Hewlett Packard
Enterprise

CHAPEL 1.25 RELEASE NOTES: LIBRARY IMPROVEMENTS

Chapel Team

September 23, 2021

OUTLINE

- Argument Parser Library
- Ordered Map Library
- LinearAlgebra Library Improvements

ARGUMENT PARSER LIBRARY

ARGUMENT PARSER LIBRARY
Background and This Effort

Background:

- Chapel supports configuration variables, but they are not always sufficient
  - No support for supplying a list of values (e.g., '--files file1 file2 file3 file4')
  - No support for positional arguments or sub-commands (e.g., './myChapelProgram setup inputfile')
- Chapel also supports accepting command-line arguments to 'main()'
  - Requires developers to sanitize and convert complex inputs and to do their own validation

This Effort: Provide an argument parser library to help with arguments passed to 'main()'

- Allow for flags, options, positional arguments, sub-commands, and pass-through arguments
- Can be used in combination with, or independent of, configuration variables
- Perform validation on the number of values, required flags/options, sub-command names
- Relieve developer from checking for properly formatted input

ARGUMENT PARSER LIBRARY

Example Usage

This program accepts a list of filenames to process, as well as an optional debug flag.

```
$ ./myChapelDemo --debug ~/file.txt ~/dirname/file2.txt file3.txt
$ ./myChapelDemo file1.txt file2.txt
```

```
use ArgumentParser;
proc main(args: [] string){                                      // 'main' needs to be defined to accept arguments
  var parser = new argumentParser();                             // create a parser object
  var debugFlag = parser.addFlag(name="debug",                   // add a debug flag
                                 defaultValue=false);
  var myFiles = parser.addArgument(name="files",                 // accept one or more filenames
                                   numArgs=1..);
  try! { parser.parseArgs(args); }                               // try to parse command line input
  catch ex: ArgumentError { exit(1); }                           // parser will throw an error on invalid input
  if debugFlag.valueAsBool() then …                              // check if debug mode is specified
  for filename in myFiles.values() do …                          // process all the files
}
```

ARGUMENT PARSER LIBRARY

Status and Next Steps

Status: Available as a package module starting this release

- Developer can define flags, options, positional arguments and sub-commands
- Offer long and short options for flags and options
- Most of ‘mason’ refactored to utilize ArgumentParser
- See more example usage and docs at chapel-lang.org/docs/main/modules/packages/ArgumentParser.html

Next Steps: Continue to add new features.

- Provide a standard help message listing available options/flags/arguments
- Improve error-handling
- Constrain option values
- Conditionally require/exclude other arguments

ORDERED MAP LIBRARY

ORDERED MAP LIBRARY

Background and This Effort

Background

- Chapel has many data structure implementations
  - Standard Modules: List, Set, Map, Heap
  - Package Modules: DistributedBag, DistributedDeque, LinkedList, OrderedSet, UnrolledLinkList

This Effort

- 1.25 introduces the ‘OrderedMap’ package module

- Implemented as a Google Summer of Code 2020 project
  - Student: Yujia Qiao
  - Mentors: Krishna Kumar Dey (Chapel GSoC 2019 Alum), Paul Cassella, Engin Kayraklioglu

ORDERED MAP MODULE

Impact

- ‘orderedMap’ can be used to store key-value associations with the keys in sorted order

```
use OrderedMap;
var m = new orderedMap(int, int);
for (randomInt, count) in zip(someRandomIntStream(), 1..) do
  m.add(randomInt, count);
for (key, value) in m.items() do
  writeln("Key: ", key, " Value: ", value);   // print items sorted by key
```

- Different comparators can be used to order keys

```
var m = new orderedMap(int, int, comparator=myComparator);
```

- Enable parallel-safety by setting the ‘parSafe’ param to true

```
var m = new orderedMap(int, int, parSafe=true);
```

- See ‘OrderedMap’ documentation: chapel-lang.org/docs/modules/packages/OrderedMap.html

ORDERED MAP LIBRARY

Next Steps

- Should the module be named ‘SortedMap’?
  - “Ordered” may imply the order of insertion
  - See issue #18449

LINEAR ALGEBRA LIBRARY IMPROVEMENTS

LINEAR ALGEBRA LIBRARY IMPROVEMENTS

Background and This Effort

Background:

- LinearAlgebra library created in 1.15 release for high-level linear algebra operations and procedures
  - Includes matrix and vector operations
  - Some operations were missing

This Effort:

- Added ‘sinm()’, ‘cosm()’, and ‘sincos()’ routines to compute sines and cosines of square matrices
- Added ‘expm()’ to compute exponentials of square matrices
- Enabled ‘dot()’ to multiply sparse and dense matrices, and vice versa

$$\begin{bmatrix} 1 & 3 & \\ & 2 & 1 \\ & 3 & \end{bmatrix} \bullet \begin{bmatrix} 3 & 1 & 2 \\ 8 & 7 & 5 \\ 2 & 2 & 1 \end{bmatrix}$$

- Implemented as a Google Summer of Code 2021 project
  - Student: Prasanth Duvvuri
  - Mentors: Nikhil Padmanabhan (Yale), Lydia Duncan, Engin Kayraklioglu

LINEAR ALGEBRA LIBRARY IMPROVEMENTS

Impact and Next Steps

Impact:

- Matrix functionality has been extended to support more common cases

Next Steps:

- Merge support for estimating 1-norms of a matrix (PR #18149)
  - A 1-norm of a square matrix is the maximum of the absolute column sums
  - E.g., the following matrix has a 1-norm of 11
    - Column 3's absolute column sum is 11 and the other columns sum to 10 and 8

$$\begin{bmatrix} 1 & 3 & -7 \\ -3 & 2 & 2 \\ 6 & 3 & 2 \end{bmatrix} = 11$$

  - Normally this computation is O(n^2), but estimating can lower that to O(k*N) time
- Merge support for finding the action of a matrix's exponential (PR #18293)
  - Avoids the cost of computing the matrix's exponential when combining with vector or another matrix

OTHER LIBRARY IMPROVEMENTS

OTHER LIBRARY IMPROVEMENTS

For a more complete list of library changes and improvements in the 1.25 release, refer to the following sections in the CHANGES.md file:

- 'Name Changes in Libraries'
- 'Deprecated / Removed Library Features'
- 'Standard Library Modules'
- 'Package Modules'
- 'Performance Optimizations / Improvements'
- 'Documentation'
- 'Portability'
- 'Bug Fixes for Libraries'

THANK YOU

https://chapel-lang.org
@ChapelLanguage