CS348K: Visual Computing Systems
Class/Reading Response Template

Reminder: please make sure the PDF you submit to Gradescope DOES NOT have your name on it. We will concatenate all responses and give everyone in the class a PDF of all responses.

Part 1: Top N takeaways from discussions in the last class. Note: this part of the response is unrelated to the current reading, but should pertain to the discussion of the prior reading in class (or just discussion in the class in general, if there was no reading):

- What was the most surprising/interesting thing you learned?
    - I believe that the most interesting aspect of Halide is that it allows programmers to have complete control over the optimization process. Initially, I thought that this feature was not important, as Halide merely separates algorithm and execution. Without acknowledging programmers, Halide should still be able to perform any optimizations. By granting programmers full control over optimization, Halide enables them to make their own decisions and improve the performance with human knowledge.
- Is there anything you feel passionate about (agreed with, disagreed with?) that you want to react to?
    - Relying solely on ML models to complete a complicated compiler optimization task is irresponsible. I still believe it to be true three years later after the paper was originally proposed. While ML models may perform well in general, it is hard to ensure their robustness, and they may produce undesirable outcomes. This unpredictability is rooted in ML models themselves, as they rely on complex algorithms and large datasets to make predictions or decisions.
- Did class cause you to do any additional reading on your own? If so, what did you learn?
    - I tried to walk through the TVM code, and I found it to be quite complex. I spent most of my time on tvm::te::Tensor because I was thinking that adding some fancy tensors that are hardware-specific would benefit performance. A Tensor contains data members such as op, value_index, shape, dtype, and others. The member op has a member function InputTensors, the input and output of value_index are both of type Tensor. Besides, an op may have multiple outputs, and the value_index in indicates which output of the op this tensor belongs to. The axis in BaseComputeOpNode represents a loop variable used in a computation, and reduce_axis records which dimensions are being reduced in the computation.
- Major takeaways in general?
    - The most important principle of Halide: separating the algorithm from the execution and acknowledging programmers the optimization.
    - ML can be a valuable component in Halide's automatic optimization process, but it cannot take full control of the optimization process.

Part 2: Answers/reactions to instructor's specific prompts for this reading. (Please see course website for prompts).

Part 3: [Optional] Questions I'd like to have specifically addressed via in class. (We also encourage you to just post these questions on Ed immediately so anyone can answer!)