High performance research computer in TAMU: Applications, Usage, and solutions to common questions

1. Account

Home website of High-Performance Computing Center: https://hprc.tamu.edu/

`Contact information of HPRC` , you will need this many times and usually you will also need to make a phone call in case that your emails are not replied in time.

Hours: 9am-5pm

Location: 222 Jones St, Henderson Hall, Room 114A

Phone: 979-845-0219

Email: help@hprc.tamu.edu

Fax: 979-847-8643

HPRC in tamu provides three tiers of accounts, including **basic, startup, and research**, as below, and different account will be allocated with different amount of **Service Units (SUs)**. For detailed

definition and calculation method of SU, please refer to the official policy. Based on your identifies, you are eligible to apply for different accounts.

> **Review table below for Service Units (SUs) of each allocation type in detail. 1 SU = 1 wallclock hour on 1 CPU.*
> *** Basic allocations are 5000 SUs*

Type of Applicant	Need 5000 SUs?**	Need up to 100,000 SUs?	Need > 100,000 SUs?
Students	Basic		
Faculty	Basic	Startup	Research
Qualified Staff	Basic	Startup	Research
Post-Docs	Basic		
Research Associates	Basic	Startup	
Research Scientists	Basic	Startup	Research

1.1. Application form

To apply for an account, you will be asked to fill out an application form, which includes the following information. Take my application as an example.

1) Dr. O'Neill's personal information:

1. NetId: zoneill
2. 979-845-4931
3. zoneill@tamu.edu

2) Project description and software you need

We need to run a significant amount of building energy simulation cases to find the optimal scenarios. The software we will use include: EnergyPlus and GenOpt.
a. EnergyPlus is a whole building simulation program, and we will use version 9.3. Download link: https://github.com/NREL/EnergyPlus/releases/tag/v9.3.0

b. GenOpt, an optimization software which provides an easy-to-use interface with EnergyPlus. Download link: https://simulationresearch.lbl.gov/GO/download.html

3) Justification for using HPRC resources

Based on our estimation, the total execution time of our simulation experiment is more than 10,000 hours in our personal computer. Therefore, we think the high-performance computing system could help to resolve our execution time issues.

4) Justification for the number of hours requested..

Each of our simulation instance is involved with 6~24 parameters. To find the optimal scenarios, the number of required simulation instances with 6~24 parameters would be approximately 10,000 for this experiments. Besides, each instance takes approximately one cpu for 2 minutes. This process need to be repeated for about 100~200 times for different experiment settings.
Therefore, the total SUs need is 10,000*1/30*200 = 60,000 SUs.

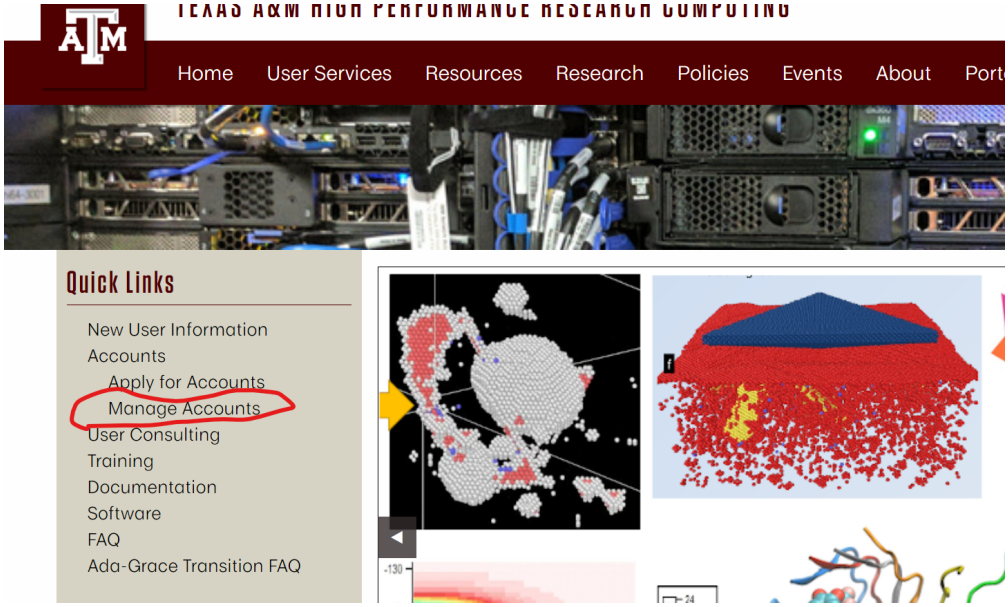
Other information

funding sources, researchers who will receive sub-allocations, etc.

