Distributed Drug Discovery

MpiDock

Distributed Drug Discovery

Cal Poly Computer Engineering Capstone

Client

Dr. Scott Eagon

Team

Lucy Brantley
lrbowen@calpoly.edu

Derek Nola
dnola@calpoly.edu

Dennis Li
dli28@calpoly.edu

3-15-2018

Table of Contents

System Overview

The **MPiDock** system is a series of different programs layered on top of each other to perform various tasks (see **Figure 1**).

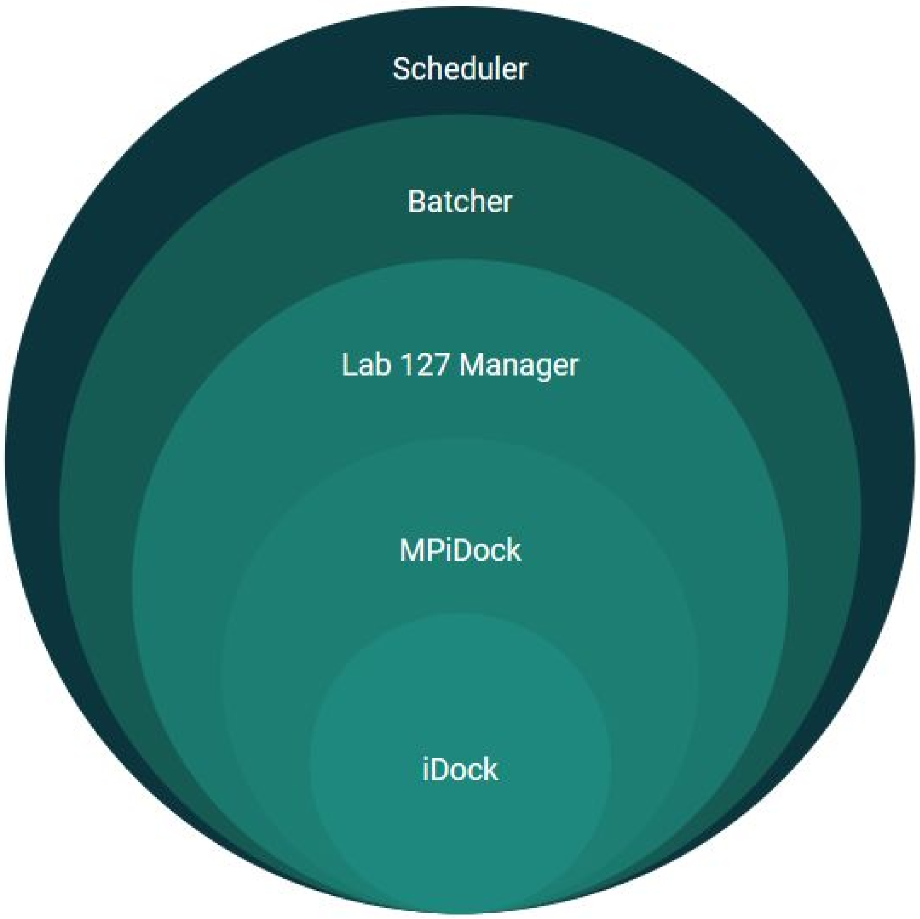


Figure 1: System Overview

At the outermost level, there is the scheduler, written as a command line interface for crontab. The scheduler would then call the batcher, a program which divides up input files and then executes commands remotely with those input files. The batcher executes the lab manager on a machine within Cal Poly's MPAC Lab, which sets up an MPI Cluster from the lab computers. The lab manager then executes MPiDock, an MPI wrapper for the iDock molecular docking program. Multiple instances of iDock are executed on each machine and the output is then stored and processed by the lab manager.

Setting Up the System

SSH

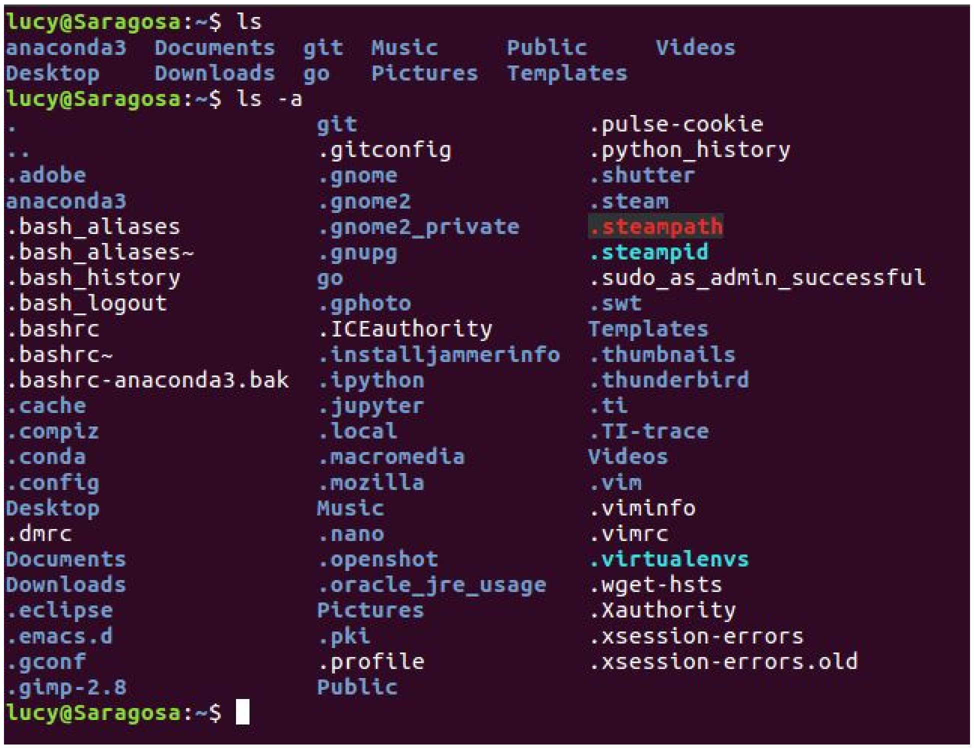
The first part of using the system is setting up both an SSH alias and SSH keys for logging in to one of the MPAC lab computers.

Setting up an SSH alias

An SSH alias will allow you to have a shorthand for a remote location you want to ssh into. Not only will this save time, but it can also be more descriptive within scripts and commands.

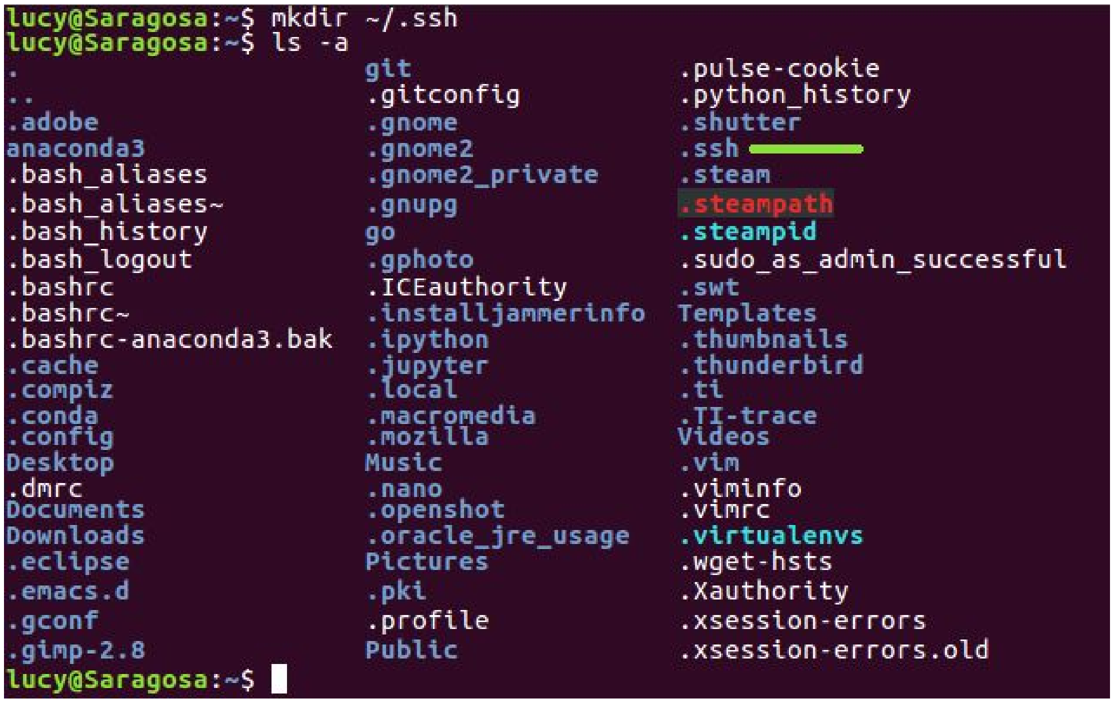
1) Make a folder within your home directory named ".ssh"

```
mkdir ~/.ssh
```



```
lucy@Saragosa:~$ ls
anaconda3  Documents  git  Music     Public     Videos
Desktop    Downloads  go   Pictures  Templates
lucy@Saragosa:~$ ls -a
.                       git                 .pulse-cookie
..                      .gitconfig          .python_history
.adobe                  .gnome              .shutter
anaconda3               .gnome2             .steam
.bash_aliases           .gnome2_private     .steampath
.bash_aliases~          .gnupg              .steampid
.bash_history           go                  .sudo_as_admin_successful
.bash_logout            .gphoto             .swt
.bashrc                 .ICEauthority       Templates
.bashrc~                .installjammerinfo  .thumbnails
.bashrc-anaconda3.bak   .ipython            .thunderbird
.cache                  .jupyter            .ti
.compiz                 .local              .TI-trace
.conda                  .macromedia         Videos
.config                 .mozilla            .vim
Desktop                 Music               .viminfo
.dmrc                   .nano               .vimrc
Documents               .openshot           .virtualenvs
Downloads               .oracle_jre_usage   .wget-hsts
.eclipse                Pictures            .Xauthority
.emacs.d                .pki                .xsession-errors
.gconf                  .profile            .xsession-errors.old
.gimp-2.8               Public
lucy@Saragosa:~$
```

