



## Season 3, Episode 5:

# Beyond Exascale: Exploring Emerging Technologies

### SPEAKERS

Sarah Webb, Luis Ceze, Bert de Jong, Katie Schuman

#### **Sarah Webb**

Just over a year ago, the Frontier supercomputer at Oak Ridge Leadership Computing Facility became the first system to cross the exascale barrier on the TOP500 list, achieving more than one quintillion floating point operations per second. The exascale era is still coming to fruition as the Department of Energy's Aurora and El Capitan systems near completion and researchers continue to develop software and compilers that can efficiently harness thousands of nodes that include CPUs and GPUs.

#### **Sarah Webb**

But entering the exascale era leads to the inevitable question of what's next. Just as generative AI is showing capabilities that could reshape science and society, computing is facing a crossroads, the end of Moore's law. Computing and its applications will need faster, more powerful processors. But the machines that do the work will change altering software and programming, too.

#### **Sarah Webb**

I'm your host, Sarah Webb, and this is Science in Parallel, a podcast about people and projects in computational science. In this episode, we'll examine one part of this puzzle, emerging hardware that could have a place in future systems. We'll discuss quantum devices, neuromorphic, chips, and molecular storage and processing. I'll be sharing parts of my conversations with three separate guests, Luis Ceze, Bert de Yong and Katie Schuman. They'll share their perspectives on individual technologies and where they might fit in the high-performance computing ecosystem. What are their strengths? What are the challenges? And how will we know when new processors are ready for prime time?

#### **Sarah Webb**

We'll set the stage with Luis Ceze, who wears dual hats as both computer science professor at the University of Washington and CEO at the AI startup Octo ml. He describes himself as a computer systems architect, a role he explained to me.

#### **Luis Ceze**

Computer architecture is not just about making a better chip. It's really thinking from a whole systems perspective is how do you co-design software and hardware? How do you think about future

applications and enable them? And how do you think about thermal aspects? You're gonna get so hot. How are you going to cool it and all of these things?

**Sarah Webb**

I asked Luis to describe our current situation in high performance computing.

**Luis Ceze**

Yeah, I mean, we've enjoyed an incredible run in the progress of basic CMOS electronics. We have exascale systems today, Not only that, it's just seems common right to have petascale computing devices. And I think, interestingly, we have really exciting applications to drive high performance computing systems, not just in very specialized settings in in science, in scientific development, in R&D, but also in applications like AI. And you've seen just recently we've seen this transformation just in the last couple of months in people using these massive-scale language models like ChatGPT and GPT-4 that does amazing things. And those are just about the most compute-intensive applications that captured people's imaginations in the history of computing. It's also very exciting, because now we have exciting applications.

**Luis Ceze**

And we have developed methodologies to design very complex systems that are highly heterogeneous. Today, you not only have, you know, CPUs, but also you have specialized compute units in the form of GPUs and specialized AI accelerators and accelerators for cryptography, compression. And so we've designed very complex systems. That's great-- that's going to take us a bit further building better and better computers. But I think everyone probably heard it multiple times that we are nearly the end of scaling of CMOS, you know, some people call that the end of Moore's law. But people have been predicting the end of Moore's law for a while. It's unavoidable that technology is getting close to its limit because you cannot miniaturize beyond a certain point. Emerging technology is just unavoidable if you're going to continue to build better and better computers.

**Sarah Webb**

Bert de Jong of Lawrence Berkeley National Laboratory made a similar point in our conversation about quantum computing.

**Bert de Jong**

One of the things that I realized about eight years ago is that classical computers cannot scale forever. It's just we cannot make them smaller; we cannot make them faster. There's too much energy that we have to put into that. So I started to look more at what are other simulation, computational tools that we could use. That's where AI and quantum computers came around, and it's like quantum computers has a lot of opportunities. They have the opportunity to do simulations that are of an exponential complexity, which means they can do a lot of computation much quicker than they can do on a classical computer.

**Sarah Webb**

Over the last several years, quantum computing has become a key part of Bert's research.

**Bert de Jong**

I do a lot of virtual experiments on computers. So I do chemistry-, materials-type of experiments that you would do in a lab. I just do them on a computer. And so my interest really is trying to solve important scientific problems relevant to society. Can we find better batteries, a better way to capture sunlight and convert that to energy, capture greenhouse gases, and all of those kinds of big scientific challenges.

### **Sarah Webb**

Quantum computers are exciting because subatomic particles can exist in multiple states at the same time until they're observed, a property known as superposition. Those same particles can also experience entanglement when the behavior of one particle is connected to that of others distant from it. For computing, that means the qubits -- the basic units of a quantum system -- are not tied to simple 1s and 0s.

### **Bert de Jong**

The challenge with quantum computers, which we realized when I started to work on that, is, one, they are noisy. And, two, we don't really know how well we can run calculations, simulations on these quantum computers. So that's where I got really interested in trying to find better ways to run simulations on quantum computers, understand where the challenges are, and actually develop strategies and approaches and algorithms and software to make quantum simulation or simulations of chemistry models to actually get chemistry simulations run on quantum computers.

### **Sarah Webb**

In addition to his own research, Bert is deputy director of the Quantum Systems Accelerator, one of five centers funded by the Department of Energy as part of the National Quantum initiative. QSA is led by Berkeley Lab and includes 15 institutions across the national labs and academia.

### **Bert de Jong**

What we're really focusing on is scaling up and building more accurate quantum computers in three different types of superconducting qubits, trapped ions and cold atoms.

### **Sarah Webb**

I'm going to pause briefly and include background information from a version of my interview with Bert that appears in ASCR Discovery. There's a link to that in our show notes. Superconducting qubits are qubits constructed with Josephson junctions. Those structures, weakly link superconductors, and current flow between them can produce superposition and entanglement. Trapped-ion and cold-atom qubits are both trapped-based approaches. Trapped-ion qubits are held in place by an electric field. Cold-atom, qubits are trapped by tweezers made of light. Lasers coax entanglement and rotate qubits between 0 and 1.

### **Bert de Jong**

Why are we doing all three? Let's say we learned something in trapped ions. It's something that we potentially could apply to a quantum computer build superconducting qubits. So we're really trying to not just build architectures to better qubits, so to speak, but also trying to figure out how we can scale them up to large number of qubits. To give you a couple of examples, we've been there for about two

and a half years. We actually have a cold-atom system that is of the scale of 256 qubits; we've been able to do real material science simulations at that scale. It's a very different technology and a different approach to quantum simulation compared to trapped ions and superconducting qubits.

### **Bert de Jong**

Where with the trapped ions, we just are in the process now of building the first trap that can hold about 200 ions. So that gets to the same scale as what I just mentioned for the cold atoms. If you look at superconducting qubits, we are now in the order of 25 qubits that we can actually do. And it's not just building more, but also trying to find better ways to control the qubits while you have them. You need to operate on them. So how do you do that with large numbers of qubits? That's a completely different problem. So we're working on control software, control hardware to be able to run these kinds of qubits. Eventually, we also need to be able to build these in larger systems, because we cannot just grow one trap or one system continuously. So we have to start thinking about what do we do next? If you have 1000, 10,000? How do we actually group them, communicate between them, operate on them? That is lots of challenge, and then that's more on the hardware side. We're also doing a lot of work on the algorithms and software side? Because eventually, even if we built his quantum computers at this scale, we want to actually solve real scientific problems. Well, there's lots of questions one can ask: How do you know that you get actually a quantum advantage with a quantum computer? So we have been working on developing protocols to actually quantify if you're getting a quantum advantage with a quantum computer.

### **Sarah Webb**

Work at the Quantum Systems Accelerator has demonstrated a theoretical advantage on a trapped ion system with 15 to 16 qubits. Bert says,

### **Bert de Jong**

We do a lot of research around how we can run real science on there. So we have some research programs focused again on the chemistry materials, but also in nuclear physics and high-energy physics. And trying to understand how we can actually use these quality computers within the quantum system accelerated to do your real science.

### **Sarah Webb**

Neuromorphic computing offers a completely different approach and other strengths, including the potential to use far less energy, up to thousands of times less power than today's CPUs and GPUs. I talked with Katie Schuman, an assistant professor of computer science at the University of Tennessee in Knoxville, about that.

