Historical Perspective and Further Reading

There is a tremendous amount of history in multiprocessors; in this section, we divide our discussion by both time period and architecture. We start with the SIMD approach and the Illiac IV. We then turn to a short discussion of some other early experimental multiprocessors and progress to a discussion of some of the great debates in parallel processing. Next we describe the historical roots of the present **multiprocessors** and conclude by discussing recent advances.

SIMD Computers: Attractive Idea, Many Attempts, No Lasting Successes

> *The cost of a general multiprocessor is, however, very high and further design options were considered which would decrease the cost without seriously degrading the power or efficiency of the system. The options consist of recentralizing one of the three major components.… Centralizing the [control unit] gives rise to the basic organization of [an] … array processor such as the Illiac IV.*
>
> Bouknight et al. [1972]

The SIMD model was one of the earliest models of parallel computing, dating back to the first large-scale multiprocessor, the Illiac IV. The key idea in that multiprocessor, as in more recent SIMD multiprocessors, is to have a single instruction that operates on many data items at once, using many functional units (see Figure e6.16.1).

Although successful in pushing several technologies that proved useful in later projects, it failed as a computer. Costs escalated from the $8 million estimate in 1966 to $31 million by 1972, despite construction of only a quarter of the planned multiprocessor. Actual performance was at best 15 MFLOPS, versus initial predictions of 1000 MFLOPS for the full system [Hord, 1982]. Delivered to NASA Ames Research in 1972, the computer required three more years of engineering before it was usable.

These events slowed the investigation of SIMD, with Danny Hillis [1989] resuscitating this style in the Connection Machine, which had 65,636 1-bit processors.

Real SIMD computers need to have a mixture of SISD and SIMD instructions. There is an SISD host computer to perform operations such as branches and address calculations that do not need parallel operation. The SIMD instructions are broadcast to all the execution units, each of which has its own set of registers. For flexibility, individual execution units can be disabled during an SIMD instruction. In addition, massively parallel SIMD multiprocessors rely on interconnection or communication networks to exchange data between processing elements.

FIGURE e6.16.1 The Illiac IV control unit followed by its 64 processing elements. It was perhaps the most infamous of supercomputers. The project started in 1965 and ran its first real application in 1976. The 64 processors used a 13-MHz clock, and their combined main memory size was 1 MB: 64 × 16 KB. The Illiac IV was the first machine to teach us that software for parallel machines dominates hardware issues. *Photo courtesy of NASA Ames Research Center.*

The basic tradeoff in SIMD multiprocessors is performance of a processor versus the number of processors. More recent SIMDs emphasize a large degree of parallelism over performance of the individual processors. The Connection Multiprocessor 2, for example, offered 65,536 single-bit-wide processors, while the Illiac IV had sixty-four 64-bit processors.

After being resurrected in the 1980s, originally by Thinking Machines and then by MasPar, the SIMD model has once again been put to bed as a general-purpose multiprocessor architecture, for two main reasons. First, it is too inflexible. A number of important problems cannot use such a style of multiprocessor, and the architecture did not scale down in a competitive fashion; that is, small-scale SIMD multiprocessors often had worse cost performance than that of the alternatives. Second, SIMD did not take advantage of the tremendous performance and cost

advantages of microprocessor technology. Instead of leveraging this low-cost technology that was improving rapidly during the height of Moore's Law and Dennard Scaling, designers of SIMD multiprocessors built custom processors for their multiprocessors.

Although SIMD computers have departed from the scene as general-purpose alternatives, this style of architecture plays an important role in special-purpose designs. Many special-purpose tasks are highly data parallel and require a limited set of functional units. Thus, designers can build in support for certain operations, as well as hardwired interconnection paths among functional units. Such organizations are often called array processors, and they are useful for tasks like image processing, signal processing, and machine learning, as we see with TPUs.

Multimedia Extensions as SIMD Extensions to Instruction Sets

Many recent architectures have laid claim to being the first to offer multimedia extensions, in which a set of new instructions takes advantage of a single wide ALU that can be partitioned so that it will act as several narrower ALUs operating in parallel. It's unlikely that any appeared before 1957, however, when the Lincoln Lab's TX-2 computer offered instructions that operated on the ALU as either one 36-bit operation, two 18-bit operations, or four 9-bit operations. Ivan Sutherland, considered the Father of Computer Graphics, built his historic Sketchpad system on the TX-2. Sketchpad did, in fact, take advantage of these SIMD instructions, despite TX-2 appearing before invention of the term SIMD in 1972.

Other Early Experiments

It is difficult to distinguish the first MIMD multiprocessor. Surprisingly, the first computer from the Eckert-Mauchly Corporation, for example, had duplicate units to improve **availability**.

Two of the best-documented multiprocessor projects were undertaken in the 1970s at Carnegie Mellon University. The first of these was C.mmp, which consisted of 16 PDP-11s connected by a crossbar switch to 16 memory units. It was among the first multiprocessors with more than a few processors, and it had a shared memory programming model. Much of the focus of the research in the C.mmp project was on software, especially in the OS area. A later multiprocessor, Cm*, was a cluster-based multiprocessor with a distributed memory and a nonuniform access time. The absence of caches and a long remote access latency made data placement critical. Many of the ideas in these multiprocessors would be reused in the 1980s, when the microprocessor made it much cheaper to build multiprocessors.

Great Debates in Parallel Processing

The turning away from the conventional organization came in the middle 1960s, when the law of diminishing returns began to take effect in the effort to increase the operational speed of a computer.… Electronic circuits are ultimately limited in their speed of operation by the speed of light … and many of the circuits were already operating in the nanosecond range.

W. Jack Bouknight et al.
The Illiac IV System [1972]

… sequential computers are approaching a fundamental physical limit on their potential computational power. Such a limit is the speed of light …

Angel L. DeCegama
The Technology of Parallel Processing, Volume I [1989]

… today's multiprocessors … are nearing an impasse as technologies approach the speed of light. Even if the components of a sequential processor could be made to work this fast, the best that could be expected is no more than a few million instructions per second.

David Mitchell
The Transputer: The Time Is Now [1989]

The quotes above give the classic arguments for abandoning the current form of computing, and Amdahl [1967] gave the classic reply in support of continued focus on the IBM 360 architecture. Arguments for the advantages of parallel execution can be traced back to the 19th century [Menabrea, 1842]! Aside from these debates about the advantages and limitations of parallelism, several hot debates have focused on how to build multiprocessors.

From today's perspective, it is clear that the speed of light was not the brick wall; the brick wall was, instead, the power consumption of CMOS as the clock rates increased.

It's hard to predict the future, yet in 1989 Gordon Bell made two predictions for 1995. We included these predictions in the first edition of *Computer Architecture: A Quantitative Approach*, when the outcome was completely unclear. We discuss them in this section, together with an assessment of the accuracy of the prediction.

The first was that a computer capable of sustaining a tera FLOPS—one million MFLOPS—would be constructed by 1995, using either a multicomputer with 4K to 32K nodes or a Connection Multiprocessor with several million processing elements. To put this prediction in perspective, each year the Gordon Bell Prize acknowledges advances in parallelism, including the fastest real program (highest MFLOPS). In 1989, the winner used an eight-processor Cray Y-MP to run at 1680 MFLOPS. On the basis of these numbers, multiprocessors and programs would had to have improved by a factor of 3.6 each year for the fastest program to achieve 1 TFLOPS in 1995. In 1999, the first Gordon Bell prize winner crossed the 1 TFLOPS bar.

Using a 5832-processor IBM RS/6000 SST system designed specially for Livermore Laboratories, they achieved 1.18 TFLOPS on a shock wave simulation. This ratio represents a year-to-year improvement of 1.93, which is still quite impressive.

What has been recognized since the 1990s is that although we may have the technology to build a TFLOPS multiprocessor, it was not clear that the machine was cost-effective, except perhaps for a few very specialized and critically important applications related to national security. We estimated in 1990 that achieving 1 TFLOPS would require a machine with about 5000 processors and would cost about $100 million. The 5832-processor IBM system at Livermore cost $110 million. As might be expected, improvements in the performance of individual micro-processors both in cost and performance directly affect the cost and performance of large-scale multiprocessors, but a 5000-processor system would cost more than 5000 times the price of a desktop system using the same processor. Since that time, much faster multiprocessors have been built, but the major improvements have increasingly come from the recent processors, rather than fundamental breakthroughs in parallel architecture.

The second Bell prediction concerned the number of data streams in supercomputers shipped in 1995. Danny Hillis believed that although supercomputers with a small number of data streams might be the best sellers, the biggest multiprocessors would be multiprocessors with many data streams, and these would perform the bulk of the computations. Bell bet Hillis that in the last quarter of calendar year 1995, more sustained MFLOPS would be shipped in multiprocessors using few data streams (<100) rather than many data streams (>1000). This bet concerned only supercomputers, defined as multiprocessors costing more than $1 million and used for scientific applications. Sustained MFLOPS was defined for this bet as the number of floating-point operations per month, so availability of multiprocessors affects their rating.

In 1989, when this bet was made, it was totally unclear who would win. In 1995, a survey of the current publicly known supercomputers showed only six multiprocessors in existence in the world with more than 1000 data streams, so Bell's prediction was a clear winner. In fact, in 1995, much smaller microprocessor-based multiprocessors (<20 processors) were becoming dominant.

In 1995, the Top 500 survey of the 500 highest-performance multiprocessors showed that the largest number of multiprocessors were bus-based shared memory multiprocessors! By 2005, various clusters or multicomputers played a large role. For example, in the top 25 systems, 11 were custom clusters, such as the IBM Blue Gene system or the Cray XT3, 10 were clusters of shared memory multiprocessors (both using distributed and centralized memory), and the remaining four were clusters built using PCs with an off-the-shelf interconnect.

More Recent Advances and Developments

With the primary exception of the parallel vector multiprocessors and more recently of the IBM Blue Gene design, all other modern MIMD computers have been built from off-the-shelf microprocessors using a bus and logically central memory or an interconnection network and a distributed memory. A number of experimental multiprocessors built in the 1980s further refined and enhanced the concepts that form the basis for many of today's multiprocessors.

The Development of Bus-Based Coherent Multiprocessors

Although very large mainframes were built with multiple processors in the 1960s and 1970s, multiprocessors did not become highly successful until the 1980s. Bell [1985] suggests the key was that the smaller size of the microprocessor allowed the memory bus to replace the interconnection network hardware and that portable operating systems meant that multiprocessor projects no longer required the invention of a new operating system. In this paper, Bell defined the terms multiprocessor and multicomputer and set the stage for two different approaches to building larger-scale multiprocessors. The first bus-based multiprocessor with snooping caches was the Synapse N + 1 in 1984.

The early 1990s saw the beginning of an expansion of such systems with the use of very wide, high-speed buses (the SGI Challenge system used a 256-bit, packet-oriented bus supporting up to eight processor boards and 32 processors) and later the use of multiple buses and crossbar interconnects, for example, in the Sun SPARCCenter and Enterprise systems. In 2001, the Sun Enterprise servers represented the primary example of large-scale (>16 processors), symmetric multiprocessors in active use.

Toward Large-Scale Multiprocessors

In the effort to build large-scale multiprocessors, two different directions were explored: message-passing multicomputers and scalable shared memory multiprocessors. Although there had been many attempts to build mesh and hypercube-connected multiprocessors, one of the first multiprocessors to successfully bring together all the pieces was the Cosmic Cube built at Caltech [Seitz, 1985]. It introduced important advances in routing and interconnect technology and substantially reduced the cost of the interconnect, which helped make the multicomputer viable. The Intel iPSC 860, a hypercube-connected collection of i860s, was based on these ideas. More recent multiprocessors, such as the Intel Paragon, have used networks with lower dimensionality and higher individual links. The Paragon also employed a separate i860 as a communications controller in each node, although a number of users have found it better to use both i860 processors for computation as well as communication. The Thinking Machines CM-5 made use of off-the-shelf microprocessors. It provided user-level access to the communication channel, significantly improving communication latency. In 1995, these two multiprocessors represented the state of the art in message-passing multicomputers.

Clusters

Clusters were probably "invented" in the 1960s by customers who could not fit all their work on one computer, or who needed a backup machine in case of failure of the primary machine [Pfister, 1998]. Tandem introduced a 16-node cluster in 1975. Digital followed with VAX clusters, introduced in 1984. They were originally independent computers that shared I/O devices, requiring a distributed operating system to coordinate activity. Soon they had communication links between computers, in part so that the computers could be geographically distributed to increase availability in case of a disaster at a single site. Users logged on to the cluster and were unaware of which machine they are using. DEC (now HP) sold more than 25,000 clusters by 1993. Other early companies were Tandem (now HP) and IBM (still IBM). Virtually company has cluster products. Most of these products are aimed at availability, with performance scaling as a secondary benefit.

Scientific computing on clusters emerged as a competitor to MPPs. In 1993, the Beowulf project started with the goal of fulfilling NASA's desire for a 1-GFLOPS computer for less than $50,000. In 1994, a 16-node cluster built from off-the-shelf PCs using 80486s achieved that goal. This emphasis led to a variety of software interfaces to make it easier to submit, coordinate, and debug large programs or a large number of independent programs.

Efforts were made to reduce latency of communication in clusters as well as to increase bandwidth, and several research projects worked on that problem. (One commercial result of the low-latency research was the VI interface standard, which has been embraced by Infiniband, discussed below.) A low latency then proved useful in other applications. For example, in 1997 a cluster of 100 UltraSPARC desktop computers at U.C. Berkeley, connected by 160 MB/sec per link Myrinet switches, was used to set world records in database sort (sorting 8.6 GB of data originally on disk in 1 minute) and in cracking an encrypted message (taking just 3.5 hours to decipher a 40-bit DES key).

This research project, called Network of Workstations, also developed the Inktomi search engine, which led to a start-up company with the same name. Google followed the example of Inktomi to build search engines from clusters of desktop computers, rather than large-scale SMPs, which was the strategy of the leading search engine, Alta Vista, that Google took over. In 2020, all Internet services rely on clusters to serve their millions of customers.

Clusters are also popular with scientists. One reason is their low cost, which enables individual scientists or small groups to own a cluster dedicated to their programs. Such clusters can get results faster than waiting in the long job queues of the shared MPPs at supercomputer centers, which can stretch to weeks.

For those interested in learning more, Pfister [1998] has written an entertaining book on clusters.

Recent Trends in Large-Scale Multiprocessors

In the mid-to-late 1990s, it became obvious that the hoped-for growth in the market for ultralarge-scale parallel computing was unlikely to occur. Without this market growth, it became increasingly obvious that the high-end parallel computing

market was too small to support the costs of highly customized hardware and software designed for a small market. Perhaps the most important trend to come out of this observation was that clustering would be used to reach the highest levels of performance. There are now three general classes of large-scale multiprocessors:

1. Clusters that integrate standard desktop motherboards using interconnection technology, such as Ethernet or Infiniband
2. Multicomputers built from standard microprocessors configured into processing elements and connected with a custom interconnect, such as the IBM Blue Gene
3. Clusters of small-scale shared memory computers, possibly with vector support, including the Earth Simulator

Blue Gene is constructed using a custom chip that includes an embedded PowerPC microprocessor offering half the performance of a high-end PowerPC, but at a much smaller fraction of the area and the power. This allowed more system functions, including the global interconnect, to be integrated onto the same die.

Looking Further

There is an almost unbounded amount of information on multiprocessors and multicomputers: conferences, journal papers, and even books seem to appear faster than any single person can absorb the ideas. No doubt many of these papers will go unnoticed—not unlike the past. Most of the major architecture conferences contain papers on multiprocessors. An annual conference, SC XY (where X and Y are the last two digits of the year), brings together users, architects, software developers, and vendors and publishes the proceedings in book, CD-ROM, and online (see www.scXY.org) form. Two major journals, *Journal of Parallel and Distributed Computing* and the *IEEE Transactions on Parallel and Distributed Systems*, contain papers on all aspects of parallel processing. Several books focusing on parallel processing are included in the following references.

Asanovic et al. [2006] surveyed the wide-ranging challenges for the industry in this multicore challenge. That report may be helpful in understanding the depth of the various challenges.

In addition to documenting the discovery of concepts now used in practice, these references also provide descriptions of many ideas that have been explored and found wanting, as well as ideas whose time has just not yet come. Given the move toward multicore and multiprocessors as the future of high-performance computer architecture, we expect that many new approaches will be explored in the years ahead. A few of them will manage to solve the hardware and software problems that have been the key to using multiprocessing for the past 40 years!

History of Domain-Specific Architectures

An example of a DSA for simulation goes back to 1990 [Agrawal, 1990]. These simulator chips modeled VLSI chips at the gate level and improved performance about 50 times over that of a mainframe computer running a production-quality software simulator while retaining the same accuracy.

Two survey articles document that DSAs for DNNs are just as old [Ienne, 1996; Asanović, 2002]. For example, in 1990 CNAPS chips contained a 64 SIMD array of 16-bit by 8-bit multipliers, and several CNAPS chips could be connected together with a sequencer [Hammerstrom, 1990]. Twenty-five SPERT-II workstations, accelerated by the T0 custom ASIC, were deployed starting in 1995 to do both NN training and inference for speech recognition [Asanović, 1998]. The 40-Mhz T0 added vector instructions to the MIPS ISA. The eight-lane vector unit could produce up to sixteen 32-bit arithmetic results per clock cycle based on 8-bit and 16-bit inputs, making it 25 times faster at inference and 20 times faster at training than a SPARC-20 workstation. More recently, the DianNao family of four DNN architectures minimizes memory accesses both on the chip and to external DRAM by having efficient architectural support for the memory-access patterns in DNN applications [Chen, 2016].

DSAs would seem to be a good use case for FPGAs as a computing platform in data centers. One deployed example is Catapult [Putnam, 2016]. Catapult deployed Stratix V FPGAs into Microsoft data centers concurrently with TPUv1 in 2015. Perhaps the most significant difference between Catapult and the TPU is that to achieve best performance, users must write programs in the low-level hardware-design language Verilog versus porting programs using the high-level TensorFlow framework; that is, "reprogrammability" comes from porting software for TPUv1 rather than from writing firmware from scratch for the fastest FPGA.

Undoubtedly the date that the industry awoke to the commercial importance of DSAs for DNNs was May 18, 2016, when Google's CEO announced [Jouppi, 2018]:

> *We've been running TPUs inside our data centers for more than a year, and have found them to deliver an order of magnitude better-optimized performance per watt for machine learning.*

Over the next year, dozens of startups were formed to build DSA for DNNs, and Intel acquired a few DSA DNN companies. We are starting to see the fruits of the billions of dollars invested in DSAs for DNNs in 2020.

Further Reading

Agrawal P. and W. J. Dally [1990]. A hardware logic simulation system, *IEEE transactions on computer-aided design of integrated circuits and systems* 9(1): 19–29.

Almasi, G. S. and A. Gottlieb [1989]. *Highly Parallel Computing*, Benjamin/Cummings, Redwood City, CA.

A textbook covering parallel computers.

Amdahl, G. M. [1967]. "Validity of the single processor approach to achieving large scale computing capabilities," *Proc. AFIPS Spring Joint Computer Conf.*, Atlantic City, NJ (April), 483–85.

Written in response to the claims of the Illiac IV, this three-page article describes Amdahl's law and gives the classic reply to arguments for abandoning the current form of computing.

Andrews, G. R. [1991]. *Concurrent Programming: Principles and Practice*, Benjamin/Cummings, Redwood City, CA.

A text that gives the principles of parallel programming.

