Performance Results

Chapel Team, Cray Inc.

Chapel version 1.18

September 20, 2018

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.

Summary

- **Generally speaking, performance has improved with 1.18**

- **Previous slides have shown performance improvements:**

  …due to communication reductions
  …due to compiler and library optimizations
  …due to runtime optimizations

- **These slides contain additional 1.18 performance results**
  - not tied to any specific effort, just comparisons across releases

Outline

- Single-Locale Performance Trends
- Multi-Locale Performance Trends
- Scalability Trends
- Priorities and Next Steps

Single-Locale Performance Trends

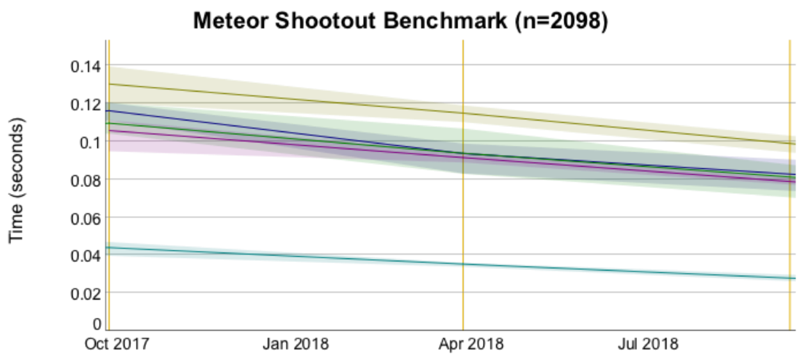
Single-Locale Performance Configuration

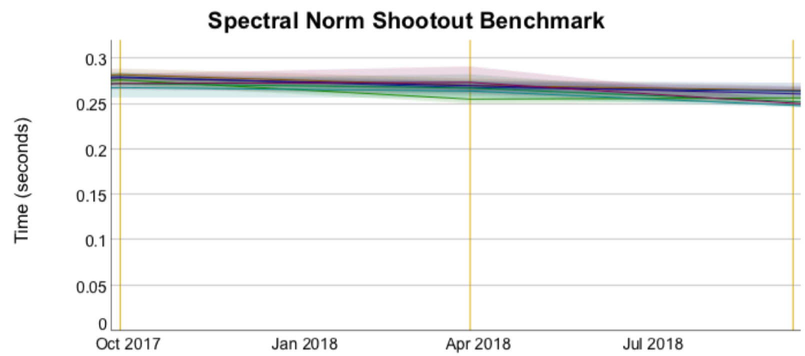
- **Hardware:**
  - 24-core, 128GB RAM
    - (2) 12-core "Haswell" 2.6 GHz processors

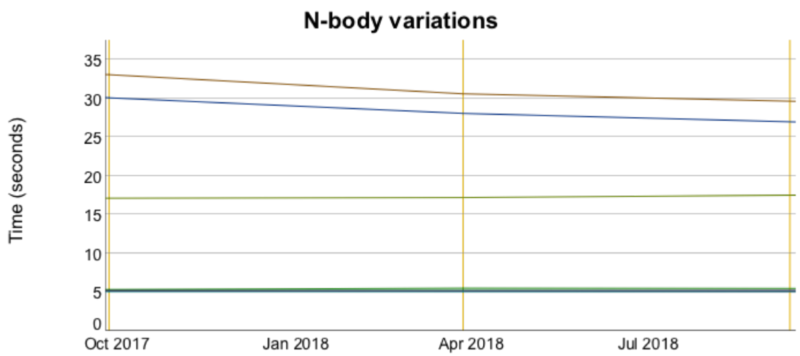
- **Software:**
  - SLES 12
  - GCC 6.3
  - Chapel 1.16.0, 1.17.1, 1.18.0

