#!/bin/bash -l

# Specific course queue and max wallclock time

#SBATCH -p course -t 5

# Defaults on Barkla (but set to be safe)

## Specify the current working directory as the location for executables/files

#SBATCH -D ./

## Export the current environment to the compute node

#SBATCH --export=ALL

# load module for intel compiler

module load compilers/intel/2019u5

if [[ $# -ne 1 ]]; then

echo

echo ERROR

echo Usage: sbatch -c NUM\_THREADS\_TO\_USE $0 NAME\_OF\_INPUT.c

exit -1

fi

# SLURM terms

## nodes relates to number of nodes

## ntasks-per-node relates to MPI processes per node

## cpus-per-task relates to OpenMP threads (per MPI process)

# determine number of cores requested (NB this is single node implementation)

## further options available via examples: /opt/apps/Slurm\_Examples/sbatch\*sh

echo "Node list : $SLURM\_JOB\_NODELIST"

echo "Number of nodes allocated : $SLURM\_JOB\_NUM\_NODES or $SLURM\_NNODES"

echo "Number of threads or processes : $SLURM\_NTASKS"

echo "Number of processes per node : $SLURM\_TASKS\_PER\_NODE"

echo "Requested tasks per node : $SLURM\_NTASKS\_PER\_NODE"

echo "Requested CPUs per task : $SLURM\_CPUS\_PER\_TASK"

echo "Scheduling priority : $SLURM\_PRIO\_PROCESS"

# check expected inputs (OpenMP is only supported on a single node)

if [ "$SLURM\_NNODES" -gt "1" ]; then

echo more than 1 node not allowed

exit

fi

# parallel using OpenMP

SRC=$1

EXE=${SRC%%.c}.exe

rm -f ${EXE}

echo compiling $SRC to $EXE

icc -qopenmp -O0 $SRC -o $EXE

if test -x $EXE; then

# set number of threads

export OMP\_NUM\_THREADS=${SLURM\_CPUS\_PER\_TASK:-1} # if '-c' not used then default to 1

echo using ${OMP\_NUM\_THREADS} OpenMP threads

# run multiple times

./${EXE};echo '---'; echo

./${EXE};echo '---'; echo

./${EXE};echo '---'

else

echo $SRC did not built to $EXE

fi