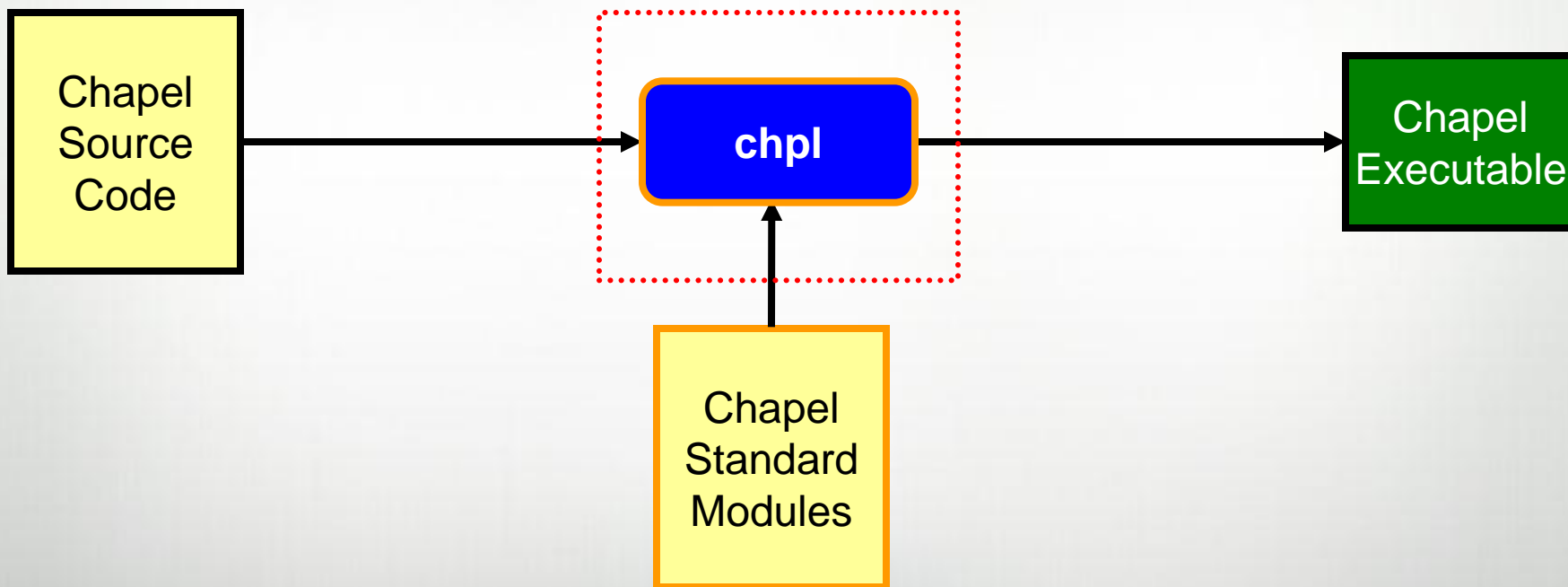
“What can I do?”

- Work through prepared Mandelbrot programming exercises
- Read and execute sample programs (`$CHPL_HOME/examples`)
- Write your own Chapel computation of interest

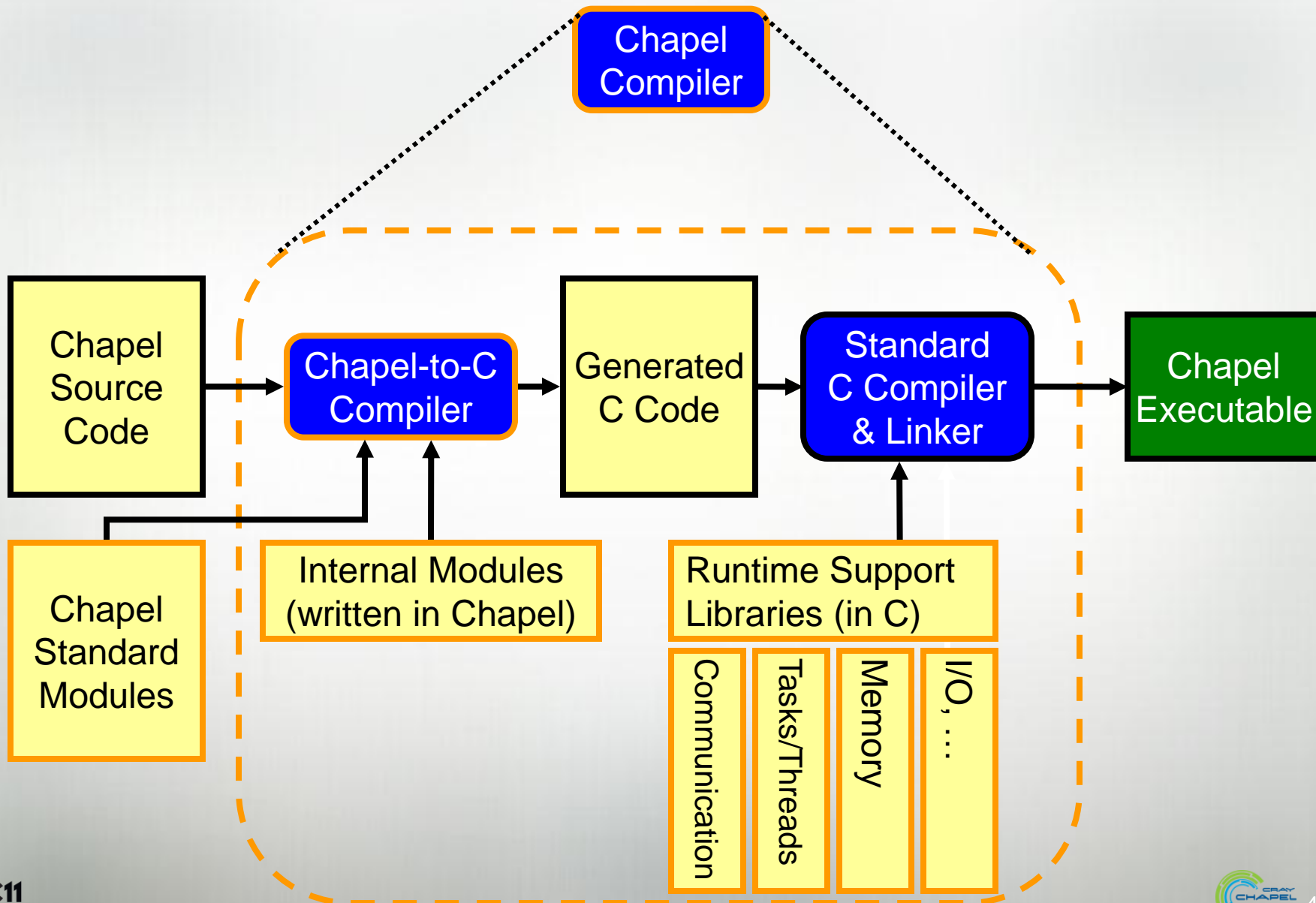
Please feel free to:

