

# UF Research Computing

---

## Environment Modules for trouble-free biocomputing

### PBS Introduction

Oleksandr Moskalenko <om@hpc.ufl.edu>

Matt Gitzendanner <magitz@ufl.edu>

October 2011 UFGI Training Session

# Outline

---

## ◆ Environment Modules

- Need for speed i.e. reason for Env. Modules
- Introduction and basic use
- Module Tree
- Writing modulefiles

## ◆ HPC PBS with modules

- How to use modules in PBS pipelines

---

# Environment Modules

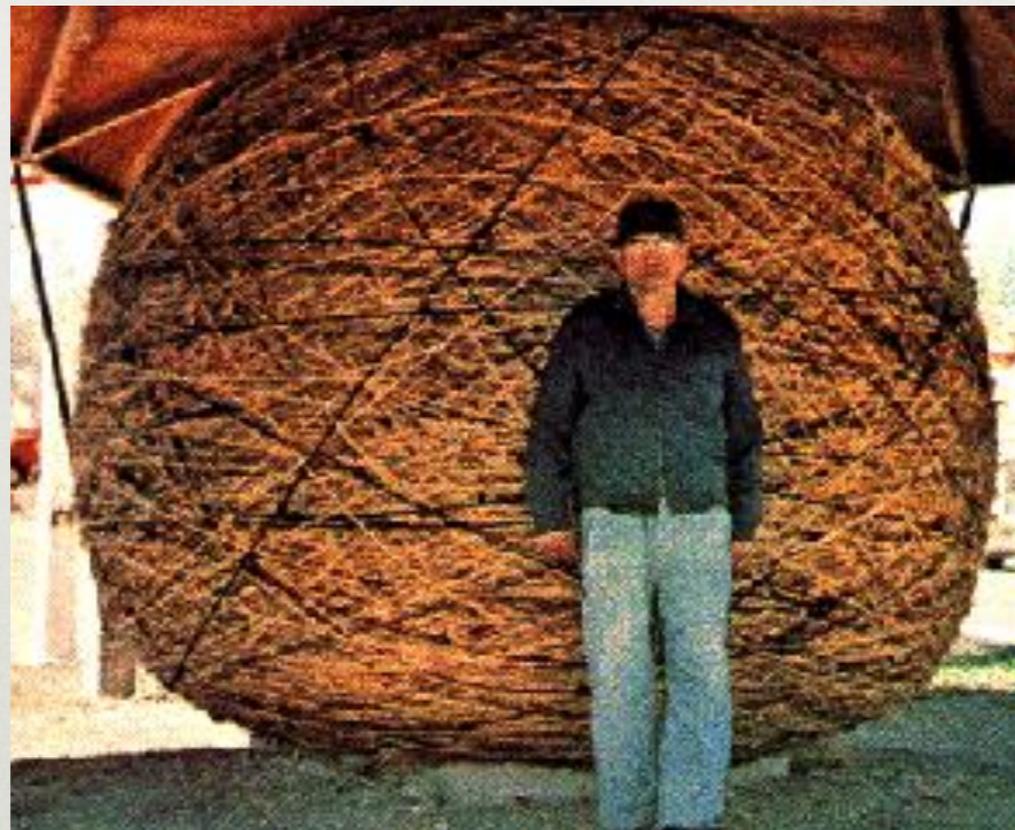
# The biggest ball of twine in...

---

App versions

Library versions

Compilers



ENV vars

Dependencies

Paths

Python versions

MPI implementations

# The Ghost of the CLI

---

- Impossible to maintain many versions
- Environment setup is very complex and must be done by every user
- Things break on every upgrade
- No stable pipelines
- No way to switch Comp / MPI easily
- Hard to find the right application
- Result: the nightmare of CLI

# Modules with benefits

---

- ◆ Users are free of the gory details
- ◆ Many application versions coexist
- ◆ Easy switching of compiler / MPI app trees
- ◆ Easy addition of dependencies
- ◆ Automated environment setup (paths, variables, licenses, etc....) and cleanup
- ◆ Powerful search and help system
- ◆ Stable pipelines

**Focus on Science, not busywork**

# Untangle Compilers/MPIs

---

- ◆ mpich2\_intel-1.0.8
  - ◆ mvapich2\_intel-1.4.1
  - ◆ mvapich2\_intel11-1.5.1
  - ◆ mvapich\_intel-0.9.9
  - ◆ mvapich\_intel10-0.9.9
  - ◆ openmpi\_intel-1.2.7
  - ◆ openmpi\_intel-1.3.4
  - ◆ openmpi\_intel11-1.3.4
  - ◆ openmpi\_intel11-1.4.3
- 2 Compilers  
x  
7 MPI  
implementations  
=  
Barely used beyond  
the default; constant  
support issues

