High Performance Computing

# Assignment 1

# High Level basic theory

1. Moore’s Law says that the number of transistors on microchips will double almost every 2 years. Although when this was stated, there was nothing known as the “Node”. A node is like numbers a unit of nanometres that indicate a key dimension in a device. The way these “dimensions” were measured depended on the dimensions of the key features in the IC chip. The 2 key features in the Integrated circuit - “Gate Length” and the “Metal Half-Pitch”.

“Gate Length” is simply the length of the gate/the path that links the charge carriers between the drain and the source. “Metal Half-Pitch” is half of the distance from the start of one metal that interconnects to the start of next metal on a chip. To accommodate the Moore’s Law, as we tried to decrease the size, these 2 decreases, equivalently and therefore this became the node number. Until almost mid 1990’s, after which the industry decided to only decrease the gate length. And so, the gate length was reduced at almost 3 times the factor of the metal pitch. Ultimately, Moore’s Law was still held true – Transistors increased, lengths and width decreased, density increased (number of transistors per square unit), heat increased. Even though these proportions were not equivalent anymore and the numbers were far from what the ”measurement” said, the measurement unit was still decided to be kept as node number.

Another thing to consider is that, at one point the size of the chip will stop decreasing. This does not mean that Moore’s Law is not valid anymore. This is the limit of Lithography and not Moore’s Law. Currently we are using the scalar or 2-D integration. However, sooner or later we will need to adapt the 3-D integration, which means, adding tiers or layers of devices in order to keep the density increasing per unit area.

**What can be the new measurement?** Referring to the “The Node is Nonsense” article by Samuel K Moore, there are a few possible solutions. At the end we need to divide what “number” can be aligned to the “node number”. Gargani, the chairman of the IEEE International Roadmap for Devices and Systems (IRDS) proposed that we should be adopting a 3-number metric that includes the ‘Gate Length (G)’, ‘Metal Pitch (M)’ and the ‘Number of Layers/tiers of devices on the chip (T)’. The Tiers will be useful in the future when we will require tiers in the integration. This could be the **GMT** **Method** for measurement.

We can no longer rely on a single number as a way of measurement of the progress. I believe therefore, the 3-number metric system should be a good ‘next way’ of measurement for semiconductor nodes. There might even come a time when these size units of the processors won’t be very ‘attractive’ and instead people might want to use a few key-performance impacting aspects of the entire computer system. This has also been proposed by a Stanford University professor, H -S Philip Wong. Wong suggests that we can use the density of each of the above components as Density of Logic (DL), Density of Main Memory (DM), and Density of the interconnects linking them (DC) and call it the **LMC Method**. The overall speed and energy efficiency of computing systems depends on these 3 values and therefore taking these 3 as the parameters of measurement would be a better option than the GMT Method.

If there is a possibility that the values in the GMT method might get ‘too small’ or might prove again to be not useful in some years, then why adopt for that method? LMC method includes DL which is the number of transistors that is being measured since the start, DM is the DRAM (main memory) and DC is the measurement of interconnection between the processor and the memory. All these 3 values are the best parameters in terms of measuring the performance and providing the ‘specs’ of a semiconductor correctly or at least in not a misleading way. Therefore, giving the **LMC** method a chance might go a long way.

**Reference**: Moore, S. K. (2020, August). *The node is nonsense*. Digital Content Experience Platform. Retrieved January 30, 2023, from https://read.nxtbook.com/ieee/spectrum/spectrum\_na\_august\_2020/the\_node\_is\_nonsense.html

1. Estimating the cost of replacing the nodes based on modern equivalent architecture and configurations:

**Dual Sockets**

For 150 nodes with GPU, **Gigabyte Server R283-Z93 Rev AAF1 – 2U AMD Epyc 9004 – 2 GPUs – 8x 3.5” SAS/SATA + 4x3.5” Hybrid – Dual GbE (RJ45) – 2400W Rdt**

