1 Computing and the Transformation of Society

This book is for the growing community of scientists and even engineers who use computing and need a scientific understanding of computer architecture – those who view computation as an intellectual multiplier, and consequently are interested in capabilities, scaling, and limits, not mechanisms. That is, the *scientific principles* behind computer architecture, and how to reason about hardware performance for higher-level ends. With the dramatic rise of both data analytics and artificial intelligence, there has been a rapid growth in interest and progress in data science. There has also been a shift in the center of mass of computer science upward and outward, into a wide variety of sciences (physical, biological, and social), as well as nearly every aspect of society.

For these audiences, the book provides an understanding and the scientific principles that describe the unique scaling that enabled computing to improve over a billion-fold in performance, cost, and size. It frames the fundamental characteristics of computing performance today, and how it can be tapped for applications. Further, it provides insights into the constraints that limit further improvement of computing, and likely directions for future advances. In short, it frames where computing hardware is going in the coming decades.

In this chapter, we survey the broad and pervasive impact of computing on society, and discuss computing's unique characteristics that have driven its proliferation. We describe the four pillars that account for the remarkable improvement of computers, and discuss expectations for readers as well as specifics of the book's organization.

1.1 Computing Transforms Society and Economy

It is no exaggeration to say computers are endemic, present in a staggering breadth of human-made devices, and used to deliver many of the new innovations and capabilities that drive our society and economy forward. Through the 1980s and 1990s it became common for professionals in the developed world to have *personal computers*. In the past decade, since the introduction of the smartphone by Apple in 2007, it has become common for not only professionals, but all adults and even children to have *smartphones*. These device categories represent 250 million, and one billion units sold per year, respectively. The cloud, for all its glamour, is a relatively small number

of processors, accounting for tens of millions of server sales each year. However, these evident computers, while important, are but a small fraction of computing in the world today.

Embedded computers have long accounted for the largest numbers of computers in the world. For decades these computers have been used to implement fixed functions (thermostats, engine timing, factory machines, point of sale terminals, etc.), doing so in a cost-effective and compact fashion for the need at hand. These systems use computers to implement control systems, communication protocols, and in some cases user interfaces. Since 2010, as embedded computers with integrated networking and storage have become quite inexpensive, driven by the explosive growth of the smartphone ecosystem, momentum has built behind the vision of the "Internet of Things" (IoT). The IoT can be more precisely described as embedded networked computers [69] and is expected to grow to 40 billion devices in 2025. These devices combine low-cost computing in everyday devices such as motion detectors, video cameras, doorbells, smoke detectors, hotel door locks, automobile keys, stereo speakers, holiday lights, smart TVs, public trash cans, and more with network access that enables intelligent data processing, smart group analytics, and in-field upgrades. Together, these form a potent combination for new services, and acceleration of the data economy.

To get a sense of how pervasive computing is, here we consider several dimensions of its ubiquity.

1.1.1 Home

When personal computers became cheap enough in the late 1980s for individuals to afford them, affluent families began to buy home computers. As laptops, a standard package of computer, display, and Wi-Fi networking, became widely available in the early 2000s families might have had several computers. In today's home, a collection of smartphones, tablets, laptops, and more is typical. As for embedded computers, a modern collection might include smart keys, thermostats, smart TVs, intelligent voice assistants, wireless routers, smart appliances, security cameras, and smart locks, just to name a few. Each of these devices includes computing, storage, and networking – for amplified capability through the cloud. Today, a family of four might have 50 or more such computing devices in their home.

1.1.2 Automobiles and Transportation

Similarly, the automobile has progressed through phases of computing adoption. Early uses included engine timing control and anti-lock brakes (traction control), both applications of high-speed monitoring and response. From there, applications exploded into a broad range of areas ranging from cabin heating and cooling, lighting, turn signals, windshield washers, wireless locking, and more recently USB power, as well as entertainment and navigation systems. An automobile generally depends on more than 50 computers, and the recent addition of wireless connectivity has created new "networked" features such as LoJack, Onstar, and Tesla's software updates. For more

than two decades computing systems have accounted for the vast majority of new features in cars, and despite the low cost of electronics, a growing fraction of the vehicle cost.