Directory before command



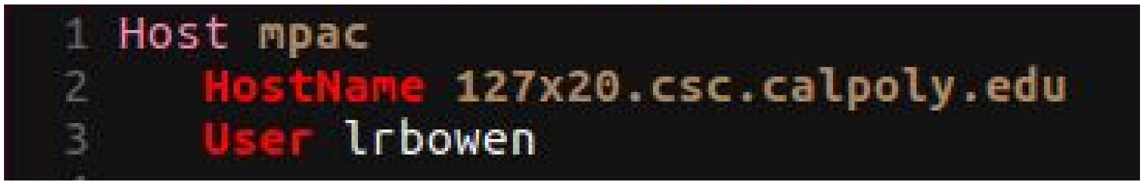
```
lucy@Saragosa:~$ mkdir ~/.ssh
lucy@Saragosa:~$ ls -a
.                       git                     .pulse-cookie
..                      .gitconfig              .python_history
.adobe                  .gnome                  .shutter
anaconda3               .gnome2                 .ssh
.bash_aliases           .gnome2_private         .steam
.bash_aliases~          .gnupg                  .steampath
.bash_history           go                      .steampid
.bash_logout            .gphoto                 .sudo_as_admin_successful
.bashrc                 .ICEauthority           .swt
.bashrc~                .installjammerinfo      Templates
.bashrc-anaconda3.bak   .ipython                .thumbnails
.cache                  .jupyter                .thunderbird
.compiz                 .local                  .ti
.conda                  .macromedia             .TI-trace
.config                 .mozilla                Videos
Desktop                 Music                   .vim
.dmrc                   .nano                   .viminfo
Documents               .openshot               .vimrc
Downloads               .oracle_jre_usage       .virtualenvs
.eclipse                Pictures                .wget-hsts
.emacs.d                .pki                    .Xauthority
.gconf                  .profile                .xsession-errors
.gimp-2.8               Public                  .xsession-errors.old
lucy@Saragosa:~$
```

Directory after command

2) Go into the ".ssh" directory and make a file named config

```
cd ~/.ssh
touch config
```

3) Edit the config file to place in your desired SSH alias. Here's an example for an SSH alias to the MPAC lab. After inserting this, save and quit.

