Theory Homework

1. What are the key architectural features that make these systems suitable for AI workloads?
    a. The huge increase of memory on chip where memory is distributed across the cores allows for faster processing as data does not need to be extracted from a drive to be used.
    b. The overwhelming increase in cores allows for more computations to run in parallel
    c. The transformation of Von Neuman to Data Flow/Spatial Architectures allows these systems to be designed in a domain-specific way to optimize for the types of computations needed to training models.

2. Identify the primary differences between these AI accelerator systems in terms of their architecture and programming models.

The architectural difference in the AI accelerator systems can best be summed up by how data flows through the chips. Instead of RAM and CPU/GPU the memory is distributed across the processors. The compiler maps the processes to a graph structure flowing data across the processors. One big advantage of this is that data does not need to be saved or extracted from memory before being operated on. Programmatically it differs in that instructions are not executed sequentially, but triggered by the arrival of data packets.

When data flows through the graph-like structure of the chip it activates different processes flowing data across the chip and connecting to other sections through networks. This can speed things up as data is not saved or extracted from memory. However, there are still some memory stores and networks to transfer data between chips. Additionally, the compiler can map a program across multiple chips.

3. Based on hands-on sessions, describe a typical workflow for refactoring an AI model to run on one of ALCF's AI testbeds (e.g., SambaNova or Cerebras). What tools or software stacks are typically used in this process?

For the Cerebras CS2 machine, the big differences in processing are abstracted away from the user as a model can still be written in Python and trained using a variant of PyTorch. You can ssh into the machine just as you would with a super computer, create a virtual environment, and run your python code that is adapted to the specific version of PyTorch. The PyTorch code changes are in how the data is loaded and the model definition while much of the codebase can stay the same by using the wrapper scripts provided. Distributed or parallel training is automatically configured.

4. Give an example of a project that would benefit from AI accelerators and why?

One project that could have benefits from AI accelerators is a foundational weather model based on convolutional operations. Sid's work showed a big computational speed up in these layers. Foundational models are very large and require lots of computational power, the exact thing a domain specific system is designed to handle. Another type or project that could see major benefits would be training a LLM as one AI-accelerator showed a 66x computational speed up in training a GPT-2 model.