# What is a module?

---

- ◆ Lmod software (R. McClay, TACC)
  - <http://www.tacc.utexas.edu/tacc-projects/mclay/lmod>
- ◆ Shell environment
- ◆ Modulefiles – systemwide and user made
  - /apps/lmod/modulefiles
- ◆ Discover, evaluate, use, swap - DEUS

# Module Tree

---

- ◆ Root of the tree - `/apps/lmod/modulefiles`
  - Core – does not depend on a particular Compiler/MPI combination  
`/apps/lmod/modulefiles/core/clustalw/2.1.lua`
  - Compiler – depends on a particular compiler  
`/apps/lmod/modulefiles/compiler/intel/10_1/mrbayes/3.1.2.lua`
  - MPI – depends on a compiler/MPI combination  
`/apps/lmod/modulefiles mpi/intel10_1/openmpi_1_2/raxml/1.0.5-light.lua`

# Basic usage - Discover

---

- `module spider`
- `module spider clu`
- `module spider netcdf/3.6.3`
- `module avail`
- `module avail clu` (what does it find?)
- `module key clu`

# Discover – module spider

---

weblago: weblago/3.0

generation of sequence logos

weka: weka/3.6.2

WEKA - algorithms for data mining

wgs: wgs/6.1

Celera Whole Genome Shotgun Assembler

wublast: wublast/2.0

Basic Local Alignment Search Tool - WUSTL version

# Discover – module spider clu

---

clustalw: clustalw/2.1

---

Description: multiple sequence alignment software

Help: This module enables the use of the ClustalW software

Clustal is popular multiple sequence alignment software.

ClustalW produces biologically meaningful multiple sequence alignments of divergent sequences. It calculates the best match for selected sequences and lines them up, so identities, similarities, and differences can be identified.

Version 2.1

UF Research Computing/HPC Center

# Discover – module avail

---

----- /apps/lmod/modulefiles/compiler/intel/10\_1 -----

amber/10      metis/4.0      mrbayes/3.1.2      openmpi/1.2.7

fftw/3.2.2      mkl/10.0      mvapich/0.9.9      openmpi/1.3.4 (default)

gromacs/4.0.7      mpich2/1.0.8      mvapich2/1.4.1

----- /apps/lmod/modulefiles/core -----

HPC                      gnuplot/4.4.3                      python/2.6.4

R/2.13.1                      hdf5/1.8.7                      python/2.6.5

abaqus/6.9                      hyphy/2.0020110824beta      python/2.7.2 (default)

add\_scores/r1                      hypre/2.0.0                      raxml/1.0.5-light

admb/10.0                      intel/10.1                      raxml/7.3.0 (default)

admb/9.1 (default)                      intel/11.1 (default)      repeatmasker/3.3.0

amos/3.0.0                      lastz/1.02.00                      rosetta/2.1.1

amos/runAmos.log (default)      lps/2.2                      rosetta/2.1.2 (default)

# Discover – module avail

---

## Core:

----- / apps / lmod / modulefiles / core -----

clustalw / 2.1

## Compiler

----- / apps / lmod / modulefiles / compiler / intel / 10\_1 -----

mrBayes / 3.1.2

## MPI

----- / apps / lmod / modulefiles / mpi / intel10\_1 / openmpi\_1\_2 -----

amber / 10   dock / 64   gromacs / 4.0.7   mpiblast / 1.6.0   mrBayes / 3.1.2   raxml / 1.0.5-light

# Discover – module key

---

- module key clu

---

The following modules match your search criteria: "clu"

---

clustalw: clustalw/2.1

multiple sequence alignment software

# Basic usage - Evaluate

---

- `module spider clustalw`
- `module whatis clustalw`
- `module show clustalw`

# Discover – module spider clustalw

---

clustalw: clustalw/2.1

---

Description: multiple sequence alignment software

Help: This module enables the use of the ClustalW software

Clustal is popular multiple sequence alignment software.

ClustalW produces biologically meaningful multiple sequence alignments of divergent sequences. It calculates the best match for selected sequences and lines them up, so identities, similarities, and differences can be identified.

