EECS 645

Hao Luo

Sep 18th

Homework 3

1.4 [20/10/20] <1.5> Figure 1.23 presents the power consumption of several computer system components. In this exercise, we will explore how the hard drive affects power consumption for the system.

a. [20] <1.5> Assuming the maximum load for each component, and a power supply efficiency of 80%, what wattage must the server's power supply deliver to a system with an Intel Pentium 4 chip, 2 GB 240-pin Kingston DRAM, and one 7200 rpm hard drive?

Solution: Based on the maximum load for each component, the server's peak power consumption = 1 * 66W (a Pentium 4 chip) + 2 * 2.3W (1 GB 240-pin Kingston DRAM) + 1 * 7.9W (7200 rpm hard drive (read/ seek)) = 66 + 4.6 + 7.9 = 78.5 W

Because of the supply efficiency of 80%,

Total Power Supply = useful Power / efficiency = 78.5W / 80% = 78.5 / 0.8 = 98.125 W

b. [10] <1.5> How much power will the 7200 rpm disk drive consume if it is idle roughly 60% of the time?

Solution: The disk drive has roughly 60% time, and it means it only read / write in remaining 40% time. Therefore, for a 7200 rpm disk drive, the current power consumption = 4.0W (idle) * 60% + 7.9W (read / seek) * 40% = 4.0 * 0.6 + 7.9 * 0.4 = 5.56 W

c. [20] <1.5> Given that the time to read data off a 7200 rpm disk drive will be roughly 75% of a 5400 rpm disk, at what idle time of the 7200 rpm disk will the power consumption be equal, on average, for the two disks?

Solution: Assuming the busy time of 7200 rpm disk as x, then the busy time of 5400rpm disk will be x/0.75. When power consumption of both two disk is equal, we get this equation that

$7.9 * x + 4.0 * (1 - x) = 7.0 * x / 0.75 + 2.9 * (1 – x / 0.75)$

$3.9x + 4.0 = 5.47x + 2.9$

X = 0.70, hence the idle time of the 7200 rpm disk is (1 – 0.70) = 0.30

1.5 [10/10/20] 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 Figure 1.23 for your power calculations.

a. [10] 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 240-pin DRAM, and a single 7200 rpm hard drive can you cool with one cooling door?

Solution: The dissipation of a cooling door is 14KW = 14000 W, also, in order to dissipate the heat from one server, we need consider about the heat from the condition of the maximum load in this server. Therefore, by calculating, the power consumption of one server in max load condition = 1 * 66W (a Pentium 4 chip) + 1 * 2.3W (1 GB 240-pin Kingston DRAM) + 1 * 7.9W (7200 rpm hard drive (read/seek)) = 76.2 W

The max numbers of server = the maximum dissipation / the power consumption of one server (max load) = 14000W / 76.2W = 183.7 => 183

Hence, one cooling door is able to cool with 183 servers.

b. [10] 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?

Solution: After the number of disk doubled, the total power consumption of one server (max load) = 66 + 2.3 + 7.9 * 2 = 84.1 W

The max numbers of server = = the maximum dissipation / the power consumption of one server (max load) = 14000 / 78.5 = 166.47 => 166

c. [20] 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?

Solution: Total Power Consumption of one server (max load) = 66 + 2.3 +7.9 = 76.2 W

The amount of heat can dissipate in a 11 square rack = 200 * 11 = 2200 W

The numbers of server in a single rack = 2200 / 76.2 = 28.87 => 28

Hence, 28 servers can be placed on a single rack.

The total consumption of the server rack = 76.2 * 28 = 2133.6 W, which is less than a cooling door's dissipation, 14KW. Therefore, it only needs 1 cooling door.

1.7 [20/20/20/20] <1.6, 1.9> Your Company's internal studies show that a single-core system is sufficient for the demand on your processing power; however, you are exploring whether you could save power by using two cores.

a. [20] <1.9> Assume your application is 80% parallelizable. By how much could you decrease the frequency and get the same performance?

Solution: The parallelizable part = 0.8, and the not parallelizable part = 0.2. Based on Amdahl's law, following the formula, the speedup = 1 / (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)). By using two cores, the ideal speedup will be 2 times.

