



User Guide

[Cluster Partnership Program \(/support-and-services/cluster-partnership-program\)](/support-and-services/cluster-partnership-program)

[Consultant Partnership Program \(/support-and-services/consultant-partnership-program\)](/support-and-services/consultant-partnership-program)

[New Faculty Program \(/support-and-services/new-faculty-program\)](/support-and-services/new-faculty-program)

[Workshops and Training \(/support-and-services/workshops-and-training\)](/support-and-services/workshops-and-training)

[Data Sharing Services \(/support-and-services/data-sharing-services\)](/support-and-services/data-sharing-services)

[Data Management \(/support-and-services/data-management\)](/support-and-services/data-management)

[Consulting and Technical Support \(/support-and-services/consulting-and-technical-support\)](/support-and-services/consulting-and-technical-support)

[User Guide \(/docs\)](/docs)

[Connecting to RCC Resources \(/connecting/index.html\)](/connecting/index.html)

[Using Midway \(/using-midway/index.html\)](/using-midway/index.html)

[Data Storage \(/data-storage/index.html\)](/data-storage/index.html)

[Software \(/software/index.html\)](/software/index.html)

[Running jobs on midway \(/index.html\)](/index.html)

[Login nodes vs. compute nodes \(/index.html#login-nodes-vs-compute-nodes\)](/index.html#login-nodes-vs-compute-nodes)

Hybrid MPI/OpenMP jobs

MPI and OpenMP can be used at the same time to create a Hybrid MPI/OpenMP program.

Let's look at an example Hybrid MPI/OpenMP hello world program and explain the steps needed to compile and submit it to the queue. An example hybrid MPI hello world program: **hellohybrid.c** ([./../_downloads/hellohybrid.c](#))

```
#include <stdio.h>
#include <omp.h>
#include "mpi.h"

int main(int argc, char *argv[]) {
    int numprocs, rank, namelen;
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int iam = 0, np = 1;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Get_processor_name(processor_name, &namelen);

    #pragma omp parallel default(shared) private(iam, np)
    {
        np = omp_get_num_threads();
        iam = omp_get_thread_num();
        printf("Hello from thread %d out of %d from process %d out of %d on %s\n",
            iam, np, rank, numprocs, processor_name);
    }

    MPI_Finalize();
}
```

To run the program on the RCC cluster, copy **hellohybrid.c** and **hellohybrid.sbatch** to your home directory, then compile the code interactively by entering the following commands into a terminal on a Midway2 login node:

```
module load openmpi
mpicc -fopenmp hellohybrid.c -o hellohybrid
```

[Illustrative examples](#)

[\(../index.html#illustrative-examples\)](#)

[Data Transfer \(../data-transfer/index.html\)](#)

[Frequently Asked Questions \(../faq/index.html\)](#)

[Tutorials \(../tutorials/index.html\)](#)

[Get an Account \(../getting-started/request-account\)](#)

[User Guide \(../docs\)](#)

[Get Support \(../support-and-services/consulting-and-technical-support\)](#)

Here we load the default MPI compiler, but it should be possible to use any available MPI compiler to compile and run this example. Note that the option **-fopenmp** must be used here to compile the program because the code includes OpenMP directives (use **-openmp** for the Intel compiler and **-mp** for the PGI compiler).

hellohybrid.sbatch ([../_downloads/hellohybrid.sbatch](#)) is a submission script that can be used to submit a job to Midway2 to run the `hellohybrid` program.

```
#!/bin/bash

# A job submission script for running a hybrid MPI/OpenMP job on
# Midway2.

#SBATCH --job-name=hellohybrid
#SBATCH --output=hellohybrid.out
#SBATCH --ntasks=4
#SBATCH --cpus-per-task=8
#SBATCH --partition=broadwl
#SBATCH --constraint=edr

# Load the default OpenMPI module.
module load openmpi

# Set OMP_NUM_THREADS to the number of CPUs per task we asked for.
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

# Run the process with mpirun. Note that the -n option is not required
# in this case; mpirun will automatically determine how many processes
# to run from the Slurm settings.
mpirun ./hellohybrid
```

The options are similar to running an MPI job, with some differences:

- **--ntasks=4** specifies the number of MPI processes ("tasks").
- **--cpus-per-task=8** allocates 8 CPUs for each task.
- `export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK` sets the number of OpenMP threads to the number of requested cores (CPUs) for each task.

You can submit `hellohybrid.sbatch` using the following command from one of Midway2 login nodes:

```
sbatch hellohybrid.sbatch
```

Here is an example output of this program submitted to the `broadwl` partition on Midway2:

[illegible]

```
Hello from thread 1 out of 8 from process 3 out of 4 on midway2-0270.rcc.local  
1
```

Research Computing Center

5607 S. Drexel Ave.

Chicago IL 60637

2014 **The University of Chicago**

Need Help?

Email:

or **Request Support**

Walk-In Laboratory: **Regenstein Library**, suite 216

Call us at **(773) 795-2667** or consult the **User Guide**