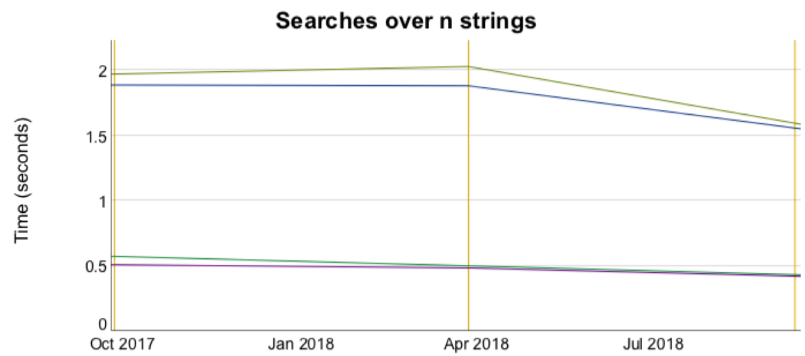
Single-Locale Performance

- **No major single-locale performance changes**
  - Minor improvements for a handful of benchmarks



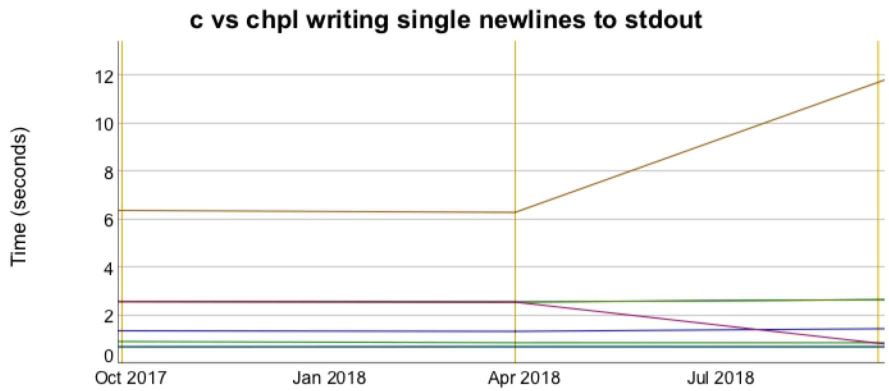






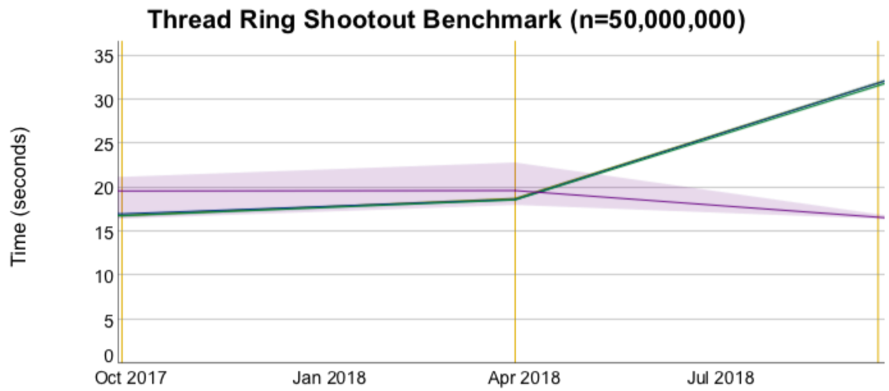
Single-Locale Performance

- **Some known/expected single-locale regressions**
  - Serial I/O sacrificed for better Parallel I/O
    - plan to address in later release with a hybrid lock for I/O
    - users can opt into unlocked I/O if needed



Single-Locale Performance

- **Some known/expected single-locale regressions**
  - Producer/consumer style sync variable performance hurt
    - but significantly improved performance for sync-based locks/barriers
    - working with qthreads team for a better solution



Multi-Locale Performance Trends

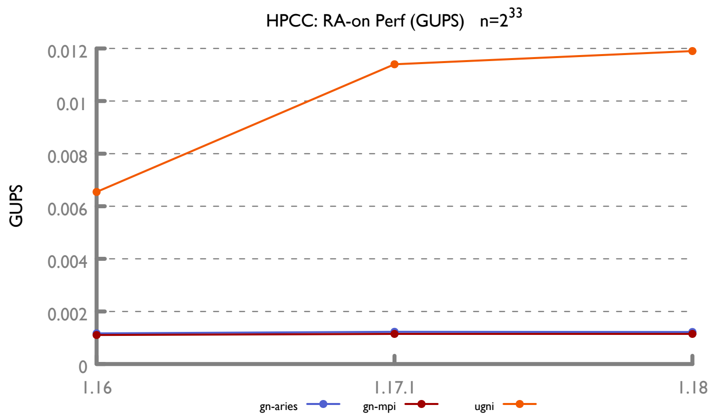
Multi-Locale Performance Configuration

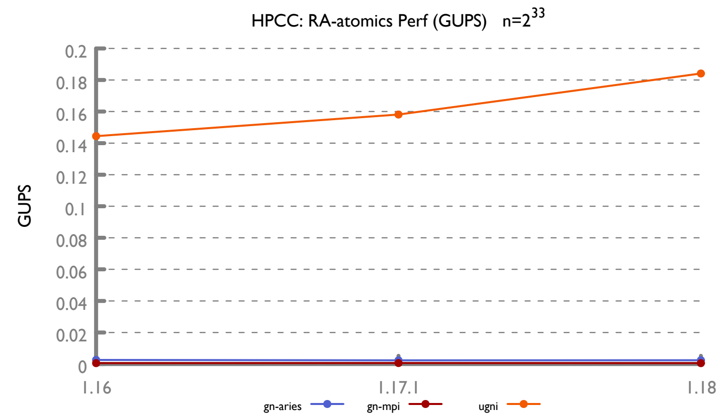
- **Hardware: 16 nodes of a Cray XC**
  - 28-core, 128GB RAM
    - (2) 14-core "Broadwell" 2.6 GHz processors

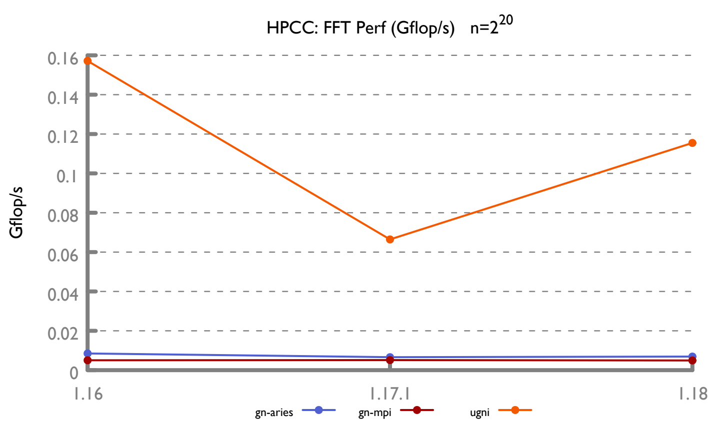
- **Software:**
  - CLE6
  - GCC 6.3
  - Chapel 1.16.0, 1.17.1, 1.18.0

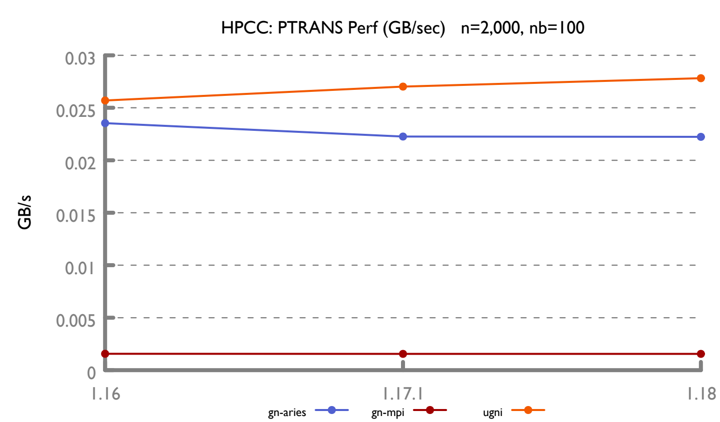
Multi-Locale Performance

- **Modest multi-locale performance improvements**
  - Performance graphs (up is better)



