1. 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 2.0 GHz and CPIs of 1, 2, 2, and 1, and P2 with a clock rate of 4 GHz and CPIs of 2, 3, 4, and 4.

a. 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 (in total execution time)?

b. What is the global CPI for each implementation?

c. Find the clock cycles required in both cases.

d. Which processor has the highest throughput performance (instructions per second) ?

e. Which processor do you think is more energy efficient? Why?

1a.

*CPI of P1:0.1*1+0.2*2+0.5*2+0.2*1=1.7*

*CPI of P2:0.1*2+0.2*3+0.5*4+0.2*4=3.6*

*Time of p1:(CPI/FCLK)*IC=(1.7/2.0GHz)*1.0E6=8.5*10^{-4}S*

*Time of p2:(CPI/FCLK)*IC=(3.6/4.0GHz)*1.0E6=9*10^{-4}S*

P1 is faster

1b.

*CPI of P1:0.1*1+0.2*2+0.5*2+0.2*1=1.7*

*CPI of P2:0.1*2+0.2*3+0.5*4+0.2*4=3.6*

1c.

Clock cycles required = CPI x IC

*P1:1.7*1.0E6=1.7M cycles*

*P2:3.6*1.0E6=3.6M cycles*

1d.

*instructions per second=1S*FCLK/CPI*

P1 instructions per second:2/1.7=1.1765G

P2 instructions per second:4/3.6=1.1111G

So P1 has the highest throughput performance.

1e.

P1.P1 has less execution time,lower clock rate,lower CPI and better throughput performance.

2. You are designing a system for a real-time application in which specific deadlines must be met. Finishing the computation faster gains nothing. You find that your system can execute the necessary code, in the worst case, twice as fast as necessary.

a. How much energy do you save if you execute at the current speed and turn off the system when the computation is complete?

b. How much energy do you save if you set the voltage and frequency to be half as much?

2a.

No energy is saved.According to the formula $E=1/2*C*V^2$.So completing the task quickly won't save any energy.

2b.

Energy=$1/2C*V^2$.After the change,the new energy is $1/2Cx(0.5V)^2$,reducing it to 1/4 of the old energy.So the energy will save 75%.

3. Server farms such as Google and Yahoo! provide enough compute capacity for the highest request rate of the day. Imagine that most of the time these servers operate at only 60% capacity. Assume further that the power does not scale linearly with the load; that is, when the servers are operating at 60% capacity, they consume 90% of maximum power. The servers could be turned off, but they would take too long to restart in response to more load. A new system has been proposed that allows for a quick restart but requires 20% of the maximum power while in this "barely alive" state.

a. How much power savings would be achieved by turning off 60% of the servers?

b. How much power savings would be achieved by placing 60% of the servers in the "barely alive" state?

c. How much power savings would be achieved by reducing the voltage by 20% and frequency by 40%?

d. How much power savings would be achieved by placing 30% of the servers in the "barely alive" state and 30% off?

3a.

Assuming the servers number is N and one server maximum power is P.

*So original power: 0.9NP current power:(1-0.6)N*0.9P*

The saving is : (original power-current power)/original power=0.6=60%

3b.

After replacing,the consumption :0.4NP+0.4(0.2/0.9)NP=0.533NP*

So the saving is (1-0.533)=0.467=46.7%

3c.

*After reducing the power :C*0.8V*0.8V*0.6FCLK*

So the saving is :(1-0.384)=0.616=61.6%

3d.

After achieved,the consumption :0.4NP+0.3(0.2/0.9)NP=0.4667PN*

So the saving is :(1-0.4667)=0.5333=53.33%

4. In a server farm such as that used by Amazon or eBay, a single failure does not cause the entire system to crash. Instead, it will reduce the number of requests that can be satisfied at any one time.

a. If a company has 10,000 computers, each with a MTTF of 35 days, and it experiences catastrophic failure only if 1/3 of the computers fail, what is the

MTTF for the system?

b. If it costs an extra $1000, per computer, to double the MTTF, would this be a good business decision? Show your work.

4a.

$MTTF_{computer}$=35days Failure rate=1/35days

The system experiences catastrophic failure only if 1/3 of the computers fail, so for the system Failure rate=3/35days $MTTF_{system}$=35/3days=11.67days

4b.

If one computer's price is more than $1000,it will be a good decision.Otherwise it's not.

5. In this exercise, assume that we are considering enhancing a machine by adding vector hardware to it. When a computation is run in vector mode on the vector hardware, it is 10 times faster than the normal mode of execution. We call the percentage of time that could be spent using vector mode the percentage of vectorization. Vectors are discussed in Chapter 4, but you don't need to know anything about how they work to answer this question!

a. Draw a graph that plots the speedup as a percentage of the computation performed in vector mode. Label the y-axis "Net speedup" and label the x-axis "Percent vectorization."

b. What percentage of vectorization is needed to achieve a speedup of 2?

c. What percentage of the computation run time is spent in vector mode if a speedup of 2 is achieved?

