

Bolstering Extreme Scale Computational Biology

According to Dr. Klaus Schulten from the University of Illinois, the molecular dynamics and visualization programs NAMD and VMD, which serve over 300,000 registered users in many fields of biology and medicine, are pushing the limits of extreme scale computational biology. Schulten says these programs can operate on a wide variety of hardware and offer new inroads to medical discovery.



In addition to outlining NAMD and VMD on ARM and GPU developments over the last several years that led to the programs' extreme performance on Blue Waters, Titan and Stampede, Schulten is known for shedding light on how these fields and programs are enabled by petascale computing. The following Q and A highlights some key features of his research.

HPCwire: Can you describe the growth of NAMD and VMD and give us a sense of how these developments have helped computational biology evolve?

Klaus Schulten: NAMD and VMD are programs that permit you to simulate very large biomolecules and effectively taking on the role of a computational microscope—you simulate these molecules and thereby you visualize them. You know their properties from chemistry and biochemistry; you know their structures from biology. Then, just like you simulate a Boeing before you actually build it, you simulate a molecule in the computer to optimize it.



The difference between us and others building similar programs is that we designed the program for parallel computers and for modern software and computer science concepts from the get-go. That meant designing software that went on clusters and then later on parallel computers built around clusters.

That was all until about 2006 when the National Science Foundation decided to invest in a large computer that was a hundred times larger than could be foreseen otherwise, called the petascale computer. We wanted to take advantage of this huge power increase—but not just because we wanted to be 100 times more powerful in what we could simulate, but rather we realized all along that all of our simulations were too small, meaning that a living cell is made of millions and millions of molecules that form associations that cooperate, and we needed to understand how these proteins worked together rather than worked by themselves.

With this big computer we wanted to explore how the molecules of life associate into structures and then cooperate, and this is exactly what we achieved. We solved the structure of the HIV virus, which made of way over 1,000 proteins that form a capsid, and we can now describe it atom-by-atom. Without petascale computing that would have been impossible.

Achieving this meant using the computer in two ways. On one side, we made the computer part of the experiment, literally. When you want to see a virus for a traditional experiment, you must have the physical virus on-hand. But just as Boeing can simulate an airplane, and we can simulate many molecules that you find in living cells, bypassing the physical study and making the computer an integral part of the experiment itself.

So we got data from different kinds of experiments via sources such as crystallography and electron microscopy, and then integrated them into one picture of the virus that gave us a view of the virus at the level of the atom. We could then test it in the second step. Finally we could take this model and simulate it in the computer, carrying out the world's largest simulation ever done I think even to this point.

At this point we have reached our goal—we could show that the structure is stable to simulate in the computer and could look at its physical properties—but now of course comes the question of what we learned from it.

First we resolved structure atom-by-atom because we wanted to make the container of the virus, the so-called “capsid,” a target of drug treatment. That requires that we know the chemistry of that target, because when you deal with drugs that are molecules, you need to know both sides of direct treatment in chemical detail: you need to know the drug, of course, (very small molecules that are pretty straightforward,) and you need to know the target, which in this case is a huge system of over 1,000 proteins, and each protein itself in a big molecule containing itself several tens of thousands of atoms.

Once we applied drugs to our computational virus then we learned that the drugs most likely work very differently than we thought—we found that the HIV virus is in a way more dangerous and intelligent than we thought.

HIV is like a con artist that that smuggles itself into the cell then persuades the cell to help the virus infect it. Otherwise it's not at all easy to infect a cell: the virus has to put its own genetic material into the nucleus of the cell, where the living cell has its genetic material, which is so difficult because the nucleus is very protected and very well organized against this kind of intrusion. But the virus talks a cell into helping it to get its genes into the nucleus.

And it is this cooperation that is acted upon by antiviral drugs.

And so now we have the stage of this drama: on the enormous surface of the virus, which is made up of over a thousand proteins, the virus recruits proteins from the infected cell to help it in its vicious strategy to get the virus' genes inside. That is where antiviral drugs apparently interfere with this coordination with the cell.

