Perl on HPC

What is Perl?

Perl is a high-level, general-purpose, interpreted programming language originally developed by Larry Wall in 1987. It is often used for text processing, system administration, web development, and other tasks that require a powerful and flexible scripting language. Perl has a large and active community, and many useful libraries and modules are available for it. It is known for its powerful regular expression handling, and its ability to manipulate text and data in a variety of ways.

In high-performance computing (HPC), Perl can be used for a variety of tasks such as scripting, automation, and data processing. Its powerful text processing capabilities make it well-suited for tasks such as parsing log files, manipulating large datasets, and generating reports. Additionally, Perl's ability to handle parallel processing and multi-threading can make it useful for certain types of HPC workloads, such as Monte Carlo simulations and other scientific computations. However, it is important to note that it is not as popular as other languages such as C, C++, Fortran, Python, etc in HPC.

Links:

Official Website

Manual

Versions Available:

The following versions are available on the cluster:

- perl/5.22.1
- perl/5.22.1threaded

How to load Perl?

To load Perl, use the following commands:

```
#Load the Perl module
module load perl/5.22.1threaded
```

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 should only load perl since it is an independent programming language.

How to use Perl?

Perl can be used in several ways:

- **Scripting:** Perl's powerful text processing capabilities make it well-suited for tasks such as parsing log files, manipulating large datasets, and generating reports.
- **Automation:** Perl can be used to automate many of the repetitive tasks that are often required in HPC environments, such as job scheduling, resource management, and data transfer.
- **Data Processing:** Perl can be used to process large data sets and generate reports. Perl's ability to handle parallel processing and multi-threading can make it useful for certain types of HPC workloads, such as Monte Carlo simulations and other scientific computations.
- **Wrapper Programs:** Perl can be used to create wrapper programs for other HPC tools that can be used to automate complex workflows.

Here is a sample script demonstrating the use of perl language,

```
use strict;
use warnings;
use Parallel::ForkManager;

my $pm = new Parallel::ForkManager(5); # 5 is the number of parallel
processes

# Print numbers in parallel
for my $i (1..10) {
    $pm->start and next; # fork a new process

    # Print the number
    print "Process $$: $i\n";

    $pm->finish; # Terminate the child process
}

$pm->wait_all_children;
```

This script uses the **Parallel::ForkManager** module to print the numbers from 1 to 10 in parallel, using 5 parallel processes. It forks a new process for each number and prints the number in the child process. The script also prints the process ID (PID) so that you can see which process printed each number. Once the printing is completed, the child process is terminated. Finally, the script waits for all child processes to complete before exiting.

To submit the job to the scheduler, use the following slurm script,

```
#!/bin/bash
#SBATCH --job-name=parallel_print    # Job name
#SBATCH -p main             # Partition (queue)
#SBATCH --qos main
#SBATCH --ntasks=1                   # Number of tasks/cores
#SBATCH --cpus-per-task=5            # Number of cores per task
#SBATCH --mem=1024MB                 # Memory per node
#SBATCH --time=00:05:00              # Time limit hrs:min:sec
#SBATCH --output=parallel_print.out  # Standard output and error log

module load perl/5.22.1threaded

srun perl scipt.perl
```

To learn more about perl, visit the official website.

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