Performance Results

Chapel Team, Cray Inc.

Chapel version 1.13

April 7, 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 1.13**
- **Previous slide decks have shown performance changes:**
  …due to communication and locality optimizations
  …due to performance optimizations in the compiler and libraries
  …due to making jemalloc the default allocator
  …due to improving the ugni communication layer
- **These slides contain additional v1.13 performance results**
  - Not tied to any specific effort, just comparisons across releases

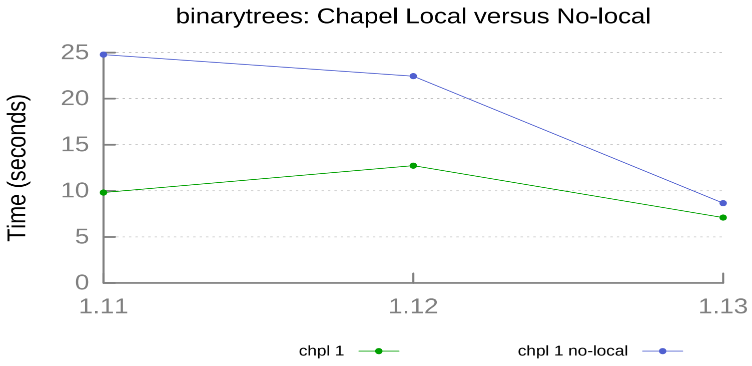
Outline

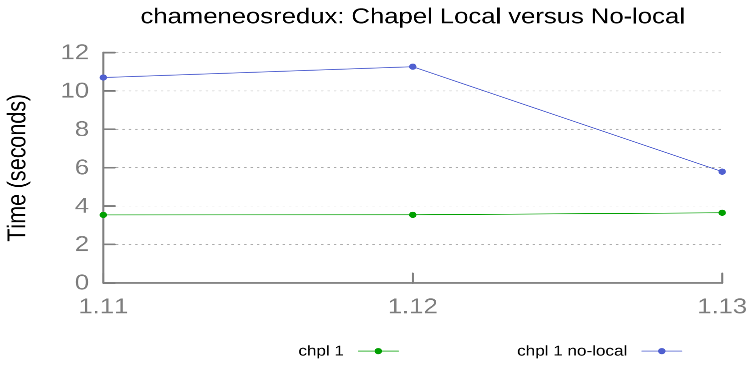
- Shootout Benchmarks Status
- Single-Locale Performance Trends
- Multi-Locale Performance Trends
- Performance Scalability Study

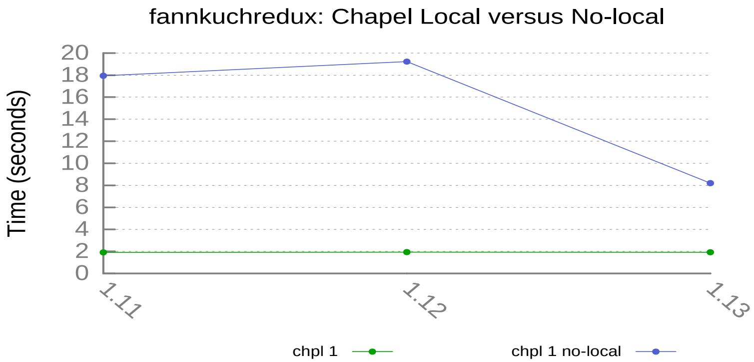
Shootout Benchmarks Status

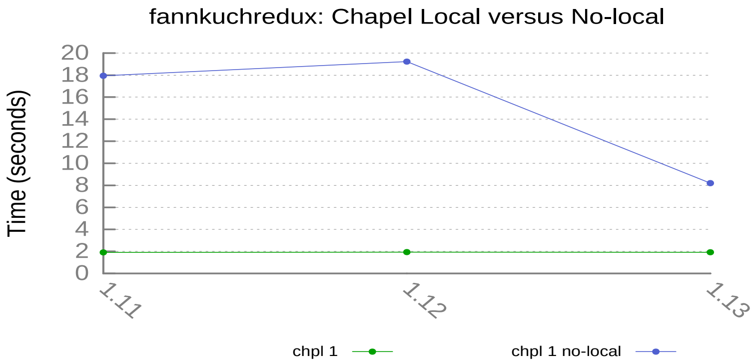
Shootout Benchmark Summary

- **By design, not much effort put into shootouts for 1.13**
  - however, other work resulted in significant speedups
    - particularly for --no-local timings



