General Description

Chapel is an emerging parallel programming language whose design and development is being led by Cray Inc. Chapel is being developed as an open-source effort with contributions from academia, industry, and scientific computing centers. Chapel emerged from Cray's entry in the DARPA-led High Productivity Computing Systems program (HPCS).

Chapel is designed to improve the productivity of high-end computer users while also serving as a portable parallel programming model that can be used on commodity clusters or desktop multicore systems. Chapel strives to vastly improve the programmability of large-scale parallel computers while matching or beating the performance and portability of current programming models like MPI.

Chapel supports a multithreaded execution model via high-level abstractions for data parallelism, task parallelism, concurrency, and nested parallelism. Chapel's locale type enables users to specify and reason about the placement of data and tasks on a target architecture in order to tune for locality. Chapel supports global-view data aggregates with user-defined implementations, permitting operations on distributed data structures to be expressed in a natural manner. In contrast to many previous higher-level parallel languages, Chapel is designed around a multiresolution philosophy, permitting users to initially write very abstract code and then incrementally add more detail until they are as close to the machine as their needs require. Chapel supports code reuse and rapid prototyping via object-oriented design, type inference, and features for generic programming.

Chapel was designed from first principles rather than by extending an existing language. It is an imperative block-structured language, designed to be easy to learn for users of C, C++, Fortran, Java, Perl, Matlab, and other popular languages. While Chapel builds on concepts and syntax from many previous languages, its parallel features are most directly influenced by ZPL, High-Performance Fortran (HPF), and the Cray MTA?/Cray XMT? extensions to C and Fortran.

Compiling Chapel Code

To compile the code in multi-locale mode, one needs to load the multi-locale version of Chapel by typing

\texttt{module load chapel/1.5.0-multi.}

To compile, type the following in the directory containing the source code:

\texttt{chpl -o executable\_name source\_code.chpl}

To compile the code in single locale mode, one needs to load the single locale version of Chapel by typing

\texttt{module load chapel} or \texttt{module load chapel/1.5.0-single.} To compile, one uses the same command as listed above.

\textbf{Note:} One CANNOT compile the code in one mode and run it in a different mode. For example, one
Chapel

CANNOT compile in single locale mode and attempt to run the code with the -nl flag. Similarly, one CANNOT compile the code in multi-locale and attempt to run it without defining the number of locales one wants to use with the -nl flag.

Basic Usage

- Submitting a Chapel job to SGE

Further Information

See the official website: Chapel