CHIUW 2015:
The ACM SIGPLAN 2nd Annual Chapel Implementers and Users Workshop

a PLDI 2015/FCRC 2015 workshop
June 13-14, 2015
Portland, OR

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.

Welcome to CHIUW!

Brad Chamberlain, Cray Inc.

June 13, 2015

CHIUW 2015

What is Chapel?

- **An emerging parallel programming language**
  - Design and development led by Cray Inc.
    - in collaboration with academia, labs, industry; domestically & internationally
- **Goal: Improve productivity of parallel programming**
- **Being developed as open-source at GitHub**
  - Licensed as Apache v2.0 software
  - Portable design and implementation
  - a work-in-progress

Chapel's 5-year push

- **Due to positive user response to Chapel under HPCS, Cray undertook a five-year effort to improve it**
  - we've just completed our second year
- **Focus Areas:**
  1. Improving **performance** and scaling
  2. **Fixing** immature aspects of the language and implementation
     - e.g., strings, memory management, error handling, …
  3. **Porting** to emerging architectures
     - Intel Xeon Phi, accelerators, heterogeneous processors and memories, …
  4. Improving **interoperability**
  5. Growing the Chapel user and developer **community**
     - including non-scientific computing communities
  6. Exploring transition of Chapel **governance** to a neutral, external body

CHIUW 2014

- **As part of community development, we kicked off CHIUW**
  - CHIUW = Chapel Implementers and Users Workshop
    (name chosen to complement CHUG: the Chapel Users Group)

- **CHIUW 2014 was held as an IPDPS workshop**
  - May 23, 2014, Phoenix, AZ
  - presentations archived at chapel.cray.com

CHIUW 2014 Talks and Speakers

User Experiences with a Chapel Implementation of UTS
Jens Breitbart, Technische Universität München

Evaluating Next Generation PGAS Languages for Computational Chemistry
Daniel Chavarria-Miranda, Pacific Northwest National Laboratory

Programmer-Guided Reliability in Chapel
David E. Bernholdt, Oak Ridge National Laboratory

Towards Interfaces for Chapel
Chris Wailes, Indiana University

Affine Loop Optimization using Modulo Unrolling in Chapel
Aroon Sharma, University of Maryland

Keynote: Walking to the Chapel
Robert Harrison, Stony Brook University / Brookhaven National Laboratory

LLVM Optimizations for PGAS Programs
Akihiro Hayashi, Rice University

Opportunities for Integrating Tasking and Communication Layers
Dylan T. Stark, Sandia National Laboratories

Caching in on Aggregation
Michael Ferguson, Laboratory for Telecommunication Sciences

CHIUW 2015 at-a-glance

Today: mini-conference

- Keynote by Bill Carlson (IDA)
- Six “Technical talks” (30 minutes)
- Six “Hot Topics talks” (10 minutes)
- Wrap-up discussion session

Tomorrow: code camp

- Have identified ~a half dozen pair-programming activities and groups
  - feel free to sit in on one or propose your own
- Officially a half-day workshop (w.r.t. the room, breaks)
  - groups could go longer (in hallways and ad hoc spaces) if desired

CHIUW 2015: Today's Schedule

8:30: Chapel Boot Camp (optional)

9:00: **Welcome, State of the Project**

9:30: **Technical Talks**

11:00: **Break** (provided)

11:30: **Technical Talks**

12:30: **Lunch** (provided?)

2:00: **Keynote: Bill Carlson (IDA)**

"Shared Memory HPC Programming: Past, Present, and Future"

3:00: **Technical Talks**

3:30: **Break** (provided)

4:00: **Hot Topics Talks**

5:00: **Community Discussion**

6:00: **Dinner** (on our own)

CHIUW 2015 Technical Talks

9:30-11:00

Practical Diamond Tiling for Stencil Computations Using Chapel Iterators
Michelle Strout, Ian Bertolacci, and Catherine Olschanowsky (*Colorado State University*), **Ben Harshbarger and Brad Chamberlain** (*Cray Inc.*), **David G. Wonnacott** (*Haverford College*)

A Study of Red-Black SOR Parallelization Using Chapel, D, and Go Languages
Sparsh Mittal (*Oak Ridge National Lab*)

Data-Centric Locality in Chapel
Ben Harshbarger (*Cray Inc.*)

11:30-12:30

Parallac: Using Chapel with ARM Clusters
Brian Guarraci (*Twitter Inc.*)

Hierarchical Locale Models in Chapel
Sung-Eun Choi, David Iten, Elliot Ronaghan, Greg Titus (*Cray Inc.*)

3:00-3:30

Vectorization of Chapel Code
Elliot Ronaghan (*Cray Inc.*)

CHIUW 2015 Hot Topic Talks

4:00-5:00

The Chapel Memory Consistency Model
Sung-Eun Choi, Michael Ferguson, Elliot Ronaghan, Greg Titus (*Cray Inc.*)

Fast Fourier Transforms in Chapel
Doru Thom Popovici, Franz Franchetti (*Carnegie-Mellon University*)

A Preliminary Performance Comparison of Chapel to MPI and MPI/OpenMP
Laura Brown (*US Army Engineer Research and Development Center*)

Data flow programming—a high performance and highly complicated programming concept?
Jens Breitbart (*Technische Universität München*)

If you can dodge a wrench, you can dodge a ball
Dylan Stark, George Stelle (*Sandia National Laboratories*)

A Progress Report on COHX: Chapel on HSA + XTQ
Mauricio Breternitz, Bibek Ghimire, Mike Chu, Steve Reinhardt (*Advanced Micro Devices (AMD)*)

CHIUW 2015 Code Camp Activities (Tomorrow)

Active Libraries in Chapel
- Michelle Strout, Ben Harshbarger, …

Native FFTs in Chapel
- Doru Thom Popovici, Kyle Brady, David Iten, …

Data Processing Workloads in Chapel
- Brian Guarraci, Michael Ferguson, …

Active Message Data Transfer Optimizations
- Mauricio Breternitz, Greg Titus, Elliot Ronaghan, …

Code Generation for HSA (via LLVM?)
- Mauricio Breternitz, Greg Titus, Elliot Ronaghan, Michael Ferguson, …

Libraries for plotting, image processing, linear algebra, …
- Chris Taylor, Brad Chamberlain, Lydia Duncan, …

ZeroMQ in Chapel
- Nick Park, Lydia Duncan, …

Your Idea Here?

CHIUW 2015 Planning Committee

General Chair: **Tom MacDonald**, *Cray Inc.*

Program Committee:

- **Brad Chamberlain** (chair), *Cray Inc.*
- **Rafael Asenjo**, *Universidad de Málaga*
- **Richard Barrett**, *Sandia National Laboratories*
- **Jens Breitbart**, *Technische Universität München*
- **Mauricio Breternitz**, *AMD*
- **Jeff Hammond**, *Intel*
- **Rob Neely**, *Lawrence Livermore National Laboratory*
- **Michelle Strout**, *Colorado State University*
- **Michele Weiland**, *EPCC*

In forming the committee, we strived to balance…

- developers and users
- academics, lab employees, and industry
- domestic and international

State of the Chapel Project

Brad Chamberlain, Cray Inc.