### **Katie Schuman**

You might be able to tell from the name that neuromorphic means brain-shaped computing. And so in the field of neuromorphic computing, we're looking to rebuild computing from the ground up starting at the materials and devices all the way up to the way that I think about it from a computer science perspective.

### **Sarah Webb**

Katie started working on neuromorphic computing and machine learning as a graduate student, and for six years at Oak Ridge National Laboratory, before taking her new job last year,

### **Katie Schuman**

The algorithms and applications rethink every aspect of the compute stack and look at all of it from a more brain inspired perspective. So we're talking about looking at new materials that can more accurately replicate what's going on in biological neural systems, all the way up to what I'm working on, which is figuring out what the algorithms are that we can leverage these properties of brain-inspired computing to do things like machine learning, but also computing applications more broadly in a new way. My personal research is really diverse because I interact with people who are doing hardware development and coming up with new ways to build computers. But I also interact with neuroscientists to take inspiration from the brain into how we're developing the way we're doing computing. Then I also work with people who want to use neuromorphic computers to do something interesting. So my research is all about coming up with the new approaches to do computing, but also figuring out how to design the future hardware to meet the computing needs that we have maybe for an edge computing application. How do we actually make them do something useful? And are they better than other computing applications or computing systems for those applications? Those are the questions we're trying to answer.

### **Sarah Webb**

Like neurons, brain inspired hardware typically spikes to send a signal after it reaches a threshold level.

### **Katie Schuman**

One of the big advantages to neuromorphic and brain-inspired hardware more broadly, is the energy efficiency aspect of it. We're really targeting orders of magnitude lower power than conventional computing systems. And so with that particular property in mind, there's a lot of opportunity to put neuromorphic at the edge for edge applications, everything from your smartwatch to your smartphone. So you know, conventional things that people are interacting with, on a regular basis, your smart speaker at your house, things like that towards, you know, smart sensors that are deployed out in the world. As we do increasingly integrated and smart cities and transportation, we really are going to need these very low-power computing operations. Because we are increasingly devoting more and more power to computing.

### **Katie Schuman**

One of my favorite projects that we're working on right now is a collaboration with researchers at Oak Ridge National Lab looking at applying neuromorphic computing to an edge computing application in the nuclear radiation detection space. And this is an application where they want a very, very low power implementation, they could operate at the edge in the field indefinitely on like a small solar panel. And so the question was, for this particular application, can we come up with the right machine learning approach that will map to neuromorphic hardware and that will meet the power requirements for that edge application and will get as good of an accuracy as other techniques all at the same time? And so having all of that constrained space and working directly with the domain experts in nuclear engineering and dealing with both simulated data and real-world data, it's really exciting to try to figure out how to

make all of these pieces come together to build a neuromorphic computer that can actually operate on a real-world application.

### **Sarah Webb**

Katie is also working with transportation researchers who are exploring neuromorphic computing, for self-driving cars, and to optimize fuel flow within engines.

### **Katie Schuman**

We're also working with people in the high-energy physics community where we're looking to apply neuromorphic computing to monitoring data that's being collected in real time at their large-scale detectors. My meeting schedule on any given day, I'm pivoting between a discussion about radiation detection to internal combustion engines to neutrinos, high-energy physics just in applying neuromorphic computing.

### **Sarah Webb**

Katie also works with researchers who build neuromorphic hardware and want to apply it to relevant problems. For example, she's working with a large group of researchers and institutions, including Sandia National Laboratories, the University of Texas at Austin and New York University, on emerging device types with magnetic tunneling junctions and tunnel diodes. They're potentially interesting for neuromorphic computing, but they also operate in an inherently random way, which could be useful for probabilistic computing. That's another strategy for addressing problems that are difficult for classical computers.

### **Katie Schuman**

Think about using these devices as like flipping a biased coin, so that you're more heavily weighted to get heads than a 50/50 shot. So that project is actually called coin flips because we frame everything in terms of flipping these biased coins, which is how these devices function. And so with that project, we're working with material scientists, and physicists and electrical engineers who are working with the materials and the devices level to build these. And then I work at the architecture and algorithms level as we're trying to come up with the circuits and the algorithms that can actually benefit from using these devices at the low level. And again, then we're interacting with the application users who would ultimately use these. And so from that perspective, it's very much like we're starting with the hardware. And then what can we do with this really cool hardware implementation, rather than the opposite side. Where we're starting with the application than nuclear engineering? And saying, how do we make the hardware that's going to do that? And with these two different sets of projects, the hope is that we can bridge the gap between the two of them and actually say, "Oh, well, maybe that nuclear engineering project will work with this particular set of devices." I also work with people who are doing quantum materials-based neuromorphic systems and biomimetic materials-based neuromorphic systems and memristive neuromorphic. So all these weird devices that I as a computer scientist only sort of know what's going on, but enough to where we can try to figure out how to use them effectively.

### **Sarah Webb**

Luis Ceze, who we heard from at the top of this episode, spends a significant portion of his time working on molecular data storage and computing, yet another emerging technology.

**Luis Ceze**

We've been using DNA, but also been looking at proteins, too. These molecules can do amazing things, right. So they can store information in a very dense way. And you can also manipulate information in a very parallel, an energy efficient way. It's not particularly fast in terms of latency is extremely parallel.

**Sarah Webb**

There is an incredible amount of biological data stored in DNA.

**Luis Ceze**

So nature evolved DNA to store data that maps to genes, right. So DNA contains blueprints to build proteins and proteins are, do all sorts of amazing things in living organisms, right? So, but DNA is a fairly general data storage molecule just four bases, ACTG, you know. If you can develop mechanisms to map digital data strings of 1s and 0s into strings of ACTGs, and then manufacture the DNA molecule physically make the molecule that encodes that information, you can store them away. And that's when things get interesting. DNA is extremely dense, like each base is on the order of a fraction of a nanometer like 0.34 nanometers that you're talking about densities that are thousands and thousands of times denser than the limits of what we could get even with 3D flash storage, for example, to denser than anything that we have in mainstream digital data storage today. And that's the process of writing.

**Luis Ceze**

And when you read it, you run it through a DNA sequencer—the same exact sequencer used in genomics—you get long sequences of ACTGs back, and you can decode that back to the digital information. And you know, this is just amazing. Look, this looks general, but it took a while for the technology to write and read DNA to mature enough for this to start to be viable. DNA storage started becoming more and more interesting and viable in the past decade as our ability to write DNA improved very, very fast, in cases faster down the progress of CMOS electronics that we just talked about. So the process works is becoming closer and closer to reality, right? So I've simplified a lot how this works. In fact, there's quite a bit of computing involved in mapping bits to DNA that relay it back. And guess what, that uses a lot of AI as well, because it's a very, very noisy environment. And the way you sift the actual information through the noise is by applying a lot of error-correcting code and a lot of AI-like techniques to encode and decode the information. So our lab has been doing quite a bit of work in this, but there's several other, you know, significant contributors around the world. There is a, an alliance today called the DNA Data Storage Alliance, and that shows that this is this is becoming closer and closer to reality,

**Sarah Webb**

The DNA Data Storage Alliance was founded in 2020. By Illumina, Microsoft, Twist Bioscience and Western Digital Check out the link in our show notes. A limit for molecular systems is that the process of writing and reading DNA molecules remains slow compared with the speed that today's computers can manage electronic memory. Luis notes that it's improving rapidly but remains at least 10 times



slower than today's electronics. At the high end, researchers have stored and retrieved several gigabytes of data using molecular methods.

### **Sarah Webb**

But molecules offer some advantages for certain types of computational problems, Luis talked about how his group has harnessed molecules' ability to bind to one another as a strategy for doing similarity searches the heart of many AI algorithms. For example, in searching for an image of a bird, a computer isn't looking for an exact match to an ostrich or a hummingbird,

### **Luis Ceze**

You don't map it by let's just say birds: blue bird, red bird, different species and all they look like birds. And so that doing that similarity search in regular computers today is pretty computationally intensive, you have to compute distance in this high-dimensional space, and we demonstrated the ability to do that in directly molecular form.

### **Sarah Webb**

That's because they realized that a similarity search algorithm can be matched to molecules, taking advantage of characteristics such as shape, charge, and other features that allow molecules to bind to each other.

### **Luis Ceze**

So we developed algorithms that map feature vectors, descriptors of images, into DNA sequences, such that if the vectors that correspond to images are similar enough in their high dimensional space, their corresponding molecules are more likely to bind to each other. So we built essentially this pool, a database of a bunch of images encoded in this form and put a magnetic particle with a DNA attached to it, that corresponds to the image that you want to search. The reason this is interesting is that here's an algorithm that is very, very data-intensive. Like when people do this today, in regular computers, there's over trillions and trillions of images, that takes quite a bit of compute. But now in molecular form, it just happens in a very parallel way, a very energy-efficient way. That's one clear example of why molecular computing, then now putting on my computer architecture hat, looks very interesting. But it's not general. Like we're not saying multiply a matrix or evaluate this expression. We're talking about execute this search, this this search stat. It's very, very specialized, but it's very efficient to do in molecular form.

### **Sarah Webb**

The potential for all these technologies is tantalizing, but as hardware is improving and changing that affects software. At the same time software developments influence hardware.

As I was talking with Katie, I was trying to find an analogy to express that idea.

I'm not sure if this analogy is appropriate, but it's almost like you're having to think about constructing the highway system and the cars that go on it all at the same time.

### **Katie Schuman**



Absolutely, yes, this is the most challenging and most exciting part of this field is that we're building software for computers that don't exist yet. And we're building computers for software that doesn't exist yet. And everybody's just out there working on it at the same time, and occasionally, we're talking to each other. That's something we're trying to do in the neuromorphic community is really do this collaboration between the two communities, so we're doing a co-design process. But that's not happening across the board for all of these emerging processors. but it's increasingly going to have to happen where you have to be having these discussions so that we're not just building them independently and that we end up at the end with a car that really doesn't map to the highway that we built, for example, or vice versa.

### **Katie Schuman**

Neuromorphic computing hardware is still very much in its research and development phase. So large-scale neuromorphic chips exist, but really only for the research community, and they're still very much under development. And so it's a challenge because the hardware is constantly evolving. And we're trying to you know, I'm the programmer trying to be like, what does it look like today, the hardware, what's changed? So that's challenging, but at the same time, it's a, it's an opportunity for us to say, Okay, I understand the hardware changing, could we change it this way, so that it is better for my application? I always like to frame my challenges as also as opportunities to remind myself this is really hard, but it's also an opportunity to do something really interesting.

### **Katie Schuman**

The other major challenge is that we're still actively developing the algorithms for these systems. So nothing is set in stone, all of the algorithms that we've developed over the course of machine learning, and all of the advances that we've seen, some of those translate into neuromorphic computing, but not all of them. And so we have to make adaptations. And we have to shift and change the way we're actually applying machine learning algorithms to our data for this hardware and looking to neuroscience, for inspiration on how to do that. Again, that's a huge challenge that I think that is the big challenge that I work on is how to actually make those algorithms do something useful for us. But it's also an opportunity to pull in all of that inspiration and potentially innovate on what we can do with our machine learning algorithms, particularly on these sorts of applications where we have a resource-constrained environment,

### **Sarah Webb**

Similar challenges exist in quantum computing, Bert and I discussed the problem of quantum software and translating science into a programming language.

### **Bert de Jong**

It's not a trivial thing to do. There is no programming languages that allow us to just take a problem and then say: Here, write it for quantum computer, compile it and it runs on a quantum computer. That didn't exist. So we started to really work on how can we build compilers? How can we build programming languages? So we can take a scientific problem [and] translate it on aquatic computer.

### **Sarah Webb**

And then there's addressing noise from the science that underlies these experimental processors. Bert also talked about how this plays out in quantum computing.

### **Bert de Jong**

We're doing effectively physics experiments when it comes to all of these quantum technologies. Well, physics experiments don't happen in isolation. Nothing around us lives in isolation, we always have an environment that perturbs the things. So that's no different for these quantum systems. They're just a lot more sensitive to it. So in case people don't realize this, but classical computers, the chips have gotten so small that these chips make errors. So what do we do? We actually do error correction. So we run three, four or five of the same experiments at the same time and just do a majority vote to decide what the right answer should be. Well, doing that kind of an approach on a quantum computer is a lot harder. So you could then you could do full error correction. But we know that if you want to do full error correction, knowing how much noise these quantum computers have right now that we would need tens of thousand or hundreds of thousands, potentially millions of qubits to even have maybe a couple of 100 really good working qubits. So we need a lot of other qubits to make sure that we can correct the errors that we see. So the root of it is, as I said, it's physics. We tried to engineer as much as we can the noise away. So we learn what the noise source is, and find a solution for it. Early superconducting qubits were found to be very sensitive to electromagnetic fields, for example. If you came by with your phone, you would create some problems. So they put copper shielding around it to minimize those effects. But even the smallest effect, at these superconducting, low temperatures can perturb the system. We're never getting a perfect quantum computer. We don't even have a perfect classical computer.

### **Sarah Webb**

So Bert has focused on solving the error problem.

### **Bert de Jong**

So what we are trying to learn is how we can control these errors, how we can mitigate these errors both on the hardware software and algorithms side. How can we correct these errors? How can we build algorithms that are more efficient, so that we don't have to do as many operations that will lead us to get a more reliable result? We take a science problem and try it on a quantum computer, see where it breaks down, learn from it, develop new ways to get more reliable results out of a quantum computer, and then try that. If it still doesn't work, we add another layer of error mitigation, till we get the reliable end so that we can get. That's just us effectively learning how a quantum computer works, where its flaws are and finding ways to do it better.

### **Sarah Webb**

Luis Ceze and I also talked about noise and what that means for molecular storage and computing.

### **Luis Ceze**

I risk sounding overly philosophical, it may seem like it's a problem. But I think it's also it's just inherent. That's how you build robust systems. And it's hard to argue something more robust than what nature has. Sure we still have, you know, single points of failure in organisms. But, by and large, they're very, very resilient and robust, especially on a population scale. So the way we deal with that is, first of all, to

remind, you know, the audience here that electronics in general, is, is noisy, right. So the reason that we did the we didn't build more analog computers is because our computers are noisy. So we go in and overlay a digital abstraction on top of it to deal with noise. So that's like a very, very, very basic level. And then on top of a digital system, they can still be noisy. You build error-correcting algorithms, your build, essentially give an abstraction of very robust writing and reading information in whatever medium you have. So that's whole feuds in coding theory and how you encode information in a in a reliable way. But then even on top of that, it can be the algorithms that are inherently, by the way they're constructed, robust to imperfections, right. So we just talked about similarity search. The whole point there is that you're going to find something that's similar enough, not hat it is exactly, you know, a perfect match. And in a lot of modern machine learning algorithms work when you train them, it's all about approximation. It's all building essentially interpolating functions, building functions that can make predictions out of a large collection of data that's deemed noisy to begin with.

### **Sarah Webb**

In molecular storage and computing, Luis emphasizes that large amounts of data aren't encoded within single molecules.

### **Luis Ceze**

So what we do is we chop this up into small little chunks. Each chunk is on the order of say, 100 bits or so. And each chunk itself has error-correcting methods that we encoded in a way that uses more than 100 bits, but there are several mappings of different bits to the original information. So even if you have some bits that are off, you can see reconstruct it. So there's error-correcting code within each one of the chunks, but then across chunks, we add extra chunks such that if any chunk is missing, you can still recover the whole information. So it's both within and across. You don't have a single copy of every given item, right, we can have quite a few, you know, think on the order of tens or hundreds of physical copies of the same original molecule. Each one of them might be slightly different, but they originally mapped to the same intended one. So now we have three forms of redundancy, within this chunk, across chunk and in physical redundancy. And with all of that we demonstrated by really stretching to the limit the ability to recover information with all sorts of adding here, removing molecules, and entirely, you know, diluting it to an extreme degree, we've been showing, demonstrating very, very strong robustness in writing and reading information.

### **Sarah Webb**

An important challenge in this work is figuring out software, not just for quantum systems as Bert mentioned, but developing compilers and frameworks so that researchers outside these specialized areas can use new hardware. Katie emphasize this issue in neuromorphic computing, which is a field of just a couple thousand researchers, many of them are hardware developers, or people interested in applications rather than software developers.

### **Katie Schuman**

What we are missing right now more than anything, from my perspective is the software. We really need software frameworks in place. We need things like compilers. We need software abstractions to be defined that just don't exist right now in the field. And that's partially a consequence of the makeup of the community because we have a lot of people who are working on hardware. And then we have

some people like me who are more focused on things like machine learning algorithms are people who are interested in applications. But what we don't have a lot of— in fact, hardly any of— people who know how to build system software, who know how to build things, like compilers, who know how to build programming languages, who know how to define these abstractions. Even though I'm a computer scientist, that's not my background. So I'm not equipped to do that effectively for a new, emerging hardware platform. In neuromorphic, we have the hardware. It exists— people are building it all the time. We have these research chips that you can have access to. What we don't have the software tools in place that make it easy for people to come in and start using that hardware on their new applications quickly. We need the ability to build out the software tools so that anybody can sit down and start using a neuromorphic computing system, and not just people who are experts in the field.

### **Sarah Webb**

It's incredibly difficult to put a timeline on when and how quantum, molecular and neuromorphic systems could enter the mainstream computing landscape. But I asked about innovations that could signal that they're ready for broader use. Here's Luis on what that could look like for molecular storage.

### **Luis Ceze**

I know folks have been using this to store very precious information as you can think of from things like cryptocurrencies that you really want to store completely off the grid, let's say to things that are nation-states information that is secret or, you know, family information is very precious. I call this boutique use cases. And that's viable today. But the U.S. government has been very active in investing in molecular data storage technologies, and industry has been very active, too.

### **Sarah Webb**

But he thinks a jump to nearly terabyte-level storage would be the point where molecular technologies will become much more interesting.

### **Luis Ceze**

My academic speculation here, I think we're going to see something really interesting within the five-to-ten-year range where we're going to select pretty exciting archival storage applications, there. You're going to start see applications like data-intensive compute tasks, like search appearing and being interesting. Can you imagine large repositories of media information stored in DNA and searching for it for various purposes? I think that's probably within the one to two decades away. We tend to overestimate what we can do in the short term and underestimate what we can do in the long term.

### **Sarah Webb**

Bert highlighted scientific milestones that could signal that quantum systems are large enough and robust enough to make their mark.

### **Bert de Jong**

In the scientific community, if we can use quantum computer to discover a new solar-cell material that is 10 times better, that would be a metric for us chemists. We have set for a long time a benchmark around what's called the FeMoco system. We're actually trying to find a natural way to convert nitrogen

to ammonia, which is a very important fertilizer problem, which takes like about a percent of worldwide energy these days, and we're trying to build a natural alternative to that. If we can actually use that to build an alternative, that would be a win. But are we there yet? That requires 100 to 200 very good qubits, and we're not there yet.

### **Sarah Webb**

I asked each of them to talk about what future supercomputers might look like. Would emerging processors be integrated into systems that also have GPUs and CPUs? Or would we see smaller systems focused on specific applications? And the key take home message here is heterogeneity, and here's what Katie had to say about that.

### **Katie Schuman**

I suspect we're always going to have the very large scale systems. But right now, for the most part, when you look at a large-scale system, like Frontier at Oak Ridge, all of your nodes in Frontier are the same. They have heterogeneous within the node, though, so there's CPUs and GPUs, but every node is the same. I suspect that in the future, our nodes are going to be heterogeneous across the compute system, where not every node is going to have a neuromorphic system, not every node is going to have a quantum system, but some of them are. You're just going to have more heterogeneity not just within the node, but across nodes.

### **Katie Schuman**

And so I think these systems are still going to be large, because there are certain applications that map well to those types of systems. But I think that they're going to be increasingly heterogeneous. And it's not just going to be every node is identical, but you're going to have a lot of specialization across one machine. Now, how we program that and how we actually that's a big challenge that the computer science community is going to have to grapple with. But I think that's the direction that will end up going is that, yeah, maybe you're not going to have a quantum computer in every node, but maybe you've got a couple in your center that are integrated in as part of your system. And you want to be able to use those effectively, as you do everything else.