Archibald, J. and J. -L. Baer [1986]. "Cache coherence protocols: Evaluation using a multiprocessor simulation model", *ACM Trans. on Computer Systems* 4 (November), 273–98.

Classic survey paper of shared-bus cache coherence protocols.

Arpaci-Dusseau, A., R. Arpaci-Dusseau, D. Culler, J. Hellerstein, and D. Patterson [1997]. "High-performance sorting on networks of workstations," Proc. ACM SIG MOD/PODS Conference on Management of Data, Tucson, AZ (May), 12–15.

How a world record sort was performed on a cluster, including architecture critique of the workstation and network interface. By April 1, 1997, they pushed the record to 8.6 GB in 1 minute and 2.2 seconds to sort 100 MB.

Asanovic´, K. [2002]. Programmable neurocomputing. In: Arbib M. A., editor: *The Handbook of Brain Theory and Neural Networks, Second Edition*, Cambridge, MA, (November), MIT Press. (edu/~krste/papers/neurocomputing.pdf. https://people.eecs.berkeley).

Asanović, K., Beck, Johnson J, J. Wawrzynek, B. Kingsbury, N. Morgan. [1998]. Training Neural Networks with Spert-II. Chapter 11. In: *Paradigms and Implementations*. N. Sundararajan and P. Saratchandran, editors: *Parallel Architectures for Artificial Networks*, IEEE Computer Society Press, (November). (ISBN 0-8186-8399-6). https://people.eecs.berkeley.edu/~krste/papers/annbook.pdf.

Asanovic, K., R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick. [2006]. "The landscape of parallel computing research: A view from Berkeley." Tech. Rep. UCB/EECS-2006-183, EECS Department, University of California, Berkeley (December 18).

Nicknamed the "Berkeley View," this report lays out the landscape of the multicore challenge.

Bailey, D. H., E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga. [1991]. "The NAS parallel benchmarks—summary and preliminary results," *Proceedings of the 1991 ACM/IEEE conference on Super-computing* (August).

Describes the NAS parallel benchmarks.

Bell, C. G. [1985]. "Multis: A new class of multiprocessor computers," *Science* 228 (April 26): 462–467.

Distinguishes shared address and nonshared address multiprocessors based on micro processors.

Bienia, C., S. Kumar, J. P. Singh, and K. Li [2008]. "The PARSEC benchmark suite: characterization and architectural implications," Princeton University Technical Report TR-81 1-008 (January).

Describes the PARSEC parallel benchmarks. Also see http://parsec.cs.princeton.edu/.

Bouknight, W. J., Denenberg, S. A., McIntyre, D. E., Randall, J. M., Sameh, A. H., and Slotnick, D. L. [1972]. The Illiac IV system, *Proceedings of the IEEE*, 60(4), 369–388.

This describes the most infamous SIMD supercomputer.

Chen, Y., T. Chen, Z. Xu, N. Sun, and O. Teman. [2016]. DianNao Family: Energy-efficient hardware accelerators for machine learning, *Commun. ACM* 59(11): 105–112 (November).

Culler, D. E. and J. P. Singh, with A. Gupta [1998]. *Parallel Computer Architecture*, Morgan Kaufmann, San Francisco.

A textbook on parallel computers.

Dongarra, J. J., J. R. Bunch, G. B. Moler, G. W. Stewart [1979]. *LINPACK Users' Guide*, Society for Industrial Mathematics.

The original document describing Linpack, which became a widely used parallel bench mark.

Falk, H. [1976]. "Reaching for the gigaflop," *IEEE Spectrum* 13: 10 (October), 65–70.

Chronicles the sad story of the Illiac IV: four times the cost and less than one-tenth the performance of original goals.

Flynn, M. J. [1966]. "Very high-speed computing systems," *Proc. IEEE* 54 12 (December), 1901–1909.

Classic article showing SISD/SIMD/MISD/MIMD classifications.

Hammerstrom, D. [1990]. A VLSI architecture for high-performance, low-cost, on-chip learning. In: *Proceedings of the International Joint Conference on Neural Networks*, San Diego, CA, IEEE Press. (June 17–21).

