ARC Advanced Research Computing

Introduction to HPC

Presented by: Jacob Boschee and Venkat Mahadevan

arc.ubc.ca

Summary of Topics

1. Course Prerequisites and Interactive Examples
2. Allocations
    a. Sharing Allocation Resources
    b. Lifecycles
3. Scheduling
    a. Job Submission
    b. Resource Requests
    c. Fairshare
4. Software Modules
    a. Loading an Environment
    b. Heirarchy
    c. Commands
5. Storage
    a. Home
    b. Project
    c. Scratch
    d. Local Scratch

Prerequisites

- Familiarity with command line shells
- Connecting to a system via SSH
- Editing files in terminal

Allocations

- Basic structure of accounting and assigning resources
- Often comprised of multiple members of a research team
- Groups usage of a system together to allow easier collaboration between members
- Can be used to organize system usage for reporting metrics

Allocations – Sharing Resources

- All members of an allocation have equal usage of the same resources
  - Storage, compute, and GPUs
- Many systems have reporting systems to check usage
  - [Sockeye Example: print_quota command]
- Communication between allocation members can be key to efficiently sharing

Allocations - Lifecycles

- Most allocations on HPC systems have a term
- Often terms can be extended by reapplication
- Users and allocation managers should have an end of term plan.
    - This often involves setting up methods for migrating research data and storing it
- Similarly plans should be made for when researchers leave the allocation

Scheduling

- Backbone of an HPC system
- Resources are allocated to jobs and jobs are run non-interactively from a queue
- Works to make the most efficient and fairly distributed use of available resources
- Multiple resource schedulers are available with their own unique interfaces
    - Sockeye uses OpenPBS
    - Compute Canada systems use SLURM

Scheduling – Job Submission

- Jobs are submitted to a queue for execution
- Users can view their own active and pending jobs
- Variations depending on type of scheduler used
- [Example of Job submission Demo]

Scheduling – Resource Requests

- Jobs must contain instructions for the scheduler
  - Amount of time needed, Memory limitations, Number of CPUs and other resources
- Systems often place limitations on the maximum duration and number of resources requested
- Requesting resources does not necessarily mean your software will utilize them
  - Ensure your software can properly scale to the resources you request

Scheduling - Fairshare

- To balance use of the system the scheduler can assign a ranking to allocations
- This allows users who have less use of a system to more rapidly access resources when there is contention
- The ranking is determined based on previously requested resources
  - Note: This is considered on requested resources not utilized resources
- As this is often balanced across allocations ensure you speak with other members about your usage if it will impact their ability to start jobs.

Software Modules

- Modules are configuration files which modify your software environment.
- The purpose is to make pre-installed software easily available to the user.
- Module files contain instructions to modify environment variables such as PATH and LD_LIBRARY_PATH to allow various installed software to run correctly.

Software Modules

- On the Sockeye HPC, for example, many kinds of software packages are provided.
- Programming languages: C/C++ compilers, Fortran, Java, Matlab
- Parallel libraries: OpenMPI, OpenMP
- GPU development: CUDA
- Many others: https://arc.ubc.ca/list-available-software-sockeye