Version 2.1

# Evaluate – module whatis clustalw

---

clustalw/2.1 : Name: ClustalW  
clustalw/2.1 : Version: 2.1  
clustalw/2.1 : Category: biology,bioinformatics,alignment  
clustalw/2.1 : URL: <http://www.clustal.org/>  
clustalw/2.1 : Description: multiple sequence alignment software

# Evaluate – module show clustalw

---

**Top part – like module spider clustalw**

**Bottom part:**

```
whatis("Name: ClustalW")
```

```
whatis("Version: 2.1")
```

```
whatis("Category: biology,bioinformatics,alignment")
```

```
whatis("URL: http://www.clustal.org/")
```

```
whatis("Description: multiple sequence alignment software")
```

```
prepend_path("PATH", "/apps/clustalw/2.1/bin")
```

```
setenv("HPC_CLUSTALW_DIR", "/apps/clustalw/2.1")
```

```
setenv("HPC_CLUSTALW_BIN", "/apps/clustalw/2.1/bin")
```

```
set_alias("clustalw", "clustalw2")
```

# Basic usage - Use

---

- `module list`
- `module load clustalw`
- `module load python/2.6.5`
- `module add`
- `module del, rm, unload clustalw` – pick one!
- `module purge` – nuclear option
- E.g. `module load intel openmpi mrbayes`

# Use – load/unload

---

- module load clustalw
- module list

Currently Loaded Modules:

1) clustalw/2.1

- module unload clustalw OR module purge
- module list

Warning: No modules installed

# Basic usage - Swap

---

- `module spider mrbayes / 3.1.2`

This module can be loaded through the following modules:

`intel / 10.1, openmpi / 1.2.7`

`intel / 10.1`

`intel / 10.1, mvapich / 0.9.9`

# Swap – module swap

---

- module load intel openmpi mrbayes
- module list

Currently Loaded Modules:

1) intel/10.1 2) openmpi/1.2.7 3) mrbayes/3.1.2

- module swap openmpi mvapich

Due to MODULEPATH changes the follow modules have been reloaded:

1) mrbayes

- module list

Currently Loaded Modules:

1) intel/10.1 2) mvapich/0.9.9 3) mrbayes/3.1.2

# Your own environment

---

- module list

Currently Loaded Modules:

1) intel/10.1 2) mvapich/0.9.9 3) mrbayes/3.1.2

- module setdefault mymodz1

- module listdefault

Possible defaults:

1) mymodz1 2) test1 3) pipeline2

- module getdefault mymodz1

- module list

Currently Loaded Modules:

1) intel/10.1 2) mvapich/0.9.9 3) mrbayes/3.1.2

# Hands On Time

---

## ◆ Environment modules commands - module

- spider, avail
- load, add; unload
- del, rm, purge
- spider, whatis, show
- list, swap
- setdefault, listdefault, getdefault

## ◆ E.g. module load bio

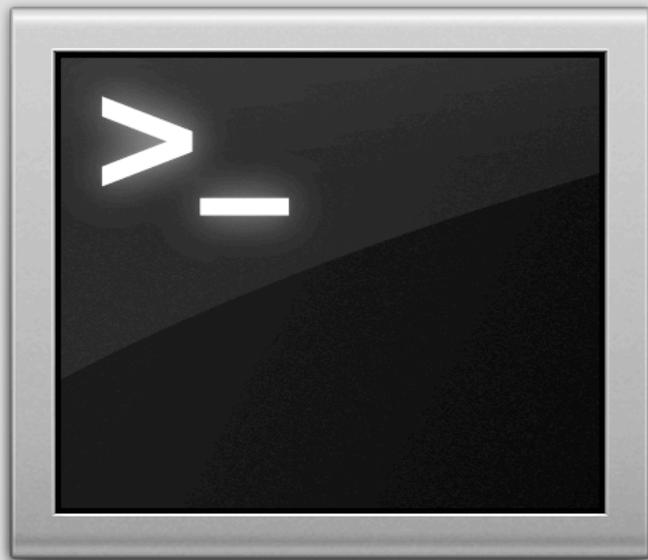
# Job Scheduling and Usage

---

PBS

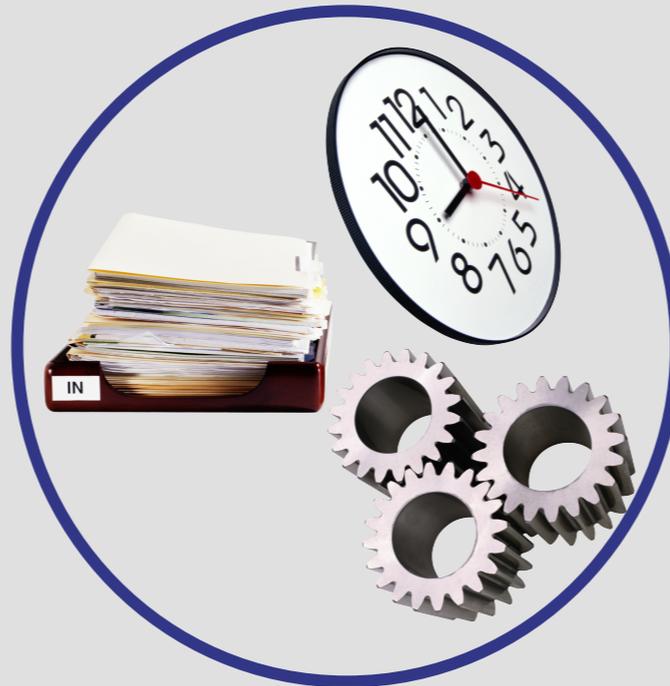
# Cluster basics

Head node(s)



Login to  
head  
node

Scheduler



Tell the  
scheduler what  
you want to do

Compute  
resources



Your job  
runs on the  
cluster

# Scheduling a job

---

- ◆ Need to tell scheduler what you want to do
  - Information about **how long** your job will run
  - How many **CPUs** you want and how you want them grouped
  - How much **RAM** your job will use
  - The **commands** that will be run

## Scheduler



Tell the scheduler what you want to do

# UF Research Computing

---

## ◆ Ordinary Shell Script

```
#!/bin/bash
```

```
pwd
```

```
date
```

```
hostname
```

Read the manual for your application of choice.

Commands typed on the command line can be put in a script.

# UF Research Computing

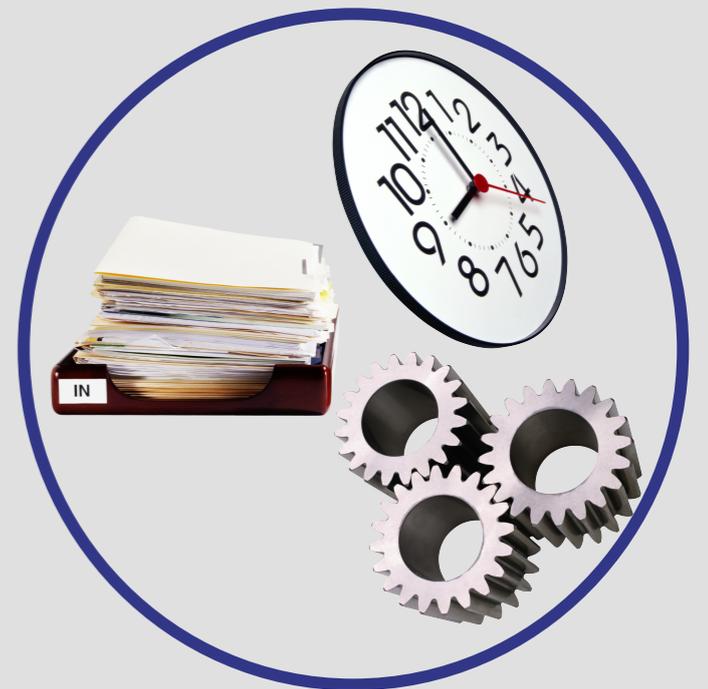
## ◆ Submission Script

```
#!/bin/bash  
#  
#PBS -N My_Job_Name  
#PBS -M Joe_Shmoe@ufl.edu  
#PBS -m abe  
#PBS -o My_Job_Name.log  
#PBS -j oe  
#PBS -l nodes=1:ppn=1  
#PBS -l pmem=900mb  
#PBS -l walltime=00:05:00
```

```
cd $PBS_O_WORKDIR  
pwd  
date  
hostname
```



## Scheduler



Tell the scheduler what you want to do

`-l nodes=1 : ppn=1`

---

- ◆ A node is one physical computer
  - Some applications can use multiple nodes (MPI applications)
  - Others can use multiple cores as long as they are all on the same node (OpenMP, threaded, shared memory)
- ◆ `nodes=1 : ppn=4` 4 cores on one node
- ◆ `nodes=2 : ppn=2` 2 cores on each of 2 nodes

# Node configuration

---

- ◆ Phase II: 2 x 2-core (4 cores / node)
  - 1316 total cores
- ◆ Phase III: 2 x 4-core (8 cores / node)
  - 872 total cores
- ◆ Phase IV: 2 x 6-core (12 cores / node)
  - 1612 total cores
- ◆ There are some additional resources, including nodes with 48 cores

