# Benchmarks and Performance Results

Chapel Team, Cray Inc.

Chapel version 1.12

October 1st, 2015

# 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 1.12**
- **Previous slide decks have shown performance changes:**
  …due to new interpretation of formal array arguments
  …due to optimizations motivated by stream
  …due to compiler locality optimizations
- **These slides contain additional v1.12 performance results**
  - Not tied to any specific effort, just comparisons across releases

# Outline

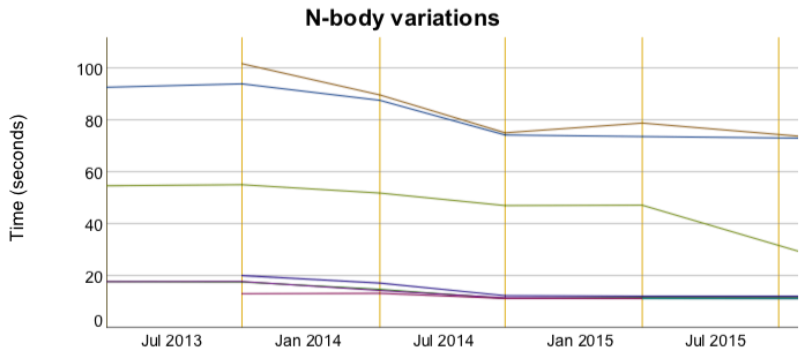
- Shootout Benchmarks Status
- Single-Locale Performance Trends
- Multi-Locale Performance Trends
- ugni+qthreads as default
- Performance Scalability Study

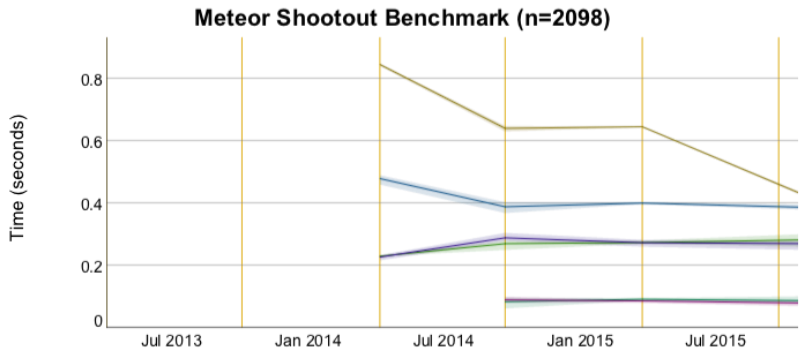
# Shootout Benchmarks Status

# Shootout Benchmark Summary

- **By design, not much effort put into shootouts for 1.12**
  - Wanted to focus primarily on multi-locale performance improvements
  - Official entry blocked by improved string support
  - A few of our fastest versions improved, but most stayed the same
  - Several of our non-fastest versions also improved