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Product/ Specs PER NODE** | **Cost** |
| Processors | 2 X 4th Generation AMD Epyc 9354 Processor 32-core 3.25GHz 256MB Cache (280W) ($15,409.00) | **$30,818** |
| Memory (RAM) 512GB | 2X 16 X 32GB PC5-38400 4800MHz DDR5 ECC RDIMM (+$1600.00) | **+ $3200.00** |
| Local Storage (SSD) 4.0TB | 2 X 2.0TB Kioxia XG6-P M.2 PCIe 3.1 x4 NVMe Solid State Drive | **+932.00** |
| Hard Drive (HDD) 2.0TB | 2TB SATA 6.0GB/s 7200RPM - 3.5" - Seagate Exos 7E10 Series (512e/4Kn) | **+$149** |
| GPU | 4 X NVIDIA® A16 GPU Computing Accelerator - 64GB (4x 16GB) GDDR6 - PCIe 4.0 x16 - Passive Cooler ($3499) | **+ $13,996.00** |
| **Total for 1 Node Configure Price Silicon Mechanics** | | **= $50,695** |
| **For 150 Nodes** | | **= $7,604,250** |

For 800 Nodes without GPU, **Gigabyte Server R283-Z93 Rev AAF1 – 2U AMD Epyc 9004 – 2 GPUs – 8x 3.5” SAS/SATA + 4x3.5” Hybrid – Dual GbE (RJ45) – 2400W Rdt**

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Product/ Specs PER NODE** | **Cost** |
| Processors | 2 X 4th Generation AMD Epyc 9354 Processor 32-core 3.25GHz 256MB Cache (280W) ($15,409.00) | **$30,818** |
| Memory (RAM) 512GB | 2X 16 X 32GB PC5-38400 4800MHz DDR5 ECC RDIMM (+$1600.00) | **+ $3200.00** |
| Local Storage (SSD) 4.0TB | 2 X 2.0TB Kioxia XG6-P M.2 PCIe 3.1 x4 NVMe Solid State Drive | **+932.00** |
| Hard Drive (HDD) 2.0TB | 2TB SATA 6.0GB/s 7200RPM - 3.5" - Seagate Exos 7E10 Series (512e/4Kn) | **+$149** |
| **Total for 1 Node Configure Price Silicon Mechanics** | | **= $35,099** |
| **For 800 Nodes** | | **= $28,079,200** |

**Total Cost for replacing 950 Nodes Dual Socket = $7,604,250 + $28,079,200 = $35,683,450**

**Single Socket**

For 150 nodes with GPU, **Gigabyte Server R162-Z10 - 1U - AMD EPYC 7003 - 10x U.2 NVMe - Dual-Port Ethernet - Redundant 1200W**

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Product/ Specs PER NODE** | **Cost** |
| Processors | 1 3rd Generation Single Socket A MD EPYC™ 7313P Processor 16-core 3.00GHz 128MB Cache (155W) | **$5,401.00** |
| Memory (RAM) 512GB | **8** X 64GB PC4-25600 3200MHz DDR4 ECC RDIMM | **$1440.00** |
| Local Storage (SSD) 4.0TB | 2 X 2.0TB Kioxia XG6-P M.2 PCIe 3.1 x4 NVMe Solid State Drive | **$932.00** |
| Hard Drive (HDD) 1.0TB | 1.0TB SATA 6.0Gb/s 7200RPM - 2.5" - Seagate Exos 7E2000 Series (512e) | **$213.00** |
| GPU | 2X NVIDIA® Tesla™ T4 GPU Computing Accelerator - 16GB GDDR6 - PCIe 3.0 x16 - Passive Cooling | **4598.00** |
| **Total for 1 Node Configure Price Silicon Mechanics** | | **$11,652.00** |
| **For 150 Nodes** | | **$1,747,800** |

For 800 Nodes without GPU, **Gigabyte Server R162-Z10 - 1U - AMD EPYC 7003 - 10x U.2 NVMe - Dual-Port Ethernet - Redundant 1200W**

