Concurrency and Parallelism: Fraternal Twins in Computer Science

Mya Souvanna

CSTEM, University of Arkansas at Little Rock

CPSC 3369: Computer Systems and Assembly

Professor Orme

May 10, 2022

Concurrency and Parallelism: Fraternal Twins in Computer Science

Computers have come a long way in history. Initially, people built early computers to calculate mathematical calculations that people would solve by hand. However, just like people, computers have progressed throughout history as well. Today's computers can do much more than solve mathematical calculations. Computers help better and bring together our society. The versatility of computers is never-ending. For example, computers can do simple tasks such as addition; they can also do complex tasks like artificial intelligence and three-dimensional printing. The computer is made up of multiple layers like an onion. Whether the layers are hardware or software, these layers work together to carry out the computer's tasks. Concurrency and parallelism allow the computer to execute multiple tasks at a time. Many people confuse the two processing techniques to be identical to one another. Like fraternal twins, concurrency and parallelism are similar but have more differences than one may initially think. A real-life example of the two programming paradigms is a hot dog stand. At the hot dog stand, the workers must do two tasks. The two tasks are taking the customers' orders and cooking the hotdogs. At the concurrency hot dog stand, one person works at this hot dog stand; the parallelism hot dog stand has two workers. The only worker of the concurrency stand will be responsible for both tasks. On the other hand, both workers will split the workload and do only one of the two tasks at the parallelism stand. Concurrency and parallelism are just two programming paradigms that help the computer function properly.

Concurrency

The word "concurrency" typically describes things that are done or occur at the same time. Concurrency is seen in almost every aspect of our lives and is a natural phenomenon. An example of concurrency would be a student taking a class that gives both high school and college credit. In computer science, however, concurrency is “multiple computations . . . happening at the same time” (“Reading 17: Concurrency,” n.d.). Concurrency is seen everywhere in modern computers today. For example, it allows a computer to run multiple applications simultaneously in an instance. While concurrency and parallelism are essentially the same thing, there is a huge difference that separates the two techniques apart. Concurrency allows multiple tasks to execute at the same time, but these tasks do not execute at the same instant. Instead, the processor may start one task before the last one finishes (Anuradhac & Arvindpdmn, 2021). This is how the computer’s processor uses concurrency to “check” on the tasks that it is running to make sure that no errors occur within the computer’s software.

Concurrency and its Many Forms

Like most things in life, there are many ways to do one task. For instance, a morning routine for some people may be to shower, dress, and then eat; others may eat, shower, and then dress for their morning routine. There is not an exact and correct way to do either morning routines or concurrent processing. There are three main types of concurrent processing: multithreading, asynchrony, and preemptive multitasking (“What is Concurrent Computing?”, 2020). The way that concurrency is achieved will differ from computer to computer. However, it will usually fall under one of those categories. While these are the three main types of concurrent processing, concurrent processing is not limited to just these three categories.

Multithreading

In computer science, a thread is a set of instructions—independent from its "parent" process—executed by the computer's CPU (Computer Hope, 2019a). Therefore, multithreading is the process of a computer, program, or operating system (OS) that executes multiple threads at once. Multithreading helps improve a computer's response time. The response time improves by "[allowing] a computer to maximize the use of available [CPU] power and [saving] the user time ("What is Concurrent Computing?", 2020)." For example, while a thread waits for its execution, the computer can do other tasks until it eventually executes. A few examples of multithreading applications include editing applications and programming code compilation software (Computer Hope, 2019a). According to Computer Hope (2019a), Intel—an American technology company—created a technology called "hyper-threading," which helps improve CPU performance and allows the processor to execute additional instructions at a time. The use of this technology permits Intel devices to use processor resources more efficiently and improves the overall performance of the device as well. Even after twenty years since the creation of "hyper-threading," Intel still uses this technology, thus proving that its usefulness in today's Intel CPUs.

Asynchrony and Preemptive Multitasking

The general definition of the word *asynchrony* refers to a lack of concurrency during a given time. This lack of concurrency helps a computer achieve concurrent processing despite the ironic definition of *asynchrony*. Asynchronous computing happens when processes in the computer work independently ("What is Concurrent Computing?", 2020). When these processes work by themselves, the computer achieves shorter execution times due to the simultaneous executions of the multiple yet independent processes. Though these processes do not interfere with others, they occur in the same timeframe but in overlapping intervals (Computer Hope, 2019b). Asynchrony allows independent execution without the fear of interference or waiting from other processes within the computer. Preemptive multitasking is like asynchrony to an extent. Instead of dealing with processes, preemptive multitasking deals with software or programs. The operating system will switch between the software to "[prevent] a program from fully taking control of the computer’s processor" (“What is Concurrent Computing?”, 2020). While the three main types of concurrent processing are different in meaning and practice, they are essentially the same at the base level of what concurrency is. Concurrent processing is diverse with the different ways that a computer can achieve concurrency.

Advantages of Concurrency

Concurrency allows multiple things to execute during a single time interval; as a result, today’s computers can experience many advantages from concurrent processing. Some advantages include running many applications, increased responsiveness, the allowance of overlapping input and output, and efficiently used computer resources. The computer can run multiple applications at the same time since the processor runs many threads at the same time, but not in the same instance. As stated earlier, preemptive multitasking—one of the main types of concurrent processing—deals with the applications and software of the computer. The operating system will switch between the running software to prevent a monopoly on the processor. By doing this action, the computer can run multiple programs at once without crashing or freezing the computer (“What is Concurrent Computing?”, 2020). According to “What is Concurrent Computing?” (2020), computer programs become more responsive due to the use of multiple threads. According to “What is Concurrent Computing?” (2020), computer programs become more responsive due to using multiple threads. Instead of waiting for other programs to finish, a program can use that standby time to complete other tasks (“What is Concurrent Computing?”, 2020). Without concurrency, an application would have to wait to run after the previous application was complete in execution, thus delaying the overall response time of the computer (GeeksForGeeks, 2021a). Programs that can overlap input/output (IO) data and computations give higher throughput by maximizing CPU power (“What is Concurrent

Computing?", 2020). By doing so, computer resources are efficiently used and not wasted. If a program does not use a computer's resources, another program may use them. This leads to better resource utilization and less resource waste (GeeksForGeeks, 2021a).

Disadvantages of Concurrency

While concurrency is a good programming paradigm, it has disadvantages that make it unfavorable in some aspects. Some disadvantages of concurrent processing are the difficulty to debug, the global resources problem, and its slow performance. In computer science, a race condition is when the program's outcome is dependent based on "the relative timing of events in concurrent computations A and B" ("Reading 17: Concurrency"). Race conditions are hard to detect during testing since it is hard to reproduce due to their non-deterministic nature; these "heisenbugs" found in parts of the program will be hard to localize since it is difficult for the program to produce the same results twice ("Reading 17: Concurrency"). According to GeeksForGeeks (2021a), sharing global resources between two processes is difficult since the sequence in which the program executes both processes is important since both processes read and write on the same variable. Sometimes, too much of a good thing is harmful. If the computer runs too many programs all at once, the performance of the computer severely hinders the speed of the computer (GeeksForGeeks, 2021a). Although concurrency has its disadvantages, it should not stop one from using the programming paradigm altogether since the advantages outweigh the disadvantages.

Parallelism

Like concurrency, parallelism can run multiple tasks at the same time and is used in modern computers today. However, it allows the processor to run multiple tasks simultaneously at the same instance of time unlike concurrency (Nagarajan, 2019). This means that the processor does not have to check each individual task to determine when the next task can run. Parallelism is whenever “large problems break into independent, smaller. . . parts . . . that can be processed in one go” (“What is Parallel Computing,” n.d.). This means that instead of tackling huge tasks and having it run one at a time like concurrency, the processor will break down a single task into smaller parts and then have execute all the smaller parts at the same time. In some instances, parallelism does not even need a minimum of two tasks to begin executing. According to Nagarajan (2019), parallelism can “run parts of tasks OR multiple tasks.” This means that parallelism is versatile and can execute whatever tasks that the computer hands to the processor. Parallelism is a sub-unit of concurrency as it concurrently computes threads and processes at the same time. Parallel computing can be found in several pieces of technology such as data mines, databases, and even augmented and virtual reality (“What is Parallel Computing,” n.d.)!

Parallelism and its Many Forms

Similar to concurrent processing, different techniques can be used to achieve parallel processing. Parallelism is more or less an umbrella term where "large problems break into independent, smaller, usually similar parts that can be processed in one go" reality (“What is Parallel Computing,” n.d.). Like concurrent computing, there are generally three types of parallel computing: bit-level parallelism, instruction-level parallelism, and task parallelism (GeeksForGeeks, 2021b). While these three types of parallelism will differ in technique, they still achieve parallelism in the end in one way or another. Because these are the three generally used types of parallelism, parallelism is not limited to just these three types of parallel computing. Similar to concurrent computing, there are several other techniques out there that can achieve parallelism. Like the three main types of parallelism, the only difference between these other techniques is the way that they achieve parallelism.

Bit-Level Parallelism

According to GeeksForGeeks (2021b), bit-level parallelism is dependent on increasing the processor's size. Bit-level parallelism works by "[reducing] the number of instructions that the processor must execute" (“What is Parallel Computing,” n.d.). By doing so, the operation is split into a series of these instructions. For example, if a computer had an 8-bit processor, but the computer had to execute on a 16-bit number, then the processor will operate on the lower-order 8-bit and then operate on the higher-order 8-bit (GeeksForGeeks, 2021b). By operating on an 8-bit at a time, an 8-bit processor would execute two instructions for a 16-bit operation. Else, if there the processor had an extra 8 bits, then the processor would only execute one instruction since it is a 16-bit processor.

Instruction-Level Parallelism and Task Parallelism

In instruction-level parallelism, the processor will decide what instructions are executed during each clock cycle phase. During this type of parallelism, the processor can only address less than one instruction for each CPU clock cycle; the processor can also change the order of the instructions and even group them (GeeksForGeeks, 2021b). According to "What is Parallel Computing" (n.d.), the software used for instruction-level parallelism is called "static parallelism." Static parallelism allows the computer to decide which instructions are simultaneously executed at a time and the executions of these instructions do not alter the outcome of the program (GeeksForGeeks, 2021b). While bit-level parallelism deals with bits and instruction-level parallelism deals with instructions, one can safely assume that task-level parallelism deals with tasks. Task-parallelism will break down the computer's tasks into subtasks before allocating resources for each subtask; once the resource allocation is done for all of the subtasks, the subtasks will be concurrently executed by the multiple processors of the computer (What is Parallel Computing," n.d.).

Advantages of Parallelism

Computers that use parallel computing will experience many benefits. Because parallelism allows tasks to execute at the same time during the same instance, it is essentially a sub-unit of concurrency. Because parallelism is a concurrency sub-unit, the computer can provide concurrent processing, which helps provide a faster response time to the user. Another benefit of parallelism is the lower processing cost and time for the computer. According to "What is Parallel Computing" (n.d.), cheap components are used to create parallel clusters, which leads to an increase in parallel cluster creations. Because more resources were used to create these parallel clusters, the processing time for the computer will decrease significantly, leading to a saving in both time and money. Parallelism also makes use of the non-local resources of a computer. When local resources are finite or scarce, parallelism will take advantage of non-local resources and can use those resources to continue executing tasks and whatnot like normal (GeeksForGeeks, 2021b).

Disadvantages of Parallelism

Like many good things in one's everyday life, there are many disadvantages to it. For instance, a knife is beneficial for cooking since people can use it to cut meat; if one isn't careful enough, they can accidentally cut themselves or others while they are using a knife. Parallelism makes use of several CPUs to achieve the ability to execute tasks simultaneously at the same instance. As a result of the multiple CPUs used, parallelism will have a high-power consumption to the multi-core architectures that it uses ("What is Parallel Computing," n.d.). Like concurrency, it can be quite difficult to debug in parallelism since it is hard to detect bugs. However, parallelism bugs are much easier to fix when compared to concurrency bugs.

Conclusion

In terms of technological advancement, computers have come a long way. From the first instance of computers being made from wood, today's computers are made from things such as a CPU, keyboard, and mouse. Early computers were designed to calculate mathematical computations that would otherwise be solved by hand. Computers today are capable of far more than just solving math problems. Computers, like humans, have evolved over time as history progressed. Computers make our civilization more efficient and cohesive. Computers' capabilities are virtually limitless. Computers, for example, may perform simple activities like addition, as well as more complicated jobs such as artificial intelligence and three-dimensional printing. A doting parent can even use computers to post about their children on social media. Like an onion, the computer has many layers. These layers, whether hardware or software, work together to complete the functions of the computer. Concurrency and parallelism enable a computer to do numerous tasks simultaneously. Many individuals consider the two processing methods to be interchangeable. Concurrency and parallelism are akin to fraternal twins, yet they have more differences than one might imagine. A hot dog stand is a real-life example of the two programming paradigms. Workers at the hot dog station must perform two jobs. The two jobs are collecting orders from clients and cooking hotdogs. One person works at the concurrency hot dog stand, whereas two people work at the parallelism hot dog stand. Both duties will be handled by the concurrency stand's one employee. At the parallelism stand, on the other hand, both workers will split the workload and perform only one of the two tasks. Concurrency and parallelism are two programming concepts that aid in the proper operation of a computer.

Without these two programming paradigms, the computer wouldn’t run efficiently and would be slow.

References

Anuradhac & Arvindpdmn. (2021, June 27). *Concurrency vs Parallelism*. Devopedia. Retrieved April 10, 2022, from https://devopedia.org/concurrency-vs-parallelism

Computer Hope. (2019a, May 4). *What is a Thread?* https://www.computerhope.com/jargon/t/thread.htm

Computer Hope. (2019b, June 30). *What is Asynchronous?* https://www.computerhope.com/jargon/a/asynchro.htm#asynchronous-programming

GeeksforGeeks. (2021a, November 26). *Concurrency in Operating System*. Retrieved April 11, 2022, from https://www.geeksforgeeks.org/concurrency-in-operating-system/

GeeksforGeeks. (2021b, June 4). *Introduction to Parallel Computing*. https://www.geeksforgeeks.org/introduction-to-parallel-computing/

Nagarajan, M. (2019, December 2). *Concurrency vs. Parallelism — A brief view - Madhavan Nagarajan*. Medium. Retrieved April 10, 2022, from https://medium.com/@itIsMadhavan/concurrency-vs-parallelism-a-brief-review-b337c8dac350

Reading 17: Concurrency. (n.d.). Web.Mit.Edu. Retrieved April 10, 2022, from https://web.mit.edu/6.005/www/fa14/classes/17-concurrency/

What is Concurrent Computing? (2020, December 15). Techslang. https://www.techslang.com/definition/what-is-concurrent-computing/

What is Parallel Computing. (n.d.). Javatpoint. Retrieved April 11, 2022, from https://www.javatpoint.com/what-is-parallel-computing