With the advent of electrically powered cars, the adoption of computing in cars is deepening and accelerating. Per-car navigation and efficient dispatch has given rise to ridesharing companies such as Uber, Didi, and Lyft, which dispatch millions of drivers for billions of rides each month! With public-transit tracking and sensors on buses and trains combined with cloud-based aggregation/integration, commuters who choose not to drive can optimize their trips. These moving sensors allow cloud services to infer dynamic traffic maps, and they are used by drivers and logistics companies to optimize delivery, pickup, and more. The dream of self-driving vehicles is built on the integration of sensors, artificial intelligence (AI), and large quantities of computation (>100 tera-operations/second computing power) in cars and trucks. This growth has been remarkable, and we're likely to see more and more!

1.1.3 Commerce

There are many examples where computing has revolutionized commerce, ranging from finance to retail to logistics to manufacturing. Let's describe retail as a relatable example – everyone shops. Computing has totally transformed the retail process. We depict elements of computing in retail and more in Figure 1.1. Here's one slice through it!

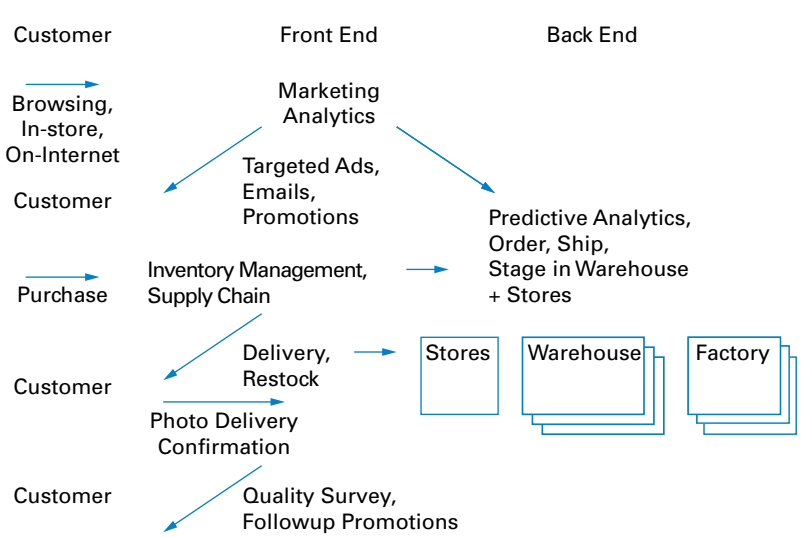

Figure 1.1 Examples of computing applied to retail, supply chain, and manufacturing. Each box includes both computation and extensive sensing (location, scans), as well as large-scale data collection and analytics.

- **Marketing:** Products are extensively advertised through social media, web ads, and emails. In virtually every online activity we have become accustomed to the idea that ads will be placed next to whatever we are doing. These ads are clickable and transition directly to web pages to buy the product.
- **Purchasing:** Buying products online, once an oddity, is now a mainstream method of shopping. With online shopping comes delivery, from services such as Amazon Prime, Instacart, Grubhub, and Walmart Delivery surpassing 10 percent of all retail in 2019. The online purchasing process is made possible by computing and the Internet.
- **Tracking:** Both before purchase (stock-check) and while in transit, buyers can track the location of their purchase, following it from the warehouse, to their city, and then en route to their home. These capabilities expose information from the store's inventory system and the delivery service's internal logistics system – both complex computing applications. And the end result is continued engagement, anticipation, and diagnosis of any mix-ups all the way to your door.
- **Delivery:** The hardest step is often to find the customer's front door, and delivery services employ global-positioning service (GPS) tracking, per-address notes, and time-of-day customized logistics, all implemented as computing systems, to not only get the purchase to the customer, but to report the package's delivery with a photo – sent to your smartphone. This ensures your immediate awareness, and also documents the delivery of the product to you.

