Episode 20: Acceleration at the Edge
Host: Nicole Huesman, Intel
Guests: John Melonakos, ArrayFire; James Reinders, Intel

Nicole Huesman: Welcome to *Code Together*, an interview series exploring the possibilities of cross-architecture development with those who live it. I'm your host, Nicole Huesman.

The industry is increasingly embracing open standards like modern C++ and SYCL to address the need for programming portability and performance across heterogeneous architectures. This is especially important with the explosion of data-centric workloads in areas like HPC, AI, and machine learning.

John Melonakos joins us today. John is the CEO and co-founder of ArrayFire, a software development and consulting company with a passion for AI and GPU acceleration projects and expertise in machine learning and computer vision. Thanks so much for being here, John!

John Melonakos: Thank you for having me. I look forward to the conversation.

Nicole Huesman: And James Reinders also joins us. As an engineer at Intel, James brings deep expertise and parallel computing having authored over 10 technical books on the topic, including a recent book about Data Parallel C++, or DPC++. He focuses on promoting parallel programming in a heterogeneous world. Great to have you with us, James!

James Reinders: Happy to be here.

Nicole Huesman: This episode was really inspired when I saw a recent blog that John authored. It really caught my attention. While ArrayFire has expertise in CUDA and OpenCL, in this blog, John writes about the open SYCL standard and its implementations. John, I understand you're a champion of things that are open, programmable and maximize performance. Can you tell us a little bit more about yourself and ArrayFire?

John Melonakos: Yeah, sure. So ArrayFire, we're 14 years old. A bunch of us were doing our PhDs together at Georgia Tech and we were doing medical image processing, which is one of the fields that has lots and lots of data. And it was taking a long time to graduate because we'd have to sit there and run our algorithms and they'd take multiple hours to run. So, while the simulations were running, we decided, why not speed these things up? And this was before really anybody was doing much of heterogeneous computing and you could use OpenGL shaders to do math on GPUs. Well, my co-founder Gallagher Pryor, he was just twiddling away and figured out how to run MATLAB code through OpenGL shaders. And it was off to the races from there. From there, we were able to build an accelerator from MATLAB and that led into a library of acceleration for other languages that's now open source. And we're passionate, at the end of the day, about helping people do real projects in science, in engineering, in finance. When you can do things faster, that means you can run more simulations per hour, which means you can refine and iterate and get a better product at the end of the day. And we've seen an explosion in that as parallelism has taken off.

James Reinders: So John, you do consulting on parallelism. What sort of challenges are you being asked the most to address or finding yourself to need to address?

John Melonakos: It's changed over the years, but recently we've had a lot of projects at the edge. So things that people are looking to put in very small end products, but doing pretty heavy computation at the camera, essentially. In a lot of cases, these are computer vision problems, and it's really exciting to see all the sorts of things that are becoming smart in our daily lives. A lot of times, we don't even know that these things are becoming smart and it's really fun to work through so many different projects in defense, in healthcare, in energy, and even in entertainment.

James Reinders: Are the computers you're using helping speed up on the edge, the ones that are in our cameras, are they more powerful than the ones you were working on with your PhD?

John Melonakos: Exactly! Isn't that wild?! That's how fast the explosion in performance is going. I actually tweeted this a couple of days ago. The rate of innovation and discovery in this world is astounding. I don't even think that humanity understands. We do not understand how fast this stuff is coming along, faster than we can actually sort of realize it in our own heads.

James Reinders: I couldn't agree more. So the ArrayFire library and the techniques you've developed previously, you really focus on performance. Is there a theme to what you do that gets access to that performance? Anything you could share?

John Melonakos: So, we're sort of middlemen. We work with partners like Intel who develops the chips that do the computation and we try to build a software that's closer to the programmer, that's what they can use those architectures in order to be so fast are incredibly complicated and it's not the best use of everyone's in the world's time to go learn all those complicated details. It's better for them to stay in their domains and be experts in their domains. So, we try to bridge as much of that complication out of the picture for them while retaining the ability to get really, really, really the full performance that's available from the different hardware. And ultimately that's what we're passionate about, and you know, I think we've done a pretty good job at. Our guys are able to go hand tune very common math functions for devices across architectures, across generations and across competing APIs, and then present it to the end user in sort of a math notation that's familiar to them from the pure math perspective and they don't really have to know the details of the architecture.

James Reinders: Well, you've sort of brought us back to MATLAB, or I feel that way. You mentioned MATLAB before. Thinking about speeding it up, I understand some of the work that you've done is actually in MATLAB these days. Is that correct?

John Melonakos: That's right. That's right. We started in 2007 with this objective to accelerate MATLAB. We built a product called Jacket. Our first sell was in January of 2009 and that product took off. It was a great product. It served all sorts of MATLAB projects, including helping me and my co-founders get out of our PhDs, like I mentioned. Ultimately, we partnered with the MathWorks and licensed it to them. And so now if you want to find that, it's in the parallel computing toolbox that you can buy from the MathWorks. We took that same know-how and retained the rights to the engine of computation for other languages. So, we no longer participate in MATLAB, but you can do the same sorts of things, the same math in science, in other languages. And we don't sell this library anymore. It's free and open source. We shifted our business model to doing services. So, people that pick it up, recognize our expertise, recognize the performance of the things that we do, but don't want to do it themselves, they hire us.

James Reinders: Well, that's super exciting. I worked with a startup a few years ago with some data scientists, and the first thing they wanted is to run the MATLAB to do their work and it's people like you that helped make MATLAB fast, you know, the plumbing underneath. So, it's exciting. Hey, IWOCL and SYCLcon are coming up in the end of April. Any thoughts about that? Are you going to participate? You got some thoughts on SYCL and OpenCL?

John Melonakos: Yes. I think this conference, I don't know what year it is, but it's approaching 10 years old. I was at the first one when OpenCL was first coming up. We've been champions of open standards from the beginning. We spent a lot of money building an OpenCL back into ArrayFire so that our customers could run on any OpenCL-compatible device. Even when it wasn't, you know, it's much more popular these days than it was when it first came out, just because we really feel that this longevity of code and the ability to build things that can evolve with the pace of innovation that we just talked about. You know, what's great today is not going to be the same as next year. And that for a business that's building products is hard to grapple with often. So we've been keeping a pace of those innovations so that people that build with ArrayFire don't have to change their code, they just upgrade the library. And so then these conferences I'm very, very excited about. SYCL is more new. That's going to be exciting to sort of see that it brings a lot of user-friendliness that was sort of missing an OpenCL, I think. It makes it much, much easier. And for that reason, a broader consortium of folks that are building with that as a core component of their projects. So I'm excited to see a lot more community-driven approach to heterogeneous computing in the coming years. And these conferences are well positioned to be the conferences for that kind of, you know, understanding and community building.

James Reinders: Yeah, for people like you and I that really care about getting our hands dirty, playing with the hardware and getting the most out of it, I think I'm looking forward to the conversations. I love the maturity, too, watching OpenCL grow and SYCL come from that experience. I think it's all taking dead aim at C++. And you know, we've got a lot to learn and experience there yet, but do you have any thoughts about parallel programming for C++ developers in the C++ language?

John Melonakos: I think you're right. I hadn't actually really thought much about that 'cause, you know, as a small company, we don't think much about influencing the C++ standards, but you know, Intel's much bigger. And I think that, at the end of the day, C++ is a language upon which the world relies, and it needs to be able to more wholly grapple with the new world of parallelism and directly in the language. And that's absolutely correct. And it needs these building blocks efforts, OpenCL and SYCL, to make that happen in a very standard way for the whole world.

James Reinders: Well, Intel may be bigger than ArrayFire, I'll take your word for that. But I think the thing that matters the most to me is experience with it. And that means everybody. And I would think nobody more than the sort of development you do in C++, that experience is valuable to instruct the standard, not the size of the company.

John Melonakos: Exactly. Sure.

James Reinders: So, are there any other open standards or industry projects that you follow that help underscore this interest in this movement to help heterogeneous computing?

John Melonakos: Of course, we follow the AI frameworks, the machine learning frameworks quite closely. We follow Python very closely. Python today is leaps and bounds more used than it was 10 years ago. And that's very interesting to us. I think those things are the building blocks for the next 20 years of innovation and making sure that we have the best tools in hand. That's my whole thought process for the last 14 years has been purely, how do I get good tools into the hands of domain experts that are doing the real innovation? And I think we're getting pretty close to a really nice set of tools for people to go change the world.

James Reinders: I agree. It's a very diverse set of tools, too. You know, I grew up around Detroit and mechanics will have mechanics toolboxes full of different types of wrenches and screwdrivers and what not, and it never confused me. I'll grab a Phillips screwdriver when I need it and I'll grab a wrench when I need it. And I get a feeling you're the same type. It's like, you'll work on whatever tools necessary, right. And different people can pick what tool works for them. Great.

Hey, I also know that you're doing some certified oneAPI training or you're offering that as part of your consultancy. Any thoughts, or how should people look you up if they're interested in learning more about that?

John Melonakos: We've been doing training for heterogeneous computing since Day One. And we really love teaching. It's something that we're passionate about and there's some great feeling of being able to walk into a classroom and inspire the people that are going to be building the next big thing. So we have a training page on our website, ArrayFire.com. We do virtual training, and as we come out of Coronavirus … we haven't traveled to visit anyone and do a training … but we typically do that where we will go onsite, or pick a customer, and train 20 people or something.

Nicole Huesman: Well, I think that's about all the time we have today. John, I know that you're going to continue the blog series on SYCL, SYCL implementations.

John Melonakos: Yes.

Nicole Huesman: What do you foresee writing about next?

John Melonakos: Well, I don't know what the exact next blog post will be, but I'll tell you the trajectory. Ultimately, we want to show some exciting projects using those standards, getting real work done. And as new hardware comes along and as these things mature, we'll get more and more into showing those things.

And then also, you know, at the same time, we're going to be building oneAPI back into ArrayFire. So that means that existing ArrayFire customers won't even have to change their code at all, they just will upgrade to the ArrayFire version that is oneAPI-compatible and they'll be able to get all the performance from oneAPI devices. So yeah, we're excited to sort of, as the devices come along and as the community builds in each domain, we'll show little examples of how to get it done and try to inspire folks to go build some really cool things.

Episode 20: Acceleration at the Edge
Host: Nicole Huesman, Intel
Guests: John Melonakos, ArrayFire; James Reinders, Intel

Nicole Huesman: Well, that's really, really exciting. And we're excited for the upcoming IWOCL and SYCLcon conference. So, thank you both for being here. Fantastic discussion! John, as we wrap up today, where can listeners go to learn more?

John Melonakos: Well, ArrayFire.com. We have a blog there: ArrayFire.com/blog. And also, we're pretty active on Twitter. We can connect with you there.

Nicole Huesman: Excellent. Thank you. And James, any pointers for folks? Any places where they can go to learn more?

James Reinders: Well, absolutely. I'd be remiss if I didn't say there's huge amount of great content on Intel.com, but I'd encourage people to follow me on Twitter or LinkedIn for specific posts. This topic is near and dear to my heart and my handle is easy enough. It's just @jamesreinders, all lower case run together.

Nicole Huesman: And I would probably be remiss in not pointing out again that there is a new DPC++ book out there that folks can also take a look at.

James Reinders: I've heard about it!

Nicole Huesman: I'll bet you have! Absolutely. Well, thank you both for being here. John. It's so exciting to hear about all the work that you have going on at ArrayFire. Thanks so much for joining us.

John Melonakos: Thank you.

Nicole Huesman: And James, thanks so much for sharing your insights. You have such deep expertise in parallel computing. Thanks for being here.

James Reinders: Well, it was a great conversation. Thank you both.

Nicole Huesman: And a big thank you to all of our listeners for joining us today. Let's continue the conversation at oneapi.com.