

# Hyper-threading on Cheyenne

Hyper-threading is enabled on the Cheyenne system's 36-core batch nodes. While CISL recommends running no more than 36 tasks or threads per node in most cases, users can experiment with running up to 72 per node to see how it affects their code's performance.

Below are examples of job configurations that use more than 36 tasks and/or threads per node.

When your script is ready, submit your batch job for scheduling as shown in the [Submitting jobs](#) documentation.

## Page contents

- [Example 1 - MPI job](#)
  - [Example 2 - OpenMP job](#)
- 

## Example 1 - MPI job

There are 72 tasks in your job. For hyper-threading, specify that you want to run them on one node as shown in the select statement. Also be sure to customize the script with your own project code, executable name, and so on.

**A tcsh user's PBS script for submitting the job would look like this:**

```
#!/bin/tcsh
#PBS -A project_code
#PBS -N hyperthreaded_mpi
#PBS -j oe
#PBS -k eod
#PBS -q queue_name
#PBS -l walltime=01:00:00

### Use two processes for each of the 36 CPUs on one node
#PBS -l select=1:ncpus=72:mpiprocs=72

setenv TMPDIR /glade/scratch/$USER/temp
mkdir -p $TMPDIR

### Run MPI program with HPE MPT
mpiexec_mpt ./executable_name
```

**A bash user's PBS script for submitting the job would look like this:**

```
#!/bin/bash
#PBS -A project_code
#PBS -N hyperthreaded_mpi
#PBS -j oe
#PBS -k eod
#PBS -q queue_name
#PBS -l walltime=01:00:00

### Use two processes for each of the 36 CPUs on one node
#PBS -l select=1:ncpus=72:mpiprocs=72

export TMPDIR=/glade/scratch/$USER/temp
mkdir -p $TMPDIR

### Run MPI program with HPE MPT
mpiexec_mpt ./executable_name
```

---

## Example 2 - OpenMP job

There are 72 tasks in your OpenMP job. For hyper-threading, you can use two threads for each CPU on a single node with the select statement as shown in the script.

**A tcsh user's PBS script for submitting the job would look like this:**

```
#!/bin/tcsh
#PBS -A project_code
#PBS -N hyperthreaded_openmp
#PBS -j oe
#PBS -k eod
#PBS -q queue_name
#PBS -l walltime=01:00:00

### Use two threads for each of the 36 CPUs
#PBS -l select=1:ncpus=72:ompthreads=72

setenv TMPDIR /glade/scratch/$USER/temp
mkdir -p $TMPDIR

### Run program
./executable_name
```

**A bash user's PBS script for submitting the job would look like this:**

```
#!/bin/bash
#PBS -A project_code
#PBS -N hyperthreaded_openmp
#PBS -j oe
#PBS -k eod
#PBS -q queue_name
#PBS -l walltime=01:00:00

### Use two threads for each of the 36 CPUs
#PBS -l select=1:ncpus=72:ompthreads=72

export TMPDIR=/glade/scratch/$USER/temp
mkdir -p $TMPDIR

### Run program
./executable_name
```

---