the Exascale Report™

A Global Perspective

The voice of the emerging exascale community

http://TheExascaleReport.com
August 2013

Argonne's Pete Beckman Discusses the ARGO Project

The Quest for an Exascale Operating System

By Mike Bernhardt

We are three years away from a prototype exascale operating system.

On the heels of the ARGO announcement, we sat down with Pete Beckman, Director of the Exascale Technology and Computing Institute at Argonne National Laboratory, and a Senior Fellow at the University of Chicago's Computation Institute.

We asked Pete to expand on the discussion of the quest for an exascale operating system.

This interview is also available as an audio podcast at either of the following links:

https://archive.org/download/ArgoPodcastWithPeteBeckman/Argo%20Podcast%20with%20Pete%20Beckman.mp3

https://dl.dropboxusercontent.com/u/8413402/Pete%20Beckman%208-8-2013-final.mp3

"We believe that getting the scalability right from the beginning is a key design point. I'm not worried so much about the scalability as I am about the functionality."

The Exascale Report: *Thanks for joining us Pete. Why don't we start with a discussion of the team – or teams that are involved in the ARGO project.*

BECKMAN: There are two teams that were funded. One is led by Argonne, and that team is called ARGO. Our team includes Lawrence Livermore National Laboratory, Pacific Northwest National Laboratory, and several universities - the University of Tennessee (UTK), Boston University, University of Illinois at Urbana-Champaign (UIUC), the University of Oregon and the University of Chicago.

The other team is the Hobbes team, lead by Sandia National Laboratory. A very high level overview of both of those projects was presented at the ROSS workshop, run by Kamil Iskra and Torsten Hoefler. The ROSS workshop is held in conjunction with the International Conference on Supercomputing. Both ARGO and Hobbes gave brief introductions on their projects.

TER: *Are these parallel projects?*

BECKMAN: Yes. We recently had a kickoff meeting to launch the ARGO and Hobbes research projects. A total of 30 research scientists attended. The goal of the meeting was what you are hinting at - to find out where our approaches are complimentary, where they are competitive, and maybe where we would work together. In the standard research funding model, proposals go in and are selected based on their merit, and we assume proposals have competing ideas, and that the ideas map our different spaces in the research arena.

TER: *Now that the programs / projects have been awarded, so to speak, it seems that you are not necessarily competing with the Hobbes program. It makes more sense that both programs move forward in much more of a collaborative fashion – or as much as possible anyway. Would you agree with that?*

BECKMAN: Yes, that's exactly right. In the areas where we have similar ideas, we want to see if maybe we can actually share work – do work that is collaborative or maybe even share code that we write.

Of course there are also areas that are quite different. For example, the Hobbes proposal, which I'll let you speak with the Hobbes folks about, includes virtualization, and they have some scientists who are experts in virtualization on their team. That's a concept that the Hobbes project believes is very important for looking at exascale operating systems.

We're using a different technique in our operating system research. We're looking at specialization across cores and partitioning cores as needed. So these are two different techniques and we'll be comparing them. We'll look at how well our idea works for exascale and they will be looking at how well their idea works for exascale.

TER: *The ARGO project was awarded $9.75 million. Is that evenly distributed over the period of 3 years? And is it based on milestones that are required to release the subsequent payments?*

BECKMAN: Yes, it is divided up evenly across all three years and the milestones are broken out to begin in the first year to explore the space and understand some of the key scaling and design issues, and in the end, after three years, we will have a prototype that we'll be evaluating on a handful of architectures.

Argonne's Pete Beckman

TER: *And how many researchers do you have involved in the ARGO project?*

BECKMAN: Around 27 looking at the set of Principal Investigators (PI's) and senior staff across all the organizations. For a project this big, that's a pretty standard number. There are eight institutions, including Argonne, working on ARGO, and each institution has 2-4 people in the mix and doing something for the project.

TER: *Pete, are there any system or software vendors from the manufacturing side involved as collaborating partners in this?*

BECKMAN: Yes, in fact, part of the ARGO proposal we had submitted was the creation of an advisory committee of the large computer vendors who are interested in exascale software. That includes NVIDIA, IBM, Intel, CRAY, ARM, and AMD.

So there is a representative from each of those companies in an advisory committee that we intend to convene as soon as we get started. Because the intent of this project is to develop a new prototype operating system as open source that can be used for extreme scale systems, that's not something that the Department of Energy wants to productize themselves. This is research in the same way that MPICH, which is here at Argonne, is research. Nine out of ten of the biggest supercomputers deploy the MPICH research software after it is productized by a vendor and shipped with their systems. So the same sort of commercialization path is what we would expect for ARGO.

TER: *In some of our very first articles for The Exascale Report, there was widespread agreement that we would never make the necessary progress in terms of exascale software without strong international collaboration. But this does not sound like it has a strong international collaborative aspect.*

BECKMAN: Well actually there is. As for government sponsored research, it's very difficult for funding to go across the ocean. It's very complex as you can imagine. So instead what's been happening is the individual project is of course funded here and at the universities in the U.S., but there's always been a very clear partnership with different places in the world. Bill Harrod and I were in Leipzig, Germany recently and met with the officials from MEXT (Japan). We have a memo of understanding already with the Japanese, and the next step is moving forward together with joint research pieces that fit together. We also have students from France, Japan and other places visiting this summer. So the research happens that way, even though with funding - it's difficult to take U.S. funds and spread it across oceans.

TER: *So Pete, let's jump ahead just a bit. At the end of the 3 year project, how far do you think the prototype operating system will scale?*

BECKMAN: That's a good question. We believe that getting the scalability right from the beginning is a key design point. I'm not worried so much about the scalability as I am about the functionality. In other words, we're going to start by making sure that everything we do is scalable and that we can see a path toward millions of threads of control. The real challenge for us is seeing how much functionality we can deliver that is scalable like that in the time frame and given the budget we have for the three years.

We'd love to be able to deliver functionality that does automatic balancing of electrical power and managing our needs for applications, while also doing resilience and looking at ways to automatically restart parts of computation from the OS side. But those considerations, if you are looking at the long list of functions that the operating system needs to support, can be pretty daunting, especially as we look at extreme scale and the constraints of power and the massive parallelism we have. So we'll be rolling out several areas, sort of in turn, and the real question for us is how many areas can we plot through and have really great functionality, as we attack scalability.

TER: *Is it correct that your goal is platform independence in terms of an operating system?*

BECKMAN: Yes. Of course on an operating system, you have hooks down to the low level hardware, so we are going to have to in some sense, port to the low level pieces of hardware, but the architecture and the structure is something that we want to be portable and we hope in partnerships with vendors that they will be willing to move forward in porting it to some of their future architectures that we don't yet have. As a good example, there is currently hardware on several of the chips that allows for a very special way to notify the processor when a very lightweight message arrives. On the

Blue Gene/Q hardware, this particular feature is called the "Wake On" feature. This feature is very nice because rather than constantly checking, or what we call 'polling' for whether a message has arrived, there's a way to put a hardware thread to sleep and then when a message comes in, it automatically gets scheduled within just a few clock cycles. That's a very powerful hardware mechanism that we're hoping other vendors are also adding. There is a similar technique being used by other vendors. So, those are hardware-specific. As we look forward to the operating system being portable, we absolutely have the design for portability, but there will be some work required to take advantage of the individual pieces of hardware that the vendors, honestly, are competing on to come up with the best ideas there.

TER: *Let's talk about some of the specific items you will attempt to address. What about the dynamic reconfiguring of the node resources. What are you going to face in terms of that challenge?*

BECKMAN: We're really looking at four big areas that we think are the key places for exascale – or extreme scale – at least those we're going to tackle. We realize the Hobbes team will have other ideas. But from our perspective, these four areas – one is the node OS itself and we have to be a little careful here because usually what people think of when they think of an exascale OS, they immediately drill down and say, ok – what are you running on the node? But we don't view it that way.

I'm going to hold the node OS thought for a minute and jump to one of the other areas which is the global view and global optimization. Our whole project is designed around this concept that we have a global view of the machine.

So rather than the machine being just a single collection of individual nodes, a collection of what a lot of times you would think of as individual machines, we're looking at it as an operating system running across the entire platform. In that way, we can have handles on power management, on bandwidth management, on how to manage memory and other things that go beyond just looking myopically at a single node.

So we will have a node OS, and we're going to leverage some of the work the community has already done and explored – things like FUSE OS and partitioned operating systems and quality of service. We're going to be leveraging some of those ideas and we have some of our own ideas in those areas – especially with respect to memory management. So the node OS is one area, and then as I mentioned this global view and global optimization of resources.

Right now systems are not very reactive. In other words, a system will run and something will happen – and it will get logged into a log file, but the system itself isn't responding autonomously, looking at what's happening on the system and then responding to it. So that global view is number two on our list of items.

A third thing on our list is a supporting construct for that – and that's really the idea, that information about the system – the control information and the performance information - needs to flow up and down through the system, from the nodes all the way up to essentially the console. We call this the backplane. As I said, right now, most users think of themselves as having a collection of nodes that they own but we really need that performance information, for example how much energy processors are using, to bubble up and make its way to the console because the system has to be balanced as a whole.

The fourth area is really parallelism and light weight threads and concurrency.

As you know, everyone is predicting very large multi-core machines with hundreds or thousands of threads of control, and hundreds of cores. Right now, the structures in an operating system are really designed for tens or hundreds of UNIX processes. If you just think about – when you get on your machine and you might do a list of all the processes – the processes are very heavyweight and you can in fact do a list of them and you expect to see one or two screens of data. But in reality, as we look toward really fine grained parallelism with lots of threads of control and maybe even over-decomposition of our science problems to generate even more user-level threads, then we have to really move to a different model and the operating system has to be able to manage and schedule appropriately very lightweight constructs. So we've partnered with Laxmikant Kale who has been doing this for quite some time at the UIUC, he is the force behind the runtime system Charm++ that's used in NAMD and many other computations.

And then we partnered with the folks who are doing the runtime system and the messaging inside MPICH. Putting those two teams together gives us a new way to handle massive parallelism.

TER: *So Pete, how do you build and test all of this when you don't have physical exascale systems to test it on?*

BECKMAN: Well, that's a good question. The first thing is that we have a couple small test systems that we just use internally but that does not test the scalability. To test scalability we have to do either some modeling and maybe a little bit of simulation – or emulation - but in the end, that only gets you so far. In the end you really have to be able to run these components at the largest machines.

Now we've been very fortunate here at Argonne in that we have a couple large Argonne Leadership Computing Facility (ALCF) machines. Intrepid, our IBM BG/P can run our experimental OS called ZeptoOS. It has been used to study extreme-scale systems around the world, and in some locations, used in production. Our new 10PF system, Mira is an IBM BG/Q, and we hope to test some of our future OS designs on the system, including memory management, lightweight threading, and the global backplane. We hope to also test on Titan and other extreme-scale machines.

TER: *So speaking of ALCF, how does the Exascale Computing Technology Institute align with ALCF?*

BECKMAN: Another good question. Both at Oak Ridge and at Argonne, the Leadership Computing Facilities are production computing facilities. They themselves are not computer science research organizations. They are meant to stand up and create a resource that is then provided through Innovative & Novel Computational Impact on Theory and Experiment (INCITE) to the entire community. So, we work with them to explore new architectures and understand what the vendors are producing, but the computer science research happens inside the computer research division which is our Mathematics and Computer Science division – and we partner with the ALCF on the deployment of that technology.

TER: *Switching gears again, it's only been a short time since we announced a new Secretary of Energy, but do you see anything coming down the road in terms of changes to the way you guys operate.*

BECKMAN: It's not quite clear yet. I know there have been some discussions already about organizational changes that could improve how the DOE works in general. I think that's a fantastic idea. From our perspective – the scientist's perspective – fixing some of the complications that we currently have with respect to conferences and travel has been on all of our minds, as you know from Supercomputing (SC). Hopefully the Secretary will be able to address that.

TER: *Tell us a little bit more about the 'Enclaves' that you announced. Where did that idea come from*?

BECKMAN: It comes actually from a frustration that system software folks have – which is – if you want to run a component that is in some sense a meta component, a piece of software that you use to manage a set of nodes, on our current supercomputers there's no place to run that. Let me give you an example. Suppose that you wanted to write a little piece of code that would cache file I/O for your entire computation. So, in your computation, you know there's a certain pattern of I/O and you know that you could improve performance with a little cache piece of code that caches your specific I/O. Where do I run that? Well, the right place to run that is somewhere between all of the small nodes that are generating the I/O requests and the set of file servers. But it's very difficult to schedule -- where do I put that code? Where does it actually run? Now on the Blue Gene, there always was an intermediate set of nodes called the I/O nodes, and our IOFSL and ZOID research pushing bits of code there. Now, it isn't just I/O that I'm talking about. Another great example would be performance data. Suppose you would like to gather some information about how your code is operating and then based on that information, maybe change how it's running or some parameters of its execution. Again, it's really a piece of code that collects status information from all of the nodes – and then an algorithm is run on that information, a determination is made, and then it tells the other compute pieces what to do or maybe to change its plan. And again, there's no place to run that on our current systems. It's a separate executable and I'll give you one more example of this: the folks who are doing 'many task' computing. So these are people who start their application with a single problem – they're doing some exploration say of compounds that would be good candidate therapeutic drugs – and looking at potential matches. They start with a very large search space and then they want to manage hundreds of thousands of sub-tasks that they generate as they explore the space.

Well, that task scheduler is a piece of code that has to run somewhere. It's not the code that is doing the matching or doing the evaluation of the potential targets, but it has to be the scheduler in a sense – a micro scheduler of hundreds or thousands or millions of tasks. Again, there's no real place to run that. We decided that rather than trying to shoehorn this in to bits of hardware or servers that sit next to the supercomputer, really what we need to do is re-think the model. We chose the word 'enclave' to represent this notion of a group of nodes that can be collectively managed. We use a different word because we don't want to overload the word partition or job because that immediately has a lot of connotations that people follow behind it with. So instead we just said let's assume that for every large group of nodes we assume that there's some management there – some place where I can run some meta code. And we'll call that large group an enclave. And the beauty of computer science is that we can do this recursively. We can just start at the top of the machine and say the entire machine is one enclave – which is the system console – and then you can break it up and for a particular job you might have another enclave and that job might have both a climate modeling piece and an economic modeling piece that might run on separate partitions. So that recursive decomposition of groups of nodes that behave together – in unison – but also has a place where I can run some management code or even user code to coordinate those resources is our concept. We are exploring several new spaces with both where we are looking at NVRAM, and what will happen in the future with 3D Stacked Memory, so this will be a very fast-paced and exciting project over the next three years.

TER: *Do you expect to be able to maintain a very high level of transparency with this so we can follow the progress on this project?*

BECKMAN: Absolutely. In our previous project, the ZeptoOS project, where we worked on operating systems in this area with a different idea – but explored the space, we had a public repository of our code and we released it and packaged things up so people around the world were using it and we would expect the same to be true just like other Argonne codes such as PETSc, MPICH, and all of our toolkits. We make them available as open source and the repository is made public so it's easy for people to check in and see how things are going.

TER: *Any other points about ARGO you would like to share with the community?*

BECKMAN: I think it's important to understand that this really is a research project and that we're going to explore these topics. Our intent of course is to produce a prototype. We're trying to design the next generation view of how to build these large platform operating systems for supercomputers. There are parts of the project that we would consider high risk – high impact. It's a project that involves several universities and laboratories and moving parts and we expect to see some exciting breakthroughs in some areas, and in other places we might try an idea, and then have to return to the drawing board.

TER: *Exascale by 2020?*

BECKMAN: I hope so. It will depend on the investment that finally happens after Congress gets back in session.

# # #

Pete Beckman bio
http://www.anl.gov/contributors/pete-beckman