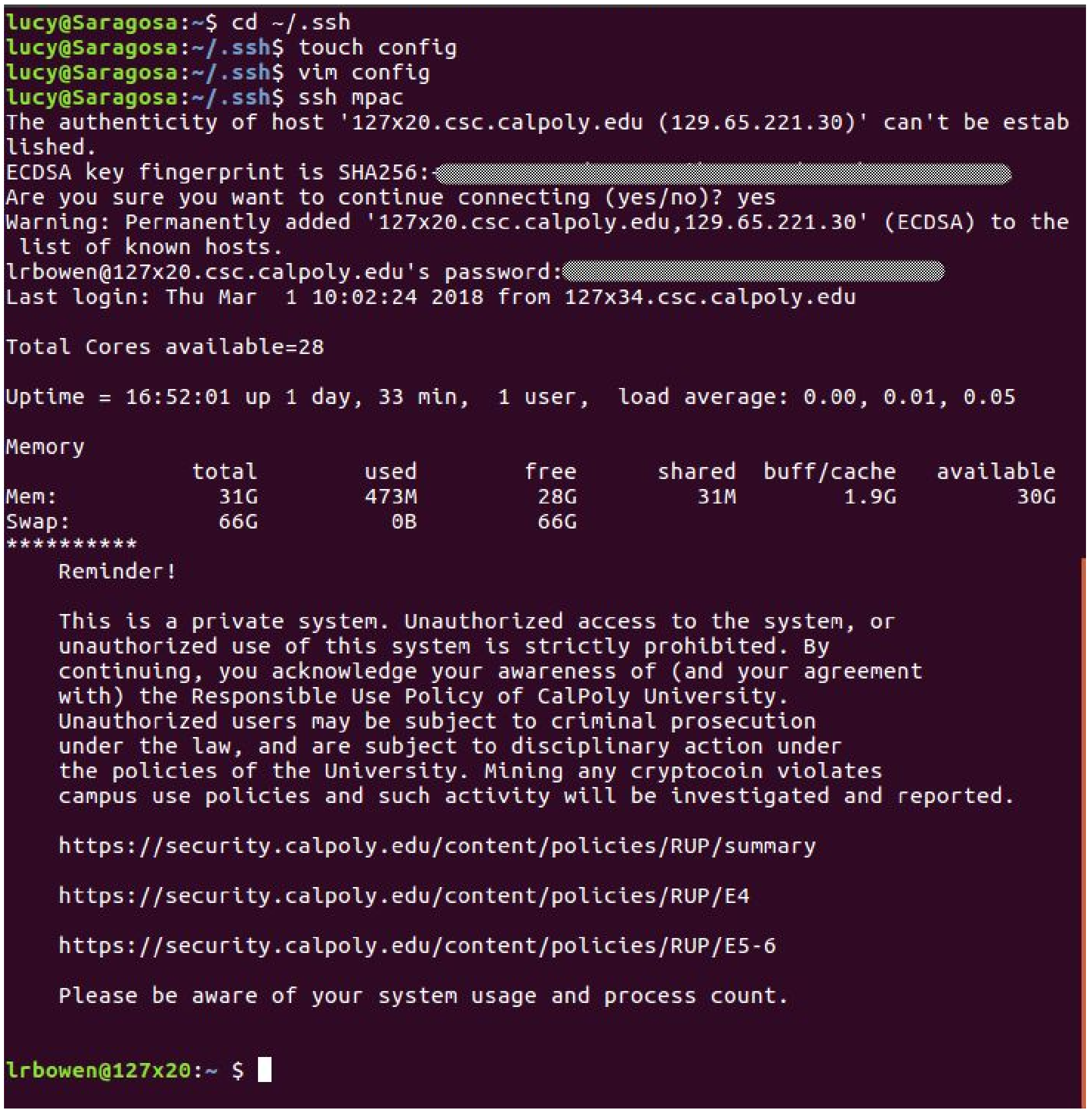


```
Host mpac
    HostName 127x20.csc.calpoly.edu
    User lrbowen
```

Example SSH alias with formatting

4) Test out the alias. See if it gets you where you want to go.

```
ssh mpac
```



```
lucy@Saragosa:~$ cd ~/.ssh
lucy@Saragosa:~/.ssh$ touch config
lucy@Saragosa:~/.ssh$ vim config
lucy@Saragosa:~/.ssh$ ssh mpac
The authenticity of host '127x20.csc.calpoly.edu (129.65.221.30)' can't be estab
lished.
ECDSA key fingerprint is SHA256:[redacted]
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '127x20.csc.calpoly.edu,129.65.221.30' (ECDSA) to the
 list of known hosts.
lrbowen@127x20.csc.calpoly.edu's password:[redacted]
Last login: Thu Mar  1 10:02:24 2018 from 127x34.csc.calpoly.edu

Total Cores available=28

Uptime = 16:52:01 up 1 day, 33 min,  1 user,  load average: 0.00, 0.01, 0.05

Memory
              total        used        free      shared  buff/cache   available
Mem:            31G        473M         28G         31M        1.9G         30G
Swap:           66G          0B         66G
**********
    Reminder!

    This is a private system. Unauthorized access to the system, or
    unauthorized use of this system is strictly prohibited. By
    continuing, you acknowledge your awareness of (and your agreement
    with) the Responsible Use Policy of CalPoly University.
    Unauthorized users may be subject to criminal prosecution
    under the law, and are subject to disciplinary action under
    the policies of the University. Mining any cryptocoin violates
    campus use policies and such activity will be investigated and reported.

    https://security.calpoly.edu/content/policies/RUP/summary

    https://security.calpoly.edu/content/policies/RUP/E4

    https://security.calpoly.edu/content/policies/RUP/E5-6

    Please be aware of your system usage and process count.


lrbowen@127x20:~ $ 
```

Example of what should happen if the alias is set up correctly

Setting up SSH keys

Setting up SSH keys will allow you to ssh into a remote machine without having to enter in your password every time. Not only does this save time, but it also allows other programs, including our **Batcher**, to log in on your behalf without needing to have a plain text password stored somewhere (a huge security risk).

1) Navigate to your ".ssh" folder.

```
cd ~/.ssh
```

2) Generate a set of SSH keys.

```
ssh-keygen
```

When it asks for a location, choose the default unless you already have a set of keys.

When it asks for a passphrase, press enter to not have a passphrase.

When it asks to confirm the passphrase, press enter again.

3) Copy the public key to the remote machine, in our case, the MPAC lab computer.

