**Andy:** 0:00

What we're looking at here is a square piece of silicon that was cut out of a circle of silicon that was 300 millimeters in diameter. So you see data scientists and ML researchers using libraries like DeepSpeed and Megatron, specialized versions of ML frameworks like Distributed PyTorch or Distributed TensorFlow, tools like OpenMPI, horovod to effectively make the problem of spreading a large model out over many small workers make it incrementally easier. They spend tons of their time doing sort of supercomputer engineering, not thinking as much as we might like about the AI, the application. And then, once you actually get that model up and running because you're training it, then on a cluster of many small workers, there's inefficiencies. If you go from one to say, 512 GPUs don't get 512 times faster because your bottleneck by memory bandwidth and communication bandwidth between chips. That's why we ended up with such a unique looking processor, because, at the end of the day, ai work is a different kind of work than the computational workloads that preceded it.

**Craig:** 1:09

Hi, I'm Craig Smith and this is Eye on AI. I'm at NeurIPS 2023 in New Orleans this week and I'm thrilled to have Andy Hawk, head of product for Cerebrous Systems, here to discuss the company's revolutionary wafer scale chip technology. Andy talked about how this new chip architecture is being used to train foundation models and how it may help break the bottleneck in generative AI caused by the ongoing GPU shortage. I hope you enjoy the conversation as much as I did. I want to give a shout out to our sponsor this week, babbel, the Science Backed Language Learning app. Babbel is something I feel strongly about because language is the key to opening the world and broadening horizons. I know because, as a journalist, I've reported out of more than 40 countries around the world, and it's amazing how knowing just a few words of the local language will open doors and build bonds. Be a better you in 2024 with Babbel. Don't pay hundreds of dollars for private tutors or waste hours on apps that don't really help you speak the language. Babbel's quick 10-minute lessons are designed by over 150 language experts to help you start speaking a new language in as little as three weeks. Babbel's designed by real people for real conversations. Babbel's tips and tools are approachable, accessible, rooted in real-life situations and delivered with conversation-based teaching, so you're ready to practice what you've learned in the real world. It's so easy to learn how to order food, ask for directions, speak to merchants without having to consult language apps while on vacation. The key to learning languages is not to be embarrassed for speaking poorly. Plus, babbel's speech recognition technology helps you to improve your pronunciation and accent, something that I need a little work on. Studies from Yale Mish state university and others continue to prove Babbel is better. Right now, get 55% off your Babbel subscription, but only for our listeners at babbelcom.

**Andy:** 3:25

slash ionai this is an actual production wafer scale engine from Serebus. This processor is the heart of the system that we build and deliver to customers and make available remotely and through cloud. It is a beast 850,000 cores in this generation and 40 gigabytes of on-ship memory, and all those cores are directly connected over silicon. You can think of this in a sense as a cluster worth of AI compute all on one device. You can also serve breakfast on it.

**Craig:** 4:04

So that would be the equivalent. I had a couple of questions about that. People are using it now for training large models. Is that right, absolutely, and what is the difference between that and an NVIDIA GPU?

**Andy:** 4:27

Sure. So this is a fundamentally different device from the architectural level of the cores all the way out to the wafer. So this isn't just a bunch of GPUs together. This is a processor that is built from the ground up for AI, not for graphics, not for database management, but really built to accelerate large-scale AI. And so there's a couple of fundamental differences. First and foremost is that the processor level all those cores, those 850,000 cores they're designed to handle sparse tensor-based linear algebra operations from the get-go. This is not different functional units on that chip that are designed for ray tracing or, say, 64-bit computations, for physics-based simulations. The cores on this thing are built to accelerate the sparse linear algebra ops that are fundamental to all of AI compute.

**Craig:** 5:24

What was the inference and training?

