Embedded System Design and Modeling

ECE382N.23, Fall 2024

Homework #1

Application Models

Assigned: September 5, 2024
Due: September 18, 2024

Instructions:

- Please submit your assignment via Canvas. Submissions should include a single PDF with the writeup and a single Zip or Tar archive for any supplementary files (e.g., source files, which has to be compilable by simply running 'make' and should include a README with instructions for running each model).
- You may discuss the problems with your classmates but make sure to submit your own independent and individual solutions.
- Some questions might not have a clearly correct or wrong answer. In general, grading is based on your arguments and reasoning for arriving at a solution.

Problem 1.1: Process Networks

(a) In class we discussed process network models such as Communicating Sequential Processors (CSP) that restrict all communication to queues with zero size, i.e. rendezvous- or barrier-style communication where both sender and receiver must meet to exchange data and be able to continue. Are such models deterministic? Why or why not?

(b) Are KPN models that are restricted to queues with arbitrary but fixed and pre-determined size (greater than zero) in general deterministic or not? Why or why not?

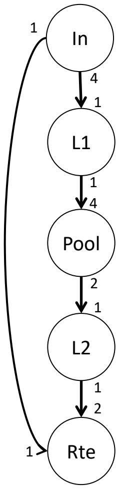
Problem 1.2: Dataflow

Given the following SDF graph of a 2-layer neural network in which the first and second layers are partitioned into 4 and 2 tiles, respectively.

Show down the balance equations and repetition vector for the SDF, and convert the application into a corresponding task graph, i.e. HSDF model. Write down a periodic single-processor schedule that minimizes buffer sizes. How much memory (in number of tokens) does your implementation require?

Assuming the task execution times as given in the table, what is the input-output latency for one iteration of the graph? What is the throughput rate at which your implementation can process inputs?

In	1
L1	5
Pool	1
L2	10
Rte	1



Problem 1.3: Performance Estimation

Assume that you are given a code like the one below, which is part of a C/C++ based machine learning inference model. You are asked to estimate the amount of time it takes to run on a CPU. The value of *L* is not known at design time

```
void inference(net* network){

  L = network->num_layers;

  for (int i=1; i<L; ++i){

    feat = network->layers[i-1]->out;

    layer_params = network->layers[i]->params;

    forward(layer_params, feat);

 }
}
```

(a) Can you derive an expression to model the execution time of this function? How does the answer change depending on whether *forward*() executes in constant time or not?

(b) How could machine learning strategies be applied to predict and hence estimate the execution time of this function?

Problem 1.4: Reading Assignment

Given the following paper (also linked from the class webpage):

L. Li, T. Flynn, A. Hoisie, "Learning Generalizable Program and Architecture Representations for Performance Modeling," *Supercomputing Conference (SC)*, November 2024.

Read the paper and submit a written review. We will be mimicking a typical conference peer review and paper discussion process, so read the paper critically and provide your assessment of the work including comparing and contrasting to other existing work, challenging assumptions, and identifying aspects that are unclear, need clarification or should have been explored in more detail. Following a typical conference review form, specifically structure your review to address these items:

1. Summarize the main points (problem being addressed, main idea, key results) of the paper. You can include figures and graphs to present a so-called visual abstract of the paper.
2. List 5 strengths.
3. List 5 weaknesses.
4. Detailed comments justifying your answers and assessment.