

Running Parallel Gaussian on the Shared Research Computing Resources

Introduction

TCP Linda is a parallel execution environment to parallelize Gaussian jobs. By default, Gaussian jobs are single processor core jobs. TCP Linda must be used to parallelize Gaussian. In order to use TCP Linda you must configure your Gaussian input file to use multiple processors (and possibly multiple nodes). For multiple node jobs this requires that you know which nodes you will be running on. However, this information is not known ahead of time. We have developed a script that will configure your job for you. Please follow the instructions below.

i **Linda not available on all systems**
Linda is only available on DAVinCI.

i **Not all Gaussian calculations can be run in parallel**
Please see the [Gaussian documentation](#) about the types of calculations that can be done in parallel.

Configure SSH Keys

In order for Linda to communicate between nodes you must enable passwordless SSH (SSH keys). Please see our [FAQ](#) for instructions.

✓ You only need to set up SSH keys once. It will work for all subsequent jobs for the lifetime of your account.

Configure Gaussian

In order to run a Gaussian job on our clusters you must first load the Gaussian module:

```
module load gaussian
```

Subsequently, you should source the Gaussian command file to set up your environment:

tssh users:

```
source $g09root/g09/bsd/g09.login
```

bash users:

```
source $g09root/g09/bsd/g09.profile
```

Build Your Gaussian Job

You must have a Gaussian input file. For our example we will use a file called *example.gjf*.

This input file must have, as its first lines, the number of processors per node and a list of the node names. An example is shown in the [Linda](#)

documentation.

An example of these lines looks like this (taken from the Linda documentation)

```
%NProcShared=8
%LindaWorkers=node1,node2,node3,node4
```

As stated previously, the node names for your job will not be known ahead of time. So this must be configured dynamically. To build your job submission script such that these jobs can be configured dynamically, use the following command:

```
/opt/apps/scripts/create-gaussian-jobscrip inputfile number_of_nodes
```

Using our example input file *example.gjf* and 6 nodes, the command would look like this:

```
/opt/apps/scripts/create-gaussian-jobscrip example.gjf 6
```

This will create a SLURM job script named *example.slurm* in your current working directory, using *example.gjf* as the input file and using 6 nodes. It will build a submission script that will set the Gaussian parameters necessary to run the parallel job.

To submit the job, simply use `qsub` as follows:

```
sbatch ./example.slurm
```

Check the contents of the job submission script for the location of your scratch directory (`GAUSS_SCRDIR`) and the location of your output file.