-l pmem=900mb

---

## ◆ Per processor memory limit

- If your job tries to use more than this amount, it will be killed
- The scheduler reserves this amount for your job, so if you request lots more than you need you are wasting resources

# Node configuration

---

- ◆ Phase II: 4 or 8GB (1 or 2GB / core)
  - 1316 total cores (~50% 1GB, 50% 2GB)
- ◆ Phase III: 32 or 64GB (4 or 8GB / core)
  - 872 total cores (~80% 4GB, 20% 8GB)
- ◆ Phase IV: 32GB (2.6GB / core)
  - 1612 total cores
- ◆ There some additional resources, including nodes with 512GB RAM

# Processor equivalents

---

## ◆ Complicated formula

- Accounts for RAM usage above the per core average
- Phase IV node: 12 cores, 32GB of RAM
  - A single core job using 24GB of RAM is using  $\sim 3/4$  of the resources on that node
  - It will have a PE of about 9
    - Non-investor jobs limited to 8 PEs

-l walltime=00:05:00

---

◆ Wall time in dd:hh:mm:ss

◆ Queues:

- long: ~10% of resources

- Investor: > 7 days (168 hrs)

- Non-investor: >3 days (72 hrs)

- short: ~90% of resources

◆ Keep your queue full!!!

# Run a job

---

```
ssh <name>@submit.hpc.ufl.edu
```

```
cd /scratch/hpc/$USER
```

```
cp -r ../galaxy/examples/blast_demo/ .
```

```
cd blast_demo
```

```
more BLAST.qsub
```

```
qsub BLAST.qsub
```

◆ Use \$TMPDIR for single node jobs!

# Job Management

---

- `qsub`: job submission
- `qstat`: check queue status
- `qdel`: job deletion
- `qdelmine`: delete all my jobs
- Documentation
  - [http://wiki.hpc.ufl.edu/index.php/Job\\_Submission\\_Queues](http://wiki.hpc.ufl.edu/index.php/Job_Submission_Queues)

# How to get help

---

Help!!!

# How to get help

---

## ◆ Asking for help

- Support Request Tickets

- <http://support.hpc.ufl.edu>

- Use for everything - not just software bugs but for any questions or help requests

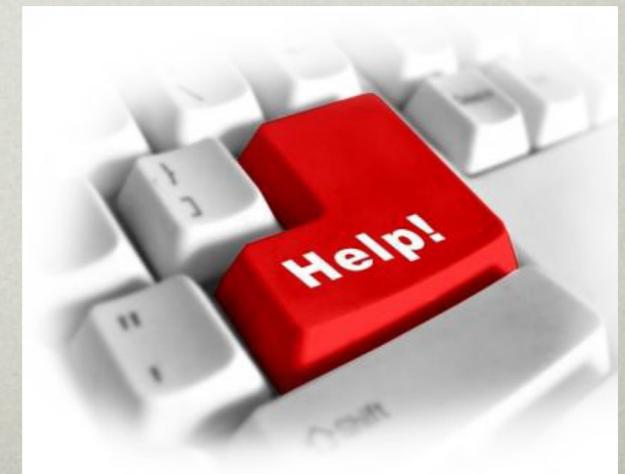
- Searchable database of solutions

- When you don't have access to web

- [support@hpc.ufl.edu](mailto:support@hpc.ufl.edu)

- [om@hpc.ufl.edu](mailto:om@hpc.ufl.edu) (Biological Support)

- [magitz@ufl.edu](mailto:magitz@ufl.edu) (Bio training and Q/A)



# Documentation

---

## ◆ UF HPC Encyclopedia

- <http://wiki.hpc.ufl.edu>
  - Documents on hardware and software resources
  - User guides
  - Sample submission scripts
  - Research-specific sections
- <http://hpc.ufl.edu/support>
  - Frequently Asked Questions
  - Account set up and maintenance



# Training Schedule

---

- ✓ October 20th: Intro to UFHPC, getting started
- ✓ October 27th: Modules and basic submission scripts
- ◆ November 3rd: Galaxy overview
- ◆ November 10th: No session - Florida Genetics 2011
- ◆ November 17th: Introduction to working in Linux shell environment
- ◆ November 24th: No session – Thanksgiving
- ◆ December 1st: Working with NGS data (quality filtering, format conversion, etc.)

2011-10-27 – HPC modules

---

Thank  
you!