June 13, 2015

CHIUW 2015

The Chapel Team at Cray

A Year in the Life of Chapel

- **Two major releases per year** (April / October)
  - **~a month later:** detailed release notes
- **CHIUW:** Chapel Implementers and Users Workshop (May/June)
- **SC** (Nov)
  - annual **Lightning Talks BoF** featuring talks from the community
  - annual **CHUG happy hour**
  - plus tutorials, panels, BoFs, posters, educator sessions, exhibits, …
- **Talks, tutorials, research visits, blogs, …** (year-round)

A Year in the Life of Chapel

- **Two major releases per year** (April / October)
  - **~a month later:** detailed release notes
- **CHIUW:** Chapel Implementers and Users Workshop (May/June)
- **SC** (Nov)
  - annual **Lightning Talks BoF** featuring talks from the community
  - annual **CHUG happy hour**
  - plus tutorials, panels, BoFs, posters, educator sessions, exhibits, …
- **Talks, tutorials, research visits, blogs, …** (year-round)

Release Highlights since CHIUW 2014

- **Standard Library Improvements:**
  - FFTW
  - file system utilities
  - I/O support for Lustre- and cURL-based resources
  - bit operations
- **Documentation:**
  - standard library modules documented at chapel.cray.com/docs/latest
- **Performance Optimizations:**
  - vectorization of data parallel loops and operations
  - locality optimizations
  - now use Qthreads tasking by default
  - now use ugni/muxed runtime on Cray systems by default
- **Generated Code Improvements:**
  - now generate readable C for-loops for many common Chapel loops

Release Highlights since CHIUW 2014 (featured today)

- **Standard Library Improvements:**
  - FFTW
  - file system utilities
  - I/O support for Lustre- and cURL-based resources
  - bit operations
- **Documentation:**
  - standard library modules documented at chapel.cray.com/docs/latest
- **Performance Optimizations:**
  - vectorization of data parallel loops and operations
  - locality optimizations
  - now use Qthreads tasking by default
  - now use ugni/muxed runtime on Cray systems by default
- **Generated Code Improvements:**
  - now generate readable C for-loops for many common Chapel loops

More Release Highlights since CHIUW 2014

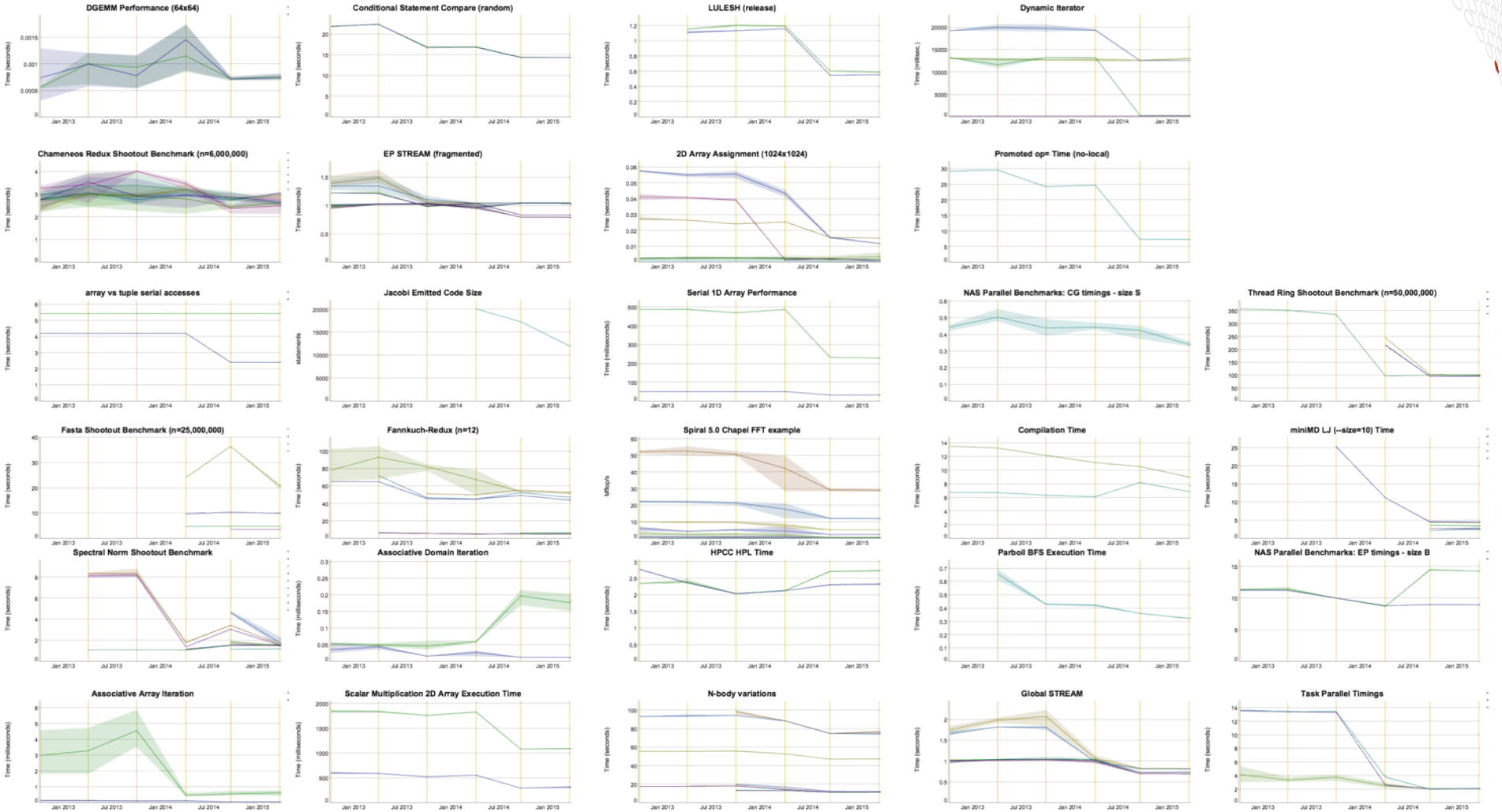
- **New Tools:**
  - chpldoc support for source-based documentation
  - chpltags for emacs/vim Chapel source navigation
  - early prototype Chapel interpreter
- **Language Improvements:**
  - standalone (non-zippered) parallel iterators
  - support for vector/set operations on arrays
  - task intent improvements
  - ability to query local portions of domains/arrays
- **Interoperability Improvements**
  - initial support for Python-to-Chapel interoperability
  - can now pass contiguous Chapel arrays to external procedures
- **Portability Improvements:**
  - Initial support for Intel Xeon Phi Knights Corner (KNC)

Process Improvements (2014-present)

