Computer Architecture TMP10284
Guangdong University of Technology, Fall 2021

Homework Assignment 1 [released Sept. 8th 2021] [due* Friday Feb 18th 2021, before 5 PM, BEIJING TIME] You are allowed to discuss homework assignments only with other colleagues taking the class. You are not allowed to share your solutions with other colleagues in the class. Please feel free to reach out to the TA or to the Instructor during office hours or by appointment if you need any help with the homework. Please enter your responses in this Word document after you download it from course website. Please use the Online Classes portal to send in your completed homework (Your student no. is the account no., password is 888888). If you are having difficulty doing this before the deadline, please convert it to PDF when you are done and email it to ustchenlong@gdut.edu.cn before 5 PM on Saturday Sept. 18th 2021.

---

1. Assume a Computer uses these components

Component	Mean time to failure (in units of 10^6 hours)	Number of these
CPU	8	1
Memory stick	6	3
Hard drive	1	2
Power supply	4	1

Assume 106 hours = 114.1 years.

Determine: (i) [10] Mean time to failure of the computer, assuming it fails if a single component fails.

MTTF = 1/failure rate;
failure rate = (1/8*1000000)+(3/6*1000000)+(2/1*1000000)+(1/4*1000000);
= (1/1000000)+(1/8+1/2+2/1+1/4)= (1+4+8+2)/8 = 23/8*1000000;
MTTF = 8000000/23 = 347826hours or 115*(8/23) = 40 years;

(ii) [10] Mean time to failure of a cluster of 144 computers, assuming the cluster fails if a single computer fails.

Failure rate of 144 computers = 144*(23/8*1000000)
MTTF = 1/failure rate = 8000000/144*23 = 8000000/3312 = 2415.46 hours

(iii) [10] Number of working computers in a cluster of 144 computers after a year

Working computers = $144*(23/8*1000000)*10^6/114.1 = 3.628$ computers
failure rate in a year (As per proff. number of working computers is equal to failure in time multiplied by the time units corresponding to a year.)

Total working computers by the end of the year = 144 – (~4) = 140

2. Power Consumption in Computer Systems [Case Study 2 from Textbook reproduced here in case you do not have the text yet]

Power consumption in modern systems is dependent on a variety of factors, including the chip clock frequency, efficiency, and voltage. The following exercises explore the impact on power and energy that different design decisions and use scenarios have.

[A] A cell phone performs very different tasks, including streaming music, streaming video, and reading email. These tasks perform very different computing tasks. Battery life and overheating are two common problems for cell phones, so reducing power and energy consumption are critical. In this problem, we consider what to do when the user is not using the phone to its full computing capacity. For these problems, we will evaluate an unrealistic scenario in which the cell phone has no specialized processing units. Instead, it has a quad-core, general-purpose processing unit. Each core uses 0.5 W at full use. For email-related tasks, the quad-core is 8× as fast as necessary.

a. [10] How much dynamic energy and power are required compared to running at full power? First, suppose that the quad-core operates for 1/8 of the time and is idle for the rest of the time. That is, the clock is disabled for 7/8 of the time, with no leakage occurring during that time. Compare total dynamic energy as well as dynamic power while the core is running.

quad core = 4 core, then full power should be 2W i.e the nominal energy will be 2T, and emails will do 8x frequency.

So, for time x/8, voltage is applied and there is consumption, and for 7x/8 no energy consumption.
So, for 1/8 time, the dynamic energy is reduced to 1/8 of the comsumption when core is running for full time. $E1 = 2*(1/8)T = T/4 = E_0/8$
power is instantaneous rate of energy consumption, so no change when the core is running, so power consumption is unchanged.

Note that Energy=Power X Time

b. [10] How much dynamic energy and power are required using frequency and voltage scaling? Assume frequency and voltage are both reduced to 1/8 the entire time.

General Formulaes:-

Dynamic_Energy(Edy) = $k*Cl*V^2$; k is a proportionality constant, Cl is the capacitive load, and V is the voltage

Dynamic_Power(Pdy) = $k*Cl* V^2*Fclk$; k is a proportionality constant, Cl is the capacitive load, and V is the voltage, Fclk is the switching frequency.

