Undergraduate Computer Architecture, Fall 2024

Final Exam, 2024-12-17

If you agree with the following sentence, please sign your name below it.

I have not cheated nor have I received any help from other students in the exam.

Student ID & Nam

1. (30 pts) Memory Hierarchy

At the age of large language models and big data, memory hierarchy has become more and more important because computer systems often need to process the models and datasets. Please answer the following questions:

(a) (5%) Explain "memory hierarchy" in a typical computer system, as mentioned in Chapter 5.

(b) (5%) Suppose you are writing a program to multiply two large matrices A and B. Assume both A and B are too large to store in the main memory. Could you set up a virtual memory with enough swap space to perform the multiplication? Explain how to do it and what the resulted performance would be.

(c) (5%) Following Question (b), can you think of a way to optimize the performance for large matrix multiplication? Hint: We talked about how to optimize cache utilization for matrix multiplication in the class.

(d) (5%) Can multithreading be used to optimize the large matrix multiplication workload in Question (b)? If yes, please describe how it works. performance ?

(e) (5%) The three Cs (compulsory/capacity/conflict) model is often used for understanding the cause of cache misses. Please explain the three Cs model.

(f) (5%) Following Question (e), for applications which process large language models and big data, which kind of cache misses would occur most frequently? Why? How would you modify the hardware to reduce such cache misses?

2. (30 pts) Multiprocessor

Now that we have discussed memory hierarchy, let us investigate on multiprocessor. The most straightforward way to build a multiprocessor in theory is to add more processors to a computer system and implement a hardware mechanism for the processors to share the memory. Please answer the following questions:

(g) (5%) If we have a shared memory multiprocessor system, what would happen if multiple processors attempt to read and write the same variables within a program region? How would you make sure that the program run correctly? What kind of support from the instruction set architecture (ISA) is required?

(h) (5%) Following Question (~~e~~g), even if you can make the program run correctly, you may observe poor performance because the memory bandwidth is limited. How would you improve the memory bandwidth by adjusting the memory design?

(i) (5%) Cache is important to a shared memory multiprocessor because it can significantly reduce the amount of access to the memory. To make sure that the primary cache has the smallest hit time, each processor has its own (private) primary cache. Please explain how to resolve coherence issues when multiple processors attempt to read and write the same variables within a program region.

(j) (5%) In the following program, OpenMP is used to parallelize a matrix multiplication program.

```
void matrix_multiply(int **A, int **B, int **C, int N) {
    #pragma omp parallel for
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            C[i][j] = 0;
            for (int k = 0; k < N; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
}
```

0 ~ 15

Do not be afraid, because this simple example shows how easy it is to parallelize loops where there is no loop-carry dependency. (Loop-carried dependencies occur when the code in a loop iteration depends on the output of previous loop iterations). In this example, a special directive **#pragma omp parallel** is placed before the loop indexed by i to instruct the compiler to partition the iteration space {0, 1, 2, …N-1} to multiple processors. Suppose N=64 and we have 4 processors, please think of a way to partition the iteration space and describe how each processor read/write the variables. Do you see any interprocessor data dependency, i.e. the code run by one processor depends on the output of another processor)?

(k) (5%) When the matrices are very large, it is not only difficult, but also very costly, to build a multiprocessor computer with enough memory to perform the entire matrix multiplication in memory without swapping. Another option is performing the matrix multiplication on message-passing multiprocessor on a cluster. Please describe how message-passing works and how it would be more cost-effective than shared memory multiprocessor?

(l) (5%) For the code in Question (j), can it can be converted to run on message-passing multiprocessor? If yes, please describe how it works.