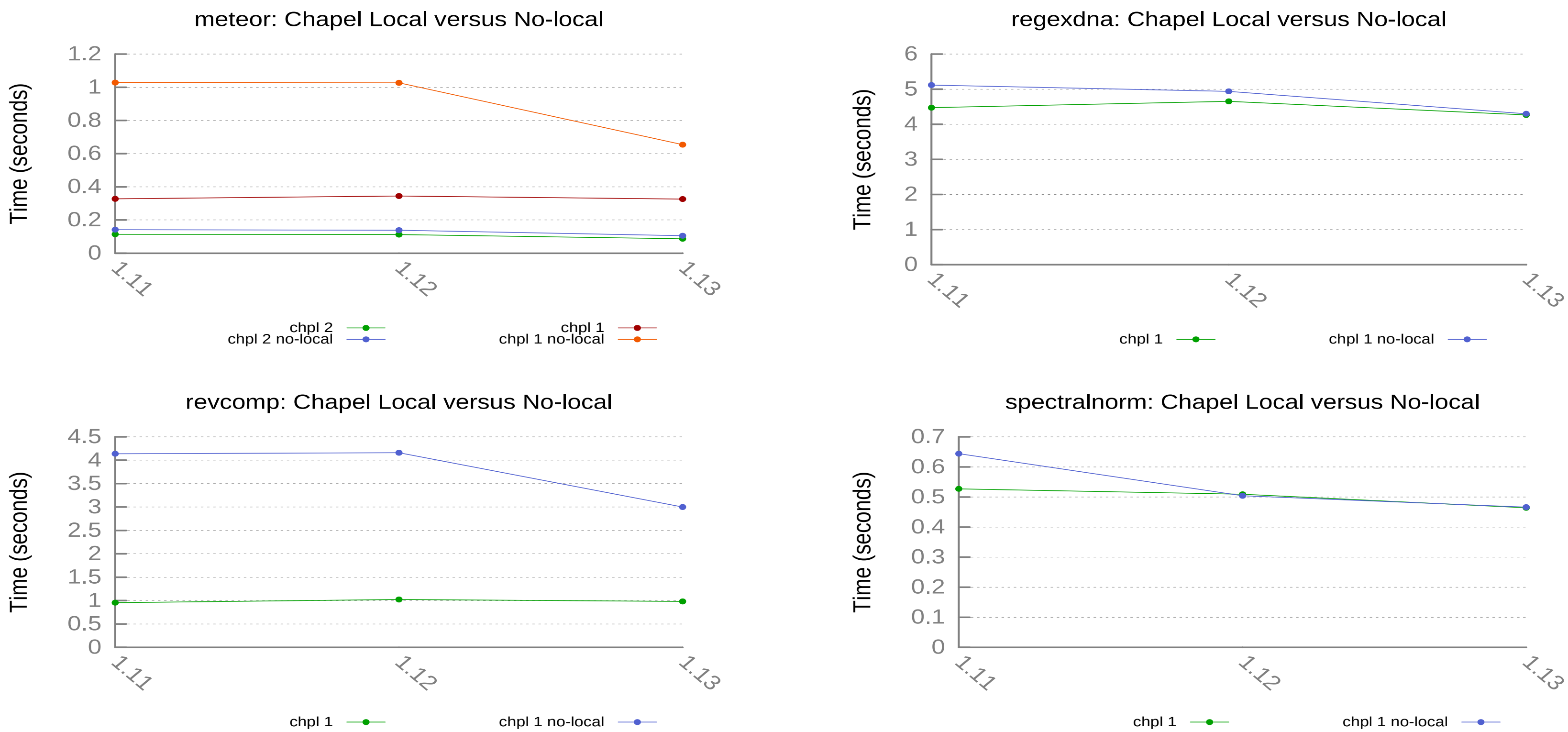






Shootout Benchmark Summary

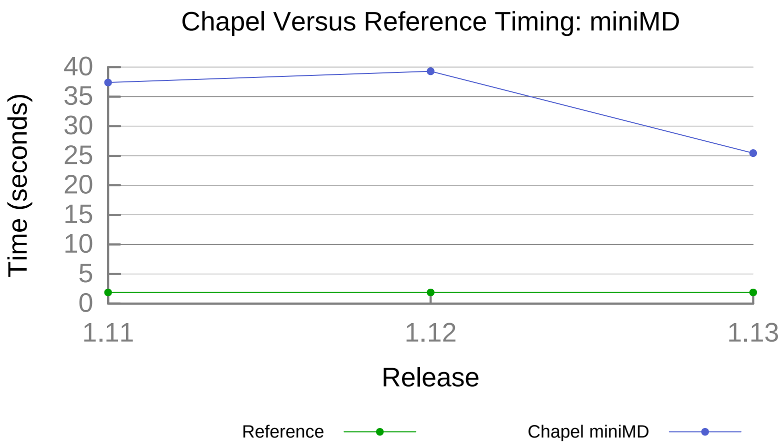
- **By design, not much effort put into shootouts for 1.13**
  - however, other work resulted in significant speedups
    - particularly for --no-local timings

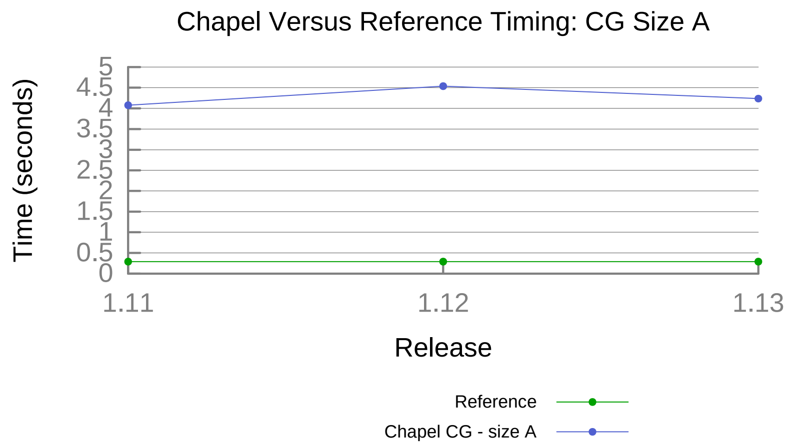


Single-Locale Performance Trends

Single-Locale Performance

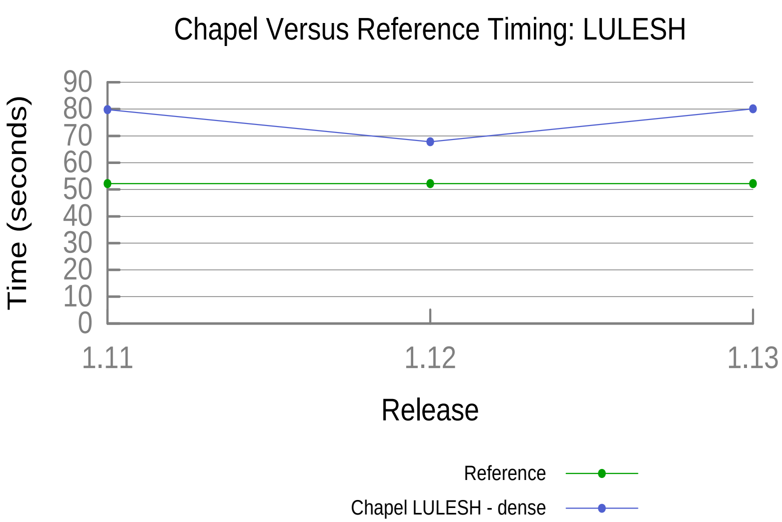
- Overall, single-locale performance improved

