Toward Speed and Simplicity: Creating a Software Library for Graph Analytics

featuring Eric Werner and Scott McMillan as Interviewed by Suzanne Miller

----

Suzanne Miller: High performance computing is now central to the federal government and industry as evidenced by the shift from single-core and multi-core or homogeneous central processing units, also known as CPUs, to many core and heterogeneous systems that also include other types of processors such as graphics processing units also known as GPUs.

Researchers at the SEI's Emerging Technology Center are working to create a software library for graph analytics that would take advantage of these more powerful heterogeneous supercomputers to perform graph analytics at larger scales and more quickly, while making them simpler to program. They are more complex, and thus, more difficult to program.

These algorithms are used in DoD-mission applications including intelligence analysis, knowledge representation and reasoning in autonomous systems, cyber intelligence and security, routing planning, and logistics optimization.

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

My name is Suzanne Miller. I am a principal researcher here at the SEI. Today, I am pleased to introduce you to Eric Werner, who has spoken with us before [Architecting Systems of the Future Podcast], and Scott McMillan, both of the SEI's Emerging Technology Center. Before we delve into their research, let me first tell you a bit about our guests.

Scott McMillan joined the SEI Emerging Technology Center in May 2013 as a senior software developer, bringing with him more than 20 years of experience in high-performance computing across a wide range of application areas including scientific computing, 2D/3D geographic information systems, and very large-scale database systems. Scott comes to us from Raytheon where he contributed to a wide range of projects from cybersecurity, wireless communications, embedded systems, and Windows-device driver research.

Eric Werner is the technical director and chief architect for the SEI Emerging Technology Center at Carnegie Mellon University where he sets the technical direction of the center in the areas of software development, high-performance computing, cloud computing, and data analytics.

Werner has more than 15 years of professional software development experience, primarily focused on delivering global, mission-critical software systems. Prior to joining the SEI, he was the lead architect and systems engineer for the United States Army's Command and Control System with General Dynamics C4 Systems.

Welcome to you both.

Scott McMillan: Thanks for having us.

Eric Werner: Thank you.

Suzanne: Absolutely. There are a lot of different aspects to the graph analytics work that we are going to talk about. Let's begin by having you talk about graph analytics and the role of graph analytics in high-performance computing.

Eric: Graphs are a fundamental data structure in mathematics and computer science. They are used to represent all sorts of problems, as you said in the introduction, from logistics planning, intelligence analysis, social network analysis, et cetera. They are different from standard high-performance computing because they have certain properties that make them hard to compute. They're hard to parallelize. I think Scott can tell us a little bit about why.

Scott: Sure. In traditional scientific applications, memory accesses are pretty regular. Computer systems are designed to optimize that kind of access. In graph systems, systems described by graphs, generally those connections aren't so regular. So when you are computing analytics on a graph, you want to traverse from node to node. It is unpredictable where you are going to go next in the graph.

Suzanne: So parts of the graph may not get very much access, and other parts of the graph may get a lot of intense access depending on the problem you're trying to solve. So, that's very irregular.

Scott: Right. That's one aspect of it, but it's more like, *If I'm at a node, the computer can't predict where I'm going to go next. It's not the next location in memory. It could be …*

Suzanne Miller: It's not sequential.

Scott: Yes. It's not sequential. It could go anywhere in the memory system, so it's very hard for a computer system to optimize access like that. Again in the scientific community, you have benchmarks like the Top 500 that measure the performance of scientific codes. They measure that performance in what is called FLOPS. That is floating-point operations per second. In the graph community, the benchmark is Graph 500, and so they measure performance in TEPS, which is traversed edges per second. So going from one node to another in that data structure is one edge traversed.

Suzanne: So this means that parallelism, parallel computing, gives some advantage because you've got multiple ways of accessing memory at any one time. You have got multiple addresses, essentially, that you can access. So when you make that decision to traverse, you have got more options and more performance available to you than when you are in a single processing system, correct?

Scott: That is one part of it. In a supercomputer, you generally have lots of compute nodes, each with the dedicated memory bank. The graph analytics problem comes down to dividing the data across these nodes. It doesn't prevent traversal from one node to the next.

So there's a lot of work done in graph analytics to organize what data is where; to minimize not only traversal to other parts of the computer, but making sure that traversal ends up being in sequential memory locations as much as possible…

Suzanne: To improve the efficiency and some of the predictability…

Scott: …of the existing, general-purpose memory architectures.

Suzanne: In late 2013, leading researchers in the graph analytics community released a position paper calling for an open effort known as GraphBLAS. BLAS in the scientific community is basic linear algebra subprograms. So trying to do something similar for the graph community that has already been done in the scientific community. And that is to define a standard for graph algorithms in terms of linear algebraic equations. Tell us a little bit about this position paper and how it figures into the SEI's ongoing research into graph analytics.

Eric: In the scientific computing community for over 30 years, basically the community has fallen in on a standard set of APIs and a standard interface to doing certain kinds of computations, and that's BLAS, the basic linear algebra subprograms. What this is, is effectively matrix algebra. What this does is it creates a line of separation between the application programmers and the people who are implementing the matrix math on either standard CPUs or other kinds of special purpose hardware. So it really creates an abstraction layer where application programmers can think about their application stuff, and the implementers of the high-performance computing can target it to their hardware architecture.

The graph analytics community realized that this might be applicable to their space as well. Since graphs can also be represented as matrices, and operations on graphs can be translated into operations on matrices, the idea is, *Can we do a similar thing for graph analytics?*

Suzanne: So where are you with this?

Scott: This came out in the middle of the research. We are collaborating with one of those leading researchers, Andrew Lumsdaine at Indiana University. We decided to tackle the problem, evaluate whether or not GraphBLAS can do what we want it to do. We have a set or a list of 10 analytics that we'd like to try and do on graphs.

Suzanne Miller: So they're like your model problems that you're applying your ideas to?

Scott: The basic one is breadth first search. So, can you traverse the graph and find all the shortest paths from one node to everything else? It goes on to more complicated things called community detection, something pretty standard. There is 30 years of expertise in fine tuning basic linear algebra subprograms for the regular access that is found in scientific communities.

Suzanne: How is this standards effort affecting your research? Or how has it affected your research?

Eric: After this position paper we pivoted our research. So what we were doing initially is we were creating a graph analytic library. We were creating a software library, and we were kind of doing it as we knew, as we could implement it on GPUs.

What happened was in the middle of this research, all of these people in the graph analytics community said *Hey, all you different people are doing this in different ways. Some of you are doing it in an object-oriented type of way. Some of you are doing it in a traversal kind of way. Some of you are doing it in a matrix algebra kind of way. The scientific community has had great success in implementing this in terms of matrix algebra. Why don't we all fall in on this way of doing it*? *That way the hardware people can implement and optimize the matrix algebra libraries, and the software people can implement their graph analytics*.

We actually pivoted our research and said, *Hey, this is a great idea. Let's do that.* Since we were partners with Andrew Lumsdaine of Indiana University, it made great sense to do it. Then there is the separation of concerns.

Scott: Most of the 19 leading researchers on that position paper are focusing on general-purpose, CPU architectures, so regular memory architectures. But, at most of the government labs and the large supercomputing facilities, there is a shift from general CPU architectures to all these different kinds of accelerators.

So, you've heard of the large Cray systems that have GPUs as coprocessors. There's also the Xeon Phi that's come out. This separation of concerns between the application and analytics developer and the underlying hardware becomes even more important when you're talking about these vastly different memory architectures, even computing architectures.

We're are focusing our evaluation of GraphBLAS on GPU architectures. *So what does it take when you have an algorithm or an analytic implemented in GraphBLAS? What does it take underneath the line of separation to organize the data in a way that a GPU can use*?