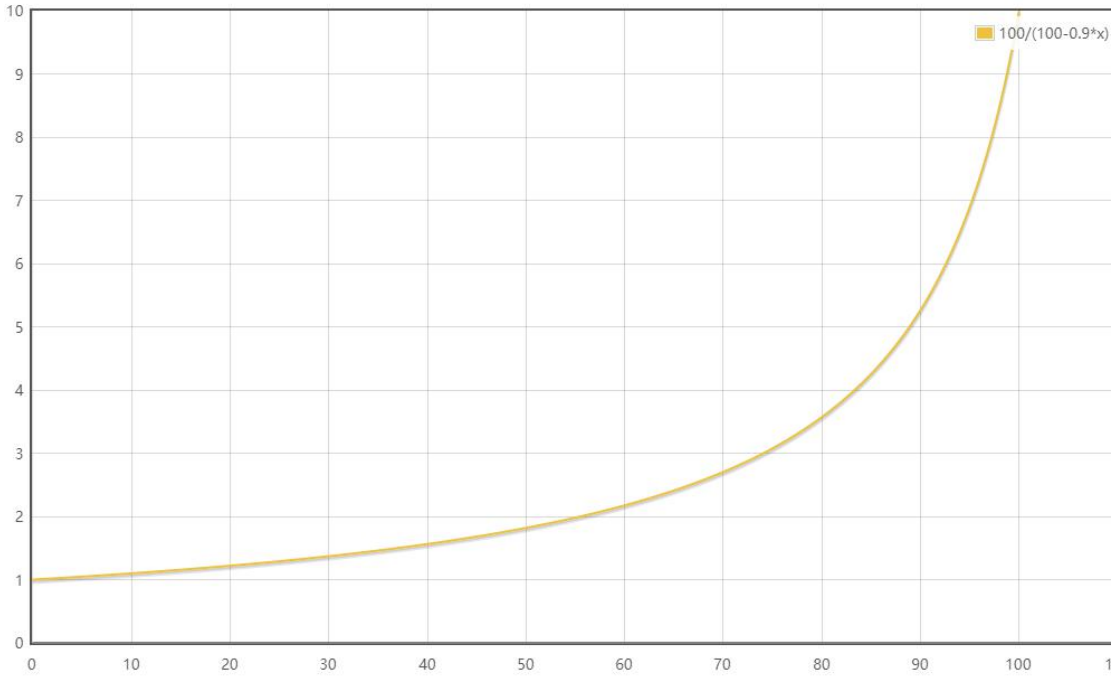
d. What percentage of vectorization is needed to achieve one-half the maximum speedup attainable from using vector mode?

e. Suppose you have measured the percentage of vectorization of the program to be 70%. The hardware design group estimates it can speed up the vector hardware even more with significant additional investment. You wonder whether the compiler crew could increase the percentage of vectorization, instead. What percentage of vectorization would the compiler team need to achieve in order to equal an addition 2× speedup in the vector unit (beyond the initial 10×)?

5a.

Assume T is the time taken for the non vectorized code to run.So the vector one is T/10

*Net speedup y=100T/[(100-x)T+x*T/10],the graph is below:*



5b.

When the speedup is 2 , x=5/9=55.56%

55.56% of vectorization is needed to achieve a speedup of 2.

5c.

(x/10)/[x/10+(100-x)]=11.11%

11.11% of the computation run time is spent in vector mode if a speedup of 2 is achieved.

5d.

*Half of max : 5 5=100T/[(100-x)T+x*T/10] => x=88.88%*

88.88% of vectorization is needed to achieve one-half the maximum speedup attainable from using vector mode.

5e.

When x is 70%, speedup is 2.7. If the aim is to double the speedup(2.7) to 5.4.

The x should be 91%

91% of vectorization would the compiler team need to achieve in order to equal an addition 2× speedup in the vector unit (beyond the initial 10×).

6. (i) A program (or a program task) takes 150 million instructions to execute on a processor running at 2.7 GHz. Suppose that 50% of the instructions execute in 3 clock cycles, 30% execute in 4 clock cycles, and 20% execute in 5 clock cycles. What is the execution time for the program or task?

(ii) Suppose the processor in the previous question part is redesigned so that all instructions that initially executed in 5 cycles and all instructions executed in 4 cycles now execute in 2 cycles. Due to changes in the circuitry, the clock rate also must be decreased from 2.7 GHz to 2.1 GHz. What is the overall percentage improvement?

