Compiler Improvements

Chapel Team, Cray Inc.
Chapel version 1.13
April 7, 2016
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.
Outline

● Controlling / Exposing Chapel’s Configuration
● Complex Type Improvements via C99 Complex
● Other Compiler Improvements
Controlling / Exposing Chapel’s Configuration
Chapel Configuration: Background

**Background:**

- Chapel’s configuration specified by ~23 environment variables
  - CHPL_HOME, CHPL_COMM, CHPL_MEM, CHPL_TASKS, ...

- Changing configurations required changing env. vars:
  ```
  export CHPL_COMM=none
  chpl foo.chpl -o foo-shared
  export CHPL_COMM=gasnet
  chpl foo.chpl -o foo-dist
  ```

- Built-in CHPL_* params have permitted Chapel code to access config.
  - yet never documented with chpldoc
Chapel Configuration: This Effort

- **Added compiler flags for specifying configuration**
  - `--home`, `--comm`, `--mem`, `--tasks`, ...
- Backed by existing CHPL_* environment variables
  - more consistent with other compiler flags
- Simplifies ability to switch between configurations:
  - `chpl --comm=none foo.chpl -o foo-shared`
  - `chpl --comm=gasnet foo.chpl -o foo-dist`
- Flags documented in man page and `chpl --help` output

- **Refactored params into their own (internal) module**
  - Location:
    - `modules/internal/ChapelEnv.chpl`
  - Documented with chpldoc, accessible in online documentation:
    - [http://chapel.cray.com/docs/1.13/modules/internal/ChapelEnv.html](http://chapel.cray.com/docs/1.13/modules/internal/ChapelEnv.html)
Chapel Configuration: Next Steps

- **.chplrc file**
  - Add ability to specify default Chapel configuration / compilation flags
  - Will benefit package-based deployment and system-wide installations
  - Will simplify managing multiple configurations

- **Rename ChapelEnv.chpl and reword documentation**
  - Name/text reflects historical nature: environment variable-centric
Complex Type Improvements via C99 Complex
Complex Types: Background

- Complex types were implemented as homegrown records
  - Stored two *real* valued components
  - Performed math operations on the individual components
Complex Types: This Effort

- Replace our homegrown records with C99 complex types
  - Generate cleaner code
  - Generate safer floating point code

- Give back-end compiler more complete type information

- Simplify the compiler implementation
  - Removed complexToRecord pass

- Simplify interoperability with C libraries
Complex Types: Impact

● Improved generated code involving complex types

```c
proc f(a: complex, b: complex, c: complex) {
    return a * b / c;
}
```

```c
static _complex128 f(_complex128 a4,
    _complex128 b2,
    _complex128 c2) {
    _complex128 call_tmp;
    _complex128 call_tmp2;
    call_tmp = (a4 * b2);
    call_tmp2 = (call_tmp / c2);
    return call_tmp2;
}
```

v1.12

```c
static _complex128 f(_complex128 a4,
    _complex128 b2,
    _complex128 const c2) {
    _complex128 call_tmp;
    call_tmp = (a4 * b2);
    call_tmp2 = (call_tmp / c2);
    return call_tmp2;
}
```

v1.13

```c
_v1.13
```

Copyright © 2016 Cray Inc.
Complex Types: Performance-Safety Tradeoff

- Caused by back-end multiply and divide implementation
  - Safer for “weird” floating point values, but slower overall
    - NaNs
    - Infinities
    - Denormalized

- Compile with ‘--no-ieee-floats’ to regain the performance

Mandelbrot performance

Execution Time (seconds)

0 2 4 6 8 10

12
Other Compiler Improvements
Other Compiler Improvements

- **enum config values can now be fully qualified**
  (contributed by Nick Park)
  - supports,
    ```
    ./a.out --fillColor=color.red
    ```
  - whereas previously, one had to use:
    ```
    ./a.out --fillColor=red
    ```

- **the --ldflags flag now “stacks” when several are used**

- **the --no-warnings flag can now be reversed via --warnings**

- **improved filename representation in generated code**
  - switched from string literals to a table of filenames + indices into it

- **also:**
  - many performance improvements (covered elsewhere)
  - many developer-oriented improvements (see [CHANGES.md](#))

---

Copyright 2016 Cray Inc.
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.