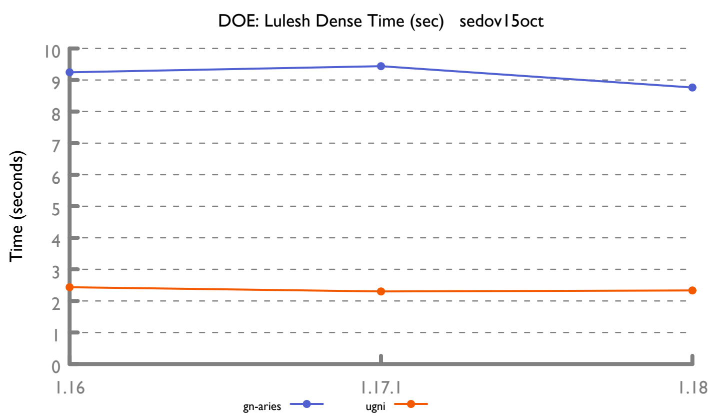


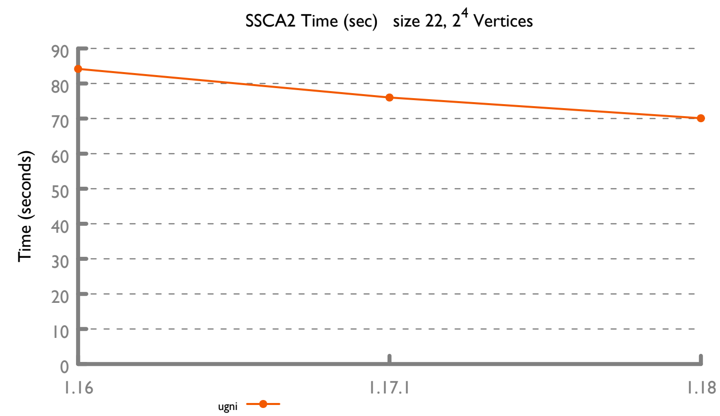


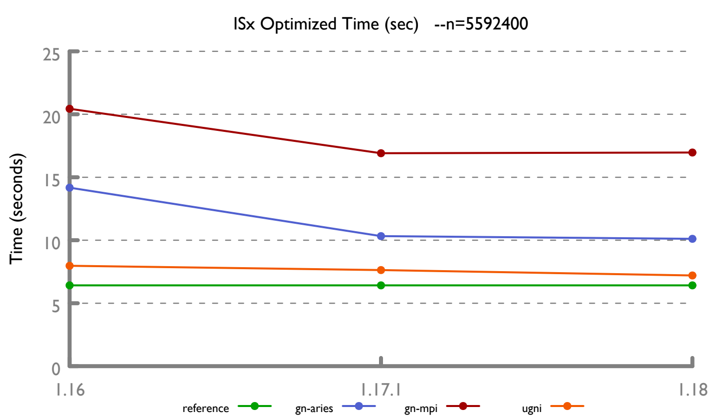


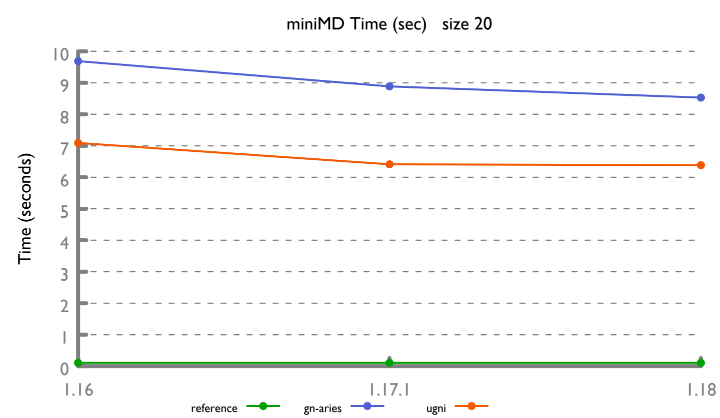
Multi-Locale Performance

- **Modest multi-locale performance improvements**
  - Time graphs (down is better)









