SCIENCE CHINA
Information Sciences

• PERSPECTIVE •

September 2019, Vol. 62 196101:1–196101:3
https://doi.org/10.1007/s11432-019-9860-1

Software system research in post-Moore's Law era: a historical perspective for the future

Xiaodong ZHANG

Department of Computer Science and Engineering, The Ohio State University, Columbus OH 43210, *USA*

Received 12 February 2019/Revised 12 March 2019/Accepted 31 March 2019/Published online 5 August 2019

Citation Zhang X D. Software system research in post-Moore's Law era: a historical perspective for the future. Sci China Inf Sci, 2019, 62(9): 196101, https://doi.org/10.1007/s11432-019-9860-1

In 1965, Gordon Moore published his observation that the number of transistors in an integrated circuit doubles about every 18 months [1]. Based this observation, the computer architecture community predicted that the CPU performance improvement would also follow the same pace, and called the observation as "Moore's Law". Tracing back the history, we have recognized that the CPU performance improvement has not exactly followed the Moore's Law but in its trajectory [2]. For example, from 1970s to the middle 1980s, the processor speed was doubled every two and half years. In a long period of 20 years from the middle of 1980s to the middle of 2000, the processor speed did accurately follow the Moore's Law to have been doubled every 18 months. After then, the speed improvement has started to drop. From middle 2000s to 2010, the CPU speed is doubled every 3.5 years, and in the next 5 years, it was doubled every 6 years. Since 2015, the CPU performance has only improved 3% every year, which is equivalent to double the speed in 20 years. Besides reaching the physical limit of the number of transistors that can be placed on a small chip, the CPU performance has also been limited by power consumption. In 1974, Robert Dennard and his colleagues at IBM recognized the power effectiveness potential in transistors: as transistors get smaller on a chip, their power consumption per unit area remains constant, which is called Dennard Scaling. Both Moore's Law and Dennard Scaling give a solid engineering basis and opportunities of CPU performance improvement without a power concern for a long period. The Dennard Scaling stopped in 2000, and 20 years after, we are preparing for the ending of Moore's Law.

Systems software including operating systems and other infrastructure software on top of the architecture had been well developed in the Moore's Law era. Reviewing the structure of the established ecosystem, we strongly believe the designing principle of the ecosystem is standardization and unification first, and performance second. Instead of controlling the key hardware components, such as ALU and memory, in a customized way, this critical control component becomes a unified abstract model, referred as instruction set architecture (ISA), which serves as the direct interface between software and hardware. On top of ISA, another computing abstraction is established, which is the operating system (OS). The OS supports a simple and one-size-fit-all environment to users of many different types. In a conventional OS environment, any computing task is divided into a sequence of executable subtasks. Under a multiprogramming model, OS handles multiple tasks, and assigns a computing time unit to each subtask of a task alternatively. Data blocks are moved around in the hardware memory hierarchy. Due to the standardization of ISA and

Email: zhang@cse.ohio-state.edu

OS, program execution is independent of application types. In addition, programming model is also standardized, which is even more architecture and system independent. In reality, after one expresses his/her ideas to a computer in simple but regulated English, such as Java or Python; computer will quickly return computing results. This highly general-purpose computing ecosystem has attracted billions of users to advance all the fields in the human society. Moore's Law is the driving force of building this ecosystem, which had given us luxury to tolerate the inefficiency in the software design. As Moore's Law is ending, the inefficiency in both unified software and hardware has started to affect increasingly more applications in this well-established computing ecosystem consisting of many commodity components in a deep software stack.

"Reforming and opening" in the existing computing ecosystem is timely desirable. This study will address several related issues with author's perspectives.

Critical issues in existing ecosystems. In order to answer the questions of "what we are reforming, and what we are opening", we identify several critical issues in the existing computing ecosystems.

- **Kernel-centric OS management is not sustainable.** OS kernel has been designed to control everything, thus CPU is frequently interrupted to handle many requests. Moore's Law supported this design, because CPU has a plenty of cycles to do a lot of extra work besides normal computation. This competitive advantage is decreasing for two reasons. CPU speed improvement pace is very slow now, and in-memory computing and NVM technology force CPU to process data very frequently. Performance data shows that the post Moore's Law era is also an era of CPU-IO performance inversion, where CPU becomes a bottleneck [3].

- **Memory technology advancement gradually flattens the memory hierarchy.** In-memory computing is a common practice now in many applications, such as in-memory data analytics and in-memory databases. Industries have made efforts to narrow the performance gap between DRAM and SSD by adding a new layer called persistent memory (PMEM). This middle layer connecting DRAM and SSD has a larger capacity and lower cost than that of DRAM, but higher performance than that of SSD. Most importantly, PMEM is byte addressable, which is an important step to reach a convergence of memory and storage [4]. Moore's Law made memory performance in both capacity and speed increasingly lag behind the CPU performance. Thus, a deep memory hierarchy is built, and the OS memory management for both DRAM and storage has been designed accordingly. This part of the memory system will have to be reconstructed in the post Moore's Law era.

