1st Homework for Computer Architecture

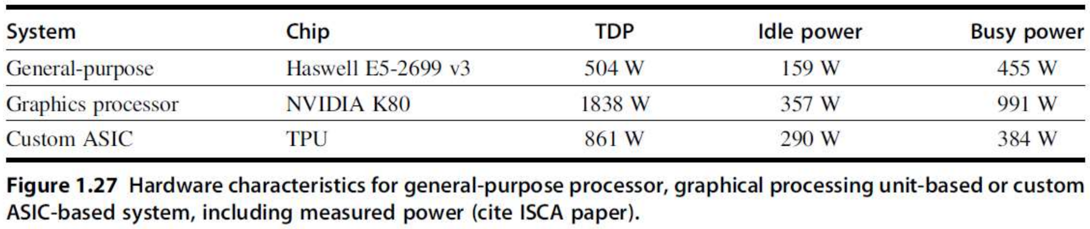
Submission deadline: Oct. 7 , 11：55pm

Read Chapter1 then do the following problems.

(Total 210 points)

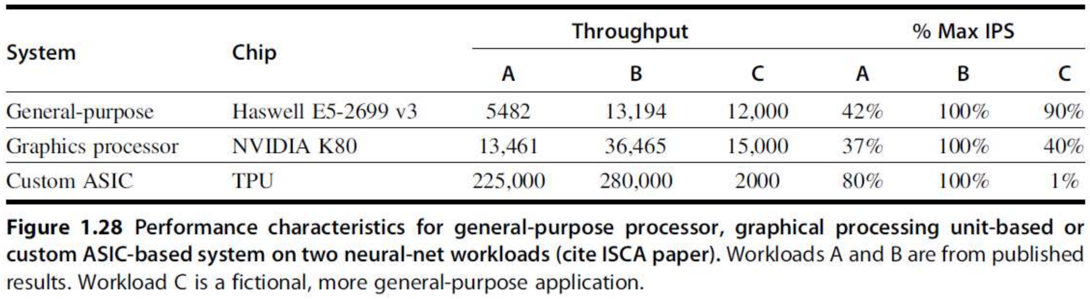
In 6th Edition

1.6 1.8 1.11 1.15



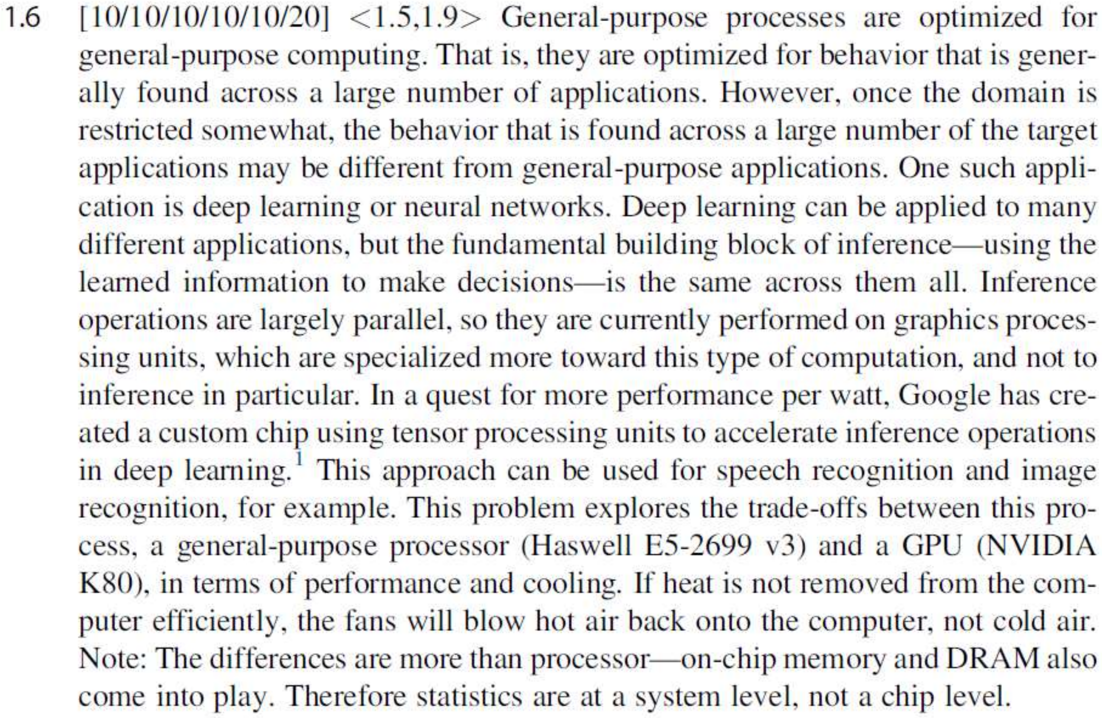
System	Chip	TDP	Idle power	Busy power
General-purpose	Haswell E5-2699 v3	504 W	159 W	455 W
Graphics processor	NVIDIA K80	1838 W	357 W	991 W
Custom ASIC	TPU	861 W	290 W	384 W