    ⇒ Chapel becoming less sensitive to writing in a specific style

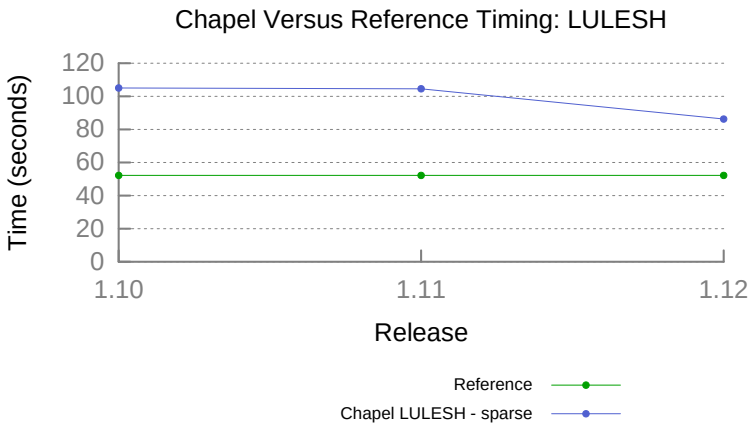


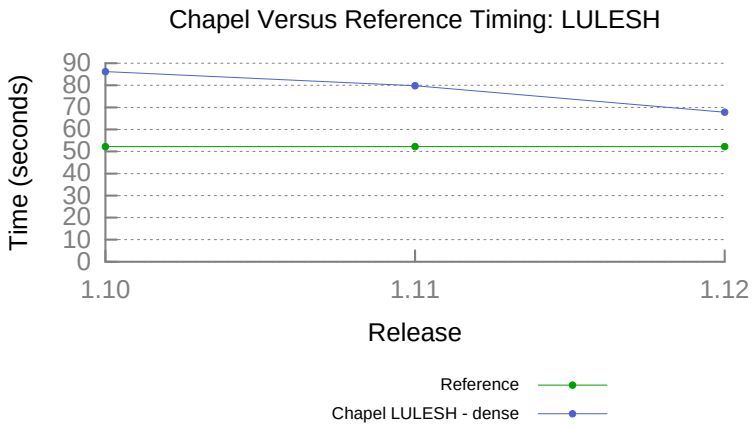



# Single-Locale Performance Trends

# Single-Locale Performance

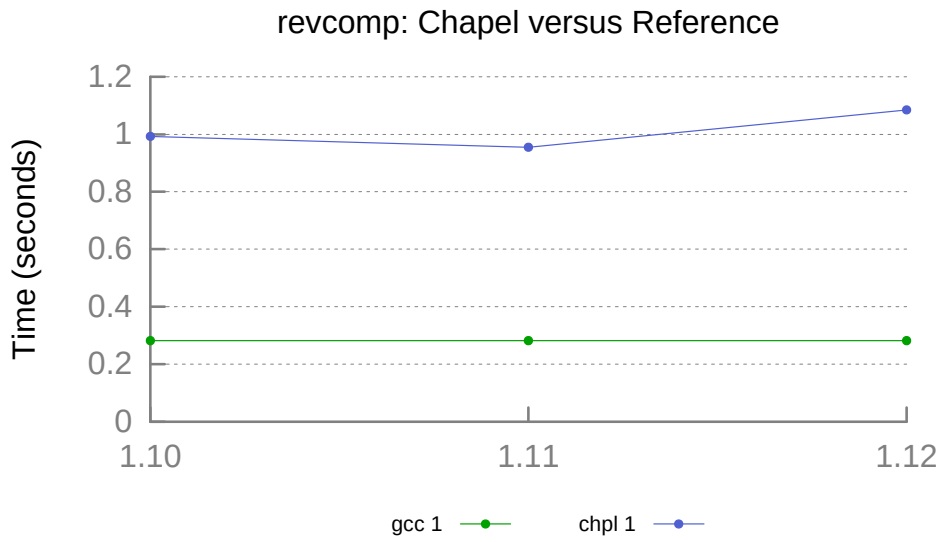
- **Overall, single-locale performance remained the same**
  - A few benchmarks such as lulesh showed improvements



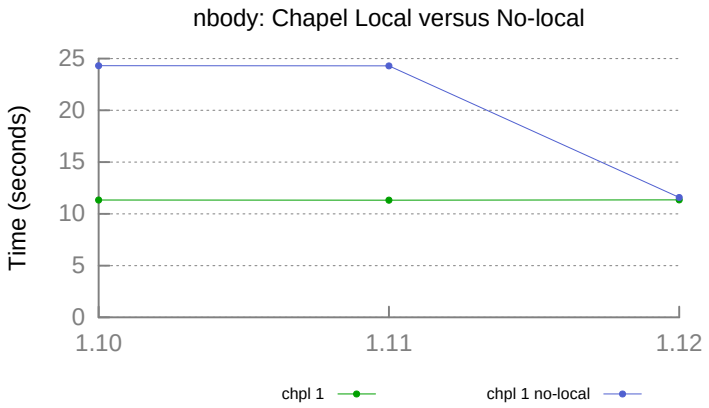
# Single-Locale Performance

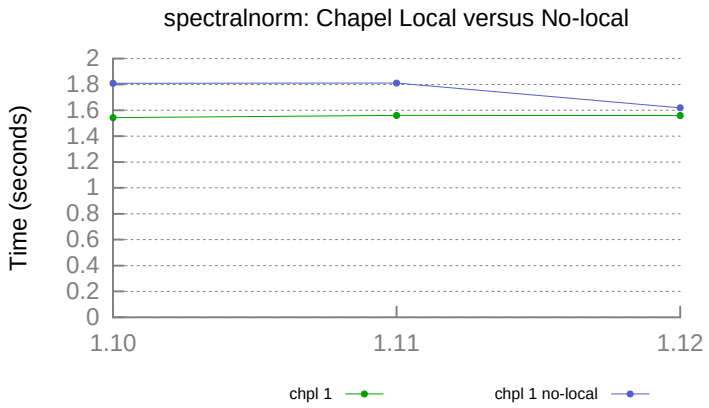
- **Overall, single-locale performance remained the same**
  - One known (and acceptable) regression for reverse-complement
    - result of a bug fix in I/O code (original code was fast, but wrong)

