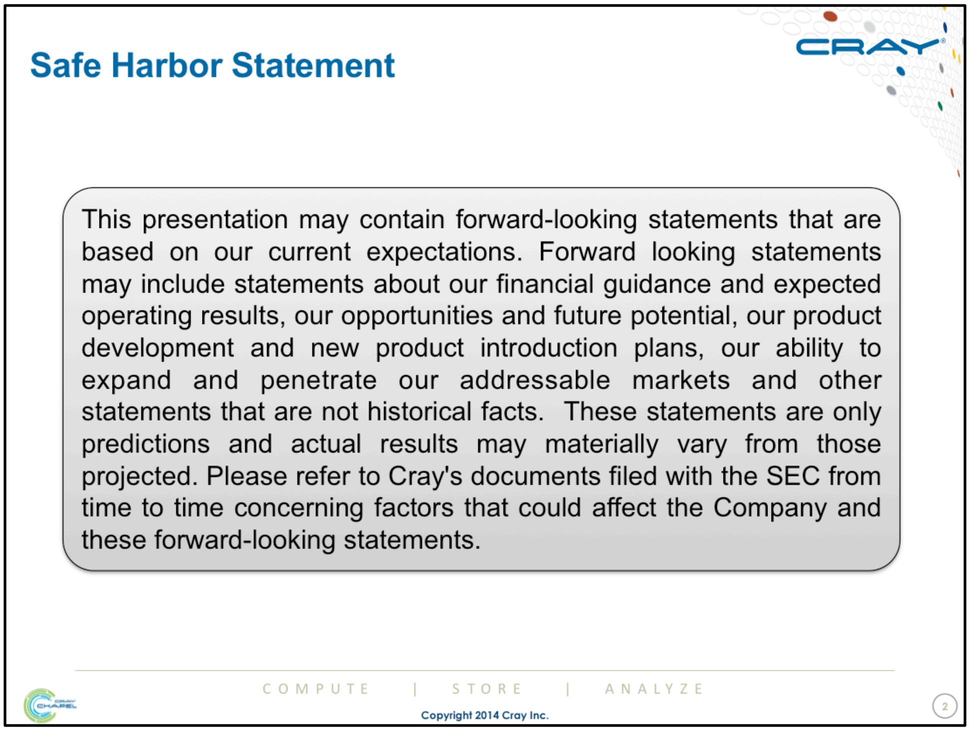
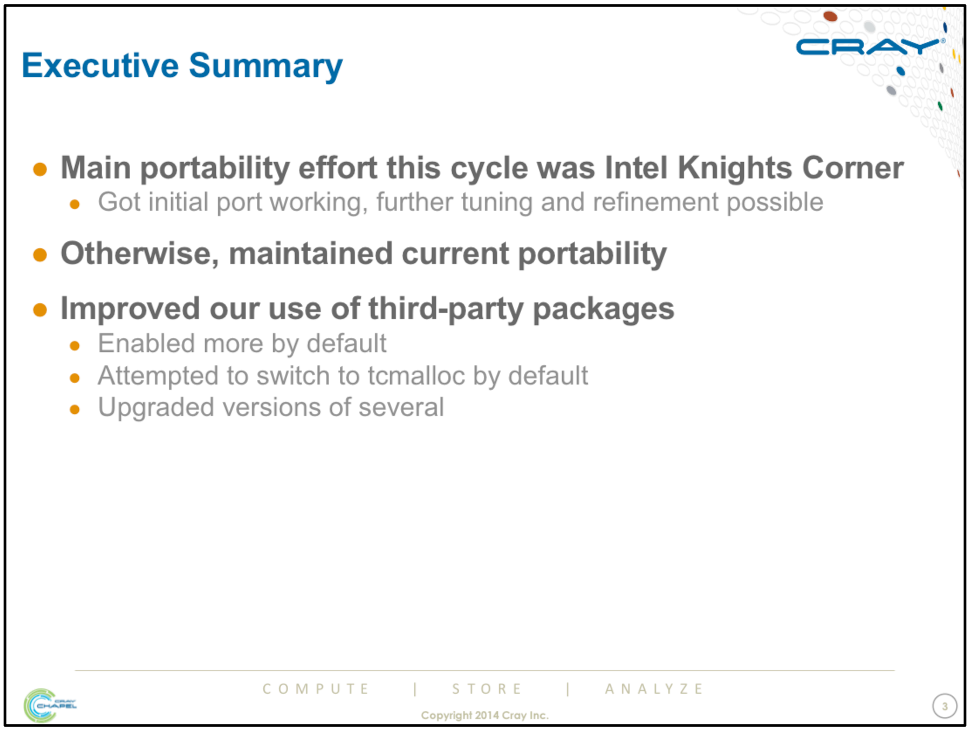
Portability and Third-Party Improvements

Chapel Team, Cray Inc.
Chapel version 1.10
October 2nd, 2014



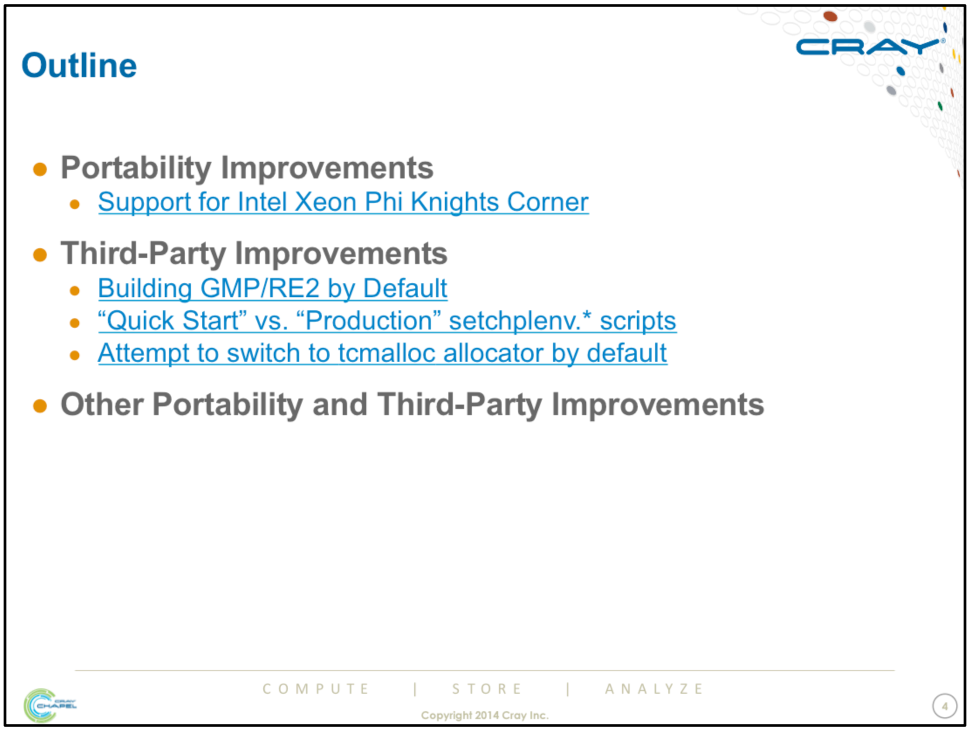
Safe Harbor Statement

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.



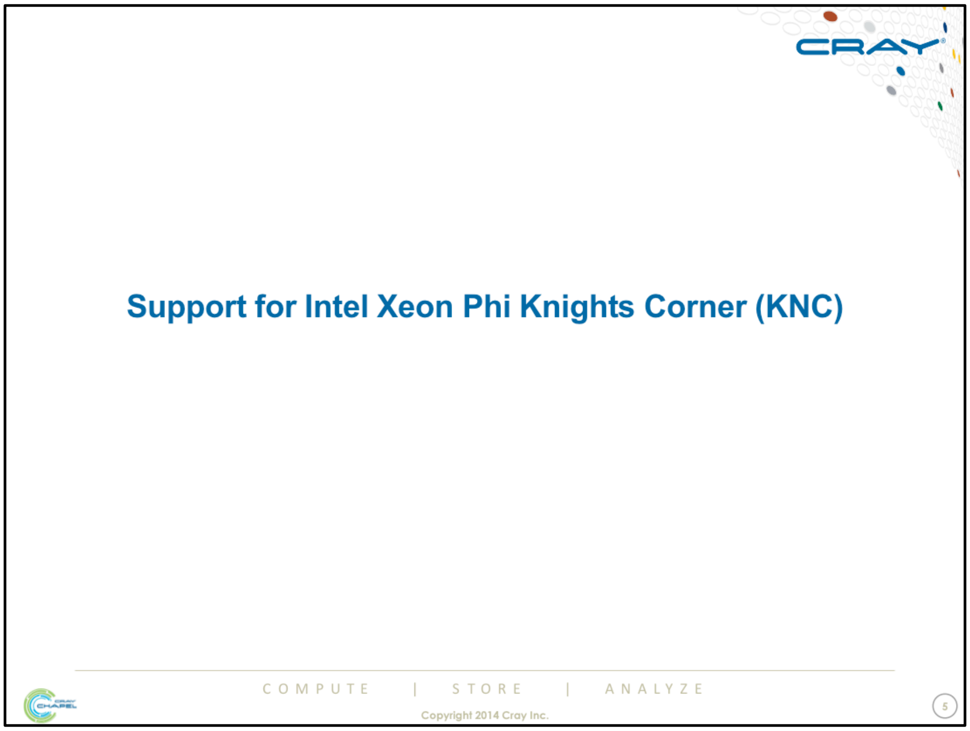
Executive Summary

- **Main portability effort this cycle was Intel Knights Corner**
  - Got initial port working, further tuning and refinement possible
- **Otherwise, maintained current portability**
- **Improved our use of third-party packages**
  - Enabled more by default
  - Attempted to switch to tcmalloc by default
  - Upgraded versions of several

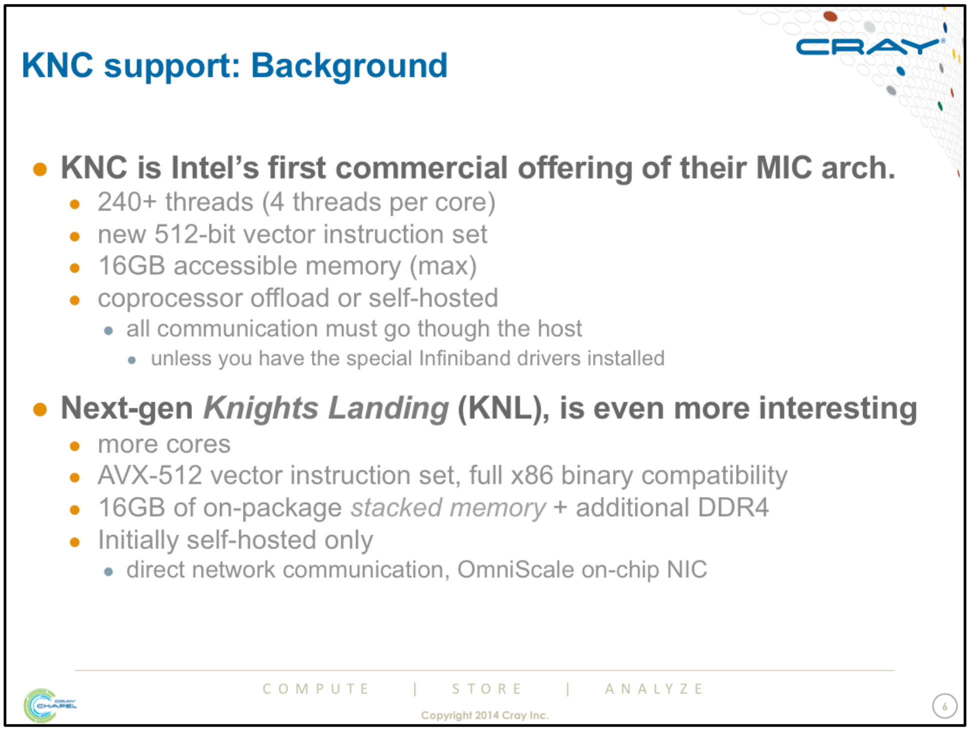


Outline

- **Portability Improvements**
  - Support for Intel Xeon Phi Knights Corner
- **Third-Party Improvements**
  - Building GMP/RE2 by Default
  - “Quick Start” vs. “Production” setchplenv.* scripts
  - Attempt to switch to tcmalloc allocator by default
- **Other Portability and Third-Party Improvements**

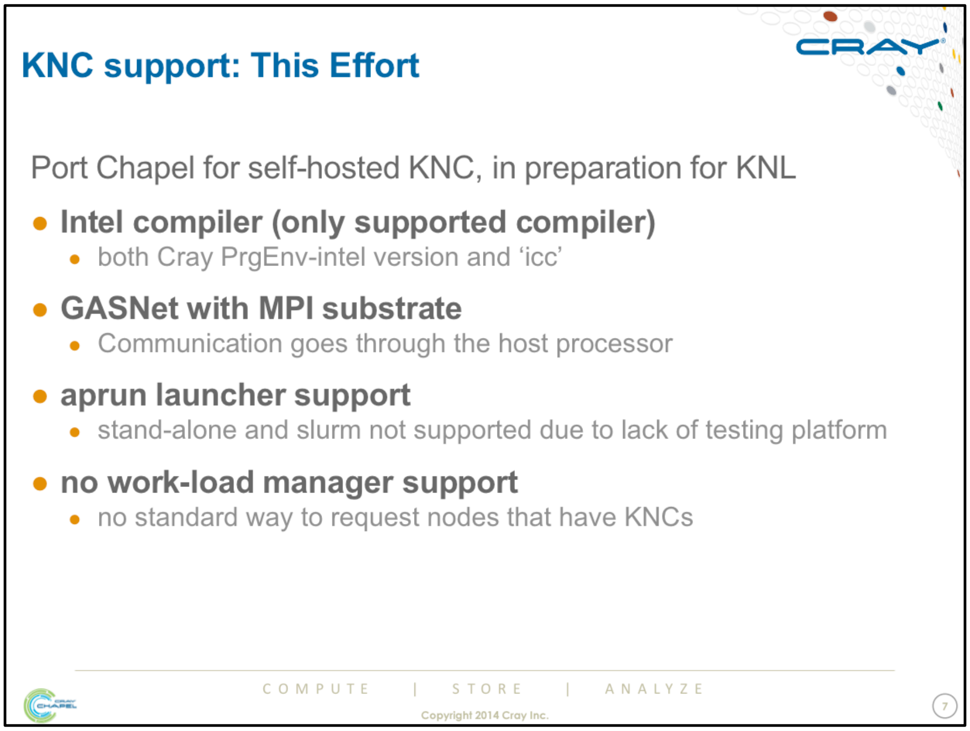


Support for Intel Xeon Phi Knights Corner (KNC)



