duhpc

High Performance Computing Cluster

User Guidelines and Best Practices

Department of Chemistry · University of Delhi
Version 1.0 | May 2026
HPC Administrator: ighani@ducc.du.ac.in

Quick Reference — Key Rules

- Use compute partition for ALL Gaussian jobs (max 16 CPUs, 5 days walltime)
- Use mpi partition ONLY for multi-node parallel jobs (GROMACS, NAMD, Amber MPI)
- Always add %Chk line to Gaussian input files for checkpoint/restart support
- Storage quota: 1 TB per user — clean old files regularly
- Never run jobs directly on login node — always use sbatch or srun
- For help: email ighani@ducc.du.ac.in

1. Cluster Overview

The duhpc HPC cluster is a shared research computing facility serving all faculty and research scholars of the Department of Chemistry. It supports computationally intensive research in quantum chemistry, molecular dynamics, materials science, crystallography, and related fields.

Resource	Specification	Notes
Compute nodes	cn01, cn02, cn03	3 × Dell PowerEdge R640
CPU cores	48 cores per node (144 total)	2 × 24-core Intel Xeon
RAM per node	170-185 GB	Available to SLURM jobs
GPU node	gpu1 — 2× NVIDIA V100 32GB	Dell PowerEdge R740
Storage	130 TB Lustre (/scratch)	High-speed parallel filesystem
Scheduler	SLURM 18.08	Job submission and management
Interconnect	100 Gbps InfiniBand	Fast node-to-node communication

2. Connecting to the Cluster

2.1 SSH Login

Connect to the cluster master node using SSH from within the university network:

```
ssh username@10.58.5.101
```

```
# From off-campus (via VPN first):
ssh username@10.58.5.101
```

Important: Do NOT run computational jobs on the master node!
The master node is for job submission and file management only.
Running jobs directly on master slows down the cluster for all users.
Always submit jobs using sbatch or srun commands.

2.2 File Transfer

Transfer files using SCP or SFTP from your local computer:

```
# Upload file to cluster
scp myfile.com username@10.58.5.101:/scratch/home/username/
```

```
# Download results from cluster
scp username@10.58.5.101:/scratch/home/username/output.log ./
```

```
# Use WinSCP or FileZilla for graphical file transfer (Windows)
```

3. Storage — Important Rules

3.1 Filesystem Layout

Path	Purpose	Quota	Notes
/scratch/home/username	Your home directory	1 TB	Use for all job files
/scratch/apps	Installed software	Shared	Read only — do not modify
/scratch/apps/templates	Job script templates	Shared	Copy to your directory
/home/username	NFS home	5 TB shared	For personal files only
/tmp on nodes	Temporary files	No limit	Deleted after job ends

3.2 Storage Quota Policy

- Each user has a 1 TB hard limit on /scratch
- Warning email sent when you reach 800 GB (80% of quota)
- At 1 TB limit your jobs will FAIL with disk full errors
- **Important:** Check your usage regularly:

```
lfs quota -u $USER /scratch
```

3.3 Storage Best Practices

- Delete old checkpoint files (.chk) after jobs complete successfully
- Archive completed research data to /home or external storage
- Do not store raw experimental data on /scratch — it is for computation only
- Large trajectory files (GROMACS .xtc, Amber .nc) should be deleted or compressed after analysis
- **Reminder:** Run storage check monthly:

```
du -sh /scratch/home/$USER/*/
```

4. Partitions — Which One to Use

Choosing the correct partition is critical for efficient cluster use. Using the wrong partition wastes resources and increases queue wait times for all users.

Partition	Max Time	Max CPUs/Job	Purpose	Use For
compute (default)	5 days	16 CPUs	Single-node CPU jobs	Gaussian, NWChem, COMSOL, Olex2
mpi	5 days	48 CPUs	Multi-node parallel	GROMACS, NAMD, Amber MPI, LAMMPS
short	4 hours	8 CPUs	Quick tests	Testing scripts, short calculations
gpu	2 days	48 CPUs	GPU accelerated	Amber CUDA, GROMACS GPU, NAMD GPU

⚠ Common Mistake — Using mpi partition for Gaussian jobs
Many users select mpi partition thinking it provides more walltime.
The compute partition NOW provides 5 days walltime — same as mpi.
The mpi partition requires minimum 2 nodes and is exclusively for parallel MPI applications. Single-node Gaussian jobs submitted to mpi will be automatically redirected to compute partition by the system.