Hennessy, J. and D. Patterson [2019]. Chapters 6 and 8 in *Computer Architecture: A Quantitative Approach*, sixth edition, Morgan Kaufmann, Cambridge, MA.

A more in-depth coverage of a variety of multiprocessor and cluster topics, including programs and measurements.

Henning, J. L. [2007]. "SPEC CPU suite growth: an historical perspective", *Computer Architecture News*. Vol. 35, no. 1 (March).

Gives the history of SPEC, including the use of SPECrate to measure performance on independent jobs, which is being used as a parallel benchmark.

Hillis, W. D. [1989]. *The connection machine*. The MIT Press.

PhD Dissertation that makes case for 1-bit SIMD computer.

Hord, R. M. [1982]. *The Illiac-IV, the First Supercomputer*, Computer Science Press, Rockville, MD.

A historical accounting of the Illiac IV project.

Hwang, K. [1993]. *Advanced Computer Architecture with Parallel Programming*, McGraw-Hill, New York.

Ienne, P., T. Cornu, G. Kuhn. [1996]. Special-purpose digital hardware for neural networks: An architectural survey, *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology* 13(1): 5–25.

Jouppi, N. [2018]. Google supercharges machine learning tasks with TPU custom chip. https://cloud.google.com/blog/products/gcp/google-supercharges-machine-learning-tasks-with-custom-chip, (May 16).

Another textbook covering parallel computers.

Kozyrakis, C. and D. Patterson [2003]. "Scalable vector processors for embedded systems", *IEEE Micro* 23:6 (November–December), 36–45.

Examination of a vector architecture for the MIPS instruction set in media and signal processing.

Menabrea, L. F. [1842]. "Sketch of the analytical engine invented by Charles Babbage", *Bibliothèque Universelle de Genève* (October).

Certainly the earliest reference on multiprocessors, this mathematician made this comment while translating papers on Babbage's mechanical computer.

Pfister, G. F. [1998]. *In Search of Clusters: The Coming Battle in Lowly Parallel Computing*, second edition, Prentice Hall, Upper Saddle River, NJ.

Putnam, A, et al. [2016]. A reconfigurable fabric for accelerating large-scale datacenter services, *Commun. ACM* 59(11):114–122 (November).

An entertaining book that advocates clusters and is critical of NUMA multiprocessors.

Regnier, G., S. Makineni, I. Illikkal, R. Iyer, D. Minturn, R. Huggahalli, and A. Foong [2004]. TCP onloading for data center servers. *Computer*, 37(11), 48–58.

A paper describing benefits of doing TCP/IP inside servers vs. external hardware.

Seitz, C. [1985]. "The Cosmic Cube", *Comm. ACM* 28 1 (January), 22–31.

A tutorial article on a parallel processor connected via a hypertree. The Cosmic Cube is the ancestor of the Intel supercomputers.

Slotnick, D. L. [1982]. "The conception and development of parallel processors—a personal memoir", *Annals of the History of Computing* 4: 1 (January), 20–30.

Recollections of the beginnings of parallel processing by the architect of the Illiac I V.

Williams, S., J. Carter, L. Oliker, J. Shalf, and K. Yelick [2008]. "Lattice Boltzmann simulation optimization on leading multicore platforms," *International Parallel & Distributed Processing Symposium (IPDPS)*.

Paper containing the results of the four multicores for LBMHD.

Williams, S., L. Oliker, R. Vuduc, J. Shalf, K. Yelick, and J. Demmel [2007]. "Optimization of sparse matrix-vector multiplication on emerging multicore platforms," *Supercomputing* (SC).

Paper containing the results of the four multicores for SPmV.

Williams, S. [2008]. *Autotuning Performance of Multicore Computers*, Ph.D. Dissertation, U.C. Berkeley.

Dissertation containing the roofline model.

Woo, S. C., M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. "The SPLASH-2 programs: characterization and methodological considerations," *Proceedings of the 22nd Annual International Symposium on Computer Architecture (ISCA '95)*, May, 24–36.

Paper describing the second version of the Stanford parallel benchmarks.