HW2

Luke Logan, Gurunath Reddy

February 2020

1 Question 1 Processors

(a) 16-core/32-thread processors will be a central feature of the mainstream AMD market in 2020-2024 (https://www.techradar.c
ryzen-4000).

(b) New AMD processors will have an increase in the number of cores and an increase the number of hardware threads per core. AMD's new processors are being designed with 7nm technology, and include between 2 and 4 hardware threads per core.

(c) Two areas for concern are power and heat. The more cores that are packed into a single processor, the more heat it will produce. Laptop processors with 16 cores will need to sacrifice space and weight in order to cool the processor efficiently when doing computationally intensive workloads (such as gaming). Furthermore, packing more cores into the processor (while keeping clock frequencies similar) will likely increase the power consumption. For laptops, this could decrease battery life.

To improve the performance of processors further, chip makers have staked their future on putting more and more processor cores on the same chip. Limited memory bandwidth and memory-management schemes that are poorly suited to supercomputers, the performance of these machines would level off or even decline with more cores. The performance is especially bad for informatics applications—data-intensive programs that are increasingly crucial to the labs' national security function. These are the challenges future processors will face. (https://spectrum.ieee.org/computing/hardware/multicore-is-bad-news-for-supercomputers)

(d)

(e) Hardware threads are for data-intensive workloads. Hardware threads on a particular core share the same ALU, and the performance increase comes from stalls produced by IO and memory operations.

(f)
- CPU is designed for SISD workloads whereas GPUs are designed for SIMD workloads.
- CPU is designed more for low latency whereas GPU is designed for high throughput.
- The individual cores of a CPU are faster than the individual cores in a GPU.
- The GPU has more cores than a CPU.
- CPU requires less power than a GPU typically.
- CPU is more programmable than GPU. It has a much more general instruction set and has the ability to handle both parallel and serial workloads.

(g)

2 Question 2

(a)

(b) Threads have independent call stacks, instruction pointers, and registers. However, threads live in the same virtual address space and have access to the same file pointers.

(c)

(d)

(e)

(f)

(g) OpenMP has the benefits of being cross-platform, and for some operations, easier. It handles threading in a different way, providing you with higher-level threading options, such as loop parallelization

```
#pragma omp parallel for
for (i = 0; i ¡ 400; i++)
arr[i] = 3 * i;
```

It can be as simple as adding a single pragma, and with linear speed up you'll be 90 percent of the way to properly multithreaded code. It takes a great deal of effort to get the same performance boost with pthreads.

Pthreads is a lower level API for generating threads and synchronization explicitly. In that respect, it provides more control.

3 Question 3

The sequence of events that take place when you visit www.iit.edu (https://medium.com/@RadixRegistry/what-is-dns-and-how-does-it-work-domain-name-system-explained-bcb48c1ff61d):