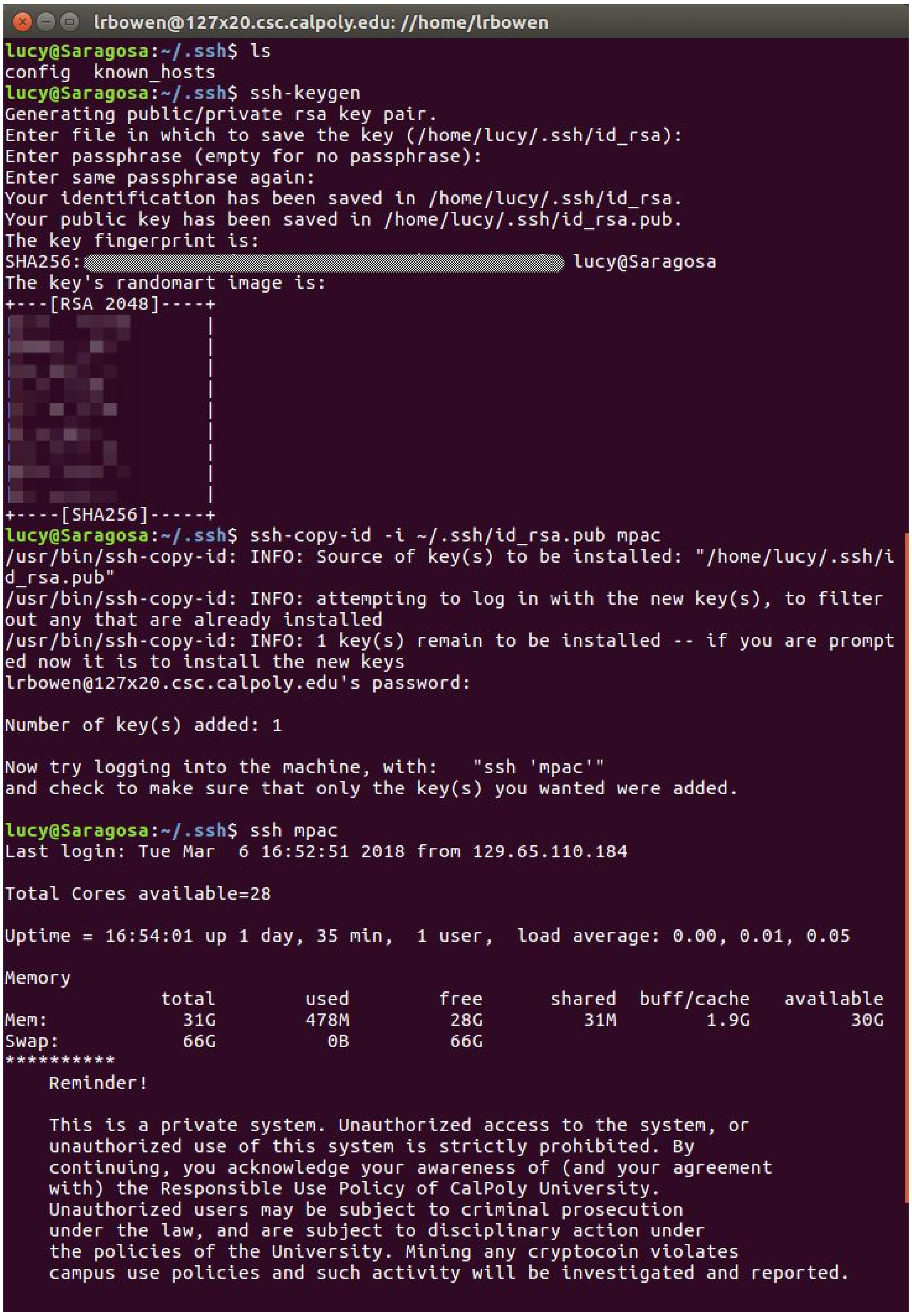
```
ssh-copy-id -i ~/.ssh/id_rsa.pub mpac
```

This command sends over our newly made public key using our ssh alias.

It will ask you for your password to log on to the MPAC computer. This is so that it can copy over your credentials.

4) Test out the ssh key configuration by trying to log in to the MPAC lab. You should not need to enter in your password.

```
ssh mpac
```



```
lrbowen@127x20.csc.calpoly.edu: //home/lrbowen
lucy@Saragosa:~/.ssh$ ls
config  known_hosts
lucy@Saragosa:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lucy/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/lucy/.ssh/id_rsa.
Your public key has been saved in /home/lucy/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:[redacted] lucy@Saragosa
The key's randomart image is:
+---[RSA 2048]----+
[redacted]
+----[SHA256]-----+
lucy@Saragosa:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa.pub mpac
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/lucy/.ssh/i
d_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
lrbowen@127x20.csc.calpoly.edu's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'mpac'"
and check to make sure that only the key(s) you wanted were added.

lucy@Saragosa:~/.ssh$ ssh mpac
Last login: Tue Mar  6 16:52:51 2018 from 129.65.110.184

Total Cores available=28

Uptime = 16:54:01 up 1 day, 35 min,  1 user,  load average: 0.00, 0.01, 0.05

Memory
              total        used        free      shared  buff/cache   available
Mem:            31G        478M         28G         31M        1.9G         30G
Swap:           66G          0B         66G
**********
    Reminder!

    This is a private system. Unauthorized access to the system, or
    unauthorized use of this system is strictly prohibited. By
    continuing, you acknowledge your awareness of (and your agreement
    with) the Responsible Use Policy of CalPoly University.
    Unauthorized users may be subject to criminal prosecution
    under the law, and are subject to disciplinary action under
    the policies of the University. Mining any cryptocoin violates
    campus use policies and such activity will be investigated and reported.
```

