Cerebras Wafer Scale Engine: An Introduction

1. The Need for Speed

Deep learning has emerged as the most important computational workload of our generation. In the past five years, Artificial Intelligence (AI) has risen from obscurity to top-of-mind awareness because of advances in deep learning. Tasks that historically were the sole domain of humans are now routinely performed by computers at human or superhuman levels.

Deep learning is profoundly computationally intensive. A recent report by OpenAI showed that, between 2012 and 2018, the compute used to train the largest models increased by 300,000X. In other words, AI computing is growing at a rate that is 25,000X faster than Moore's law at its peak. AI compute demand is doubling every 3.5 months.

This voracious demand for compute means AI is constrained not by applications or ideas, but by the availability of compute. Testing a single new hypothesis—training a new model—takes weeks or months and can cost hundreds of thousands of dollars in compute time. This has slowed innovation to a crawl. Google, Facebook, and Baidu, among others, have noted that long training time is the fundamental impediment to AI progress; that many important ideas are ignored simply because these models take too long to train.

To meet the growing computational requirements of AI, Cerebras has designed and manufactured the largest chip ever built. The Cerebras Wafer Scale Engine (WSE) is 46,225 millimeters square, contains more than 1.2 trillion transistors, and is entirely optimized for deep learning work. By way of comparison, the WSE is more than 56X larger than the largest graphics processing unit, containing 3,000X more on chip memory and more than 10,000X the memory bandwidth.

Bigger is better in Artificial Intelligence compute. Larger chips process information more quickly and produce answers in less time. The WSE reduces the time it takes to do the most complicated AI work from months to minutes.

2. The Deep Learning Workload

At Cerebras, we began with a clean sheet of paper and a desire to build a new type of computer optimized exclusively for deep learning. Choosing the right computer architecture for a specific workload is like finding the right design for a car. The first question to ask is: What is its job? Will it be taking children to soccer practice? Or will it be moving bricks and lumber? These questions determine whether a minivan or a pickup truck is the right architecture. In computer design, too, understanding the computer's workload—in this case neural network processing – is the first step.

A neural network is a sequence of layers, arranged from the input (where data flow in) to the output (where, for example, classification predictions or translated sentences or Go moves flow out). The function of each layer is parameterized by the model parameters of that layer. That function is dominated by a simple, highly parallel operation, such as multiplying a vector (the input data) by a matrix (the model parameters).

In **inference**, the parameters are fixed and data flows through in one direction, from input to output (I to O). At O, we get a result: it's a dog, or a cat, or your most strategic board game move.

But during **training**, data flows both ways. A labeled set of input data, called the training set, flows from I to O. At the output end, rather than providing an answer, the output is compared to the correct answer for that particular input. Then, with the help of a little calculus, the network improves the accuracy of I-to-O calculation by reducing the errors it made. The process is as follows: with each new training sample, the network changes how it makes its output predictions such that it is continuously reducing the error; error being the difference between the predicted output and the correct output.

This information takes the form of something called a *gradient*. The gradient begins to work its way backwards, in the O-to-I direction. As this data flows through each layer, it interacts with the I-to-O data that produced it (that data has been parked there, waiting for the gradient to come back) and, together, they determine how to change the layer's parameters to most effectively reduce the error. The parameters are then adjusted, and this process continues for numerous passes over the set of training examples. Basically, a multi-stage training loop is created.

The time required to train a network depends on the rate at which inputs can be processed through this feedback loop. The faster the inputs can move through the loop, the more inputs are sent through the loop per unit time and the faster the network trains.

2. Understanding Performance

By thinking about a deep neural network as a multistage computational feedback loop we demystify performance. And, we can better understand the architectural choices made by startups and incumbents alike.

The only way to reduce training time is to reduce the time it takes for the inputs to travel through the feedback loop. The only things happening in the feedback loop are calculations and communications. So, calculations and communication must be accelerated.

Accelerating calculation is most directly achieved by increasing the number of compute cores. More cores—specifically more floating point multiply accumulate units-- do more calculations in less time. Placing high-speed memory close to cores ensures that the cores are always occupied doing calculations. Placing memory far away off chip means the cores are frequently waiting for the data on which to do their calculations.

High-bandwidth, low-latency communication ensures that compute cores can be gathered together into communicating groups that quickly solve problems that would take a single core too long to do. In addition, High-bandwidth, low-latency communication ensures that the results of each stage in the loop are quickly delivered to the subsequent stage.

With this simple understanding of AI performance, we understand the competitive landscape in AI compute. For training, graphics processing units have dominated central processing units. The graphic processing unit has thousands of cores. The central processing unit has dozens of cores. At first pass, it's that simple.

