Homework 2: Linear Algebra on MTE

1. Follow instructions on the MTE simulation slide set (posted as part of the 5/12/2016 lecture): download and install Oracle Virtual Box, install MTE (virtual machine on Windows), and get the examples to execute.

2. Compute the linear part of the first layer of the neural network of HW1. Use the $1000 \times 784$ first layer weight matrix from HW1, and one of the $784 \times 1$ input vectors. Following the examples included in the slides and the sample projects included in the VM (virtual machine), perform matrix-vector multiplication using task instances that perform one dot product each. Insert performance counters as needed. Insert also energy counters.

3. Evaluate performance for 1,2,4,8,16,32,64,128,256,512 and 1024 cores, following examples in the slides. Evaluate energy measurement and develop a methodology to present it.

Clearly, this HW exploits work you have done in HW1 and leads towards implementing the entire NN of HW1 on MTE in HW3.

4. Submit by 19 December 2016, using email to ran@ee with subject line “048874-F2016-HW2”:
   a. Your code
   b. Performance report

Q&A published after posting this HW:

Q&A: The parameter for changing number of cores is “-cores=13” (in case you wish p=13).

Q&A: How do we monitor and collect info about energy? Well, this is what I want you to think about and invent. Having seen the innovative ways most people presented performance in HW1, I am sure you will come up with similarly innovative ways to handle energy measurements.

Q&A: template not accessible? Ignore, copy one of the given projects into a new one, or make a fresh new project and import to it files from an existing project.

Comment: In some of the reports I have received, the speedup-vs-number of cores chart is shown using axes other than log-log. You should use only the log-log scale and a scatter/line chart, because it is the only one that shows ideal speedup as a straight line and it is easy to compare actual speedup to the ideal one (and detect suspicious results).

Comment: In some other works, automatic time reporting of the simulator was used instead of creating your own performance and energy counters. This is unreliable, please use your own counters and make them proportional to actual work (as we have discussed, accurate coefficients are hard to get and are not needed for relative analysis).

Comment: In yet other cases, I have seen counted input and output, such as reading the data from files and printing results. These times are irrelevant because we are not really simulating a full real system with I/O and the I/O time for any target manycore is unknown. Ignore these times, start counting after all the data resides in shared memory. In contrast, counting load and store from shared memory to the core is useful.