CS553 Spring 2020 - HW2 report

Team 25

Ji Tin Justin Li A20423037
Hsuan-An, Weng Lin A20450355
Kevin Tchouate Mouofo, A20454613

Q1. Processors (15 points):

a. Today's commodity processors have 1 to 64 cores, with some more exotic processors boasting 72-cores, and specialized GPUs having 5000+ CUDA-cores. About how many cores/threads are expected to be in future commodity processors in the next five years?

According to a tenet of Moore's Law, the growth of microprocessors is exponential. Hence, the previous evolution of the number of cores in a microprocessor indicates that in 5 years, a commodity microprocessor will have more than 100 cores.

b. How are these future processors going to look or be designed differently than today's processors?

As our current processors are made of transistors of about 10nm, reducing the size of the transistors is becoming a challenge due to the manufacturing process. Currently, the tiniest transistor in a processor has 7nm, and the limit for silicon transistor is 3nm. To keep satisfying Moore's Law on future processors, we will have to find an alternative to the size of the transistors. A solution to that might be the "specialized processors". Our computer will not be made of one processor, but many "hybrid processors" that could handle specific tasks. For instance, Intel is planning to adopt a new design called Foveros design. In this 3-D hybrid CPU, the parts of the CPU will be manufactured separately and then assemble together one layer on top of another. The first 2 layers will be the DRAMs, followed by a chiplet with CPU and GPU (10nm), and a lower Die with cache and I/O.

c. What are the big challenges they need to overcome?

The big challenges they will need to overcome are:

· The scale of the chips. The current chips are made by the process of photolithography. With the reduction of the scale of transistors, we will have to see if this process is still sustainable.

· The power leakage due to the size of the transistor. A reduction of the scale of the transistor, increases the risk of power leakage.

· The heat problem. With smaller transistors, we could think of a higher gate voltage to deal with leakage, this might cause a heating effect that can compromise the integrity of the whole chip.

· The power consumption. The way the power will be handled in those processors.

. The speed of the processor, how will that be organized.

e. What type of workloads are hardware threads trying to improve performance for?

Hardware thread are trying to improve performance for the workloads that are IO intensive, and heavily multithreaded. In that case, the time needed to access the data of the workload is non negligible. As a solution to that, the hardware thread (pipeline) are used to hide IO latency.

f. Compare GPU and CPU chips in terms of their strength and weakness. In particular, discuss the tradeoffs between power efficiency, programmability and performance.

CPUs are made to handle many different tasks, while GPUs are made for specific tasks. Hence, a GPU is more performant than a CPU if we look at what the GPU was meant for. However, CPUs are more versatile than GPU because they have a larger instruction set. Generally, a CPU is less power consuming than GPU.

Q2. Threading (21 points):

a. Why is threading useful on a single-core processor?

Multithreading allows IO-intensive applications to have higher performance by hiding latency during IO operations. E.g. When a thread is waiting on IO, the scheduler can schedule another thread to run on the processor, effectively increasing performance.

b. Identify what a thread has of its own (not shared with other threads):

The execution environment maintains the thread identifier and the thread context for each thread. The thread context includes the thread's set of machine registers, the kernel stack, a thread environment block, and a user stack in the address space of the thread's process

c. Do more threads always mean better performance?

Threads have to be maintained properly (both by the programmer, the OS, as well as the threading library) in order to give better performance.

d. Is super-linear speedup possible? Explain why or why not.

Yes. Super-linear speedup is possible if the task given is embarrassingly parallelizable. As we split the data into smaller pieces, it is possible that the size of the data becomes small enough such that it fits into higher memory hierarchy component (e.g. from memory → L3 cache, or L3 cache → L2 cache etc), which often times the speeds up comes in chunks. Together with more cores and multithreading, the speedup is possible to be super-linear.

e. Why are locks needed in a multi-threaded program?

Locks are abstractions provided by the operating systems to coordinate shared resources across threads. They are used to avoid race-conditions in programs. For instance, if two threads increment the same variable, at the end it’s not guaranteed that we will have the expected value. To avoid that, we have to use locks to control the access to the variable.

f. Would it make sense to limit the number of threads in a server process?

Yes. More threads incurs higher coordination overheads. The cost of locks (for preventing race-conditions and thread starvation) increases exponentially with the increase of threads, which may hinder overall performance.

g. What is the advantage of OpenMP over PThreads?

OpenMP provides an API that supports multi-platform, parallel programming across many machines, regardless of the machines’ underlying instruction sets and operating systems. PThreads, on the other hand, is a multithreading library for the programming language C, which only provides communication and synchronization mechanisms between threads in a single process.