Example output from all above commands

Batcher

The batcher allows us to make use of the MPAC lab computers without worrying about file size restrictions on the Cal Poly computer science accounts. It sends files in batches, waits for batch processing to finish, then retrieves the output and moves the input that was processed over to the processed folder. It continues doing this until it has finished iterating through the entire input or after the time limit has been reached.

Setting up a docking job

The general folder structure of a docking job is included within the repository for quick set up.

1) Copy over the “DockingJob” folder from the GitHub repository into your home directory.

```
cp -r [Repo Location]/DockingJob ~/DockingJob
```

2) Add your Receptor Molecule and “idock.conf” file to the “Config” folder within the “Misc” directory of “DockingJob”.

```
cp 5H4I_DOCK.pdbqt ~/DockingJob/Misc/Config
cp idock.conf ~/DockingJob/Misc/Config
```

This folder is where **LABMNGR** will look for your receptor and iDock configuration.

3) Add your ligands to the “Ligands” folder within “DockingJob”.

```
cp MyLigands/* ~/DockingJob/Ligands
```

Supposing you have your ligands currently stored in a folder named MyLigands.

4) Copy over “batcher.py” from the “scheduler” folder of the GitHub repository into your home directory.

```
cp [Repo Location]/scheduler/batcher.py ~/
```

Setting up an Autodock Vina job

This is supposing you want to modify your iDock job to be an Autodock Vina job.

1) Open up the “run_labmngr.sh” and add “--Vina” to the end of the last line in the file.

The end of the line should have looked something like this:

```
[other stuff…] -t $timeout -v
```

Now, the end of the line should look something like this:

```
[other stuff…] -t $timeout -v --Vina
```

2) Make sure that you have your Autodock Vina “conf.txt” file in the “Config” folder within the “Misc” folder of “DockingJob”.

```
ls ~/DockingJob/Misc/Config/conf.txt
```

This should show a file.

And that’s it.

Scheduler

The scheduler is a command line interface for crontab. Because it aims to be simple to use, it does not expose every crontab feature.

No set up is required to use the scheduler, it can be run from anywhere on the system.

Setting up the Scheduler to run the docking job

1) Execute the scheduler.

```
~/scheduler.py
```

2) Choose Option 2.

```
2
```

3) Input the desired start time.

```
23
```

4) Input the location of “docking_job.sh”

```
~/DockingJob/docking_job.sh
```

5) Quit from the scheduler.

```
5
```

Setting up the Scheduler to run a docking job with Vina

Setting up a job with Vina is nearly the same as that for iDock. You will simply need to change the job executable you are running.

1) Execute the scheduler.

```
~/scheduler.py
```

2) Choose Option 2.

```
2
```

3) Input the desired start time.

```
23
```

4) Input the location of “docking_job.sh”

```
~/DockingJob/docking_job_vina.sh
```

5) Quit from the scheduler.

```
5
```

Lab Manager

The lab manager is used to create an ad-hoc cluster on the MPACT lab. It automates the ssh configuration and the configuration of MPiDock through a variety of flags. The lab manager can be used as a standalone program, its simply requires sshing into the MPACT lab directly and does not deal with the problem of storage limits imposed upon lab users. This section describes how to use it without the scheduler or batcher.

Setting up the lab manager for the first time

1) SSH into one of the lab machines

```
ssh user@127x21.csc.calpoly.edu
```

2) Clone the git repo onto the lab machine

```
git clone git@github.com:lrbrantley/Capstone.git
```

3) Navigate to the “LabManager” folder

```
cd [Repo Location]/LabManager
```

4) Generate ssh keys for this machine, click enter when asked for a passphrase

```
ssh-keygen
```

5) Open “bashrc” file

```
vim ~/.bashrc
```

6) Add the following 4 lines to the “bashrc” file in 2 locations, at the top of the file and at the bottom of the file.

NOTE: The reason you need to add it twice is because the default bashrc contains separate locations for non interactive and interactive shells. The top part of the file deals with non interactive and the bottom with interactive. Both types of shells are needed for MPI to work correctly in the MPAC lab.