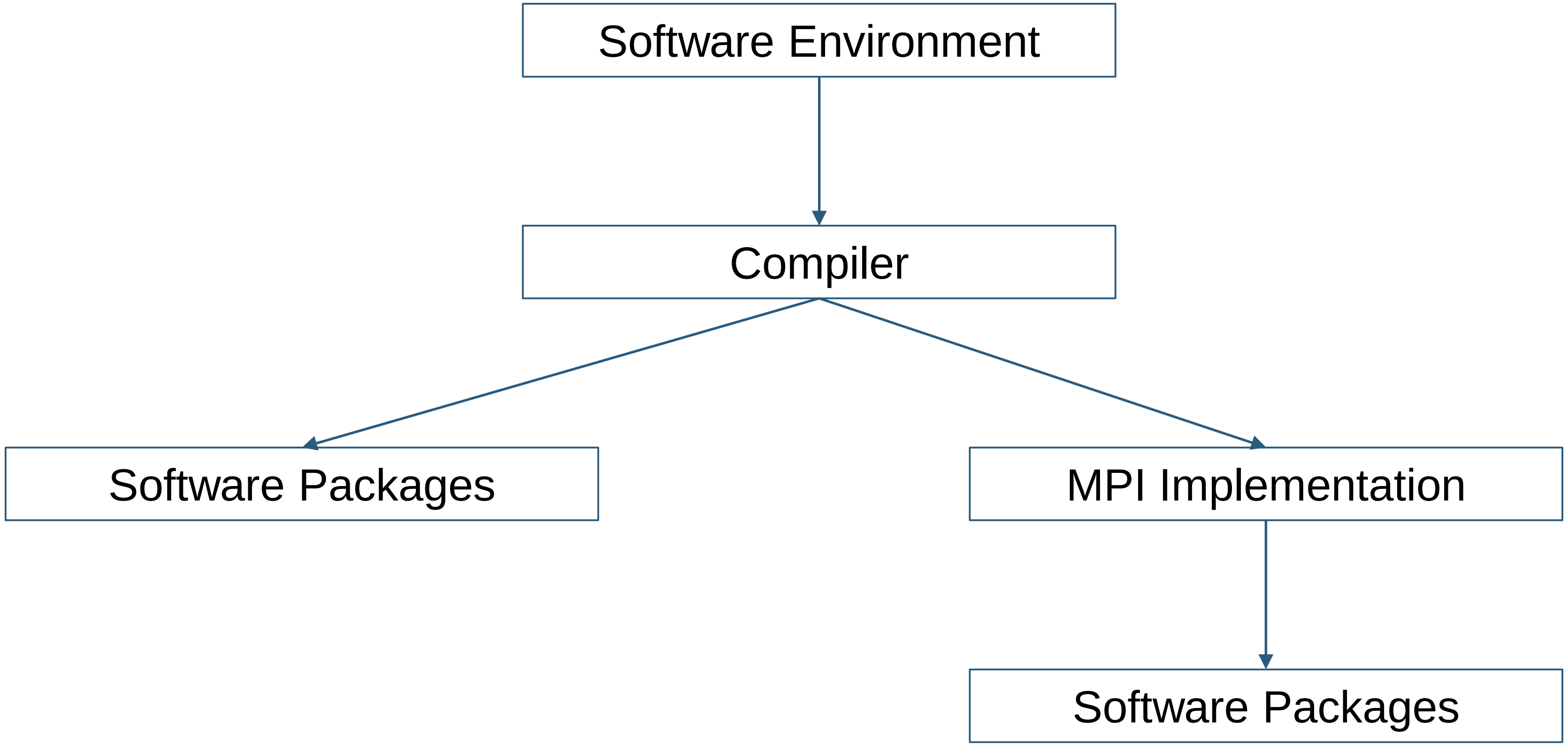
Software Modules – Loading Environments

- On Sockeye, there are a couple of different software environments which load a pre-determined set of modules.

- default-environment: current environment loaded by default.

- Sockeye_2021_Software: new environment, will become the default upon completion of testing.