The big cores in central processing units weren't designed for the type of calculations in a multistage training loop. The little cores in graphics processor weren't designed for AI work either, but there were more of them and that carried the day. This logic also explains why, over the past six years, Nvidia's graphics processing units have grown from 551 square millimeters to 815 square millimeters. More silicon area provides more room for cores and **more cores deliver more calculations**. More silicon area also provides more room for memory close to cores.

The same insights explain the recent focus on **communication fabrics**. As AI compute became more demanding, the number of cores needed in the training loop exceeded the number of cores on a single chip. Having reached the traditional limit to chip size, the only way to get more cores was to add chips by building a cluster. Since off-chip communication is tens of thousands of times slower than on chip communication, linking together cores from different chips to work on a single problem became an important problem to solve. This logic informs why NVIDIA designed NVLink as a way to improve communication among chips and why they paid $6.8 billion to acquire Mellanox – Mellanox pioneered InfiniBand, which is a communication technology for linking chips together.

This same logic helps to parse through claims made by various startups in the AI space. Some will claim that they have moved memory onto the chip and thereby increased the calculation performance of nearby cores. This is true. More on-chip memory close to the cores increases efficiency of calculations. Other companies will say they more tightly tuned their cores for AI work and thereby get more calculations per unit time or that they avoid doing useless work like multiplying by zero. Optimizations that provide more calculations and avoid wasting time on non-useful work will also improve performance.

But in the end, all of these strategies are trying to drive up calculation and accelerate communication through one or more of three strategies: 1) more/better cores, 2) more memory close to cores, and 3) more low-latency bandwidth between cores.

But what would happen if you took these three approaches to their logical extreme - what would you get? A very big chip, with memory close to cores, all connected by a high-bandwidth, low-latency fabric.

How might this be achieved? Small chips are built as arrays of identical chips on a 12-inch diameter silicon wafer. To get a really big chip, instead of cutting the wafer into small chips, you would increase the size of the chip to a full wafer. That would dramatically increase the number of cores available to do calculations. Now that you had sufficient silicon area, you would distribute memory across the chip, giving each core its own local memory.
Finally, you would build a communication fabric across all the cores since the communication would all be on-die, where it is many thousands of times faster than off-chip technologies like InfiniBand.

Why hasn’t this been done before? Because it is very difficult. Nobody has ever built and brought to market a chip larger than 840 mm^2. The design, manufacturing, power, cooling, communication and coordination challenges are immense. But the promise is huge. The resulting chip would be 50x larger than the largest chip on the market today. It would have hundreds of thousands of AI optimized cores and Gigabytes of on chip memory distributed across the cores. And it would have petabytes/s of core to core bandwidth.

4. The Cerebras Architecture: Accelerating Deep Learning with the Computational Trifecta

With a focus on AI and only AI, the Cerebras Wafer Scale Engine accelerates both calculation and communication, and by doing so, reduces training time. The approach is a straightforward function of the size of the WSE. With 56 times more silicon area than the largest graphics processing unit, the WSE provides more cores to do calculations, more memory closer to the cores so the cores can operate more efficiently, and more low latency bandwidth between cores so groups of cores can collaborate effectively.

The computational trifecta- more cores, more memory close to cores, more bandwidth between cores-- allows the Wafer Scale Engine to avoid the old performance bugaboos of slow off-chip communication, distant memory, low memory bandwidth, and wasting computation resources on useless work. In other words, the WSE achieves cluster scale performance without the penalties of building large clusters.

3.1 More Silicon Area Means More Space for Compute Cores

The Cerebras Wafer Scale Engine delivers 400,000 programmable compute cores. These Sparse Linear Algebra Cores, called SLA™ , are, unsurprisingly, optimized for the sparse linear algebra that is fundamental to neural network calculation. These cores are designed specifically for AI work. They are small and fast, contain no caches, and have eliminated other features and overheads that are needed in general purpose cores but play no useful role in a deep learning processor.

The SLA cores are programmable, ensuring that they can run all neural network algorithms in the constantly changing field of deep learning. Each core performs both control processing and data processing. The control processing is used for parallel processing coordination and the data processing is for the math operations at the heart of neural networks.

Control processing is achieved with a full set of general-purpose instructions. These instructions provide programmable primitives such as arithmetic, logical, and branching operations. These primitives provide the foundation on which any parallel algorithm can be mapped.

Tensor operations are the core of the neural network workload. To achieve high performance, the SLA cores have a specialized tensor processing engine where full tensors are first-class operands in architecture. The tensor operations are programmable, so the same engine can be programmed to perform a variety of tensor operations such as convolution or matrix multiply. The hardware internally optimizes the tensor processing to achieve datapath utilization three of four times greater than graphics processing units.

