ECE 498 ICC: IoT and Cognitive Computing

Spring 2020, Homework 3

Due on May 18, 2020

Question 1: IoT (25 pts)

1. Great job! You have stayed at home for almost a month to stop the spreading. Now take a look around the place you live in, please select at least one IoT-enabling device different from the **"smart lighting"** example given in the lecture (Hint: Smart body weight scale; Internet TV; Amazon Echo; any App-connectable device). Then describe its "Purpose", "Behavior" and at least two unique requirements that will be necessary to fulfill its functionality. (See slides 4-6 from lecture 18) **[4 pts]**

2. Designing an IoT at scale is more serious and thus require more cautions to complete. What will be the possible challenges if we want to deploy a larger scale IoT based on a relatively private one? (An example of reliability has been given; clues could be found through slides of lecture 18). Please list three more problems and their solutions **[6 pts]**

Possible Problem	Solution
Reliability	Verifying system behaviors through behavior diagrams.

3. Amazed by your work done in this course, Chicago Department of Transportation (CDOT) would like to hire you as the chief engineer for upgrading the current I-Pass Toll System to a Unified Vehicle Registration System so a real-time plate recognition could be deployed.

   (a) Soon, a staff from CDOT hold a virtual meeting with you regarding the requirement of this exciting new system:

"We would like to set up an online database that allow each household to set up an account which includes multiple users and vehicles. For each active user under the household, they must share the same home address but age, payment method and billing history separately. For each vehicle registered, we need the following information: license plate number, plate sticker valid through, garage address, manufacturer, vehicle type and number of axis. Each vehicle needs to be registered under exactly one valid user."

Please draw an E-R model diagram like the one in slide 16 from lecture 18. You should follow the format (with proper shapes and label) and include all entities in the following table. **[10 pts]**

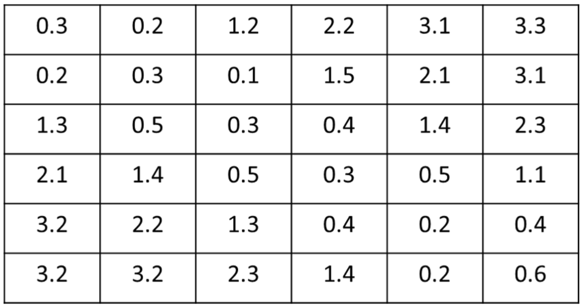
Account
AcctName
Password
HomeAddress
User
UserName
Age
BillingHistory
PaymentMethod
Vehicle
VehicleName
PlateNumber
ValidTill
GarageAddress
Manufacturer
Type
AxisNumber
USPS_Standardized_Address

(b) CDOT is very happy with your design on database and ask you to give an estimation of cost per month if both database and license plate (image) recognition are deployed on AWS with the given information in the following table. Justify your answer with proper reference. (AWS homepage: https://aws.amazon.com/) **[5 pts]**

Item	Count
System-Wide Average Daily Traffic	1,575,670
Estimated Number of Account by the end of 2020	2,706,000
Average number of users per account	3.2
Average number of vehicles registered per account	2.1
Data per user	64 kB
Data per vehicle registered	128 kB
Overhead data per account	16 kB

Question 2: Weight Quantization and Huffman Coding (20 pts)

After trying to implement a DNN on a tiny IoT device, you figure out that the on-chip memory of the targeted device is not sufficient for keeping DNN parameters (weights). So, you decide to apply weight quantization and huffman coding and to reduce the number of bits required to represent each weight.
Assuming you have the following weights (in a 6×6 matrix) and each of them is represented by FLOAT32 format. In total, these weights require $32 \times 6 \times 6 = 1152$bits.