While we have focused on the consumer retail process, many other dimensions of commerce, such as supply chains, billing, and finance, have been equally transformed. Each e-commerce purchase transaction triggers a collection of follow-on activities, all of which are enabled by the low cost and extraordinary speed of computing.

For example, a purchase triggers artificial intelligence (AI) applications. A purchase updates customer classifiers, and these classifiers drive predictors used to target advertising. Many e-commerce sites also host customer reviews singing the praises (or excoriating the flaws) of products. One way to think of this is as computerized and online "word of mouth." We can easily search for reviews of a particular product – and its competitors – as well as find the most positive and negative reviews – out of a sea of thousands. All of this is provided courtesy of fast, inexpensive computing.

Another dimension of the purchasing process is payment. With credit/debit cards and even electronic payment apps on smartphones, customers can pay "with a tap," or in the case of e-commerce "with a click." Charges are customized for local taxes, shipping, and personalized reward programs before they are charged against a credit account or directly removed from our bank account. Purchase histories are tabulated against complex loyalty programs, from total use (membership miles) to customized promotions (2 percent off on gasoline or grocery stores), and even temporary promotions (2 percent off on clothing this month only). This dizzying array is beyond the ability of most consumers to keep straight, and the retailers can only manage with complex computer applications that seek to optimize revenue and profit against products, inventory, and customers! One more important dimension is the supply

chain, which triggers manufacture and retailer purchase of items in response to a sale. That supply chain may include multiple parties, span 15,000 kilometers, and weeks of latency.

1.2 Computing Transforms Science and Discovery

Science and discovery are about the creation of new knowledge and understanding, so they are fueled by tools that speed information access, analysis, and propagation. Consequently, computing has transformed them, and it's now unthinkable to pursue discovery without computing as a fundamental tool. We outline several specific dimensions below:

- **Finding information:** The growth of the Internet and the power of text-based search enables scientists to find relevant information and recent discoveries from around the world. Increasingly sophisticated indexing enables automated filtering and notification of potentially relevant advances.
- **Disseminating knowledge:** Electronic publication combined with rapidly expanding open-access publications and innovations such as arXiv, and digital libraries allow new insights to rapidly spread around the world. With growing government-led requirements for open access to government-funded scientific research fueling growing global access, new discoveries are propagated both faster and more widely than ever before.
- **Scientific modeling:** Some of the earliest computers modeled the flight of projectiles. Another major task was fluid dynamics – flows and explosions – and soon thereafter, nuclear physics. Today, computational modeling at the molecular level underlies drug design, materials design, nanotechnology, advanced electronics, and more. Climate change, epidemiology, and ecology are all studied with computational models. Increasingly, computational modeling is used for human behavior, modeling psychology, economics, and more.
- **Analytics:** Computer simulations are used extensively for scientific and engineering modeling. The explosion of data from internet usage, rich sensors, video, smartphones, and the IoT enables "digital twins" that track the real world. With terabytes of data, large-scale computing is required for analysis, including precise analysis to extract even small signals. For example, analysis to understand the spread of the virus that causes COVID-19, and the effectiveness of antivirals and vaccines, or the health impacts of the widespread economic disruption caused by it. Bioinformatics and clinical data are used to understand its evolution and spread (analysis of its lineage).

Despite the broadly transformative impact of computing on society, economy, science, and discovery, few people understand why such a transformation has been possible. This book explains the science behind why it has been possible, how it came to pass, and where it is going!

1.3 Extraordinary Characteristics of Computing

How has this radical transformation been possible? The proliferation of computing into every sector of society and economy has been enabled by the central importance of information – for intelligent behavior, control, and organization. Access to up-to-date and detailed information is well understood to be a prerequisite for success in an increasingly fast-paced, competitive, and global environment. So industry has been a key driver for ambitious use of computing, but recent developments have demonstrated that individuals are equally interested in up-to-date and detailed information to drive their personal connection, enjoyment, and even social status.