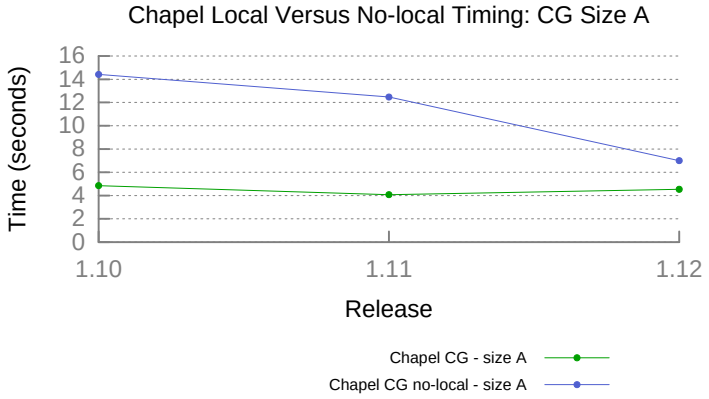


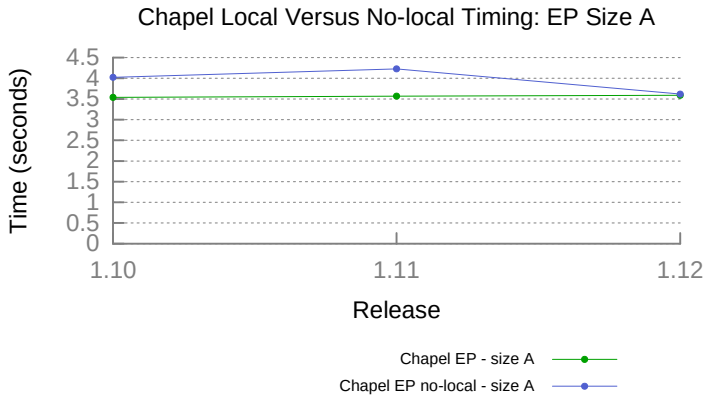
# Single-Locale Performance

- No-local execution improved due to locality optimizations



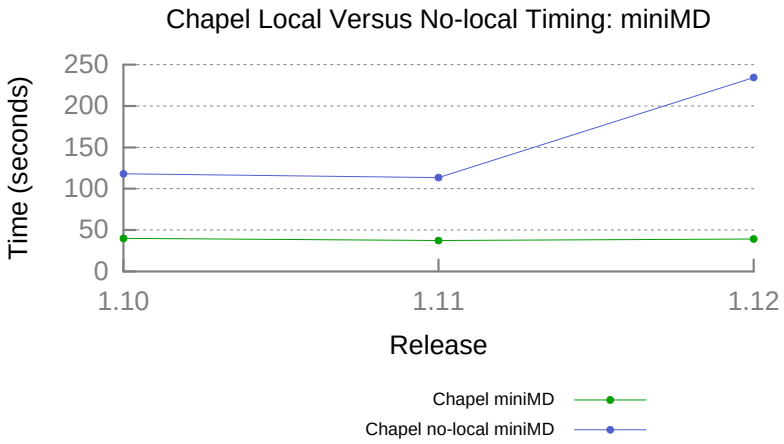



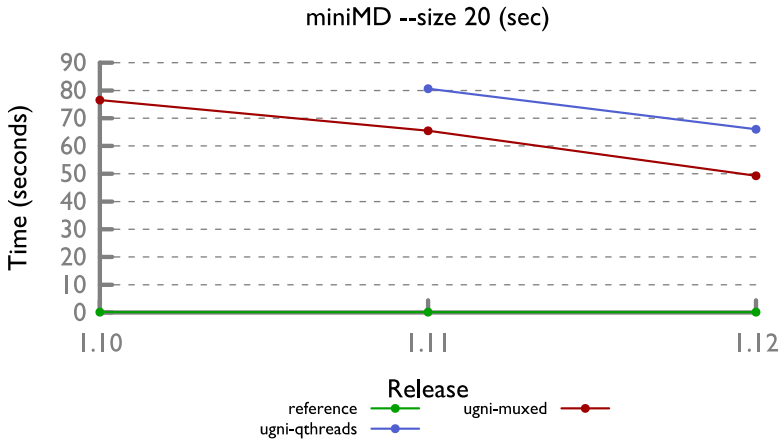



# Single-Locale Performance

- **Surprising no-local regression for miniMD**
  - Still investigating the root cause
  - Not seen in our nightly multi-locale or no-local performance testing
    - in fact multi-locale performance improved since last release