Module Hierarchy



Software Environment

Compiler

Software Packages

MPI Implementation

Software Packages

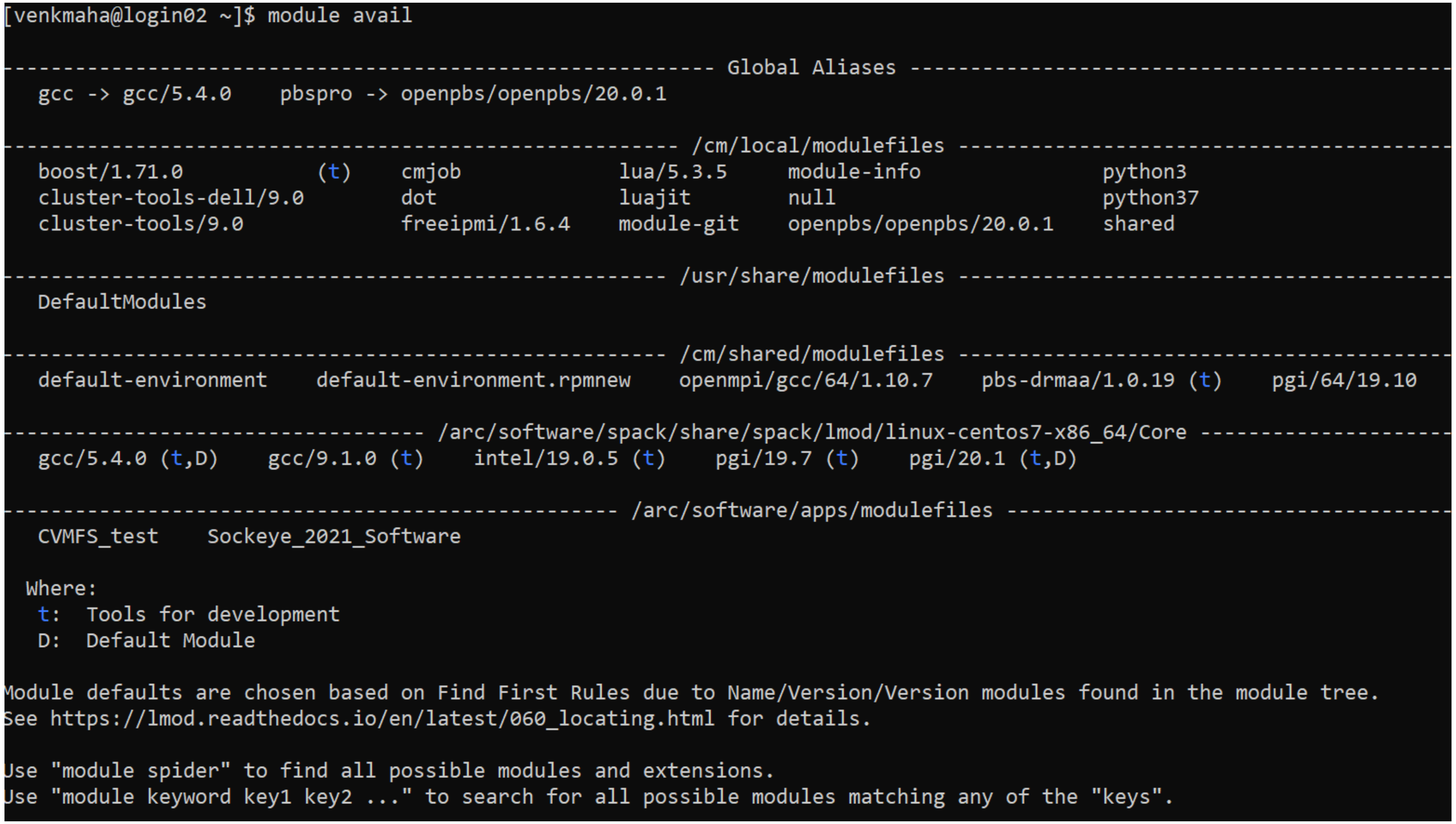
Software Modules - Commands

- module avail <name> → list available modules
- module spider <name> → provide detailed info. about modules and versions
- module list →list loaded modules
- module load <name> → load the module
- module unload <name> → unload the module
- module show <name> → show the detailed commands

Software Modules – Commands

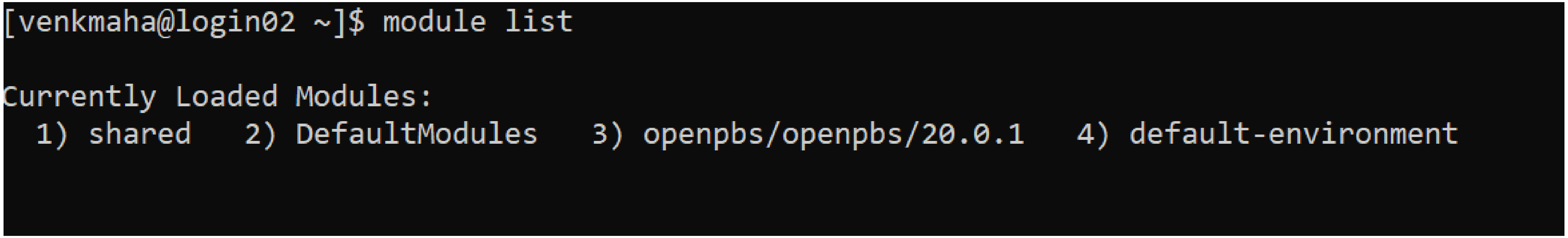
- Modules must be loaded before a job is submitted.
- Or they can be loaded in the submission script.
- Pre-requisite modules for a module will also be loaded automatically.
- It is not advised to load modules in your .bashrc; load them as required.

Exercises – list available modules using “module avail”



