INTEL MIC (Xeon Phi) EXERCISES

GETTING STARTED
Before class begins or at the beginning of the hands on portion of the class, please do the following:

MAC OS X
1. Locate the slip of paper with your username (e.g., hpc02) and password
2. NLI provides a Shared Folder on your desktop. Copy the MIC folder from the Shared Folder onto your desktop.
3. Open a terminal:
   a. Click the magnifying glass icon in the top right corner of the screen
   b. Type “Terminal” and select it when it appears in the list of results
4. Log into BlueRidge with the username that you identified in step 1:
   `ssh hpc02@blueridge2.arc.vt.edu`
5. When prompted, enter the password provided to log in.
6. Clear your Home directory:
   `rm -rf *`
7. Copy the MIC files into your Home directory:
   `cp /archive/groups/training/mic/* .`
8. Request an interactive job on one MIC node by running the provided submission command:
   `./mic_interactive.sh`

WINDOWS
1. Locate the slip of paper with your username (e.g., hpc02) and password
2. NLI provides a Shared Folder on your desktop. Copy the MIC folder from the Shared Folder onto your desktop.
3. Open PuTTY from the Shared Folder.
4. Under “Host Name (or IP address)” type (replace “hpc02” with the username from step 1):
   `hpc02@blueridge2.arc.vt.edu`
5. Click “Open”
6. When prompted, enter the password provided to log in.
7. Clear your Home directory:
   `rm -rf *`
8. Copy the MIC files into your Home directory:
   `cp /archive/groups/training/mic/* .`
9. Request an interactive job on one MIC node by running the provided submission command:
   `./mic_interactive.sh`
**Native Jobs**

1. Compile the program:
   
   ```
   icc -mmic -openmp -O3 helloflops3.c -o helloflops3
   ```

2. Log into the MIC (either mic0 or mic1 will work):
   
   ```
   ssh mic0
   ```

3. Set the number of threads and thread affinity:
   
   ```
   export OMP_NUM_THREADS=120
   export KMP_AFFINITY=balanced
   ```

4. Run the program:
   
   ```
   ./helloflops3
   ```

5. Options to try:
   
   a. Add the -vec-report=3 flag to see vectorization information on compilation
   b. Compare performance for:
      i. Different numbers of threads
      ii. KMP_AFFINITY set to balanced, compact, and scatter
   c. Try compiling for and running on the host
   d. Review the helloflops3.c code - note the OpenMP parallelism and how the arrays are declared with `__attribute__((align(64)))` to ensure performance

6. Exit the MIC:
   
   ```
   exit
   ```

**Automatic Offload (with MKL)**

1. Reset your modules:
   
   ```
   module reset
   ```

2. Compile the program:
   
   ```
   icc -std=c99 -O3 -mkl matmul_mkl.c -o mm_mkl
   ```

3. Load the MKL and MIC modules:
   
   ```
   module load mkl mic
   ```

4. Run the program:
   
   ```
   MKL_MIC_ENABLE=1 ./mm_mkl -s 8192
   ```

5. Options to try:
   
   a. Run entirely on the host by disabling offload:
      
      ```
      MKL_MIC_ENABLE=0 ./mm_mkl -s 8192
      ```
   b. Enable offload reporting to see what MKL offloaded: OFFLOAD_REPORT=2
   c. Run with smaller problem sizes to see where MKL does and does not offload
**MANUAL OFFLOAD**

1. Make sure that you have the Intel module and an MPI stack module loaded. For example:
   ```
   module purge
   module load intel mvapich2
   ```

2. Compile the program:
   ```
   mpicc mpi_hello_offload.c -o mpihw.offload
   ```

3. Load the MKL and MIC modules:
   ```
   module load mkl mic
   ```

4. Set the number of threads to be used on the MIC:
   ```
   export MIC_OMP_NUM_THREADS=16
   ```

5. Execute the program with two MPI processes:
   ```
   mpiexec -np 2 -ppn 2 ./mpihw.offload
   ```

6. Other options:
   a. Vary the number of threads on the MIC
   b. Exit your current interactive job and try requesting a new one with two nodes. Then try running with `-np 4`, offloading to four MICs across two nodes.
   c. Try running on two nodes as part of a batch job:
      [http://www.arc.vt.edu/resources/hpc/blueridge_mic.php#mult_mic_offload_batch](http://www.arc.vt.edu/resources/hpc/blueridge_mic.php#mult_mic_offload_batch)

**OFFLOAD TO MULTIPLE MICs**

1. Reset your modules:
   ```
   module reset
   ```

2. Compile the program:
   ```
   icc -openmp -O3 omp_hello_offload.c -o omphw.offload
   ```

3. Load the MKL and MIC modules:
   ```
   module load mkl mic
   ```

4. Set the number of threads to be used on the MIC:
   ```
   export MIC_OMP_NUM_THREADS=16
   ```

5. Run the program:
   ```
   ./omphw.offload
   ```

6. Other options:
   a. Try changing MIC_OMP_NUM_THREADS and check that it changes the output. If you clear MIC_OMP_NUM_THREADS and then try to run, what happens?
   b. Try disabling offload during compilation using the `-no-offload` flag. How many threads does it use now?