**Andy:** 5:27

Both inference and training, and what we've focused on in terms of our software and our go-to-market has been on accelerating training in the data center. But this thing is built from the ground up for that work, and what that really means is that, look, what we observed is that AI compute isn't just, in particular, data center training. It isn't just a problem of bringing a lot of compute to bear. It's also a problem of bringing that compute to bear with high memory bandwidth and high communication bandwidth. If we look today at problems like training GPT models or chat GPT size models, right, these are models that don't fit on a single GPU anymore and they would be slow if you did. But they don't fit on a single GPU anymore. So it requires you to train these large models. If you're using GPU on large clusters of, say, hundreds to thousands of tiny general-purpose chips and look, that's gotten the industry a long way right, that is an extraordinary machine and it's a suitable engine for large-scale AI. But when you're training a large model on a cluster of many small chips, you're into a couple problems. The first is that, first of all, it's just hard to distribute that model, to program the model to run on so many small workers. So you see data scientists and ML researchers using libraries like DeepSpeed and Megatron and specialized versions of ML frameworks like distributed PyTorch or distributed TensorFlow, tools like OpenMPI and Coravod to effectively make the problem of spreading a large model out over many small workers make it incrementally easier, but that still requires thousands or tens of thousands of lines of code. It takes days or weeks, or even months sometimes of software engineering just to get the model set up to run. And then, well, good luck if you change something right. If you change the model architecture, you change something about the data or you change the size of your cluster, often have to go back to the beginning and redo the distribution problem. So what ends up happening in that case is that your data scientists and your ML researchers are often the most valuable people in an organization that's pursuing AI. They spend tons of their time doing what you think of as sort of supercomputer engineering, right Parallel programming, not thinking as much as we might like about the AI, the application. And then, once you actually get that model up and running because you're training it, then on a cluster of many small workers, there's inefficiencies. If you go from one to say, 512 GPUs, you don't get 512 times faster because you're bottlenecked by memory bandwidth and communication bandwidth between ships. Okay, so then let's come back to our machine, right, the wafer and clusters of our machines. Our machine, because of its physical scale and because of our cluster architecture, we can run even the largest models on the planet today, that is, 10 billion, 100 billion, even trillion, parameter or larger models on a single machine. And then if you want to scale out, that is, if you want to add more machines to go faster because we can run the whole model on just one machine, then we can run the whole model on every machine in the cluster and just ask every machine in the cluster to work on a different part of the dataset. And that's called data parallel scaling, right, and it's fundamentally different than the distribution techniques that are required for a cluster of GPUs, where you can't do just data parallel, you also have to do model parallel, tensor parallel. And what this? Being able to run the largest models on one machine and scale with simple data parallelism, only on cerebrus. What that really means to an end user is I can program a cluster of Cerebrous machines that give me the horsepower of, say, thousands of GPUs with the same code that I would use to train my model on a single desktop machine and that takes that days or weeks of months of software engineering and all the required expertise of parallel programming and supercomputer architecture. It just makes it all go away and makes it far simpler for users to get up and running. And then when we run on say 16 or 32 or 64 of our machines, like our latest clusters, you actually get 16 and 32 and 64 times faster than one. And at the end of the day especially for someone like me coming from a research background what that means for our end users that are researchers themselves or researchers in an enterprise organization building new applications, it just means that they can go that much faster, they can ask and answer questions more quickly, they can set up and train these billion, 10 billion, 100 billion parameter models in days or weeks rather than weeks or months, and bring new applications to life that much sooner.

**Craig:** 10:44

Yeah, it's interesting. I was talking to somebody from DeepMind yesterday or I was at a talk at which somebody from DeepMind was speaking along with somebody from Meta, and the question was how difficult is, would it be, for a master's level student, computer science student, to train an LLM on one GPU? Could they do it with all the tools available? And they were saying, yeah, up to about 70 billion parameters is what they were claiming. But once you get beyond that, once you need more than one GPU, it just becomes this massive problem exactly what you're talking about and that, how much time they spend and how much expertise is required to coordinate multiple GPUs. So two questions how many GPUs the GPU equivalents does one wafer represent?

**Andy:** 12:08

Sure, you see a lot in marketing materials about flops, about floating point operations per second. But what really matters to an end user in this business is time to solution that is how quickly can I train a model to stay to the accuracy? So sometimes it's not even just throughput, that is training samples per second. Sorry about that. It's training throughput that yields state of the art accuracy. So no tricks, just tell me how fast I can train a model to state of the art accuracy. And in some recent work actually one of our customers published an independent study of novel AI accelerators for large AI model training. This is some work from Argonne National Labs, department of Energy, us Research Laboratory, and they're looking at big AI models for science. Some really, really interesting stuff under the hood. But the result that they found when comparing our CS2 machine to A100 was that for training particularly large GPT models, they found the CS2 was 152 times faster than an A100. And that's a meaningful difference. When you couple that together with a simple or programming model, it just lets you iterate that much faster. So, long story short, each one of our machines typically delivers the compute equivalent of several tens to, in some cases, hundreds or more GPUs for the same AI training task.

**Craig:** 13:53

The? Yeah, and you mentioned efficiency or utilization. That was one of the things that the deep mind person was saying that it's one thing is getting them all to work together. The other is to you're running it I can't remember the technical term, but it basically at 50% capacity, and to get above that then you need to do a lot of other stuff. So do you have that problem on Cerebras that you're actually using the power of the chip?