Scalability Trends

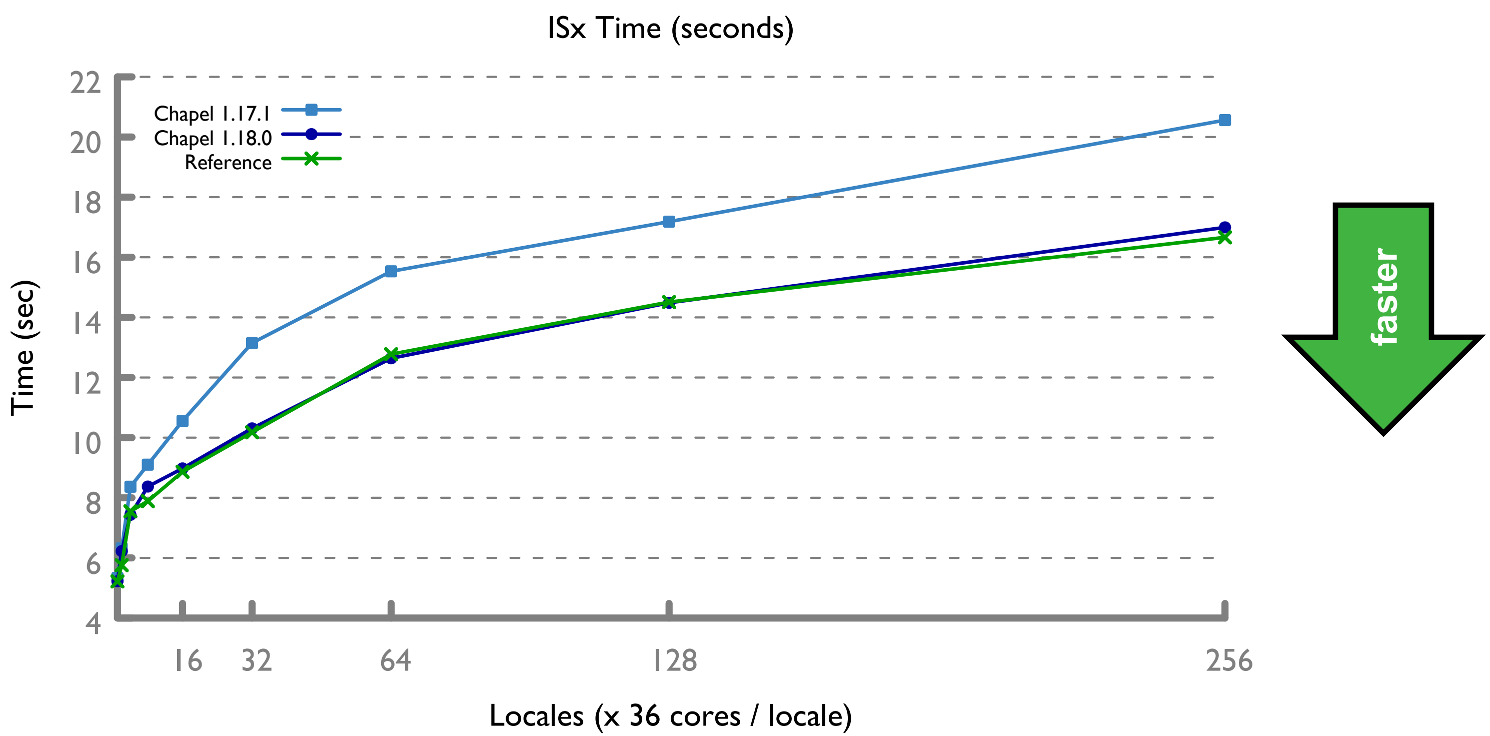
Scalability Configuration

- **Hardware: Up to 256 nodes of a Cray XC**
  - 36-core, 128 GB RAM
    - (2) 18-core "Broadwell" 2.1 GHz processors

- **Software:**
  - CLE6
  - GCC 6.3
  - Cray mpich/shmem 7.6.2
  - Chapel 1.17.1, 1.18.0

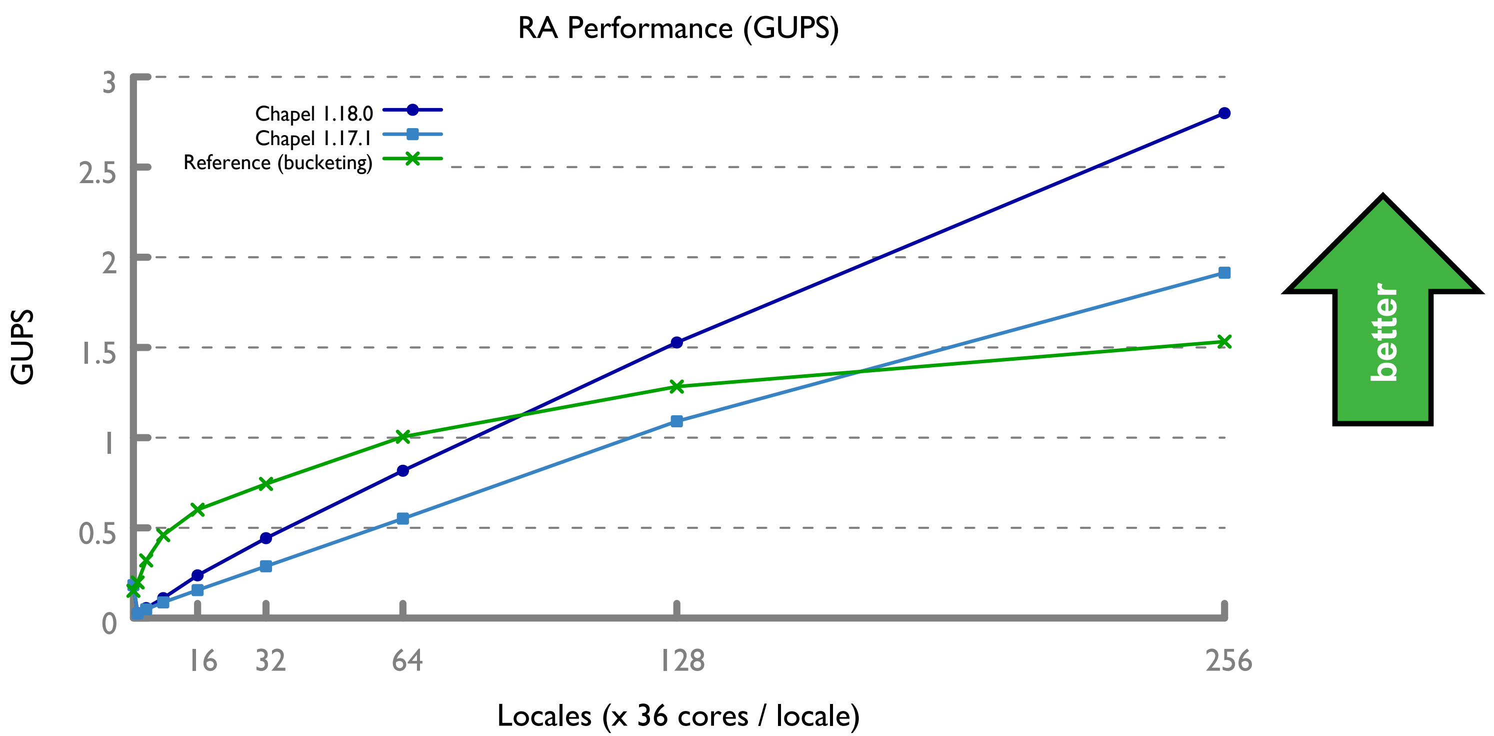
Scalability

- **Significant scalability improvements**
  - ~25% improvement for ISx, now on par with reference