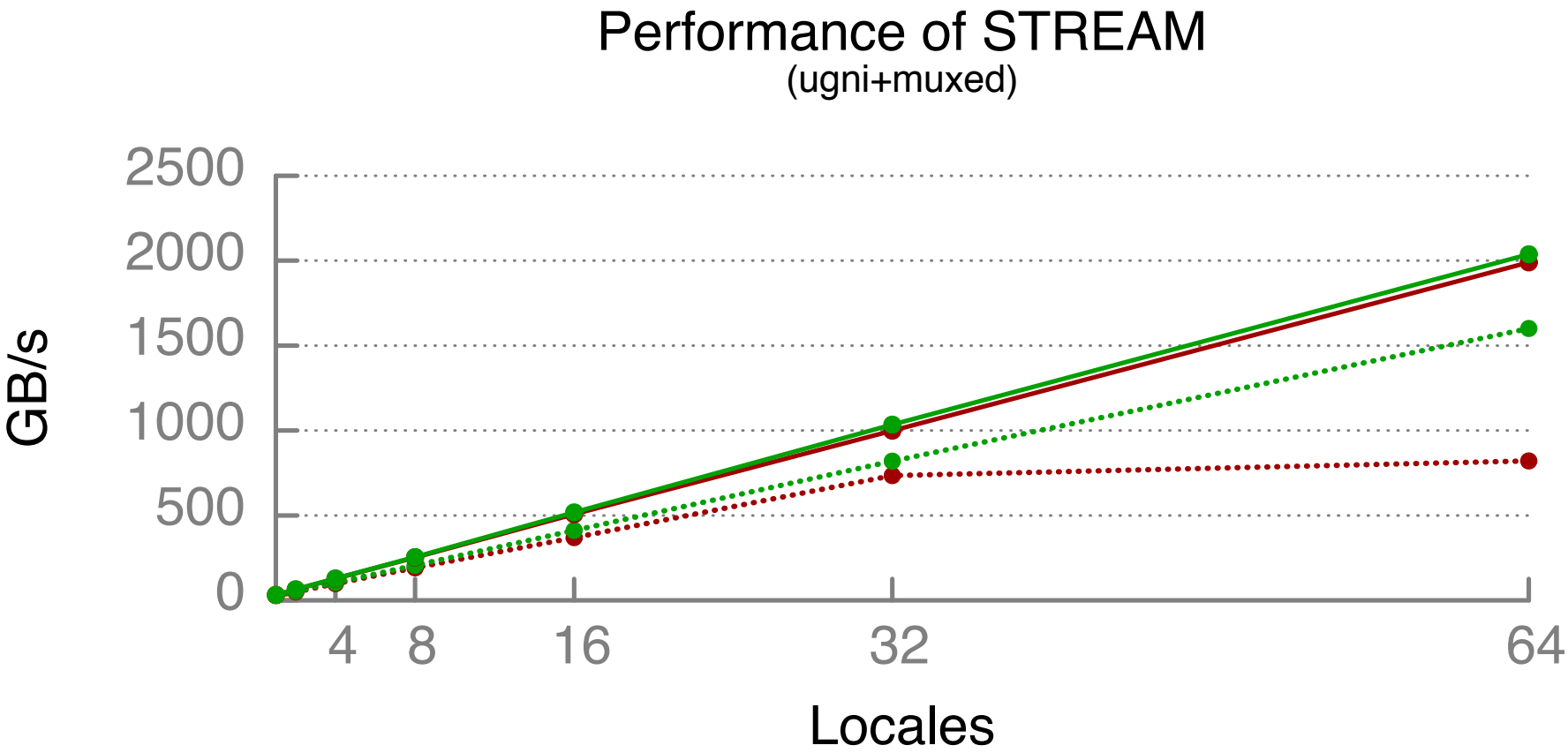
- Migrated from SVN/SourceForge to Git/GitHub
- Converted testing from crontabs to Jenkins
- Began using Travis for pre-commit sanity checks
- Began using Coverity Scan to catch code quality issues
- Started tracking tasks in Pivotal
- Kicked off a Facebook page
- Created web documentation with Sphinx
- Started a Jira-based issue tracker
- Started a #chapel-developers IRC channel
- Owned the Chapel entry in OpenHUB

Single-Locale Execution Time is Improving

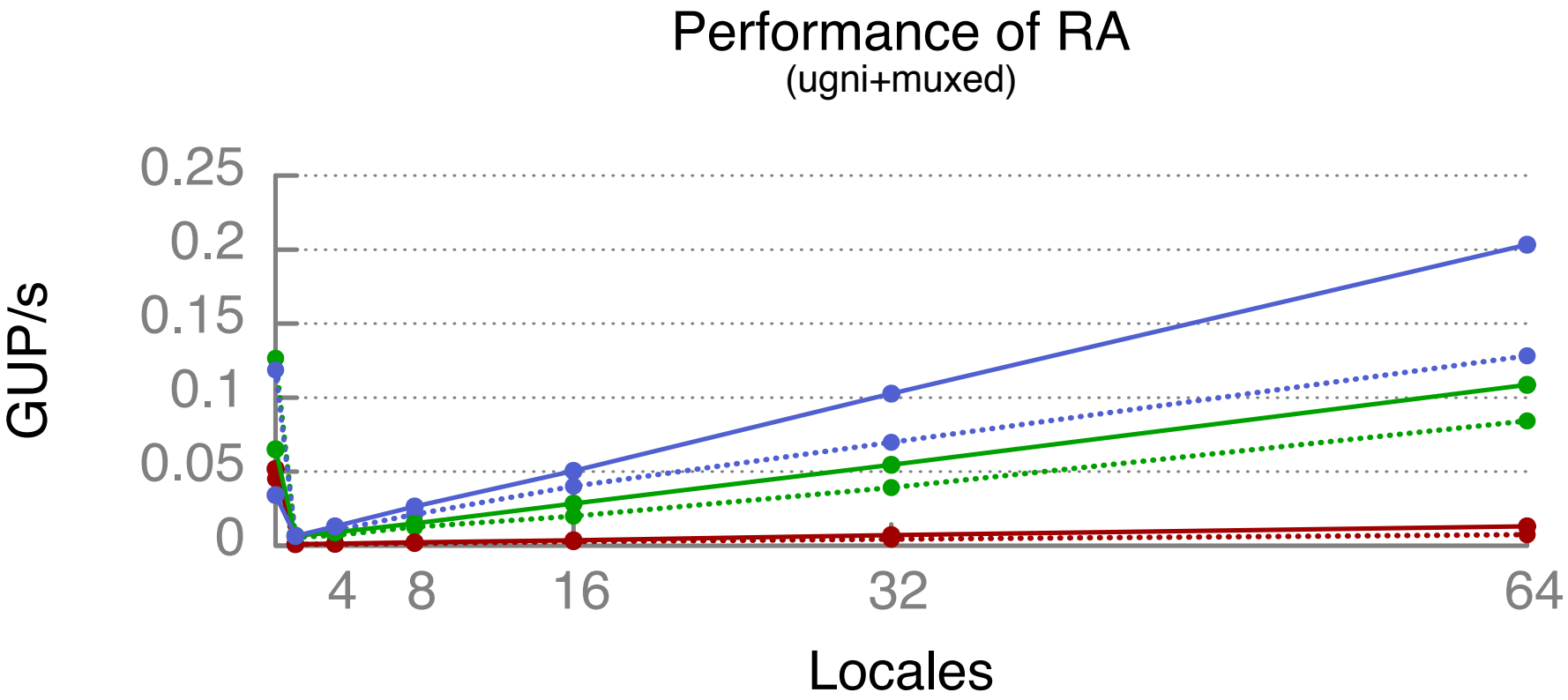
lower is better, yellow lines indicate releases (1.6-1.11)



STREAM Scalability, 1.9 vs. 1.10 (higher is better)