Figure 1.27 Hardware characteristics for general-purpose processor, graphical processing unit-based or custom ASIC-based system, including measured power (cite ISCA paper).



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 (cite ISCA paper). Workloads A and B are from published results. Workload C is a fictional, more general-purpose application.



1.6 [10/10/10/10/10/20] <1.5,1.9> 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. [10] <1.9> 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. [10] <1.9> 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?

c. [15] <1.5, 1.9> 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. [10] <1.9> 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?

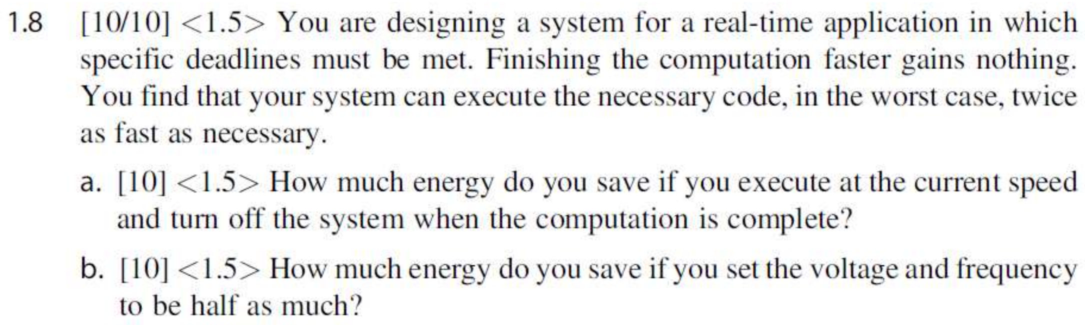
e. [10] <1.5> A cooling door for a rack costs $4000 and dissipates 14 kW (into the room; additional cost is required to get it out of the room). How many Haswell-, NVIDIA-, or Tensor-based servers can you cool with one cooling door, assuming TDP in Figures 1.27 and 1.28?

f. [20] <1.5> 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?

Answer	
a	TPU: (0.7*13461)/225000+(0.3*36465)/280000≈0.081 1/0.081≈12.35 So the speed up is 12.35.
b	General-purpose: 0.7*0.42+0.3*1=0.594 GPU: 0.7*0.37+0.3*1=0.559 TPU: 0.7*0.8+0.3*1=0.86
c	TPU power: 290+(384-290)*0.86=370.84W GPU power: 357+(991-357)*0.559=711.406W TPU performance: (225000*0.7+280000*0.3)/370.84=241500/370.84 GPU performance: (13461*0.7+36465*0.3)/711.406=20362.2/711.406 TPU/GPU -> (241500/370.84)/(20362.2/711.406)≈22.75
d	GPU: (0.4*5482)/13461+(0.1*13194)/36465+(0.5*12000)/15000 ≈ 0.599 TPU: (0.4*5482)/225000+(0.1*13194)/280000+(0.5*12000)/2000 ≈ 3.015 Speed up of GPU/general-purpose = 1/0.599 ≈ 1.67 Speed up of TPU/general-purpose = 1/3.015 ≈ 0.33

e	Haswell-based: 14000/504≈27.8, then 27 servers;
	NVIDIA-based: 14000/1838≈7.6, then 7 servers;
	Tensor-based: 14000/861≈16.3, then 16 servers.

f	200*11=2200W Haswell-based: 2200/504≈4 NVIDIA-based: 2200/1838≈1 Tensor-based: 2200/861≈2 Only 1 cooling door is required.

