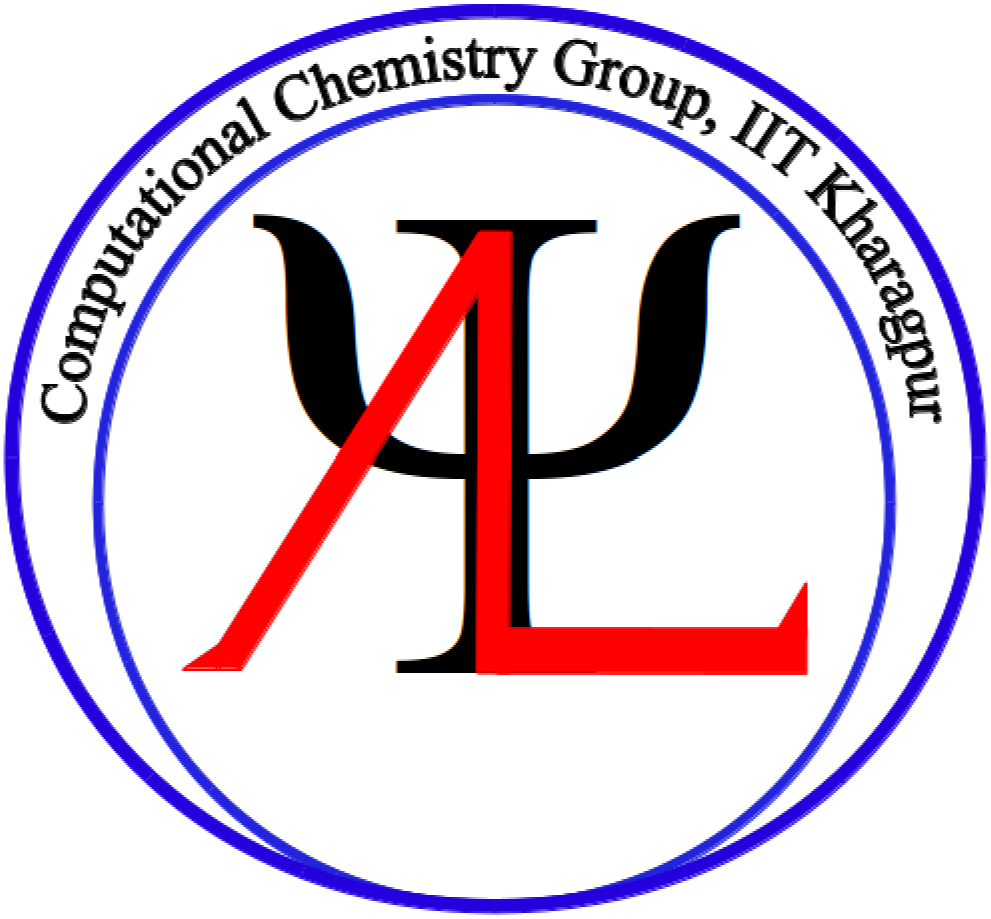
# AnoopLab Wiki

**Saikat**

**May 14, 2022**

# SLURM

Hello there !!



We are a theoretical and computational chemistry research group at the Indian Institute of Technology Kharagpur, India.

**Want to know more about our research ?** Click here

This preliminary guide is created for the following reasons

- Help beginners to quick start computational chemistry
- Set up different softwares in the IIT KGP supercomputing facility.
- Some essential tutorials about linux and programming

**Want to help?**

Please feel free to contribute. Send PR to our github repo.

# CHAPTER ONE

# QUANTUM CHEMISTRY PACKAGES

Currently we can offer help for these softwares

- Gaussian
- ORCA
- Turbomole
- xtb
- PyAR
- Psi4
- CFour
- Mopac

CHAPTER

TWO

# EXTERNAL LIBRARIES/PACKAGES

- Conda Python
- OpenMPI
- OpenBLAS
- Scalapack
- Lapack

## 2.1 SLURM Queuing system

The Slurm is a Workload Manager, formerly known as Simple Linux Utility for Resource Management (SLURM), or simply Slurm.

If you are not familiar with this please read about this first before proceed to the tutorials. There are plenty of resource available in the web. Go ahead.

Here are some links

https://slurm.schedmd.com/overview.html

**Note:** You should also follow the IIT Kgp hpc website specially for the paramshakti configuration and quick start guide.

http://www.hpc.iitkgp.ac.in

**Date** 18.01.2021

**Authors**

- saikat R

## 2.2 ORCA

This guide deal with setup ORCA package in paramshakti supercomputing facility in user's own account

**Contents**

### 2.2.1 Getting the Program

Goto the orca forum website. Register yourself if you are new or login. Download the preferred version for Linux x86 and copy it to your account. Now create a `apps/` directory inside your `home` and move the tar file there and untar it.

```
mkdir -p ~/apps
mv orca_4_2_1_linux_x86-64_shared_openmpi314.tar.xz ~/apps
cd ~/apps
tar -xvf orca_4_2_1_linux_x86-64_shared_openmpi314.tar.xz
```

### 2.2.2 Installation

ORCA is distributed as precompiled binary files. So there is nothing to install other than the environment paths.