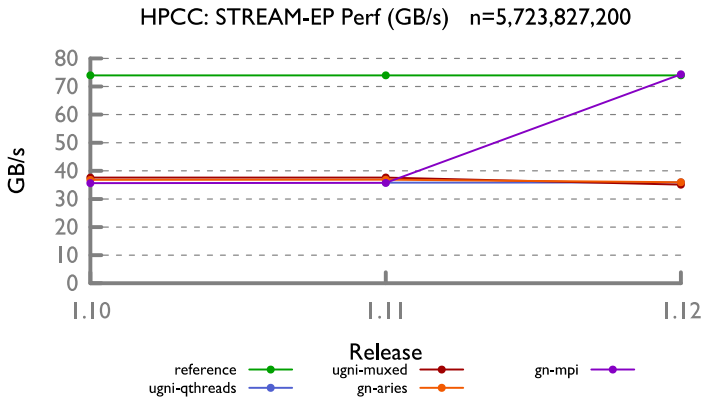


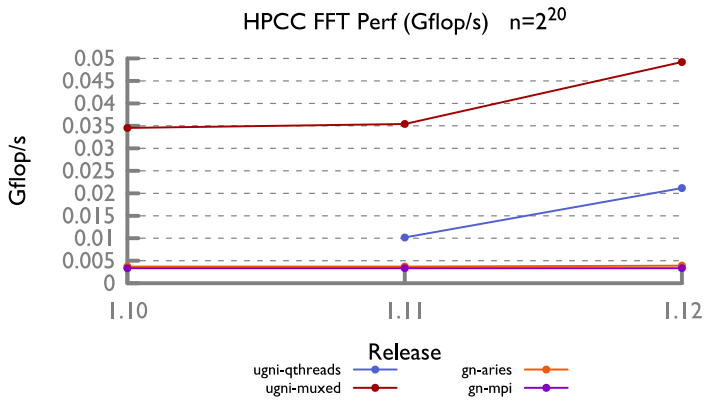



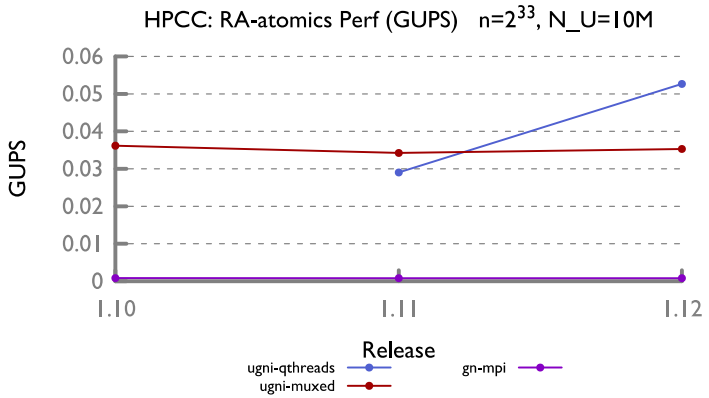
# Multi-Locale Performance Trends

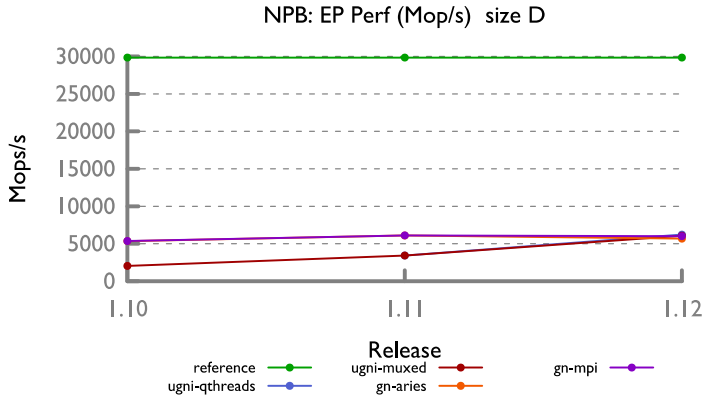
# Multi-locale Performance

- **Multi-locale improvements for many benchmarks**
  - Result of optimizations motivated by stream case study



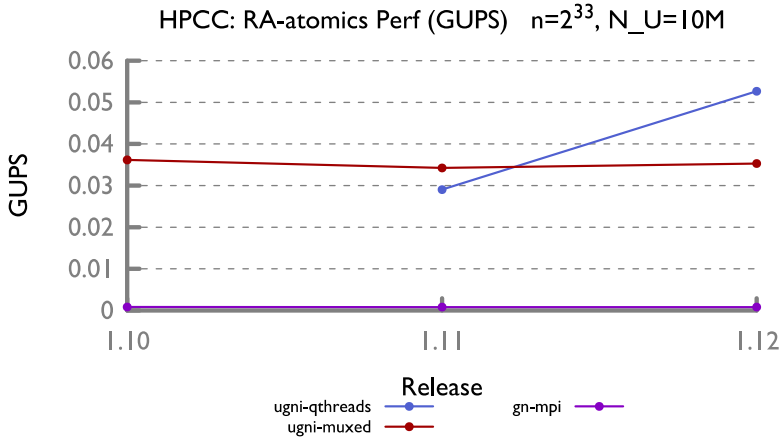



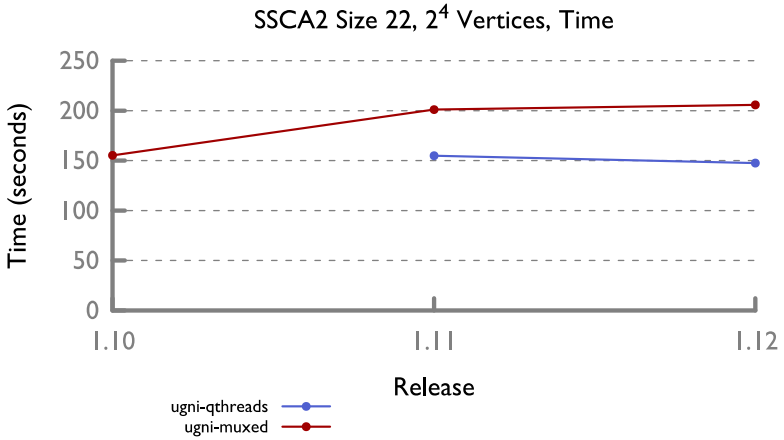



# ugni+qthreads as default

# ugni+qthreads as default

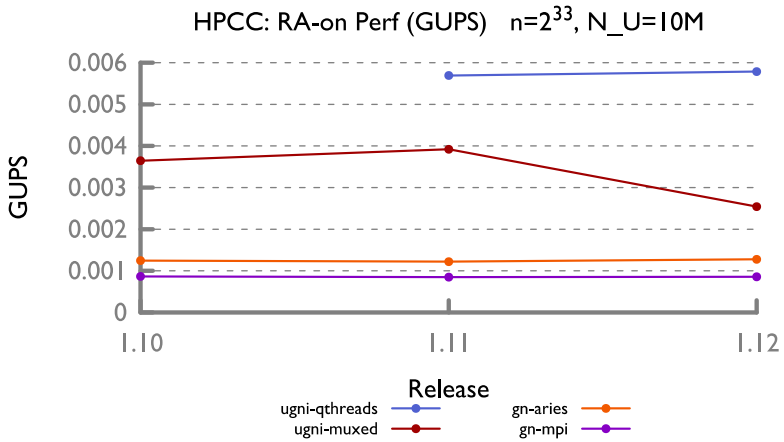
- **Saw ugni+qthreads becoming more competitive**
  - Decided to make it our default instead of ugni+muxed
    - puts us one step closer to retiring muxed

