Tony (00:05):

Welcome to Code Together, a podcast for developers by developers, where we will discuss technology and various technology trends in the industry. I am your host, Tony Mongkolsmai, and today we'll be talking about the convergence of HPC, AI, and big data analytics in the era of exascale computing. We have two wonderful guests that are going to talk to us about what's going on at Argonne National Laboratory and how it affects both us and society. First, we have Henry Gabb, who is a former computational life scientist who has been working in high performance computing for several decades. His first supercomputer was Cray X-MP, and Henry one API evangelist in the Intel software and advanced technology group. He is also the editor of the Parallel Universe, Intel's quarterly magazine for software innovation. Welcome, Henry.

Henry (00:52):

Thanks, Tony.

Tony (00:54):

Also joining us today is Chris Knight. Chris is the lead for chemistry and material science at Argonne Leadership Computing Facility, and manages a team of computational scientists that work directly with current and future ALCF users to accelerate their research efforts. He has been working with Intel and application developers over the past several years to ready codes for the Aurora exascale system. Nice to have you, Chris.

Christopher (01:18):

Thanks, Tony. Happy to be here.

Tony (01:20):

So as we know, exascale computing is upon us. It's very exciting how computing has grown over the last several years, and one of the places that we have a great opportunity to make strides is with the Aurora system at Argonne National Laboratory. Chris, can you tell us a little bit about how you guys are planning on leveraging it?

Christopher (01:37):

Yeah. I think by far one of the biggest ways that we're going to use Aurora, in really the research community, is by way of the challenging questions that researchers are asking, and these are questions that are becoming more and more challenging to answer, and I think this is true across multiple science domains, that all of these questions are demanding more computational resources, maybe for higher fidelity simulation models or increased flexibility in the types of calculations that researchers are using in their work. Really, and then thinking about the past several years, the nature and scope of projects at the ALCF has really grown in complexity, and I think this is very much where simulation, data and learning aspects really are being adopted by many in the community.

(02:23):

There are certainly some that still run the large hero size simulations that they will need the full Aurora exascale system for their one or two simulations, but there are many others that have many components in their calculations and the work that they want to do, and they need frameworks to tie them all together and to orchestrate these workflows on a large system like Aurora, and then I think that the great thing is really from the start and early on Aurora, was designed from the get-go to be

focused and to enable simulation data, and learning, and the coupling of them at scale on a large system, and really, all of this is with the mission of DOE, and the ALCF is just really to help researchers take that next big leap in breakthroughs in scientific challenges. I was curious, Henry, with all of the technologies that we have in the Aurora system, if there are any, in particular, that you think are going to be important for the community?

Henry (03:20):

Obviously, it will have the CPU and the GPU heterogeneity. That will turn out to be critical, but I'll turn the question back; how do we program it? From your point of view, you're a computational chemist, and so when I think of exascale, and computational chemistry, and the kind of compute power you'll have in Aurora, is it really just as simple as saying, "Okay, I've got exascale now. I've got Aurora. When I solve, now I can solve much bigger molecular systems than I could ever have attempted before," or is it something more than that?

Christopher (04:01):

I think it's something more than that.

Henry (04:03):

Yeah, I would hope so. I would hope it's something more than that, because I can appreciate being able to solve bigger molecular systems. You're a chemist, I'm a biochemist, but I'm looking for that convergence of traditional HPC, AI, big data analytics that Aurora can deliver to solve the really big problems, but also to answer new scientific questions that weren't possible before.

Tony (04:30):

Yeah, that's interesting. Chris, what type of problems are you looking forward to solving that you feel like Aurora's going to give you a chance to tackle that you weren't able to tackle with previous systems you've had access to?

Christopher (04:41):

Well, in the area of chemistry and material science, there will be, again, the hero-sized calculations, very large-scale advanced electronic structure calculations with quantum chemistry codes, and then there'll be solving challenging problems, looking at excited states within a protein, for example, but I do also think that there's going to be many other projects that will be more of a discovery type of research; trying to identify the next best battery material, or the next best catalyst for certain chemical reactions, or to generate a new type of a solar cell for sustainable energy.

(05:19):

And they'll need a large machine like Aurora to run many hundreds of thousands, millions of calculations of using some advanced methods, and then not only just running those calculations and getting output, but really using machine learning methods to learn, and to navigate all of that output, and to pull out useful information and then maybe guide additional calculations to guide experimental work. It'd be great if Aurora was used for automated workflows to predict materials and their properties, and that result was then passed off to an experimental research group that then created the material, and it was actually demonstrated in practice. These are the types of problems that a machine like Aurora would hopefully enable.

Henry (06:07):

Yeah, I guess if it explores that very large parameter space. When I think of drug design, I used to do research in molecular docking and how molecules fit together, and it was all just shape fitting, but with AI now, scanning that structural conformational space, the AI then is predicting what will fit together best, and say, what drug will fit into the active side of what molecule? And so you have the AI doing prediction, predictive analytics, but then returning to first principles and doing the actual computational chemistry calculation that says, "Okay, this is at least chemically and physically viable. Now let's hand it off to experimentalist," like you said, to do the expensive bench work, to do the actual experiments to say, "Okay, is this new material or is this new drug doing what we think it should be doing, and is it beneficial?"

Christopher (07:08):

And these are just examples in chemistry, and material science and biology. Industry will benefit in similar ways, engineering applications, researchers using computational fluid dynamics models coupled with machine learning models to construct databases of maybe modeling an industrial process or airflow over an airplane wing, and having automated workflows potentially to design those types of objects, and really taking a lot of the guesswork out and getting to solutions faster, meaningful solutions.

Henry (07:40):

Right, because in every example you just gave, the search space is very large.

Christopher (07:46):

Indeed, indeed, and I think that's the great thing, is that these are challenging problems, and many of them need a machine like Aurora, and I think as the community gets on to Aurora, and they learn how to push Aurora to the limits, and learn what new questions they can begin to ask, and really, it just continues this never ending process of the need for larger and larger machines, more flexible machines, to enable science to continue moving forward. It's exciting.

Henry (08:16):

Yeah, back when I was in graduate school, and this would've been 1988, '89 timeframe, I went to a conference that NSF, the National Science Foundation, had sponsored as we looked ahead to terascale, and I remember an experimentalist, I was an x-ray crystallographer at the time, and an experimentalist said, "Well, you people don't need a terascale computer; you just have to be more clever with the resources you have," and that always stuck with me, and when we did hit terascale, by that time, there were problems where you needed a terascale computer, so now we're looking to exascale, and like you said, we may not even know the problems that we're going to solve with it, until we start running on a system that size and seeing what we can actually do now.

Tony (09:08):

Yeah. Chris, how does your team prepare? I mean, I'm sure the software stack has to change from running on, I'll say smaller scale systems, which are still quite large, that you have access to, to really running on an exascale system. Is that something that your team has to prepare for and think about as you're kind of doing your research planning?

Christopher (09:25):

Yes, definitely. We have to sort out as much of the process as we can ourselves so that we can then help the community begin the process themselves, and then this is something that started several years ago, the beginning of the Aurora project itself, understanding the changes in the hardware, understanding the changes in the software stack, and the implication of that on the application teams and the software that needs to be ported to Aurora. And actually, I think one of the great things in the recent years, and I believe some of this was driven in part by the Exascale Computing Project and the exascale machines, is just the diversity in the ways that application developers can program for the GPU, the various programming models now: DPC++, OpenMP, Kokkos is continuing to rise in popularity on the frameworks side, and making use of accelerated math libraries, and so there's a lot more opportunities for application developers to really take their important codes and make use of accelerated architectures.

Henry (10:27):

Well, Chris, I have a question for you; do you like to program?

Christopher (10:32):

Yes, when I've got the time to do it.

Henry (10:37):

Okay, fair enough. Oh, and do you like heterogeneous parallel programming?

Christopher (10:42):

Yes.

Henry (10:42):

Okay.

Christopher (10:44):

And I would say that there are many in the community that are, let's say, not afraid, but that don't want to deal with the perceived challenges of programming for an accelerated architecture, and the look of surprise on their face when, say, a proof of concept example with something like OpenMP Target Offload is presented to them, and a few lines of additional code, and with OpenMP, that's usually chosen because they may have familiarity with multi-threaded programming. The look of joy on their face when they see an acceleration on the GPU with a very low effort for the initial proof of concept, that is definitely something I've noticed more and more in recent years, and more of the community.

(11:31):

Those that are primarily using multi-core architectures, for example, in their work, I think the barrier for them to program for hybrid architectures, accelerators, and so on is becoming much lower, and it's much more accessible to them maybe nowadays. And that may be perception, but still, I do think it's real. It is becoming much easier for the community as a whole to take large legacy-based applications and software stacks and move them to an accelerator model, and with that, of course, comes a number of benefits, usually with optimizations that are motivated by running on an accelerator that are able to

also run on the CPU. They see benefits. I think there's a number of good things coming soon when the rest of the community gets access to these exascale systems.

Henry (12:17):

Yeah, I kind of feel the same way. There is that joy of acceleration that you take a problem with a little bit of coding, a little bit of modification, and you get this a hundredfold speed up, and the joy of the hundredfold speed up is that now it opens up new possibilities. I say I don't like to program, but actually, I do; I'm just not good at it, and so when I have a tool, whether it's OpenMP Target Offload, Kokkos, oneAPI, that makes it easier for me to do the acceleration and to take advantage of the accelerator, and I see that really large improvement of performance, that yes, there is that sense of accomplishment from getting the performance, but then there's that scientific joy that what new questions have just opened up?

Christopher (13:07):

Exactly. That's the driver for a lot of us in ALCF, is helping researchers run at scale on the large systems that we stand up, and we do that in part to help them be successful, but we're really interested in having that conversation with them. You've solved this problem. What's the problem that you really want to solve? What's the problem that you're not able to solve today, and how can we help you to solve that? Whether it's advancements through software, improving the application, or helping them to get ready for the next big machine, like Aurora, that's definitely an important driver and a big motivator for helping the community to work towards accelerated programming models.

Tony (13:46):

Chris, you mentioned that you were looking forward to be able to use machine learning, to kind of mine through the data that you have and simulations that you're doing. Is that something you guys did before, or is that something that's going to be brand new because you have all this additional horsepower with Aurora?

Christopher (14:00):

No, we've actually been a champion of this for a while, the coupling up simulation, data and learning at ALCF. Maybe several years ago, we started running a Simulation, Data and Learning Workshop, where we invited those in the community to come to ALCF to understand their needs, first and foremost, but to understand how to help them run at scale their machine learning workflows, their data-intensive workflows. The ALCF, we have at Argonne, the Data Science Program. It's a proposal-driven process, where we invite teams to submit proposals, and the awardees are given considerable amounts of time on the resources, but also maybe more importantly, they're also given people resources, and so they can work closely with the facility to help enable a new scientific capability, melding simulation, in some cases, but mostly learning and data, really with the driver to do it at scale.

(14:55):

This started with the Theta system that we have at the ALCF and Intel Knights Landing based architecture. That was really the initial stepping stone for Aurora, and we're continuing that now with the Polaris system that we've just made available to users. And this, for us, is particularly important, as it helps us test out technologies like Slingshot, but also programming models and frameworks, DPC++ based applications at scale. That's something that we're very interested in looking at, and, of course,

helping teams to run their workloads at scale, and be even better prepared for Aurora when they get access to it.

Henry (15:33):

And so you said at ALCF, let's say I'm a scientist, and I have an allocation on Aurora, and I'm not particularly good at parallel computing, much less heterogeneous parallel computing. There are software engineering resources that can be devoted to make my project worthy of a system like Aurora?

Christopher (15:57):

Yeah, indeed. One of the great things about the programs, this is true of both leadership computing facilities, but at ALCF, we have a very lightweight mechanism for getting new users onto our systems. These are called Director's Discretionary projects, and so any researcher at a university, industry, other government agencies, that have interesting problems, and potentially, a need for large-scale computing, can submit a very lightweight proposal submission, if you would like to call it that. Then within an order of weeks, two or three weeks, maybe, in some cases, they can get access to the ALCF systems; they could get access to the Aurora system.

(16:38):

And that's an excellent opportunity for them to kick their tires on the system, to explore, to test their specific workloads, and then raise questions, and facility staff, such as the team that I help manage, are there to provide assistance, whether it's how to compile their particular application with oneAPI, or to help explain some unexpected results that they might be seeing, either performance or hopefully not correction. And then when they've had enough success, and they're comfortable, and they've confirmed that they have a need for running at-scale, they're able to definitively use significant fractions of an exascale machine, for example, there are allocation programs, ALCC, and INCITE, that they can submit proposals to, and really they can be awarded substantial amounts of times on these systems.

Henry (17:29):

Yeah, I guess saying to make a project worthy of Aurora sounds a bit snobby, but I worked on a system, it was a vector system, and the policy was if your codes were not vectorized, if they were not taking advantage of the architecture, you could run elsewhere, that it was a scarce supercomputing resource, and you were expected to put some effort into taking advantage of the architecture. That's really what I meant.

Christopher (18:00):

Yeah, but that's still true today in many ways, and I think that comes in part of the team demonstrating that they're making use of the capabilities and features of the systems, Aurora, in this case, and that they really do have a need for the large-scale computing resources. Working with teams, maybe we help them and we improve the performance of their code by some speed up. In some cases, we have that joy of helping to accelerate their code, their workflow, and now, with that new, faster code, they no longer necessarily have a need for the large-scale computing, because they're able to do the work now on a local resource. That's a happy moment, but of course, that's all part of the process.

Henry (18:43):

But architecturally, Aurora has a lot to offer. It's got high speed networking, high speed I/O, the multi-core vector, and parallel CPUs, massively parallel GPU. Architecturally, it's interesting, and it has a lot to offer for projects that can take advantage of it.

Christopher (19:04):

Definitely. There's something for everyone on this machine. For those that are doing the traditional simulations, if you will, there's a lot of computer resources on these nodes, and the high speed interconnect between nodes, they're definitely going to help with accelerating the simulations. The fact that the nodes have a unified memory architecture will greatly facilitate new teams moving their workloads over to the Aurora system so they can worry less about migrating memory between the CPUs and GPUs and worry more on getting the science done. The large amount of memory on the system, coupled with a very fast file I/O via DAOS, Distributed Asynchronous Object Store technology, that we have on Aurora, will be very beneficial to data-intensive workloads.

(19:53):

Users being able to quickly load a large database, either from outside of the ALCF, or from one of the ALCF file systems onto the Aurora compute nodes, and then execute machine learning type of workloads on the nodes, and maybe somewhere else on the Aurora system, they have some simulation calculations that are running that may be coupled to the machine learning model. There's a lot of technologies in the Aurora system that are going to enable a lot of unique and very interesting workflows, and this is all in addition to the software part of the story that will be on the system; the programming models, the libraries, the tools, and really having all of that work supporting all of these different workloads is really going to be a benefit to the research community.

Henry (20:39):

But in terms of the actual environment, it will be familiar to anyone who has worked on a large HPC system, not necessarily exascale.

Christopher (20:48):

Yeah, and I think that's one of the nice things, really, that has emerged, is users that log into a supercomputer today, that experience will look very similar to the experience when they log into Aurora, and so here is one of the first exascale systems in the world, and when they log in SSH their username, it will look exactly like any other supercomputer. Now, there's a lot of compute power that they have access to, but the interface itself will look very familiar. I think that's a great thing, is we're helping to build these advanced machines with significant amounts of computational power, and they're still accessible to the research community, that the research community can focus more on the science and their codes, and less so how to use a new supercomputer. Of course, the facility staff is there to help when needed.

Tony (21:40):

So for the people that you're talking to, Chris, a lot, is there one particular group of customers or people that you work with that are just super excited to basically jump on Aurora and use the resources, or is it just everybody?

Christopher (21:52):

It's just everybody. I was giving a talk at the American Chemical Society here in Chicago, a very high level talk on preparing codes for exascale. Six people, maybe, afterwards asked when could they get time and when were they going to get access? There's a lot of excitement about these systems, and I think that's the great thing, is that there isn't a secret process to getting access, that once the machine is online and available, anyone in the public can get access to it, as we talked about earlier, with the Director's Discretionary projects. There's a lot of excitement, and especially so with some recent announcements of performance results from applications; the recent talks at the ISC conference and Hot Chips, and really talking about some of the initial application results that real applications and the real application developers are themselves experiencing. There's a lot of excitement going on right now, as we get closer and closer to launch.

Henry (22:52):

Talking about it makes me want an allocation, myself.

Christopher (22:57):

Believe me, we are all very eager to get some science done on the system.

Tony (23:02):

So, Henry, I guess I'll toss one to you. We've been asking Chris a lot of questions. Obviously, you're working for Intel, so you're more trying to make sure that the infrastructure is set up for them. What are you excited about? You said you want an allocation. What type of problems would you want to throw at Aurora?

Henry (23:17):

Oh, I'd go and look at the old problems I used to do when I was still doing basic research. I love structural biology, so I would look for structural biology problems worthy of the architecture. I still have that sense that you should be taking advantage of the architecture or running elsewhere, but I would be looking to see how I could leverage machine learning to some of the problems I used to work on in molecular docking, and one thing I'd really like to try, and certainly not as good a computational chemist as Chris, but I used to study nucleic acid structure, and while everybody else is paying attention to the hydrogen bonds that keep the two strands of DNA together, I was looking at something called base stacking, which was poorly understood, and it might still be poorly understood.

(24:06):

I haven't kept up with literature, but I would love to go back to those kinds of problems, because it required serious computational chemistry. Classical molecular simulation, where you ignore the electrons, and you're only considering the atoms as point masses, that doesn't work when you're looking at something like base stacking. You really have to go to first principles. You really have to Schrödinger's equations, but we just didn't have the compute power at the time, when I really wanted to study base stacking. I'm pretty sure we've got the compute power now, and if I return to chemistry, that would probably be where I'd go first. Now that I think the compute power is available, I'd love to give it a shot.

Tony (24:54):

It sounds like we need to put together a proposal for Chris and get into that director program.

Christopher (25:03):

It would definitely be a winner.

Henry (25:04):

Yeah, Chris, you have any interest in nucleic acid based stacking?

Christopher (25:08):

Yeah, I mean, it's Van der Waals interactions, and so there's been a lot of development in recent years on efficient ways to compute those with accurate methods, and so I could easily see the need for using electronic structure calculations to get those interactions right, and then to really get the sampling done correctly, you most likely want to fit a machine learned force field, for example, to reproduce your quantum calculations, and then you could use that force field in probably an ensemble of molecular simulations, and really get at the heart of what's going on and why some certain phenomena arise the way that they do. That's definitely a project.

Henry (25:50):

Yeah, that's a good idea, and I'm going to pretend it was mine.

Christopher (25:57):

That's okay.

Tony (25:57):

And with that, I think that's probably about the end of our time today. Is there anything that you guys want to point people to kind of understand a little bit more about how Aurora works, or how we're putting together the system, the partnership between Argonne National Laboratory and Intel?

Christopher (26:10):

I think I would direct folks to the Aurora webpage that we have on the ALCF website. That's probably a great first stop for anyone who's interested in the Aurora architecture and the ALCF facility as a whole, and from there, I think folks may find themselves wandering through the website, in particular, the stories that are available there, as more stories come out of teams getting ready for the system and preparing for Aurora. There's a lot of good content there.

Henry (26:42):

And I guess, for me, it would be take a look at oneAPI and the different heterogeneous parallel programming approaches to really take advantage of the compute architecture. I was never really a big fan of parallel programming, but I was even less of a fan of heterogeneous parallel programming, but now that I'm starting to see that really good approaches, whether it's OpenMP Offload, whether it's SYCL, whether it's existing highly tuned frameworks to take advantage of these heterogeneous parallel architectures, I'm a lot more enthusiastic about it than I was before, and so I would say, take a look at the different programming approaches that have come on scene, and start learning how to use them.

Tony (27:27):

All right. Well, thank you, Chris and Henry for joining us today and teaching us a little bit about the exciting possibilities that Aurora will make happen for us, and with little side discussion about chemistry, and thank you to our listeners for joining us. We hope to talk to you soon with some more interesting technology topics.