To run ORCA in parallel it requires OpenMPI package library. So either you have to load OpenMPI by `module` system in paramshakti (easy way!!) or you have to install OpenMPI. Please visit the OpenMPI installtion for building it from source. In this tutorial we will use the pre-installed OpenMPI library in paramshakti for ORCA. The supported MPI version are usually mentioned in the orca folder name when you extract it e.g `orca_4_2_1_linux_x86-64_shared_openmpi314` means ORCA version is 4.2.1 and supported OpenMPI version is 3.1.4

#### Module File setup

Packages in paramshakti are usually maintained by using the `module` system. You can easily maintain multiple version of same/different packages with this system. First create a `modulefiles` directory inside your `\home`

```
mkdir -p ~/modulefiles/orca
cd ~/modulefiles/orca
```

create a file with your preferred name e.g `421` (usually same with the orca version)

Here is a sample modulefile(`421`)

```
#%Module1_0
## ORCA modulefile
##
```

(continues on next page)

(continued from previous page)

```
proc ModulesHelp { } {

        puts stderr "\tAdds ORCA to your environment"
}

module-whatis   "Adds ORCA to your environment"
module try-add compiler/gcc/8.3.0
module try-add openmpi3/3.1.4

setenv ORCAPATH /home/<username>/apps/orca_4_2_1_linux_x86-64_shared_openmpi314

prepend-path    PATH    /home/<username>/apps/orca_4_2_1_linux_x86-64_shared_openmpi314
prepend-path    LD_LIBRARY_PATH /home/<username>/apps/orca_4_2_1_linux_x86-64_shared_
↪openmpi314
```

**Note:**

- `module try-add` will load the gcc (version 8.3.0) and openmpi (version 3.1.4) these two are pre-installed.
- `setenv ORCAPATH` path to your orca directory. In this case it is located under `apps` subdirectory inside the `\home`
- `prepend-path` usually the same directory required for ORCA

Modify these variables according to your account

Now modulefile for ORCA is created and module path needs to be set in the environment `~/.bashrc`

open `~/.bashrc` file with an editor and paste this

```
export MODULEPATH=$MODULEPATH:/home/<username>/modulefiles
```

save it and `source` it

```
source ~/.bashrc
```

try this in command line

```
module load orca/421
```

if no error is coming out try

```
$ which orca
```

it should print out

```
$ /home/<username>/apps/orca_4_2_1_linux_x86-64_shared_openmpi314/orca
```

Congratulation !! You have successfully installed ORCA in your account.

Follow the same steps if you want to install a different version Create a new module file in the same location and add the `paths` accordingly. Also, don't forget to load the new version in the submit script.

Still having problem ? Don't worry, create an issue with proper error output here.

We are happy to help!!

### Submit Script for SLURM in paramshakti

It's time to test the ORCA Program.

Go to your `scratch` directory and submit a test job. Here is a sample submit script.

```
#!/bin/bash
#SBATCH -J orca-testjob     # name of the job
#SBATCH -p standard-low     # name of the partition: available options "standard,␣
↪standard-low, gpu, hm"
#SBATCH -n 16                   # no of processes or tasks
#SBATCH -t 1:00:00          # walltime in HH:MM:SS, Max value 72:00:00

#list of modules you want to use, for example
module load orca/421
#name of the executable
exe=$ORCAPATH/orca
#run the application
$exe opt.inp >& result.out
```

**Date** 17.12.2021

**Authors**

- saikat R

## 2.3 xtb

This guide deal with setup XTB package in paramshakti supercomputing facility in user's own account

**Contents**

### 2.3.1 Getting the Program

**Date** 18.01.2021

**Authors**

- AnoopLab

## 2.4 Conda Python

Anaconda is a python distribution. It also has its own package manger `conda`. You can install python and other softwares/packages through conda. Main advantage is that it comes with most of the necessary packages. Here we will tell you how to install anaconda python in paramshakti.

### 2.4.1 Installation

Goto the download page for anaconda individual edition. Download the `Anaconda3-20XX.XX-Linux-x86_64.sh` file and copy it to your paramshakti `\home` directory.

Run it

```
bash Anaconda3-20XX.XX-Linux-x86_64.sh
```

Accept the terms and condition by typing `yes` You can set up your custom install directory or default location is also fine. It will take some time to install

At the end it will ask you

```
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
```

Although the default is `no` type `yes`.