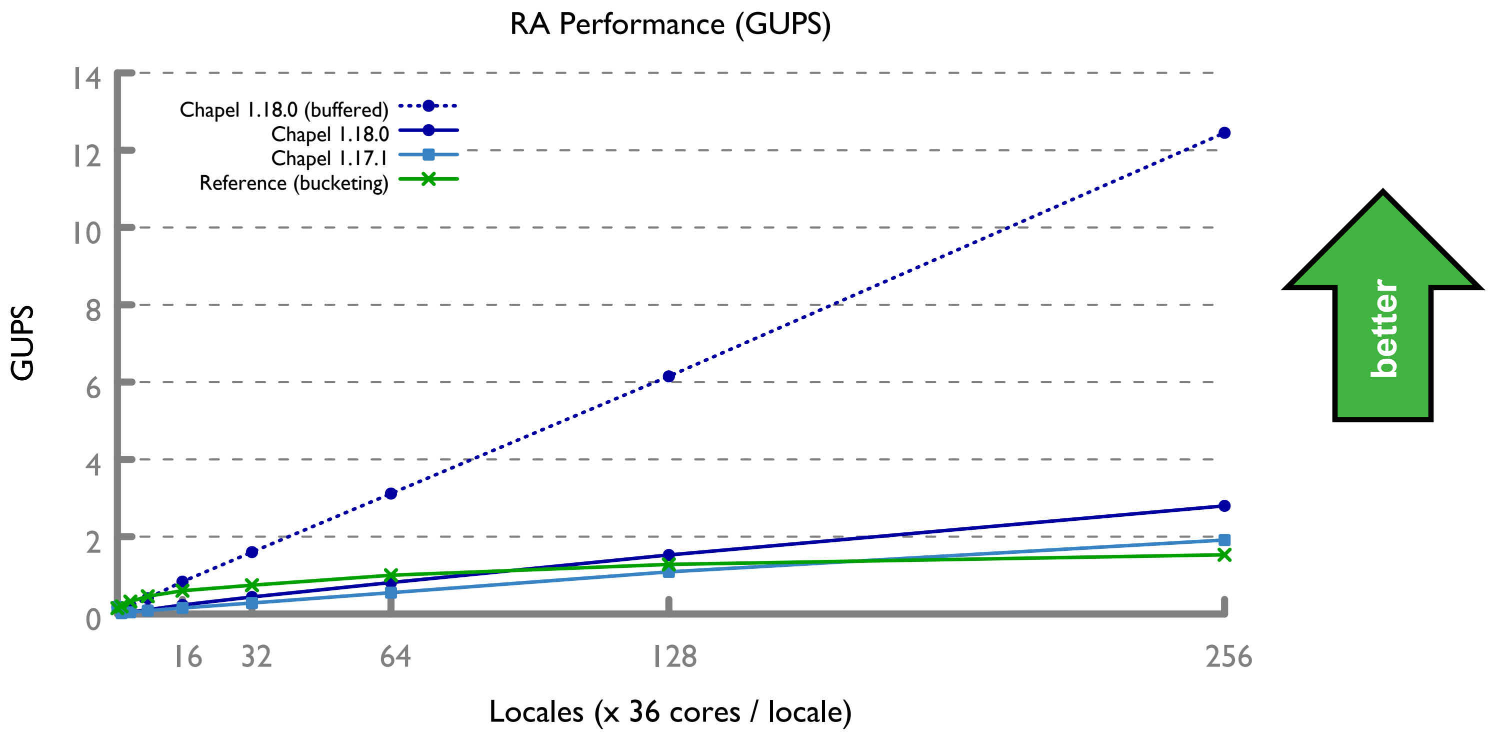
Scalability

- **Significant scalability improvements**
  - ~45% improvement for RA-atomics



Scalability

- **Significant scalability improvements**
  - 6x improvement for RA-atomics with buffered atomics



Performance Priorities and Next Steps

Performance Priorities and Next Steps

- **Continue benchmark-driven improvements**
  - Scalability:
    - add support for unordered GETs/PUTs
    - add a bulk-spawning mechanism for more scalable task-spawning
    - run scalability tests at higher scales
  - Multi-locale:
    - focus on user applications, Bale, PRKs, and DOE proxy apps
    - reduce unnecessary communication
  - Single-locale:
    - improve performance for shootouts (requires better vectorization)

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.

CRAY CHAPEL

CRAY

THE SUPERCOMPUTER COMPANY