Cerebras' unwavering focus on the deep learning workload is further evidenced by the handling of sparsity. Cerebras invented sparsity harvesting technology, which allows the Sparse Linear Algebra Cores to improve performance by harvesting the sparsity that abounds in neural network workloads.

Multiplying by zero is a waste—a waste of silicon, power, and time, all while creating no new information. In deep learning, the data are often very sparse. Half to nearly all the elements in the vectors and matrices that are to be multiplied together are zeros. The source of the zeros are fundamental deep learning operations, such as the rectified linear unit nonlinearity (ReLU) and dropout, both of which introduce zeros into neural network tensors. Newer methods are emerging around weight sparsity that add even more zeros. Put simply, there is a growing interest in making tensors sparser still.

Graphics processing units and tensor processing units are dense execution engines; they perform the same computation task on an entire vector or matrix of data. This is a wise approach when the vector or matrix is dense (all nonzero). In the dense environment, efficiencies are gained by bunching the data together and providing a single instruction to be applied to all of the data. But when the data is 50 to 98% zeros, as it often is in neural networks, then 50 to 98% of your multiplications are wasted.

Because the Cerebras SLA core was designed specifically for the sparse linear algebra of neural networks, it never multiplies by zero. To take advantage of this sparsity, the core has built-in, fine-grained dataflow scheduling, so compute is triggered by the data. The scheduling operates at the granularity of a single data value so only non-zero data triggers compute. All zeros are filtered out and can be skipped in the hardware. In other words, the SLA core never multiplies by zero and never propagates a zero across the fabric. Thus, not only does the SLA core save the power and energy by skipping the useless computations, but it also gains performance advantage by using that same time to do useful work instead while other architectures are stuck multiplying by zero.

Sparsity is not evenly distributed nor is it of uniform size. It can be fine-grained where individual activations or weights become zero or coarse-grained or where adjacent blocks of activations and weights are all zero. To maximize the performance opportunity available from sparsity, it is vital that the architecture can harvest both fine-grained and coarse-grained sparsity. The SLA core architecture was designed to do just that.

Sparsity is an important characteristic of neural networks that must be leveraged to achieve the maximum performance. The computer architect's mantra—do only useful work—is brought to the forefront with sparsity. The WSE houses hundreds of thousands of AI optimized cores each with the ability to harvest sparsity and avoid doing senseless work, providing an architectural foundation for extraordinary performance.

3.2 More Silicon Area Means More on Chip Memory

Memory is a key component of any computer architecture. The closer memory is to a compute core, the faster the calculation, the lower the latency, and the less power is used moving data. High-performance deep learning requires each core to operate at maximum levels, and this requires close collaboration and proximity between the core and memory.

In deep learning, memory is used to hold the model's parameters, the activations, the model configuration, and more. In state-of-the-art-networks, the model parameters already run into the gigabytes, and this memory requirement is expected to grow. But large models, needing a lot of memory and a lot of compute, create fundamental challenges for traditional architectures.

For maximum performance, the entire model should fit in the fastest memory, which is the memory closest to the computation cores. This is not the case in CPUs, TPUs, and GPUs, where main memory is not integrated with compute. Instead, the vast majority of memory is based off-chip, far away on separate DRAM chips or a stack of these chips in a high bandwidth memory (HBM) device. As a result, main memory is excruciatingly slow.

While computer architects have tried for years to address the memory bottleneck, the primary solution has been memory hierarchies, based on many levels of on-chip and near-chip caches. These are costly, small, difficult to use, and provide sometimes unpredictable benefits. The simple and unavoidable truth is that once the on-chip/off-chip boundary is crossed, the latency-to-memory explodes and the bandwidth plummets. The extreme latency penalty in accessing off-chip memory forces a downward spiral of performance. This is one of the fundamental reasons graphics processors are slow when doing artificial intelligence work.

Cerebras solved this problem. The Cerebras WSE has 18 Gigabytes of on chip memory and 9.6 bytes of memory bandwidth. Respectively this is 3,000 and 10,000 times more than the leading graphics processing unit. As a result the WSE keeps the entire neural network model, that is all the parameters to be learned, on the same silicon as the compute cores, where they can be accessed at full speed. This is possible because memory on the Wafer Scale Engine is widely distributed alongside the computational elements, allowing the system to achieve extremely high memory bandwidth at single-cycle latency, with all model parameters in on-chip memory, all of the time.

The Cerebras WSE delivers more compute cores, more local memory, and more memory bandwidth than any chip in history. This enables fast computation, reduces the time it takes to train a model, and uses less energy as well.

3.3 More Silicon Area Enables Blisteringly Fast Communication—Swarm™ Fabric

AI-optimized cores and high-speed, local memory drive up performance by improving the number, rate, and flexibility of calculations. Performance improvements also come from accelerating communication. Artificial intelligence is a communication intensive workload—layers, and therefore cores, communicate constantly—so the fabric that links cores together is fundamental to performance. Maximum performance is achieved when cores can communicate at high bandwidth and low latency since the cores can be clustered together to do in a group what would take an individual core too long to do.