Now logout and login again. You will find there is `(base)` in front of login prompt. you can disable it by

```
conda config --set auto_activate_base false
```

Don't worry it will go away next time you login

you can also notice there are some config in your `~/.bashrc` file written by conda installer don't remove it. If you install it in your home directory it will be something like this

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/home/username/anaconda3/bin/conda' 'shell.zsh' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/home/username/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/home/username/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/home/username/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

Now you can activate base environment by

```
conda activate base
```

and deactivate it by

```
conda deactivate
```

You can also install different environment with required package. Details documentation is here.

**Tip:** We strongly recommend to create new environment each time when you work on a different projects.

You can read more documentation here

https://docs.anaconda.com/anaconda/user-guide/getting-started/

**Date** 18.01.2021

**Authors**

- saikat R

## 2.5 Openmpi

This guide leads to the setup openmpi in paramshakti supercomputing facility in user's own account

**Contents**

### 2.5.1 Requirement

gcc, g++, and gfortran compiler

In paramshakti try this

```
module avail | grep gcc
```

This will output all the available gcc versions. Load the latest (preferably) version or any other depending on your requirement. I loaded gcc 10.2.0

```
module load compiler/gcc/10.2.0
```

### 2.5.2 Getting the Program

Goto the openmpi website. Register yourself if you are new or login. Download the required version for Linux x86 and copy it to your account. Now create a `apps/` directory inside your `home` and move the tar file there and untar it. I am writing this tutorial for openmpi v *4.1.1*.

```
mkdir -p ~/apps
mv  ~/apps
cd ~/apps
tar -xzvf openmpi-4.1.1.tar.gz
```

This will create a directory *openmpi-4.1.1*. It amy be different depending upon the version. Now create an installation directory. I created with the name *openmpi411* and copy the path to this directory. For my case the path will be */home/<username>/apps/openmpi411*

Now go to the *openmpi-4.1.1* directory and follow the code below

```
cd openmpi-4.1.1
./configure --prefix=/home/<username>/apps/openmpi411 CC=gcc CXX=g++ F77=gfortran␣
↪FC=gfortran
make -j10
make install
```

This will take some time. If it's finished then the installation is successfully, otherwise check the error log for details. After finishing this go to *openmpi411* directory. You wil find several directories, *bin*, *lib*, *etc*, *include*, *share*.

Next step is to create a modulefile for openmpi 4.1.1.

#### Module File setup

Packages in paramshakti are usually maintained by using the `module` system. You can easily maintain multiple version of same/different packages with this system. First create a `modulefiles` directory inside your `\home`

```
mkdir -p ~/modulefiles/openmpi
cd ~/modulefiles/openmpi
```

create a file with your preferred name e.g `411` (usually same with the openmpi version)

Here is a sample modulefile(`411`)

```
#%Module1_0
## ORCA modulefile
##
proc ModulesHelp { } {

puts stderr "\tAdds OPENMPI to your environment"
}

module-whatis   "Adds OPNMPI to your environment"

module try-add compiler/gcc/10.2.0

set openmpiversion  4.1.1
set MPI_HOME    /home/17cy91r04/apps/openmpi411

```

(continues on next page)

(continued from previous page)

```
prepend-path    LD_LIBRARY_PATH /home/<username>/apps/openmpi411/lib
prepend-path    PATH /home/<username>/apps/openmpi411/bin
```

Modify these variables according to your account

Now modulefile for openmpi is created and module path needs to be set in the environment `~/.bashrc`

open `~/.bashrc` file with an editor and paste this

```
export MODULEPATH=$MODULEPATH:/home/<username>/modulefiles
```

save it and `source` it

```
source ~/.bashrc
```

try this in command line

```
module load openmpi/411
```

if no error is coming out try

```
$ which mpirun
```

it should print out

```
$ ~/apps/openmpi411/bin/mpirun
```

Congratulation !! You have successfully installed ORCA in your account.

**Tip:** If anything goes wrong, you have find what is the problem and after fixing the issue you have to follow all the steps from *configure* step. Before configuring do *make clean* first, otherwise it will take the old configuration setup.

Still having problem ? Don't worry, create an issue with proper error output here.

We are happy to help!!

**Date** 14.05.2022

**Authors**

- saikat R

## 2.6 How to Contribute

This website is build with the help of readthedocs.org and Sphinx with `sphinx_rtd_theme`

You can read their documentation for details.

### 2.6.1 Installation

Clone the github repo

```
$ git clone https://github.com/Saikat248/anooplab-docs.git
```

and install this two packages

```
$ pip install sphinx
$ pip install sphinx_rtd_theme
```

Navigate to `anooplab-docs/docs` directory

there is a `Makefile` and start

```
$ make html
```

It will create a `_build` directory and the `html` files

```
$ firefox _build/html/index.html
```

The whole website will run locally. Go ahead start editing the documents and before pushing make sure the build is successful.

Enjoy !!

## 2.7 License

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International (**CC BY-SA 4.0**). To view a copy of this license, visit creative commons or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

## 2.8 Credits

AnoopLab Research group