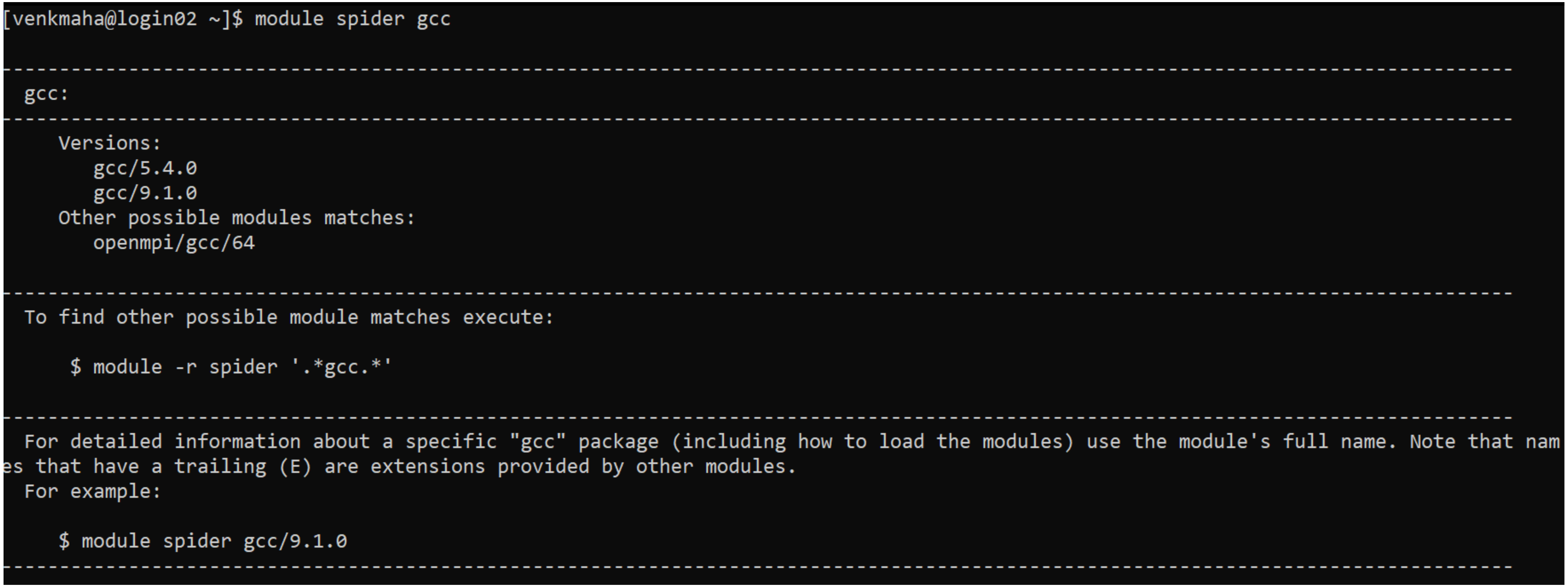
```
[venkmaha@login02 ~]$ module avail

----------------------------------------------------------- Global Aliases --------------------------------------------
   gcc -> gcc/5.4.0    pbspro -> openpbs/openpbs/20.0.1

-------------------------------------------------------- /cm/local/modulefiles ----------------------------------------
   boost/1.71.0           (t)    cmjob             lua/5.3.5     module-info               python3
   cluster-tools-dell/9.0        dot               luajit        null                      python37
   cluster-tools/9.0             freeipmi/1.6.4    module-git    openpbs/openpbs/20.0.1    shared

------------------------------------------------------- /usr/share/modulefiles ----------------------------------------
   DefaultModules

------------------------------------------------------- /cm/shared/modulefiles ----------------------------------------
   default-environment    default-environment.rpmnew    openmpi/gcc/64/1.10.7    pbs-drmaa/1.0.19 (t)    pgi/64/19.10

----------------------------------- /arc/software/spack/share/spack/lmod/linux-centos7-x86_64/Core --------------------
   gcc/5.4.0 (t,D)    gcc/9.1.0 (t)    intel/19.0.5 (t)    pgi/19.7 (t)    pgi/20.1 (t,D)

--------------------------------------------------- /arc/software/apps/modulefiles ------------------------------------
   CVMFS_test    Sockeye_2021_Software

  Where:
   t:  Tools for development
   D:  Default Module

Module defaults are chosen based on Find First Rules due to Name/Version/Version modules found in the module tree.
See https://lmod.readthedocs.io/en/latest/060_locating.html for details.

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

Exercises – show currently loaded modules using “module list”



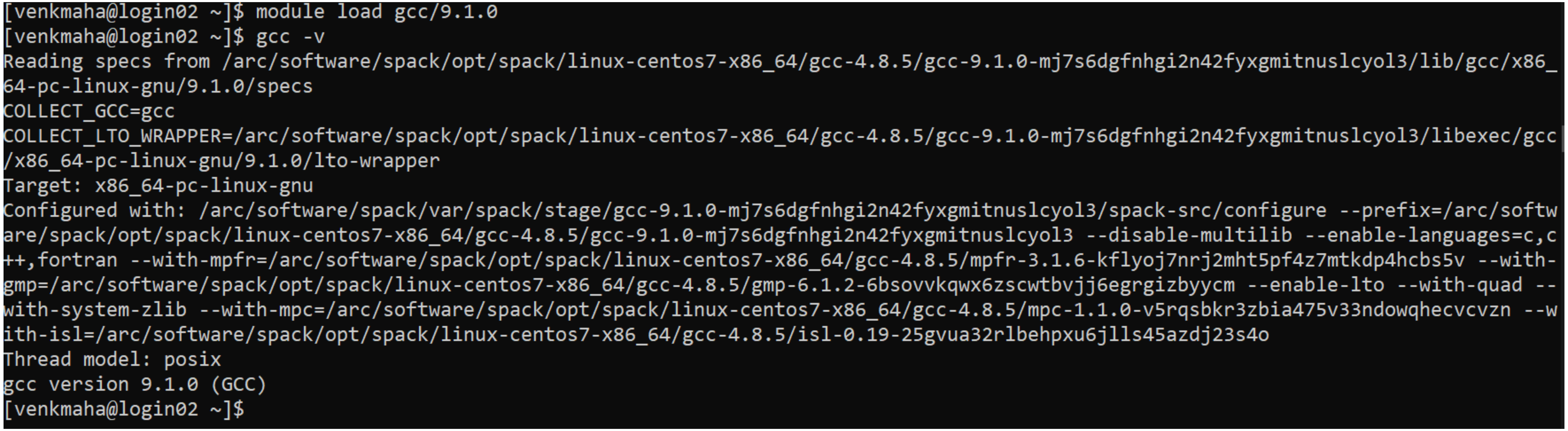
```
[venkmaha@login02 ~]$ module list