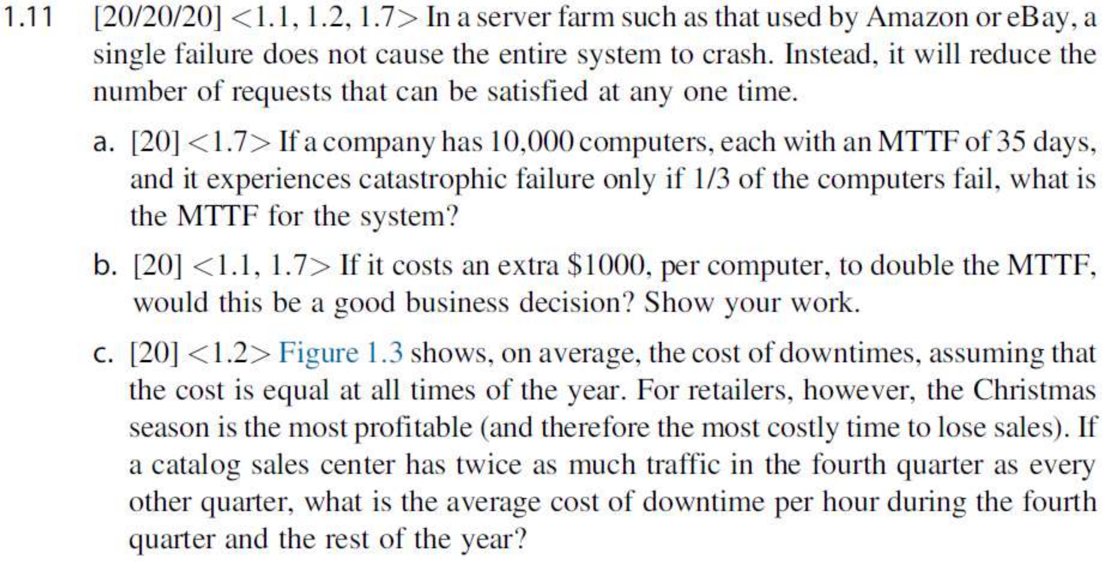


1.8 [10/10] <1.5> You are designing a system for a real-time application in which specific deadlines must be met. Finishing the computation faster gains nothing. You find that your system can execute the necessary code, in the worst case, twice as fast as necessary.

a. [10] <1.5> How much energy do you save if you execute at the current speed and turn off the system when the computation is complete?

b. [10] <1.5> How much energy do you save if you set the voltage and frequency to be half as much?

Answer	
a	Energy_new/Energy_old = (0.5*time_old/time_old)*100%=50% So 50% energy can be saved.
b	Energy_new/Energy_old = ((0.5*voltage_old)^2/voltage_old^2)*100%=25% So 75% energy can be saved.