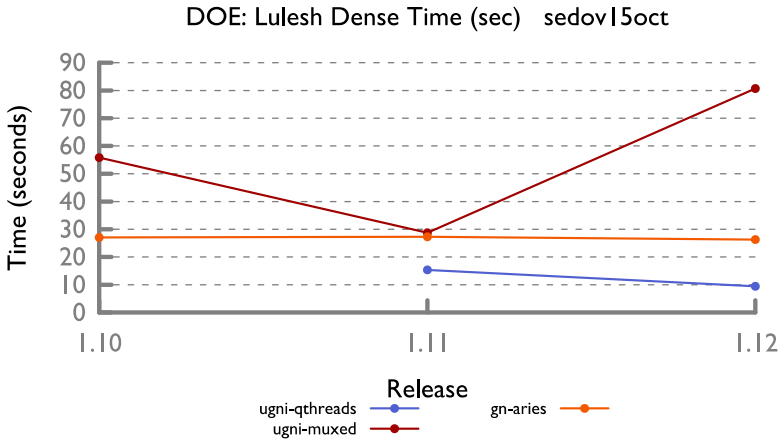


# ugni+qthreads as default

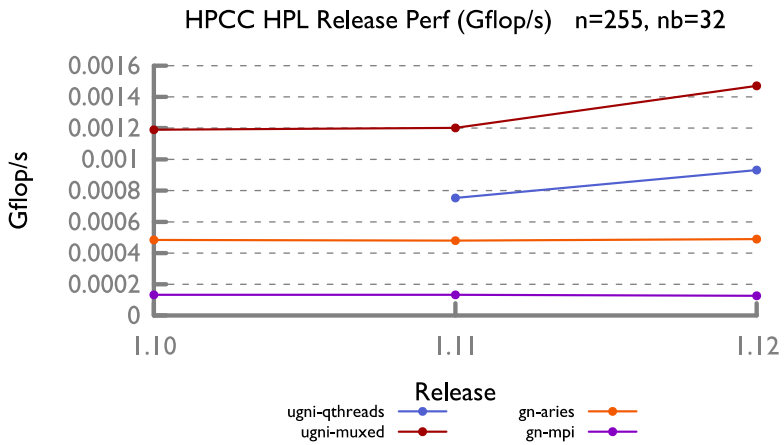
- **As a result, we left some muxed regressions unresolved**
  - For cases where qthreads was already outperforming muxed

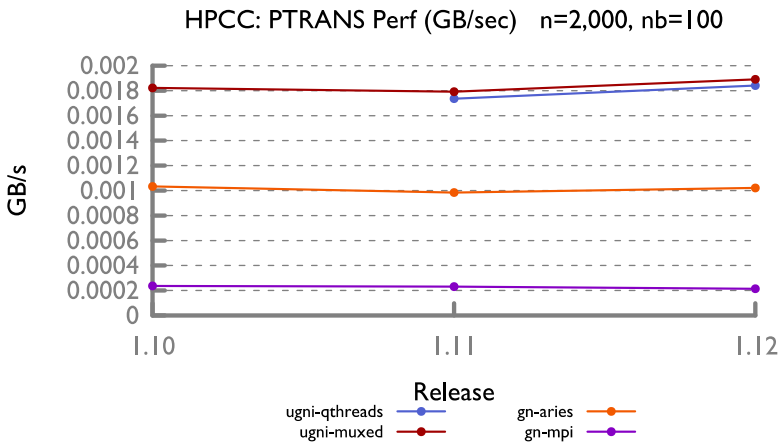


# ugni+qthreads as default

- **Still some cases where ugni+muxed outperforms**
  - Mostly for benchmarks we have not examined very closely yet
  - Will work on these remaining cases before retiring muxed



