Performance Results

Chapel Team, Cray Inc.
Chapel version 1.14
October 6, 2016

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.

Executive Summary

- **Generally speaking, performance has improved with 1.14**

- **Previous slide decks have shown performance changes:**
  …due to array optimizations
  …due to compiler and library optimizations
  …due to runtime improvements

- **These slides contain additional v1.14 performance results**
  - not tied to any specific effort, just comparisons across releases
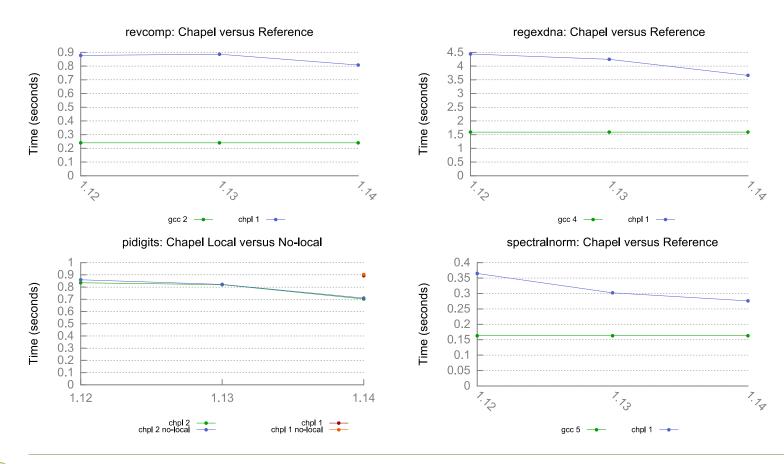
Outline

- **Shootout Benchmarks Trends**

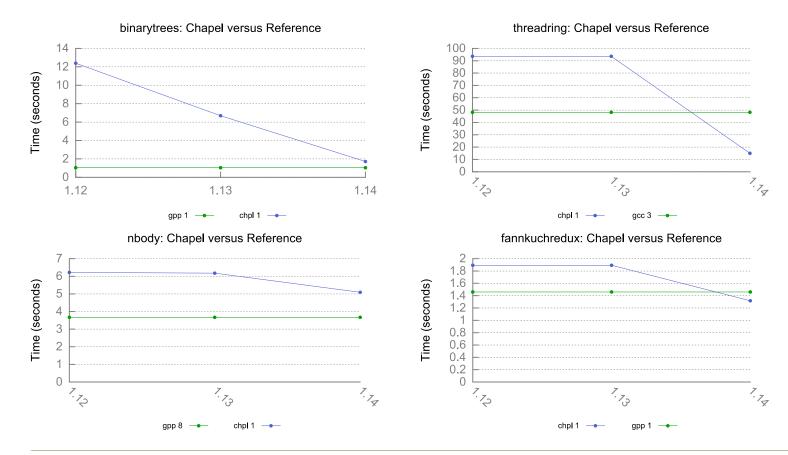- **Single-Locale Performance Trends**
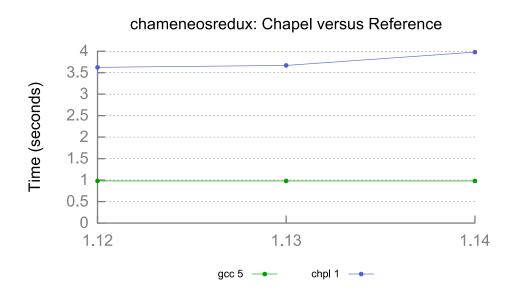
- **Multi-Locale Performance Trends**

