

#### **Additional Progress and Status**

Chapel Team, Cray Inc. Chapel version 1.9

April 17th, 2014 (released) / May 2014 (documented)



#### **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.

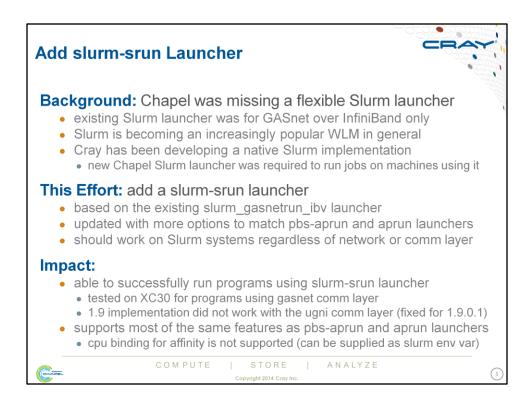


COMPUTE | STORE | ANALYZE

Copyright 2014 Cray Inc.

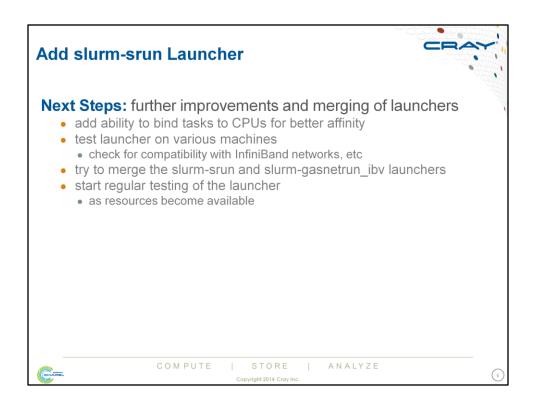
# Outline • Launcher Improvements • Portability Improvements • Outreach and Community Engagement • Laundry Lists



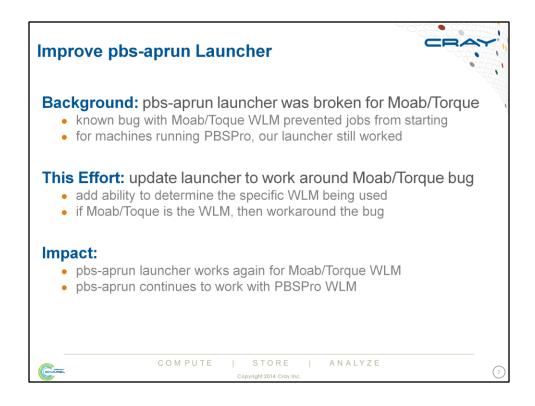


"Native" slurm isn't real term. It is more of the term that Cray is using for its slurm launcher since the initial efforts towards a slurm launcher was slurm over alps. Note that this launcher also works for Slurm outside of Cray systems

Initially, the slurm-srun launcher was put together close to the 1.9 release and had some bugs in it. Theses bugs led to suboptimal performance and didn't work with the UGNI comm layer. These issues have since been fixed and will go out with the 1.9.0.1 release (Cray module only).



The slurm-gasnetrun\_ibv uses gasnetrun for infiband to launch jobs instead of srun. It's not clear yet if we will be able to merge them or if there are cases where srun will not be able to launch jobs over infiniband; or whether maintaining the GASNet launcher will have other benefits.

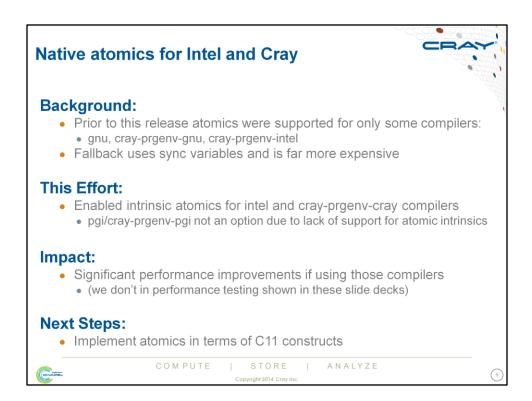


Moab/Torque recently (since Chapel 1.8) stopped starting jobs that have mppdepth=<numcpus> as part of the resource requirement. This is a known problem and as a workaround we now simply specify the number of nodes and walltime for Moab/Torque jobs and let aprun specify the number of cpus to use.