**Andy:** 14:33

We're actually using the power of the chip, so our software takes care of that challenge also under the hood. By virtue of our architecture and the way we execute a model, we can achieve. Very high utilization is the term of our very high utilization for this chip. I would actually say that, while it's true that some of the leading AI research labs in the world that is the cream of the crop while it's true that some of those researchers might be able to get something like 40 to 50% utilization out of a GPU, it's not that common and it's very difficult. In order to achieve that kind of utilization on a general purpose processor like a GPU for an AI workload, you end up tweaking sort of lower level micro code kernels and tweaking communication patterns between devices. I think that's probably what they were alluding to. That is, it is possible to squeeze that much performance juice from a GPU, but it's very difficult and it requires highly specialized expertise that really fundamentally only lives in a few organizations on the planet, and I know this is a related point a little bit of a jog. But I think one of the things that we are really trying to do at Cerebrus is not just build the most powerful AI processors and systems on the planet to accelerate AI research, but also put that power into more users' hands, that is, democratize access to the highest performance AI compute, and I think part of that is actually making the programming easier and giving users who might not know how to architect their own micro code kernels for compute operations or communication patterns, giving the tools that they need to harness that power from a processor on their own, with the tools that they use today. So the short answer is although our chip looks different than a regular processor, you program our machines with standard ML frameworks like PyTorch and, as we said before, you don't need any specialized libraries. You don't need to go underneath the hood and optimize kernels. All of that power gets put in your fingertips through our compiler and our software stack and you get to use the tools that you're comfortable with.

**Craig:** 17:22

Yeah, actually, can you hold that up to the camera again and just explain what we're looking at? I mean, it's a piece of silicone, and what are the individual rectangles? Sure, yeah.

**Andy:** 17:43

So, as I mentioned, Cerebrus was born out of an interest to really fundamentally transform the AI compute landscape and as a startup, we got our start back in 2016,. We had the opportunity to approach the processor design problem from first principles. That is, we could ask what is the right processor shape, size, architecture for AI full stop, Whereas I think if you're already in the industry and already making computer chips, then you can make small changes to the architecture to move it towards the AI workload. But it's much harder to take a clean slate approach. So we had this opportunity in front of us to take a clean slate approach and that's why we ended up with such a unique looking processor, because, at the end of the day, AI work is a different kind of work than the computational workloads that preceded it. So what we're looking at here is a square piece of silicon that was cut out of a circle of silicon that was 300 millimeters in diameter. That's the largest.

**Craig:** 19:02

And that's the 330 millimeter diameter. That's what do you call that? It's a loaf, it's a wafer, a wafer.

**Andy:** 19:08

Yeah, that's a wafer.

**Craig:** 19:10

It's cut into a wafer.

**Andy:** 19:13

Oh, oh, oh, you might be right.

**Craig:** 19:16

Yeah, but it's a long cylinder, it's like a long cylinder, that's right.

**Andy:** 19:20

That's right. And it gets sliced into very thin wafers. So you probably can't resolve this in the camera, right? But what we're actually looking at here is basically as thin as a pane of glass, right? If I tried to hold this in my fingers, you might cut yourself on the edges. So, yeah, we start with a blank circular wafer, and then our fabrication partner uses optical lithography, basically to etch the circuitry onto the silicon wafer, and the way they do that optical lithography is in these individual steps. So each one of these tiny rectangles that you might be able to see in the camera, each one of these tiny rectangles and there's 84 of them on my wafer each one of those tiny rectangles is called a reticle, and so that's the physical unit of optical lithography exposure. It's like a camera shot right, and so we take one and then step it to the right, take another, take another, take another On our wafer. All of these are identical, and for other processors they might be identical too. The big difference is that once we're done writing all the circuitry on this wafer, we don't cut it any further. If I was making other chips, like a GPU or a CPU, I'd start with something that looks like this, but then I would dice it up into many tiny chips. Sort of an interesting way to think about it, right? Because then what they end up doing is they end up mounting each one of those little chips on a motherboard with memory and interconnect, and then we spend a lot of time as a community figuring out how to cluster them and get them to work together again. And we just said, well, let's not do that, let's not cut it apart, let's keep it all together so that we can have all this compute close to memory and able to talk to one another without having to put it into individual chips and then string them all back together with a bunch of cabling and other machines. So you're looking at 84 reticles, and this wafer, our wafer, is fabricated by TSMC. There's only a few fabs in the world that can produce chips and circuitry at this resolution, in this generation. We're fabricating at the 7 nanometer process.

**Craig:** 21:40

And that 7 nanometer, just that's kind of a reference term. It doesn't mean that the circuits are literally 7 nanometers wide.

**Andy:** 21:51

That's right. That's right, yeah, and in the history of optical lithography they were actually very close to that and they still are. But it's not exactly 7 nanometers and, particularly as we get the future higher resolution, finer scale nodes like 5 nanometer, 3 nanometer, 2 nanometer. They're not exactly that either, but they're very close. And that fine pitch just means I can put more transistors per unit area. And as a chip designer, our primary resource just like an artist working with a canvas, our primary resource is how much silicon do we have and what do I want to do with that area? So if I can put more transistors per unit area, that means more cores or more memory and therefore more capability. So, coming back to this guy, we have 84 identical reticles, 850,000 identical cores, all interconnected, all with their own memory, one clock cycle away. As I think of it as the fundamental unit of compute processing or a single processing element, so it includes a data path with transistors that, when data comes in, it executes the computation. It also includes interconnect to adjacent cores and interconnect to memory, so it's sort of the fundamental cell of your unit of computation on a machine.