$E_{dy,new} = k*Cl*(V/8)^2 = k*Cl*V^2/64;$

Edy,new/Edy,old = k*Cl*(V/8)^2 / k*Cl*(V)^2; Edy,old = 64*Edy,new;

$P_{dy,new} = k*Cl*(V/8)^2*(F/8) = k*Cl*V^2*F/512$

Pdy,new/Pdy,old = k*Cl*(V/8)^2*(F/8) / k*Cl*(V)^2*F; Edy,old = 512*Edy,new;

c. [10] Now assume the voltage may not decrease below 50% of the original voltage. This voltage is referred to as the voltage floor, and any voltage lower than that will lose the state. Therefore, while the frequency can keep decreasing, the voltage cannot. What are the dynamic energy and power savings in this case?

Here, voltage can go below upto V/2, frequency can go below upto f/8 only.

So,

$E_{dy,new} = k*Cl*(V/2)^2 = k*Cl*V^2/4 = E_{old}/4$

$P_{dy,new} = k*Cl*(V/2)^2*F/8 = k*Cl*V^2/32 = P_{old}/32$

d. [10] How much energy is used with a dark silicon approach? This involves creating specialized ASIC hardware for each major task and power gating those elements when not in use. Only one general-purpose core would be provided, and the rest of the chip would be filled with specialized units. For email, the one core would operate for 25% the time and be turned completely off with power gating for the other 75% of the time. During the other 75% of the time, a specialized ASIC unit that requires 20% of the energy of a core would be running.

Consider the Energy required by the original processor with 4 cores be E,

So for first 25%, Total Energy = (25/100)*E/4 (Since, we are using only one core, so E/4).
And for remaining 75%, Total Energy = (75/100)*E/4*(20/100) (Since ASICs use 20% of the Energy of a single core)

Total energy = (25/100)*E/4 + (75/100)*E/4*(20/100) = (E/100)*(25/4 + 15/4) = E/10 =0.1E

[B] As mentioned in [A], cell phones run a wide variety of applications. We'll make the same assumptions for this exercise as the previous one, that it is 0.5 W per core and that a quad

core runs email 3× as fast.

a. [10] Imagine that 80% of the code is parallelizable. By how much would the frequency and voltage on a single core need to be increased in order to execute at the same speed as the four-way parallelized code?

$Speedup_{original} = 1/(1-f_{enhanced}) + f_{enhanced}/speedup_{enhanced}$

So, $f_{enhanced} = 0.8$ and $speedup_{enhanced}$ will be 4 since the code is parallelized 4 way.

$Speedup_{original} = 1/(0.2+0.8/4) = 2.5$.

So a single core's performance and voltage would need to increase 2.5X to execute at the same speed as a 4 way parallized code.

b. [10] What is the reduction in dynamic energy from using frequency and voltage scaling in part a?

With each of the 4 cores running at 1/2.5 voltage and frequency are

Energy: $Energy_{quad} / Energy_{single} = 4(Voltage*(1/2.5))^2 / (Voltage)^2 = 0.64$

Power: $Power_{quad} / Power_{single} = 4((Voltage*(1/2.5))^2 * Freq*1/2.5) / (Freq* (Voltage)^2) = 0.256$

c. [10] How much energy is used with a dark silicon approach? In this approach, all hardware units are power gated, allowing them to turn off entirely (causing no leakage). Specialized ASICs are provided that perform the same computation for 20% of the power as the general-purpose processor. Imagine that each core is power gated. The video game requires two ASICS and two cores. How much dynamic energy does it require compared to the baseline of parallelized on four cores?