Previously we didn't differentiate WLMs as well as we could have. PBSPro was detected, but Moab/Torque got detected as 'unknown' and just had default settings used. The fix updated the launcher so we identify when we are using moab/torque and change our resource request appropriately.

Fixed with r23049 on 04/02/2014





To clarify, the difference between the 'foo' compiler and the 'cray-prgenv-foo' compiler is that the former refers to running the compiler directly (e.g., fcc) and the latter refers to running the Cray Programming Environment's 'cc' that wraps 'fcc'. These are treated as distinct compiler targets within Chapel to account for any distinct flags or default configurations that the Cray Programming Environment adds.

#### **Third-party Improvements**



#### **GASNet:**

- Upgraded to version 1.22.0
- Made 'aries' the default conduit for cray-xc platform by default
- Disabled 'pshm' for all non-udp GASNet conduits

#### GMP:

• Updated to version 6.0.0

#### hwloc:

• Added a snapshot of 1.7.2 to the release

#### re2:

• Added a snapshot of 20140111 to the release

#### **Qthreads:**

• made various minor improvements to our snapshot of 1.10

#### dygraphs:

• added a snapshot for generating performance graphs





### **Other Portability Improvements**



- Significantly Improved Portability to Mac OS X
  - used by default by most new hires to the team
  - changes included:
    - back-end compiler portability improvements
    - third-party package portability improvements
    - generated code improvements



#### **Presentations**



#### Pacific Northwest Numerical Analysis Seminar (PNWNAS):

- The Chapel Parallel Programming Language
  - 45-minute invited talk
  - October 19<sup>th</sup>, Seattle WA
  - ~65 attendees: students, faculty, and industry from WA, OR, ID, BC

#### SC13 Talks:

- Hierarchical Locales: Exposing the Node Architecture in Chapel
  - 20-minute invited talk
  - November 19<sup>th</sup>, KISTI Booth, Denver CO
  - ~12 attendees
- Chapel: An Emerging Parallel Programming Language
  - 15-minute talk in support of the Emerging Technologies booth
  - November 20<sup>th</sup>, HPC Impact Theatre, Denver CO
  - ~35 attendees





#### **SC13 Chapel Activities (Cray-led)**



- Emerging Technologies Booth presence/participation
  - poster, handouts, rotating booth-wide slideshow
  - continual stream of visitors for 3-1/2 days
    - some colleagues and collaborators, but at least as many new faces

#### Chapel Lightning Talks 2013 BoF

- · Cray-delivered overview talk
- Six talks from community members:
  - MiniMD (Brad)
  - Education (Tim Stitt, University of Notre Dame)
  - Chapel Over MPI-3 (Pavan Balaji, Argonne)
  - Autotuning (Ray Chen, U Maryland)
  - HDFS in Chapel (Michael Ferguson)
  - Futures (Shams Imam, Rice university)



#### **SC13 Chapel Activities (Community-led)**



#### Technical Poster: Ray Chen (UMD)

- topic: autotuning and Chapel
- a semi-finalist in the ACM Student Research Competition

#### HPC Educators Program: David Bunde (Knox), Kyle Burke (Colby)

- topic: use of Chapel in academic courses
- ~30 attendees for first half, ~10 for second
  - (apparently, many non-educators came to hear more about Chapel)

#### **More Presentations**



#### SIAM PP14:

- <u>Chapel Language Features for Hierarchical Tiling and Exascale</u> <u>Architectures</u> [audio + slides]
  - 20-minute minisymposium talk
  - February 19th, Portland OR
  - ~30 SIAM PP14 attendees: DOE, academics, etc.
- Co-Design Via Proxy Applications: MiniMD in Chapel
  - 20-minute minisymposium talk
  - February 21<sup>st</sup>, Portland OR
  - ~15 SIAM PP14 attendees, primarily DOE, some academics

#### SICM<sup>2</sup> workshop (computational chemistry):

- Chapel, Life, the Universe
  - 30-minute invited talk
  - March 29<sup>th</sup>, Manhattan, NY
  - ~45 attendees, 2/3 computational chemists, 1/3 computer science



COMPUTE | STORE | ANALYZE

Copyright 2014 Cray Inc.

