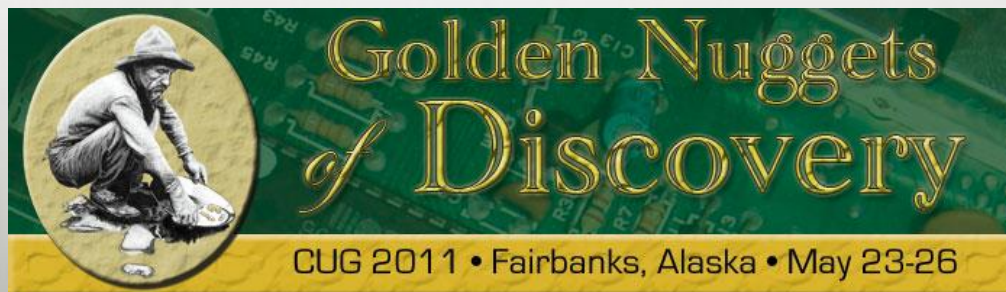


An Overview of the Chapel Parallel Programming Language

Brad Chamberlain, Chapel Team, Cray Inc.

CUG2011: May 23, 2011



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
 - (in-progress: CPU+accelerator hybrids)

Today's Goals

- Introduce you to the Chapel language
 - project themes and status
 - central language concepts
- Demonstrate Chapel in an interactive manner
- Point you toward resources for future reference
- Get your feedback on Chapel

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 me questions throughout CUG
(I'll be here M-W)

Today's Plan

8:00 – Welcome

8:10 – [Background](#)

8:30 – Base Language*

9:00 – Data Parallelism*

9:25 – Break?

9:30 – Task Parallelism*

10:00 – Locality*

?:?? – Project Status (as time permits)

10:30 – Done!

* = I'll try to wrap up these sections with a live demonstration

Today's Plan

8:00 – Welcome

8:10 – Background

8:30 – Base Language*

9:00 – Data Parallelism*

9:25 – Break?

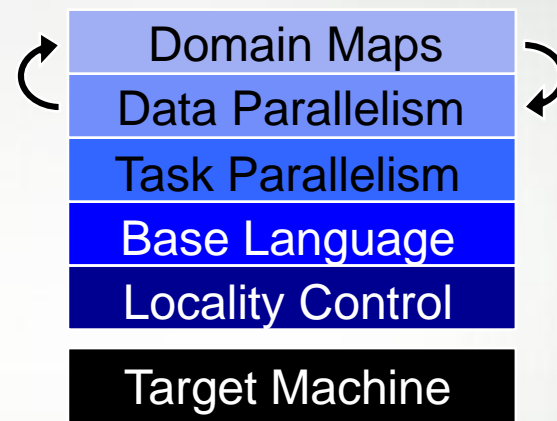
9:30 – Task Parallelism*

10:00 – Locality*

?:?? – Project Status (as time permits)

11:00 – Done!

Chapel language concepts



* = I'll try to wrap up these sections with a live demonstration