Accessing Software

- Policies
- Installed software
- Modules
- Installing additional software
- Using and Customizing Perl
- Using and Customizing Python
- Using and Customizing R Packages
- Using Matlab
- Using Anaconda
- Spark
- Compilers

Policies

The policies regarding the installation of software is on this page. In general, scientific software is installed as requested with the caveats noted in that section.

Installed software

A list of installed software is kept at this page.

On Ocelote, many libraries like FFTW and MPICH came with the Cluster Manager. As a consequence, they have different naming from the other clusters. For example, module load blas on the older clusters will need to be module load blas/gcc/64. There are frequently versions of these libraries that have several compilation options, for example, blas was created with four different compilers on Ocelote.

Modules

Many popular software packages are installed and available as modules. There may be several versions of a package available.

<table>
<thead>
<tr>
<th>Module Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>module avail</td>
<td>Display all the software and versions installed on the system</td>
</tr>
<tr>
<td>module list</td>
<td>Display the software you have loaded in your environment</td>
</tr>
<tr>
<td>module load modulename</td>
<td>Load a software module in your environment</td>
</tr>
<tr>
<td>module purge</td>
<td>Unload all the software modules from your environment</td>
</tr>
<tr>
<td>module unload modulename</td>
<td>Unload a specific software package from your environment</td>
</tr>
<tr>
<td>module help</td>
<td>Display a help menu for the module command</td>
</tr>
</tbody>
</table>

Installing additional software

To submit a request to have software installed on the UA HPC systems use the HPC Software Install Request form: [https://it.arizona.edu/hpc-software-install-request](https://it.arizona.edu/hpc-software-install-request)

There is no expected timeframe for how long it takes to install software. There are many variables. If you haven't heard back in a week, it is reasonable for you to follow up with [hpc-consult@list.arizona.edu](mailto:hpc-consult@list.arizona.edu)
You can install software packages into your home directories with the space that is allocated to you with your HPC account. However, you cannot install software that requires root permission, or use a method like "yum install" that accesses system paths.

Follow this link for detailed information on how to install your own software.

**Using and Customizing Perl**

Follow this link for more information on using Perl.

**Using and Customizing Python**

Follow this link for more information on using Python.

**Using and Customizing R Packages**

You can install your own R packages which is similar to using virtualenv with Python:

1. Make directory to store packages
   
   ```bash
   $ mkdir -p ~/R/library
   ```

2. Tell R where the directory is by creating an environment file:
   
   ```bash
   $ echo 'R_LIBS=~/R/library/' >> ~/.Renviron
   ```

3. For example to install and load the package "ggplot2":
   
   ```bash
   $ module load R
   $ R
   ...
   > install.packages("ggplot2")
   > library(ggplot2)
   ```

4. After this you'll only need the library command to load your custom package

5. For more information:
   

**Using Matlab**

There are three ways to run Matlab:

1. The command line version using modules. This is the most common as you will typically submit a job using PBS.
2. Graphical mode using the Ocelote Desktop of OnDemand

Here are the details for method 1.

MATLAB performs its own hardware discovery and it might try to access all the cores and the memory of the node even if the full node wasn't allocated. That will result in scheduler killing the job. To prevent that the full Ocelote node of 28 cores and 168GB of memory should be allocated to run a MATLAB job.

Like any other application, MATLAB has to be loaded as a module before you can use it. To see all the installed versions of the MATLAB use command `module avail matlab`.

The typical procedure for performing calculations on UA HPC systems is to run your program non-interactively on compute nodes. The easiest way to run MATLAB non-interactively is to use input/output redirection. This method uses Linux operators < and > to point MATLAB to the input file and tell where to write the output (see the example script). The other method is to invoke MATLAB from the PBS script execute specified statement using `-r` option. For details please refer to the manual page of `matlab` command:

#!/bin/bash
#PBS -N job_name
#PBS -W group_list=group_name
#PBS -q standard
#PBS -l select=1:ncpus=28:mem=168gb:pcmem=6gb
#PBS -l walltime=01:00:00
#PBS -l cput=28:00:00

cd ${PBS_O_WORKDIR}
module load matlab
matlab -nodisplay -nosplash < script_name.m > output.txt

The options `-nodisplay` and `-nosplash` in the example prevent MATLAB from opening elements of GUI. To view the full list of options for `matlab` command load the MATLAB module and type `matlab -h` in Linux prompt, or use the link above to the manual page on MathWorks website.

**Using Anaconda**

Anaconda is available as a module and comes with Glueviz, JupyterLab, Jupyter Notebook, Spyder, Orange and RStudio. It also supports the conda commands for building a personal environment.

More details are at this page.

**Spark**

Apache Spark is a fast and general-purpose cluster computing system. However it has not been installed for support in a multi-node environment as yet. That functionality is planned for the future. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming.

There are some simple tests here.

**Compilers**

- **El Gato**

  El Gato is now running CentOS 7. The compilers available are:
  - Intel/2019.3: "module load intel"
  - GCC 5.4.0: "module load gnu"
  - GCC 8.3.0: "module load gnu8"
  - LLVM 5.0.1: "module load llvm5"

- **Ocelote**

  GCC is available without loading a module. gcc --version shows that it is 6.1.0. If you also need the GNU Scientific Libraries (gsl), that is available using `module load gsl` which will get you version 2.1
  - The Intel Compiler suite 2016 is available in both 32 and 64 bit versions. The math kernel libraries (mkl) are provided as separate modules, also in 32 and 64 bit versions.
  - The Intel Compiler 2019 is available and is no longer separated like 2016. So there are not separate modules for mkl, mpi, etc.
  - PGI compilers 2018 (Portland Group) are installed as a module. We have provided these in particular for their OpenACC support. We do not build software with the PGI compilers.
  - MPI. There are more choices now so pay attention.
    - There are standard Red Hat versions of mpich, mvapich, mvapich2 and openmpi. Some extra options are invoked when you load one - use `module avail` to see the specific name.
    - The same four MPI are available with more detailed options for gcc, intel and open64. Again use `module avail` for the appropriate choices.
  - AVX2 Ocelote has Intel V3 Haswell processors. A key feature of these is AVX2. Read this Intel document.
For AVX2 support, compile with the -xHOST option. Note that -xHOST alone does not enable aggressive optimization, so compilation with -O3 is also suggested. The -fast flag invokes -xHOST, but should be avoided since it also turns on interprocedural optimization (-ipo), which may cause problems in some instances.

For GNU compilers, AVX support is only available in version 4.6 or later. For AVX support, compile with -mavx.