Currently Loaded Modules:
  1) shared   2) DefaultModules   3) openpbs/openpbs/20.0.1   4) default-environment
```

Exercises – search for versions of gcc



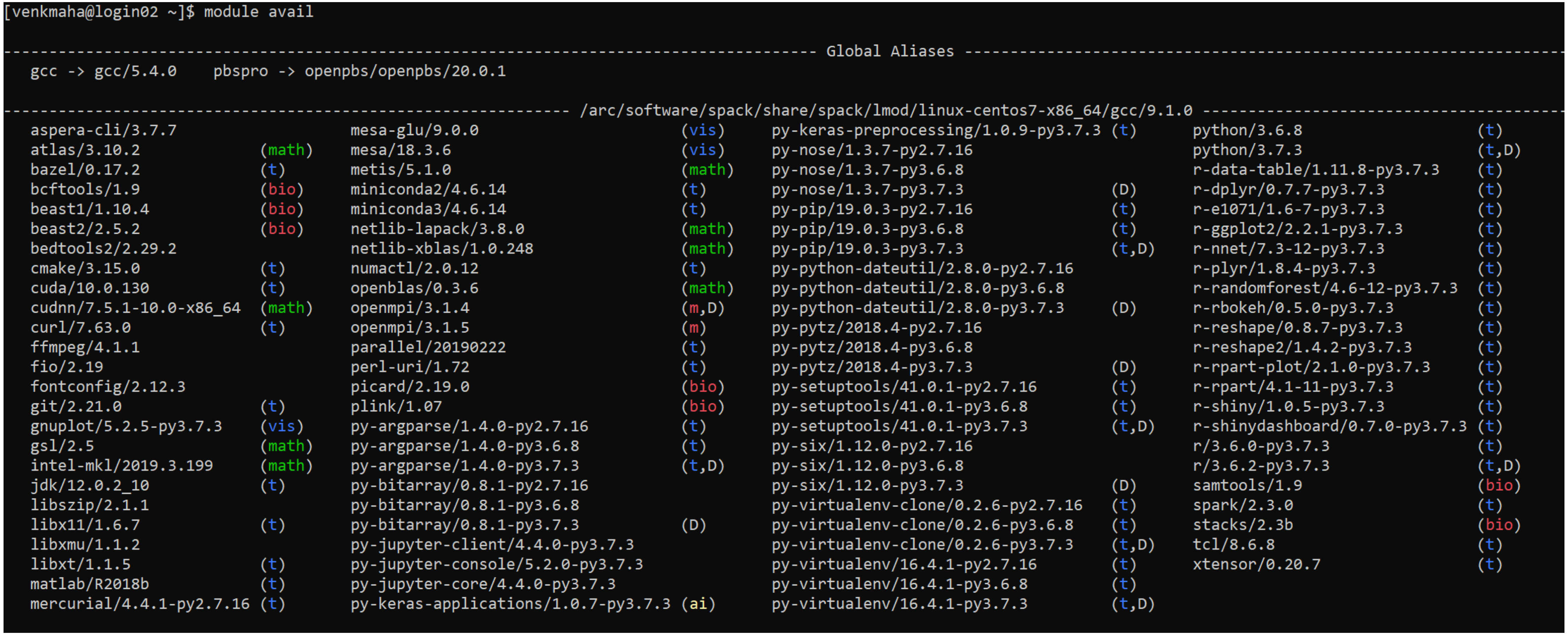
```
[venkmaha@login02 ~]$ module spider gcc

-----------------------------------------------------------------------------------------------------------------------------------
  gcc:
-----------------------------------------------------------------------------------------------------------------------------------
     Versions:
        gcc/5.4.0
        gcc/9.1.0
     Other possible modules matches:
        openmpi/gcc/64

-----------------------------------------------------------------------------------------------------------------------------------
  To find other possible module matches execute:

      $ module -r spider '.*gcc.*'

-----------------------------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "gcc" package (including how to load the modules) use the module's full name. Note that nam
es that have a trailing (E) are extensions provided by other modules.
  For example:

     $ module spider gcc/9.1.0
-----------------------------------------------------------------------------------------------------------------------------------
```

Exercises – load the gcc compiler “module load gcc/9.1.0”



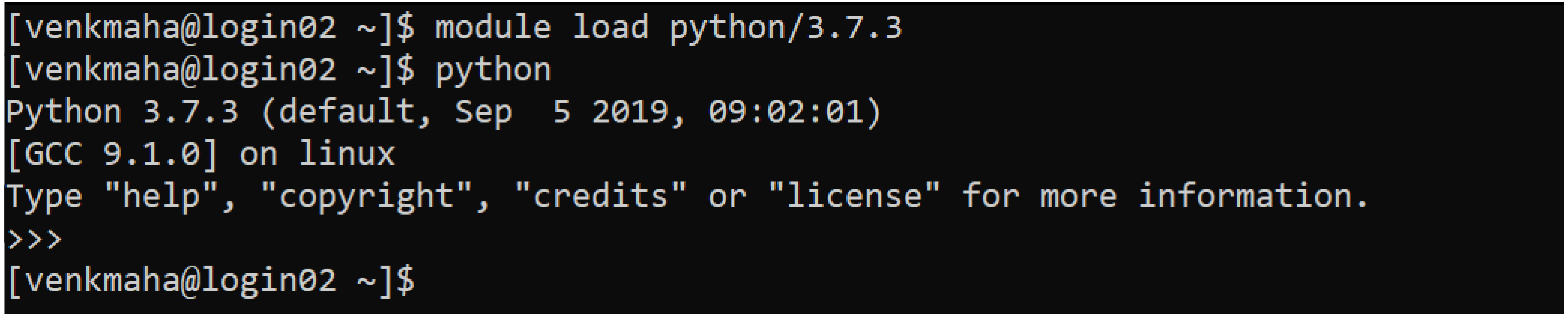
```
[venkmaha@login02 ~]$ module load gcc/9.1.0
[venkmaha@login02 ~]$ gcc -v
Reading specs from /arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/gcc-9.1.0-mj7s6dgfnhgi2n42fyxgmitnuslcyol3/lib/gcc/x86_
64-pc-linux-gnu/9.1.0/specs
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/gcc-9.1.0-mj7s6dgfnhgi2n42fyxgmitnuslcyol3/libexec/gcc
/x86_64-pc-linux-gnu/9.1.0/lto-wrapper
Target: x86_64-pc-linux-gnu
Configured with: /arc/software/spack/var/spack/stage/gcc-9.1.0-mj7s6dgfnhgi2n42fyxgmitnuslcyol3/spack-src/configure --prefix=/arc/softw
are/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/gcc-9.1.0-mj7s6dgfnhgi2n42fyxgmitnuslcyol3 --disable-multilib --enable-languages=c,c
++,fortran --with-mpfr=/arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/mpfr-3.1.6-kflyoj7nrj2mht5pf4z7mtkdp4hcbs5v --with-
gmp=/arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/gmp-6.1.2-6bsovvkqwx6zscwtbvjj6egrgizbyycm --enable-lto --with-quad --
with-system-zlib --with-mpc=/arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/mpc-1.1.0-v5rqsbkr3zbia475v33ndowqhecvcvzn --w
ith-isl=/arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/isl-0.19-25gvua32rlbehpxu6jlls45azdj23s4o
Thread model: posix
gcc version 9.1.0 (GCC)
[venkmaha@login02 ~]$
```

Exercises – list available modules



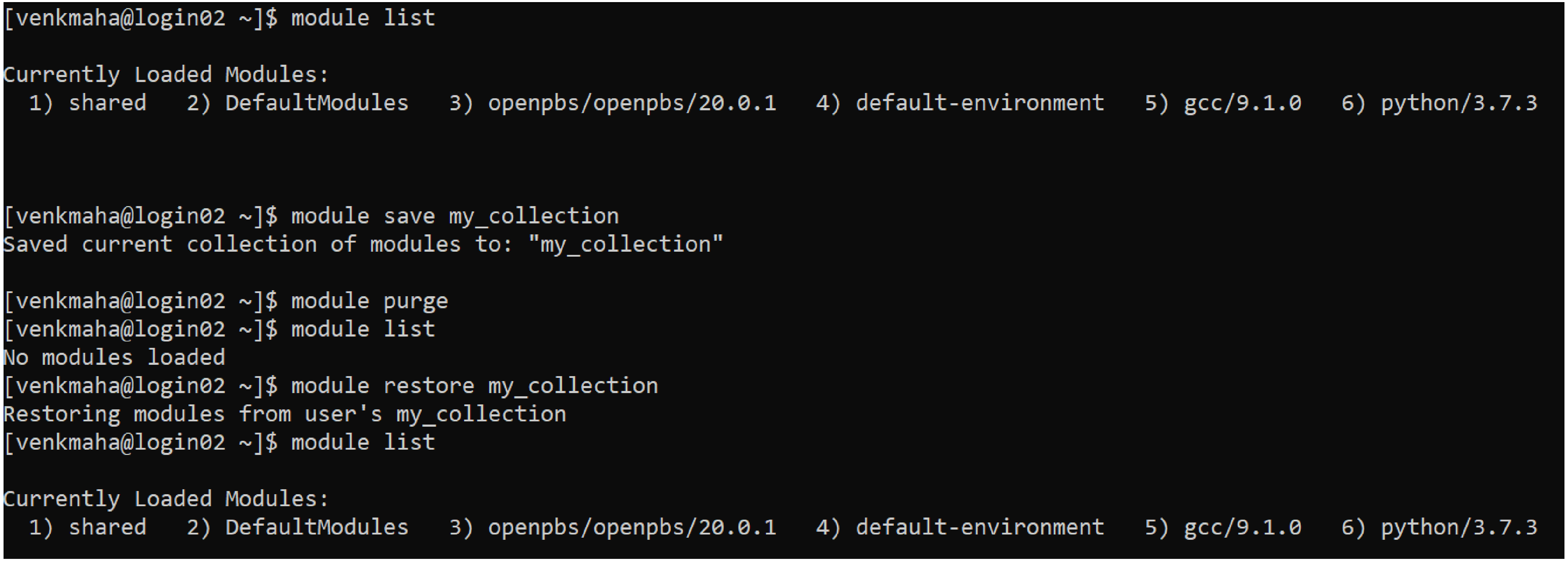
```
[venkmaha@login02 ~]$ module avail