Q3. Network (11 points):

a. A user is in front of a browser and types in www.iit.edu, and hits the enter key. Think of all the protocols that are used in retrieving and rendering the main webpage from IIT. Describe the entire sequence of operations, commands, and protocols that are utilized to enable the above operation.

When the user enters www.iit.edu in the address bar:

1. The web browser segments the URL to find the protocol, the host, the path and the port.

2. The web browser, via a DNS Lookup command on a DNS server, gets the IP number (IPv4 or IPv6) of the host. Then, a socket is open from the user's computer to the IP number, on the specified port (80 generally).

3. The web browser makes a TCP connection and sends an HTTP GET request to the host. The host forwards the request to the server of IIT. Then, the server inspects the request and launches the server plugin needed to handle the request. After accessing the full request, the plugin gets a HTTP response ready.

4. The response is made out of the elements on the IIT database (assuming it is not a static web page), together with other information added by the plugin. We got a HTML text. A HTTP response is then constructed and sent back to the web browser by the plugin.

5. When the web browser receives the response, a DOM tree is built up out of the HTML. New requests are made for every resource in the HTML to assemble up the IIT web page (images, JavaScript files, Stylesheets, etc). Every rendering information on the stylesheet is attached to the right node in the DOM tree, DOM nodes are moved, and information style is updated accordingly, JavaScript is executed, and the web browser renders the page.

Source mostly used: https://friendlybit.com/css/rendering-a-web-page-step-by-step/

Q4. Power (12 points):

a. Why is power consumption critical to datacenter operations?

Data Centers are online 24/7. Power cost contributes a significant percentage to the overall operational cost.

b. What is dynamic voltage frequency scaling (DVFS) technique?

DVFS is a computer architecture technique where the frequency of a microprocessor can be dynamically adjusted during operation. When the cpu load is low, the processor frequency can be lowered to conserve power and reduce heat generation, thus reducing power cost and cooling cost.

c. If you were to build a large $100 million data center, which would require $5M/year in power costs to run the data center and $5M/year in power costs to cool the data center with traditional A/C and fans. Name 2 things that the data center designer could do to significantly reduce the cost of cooling the data center?

1. Implement air flow management technologies, which efficiently maintain uniform temperatures across the entire data center floor.
2. Build the data center in areas that have milder climate, which allows better utilization of ambient air temperatures.

d. Is there any way to reduce the cost of cooling in (C)? If yes, how low could the costs go? Explain why or why not?

Using hardwares that are optimized for power consumption may further reduce heat dissipated during operation. However there is still a certain limit on how low the cost can go. Cooling systems for server applications have a baseline cost that cannot be completely nullified. e.g. for dust filtering, humidity control, temperature control, etc.

Q5. Storage (15 points):

a. If a manufacturer claims that their HDD can deliver sub-millisecond latency on average, can this be true? Justify your answer?

I would say it is quite impossible that HDD can deliver sub-millisecond latency on average. The latency of HDD mostly relates to the rotational delay. Therefore, faster the rotational speed of HDD shorter the latency. Nowadays, the highest rpm of top performance HDDs are about 15000 which come with 2ms latency on average. If we want to get a sub-millisecond latency HDD, we must at least double the rotational speed to 30000 rpm. In this case, the HDD will cost too much power and the high rotational speed will threaten the lifetime of the HDD.

b. Explain why flash memory SSD can deliver better performance for some applications than HDD.

Because SSDs have higher data-transfer rates and much lower latency and access times. Unlike HDDs, SSDs don’t need to wait for the read/write head physically move to the location where the data is stored.

c. What types of workloads benefit the most from SSD storage?

Although SSD storage gives laptops a faster solution, I believe that for some workloads like a database server benefit the most from SSD storage. Because SSD not only has better performance but is also a more reliable solution. It costs less power and it produces lower heat. These features are more important to a huge database server.

d. If a manufacturer claims they have built a storage system that can deliver 1 Terabit/second of persistent storage per node, would you believe them? Justify your answer to why this is possible, or not. Make sure to use specific examples of types of hardware and expected performance.

I believe it. Actually there already exist some solutions that can deliver 1Tb/s on the market. Although an individual SSD has a limit of bandwidth. By combining lots of SSDs, we can actually achieve 1Tb/s per node(very expensive).

e. In this problem you are to compare reading a file using a single-threaded file server with a multi- threaded file server. It takes 8 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming the data are in the block cache. If a disk operation is needed (assume a spinning disk drive with 1 head), as is the case one-fourth of the time, an additional 16 msec is required. What is the throughput (requests/sec) if a multi-threaded server is required with 4-cores and 4-threads, rounded to the nearest whole number?

