Rashmi Mohan: This is ACM ByteCast, a podcast series from the Association for Computing Machinery, the world's largest educational and scientific computing society. We talk to researchers, practitioners, and innovators who are at the intersection of computing research and practice. They share their experiences, the lessons they've learned, and their own visions for the future of computing. I am your host, Rashmi Mohan. Having extensible skills in our industry is always a boom, but rare is the individual that straddles the world of hardware engineering and software programming with Ellen and expertise. Our next guest started out as a hardware engineer, pioneering significant areas of computer architecture, and then seamlessly went over to the world of computer programming to scale and conquer new heights.

Luiz Barroso is a Google fellow and the head of the Office of Cross-Google Engineering, which is responsible for company-wide technical coordination. He drove the transformation of the company's hyperscale computing infrastructure, and has also led engineering for key products like Google Maps. He's a published author, a prolific researcher, and most recently, the winner of the 2020 ACM-IEEE CS Eckert-Mauchly Award for leading the design and development of warehouse scale computing in the industry. Luiz, welcome to ACM ByteCast.

Luiz Barroso: Oh, thank you very much, Rashmi. I'm delighted to be here.

Rashmi Mohan: We're super excited to have you, and I'd love to start with a simple question that I ask all my guests, Luiz. If you could please introduce yourself and talk about what you currently do, and also give us some into you, what brought you into the field of computing?

Luiz Barroso: Sounds good. Yes, I am Brazilian by birth. I've been in the U.S. for over 30 years now, I came to the U.S. for my PhD, like so many of us immigrants in this field. What I currently do, I worked at Digital Equipment Corporation before I came to Google, and I've been at Google for just about 20 years now. As you mentioned, I'm currently leading the Office of Cross-Google Engineering, and this is a relatively new office in which my responsibilities are to coordinate the technical roadmaps that actually needs to be consistent across all of our products. So if you think about technology behind Gmail or Photos or Search or Assistant, there are bits and pieces of it that are common to all of our products. And those are the kinds of things I tend to focus on the building blocks that are all our developers internally at Google use.

And in particular, this year, I've been focusing on two key areas for us. One is of course, machine learning infrastructure, the other one being privacy and security. And I should tell you what drove me into this field of work, to the second part of your question. I always since I was eight years old, I guess, I knew that I wanted to become an engineer and actually an electrical engineer. My grandfather, who was a physician in the Navy, for some reason had a hobby in amateur radio in Brazil. If you had amateur radio back in the day, you have to be a little bit of an electronics hobbyist because those things are breaking all the time. So I spent some time with him. I was fascinated by what he was doing. Very early on, I decided that that was really cool and that's what I wanted to do. And from then on, it went from electrical engineering to then sort of computer engineering and then from computer science over the years throughout grad school and a couple of companies.

Rashmi Mohan: That's great. I mean, it's nice to have that inspiration at home and definitely somebody to sort of tinker with. Were there other role models, Luiz, that as you sort of got into the world of computing, as you studied computer science in college or electrical engineering, were there other role models that sort of influenced your interest, especially in the field of distributed computing?

Luiz Barroso: Yeah, I mean, of course, lots of role models across. I was lucky to be able to work on very early efforts in local area networks during my undergrad, and I found it fascinating, the idea that relatively inexpensively at the time, you could actually have multiple PCs sort of talking together and coordinating to resolve a task. There was a project in my university that was looking at some categories of local area networks using CSMA/CD, which is a collision based arbitration protocol for access, but we had interest in real time. So we wanted actually have limits so that when the network was very congested, we still could guarantee that some packets would get through by a given deadline. I found that fascinating. It went from there to a class in queuing theory. And I think queuing theory was the area that just got me infatuated with computing itself, as opposed to electrical engineering.

I had a wonderful master's advisor in Brazil, Professor Danielle [Menase 00:05:07], who probably is very much responsible for my going to this field because he was not only a great mentor, but a wonderful teacher. So he probably was the one who got me going this general area. I just found queuing theory magical. You could write these equations that you knew that were based on principles that real computing systems could absolutely not adhere to, and yet somehow they worked, they predicted what would happen, say, when a particular why they're in network was getting congested or not, and help to design systems that performed well by just using math. I just thought that was just mind blowing.

Rashmi Mohan: That certainly sounds like something that is, I mean, incredibly fascinating and to see the application of that into what might be a real world problem that the industry might be facing. What was that journey like, Luiz? So you were a student and you were working on these problems with your advisor, how did you then start to get it as like, how do I actually apply this into the real world? I mean, did you get a job straight out of college or did you stay in academia for some amount of time?

Luiz Barroso: Yeah. So during my masters in Brazil, Danielle actually got me an internship at the IBM Rio Research Center. And that was my first, I guess, computing job, so to speak. And it was a real privilege to actually have access to the kind of computing and not just computing, but just the IBM library. I could actually read papers that were otherwise very hard to get in Brazil. So it was a tremendous boost for my enthusiasm in computer architecture to be part of that group during my masters. When my masters ended, as opposed to the situation today where Brazil was blooming with entrepreneurial efforts in computer science and lots of great universities graduating, lots of great students and really interesting jobs in Brazil for computer scientists, back in the day, it wasn't quite as much. It was very early. And the idea of actually going for a PhD seemed to be the most fun thing to do because the jobs that were going to be available for me at that time in Brazil didn't seem quite as exciting.

So I applied for a PhD program in the U.S. and with the help of Danielle as well, I ended up going to get my PhD at USC in Los Angeles.

Rashmi Mohan: Got it. Yeah. And what did you do during your PhD? What were kind of some of the problems that you were solving then?

Luiz Barroso: Yeah. So during my PhD, I think that's when I really began to get interested in computer architecture. We had a strong computer architecture group at USC at the time. And my advisor, Michelle [Dubbo 00:07:43], was working on a lot of interesting problems in memory systems, in particular cache consistency models, cache coherent protocols. You have to remember, those are the times where we're beginning to think about shared memory multiprocessors are going to be the way to go. And for shared memory multiprocessors to shine, we needed to solve coherence and consistency problems. So much of my PhD was working both on the practice, as well as the research aspects of memory systems. My PhD thesis was on different kinds of snooping cache coherency protocols for non bus-based systems at the time, but I also was part of an NSF sponsored project to build a multiprocessor emulation platform, which was possibly one of the first large scale emulations using FPGAs, which is a technology that's very useful and very widespread today. It was quite exotic at the time.

So I had the chance at USC both to work on real system design, as well as to do some research, which was a treat.

Rashmi Mohan: Absolutely. I think that kind of the opportunity to do that sort of work, which is not just research based, but also the collaboration with the real world problem is definitely something that validates what you're doing as well as gives you real data to actually take your research forward.

Luiz Barroso: Yeah, I think you're right. I have a feeling we probably may talk a little bit more about this, that research was my way to find interesting problems to work in and computing, and yet my passion was always more in the very, very applied end of research and really engineering. So the opportunity to be able to actually put in practice as opposed to sort of only do the theoretical work was really, really important to me.

Rashmi Mohan: Yeah. No, that makes a lot of sense. And I would say that's probably one of the main reasons why we thought you would be an excellent guest on our podcast because that's what we try to focus on as well, like how do you bring that research into practice. Most of our audience are also practitioners who are saying, okay, there are these great problems that are being solved in the academic world, but here are some of the on the ground problems that I have and how can I apply that research to make my life easier. But from there, Luiz, I mean, obviously some of the most perfect work that you are famous for is the warehouse style computing. I was wondering if you could just explain that concept to our audience and how did you get into it?