- Work with a partner
- Ask questions/talk with the team

Compiling Chapel



Chapel Compiler Architecture



Chapel Directory Structure (Partial)

- chapel-1.4.0/ – \$CHPL_HOME refers to this directory
- README – quick-start instructions for building & using **chpl**
- also contains pointers to key documentation
- doc/ – language spec, READMEs, quick reference
- examples/ – sample codes written in Chapel
- man/ – man page for **chpl**
- bin/ – location of the Chapel compiler
- lib/ – location of the Chapel runtime libraries
- modules/ – location of the Chapel standard/internal modules
- README.files – complete directory structure description

Materials For the Hands-On Sessions

1. A Chapel Quick Reference Sheet
 - Provides a succinct overview of the language
2. A prepared series of programming exercises based on Mandelbrot set calculation

Mandelbrot Exercises

Goal: Draw the Mandelbrot set

What we give you:

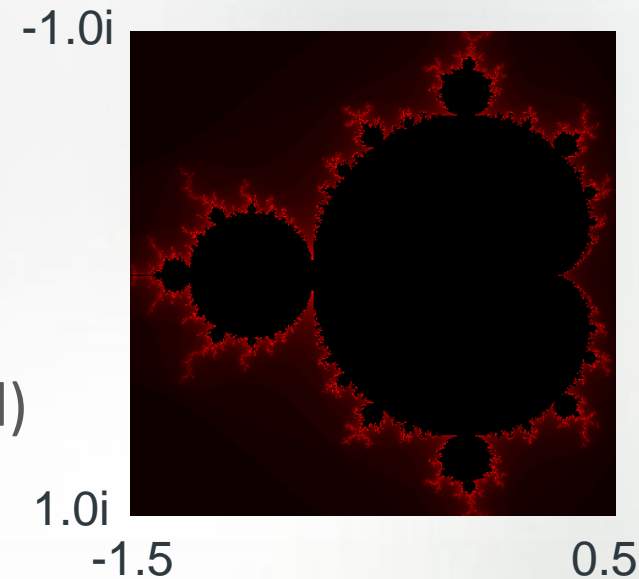
- A program template (mandelbrot.chpl)
- Image file utils for P*M files (MPlot.chpl)

What you'll do:

- Compute the Mandelbrot set image array

Five main variations (and other optional explorations):

- | | | |
|-------------------------------|---|--------------------|
| 1. serial | } | morning hands-on |
| 2. data parallel | | |
| 3. task parallel | } | afternoon hands-on |
| 4. multi-locale task parallel | | |
| 5. multi-locale data parallel | | |



Using Chapel Today

- Install your own version of Chapel...

...on a laptop

- Linux/UNIX
- Mac OS X
- Windows if you have Cygwin installed (*but* it tends to be slow)

...or by ssh-ing to a remote Linux/UNIX-based system

- Feel free to partner with someone else

- For I/O enthusiasts:

- One of our collaborators, Michael Ferguson (LTS) has just released a “technology preview” release containing nice new I/O features. If you’d like to give it a try, look in

<https://sourceforge.net/projects/chapel/files/chapel/1.4.0/QIO-techpreview/>

(or on your USB stick)

Q&A

Q: Where do I get the Chapel release?

A: From your Chapel USB stick or <http://sourceforge.net/projects/chapel/> after unpacking, start with the top-level README

Q: Where do I get the Mandelbrot exercises?

A: From your USB stick, or <http://chapel.cray.com/tutorials/SC11>

Q: Where can I get the final tutorial slides?

A: On your Cray USB stick -- the “official” version from SC11 is an earlier draft

Q: Where can I get more documentation on the language?

A1: \$CHPL_HOME/doc/chapelLanguageSpec.pdf or <http://chapel.cray.com/spec/spec-0.82.pdf>

A2: The quick reference handout (also in \$CHPL_HOME/doc/)

Q: Where can I get more documentation on the compiler itself?

A: (1) chpl --help; (2) man chpl; (3) \$CHPL_HOME/doc/README.compiling

Executing Multi-Locale Programs

- By default, Chapel compiles for a single locale
 - environment variable `CHPL_COMM` defaults to 'none'
 - Effect: no communication inserted by compiler
 - The `Locales` array exists, but has just one element

- To execute using multiple locales...
 - Requires making some additional settings and building a second version of the runtime
 - See `$CHPL_HOME/doc/README.multilocale` for instructions
 - Although a laptop is typically considered a single-locale machine, you can oversubscribe it to try multi-locale runs