MKL-lab2 : FFT with pragma offload and Native Execution

In this example we learn how to offload MKL function calls using offload pragmas.

1. Objectives and learning goals

- To control memory allocation on MIC
- Understand data transferring and data persistence
- Set up the build environment
- Compile the program on the host without modifying the original code
- `icc --no-offload --mkl mkl_fft.c --o mkl_fft`
- Run the program: `.mkl_fft`
- Code to offload, replace the LRZ WORK FOR YOU comments with MKL calls
- Compile the program for offload: `icc --mkl mkl_fft.c --o mkl_fft`
- Run the program: `.mkl_fft`
- Check the performance results
- Set the following environment:
  
  ```sh
  export MIC_LD_LIBRARY_PATH=
  /lrz/sys/intel/compiler140_144/composer_xe_2013_sp1.2.144/compiler/lib/mic:
  /lrz/sys/intel/compiler140_144/composer_xe_2013_sp1.2.144/mpirt/lib/mic:
  /lrz/sys/intel/compiler140_144/composer_xe_2013_sp1.2.144/mkl/lib/mic:
  /lrz/sys/intel/compiler140_144/composer_xe_2013_sp1.2.144/tbb/lib/mic
  ```

- What about the performance:
  
  Compile the program for Native execution:
  
  ```sh
  icc --mmic --mkl_fft.c --o mkl_fft.mic
  ```
  
  add this setting:
  
  ```sh
  export KMP_AFFINITY=explicit,granularity=fine,proclist=[1-240:1]
  ```
  
  and run again.

Try now to understand the performance numbers.