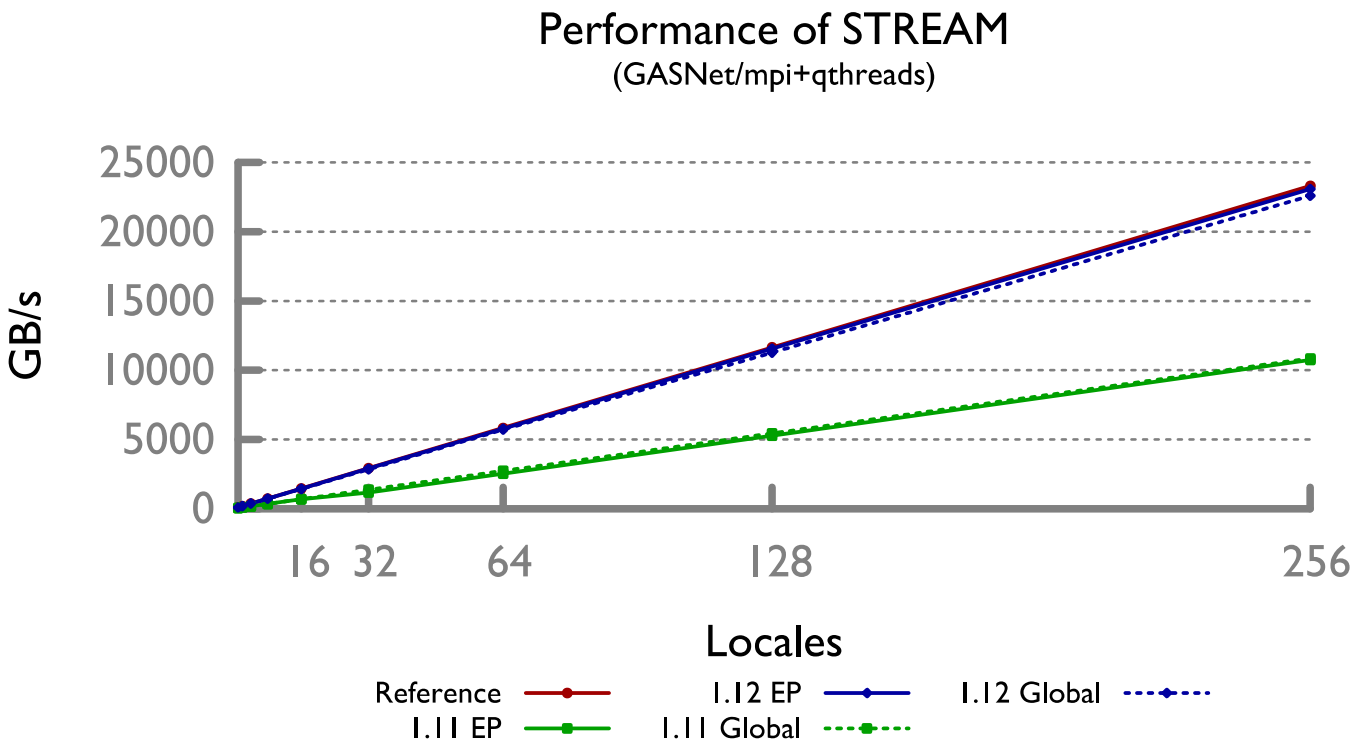
# Performance Scalability Study

# Scalability Study: Background

- **We continued the scalability study from the last 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 will highlight a few notable cases**
  - Stream improved dramatically as seen in previous slides decks
  - RA generally improved for qthreads, but got worse for muxed
    - 1.12 qthreads is close to 1.11 muxed performance
    - muxed declined due to using more tasks as described in optimization slides
      - not very concerned about muxed regressions now that qthreads is the default
  - Reduction performance has not changed since last release
    - (graph omitted for this reason)

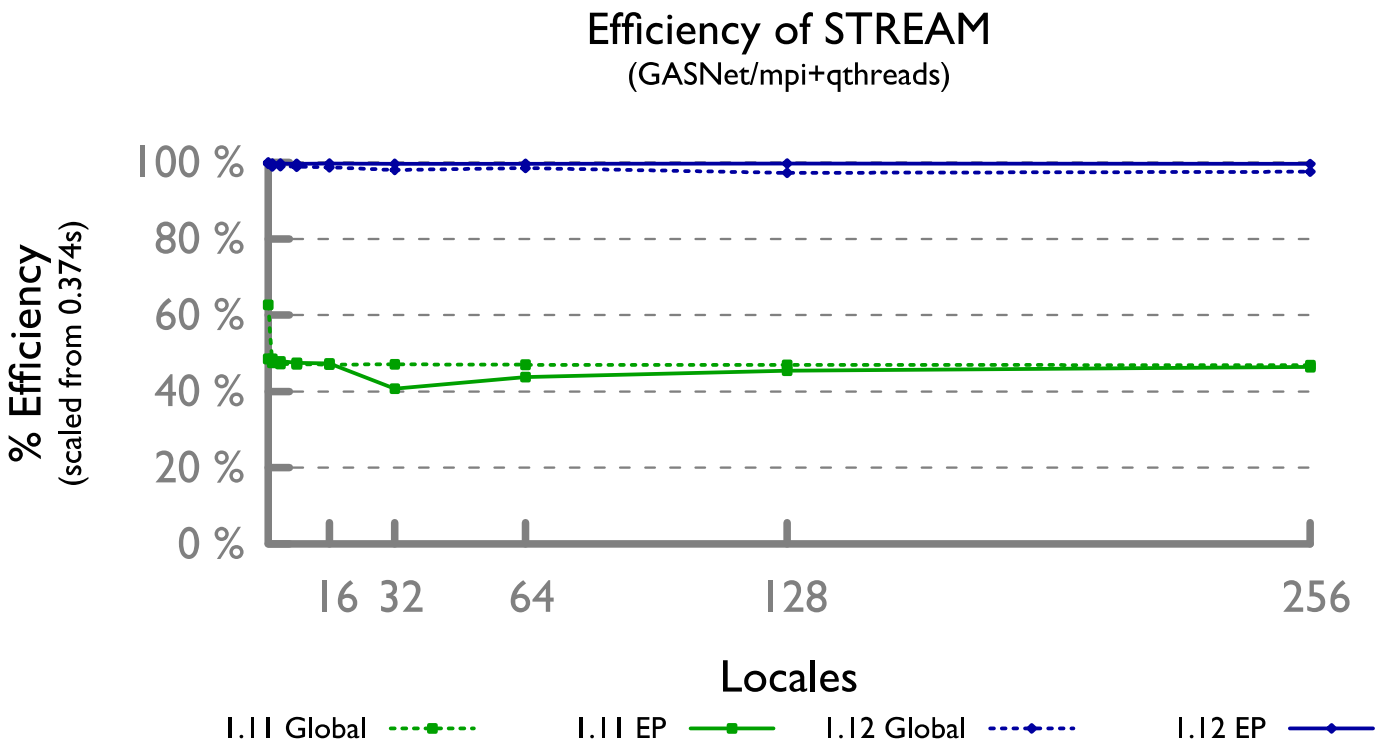
# Scalability: STREAM Performance

- **Stream performance more than doubled since last release**
  - EP is on par with the reference
  - Global is very competitive



