Parallel Computing Architecture 048874 (aka Manycores for Machine Learning)

December 2017

Homework 2: MNIST inference with 2-layer 1000 hidden units NN as a PLURAL program

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 1000×784 first layer weight matrix from HW1, and one of the 784×1 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 (T(p), SUP(p) and Eff(p)) for 1,2,4,8,16,32,64,128,256,512 and 1024 cores, following examples in the slides. Make sure to use your own run-time (cycle) counter for your own imaginary machine, rather than using the automatic time reporting of the simulator. Make sure to use log-log scale and a scatter/line chart for speedup (and maybe other measures), because it shows ideal speedup as a straight line and makes it easy to compare actual speedup to the ideal one (and detect suspicious results).
4. The course web page links to two alternative scripts for automating this process, contributed by Firas Shama and Roye Mazor. You are welcome to try them out and provide feedback.
5. In these performance evaluations, ignore time for I/O and file read/write. 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 the entire dataset resides in shared memory. Note that at this stage we assume unlimited size of shared memory. If we were doing a more detailed study, we would instrument the simulator to work with a limited shared memory (but we don’t do it in HW2).
6. In contrast to I/O times, counting cycles for load and store from shared memory to the core is useful.
7. Evaluate energy. Develop a methodology to measure and present it.
8. Port the rest of your inference machine (created in HW1) onto the PLURAL architecture as simulated by MTE.
9. Evaluate and report performance and energy for the entire inference machine for the 11 cases.
10. Submit by 31 December 2017, using email to ran@ee with subject line “048874-F2016-HW2”:
	1. Your code
	2. Performance report

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.