0.3	0.2	1.2	2.2	3.1	3.3
0.2	0.3	0.1	1.5	2.1	3.1
1.3	0.5	0.3	0.4	1.4	2.3
2.1	1.4	0.5	0.3	0.5	1.1
3.2	2.2	1.3	0.4	0.2	0.4
3.2	3.2	2.3	1.4	0.2	0.6

1. By applying the weight quantization mentioned in lecture 16 (slide 8), you need to first cluster the weights into 4 groups according to their values:

$$
\begin{aligned}
Group_0 &: [0, 1) \\
Group_1 &: [1, 2) \\
Group_2 &: [2, 3) \\
Group_3 &: [3, 4)
\end{aligned}
$$

   Please present the matrix of cluster indices (shape: 6×6) and the vector of centroids (shape: 4×1) **[8 pts]**

2. After weight quantization, how many bits are required for keeping cluster indices and centroids, respectively (assuming the centroids still require FLOAT32 format, and each index (0, 1, 2, or 3) needs 2 bits to represent)? What is the compression ratio between the original design and the quantized one regarding the number of bit? . **[6 pts]**

3. To further compress the space for keeping the 6×6 index matrix, you start using Huffman code together with weight quantization. You need to first encode every element in the centroid vector, and then you fill the encoded data into the index matrix instead of using the 2-bit indices in Q2.2. Please present the variable-length code after Huffman coding for each centroid, and illustrate how this method helps reduce the memory overhead for keeping the index matrix. [Hint: each centroid may appear in the cluster index matrix different times.] **[6 pts]**

Question 3: POWER and CAPI (21 pts)

1. **POWER System and Architecture**

   (a) IBM has a series of high performance microprocessors called POWER, which have been used by servers and supercomputers in the industry. POWER is the acronym for "Performance Optimization With Enhanced RISC". Please describe what RISC is. What are the main characteristics of RISC ISAs? **[2 pts]**

   (b) As an enhanced version of RISC, Power ISA has added expanded features over the years of evolution. Please name TWO main characteristics of the Power ISA and describe what POWER systems are. **[3 pts]**

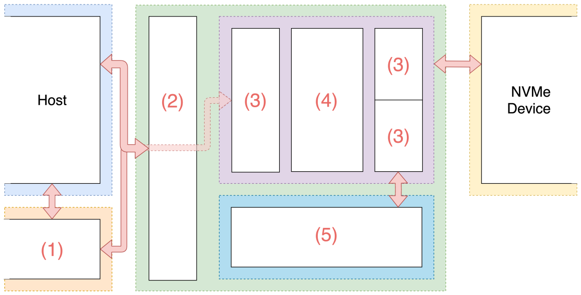
   (c) Briefly compare the following architectures: Intel X86, ARM, and Power ISA. You shall summarize at least four similarities or differences in total among these three architectures. **[4 pts]**

2. **CAPI SNAP**

   (a) Describe what CAPI is. What is the ONE significant problem that CAPI aims to solve? **[2 pts]**

(b) Describe what SNAP is. What is its relationship with CAPI? **[2 pts]**

(c) Imagine that you are asked to design a Near Memory Acceleration action with the CAPI SNAP framework. You attach a CAPI-enabled SSD to your host system, meaning that there is a FPGA in between the NVMe SSD and your host. Instead of fetching data directly from the SSD, you want to do some operations with SNAP in the FPGA. Look at this high-level block diagram and answer the following questions:



i. Choose among these options: PSL, AXI, Hardware Action, HOST memory, and Device Memory. Mark each of the (1) to (5). **[5 pts]**

ii. With the components mentioned above, on a high level, briefly describe the life cycle of an SNAP function from the host making the call to it retrieving the results. You can start from "the host issues an NVMe read request to the FPGA through...". **[3 pts]**

Question 4: Near Memory Computing (20 pts)

1. **System performance without NMA**

   Consider a simple system architecture that consists of a simple CPU with three levels of SRAM on-chip cache and DRAM main memory. For this question, we can calculate the total energy and time spent on memory accesses using the following equations:

$$\begin{aligned}\text{total memory access time} &= \text{mean latency} \times \text{total number of accesses} \\ &\quad + \text{compulsory cache misses latency} \\ \text{total memory access energy} &= \text{mean energy} \times \text{total number of accesses} \\ &\quad + \text{compulsory cache misses energy}\end{aligned}$$

   For the warm cache accesses, that is, the data is at least accessed once and therefore may already present in the cache levels, the mean memory access latency and energy of a level of the memory hierarchy can be computed using the following equations:

$$\begin{aligned}\text{mean memory access latency} &= (1 - \text{miss rate}) \times \text{access latency of current level} \\ &\quad + \text{miss rate} \times \text{mean access latency of next level} \\ \text{mean memory access energy} &= (1 - \text{miss rate}) \times \text{access energy of current level} \\ &\quad + \text{miss rate} \times \text{mean access energy of next level}\end{aligned}$$

Notice that, to get the mean values for the whole system, these relationships needs to be applied recursively.