3. (20 pts) Edge AI and Roof-line Model

A group of researchers at Massachusetts Institute of Technology developed a LLM called TinyChat to run on weak, power-constrained edge devices. First, they profiled the execution time of TinyChat in the target application scenario to understand the workload. The LLM workload was broken down into two stages, Context stage and Generation Stage, and the profile revealed that the Generation Stage dominated the execution time, as Generation occupies 310ms of the total execution time (320ms) for a request. Furthermore, they found arithmetic intensity for the Context stage is la~~rger than~~ ~~165,~~ 170 but is only 1 for the Generation Stage. The roofline model for the edge device is shown below. Please answer the following questions:

10^{-3}

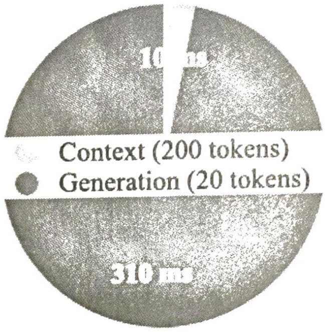
heavy memory accd.

few FLOPs

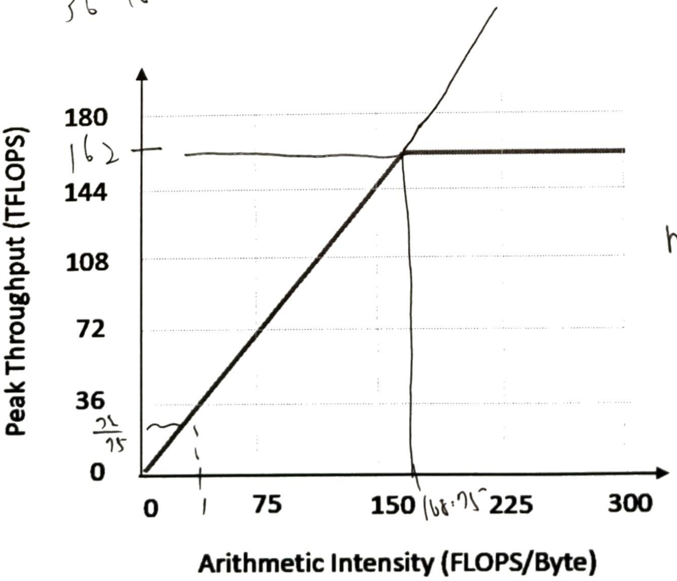
k M G T

10^3 10^6 10^9 10^{12}

μ 10^{-6}



36 18



$$y = \frac{72}{75}x = 162$$

$$x = \frac{162 \cdot 75}{72}$$

$$= 168.75$$

$$m = \frac{\cancel{72}\,24}{\cancel{75}\,25}$$

$$\frac{72}{75} \times 170$$

163.2

(m) (5%) Please use the roofline model to estimate the peak throughput of the Context Stage and the Generation Stage.

(n) (5%) According to the profile and the roofline model, is TinyChat mostly compute-bound or memory-bound? Why?

(o) (5%) Can you estimate the total number of floating-point operations executed for one request? How many bytes of memory access have been done?

(p) (5%) Quantization is a popular way to optimize neural networks and language models. Basically, it uses a lower precision floating-point format to perform the computation. Suppose the original TinyChat uses FP16, and we choose to use FP4, will it reduce the execution time? Can you estimate the execution time if FP4 is used?

4. (30 pts) More about Computer Organization

The following questions are to evaluate your knowledge about computer organization.

