Answer 1)
Formula utilized : Time = (Number of Instructions * CPI) / Clock rate
As per the question, O is optimized and UO is unoptimized.

Given -
Clock rate_O = 0.95* Clock rate_UO
NOI_UO = 0.30 (Load and store) NOI_UO + 0.7 (Others instructions)
NOI_O = ⅔ * NOI_UO (Number of instructions of load and store)
CPI=1

Time_O = (NOI_O) * CPI / Clock rate_O
= (⅔ *0.3) + (0.7) NOI_OU * 1 / 0.95* Clock rate_UO
= 0.9 NOI_OU / 0.95* Clock rate_UO

Time OU = (NOI_OU) * CPI / Clock rate_OU
= NOI_OU * 1 / Clock rate_UO

Speedup = Time_OU/Time_O
= (NOI_OU * 1 / Clock rate_UO) / ⅔ * NOI_OU * 1 / 0.95* Clock rate_UO
= 0.95/0.9
= 1.06

The optimized system is faster by 6% than unoptimized version while optimizing the 6%.

Answer 2)

a)
Formula utilized: Speedup of B wrt A = throughput_A / throughput_B
Using table 2,
Speedup of TPU with GPU A = throughput of TPU A/ throughput of GPU A
= 2,25,000/13,461 = 16.715
Speedup of TPU with GPU B = throughput of TPU B/ throughput of GPU B
= 2,80,000/36,465 = 7.678

Total time of TPU over GPU = 0.3 T (A) +0.7 T (B)

As per the given information,
16.715 * Time_taken_TPU_A = 0.7 * T (Time taken by GPU of A)
7.678 * Time_taken_TPU_B = 0.3 * T(Time taken by GPU of B)

Speedup of TPU wrt GPU = Time_taken_GPU / Time_taken_TPU
= T / (0.7/16.715 + 0.3/7.678) T
= 1/ (0.0418 + 0.039)
= 12.37 times

b)
Formula utilized : Performance = clock rate/CPI (Cycles per instruction)
= Max IPS of A * time + Max IPS of B * time
Time spent on A = 0.70
Time spent on B = 0.30

I. General Purpose Performance = 0.42 * 0.70 + 1.00* 0.3 = **0.594 or 59.4%**
II. Graphics Processor Performance (GPU) = 0.37 * 0.70 + 1.00* 0.3 = **0.559 or 55.9%**
III. TPU Performance = 0.80 * 0.70 + 1.00* 0.3 = **0.86 or 86%**

c)
Formula utilized:
Power consumed = IDLE Power + (Busy Power - IDLE Power) * Max IPS of system
Performance per watt = Power consumed * number of instructions with max IPS (n)

PC_GPU = 357 + [(991 - 357) * 0.559]
= 711.406 W

PC_TPU = 290 + [(384-290) * 0.86]
= 370.84W

Performance per watt of TPU system over GPU system
= ((0.86 * n) / 370.84) / ((0.559*n) / 711.406)
= 2.95

d)
Formula utilized : Speedup of B wrt A = throughput_A / throughput_B

Speedup of general processor (GP) over TPU
= 0.4 / (Speedup of TPU/GP of A) + 0.1 / (Speedup of TPU/GP of B) + 0.5 / (Speedup of TPU/GP of C)
= 0.4 / (Throughput of TPU/GP of A) + 0.1 / (Throughput of TPU/GP of B) + 0.5 / (Throughput of TPU/GP of C)
= 0.4 / (225000/5482) + 0.1 / (280000/13194) / + 0.5 / (2000/12000)
= 0.4 / 41.043 + 0.1 / 21.22 + 0.5 / 0.167
= 3.0084

Speedup of TPU over GP becomes 1 / 3.0084 i.e., 0.332 times

Similarly,
Speedup of GPU over GP becomes 1/ X i.e., 1.789 times

X = 0.4 / (Speedup of GPU/GP of A) + 0.1 / (Speedup of GPU /GP of B) + 0.5 / (Speedup of GPU /GP of C)

= 0.4 / (Throughput of GPU /GP of A) + 0.1 / (Throughput of GPU /GP of B) + 0.5 / (Throughput of GPU /GP of C)
= 0.4 / (13461/5482) + 0.1 / (36,465/13194) / + 0.5 / (15,000/12000)
= 0.4 / 2.45 + 0.1 / 2.76 + 0.5 / 1.25
= 0.599

e)
Cooling door = 14000W
GP = 14*10^3W / TDP (504W) **=~ 28 servers** that can be cooled from 1 cooling door.
GPU = 14*10^3W / TDP (1838W) **=~ 8 servers** that can be cooled from 1 cooling door.
TPU = 14*10^3W / TDP (861W) **=~ 16 servers** that can be cooled from 1 cooling door.

GP from Haswell having ~28 servers can be cooled from 1 cooling door.

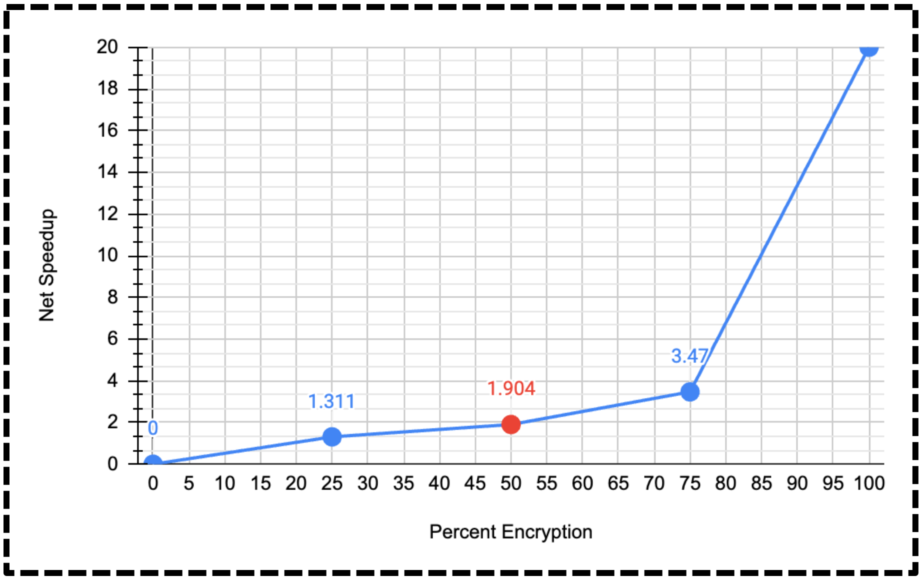
f)
Server Rack = 11 sq feet
Max dissipation = 200 W per sq feet
Maximum power per server rack = max power * length
= 200W / (feet)^2 * 11 (feet)^2 = 2200 W
Therefore,
Max. # of GP Servers = 2200 W / 504 W = ~ **4 servers**
Max. # of GPU Servers = 2200 W / 1838 W = ~ **1 servers**
Max. # of GP Servers = 2200 W / 861 W = ~ **2 servers**

One cooling door (14kW) is enough to dissipate 2200W of energy.
Hence, with a single cooling door following number of racks are required -
GP = 14kW / (504 * 4) **= ~7 GP server racks**
GPU = 14kW / (1838 * 1) **= ~8 GPU server racks**
TPU = 14kW / (861 * 2) **= ~8 TPU server racks**

Answer 3)
a)
Using Amdahl's law -
Formula utilized: Net Speedup = 1 / (1 - PE) + (PE / faster times)

Computations for plotting the graph as Net Speedup being 1 initially =

Speedup of 25% encryption = 1 / (0.75 + 0.25/20) = 1.311
Speedup of 50% encryption = 1 / (0.5 + 0.5/20) = 1.904
Speedup of 75% encryption = 1 / (0.25 + 0.75/20) = 3.47
Speedup of 100% encryption = 1 / (1/20) = 20



b)

Formula utilized: Net Speedup = 1 / (1 - PE) + (PE / faster times)

2 = 1 / (1 - x) + (x/20)

x= 10/19 = **52.6% encryption required for Speedup of 2.**

c)

Formula utilized: Computation run time = encrypted / total time taken

From above, the value (x = speedup = 2) on Y corresponds to 52.6%

Original time taken for non-encrypted machine = (100-52.6) * t / 100 = 0.474 t

When the speedup is achieved, the total time becomes t/2. Hence, the time taken for encrypted machine will be = (t/2) – 0.474 t = 0.5t – 0.474t = 0.026t

Therefore, computation run time = encrypted / total time taken

= 0.026t / 0.5 t

= 0.26 / 5

= 0.052

I.e., 5.2% of computation run time is spent in encryption mode if 2% speedup is achieved.

Answer 4)

a)

Formula utilized: Speedup = 1/ (1 - time) + (time / faster times) [According to Amdahl's law]

Assumption –

Faster times of new = 10 = 1/ faster times of old

Time original (old) is 1 and Time enhanced (new) becomes 0.5

Speed_new/Speed_old = 1/ [(1 - time_new) + (time _new / faster_times_new)] - Equation 1

Speed_old/Speed_new = 1/ [(1 - time _old) + (time _old / faster_times_old)]

= 1/ [(1 - 0.5) + (0.5 * 10)]

= 1/ [(0.5) + (0.5 * 10)]

= 1/5.5

Therefore,

Speed_new/Speed_old = **5.5 is the speedup obtained from fast mode** - Equation 2

b)

Equating both the above equations –

5.5 = 1 / [(1 - time_new) + (time_new / 10)]

(1 - time_new) + (time_new / 10) = 1 / 5.5

1 - time_new + 0.9 time_new = 1 / 5.5

1 - 0.9 time_new = 0.1818

0.9 time_new = 0.818

time_new = 0.909 i.e., 90.9% time of the original execution time has been converted to fast mode

Answer 5)

a)

The speed will increase by the 80% parallelizing.

Formula utilized: Net Speedup = 1 / (1 – speed_new) + (speed_new / faster times) [Using Amdahl's law]

New Speedup with N processors = 1 / (1 - 0.8) + (0.8/x)

= **1 / (0.2) + 0.8 x**

b)

The number of processors = 8 units

Formula utilized: Net Speedup = 1 / [(1 - speed_new) + (speed_new / faster times * no of processors) + ((delay * no of processors)/100)

= 1 / [(1 - 0.8) + (0.8 / 8 processors) + ((0.5 * 8 times) / 100)]

= 1 / (0.2) + (0.1) + (0.04)

= 1 / (0.34)

= **2.94 is the net speedup for 8 times adding each of the processors**

c)

Formula utilized: Net Speedup = 1 / [(1 - speed_new) + (speed_new / faster times * no of processors) + ((delay * no of processors)/100)

The number of processors will be doubled = 1 -> 2 -> 4 -> 8 i.e., 3 times

OR

2 ^ 3 = 8

= 1 / [(1 - 0.8) + (0.8 / 8 processors) + ((0.5 * 3 times) / 100)]

= 1 / (0.2) + (0.1) + (0.0015)

= 1 / (0.3015)

= **3.316 is the net speedup for 3 times doubling the processors**

d)

The number of processors will be doubled by the formula 2^x = N (given N is the number of processors). The final equation becomes x = $\log_2 N$ (as per the above derived equations.

Speedup = 1 / [(1 - 0.8) + (0.8 / N processors) + ((0.5 * $\log_2 N$ times) / 100)]

e)

The original execution time (here PV) is P% hence, the equation becomes

Speedup = 1 / [(1 - P) + (P / N processors) + ((0.5 * $\log_2 N$ times) / 100)]

Further solving the term -

For speedup to be highest, the derivative of N should be 0 (derived from motion equation i.e., when ball tossed in air reaches the max height graph)

D (Speedup) / DN = $[1 / ((1 - P) + (P / N \text{ processors}) + (0.005 * \log_2 N \text{ times}))]^{-2}$ * $[(0.005 /N*\log_e 2) + (P/ N)^2]$ * (-1) = 0

Equating the later term with 0, we get -

N = 200 P $\log_e 2$ as the final equation for getting the highest speedup depending on number of processors available.

Answer 6)

Percentage Parallelizable (PP)

Formula utilized: Net Speedup = 1 / (1 - PP) + (PP / optimization /parallel processors)

a)

Speedup = Speed_serial / Speed_parallel

= 1 / [(1 - 0.5) + (0.5/22 cores)] = 1/ (0.5227) **= 1.913 i.e., 91% speedup**

b)

Similarly, as done in part a,

Speedup = Speed_serial / Speed_parallel

= 1 / ((1 - 0.9) + (0.9/22 cores) = 1/ (0.1409) = 7.096 **i.e., 610% speedup**

c)

41% cores (resources) are allocated to A i.e., 22 * 0.41 = ~9 cores

Speedup of A

= 1 / ((1 - 0.5) + (0.5/9 cores)

= 1.8 i.e., **80% speedup in A**

Overall speedup = 1 / (1- 0.41) + (0.41/1.8)

= 1.22 **i.e., 22% overall speedup of A**

d)

27% cores for B = 22 * 0.27 = ~6 cores

18% cores for C = 22 * 0.18 = ~4 cores

14% cores for D = 22 * 0.14 = ~3 cores

Speedup of A (from c part) = **1.8 times**

Speedup of B = 1 / (1 - 0.8) + (0.8/6 cores) = **3 times**

Speedup of C = 1 / (1 - 0.6) + (0.6/4 cores) = **1.818 times**

Speedup of D = 1 / (1 - 0.9) + (0.9/3 cores) = **2.5 times**

e)

Speedup overall = 1 / (0.41/1.8 + 0.27/3 + 0.18/1.818 + 0.14/2.5)

= 1 / (0.227 + 0.09 + 0.01 + 0.056)

= 1/0.383

= **2.61 % overall speedup** of the resources, considering only active time on their statically assigned cores

Answer 7)

Formula utilized: Net Speedup = 1 / (1 – time_taken_new) + (time_taken_new / faster times)

Here, faster times is 2 (given)

Time is 1.

a)

Time taken new for floating point is 1*20/100 = 0.2

Speedup = 1 / (1 - 0.2) + (0.2/2)

= 1 / (0.8) + 0.1

= 1.11 i.e., 11 % overall speedup for fast floating-point operation

b)

The Speedup would include the cache time of 10% time consumed and ⅔ speedup in addition to floating point.

Time taken new for data cache is 1*20/100 = 0.2

Speedup = 1 / [(1 - 0.2 - 0.1) + (0.2/2) + (0.1*3/2)]

= 1 / [(0.7) + (0.1) + (0.15)]

= 1 / 0.95 **= 1.05 i.e., 5% overall speedup**

c)

Formula utilized: Run time for each = (TT_new / faster times) / (1 - TT_new) + (TT_new / faster times)

Run time for floating point = (0.2/2) / [(1 - 0.2 - 0.1) + (0.2/2) + (0.1*3/2)]

= 0.1 / 0.95

= 0.105 i.e., **10.5% of time**

Run time for data cache access = (0.1*3/2) / [(1 - 0.2 - 0.1) + (0.2/2) + (0.1*3/2)]

= 0.15 / 0.95

= 0.157 i.e., **15.7% of time**

Answer to Question 8 is the same as Question 7.

Answer 8)

Formula utilized: Net Speedup = 1 / (1 – time_taken_new) + (time_taken_new / faster times)

Here, faster times is 2 (given)

Time is 1.

a)

Time taken new for floating point is 1*20/100 = 0.2

Speedup = 1 / (1 - 0.2) + (0.2/2)

= 1 / (0.8) + 0.1

= 1.11 i.e., 11 % overall speedup for fast floating-point operation

b)

The Speedup would include the cache time of 10% time consumed and ⅔ speedup in addition to floating point.

Time taken new for data cache is 1*20/100 = 0.2

Speedup = 1 / [(1 - 0.2 - 0.1) + (0.2/2) + (0.1*3/2)]

= 1 / [(0.7) + (0.1) + (0.15)]

= 1 / 0.95 **= 1.05 i.e., 5% overall speedup**

c)

Formula utilized: Run time for each = (TT_new / faster times) / (1 - TT_new) + (TT_new / faster times)
Run time for floating point = (0.2/2) / [(1 - 0.2 - 0.1) + (0.2/2) + (0.1*3/2)]
= 0.1 / 0.95
= 0.105 i.e., **10.5% of time**

Run time for data cache access = (0.1*3/2) / [(1 - 0.2 - 0.1) + (0.2/2) + (0.1*3/2)]
= 0.15 / 0.95
= 0.157 i.e., **15.7% of time**