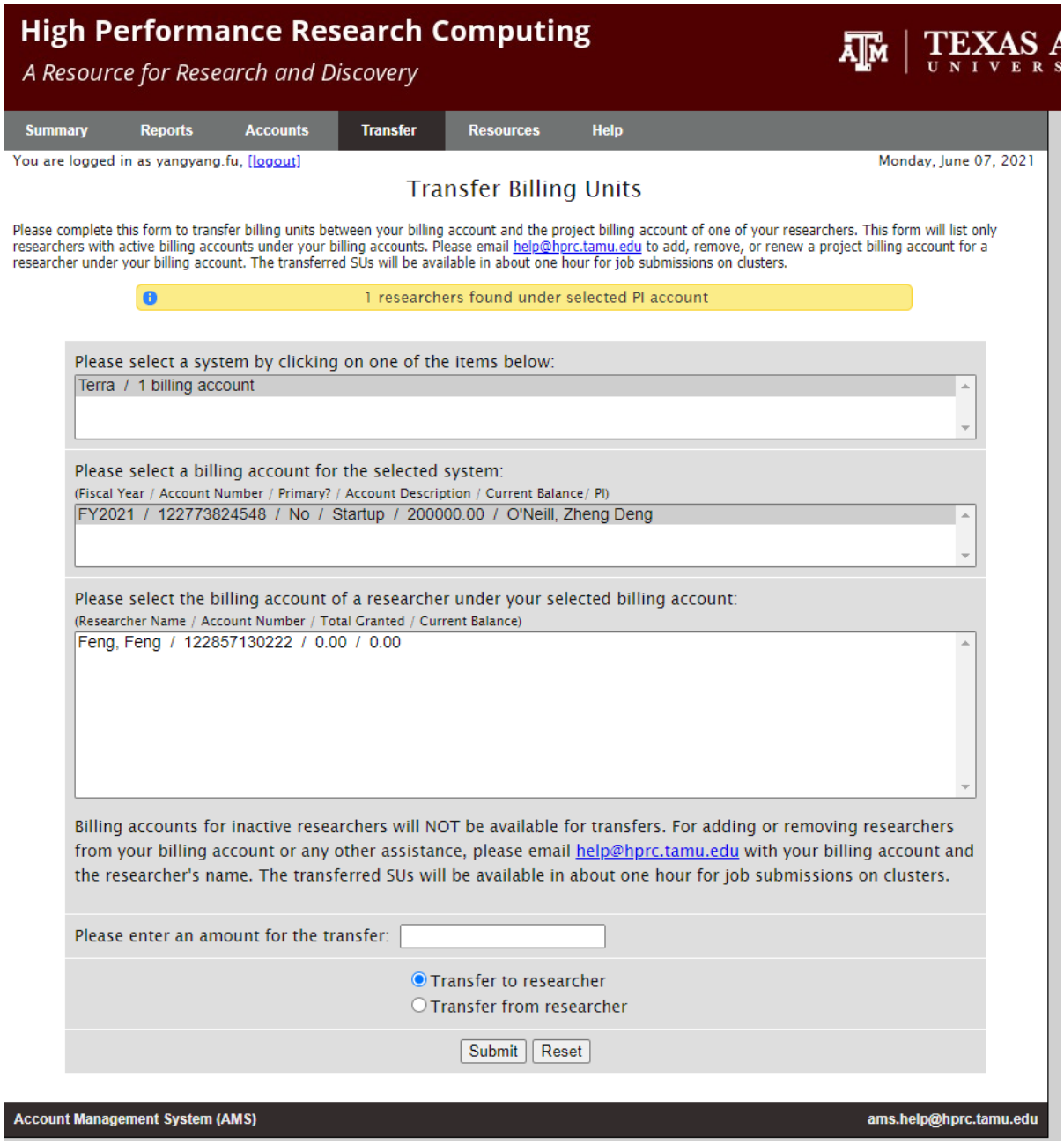
1.2. Account manage

Usually after one or two office days, your application will be approved.However, as an student, we are only able to obtain less than 5,000 SUs and that is not enough for most cases. In this case, we could as Dr.O'Neill (or your supervisor, research assistant in your group) to apply for a high tier account and then transfer their SUs to you.

After their applications completed, they could transfer their SUs in the account management system. You are able to find this in the home page of HPRC.



