Universidade de Aveiro

Mestrado em Engenharia de Computadores e Telemática

Arquitecturas de Alto Desempenho

Performance Evaluation

Academic year 2022/2023 Exercises adapted from *Computer Architecture: a Quantitative Approach*

1. Your company is trying to decide between purchasing Opteron or Itanium 2 processors for it new computer system. You have analyzed the applications to be run there and you have concluded that 60% of time the system will be running applications similar to `wupwise`, 20% of the time applications similar to `ammp` and 20% of the time applications similar to `apsi`.

SPECfp2000 execution times (s) for Sun Ultra 5 (reference computer) and execution times and SPECRatios for AMD Opteron and Intel Itanium 2

Benchmark	Ultra 5 exec time (s)	Opteron exec time (s)	SPECRatio	Itanium 2 exec time (s)	SPECRatio	Opteron/Itanium 2 exec time	Itanium 2/Opteron SPECRatio
wupwise	1 600	51,5	31,06	56,1	28,53	0,92	0,92
ammp	2 200	136,0	16,14	132,0	16,63	1,03	1,03
apsi	2 600	150,0	17,36	231,0	11,27	0,65	0,65
...							
geo mean			20,86		27,12	1,30	1,30

1.1. If your selection criterion was just based on overall SPEC performance for SPECfp2000, which processor would you choose and why?

$$geo\,mean(\text{Itanium 2/Opteron SPECRatio}) = \sqrt[N]{\prod_i \frac{SPECRatio_{Itanium\,2}(i)}{SPECRatio_{Opteron}(i)}} =$$

$$= \sqrt[N]{\prod_i \frac{execTime_{Opteron}(i)}{execTime_{Itanium\,2}(i)}} = geo\,mean(\text{Opteron/Itanium 2 execTime}) \ .$$

I would choose Itanium 2, because it has a better SPEC performance for SPECfp2000 as a whole.

1.2. What is the weighted average of execution time ratios for this particular mix of applications in the Opteron and in the Itanium 2?

$$weighted\ geo\,mean\,mix(\text{Opteron SPECRatio}) = SPECRatio_{\text{wupwise}}^{w_{\text{wupwise}}} \cdot SPECRatio_{\text{ammp}}^{w_{\text{ammp}}} \cdot SPECRatio_{\text{apsi}}^{w_{\text{apsi}}} =$$

$$= 31{,}06^{0.6} \cdot 16{,}14^{0{,}2} \cdot 17{,}32^{0{,}2} = 24{,}26$$

$$weighted\ geo\,mean\,mix(\text{Itanium 2 SPECRatio}) = SPECRatio_{\text{wupwise}}^{w_{\text{wupwise}}} \cdot SPECRatio_{\text{ammp}}^{w_{\text{ammp}}} \cdot SPECRatio_{\text{apsi}}^{w_{\text{apsi}}} =$$

$$= 28{,}53^{0.6} \cdot 16{,}63^{0{,}2} \cdot 11{,}27^{0{,}2} = 21{,}27 \ .$$

1.3. What is the speed up of the Opteron over the Itanium 2?

$$speed\,up(\text{Opteron/Itanium 2}) = \frac{weighted\ geo\,mean\,mix(\text{Opteron SPECRatio})}{weighted\ geo\,mean\,mix(\text{Itanium 2 SPECRatio})} =$$

$$= \frac{24{,}26}{21{,}27} = 1{,}14 \ .$$

2. The company you work for has bought a IntelCore i5 dual-core processor and your boss has challenged you to optimize the code for the processor. You are supposed to run two independent applications on it whose resource requirements are unequal. The first application requires 80% of the resources of the computer system, while the second only requires 20%. Assume that when you parallelize a portion of the code, the speed up for that portion is 2.

   Since the totality of resources of the computer system required by the two applications is 100% and they are supposed to be independent, they may run in parallel.

$$speed\ up = \frac{t_1 + t_2}{max(t_1, t_2)} .$$

2.1. Given that 40% of the first application is parallelizable, how much speed up would this application observe if run in isolation?

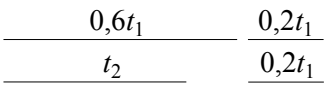
Amdahl's Law $$speed\ up = \frac{1}{0{,}6 + 0{,}4 / 2} = \frac{1}{0{,}8} = 1{,}25 .$$

2.2. Given that 99% of the second application is parallelizable, how much speed up would this application observe if run in isolation?

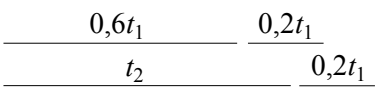
Amdahl's Law $$speed\ up = \frac{1}{0{,}01 + 0{,}99 / 2} = \frac{1}{0{,}505} = 1{,}98 .$$

2.3. Given that 40% of the first application is parallelizable, how much overall speed up would you observe if you parallelized it and run it together with the second application?

