Parallel Programming in Chapel: The Cascade High-Productivity Language

Chapel Team, Cray Inc.

SC10: November 15\textsuperscript{th}, 2010
What is Chapel?

- A new parallel programming language
  - Design and development led by Cray Inc.
  - Initiated under the DARPA HPCS program

**Overall goal:** Improve programmer productivity
- Improve the *programmability* of parallel computers
- Match or beat the *performance* of current programming models
- Support better *portability* than current programming models
- Improve the *robustness* of parallel codes

- A work-in-progress
Chapel's Implementation

- Being developed as open source at SourceForge
- Licensed as BSD software
- **Target Architectures:**
  - multicore desktops and laptops
  - commodity clusters
  - Cray architectures
  - systems from other vendors
  - (more recently: CPU+GPU hybrids)
Today's Goals

- Introduce you to the Chapel language in-depth
- Give you experience...
  ...using the Chapel compiler
  ...writing Chapel code
- Point you toward resources for future reference
- Get your feedback on Chapel
Our team for this tutorial

Cray Employees:
- Brad Chamberlain
- Sung-Eun Choi
- David Iten
- Vass Litvinov

Past Interns/Collaborators:
- Jonathan Claridge (UW Amath)
- Albert Sidelnik (UIUC CS)
Who Are You?

**Type of Institution?**
- Academic, Industry, HPC Lab, Gov’t, ...

**Role?**
- Student, postdoc, faculty, developer, researcher, ...

**Favorite Languages?**
- Fortran, C, C++, Java, Matlab, Python, Perl, C#, ...

**Parallel Programming Models?**
- MPI, OpenMP, Co-Array Fortran, UPC, pthreads, ...
Ground Rules

- Please ask questions as we go
- Also feel free to ask questions of any of us during the breaks, meals, and hands-on sessions
This Morning You Should Receive

1. A Cray USB Stick with...
   - the final tutorial slides
   - the Chapel release
   - today’s hands-on exercises

2. A Chapel-specific survey on the tutorial and language
   - please complete during breaks/hands-on
     - return to us by the end of the day
   - Note: SC10 also has a survey you should complete today
     - return these to the student volunteers
Today's Plan

8:30 – Welcome
8:40 – Background
9:00 – Base Language
9:45 – Data Parallelism
10:00 – Break
10:30 – Data Parallelism
11:00 – Hands-On I
12:00 – Lunch

1:30 – Task Parallelism
1:55 – Locales
2:20 – Domain Maps
2:40 – Project Overview
3:00 – Break
3:30 – Sample Codes
4:00 – Hands-On II
4:50 – Wrap-up
5:00 – Done!
5:30ish – Happy Hour/Meet-Up