The account manage system is slightly different for basic accounts and startup/research accounts. That is, the manage system for startup/research accounts have a "transfer" page where SUs transfer requests could be submitted. Yo might need to ask help desk to add the billing account of a researcher who will receive sub-allocations.



.

After the transfer is completed, you are supposed to have sufficient SUs in your account and next you will move forward to access and usage of the HPC.

2. HPRC usage tutorial

HPRC provides us detailed user guides, which basically covers every aspects you will need, including access the HPRC, file transfer, Job submission and etc. You could find these user guides here.

HPRC is a Linux system with GUI, and if you are familiar with Linux commands you will feel comfortable with this system. If not, you might need to learn it from scratch. Good luck.

2.1. Access your account

On windows computers, you could use either Putty or MobaXterm. Both these are SSH shells that allow you connect to HPRC remotely.
If you need to connect to HPR from outside of the campus, you will need to activate TAMU VPN first.

Then, after you log into HPC, you could use the following command to check your account balance..

> myproject -l

```
(base) [ffeng@terra2 ~]$ myproject -l
================================================================================================
               List of ffeng's Project Accounts
------------------------------------------------------------------------------------------------
|  Account   |  FY  | Default | Allocation |Used & Pending SUs|   Balance  |          PI        |
------------------------------------------------------------------------------------------------
|122774970664|  2021|        N|   150000.00|              0.00|   150000.00|O'Neill, Zheng Deng |
------------------------------------------------------------------------------------------------
|122774975429|  2021|        Y|     5000.00|             -1.22|     4998.78|O'Neill, Zheng Deng |
------------------------------------------------------------------------------------------------
```

2.2. file transfer

To transfer file into/from the HPC, you could use `WinSCP` . You could find the detailed user guides here.

Besides, you could also use GIT to download files from the online Repositories.

2.3. Software

HPRC has many pre-installed software, including python, cmake, and etc. You could search this by keywords using the following command, taking python as an example..

> mla python

You will get a list of python modules in the server. Then, you could select one of them by loading it to your account using the following command.

> ml Python/3.8.6-GCCcore-10.2.0

Generally, you need to load these software every time when you start a new HPR connection. So it is highly recommended to write all commands in a bash file.

2.3.1. Installing new software

Installing new software is the most tricky me for me.. Theoretically, you could ask the help desk to install software for you. However, after several request emails, the reply I got from the help desk is to install the software by myself, and here is the instructions they sent me.

> export INSTALL_LOC=$SCRATCH/programs/EnergyPlus-9-5-0 # you can edit
> mkdir -p $INSTALL_LOC
> ml CMake/3.18.4-GCCcore-10.2.0
> cd $SCRATCH
> wget url(for your package)
> tar -xzvf v9-5-0.tar.gz
> cd EnergyPlus-9.5.0
> cmake -DCMAKE_INSTALL_PREFIX=$INSTALL_LOC
>
> make -j4 install
> echo “Program located at $INSTALL_LOC”

Basically, this instruction is correct. However, you will get some error when you follow this.
First, this cmake version might not work for you case. You could check the building instruction for you software which will give the cmake version requirements and etc.
The problem I got is

> **“CMake: The C Compiler is not able to compile a simple test program”**

There are several solutions for this problem.
1) You could add an option to cmake command to walk around this problem as below.

> cmake -CMAKE_TRY_COMPILE_TARGET_TYPE=STATIC_LIBRARY -
> DCMAKE_INSTALL_PREFIX=$INSTALL_LOC
> However, this doesn’t work for my case.

2) Use a different cmake version, here is the version that I used.

> ml CMake/3.15.3-GCCcore-8.3.0

Second, some software have some dependencies that is not installed or inconsistent with the existing software in the system. For example. EnergyPlus v9.3. require at least python 3, and if you have python 2 as default python version this installation process will raise a problem. So load python 3 to your system.

> ml Anaconda3/2020.11

Lastly, after installation is completed, you need to add the installation path to you system path as below.

```
export PATH= $SCRATCH/programs/EnergyPlus-9-4-0:$PATH
```

2.4. Job submission, monitor, and etc.

Then, you could submit your Job and carry out your simulation. For this part, refer to the official guide for instructions.

1) Job submission.

In the first experiment, ntasks is set as 30, and ntasks-per-node equals 10. However, I came into an problem with this setting. That is, although 30 processes were created and successfully executed, these processes have problem communicating results back to the main process.

```
**Solutions tried:**
```

Potential solution1: Python's multiprocessing module does not support cluster distributed computation.
Answer: No. This is not the case…

Memory allocated isn