CRAY
THE SUPERCOMPUTER COMPANY

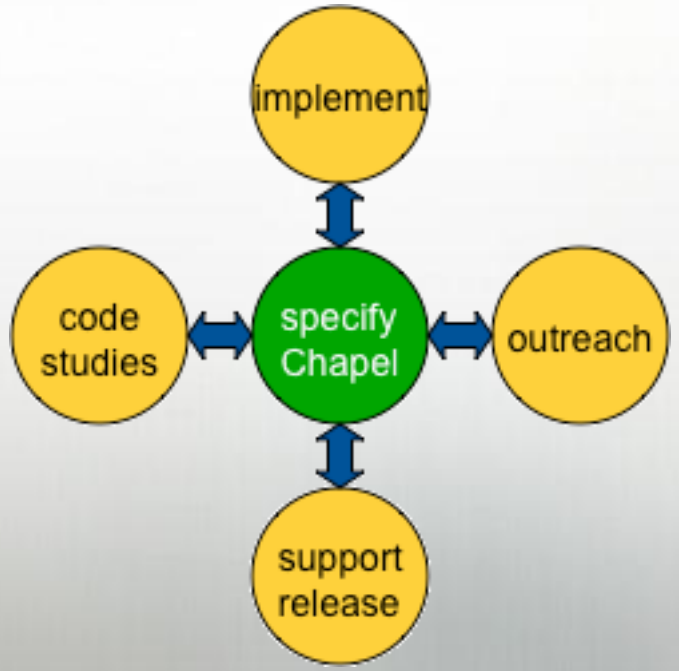
Chapel: Project Overview

Outline

- What we do
- Who we are
- What's next?

Chapel Work

- Chapel Team's Focus:
  - specify Chapel syntax and semantics
  - implement open-source prototype compiler for Chapel
  - perform code studies of benchmarks, apps, and libraries in Chapel
  - do community outreach to inform and learn from users/researchers
  - support collaborators and users of code releases
  - refine the language based on all these activities



Implementation Status -- Version 1.4.0

In a nutshell:

- Most features work at a functional level
- Many performance optimizations remain

This is a good time to:

- Try out the language and compiler
- Give us feedback to improve Chapel
- Use Chapel for parallel programming education
- Use Chapel for non-performance-critical projects

In evaluating the language:

- Try to judge it by how it should *ultimately* perform rather than how it does today
  - lots of low-hanging fruit remains, as well as some challenges

Chapel and Education

- If we were teaching parallel programming, we'd want to cover:
  - data parallelism
  - task parallelism
  - concurrency
  - synchronization
  - locality/affinity
  - deadlock, livelock, and other pitfalls
  - performance tuning
  - …
- We don't think there's a good language out there…
  - for teaching *all* of these things
  - for teaching some of these things well at all
  - ***until now:*** We believe Chapel can potentially play a crucial role here

"I Like Chapel, how can I help?"

- **Let people know that you like it and why**
  - your colleagues
  - your employer/institution
  - Cray leadership (stop by the Cray booth this week)
- **Help us evolve it from prototype to production**
  - contribute back to the source base
  - collaborate with us
  - help fund us to grow the team
  - help us get from "How will Cray make Chapel succeed?" to "How can we as a community make Chapel succeed?"

Join Our Team

- Cray:


Brad Chamberlain


Sung-Eun Choi


Greg Titus


Vass Litvinov


Tom Hildebrandt

- External Collaborators:


Albert Sidelnik
(UIUC)


Jonathan Turner
(CU Boulder)

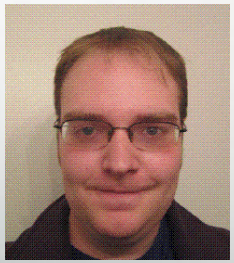

Kyle Wheeler
(Sandia)

You? Your Friend/Student/ Colleague?

- Interns:


Jonathan Claridge
(UW)


Hannah Hemmaplardh
(UW)


Andy Stone
(Colorado State)


Jim Dinan
(OSU)


Rob Bocchino
(UIUC)


Mackale Joyner
(Rice)

Featured Collaborations (see chapel.cray.com/collaborations.html for details)

- **Tasking using Qthreads:** Sandia (Rich Murphy, Kyle Wheeler, Dylan Stark)
  - **paper at CUG, May 2011**
- **Interoperability using Babel/BRAID:** LLNL (Tom Epperly, Adrian Prantl, et al.)
  - **paper at PGAS, Oct 2011**
- **Dynamic Iterators:**
- **Bulk-Copy Opt:**
- **Parallel File I/O:**

  U Malaga (Rafael Asenjo, Maria Angeles Navarro, et al.)
  - **paper at ParCo, Aug 2011**
- **Improved I/O & Data Channels:** LTS (Michael Ferguson)
- **CPU-GPU Computing:** UIUC (David Padua, Albert Sidelnik, Maria Garzarán)
  - **tech report, April 2011**
- **Interfaces/Generics/OOP:** CU Boulder (Jeremy Siek, Jonathan Turner)
- **Tasking over Nanos++:** BSC/UPC (Alex Duran)
- **Tuning/Portability/Enhancements:** ORNL (Matt Baker, Jeff Kuehn, Steve Poole)
- **Chapel-MPI Compatibility:** Argonne (Rusty Lusk, Pavan Balaji, Jim Dinan, et al.)

Collaboration Ideas (see chapel.cray.com/collaborations.html for details)

- memory management policies/mechanisms
- dynamic load balancing: task throttling and stealing
- parallel I/O and checkpointing
- exceptions; resiliency
- language interoperability
- application studies and performance optimizations
- index/subdomain semantics and optimizations
- targeting different back-ends (LLVM, MS CLR, …)
- runtime compilation
- library support
- tools: debuggers, performance analysis, IDEs, interpreters, visualizers
- database-style programming
- (your ideas here…)

Chapel Team's Next Steps

- Continue to improve performance
- Continue to add missing features
- Expand our set of data distributions
- Expand the set of codes that we are studying
- Expand the set of architectures that we can target effectively
- Support the public release
- Continue to support collaborations and seek out new ones
- Continue to expand our team

- Determine Chapel's future after HPCS ends (October 2012)

Chapel 5-year Plan: Key Components

- **Advisory Board**
  - help steer Chapel team's priorities on a regular basis
    - performance vs. features vs. a mix of both
    - which optimizations and features to prioritize
    - which benchmarks, idioms to focus on
- **Agile milestones rather than *a priori***
  - dynamically react to community's needs, R&D challenges
- **Improve openness of project, transition to community**
- **Unified Chapel reporting**
  - rather than reporting to several programs, Chapel is the program
  - reduces reporting burden, permitting team to focus more on work
  - brings those interested in Chapel to a single meeting

Chapel at SC11 (see chapel.cray.com/events.html for details)

- **Mon:** full-day tutorial
- **Mon:** 2nd annual CHUG happy hour/meet-up
- **Tues:** "HPC Challenge" BoF (12:15-1:15)
- **Wed:** "Chapel Lightning Talks" BoF (12:15-1:15)
- **Thurs:** "Punctuated Equilibrium at Exascale" Panel (5:30-7:00)
- **Fri:** half-day tutorial

- **T-Th:** Chapel posters in PGAS booth, Chapel team members staffing (T 2-4, W 10-12, W 4-6, Th 10-12)

Questions?

- What we do
- Who we are
- What's next?