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Product/ Specs PER NODE** | **Cost** |
| Processors | 1 3rd Generation Single Socket A MD EPYC™ 7313P Processor 16-core 3.00GHz 128MB Cache (155W) | **$5,401.00** |
| Memory (RAM) 512GB | **8** X 64GB PC4-25600 3200MHz DDR4 ECC RDIMM | **$1440.00** |
| Local Storage (SSD) 4.0TB | 2 X 2.0TB Kioxia XG6-P M.2 PCIe 3.1 x4 NVMe Solid State Drive | **$932.00** |
| Hard Drive (HDD) 1.0TB | 1.0TB SATA 6.0Gb/s 7200RPM - 2.5" - Seagate Exos 7E2000 Series (512e) | **$213.00** |
| **Total for 1 Node Configure Price Silicon Mechanics** | | **$7054.00** |
| **For 800 Nodes** | | **$5,643,200** |

**Total Cost for replacing 950 Nodes Dual Socket = $1,747,800 + $5,643,200 = $7,391,000**

1. The power consumption depends on various factors and components of the system and then type of operations and workload as well. But making a few general assumptions, 2 socket 4 GPU setup can take up to 2400Watt of power (as seen while doing the previous questions).

**Power** = 2400 Watt = 2.4 kW

**Hours in a Week =** 24 hours \* 7 days = 168 hours

**Power consumed in a week** = 2.4 kW \* 168 hours = 403.2 kWh

**Power consumed in a month** = 403.2 kWh \* 4 weeks = 1612.8 kWh

Considering the Toronto Hydro rates (Non-residential customers: for electricity used above 750 kWh/month)

**Unit Power Cost** = 10.3c per kWh = $0.103/kWh

**Power consumed in a year** = Power consumed in a week \* 52 weeks

**=** 403.2 kWh \* 52 = 20,966.4 kWh

**Power Cost for a year =** Power consumed in a year\* Unit Power Cost

**=** 20966.40 kWh \* $0.103/kWh

**=**  $2159.54

**Power cost of running ONE NODE for a year = ~$2159.54**

**Assuming 950 Nodes = $2159.24 \* 950 = $2,051,278**

# Programming

1. CPU Benchmarking Program
2. Throughput – every 1 second print off how many times it was able to run the solutions, run this for 10 seconds.



1. Capacity computing (job time) – calculate how long it takes for 100, 200, 500, 1000, 2500, 5000, 7500 and 10 000 runs of the calculation. (If that takes absurdly long on your machine pick smaller numbers, I don’t actually care what it is, it seems like 10 000 should take about 5 minutes). Plot the result (feel free to use excel for the plot). Text

   Description automatically generated
2. Parallelize with OpenMP and see how much (if any) improvement you get to one of the longer calculations in b).

Text

Description automatically generated

# Linux Commands

1. Demonstrate a basic understanding of Linux Commands based on the following
2. Using the Linux pipe command, output the result of your program in 4 to a file (even if your program does not work you should know how to do this). Paste that file here.

**g++ assn1.cpp -o ass1\_output | export OMP\_NUM\_THREADS=8 | ./ass1\_output > outputfile.txt**

Graphical user interface, text

Description automatically generated Graphical user interface

Description automatically generated with low confidence

1. While your program in 4 is running doing its thing (or something else is running, doesn’t matter what) use *ncdu* and *top* to see how much memory and what percentage of CPU your program uses. Show your output here. (You can do that with WSL if you need to).

Text

Description automatically generated with medium confidence

1. Use Grep to find all instances of the word “and” on the syllabus (you’ll need to save the syllabus to .txt first), and print out the lines they are on.

**grep -n “ and ” syllabus.txt**

Text

Description automatically generated

1. Tar up your assignment and submit it (name it for your trent username.tar.gz), on future assignments feel free to use zip or rar like a normal person.

**tar -cf Punyaja\_Mishra\_Assn1.tar assn1.cpp ass1\_output outputfile.txt syllabus.txt MishraAssignment1.docx**

**Text

Description automatically generated**