KNC support: Background

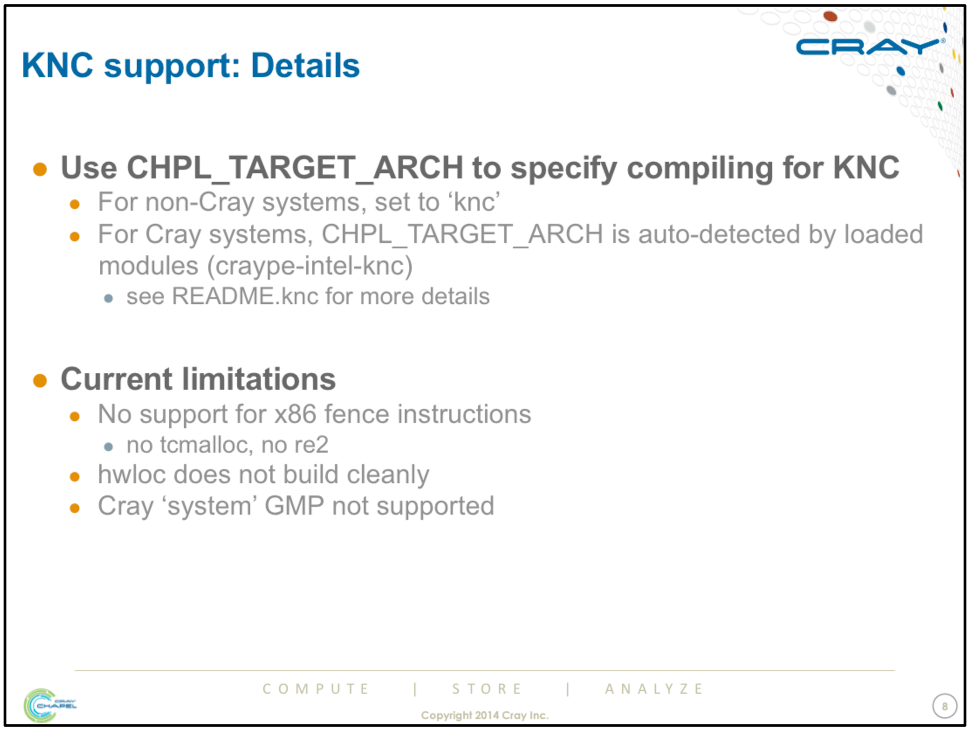
- **KNC is Intel's first commercial offering of their MIC arch.**
  - 240+ threads (4 threads per core)
  - new 512-bit vector instruction set
  - 16GB accessible memory (max)
  - coprocessor offload or self-hosted
    - all communication must go though the host
      - unless you have the special Infiniband drivers installed
- **Next-gen *Knights Landing* (KNL), is even more interesting**
  - more cores
  - AVX-512 vector instruction set, full x86 binary compatibility
  - 16GB of on-package *stacked memory* + additional DDR4
  - Initially self-hosted only
    - direct network communication, OmniScale on-chip NIC



KNC support: This Effort

Port Chapel for self-hosted KNC, in preparation for KNL

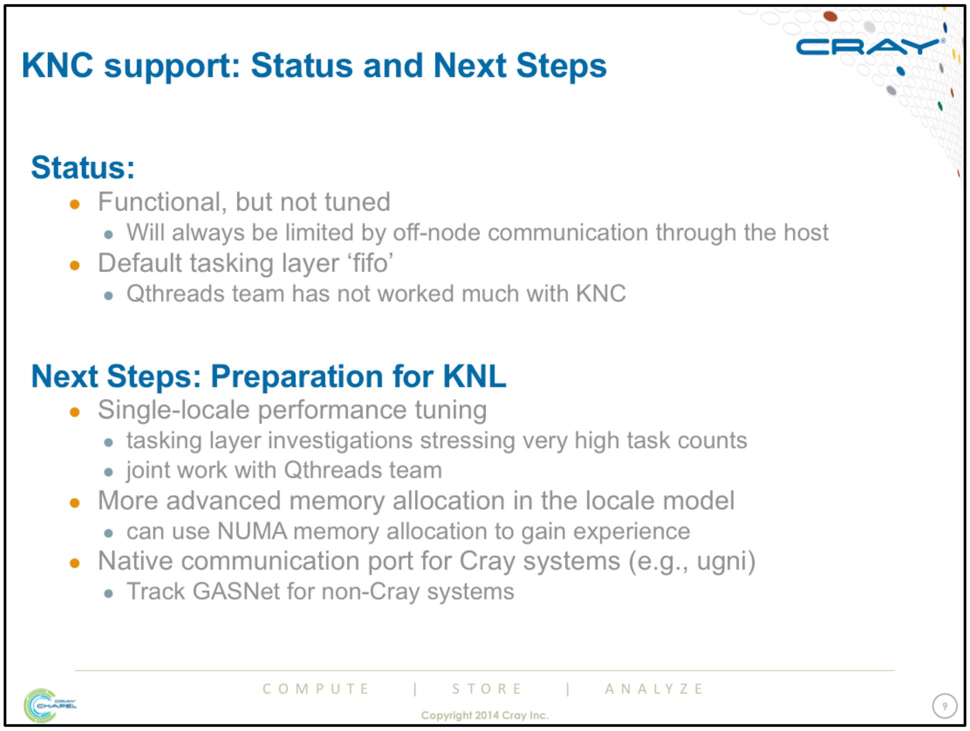
- **Intel compiler (only supported compiler)**
  - both Cray PrgEnv-intel version and 'icc'
- **GASNet with MPI substrate**
  - Communication goes through the host processor
- **aprun launcher support**
  - stand-alone and slurm not supported due to lack of testing platform
- **no work-load manager support**
  - no standard way to request nodes that have KNCs



KNC support: Details

- **Use CHPL_TARGET_ARCH to specify compiling for KNC**
  - For non-Cray systems, set to ‘knc’
  - For Cray systems, CHPL_TARGET_ARCH is auto-detected by loaded modules (craype-intel-knc)
    - see README.knc for more details

- **Current limitations**
  - No support for x86 fence instructions
    - no tcmalloc, no re2
  - hwloc does not build cleanly
  - Cray ‘system’ GMP not supported



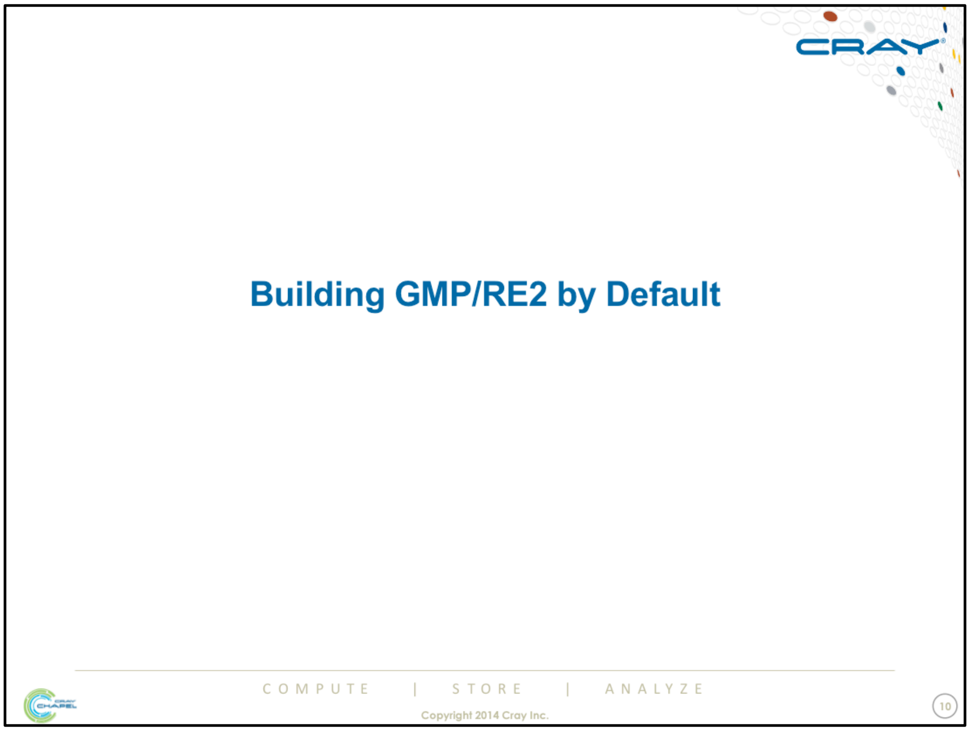
KNC support: Status and Next Steps

Status:

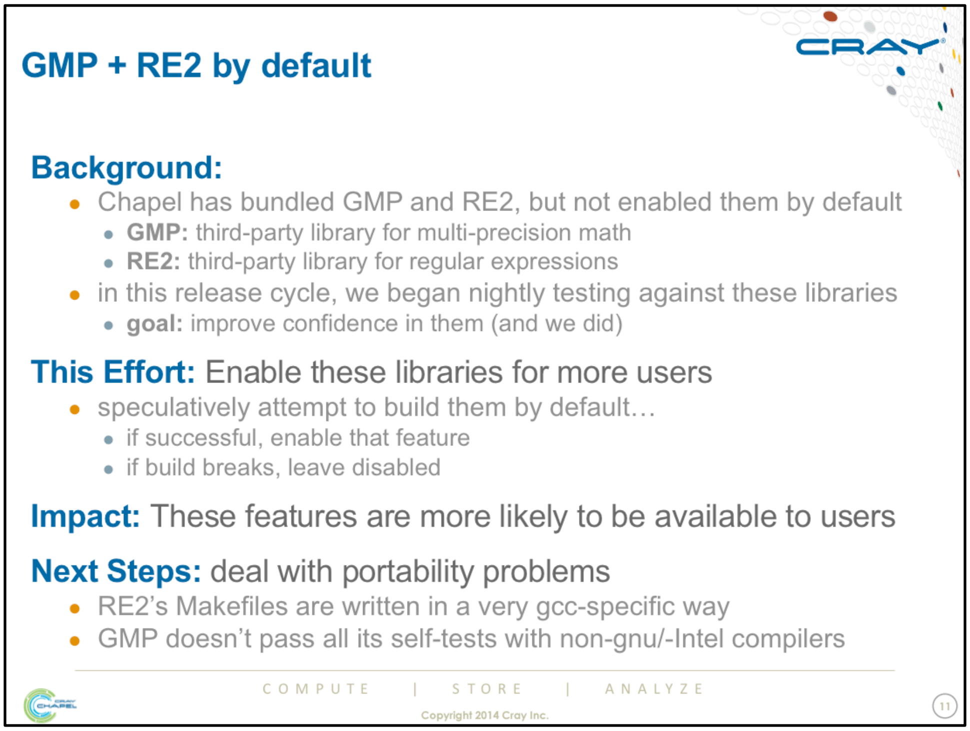
- Functional, but not tuned
  - Will always be limited by off-node communication through the host
- Default tasking layer ‘fifo’
  - Qthreads team has not worked much with KNC

Next Steps: Preparation for KNL

- Single-locale performance tuning
  - tasking layer investigations stressing very high task counts
  - joint work with Qthreads team
- More advanced memory allocation in the locale model
  - can use NUMA memory allocation to gain experience
- Native communication port for Cray systems (e.g., ugni)
  - Track GASNet for non-Cray systems



Building GMP/RE2 by Default



GMP + RE2 by default

Background:

- Chapel has bundled GMP and RE2, but not enabled them by default
  - **GMP:** third-party library for multi-precision math
  - **RE2:** third-party library for regular expressions
- in this release cycle, we began nightly testing against these libraries
  - **goal:** improve confidence in them (and we did)

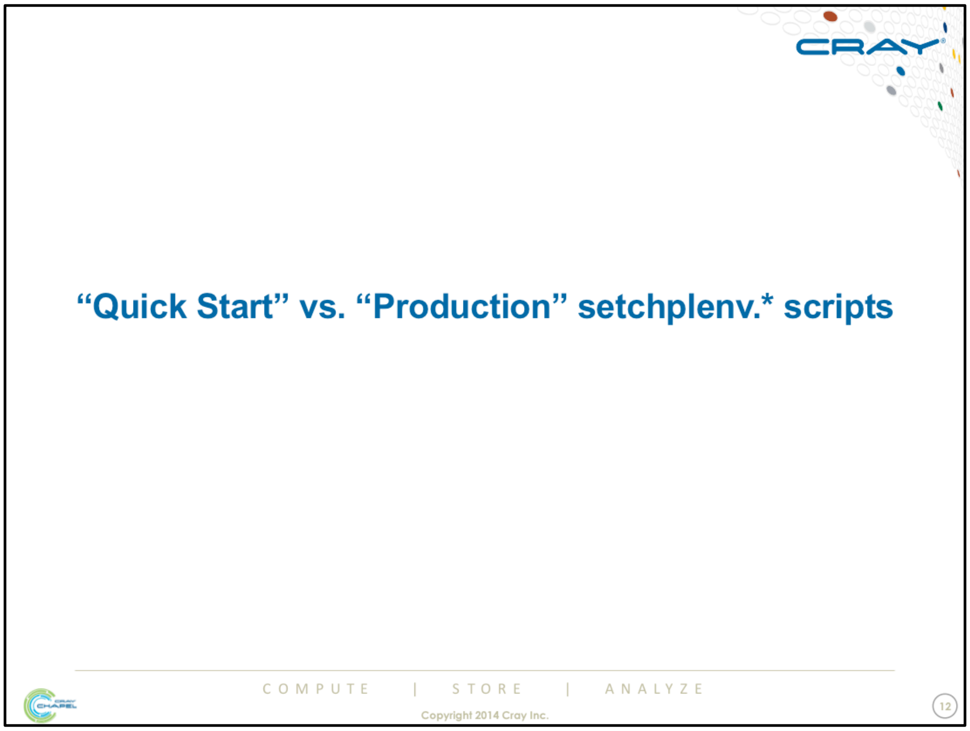
This Effort: Enable these libraries for more users

- speculatively attempt to build them by default…
  - if successful, enable that feature
  - if build breaks, leave disabled

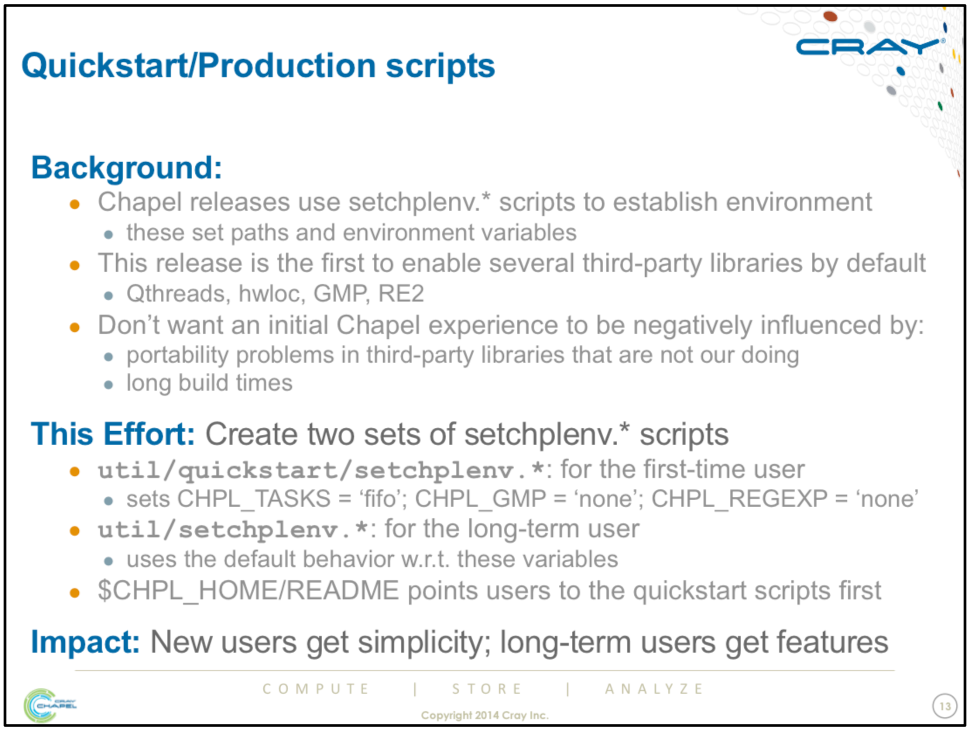
Impact: These features are more likely to be available to users

Next Steps: deal with portability problems

- RE2's Makefiles are written in a very gcc-specific way
- GMP doesn't pass all its self-tests with non-gnu/-Intel compilers



“Quick Start” vs. “Production” setchplenv.* scripts



Quickstart/Production scripts

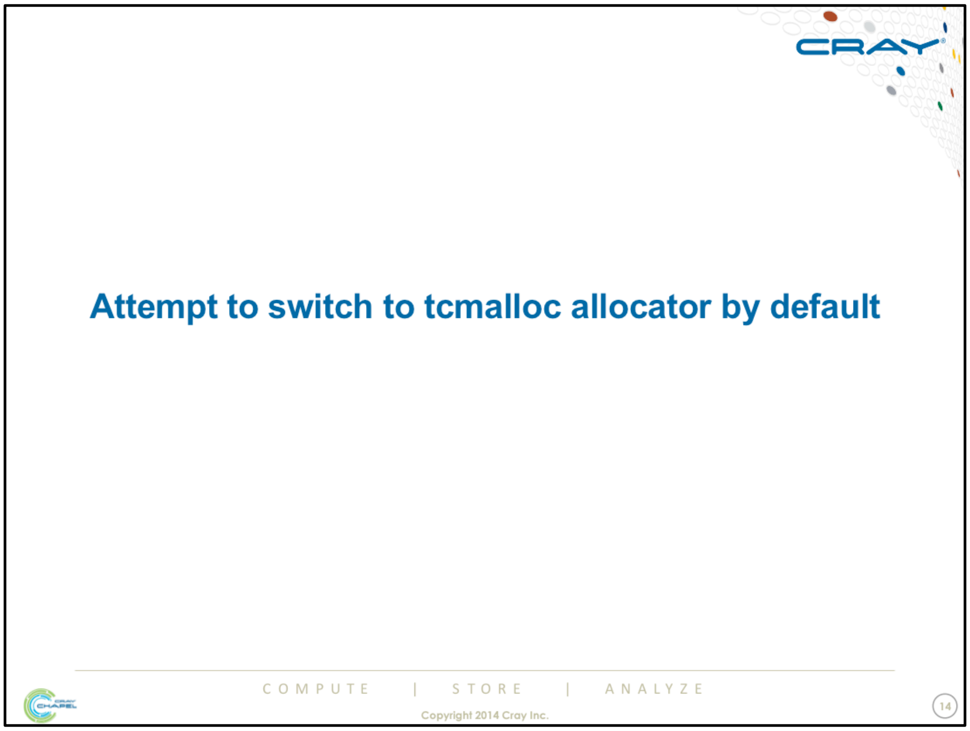
Background:

- Chapel releases use setchplenv.* scripts to establish environment
  - these set paths and environment variables
- This release is the first to enable several third-party libraries by default
  - Qthreads, hwloc, GMP, RE2
- Don’t want an initial Chapel experience to be negatively influenced by:
  - portability problems in third-party libraries that are not our doing
  - long build times

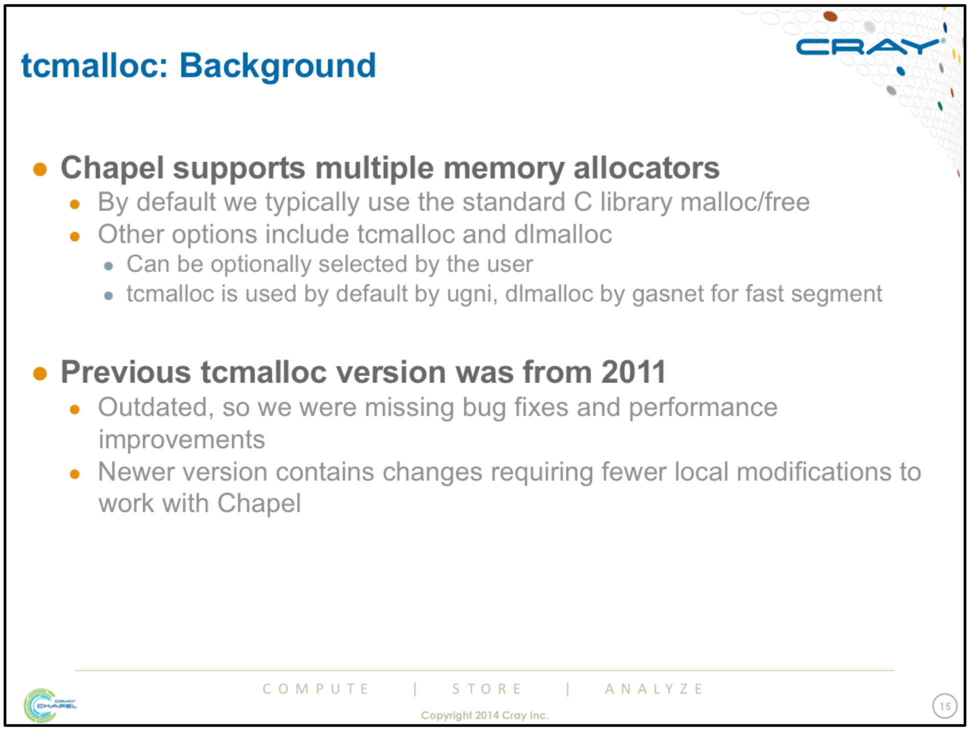
This Effort: Create two sets of setchplenv.* scripts

- `util/quickstart/setchplenv.*`: for the first-time user
  - sets CHPL_TASKS = ‘fifo’; CHPL_GMP = ‘none’; CHPL_REGEXP = ‘none’
- `util/setchplenv.*`: for the long-term user
  - uses the default behavior w.r.t. these variables
- $CHPL_HOME/README points users to the quickstart scripts first