Thus, the speedup = 1 / (1 - 0.8 + 0.8 / 2) = 1 / 0.6, in order to finish the same word load in same execution, the frequency new = 1 / frequency old = 1 / speedup = 1 / (1 / 0.6) =0.6. Comparing with original frequency 1, the decrease of frequency = 1 – 0.6 = 40%.

b. [20] Assume that the voltage may be decreased linearly with the frequency. Using the equation in Section 1.5, how much dynamic power would the dual-core system require as compared to the single-core system?

Solution: From the question A, we get the decreased frequency 0.6 by using two cores. Based on the formula of dynamic power, dynamic power = ½ * Capacitive Load * Voltage ^ 2 * Frequency switched. Since the capacitance is unchanged, the ratio of the dynamic memory of dual core system = Power new / Power old = (new Voltage ^2 * new Frequency switched) / (old Voltage ^ 2 * old Frequency switched) = ((0.6V) ^2 * 0.6F) / (V^2 * F) = 0.36V ^ 2 * F / V ^ 2 * F = 0.216

c. [20] now assume the voltage may not decrease below 25% of the original voltage. This voltage is referred to as the voltage floor, and any voltage lower than that will lose the state. What percent of parallelization gives you a voltage at the voltage floor?

Solution: When the voltage may not decrease below 25%, it means the voltage floor will be at least 75%, which is the requirement to keep the state.

Assuming the percent of parallelization as x, following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)). As we already know, the number of processors are 2, so the ideal speedup is 2 times. Now, the only job is to ensure the voltage floor work normally (higher than voltage floor), thus it just need speedup 1 to keep it work there. Therefore, the equation is

1 = 0.75 / (1 – x + x / 2) => 0.5 x = 0.25 => x = 0.5

d. [20] Using the equation in Section 1.5, how much dynamic power would the dual-core system require as compared to the single-core system when taking into account the voltage floor?

Solution: From the dynamic power, dynamic power = ½ * Capacitive Load * Voltage ^ 2 * Frequency switched. From the problem C, we find the 75% is the lowest voltage to keep the state of voltage floor. In problem A, we get the decreased frequency 0.6 by using two cores. So, since the capacitance is unchanged, the ratio of dynamic memory of the dual-core system = Power new / Power old = (new Voltage ^2 * new Frequency switched) / (old Voltage ^ 2 * old Frequency switched) = (0.75V) ^2 * 0.6F / (V^2 * F) = 0.5625V^2 * 0.6F / V^2 * F = 0.3375

1.12 [20/20/20] 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] if a company has 10,000 computers, each with a MTTF of 35 days, and it experiences catastrophic failure only if 1/3 of the computers fail, what is the MTTF for the system?

Solution: MTTF for the system = 1 / FailureRatesystem

Failure rate for the system = 10000 * 1 / 35 = 285.7

MTTF for the system = 1 / Failure rate * numbers of failure computers = 1 / 285.7 * (10000 / 3) = 11.667 days

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

Solution: The time of MTTF = 35 / 10000 = 0.0035 days = 5.04 min, which means the mean time to failure about 5 min for one computer. Actually, for an engineer to handle the failure, 5 min might not be enough cause there might be multiple problems about the failure. Therefore, spending $1000 per computer to update will provide more time and flexibility for engineers to repair these computers. If

the failure for the system happens, the 1/3 of computers will be suffered. It will be a painful loss for a server farm.Hence, it would be a good business decision.

c. [20] 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?

Solution: Assuming the cost of downtime of other three quarters as x, and the fourth quarter will be 2x. From the chart, we get the cost of downtime per hour $ 90000, and the cost is equal at all the time of the year. Thus, we get the average cost of down time for each quarter in this month as