1. You enter a URL into the browser.
2. The browser looks up the IP address for the domain name. DNS (Domain Name System) is a database that maintains the name of the website (URL) and the particular IP address it links to. Every single URL on the internet has a unique IP address assigned to it. The IP address belongs to the computer which hosts the server of the website we are requesting to access. The DNS lookup proceeds as follows:
   (a) ***B*rowser cache** – The browser caches DNS records for some time for websites you have previously visited. So, it is the first place to run a DNS query. Interestingly, the OS does not tell the browser the time-to-live for each DNS record, and so the browser caches them for a fixed duration (varies between browsers, 2 – 30 minutes).
   (b) ***O*S cache** – If the browser cache does not contain the desired record, the browser makes a system call (gethostbyname in Windows). The OS has its own cache.
   (c) ***R*outer cache** – Third, it checks the router cache. If it's not on your computer the request continues on to your router, which typically has its own DNS cache
   (d) ***I*SP DNS cache** – The next place checked is the cache ISP's DNS server. Your ISP maintains its' own DNS server, which includes a cache of DNS records.
   (e) ***R*ecursive search** – If the requested URL is not in the cache, ISP's DNS server initiates a DNS query to find the IP address of the server that hosts www.iit.edu. DNS query searches multiple DNS servers on the internet until it finds the correct IP address for the website. For www.iit.edu, first, the DNS recursor will contact the root name server. The root name server will redirect it to the .edu domain name server. .edu name server will redirect it to the iit.edu name server. The iit.edu name server will find the matching IP address for www.iit.edu in its' DNS records and return it to your DNS recursor, which will send it back to your browser. These requests are sent using small data packets that contain information such as the content of the request and the IP address of the DNS recursor. These packets travel through multiple networking equipment between the client and the server before it reaches the correct DNS server.
3. The browser initiates a TCP connection with the server. Once the browser receives the correct IP address, a connection will be established to the server that matches the IP address for the information transfer. Browsers build such connections using internet protocols. There are several different internet protocols to use, but TCP is the most commonly used protocol for many types of HTTP requests. To transfer data packets between your computer(client) and the server, it is important to have a TCP connection established. This connection is established using a process called the TCP/IP three-way handshake. This is a three-step process where the client and the server exchange SYN(synchronize) and ACK(acknowledge) messages to establish a connection.
   (a) The client machine sends a SYN packet to the server over the internet, asking if it is open for new connections.
   (b) If the server has open ports that can accept and initiate new connections, it'll respond with an ACKnowledgment of the SYN packet using a SYN/ACK packet.
   (c) The client will receive the SYN/ACK packet from the server and will acknowledge it by sending an ACK packet.
4. The browser sends an HTTP request to the webserver. The browser will send a GET request asking for www.iit.edu web page. The GET request names the URL to fetch: "http://www.iit.edu". The browser identifies itself (User-Agent header), and states what types of responses it will accept (Accept and Accept-Encoding headers). The Connection header asks the server to keep the TCP connection open for further requests.

```
GET http://www.iit.edu/ HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, [...]
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; [...]
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: iit.edu
Cookie: datr=1265876274-[...]; locale=en_US; lsd=WW[...]; c_user=2101[...]
```

5. The server handles the request and sends out an HTTP response. The server response contains the web page you requested as well as the status code, compression type (Content-Encoding), how to cache the page (Cache-Control), any cookies to set, privacy information, etc.

6. The browser displays the HTML content (for HTML responses). The browser shows the contents of the HTML in phases. First, it'll render the HTML skeleton of bone bare. It will then search the HTML tags and submit GET requests for additional web page things such as images, CSS stylesheets, JavaScript files, etc. The browser caches these static files, so the next time you visit the page it doesn't have to fetch them again.

4 Question 4

(a) The more power a system requires, the higher the energy bill. Modern data centers require many computers running all the time in order to meet the demands of consumers, which requires a substantial amount of energy, which becomes very costly.

(b) DVFS changes the clock frequency of cores on a processor in order to decrease the power consumption of the processor. For example, in a memory-bound workload, if a processor is stalled waiting for data, the frequency of the processor is dropped to match the speed of memory. In a multi-core environment, some cores are given a lower frequency if the workload is highly parallel, whereas other cores are given a higher frequency if the workload is largely serial.

(c) (a) **Hot Aisle/Cold Aisle Containment**: Keeps hot air from mixing with cold air produced by AC.

  (b) **Portable cooling**: Allows you to be more flexible with where cold air goes so that "hot spots" are avoided.

(d) Immersion cooling can reduce the cost of traditional air cooling by as much as 50% (https://www.grcooling.com/air-based-cooling-vs-liquid-based-cooling). Thus, the cost of cooling this data center would drop to $2.5M.

5 Question 5

(a) The fastest HDDs today are 15,000RPM.
Latency = Seek + Rotation + Transfer.
On average, you will perform half of a rotation in order to access a piece of data on the disk.
On a 15,000RPM drive, the rotational cost is:
(15,000 rotations/minute)(1 minute/60 seconds)(1 second/1000 ms) = .25 rotations/ms
(.5 rotations) / (.25 rotations/ms) = 2ms.
Thus, the HDD should be at least 2ms on average for data access rates. Sub-millisecond data access on average is not currently possible.

(b) SSDs have no moving parts; they are essentially HDDs without the seek or rotation penalties. SSDs access any memory location at about the same speed.