- **General-purpose computing is very flexible for users but not efficient in computing.** Moore's Law makes an architecture- and system-independent interface available so that billions of users easily use computers. However, this general-purpose computing ecosystem is built at the cost of inefficiency of computing resource usage of both hardware and software. As Moore's Law is ending, many users start to have performance concerns in their applications. Instead, they are using special computing devices for their applications, such as GPU, FPGA, or even ASIC (application specific integrated circuits). These devices are highly efficient for certain domain applications, but not very flexible.

- **Existing ecosystem is non-inclusive to specialized devices.** Increasingly more researchers and practitioners use specialized devices to accelerate their applications for high performance and high efficiency. However, these highly efficient devices are not in the management scope of the existing computing ecosystem that has been designed for general-purpose CPU chips only. There are three reasons for this isolation. First, their programming environments are very different. Specialized programing is required for specialized devices. Second, existing ecosystem does not support unique execution models, such as SIMT (single instruction multiple threads) for GPU, customer-designed execution streams for FPGA and ASIC. Finally, the management of the computing ecosystem does not include these devices. For example, OS does not manage GPU and other devices. Moreover, these devices do not have their own operating systems.

Evolution of the computing ecosystem. System and architecture researchers have recognized the existing system must be reformed to respond the end of the Moore's Law, and have made two major efforts.

- **Weakening the central control of OS by delegating its powers to other system components.** Recognizing that kernel-based OS management is becoming a bottleneck; researchers have reconstructed the ecosystem by delegating certain OS powers to other system components. For example, RDMA (remote direct memory access) is a widely used facility to allow a direct memory access from a remote computer to another without either's OS getting involved, which greatly reduces CPU's burden on processing too many requests [5]. Another effort is to push computing close to data. The firmware in high-end SSD today is increasingly powerful with sophisticated computing capability including coding, decoding, compression and decompression and others, which were the jobs of centralized CPU.

Tracing back the history of human society, we recognize that the evolution of the computing system follows a similar pattern. In the Chinese Qin Dynasty more than 2000 years ago, Prime Minister Li Si suggested a kernel-based rule to Emperor Qin: only the Emperor can make the final judgement and decisions to control the country. His 4-charcter concise expression is also a well-known idiom in the Chinese literature. This kernel-based principle lays a solid foundation in the Chinese political system for many years. On the other hand, in the modern world history, French philosopher and thinker de Montesquieu published his book entitled “The Spirit of the Law”, where he first presented the concept of “separation of powers”. This concept has been widely applied to many countries in the world for their governance of public powers. The computer systems community is following the direction of separation of powers to reconstruct the computing ecosystem in the post Moore’s Law era [6].

• **Accelerators and domain-specific architecture.** In the transition of between general-purpose computing ecosystem and a heterogeneous computing ecosystem, researchers have made efforts in two directions. One is to utilize accelerators, such as GPU and FPGA, significantly raising the efficiency and performance of applications. The growth of both GPU and FPGA communities is strong, where programming environment, application libraries and system tools are developed. Another direction is to build domain-specific architecture. A representative example is the tensor-processing unit (TPU), an ASIC for machine learning via neural networks, which is developed by Google. The efforts in the two directions aim to achieve the same goal, namely, to best utilize the hardware and software resources for different domains of applications based on their unique execution and data access patterns.

Again, tracing back the history, domain-specific approach has been applied for many years in production systems. In 1776, British economist Adam Smith published his book of “The Wealth of Nations”, where he proposed the concept of specialization in economics. Here is a classical paragraph in his book: “Economic growth is rooted in the increasing division of labor, which is primarily related to the specialization of the labor force, essentially the breaking down of large jobs into many tiny components. Each worker becomes an expert in one isolated area of production, increasing his efficiency.”

Conclusion. Our existing ecosystem must be reconstructed for the end of the Moore’s Law. The reforming of the ecosystem is in the direction of separation of powers and the opening of the ecosystem is to include many different specialized devices. In 1913, Henny Ford divided his T-model car into 8772 units of specialized work, and built the first assembly line in the world to make cars. This assembly line greatly improved the productivity and lowered the price of the automobile products. Post Moore’s Law era is also a computing specialization era; however, we will have a long way to go towards building an inclusive assembly line for all the specialized devices.

References

1 Moore G E. Cramming more components onto integrated circuits. Electronics, 1965, 38: 114

2 Hennessy J L, Patterson D A. A new golden age for computer architecture. Commun ACM, 2019, 62: 48–60

3 Waddington D, Harris J. Software challenges for the changing storage landscape. Commun ACM, 2018, 61: 136–145

4 Moore S K. Researchers invent a way to speed Intel’s 3D XPoint computer memory. IEEE Spectrum, 2018. https://spectrum.ieee.org/tech-talk/

5 Dragojevic A, Narayanan D, Hodson O, et al. FaRM: fast remote memory. In: Proceedings of 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI’14), Seattle, 2014

6 Zhang S, Xiao M, Guo C, et al. HYPHA: a frame work on separation of parallelisms to accelerate persistent homology matrix reduction. In: Proceedings of the 33rd ACM International Confenrence on Supercomputing (ICS’19), Phoenix, 2019