---------------------------------------------------------------------------------------- Global Aliases -----------------------------------------------------------------
   gcc -> gcc/5.4.0    pbspro -> openpbs/openpbs/20.0.1

------------------------------------------------------------- /arc/software/spack/share/spack/lmod/linux-centos7-x86_64/gcc/9.1.0 ----------------------------------------
   aspera-cli/3.7.7                   mesa-glu/9.0.0                      (vis)     py-keras-preprocessing/1.0.9-py3.7.3 (t)      python/3.6.8                   (t)
   atlas/3.10.2             (math)    mesa/18.3.6                         (vis)     py-nose/1.3.7-py2.7.16                        python/3.7.3                   (t,D)
   bazel/0.17.2             (t)       metis/5.1.0                         (math)    py-nose/1.3.7-py3.6.8                         r-data-table/1.11.8-py3.7.3    (t)
   bcftools/1.9             (bio)     miniconda2/4.6.14                   (t)       py-nose/1.3.7-py3.7.3                (D)      r-dplyr/0.7.7-py3.7.3          (t)
   beast1/1.10.4            (bio)     miniconda3/4.6.14                   (t)       py-pip/19.0.3-py2.7.16               (t)      r-e1071/1.6-7-py3.7.3          (t)
   beast2/2.5.2             (bio)     netlib-lapack/3.8.0                 (math)    py-pip/19.0.3-py3.6.8                (t)      r-ggplot2/2.2.1-py3.7.3        (t)
   bedtools2/2.29.2                   netlib-xblas/1.0.248                (math)    py-pip/19.0.3-py3.7.3                (t,D)    r-nnet/7.3-12-py3.7.3          (t)
   cmake/3.15.0             (t)       numactl/2.0.12                      (t)       py-python-dateutil/2.8.0-py2.7.16             r-plyr/1.8.4-py3.7.3           (t)
   cuda/10.0.130            (t)       openblas/0.3.6                      (math)    py-python-dateutil/2.8.0-py3.6.8              r-randomforest/4.6-12-py3.7.3  (t)
   cudnn/7.5.1-10.0-x86_64  (math)    openmpi/3.1.4                       (m,D)     py-python-dateutil/2.8.0-py3.7.3     (D)      r-rbokeh/0.5.0-py3.7.3         (t)
   curl/7.63.0              (t)       openmpi/3.1.5                       (m)       py-pytz/2018.4-py2.7.16                       r-reshape/0.8.7-py3.7.3        (t)
   ffmpeg/4.1.1                       parallel/20190222                   (t)       py-pytz/2018.4-py3.6.8                        r-reshape2/1.4.2-py3.7.3       (t)
   fio/2.19                           perl-uri/1.72                       (t)       py-pytz/2018.4-py3.7.3               (D)      r-rpart-plot/2.1.0-py3.7.3     (t)
   fontconfig/2.12.3                  picard/2.19.0                       (bio)     py-setuptools/41.0.1-py2.7.16        (t)      r-rpart/4.1-11-py3.7.3         (t)
   git/2.21.0               (t)       plink/1.07                          (bio)     py-setuptools/41.0.1-py3.6.8         (t)      r-shiny/1.0.5-py3.7.3          (t)
   gnuplot/5.2.5-py3.7.3    (vis)     py-argparse/1.4.0-py2.7.16          (t)       py-setuptools/41.0.1-py3.7.3         (t,D)    r-shinydashboard/0.7.0-py3.7.3 (t)
   gsl/2.5                  (math)    py-argparse/1.4.0-py3.6.8           (t)       py-six/1.12.0-py2.7.16                        r/3.6.0-py3.7.3                (t)
   intel-mkl/2019.3.199     (math)    py-argparse/1.4.0-py3.7.3           (t,D)     py-six/1.12.0-py3.6.8                         r/3.6.2-py3.7.3                (t,D)
   jdk/12.0.2_10            (t)       py-bitarray/0.8.1-py2.7.16                    py-six/1.12.0-py3.7.3                (D)      samtools/1.9                   (bio)
   libszip/2.1.1                      py-bitarray/0.8.1-py3.6.8                     py-virtualenv-clone/0.2.6-py2.7.16   (t)      spark/2.3.0                    (t)
   libx11/1.6.7             (t)       py-bitarray/0.8.1-py3.7.3           (D)       py-virtualenv-clone/0.2.6-py3.6.8    (t)      stacks/2.3b                    (bio)
   libxmu/1.1.2                       py-jupyter-client/4.4.0-py3.7.3               py-virtualenv-clone/0.2.6-py3.7.3    (t,D)    tcl/8.6.8                      (t)
   libxt/1.1.5              (t)       py-jupyter-console/5.2.0-py3.7.3              py-virtualenv/16.4.1-py2.7.16        (t)      xtensor/0.20.7                 (t)
   matlab/R2018b            (t)       py-jupyter-core/4.4.0-py3.7.3                 py-virtualenv/16.4.1-py3.6.8         (t)
   mercurial/4.4.1-py2.7.16 (t)       py-keras-applications/1.0.7-py3.7.3 (ai)      py-virtualenv/16.4.1-py3.7.3         (t,D)
```

Exercises – load python/3.7.3



