

HPC User Forum Update

Interviews with Steering Committee Members: Paul Muzio, Committee Chair

Steve Conway and Thomas Gerard
June 2020

IN THIS UPDATE

After the global pandemic forced Hyperion Research to cancel the April 2020 HPC User Forum planned for Princeton, New Jersey, we decided to reach out to the HPC community in another way—by publishing a series of interviews with members of the HPC User Forum Steering Committee. Our hope is that these seasoned leaders' perspectives on HPC's past, present and future will be interesting and beneficial to others. To conduct the interviews, Hyperion Research engaged Rich Brueckner (1962-2020), president of insideHPC Media. We welcome comments and questions addressed to Steve Conway, sconway@hyperionres.com or Earl Joseph, ejoseph@hyperionres.com.

This interview is with Paul Muzio, the present chair of the HPC User Forum Steering Committee. Most recently, Muzio was Director of the City University of New York Interdisciplinary High Performance Computing Center. In this position, he was responsible for the strategic development and management of the center, which is one of the largest academic HPC facilities in the New York City area. Prior to joining the City University, he was vice-president, HPC Programs, at Network Computing Services, Inc. (Minnesota Supercomputing Center, Inc.). From 1980 to 1990, he was Director, Special Systems, at Grumman Aerospace Corporation in Bethpage, NY. Muzio has been involved in acquiring approximately 30 HPC systems, beginning with a Cray-1M at Grumman in 1982.

The HPC User Forum was established in 1999 to promote the health of the global HPC industry and address issues of common concern to users. More than 75 HPC User Forum meetings have been held in the Americas, Europe and the Asia-Pacific region since the organization's founding in 2000.

PAUL MUZIO INTERVIEWED BY RICH BRUECKNER, insideHPC

Brueckner: Welcome, Paul.

Muzio: Great to be here.

Brueckner: Why don't we start at the beginning? How did you get involved with HPC?

Muzio: There was no such thing as HPC when I got involved. I was very fortunate to have a number of great high school teachers at Grover Cleveland High School in New York City. This was back in the Spring of 1960, when I was a junior. One of those faculty members called me into his office and said, "Columbia University is having this science honors program, would you be interested?" Of course. So, I went up to Columbia University on a Saturday morning to take an exam, and a while later I got a letter saying I was accepted. It was really a great program. It was funded by the National Science Foundation. It included lectures by outstanding faculty, including I.I. Rabi [Nobel Laureate in physics], and lunch with faculty in the faculty dining room. I started the program in the fall of 1960.

I was provided access to an IBM 650 computer but was given no instructions. They said, "here's the computer, here's the keypunch machine, here are the manuals." So, I had to learn machine language to program this machine, which was basically just a smart accounting machine with 4,000 words of drum memory - it was a decimal-based computer. The instructions consisted of a two-digit operation code and a four-digit data address and a four-digit address of the next instruction. If you wanted to add two numbers you needed to numerically specify where the numbers started, where the numbers ended. This was all very basic and fundamental, more like a fancy, programmable adding machine. But you learned the most basic level of how computers really work.

Except for a one-credit course in college, this was the only formal computer science course I ever had and it has served me well for 60 years. Why? Because it taught me the basics and it provided me with a foundation for understanding both the limitations and potential for future developments of computing hardware. Unfortunately, I don't think the basics of hardware and machine instructions are taught anymore and I think that is a mistake because there is a huge correlation between computing hardware and the applications that can be mapped to the hardware. Conversely, what algorithms and applications we want to solve should drive the hardware design.

After graduate school I went to work as an operations research analyst and I had to write software to conduct simulations and do statistical analyses. I mostly used Fortran but on occasion I also learned to write COBOL programs. I used a number of different systems, including IBM 360s and 370s, UNIVAC 1108s and CDC 6600s. Then, over time, I moved into other positions at other companies and into management roles regarding technical computing and managing computing system resources and operations. Not so much fun, more bureaucracy and less hands-on work.

Brueckner: When did you get into the simulation, the science and the research?

