How to use the Julia programming language on the HPC Prince cluster

How to start using Julia on NYU HPC clusters:

- **Step 1:** Login to prince

  From your terminal on your desktop/laptop, run:

  ```
  ssh <your net id>@prince.hpc.nyu.edu
  ```

  Output:

  ![Login output]

  For more extensive instructions on how to login on the HPC Prince cluster, read the following wiki page.

- **Step 2:** Check for julia modules

  Once you are logged on the Prince cluster, find out what versions of Julia are installed, by get a listed of all Julia software modules that are installed.

  ```
  log-1 ~]$ module spider julia
  ```

  Output:
For more extensive instructions on the usage of module commands, read the following wiki page.

- **Step 3:** Load the version of Julia that you want to use

Select the version of Julia you would like to use. Usually the latest version (in our case 1.1.0) is the one most users pick:

```
log-1 ~]$ module load julia/1.1.0
```

- **Step 4-1:** Run Julia in julia terminal

```
log-1 ~]$ julia
```

output:

```
| __|
| | |
| |___| |
| |   |
| |   |
| |___| |
|___|
```

Documentation: https://docs.julialang.org

Type "?" for help, "?]" for Pkg help.

Version 1.1.0 (2019-01-21)

Official https://julialang.org/ release

you can exit julia terminal by running the following code:
• **step 4-2: use julia command to run code**

    # copy julia sample code to current directory
    log-1 ~]$ cp /share/apps/examples/julia/test.jl .
    # run julia code
    log-1 ~]$ julia ./test.jl

    ```julia
    n = 10000
    a = SharedArray(Float64, n, n);
    @sync @parallel for j in 1:size(a,2)
        for i in 1:size(a,1)
            a[i,j] = min(i,j)
        end
    end
    b = SharedArray(Float64, n);
    @sync @parallel for i in 1:n;
        b[i] = sum(a[i,:])
    end
    for i in 1:2000:n;
        @printf "%d %f\n" i b[i]
    end
    ```

• **step 4-3 using julia with slurm**

  ```julia
  sample_code.jl
  log-1 ~]$ cat /share/apps/examples/julia/test.jl
  ```

  ```julia
  n = 10000
  a = SharedArray(Float64, n, n);
  @sync @parallel for j in 1:size(a,2)
      for i in 1:size(a,1)
          a[i,j] = min(i,j)
      end
  end
  b = SharedArray(Float64, n);
  @sync @parallel for i in 1:n;
      b[i] = sum(a[i,:])
  end
  for i in 1:2000:n;
      @printf "%d %f\n" i b[i]
  end
  ```

• **copying batch files from shared folders**

    # copy julia sample code from example folder to current directory
    log-1 ~]$ cp /share/apps/examples/julia/ ./
```bash
#!/bin/sh
#SBATCH --time=00:15:00
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=1
# the resources requested above must be within the allocation
# we need to load the julia module so that the paths are set up right.
module load julia

# this starts the julia script which will srun its own processes
julia test.jl
```

For more extensive instructions on the using Slurm, read the following wiki page.

**Additional Helpful Resources:**

New to Julia? Here's a full scale tutorial for you on Julia's official website:

https://docs.julialang.org/en/v1/manual/getting-started/

Here are some great online training resources for you to jump start your projects(including youtube videos and books):

https://julialang.org/learning/

PS: for some of the tutorials, Jupyter notebook is required. Jupyter notebook is installed on the clusters and you can read more about them here.

Not what you were looking for? contact hpc@nyu.edu for more information on Julia and other information.