In addition, we also should not ignore the effect of compulsory misses for caches. Compulsory misses happen when a piece of data is accessed for the first time and these cache misses are inevitable. Thus, we can assume that the data needs to be pulled from the DRAM through all the cache levels until it reaches L1 cache. That means, such accesses will cause cache miss in all levels. For simplicity, we assume that there will be around 10^6 such accesses.

Now, suppose that for a certain computational kernel, the miss rate, average value of latency, and energy consumption of accessing each level of the memory hierarchy are as follows:

Level	Miss rate	Average Latency	Average Energy
L1	10%	5 ns	15pJ
L2	5%	15 ns	26pJ
L3	2%	45 ns	47pJ
DRAM	NA	80 ns	2560pJ

It is worth mentioning that we are using an extremely simplified model in this question. For a real processor, these values will be affected by a large number of architecture specific parameters, such as cache line size, out-of-order execution, cache prefetching, etc. You are encouraged to consult computer architecture textbooks if interested.

Given the information above, assuming that the total number of accesses issued from the CPU is 10^9 for this kernel. Calculate the following values:

(a) The total time spent on memory access. **[4 pts]**

(b) The total energy spent on memory access. **[4 pts]**

(c) Now, suppose that on average each computational operation requires 2 reads and 1 write to the memory, and each of the operation consumes 18pJ. Calculate the percentage of total energy consumed by the computation. For simplicity, you may ignore the compulsory memory accesses when calculating the total number of computational operations. (Assuming that the energy is only consumed by memory accesses and computational operations, and the data type remains the same for all operations and memory accesses) **[4 pts]**

2. **System performance with NMA**
   Now, assuming that a near memory dataflow accelerator specialized for the aforementioned kernel is added to the system. Since the accelerator is directly connected to the DRAM, we assume that the access latency and energy consumption to the DRAM can be reduced by 50%. Also, assume that the number of direct accesses to the DRAM remains the same as in the previous case, considering both the regular and compulsory cache misses. Again, compute the total energy consumption AND latency of the kernel on memory access for the NMA. **[4 pts]**

3. **Advantages of using NMA**
   Comparing the result above for systems with and without NMA, discuss the reasons why NMA can improve the system performance in terms of energy and latency.**[4 pts]**

Question 5: Efficient DNN (14 pts)

1. **Standard Convolution**
   Standard convolution takes an $h_i \times w_i \times d_i$ (representing height, width, and depth) input tensor L_i, and applies convolutional kernel $K \in R^{k \times k \times d_i \times d_j}$ to produce an $h_i \times w_i \times d_j$ output tensor L_j with $stride = 1$. Please compute the number of operations (regarding both multiplications and additions). **[3 pts]**

2. **Depthwise Convolution**
   Depthwise convolution takes an $h_i \times w_i \times d_i$ input tensor L_i, and applies convolutional kernel $K \in R^{k \times k \times d_i}$ to produce an $h_i \times w_i \times d_i$ output tensor L_j with $stride = 1$. Please compute the number of operations (regarding both multiplications and additions). **[3 pts]**

3. **Pointwise Convolution**
   Pointwise convolution takes an $h_i \times w_i \times d_i$ input tensor L_i, and applies convolutional kernel $K \in R^{1\times1\times d_i \times d_j}$ to produce an $h_i \times w_i \times d_j$ output tensor L_j with $stride = 1$. Please compute the number of operations (regarding both multiplications and additions). **[3 pts]**

4. **Reduction in Computation**
   By combining the depthwise and pointwise convolutions, we can create a depthwise separable convolution and replace the standard one. Compute the number of operations in a depthwise convolution and a pointwise convolution (Q5.2 and Q5.3), and compare it to the operation number of a standard convolution (Q5.1). What is the ratio of operation reduction if using a depthwise separable convolution instead of a standard one? Please explain why it can save operations. **[5 pts]**