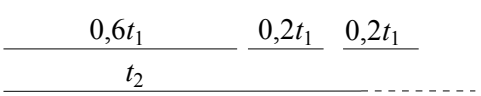
Amdahl's Law is not applicable.



$$t_2 \le 0{,}6\,t_1 \Rightarrow speed\ up = \frac{t_1 + t_2}{0{,}8\,t_1}$$



$$0{,}6\,t_1 < t_2 \le 0{,}8\,t_1 \Rightarrow speed\ up = \frac{t_1 + t_2}{0{,}2\,t_1 + t_2}$$



$$t_2 > 0{,}8\,t_1 \Rightarrow speed\ up = \frac{t_1 + t_2}{max(t_1, t_2)} .$$

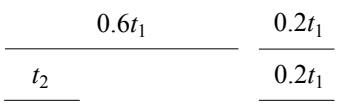
2.4. Given that 99% of the second application is parallelizable, how much overall speed up would you observe if you parallelized it and run it together with the first application?

$$t_1 \le 0{,}01\,t_2 \Rightarrow speed\ up = \frac{t_1 + t_2}{0{,}505\,t_2}$$

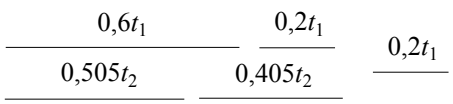
$$0{,}01\,t_2 < t_1 \le 0{,}505\,t_2 \Rightarrow speed\ up = \frac{t_1 + t_2}{t_1 + 0{,}495\,t_2}$$

$$t_1 > 0{,}505\,t_2 \Rightarrow speed\ up = \frac{t_1 + t_2}{max(t_1, t_2)} .$$

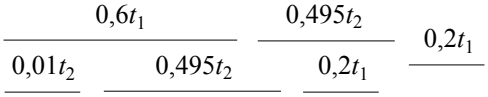
2.5. Given that 40% of the first application and 99% of the second application are parallelizable, how much overall speed up would you observe if you parallelized both of them and run them together?



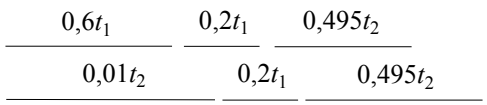
$$t_2 \le 0{,}6\,t_1 \quad \Rightarrow \quad speed\ up \ = \ \frac{t_1 + t_2}{0{,}8\,t_1}$$



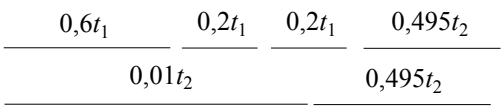
$$0{,}6\,t_1 < t_2 \le \frac{0{,}6}{0{,}505}\,t_1 \quad \Rightarrow \quad speed\ up \ = \ \frac{t_1 + t_2}{min(0{,}8\,t_1, t_2) + 0{,}2\,t_1}$$



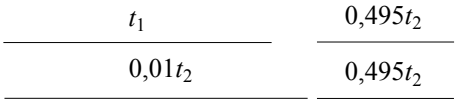
$$\frac{0{,}6}{0{,}505}\,t_1 < t_2 \le \frac{0{,}6}{0{,}01}\,t_1 \quad \Rightarrow \quad speed\ up \ = \ \frac{t_1 + t_2}{min(0{,}2\,t_1 + 0{,}505\,t_2, 0{,}6\,t_1 + 0{,}495\,t_2) + 0{,}2\,t_1}$$



$$\frac{0{,}6}{0{,}01}\,t_1 < t_2 \le \frac{0{,}8}{0{,}01}\,t_1 \quad \Rightarrow \quad speed\ up \ = \ \frac{t_1 + t_2}{min(0{,}8\,t_1 + 0{,}495\,t_2, 0{,}2\,t_1 + 0{,}01\,t_2) + 0{,}495\,t_2}$$



$$\frac{0{,}8}{0{,}01}\,t_1 < t_2 \le \frac{t_1}{0{,}01} \quad \Rightarrow \quad speed\ up \ = \ \frac{t_1 + t_2}{min(t_1, 0{,}505\,t_2) + 0{,}495\,t_2}$$



$$t_1 < 0{,}01\,t_2 \quad \Rightarrow \quad speed\ up \ = \ \frac{t_1 + t_2}{0{,}505\,t_2} \ .$$

3. When parallelizing an application, the maximum speed up that can be attained is equal to the number of processors running the application. This value, however, is limited by two factors: the amount of the application that can be run concurrently and the communication cost among the processors. Amdahl's law takes into account the former, but not the latter.

   3.1. What is the speed up with N processors if as much as 80% of the application is concurrent and the communication cost is ignored?

   $$speed\ up = \frac{1}{0{,}2 + \frac{0{,}8}{N}} .$$

   3.2. What is the speed up with 8 processors if, for every processor added, the communication overhead is increased by 0.5% of the original execution time?

