

Gaussian



Gaussian ist ein kommerzielles Programm aus dem Bereich der Computerchemie zur Berechnung und Modellierung von Elektronenstrukturen.

Gaussian ist von Hause aus nur mittels OpenMP parallelisiert, d.h. sollen mehrere Cores für eine Rechnung genutzt werden, müssen diese Cores auf den gleichen Speicherbereich zugreifen können.

Eine Parallelisierung mit MPI ist *nicht* implementiert.

Eine Beschleunigung durch GPGPUs ist *nicht* implementiert.



Nach unserer Erfahrung, wird Gaussian nur sehr bescheiden schneller mit steigender Anzahl an verwendeten Cores. Außerdem kommt es immer mal wieder vor, dass eine Rechnung mit einer bestimmten Zahl an Cores nicht konvergiert. Wir vermuten hier eine ungenaue Implementierung seitens der Gaussian-Entwickler.

- Offizielle Webseite: <http://www.gaussian.com>
- **Gaussian 09 Users's Reference:** http://www.gaussian.com/g_tech/g_ur/g09help.htm

Job Script für die SGI UV2000 (hilbert1, hilbert2)

Gaussian09_uv2000.job

```
#!/bin/bash

#PBS -l select=1:ncpus=2:mem=4gb
#PBS -l walltime=12:00:00
#PBS -r n
#PBS -N ###JOB NAME###
#PBS -A ###PROJECT NAME###

GaussianInputFilename=###GAUSSIAN INPUT FILE###

FileBasename=$(basename $GaussianInputFilename)
GaussianOutputFilename="${FileBasename%.*}.${PBS_JOBID}.log"

user=`whoami`

#make unique scratch directory on GPFS filesystem
SCRATCHDIR=/gpfs/scratch/$USER/$PBS_JOBID
mkdir -p "$SCRATCHDIR"

#load Gaussian Environment and set scratch directory
module load Gaussian/g09
export GAUSS_SCRDIR=$SCRATCHDIR

#some (useful?) output
LOGFILE=$PBS_O_WORKDIR/$PBS_JOBNAME"."$PBS_JOBID".log"
cd $PBS_O_WORKDIR

echo "$PBS_JOBID ($PBS_JOBNAME) @ `hostname` at `date` in "$PBS_O_WORKDIR" START" > $LOGFILE
echo "`date +%d.%m.%Y-%T`" >> $LOGFILE

echo >> $LOGFILE
echo "GLOBAL PARAMETERS">> $LOGFILE
echo "-----" >> $LOGFILE
echo "Node      : "$HOSTNAME >> $LOGFILE
echo "Arch      : "$ARCH >> $LOGFILE
echo "-----" >> $LOGFILE
echo "RunDir     : "$PBS_O_WORKDIR >> $LOGFILE
echo "InputFile  : "$GaussianInputFilename >> $LOGFILE
echo "OutputFile : "$GaussianOutputFilename >> $LOGFILE
echo "ScratchDir : "$GAUSS_SCRDIR >> $LOGFILE
echo "GaussianDir: "$GAUSS_EXEDIR >> $LOGFILE

#execute gaussian IN the (fast) scratch directory
cd $SCRATCHDIR
g09 < $PBS_O_WORKDIR/$GaussianInputFilename > $GaussianOutputFilename

#copy files back from scratch directory
cp -r "$SCRATCHDIR"/* $PBS_O_WORKDIR/.
cd $PBS_O_WORKDIR

#print the last known statistics of the job (memory usage, cpu time, etc...)
echo >> $LOGFILE
qstat -f $PBS_JOBID >> $LOGFILE

echo "$PBS_JOBID ($PBS_JOBNAME) @ `hostname` at `date` in "$RUNDIR" END" >> $LOGFILE
echo "`date +%d.%m.%Y-%T`" >> $LOGFILE
```

In diesem Beispiel werden **2 Cores** und **4 GByte RAM** angefordert. Bitte denken Sie daran, in dem entsprechenden Gaussian-Input auch die Anzahl der genutzten Cores und des Speichers zu setzen:

Ausschnitt Gaussian-Input

```
[...]  
%nproc=2  
%mem=4096MB  
[...]
```

###JOB NAME### und ###PROJECT NAME### müssen noch entsprechend von Ihnen angepasst werden, ebenso wie *Walltime* (die hier beispielsweise auf 12 Stunden limitiert ist).