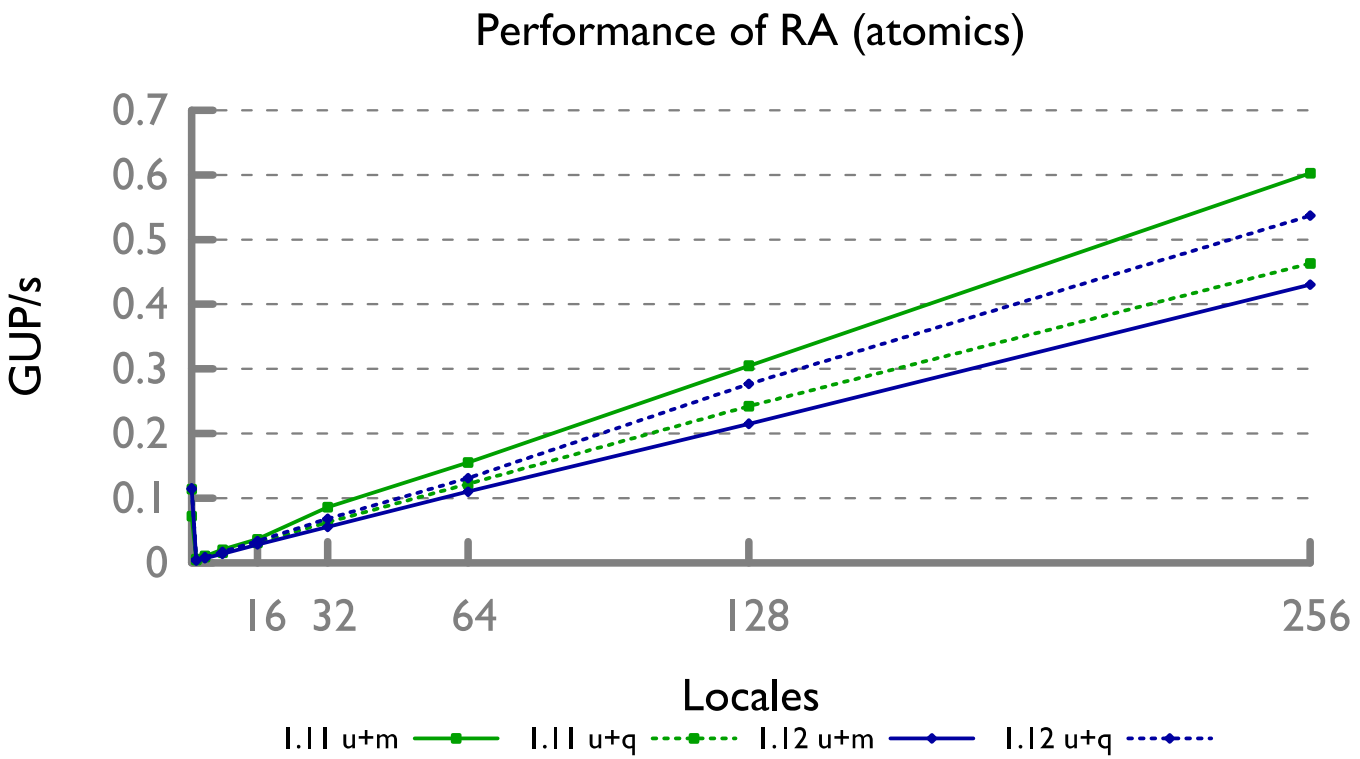
# Scalability: STREAM Efficiency

- **Stream efficiency more than doubled since last release**
  - EP efficiency is 100% at all locale counts
  - Global efficiency is 97% at 256 locales



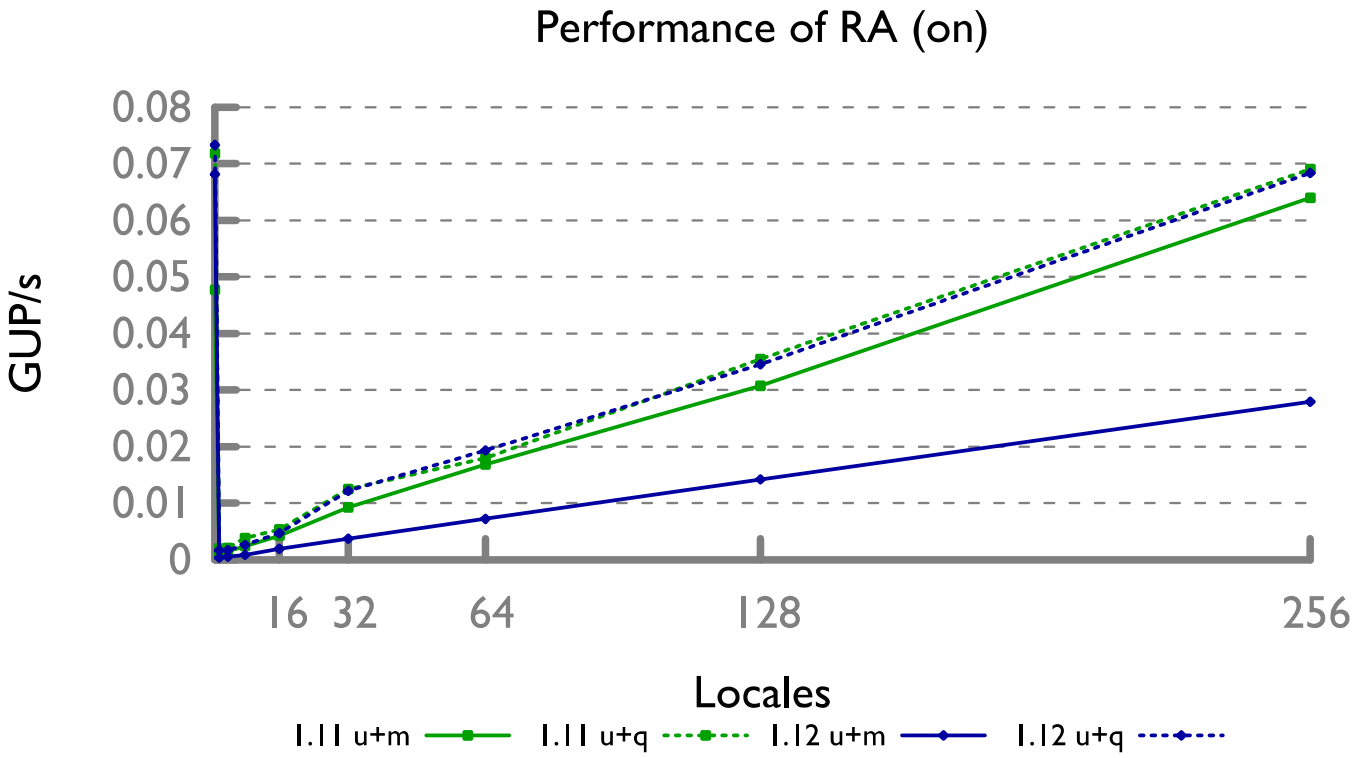
# Scalability: RA (atomics) Performance

- **RA (atomics) summary:**
  - qthreads performance improved (very close to 1.11 muxed)
  - muxed performance got worse



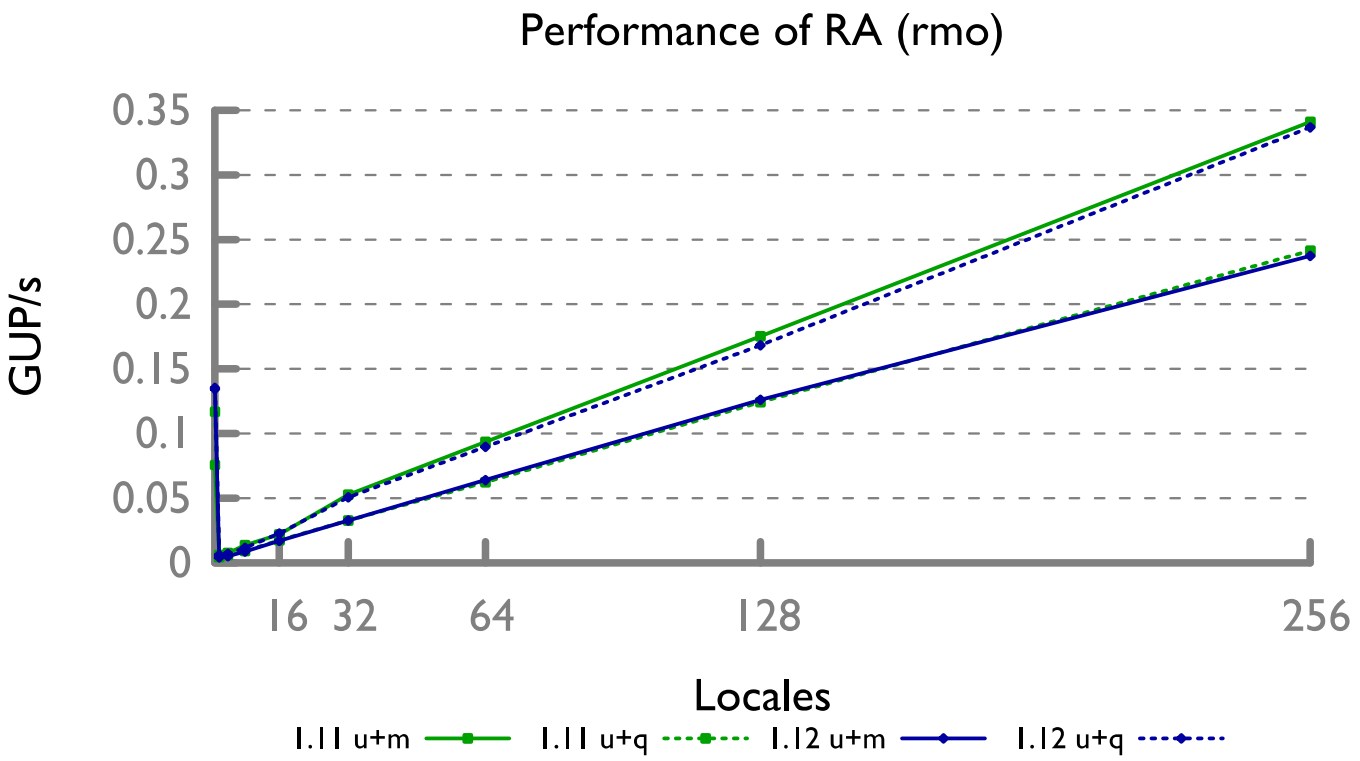
# Scalability: RA (on) Performance

- **RA (on) summary:**
  - qthreads performance remained the same (better than muxed)
  - muxed performance got worse



# Scalability: RA (rmo) Performance

- **RA (rmo) summary:**
  - qthreads performance improved (on par with 1.11 muxed)
  - muxed performance got worse



# 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**
  - more focus on NUMA locale model
    - particularly address representation at execution time
  - 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
- **Continue scalability studies**
  - Reduce unnecessary communication code
  - Improve implementation of reductions

# 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, URIKA, and YARCDATA. 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.*


CRAY
THE SUPERCOMPUTER COMPANY

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