**Craig:** 23:27

Yeah, okay, because when they talk about dual core, you're talking about one chip, but on that there are two processing units, many tiny computers.

**Andy:** 23:39

Exactly, exactly, yeah, right, if you're like me, you remember maybe when the first dual core CPUs came out, or four core CPUs, and even state of the RC, if you use today, maybe have 24 or 48 cores, right, we're talking about 850,000 on this device.

**Craig:** 24:00

And we were talking yesterday. So that's a very thin slice of this loaf, this cylinder of silicon that's etched. It's got a copper color. Is there? Is it then covered with something?

**Andy:** 24:17

Yeah, there are many layers, right? So the actual fabrication process will be laying down many layers, one on top of another, of circuitry and then a final upper layer that will connect all the cores and maybe connect the chip itself to other IO systems, right? So what you're looking at here on our device is we have a uniform 2D array of those 850,000 identical cores, so it's a very simple design. And then on the edges we have IO circuitry so that when we hook this into a machine we can get data from the outside world Say, in AI we can get the weights of the model and the input data batches actually onto the wafer. And then the wafer those 850,000 cores on the wafer all execute the multiply and accumulate operations that constitute the AI training problem. But just sort of a fun fact and I know we're talking a lot about the wafer today but once we landed on this architecture because we thought it was the right chip architecture for AI and solved this decades long challenge of integrating a single device at wafer scale, that was really just the beginning of the engineering challenges that we faced, because then we had to figure out how to package this thing and put it into a system that could go into a data center and connect with standard power, standard interconnect to other machines. So that meant we had to figure out how to power this thing, how to keep this thing cool. This is consuming about 17 to 18 kilowatts worth of power at about 20,000 amps when it's running in a machine. So how do I keep all that cool and uniform? So building the wafer itself was in some sense just the beginning. Then we had to figure out how to package it, how to power it, how to cool it, how to deliver data so that it could live and breathe in today's standard data center environments. And so we couldn't be just a chip company, we had to be a full systems company and then write the software on top of it to make that whole thing programmable by data scientist or ML researcher who's probably miles away from the data center. I think one of the things that I enjoy about SRE versus a company and a team is that there's this common thread amongst our engineers, both from the hardware and the software and the ML research side, that they in some sense they are unafraid of these big challenges, right, Whether it's architecting a wafer, scale chip or bending metal and cutting gaskets to cool this beast in a data center machine, or building software on top of it, or training new devices or new world leading models, right, we get excited by those kinds of challenges that I think might throw other people for a loop. Yeah, and this is the second version right, yeah, yeah, this one right here is a second generation wafer scale engine. We introduced this part and the associated system called our SRE versus CS2 in 2022. And we introduced the first generation machine, which we call the CS1, you can sense a theme here CS1, cs2. We introduced the first machine in 2020. And we've got a roadmap into the future of multiple wafers and multiple machines that will be coming in future years.

**Craig:** 28:21

Right now it's one wafer per system, per machine.

**Andy:** 28:24

Right now it's one wafer per machine and the work that we've done recently has allowed us to cluster these things together very, very efficiently. Like I said, we're building clusters right now of 1632, 64 systems and larger, and we're seeing really good scaling across those. When we train big AI models, we're getting 64 times faster than one in a cluster of 64. So right now, that approach is serving us really

well. That is one wafer in one machine, yeah.

**Craig:** 28:59

Yeah, and the cooling thing that's after I spoke to Andrew, I got a lot of emails and comments saying, yeah, that's great, but how do you cool it? And so that, I would imagine, was one of the biggest challenges how do you cool it?

**Andy:** 29:21

It was one of the biggest challenges, yeah, so unfortunately I don't have a picture today, but you can imagine, and your listeners can imagine, this thing's about the size of a dinner plate. The system that it goes into, that is, the physical box that it goes into, is about the size of a dorm room refrigerator or a kitchen mini fridge and it weighs in at about 600 pounds and about the top third and quarter of the machine is power IO and the wafer package, which we call the engine block itself is about the top third quarter of that tiny refrigerator size device. Everything else in the box is cooling. So the way we cool it is we actually have a closed internal water loop that is taking cold water, pumping it towards the back of the chassis, and then we have a series of custom built manifolds that take that cold water flow and distribute it uniformly across the back of the wafer where there's a copper cold plate. So we have a thermal contact to the back of the wafer that's got basically a series of micro etched grooves that are a manifold for the cold water to spread out across the back of the wafer, so that we're not only keeping the wafer cool but we're keeping the temperature of the wafer across the face uniform. Then we have warm water and we have to. We move that warm water by pumps down to the bottom half of the machine, which is basically a big heat exchanger. Andrew often makes a good joke about this heat exchanger that we're effectively taking the 1970s radiator technology and building it into a, you know, a 21st century machine. Sounds like a radiator.

**Craig:** 31:22

Yeah, Even 19,. Did you say 70s, even 20s Earlier?