Muzio: After doing the research and the simulations and the statistical analyses, I went to work for Grumman Aerospace Corporation. There I was in more of a management role and I managed a number of systems for different projects, both the acquisition, installation, and operation of the systems. In the procurement of the systems I was an advocate, I guess it was in 1982 when Grumman acquired its first supercomputer, a Cray 1M. I was doing that kind of stuff up until 1990. In 1990 I moved to Minnesota, where I worked for the Minnesota Supercomputing Center and was the

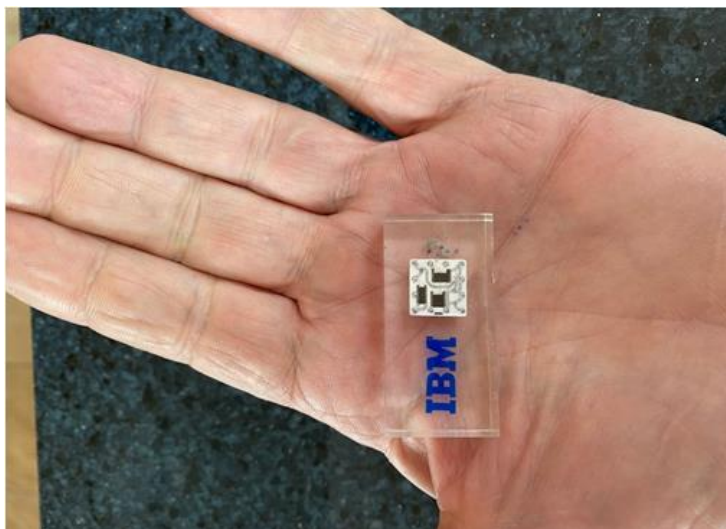
infrastructure director for the Army High-Performance Computing Research Center. There we had a number of very interesting machines, including a Thinking Machine's CM-200, Thinking Machine's CM-5, and then a Cray T3E, and finally a Cray X1. The T3E and the Cray X1 were very interesting machines. The T3E had global address space programming models incorporated in it, in hardware. And we'll get on to that a little bit later in terms of future technologies. And the Cray X1, of course, was a scalar/vector machine with global address space programming models. I thought it was a great machine. It had two problems: scalar performance was not good and it was very expensive.

So, after that, I was director of high-performance computing at the City University of New York. And there I acquired and installed a number of systems, ranging from traditional Dell clusters to an SGI system with 12 terabytes of memory and a lot of GPUs, and other systems that were pretty cutting-edge technology at the time. I've been retired now for three years and enjoying the beach, but still keeping my fingers in high-performance computing.

Brueckner: Paul, you've certainly seen a lot of change in your full career, but what are the changes that really struck you in HPC?

Muzio: In certain respects, nothing has changed over 30 years. In other respects, lots of things have changed. So, I'm going to cover what was and what is currently happening and discuss trajectories. So, there's been advances in computing made by architectural developments and transistor and chip technology improvements. In the 1960s to 1980s, architectural advances such as pipelining and vector processing resulted in significant performance improvements for scientific computing on specialized computers as compared to more general-purpose computers. But these specialized machines were to lose out to microprocessor-based chip-based technology. The improvements in chip technology have been revolutionary and astounding. Somewhere in my files, I have a chip from 1964 from an IBM 360 with about four transistors on it. Now we see chips with up to 40,000,000,000 MOSFETs. These improvements in chip technology have resulted in a million times improvement in systems performance, coupled with a reduction in price per computation by a factor of at least a billion. Figure 1

IBM 360 Solid Logic Technology (SLT) Chip



Source: IBM Corporation

In my opinion, the advances in chip technology overshadowed, or derailed for a while, developments in computer architecture for two coupled reasons – the high cost of building a microprocessor chip fabrication facility required amortization through the sales of millions of chips, not the limited subset required by the scientific computing community; and the extraordinary performance gains in chip technology obviated the need for specialized architectures.

Today, we are seeing a shift back to building processors with specialized architectures. This is a result of a flattening performance improvement in chips, particularly related to the inability to increase processor frequency speeds and also the creation of open-fab facilities such as TSMC, which has changed the economic picture. Now, in a way, it's back to the future: we have the proliferation of processors chips connected to GPUs, which are vector processors. It reminds me of 1985's Digital Equipment Corporation VAX-11/780 coupled to a Floating Point Systems array processor.

By the way, that 2020 solution which is much, much faster and cheaper still suffers from the same architectural bottleneck as that 1985 configuration: a poor interconnect compared to the processor performance. I expect, however, that we are seeing that problem going away with more advanced microprocessors which will have integrated scalar/vector processors with equal addressability to memory, or equal access to memory as in systems like the Cray 1. By the way, along those lines, it was fun for me, a few years back, to go to classes on programming and optimizing codes for Nvidia GPUs or Intel Phis. It's no different from classes for programming the Cray 1 or FPS [Floating Point Systems] boxes, so it made me feel 30 years younger. Also, along those same lines, now we are having Micron and Intel offering SSD memory. This is no different than extended core memory on CDC 6600s or SSDs on Cray systems. And it's word addressable, which I think is kind of important.

Another architectural feature of yesteryear that I hope to see come back is hardware support for global address space such as was implemented on the Cray T3E. Writing programs using UPC or Coarray Fortran was so much easier compared to using MPI and had better performance on machines that had hardware support for global address space. So, I think that's a feature that we'll see more of in the future because I think it's very important for AI and machine learning.