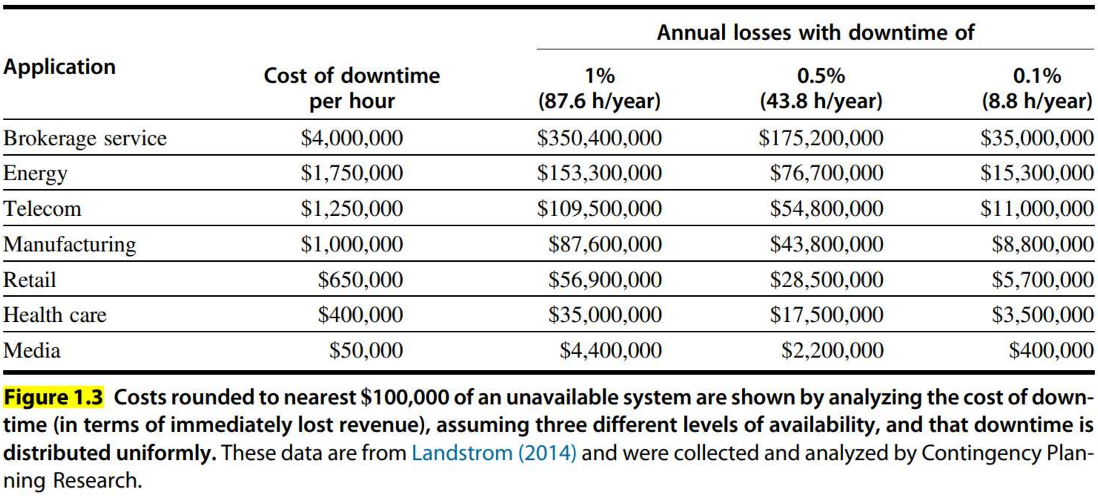


1.11 [20/20/20] <1.1, 1.2, 1.7> In a server farm such as that used by Amazon or eBay, a single failure does not cause the entire system to crash. Instead, it will reduce the number of requests that can be satisfied at any one time.

a. [20] <1.7> If a company has 10,000 computers, each with an MTTF of 35 days, and it experiences catastrophic failure only if 1/3 of the computers fail, what is the MTTF for the system?

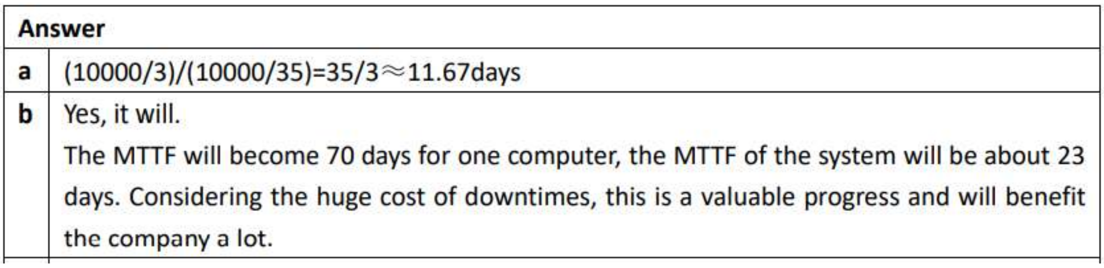
b. [20] <1.1, 1.7> If it costs an extra $1000, per computer, to double the MTTF, would this be a good business decision? Show your work.

c. [20] <1.2> Figure 1.3 shows, on average, the cost of downtimes, assuming that the cost is equal at all times of the year. For retailers, however, the Christmas season is the most profitable (and therefore the most costly time to lose sales). If a catalog sales center has twice as much traffic in the fourth quarter as every other quarter, what is the average cost of downtime per hour during the fourth quarter and the rest of the year?