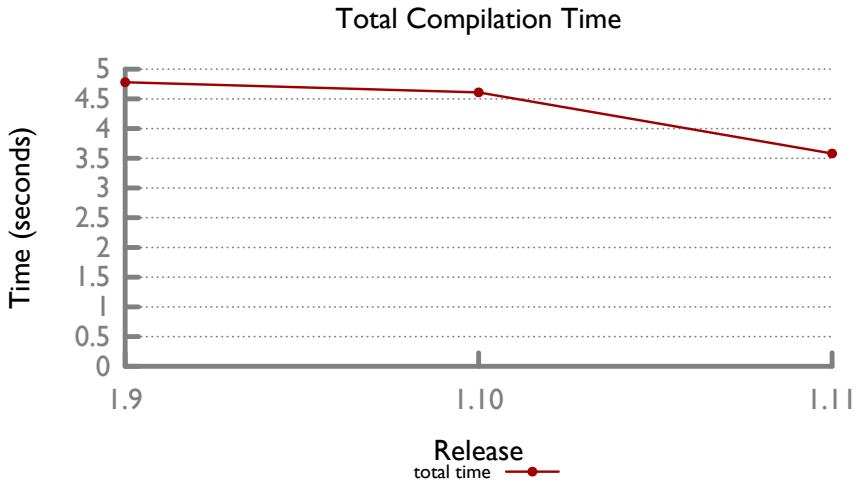


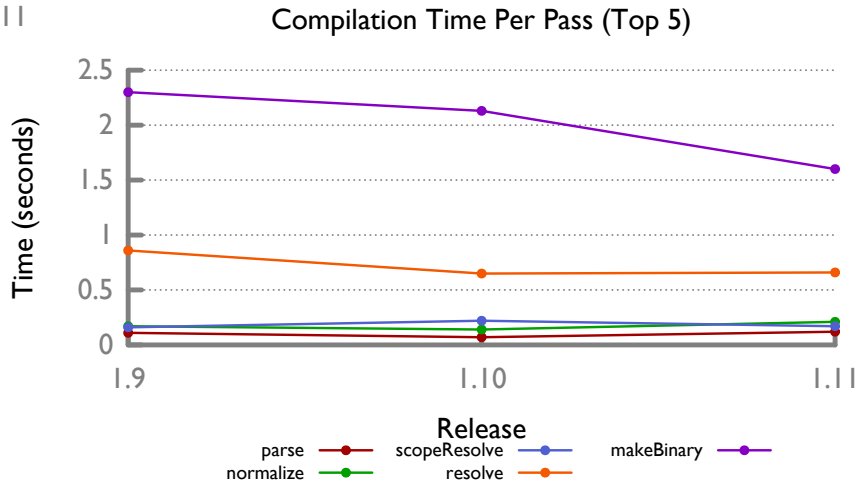
RA Scalability, 1.9 vs. 1.10 (higher is better)



Compiler Performance

- Compilation time has improved over the past several releases





A Year in the Life of Chapel

- **Two major releases per year** (April / October)
  - **~a month later:** detailed release notes
- **CHIUW:** Chapel Implementers and Users Workshop (May/June)
- **SC** (Nov)
  - annual **Lightning Talks BoF** featuring talks from the community
  - annual **CHUG happy hour**
  - plus tutorials, panels, BoFs, posters, educator sessions, exhibits, …
- **Talks, tutorials, research visits, blogs, …** (year-round)

Chapel at SC14

Chapel Tutorial (Sun @ 8:30)
"A Computation-Driven Introduction to Parallel Computing in Chapel"

Hierarchical Locales Exhibit at Emerging Technologies Booth (all week, booth #233)
poster staffed by members of the Chapel team

4th Annual Chapel Lightning Talks BoF (Tues @ 12:15, room 293)
5-minute talks on Chapel + HSA, HDFS/Lustre/cURL, tilings, LLVM, ExMatEx, Python

Talk on Hierarchical Locales (Tues @ 4:30, Emerging Technologies Theater, booth #233)
"Chapel Hierarchical Locales: Adaptable Portability for Exascale Node Architectures", Greg Titus (Cray)

Poster on Advanced Tilings in Chapel (Tues @ 5:15, New Orleans Theater Lobby)
"Orthogonal Scheduling of Stencil Computations with Chapel Iterators", Ian Bertolacci (Colorado State)

Chapel Users Group (CHUG) BoF (Wed @ 5:30, room 383-84-85)
Chapel overview and current events, followed by community Q&A and discussion

5th Annual CHUG Happy Hour (Wed @ 7:15, Mulate's at 201 Julia St)
social gathering just across the way; open to general public, dutch treat

Participation in other BoFs:

- **LLVM in HPC** (Tues @ 12:15, room 283-84-85)
- **Programming Abstractions for Data Locality** (Wed @ 12:15, room 391-92)
- **PGAS: Partitioned Address Space Programming Model** (Wed @ 12:15, room 273)

Chapel at SC14

Chapel Tutorial (Sun @ 8:30)
"A Computation-Driven Introduction to Parallel Computing in Chapel"

Hierarchical Locales Exhibit at Emerging Technologies Booth (all week, booth #233)
poster staffed by members of the Chapel team

4th Annual Chapel Lightning Talks BoF (Tues @ 12:15, room 293)
5-minute talks on Chapel + HSA, HDFS/Lustre/cURL, tilings, LLVM, ExMatEx, Python

Talk on Hierarchical Locales (Tues @ 4:30, Emerging Technologies Theater, booth #233)
"Chapel Hierarchical Locales: Adaptable Portability for Exascale Node Architectures", Greg Titus (Cray)

Poster on Advanced Tilings in Chapel (Tues @ 5:15, New Orleans Theater Lobby)
"Orthogonal Scheduling of Stencil Computations with Chapel Iterators", Ian Bertolacci (Colorado State)

Chapel Users Group (CHUG) BoF (Wed @ 5:30, room 383-84-85)
Chapel overview and current events, followed by community Q&A and discussion

5th Annual CHUG Happy Hour (Wed @ 7:15, Mulate's at 201 Julia St)
social gathering just across the way; open to general public, dutch treat

Participation in other BoFs:

- **LLVM in HPC** (Tues @ 12:15, room 283-84-85)
- **Programming Abstractions for Data Locality** (Wed @ 12:15, room 391-92)
- **PGAS: Partitioned Address Space Programming Model** (Wed @ 12:15, room 273)

Chapel Lightning Talks 2014 (SC14 BoF)

Chapel Overview
Greg Titus, Cray Inc.

CoMD in Chapel: The Good, the Bad, and the Ugly
David Richards, Lawrence Livermore National Laboratory

Chapel for Python Programmers
Simon Lund, University of Copenhagen

Chapel Iterators: Providing Tiling for the Rest of Us
Ian Bertolacci, Colorado State University

Chapel I/O: Getting to Your Data Wherever It Is
Tim Zakian, Indiana University

LLVM-based Communication Optimizations for Chapel
Akihiro Hayashi, Rice University

COHX: Chapel on HSX + XTQ
(Adventures of a PGAS Language in a Heterogenous World)
Deepak Majeti, Rice University / AMD intern

Chapel at SC14

Chapel Tutorial (Sun @ 8:30)
"A Computation-Driven Introduction to Parallel Computing in Chapel"

Hierarchical Locales Exhibit at Emerging Technologies Booth (all week, booth #233)
poster staffed by members of the Chapel team

4th Annual Chapel Lightning Talks BoF (Tues @ 12:15, room 293)
5-minute talks on Chapel + HSA, HDFS/Lustre/cURL, tilings, LLVM, ExMatEx, Python

Talk on Hierarchical Locales (Tues @ 4:30, Emerging Technologies Theater, booth #233)
"Chapel Hierarchical Locales: Adaptable Portability for Exascale Node Architectures", Greg Titus (Cray)

Poster on Advanced Tilings in Chapel (Tues @ 5:15, New Orleans Theater Lobby)
"Orthogonal Scheduling of Stencil Computations with Chapel Iterators", Ian Bertolacci (Colorado State)

Chapel Users Group (CHUG) BoF (Wed @ 5:30, room 383-84-85)
Chapel overview and current events, followed by community Q&A and discussion

5th Annual CHUG Happy Hour (Wed @ 7:15, Mulate's at 201 Julia St)
social gathering just across the way; open to general public, dutch treat

Participation in other BoFs:

- **LLVM in HPC** (Tues @ 12:15, room 283-84-85)
- **Programming Abstractions for Data Locality** (Wed @ 12:15, room 391-92)
- **PGAS: Partitioned Address Space Programming Model** (Wed @ 12:15, room 273)

Recent Chapel Publications

Parameterized Diamond Tiling for Stencil Computations with Chapel Iterators, Ian Bertolacci, Michelle Strout, Catherine Olschanowsky (*Colorado State University*); **Ben Harshbarger, Bradford Chamberlain** (*Cray Inc.*), **David Wonnacott** (*Haverford College*), *29th International Conference on Supercomputing (ICS 2015)*, Newport Beach, CA, June 8-11, 2015.