#### **Even More Presentations**



#### Northwest C++ Users Group:

- Chapel: An Emerging Parallel Programming Language [video]
  - 60-minute invited talk
  - April 16th, Redmond WA
  - ~30 attendees: primarily local mainstream/"big data" professionals

#### **Notable Talks by non-Cray Personnel**



- SIGCSE 2014 Workshop
  - Chapel: A Versatile Tool for Teaching Undergraduates Parallel **Programming** 
    - David Bunde (Knox College) and Kyle Burke (Colby College)
    - March 8th, Atlanta GA

#### CCSC Regional Conference

- Using Chapel to Teach Parallel Concepts
  - David Bunde (Knox College)
  - April 4<sup>th</sup>, Fulton, MO

(see <a href="http://chapel.cray.com">http://chapel.cray.com</a> for links to materials)



#### Other presentations in the interim



ETH Zurich: 2-hour seminar

PADAL Workshop (Lugano): 15-minute invited panel talk

(see <a href="http://chapel.cray.com">http://chapel.cray.com</a> for links to materials)



#### **Notable Upcoming Outreach Opportunities**



- CHIUW: Chapel Implementers and Users Workshop
  - May, Phoenix AZ
  - http://chapel.cray.com/CHIUW.html
- SC14
  - Submitted a tutorial proposal
  - November, New Orleans LA
- Repeat Invitation to speak at Argonne Summer School
- DOE ASCR Productivity Workshop
- Third Cray blog article ("Why Chapel?")





## Other Improvements not covered here --minimal-modules compilation mode for developers avoids the vast majority of internal/standard module contents useful for rapidly testing core language features

• removed --serial and --serial-forall flags

#### **Notable Community Contributions**



#### Michael Ferguson (LTS):

- Significant LLVM back-end improvements
- Implemented improved operator precedence
- Improved support for c\_ptr / c\_ptrTo
- Improved casts to boolean types
- Propagated CHPL \* environment variables to GASNet launcher
- Optional runtime support for thread-local storage using \_\_thread
- QIO, GMP bug fixes
- Portability improvements for Debian, Mac

#### **Chris Wailes (Indiana):**

• function resolution improvements including partial instantiation

#### Rafael Larrosa (Malaga):

- Launcher improvements
- GASNet/ibv portability improvement

#### **Brandon Ross (Buffalo):**

string utilities



#### **Notable Bug Fixes**



- Improved const checking on records and fields
- Fixed a bug in string-to-c\_string conversions
- Fixed a race condition in task reporting
- bug fix for abs(real) (and support for abs(imag)
- semantic check for tuple size mismatches
- bug fix for timer.clear()
- bug fixes for representation of literals in generated code
- bug fix for usernames containing spaces
- bug fix relating to do...while warnings on local constants



#### **Language Specification Improvements**



- Documented atomic variables
- Documented new assignment signature
- Documented 'noinit'
- Documented expression-less 'serial' statement
- Documented 'ref this' intents
- Clarified casting from numeric types to 'bool'
- Clarified rules related to nil casting and deletion
- Updated operator precedence table to reflect new choices
- Various other improvements and cleanups



#### **Other Documentation Improvements**



- Added a PERFORMANCE file to address FAQs
- Rewrote top-level LICENSE files to clarify third-party terms
- Removed user AGREEMENT to avoid compromising BSD
- Refreshed and reorganized README.tasks
- Noted 'clang' as a CHPL\_\*\_COMPILER option
- Refreshed and reorganized README.tasks
- Improved description of Cray environment variables
- Clarified formatted I/O documentation
- General refresh and improvements to various READMEs



#### **Correctness Testing Improvements**



- Made testing system less Cray-centric
- Moved test result emails to SourceForge mailing lists
- Added numtrials capability
- Added support for nightly testing against --fast



#### **Performance Testing Improvements**



- pushed performance testing graphs to public web
- automated SVN tracking of nightly performance data
- added verification capability
- added ability to track compiler performance
- implemented an annotation capability for perf. graphs
  - not enabled yet, but can be utilized on a per-user basis
- scripts for release-over-release testing
  - and append nightly results to historical data for instant comparisons
- switched to using --static and --fast by default
- utility script for splicing .dat files from performance runs



#### **Legal Disclaimer**



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, APDEKARE, 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.



COMPUTE | STORE |

Copyright 2014 Cray Inc

ANALYZE



