MIT CSAIL Alliances | Neil Thompson Final Podcast

Welcome to MIT'S Computer Science and Artificial Intelligence Labs Alliance's podcast series. My name is Steve Lewis. I'm the Assistant Director of Global Strategic Alliances for CSAIL at MIT. In this podcast series I will interview principal researchers at CSAIL to discover what they're working on and how it will impact society.

Neil Thompson is the Director of the Future Tech Research Project at MIT's Computer Science and Artificial Intelligence Lab and a Principal Investigator at MIT's Initiative on the Digital Economy. Previously, he was an Assistant Professor of Innovation Strategy at the MIT Sloan School of Management, where he co-directed the Experimental Innovation Lab, or X-Lab, and a visiting professor at the Laboratory for Innovation Science at Harvard. He has a PhD in Business and Public Policy from Berkeley, where he also did a Masters degree in Computer Science and Statistics. He also has a Masters in Economics from the London School of Economics, and undergraduate degrees in Physics and International Development.

Prior to academia, he worked at organizations such as Lawrence Livermore National Laboratory, Bain and Company, the United Nations, the World Bank, and the Canadian Parliament. He has advised businesses and governments on the future of Moore's law and has been on National Academies panels on transformational technologies and scientific reliability. Neil, thanks very much for your time today.

Oh, it's my pleasure.

And in your introduction, I mentioned that you lecture about Moore's law. For those who aren't familiar, can you just tell us what Moore's law is all about?

Sure. So the short answer is that Moore's law is a description of just how fast computer hardware has been improving. Now if you go a little bit deeper than that though, it gets a little bit more disagreement about what people mean by Moore's law. So in 1965, Gordon Moore, who worked at Fairchild Semiconductor, was the first person to write this down.

And he said, look, I see this projection of us doubling the number of transistors that we can put on these chips very, very rapidly over time. And so he made this projection. And that turns out to have been basically true, that there was this very, very rapid increase in the number of transistors we could put on a computer chip. And because transistors are sort of like the fundamental unit that allow us to do computation, that means that our chips have gotten more and more powerful over time.

Now in common parlance, I think we actually use Moore's law even a little bit more broadly than that to include just sort of all of the improvement that has happened in computer hardware. Not just the number of transistors. And actually that makes a lot of sense. So at a very deep level, the reason that computers have gotten so good, so fast is that we've done a lot of miniaturizing of the components of them.

And turns out, every time you miniaturize these transistors-- again, these sort of fundamental units of computing-- every time you miniaturize them, you can fit more on a chip. That's just geometry. We can handle that. But the second thing that happens is they produce less heat. And that turns out to be really, really important. So I think we all have this experience of heaters in our house, where you have wires and you have electrons flowing through them. They produce a lot of heat. You produce more and more heat, you can imagine at some point it melts.

That's also the limit for our computers. If you keep running them faster, at some point they're going to melt. And that's not good. And so that puts a limit on it. But it turns out that as the wires get smaller, you need fewer and fewer electrons to go through there, you produce less heat, and that allows us to run our chips faster and faster. And so those are really the two things that people mean most often when they talk about Moore's law.

I see. So what do you mean by the end of Moore's law?

So in this sort of large trend about miniaturization and the potential that it had there-- so that was actually first articulated in 1959 by Richard Feynman, and he sort of had this incredible speech that he gave-- I'm sort of in awe of it-- where he gave his after dinner speech at the American Physical Society, and he said, look, I see a whole bunch of things in the world that are sort of macroscale objects-- like the stuff we sort of look at every day. I think we can miniaturize them a ton. And actually, this was basically the talk that introduced nanotechnology.

In that discussion, one of the things he said was, hey, we can take these computers that we're designing, and we can take these wires, and we can make them small. And by small he meant small numbers of atoms. This was a huge level of miniaturization from what we had had before.

And so this is a long way of answering your question of what is the end of Moore's law mean. It means that we are coming to this point that Feynman had identified, that once we get down to wires that are just a few atoms in diameter, all of a sudden we start having a whole bunch of stuff that gets more complicated. Because quantum effects start coming in and stuff like that. And so it gets much, much harder to build the chips at that point. And so all of these sort of things that were allowing us to make enormous amount of progress are slowing.

And what in practice that means is that this sort of incredible source of innovation that society has had, that has made all of our computers better, and that we've used to then make everything else better, actually that's really drying up. And so I like to think about this as society is riding a horse, and for a long, long time that horse was running incredibly fast, and now we're just kind of at a saunter. We're not making as much progress as we used to be making.

So what happens now? What happens at the end of Moore's law? Is it more about parallelism? To get more speed?

Yeah, so that's certainly one of the things we're going to do, right? And so some of our listeners may be familiar with GPUs-- these specialized chips that NVIDIA produces-- that were first used for gaming, but we're realizing actually we can use it for lots of other things. And one of the key characteristics about them is they do more things in parallel. So that's one particular example.