(x + x + x + 2x) / 4 = 90000 => x = 72000

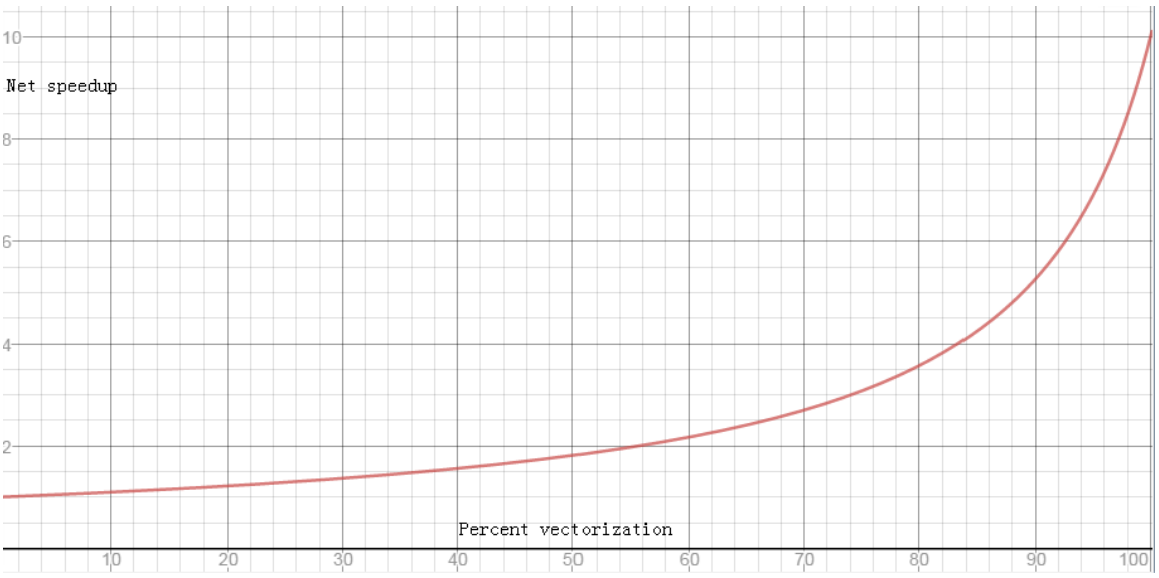
So, the cost of downtime per hour of the fourth quarter is 72000 * 2 = 144000 $ / hour

1.14 [20/10/10/10/15] in this exercise, assume that we are considering enhancing a machine by adding vector hardware to it. When a computation is run in vector mode on the vector hardware, it is 10 times faster than the normal mode of execution. We call the percentage of time that could be spent using vector mode the percentage of vectorization. Vectors are discussed in Chapter 4, but you don't need to know anything about how they work to answer this question!

a. [20] Draw a graph that plots the speedup as a percentage of the computation performed in vector mode. Label the y-axis "Net speedup" and label the x-axis "Percent vectorization."

Solution: Following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)).

The equation between "Net speedup" and "Percent vectorization" is y = 100 / (100 – x + x / 10)



b. [10] what percentage of vectorization is needed to achieve a speedup of 2?

Solution: Following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)). When the computation run in vector mode, it will 10 times faster than the normal mode. So, the ideal speedup should be 10 times. Also, the speedup is 2, assuming the percentage of vectorization as x, and we get this equation 1 / (1 – x + x / 10) = 2

=> x = 5 / 9 = 0.556

c. [10] what percentage of the computation run time is spent in vector mode if a speedup of 2 is achieved?

Solution:

d. [10] what percentage of vectorization is needed to achieve one-half the maximum speedup attainable from using vector mode?

Solution: The ideal maximum speedup is 10, the half of this speedup is 5

Following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)). Assuming the percentage of vectorization to achieve one-half the maximum speedup as x,

5 = 1 / (1 – x + x / 10) => x = 8 / 9 = 0.89

e. [15] Suppose you have measured the percentage of vectorization of the program to be 70%. The hardware design group estimates it can speed up the vector hardware even more with significant additional investment. You wonder whether the compiler crew could increase the percentage of vectorization, instead. What percentage of vectorization would the compiler team need to achieve in order to equal an addition 2× speedup in the vector unit (beyond the initial 10×)?

Solution: The current speedup = 1 / (1 – 0.7 + 0.7 / 10) = 2.7, the addition 2x speedup is 5.4. In vector mode, the ideal speedup is 10 times because vector mode is 10 times faster than normal mode.

Following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)).