Computing technology has made remarkable advances over the past 75 years since the pioneering digital electronic computer, ENIAC, in 1945 [104], delivering dramatic improvements in cost, size, power, and performance. These advances are explained below:

- **Cost:** The original ENIAC system cost $487,000 (about $7 million in 2019 dollars), in contrast to today's ARM Cortex-M3 microprocessor, which costs a few cents. Adjusted for inflation, today's cost of computing is 1/350,000,000 that of 75 years ago, not even taking into account the much higher performance of the modern processor. This cost reduction is primarily due to miniaturization and high volumes of computers sold.
- **Size:** The original ENIAC system occupied 2,350 cubic feet (65 m^3) of space, filling a 50-foot long basement room. In contrast, a small microprocessor today occupies the area of 0.11 mm^2 (silicon) and is 1.1×10^{-9} m^3 in volume. This area corresponds to the cross-section of about 10 human hairs. Thus, today's microprocessor occupies a space approximately 1/60,000,000,000 of the ENIAC system.
- **Power:** The original ENIAC system consumed 150 kW of power; in contrast, the Cortex-M3 processor is dramatically more energy efficient, consuming less than 5×10^{-6} watts. This means that today's processor has a power requirement less than 1/30,000,000,000 that of the original ENIAC 75 years ago.
- **Performance:** The original ENIAC system could complete approximately 5,000 operations per second, whereas the Cortex-M3 is approximately 50,000 times faster in computation. Today's fastest microprocessors achieve single-thread performance of about 10 billion instructions/second, which translates to a performance increase of nearly 2 million times compared to 75 years ago.

Overall, the performance of a single computer – at first a collection of racks but now down to a single silicon chip – has increase by a factor of a trillion, 10^{12}, as illustrated in Figure 1.2. These remarkable advances have enabled the widespread adoption of computers and computation to replace and improve the capabilities of our modern products and systems. These million- and billion-fold improvements have allowed computing systems to acquire new functionalities (features!) and a staggering complexity of software systems (hundreds of millions of lines of code across iOS and

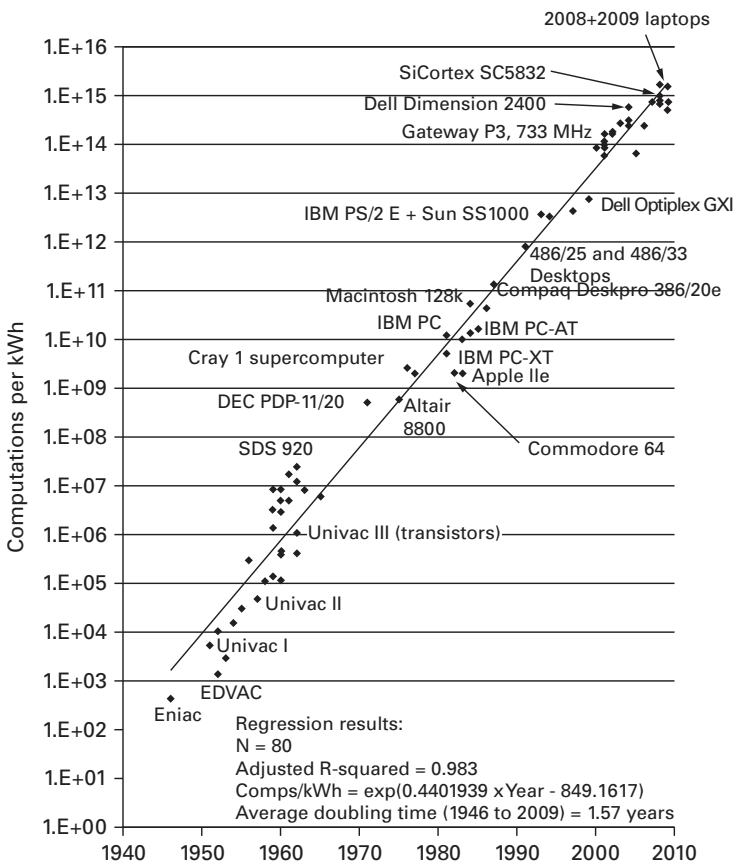

Figure 1.2 Increase in the power/speed of computing over the past seven decades.