And of course people have probably heard of cloud computing and being able to run on lots of different machines on the cloud. So that is definitely one of the things that's going to happen. But we're going to have to look for lots of other sources as well. And so in one of the articles we wrote, we actually thought about this question, and we said, actually, we can sort of put this in an umbrella category of what we might call, things at the top.

So if you think about at the very bottom, we have these transistors that are getting smaller and making everything faster. Well, there's a whole bunch of things that sit on top of that and we can make them better as well, right? And so you can think about like actually designing the circuits of a chip better. And that's a little bit what we were just talking about with that NVIDIA example. And then sitting on top of that, you have the operating systems and the applications.

And in those top two things, you again can have this stuff where, like, maybe we've done some programming, and we've made a bunch of choices to make it really easy to program, even if it's not very efficient for our computer to run, right? And so we sort of made this choice that we're going to be less efficient in order to make things easier for ourselves. And I think what we're likely to do is take back some of that.

And you can call this something like performance engineering, and it's this idea of now we're going to look at this code really carefully, and find all the ways that we can make it more efficient, and then speed it up. And so I think this is a big thing that's going to be happening over this time.

I see. And where do you think it's more important to focus? On the bottom or on the top, as far as improvements are concerned?

Yeah, well, so I think both are obviously very good, right? I think the problem we're having is that these improvements at the bottom are becoming so much more difficult. So again, we're sort of reaching that end of miniaturization. If you actually look at the cost of building the factories that build new chips, those are going up exponentially. These are already like tens of billions of dollars. So you should be like-- searching for those improvements at the bottom is getting harder, and harder, and harder. And so I think what we are increasingly going to do is look to the top-- these other areas-- in order to get that kind of progress.

So switching gears a little bit. Let's talk about some economic trends. I found out that you went to the London School of Economics, and I did as well. Let's talk about GPTs and what sort of economic trends have we traditionally seen around them.

GPTs, or general purpose technologies, are actually one of the most important concepts in innovation economics. And what we mean by general purpose technologies is sort of two things. So I think the first thing is actually very intuitive for people, right? It's the technology that can be applied in lots of areas. And of course computers are a really fantastic example of that. We use them in all of these different parts of our lives and that's obviously very, very helpful.

But when economists talk about it they also mean a second thing. And the second thing they mean is that, when innovation happens in that technology, it spills over into all of the adjacent areas, right? And so as we get better algorithms for compressing data, Netflix gets better for us. Or as we improve what computer vision can do, our cars can keep on the road a bit better. So all of those things are examples of this innovation in one place spilling out into all of these other areas.

And we think that that's really, really important, and we think that computing in particular is an example of that, that has been doing this for decades and decades. And it has been very, very important to overall prosperity and overall innovation. But a really crucial thing that underlies this, and comes to this trends question you were asking, is, well, why is it that we've been able to have so many decades of progress in computing in a way that has been so useful in all of these different areas?

And the answer, I think, is that there is-- underlying this trend-- there's a self-reinforcing cycle. So that cycle is that, as computers improve, people buy more of them. That additional market size means that that finances the next round of improvement of the chips, and then we get new chips that are even better, and then we have more market, and so on. And we go around this cycle many times.

So that's really fantastic because it means that as the costs of doing this innovation goes up, as the cost of building these factories goes up, we can afford to actually do that because the market is getting bigger, because lots of people want computers. And that's pretty great. But that cycle really, really depends on the fact that we can continue to improve processor technology by investing more in it.

And what we've seen is actually that slowing down a lot. And so the real threat here is that this general purpose technology cycle that has been fueling so many advances throughout society is actually slowing down at the moment, and we might get less innovation and less benefit to society because of it.

So is there anything people should do to prepare their industry for these changes?

I think this is a particularly important question because I think we are at a moment right now where the role of CTOs in particular is changing a lot because of these changes. And in particular I think what's going to happen is that the role of the CTO, which has traditionally been focused on just these very high-level applications, right-- what database system do you want to use? How do you want to use it?

And then at the hardware level it might be like, how many servers do you want to buy? Or how often do you want to refresh your laptops? I think that's going to become a significantly more technically rigorous job in the sense that it's going to require understanding more details about the algorithms that are being used and the hardware that is going to underlie that.

So let me explain a little bit why I think that that's true. So we talked before about this idea of how rapidly computers have been improving, and what's really important to understand is that most of that improvement is coming at the very bottom. And so this is-- when I talk about bottom here, I'm thinking of what's sometimes called the computing stack-- and you can think about at the very bottom you have sort of the way that chips are built, and then on top of that you have the hardware level where people take those transistors at a very small level, turn them into circuits that do things. Then people have operating systems on top of that, and applications that sit on top of that, which are the things that we actually use.

So if you think about that-- all of those different levels-- everything we've talked about with Moore's law is improvement that's happening at the very bottom of that stack. And that has a really big impact because when you have so much improvement at the bottom, it flows to all of the areas that are above it. And in fact, all of the levels above it have taken advantage of it.