Scripting Language Performance Through Interoperability, Simon Lund (University of Copenhagen), **Bradford Chamberlain** (Cray Inc.), **Brian Vintner** (University of Copenhagen), *First Workshop on the High Performance Scripting Languages (HPSL, a PPoPP 2015 workshop)*, San Francisco CA, February 7, 2015.

A Study of Successive Over-relaxation (SOR) Method Parallelization Over Modern HPC Languages, Sparsh Mittal (ORNL), *International Journal of High Performance Computing and Networking (IJHPCN)*, vol. 7, no. 4, 2014.

Under review: **LLVM-based Communication Optimizations for PGAS Programs, Akihiro Hayashi and Jisheng Zhao** (Rice University), **Michael Ferguson** (Cray Inc.), **Vivek Sarkar** (Rice University), *in submission to the 24th International Conference on Parallel Architectures and Compilation Techniques (PACT 2015)*, San Francisco, CA, October 18-21, 2015.

Recent Chapel Publications (featured today)

Parameterized Diamond Tiling for Stencil Computations with Chapel Iterators, Ian Bertolacci, Michelle Strout, Catherine Olschanowsky (*Colorado State University*); **Ben Harshbarger, Bradford Chamberlain** (*Cray Inc.*), **David Wonnacott** (*Haverford College*), *29th International Conference on Supercomputing (ICS 2015)*, Newport Beach, CA, June 8-11, 2015.

Scripting Language Performance Through Interoperability, Simon Lund (University of Copenhagen), **Bradford Chamberlain** (Cray Inc.), **Brian Vintner** (University of Copenhagen), *First Workshop on the High Performance Scripting Languages (HPSL, a PPoPP 2015 workshop)*, San Francisco CA, February 7, 2015.

A Study of Successive Over-relaxation (SOR) Method Parallelization Over Modern HPC Languages, Sparsh Mittal (ORNL), *International Journal of High Performance Computing and Networking (IJHPCN)*, vol. 7, no. 4, 2014.

Under review: **LLVM-based Communication Optimizations for PGAS Programs, Akihiro Hayashi and Jisheng Zhao** (Rice University), **Michael Ferguson** (Cray Inc.), **Vivek Sarkar** (Rice University), *in submission to the 24th International Conference on Parallel Architectures and Compilation Techniques (PACT 2015)*, San Francisco, CA, October 18-21, 2015.

Community Blog Article

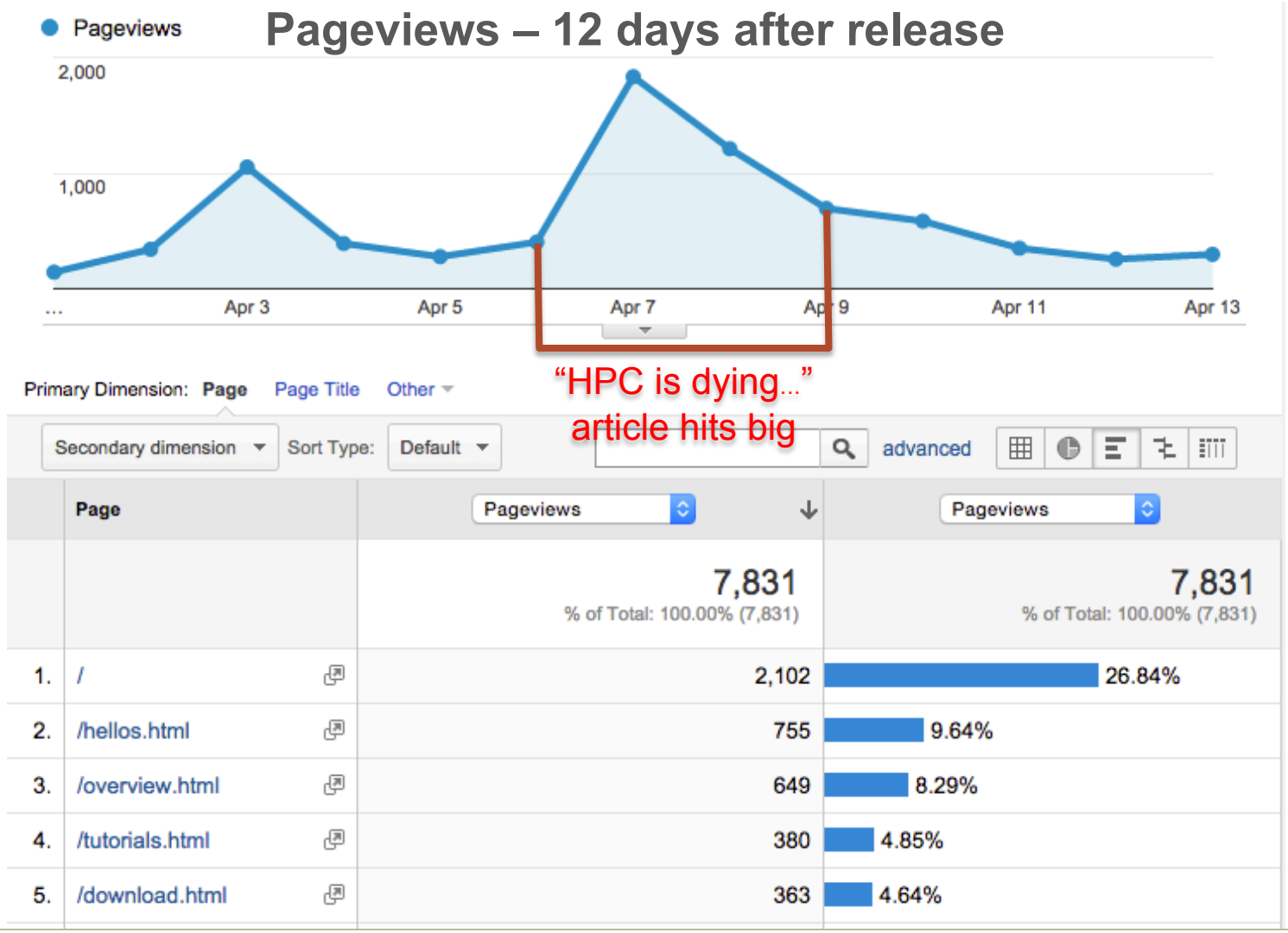
"HPC is Dying and MPI is Killing it" blog article