Impact: New users get simplicity; long-term users get features

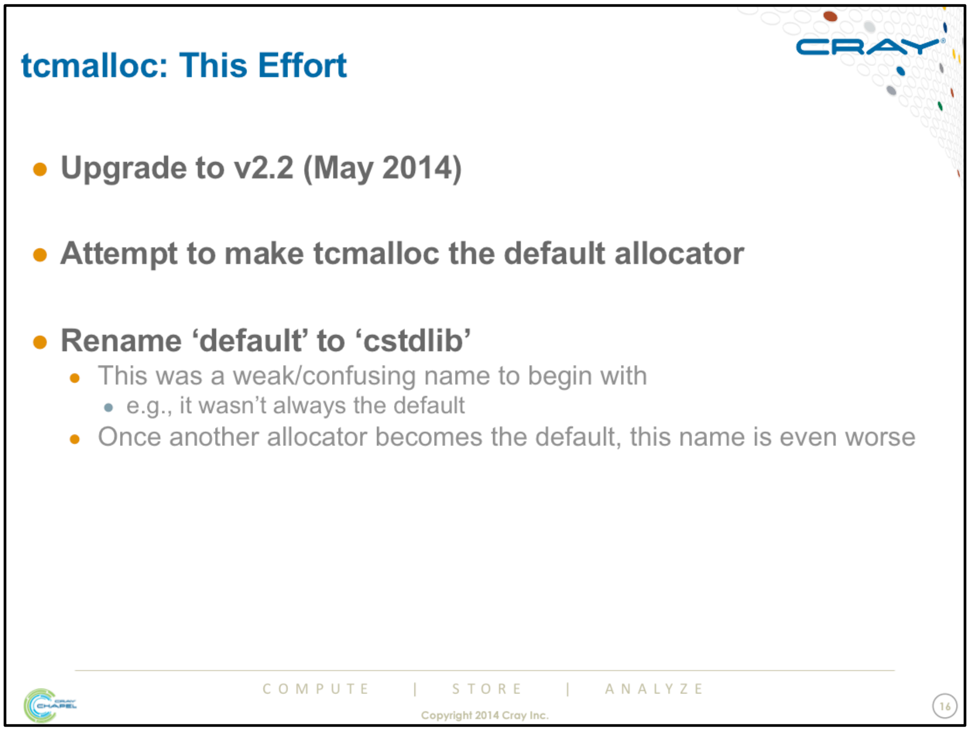


Attempt to switch to tcmalloc allocator by default



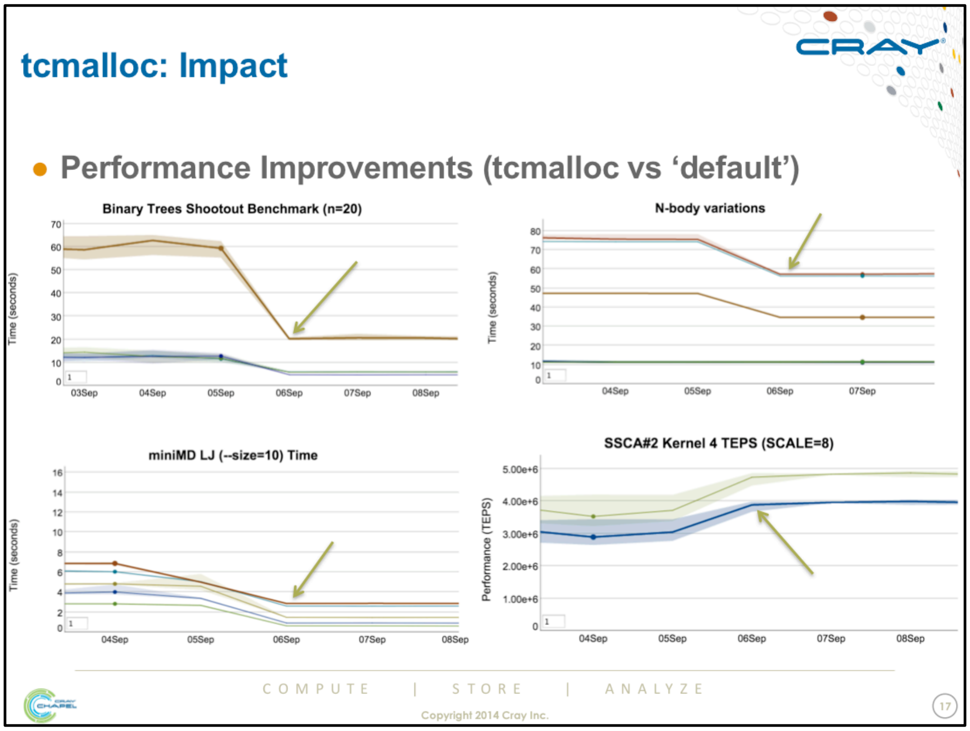
tcmalloc: Background

- **Chapel supports multiple memory allocators**
  - By default we typically use the standard C library malloc/free
  - Other options include tcmalloc and dlmalloc
    - Can be optionally selected by the user
    - tcmalloc is used by default by ugni, dlmalloc by gasnet for fast segment
- **Previous tcmalloc version was from 2011**
  - Outdated, so we were missing bug fixes and performance improvements
  - Newer version contains changes requiring fewer local modifications to work with Chapel



tcmalloc: This Effort

- **Upgrade to v2.2 (May 2014)**
- **Attempt to make tcmalloc the default allocator**
- **Rename ‘default’ to ‘cstdlib’**
  - This was a weak/confusing name to begin with
    - e.g., it wasn’t always the default
  - Once another allocator becomes the default, this name is even worse



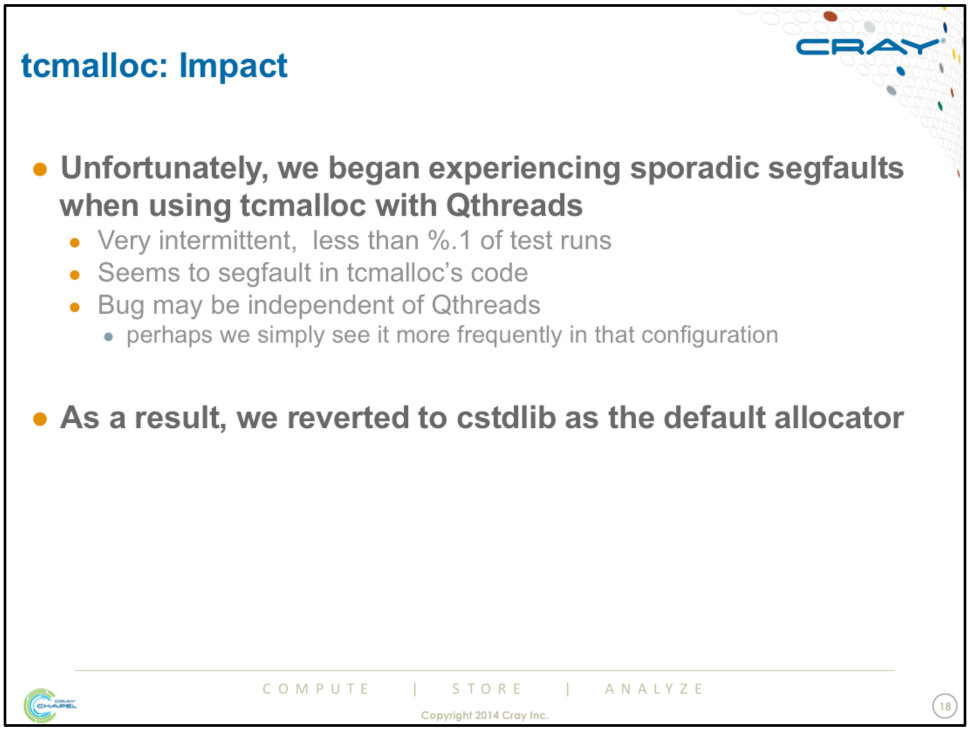
tcmalloc: Impact

- Performance Improvements (tcmalloc vs 'default')