Luiz Barroso: Sure. We're fast forwarding now to around maybe 2004, 2005 when the size of the minimum Google data center, if you will, was beginning to be much bigger than any third party colocation facility could provide us. So we had to build our own data centers. When we began doing that, we suddenly were a very vertically integrated company because we are now in the positioned, we had the right be designing our own servers. We're beginning to design our own networking switches. We were designing our own storage systems, our own sort of distributed file system appliances as well. So at that stage, I don't know if it was the first time in history, probably not, but certainly it was a unique time for us in which we were designing just about every piece of that thing of that data center, from the building shell to cooling infrastructure, to the power substations and power distribution, UPS systems, the emergency generators, and of course, the servers and the hardware and the software.

And the interesting thing about Google scale at the time is that products like Google Search or Gmail didn't run in one or two machines or in one or two racks of machines. These were such large scale deployments at the time that Gmail is a software that runs in a building and Search is a piece of software that runs in the building. The moment you kind of realize that, it becomes pretty obvious that that building is indeed the hardware in which your software, say, Search is running on. And when you begin to look at the design of that whole facility, from the perspective of a computer architect that that is your design, your design is not this machine that then you wired up with networking with other machines. The computer you're designing is that entire building. Opportunities for efficiency, both in terms of performance, cost, and then energy efficiency appear from everywhere, because it was an area that really didn't have that much work. We had no other known team that was taking such a holistic approach.

So that was a really exciting moment, because we really had blank sheets of paper to think through, a lot of engineers with very little expertise, but not short of confidence. It was a fun time.

Rashmi Mohan: Yeah, that's great. I actually listened to the six-part series that you put out on Google's data centers. I thought it was fascinating. That was definitely something I'd recommend to the others, but I think I heard you say in one of those interviews that you were thinking about being thrifty. So it was not that as a company, maybe you were early enough where you didn't have like a massive bank balance that you probably do now, and in terms of money is no cost is as of no consideration, but how did that play into some of the design choices that you made?

Luiz Barroso: That's a great question, Rashmi. And just to be clear, money is absolutely important today.

Rashmi Mohan: Okay. I take that back.

Luiz Barroso: It's a very competitive field. But you are absolutely right in that there is a Sergey likes to say that, I don't know if that's his or he's quoting somewhere else that scarcity breeds clarity. And certainly, it was the case for us, right? We just could not afford to build the kinds of data centers or servers or buy the kinds of networking gear that other people were buying. We just didn't have that kind of money for the scale we had. I joined Google during a time where we actually were not making a profit at all, right? So we really were forced to be very, very thrifty. And honestly, even before I joined Google, I think these people that were there before me were already pioneering this idea of thrift, we were already beginning to put it together our own motherboards for our servers using desktop class components, as opposed to server class components, which were much more expensive, because we just realized we couldn't afford the server class components. So we had to make the cheaper desktop class components work. That's just one of them.

We couldn't afford the really fancy storage appliances that you could buy at the time. So we decided to put just regular desktop class disc drives in every server and instead created distributed file system. That was later we published a paper about it called GFS that has evolved into something called Colossus at Google today, because we just could not afford to buy fancy storage appliances that existed at the time. And the same thing happened with networking. Networking today, I think, is at a price point that is probably much more reasonable than it was at the time, especially high performance data center class networking. Buying the kind of bandwidth we needed within the data center using the vendors at the time was just something we could not afford. So we decided to see if we could build distributed switching fabric that was built out of inexpensive switching components as well. So a lot of it was driven by the fact that we really didn't have another option. We didn't have the money to do anything else.

Rashmi Mohan: Got it. I appreciate that very much. So the question I have then, Luiz, is today, there are probably smaller startups or smaller companies in the position that you were in. Do you see that hunger? Do you see that innovation coming from those areas as well? And also, like you said, it's still a strong consideration, even at a place like Google. Do you feel like that drives you to constantly improve and build more efficiencies either from a cost perspective or as an energy perspective? What are the other key considerations that you keep in mind as you keep looking at this problem to look for more innovation?

Luiz Barroso: Yeah, this is very much still part of our DNA, Rashmi. Many of our teams brag about 2%, 3%, 4% performance improvements, right? All the time. We have many ways internally to actually recognize people that work on performance and efficiency. So this kind of work is still very much elevated at Google, and it's something that I think it's going to continue to be part of our success going forward. Even though we actually have a little bit more cache reserves these days than we had sort of back in the day, the headwinds coming from the end of Dennard scaling and all of that are very significant. So for us to continue to have viable high performance, very compute-intensive services that do more and more amazing things, it requires an obsessive focus on improving our efficiency at the harder and the software level just about every month, right?

And to a point that I think that every once in a while, we may even overcorrect, in that we sacrifice so much complexity at times to get the next half a percentage point, that at times we look at it and scratch ourselves in the head and say, maybe that's actually was too much because we may have sacrificed other things in order to get that set of an efficiency. So if anything, I'll say that the problem we have today is to make sure that we don't over-optimize things to a point that we create complexity that makes our systems actually difficult to evolve.

Rashmi Mohan: Got it. Yeah. No, that makes sense. Do you think, Luiz, that obviously the scale of any sort of compute that Google sees is not probably what if I looked at the average business out there is going to see, is this a problem for other companies as well, other than the handful that sees the kind of scale that you do? What should everybody else be thinking about? I guess that's my question.

Luiz Barroso: Yeah. So I think there are two classes of companies out there, right? There are the companies that were not born in, if you will, the digital era, so to speak, right? And those probably will continue to have the more of your typical enterprise or your bank or your grocery store or your drug store. Those probably are less likely to have the kinds of computing requirements that a company like Google has. They have different kinds of requirements that are very important for their businesses. But we increasingly see a very large number of what some people call digital first companies, companies like Snap, for example, or companies like Twitter. And there's a longer tail of companies of that caliber that begin to have rather significant needs for computing capacity.

And in that case, I think that over time, there's going to be a larger number of companies more than the top four or five large technology companies today that will actually have the same kinds of requirements and needs that we do internally as a company. And we're excited about that because there's a lot of technology we've developed internally at Google for Google's own news that we are seeing over time becoming more and more useful for our cloud customers. And it'll be really exciting to be able to continue to offer more and more of those. We already do this today with things like Spanner, for example. And so that, I think, is a really great opportunity in which these companies can take advantage of these kinds of problems, but they don't have to actually solve the entire sort of data center building end of efficiency problem from scratch. They can take advantage of a solid infrastructure provided by good cloud providers.

Rashmi Mohan: Absolutely. Yeah. No, I think and that helps them focus on the business problems that they're trying to solve without having to worry about the underlying the hardware or just managing of infrastructure. But one of the other questions I had, Luiz, is I was listening to, or trying to do more research on the kind of work that you do, was also, I mean, we've heard about Moore's law and how it may not be as valid today, or is probably on the decline, if you will, in terms of how valid it is for us today. What are some of the strategies to continue to build efficiencies? I know that in some cases, building efficiency in the software side is considered a way to extend a Moore's law, if you will. What are your thoughts around that?

Luiz Barroso: Yeah, this is really important. I mean, to be clear, Moore's law is ancient history as originally articulated. I happened to be about the same age as Moore's law. Moore's law was articulated pretty much when I was born. I was rather sad to see it go away, but I got over it because that happened like 15 years ago. The kinds of improvements we see coming from the circuit level today are nowhere near anything that I experienced for the majority of my professional life, which is scary and exciting because it's a new challenge for all of us. You mentioned one of the aspects of it, which is how can we now work on efficiencies at the various layers of the software to continue to squeeze continuing improvements in performance over time. And we love to do that at Google.

Like one of the things that a lot of our infrastructure teams do, say, teams that build database systems or data processing systems, for example, or say machine learning accelerators is to say, how can we recreate that magic from the '80s and '90s, where you could write a program on an x86 machine, and you could go to sleep for three years and you wake up and you ran that program again, without having to do anything to even maybe recompile the program, and that program will run at least twice as fast. That is something that the hardware itself alone, or if you will, the circuits themselves are not giving it to us. So we need to find other ways of doing that. I mentioned one of them improvements in the software stack.

The other ones are a rich area of exploration for us, which is a hardware acceleration for some specialized kinds of computing. We started the TPU program, which is a set of chips that Google built to run deep learning applications several years ago, which is one way that we have been able to achieve several orders of magnitude, but certainly 30, 40 acts improvements in overall performance. So energy efficiency for these kinds of applications for building a hardware that's very dedicated to that. And this acceleration, hardware acceleration trick, if you will, it's not a panacea, but there are still quite a few areas that we can still apply them. More recently, Google talked about a chip that accelerates encoding for video processing so that video conferencing systems like the one we're using today becomes much more efficient to run at the data center level because the processing of audio and video streams is optimized by special accelerators.

Finding the right things to accelerate is tricky because if you want to build hardware for something, it has to meet these two criteria. It has to really be specialized enough to something that has a big edge over general purpose computing, but it also needs to have a big enough market because otherwise, you don't want to build something that's going to also only require 10 chips, right? You want to build an accelerator for something that's going to be very broadly used. When you need the intersection of these two factors, then the number of things that can really benefit from hardware acceleration goes from potentially on the ideas level very large number to a more modest number, that I think still quite interesting and we're still exploring it. It is actually a high bar for big gains in hardware acceleration.

Rashmi Mohan: Yeah, no, that's an incredibly valuable point that you bring is really to identify what the areas where that acceleration is going to be valuable as well as needed in the future, right? In some ways, you also have to predict where this is going to go and start to sort of think of about those problems ahead of time. But going back to your introduction, Luiz, and you're one of those few people that actually navigates the world of data center and infrastructure optimization and also Google Maps and Earth. So what drew your interest? I mean, how did you sort of build skills in both these sort of somewhat diverse areas of computing?

Luiz Barroso: Oh, I'm generally driven by areas of ignorance. I don't know, I think some of us are just wired that way. In fact, when I came to Google, I was a micro architect. I was designing a chip at the Digital. Came to Google making a relatively rather late career change to learn how to be a programmer, and a programmer of all things for internet service is something that I was certifiably incompetent at. So I tend to kind of work that way. Every four to five years, if I begin to feel like I'm finally kind of really understanding what I'm doing, the world is so interesting, and the computing and engineering is so interesting, that I'm suddenly much more interested in poking an area of complete ignorance than continuing to build my expertise in an area that I have already been sort of working on for a while.

So that was a story with Maps. It was actually a double career change in the sense that I both went to an area of complete technical ignorance, but I also went from being an engineer individual country, but to managing a team, to becoming the VP of engineering in Geo, which is our internal name for the

Google Maps team. And boy, it was just a fascinating area, fascinating technically, and we can talk a little bit about that. Fascinating because of the sense of mission that the team had, the energy on the team. This is a group of people who really understood how important the tools and technologies they were building were to people all over the world. And you could see that in enthusiasm that they brought to work sort of every day.

Imagine the idea of being able to make it easy to find local shops so that the overall vitality of our small downtown sort of all over the world, if you will, is preserved in the internet age. You can still find your local shop, you can still find your local deal. That's one of the things that Google Maps as a product that it does, which I think is truly inspiring. And in the case of Google Earth, it's a team that is really interested in storytelling and getting people to more deeply understand how amazing our planet is, and what are the things we can do to keep it this way. So if you go to Google Earth, these days went from being a platform for GIS geo-nerds to being a platform for storytelling about the world, while still actually being a great platform for mapping nerds.

It was an incredible area to be in. From a technical standpoint, it was an amazing challenge because we wanted to map the entire world, and it's kind of expensive to do that, especially if you can't imagine sort of driving every street in the world every other day to see what's changing, it's kind of a big planet. And in particular, how do you do that in an economically viable way, even in areas where Google's business was not very strong, say parts of south America, where I come from, for example. So there was a challenge to make it to go back to thrift, to make it cheaper and cheaper and more efficient, to create a real time accurate representation of the world that helped you discover what is around you.

And we went into this journey of being probably one of the most aggressive adopters of machine learning technology to automate this process of understanding the world, whether it's from imagery we took from space, or from cars, or from contributions to Google map users and using ML to make that process maintainable, to keep the map up to date and increasing its coverage. And this mission of making sure that everybody in the world has a rich digital map experience that helps them navigate and discover things in the world anywhere they live or work was just something that really drove us. It was a fantastic time to be a part of that journey.

Rashmi Mohan: The share passion that you have for the product, as well as the mission, definitely comes through in your answer. And I have to say for me, personally, as well, and I'm sure that's for many users, Google Maps is one of my favorite products as well, simply because for the amount that you I used to travel just the feature of having an offline map to be able to navigate an unknown area is fascinating. And I have to say that even Google Earth, like you said, I think it opens up the world for you. I volunteer with an organization that works in education, especially in India, we use Google Earth there to basically introduce like the Taj Mahal to somebody in a remote region in India who may have never seen it, and literally brings history to life.

So I completely understand the larger purpose behind working on a project like that. What I do also want to ask you though, Luiz, is I mean, there's two parts to your answer that I want to sort of dig into a little deeper. One is when you make a switch like that in your career, that comes at a risk. I mean, you've built your credibility in a certain area for some amount of time and you're now going back to being completely a novice in a certain area, you're going to be the person asking the dumbest of questions, even if there isn't such a thing. How do you prepare yourself for something like that? And I mean, I would say most of us want to do that. We want to try a variety of rules, but there is a certain fear of like, am I giving up all that I have built over these a number of years?

Luiz Barroso: Yeah, you're right. There's no two ways about it, right? It is scary, and it is also not the only recipe for career progression at all, eight? I mean, we, both of us, probably know so many people who are the most successful people in our field who actually continue to work on one particular area of engineering or computer science, and they're unbelievably successful at that. But it's fun that there are different ways of succeeding in our craft. In my case, I don't know what is it about, Rashmi, but it probably an engineering education gives you a little bit of hubris that somehow you're going to figure out how to do things. And I think it requires a little bit of suspension of disbelief, I think it requires as you hint it to humility, and you overcome these things when the reasons why you switched were based on excitement and passion.

If you're doing career switches because you feel like it's going to be an important step in advancing your career, you use a given kind of sort of value systems and considerations for it. And people should absolutely think about those. When you make switches that are more based on, I'm really excited about this, this thing looks really interesting, then I think you are able to tolerate, if you will, the fact that you are going back in many ways and being the least informed person in the room for a while. The magic that happens sometimes though, not all the time, is that I've seen that when we hire new senior people, in particular, at Google sometimes. And there's a magic that happens when somebody that's clever and has experience that comes into a new area with the fresh pair of eyes that can see things that those of us who think of ourselves as experts in those areas may miss.

And every once in a while, I think I was able to contribute a little bit in some of these career switches by being that guy that's going, wait, wait, wait, but why are we doing it this way? And it just turns out that the people who had been living in that area, hadn't really thought about questioning that. And sometimes it's something that was based on a good reason three years ago, and that reason had a sell by date, and it's time to do something else. So that's a part of it that I think that is reassuring, is that, especially if you do that a little bit later in your career, it might not be a great thing to do in your first few years after graduation. But after you accumulate a little bit of experience, there's that extra ingredient that you bring to a new area, which is the fact that you are probably not dumb and ignorant about that area and you can contribute by bringing a different perspective.

Rashmi Mohan: Yeah, that's great. I think you just articulated the overall business case for diversity in your team, right? And we talk about, of course, gender and other forms of diversity, but really diversity of background as well is so critical to uncover some of these challenges that maybe the team that's currently been or very close to the problem hasn't seen. And I think the other thing that I really enjoyed in what you said is that we typically tend to think that the risk is lower earlier in your career to make these sort of switches, but from what you are saying, it sounds like there are certain traits that you develop over a period of time in your career that better set you up for making these switches and still contributing and adding value. So thank you. Thank you for sharing that.

Luiz Barroso: I think you're right, by the way. And when I talk to interns coming to Google, I do think that sometimes students and interns get so anxious about the decisions they make super early in their careers, right? I don't think they're actually that important. So I think that even at that level, even if you probably don't want to be doing a huge amount of job hopping earlier on, those early decisions are not that consequential in the long run.

Rashmi Mohan: Got it. Yeah. And that's very liberating, Luiz. So thank you. Thank you for saying that. The other part of your answer from prior questions that I wanted to tap into was in one of your interviews, you spoke about teamwork being very integral as a quality for today's engineers, right? I mean, always, we all understand the value of teamwork. But you also spoke about the interdisciplinary nature of your team when you were building a data center infrastructure. Leading to the success of that, I'd love to hear more about that, because as I was hearing you talk about the fact that you had engineers from various disciplines contributing towards building the solution, it just sounded like there was some magic happening there.

Luiz Barroso: Yeah. The two aspects to your question, first is teamwork itself, and the second one is looking at things across layers of the stack quo, even across cross engineering disciplines. Let me talk about the teamwork first. When I joined the field a while ago, I think that there was this vision that computer science and programming and research, or sometimes even hardware design was more like an individual sport, certainly programming, but maybe hardware design at that time wasn't quite that way anymore. For the things that we do at Google today, the complexity of what we're tackling and the scale of the things we're building are inconsistent with individual superstars just carrying the day. We love to have individual superstars everywhere, but they're superstars because of their contributions to a team, because things at Google don't get accomplished by Luiz or anybody else, they get accomplished by teams that are incredibly high performing teams.

So we begin to judge our people and ourselves more and more, not by their individual brilliance, and more so by when they are on the team is the team better? And when they are not on the team, does the team suffer? I think this is a key aspect that I think many disciplines go through. I think if you look at certainly particle physicists these days, these are huge teams, right? A lot of things in biology, these are huge teams because of the complexity and scale of the things that they're solving. And I think computing has gotten there. So the importance of knowing how to build high performing teams is much more important today than it was when I graduated. So that's one aspect of it. The other aspect of it is how fun it is when things that seem impossible to solve at one level of the stack begin possible if you actually have us to the level, say, above or below it?

I'll give you a couple of very quick examples. We're building service at the time, and the efficiencies of our power supplies were terrible. Like it was across the industry, like you feed a power supply that transforms AC to DC with 100 watts and you lose like 20 or 25 watts just on the power supply itself. That was a bit of a bummer. And we realized that there was one way to make power supplies more efficient without making them more expensive, which is to say, well, the power supplies that we buy at the time, because of the standards they were built on, had to provide power at various voltage levels, 5, 3.3, minus 3.3, 12 volts, right? All of these power rails, because these were building blocks that could be used in sort of any kind of computer.

Well, we knew exactly what computer we were building and we didn't need all of these power rails. And it turns out that if you just let's build a power supply that just gives us 12 volts, nothing else, you can actually simplify the design of the power supply and achieve efficiencies that weren't available elsewhere. But you can only do that if you're designing the motor board as well, because now you have to design a motor board where there's no 3.3 volts coming from the power supply, that if you need that, you need actually create that yourself. So by designing these two things at the same time, in one side, as a power electronics engineer, on the other side, as a hardware engineer, that suddenly because they're working together, they find a way to create efficiencies that wouldn't be possible otherwise. Does that make sense?

Rashmi Mohan: Absolutely. Yeah. No, it makes a lot of sense. But the follow-up question to that is not every organization has the wherewithal maybe to have all of those disciplines within the organizations. So do you see opportunities for this being like industry-wide collaborations?

Luiz Barroso: Yeah. I think that those can be certainly industry like collaborations. The open compute area is one that we collaborate with the industry in these areas, for example, and that's one way of doing. But even in companies that say are not dabbling with hardware design, the kind of story that I told you could actually happen in the software layers between, say, a database layer and an application layer of a given team in a completely software company, that if those folks suddenly begin to talk to each other, they might be able to find opportunities for optimization that otherwise would not have been available if they were just trying to think about solving their problems in isolation.

Rashmi Mohan: Yeah. No, you're absolutely right. I mean, and sometimes it just takes us taking a step back from just delivery on a specific roadmap to then think about a larger problem that we're trying to solve and come together as maybe a consortium of people working in different disciplines within the company, but to solve a more sort of maybe futuristic problem that'll actually bring all of these pieces together and look for ways in which we can collaborate.

Luiz Barroso: Yeah. The other example I'll give you that does go across companies is when we're trying to figure out how to make data centers more energy efficient. And we did that in several dimensions in the cooling systems and the power distribution systems. But in this case, we're trying to figure out how to make the actual chips, the actual computing part of the data center more energy efficient. And for that, we needed help from microprocessor design companies and from memory manufacturers. And what we began to do at the time is doing some studies at Google to understand what are the features we need from microprocessors, for example, to make the whole data center more energy efficient as a whole. And this went into something we called energy proportionality, which is something that we needed help from the microprocessor manufacturers to be able to understand and engage with.

So we wrote papers about it. We had great relationships with the manufacturers at the time, we still do. And across these two companies, we were actually able to make significant advances in energy proportionality, which is one way of achieving energy efficiency in the data center.

Rashmi Mohan: Excellent. Thank you. That's a great example of that collaboration that happens across those boundaries of a company. Luiz, I'd love to know from you for our final bite, what is it that you are most excited about in the field of distributed computing or in the Cross-Google Engineering organization that you are current a part of? What are you most looking forward to in the next few years?

Luiz Barroso: I'll give you two things related to the two things I'm spending the majority of my time today. I think we're in the early days of taking advantage of what machine learning can do for society. And we are super excited about making machine learning more efficient, more easy to use, more applicable to different domains, eventually to a point where somebody who has no training in programming whatsoever can build an intelligent system based on machine learning, which I think it would be an amazing way to democratize the technology. I'm very excited about that entire kind of area. The other area that I'm fascinated by that I think is really meaningful for us as society is the issue of online safety, right? I think that all of us live a lot of our lives online and many, if not all of us, are a little bit nervous about this. It seems to be a scary world for us. For us, and especially for those of us who have children, right? And this is something that's front of mind, right?

So one of the things that we are dedicated at doing at Google is to make sure that the online experiences we have with any of our products are the safest places you can be on the internet. And this is a combination of amazing advances in security, advances in privacy, and advances in understanding the quality of content and in particular misinformation that are, to me, a grand challenge for our era in which companies like Google and computing as a whole has a very significant role to play. And I am very much motivated by the work that we are doing at Google to make sure that the safest place you could possibly be online for yourself and for your family will continue to be Google for years and decades to come.

Rashmi Mohan: That's very inspiring, Luiz, especially because I think given in the last year and a half during the pandemic, the number of people who have suddenly grown to be online has obviously exploded, and I think the average age of somebody getting on the internet has also come down significantly because that's been our primary way of sort of communicating with each other, of learning, of doing business. So yeah, I'm super excited. I think we're all looking forward to hear more about all the amazing things that you do. Thank you so much for talking to us at ACM ByteCast.

Luiz Barroso: Oh, it's a real pleasure, Rashmi. Thank you.

Rashmi Mohan: ACM ByteCast is a production of the Association for Computing Machinery's Practitioners Board. To learn more about ACM and its activities, visit acm.org. For more information about this and other episodes, please visit our website at learning.acm.org/bytecast, that's learning.acm.org/B-Y-T-E-C-A-S-T.