

## Code Together Podcast Episode #36

**Title:** Preparing Applications for Aurora in the Exascale Era

**Revision date:** 10/06/2022

Radhika Sarin ([00:05](#)):

Welcome to Code Together, a discussion series exploring the possibilities of cross architecture development with those who live it. I'm your host, Radhika Sarin.

Intel, HPE, and Argonne National Lab are collaborating closely to deliver Aurora. Built on Intel's architecture, Aurora will be Argonne's first exascale, high performance computing system. Its performance is expected to exceed 2 exaflops, which is 2 billion billion calculations per second. It's enormous scale and performance levels will accelerate research in science and engineering fields at a scale that is not possible today.

In today's podcast, let's talk to our guests about what goes behind the scenes to prepare for computing in the exascale era. Our first guest today is Scott Parker from Argonne. Scott is the lead for the performance engineering team at the ALCF. Over the last several years, he has been working with Intel on the development of Aurora and with the department of energy HPC applications community to prepare their codes for exascale. Welcome Scott, it's a pleasure to have you here.

Scott Parker ([01:32](#)):

Thank you Radhika.

Radhika Sarin ([01:34](#)):

Our next guest is Michael D'mello from Intel. Michael is the manager for the Intel Center of Excellence at Argonne National Lab. Over the last six years, he has been facilitating efforts with the DOE, HPC community to prepare applications for exascale. Welcome Michael, it's a pleasure to have you come speak about this topic.

Michael D'mello ([01:59](#)):

Thanks Radhika.

Radhika Sarin ([02:00](#)):

All right, great. Let's get started. Scott, tell us in detail about Aurora.

Scott Parker ([02:06](#)):

Sure. Happy to. So Aurora is an exascale supercomputer that will be cited at Argonne, and that we have been developing along with our partners at Intel and HPE. So the machine is planned to come up this year and when it's on the floor and operational, it will be capable of delivering a peak performance of over two exaflops, which will make it one of the fastest computers in the world, if not the fastest at the time that it's on the floor. Unlike many supercomputers these days, Aurora will use more than one type of processor for its computation. So in order to be capable of the kind of computations that we want to perform, all these nodes will have to work together and that will be achieved by using the new Cray Slingshot network technology, which will enable fast communications between the nodes and on the nodes we will be using both an Intel CPU, so it's a traditional sort of processor that's been around and

used on computers in your laptops or desktops just at a faster scale. And also the new Intel GPUs that they're in the process of developing that will enable us to accelerate the compute work that's been done on the nodes significantly.

Radhika Sarin ([03:15](#)):

Hey Michael, can you tell us a little bit about the technologies that Intel is using for Aurora?

Michael D'mello ([03:21](#)):

Sure. So Scott just covered the hardware aspects of this machine and I will talk a little bit about the software aspects. So the DOE community has a wide spectrum of scientific codes in various languages and utilizing various libraries and various other elements of software which brings special requirements for this kind of environment. Intel is meeting those requirements using our very recently developed oneAPI suite of software and tools. These include compilers, optimized libraries and performance tools of various kinds to HPC community be productive in this environment. So because of the scale involved here, this software has to provide users with not only the capacity to build in one code on this machine, but it has to be able to do so in scale. So just to give you an idea of what scale means here, Aurora has on the order of 60,000 plus Intel GPUs in addition to that, there are 20,000 plus traditional CPU type cores on this machine. There's also a specialized memory technologies like HVM et cetera. So it's a pretty wide spectrum of things that people have to contend with and leverage so they can do world class groundbreaking science. And with that of course comes the requirements I just mentioned. So essentially on the hardware side, we have the base that Scott spoke about. And on the software side, we have this oneAPI suite of tools.

Scott Parker ([05:06](#)):

And Mike, just to add a little bit to what you were saying. So as far as I understand, Argonne will be the first large scale site deploying the new Intel Ponte Vecchio GPU that's coming out the new line of XE GPU that Intel's been developing. So we're pretty excited about that to be on the leading edge of Intel's new technologies.

Michael D'mello ([05:25](#)):

So Scott, perhaps our audience would be interested in knowing more about the DOE side of things. And so could you give us a little breakdown of what types of research will be possible with Aurora exascale that were not previously possible?

Scott Parker ([05:39](#)):

Yeah, sure. So the systems we deploy at Argonne, all the computing systems that we've deployed in the past are national resources that we make available free of charge to scientific community around the world. So any researcher in the world can submit a proposal for time on Aurora. It is a competitive process so the proposals go through a multi-stage review from which only the most impactful ones are selected and then granted time on the machine. But it is used by a really diverse set of researchers around the country and around the world. And really the science that's done and will be done on Aurora is driven by the priorities of the scientific community and this spans a huge range of areas. We anticipate that we will have groups looking at things like astrophysics and trying to understand things such as dark matter and the formation of galaxies.

Scott Parker ([06:26](#)):

There will be other teams that will be looking to simulate Earth's climate and understand the impact and mitigations that may be possible for global warming. Other groups are looking at new energy technologies such as fusion reactors so that's the kind of nuclear reaction that happens in the sun. It's different from the kind of nuclear reactors we have here today which are fission, and this would let us unlock a new source of energy from basically seawater or some of the components that are in seawater. And that will be much safer than the nuclear energy that we have today. Beyond that we have teams that are looking at doing fluid dynamic simulations that allows them to understand the airflow over things like aircraft and design better more fuel efficient aircraft. They can also couple that with combustion simulations that allow them to design more efficient engines and then going even sort of down into a smaller scale, there's a lot of teams looking at drug discovery and disease.

Scott Parker ([07:23](#)):

So simulating how proteins and molecules interact in the body allows incredible amount of opportunities to understand both sort of fundamental nature of how disease occurs and also the possibilities for new drugs to cure those diseases. And beyond that, we have a large amount of work going into understanding material science and chemistry and simulating things in those areas in order to build better solar cells or higher capacity batteries for electric vehicles. So there's just an incredibly broad scope of research that happens on these platforms and beyond just the research itself, the tools that are becoming available to the research community are changing. So the new techniques in artificial intelligence are available and can be accelerated on Aurora to let the scientific research communities apply these new AI tools to the research that they've been traditionally doing and further their discovery in these areas.

Michael D'mello ([08:15](#)):

So Scott, you've been involved in a lot of behind the scenes work to make Aurora ready for science as soon as it's deployed, can you give us a feeling of how you prepare for a new computer before it hits the floor at ALCF?

Scott Parker ([08:28](#)):

Sure. And this is something you yourself have been involved in a lot. It's really large effort and it's been a multiyear effort. I mean, we've been working with you and other folks at Intel and with our applications community for several years now to get ready for Aurora. So for Aurora, Intel's been developing both new hardware and new software and the people here at Argonne have been working very closely with them on that. Intel's obviously a leader in developing processor hardware so our role has been more to provide feedback to Intel on how well their designs suit the needs of our application community as they go through the process of designing and developing the new chips that are going into Aurora. Similarly, we provide feedback on the software that they're developing such as the compilers, programming models, and libraries that we'll be using on machine.

Scott Parker ([09:17](#)):

In the software space we can play a bit more of an active role. So we've been doing a lot of testing. So we have the things that we want to run on Aurora when we get it and in advance of that, we've been trying to run those things on the early testbeds that we have with the sort of pre-release software that Intel's made available to us and then any issues or problems that we encounter or changes we'd like to see, we report to and work with Intel to see those implemented. And so a key component of creating a robust software environment on Aurora is really doing a lot of this sort of testing. New software always

has issues. It's inevitable, it's unavoidable, and the way you deal with that is you try to find them and then correct them. And so that's really a lot of focus of our work over the last couple years has been in that sort of testing. And in addition to working with Intel, we're also working closely with our applications community to help them get ready for Aurora.

Michael D'mello ([10:13](#)):

That's remarkable. That's a very nice description of how you have prepared for it. Can you give us a little more of an idea of what the scale of this effort is in terms of projects, numbers of apps, that sort of thing?

Scott Parker ([10:25](#)):

Yeah, and it's quite a big effort, both on the Argonne side, across the DOE department of energy complex and with Intel. So in addition to the work that DOE's funding to put Aurora on the floor, DOE has had a large effort called the Exascale Computing Project that it's been funding over the last four, five years. And that's providing resources to dozens of research teams across the country to help get their software and applications ready for the upcoming exascale machines of which Aurora is one of them. As part of that effort, there are 20 application projects within the ECP that we are working with to help them get ready. And in addition to that, Argonne has identified another 20 applications to work with as part of what we call our early science program, which is something we've traditionally done before we get any new machine is to help them get their codes ready for the machine that's coming, in this case it's Aurora.

Scott Parker ([11:19](#)):

So overall we're working with more than 40 project teams to help them get their codes ready for the system when it arrives. And in many cases, the teams have more than one application. In some cases, people that are for instance, simulating a nuclear reactor, have a code that simulates the fluid flow through the reactor core and at separate code that simulates the neutrons and the neutronics that are happening in the core due to the nuclear reaction. So there's multiple facets to the work that they're doing that are covered by really two distinct sets of applications in that case. And so overall we're working with over 50 different applications to get them ready for Aurora. And this involves work from more than 60 people here at Argonne and at Intel who are specifically focused on helping those teams get their applications ready. And then I would say it's easily two to three times more people than that in the broader application community that are at other universities labs that we're collaborating with to make this all happen. So it's several hundred people all so it's quite a big effort.

And so Mike, as I said, we've been working closely with Intel, can you tell us a bit more about from your side Intel's collaboration with Argonne?

Michael D'mello ([12:27](#)):

Sure, sure. So as you all know, it's a rather multifaceted collaboration. It's big, there are lots of people involved, lots of engineering meetings all the way from Intel development to support engineers, et cetera but one of the things that we are doing that I think is quite impactful when it comes to interacting with the user community, the scientists involved and all this, the training events we hold. So we break them down into three categories, workshops, dungeons, and hackathons. And of course, in addition to that, we do other outreach activities like webinars et cetera which I'll touch on later but the core training events that we handle are workshops, dungeons, and hackathons. These are randomly similar things but they differ in their focus. So a hackathon would be essentially a two day event where we'd sit

down with a group of scientists focused on one code typically, and we'll do a very deep analysis of it with the authors of the code, the developers of the code and the various experts from Argonne, the DOE and Intel involved to analyze the code and understand how efficiently it is using the proxies that we have in place for the actual Aurora machine.

Michael D'mello ([13:43](#)):

So we know it's going to have a certain kind of GPU and a certain kind of CPU so we have smaller setups that we can test that hardware on with these codes. So that's what a hackathon does. It actually takes a certain target code and examines it and optimizes it, we try to optimize it depending on what we find on the GPUs and CPUs that we have available to us. A dungeon is a slightly bigger event. You can think of it as multiple hackathons at the same time. They have multiple tracks so it's got a order of reach, you'll have four or five project teams involved. And you can actually have projects collaborating with each other. In other words, if someone discovers something, a technique to make a certain kind of algorithm run well in one project, they can share it with another project. And that's what a dungeon does. It brings together not only the experts and the code developers and all these people, but it also brings together the projects to some degree. And a workshop of course is much bigger thing and it's something on the order of several hundred people potentially from the DOE, from Intel, from Argonne, from other affiliated organizations.

Michael D'mello ([14:50](#)):

And the objective of a workshop is to share the learnings that have come out of hackathons and dungeons and other activities, other engineering activities leading up to the launch of this machine. So it's a nice little set of events that helps drive progress towards obtaining a set of scientific apps that are very well optimized for this future technology. And that we think will actually enable these scientists to do their work eventually at the kinds of scale we talked about a few minutes ago. Now there are certainly going to be portions of the community that are not able to come for these events for various reasons, geographical or other reasons. And for those folks, we have webinars where we isolate a certain topic and have an expert on that topic present to the community at large and we make these recordings available to anybody who is interested so they're freely available and that's another mechanism we use to communicate to the community, and to people who might potentially join the community to contribute towards a technology like Aurora. To date I would say we've done about 18 or so hackathons, we got about four dungeons behind us and about the same number of workshops, there is a whole number of webinars that we've done that I've pretty much lost count of so I hope that explains the kinds of collaborative efforts that we have ongoing with Argonne, Scott.

Scott Parker ([16:18](#)):

Yeah, I know there's been a lot going on over the last several years. It's been a great opportunity especially when we were able to be in the same room and sit down and talk to people face to face, but certainly, doing things remote has also worked really well over the last couple years.

Michael D'mello ([16:35](#)):

So I'm curious Scott, from your perspective, how do you see these projects approach exascale code development? So exascale is something that has never created before, it's something completely new, so what's your take on how these projects approach exascale code development or potentially how they should approach it?

Scott Parker ([16:54](#)):

That's a good question. I mean, every project is unique so I don't know that there's a sort of single approach that everybody takes. There's a lot of trial and error in this. So sort of two steps forward, one step back that we see, because it really is a challenge. People are really trying to do something that they haven't done before and each generation of computers that we've deployed is typically an order of magnitude, increase in capabilities of performance and so you're really taking a big step there. So there's a lot of experimentation to try to find the best path forward for each application. But that said, I would say, in very general terms, each code team first off needs to implement the algorithms in their code that represents the calculations that they want to perform.

Scott Parker ([17:38](#)):

So most of these teams are working with existing codes, but they may be revamping them considerably. And especially in light of the new capabilities that Aurora offers there's a lot of opportunities for new science and in many cases, the teams need to modify their codes in order to enable them to take advantage of these new possibilities. So the first step is writing the code that enables you to do your science in whatever form, even if it's just to run on a CPU today on a machine that you have access to, that gets you sort of started and gets you to a point where you have a simulation that you can perform that is capable of doing what you want. But when you talk about then how do you get that onto Aurora specifically? The real key to that is making sure that your code is written in a language or a programming model that is supported on the machine.

Scott Parker ([18:25](#)):

So on Aurora, the languages that we support of Fortran, C, and C++ and also Python, and then within those languages, you still have to code in a programming model that lets you use the various components of the CPU and the GPU in parallel and efficiently. So on Aurora, the programming models that are available to people would be SYCL, OpenMP, Kokkos, and RAJA. And one of the challenges that we see today is not all machines support the same programming models. So if you run on a different platform, you may have written your code sort of in a different way that's not going to run out of the box on Aurora. And so for people that are interested in Aurora, what they have to do is look at the options that are available on our machine. And in some cases, they do overlap with what's available on other platforms.

Scott Parker ([19:12](#)):

In some cases they're unique or different or novel. And so developing a strategy for what programming models and languages they want to use both on our machine and on other machines going forward and then if they aren't already written in one of those languages, reimplement the key components of their code in one of those models. So that would be what we call porting to Aurora so taking their code and rewriting it if needed so that it can run on the Aurora platform. And so once they have a code that's written in a way that can run on Aurora, the next step is generally testing. Like I said, this is new technology, new software, new hardware. There's always a lot of issues that occur in the development process and we want to identify as many of those issues as we can and knock them out before the broader research community gets on the machine, that's testing.

Scott Parker ([20:00](#)):

And so we have these testbeds at Argonne that allow us to do a lot of this testing with the Intel software tools that will be available on Aurora in advance of having Aurora itself. And so we really like to see folks

get on there and do a lot of testing. If they run into problems, then they need to identify, let's try to narrow down where is this problem coming from? And then we work with Intel to resolve those issues. And I would say, this is something we've gotten very good at. We've really been successful I think developing a close collaboration with Intel and making this process of identifying issues and resolving them successful. So once a code is running, you've ported it, you have your science implemented and you've eliminated all the issues that prevent it from running on our tested platforms, the next focus of activities typically on performance.

Scott Parker ([20:49](#)):

What makes the supercomputer super is that they're capable of delivering an immense amount of performance to do large scale calculations very quickly but sometimes it takes a while to unlock that performance. These machines are incredibly complex. The chips that we run on, have all these unique facets to them and a lot of detail that can be exploited. And sometimes that's an easy process and sometimes it takes more work. And so sitting down and understanding how your code is performing on the machine, what are the bottlenecks that are inhibiting your performance and then working to resolve them is a process that most applications need to spend some time going through. But once they're done with that, they can often get very good performance out of these platforms on a node itself.

Scott Parker ([21:28](#)):

So on a combination of CPU and GPU so it sit on a single circuit board in the machine and then the last step is once you've got it running well on a single node, as you mentioned, we have over 10,000 of these nodes on Aurora. And so really to get the full advantage to running on Aurora, you want to run across a large fraction of those. So two, three, four, 5,000, maybe even the whole machine and so getting your simulation to run across all of them simultaneously and efficiently is another challenge. And that's one we haven't had the opportunity to address yet because we don't have the hardware at that scale but that's another step that we'll be addressing with most of our teams once we're at that point. So and it's sort of a big picture sense that's the broad steps that I think most folks go through to get their codes ready to run on Aurora.

And Mike, I mentioned the testbeds and those have been a key resource. Without those testbeds, we wouldn't be able to do really any of this. So maybe that's something you can talk a little bit about are what are the testbeds we have available? And how does work done there translate to Aurora?

Michael D'mello ([22:32](#)):

Certainly, thank you, Scott. From my point of view, these testbeds systems are fundamental to the whole process of enabling a scientist on this machine, testbeds that we leverage are at Argonne at ALCF. We've got a whole set of machines there that span essentially three generations of Intel GPUs and what they offer our communities, they offer the ability for potential users of Aurora to come and examine the architecture, examine how the code runs on these testbeds and tweak their codes to where they can get them to run optimally. And this is one of those environments where a user can have a dedicated time with this technology and really put some effort into really understanding the nuances of Intel GPUs and CPUs and how they work together and all this and how the science code could actually leverage what's underneath the hood. And there's whole layer of software of course that one has to worry about, all of that software is also available on these testbeds so the oneAPI suite of tools is available and we've found it to be extremely effective. I'd say in many cases, we've had scientists who are able to get their codes

up and running in relatively short periods of time which was not something that was entirely possible in the past.

Michael D'mello ([23:56](#)):

Programming heterogeneous machines with CPUs and GPUs was traditionally considered sort of an acquired skill, something that one would have to really spend some time developing with the various innovations and various APIs that we offer in oneAPI. I wouldn't say they've made easy work of it, but it's a lot easier to program these heterogeneous environments than perhaps any time in the past. We also leverage a whole slew of machines at Intel and the benefit of having those machines also available to us and our users in some cases is that we were able to test things well before they're released. So we can do engineering type tests and really go deep on certain elements of the technology and prepare it in a manner that's if you're much more consumable and much more beneficial to the scientist or to the end user when actually it does appear in front of them. It was a great thing that ALCF allowed us to do to create this environment.

Michael D'mello ([24:58](#)):

And I think that's been one of the key drivers for our users. But let me mention also other aspects of this testbed environment. So one of the things that it enables us to do of course is to actually work alongside scientists. And just that kind of interaction, that one to one interaction has had a tremendous amount of side benefits. So in other words, some of the things that they tell us about the technology as it sits today, it's going to be useful for us as we develop exascale technology even further beyond where it will be say a year from now. So we're heading exascale at the one or two exascale level right now, but in the next five years, the next five, 10 years, we're going to be talking many times that in terms of achievable performance, and we want to understand ahead of time how we have to prepare that technology or design that technology so that it will meet the needs of the community. So I hope that explains a little bit more about how we're using the testbed, Scott.

Scott Parker ([26:00](#)):

Yeah, yeah, thanks Mike. And maybe you could say a little bit about what the most significant challenges that you've encountered and sort of using these testbeds and preparing applications for Aurora.

Michael D'mello ([26:11](#)):

Sure. One thing that we often talk about when we discuss HPC type things is that we just talk about applications in very generic terms. In reality, there are elements in that space which leverage a very wide spectrum, just even the computer languages. So there are Python codes and there are Fortran codes and there are C, C++ codes, et cetera, and they all use different elements of the software stack. Some of which has been developed over the last 25, 30 years. So it's not uncommon to find a code that was written maybe in the early '90s and was written in a style that was very appropriate for the machines of that time. And it has been incrementally developed to run on various iterations of the technology over the years but nothing like Aurora where you have heterogeneous machine at scale, at the scale of tens of thousands of units and things like this. So that's been a big challenge because it takes a little bit of effort to adjust one's mindset towards those kind of goals so I'd say that's definitely one of them.

Scott Parker ([27:20](#)):



Yeah, thanks Mike. I know another problem that we struggle with at this stage in particular is the performance, really ringing the performance out of these systems due to the various levels of complexity, both in the software and in the hardware. So how do you and your team go about addressing those kind of issues?

Michael D'mello ([27:38](#)):

We are very fortunate in this case because Intel has been a very big proponent of vectorization type technologies for many, many years. Vectorization is important and our CPUs vectorization is very important on our GPUs and both platforms, we put that together with parallelism and that's what gets you your scale, your scale of performance so one of the things we're able to do as we prepare for Aurora, and as we work on these testbed systems that I mentioned is we can actually examine the code in depth on the CPU first and leverage that understanding to develop the GPU code. So for example, for both our C, C++, and our Fortran users, we have the OpenMP APIs for example. So this is something that has been around in the CPU world for a long time. It's actually been available for devices for many years as well but at this point in time, it's getting increasingly larger and larger amount of attention in the scientific community.

Michael D'mello ([28:43](#)):

So that's one of the things we do. We actually analyze a code in extent and really understand it to the best of our abilities on existing technologies and then project and see how it might work on a device in conjunction with the CPU. That recipe has proven very, very powerful indeed. There are several groups of people who have been able to show to at least some degree that you can try to program a device in a manner that will have a good deal of commonality with what you're used to and to the extent possible. To the extent that it is possible, of course we leverage it. Now, I'm not saying that's going to happen in every single case. There might be cases where you might have to really think of refactoring your code in a completely unique way in order to get the best part of the GPU but that's not, as I mentioned, not necessarily true in every case. So I hope that explains a little bit more of how we are approaching things.

Scott Parker ([29:39](#)):

It does. And could you tell us a little bit about maybe some of the early successes that have come out of all this effort that's gone into getting things ready for Aurora?

Michael D'mello ([29:48](#)):

Absolutely. So there are several success stories in the pipeline right now some of which are extremely advanced in their readiness if you like for Aurora. One particular case that came out very recently was the OpenMC code. So this was highlighted at ISC International Supercomputing 2022 in the Intel keynote talk. This is part of the ExaSMR, small modular reactors, ExaSMR project. And it's one of those codes which models nuclear transport among other things. It's a Monte Carlo type code. So it has a lot of features that make it particularly interesting from a GPU computation point of view. And one of the things that a member of my group, the COE and PI team were able to show that they were able to restructure the code in a manner that would give essentially some of the highest observed performance for that code on a GPU ever. And that was described in the keynote International Supercomputing 2022. So that was one of the only success stories, but there are others in the pipeline in area of CFD, astrophysics, and AI as well which are in the process of being announced.

So, Scott, perhaps you can outline for us some of the timeframes, the dates, other elements of this picture of having this really sophisticated machine available to the public.

Scott Parker ([31:16](#)):

Yeah. So we're kind of getting to the point where we will be getting the system soon. So our expectation is that the hardware will all be delivered this year and for systems of this scale to bring up, it is a time consuming process. We'll continue to do more of the same in terms of what we have been doing, a lot of testing, development and really trying to make sure that we have things ready when the machine is here. So when the first bits of hardware hit the floor, we'll start running on what's available, whether that's a single node or a handful of nodes.

Scott Parker ([31:59](#)):

And then as more of it becomes available incrementally as it gets built out, we'll try to start using more and more of that hardware. And so really, it's just going to be a lot of testing, early and often and repeatedly to make sure that we're identifying any lingering issues, remaining problems and give the Intel folks time to work those out before the whole machine is in place. And I do think we have over the last couple of years really put a good process in place for doing this and so we've been really, I feel doing a good job of identifying issues and getting resolutions deployed in a timely manner. So if we can keep doing that, I think we'll be in good shape when the hardware is here.

Michael D'mello ([32:38](#)):

Okay Scott, any advice for users interested in preparing to use Aurora, those who have not already started?

Scott Parker ([32:48](#)):

Yeah. So for folks in the broader community that haven't had an opportunity to prepare their code for Aurora, as I mentioned, once the machine is installed and operational, it will be a resource that's available to all those folks who want to submit a proposal, if they get an award. So the first thing I think is understanding that proposal process because you do have to submit something and write a competitive proposal that outlines the research that you want to do. So folks that are interested, it is managed through the department of energy insight program and if you want to learn more about that, there's a website [www.doeleadershipcomputing.org](http://www.doeleadershipcomputing.org), where more information can be found but for people that are really wanting to focus on code development, a key task, as I mentioned earlier, is to pick a programming model and start developing your application in a programming model that enables them to run on Aurora.

Scott Parker ([33:43](#)):

So whether that's OpenMP, Kokkos, RAJA, or SYCL, and to do that development work, one platform they can use is the Intel dev cloud. So Intel is providing resources independent of what's at Argonne for the broader community to start testing out oneAPI and get some experience on Intel's GPU hardware. So those I think are the main steps that folks could do today to get ready. And I mentioned SYCL and that's something that's sort of newer to the community, that's probably the most recent programming model to show up on the scene and one that we're really using heavily on Aurora, but maybe not as widely used other places yet. And Intel, I think has sort of been spearheading some of the SYCL adoption in the community. So Mike, could you say a little bit about how SYCL fits into the picture and the path forward for folks with SYCL.

Michael D'mello ([34:33](#)):

Absolutely. So what's really nice about SYCL is that it appears as a API that's part of the C++ language and that's a tremendous thing for C, C++ developers and users because that language is so prevalent scientific computation today that it's a very timely thing to have something like this available inside the language. And what SYCL is capable of, it's capable of looking at any computing environment as a device if you like. So you could think about deploying SYCL computations on a CPU, GPU, potentially even an FBJ(FPGA?) if the right software is in the stack. I think it's got a tremendous appeal to designers to the scientific code designers because they can stay in one language if you like and just build on core concept that they're expert at basically. And I think that there's a lot more to SYCL that we will learn over time and that will cement its position in high performance computing. SYCL by itself probably deserves its own podcast because it's a rather interesting development with many facets so I'll just stop over there and help the interest of our listeners on this topic.

Radhika Sarin ([35:49](#)):

Yeah, absolutely. I think that totally demands for a separate topic itself and a podcast dedicated just to SYCL. Scott, are there any resources that you'd like to share with us?

Scott Parker ([36:01](#)):

Well, I think the main ones are Intel's development cloud resources. So people that want to get experience, hands on experience with the technology, that's a good place to go. The Argonne website on Aurora has the public information that's available for Aurora as well as any updates that we'll be putting out there as the machine hits the floor and gets built out. In addition, on the Argonne site, there are links to both past and upcoming webinars that we've done, they're called preparing for Aurora and they cover a lot of the technologies that are used on Aurora. And you provide information to folks that they can look at to understand those technologies and how to get ready. And the last thing I'd mention is just again, the [www.doeleadershipcomputing.org](http://www.doeleadershipcomputing.org) website, for folks that are interested in submitting a proposal to use Aurora in the future.

Radhika Sarin ([36:46](#)):

That sounds great. Michael, are there any resources from Intel that you'd like to recommend?

Michael D'mello ([36:51](#)):

Yeah, certainly, I would look at [www.oneapi.intel.com](http://www.oneapi.intel.com) for example and also in terms of Aurora, I would encourage you just to go to [www.alcf.anl.com/aurora](http://www.alcf.anl.com/aurora), A-U-R-O-R-A.

Radhika Sarin ([37:05](#)):

That's great. Good, we're almost out of time, but I wanted to thank our speakers for this insightful conversation. Scott, thank you for your time today, appreciate it.

Scott Parker ([37:15](#)):

Yes, thank you. Happy to have been here.

Radhika Sarin ([37:17](#)):

Michael. Thank you for joining us today.

Michael D'mello ([37:20](#)):

Thank you very much.

Radhika Sarin (37:21):

And a big thank you to all of our listeners for joining us, let's continue the conversation at oneapi.com.

## Abstract

Intel, HPE, and Argonne National Lab are collaborating closely to deliver Aurora. Built on Intel's architecture, Aurora will be Argonne's first exascale, high performance computing system. Its performance is expected to exceed exaflops, which is 2 billion billion calculations per second. It's enormous scale and performance levels will accelerate research in science and engineering fields at a scale that is not possible today.

## Learn more

RESOURCES | [share resources + links](#)

ACLF – Aurora [www.alcf.anl.com/aurora](http://www.alcf.anl.com/aurora)

DOE Leadership Project [www.doeleadershipcomputing.org](http://www.doeleadershipcomputing.org)

Intel® oneAPI [www.oneapi.intel.com](http://www.oneapi.intel.com)

Intel® oneAPI Toolkits

<https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html>

Intel® oneAPI HPC Toolkit <https://www.intel.com/content/www/us/en/developer/tools/oneapi/hpc-toolkit.html>

Exascale Computing Project website <https://www.exascaleproject.org/>

Aurora Early Science Program <https://www.alcf.anl.gov/science/early-science-program>

ALCF Aurora <https://www.alcf.anl.gov/aurora>

Preparing for science in the exascale article <https://www.alcf.anl.gov/aurora>

Argonne National Lab's Aurora Exascale System

<https://www.intel.com/content/www/us/en/customer-spotlight/stories/argonne-aurora-customer-story.html>

Intel and Argonne Developers Carve Path Toward Exascale

<https://www.intel.com/content/www/us/en/newsroom/news/developers-carve-path-toward-exascale.html>

Introducing the Aurora Supercomputer - Powered by Intel:

<https://www.intel.com/content/www/us/en/high-performance-computing/supercomputing/aurora-video.html>

#### External Guests

**Name:** Scott J. Parker

**Title:** Team Lead, Performance Engineering at ALCF

#### Intel Guest

**Name:** Michael D'mello

**Title:** Program Manager, Intel

#### Keywords

- Aurora
- Exascale computing
- SYCL
- SYCL standard
- Heterogeneous Programming
- Parallel programming
- Parallel computing
- Code Portability
- Cross Architecture
- Code Migration
- High performance computing
- exaFLOPS
- OpenMP
- Kokkos
- RAJA programming language
- SYCL

#### Hashtags:

- #SYCL
- #Cpp
- #ParallelComputing
- #Opensource
- #oneAPI
- #HeterogeneousProgramming
- #CrossArchitecture
- #CodeMigration
- #Developer
- #HPC

**Promotion timing** – Aiming for October 10<sup>th</sup>, 2022 release.