**Andy:** 31:27

earlier. Yeah, and so you know if your listeners are gamers or PC builders. Right, if you imagine this, the machine in an X-ray view, it looks a little bit like a gaming PC on steroids, right? Big engine block and then pumps with an internal closed water loop and a big heat exchanger which is effectively a radiator at the bottom, and then the heat exchanger itself can be cooled either by facility air in a data center or by facility water. So most data centers that we deploy into today whether they're our own laboratory data centers or customers or cloud partners most of those data centers have water cooling and they'll end up, you know, bringing effectively a feed of cold water to the machines that then cools the internal closed water loop of the CS2. So, closed loop cooling with a heat exchanger that's then either cooled by facility water or facility air.

**Craig:** 32:31

Yeah, that's fascinating and just a fun fact that we talked about last night. This starts out as a disk. You cut it into a square and the ends are discarded and we were talking about all the ways you could use that discarded silicone and you've got to send me one and I'll figure out a business for using the. I love it.

**Andy:** 33:02

I love it. Cut me in. Yeah, these things would make great jewelry or artwork, and actually they are. So I don't know, craig, if you've ever been to our office, but if you come to our office, I mean, first of all, this thing looks pretty cool in its final state like this, but we also have in the office large prints of the microscopic view of each one of these reticles and all the different layers of circuitry, and those are quite beautiful also. I mean, it's a very regular geometric pattern, but when you sort of imagine what all these literally trillions of transistors map out to, it looks like a city grid in a way, and it's just. It's a really cool piece of visual art also. So, yeah, we got a couple of side hustles that we could work.

**Craig:** 33:58

The chip parts are art. One of the big topics globally right now, as these models proliferate and they get bigger, is the shortage of GPUs or the shortage of compute. You were saying when we spoke earlier that these wafers are on a separate production line or in a separate process at the FAB, so you're not constrained in the way that other chip makers are constrained. Is that right? Can you talk a little bit about that?

**Andy:** 34:42

Yeah so we still end up getting in line with our state of the art FAB partners, like other chip builders do, but we have a really well controlled and redundantly sourced supply chain after that. So we're not facing, we don't have the same supply chain complexities or risks or sort of single points of failure in vendors that other manufacturers might have. And so we today are still in a position where we can deliver systems in a reasonable timeframe, something like 90 days from order, whereas other system vendors might be quoting something like six months or nine months or even a year. And so, yeah, we're in a relatively good position from that standpoint. As a proof point to that, over the past six months or so only, we've built a system with our strategic partner and customer, g42, that we are calling Condor Galaxy One, and Condor Galaxy One is a cluster of 64 of our machines, so four exaflops of Sparse AI compute, which is and sounds tremendous, but really, at the end of the day, with this system like this, we've been able to train 10 to 30 billion in larger models in a matter of days or weeks. So it's a massive machine and it's built just in the past six months only. Yeah, yeah.

**Craig:** 36:21

How does the cost compare? Because, not being in the industry as a journalist, you wonder. Well, why isn't everybody training on Cerebrus chips?

**Andy:** 36:37

Sure. So the short answer on cost is that we are going to be price performance competitive or better than a cluster of GPUs, and that's just sort of the dollar capex outlay. After that we're going to deliver more performance to your users with that simpler programming model. So your users are going to be able to do more in less time and therefore your operational cost is also better from the AI development cycle and we also, because we didn't break apart our chip and reconnect it with literally hundreds of yards of cabling. We also have inherent power advantages from a power efficiency standpoint. So we're going to be at time of purchase, we're going to be price performance equal or better, and then we're going to pile on advantages for you as a user afterwards. So that gets at your question of price and cost. And then I think the question about adoption is really less a matter of price performance and more a matter of the fact that we're just bringing a fundamentally new tool into the marketplace, and in my head it's a little bit like inventing the wheel for the first time and bringing it into market. There is a massive incumbent system that's out there and we're an alternative to that and I think we have to prove to our customers and to our users and to the market that you can use this machine to train state-of-the-art models. And we've introduced a library of GPT models this year called Cerebers GPT, that prove that. We've released multiple models with customers like Jace 13 and 30 billion, which are Arabic, english, state-of-the-art GPT models. Btlm, which is a world-leading small language model that can run on devices, and we recently released a model called Crystal Coder in partnership with MBZU, ai and Ptoom that speaks not only English language but also speaks code. So we've illustrated that proof point through our customers and our own work that you can use these machines to build state-of-the-art models and you can see these performance and operational cost advantages. And I think that's the beginning right, that's our foothold in the market to then expand from there. So my hope is honestly, when we talk, maybe next year at the next NERP you'll see even more customers, even more models, and we also hope to make these machines available to more developers in the open source community so that they can start building their own right. And at that point I think we start to see the dominoes fall and we start to see a bigger and bigger chunk of the AI data center compute market falling in our direction, but we really have had to prove ourselves and show the world the value of a fundamentally new platform for this work.