(c) HPC workloads benefit greatly from SSDs. HPC workloads generate vast amounts of data that needs to be read/written quickly. The problem with just reading/writing directly to HDDs is that HDDs are very slow. In order to account for this performance concern, intermediate storage devices (like SSDs) are introduced to temporarily cache data that needs to be accessed and modified. Eventually, changes made to data stored in intermediate storage will be persisted to the HDDs.

(d) Building a storage system that delivers 1 Terabit/s data access rates is possible through concurrency.
3D Xpoint SSDs can have up to roughly 9GB/s write performance (https://investors.micron.com/news-releases/news-release-details/micron-brings-3d-xpointtm-technology-market-worlds-fastest-ssd).
Let's say we have a storage node with 16 3D Xpoint SSDs.
We write 128GB of data (1 Terabit = 128GB) to the SSDs using striping.
In other words, each SSD gets 128/16 = 8GB of data.
(8GB)/(9 GB/s) = .89s.
The overall cost to write 1 Terabit of data to this storage node would be .89s.

(e) 8ms per request if request in cache (75% of the time).
16ms per request if request on disk (25% of the time).
4 cores and 1 thread per core.
Cache accesses can be done in parallel.
Disk accesses must be done sequentially.

Throughput = (1 request)/(avg time to complete request)

One a single-threaded file server:
Throughput = (1 req)/(8ms*.75 + 16ms*.25) = .1 req/ms = 100 req/s

On a multi-threaded file server:
Throughput = (1 req)/(8ms*.75/4 + 16ms*.25) = .18182 req/ms = 181.82 req/s

6 Question 6

6.1 Requirements

x86 processors with a total of 224 cores and at least 2.7GHz per core.
500TB of storage.
5TB of RAM (1% of total storage).
100GBe interconnect (if more than one node).

6.2 MySQL

MySQL is best used for shared-memory architectures.

6.2.1 Architecture

RAX XT24-42S1
4x Intel® Xeon® Platinum 8280L Processor 28-Core 2.7GHz 39MB Cache (205W)
24x 128GB PC4-21300 2666MHz DDR4 ECC LR-DIMM
18x 15.36TB SAS 12.0Gb/s Solid State Drive - 2.5" - HGST Ultrastar™ DC SS530 Series (1x DWPD)
3x Supermicro AOC-S3008L-L8e SAS 3.0 12Gb/s 8-Port Host Bus Adapter
1x Mellanox 56Gb/s FDR InfiniBand Adapter ConnectX®-4 VPI (2x QSFP28) - PCIe 3.0 x8
1x IEC60320 C13 to C14 Power Cable, 16 AWG, 240V/15A, Black - 1'
$250,422 per node
1366.1 Watts per node
2x nodes

6.2.2 Cost

Power Consumption: 1366.1*2 = 2732.2 Watts = 2.7322 KW
Server Cost: $250,422*2 = $500,844 for the server.
Energy Cost: (.15 $/KWH)*(2.7322KW)*(24 hrs/day)*(365 days/yr)*(5 yrs) = $17,950.55
Administrative Cost: (.2 * 100,000 $/yr)*(5 yrs) = $100,000
Total: $618,795.55

6.3 Spark

Spark is best suited for distributed architectures.

6.3.1 Architecture

RAX QS12-12E1
1x AMD EPYC™ 7282 Processor 16-core 2.80GHz 64MB Cache (120W)
8x 64GB PC4-23400 2933MHz DDR4 ECC RDIMM
3x 16.0TB SAS 12.0Gb/s 7200RPM - 3.5" - Seagate Exos X16 Series FastFormat™ (512e/4Kn)
1x Mellanox 100Gb/s EDR InfiniBand Adapter ConnectX®-5 VPI (1x QSFP28) - PCIe 3.0 x16
2x IEC60320 C13 to C14 Power Cable, 16 AWG, 240V/15A, Black - 6'
$8,318 per node.
265.4 Watts per node
14x nodes

6.3.2 Cost

Power Consumption: 265.4*14 = 3715.6 Watts = 3.7156 KW
Server Cost: $8,318*14 = $116,452 for the server.
Energy Cost: (.15 $/KWH)*(3.7156KW)*(24 hrs/day)*(365 days/yr)*(5 yrs) = $24,411.5
Administrative Cost: (.2 * 100,000 $/yr)*(5 yrs) = $100,000
Total: $240,863.5