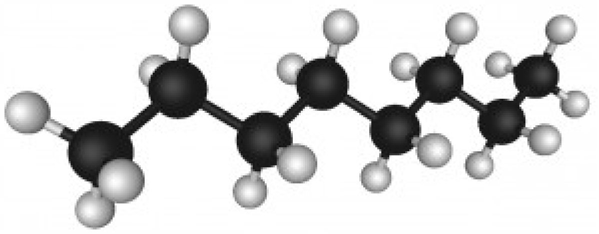
Cluster-docs
Casino

Casino - Quantum Monte Carlo on SCC



Quantum Monte Carlo (QMC) is an exciting, modern computational technique which allows us to approximately solve the equations of quantum mechanics – which are far too complicated to solve exactly – and in most cases get *essentially the right answer*. Its key advantage is that, unlike with all other known highly accurate techniques, it is still possible to do the calculations for relatively large systems with many atoms – providing you have a big enough computer (and QMC is quite capable of using the biggest). Used in combination with other cheaper methods, it provides researchers with the final building block in an atomic scale micro-laboratory on their computer which can be used to simulate small parts of the real world. Its practical application to real-life problems – via a general, widely-distributed computer program system – has been pioneered in the United Kingdom by members of the Cambridge University Theory of Condensed Matter Group working out of the Cavendish Laboratory, famous throughout the world as the venue for amazing historical discoveries such as the electron and DNA. A number of their collaborators at various other universities – whose work is also discussed here – have also made fundamental contributions, both to the software, and to the field in general.

Remarks on this manual:
The given example-commands must be executed in an Terminal-Window on the SCC so you need a valid user-account on the Cluster.
If there are no further instructions, run the commands in the directory with your input files. User-defined parameters, included in the example-commands are indicated with a $-sign.

Using Casino on SCC

1. General Cluster HPC-usage:

The SCC consists of one main-computer, called “headnode”, from where you can start and manage your calculations, and numerous slave-nodes, which are executing the calculation in parallel. In general the calculation is started simultaneously on all reserved slave-nodes

by the “qsub” command, which implies you are submitting a computing job to the cluster batch system. By running the “qsub” command, the SCC batch system figures out the free nodes for you and starts the parallel execution on the reserved hardware.
The headnode and the slave-nodes all use the same working-directories and the same software directories, whereby the input- and output-datasets of your calculations are managed centrally from the headnode. The scientific software packages are installed as modules, which must be loaded on each node before the software can be started.

2. The Casino-Package:

Casino can be accessed via a module called `Casino/2.13-foss-2017b.`
It has to be loaded in the submit script as it is done in the example scripts.

Example scripts can be found in the following directory:

```
~/cluster-public/casino/usage_instructions/examples
```

We recommend to store your input files in separate directories since Casino produces output files with the same name on each run.

Basic Workflow to run Casino in parallel:

Create a new directory for your input data or copy one of the examples. Create a new submit file in the same directory as your input data or modify an existing one according for your needs (see below). The name of the script is assumed to be “casino.sub” in the following part.

A job can be started via

```
$ qsub casino.sub
```

When the job is finished, an output file will be generated.

As long as the job is running, you can see it with the command

```
$ qstat -a -u _your_user_account_
```

The status Q means queued, so the job is waiting for free resources. R means running. Once this is the case, browse to http://.../ganglia to the a live view of the resource usage of your computation. You can find out on which node(s) your job is running by calling

```
$ qstat -n _job_id_
```

(The job id can be found via the first qstat command)
Your results will be stored in the same directory.
If you want to delete a running job, execute the following command:

```
$ qdel _jobid_
```

(we need this to abort the reservation when pbstop shows the reserved hardware, but ganglia shows no real load)

Please be aware that to start a job a second time, you will have to delete the file vmc.hist first

---

Example:

Copy the example in your home directory:

```
$ cp -r ~/cluster-public/casino/usage_instructions/examples/argon
~/
and change into that directory
```

```
$ cd argon
```

Please edit the file casino.sub with the editor of your choice, for example nano:

```
$ nano casino.sub
```

Enter your email address in the line starting with `#PBS -M`

Lines beginning with #PBS are commands for the batch systems. The other lines starting with a # are comments and are ignored.

The file could look like the following example:

```
#The output will be written to this file
#PBS -o output_casino.txt
#How long should the job run maximal, on how many nodes will it
run and how many processors per node should be allocated. It is
best to reserve complete nodes, this means try to keep the value
at 24 if running parallelized jobs
#PBS -l walltime=4:00:00,nodes=2:ppn=24
#The queue that is used
#PBS -q scc-default   (this que named “scc-default” makes nodes
01-15 available which have 24 cores each)
#A description to distinguish your jobs
#PBS -N _job_description_
#Where should messages be sent to
#PBS -M _your_email_address_here_
#When to send messages: b(egin), a(bort), e(nd)
#PBS -m bae
```

```
#Join output and error log
#PBS -j oe

#The following “module” commands load the modules for the
toolchain foss (free and open source, the GNU Compiler and
OpenMPI) and Casino itself. The “source” command is necessary to
make the module command available in remote sessions

source /etc/profile
module add foss
module add Casino/2.13-foss-2017b
cd $PBS_O_WORKDIR   (change to the directory on the compute node
where the submit file is located)
mpirun --hostfile $PBS_NODEFILE -np 48 casino
#the number behind np has to be the product of the nodes reserved
in the #PBS -l line times ppn
```

The example will run within seconds, so a walltime of 4 hours is more than enough. If you run longer computations, you can increase this value to up to 48 hours, if needed.

Start the Calculation:

Submit the file to the batch system via the command:

```
$ qsub casino.sub
```

After submitting your job, you can observe it’s reserved hardware as written above.

Inspect the results

The files out and casino_output.txt (if not changed in the submit file) will be generated. Inspect them to see if Casino ran properly.