**Craig:** 40:01

Yeah, and that's the most of the people, most of your customers, using it through the cloud or buying it and installing it on-prem.

**Andy:** 40:11

Great question. This has changed over time. The short answer today is that most of our users and customers are using these systems remotely, either directly through us or through our Cloud Partner Serious Scale. In the beginning it wasn't that way. Some of the earliest adopters of our machines were traditional super computing centers and research laboratories like the AI Research Group at GlaxoSmithKline, and Department of Energy Labs like Argonne National Laboratory and Lawrence Livermore National Laboratory. These folks have data centers. They know how to build supercomputers. They were very comfortable with the idea of bringing a new refrigerator-sized box into their data center and working with us to figure out how to power and cool that with their infrastructure. They're comfortable with that. Part of their charter is to adopt these new classes of machines and see what they're good at. Our earliest customers we were deploying direct on-prem. We still do that. If you're a customer that happens to have a couple of megawatts worth of power and wants to build a big cluster, we will do that. Alternatively, if you're a customer that is, say, a new AI startup or just a software-oriented enterprise organization that doesn't own or operate your own data center and is used to consuming compute through the cloud, we can roll with that too. We've built that out over the past year, basically just letting users be able to log in to our machines just like any other virtual machine, bring their data to the machines and get to work right away without ever having to set foot in the data center.

**Craig:** 41:54

Yeah, who are the cloud partners? Do you have your own cloud?

**Andy:** 41:59

Yeah, right now we can make our systems available directly. We could work directly together and get you remote access to machines in our data center. We also have a cloud partnership with a company called Ciriscale. They are offering pre-trained models as a service and training compute time as a service through their cloud. We've also, of course, been talking to some other cloud partners as we expand our reach. I should say also additionally, some of our super computing partners, like I mentioned to Argonne earlier, but we also have systems at Pittsburgh Super Computing Center funded by the National Science Foundation, and in Europe at EPCC, which used to be Edinburgh Parallel Computing Center in Scotland, and LRZ in Germany. Those centers in particular ANL, psc, epcc and LRZ also offer time on their machines to local and regional researchers. You can actually apply for grants of time if you're a researcher. While that's not a traditional commercial cloud kind of engagement, they are offering their machines up for researchers in that cloud consumption model.

**Craig:** 43:19

If a couple of questions, I can see the value for training foundation models on a Cerebrous machine. Are there use cases where you don't need the entire wafer? Can you partition the wafer and run multiple workloads on it?

**Andy:** 43:45

Yeah, great question. The short answer is yes For big foundation models and GPTs. The way we train is we load a mini batch of data onto the wafer and then compute one big model layer at a time. You can imagine these big rectangles, basically large matrices of compute, coming down and landing on the wafer where there's already data and the compute happens there. That's how we execute training for large models today. If the individual layer isn't that big, we can run multiple layers. If the whole model fits on the device, we can actually run the whole model on the device at one time, to your point. If your workload is smaller still that is, it doesn't require even a whole wafer then we have the capability to do that too. All of that is in software, effectively describing to the machine how to compile the compute job onto that array of cores and then how to move data to or through that compute job. The only reason I say that is because what we've really optimized the software for recently, particularly past 12 to 18 months, is for these large language model training tasks, these foundation model training tasks. We can run smaller models, but most of our users are focused on a lot bigger things.

**Craig:** 45:22

You also mentioned, along that same idea, that I remember exactly what you said that people can train models that run on the edge, did you say?

**Andy:** 45:40

that. Yeah, maybe there's two related points. One is that one of our design principles from the beginning, from a product standpoint, was that you should be able to take code that runs on a GPU and you should be able to run that on our machine. That is, if you train a model on a GPU, you should be able to train it on us. That's true for PyTorch transformer models today. You should also be able to take a model that was trained on our machine and run it elsewhere for inference. If you want to train a model on our machine and then run inference in the data center on CPU, or if you want to run inference on that model in an edge device like a phone or a MacBook, you can do that too. Yeah, I mentioned one of our particular models is a model that we developed with a customer that's called BTLM. It stands for Bittensor Language Model. It's a three billion parameter language model. It's a large language model, just not as big as the others. That three billion parameter model can, because of its size, once it's trained, it can run, say, on a MacBook Pro or another capable edge device For academic developers or people that are the sort of hobbyist developers or just researchers interested in highly efficient and capable small models. That's been a real hit in the market. It's one of the top, if not the leading model for its size posted out on Huggingface and the open source community. Because of that, your partners.

**Craig:** 47:30

I don't know how freely you can talk about them. Are you working with OpenAI?

