

Ruby on HPC

What is Ruby?

Ruby is a high-level, interpreted, object-oriented programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan. Ruby is known for its simplicity and readability, making it a popular choice for beginner programmers. However, it also provides many advanced features for experienced developers, including metaprogramming, dynamic typing, and garbage collection.

Ruby has a strong emphasis on code readability and high-level abstractions, which makes it well-suited for building web applications, especially using the Ruby on Rails framework. Ruby on Rails is a full-stack framework for building web applications that provides a high level of abstraction and a "convention over configuration" philosophy, making it easier to build and maintain complex web applications.

In recent years, Ruby has become increasingly popular, especially in the start-up and tech communities, and has been used to build many successful web applications, such as GitHub, Shopify, and Airbnb.

Links:

[Official Website](#)

[Tutorial](#)

Versions Available:

The following versions are available on the cluster:

- ruby 2.1.2

How to load Ruby?

To load Ruby, use the following commands:

```
#Load the Ruby module  
module load ruby/2.1.2
```

To verify if the module is loaded correctly, use the following command,

```
# List all the module loaded in the environment  
module list
```

In a fresh environment, this only loads Ruby module without any dependencies.

How to use Ruby?

Ruby can be used effectively for scientific computations, especially for prototyping and data analysis.

Here are some tips for using Ruby in HPC for scientific computations:

1. Use optimized libraries: There are several optimized libraries available for Ruby that can be used for scientific computations. For example, Numo::NArray is a fast numerical array library for Ruby that provides optimized numerical computation capabilities.
2. Parallelize your code: Ruby provides several libraries and tools for parallelizing your code, such as Parallel and concurrent-ruby. These libraries can be used to distribute your computations across multiple CPU cores or computers, making your code run faster.

3. Optimize your code: Optimize your code for performance by profiling it and eliminating bottlenecks. You can use the built-in Ruby profiler to determine which parts of your code are slow, and then optimize those parts.
4. Use a multi-language approach: If necessary, you can use Ruby to prototype your scientific computations, and then translate your code into a lower-level language, such as C or Fortran, for performance-critical parts of your code. This can give you the best of both worlds: the ease of development in Ruby, combined with the performance of a lower-level language.

To demonstrate the usage of the ruby library, here is a sample parallel matrix multiplication,

```
require 'parallel'

# An array of matrices
matrices = [
  [[1, 2], [3, 4]],
  [[5, 6], [7, 8]],
  [[9, 10], [11, 12]]
]

# The size of the matrices
size = matrices.first.size

# The product of the matrices
product = Array.new(size) { Array.new(size, 0) }

# Use parallel processing to calculate the product of the matrices
Parallel.each(0...size, in_threads: size) do |i|
  (0...size).each do |j|
    (0...size).each do |k|
      product[i][j] += matrices[i][k] * matrices[k][j]
    end
  end
end

puts "The product of the matrices is: #{product}"
```

To install parallel library, use gem package manager,

```
gem install parallel
```

Here is a slurm script to submit the job,

```
#!/bin/bash
# Request 1 node and 2 tasks
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --p main
#SBATCH --qos main
# Request 2 CPUs per task
#SBATCH --cpus-per-task=2
# Request 2GB of memory per task
#SBATCH --mem-per-task=2GB
# Set the job name
#SBATCH --job-name=parallel_ruby
# Set the output file
#SBATCH --output=parallel_ruby.out

# Load the Ruby module
module load ruby/2.1.2

# Run the parallel Ruby script
ruby parallel_script.rb
```

Where to find help?

If you are confused or need help at any point, please contact OIT at the following address.

<https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp>

