1. Introduction

This guide is for users who submit Amsterdam Modeling Suite (AMS) jobs on Penn State's Roar supercomputer. The AMS installation guide provides a brief guide to submitting remote jobs from the AMS interface. However, there are additional troubleshooting steps required to successfully submit AMS remote jobs to Roar. Performing the necessary additional setup outlined in this document will allow users to allow users to submit remote jobs onto the Roar computers directly from their local AMS graphical user interface (GUI). Further, we hope these troubleshooting steps maybe helpful for users who wish to submit remote jobs to other servers.

There are two primary requirements in order to submit remote jobs for the AMS GUI.

1. Setting up ssh keys such the the AMS GUI can communicate with the remote servers without user intervention.
2. Defining the appropriate queues for remote jobs on the AMSjobs interface.

Before we go into these steps, here we present a brief overview of the Amsterdam modeling suite and Penn State's Roar supercomputer. Readers who are familiar with these may skip ahead to sections 2 and 3.

Amsterdam modeling suite

Amsterdam modeling suite is software suite for computational chemistry calculations, developed by SCM - Sofware for Chemistry and Materials. AMS contains software products capable of handling density functional theory (DFT), molecular Dynamics (MD), and other ab-initio and semi empirical computational chemistry simulations. In addition to the integrated software modules, AMS can be used as an interface to handle QuantumEspresso(QE) and Vienna ab-initio simulation package(VASP) jobs. Detailed installation guides and documentation for AMS, on all major operating systems, can be found on the AMS installation guide and accompanying video guides.

Note:

Version compatibility with the AMS installation on the remote server needs to be taken into consideration when installing AMS on the local Machine. If the server AMS version is AMS2019 or older, the local installation should also be AMS2019 or older. AMS2020.xxx and newer versions on the remote server are compatible with any local AMS version.

The AMS suite consists of several GUIs tailored to carry out various tasks related to DFT and other computational chemistry calculations. Please refer to the AMS guide for full details on the different GUIs. In this document we will be primarily concerned about the AMSjobs interface, where all the requisite setup for remote jobs will be performed.

Roar Supercomputer

The Roar supercomputer is a high-performance computing(HPC) infrastructure available to researchers affiliated to Penn State university. Roar managed by the Institute for Computational and Data Sciences(ICDS) at Penn State. For a detailed overview of Roar, readers are referred to the Roar user guide.

Here, we will review the details about Roar that our relevant to our task of submitting AMS remote jobs. There are two versions AMS available to be loaded as modules on Roar: AMS2020.103, and ADF2019.303. We recommend using the AMS2020.103 version, since it is compatible with any version of AMS installed on the local machine. On Roar the following command will load AMS2020.103,

```
$ module load ams
```
.

The remote hostname for ssh logins onto Roar is, `submit.aci.ics.psu.edu` . Additionally, batch jobs on Roar are schedulin is handled by Portable Batch System (PBS).

2. Establishing ssh connection between local AMS and remote machine

To submit remote jobs from the AMSjobs interface, we need to setup ssh keys, such that the ssh connection does not require entering the password or any other user intervention. However, setting up the ssh keys does not bypass the two factor authentication (2FA) requirement when logging onto Roar computers. Once a login is authorized using 2FA, subsequent logins for (x time?) do not require 2FA. So unfortunately, an ssh login from the terminal is required before performing any remote task (submitting remote jobs, retrieving remote files etc.).

check

The default timeout for ssh connections used by AMS is typically too low. A higher timeout can be set as follows. On AMSjobs interface, go to `SCM -> Preferences` and then on the AMSpreferences window on the Module tab select AMSjobs. On this window, make the following changes:

- `Local Ssh Command:` set to `ssh -q` .
- `Timeout For Remote Commands:` set to 60 seconds.
- `rsync command:` set to `rsync -q` .
- `Refresh result file every:` set to 60 seconds.

Next create the required ssh keys from the terminal. Users on Linux and Mac operating systems, who already have set up ssh keys for password-free (even the ssh keys should have no passphrase) login to Roar may skip ahead to section . For users on Windows please complete the steps outlined in the highlighted region below before performing the following steps.

cref

For Windows local machines only, perform this step first.

AMS handles remote jobs by sending commands to a bash-like shell in the background. This shell is accessible to users from the context menu `Help -> Command-line` on the AMSjobs interface and entering the command `bash` in the command prompt that pops up.

