

## ***Moose on HPC***

### **What is Moose?**

MOOSE (Multiphysics Object Oriented Simulation Environment) is an open-source simulation framework for solving multiphysics problems. It is written in C++ and is designed to solve large-scale, complex, nonlinear systems of coupled partial differential equations using finite element methods. MOOSE is particularly well-suited for problems involving nonlinear, time-dependent, multiphysics phenomena in complex geometries. The MOOSE framework allows for the easy creation of custom physics modules, and it has been used to simulate a wide range of problems in fields such as nuclear engineering, materials science, and geophysics.

Links:

[Official Website](#)

[Usage Documentation](#)

### **Versions Available:**

The following versions are available on the cluster:

- Moose

### **How to load Moose?**

To load MOOSE, use the following commands:

```
#Load the MOOSE module  
module load physical/moose
```

MOOSE is designed such that each user must maintain their own copy of the MOOSE code. The full size of the code, when all the basic programs are built (i.e. libmesh) is about 5 G. User home directory quotas are 50 G so this will easily fit.

Instructions for downloading and building moose, which are discussed here, can also be accessed at:

[https://mooseframework.inl.gov/getting\\_started/installation/install\\_moose.html](https://mooseframework.inl.gov/getting_started/installation/install_moose.html)

It is cloned using GIT from the central MOOSE repository and can be updated when desired. Start by creating a projects subdirectory. The best place for this is your home directory, since that is backed-up nightly. The tilde (~) in these instructions is a shortcut for your home directory name. If you want the projects directory somewhere else, cd to that directory and leave off "~/".

```
# Create Project Directory
mkdir ~/projects
cd ~/projects
```

Next you will download the MOOSE code and make sure that you are pointed to the stable master branch.

```
git clone https://github.com/idaholab/moose.git
cd moose
git checkout master
```

You will then need to download and compile your own copy of petsc as well as the libMesh component. First, make sure you are still in the moose subdirectory, then run the script. This can take quite a long time to finish.

```
cd ~/projects/moose
./scripts/update_and_rebuild_petsc.sh
./scripts/update_and_rebuild_libmesh.sh
```

Next, it is best to run the test code to make sure all is built correctly:

```
cd ~/projects/moose/test
make -j 2
./run_tests -j 2
```

At this point you should be ready to create your own projects. There is a script (stork.sh) that will create the framework for your application. This is discussed in the "Getting Started" information.

```
https://mooseframework.inl.gov/getting\_started/index.html
```

Note that you will want to update your installation periodically. Information on how to do that is in the "install\_moose" web page referenced above. You will need to recompile your projects if you run an update on MOOSE.

## How to use Moose?

When a project is initiated with stork.sh it should put in the framework to compile the application. The customization is up to the user and is beyond the scope of this document.

The project is compiled with "make" from the project directory. So, if the project is call myApp, for instance

```
cd ~/projects/myApp
make -j 2
```

At this point you should not run it on the master node. You should write an sbatch file to run the program.

```
#!/bin/bash
#SBATCH -n 12
#SBATCH -p main
```

```
#SBATCH --qos main
cd ~/projects/myApp
{put command to run moose application here}
```

Submit this sbatch file (example here assumed to be "mysbatch") to SLURM with sbatch mysbatch

### ***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>