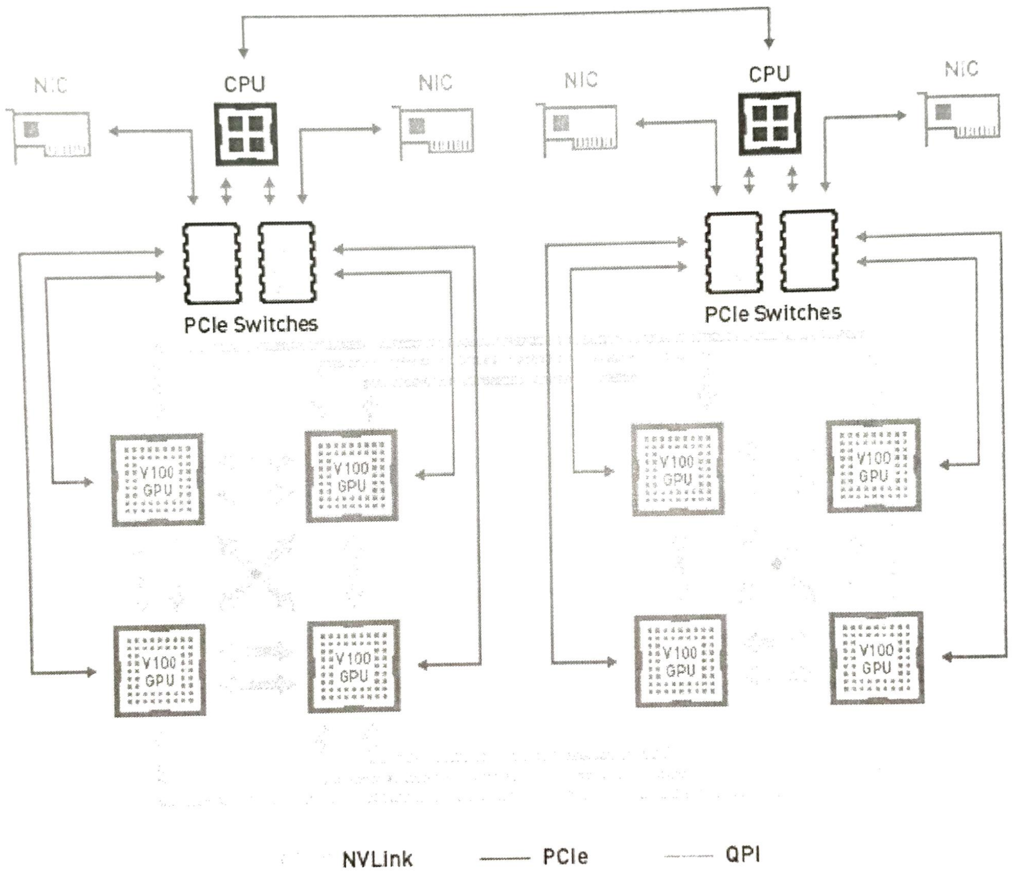
(q) (5%) To support virtual memory, a cache can be virtually indexed and virtually tagged, so that the cache can search for the data before the TLB translates the virtual address into physical address. However, this can lead to aliasing. Please explain how aliasing can happen and how to address this issue.

(r) (5%) Suppose our computer has a GPU, and we would like GPU to accelerate large matrix multiplication. Unfortunately, our GPU does not have sufficient memory to hold the matrices. We can overcome this memory issue by storing the matrices in the main memory and have the GPU perform matrix multiplication block by block. Can you describe how it works with a pseudo code?

(s) (5%) Following Question (r), do you think that GPU can perform the large matrix multiplication effectively? What would be the performance bottleneck? "Prefetching" is a technique that leverages predictable address patterns to speculatively bring in additional data items when a particular data item is accessed. One example of prefetching is a fast stream buffer that prefetches sequentially adjacent data items into a separate buffer when a particular data item is brought in. If the data are found in the prefetch buffer, it is considered as a "hit", because the processor can access the data without waiting, and the next data item is prefetched. Is prefetching useful for improving GPU performance in Question (r)? Can you describe how it works with a pseudo code?

(t) (5%) The following figure shows the system topology for NVIDIA DGX V100 system. It appears that the system has 2 CPU chips. The CPU chips are connected with Intel's QuickPath Interconnect to form a shared memory system. The network interface cards (NIC) and GPUs are connected to the CPUs via 4 PCIe switches. Interestingly, NVIDIA developed NVLink technology for the GPUs to share their GPU memories, as a way to expand GPU memory. This is why people buy high-end NVIDIA GPUs to run large language models.

Now, according to the figure, suppose each NVLINK provides 50GBytes/sec bandwidth, what is the bisection bandwidth for the NVLink-based GPU interconnection network? (Hint: In computer networking, a network may be bisected into two equal-sized partitions. The bisection bandwidth of a network topology is the minimum bandwidth available between any two such partitions).


NIC
CPU
NIC
NIC
CPU
NIC
PCIe Switches
PCIe Switches
V100 GPU
V100 GPU
V100 GPU
V100 GPU
V100 GPU
V100 GPU
V100 GPU
V100 GPU
NVLink
PCIe
QPI