

# Hands-On I: Hello worlds



# Safe Harbor Statement

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts. These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.



# Live Walkthrough of Online Resources



COMPUTE | STORE | ANALYZE

# "Hello World" in Chapel: Two Versions

- Fast prototyping

```
writeln("Hello, world!");
```

- "Production-grade"

```
module Hello {  
  
    proc main() {  
        writeln("Hello, world!");  
    }  
}
```



# "Hello World" in Chapel: Two Versions

- Fast prototyping (configurable)

```
config const audience = "world";
writeln("Hello, ", audience, "!!");
```

- “Production-grade” (configurable)

```
module Hello {
  config const audience = "world";

  proc main() {
    writeln("Hello, ", audience, "!!");
  }
}
```

- To change ‘audience’ for a given run:

- ./hello --audience=ACCU2017

# Steps to “Hello World”

- **Point web browser to online resources**
  - <http://chapel.cray.com/tmp/ACCU2017>
- **Download and build Chapel**
  - most find it nicest to start with a familiar machine...
  - can also get a Cray account and use a pre-installed version
- **Create hello.chpl with your favorite editor:**
  - `writeln("Hello, world! ");`
- **Compile it**
  - `chpl -o hello hello.chpl`
- **Run it**
  - `./hello`



# Check out other “Hello world” versions

- **Take a look at \$CHPL\_HOME/examples/hello\*.chpl**
  - Six variations that show off different language features
  - Also rendered online at: <http://chapel.cray.com/docs/latest/examples/>
- **Do whatever else you can / want before the morning break**
  - Additional examples can be found in \$CHPL\_HOME/examples
    - **primers/** introductions to various language concepts
    - **benchmarks/** some standard benchmarks written in Chapel
  - Primer examples also online:
    - <http://chapel.cray.com/docs/latest/primers/index.html>
- **After the break:**
  - Base language
  - Data parallelism
  - More hands-on
  - Task Parallelism, Locality, ...



# Optional Compiler Architecture Sidebar

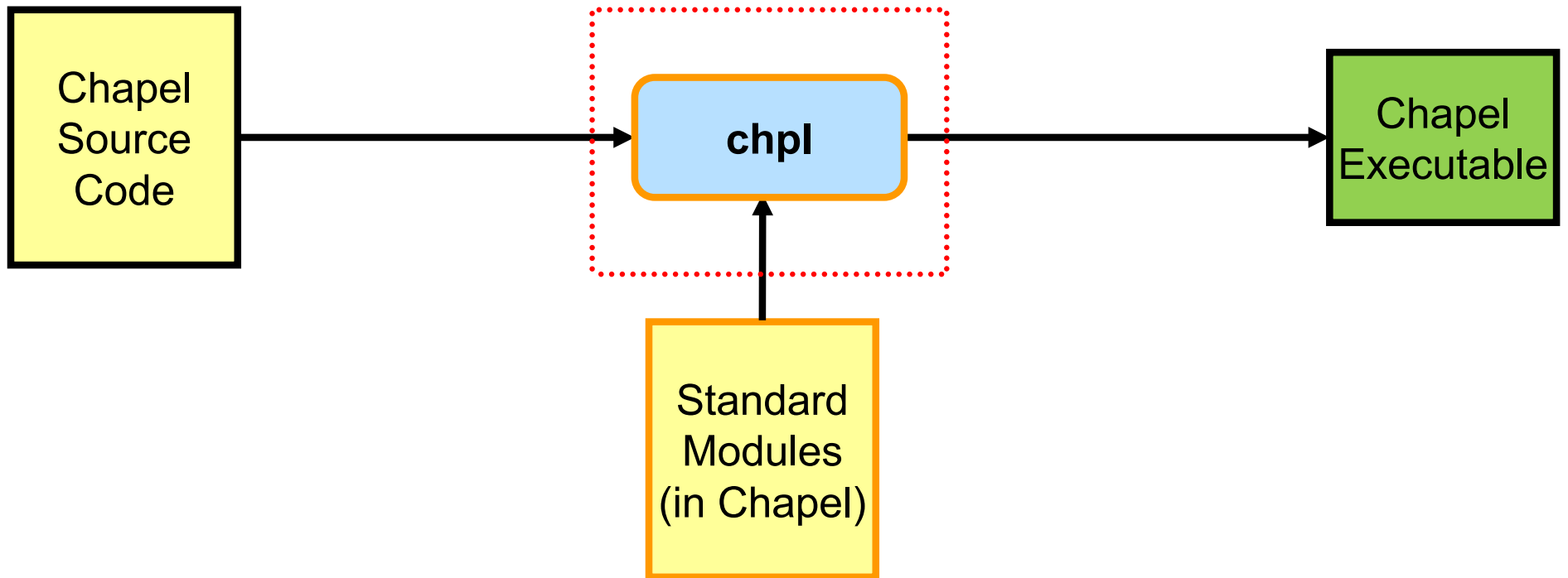


COMPUTE | STORE | ANALYZE

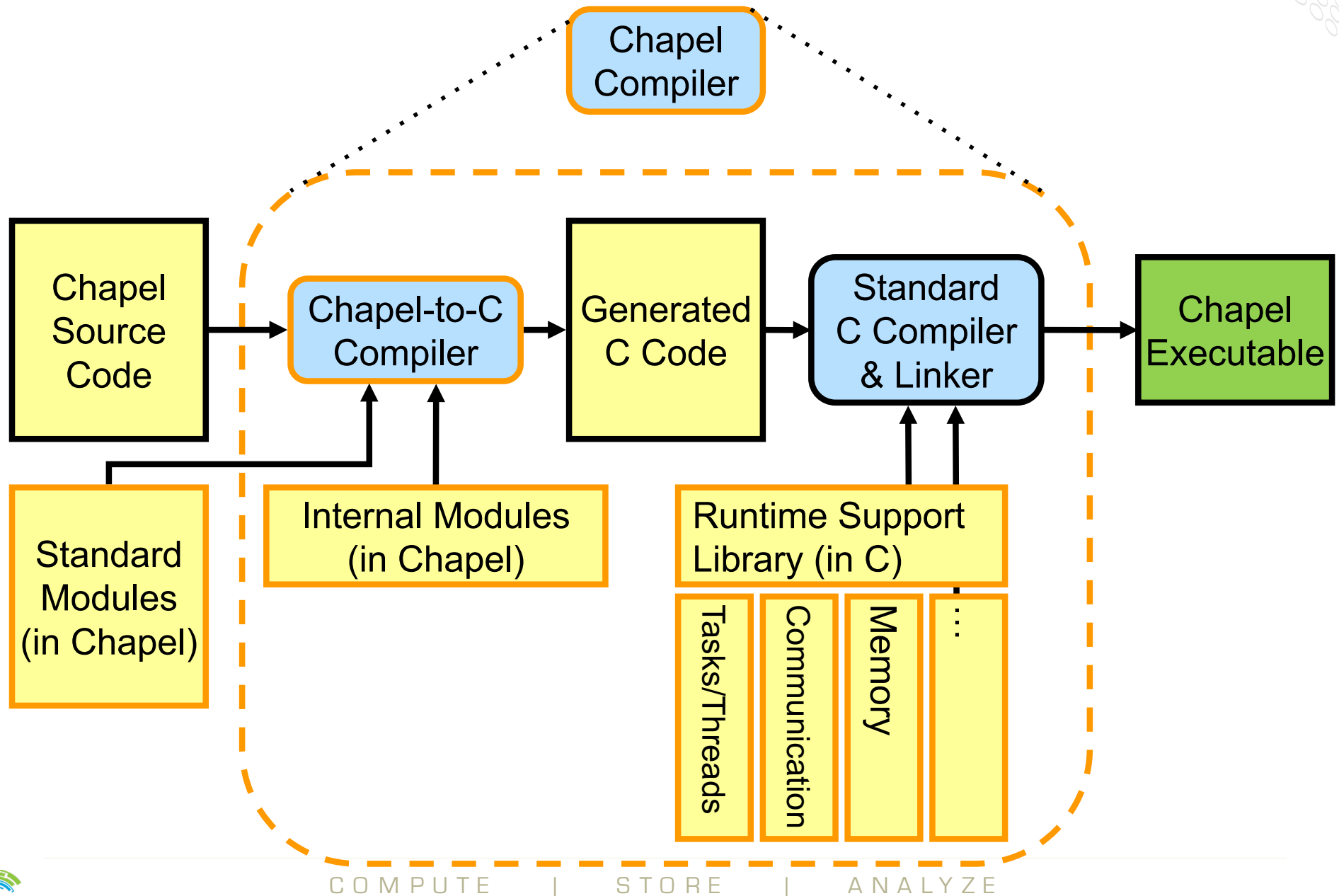
Copyright 2017 Cray Inc.



# Compiling Chapel



# Chapel Compiler Architecture



# LLVM back-end

- Chapel typically code-generates C
- Targeting LLVM is also an option
  - Not on by default due to build-time required
- To enable:
  - set CHPL\_LLVM=llvm and re-'make'
  - compile with 'chpl --llvm' (or set CHPL\_LLVM\_CODEGEN)
- Enabling LLVM also supports an interoperability capability

```
extern {
    void myCFunc() {                                     // myCFunc() can be called from Chapel
        printf("Hello from C!");
    }
    #include "gsl.h"                                     // as can functions defined in gsl.h...
}
```



# 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.*

