CSCI 510: Computer Architecture Written Assignment 1 Solutions

- 1. Your colleague at a CPU manufacture company suggests that, since the yield is so poor, you might make chips more cheaply if you placed an extra core on the die and only threw out chips on which both cores had failed. We will solve this exercise by viewing the yield as a probability of no defects occurring in a certain area given the defect rate of 0.6 per cm². The die size is 200 mm². Calculate probabilities based on each core separately (this may not be entirely accurate, since the yield equation is based on empirical evidence rather than a mathematical calculation relating the probabilities of finding errors in different portions of the chip). Assume the process-complexity factor is 11.
 - a) (7 pts) What is the probability that a defect will occur on no more than one of the two processor cores?
 - b) (7 pts) If the old chip cost \$18 dollars per chip, what will be the cost of the new chip, taking into account the new area and yield?

Answer:

- a) Since we are considering individual dies, wafer yield is not a factor. $100 \text{ mm}^2 = 1 \text{ cm}^2 \Rightarrow 200 \text{ mm}^2 = 2 \text{ cm}^2$ Probability of a die having no defect = die yield = p = 1 / (1 + 0.6 × 2)¹¹ = 2.2⁻¹¹ \approx 0.00017 Probability that both dies have defects is $(1 - p)^2$. Probability that no more than one die has defects is $q = 1 - (1 - p)^2 = 1 - (1 - 0.00017)^2 \approx 0.00034$
- b) Probability that both dies have defects is $(1 p)^2$.

Let wafer cost be W, dies per water be N, of which $p \cdot N$ have no defect.

 $18 = W / (p \cdot N)$

The amount of double-die chips made from one wafer is N/2.

The total amount of good double-die chips made from one wafer is q $\,\cdot\,N/2$

Let the cost per double-die chip with two dies be x.

The total amount of good double-die chips made from one wafer is

$$x \cdot q \cdot N/2 = W = 18 \cdot p \cdot N$$

=> $x = 18 \cdot 2 \cdot p / q = 36 \times 0.00017 / 0.00034 = 18$

2. One critical factor in powering a server farm is cooling. If heat is not removed from the computer efficiently, the fans will blow hot air back onto the computer, not cold air. We will look at how different design decisions affect the necessary cooling, and thus the price, of a system. Use the following table for your power calculations.

Component Type	Product	Performance	Power
Processor	Intel Pentium 4	2 GHz	48.9–66 W
Hard drive	DiamondMax 9	7200 rpm	7.9 W read/seek, 4.0 W idle

DRAM Kingston X64C3AD2 1 GB	184-pin	3.7 W
-----------------------------	---------	-------

- a) (5 pts) 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 servers with an Intel Pentium 4 processor, 1 GB 184-pin DRAM, and a single 7200 rpm hard drive can you cool with one cooling door?
- b) (5 pts) You are considering providing fault tolerance for your hard drive. RAID 1 doubles the number of disks (see Chapter 6). Now how many systems can you place on a single rack with a single cooler?
- c) (5 pts) 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 (a) can be placed on a single rack, and how many cooling doors are required?

Answer:

- a) The worst case (i.e. the peak power consumption) should be accommodated. The peak power consumption per system is 66 + 7.9 + 3.7 = 77.6 W The amount of systems that can be cooled = $\lfloor 14000 / 77.6 \rfloor = 180$ ($\lfloor x \rfloor$ = the largest integer no more than x)
- b) The peak power consumption per system is $66 + 2 \times 7.9 + 3.7 = 85.5$ W The amount of systems that can be cooled = $\lfloor 14000 / 85.5 \rfloor = 163$
- c) The heat that can be dissipated from a rack is $200 \times 11 = 2200$ W. The amount of systems that can be cooled = $\lfloor 2200 / 77.6 \rfloor = 28$ Since 2200 W < 14000W, only one cooling door is required.
- 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) (5 pts) How much power savings would be achieved by turning off 40% of the servers? Assume the rest 60% have to operate at 100% capacity to process the requests.
 - b) (5 pts) How much power savings would be achieved by placing 40% of the servers in the "barely alive" state? Assume the rest 60% still operate at 60% capacity as the "barely alive" servers can quickly restart.
 - c) (5 pts) How much power savings would be achieved by reducing the voltage by 20% and frequency by 40% while keeping all servers operate at 60% capacity?
 - d) (5 pts) How much power savings would be achieved by placing 20% of the servers in the "barely alive" state and 20% off? Assume the rest 60% have to operate at 80% capacity and consume 97% of maximum power.

Answer:

a) Assume the maximum power consumption is P, and the amount of servers be N. In the old system, the power consumption is 0.9NP.

In the new system, the power consumption is 0.6N 1P = 0.6NP Power saving = $(0.9NP - 0.6NP) / 0.9NP = 1/3 \approx 33.3\%$

b) In the new system, the power consumption is 0.6N 0.9P + 0.4N 0.2P = 0.62NP Power saving = $(0.9NP - 0.62NP) / 0.9NP \approx 0.311 = 31.1\%$

c)

$$\frac{\text{new power}}{\text{old power}} = \frac{\frac{1}{2} \times \text{capacitance} \times (\text{voltage} \times 0.8)^2 \times (\text{frequence} \times 0.6)}{\frac{1}{2} \times \text{capacitance} \times \text{voltage}^2 \times \text{frequence}} = 0.384$$

The saving is 1 – 0.384 = 0.616 = 61.6%

- d) In the new system, the power consumption is 0.2N 0.2P + 0.6N 0.97P = 0.622NP Power saving = $(0.9NP - 0.622NP) / 0.9NP \approx 0.309 = 30.9\%$
- 4. (7 pts) 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. If a company has 10,000 computers, each with a MTTF of 40 days, and it experiences catastrophic failure only if 1/4 of the computers fail, what is the MTTF for the system?