Jonathan Dursi, April 3, 2015

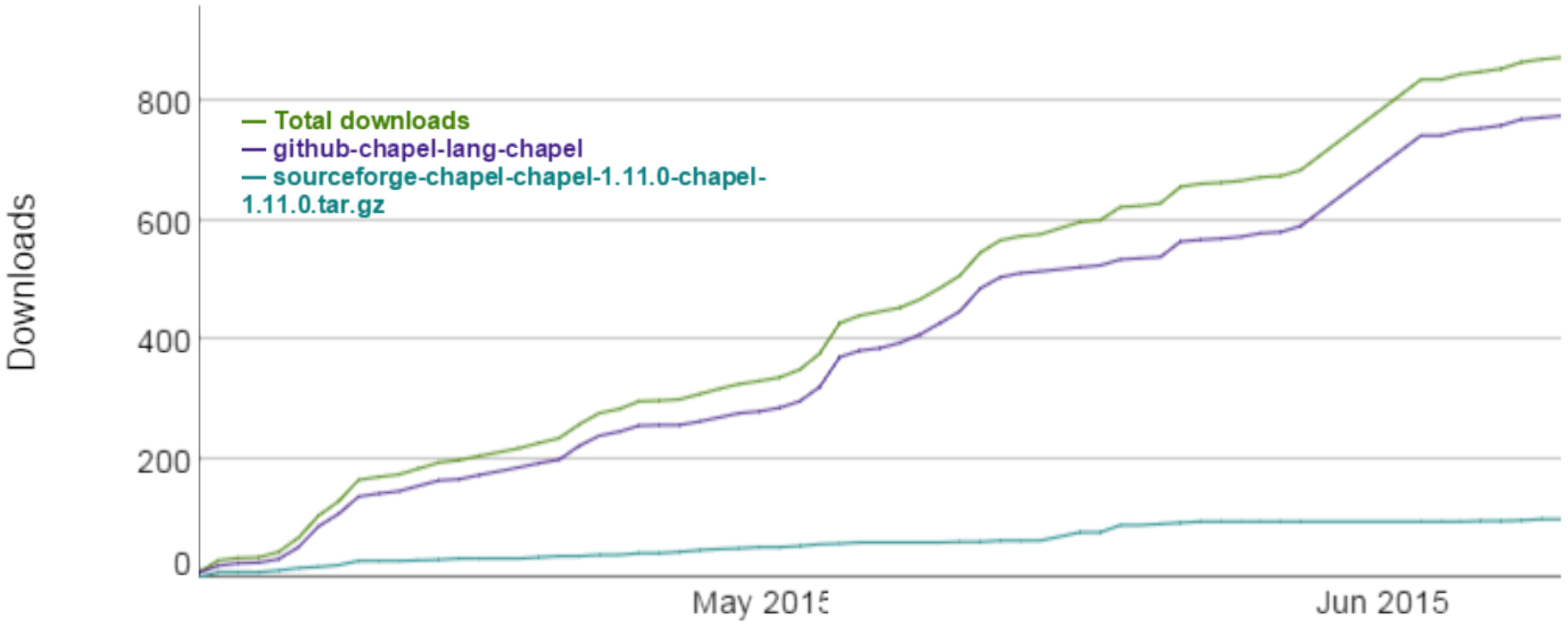
- extremely popular/debated article for a few days
- had nice things to say about Chapel and Spark

Pictured: The HPC community bravely holds off the incoming tide of new technologies and applications. Via the BBC.

chapel.cray.com pageviews – circa 1.11 release



Chapel 1.11 Downloads



- **Recent Chapel releases have seen ~1000 downloads**
  - spread fairly evenly across six months with spikes around release, SC
- **The 1.11 (April) release already has 875+ downloads**

Notable Recent Users (not able to attend today)

Nikhil Padmanabhan, Professor of Physics, Yale University

- data analytics problems regarding formation/evolution of the universe
- interested in time-to-science over performance (on distributed memory)
- contributed FFTW library bindings to 1.11 release
- currently experimenting with k-d tree n-body computations in Chapel
  - work-in-progress: https://github.com/chapel-lang/chapel/tree/master/test/users/npadmana/twopt

Greg Kreider, Co-founder, Primordial Machine Vision Systems

- focused on image processing computations
- developed an extensive suite of notes and sample computations (next slide):
  - http://www.primordand.com/
  - http://www.primordand.com/chapel_by_ex.html
- key quote: ***"I decided to learn the language by porting several of the programs here, plus one or two new ones, to Chapel. It's gone well – the language is a good fit for this problem domain, and is comfortable to program in (although there are frustrating points...)"***
  - (we're currently working our way through these points with Greg now)

Chapel by Example--Image Processing (Kreider)

www.primordand.com

Apps | Behind The Scenes | 10 Science Fiction | Okayplayer's Top

PRIMORDIAL

MACHINE VISION SYSTEMS

Feel free to poke around.

NEW

CHAPEL BY EXAMPLE – IMAGE

We've collected the notes we took while learning the Chapel language
processing programs in it and packaged them up into a set of tutorials
Chapel By Example page. The examples include color converters, Ga
FAST corner detectors, and RANSAC feature matching.

The Applications page contains some of the work we've done developing apps for Android.

The Development pa
the experience.

The About page has more information about who we are and how to c

PRIMORDIAL

MACHINE VISION SYSTEMS

Chapel by E

Image Pro

We've been using the Chapel language to learn the language and see how
written up tutorial and examples. It's grown quite a bit.

This is the central page for the project, with links to the individual sectio
of each section. The programs have been chosen to increase in complex
is the Front Page. You can also download the entire series in one PDF d

Page	Topics	File
Front	Introduction An Example Overview Install	
Image Interface	Introduction C Library C Interface from Chapel types, variables procedures, linkage structural types modules Wrap-Up / Exercises	tarb
Color Conversion	Introduction Color Spaces Language Description expressions, statements enumerations, tuples Arrays and Ranges Domains Program Organization Wrap-Up / Exercises	tarb
Gabor Filter	Introduction Edge Detectors Subdomains, Subranges Gabor Filters Wrap-Up / Exercises	tarb
Parallel Programming	Introduction Data Parallelism Task Parallelism Synchronization	tarb

CHAPEL BY EXAMPLE

IMAGE PROCESSING

More Notable Recent Users (unable to attend today)

Uwe Zimmer, Fellow of Computer Science, Australian National University

- starting to integrate Chapel into one or two central courses:
  - Concurrent and Distributed Systems
  - Real-time and Embedded Systems
- key quote: ***"We had an eye on Chapel for awhile, but now is the time when I will become serious about adding it..."***