**Andy:** 47:36

So we're working with a large number of partners, both in research and enterprise. We have lots of friends over at OpenAI. Right now, I'll say the biggest public partners that we've announced are, once again, the US Department of Energy Labs like Argonne and Livermore, pittsburgh Super Computing, glaxosmithkline, OpenTensor. Together. We also have recently announced just earlier this year a massive strategic partnership with a commercial company called G42, based in the UAE. They're a private company that is sort of the national champion of artificial intelligence in the Emirates. They're working with us to build large, state-of-the-art language models for Arabic, as well as large foundation models for medical clinical assistance. We worked with a part of G42 called M42 recently to fine tune an open source model that happened to be built by Meta on GPUs, one of their Lama models. We continuously pre-trained and fine-tuned that model with some medical knowledge and open sourced with M42, the world's first open source model that was able to pass the US medical licensing exam. So our partner G42 is working with us not just to build big compute clusters but also to build state-of-the-art models for things like Arabic language, medicine, coding, science and climate.

**Craig:** 49:26

Yeah, the reason I mentioned OpenAI is I've been talking to a lot of people in industry and what I'm hearing is that and it backs all the way up to the fabs but because of the limited GPU supply, openai limits the number of tokens per minute or requests per minute that you can push through the model on the API, and that is limiting or constraining enterprise scale production applications, because people can't go into production on something. If they're going to run into this bottleneck issue, it would. This seems to me, particularly if you don't have a supply problem, that it would be the answer to that. So are you talking to any of the really big you know, openai or Google or Amazon about? I guess Amazon's got their own solution but about using these as an alternative to smaller chips just simply to train and run inference on very large models?

**Andy:** 51:01

Right, a couple of points here. So, first and foremost, as a novel AI accelerator builder and a relatively new company in this ecosystem, yeah, we're talking to all of those organizations because I think at some level we all as a community recognize the need for, or the value of, alternative compute architectures for this really economically and maybe even societally important workload. You see Google working on TPU, you see AWS working on Tranium, you hear about other hyper scale and research organizations thinking about rolling their own silicon, because AI compute is a different workload than what we've seen before and in some sense, that's just a point that substantiates our founding hypothesis, which is that building and running these models in production on existing infrastructures okay, right, it got us off the ground, but it's not an optimal solution, and so we are talking to all of those folks about the value of cerebrous systems in their infrastructure. So that's one point, and I think, as I look into my own crystal ball for the future, I think it's inevitable in some sense that future data centers that are doing AI work or that are built for AI work are going to have different kinds of accelerators, that is, these are going to be heterogeneous clusters where you can imagine a world where you have many users that are bringing to your infrastructure AI problems of different shapes and sizes, right, big training jobs, small inference jobs, dense jobs, sparse jobs, even jobs that may be border up against, or also do HPC right things that couple, say, physics based simulation and AI. So we're going to see a broad landscape of pure AI or AI augmented workloads coming into the compute infrastructure of the future, and I think it's therefore very likely that on the back end you end up with a heterogeneous compute infrastructure that's composed of AI and HPC accelerators that are each best athletes in some sense in their own domain, right, and that in between you have a software layer that can interpret those inbound jobs and sort of decompose them into the requisite compute and communication and memory and storage attributes of the engine and then compile down to, maybe a combination of training jobs on a big CS cluster and then inference, maybe on some purpose built inference device. So we are having those conversations and I think that is sort of the direction that I see the industry going Then on. I wanted to make one comment also on business model right. Some organizations today that are doing, say, enterprise GPT training for customers end up keeping the model right. Like you asked about inference, I know, and so I don't want to dodge that question. I'll come back to it. But I also wanted to make mention of something that I think is sort of unique about our business model. That is, when you look at it say, existing businesses that do AI model training often as a customer you don't get to keep the model right. You end up the organization that built the model keeps the model and then you have to run inference there and I think that's you know. That business model has clearly worked for those organizations. But when you work with cerebrus, not only is your data safe so when you bring your data to, say, our cloud or our machines, it's kept secure by all standards but also we're not going to build other models with that data and when you have a fully trained model from our system, we give it back to you. So you get to own the weights and therefore, if you have your own, say, infrastructure for model serving, or if you want to fine tune that model a little bit further for another purpose, you can do that. You have full control and ownership. And then customers have come back to us and told us that they like this because it lets them in some sense, own their own destiny in their AI work. Right, their data is safe when they train and then when the model is done, they get the weights and they can hold on to those and of course, we hope that they come back to us to train again or update those models. But they get to own the outcome. And this leads back to your inference question, because it means that they can run inference, say, in a data center with us, or they can run, say, in their own cloud or in their own on-premise compute. Still doesn't answer the question as to whether or not we do inference, so I'll come to that now. The short answer is this is back to the discussion that we were having earlier about large models and small models. This chip doesn't care if we're running training or inference, so the chip can run inference and in fact, has some big architectural advantages in terms of being able to compute at high throughput with low latency, even at very small batch sizes. So we can run inference on the machine. Today we haven't written all the software to do that, so you can run eval, which is one form of inference, on our machine. Today. We also have some really exciting early work in inference for computer vision for some of our government customers who are interested in processing lots of imagery and video feeds. And last but not least, our team today is working on some architecture studies to be able to support large language model inference in the future. I think it's unlikely in a sense that we will be a fully general purpose inference engine, but we're taking the same approach as we did to training to inference. That is, really taking a very close look at the variety of inference workloads, what each one of them is asking of the underlying machine, everything from computer vision inference on, say, tiny images to inference on large language data sets. And from a product standpoint then we're making decisions about where to focus first, where our architecture can give the biggest performance value back to users. So long story short, yes, and stay tuned.