6(i)

*Time=(IC*CPI)/FCLK=(0.5*150M*3+0.3*150M*4+0.2*150M*5)/2.7GHz=205.56ms*

6(ii)

*Current Time=(IC*CPI)/FCLK=(0.5*150M*3+0.3*150M*2+0.2*150M*2)/2.1GHz=178.57ms*

the overall percentage improvement=(205.56-178.57)/178.57=15.11%

7. Assume for a given processor the CPI of arithmetic instructions is 1, the CPI of load/store instructions is 10, and the CPI of branch instructions is 3. Assume a program has the following instruction breakdowns: 500 million arithmetic instructions, 300 million load/store instructions, 100 million branch instructions.

a. Suppose that new, more powerful arithmetic instructions are added to the instruction set. On average, through the use of these more powerful arithmetic instructions, we can reduce the number of arithmetic instructions needed to execute a program by 25%, while increasing the clock cycle time by only 10%. Is this a good design choice? Why?

b. Suppose that we find a way to double the performance of arithmetic instructions. What is the overall speedup of our machine? What if we find a way to improve the performance of arithmetic instructions by 10 times?

7a.

*Old Time=(1*500M+10*300M+3*300M)*clock cycle time*

*New Time1= (1*500*0.75M+10*300M+3*300M)*1.1clock cycle time*

New Time1>Old Time

it is a bad design choice since the new execution time is larger.

7b.

After double, CPI of arithmetic instructions is 0.5.

*Old Time=(1*500M+10*300M+3*300M)*clock cycle time*

*New Time2=(0.5*500M+10*300M+3*300M)*clock cycle time*

The overall speedup=(Old Time-New Time2)/New Time2=0.07=7%

10 times

CPI of arithmetic instructions is 0.1

*New Time3=(0.1*500M+10*300M+3*300M)*clock cycle time*

The overall speedup=(Old Time-New Time3)/New Time3=0.13=13%

8. After graduating, you are asked to become the lead computer designer at Hyper Computers, Inc. Your study of usage of high-level language constructs suggests that procedure calls are one of the most expensive operations. You have invented a scheme that reduces the loads and stores normally associated with procedure calls and returns. The first thing you do is run some experiments with and without this optimization. Your experiments use the same state-of-the-art optimizing compiler that will be used with either version of the computer. These experiments reveal the following information:

- **The clock rate of the unoptimized version is 5% higher.**
- **30% of the instructions in the unoptimized version are loads or stores.**
- **The optimized version executes 2/3 as many loads and stores as the unoptimized version. For all other instructions the dynamic counts are unchanged.**
- **All instructions (including load and store) take one clock cycle.**

Which is faster? Justify your decision quantitatively.

8.

*Execution Time$_{old}$=IC$_{old}$*CPI*Cycle Time$_{old}$=IC$_{old}$*CPI/Clock Rate$_{old}$*

*Execution Time$_{new}$=IC$_{new}$*CPI*Cycle Time$_{new}$=IC$_{new}$*CPI/Clock Rate$_{new}$*

*IC$_{new}$=0.7IC$_{old}$+2/3IC$_{old}$*0.3=0.9IC$_{old}$*

Clock Rate$_{old}$=1.05Clock Rate$_{new}$

Clock Rate$_{old}$ / Clock Rate$_{new}$=1/0.945=1.055 1.055-1=0.055=5.5%

the optimized version is faster since it will be improved by 5.5%.

9. 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. 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?

b. 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?

c. 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?

d. 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?

e. 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?

f. 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?

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

9a.

SpeedupA=225000/13461=16.7

SpeedupB=28000/36465=7.7

$Time_{TPU}=TimeA_{TPU}+TimeB_{TPU}=0.7*Time_{GPU}/16.7+0.3*Time_{GPU}/7.7$

$Time_{GPU}/Time_{TPU}=1/(0.7/16.7+0.3/7.7)=12.376$

the speedup of the TRU over GPU is almost 12.38.

9b.

*Haswell E5-2699 v3 : 0.7*42%+0.3*100%=0.0594=59.4%*

*NVIDIA K80: 0.7*37% +0.3*100% =0.0559=55.9%*

*TPU: 0.7*80% +0.3*100% =0.086=86%*

9c.

Haswell E5-2699 v3 : 159+0.594(455-159)=334.8W*

*NVIDIA K80: 357+(991-357)*0.559=771.406W*

*TPU: 290+(384-290)*0.86=370.84W*

*Performance:(0.86*771.406)/(370.84*0.559)=2.95*

the performance per watt of TPU system over GPU system is about 2.95

9d.

*$Speedup_{GPU}$=1/(0.4/13461*5482+0.1/36456*13194+0.5/15000*12000)=1.670*

*$SPeedup_{TPU}$=1/(0.4/225000*5482+0.1/28000*13194+0.5/2000*12000)=0.332*

9e.

Number of Haswell E5-2699 v3 :14KW/504=27.78 27 Haswell-based servers

Number of NVIDIA K80 :14kW/1838W = 7.62 7 NVIDIA-based servers

Number of TPU :14kW/861w= 16.26 16 TPU servers

9f.

*Maximum power : 11*200W = 2200W*

Number of Haswell E5-2699 v3 :2200W/504=4.37 4.37 Haswell-based servers

Number of NVIDIA K80 :2200W/1838W = 1.2 1.2 NVIDIA-based servers

Number of TPU :2200W/861w= 2.56 2.56TPU servers

22KW>2200W one cooling door is enough.