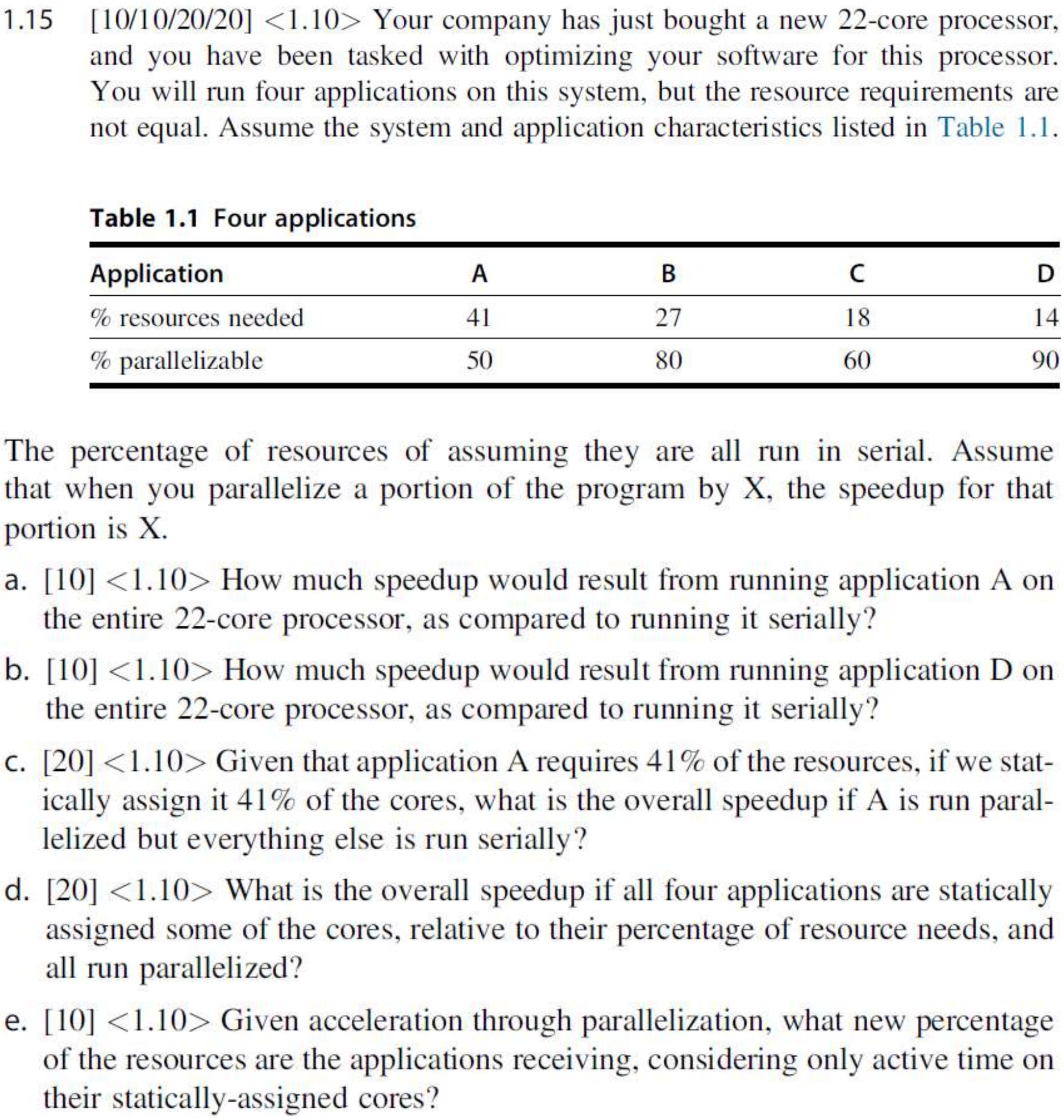
Application	Cost of downtime per hour	Annual losses with downtime of 1% (87.6 h/year)	0.5% (43.8 h/year)	0.1% (8.8 h/year)
Brokerage service	$4,000,000	$350,400,000	$175,200,000	$35,000,000
Energy	$1,750,000	$153,300,000	$76,700,000	$15,300,000
Telecom	$1,250,000	$109,500,000	$54,800,000	$11,000,000
Manufacturing	$1,000,000	$87,600,000	$43,800,000	$8,800,000
Retail	$650,000	$56,900,000	$28,500,000	$5,700,000
Health care	$400,000	$35,000,000	$17,500,000	$3,500,000
Media	$50,000	$4,400,000	$2,200,000	$400,000

Figure 1.3 Costs rounded to nearest $100,000 of an unavailable system are shown by analyzing the cost of downtime (in terms of immediately lost revenue), assuming three different levels of availability, and that downtime is distributed uniformly. These data are from Landstrom (2014) and were collected and analyzed by Contingency Planning Research.



Answer	
a	(10000/3)/(10000/35)=35/3≈11.67days
b	Yes, it will. The MTTF will become 70 days for one computer, the MTTF of the system will be about 23 days. Considering the huge cost of downtimes, this is a valuable progress and will benefit the company a lot.

c	650000 = $\frac{3}{4}x + \frac{1}{4} \cdot 2x$ 2x = 1040000 $/h for 4th quarter x = 520000 $/h for others



1.15 [10/10/20/20] <1.10> Your company has just bought a new 22-core processor, and you have been tasked with optimizing your software for this processor. You will run four applications on this system, but the resource requirements are not equal. Assume the system and application characteristics listed in Table 1.1.

Table 1.1 Four applications

Application	A	B	C	D
% resources needed	41	27	18	14
% parallelizable	50	80	60	90

The percentage of resources of assuming they are all run in serial. Assume that when you parallelize a portion of the program by X, the speedup for that portion is X.