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib64/openm
pi/lib
PATH=$PATH:/usr/lib64/openmpi/bin
export LD_LIBRARY_PATH
export PATH
```

7) Relaunch your bash shell (note the period at the beginning of the command)

```
. ~/.bashrc
```

8) Quickly check that MPI can be found

```
which mpicc
```

You should get the following line or something similar

```
/usr/lib64/openmpi/bin/mpicc
```

9) Run the first-time setup script in LABMNGR

```
./LabMngr.py --setup
```

How to Operate

Batcher

Running a docking job with the **Batcher** is straightforward. To start it manually, execute the "docking_job.sh" script within the "DockingJob" folder you set up.

```
./DockingJob/docking_job.sh
```

Scheduler

Starting the Scheduler

Execute the “scheduler.py” program.

```
~/scheduler.py
```

Supposing it’s in your home directory.

After that you’ll see a menu with the following options.

```
1. List existing jobs
2. Create a new job
3. Delete an existing job
4. Modify an existing job
5. Exit
```

To select an option, input the number of the option. See below for instructions on each option.

Listing Jobs

Select Option 1.

This lists all the jobs created by the scheduler.

NOTE: All of the jobs are saved in a file named “scheduledjobs.txt” which is stored in the same directory as “scheduler.py”. If you want to move “scheduler.py” you must also move “scheduledjobs.txt”, otherwise jobs will be lost to the ether (actually you can use crontab to edit them but still).

Scheduling a New Job

Select Option 2.

Step 1. It will ask you to input a desired start time. Input the hour you would like the job to start. This uses a 24 hour clock.

For example, our docking job would probably choose this:

```
23
```

Step 2. It will then ask you for the location of the bash script you want to execute. Placing an absolute path is recommended. An absolute path is one that starts with "~" or "/"

For example, our docking job would use this:

```
~/DockingJob/docking_job.sh
```

After performing steps 1 and 2, the **Scheduler** should display the newly updated list of jobs.

Deleting an Existing Job

Select Option 3.

A list of the existing jobs should be displayed. Select the *NUMBER* of the job you wish to delete. After doing so, you should see the newly updated list of jobs.

Modifying an Existing Job

Select Option 4.

A list of the existing jobs should be displayed. Select the *NUMBER* of the job you wish to modify. This will then move you to the modification menu, which will present these options.

```
1. Start Time
2. Job Executable Location
3. Finished Modifying
```

Select any option by entering the *NUMBER* of the option. When you're done making changes, choose Option 3 to save the changes.

Lab Manager

Run the default lab manager

1) Navigate to the “LabManager” folder

```
cd [Repo Location]/LabManger
```

1) Copy all ligands you wish to process into the “Ligand” folder

```
cp MyLigands/* ./Ligand
```

2) Copy the receptor and idock config file into the “Config” folder

```
cp Receptor.pdbqt idock.conf ./Config
```

3) Run LABMNGR

```
./LabMngr.py
```

4) View results in “Output” folder

```
cd ./Output
```

Lab Manager Flags

The following table identifies the various flags and options avaliable to the lab manager. The lab manager can display this table with the “-h” flag:

Flag(s)	Description
-h, --help	Prints this table and exits
-l <LIGAND>	Override the default Ligand Folder. Use this to specify another folder containing ligands you want processed. NOTE: the specified folder should not include an ending / (e.g not good: `-l ./p500/`) (e.g good: `-l ./p500`)
-o <OUTPUT>	Override the default Output Folder. Use this to specify another folder where you want the results to be saved. NOTE: the specified folder should not include an ending / (e.g not good: `-o ./Results/`) (e.g good: `-o ./Results`)
-p <PROCESSED>	Override the default Processed Folder. Use this to specify another folder where you want the finished ligands to be saved. NOTE: the specified folder should not include an ending / (e.g not good: `-p ./Finished/`) (e.g good: `-p ./Finished`)
-v, --verbose	Prints out more information as the program runs. This is recommended for debugging, or if your interested in what stage the program is at.
-s, --setup	Runs the first-time setup script and exits
-n [NODES...]	Directly decides which lab machines to run on (e.g `-n 06 08 13 15 32`)

-x [EXCLUDES...]	Excludes lab machines from running the program on. Best used if you know someone is working on a specific machine, or if a specific machine is running slow. (e.g `-x 05 11`)
--Vina	Run MPIVina instead of MPiDock. This is a distributed version of Autodock Vina.
--hostfile <HOSTFILE>	Overrides the auto-generated hostfile with hardcoded lab machines to use in the cluster
-t <SECONDS>, --timeout <SECONDS>	Specifies the number of seconds MPiDock will run before exiting. This is primarily used by the scheduler to stop MPiDock when it reaches the designated end clock time
-r <RATIO>	Overrides the default block ration of work. The default is 2. This ratio is used to control how large the block of ligands given to each worker node is.
--wpm <WPM>	Overrides the default number of workers nodes per machine. The default is 4. If you increase this number, more workers will be placed on each physical lab machine, but it will use up more system resources. I wouldn't recommend setting this to more than 5.

Accuracy Checking

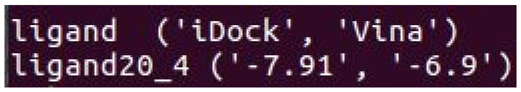
If you desire to check the accuracy of a specific run of iDock or Vina the accuracy script can be used. This is good for comparing iDocks to Vina, or for double checking that certain ligands have not fallen into a local minimum instead the global minimum.

Using the Accuracy Script

To use the accuracy script the only necessary things are two result summary files. These examples use the default accuracy of 1.0.

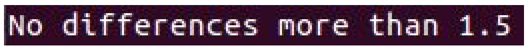
Ex: iDock vs. Vina	./accuracyComparison.py -i ./iDock/i_sum.txt -v ./Vina/v_sum.txt
Ex: iDock vs. iDock	./accuracyComparison.py -i ./iDock/i_sum1.txt ./iDock/i_sum2.txt
Ex: Vina vs. Vina	./accuracyComparison.py -v ./Vina/v_sum1.txt -v ./Vina/v_sum2.txt

Example Output



