Computer Architecture

Quiz 1 March

26, 2021

Abstract

The underlying idea this quiz explores is that computer architecture is guided by quantitative design and analysis.

1 Vocabulary

We gain power over ideas which we can associate with a word. Instead of being fleeting collections of activations of neurons, alone, named ideas can be recalled, communicated, built upon.

Thus, we need vocabulary. Provide a definition for:

- computer instruction

A computer instruction is a set of steps or machine learning instructions on carrying out specific processor functions, or more generally, a particular documentation on how to perform, or maintain a piece of software or hardware.

- register

A register is a special type of computer memory or a temporary storage unit specialized in performing tasks very rapidly. They are used to quickly fetch, storing, and quickly access pieces of data and instructions.

stack pointer

A stack pointer is a small register whose function is to store the address of the last program request on stack. The stack operates on a last in first out fashion. So, when new data is pushed onto the stack, the stack pointer moves to the next physical memory address and the new item is copied to the new address.

-

- opcode

Opcode is the section of a machine language instruction that specifies an operation to be performed.

- branching

Branching is an instruction that tells the computer to execute a different part of the program as opposed to executing several statements in a one-by-one fashion.

- scalability

Scalability refers to the ability of a system to handle an increased amount of workload.

- power density Amdahl's Law

Power density Amdahl's Law designates a formula to speed up the latency of a particular execution at a fixed workload. It is speedup = performance for entire task using enhancement when possible / performance for entire task without using enhancement.

-

- dynamic voltage-frequency scaling

Dynamic voltage-frequency scaling is a technique aimed at improving energy efficiency despite flat clock rates and constant supply voltages.

2 Relationships

Describe some relationship between the two terms, or explain why they are not related:

- Power, Energy

They are related by cause and effect. Energy causes change in a computer system, power is the rate in which energy is moved or used in said computer system.

- Bandwidth, Latency

They are related with respect to data. Bandwidth measures size of data being moved, latency measures delay/speed of data being moved.

dynamic energy, static power

They are not related. Static power remains constant even if there are activities being performed, where dynamic energy has the ability to change.

-

- temporal locality, spatial locality

They are not related. Temporal locality reuses specific data within a short period of time and spatial locality uses data elements with storage locations in close proximity.

3 Calculations

Computer Architecture is quantitative, so we perform some computations.

1. An algorithm has some data parallelism that can be exploited. The rest of the program must be carried out serially. If the entire program were executed on a single processor, it would take 16T units of time. Of this, 12T is the time devoted to the part that has parallelism. When the parallelism is exploited, the speedup of that portion is 12. What is the shortest execution time?

New Execution Time = Old Execution Time * ((1-Fraction)+Fraction/Speedup) → 16*((1-0.75)+0.75/12) →5T

2. An assembly language program has 1024 instructions, some of which areloops. If all the loops are unrolled, the number of instructions executed is 2048. This program runs on a processor whose clock rate is 2^9 clock periods per second. The pipeline accepts one instruction every clock cycle. The latency from the beginning to the end of the pipeline is 16 clock cycles. What is the CPU time to execute this program?

CPU Time = CPU Clock Cycles / Clock Rate → 16 / 2^9 = 0.03125 seconds

3. Assuming that a collection of modules has independent failures, and exponentially distributed lifetimes, in particular:
   - 2 GPUs, rated at 150,000 hours
   - 1 ATA controller, rated at 150,000 hours
   - 1 power supply, rated at 150,000 hours
   - 2 fan, rated at 150,000 hours
   - 2 disks, rated at 150,000 hours What is the overall failure rate?

Failure Rate = Total Failures / Total number of hours, but the overall failure rate is the sum of the failure rates of the modules → (2/150,000) + (1/150,000)+(1/150,000)+(2/150,000)+(2/150,000) = 8/750,000 = 0.00001067