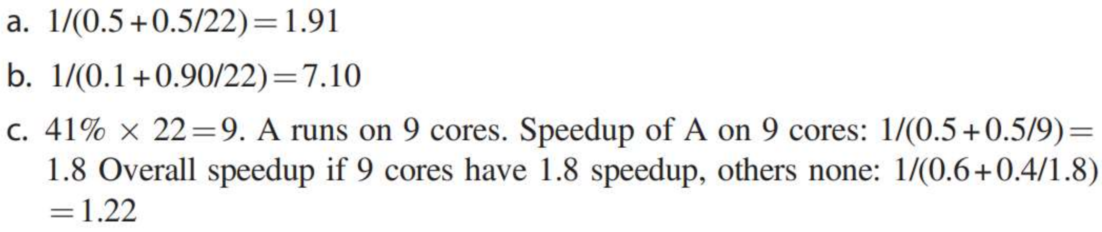
a. [10] <1.10> How much speedup would result from running application A on the entire 22-core processor, as compared to running it serially?

b. [10] <1.10> How much speedup would result from running application D on the entire 22-core processor, as compared to running it serially?

c. [20] <1.10> Given that application A requires 41% of the resources, if we statically assign it 41% of the cores, what is the overall speedup if A is run parallelized but everything else is run serially?

d. [20] <1.10> What is the overall speedup if all four applications are statically assigned some of the cores, relative to their percentage of resource needs, and all run parallelized?

e. [10] <1.10> Given acceleration through parallelization, what new percentage of the resources are the applications receiving, considering only active time on their statically-assigned cores?

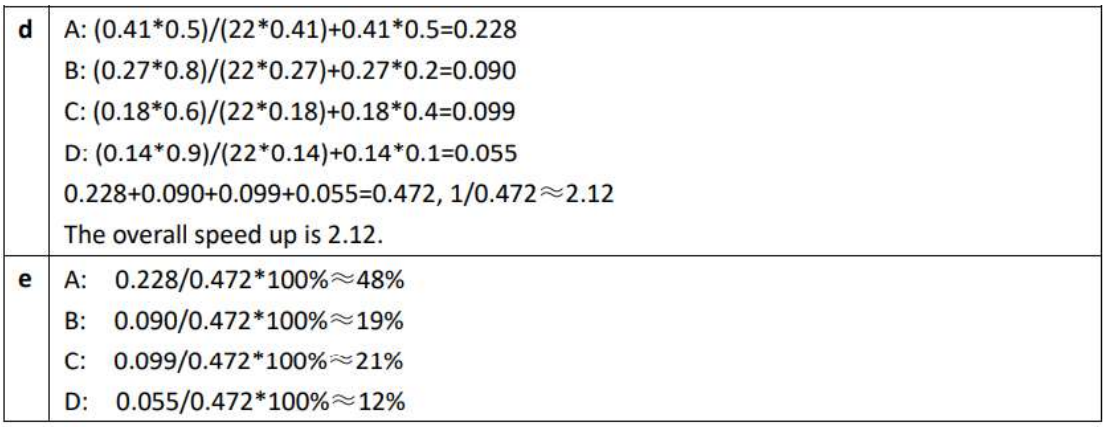


a. $1/(0.5+0.5/22)=1.91$

b. $1/(0.1+0.90/22)=7.10$

c. 41% × $22=9$. A runs on 9 cores. Speedup of A on 9 cores: $1/(0.5+0.5/9)=1.8$ Overall speedup if 9 cores have 1.8 speedup, others none: $1/(0.6+0.4/1.8)=1.22$

(注意，中间结果不要近似，只在最终结果近似**)**



d	A: (0.41*0.5)/(22*0.41)+0.41*0.5=0.228 B: (0.27*0.8)/(22*0.27)+0.27*0.2=0.090 C: (0.18*0.6)/(22*0.18)+0.18*0.4=0.099 D: (0.14*0.9)/(22*0.14)+0.14*0.1=0.055 0.228+0.090+0.099+0.055=0.472, 1/0.472≈2.12 The overall speed up is 2.12.
e	A: 0.228/0.472*100%≈48% B: 0.090/0.472*100%≈19% C: 0.099/0.472*100%≈21% D: 0.055/0.472*100%≈12%