GPUs are even more sensitive to poor memory layout of data, so there is more focus on what it takes to actually get the data organized so that the GPU can do it efficiently. There is a tradeoff there: The more work you do to organize the data, the more time *that setup* takes, but *then the algorithms* take a shorter amount of time.

Suzanne: So you are really trying to do feasibility analysis for different strategies associated with memory layout and other aspects of memory access….

Scott: Yes.

Suzanne: …so that you can help this community understand what are the options, and what are some of the solutions that might be better to try than others because you found out through this feasibility analysis that some things don't work so well.

Scott: That's right.

Suzanne: Cool.

Eric: As we like to do, the feasibility analysis is done through hands-on prototyping and technology demonstrations.

Suzanne: Excellent.

Scott: There is another aspect to that problem. There is an existence analysis that goes along with it. We come in with our stakeholders' needs. We have 10 or so analytics that we want to do. We are trying to find out if you can implement them easily in terms of the BLAS interface.

Suzanne: Is the standard really going to work?

Scott: Right. So we are doing analysis of the appropriateness of GraphBLAS.

Suzanne: Excellent. That is one of the things we do here at the SEI is help the community understand their options.

OK. So where are you with this research? What do you have coming up next? You said you started with the one analytic, the traversal. Have you progressed with the other 10, or are you in the middle of that? Where are you?

Scott: All right, so we are collaborating with Indiana University, Andrew Lumsdaine's group at the Center for Research and Extreme Scale Technologies [CREST]. So, they have a large supercomputer with GPU nodes on it. We have a few of the newer generation GPUs here in our cluster.

We've actually split the work. Here at the SEI, we are doing a top-down approach. We're looking at the feasibility or the existence of analytics, implemented in terms of GraphBLAS. We've started by prototyping these things in Python.

At the same time, Andrew Lumsdaine's group is looking at the low-level, just bare bones performance of primitives that would implement below the separation concerns on GPUs.

We have implemented in Python, all 10 algorithms to greater or lesser extent in terms of GraphBLAS functions. We are using Python's CUDA[Compute Unified Device Architecture] interface, which is the GPU programming language to then transition those high-level implementations into basic implementation on GPUs. From there we're going to try and meet in the middle with Indiana University and see what needs to be done to that CUDA interface

Suzanne: Sounds like that's going to be a fun meeting.

Scott: That happens in June.

Eric: We also intend to release the graph analytics library as open-source software to the public.

Suzanne: Excellent. That has its own challenges.

All right. So once you get done with this feasibility analysis, what do you see as coming next for this kind of work, and you have released the open-source library that you're talking about? What do you think comes next?

Eric: I'll start.

Scott: OK, you start.

Eric: We are in the third year of a three-year research project on doing graph analytics on GPUs. This is the final year of the graph-analytics-on-GPUs research. And, we are going to release that open-source library.

In terms of the research we're going to do in this area, it's going to be more focused on applications in terms of data analytics, big data analysis, and measuring and benchmarking big data analysis.

Suzanne Miller: OK. Anything to add, Scott?

Scott: No, I think you covered it.

Suzanne Miller: Excellent. I want to thank both of you for joining us today. For those who want a deeper dive into this work, we welcome you to visit the SEI Library where you can download a copy of their technical note on this topic. Just go to resources.sei.cmu.edu and in the keyword field type the name of their report, which is *Patterns and Practices for Future Architectures*. Or, you can search on the author function.

Eric [and Scott] also co-authored a blog post on this research. Please visit blog.sei.cmu.edu and search on EricWerner [or Scott McMillan] in the author field.

This podcast is available on the SEI website at sei.cmu.edu/podcasts and on Carnegie Mellon University's iTunes U site. As always, if you have any questions, please don't hesitate to email us at info@sei.cmu.edu. Thank you for listening.