COMPUTE | STORE | ANALYZE

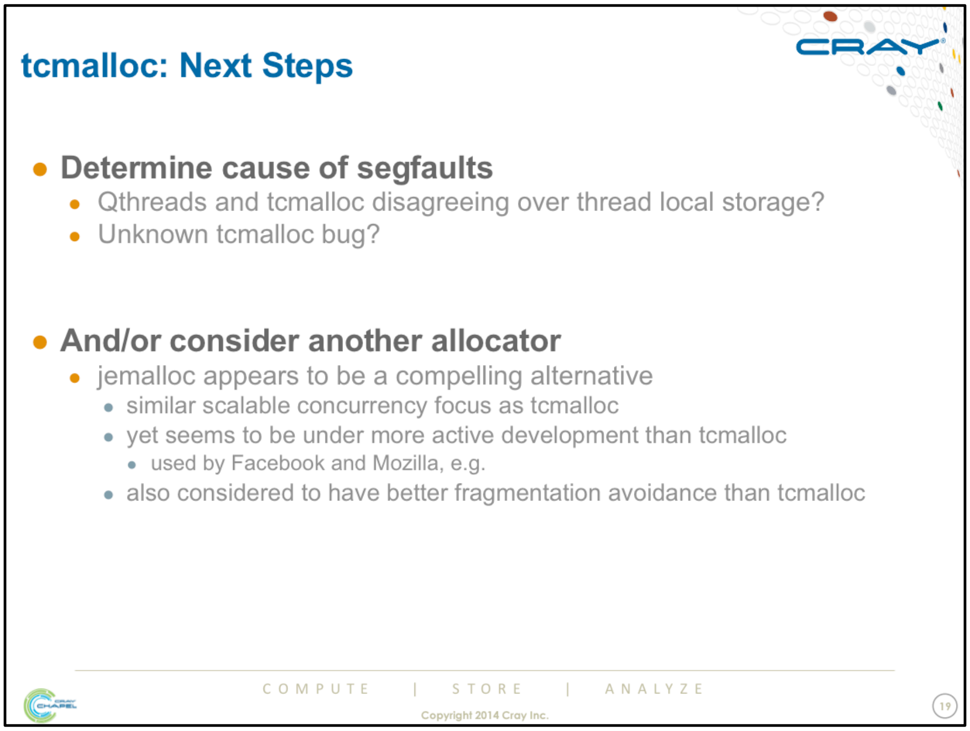
Arrows point to where tcmalloc was made the default

Note that for SSCA#2, higher is better



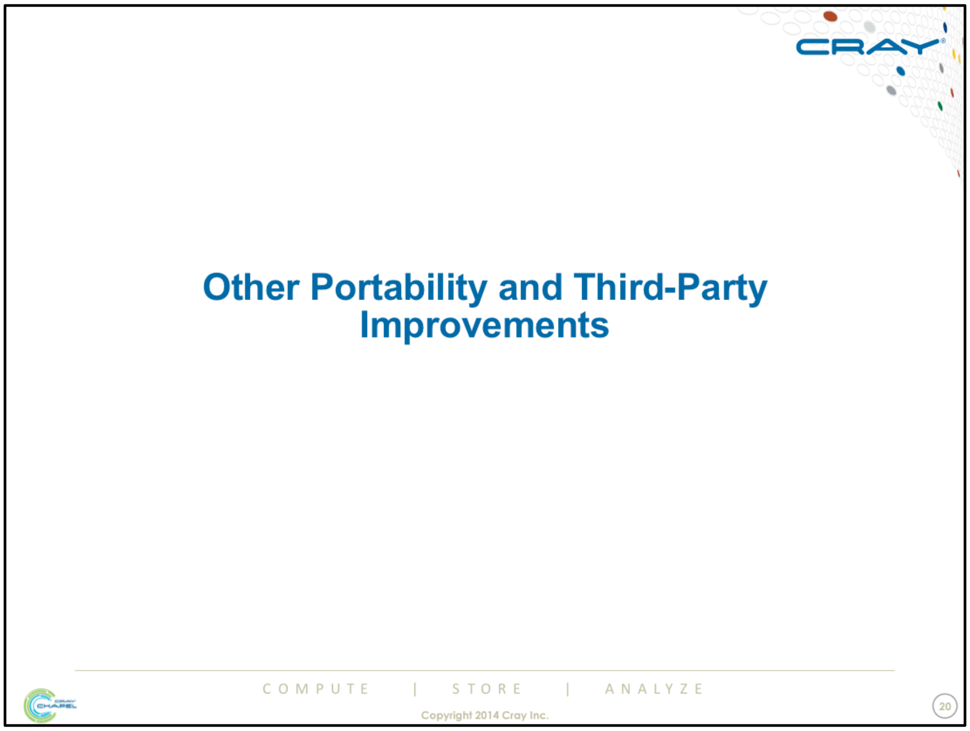
tcmalloc: Impact

- Unfortunately, we began experiencing sporadic segfaults when using tcmalloc with Qthreads
  - Very intermittent, less than %.1 of test runs
  - Seems to segfault in tcmalloc's code
  - Bug may be independent of Qthreads
    - perhaps we simply see it more frequently in that configuration
- As a result, we reverted to cstdlib as the default allocator

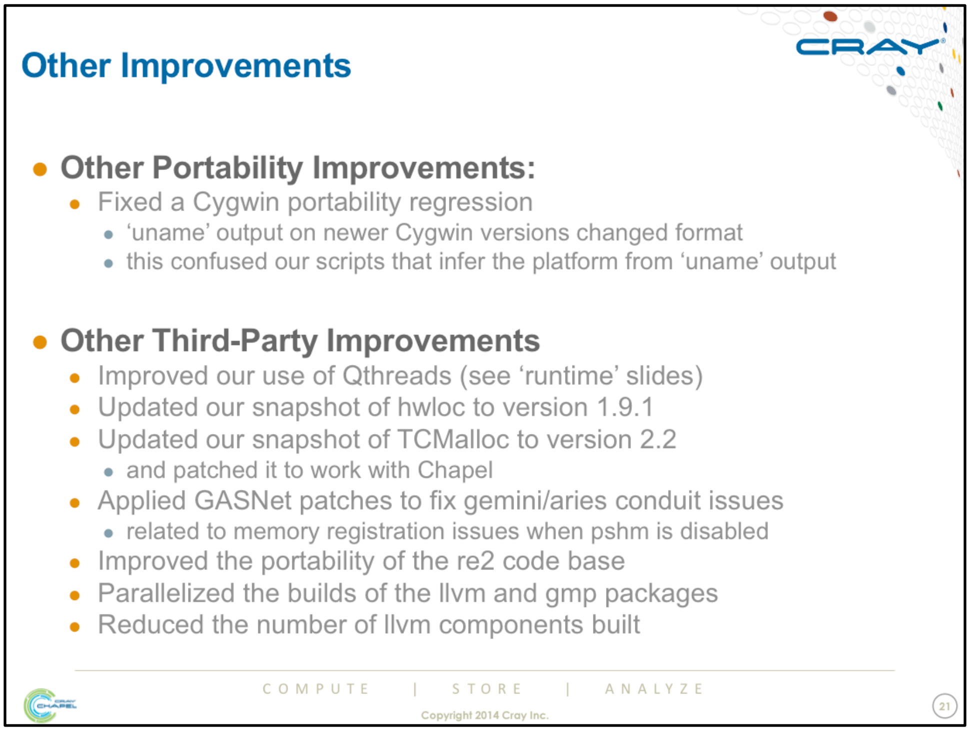


tcmalloc: Next Steps

- Determine cause of segfaults
  - Qthreads and tcmalloc disagreeing over thread local storage?
  - Unknown tcmalloc bug?
- And/or consider another allocator
  - jemalloc appears to be a compelling alternative
    - similar scalable concurrency focus as tcmalloc
    - yet seems to be under more active development than tcmalloc
      - used by Facebook and Mozilla, e.g.
    - also considered to have better fragmentation avoidance than tcmalloc