We refer to this shell as bash-like because of a few peculiarities found here. Primarily of relevance to us is the fact the `$HOME` environment variable does not point to the same location as `/home/$USERNAME/` as expected in Linux systems. AMS looks for ssh keys in the latter folder. The root folder for the terminal is located at `<path to AMS installation>/msys/` . And the directory `/home/<username>/` does not exist. This folder needs to be created before the ssh keys are created, using the following command.

```
$ mkdir -pv /home/$USERNAME
```

The discrepancy between `$HOME` and `/home/$USERNAME/` results in further complications because AMS uses both locations. The `.ssh` directory is stored at `/home/$USERNAME` , and the `.scm_gui` directory is stored in `$HOME/` . When implementing the commands provided below, use `$HOME` as is, and change `~` to `/home/$USERNAME` .

When logging in to Roar for the first time users will be prompted to add the ECDSA key fingerprint of Roar host to the `~/.ssh/known_hosts` file. Please do so, as this is required for remote job submissions from the AMS GUI. The commands to setup the ssh keys are given below.

- Create public and private ssh key pair:

  ```
  $ ssh-keygen
  ```

  When prompted for a passphrase for the ssh keys leave it blank. Also the use default location to store the key `~/.ssh/id_rsa`.

- Copy public key to `$HOME/.ssh/authorized_keys` on the remote host. On Windows this step needs to be done manually. For other os the command is (replace `xyz123` with valid remote username),

  ```
  $ ssh-copy-id -i ~/.ssh/id_rsa.pub xyz123@submit.aci.ics.psu.edu
  ```

  On Windows machines, the above can be accomplished with the following commands.

```
$ key=$(cat /home/$USERNAME/.ssh/id_rsa.pub)
$ ssh xyz123@submit.aci.ics.psu.edu "echo $key >> .ssh/authorized_keys"
```

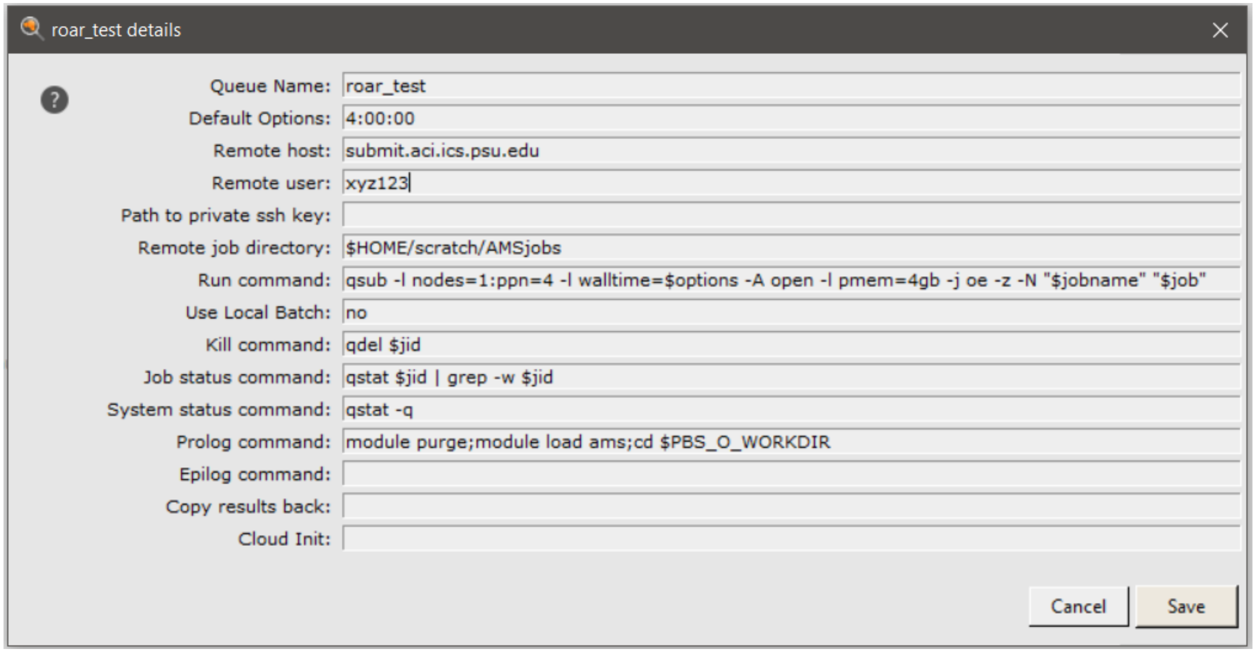
- The final step is to test if the keys are setup properly. This can be done by performing an ssh login onto Roar using the command:

  ```
  $ ssh xyz123@submit.aci.ics.psu.edu
  ```

  If the keys are setup properly you should be logged in without a password prompt (2FA may still be required).