Parallelized on 4 core:-

Lets Energy of parallelized on 4 core be 4E, So a single core be E

Two ASICs plus two two cores, i.e. 2E+.2*2*E = 2.4E

Dynamic Energy compared to parallelized 4 core = 2.24E/4E = 0.6E

[C] General-purpose processes are optimized for general-purpose computing. That is, they are optimized for behavior that is generally found across a large number of applications. However, once the domain is restricted somewhat, the behavior that is found across a large number of the target applications may be different from general-purpose applications. One such application is deep learning or neural networks. Deep learning can be applied to many different applications, but the fundamental building block of inference—using the learned information to make decisions—is the same across them all. Inference operations are largely parallel, so they are currently performed on graphics processing units, which are specialized more toward this type of computation, and not to inference in particular. In a quest for more performance per watt, Google has created a custom chip using tensor processing units to accelerate inference operations in deep learning.1 This approach can be used for speech recognition and image recognition, for example. This problem explores the trade-offs between this process, a general-purpose processor (Haswell E5-2699 v3) and a GPU (NVIDIA K80), in terms of performance and cooling. If heat is not removed from the computer efficiently, the fans will blow hot air back onto the computer, not cold air. Note: The differences are more than processor—on-chip memory and DRAM also come into play. Therefore statistics are at a system level, not a chip level.

a. [10] If Google's data center spends 70% of its time on workload A and 30% of its time on workload B when running GPUs, what is the speedup of the TPU system over the GPU system?

Speedup of computer A over Computer B = ET of B / ET of A – (1)

$(\text{Speedup of TPU Over GPU})_A = 16.7$ – (2)

$(\text{Speedup of TPU Over GPU})_B = 7.7$ – (3)

$(ET)_{A,TPU} * 16.71 = (ET)_{A,GPU} \Rightarrow (ET)_{A,TPU} * 16.71 = 0.7*T$

$(ET)_{B,TPU} * 7.7 = (ET)_{A,GPU} \Rightarrow (ET)_{A,TPU} * 7.7 = 0.3*T$

$\text{Speedup}_{\text{TPU over GPU}}$ = ET,GPU,A and B / ET, TPU, A and B

$T/ (.7/16.71 + .3/7.7)*T = 12.37$

b. [10] If Google's data center spends 70% of its time on workload A and 30% of its time on workload B when running GPUs, what percentage of Max IPS does it achieve for each of the three systems?

General Purpose Unit, 0.42*0.7+0.3*1 = 0.594

Graphical Processing Unit, 0.37*.7+0.3*1 = 0.559

TPU, 0.8*0.7 + 0.3*1 = 0.86

c. [15] Building on (b), assuming that the power scales linearly from idle to busy power as IPS grows from 0% to 100%, what is the performance per watt of the TPU system over the GPU system?

Performance per Watt = Throughput/Watt

Performance per Watt, TPU over GPU = (Throughput, TPU/Watt, TPU)/ (Throughput, TPU/Watt, TPU)

⇨ (0.86/(384-634))/(.559/(991-357)) = (0.86*634)/(0.559*94) = 545.24/52.54 = 10.376

d. [10] If another data center spends 40% of its time on workload A, 10% of its time on workload B, and 50% of its time on workload C, what are the speedups of the GPU and TPU systems over the general-purpose system?

Speedups, GPU over General Processor

$1/ (0.4/(SP_{TPU})_A + 0.1/(SP_{TPU})_B + 0.5/(SP_{TPU})_A)$

1/(.4/41.04 +.1/21.22 + .5/.167)

$1/(.0097+.0047+2.994)$

$= 0.332$

Speedups, TPU over General Processors,

$1/ (0.4/(SP_{TPU})_A + 0.1/(SP_{TPU})_B + 0.5/(SP_{TPU})_A)$

$1/(.4/20.69 +.1/2.763 + .5/1.25)$

$1/(.0193+.036+0.4)$

$= 2.19$

e. [10] A cooling door for a rack costs $4000 and dissipates 14 kW (into the room; additional cost is required to get it out of the room). How many Haswell-, NVIDIA-, or Tensor-based servers can you cool with one cooling door, assuming TDP in Figures 1.27 and 1.28?

Haswell, H = 14000/504 = 27.78

NVIDIA, N = 14000/1838 = 7.6

TPU, T = 14000/861 = 16.26

f. [20] Typical server farms can dissipate a maximum of 200 W per square foot. Given that a server rack requires 11 square feet (including front and back clearance), how many servers from part (e) can be placed on a single rack, and how many cooling doors are required?