**Craig:** 57:46

Yeah, and you mentioned Dubai and we spoke yesterday about, you know, this evolving or developing idea of a national compute resource for AI in the United States, and whether that's cloud credits or physical data centers. As you know, and I'm a believer that this generative AI and behind that, other large model technologies are going to change fundamentally change economies and even geopolitics Are you talking to nation states about building national compute resources that then, would you know, power their economies, the AI economy, and including the United States?

**Andy:** 58:58

Yeah, this is a great question. I think, following our chat last night, it's an area that I'm really passionate about. The short answer to the question is yes, and I think, like you know, we at Cerebrus see massive and positive potential for this technology to fundamentally change the way that we interact with data, make decisions on a daily basis. Everything from retail applications and things like recommendation engines that find good cat videos on YouTube to models that can help predict the next pandemic outbreak or help us develop smarter vaccines faster, or help us figure out pathways to more efficient and cleaner renewable energy generation and environment right. So I think there's tremendous and yet fundamentally untapped potential for these large AI models, both for economic and good, and I think the way that we discover these positive applications as a society is by accelerating our research and making the tools for that research more broadly available. And it comes back to my remark about how do we make systems like these accessible to more users, and that's the broad base, right. But at the meantime, back to your question we're also seeing around the world big enterprise companies and big enterprise nation state governments sort of having that aha moment and realizing that these technologies could be transformative for economy and environment and provide tremendous return on investment for their people and for all of us. And so, yeah, we are seeing a lot of interest in our systems from both large commercial enterprise and from governments to help them think about how to build that right compute infrastructure and even help building models on that infrastructure as part of their AI initiatives. And we are directly talking to the US. As you're probably familiar I think we chatted a little bit about it last night the National Science Foundation has announced a program called the National AI Research Resource, or NARE. That was even called out in the administration's recent executive order as a priority for the nation. So we're engaged and talking to our friends at NSF and NARE about that, and also in the US, the Department of Energy is formulating its own position. You know, these are the folks that have built generations of world leading supercomputers, historically primarily for physics based modeling and simulation, but know how to build these sort of massive world class computational infrastructure. The Department of Energy is also looking at how to build that next generation of supercomputers to power the US AI research initiative, both for science and for security, and so we're deeply engaged with both of those communities and at a personal level. I think it's something that can really benefit the nation, and so much of this technology like ours and GPUs have been developed here right, and so I think we're in an extraordinary position in the country and have an extraordinary opportunity, therefore, to lead the world in this with our partners and co-develop the next generation of capable and efficient, safe and responsible AI methods, and I think the way that we do that is by starting at the infrastructure level and then putting that infrastructure into the hands of users that can ask and answer questions and build those solutions. I'm stoked, I'm thrilled for what the next several years and the next decades in this field is going to bring, and I couldn't be more proud to be a part of a team that's bringing potentially one of the platforms that powers this into the future. It's cool stuff.

**Craig:** 1:03:41

It's exciting. I'm very interested in longevity research because I want to be around as soon as possible.

**Andy:** 1:03:50

We got to build you a cluster.

**Craig:** 1:03:52

Let's get to it. I want to give a shout out to our sponsor this week, babel, the science backed language learning app. Babel is something I feel strongly about, because language is the key to opening the world and broadening horizons. I know because, as a journalist, I've reported out of more than 40 countries around the world and it's amazing how knowing just a few words of the local language will open doors and build bonds. Be a better you in 2024 with Babel. Don't pay hundreds of dollars for private tutors or waste hours on apps that don't really help you speak the language. Babel's quick 10 minute lessons are designed by over 150 language experts to help you start speaking a new language in as little as three weeks. Babel's designed by real people for real conversations. Babel's tips and tools are approachable, accessible, rooted in real life situations and delivered with conversation based teaching, so you're ready to practice what you've learned in the real world. It's so easy to learn how to order food, ask for directions, speak to merchants, without having to consult language apps while on vacation. The key to learning languages is not to be embarrassed for speaking poorly. Plus, babel's speech recognition technology helps you to improve your pronunciation and accent, something that I need a little work on. Studies from Yale Mish state university and others continue to prove Babel is better. Right now, get 55% off your Babel subscription, but only for our listeners at babelcom. Slash ionai. That's it for this episode. I want to thank Andy for his time. If you want to learn more about the conversation today, you can read a transcript on our website. Ionai, that's e-y-e-o-n dot ai. In the meantime, remember the singularity may not be near, but AI is fast changing our world, so pay attention.