```
ligand  ('iDock', 'Vina')
ligand20_4 ('-7.91', '-6.9')
```

Showing a Difference



```
No differences more than 1.5
```

No Differences Found

Accuracy Script Flags

The following table identifies the various flags and options available to the accuracy script. The accuracy script can display this table with the “-h” or “--help” flag:

Flag(s)	Description
-h, --help	Prints this table and exits
-v, --Vina	The path(s) after this will be .txt files with Vina result summaries
-i, --iDock	The path(s) after this will be .csv or .txt files with iDock result summaries
-a, --acc Default Value: 1.00	This float is the the maximum that two ligands can be apart. If they are this much or further the script will flag them for review.

Troubleshooting Common Issues

Batcher

Batch runs complete instantly with no output

This probably means that nothing was processed.

1. Make sure there are ligands within the “Ligands” folder.
2. See if anything was added to the “ProcessedLigands” folder. This is more of a confirmation step.
3. Open up the “batcher.log” file to see if there is an error message. The most recent log file is the one without any dates attached to it.
    a. If there is an error, try to resolve it.

Some ligands don’t process

1. See if they run the next time the **Batcher** starts. It’s possible that some unavoidable issue caused a certain number of them to fail the first time around.
    a. If you don’t want to wait for the next scheduled job to start, you can manually start it again.
2. See if they are corrupted.
3. grep the logs to see if there is any mention of their failure.

Scheduler

Job doesn’t run

This potentially means a problem with crontab.

1. List the crontab to see if your job shows up.

   ```
   crontab -l
   ```

   a. This should contain your job.

      i. If it doesn’t reschedule a new job. This also potentially means your “scheduledjobs.txt” file got moved or modified outside of the **Scheduler** program.

Lab Manager

ORTE Error

If you get the following text error:

```
ORTE was unable to reliably start one or more daemons.
```

It means the MPI isn't configured correctly. Make sure you added the 4 lines of code in you bashrc file in the **two** locations that require it, at the top and the bottom. You can run "`which mpic++`" to and "`which mpiexec`" to verify that the lab machines know where the MPI library is.

Incorrect MPI Version

If you get the following text warning:

```
ERROR: Incorrect version of MPI identified.
```

It means the MPI installed in the systems isn't the 1.10 version required. To fix this, you need to get the version 1.10 of openmpi and change the 4 lines in the bashrc file to point to the correct folders within the 1.10 version. As a last resort, contact the creators so the lab manager can be updated to support the new MPI version.

Misc

Issue: Lab 127 is not responding due to power failure

Resolution: The scheduler will try again the next day when hopefully the power is back

Issue: Lab 127 has moved locations or in some way had its normal IP address changed

Resolution: Contact the EE/CSC departments to obtain the new IP address.

Issue: Accidentally deleted MPiDock

Resolution: Redownload from source repository.

Issue: Replaced Master Node Computer

Resolution: Redownload from source repository and rerun setup instructions.

Issue: New Version of iDock has been released

Resolution: Download the new pre-built binary and put it into the folder

`MPiDock/LabManager/src/iDock/`

Make a new folder named iDockOld and save the previous executable in that folder. You should check the that this new version is faster / as accurate / works at all before deleting the old executable. If the new version is not as fast / as accurate / does not work, simply delete the newer executable and put the old executable back into the file path above.

Issue: New Version of iDock was released, how to check accuracy

Resolution: Change the conf file to set a standard seed value. Then use the accuracy script to test with a very low number, like 0.1. If a large percentage of your ligands now differ by that amount with different seeds, the internal algorithm has changed and a closer look should be taken at the potential accuracy. If no ligands or only a very small number appear, the accuracy was not affected.