5. Gaussian 16 — Best Practices

5.1 CPU Selection — Critical for Efficiency

This is the most common mistake on duhpc. Requesting too many CPUs does NOT make Gaussian faster — it often makes it slower due to communication overhead.

Calculation Type	Recommended CPUs	Memory	Reason
Geometry optimisation	4-8 CPUs	20-32 GB	Poor parallel scaling above 8
Single point energy (DFT)	8-12 CPUs	32-48 GB	Good scaling to 12 CPUs
Frequency calculation	8-12 CPUs	32-48 GB	Moderate scaling
TD-DFT / Excited states	4-8 CPUs	32-48 GB	Poor parallel scaling
MP2 / CCSD(T)	8-16 CPUs	48-60 GB	Good scaling, memory intensive
NMR calculation	4-8 CPUs	20-32 GB	Poor parallel scaling
Large molecule (>100 atoms)	12-16 CPUs	48-60 GB	Maximum allowed on compute

Cluster CPU Limit: Maximum 16 CPUs per Gaussian job on compute partition
Research consistently shows Gaussian rarely benefits above 12-16 CPUs.
Using 8 CPUs instead of 24 CPUs = 3x more jobs in the queue simultaneously

= shorter wait times for everyone including yourself.

5.2 Correct Gaussian Input File Format

Always include these lines at the top of your Gaussian input file:

```
%Chk=/scratch/home/username/jobname.chk
%Mem=28GB
%NProcShared=8
# B3LYP/6-31G* Opt

Title: My geometry optimisation

0 1
C   0.000   0.000   0.000
...
```

Line	Purpose	Example
%Chk	Checkpoint file — REQUIRED for restart (script adds automatically if missing)	%Chk=/scratch/home/user/aspirin.chk
%Mem	Memory — must be ~4 GB LESS than --mem in job script	%Mem=28GB (if --mem=32G in script)
%NProcShared	CPUs — must match --cpus-per-task (script fixes automatically if wrong)	%NProcShared=8 (if --cpus-per-task=8)
Route card	Calculation type	# B3LYP/6-31G* Opt Freq
Blank line	Required separator	(empty line)
Title	Description of job	Aspirin geometry optimisation
Blank line	Required separator	(empty line)
Charge Mult	Charge and spin multiplicity	0 1 (neutral, singlet)

⚠ Important: %Mem must be ~4 GB LESS than --mem in job script

--mem=32G in script → %Mem=28GB in .com file
--mem=40G in script → %Mem=36GB in .com file
--mem=60G in script → %Mem=56GB in .com file

If you forget %Chk or get %NProcShared wrong — the script fixes it automatically.

5.3 Automatic Protection Features — Job Script v3.0

The duhpc Gaussian job script includes several automatic features to protect your work:

Feature	What It Does	What You See in Output
Auto-restart	Detects .chk file — restarts from checkpoint automatically	✓ RESTART MODE: Checkpoint found!
Auto %Chk inject	Adds %Chk if missing from .com file	⚠ No %Chk found — auto-adding...
NProcShared fix	Corrects CPU count in .com if it mismatches	⚠ WARNING: %NProcShared=24 ... Fixing...
Walltime warning	Saves .chk 5 minutes before job is killed	⚠ WALLTIME LIMIT APPROACHING!
Failure diagnosis	Shows specific fix for each error type	Status: ✗ FAILED — Memory error + fix
Scratch on failure	Preserves scratch directory for debugging	Scratch preserved: /scratch/gaussian/...
Results summary	Shows key output from log file	Stationary point found / SCF Done

5.4 Restarting After Power Failure or Walltime

The duhpc cluster has experienced power failures. The job script handles restarts completely automatically:

Restart Procedure — Only 1 Step Required!

sbatch gaussian_job.sh

The script automatically:

1. Detects the .chk file in your job directory
2. Creates correct restart input with Opt=Restart
3. Includes proper blank lines and charge/multiplicity
4. Runs Gaussian with the restart input

No manual editing of input files needed.

The restart input created by the script looks like this:

```
%Chk=aspirin_opt.chk
%Mem=28GB
%NProcShared=8
# B3LYP/6-31G* Opt=Restart Integral=UltraFine

Restart from checkpoint — Job 4947

0 1
```

5.5 Standard Gaussian Job Script

Copy the template and edit only the two marked lines. The script is at:

```
/scratch/apps/templates/gaussian_job.sh
```

```
#!/bin/bash
#SBATCH --job-name=Gaussian
#SBATCH --partition=compute        # Always use compute for Gaussian
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=8          # ← Edit: 4-16 CPUs (see table above)
#SBATCH --mem=32G                  # ← Edit: %Mem in .com = this minus 4G
#SBATCH --time=5-00:00:00          # Max 5 days on compute partition
#SBATCH --output=%x_%j.out
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --mail-user=your@email.com # ← Edit: your email
#SBATCH --signal=B:USR1@300        # Saves checkpoint before walltime

JobFile=molecule    # ← Edit: your input filename without .com
```

5.6 Understanding Job Output Messages

Message in Output	Meaning	Action Required
✓ RESTART MODE: Checkpoint found!	Restarting from previous run	None — automatic
Fresh start — no checkpoint found	First run of this job	None — normal
Status: ✓ COMPLETED SUCCESSFULLY	Job finished normally	Check results
Stationary point found	Geometry optimisation converged	Run frequency next
Status: ✗ FAILED — Memory error	Not enough RAM	Increase --mem and %Mem
Status: ✗ FAILED — SCF convergence	SCF did not converge	Add SCF=(MaxCycles=512,XQC)
Status: ✗ INCOMPLETE — Opt not conv.	More steps needed	Resubmit — auto-restarts
Status: ✗ FAILED — Gaussian did not start	Input file error	Check .com file format
⚠ WALLTIME LIMIT APPROACHING	5 min before kill	Resubmit with longer --time

6. Molecular Dynamics — GROMACS, NAMD, Amber

6.1 Which Partition to Use

Software	Single Node	Multi-Node	GPU	Recommended Partition
GROMACS	☑ Yes	☑ Yes	☑ Yes	compute (1 node) or mpi (2-3 nodes)
NAMD	☑ Yes	☑ Yes	☑ Yes	compute (1 node) or mpi (2-3 nodes)
Amber pmemd	☑ Yes	☑ Yes	☑ Yes	gpu (with CUDA) or compute
Amber sander	☑ Yes	✗ No	✗ No	compute
LAMMPS	☑ Yes	☑ Yes	⚠ Limited	mpi for large systems

6.2 MD Job Recommendations

- Always run equilibration on single node (compute) before production
- Use GPU partition for Amber CUDA — 10-50x faster than CPU
- For GROMACS multi-node, use mpi partition with 2-3 nodes maximum
- Save trajectory files to /scratch — they can be very large
- Set up checkpointing in GROMACS with -cpi flag to restart from crash

7. Queue Management and Fair Use

7.1 Useful SLURM Commands

Command	Purpose	Example
sbatch	Submit a job	sbatch gaussian_job.sh
squeue	View job queue	squeue -u $USER
scancel	Cancel a job	scancel 4921
sinfo	View cluster status	sinfo
scontrol show job	Job details	scontrol show job 4921
sacct	Job history	sacct -u $USER --starttime=today
srun	Run interactive job	srun -N1 -p short --pty bash

7.2 QOS Limits — Fair Use Policy

User Type	Account	Max Running Jobs	Max Pending Jobs	Max CPUs
Normal users	root	2 jobs	5 jobs	16 CPUs per job
Research group	research_group	3 jobs	8 jobs	16 CPUs per job
GPU jobs	any	1 GPU job	2 GPU jobs	48 CPUs

7.3 Fair Use Guidelines

- Do not submit hundreds of jobs simultaneously — use job arrays instead
- Cancel jobs you no longer need — free resources for others
- Do not request more CPUs or memory than your job actually needs
- Test your job script with a short run on the short partition first
- If your job needs more than 5 days, contact the HPC administrator
- Do not share your password or login credentials with others

Job Array Example — Submit 10 similar jobs efficiently:

```
#SBATCH --array=1-10
#SBATCH --partition=compute
```

Then use $SLURM_ARRAY_TASK_ID to differentiate inputs.
This is much better than submitting 10 separate job scripts.

8. Do's and Don'ts

☑ DO	✗ DON'T
Use compute partition for Gaussian	Use mpi partition for single-node Gaussian
Add %Chk to all Gaussian inputs	Run Gaussian without checkpoint file
Request only CPUs you need (4-16)	Request 24-48 CPUs for Gaussian
Check storage usage monthly	Ignore storage quota warnings
Use job templates in /scratch/apps/templates	Write job scripts from scratch
Test scripts on short partition first	Submit untested scripts to compute
Cancel jobs you no longer need	Leave failed jobs pending in queue
Email admin for software requests	Install software without permission
Save important results to /home	Keep all data only on /scratch
Report hardware issues promptly	Continue using failing hardware silently