3. Defining remote queues on AMSjobs interface.

Submitting AMS remote jobs requires defining the appropriate queues that gives the program information about the remote host and requested resources. Roar uses a PBS scheduler the defined queue must be of the same type. A new queue can be created manually from the AMSjobs interface. Navigate to `Queue -> New -> PBS` . Detailed information about each field on the queue definition window is available at the following link. Shown below in fig. 1 are the settings required to submit remote jobs to Roar. The remote user field needs to be set appropriately.



Queue Name: roar_test
Default Options: 4:00:00
Remote host: submit.aci.ics.psu.edu
Remote user: xyz123
Path to private ssh key:
Remote job directory: $HOME/scratch/AMSjobs
Run command: qsub -l nodes=1:ppn=4 -l walltime=$options -A open -l pmem=4gb -j oe -z -N "$jobname" "$job"
Use Local Batch: no
Kill command: qdel $jid
Job status command: qstat $jid | grep -w $jid
System status command: qstat -q
Prolog command: module purge;module load ams;cd $PBS_O_WORKDIR
Epilog command:
Copy results back:
Cloud Init:

Figure 1: Test queue definition on AMSjobs.

Here we provide explanations for some of the more complex fields in the above queue definition.

- `Default Options:` field can be used as an additional bash variable for run command. The label of this field is slightly misleading, since only one variable can be set here despite the trailing 's' in `Default Options:` seeming to imply otherwise. In our queue in fig. 1 we use it for the default walltime requested. The advantage of this field is that this value can be altered from job to job without editing the queue definition.
- We set the `Remote job directory:` field to `$HOME/scratch/AMSjobs` . On Roar we recommend this setting since the computational chemistry jobs performed could produce files that require considerable storage space. Using a directory in the scratch folder

here will avoid jobs being interrupted due to files exceeding storage quotas provided to users.

- `Run command:` , here we provide the `qsub` command to schedule the job on the remote machine. Listed below are the various flags and options. Additional optional arguments may be used if necessary.

Flag	Options	Description	
`-l`	nodes=1	Requested number of nodes	*Required*
	ppn=4	Processors per node	
	walltime=$options	Walltime requested. Set to `Default options:` field	
`-A`	open	Queue allocation	
`-l`	pmem=4gb	Memory per processor	*Recommended*
`-j`	oe	Redirect qsub `stdout` and `stderr`	
`-z`		Suppress qsub command printing confirmation to terminal .	

- `Prolog command:` , commands included here are effectively prepended to the job script generated by AMS. We use this field to load the AMS module on Roar. Additionally, the command `cd $PBS_O_WORKDIR` changes the compute nodes directory to the job directory. Flag options such as: queue alloction, nodes, processors per node, memory per processor can vary from job to job. To avoid editing the queue definition every time we recommend defining multiple queues with various possible combinations for these flag options.

- `Queue name:` here we used a place holder name for the queue. However, we recommend using names indicative of the variable flag options. For example, 'open_1_4_4' - indicating this queue is using the open allocation and requesting 1 node, 4 processors per node, and 4 GB memory per node - could be a suitable name for the queue defined in fig. 1.

The queue definition discussed above is saved by AMS as `<Queue name>.tid` file in the `$HOME/.scm_gui/` folder. The file is stored as follows.

```
# backup:
# batch: no
# cloudid:
# cloudinit:
# comparewithtarget: changed
# copyback:
# epilog:
# error: -1
# fakejob: 0
# hostname: submit.aci.ics.psu.edu
# jobdir: $HOME/scratch/AMSjobs
# jobscript:
# jobstatuscmd: qstat $jid | grep -w $jid
# jobtype:
# killcmd: qdel $jid
# label: roar_test
# logfile: logfile
# mtime:
# options: 4:00:00
# paused: 0
# privatesshkeypath:
# prolog: module purge;module load ams;cd $PBS_O_WORKDIR
# remoteprefix:
# runcmd: qsub -l nodes=1:ppn=4 -l walltime=$options -A open -l pmem=4gb -j oe -z -N
    "$jobname" "$job"
# sysstatuscmd: qstat -q
# username: xyz123
# warning: 0
```

For readers convenience we have created a bash script to automate defining new queues. The script produces `<Queue name>.tid` files such as the one shown above. The script accepts two mandatory arguments providing the number of nodes and processors per node to be requested by the queue. Additionally optional arguments may be passed to the script to alter the options listed in the table below.

Option	Flag	Default Values
Queue Name	-n	(Default determined based on the inputs)
Remote user	-u	(Needs to be set)
Remote host	-h	submit.aci.ics.psu.edu
Allocation	-a	open
Memory per Processor	-p	4gb
Walltime	-w	4:00:00
Remote job Directory	-d	$HOME/scratch/AMSjobs
Skip confirmation (Flag only)	-c	Not applicable