CCR Advanced Seminar: CPLEX and Accessing a Faculty Partition

Cynthia Cornelius

Center for Computational Research
University at Buffalo, SUNY

cdc at buffalo.edu

December 2015

CCR Cluster - CPLEX

- This seminar assumes that you have basic knowledge of submitting and running jobs on the CCR cluster.
- Please refer to one of the introductory presentation seminars to learn the basics of running a computation on the CCR cluster.
  - The “CCR MATLAB Example Seminar” is a good starting point.

CCR Cluster - CPLEX

- You will not find CPLEX listed by the **module avail** command.
- CPLEX is not generally available to all users on the CCR cluster.
- CPLEX is licensed to the research group or individual.
- It can be installed in either the group's projects space or the user's home directory.

CCR Cluster - CPLEX

- You will need an installation of CPLEX, a module file, and a SLURM script.
- Check your projects directory for a CPLEX installation.
  - A member of your research group may have already installed the software and created the module file.

CCR Cluster - CPLEX

- Installing CPLEX on the CCR cluster is easy.
- Register with IBM CPLEX website.
  - It is free for academic research and educational use.
  - You must apply and be approved.
- Download the Linux 64-bit version.
  - This will be an installer script.
- Transfer this script to CCR.

CCR Cluster - CPLEX

- Create the software directory in the projects space.
  - Usually, this will be /projects/group/software.
- If the software directory already exists then check that you have write permissions for the directory.
  - The owner of the directory can change the permissions using the chmod command.

CCR Cluster - CPLEX

- Create the cplex directory.
- Copy the installation script to your cplex directory.
- Install CPLEX.
- Here are the instructions.
  - (http://www-01.ibm.com/support/docview.wss?uid=swg21444285)

CCR Cluster - CPLEX

- Next create a module file.
- Usually, the modules for a group will reside in the /projects/group/software/modules directory.
- Create the modules directory if necessary.
- Copy the cplex.lua file from the /gpfs/scratch/cdc/cplex directory to your modules directory.

CCR Cluster - CPLEX

- Edit the module file.
  - The paths should point to directories in your cplex installation.
- Using the cplex module file:
  - **TYPE:** module use /projects/group/software/modules
  - **TYPE:** module load cplex
- Now you can use CPLEX.

SLURM Script Example

- Copy the SLURM scripts from the /gpfs/scratch/cdc/cplex directory.
  - The slurmCPLEX-ubhpc file is an example script to submit a job to the CCR cluster.
  - The slurmCPLEX-josewalt file is an example script to submit a job to the faculty partition josewalt.

SLURM Script Example

- The **#SBATCH** lines are directives to the scheduler. The directives are the resource requests, job output file, and email preferences.

```
#!/bin/sh
#SBATCH --partition=general-compute
#SBATCH --time=01:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --job-name="cplex_test"
#SBATCH --output=test-cplex.out
##SBATCH --mail-user=username@buffalo.edu
##SBATCH --mail-type=END
```

SLURM Script Example

- The following lines print information to the job output file.

```
echo "SLURM_JOB_ID="$SLURM_JOB_ID
echo "SLURM_JOB_NODELIST"=$SLURM_JOB_NODELIST
echo "SLURMTMPDIR="$SLURMTMPDIR

echo "working directory = "$SLURM_SUBMIT_DIR
```

- The job starts in the directory from which you submitted it. This is the working directory.

SLURM Script Example

- Load the module file for CPLEX.
- List the loaded modules.

```
ulimit -s unlimited

module use /projects/group/software/modules
module load cplex
module list
```

- The ulimit command removes the size limit on the instruction stack. This helps large programs to run.

SLURM Script Example

- Run the computation using the command line.

```
echo "running computation now"
# insert command line for CPLEX computation
date
#
echo "All Done!"
```

- Once all the lines of the SLURM have been executed, then the job completes and exits the compute node.

The josewalt Partition

- The josewalt partition resides in the mae cluster.
- The reason faculty partitions reside in a separate cluster rather the CCR ub-hpc cluster is to improve the scheduling of jobs.
  - The CCR ub-hpc cluster has so many jobs in the queue that the scheduler can fail to launch jobs in the faculty partition simply because the job is too far down in the list.
- The josewalt partition has 12 compute nodes.
  - Each node has 12 CPUs (cores) and 128 GB of memory.
- The time limit is 500 hours.

Accessing the josewalt Partition

- The josewalt partition resides in the mae cluster. This is not the default.
- All SLURM commands reference the CCR ub-hpc cluster by default.
- You have two choices for accessing the josewalt partition.
- 1. Always specify the cluster when issuing a SLURM command.

```
-M mae
or
--clusters=mae
```

Accessing the josewalt Partition

- 2. Make the mae cluster your default by setting the SLURM_CONF variable.

```
export SLURM_CONF=/gpfs/slurm/conf/mae/slurm.conf
```

  - The advantage is that you do not have to specify the cluster for every SLURM command.
  - The disadvantage is that this makes using the CCR cluster more difficult. Either you must specify the ub-hpc cluster or unset the variable.

SLURM Script Example

- Changes to SLURM script to submit the job to the josewalt partition.
  - Add the account
  - Add the cluster
  - Change partition to josewalt

```
#SBATCH --account=pi-group
#SBATCH --clusters=mae
#SBATCH --partition=josewalt
```

  - You still use **sbatch slurm-script** to submit the job.

More on the josewalt Partition

- Summary of SLURM commands:

```
squeue -M mae -p josewalt

sinfo -M mae -p josewalt

snodes all mae/josewalt

slurmjobvis jobid mae
```