NWChem HW1-Bonus Report

Hongyan Huang, Student ID: r14922156

September 30, 2025

1 Introduction

The purpose of this assignment is to compile, optimize, and execute the **NWChem** application on the Taiwania3 HPC cluster. The goal is to obtain correct results under the given input within 300 seconds, while exploring optimization techniques to improve runtime performance.

2 Build Procedure

The compilation was performed on the login node (`lgn301`) with the following steps:

1. **Environment setup**

```
source /work/b11902043/PP25/intel/oneapi/setvars.sh --force
export OMPI_HOME=/work/b11902043/PP25/openmpi
export PATH="$OMPI_HOME/bin:$PATH"
export LD_LIBRARY_PATH="$OMPI_HOME/lib:$LD_LIBRARY_PATH"
```

2. **Clone NWChem source code**

```
cd ~/hw1bonus
git clone https://github.com/nwchemgit/nwchem.git
export NWCHEM_TOP=$HOME/hw1bonus/nwchem
```

3. **Set build environment variables**

```
export NWCHEM_TARGET=LINUX64
export NWCHEM_MODULES=qm
export USE_MPI=y
export USE_MPIF=y
export USE_MPIF4=y
export ARMCI_NETWORK=MPI-PR
export USE_SCALAPACK=y
export USE_OPENMP=y
export FC=mpifort
```

```
export CC=mpicc
export CXX=mpicxx
```

4. **Build Global Arrays (GA) and ARMCI-MPI**

```
cd $NWCHEM_TOP/src/tools
./get-tools-github
MPICC=mpicc ./install-armci-mpi
```

5. **Configure and build NWChem**

```
cd $NWCHEM_TOP/src
make nwchem_config
make -j$(nproc) FFLAGS+=" -i8 -diag-disable=10448" \
                CFLAGS+=" -diag-disable=10441"
```

After compilation, the executable is located at:

```
$NWCHEM_TOP/bin/LINUX64/nwchem
```

3 Execution Setup

The job was submitted through a Slurm batch script `hw1-bonus.sh`:

```
#!/bin/bash
#SBATCH --job-name=nwchem
#SBATCH --partition=ctest
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=32
#SBATCH --cpus-per-task=1
#SBATCH --mem=64G
#SBATCH --time=0:5:00
#SBATCH --account=ACD114118
#SBATCH --output=nwchem.log
#SBATCH --error=nwchem.err

export NWCHEM_TOP=/home/r14922156/hw1bonus/nwchem
source /work/b11902043/PP25/intel/oneapi/setvars.sh --force
export OMPI_HOME=/work/b11902043/PP25/openmpi
export PATH="$OMPI_HOME/bin:$PATH"
export LD_LIBRARY_PATH="$OMPI_HOME/lib:$LD_LIBRARY_PATH"

export NWCHEM_TARGET=LINUX64
export NWCHEM_MODULES=qm
export FC=mpifort
```

```
export OMP_NUM_THREADS=1

mpirun -np $SLURM_NTASKS $NWCHEM_TOP/bin/LINUX64/nwchem \
/work/b11902043/PP25/hw1-bonus/inputs/w12_b3lyp_cc-pvtz_energy.nw
```

After job completion, the main output is found in `nwchem.log`, which ends with:

```
Total times  cpu:   46.9s   wall:   46.9s
```

4 Judge System

The correctness of the results and timing was verified using the TA-provided judge system:

```
bash /work/b11902043/PP25/setup.sh
source ~/.bashrc
hw1-bonus-judge
```

The job successfully produced valid output under 300 seconds, and results were uploaded to the scoreboard.

5 Optimization Attempts

5.1 Compiler Optimizations

Tried adding `-O3 -xHost -ipo` flags to the Intel compiler. Result: negligible improvement (< 1s).

5.2 Parallel Configurations

We tested multiple MPI/OMP configurations:

Configuration	MPI ranks/node	OMP threads	Total ranks	Wall time (s)	Notes
Baseline	32	1	128	**46.5**	Fastest
Hybrid 16×2	16	2	64	48.2	Slower
Hybrid 8×4	8	4	32	50.1	Slower
Pure OMP	1	32	4	>300 (fail)	Not scalable

5.3 Observations

- The input workload is relatively small.
- Hybrid configurations added thread synchronization overhead.
- The pure MPI configuration fully utilized all 128 cores and achieved the best performance.

6 Conclusion

- Successfully compiled and executed NWChem on Taiwania3 with the given input.
- Best performance was achieved with the **baseline pure MPI** configuration: 128 ranks (32 per node $\times$ 4 nodes), runtime $\sim$46.5 seconds.
- Hybrid parallelization and compiler flags did not yield improvements.
- For small-scale inputs, NWChem scales best with pure MPI on this cluster.

7 References

- NWChem GitHub
- NWChem Manual
- TA-provided environment and instructions for PP25 course