9. Available Software

Application	Version	Category	Module/Path	Partition
Gaussian 16	g16	Quantum Chemistry	/scratch/apps/gaussian/g16	compute
Gaussian 09	g09	Quantum Chemistry	/scratch/apps/gaussian/g09	compute
NWChem	7.2.2	Quantum Chemistry	module load nwchem	compute/mpi
GROMACS	Latest	Molecular Dynamics	module load gromacs	compute/mpi/gpu
Amber	24.0	Molecular Dynamics	/scratch/apps/amber/amber24	compute/gpu
NAMD	2.14	Molecular Dynamics	module load namd	compute/mpi/gpu
LAMMPS	Latest	Molecular Dynamics	module load lammps	compute/mpi
COMSOL	Latest	FEA/CFD	/scratch/apps/comsol	compute
Mathematica	Latest	Mathematics	module load mathematica	compute
Olex2	Latest	Crystallography	/scratch/apps/olex2	short/compute
Intel MKL/MPI	Latest	Math Libraries	module load intel	all
CUDA	11.4	GPU Framework	Loaded automatically	gpu

To request installation of new software contact the HPC administrator with the software name, version, license type, and justification. Allow 1-2 weeks for installation and testing.

10. Power Failure Procedure

⚠ Important — Power failures occur at this facility
The cluster is subject to unplanned power failures. Follow these steps to protect your running jobs and research data.

10.1 Before Running Long Jobs

- Always add %Chk to Gaussian input — this saves progress automatically
- For MD simulations — enable checkpoint output every 10,000 steps
- For jobs longer than 1 day — request walltime with 20% buffer

10.2 After Power Failure — Restart Procedure

1. Check if your job was running: check your email for SLURM notifications
2. Check job status: sacct -u $USER --starttime=yesterday

3. Check if checkpoint file exists in your job directory
4. For Gaussian: simply resubmit — script auto-restarts from .chk file
5. For GROMACS: resubmit with -cpi checkpoint.cpt flag
6. For Amber: resubmit with restart file
7. Contact HPC admin if you cannot restart your job

11. Getting Help

Email: ighani@ducc.du.ac.in

11.1 Useful Resources

- Job templates: /scratch/apps/templates/
- Gaussian manual: https://gaussian.com/man/
- GROMACS manual: https://manual.gromacs.org/
- Amber manual: https://ambermd.org/Manuals.php

Appendix A — Gaussian CPU Scaling Guide

This data is based on benchmarks run on duhpc and published Gaussian performance studies. It demonstrates why requesting fewer CPUs is often better:

CPUs	Relative Speed	Efficiency	Recommendation
1	1.0×	100%	Baseline
2	1.8×	90%	Good for small molecules
4	3.2×	80%	☑ Recommended for opt
8	5.6×	70%	☑ Recommended for SP/Freq
12	7.2×	60%	☑ Good for large molecules
16	8.0×	50%	Maximum — cluster limit
24	8.4×	35%	✗ Avoid — negligible gain
48	8.8×	18%	✗ Avoid — wasteful

Key insight: Going from 16 to 24 CPUs gives only 5% speedup but uses 50% more resources. This means your job waits longer in the queue AND other users are blocked. 16 CPUs is the sweet spot for Gaussian on duhpc.

Appendix B — Partition Quick Reference Card

Scenario	Use This Partition	CPUs	Memory	Time
Gaussian geometry opt	compute	4-8	20-32G	1-3 days

Scenario	Use This Partition	CPUs	Memory	Time
Gaussian single point	compute	8-12	32-48G	6-24 hours
Gaussian frequency	compute	8-12	32-48G	1-2 days
Gaussian large molecule	compute	12-16	48-60G	2-5 days
GROMACS single node	compute	12-24	40-80G	1-5 days
GROMACS multi-node	mpi	48-96	80-160G	1-5 days
Amber CPU	compute	8-16	20-40G	1-5 days
Amber CUDA (GPU)	gpu	8	20G	1-2 days
NAMD single node	compute	12-24	40-80G	1-5 days
NAMD multi-node	mpi	48-96	80-160G	1-5 days
Script testing	short	1-4	4-8G	< 4 hours
Olex2 refinement	short	2-4	8G	< 4 hours
NWChem calculation	compute	8-16	40-60G	1-5 days
COMSOL simulation	compute	8-16	40-80G	1-5 days

duhpc HPC Cluster | Department of Chemistry | University of Delhi
For help: ighani@ducc.du.ac.in | Templates: /scratch/apps/templates/
Version 1.0 | May 2026 | Next review: November 2026