### **Sarah Webb**

Luis agrees and points out the relationship to the question of where one computer ends, and another begins.

### **Luis Ceze**

I think it's going to be highly heterogeneous. It already is today. So it's heterogeneous with a given specific device technology, now we're going to have multiple device technologies, and we should also say that it's not just quantum or neuromorphic. So it's also, you know, 3-5 materials are coming back. People are talking about cryogenic computing. There is all sorts of different, say lower-level, device technologies here, that's going to be part of the mix. That's just unavoidable, it's already the case is going to, it's going to continue to be the case. I think that, depending on the on the type of problem, this might be actually multiple, let's say, boxes, each one with their own specific technology that you farm out and combine them later.

**Luis Ceze**

Or I can also imagine this in a single device. I guess I'm trying to get at here probably is that what is a computer anyway? These days, like when you run a task, if you go and run a piece of computing, even if it's just like a highly specialized. HPC-like application, okay, so you do the large amount of computation in certain box somewhere, probably the national lab, you get that output, you process it in different way. Now using a different computer and you go and render an image. And then you know, you have a like, what is the computer there? Are you calling the computer just what did the majority of the computation? Or everything in between, right? So it's already highly-distributed anyway, even things that seem like we're just using one computer, you're not. You're probably using, you know, 10 or 100 different computers in between when you give the input and get the output back. And I think we're going to continue seeing that, you know, so but to an extreme degree. And given that some of these more exotic technologies that we're talking about requires very specialized conditions, like a quantum computer running in these dilution fridges. So in molecular computers, or more likely data-storage systems, are going to have fluidics will be around. I think it's unlikely we're going to have this in our pockets or even in our offices. We're going to have this in a specialized data center that can provide those conditions. In the end, it's going to be a matter of building a system that's an outer loop that puts everything together.

**Sarah Webb**

Bert speculates that novel processors could influence today's hardware in ways that are hard to anticipate.

**Bert de Jong**

When we went to the moon what we had to do is develop enormous amounts of technologies, including actually chips that we use in our computers that resulted from the vision of going to the moon. I see the vision of quantum also having that same role. We learn a lot about the role of quantum in physics and in building chips. So there might be a future that things we have learned show up in classical computers. So we might, because of the work that we're doing in quantum, develop revolutionary new technologies that could be used in classical computing. We're using fast internet right now to communicate between different classical computers. If we figure out how to do quantum networking a lot more efficiently, maybe a classical computer will be having a quantum network in the future. So there is a lot of technologies and advances that are being made that I see potentially ending up in the world of classical computing. So it might not be just that we have a classical computer and a quantum computer, we might get into a situation where the classical computer is starting to look a lot more like a quantum computer, because we're starting to integrate components that we have learned to harness as part of the quantum computing revolution.

**Sarah Webb**

Luis emphasizes that generative AI and large language models are already rapidly reshaping how we program computers and interact with them and that's going to influence the future, too.

**Luis Ceze**

The final thing to say, let's reflect on how much is happening right now. And it's all the confluence of all the progress in HPC, and computer systems in AI and, everything. And just really think about the future

of what we can do with computer is just amazing. And I don't think we are likely to live a moment like we're living right now in a long time.

**Sarah Webb**

To learn more about our guests, and for additional resources and reading on neuromorphic, quantum and molecular computing, please check out our show notes at [scienceinparallel.org](http://scienceinparallel.org). This concludes season three of Science in Parallel. Thank you for listening, and please subscribe on your favorite platform. We'd love to hear your feedback and suggestions. There are links to reach us by email and Twitter in our show notes.

**Sarah Webb**

Science in Parallel is produced to the Krell Institute and is a media project of the Department of Energy Computational Science Graduate Fellowship program. Any opinions expressed are those of the speaker and not those of their employers, the curl Institute for the U.S. Department of Energy Our music is by Steve O'Reilly. This episode was written and produced by Sarah Webb and edited by Tess Hanson.