Android). These codebases depend on the modern advanced hardware capabilities to deliver interactive applications and new features in mobile and IoT devices.

The decreased cost of computers has also led to the rise of cloud computing, where millions of processors are combined to do global-scale computations. These large assemblies of computers enable applications of unthinkable scale, such as internet search and global social networks. Further, their ability to efficiently share resources among applications allows a further drop in the cost of computing.

Beyond enabling large, complex systems, the falling cost of computing, combined with its widespread availability, has driven the growth of a vibrant, international computer science community. That community of hundreds of thousands of researchers and millions of computing professionals has driven a broad advance in fundamental topics such as theory and algorithms, programming languages and compilers, software systems as we have mentioned, and a dizzying variety of applications of computing. For example, there are currently over three million software applications available in the iOS and Android mobile application stores. And these advances also fuel growing excitement in AI and statistical machine learning.

1.4 What is Computer Architecture?

Computer architecture is the science and engineering that underpins the design of computer hardware. It is both the design of the software–hardware interface that defines portability and correct execution for software, and also the organizational understanding for hardware that effectively exploits advancing microelectronics technology for increased computing performance. For this reason, computer organization is another term used for computer architecture. These forces are depicted in Figure 1.3.

Because computers are rendered useful by software programs, the flexible support of software applications, and thereby well-developed uses of computing, is critical. As a result, existing software (applications and operating systems) acts as a conservative force on computer architecture, constraining its rate of change and limiting advancement of its performance. One manifestation of this is that computer architectures typically provide **binary compatibility**, the ability to run programs written for older versions of the architecture. On the other hand, new applications often demand new computing structures and efficiencies, driving higher rates of change, and in some cases radical incompatibility. A good example of this is the new computing paradigm created by deep learning (aka deep neural networks, or more broadly AI and machine learning), in which arithmetic operation density and efficiency is favored over flexible data structures and control. A 2019 article noted over 100 venture-funded AI hardware startups – all pursuing approaches incompatible with traditional CPU software [93].

Beneath these tensions for software portability, legacy software, new applications, and new software, all computer performance depends on organizing the work of executing software programs, and the corresponding blocks of hardware to achieve high performance. This gives rise to the basic structure of computers, covered in Chapter 2. But beyond a basic organizational understanding, computer architecture is driven forward by the rapid advance of hardware microelectronics technology that has provided

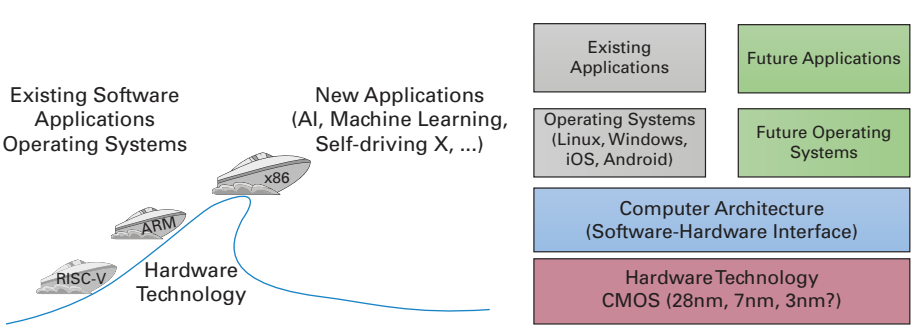

Figure 1.3 Computer architecture supports existing and future applications by providing a stable interface for software and increasing performance. Performance is driven by advances in hardware technology and in computer architecture understanding of how to organize systems, notionally (left) and formally (right). Beneficiaries of these advances are future applications – and you!

dramatically faster devices, larger numbers of devices, and lower-cost devices over the past seven decades. When building blocks change in these three dimensions by factors of one thousand, one million, and in some cases one billion, qualitative changes in computer organization become not only possible, but profitable in terms of performance. Thus, computer architecture also involves the invention of new algorithmic techniques (implemented in hardware), as well as new organizational elements that shape computer organization. We will discuss the basic trends and most important insights that enable programmers of today to efficiently exploit billions of transistors with the ease of a high-level Python or JavaScript program.