   Assuming that the communication overhead can not be parallelizable, one gets

   $$speed\ up = \frac{t}{\left(0{,}2 + 0{,}005 * 7 + \frac{0{,}8}{8}\right) \cdot t} = \frac{1}{0{,}2 + 0{,}035 + \frac{0{,}8}{8}} = 2{,}99 .$$

   3.3. What is the speed up with 8 processors if, every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

   $$speed\ up = \frac{t}{\left(0{,}2 + 0{,}005 * 3 + \frac{0{,}8}{8}\right) \cdot t} = \frac{1}{0{,}2 + 0{,}015 + \frac{0{,}8}{8}} = 3{,}17 .$$

   3.4. What is the speed up with N processors if, every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

   Assuming that N is a power of 2, $N = 2^k$.

   $$speed\ up = \frac{1}{0{,}2 + 0{,}005 * k + \frac{0{,}8}{N}} .$$

   3.5. Write the general equation that solves the following problem: What is the number of processors that gives rise to the highest speed up for the execution of an application in which P% of the original execution time is concurrent and, every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

   Assuming that N is a power of 2, $N = 2^k$.

   $$speed\ up = \frac{1}{1 - P + 0{,}005 * k + \frac{P}{N}} .$$

   The table presents the speed up results when $P = 0.8$ for different numbers of processors

K	N	speed up
0	1	1,00
1	2	1,65
2	4	2,44
3	8	3,17
4	16	3,70
5	32	4,00
6	64	4,12
7	128	4,15
8	256	4,11
9	512	4,06
10	1024	3,99

As it can be seen, speed up attains a maximum near $k = 7$. The way to determine it without trial and error is to differentiate the speed up formula in order to k, equate it zero and then solve the resulting equation in order to k (bear in mind that it is an nonlinear equation that can only be solved numerically)

$$\frac{d}{dk}\, speed\, up(k) = 0 \quad \Rightarrow \quad \frac{d}{dk}\left(\frac{1}{1 - P + 0{,}005 * k + \frac{P}{2^k}}\right) = 0 \quad \Rightarrow$$

$$\Rightarrow \; -\left(\frac{1}{1 - P + 0{,}005 * k + \frac{P}{2^k}}\right)^2 \cdot \left[0{,}005 - \frac{P \log 2}{2^k}\right] = 0 \quad .$$

4. In a warehouse-scale computer system such as that used by Amazon or eBay, a single failure in a node computer does not cause the entire system to crash. Instead, the total number of requests that can be served at any one time will be just reduced.

Unavailability costs for some businesses in a warehouse-scale computer, assuming that downtime is distributed uniformly

Business	Downtime cost per hour	Annual losses with downtime of		
		1% (87.6 hrs/yr)	0.5% (43.8 hrs/yr)	0.1% (8.8 hrs/yr)
brokerage operations	$ 6 450 000	$ 565 000 000	$283 000 000	$ 56 500 000
home shopping channel	$113 000	$ 9 900 000	$ 4 900 000	$ 1 000 000
catalog sales center	$ 90 000	$ 7 900 000	$ 3 900 000	$ 800 000
airline reservation center	$ 89 000	$ 7 900 000	$ 3 900 000	$ 800 000

4.1. If a company has 10 000 computers, forming a warehouse-scale computer system, each with a MTTF of 35 days and a MTTR of 1 day, and if it only experiences catastrophic failure when at least 1/3 of its computer nodes fail, what is the MTTF of the warehouse-scale computer?

A catastrophic failure would entail the joint failure of 3 334 computer systems, so one may say that the warehouse-scale computer has a redundancy of 3 334.
The table bellow presents some results about the reliability of aggregates of these computer systems for different levels of redundancy.

$$\text{MTTF}_{\text{mod - redund K}} = \frac{\text{MTTR}_{\text{mod}} \cdot \textit{availability}_{\text{redund K}}}{(1 - \textit{availability}_{\text{redund K}})}$$

$$\textit{availability}_{\text{redund } K} = 1 - \prod_{k=1}^{K} (1 - \textit{availability}_{k \text{ mod}})$$

$$\textit{availability}_{\text{k mod}} = \frac{\frac{\text{MTTF}_{\text{mod}}}{k}}{\frac{\text{MTTF}_{\text{mod}}}{k} + \text{MTTR}_{\text{mod}}}$$

	1 comp sys	2 comp sys	3 comp sys	10 comp sys	20 comp sys	25 comp sys
availability	0,972222	0,998498	0,999881	1,000000	1,000000	1,000000
MTTF	3,50000E+01	6,65000E+02	8,43500E+03	3,19019E+09	5,05040E+14	infinity

infinity means that the availability is closer to 1 than 1e-17

As it can be seen, the MTTF of redundancy 20 is already larger than the age of the universe. So the MTTF of redundancy 3 334 is so large that for all practical purposes may be considered infinite.

4.2. Suppose it costs an extra $200 per computer node to double its MTTF. Would it be a sound business decision to do this? Present your reasoning.

It is pointless to spend money to improve the MTTF of the individual computer systems, since the warehouse-scale computer will never suffer a catastrophic failure of this kind.

4.3. The table above illustrates the downtime cost for some businesses assuming that the cost is invariant throughout the year. For retailers, however, the Christmas season is the most profitable (and therefore is the most costly time to loose sales). If a catalog sales center has

twice as much traffic in the fourth quarter as in any other quarter, what is the average cost of downtime per hour during both the fourth quarter and the rest of the year?

Assuming that traffic is directly proportional to sales revenues, one may say that the average cost of downtime per hour is about $144 000 in the fourth quarter and about $72 000 in the rest of the year.