Damian McGuckin, Managing Director, Pacific Engineering Systems Int'l

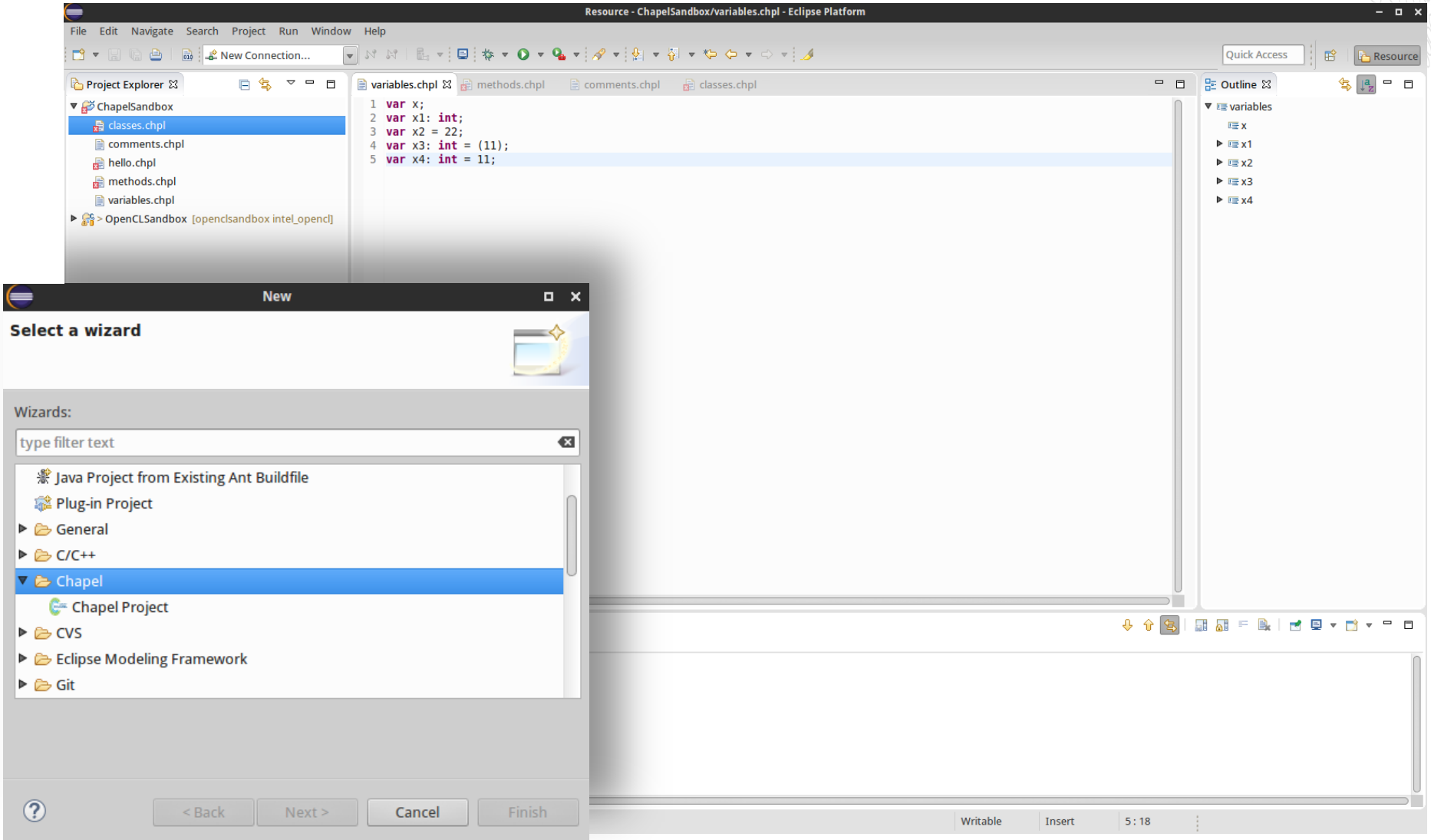
- computational focus: Automotive, Aeospace, Oil&Gas, Geophysics, …
- looking at finite element/volume computations in Chapel
- currently focused on clean, efficient, generic IEEE 754 routines
- key quotes: ***"I expect all the computationally intensive programs that we develop, or will just use, will be written in Chapel, or some language very much like it, within 5 years." "Your work in producing Chapel has yielded quite a remarkable tool."***

Notable Recent Tool Efforts (unable to attend today)

Atilla Sragli, Zoltan Matyas, engineers at TTTech Hungary (personal project)

- write aerospace tools for designing, scheduling, analyzing real-time networks
  - use C/C++, MPI, & Java in production
  - experimenting with Chapel in spare time
- developing a **Chapel IDE** using Eclipse Xtext
  - https://bitbucket.org/ngmschapel/hu.ngms.chapel/overview

Chapel IDE Screenshots (Sragli & Matyas)



Chapel IDE Screenshots (Sragli & Matyas)

variables.chpl | methods.chpl | comments.chpl | classes.chpl

```
77 // There are a few method names that cause the method to have special
78 // meaning.  A method named 'this' allows a class to be indexed like
79 // an array.  A method named 'these' allows a class to be iterated
80 // over.
81 //
82 class ArrayLike {
83   var a, b, c, d: int;
84   proc this(i:int) ref {
85     select i {
86       when 1 do return a;
87       when 2 do return b;
88       when 3 do return c;
89       when 4 do return d;
90       otherwise halt("index out of bounds", i)
91     }
92   }
93
94   iter these() ref {
95     yield a;
96     yield b;
97     yield c;
98     yield d;
99   }
100 }
101
102 //
103 // An 'ArrayLike' object can be indexed like a
104 // can be iterated over in a loop.
105 //
106 var a = new ArrayLike();
107 a(2) = 1; // call to this method
108 a(4) = 2; // call to this method
109
110 for elt in a do // invocation of these iterato
111   writeln(elt);
112 delete a;
```

variables.chpl | methods.chpl | comments.chpl | classes.chpl

```
1 /*
2  * Classes Primer
3  *
4  * This primer covers the declaration and use of classes.
5  *
6  */
7
8 //
9 // A class is a type that can contain variables and constants, called
10 // fields, as well as functions and iterators called methods.  A new
11 // class type is declared using the 'class' keyword.
12 //
13 class C {
14   var
15   proc
16     wr
17   }
18 }
19
20 //
21 // Class variables can be declared using the 'new' keyword to call the
22 // constructor for the class C.  The default constructor has an argument for
23 // each field in the class.  Once a class has been constructed, its methods
24 // can be called.
25 //
26 var c = new C(1, 2);
27 c.printFields();
28
29 //
30 // Default output is supported so a class can be written by making a
31 // call to write or writeln.  Default input is also supported.
32 //
33 writeln(c);
34
35 //
36 // Methods can also be defined outside of the class body by prefixing
```

ClassType **C**

Classes Primer This primer covers the declaration and use of classes.

Press 'F2' for focus

Notable Recent Tool Efforts (unable to attend today)

Atilla Sragli, Zoltan Matyas, engineers at TTTech Hungary (personal project)

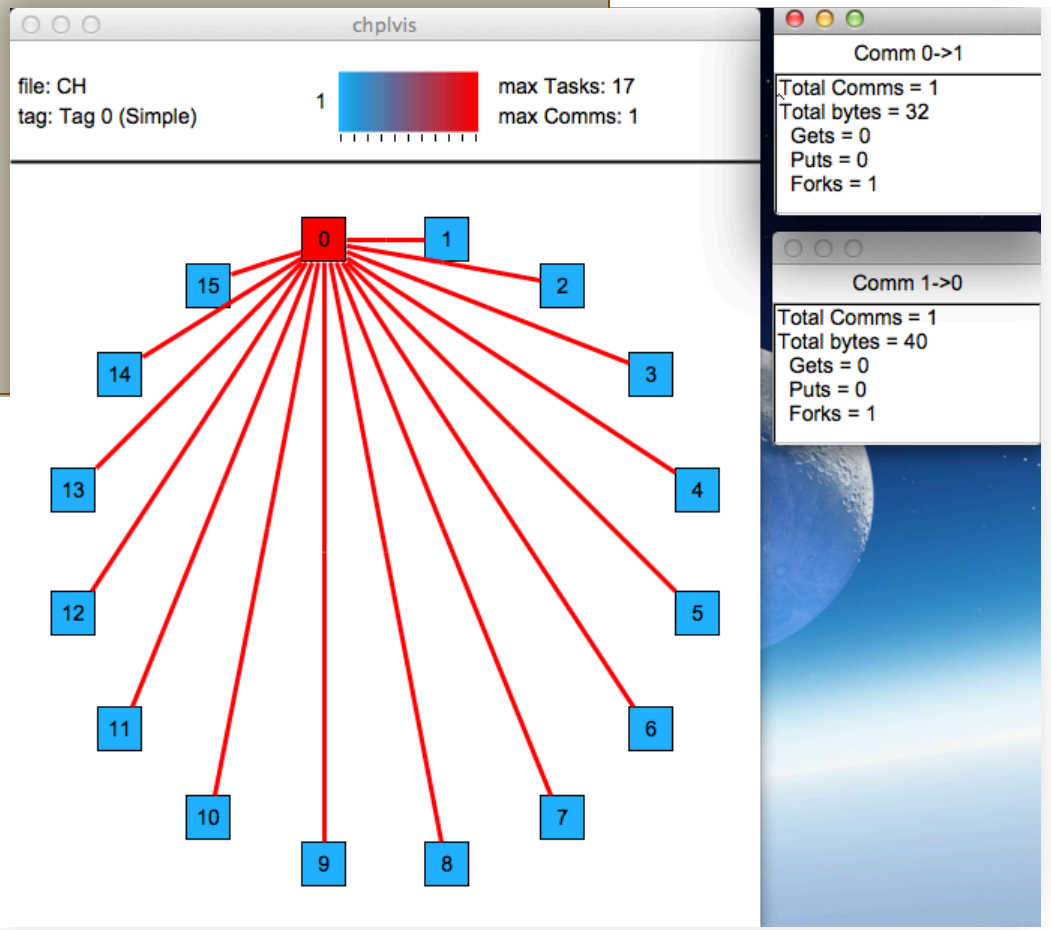
- write aerospace tools for designing, scheduling, analyzing real-time networks
  - use C/C++, MPI, & Java in production
  - experimenting with Chapel in spare time
- developing a **Chapel IDE** using Eclipse Xtext
  - https://bitbucket.org/ngmschapel/hu.ngms.chapel/overview

Phil Nelson, Professor of Computer Science, Western Washington University

- developing **chplvis,** a tool for visualizing Chapel performance
- traces tasking and communication events
  - identified by tagged program phases
- visualizes intensities after execution

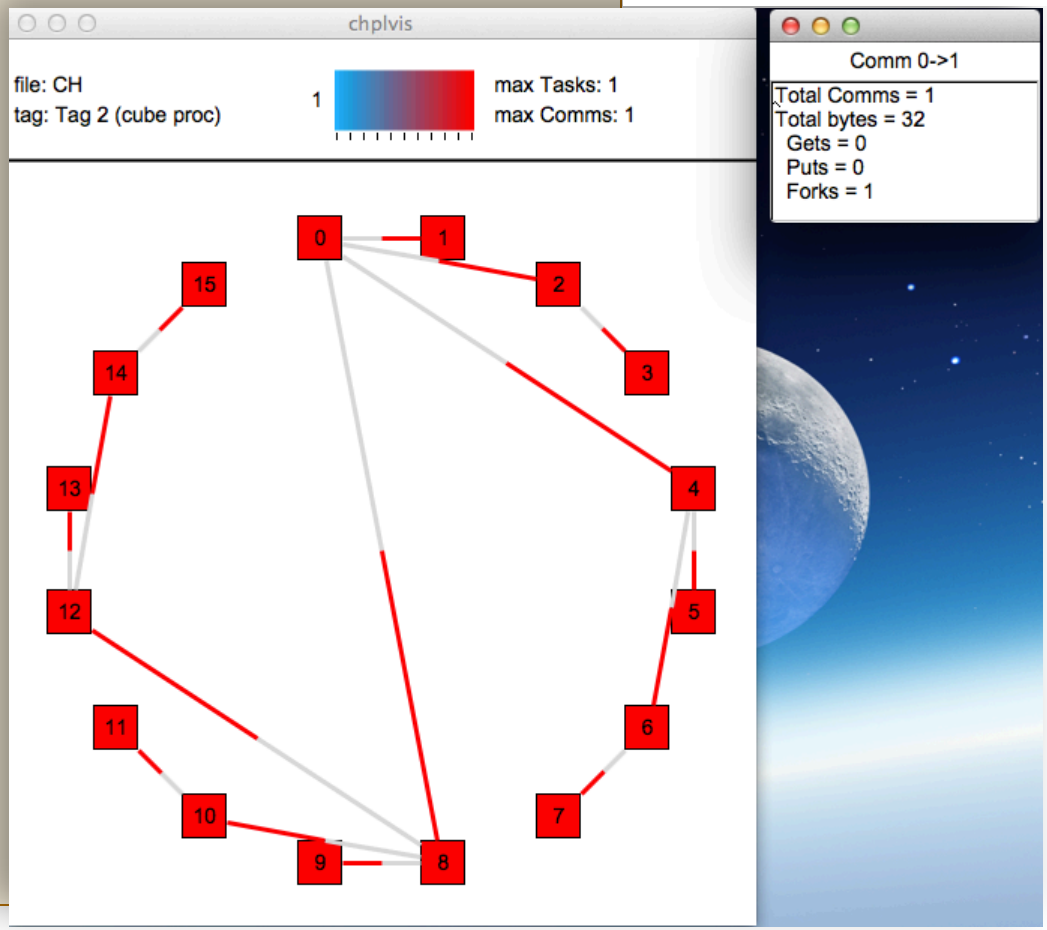
chplvis: One-to-all Example (Nelson)

```
const OnePerLoc = LocaleSpace dmapped Block(space);
var A: [OnePerLoc] int;


coforall loc in Locales do
  on loc do
    A[here.id] = here.id;
```



chplvis: Binary Hypercube Example (Nelson)

```
proc SetBHcube(n = numLocales, id = 0, offset = -1)  {
  var off = 1;
  if (offset < 0) then
    while (off*2 + id < n) do
      off = off << 1;
  else
    off = offset;

  if (id + off < n ) then
    while (off > 0) {
      on Locales[id + off] do
        SetBHcube(n, id+off,
                  off >> 1);
      off = off >> 1;
    }
  A[id] = here.id;
}
```



Summary

- **Chapel is improving by leaps and bounds**
  - features
  - performance
  - documentation
  - development process
- **Community interest also seems to be growing rapidly**

Next Steps

- **Firm up language core**
  - strings
  - memory management
  - error-handling
- **Continue performance and scalability improvements**
  - switch focus to primarily be on multi-locale / distributed memory
  - also validate the quality of our vectorization relative to hand-coded C
- **Continue expanding standard libraries**
  - focus on linear algebra and math capabilities
- **Improve architectural mappings**
  - optimize mapping for NUMA nodes
  - map well to Intel Xeon Phi
- **Continue outreach and collaboration efforts**

CHIUW 2016

- **We expect there to be a CHIUW 2016** (details TBD)
  - Interested in helping to organize? Let us know!
  - Before then, look for Chapel events at SC15 in Austin, TX
- **Sometime today, please fill out survey to help tune CHIUW**

CHIUW 2015: Today's Schedule

8:30: Chapel Boot Camp (optional)

9:00: Welcome, State of the Project

9:30: **Technical Talks** (chair: Mauricio Breternitz, AMD)

11:00: **Break**

11:30: **Technical Talks**

12:30: **Lunch (ad hoc)**

2:00: **Keynote: Bill Carlson (IDA)**

"Shared Memory HPC Programming: Past, Present, and Future"

3:00: **Technical Talks**

3:30: **Break**

4:00: **Hot Topics Talks**

5:00: **Community Discussion**

6:00: **Dinner (ad hoc)**

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/