Shootout Benchmarks Trends

Shootout Benchmarks Trends

- **Lots of improvements for shootout benchmarks**

COMPUTE | STORE | ANALYZE

Shootout Benchmarks Trends

- **Lots of improvements for shootout benchmarks**
  - in many cases, the improvements were significant

binarytrees: Chapel versus Reference

threadring: Chapel versus Reference

nbody: Chapel versus Reference

fannkuchredux: Chapel versus Reference

Shootout Benchmarks Trends

- ## Known regression for chameosredux
  - caused by an atomic load bug fix

chameneosredux: Chapel versus Reference

Shootout Benchmarks Trends

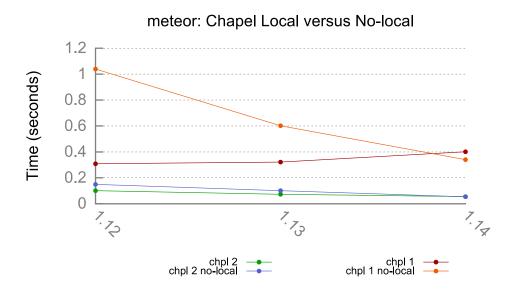- **Known regression for meteor-elegant**
  - also caused by an atomic load bug fix
    - surprisingly --no-local performance improved
    - meteor is an incredibly short benchmark, likely just cache effects (~40% of execution time is spent in runtime initialization)
  - happily, meteor-fast performance improved

meteor: Chapel Local versus No-local
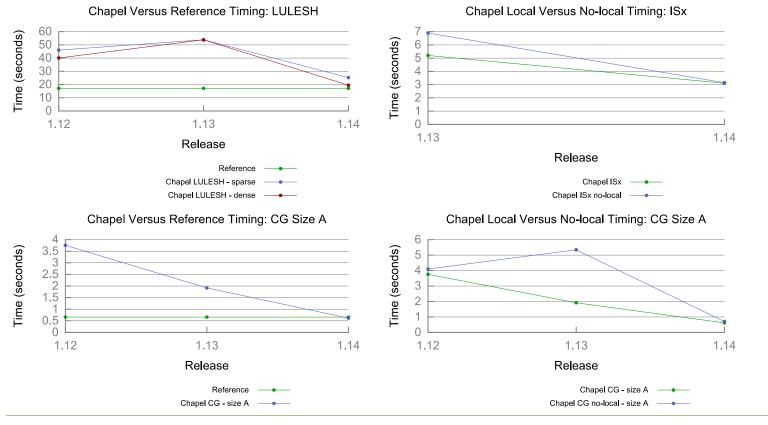
Single-Locale Performance Trends

Single-Locale Performance

- **Significant single-locale performance improvements**
  - new convergences w/ reference timings; between local and --no-local
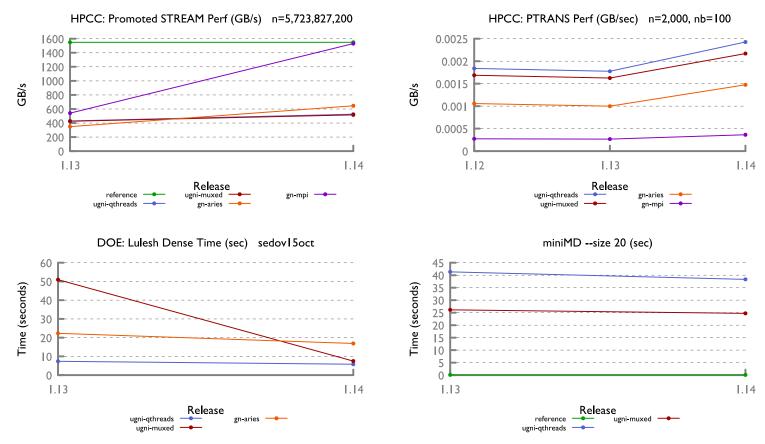 - no known regressions

Multi-Locale Performance Trends

Multi-locale Performance

- ## Multi-locale improvements for many benchmarks
  - ### no known regressions

Performance Priorities and Next Steps

Performance Priorities and Next Steps

- **Continue to improve ugni+qthreads performance**
  - understand differences compared to ugni+muxed
  - strive to close performance gaps and retire muxed tasking

- **NUMA-aware performance**
  - strive to support NUMA by default without performance loss

- **Continue benchmark-driven improvements**
  - single locale:
    - eliminate remaining performance gap for LCALS
    - improve performance for shootouts and proxy apps
  - multi-locale:
    - reduce unnecessary communication code
    - optimize scalability of core algorithms (task spawning, reductions, barriers)
    - focus on ISx, MiniMD/CoMD, LULESH
    - consider submitting an entry for 2017 HPC Challenge

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.