General Processor => 4, GPU => 1 and TPU =>2

System	Chip	TDP	Idle power	Busy power
General-purpose	Haswell E5-2699 v3	504 W	159 W	455 W
Graphics processor	NVIDIA K80	1838 W	357 W	991 W
Custom ASIC	TPU	861 W	290 W	384 W

Figure 1.27 Hardware characteristics for general-purpose processor, graphical processing unit-based or custom ASIC-based system, including measured power

System	Chip	Throughput			% Max IPS		
		A	B	C	A	B	C
General-purpose	Haswell E5-2699 v3	5482	13,194	12,000	42%	100%	90%
Graphics processor	NVIDIA K80	13,461	36,465	15,000	37%	100%	40%
Custom ASIC	TPU	225,000	280,000	2000	80%	100%	1%

Figure 1.28 Performance characteristics for general-purpose processor, graphical processing unit-based or custom ASIC-based system on two neural-net workloads

3. Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a. [10] Which processor has the highest performance expressed in instructions per second?

P1= 3GHz, CPI = 1.5; CPU Time = Instruction Count(IC) * Clock cycles per Instruction(CPI) * Clock cycle time

i.e. CPU,time = IC*CPI*(1/Fr)

So,

$(\text{Execution time})_{P1} = 1.5*(1/3*10^{9}) = 5*10^{-10}$

$(\text{Execution time})_{P2} = 1.0*(1/2.5*10^{9}) = 4*10^{-10}$

$(\text{Execution time})_{P3} = 2.2*(1/4.0*10^{9}) = 0.55*10^{-10}$

Now, Performance= 1/Execution Time

So, Best performance is given by P2

b. [10] If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

CPU Time = Instruction Count(IC) * Clock cycles per Instruction(CPI) * Clock cycle time

i.e. CPU,time = IC*CPI*(1/Fr)

$10 = (IC)_{P1} * 5*10^{-10}$; number of cycles = $(IC)_{P1} * (CPI)_{P1}$; number of cycles = $3*10^{10}$

$(IC)_{P1} = 2*10^{10}$

$10 = (IC)_{P2} *4*10^{-10}$; number of cycles = $(IC)_{P2} * (CPI)_{P2}$; number of cycles = $2.5*10^{10}$

$(IC)_{P2} = 2.5*10^{10}$

$10 = (IC)_{P3} *18.2*10^{-10}$; number of cycles = $(IC)_{P2} * (CPI)_{P2}$; number of cycles = $4*10^{10}$

$(IC)_{P2} = 1.82*10^{-9}$

c. [10] We are trying to reduce the execution time by 30%, but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

Now, Execution time = CPI * (1/clock rate or frequency)
So, clock rate = CPI/Execution Time; Therefore, new CPI = CPI(1+1/5)=6/5*CPI
New Execution time = Execution time(1-3/10)=7/10*(Exec. Time)
So $(ET)_{new} = (CPI)_{new}*(1/F_{new})$; ie, $7/10*ET = 6/5*CPI*(1/F_{new})$
$F_{new} = 12/7*f$ i.e the frequency have to be increased by 71.8 percentage!!

4. Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (classes A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2.

a. [10] Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, Which is faster: P1 or P2?

CPU Time = Instruction Count(IC) * Clock cycles per Instruction(CPI) * Clock cycle time

i.e. CPU,time = IC*CPI*(1/Fr)

P1:- $(CPUt)P1 = 10^6*(1/2.5*10^9)[1*1/10 + 2*2/10 + 3*5/10 + 3*2/10] = 26*10^6*10/10*25*10^9 = 1.05*10^{-3}$ seconds

P2:- $(CPUt)P2 = 10^6*(1/3*10^9)[2*1/10 + 2*2/10 + 2*5/10 + 2*2/10] = 2*10^6*10/10*3*10^9 = 0.667*10^{-3}$ seconds

So, P2 is faster

b. [10] What is the global CPI for each implementation?

Global CPI, P1 => $2.6*10^3 = 10^6*CPI*(1/2.5*10^9) = 2.6$

Similary, CPI, P2 = 2

c. [10] Find the clock cycles required in both cases.

Clock cycle, P1 = 10^6* [1*1/10 + 2*2/10 + 3*5/10 + 3*2/10] = $2.6*10^6$

Clock cycle, P2 = 10^6* [2*1/10 + 2*2/10 + 2*5/10 + 2*2/10] = $2.0*10^6$