8/4 + 16 = 18 msec/request

1 / 0.018 = 55.5555 = 56 (requests/sec)

Q6. SQL vs Spark (20 points):

a. You hired by a company to help them decide what software stack and hardware they should adopt to store, process, and analyze 500TB (terabyte) of data. Their choices for software stack are: MySQL (https://en.wikipedia.org/wiki/MySQL) and Spark (https://en.wikipedia.org/wiki/Spark_(software)). It has been determined that most queries will only touch 1% of the data using primarily a random-access pattern. The computation to be done seems to be scalable, and that the more computing resources, the faster the computation will run, as long as it can be maintained in memory. The requirement is that there should be at least 224-cores of computing running at 2.7GHz of faster. There are no requirements on the processors used (as long as they are x86 compatible). There should be enough memory to store 1% of the dataset in memory, and there should be enough storage to reliably store 500TB of storage. If a multi-node approach is taken, the network should be as fast as possible (e.g. 100GbE) to ensure good scalability. Assume administration cost is 20% of a full-time system administrator (at a salary of $100,000/year). Assume power costs $0.15 per KWH, and that cooling costs are in-line with the power costs of powering the hardware. Use the ThinkMate website (https://www.thinkmate.com) to come up with the a solution for MySQL and one for Spark in terms of costs over a 5 year period, including hardware, power, cooling, and administration. Note that your solution has to be rack mountable (you cannot use desktops or laptops).

MySQL

Hardware:

Main Server

- Barebone: 2U 4-Node - Intel® C621 Chipset - 24x NVMe - 2200W Redundant Power
- Processor: 8 x Intel® Xeon® Platinum 8280L Processor 28-Core 2.7GHz 39MB Cache (205W)
- Memory: 48 x 128GB PC4-21300 2666MHz DDR4 ECC RDIMM
- M.2 Drive: 4 x 1.0TB Micron 2200 M.2 PCIe 3.0 x4 NVMe Solid State Drive
- U.2 NVMe Drive: 24 x 15.36TB Micron 9300 PRO Series U.2 PCIe 3.0 x4 NVMe Solid State Drive
- I/O Modules - Networking: 4 x Supermicro SIOM 100-Gigabit EDR InfiniBand Adapter AOC-MHIBE-m1CGM (1x QSFP28 & 1x RJ45)
- PCI Express Storage Card: 4 x Intel® Optane™ SSD DC P4800X Series 1.5TB PCIe 3.0 x4 NVMe Solid State Addon Card

Configured Price: $358,741
Power: 2200W

JBOD

- Chassis: Thinkmate® STX-2312 2U Chassis - 12x Hot-Swap 3.5" SATA/SAS3 - 12Gb/s SAS Single Expander - 740W Redundant Power
- Storage Drive: 10 x 16.0TB SATA 6.0GB/s 7200RPM - 3.5" - Seagate Exos X16 Series FastFormat™ (512e/4Kn)

Configured Price: $6,386
Power: 740W

Total Cost (5y)

Hardware - $365,127
Power & Cooling - (((2200+740) * 24 * 365 * 5)/1000) * 0.15 = $19,315.8
Administration - $100,000

Total - $484,442.8

Spark

Hardware:

Servers

- Motherboard
- Intel® C622 Chipset - 12x SATA3 - 1x M.2 - Dual Intel® 10-Gigabit Ethernet (RJ45)
- Processor: Intel® Xeon® Gold 5217 Processor 8-Core 3GHz 11MB Cache

(115W)

- Memory: 6 x 8GB PC4-21300 2666MHz DDR4 ECC RDIMM
- Chassis: Thinkmate® RAX-1208-SH 1U Chassis - 8x Hot-Swap 2.5" SATA/SAS3 - 600W Single Power
- M.2 Drive: 256GB Micron 2200 M.2 PCIe 3.0 x4 NVMe Solid State Drive
- Hard Drive:
  6 x 1.92TB Intel® SSD D3-S4610 Series 2.5" SATA 6.0Gb/s Solid State Drive
  2 x 3.84TB Intel® SSD D3-S4610 Series 2.5" SATA 6.0Gb/s Solid State Drive
- Riser Cards: Thinkmate® 1U Riser Card - Left Slot - 1x PCIe 3.0 x16

Configured Price: $8,880
Power: 254.6 W

Total Cost (5y)

Hardware - $8,880*28 = $248,640
((254.6*28 * 24 * 365 * 5)/1000) * 0.15 = $46,836.216
Administration - $100,000

Total: $395,476.216