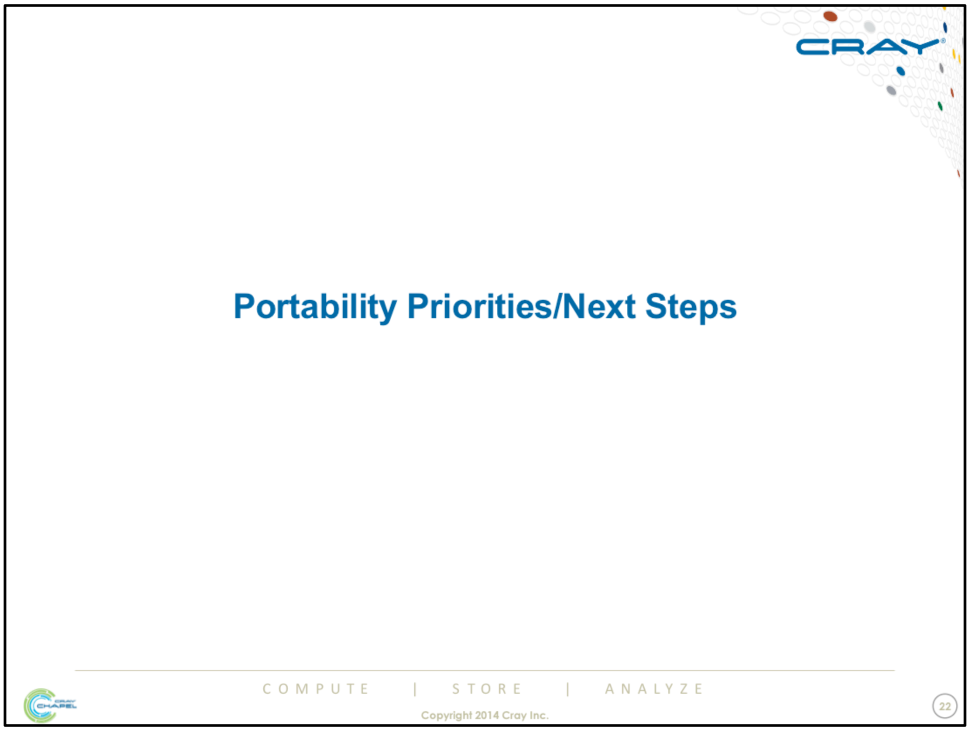
Other Portability and Third-Party Improvements



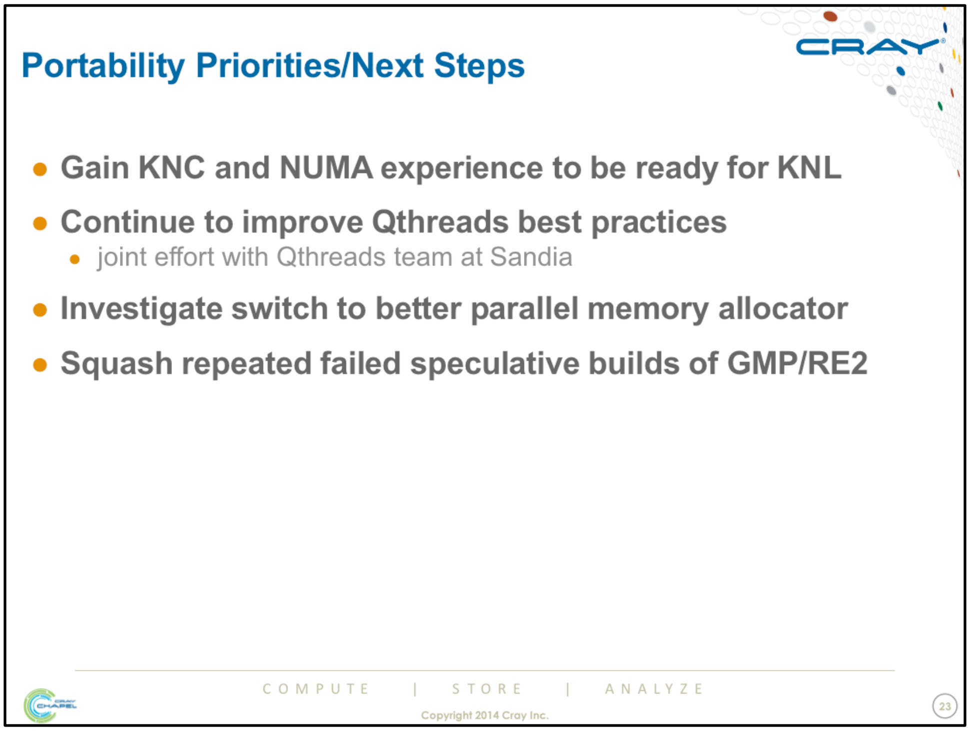
Other Improvements

- **Other Portability Improvements:**
  - Fixed a Cygwin portability regression
    - ‘uname’ output on newer Cygwin versions changed format
    - this confused our scripts that infer the platform from ‘uname’ output

- **Other Third-Party Improvements**
  - Improved our use of Qthreads (see ‘runtime’ slides)
  - Updated our snapshot of hwloc to version 1.9.1
  - Updated our snapshot of TCMalloc to version 2.2
    - and patched it to work with Chapel
  - Applied GASNet patches to fix gemini/aries conduit issues
    - related to memory registration issues when pshm is disabled
  - Improved the portability of the re2 code base
  - Parallelized the builds of the llvm and gmp packages
  - Reduced the number of llvm components built

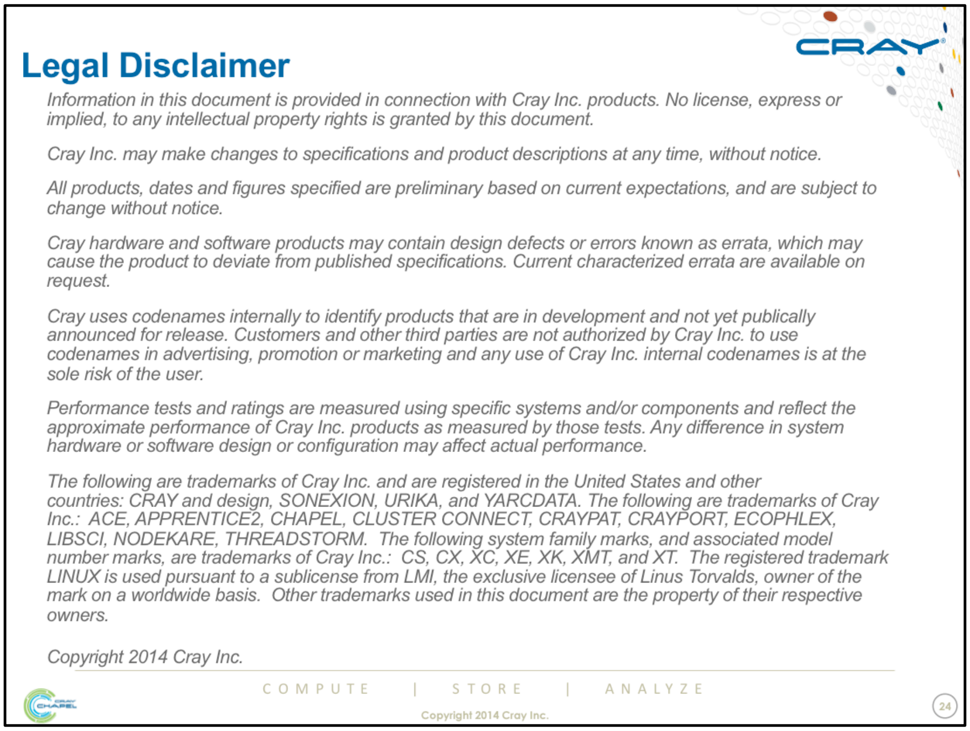


Portability Priorities/Next Steps



Portability Priorities/Next Steps

- Gain KNC and NUMA experience to be ready for KNL
- Continue to improve Qthreads best practices
  - joint effort with Qthreads team at Sandia
- Investigate switch to better parallel memory allocator
- Squash repeated failed speculative builds of GMP/RE2



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.

Copyright 2014 Cray Inc.

CRAY CHAPEL

CRAY

THE SUPERCOMPUTER COMPANY

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