```
[venkmaha@login02 ~]$ module load python/3.7.3
[venkmaha@login02 ~]$ python
Python 3.7.3 (default, Sep  5 2019, 09:02:01)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
[venkmaha@login02 ~]$
```

Bonus – save and restore your loaded modules from a collection



```
[venkmaha@login02 ~]$ module list

Currently Loaded Modules:
  1) shared   2) DefaultModules   3) openpbs/openpbs/20.0.1   4) default-environment   5) gcc/9.1.0   6) python/3.7.3



[venkmaha@login02 ~]$ module save my_collection
Saved current collection of modules to: "my_collection"

[venkmaha@login02 ~]$ module purge
[venkmaha@login02 ~]$ module list
No modules loaded
[venkmaha@login02 ~]$ module restore my_collection
Restoring modules from user's my_collection
[venkmaha@login02 ~]$ module list

Currently Loaded Modules:
  1) shared   2) DefaultModules   3) openpbs/openpbs/20.0.1   4) default-environment   5) gcc/9.1.0   6) python/3.7.3
```

Storage - Home

- Location for personal files and basic development work
- Should not be used to store active input or output from jobs
- For performance reasons some systems (such as Sockeye) do not allow writing from queued jobs
- Generally a smaller limitation on size than other allocation directories

Storage - Project

- Best location for software installed to be used by software run via the scheduler.
- Input files that do not need to be written to
- Storing final results from workflows.
- Shared among all users within an allocation
- For performance reasons some systems (such as Sockeye) do not allow writing from queued jobs

Storage - Scratch

- Generally the fastest shared storage system on an HPC system
- Primary location for files that need to be written by HPC jobs and input files that are repeatedly read during execution by software.
- As a high turnover file system users are recommended against long-term storage of data in scratch.
    - Some systems may have automated processes that will prune data older than a certain period
- Users are encouraged to move completed outputs to another location on the system once the work is completed

Storage – Local Scratch/Temporary Directories

- For workflows that write many files that are only necessary during the execution
- Some workflows have a very large number of output files that may not be suitable to write to scratch
- Storage space only exists as long as the job is still executing. Upon completion all data in the location is destroyed
- Tends to be the best performance location to write on the system but will require additional steps to ensure data is preserved

ARC Advanced Research Computing

arc.ubc.ca

email: arc.info@ubc.ca