A fundamental truth in computer architecture is that off-chip communication is tens of thousands of times slower than on-chip communication. Small chips that wish to communicate must be clustered together via Ethernet, InfiniBand, PCI-E, or other off-chip technologies, which all suffer an enormous performance penalty compared to staying on-silicon. On-chip communication is faster, and it consumes less than a thousandth the power per bit moved.

The Cerebras Swarm communication fabric creates a massive on chip network that delivers breakthrough bandwidth and low latency at a fraction of the power draw of the traditional communication techniques used to cluster graphics processing units into clusters.

The 400,000 cores on the Cerebras WSE are connected via the Swarm communication fabric in a 2D mesh with 100 Petabits per second of bandwidth. Swarm provides a hardware routing engine to each of the compute cores and connects them with short wires optimized for latency and bandwidth. The resulting fabric supports single-word active messages that can be handled by the receiving cores without any software overhead. The fabric provides flexible, all-hardware communication.

Swarm is fully configurable. The Cerebras software configures all the cores on the WSE to support the precise communication required for training the user-specified model. For each neural network, Swarm provides a unique and optimized communication path. This is different than the approach taken by central processing units and graphics processing units that have one hard-coded on-chip communication path into which all neural networks are shoehorned.

Swarm's results are impressive. Typical messages traverse one hardware link with nanosecond latency. The aggregate bandwidth of the system is measured in tens of petabytes per second. Communication software such as TCP/IP and MPI is not needed, avoiding associated performance penalties. The energy cost of communication in this architecture is well below one picojoule per bit, which is nearly two orders of magnitude lower than central processing units or graphics processing units. As a result of the Swarm communication fabric, the WSE trains models faster and uses less power.

4. Co-Designed with Software for Maximum Utilization and Usability

The Cerebras software stack has been tightly co-developed with the Wafer Scale Engine to take full advantage of its unique capabilities. At its core is the Cerebras graph compiler, which enables automatic translation of an ML researcher's neural network into an optimized executable for the massive computational resources of the WSE.

To make it easy and intuitive for ML practitioners to use the WSE, the Cerebras software stack provides a seamless interface to existing high-level ML frameworks, such as TensorFlow and PyTorch. The graph compiler begins by extracting a dataflow graph representation of the neural network from the user-provided framework representation. Portions of the network are matched to optimized microcode programs, and these programs are mapped to specific resource regions of the immense computational array. The software then configures the Swarm interconnect fabric, creating a tailored data path through these regions of compute to maximize locality and minimize communication cost. The result is a highly optimized placement of the neural network onto the WSE, ready for the user to perform training or inference.

In addition to traditional execution modes, the scale of the WSE enables novel methods of model-parallel execution. The WSE is capable of running an entire neural network on the fabric at once – mapping each layer of the network to a single stage in a multi-stage pipeline for full, layer-parallel, pipelined execution. The user can then rapidly stream data through the pipeline, running all stages and thus, all layers in the neural network, simultaneously. This approach is unique to the WSE and is only possible because of its immense scale.

In summary, the Cerebras Software Stack:

- Allows users to program the WSE using familiar, high-level ML frameworks
- Automatically compiles ML framework-defined models into WSE executables
- Maximizes utilization of all 400,000 cores and takes advantage of the WSE’s unique scale

5. Conclusion: WSE, The Future of AI Computation

Cerebras Systems is a team of pioneering computer architects, computer scientists, deep learning researchers, and engineers of all types who love doing fearless engineering. We have come together to build a new class of computer to accelerate artificial intelligence work. The first announced element of the Cerebras solution is the Cerebras Wafer Scale Engine.

The WSE is the largest chip ever built. It is 46,225 square millimeters and contains 1.2 Trillion transistors and 400,000 AI optimized compute cores. The memory architecture ensures each of these cores operates at maximum efficiency. It provides 18 gigabytes of fast, on-chip memory distributed among the cores in a single-level memory hierarchy one clock cycle away from each core. These high-performance, AI-optimized, local memory fed cores are linked by the Swarm fabric, a fine-grained, all-hardware, high bandwidth, low latency mesh-connected fabric.

By accelerating AI compute, the Cerebras WSE eliminates the primary impediment to the advancement of artificial intelligence by reducing the time it takes to train models from months to minutes and from weeks to seconds. Thus, the WSE enables deep learning practitioners to test hypotheses more quickly and to explore ideas that today are untestable with legacy architectures or are too risky to try. The WSE reduces the cost of curiosity, accelerating the arrival of the new ideas and techniques that will usher forth tomorrow's AI.