Assuming the percentage of vectorization as x, in order to achieve 5.4 times speedup, we get this

5.4 = 1 / (1 – x + x / 10) => x = 44 / 48.6 = 0.91

The percentage would need to be 91%

1.16 [20/20/15] 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. [20] If the new fast floating-point unit speeds up floating-point operations by, on average, 2×, and floating-point operations take 20% of the original

program's execution time, what is the overall speedup (ignoring the penalty to any other instructions)?

Solution: Following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)). The enhanced speedup is 2 times, and the floating-point operation is the enhanced fraction, which = 20% * 1(original Ex time) = 0.2

Speedup = 1 / (1 – 0.2 + 0.2 / 2) = 1 / 0.9 = 1.11

b. [20] 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?

Solution: Following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)). For the enhanced parts, the floating-point operation is speeding up with 2 times, and the data cache accesses is speeding up with 2/3 times at the same time. As a result of this, the speedup = 1 / (1 – 0.1 – 0.2 + 0.2 / 2 + 0.1 / (2 / 3)) = 1 / 0.95 = 1.05

c. [15] 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?

Solution: From problem B, we get the percentage of floating-point operation takes 20% in original excution time. After the speeding up of 2 processor, it is 20% / 2 = 0.1. Also, we already have the new Ex time 0.95 from problem B. Therefore, the percentage of floating-point operations in execution time = 0.1 / 0.95 = 0.105 => 10.5%

From problem B, we get the percentage of data cache accesses takes 10% in original excution time. After the slowdowning of 1.5 times, it is 10% * 1.5 = 0.15. Also, we already have the new Ex time 0.95 from problem B. Therefore, the percentage of data cache accesses in execution time = 0.15 / 0.95 = 0.158 => 15.8%

1.18 [10/20/20/20/25] 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. [10] what is the speedup with N processors if 80% of the application is parallelizable, ignoring the cost of communication?

Solution: The parallelizable part is 80%, based on Amdahl's law, the speedup = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)). The application is speedup with N processors, thus the ideal Speedup enhanced will be N times. In this way, without the cost of communication, the speedup = 1 / (1 – 0.8 + 0.8 / N) = 1 / (0.2 + 0.8 / N)

b. [20] what is the speedup with 8 processors if, for every processor added, the communication overhead is 0.5% of the original execution time.

Solution: The application is speedup with 8 processors, and it means the ideal speeding up will be 8 times. For the extra execution time of the communication overhead, it will be increasing with the numbers of processor.

Following Amdahl's law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + Fraction (enhanced) / Speedup (enhanced)).

For 8 processors, the communication overhead = 8 * 0.5% * 1 (Ex Time old) = 0.04, and the Ex Time new should be added with the communication overhead.

The speedup = 1 / (1 – 0.8 + 0.04 + 0.8 / 8) = 1 / 0.34 = 2.941

c. [20] what is the speedup with 8 processors if, for every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

Solution: In this problem, the number of processors is still 8. Also, we know that the communication overhead will be increased by 0.5% for every time the number of processors is doubled. By calculating with logarithmic formula “8 = log2(N)”, we can get that N = 3, and it means the 8 processors double the overhead of communication 3 times, as 1->2, 2->4 and 4->8. So, the extra execution time for communication overhead is 3 * 0.5% = 3 * 0.005 * 1 (Ex Time old) = 0.015

Following Amdahl’s law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + communication overhead + Fraction (enhanced) / Speedup (enhanced)).

The speedup = 1 / (1 – 0.8 + 0.015 + 0.8 / 8) = 1 / 0.315 = 3.175

d. [20] what is the speedup with N processors if, for every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

Solution: In the same way as question C, there are N processors, and the communication overhead will be increased by 0.5% for everyone time the number of processors doubled. With the logarithmic formula “N = log2a”, after applying this formula, then we get a = log2N. So, the number of processors is doubled log2N times, thus the overhead in execution time will be log2N * 0.5% * 1 (Ex Time old) = log2N * 0,005

Following Amdahl’s law, the speedup = Ex Time old / Ex Time new = 1/ (1 – Fraction (enhanced) + communication overhead + Fraction (enhanced) / Speedup (enhanced)).

The speedup = 1 / (1 – 0.8 + log2N * 0.005 + 0.8 / N) = 1 / (0.2 + 0.005 * log2N + 0.8 / N)

e. [25] 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 0.5% of the original execution time?

Solution: