Benchmark Score 1/2
Calculation

EEL 3801

Given: Two computer systems: A and B. There is also a reference computer R.

Sought: Use the reference computer R to evaluate the two computer systems A and B under a benchmark suite that has the 3 workloads as listed below.

Workload Program of SPEC suite	Time (R) [seconds]	Time (A) [seconds]	Time (B) [seconds]
CPU	20	4	5
Memory	30	5	6
Graphics	12	3	4

Partial Credit 1:

Which equation best applies for a given workload:

a) SpecRatio(A) = Time(A) / Time(B)
b) SpecRatio(A) = Time(B) / Time(A)
c) SpecRatio(A) = Time(R) / Time(A)
d) SpecRatio(A) = Time(A) / Time(R)
e) SpecRatio(A) = Time(A)+Time(B)/Time(R)
f) SpecRatio(A) = Time(A)*Time(B)/Time(R)
g) SpecRatio(A) = Time(R) / [Time(A)+Time(B)]
h) SpecRatio(A) = Time(B) / [Time(A)+Time(R)]

Partial Credit 2:

What is the Graphics SpecRatio for System B?

Partial Credit 3:

What is the ratio of SPECscores for System A relative to System B?

Partial Credit 4:

Express Part 3 result as a declarative statement to explain the relative performance in a skype teleconference with your client.

Benchmark Score 2/2
Calculation

Solution 1: SpecRatio(A) = Time(R) / Time(A)

[Time R]/[Time A]

Solution 2: SpecRatio(B_Graphics) = Time(R_Graphics) / Time(B_Graphics) = 12/4 = 3 as unitless ratio

Solution 3:

Workload Program of SPEC suite	Time (R) [seconds]	Time (A) [seconds]	SPECRatio(A) [unitless]	Time (B) [seconds]	SPECRatio(B) [unitless]
CPU	20	4	5	5	4
Memory	30	5	6	6	5
Graphics	12	3	4	4	3

SPECscore = $\sqrt[n]{\prod_{i=1}^{n} SPECratio(i)}$

SPECscoreA = $(5*6*4)^{1/3}$ = 4.93

SPECscoreB = $(4*5*3)^{1/3}$ = 3.91

ScoreA / ScoreB = 4.93 / 3.91 = 1.26

Solution 4:

"System A is roughly 5 times faster than the reference system for these 3 benchmark programs."

"System A is roughly 1.26-fold faster than System B for these 3 benchmark programs."

"System A is roughly 26% faster than System B for these 3 benchmark programs."

SPECRatios 1/2
Calculation

<u>Given</u>: A recently-released Dell server became available. We seek to determine its performance rating using the SPEC procedure according to the four benchmarks listed below:

System	**Integer**	**Encryption**	**Graphics**	**Database**
Reference	1 sec	8 sec	0.8 sec	2 sec
Dell server	500 msec	2000 msec	100msec	500msec

<u>Partial Credit 1</u>: What is the SPECRatio of the Dell server for the Integer benchmark?

a) 0.0005
b) 0.5
c) 0.707
d) 2
e) 4.707
f) 8
g) 16
h) 70.700
i) none of the choices listed

<u>Partial Credit 2</u>: What is the SPECscore of the Dell server? Round your answer to the nearest integer.

a) 4
b) 263
c) 66
d) 3
e) 5
f) 3
g) 2
h) 300
i) none of the answers listed

SPECRatios 2/2
Calculation

Solution: SPECRatio(CPU) = 1sec/0.5sec= 2.0

Thus, select choice d) for **Partial Credit 1.**

SPECRatio(Memory) = 8/2 = 4.0

SPECRatio(Graphics) = 800/100 = 8.0

SPECRatio(Floating-Point) = 2/0.5 = 4.0

SPECScore = $(2*4*8*4)^{1/4} = (2^{[1+2+3+2]})^{1/4} = (2^8)^{1/4} = 2^{[8 * 1/4]} = 2^2 = 4.0$

Thus, select choice a) for **Partial Credit 2.**

"Roughly 4-fold speedup over the reference machine for these various workloads."

"Roughly 4 times faster than the reference machine for these various workloads."

Tref/Tnew=4.0, e.g. Tref=4 sec then Tnew=1 sec so 3 seconds was reduced compared to reference. So 3/4 reduction = 75% reduction.

"Execution time reduced by 75% compared to reference machine for these various workloads."

Amdahl's Law
Evaluating Architecture Tradeoffs

Given: The running time of a program on a Dell Server System is comprised entirely of:
- CPU: 6 sec
- Main Memory access: 14 sec
- SSD access: 5 sec

Partial Credit 1: The CTO of the company committed to reduce running time by at least 25%, and she provided you with a budget for exactly one of the following which have the same cost and development effort:

a) make the processor twice as fast, or else
b) make memory access twice as fast.

As project lead, you need to determine if the CTO's performance goal is achievable, and also indicate which approach you choose for your team?

Solution 1: Two ways to answer this question: find time or find speedup:

Exec $\text{Time}_{fasterprocessor}$ = 6/2 + 14 + 5 = 22 sec

Exec $\text{Time}_{fastermemory}$ = 6 + 14/2 + 5 = 18 sec

$\text{Speedup}_{fasterprocessor}$ = 1 / [19/25 + (6/25)/2] = 25/22 = 1.14-fold

$\text{Speedup}_{fastermemory}$ = 1 / [11/25 + (14/25)/2] = 25/18 = 1.39-fold

Better to make memory faster → *select choice b)*

→ *using option b) reduces running time from 6+14+5=25 secs to 18 secs for 25-18=7 seconds reduced, thus 7 sec / 25 sec = 28% reduction.So the CTO's goal of 25% reduction is achievable.*

Partial Credit 2: Which of the choices listed provides the mathematical expression for Speedup:

a) $\log_N(f)$
b) $1/(1-f_{unenhanced})$
c) $1/f_{enhanced}$
d) ExecTime_{new} / ExecTime_{old}
e) ExecTime_{old} / ExecTime_{new}

Solution 2: e) by definition below:

$$\text{Speedup}_{overall} = \frac{\text{Execution time}_{old}}{\text{Execution time}_{new}} = \frac{1}{(1 - \text{Fraction}_{enhanced}) + \dfrac{\text{Fraction}_{enhanced}}{\text{Speedup}_{enhanced}}}$$

A remembrance hint is to recall is that if it runs faster, then speedup should be greater than unity. Thus, being faster means ExecTime_{old} must be in the numerator. The other choices don't relate to speedup.

Amdahl's Law
Evaluating Architecture Tradeoffs

EEL 3801

Given: The running time of a program on a IBM Server System is comprised entirely of:

- *CPU time:* 4 sec
- *Main Memory access time:* 10 sec
- *Disk access time:* 6 sec

The company CTO has provided you with the following approaches:

- *Approach (A):* make the processor twice as fast at a cost of $5
- *Approach (B):* make memory access twice as fast at a cost of $10
- *Approach (C):* make disk access twice as fast at a cost of $2

Partial credit 1: What is the running time of the program if only Approach (A) is applied?

Partial credit 2: What is the running time of the program if both Approach (A) and Approach (B) are applied simultaneously?

Partial credit 3: Which of the following choices provides the lowest cost option to achieve a speedup factor of at least 1.4-fold?

a) Approach A alone
b) Approach B alone
c) Approach C alone
d) Approach A and Approach B when applied together simultaneously
e) Approach B and Approach C when applied together simultaneously
f) Approach A and Approach C when applied together simultaneously
g) Approach A and Approach B and Approach C when applied together simultaneously

Amdahl's Law
Evaluating Architecture Tradeoffs

<u>Part 1</u>: What is the running time of the program if only Approach (A) is applied?

<u>Sol 1</u>: Approach (A): 4/2 + 10 + 6 sec = 2+10+6 sec = 18 sec

<u>Part 2</u>: What is the running time of the program if both Approach (A) and Approach (B) are applied simultaneously?

<u>Sol 2</u>: Approach (AB): 4/2 + 10/2 + 6 sec = 2+5+6 sec = 13 sec

<u>Part 3</u>: Which of the following choices provides the lowest cost option to achieve a speedup factor of at least 1.4-fold?

<u>Sol 3</u>: CTO requirement is a speedup of 40%.
Thus, Speedup = { Time(old) } / { Time(new) } >= 1.4 so finding
Time(old) to be 4+10+6=20 sec in the given statement. Then, 20/Tnew >= 1.4 so:
Tnew = 20 sec / 1.4 <= 14.28 sec.

So let's see which approaches listed achieve 14.28 secs or less, at the lowest cost:

- Approach(C) requires 4+10+6/2=17 sec which does not meet the performance requirement.
- Approach(AB) meets CTO's performance requirement at a cost of $15.
- Approach(BC) meets CTO's performance requirement at a cost of $12.
- Approach(AC) costs just $5+$2=$7 but requires 4/2+10+6/2=15 seconds which does not meet the performance requirement.

→ ***<u>Tip</u>**: skip checking execution time of Approach(ABC) as it costs more than Approach(BC) already found to be sufficient* ☺

Amdahl's Law

Multicore Speedup / Benefit Analysis

EEL 3801

Given: Consider a program that takes 10 seconds to run on dual-core Intel Xeon running at 2GHz under ideal conditions. This program has 30% sequential code and the remainder is infinitely parallelizable.

Sought: What is the minimum time possible to execute this program on an 8-core Intel Xeon CPU running at 2GHz, if all other parameters were to remain unchanged? Round your answer to the nearest second.

Solution: Amdahl's law can be used as the execution time, fraction of parallel code, and number of cores is known. Here, observe the parallel portion is f=0.7.

First compute idealized speedup for the dual-core case that is occurring when f=0.7. That is S=1/((1-f) + f/N) = 1/(0.3+0.7/2) = 1.54 over the uniprocessor case, which is speedup for 2 cores denoted S_2. So running on dual core is 1.54-fold faster than a single core. Since we know the execution time in seconds on the dual core, we can find the uni-core execution time since $S=T_{unenhanced}/T_{enhanced}$. Thus, $1.54=T_1/(10 \text{ seconds})$ so $T_1 = 15.4$ seconds where T_1 represents the unenhanced execution time on 1 processor core.

Thus, if the dual-core case took 10 seconds which represented 1.54 fold speedup over the uniprocessor, then by definition of speedup=Told/Tnew=T_1 /T_N we find the uni-core time to be 15.4 seconds.

Since maximum speedup on 8 cores = S=1/((1-f) + f/N) where f=0.7, but now N=8. So we find next S_8 = 1/(0.3+ 0.7/8)=2.58 then 2.58= T_1 / T_8 implying T_8 = 5.97 seconds which rounds to 6 seconds.

Intuitively, this is consistent that if the program was 70% parallizeable then running on 1 core it took 15 seconds, two cores took 10 seconds, and 8 cores requires 6 seconds.

Amdahl's Law for Speedup

AminoPak: composite use of CPI, clock rate, speedup 1/2

Given: Cellgene Biosciences Inc. is executing the AminoPak protein-folding application and has hired you to increase the performance. They're currently using one AMD single-core processor which has a 2.8 GHz clock rate with CPI=1.0 for all instructions.

The dynamic instruction mix of AminoPak is 80% floating-point instructions, 5% integer instructions, and 15% data transfer instructions. AminoPak is infinitely parallelizable except for the completely sequential **`GetAmino()`** routine (which is 2% of the total AminoPak application time) and also the completely sequential **`RotateBond()`** routine (which is 3% of the total AminoPak application time).

You're considering recommending a single alternative processor chip, as a replacement for the current AMD processor chip.

Design A: The datasheet for a 2.8 GHz Intel Broadwell single-core processor indicates that compared to a 2.8 GHz AMD single-core processor, it will:

- speedup floating-point instructions by 32-fold,
- provide CPI=3.2 for integer instructions while leaving the number of instructions unchanged, and
- double the time taken by each data transfer instruction.

What speedup factor would a 2.8 GHz Intel Broadwell single-core processor provide compared to the 2.8 GHz AMD single-core CPU?

a) 0.05-fold
b) 5%
c) 5-fold
d) 3.2-fold
e) 32%
f) 32-fold
g) 0.846
h) 84.6%
i) 2.06
j) 20%
k) 20-fold
z) none of the choices listed

Solution A: Speedup = 1/(fraction_enhanced/enhancement_factor + fraction_slowed_down*slow_down_factor).
Speedup = 1/(0.8/32 + 0.05*3.2 + 0.15*2) = 2.06. Thus, select choice i).

Amdahl's Law for Speedup

AminoPak: composite use of CPI, clock rate, speedup 2/2

Design B: An AMD quad-core processor has an identical ISA to the AMD single-core processor, but a clock rate of 4.0 GHz and twice the Effective CPI. What best-case speedup factor would it provide for AminoPak compared to the baseline AMD single-core CPU as described above?

a) 0.05-fold
b) 2.48-fold
c) 3.47-fold
d) 34.7%
e) 347%
f) 3470%
g) 5-fold
h) 5.67-fold
i) 0.714
j) 7.14%
k) 71.4-fold
z) insufficient information to determine

Solution B: Without regards to clock rate, Amdahl's law indicates:

Speedup4 = 1 / [(1-f) + (f/4)] where f=parallel fraction.

Here in this question, it is given that f=1-(0.03+0.02)=0.95, so substituting for f:

Speedup4 = 1 / [(1-0.95) + (0.95/4)] = 1 / [0.05 + 0.2375] = 1/0.2875 = 3.47

Meanwhile, the clock rate has also been increased to 4.0 GHz from the previous 2.8 GHz, thus execution is speeded up by [4.0 GHz] / [2.8 GHz] = 1.43 due to the clock rate alone while each instruction requires twice as many clocks, resulting in a relative speed of 1.43/2 = 0.714.

Resulting in a composite impact of:

3.47*0.714 = 2.48-fold speedup. Thus, select choice b).