

CPLEX on HPC cluster:

What is CPLEX?

Linear programming was revolutionized when CPLEX software was developed over 20 years ago. CPLEX was the first commercial linear optimizer on the market to be written in the C programming language. CPLEX gave operations researchers unprecedented flexibility, reliability, and performance, allowing them to create novel optimization algorithms, models, and applications.

The full documentation for CPLEX is found in the following links:

[Official Documentation](#)

[GitHub Examples](#)

Versions Available:

- CPLEX-v12.8.0
- CPLEX -v20.1.0

How to load a version of CPLEX?

To see all the versions available in the cluster, use

```
module avail math/cplex
```

To load a version of CPLEX on the HPC, use the following command:

```
math/cplex/20.1.0
```

Verify by using this command:

```
module list
```

Note: Since Cplex supports MPI, user can load MPI module for heavy multi-node tasks.

Use the following command to achieve that,

```
module load mpi/openmpi/gcc/4.1.1
```

How to use CPLEX on the cluster?

There are multiple ways to use CPLEX in the cluster. The methods are listed below,

The Interactive GUI:

To launch CPLEX in GUI mode, use the following code,

```
#Jump to a compute node
srun -p main --qos main -n 1 -c 8 --mem 16G --pty bash

#Open the GUI
oplide
```

Note: User must login using X11 forwarding to the cluster. For eg,

`ssh -X userid@uahpc.ua.edu`

The Interactive terminal:

To launch CPLEX in terminal mode, use the following commands

```
#Jump to a compute node
srun -p main --qos main -n 1 -c 8 --mem 16G --pty bash

#Open the terminal
cplex

# Follow the command here
```

The Script :

To run a slurm job, the user must prepare input files. For this example, get input files with,

```
#Download the input test file from CPLEX website
wget http://miplib2017.zib.de/WebData/instances/ex10.mps.gz
gunzip ex10.mps.gz

# Make a sbatch script
touch cplex.sbatch
```

There are two templates for the script. First is threaded without MPI and another threaded + MPI. To run threaded with MPI, user must load the MPI module.

Threaded without MPI:

```
#!/bin/bash
#SBATCH -J Distrbuted_cplex
#SBATCH --nodes=2
#SBATCH --ntasks=14
#SBATCH --cpus-per-task=2
#SBATCH --time=0-01:00:00
#SBATCH -p main
#SBATCH --qos main
#SBATCH -e err.out
#SBATCH -o out.out

module purge
module load math/cplex/20.1.0
module load mpi/openmpi/gcc/

# Some variables
```

```

MPS_FILE=$1
RES_FILE=$2
CPLEX_COMMAND_SCRIPT="command_job${SLURM_JOBID}.lst"

# Create cplex command script
cat <<EOF >${CPLEX_COMMAND_SCRIPT}
set distmip config mpi
set threads ${SLURM_CPUS_PER_TASK}
read ${MPS_FILE}
mipopt
write "${RES_FILE}.sol"
quit
EOF
chmod +x ${CPLEX_COMMAND_SCRIPT}

# Start Cplex with MPI
# On first host, the master is running
mpirun -np 1 cplex -f ${CPLEX_COMMAND_SCRIPT} -mpi : -np
${(SLURM_NTASKS - 1)} cplex -mpi

rm ${CPLEX_COMMAND_SCRIPT}

module purge

```

To submit the script to the slurm scheduler, use the following syntax:

```

sbatch cplex.sbatch <mps file> <output_file>

```

For this example, the syntax would be

```

sbatch cplex.sbatch ex10.mps solution

```

Threaded + MPI:

```

#!/bin/bash
#SBATCH -J Distributed_cplex
#SBATCH --nodes=2
#SBATCH --ntasks=14
#SBATCH --cpus-per-task=2
#SBATCH --time=0-01:00:00
#SBATCH -p main
#SBATCH --qos main

```

```

#SBATCH -e err.out
#SBATCH -o out.out

module purge
module load math/cplex/20.1.0
module load mpi/openmpi/gcc/

# Some variables
MPS_FILE=$1
RES_FILE=$2
CPLEX_COMMAND_SCRIPT="command_job${SLURM_JOBID}.lst"

# Create cplex command script
cat <<EOF >${CPLEX_COMMAND_SCRIPT}
set threads ${SLURM_CPUS_PER_TASK}
read ${MPS_FILE}
mipopt
write "${RES_FILE}.sol"
quit
EOF
chmod +x ${CPLEX_COMMAND_SCRIPT}

# Start Cplex with MPI
# On first host, the master is running
mpiexec cplex -f ${CPLEX_COMMAND_SCRIPT}

rm ${CPLEX_COMMAND_SCRIPT}

module purge

```

Use the same format as above to submit to the slurm scheduler.

All the output files will be in the same directory as the script.

Where to find help?

If you are stuck on some part or need help at any point, please contact OIT at the following address.

<https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp>