How Evolving Computer Technology Has Improved System Performance

John Kucera

CMIS 310 6380 Computer Systems and Architecture

University of Maryland Global Campus

September 8, 2020

How Evolving Computer Technology Has Improved System Performance

The competence of computer technology is constantly evolving. Every year, computer scientists and engineers develop new ways to improve the speed, storage, error prevention, and capabilities of computers. This evolution is what allows websites to be accessed within seconds, mobile devices to run when they're as small as a hand, and large applications to be run on home systems. Among these evolving computer technologies that have improved computer performance in the past 25 years are RISC (Reduced Instruction Set Computer), pipelining, cache memory, and virtual memory. Since pipelining, cache memory, and virtual memory are implemented in RISC's design, RISC's evolution is the most important for improving system performance.

RISC: Reduced Instruction Set Computer

One major approach to improving system performance is a Reduced Instruction Set Computer, more commonly known as RISC. When first designing RISC, the engineers wanted to "create a high performance single-chip processor with a fast clock rate… [and] it was necessary to discard a new minimal instruction set that could fit on a single chip" (Dowd & Severance, 2010). With the instructions modified to be fixed-length, it would take less time and power to get them translated into assembly language. Being a smaller physical size also meant RISC had the potential to be used in portable devices such as phones and handheld game consoles.

In 2010, a new ISA (Instruction Set Architecture) called RISC-V was introduced for the RISC implementations to use and is still gaining popularity today. RISC-V uses optimized instructions while maintaining "a very high level of flexibility at a much lower cost than proprietary RISC. It allows users to produce custom chips suited to specific applications" (Greengard, 2020). RISC-V is open source and royalty-free which allows it to be freely used and modified by anyone. One of RISC-V's enhancements is being able to compress instructions which allows more storage space in the implementation. RISC-V is also capable of handling IEEE 754 floating-point which enables an additional number system to be used for instructions. In addition to RISC gaining widespread usage after costs were lowered, system performance has greatly improved from the evolution of RISC due to the evolution of storage space optimization, capabilities of using multiple binary number systems, and instruction flexibility.

Pipelining

Pipelining is another major concept that was brought in to improve system performance. Computers that use pipelining will perform “iterations [that] are executed in overlapped fashion to increase parallelism” (Alan et al., 1995). With this parallelism, output will be processed as input for the next instruction as new input is being processed in the previous instruction. In other words, fetching, decoding, and execution occur simultaneously in the system. Pipelining therefore leads to faster processed instructions because many can be fulfilled at the same time. Pipelining is heavily utilized in RISC since the latter’s design is specialized in quick execution of fixed-length instructions.

In the past 25 years, Pipelining has evolved into a new form called superpipelining. With superpipelining, the “pipes” where different instructions are processed get split “to divide the work into larger number of stages, which lets the chip run at a higher clock rate” (Slater, 1995). Pipelining already sped up the instruction execution process, but superpipelining gave an extra boost to it by enhancing the clock’s speed and therefore allowing more instructions to run at once. Superpipelining became so efficient that it was utilized not only in RISC, but in CISC (Complex Instruction Set Computer) where low-level multi-step operations run in a single instruction set. The introduction of pipelining and its evolution into superpipelining have remarkably advanced system performance, especially in RISC, due to increasing the speed at which instructions can be completed.

Cache Memory

An additional approach to improving system performance developed in the past 25 years was cache memory. Cache memory is a software or hardware component of a computer that stores commonly accessed data. Later when the data is requested again, it becomes simpler for the processor to find and “access times on a specific disc become faster the more you use that disc… the most requested data from a disc is loaded into the hard disk cache, and future access proceeds from there” (Starrett, 1996). Although the cache holds less storage than RAM, the former is much faster in serving data. As opposed to the CPU tediously going to the RAM for all data, Cache memory is therefore necessary for the CPU to retrieve regularly accessed data to increase the efficiency of the system’s performance.

One common concern about cache memory is that due to its high speed and power, it consumes a lot of energy which can be inefficient for a computer. In 2019, a novel cache organization called FOS-Mt introduced an approach to lower cache memory’s energy consumption. FOS-Mt’s proposal for a new cache hierarchy has only two levels: L1 cache level where the most regularly accessed data lies, and a second cache level that is “sliced into small buggers, which are dynamically assigned to any of the running thread when they are expected to improve system performance. Those buffers that are not allocated to any core are powered off to save energy” (Puche et al., 2019). A cache memory may be essential for a speedy CPU, but it might not be worth it when considering the high energy consumption. This raises even more doubt when using cache memory with a RISC that is designed to always operate at high speeds. However, cache memory is now evolving to decrease that energy expenditure and revamp the energy distribution in the system, greatly improving system efficiency and performance.

Virtual Memory

Another concept brought in for improving system performance in the past 25 years is virtual memory. One common concern involving RAM is that there are large applications that users want to run which are too large for RAM to handle. Virtual memory is a scheme that was introduced to provide more space for large applications to run on a RAM that has less storage. With virtual memory, “some data stored in RAM that is not actively being used can be temporarily moved to virtual memory (which is physically located on a hard drive or other storage device)” (Rubens, 2019). Freed space in RAM allows it to optimize running the large applications that otherwise would not fit. In other words, it is a scheme that adds secondary memory space to RAM space. RISC also utilizes virtual memory as long as there is enough secondary storage.

A concern involving applications that heavily utilize the GPU (Graphics Processing Unit) is that the GPU and CPU are physically separated, and it takes time and user input to explicitly manage memory between them. Users are forced to manually copy and move data between the CPU and GPU depending on where instructions will be executed. To address this, a newer approach of virtual memory has been developed in 2020 called UVM (Unified Virtual Memory) which “allows GPUs and CPUs to share the same virtual memory space, and offloads memory management to the GPU driver and hardware, thus eliminating explicit copy-then-execute by the programmers” (Gu et al., 2020). Virtual memory’s base concept of sharing memory between different components in the computer has inspired other solutions like UVM to resolve ongoing issues in today’s systems. Having sowed the seeds for these modern implementations that increase useable memory, virtual memory has therefore contributed vastly to the increasing competence of system performance.

Conclusion

System performance would be much less than what it is today without the evolution computer technology concepts such as RISC, pipelining, cache memory, and virtual memory. RISC and the development of RISC-V allowed systems to operate faster from optimizing storage space, allowing instruction flexibility and modifying instructions to be fixed-length. Pipelining and superpipelining also contributed to the speed of systems by allowing instructions to process simultaneously. Cache memory increased the speed of data retrieval and FOS-Mt's new cache proposal cushioned cache memory's own weakness of energy consumption. Finally, virtual memory and its derivatives such as UVM have allowed for expanded useable memory which lets large applications run on RAM that would otherwise be too small. RISC is the most important among these approaches because it incorporates the other three concepts into its own architecture. Although all four concepts evolve alongside each other and enable each other's evolution, the other three concepts would not have evolved as they have if RISC did not exist. RISC's requirement of speedy operations and processing encouraged the development of pipelining and cache memory, and RISC's demand for additional memory invited the development of virtual memory. Today, RISC remains to be a centerpiece of computer technology around which the evolution of supporting computer technology concepts revolves in order to pursue the improvement of system performance.

References

Allan, V. H., Jones, R. B., Lee, R. M., & Allan, S. J. (1995). Software Pipelining. *ACM Computing Surveys*, *27*(3), 467–432. https://doi-org.ezproxy.umgc.edu/10.1145/212094.212131

Dowd, K., & Severance, C. (2010). 7.1.3 Fundamentals of RISC. In *High Performance Computing*. OpenStax-CNX.

Greengard, S. (2020). Will RISC-V Revolutionize Computing? The open instruction set for microprocessors promises to reshape computing and introduce new, more powerful capabilities. *Communications of the ACM*, *63*(5), 30–32. https://doi-org.ezproxy.umgc.edu/10.1145/3386377

Gu, Y., Wu, W., Li, Y., & Chen, L. (2020). *UVMBench: A Comprehensive Benchmark Suite for Researching Unified Virtual Memory in GPUs*.

Puche, J., Petit, S., Gómez, M. E., & Sahuquillo, J. (2020). An efficient cache flat storage organization for multithreaded workloads for low power processors. *Future Generation Computer Systems*, *110*, 1037–1054. https://doi-org.ezproxy.umgc.edu/10.1016/j.future.2019.11.024

Rubens, P. (2019, May 17). Virtual Memory. Retrieved September 08, 2020, from https://www.enterprisestorageforum.com/storage-hardware/virtual-memory.html

Slater, M. (1995, May 1). P6 details deliver on power promise: but don’t throw out your Pentium yet. *Computer Shopper*, *v15*(n5).

Starrett, R. A. (1996). How fast is fast enough. *CD-ROM Professional*, *9*(9), 91.