> Principle 1.1: The Role of Computer Architecture
>
> *Computer architecture enables software portability and determines the performance properties of software applications.*

1.5 Four Pillars of Computer Performance: Miniaturization, Hidden Parallelism, Dynamic Locality, and Explicit Parallelism

The development of computing technology has created advances in computing performance of ten billion billion-fold (10^{19} times) over the past seven decades. This unprecedented increase has been achieved based on four major pillars and thousands of smaller innovations (see Figure 1.4). These four pillars are:

- **Miniaturization:** Riding the rapid shrinking of electronics to smaller sizes, computers benefited in speed, power, cost, and size.
- **Hidden parallelism:** An important technique used to manage complexity in computer software the sequential abstraction (computing operations are ordered). Employing growing numbers of inexpensive transistors, computer architects invented techniques that execute a program's instructions in parallel while preserving software's key tool for reasoning – sequential abstraction. These hidden parallelism techniques significantly increase performance.

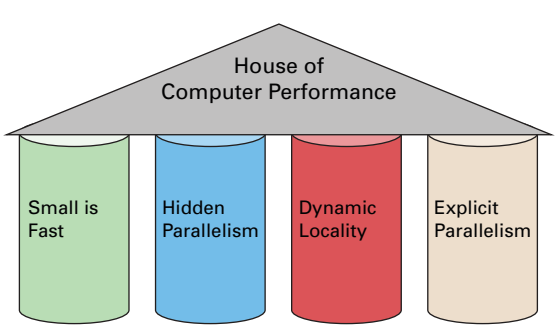

Figure 1.4 The four pillars that provide the extraordinary performance of computers are miniaturization, hidden parallelism, dynamic locality, and explicit parallelism.

- **Dynamic locality:** Software applications' growing complexity demands memory capacity increases, but large memories can only be accessed at low speed. Computers finesse this paradox by exploiting dynamic locality, making small, transient sets of data accessible at high speed, and efficiently guessing which are most important to the application.
- **Explicit parallelism:** Beyond the limits of a single processor (core), growing computing requirements are met by employing millions of cores in both supercomputers and the cloud. The primary drawback of explicit parallelism is the loss of the sequential interface to reason about computing correctness and performance. However, software has developed new application-structuring techniques to stretch the use of sequential reasoning and make ever-larger applications possible.

These pillars reflect the most important dimensions to understand how computers have accomplished their extraordinary advances in both performance and efficiency. We use these four pillars as the organizing structure for the book.

1.6 Expected Background

This book is intended for three growing communities, all of whom share an interest in understanding how computing's unique capabilities arose, including its real limitations and likely vectors of change in the future. First, there are the computer scientists pursuing careers in high-level software, large-scale applications, and even mobile and interactive systems. Second is the growing community of data scientists who use computing and the growing sea of data to analyze scientific phenomena, economies, human individual or social behavior, and every aspect of business to understand, shape, and optimize it. Third is anyone who uses computing to empower their creativity, business, or life who seeks a deeper understanding of how computing's remarkable capability came about – and insights into how it continues to evolve.

We present computer architecture concepts, not engineering, with a scientific approach. The focus is on principles and concepts that feed understanding and can be used for extrapolation. We use programs as examples of computing structure and common industry products as hardware examples. We explain material with analogies and illustrations, often tying computing's capabilities or limitations to fundamental scientific principles from physics or mathematics. Helpful background for readers includes:

- experience with scientific principles and concepts (high school or college physics);
- basic exposure to code – able to read code snippets that describe computation; and
- experience in computing for analysis, applications, or more.

While computer architecture necessarily touches on hardware technology, we have assumed no technology or hardware design background. Further, the approach takes a principled scientific approach, avoiding the need for engineering hardware knowledge and terminology.