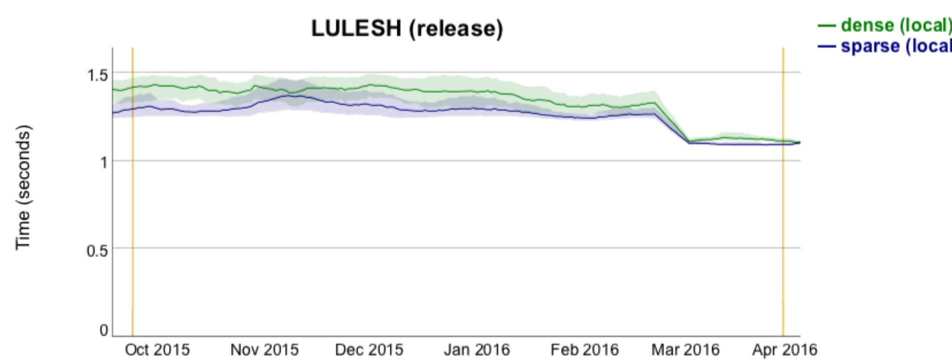




Single-Locale Performance

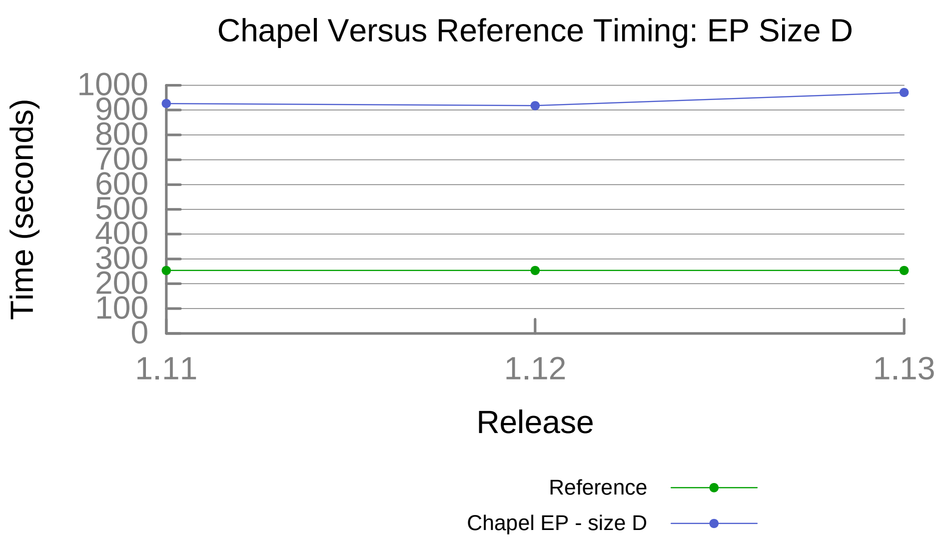
- **Surprising single-locale regression for LULESH**
  - nightly testing showed improvement
    - (uses a different problem size)
  - still need to investigate root cause





Single-Locale Performance

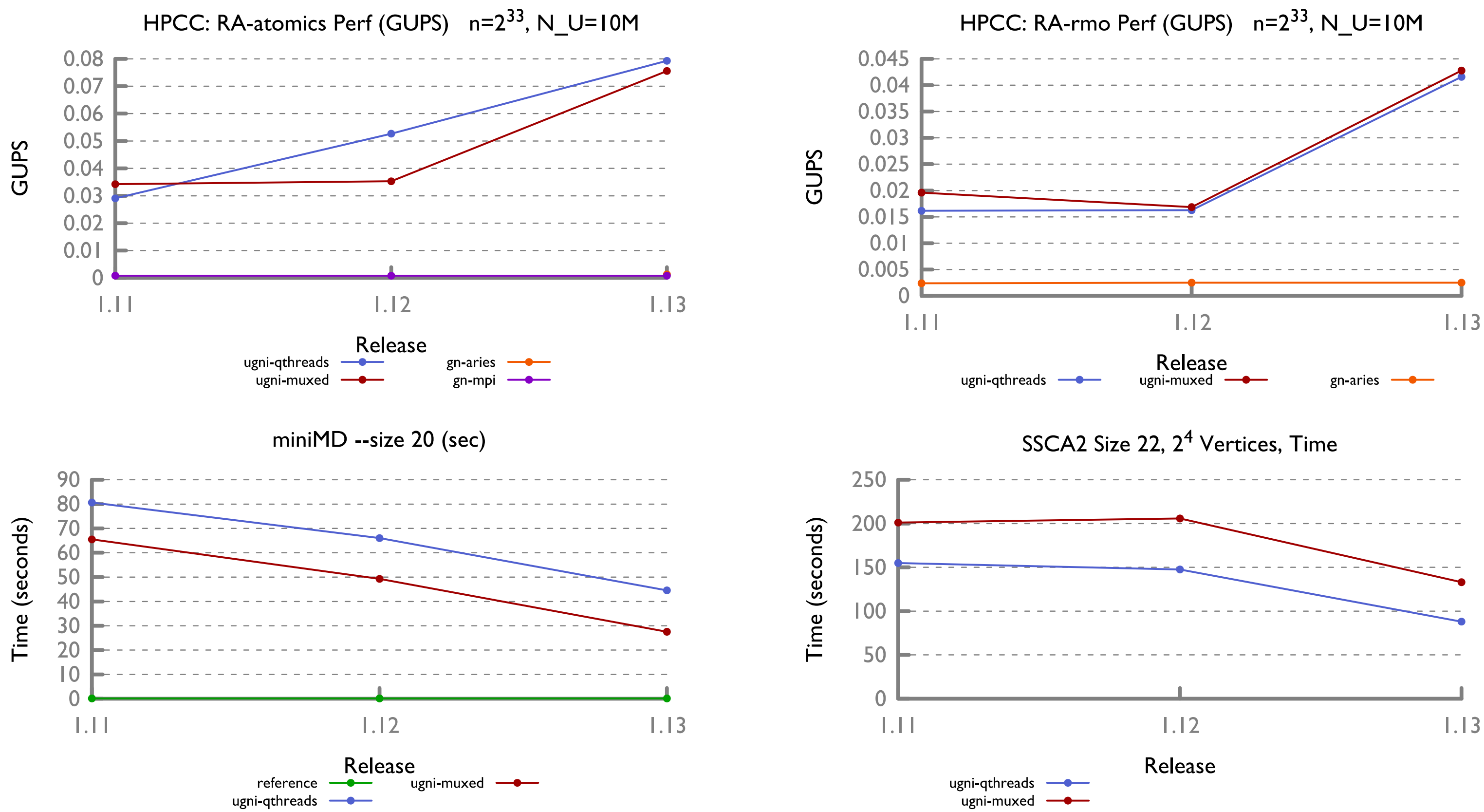
- **Known single-locale regression for NAS EP**
  - result of making jemalloc the default allocator
    - known ahead of time, but overall performance trend was extremely positive
  - have not investigated further yet



Multi-Locale Performance Trends

Multi-locale Performance

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



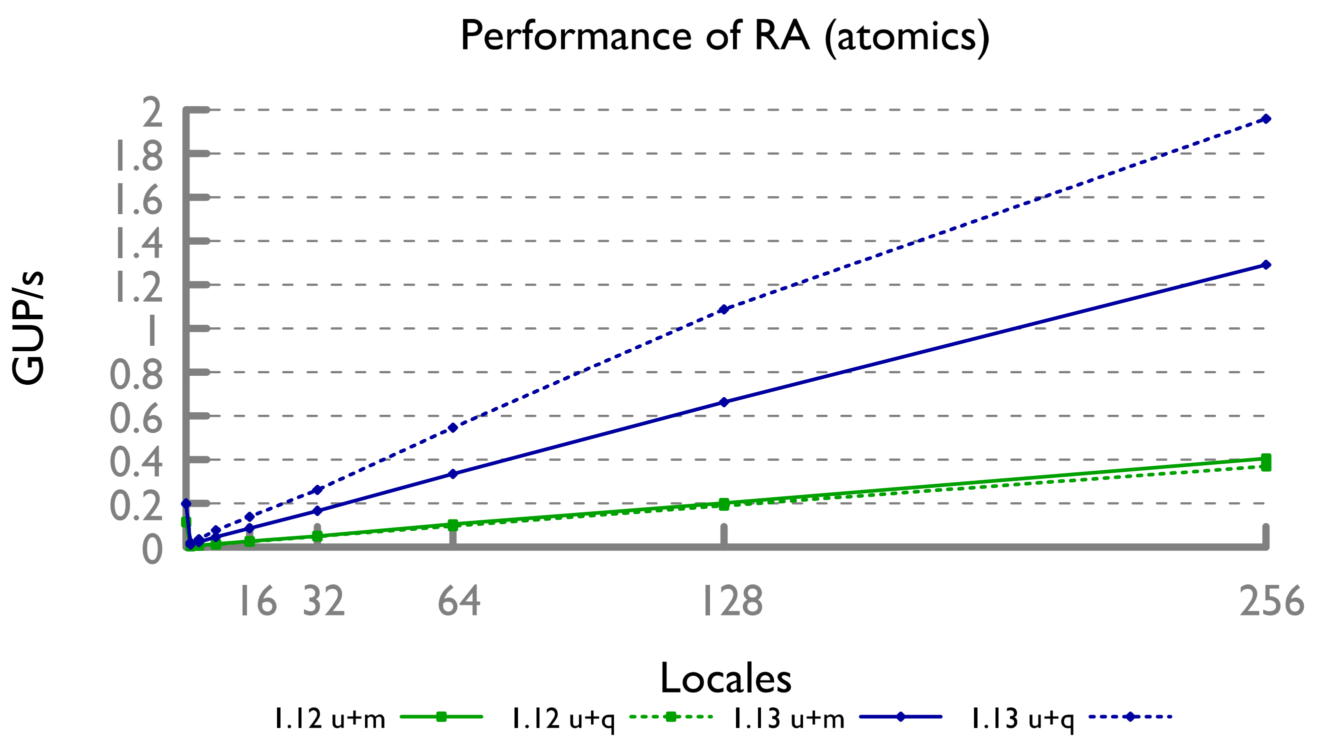
Performance Scalability Study

Scalability Study: Background

- **We continued the scalability study from past releases**
  - HPCC Stream: EP and Global
  - HPCC RA: atomic, on-based, and remote memory operations (rmo)
    - these test network atomics, active messages, and puts/gets, respectively
  - Reduction of an array
- **All experiments shown here were performed on a Cray XC**
  - 1-256 locales
- **The following slides highlight a few notable cases**
  - RA (atomics and rmo) performance has improved dramatically
    - up to 5x increase for ra-atomics and 3x for ra-rmo
  - Reductions are significantly more efficient
  - Stream and ra-on performance has not changed
    - (graphs omitted for this reason)

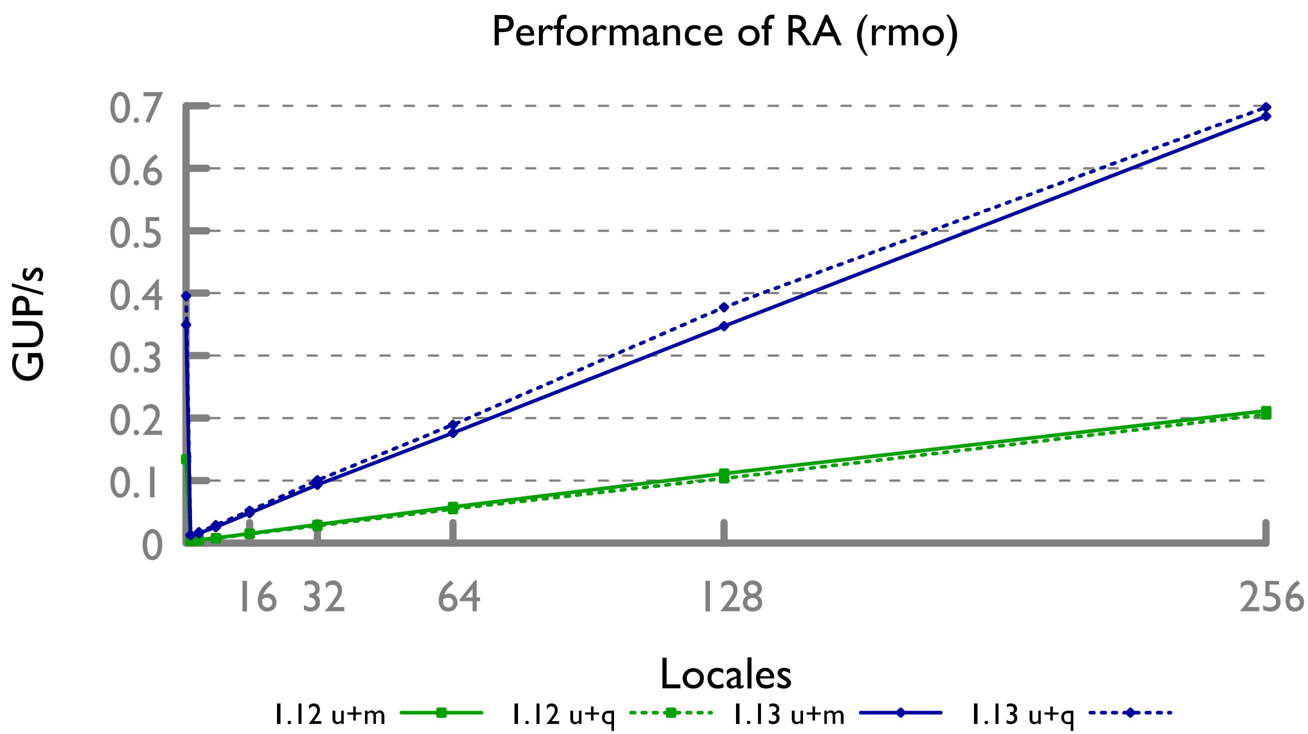
Scalability: RA (atomics) Performance

- **RA (atomics) summary**
  - 5x better performance for ugni-qthreads
  - 3x better performance for ugni-muxed



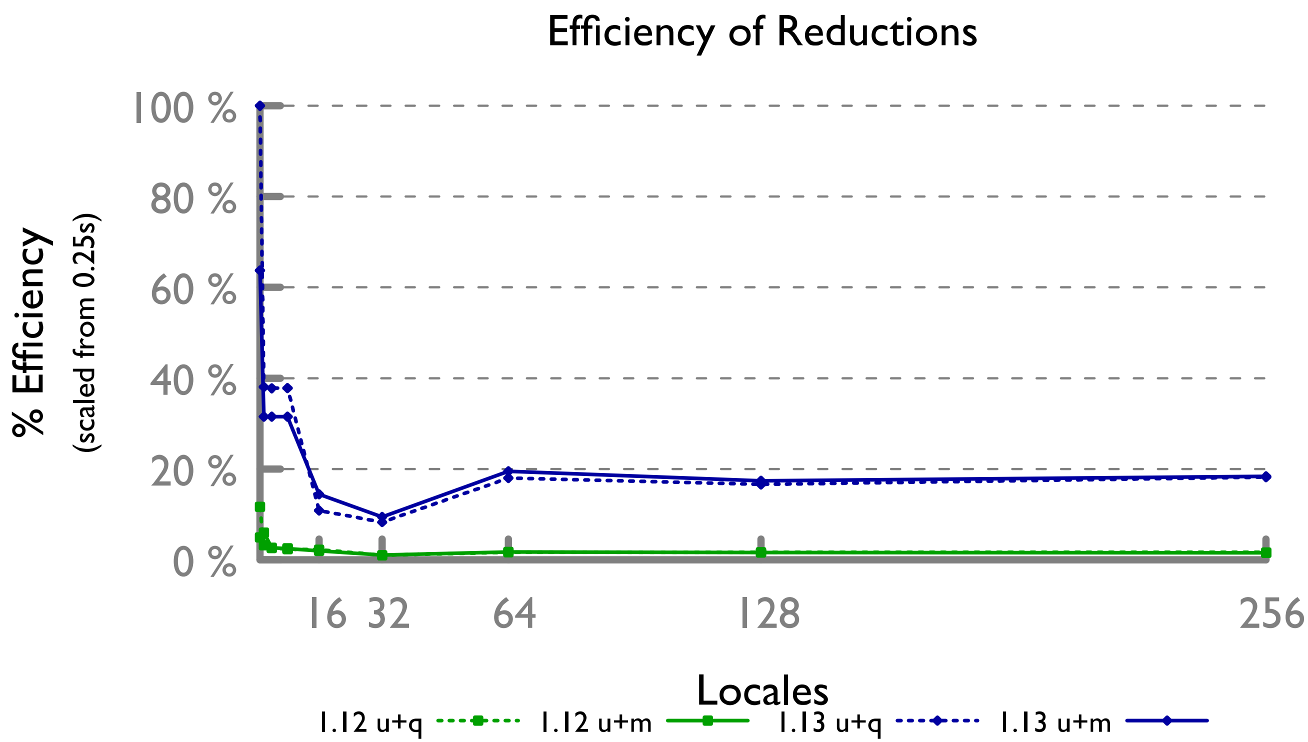
Scalability: RA (rmo) Performance

- **RA (rmo) summary**
  - 3x better performance for ugni-qthreads and ugni-muxed



Scalability: Reduction Efficiency

- **Reduction efficiency summary**
  - improved scalability
  - significantly improved raw performance



Performance Priorities and Next Steps

Performance Priorities and Next Steps

- **Continue to focus on ugni+qthreads performance**
  - understand differences compared to ugni+muxed
  - strive to close performance gaps and retire muxed tasking
- **NUMA-aware performance**
  - improve array initialization (parallel, appropriate first-touch)
    - currently gated by constructor/default init/noinit capabilities
  - strive to support NUMA by default w/out performance loss
- **KNL performance**
  - improve vectorization performance
  - explore benefits of high bandwidth memory
- **Continue benchmark-driven multi-locale improvements**
  - Reduce unnecessary communication code
  - Optimize scalability of core algorithms such as task spawning, barrier

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.

http://chapel.cray.com chapel_info@cray.com http://sourceforge.net/projects/chapel/