HPCwire: ARM and GPU use are central to your research. Can you speak to that angle?

Klaus Schulten: We are one of the technology centers funded by the National Institutes of Health. We're called the Center for Macromolecular Modeling and Bioinformatics and we've received funding for 23 years now and we will receive funding for five more years. The task of the center is to make the absolute best computing technology available to biomedical researchers in the United States.

And our task, since we have shown that we through our research that we can use the modern computing technology (particularly parallel computing) extremely well, is that we not only use it for our own research but that we also make it available for others. Our goal is to be as good or better that the physicists in using computing technology for the benefit of our particular scientific community, which in our case is biomedical research.

I think we are doing very, very well because our software runs extremely effectively on the biggest computers in the world. But it was the same software from the laptop to these big computers, so the individual researcher can learn it on his or her laptop and use it all the way to the big machine. In the same way, if we develop for the big machine it trickles down quickly back to the small machine.

Our task is now to utilize this technology constantly. So from 2006 to 2012 we have been working on making petascale computing possible. We focused on making these programs capable of simulating very, very large systems, hundreds of times larger than before, and also on analyzing and visualizing the results. This meant working on two fronts: on the front of the actual simulation that's done by the program NAMD, and on the front of the visualization.

Now this is old. Of the new technologies going on, we think that the upcoming technology for the next generations of computers will be ARM chips, which we've been very successful in integrating.

But one factor that has never before been so important is the use of power. Now we not only adopt the new generation of chips for our software perhaps two, three years before the first time any scientists outside of our own group will use it, but we have to power profile all of our algorithms and all of our computational strategies.

Before the only thing that counted was how fast we compute. Now the talk is of scalability and making bigger models that effectively make use of bigger machines, and the talk in the lab is constantly about power profiling. Where can we cut corners in power consumption? What new computational strategies should we adopt? So the issue of power consumption is coming into our development work.

HPCwire: So there really aren't a lot of on-base supercomputers at all, so where and how are you testing these ideas?

Klaus Schulten: First, are the smartphones and tablets. Our priority is to support the software that puts demands on machine so that through a smartphone or tablet you have another input device, an extra monitor or extra output device. But we're already now well on our way to have the entire programs run on tablets and smartphones.

That went through pretty well, but the main problem is that you have a very, very small monitor, so you must develop a new user interface and that takes time and created a bottleneck for when we can release our software on these devices.

The main point, however, is that computers will be built from these kinds of chips that people expect, and then these kinds of chips will be made available in a form that you can put them in for other interactions and use them for computing. For that moment we will also be ready.

We've learned that these are very intelligent chips that can handle power issues in a much more flexible way, which enabled us to add dimension to our computational strategy that we never had before: a totally new culture to prepare us for the next generation of computing.

HPCwire: Let's talk briefly about what GPUs have lended to computational biology in general.

Klaus Schulten: They were a tremendous benefit because they go two directions. The first is they make very powerful computing possible in the lab for a much less money. It's very cost-effective computing, and very powerful computing. So the kind of calculations that until just about two, three years ago required a \$50,000 computer can be done now with a few-thousand-dollar GPU cluster or even a single GPU board.

In the other direction, many smaller calculations are being made possible through GPUs. We were very early in demonstrating this with our first GPU extended paper, but today many labs work on it very well.

So that is the poor man's powerful computer, which has been essential in proliferating the methodology and the culture of computing within the biomedical community.

Finally we come to accelerators, which is where Cray has played such a large role, particularly in boosting the speed of Titan and Blue Waters by above a factor of two. So too we can often be better—we are still battling to get more power out of the GPUs. But the effect is that what we expected at first to gain from these computers is doubled or now tripled.

And that of course is when the power of the computer is delineating the scientific frontier. When all of the sudden you can go twice as far, reaching twice as fast into new territory, that's a huge scientific advancement. That's what GPUs made possible.