Brueckner: Paul, I'm supposed to ask you about where HPC is headed in the future. Where do you think it's going to go from what you're seeing today?

Muzio: Well, I think we're going to see more custom chips and chips that are better optimized for machine learning and AI as well as high-performance computing. I will not discuss quantum computers because they're beyond my brain to understand. But, you know, I was at a presentation a while back on machine learning and one of my colleagues commented to me, "Well, you know, this is trivial stuff. You have to show a million pictures for it to recognize a horse. A child can pick that up right away." I said that's not really quite true, a child has seen millions and millions and millions of pictures. And one of the differences between machine learning with a computer and machine learning with biological systems is context. So, you don't see a picture of just a horse, you see a picture of a horse in context. You see a picture of a horse relative to a dog, a cat, so forth. So, it's not just understanding the image, it's understanding the concept and the context and the relationships.

You know, there's a lot of research these days going on in how the brain functions. John O'Keefe and May-Britt Moser and Edvard Moser won the 2014 Nobel Prize in physiology and medicine for their work in discovering the role on place cells in the brain. Their function is geographic mapping. For example, one of the things they were able to show was taxi cab drivers in London who go through a

rigorous training program on maps have much larger developed areas of place cells in the brain. So, the brain adapts, it's a muscle and it grows and it learns.

But the place cells are related to relationships. Getting back to interconnects and global address space models, the issue there is not just machine learning of single things but linking these things. Artificial intelligence will require linking lots of images and building relationships and graphs between them and that's where I think the need for improved networks comes into play: interconnect networks, and global address space programming models and the hardware to support those models.

Brueckner: To kind of wrap up here, what are the trends that have you excited these days and which ones have you concerned?

Muzio: Well, the ones that have me concerned are exactly machine learning and AI. A great picture came out in 1956, *Forbidden Planet*. The movie was inspired by people in computer science and people in thermonuclear fusion. It revolved around a planet where the inhabitants had built a machine to run the entire planet and also where they could transfer their intelligence to it and their search was for immortality. They would transfer their intelligence to this machine and live on in that machine forever.

Well, I often relate that to some of the systems that we're building today. Google has a vast repository of knowledge and we are transferring evermore information into it. It's getting more capable. I'm shocked, over the last ten years, to see how much it's improved, and I expect it will continue to improve. And as hardware architectures advance, it will advance. Are we building that global machine? You know, when we die, our brain is dead, our offspring have to go to college and learn all over again. Google doesn't have to. That knowledge stays there. That information stays there.

The other thing I'm concerned about is, I've worked with many computer science students, and I would go in and ask them – I might have 30 or 40 at a time – I'd ask them, "you go to Google Maps and you say, 'I want to go from point a to point b, how long will it take?' Google Maps tells me two hours and 17 minutes and gives me the route." I would ask the students, "How does Google Maps know that?" Well, my experience is that maybe one in 100 students knew the answer. If some volunteered and answered, they would generally say GPS. It's not GPS. GPS shows you the start point and the end point but the time calculation is done by people's cell phones moving around, even if they're not active and communicating with cell towers. Where is the level of inquisitiveness among these computer science students to find out what's going on underneath? That concerns me.

Let's get back to the taxicab drivers; the brain is a muscle. The place map cells increased in number because they had to learn maps. We don't have to learn maps anymore; we just look it up on Google. So, on one hand we are transferring knowledge into this artificial machine and we are losing the capability to think ourselves. That concerns me.

Brueckner: Do you have anything else you want to add before we close it out?

Muzio: I just wish everyone stays healthy and happy. Thank you.

Brueckner: Thank you, you too.

About Hyperion Research, LLC

Hyperion Research provides data-driven research, analysis and recommendations for technologies, applications, and markets in high performance computing and emerging technology areas to help organizations worldwide make effective decisions and seize growth opportunities. Research includes market sizing and forecasting, share tracking, segmentation, technology and related trend analysis, and both user & vendor analysis for multi-user technical server technology used for HPC and HPDA (high performance data analysis). We provide thought leadership and practical guidance for users, vendors and other members of the HPC community by focusing on key market and technology trends across government, industry, commerce, and academia.

Headquarters

365 Summit Avenue

St. Paul, MN 55102

USA

612.812.5798

www.HyperionResearch.com and www.hpcuserforum.com

Copyright Notice

Copyright 2020 Hyperion Research LLC. Reproduction is forbidden unless authorized. All rights reserved. Visit www.HyperionResearch.com or www.hpcuserforum.com to learn more. Please contact 612.812.5798 and/or email info@hyperionres.com for information on reprints, additional copies, web rights, or quoting permission.