And so they've said, for example, in programming languages, people have designed programming languages like Python which are much easier to write, much more productive if you're a programmer, but actually less efficient from the point of view of the computer. And that's been OK because you've lost some efficiency, but actually you've gotten so much increased computing power from Moore's law-- from the bottom of the stack here-- that it was totally fine to do that because you were still improving overall.

As we lose that improvement at the bottom, we're going to have to start moving to finding improvement at the top, right? And so as Moore's law stops giving us that low level thing, we're going to look to other areas of the computing stack. And we're going to say, I want to make these more efficient so that I can harness my computing power to do more things. And so one very specific example of that is hardware specialization.

So I mentioned before, CTOs often are thinking about just like, well, how often do I need to replace the computer? And part of that is because the CPUs that were in our computers were very similar one laptop to another or one server to another. Not perfectly but quite similar, in the sense that they were designed to do a wide range of tasks pretty well and to be sort of agnostic to the task that you're going to apply to it.

I think where we're going is that, as we lose this improvement at the bottom, we're going to start designing specialized chips so that for our particular problem something can be even better. And so you might say, well, I'm working particularly on deep learning or machine learning, one of these areas. I'm going to design a chip that is particularly good for that but actually doesn't do other things.

And so CTOs are going to have to start thinking about those type of questions, and say, well, if there's a calculation that's particularly important for me, do I have to be designing my own chip or finding someone else who's designed their own chip that does something similar to what I care about? And so they're going to start having much more custom hardware as the fuel that is going into giving them improvement in computing.

Is quantum computing a viable solution to these problems?

So to a narrow set, probably. Well, let me say it-- probably is a little strong. For a narrow set of them I think that there's a reasonable chance that quantum will solve it, but certainly not for all of them. So I think that many people have in their minds-- because I think the way that quantum is often talked about, that it's like it's the successor technology, right? It's like we're in a relay race, and we've been in classical computing, and we're going to hand the baton to quantum computing, and it's going to run forward.

And I don't think that's the right analogy. I think the way we should think of it is like this kind of specialization we've just been talking about-- to say there's a lot of computing, and it's moving forward, and not that quickly anymore for most things. But there is going to be some little part of computing where quantum is really, really effective, and in that particular area we're actually going to get a lot of progress, assuming we can solve some of the technical hardware stuff about building these quantum computers.

Do you think that custom operating systems will sort of help in this particular problem, when you're talking about chip designs, right? Primary hardware-based solutions. What about software-based solutions?

Yeah, so I mean, I think what we're going to see is lots of specialization I would say all the way up and down the stack, right? And so I think a nice analogy here is the stuff that Google has done, where Google has-- they care a lot about machine learning, right? So every time you use Google to recognize your voice or something like that, they're using machine learning-- and in particular, deep learning-- as the way to do that.

And they've developed a full stack where they say, OK, we have TensorFlow-- which is this sort of framework for processing deep learning things-- but then they've designed all the way down from that very high level, all the way to the hardware to build new specialized hardware. And I think we're going to see that whole set of levels be optimized for particular things. And so we're going to go from-- interestingly-- this world where computing has been sort of like horizontal slices, right? It was like Intel did one thing, and then Microsoft did another thing, and these other things. And I think we're going to end up much more in silos where the kind of specialization that you're talking about and other kinds of specialization happen in these silos.

So when you think about your research, what are some of the takeaways you'd like our audience to know?

So in our discussion so far what we've really been talking about are sort of individual applications, or individual companies, and how they think about this. And what I'd really like to tell your listeners is, we need to be thinking about this as a whole society because computing has permeated so many parts of our lives. It's our work lives, our personal lives. And we can really see all of the innovation that has produced and all the benefit that it gives to us in our lives.

But if what we're seeing is that there's this slowdown in this progress, we should really worry about that, right? Like it means that some of these benefits that have been accruing to us over the last six decades are going to be accruing a little slower. Not everywhere, right? We will still see some folks, like if Google is still investing a lot in specialized chips, right? That's going to keep improving what Google does a little bit, but it's going to be much less broad based.

And so I think one of the things about computing over the last decades is that it's been sort of a rising tide that lifts all boats and we're really not going to be in that world anymore. It's going to be much more people investing in their particular things. And so the improvement is going to be much more uneven.

And you should really worry about that because there are going to be some areas of science, some areas of society that are basically going to get left behind as these things move too slowly. And I think that's something we're not thinking enough about as a society level, particularly in terms of what we might do to solve the problem.

Fascinating. Where could people go to find out more about your research?

Yeah, so I would recommend my website, which is just neil-t.com. And the other option is my Google Scholar page-- also has a lot of information on me and you can find me there.

Great. Well, Neil, thank you very much for your time today. It was a fascinating discussion.

Thanks, Steve.

[MUSIC PLAYING]

If you're interested in learning more about the CSAIL Alliances Program and the latest research at CSAIL, please visit our website at cap.csail.mit.edu, and listen to our podcast series on Spotify, Apple Music, or wherever you listen to your podcasts. Tune in next month for a brand new edition of the CSAIL Alliances Podcast and stay ahead of the curve.