Answer:

According to the way MTTF of individual computers is measured,

MTTF = total # hours all computers work / # computers failed

Let x be the average amount of time the system runs until 1/4 of all computers fail.

 $40 = x \ 10000 \ / \ (10000 \ \cdot \ \frac{1}{4})$

 $= x = 40 / 4 \approx 10 \text{ days}$

- 5. Your company is trying to choose between purchasing the Opteron or Itanium 2. You have analyzed your company's applications, and 60% of the time it will be running applications similar to wupwise, 10% of the time applications similar to ammp, and 30% of the time applications similar to apsi. Refer to Figure 1.17 in the textbook for SPEC performance measurement.
 - a) (5 pts) If you were choosing just based on overall SPEC performance, which would you choose and why?
 - b) (5 pts) What is the weighted average of execution time ratios for this mix of applications for the Opteron and Itanium 2?
 - c) (5 pts) What is the speedup of the Opteron over the Itanium 2?

Answer:

- a) Since the SPECRatio of Opteron (20.86) is less than that of Itanium 2 (27.12), the choice would be Itanium 2.
- b) The weighted execution time ratio of Opteron to Itanium 2 is $0.6 \times (51.5 / 56.1) + 0.1 \times (136 / 132) + 0.3 \times (150 / 231) \approx 0.649$
- c) For Opteron, the weighted averaged execution time is $0.6 \times 51.5 + 0.1 \times 136 + 0.3 \times 150 = 89.5$ seconds For Itanium 2, the weighted averaged execution time is $0.6 \times 56.1 + 0.1 \times 132 + 0.3 \times 231 = 116.16$ seconds The speed up of Opteron over Itanium 2 is 116.16 / 89.5 \approx 1.30
- 6. When making changes to optimize part of a processor, it is often the case that speeding up one type of instruction comes at the cost of slowing down something else. For example, if we put in a complicated fast floating point unit, that takes space, and something might have to be moved farther away from the middle to accommodate it, adding an extra cycle in delay to reach that unit. The basic Amdahl's law equation does not take into account this trade-off.
 - a) (5 pts) If the new fast floating-point unit speeds up floating-point operations by, on average, 2×, and floating-point operations take 30% of the original program's execution time, what is the overall speedup (ignoring the penalty to any other instructions)?
 - b) (5 pts) Now assume that speeding up the floating-point unit slowed down data cache accesses, resulting in a 1.5× slowdown (or 2/3 speedup). Data cache accesses consume 10% of the execution time. What is the overall speedup now?
 - c) (5 pts) After implementing the new floating-point operations, what percentage of execution time is spent on floating-point operations? What percentage is spent on data cache accesses?

Answer:

- a) The overall speedup = 1 / (0.7 + 0.3 / 2) = 1/0.85 ≈ 1.18
- b) The overall speedup = 1 / (0.6 + 0.3 / 2 + 0.1 \times 1.5) = 1/0.9 \approx 1.11
- c) The percentage of execution time spent on floating-point operations is $(0.3 / 2) / (0.6 + 0.3 / 2 + 0.1 \times 1.5) \approx 0.167 = 16.7\%$ The percentage spent on data cache accesses is $(0.1 \times 1.5) / (0.6 + 0.3 / 2 + 0.1 \times 1.5) \approx 0.167 = 16.7\%$
- 7. When parallelizing an application, the ideal speedup is speeding up by the number of processors. This is limited by two things: percentage of the application that can be parallelized and the cost of communication. Amdahl's law takes into account the former but not the latter.
 - a) (5 pts) What is the speedup with N processors if 80% of the application is parallelizable, ignoring the cost of communication?
 - b) (5 pts) What is the speedup with 8 processors if, for every processor added, the communication overhead is 1% of the original execution time?

- c) (5 pts) What is the speedup with 8 processors if, for every time the number of processors is doubled, the communication overhead is increased by 1% of the original execution time?
- d) (5 pts) What is the speedup with N processors if, for every time the number of processors is doubled, the communication overhead is increased by 1% of the original execution time?
- e) (5 pts) Write the general equation that solves this question: What is the number of processors with the highest speedup in an application in which P% of the original execution time is parallelizable, and, for every time the number of processors is doubled, the communication is increased by 1% of the original execution time?

Answer:

- a) The speedup = 1 / ((1 0.8) + 0.8 / N) = 1 / (0.2 + 0.8 / N)
- b) The speedup = $1 / ((1 0.8) + 8 \times 0.01 + 0.8 / 8) \approx 2.63$
- c) The number of processor is doubled three times in order to have 8 processors The speedup = $1 / ((1 - 0.8) + 3 \times 0.01 + 0.8 / 8) \approx 3.03$
- d) To reach N processors, the number of doubling is log_2N . The speedup = $1 / ((1 - 0.8) + 0.01 log_2N + 0.8 / N)$
- e) The speedup = $1 / (1 0.01P + 0.01 \log_2 N + 0.01P / N)$ To get the number of processors with the highest speedup, the derivative (on N) of the speedup should be 0. Therefore, the equation is $-1 (1 - 0.01P + 0.01 \log_2 N + 0.01P / N)^{-2} (0.01 / (N \ln 2) - 0.01P/N^2) = 0$ (ln2 means log_e2) For the equation to be zero, it must be that $0.01 / (N \ln 2) - 0.01P/N^2 = 0$ The equation can be solved to have N = $0.01P \cdot \ln 2 / 0.01 = P\ln 2$