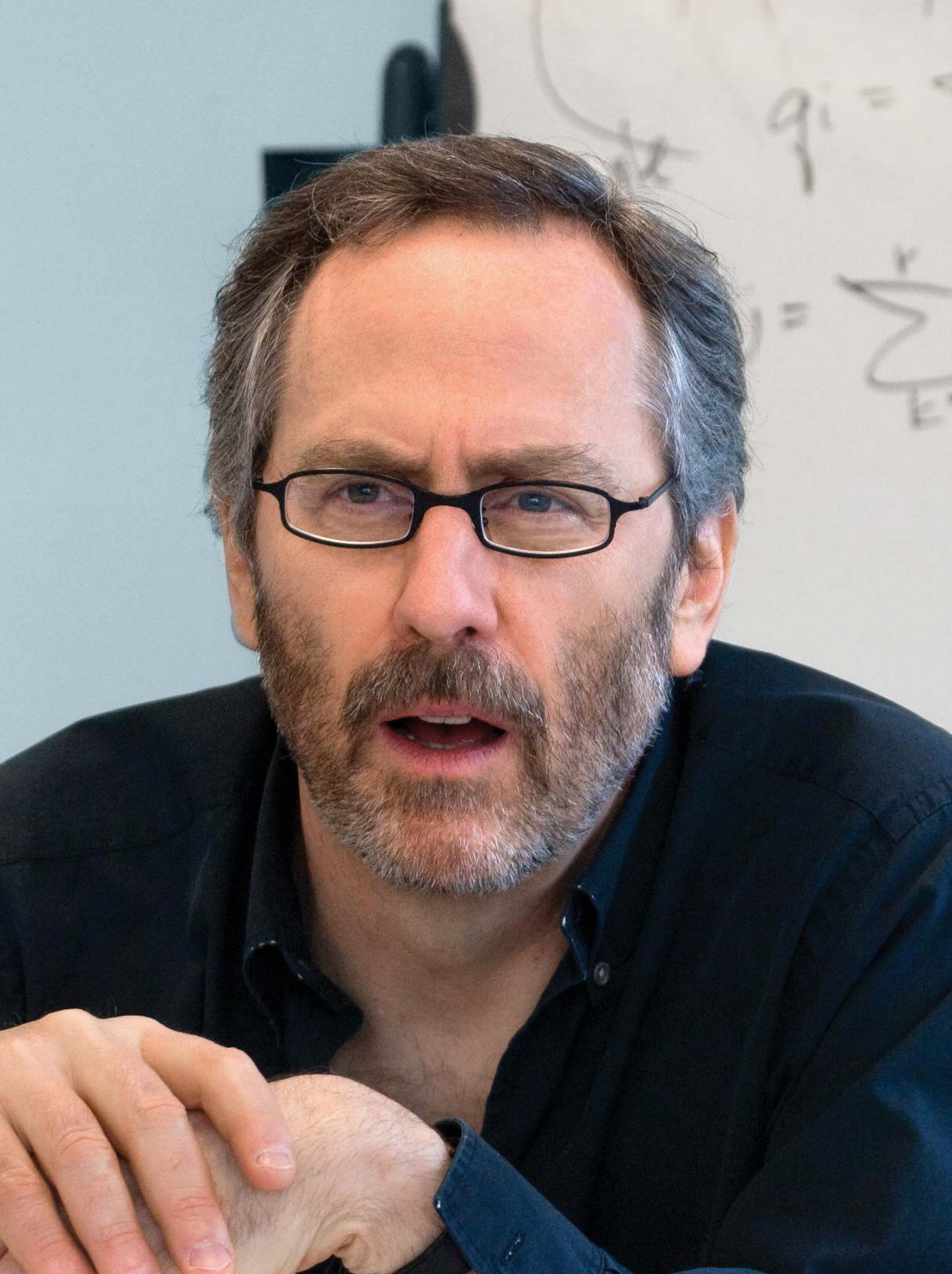
qi =

PHOTOGRAPH COURTESY OF DAVID E. SHAW

DOI:10.1145/1562764.1562782

Article development led by acmqueue
queue.acm.org

Stanford professor Pat Hanrahan sits down with the noted hedge fund founder, computational biochemist, and (above all) computer scientist.

A Conversation with David E. Shaw

DAVID SHAW CONSIDERS himself first and foremost a computer scientist. It's a fact that's sometimes overshadowed by the activities of his two highly successful, yet very different, ventures: the hedge fund D. E. Shaw & Co., which he founded 20 years ago, and the research lab D. E. Shaw Research, where he now conducts hands-on research in the field of computational biochemistry. The former makes money through rigorous quantitative and qualitative investment techniques, while the latter spends money simulating complex biochemical processes. But a key element to both organizations' success has been Shaw's background in computer science. Serving as interviewer, computer graphics researcher and Stanford professor Pat Hanrahan[a] points out that one of Shaw's unique gifts is his ability to effectively apply computer science techniques to diverse problem domains.

In this interview, Hanrahan and Shaw discuss Shaw's latest project at D. E. Shaw Research—Anton, a special-purpose supercomputer designed to speed up molecular dynamics simulations by several orders of magnitude. Four 512-processor machines are now active and already helping scientists to understand how proteins interact with each other and with other molecules at an atomic level of detail. Shaw's hope is that these "molecular microscopes" will help unravel some biochemical mysteries that could lead to the development of more effective drugs for cancer and other diseases. If his track record is any indication, the world has a lot to be hopeful for.

PAT HANRAHAN: What led you to form D. E. Shaw Research?

DAVID SHAW: Before starting the lab, I'd spent a number of years in the fi-

[a] http://graphics.stanford.edu/~hanrahan/

nancial world applying quantitative and computational techniques to the process of investment management. During the early years of D. E. Shaw & Co., the financial firm, I'd been personally involved in research aimed at understanding various financial markets and phenomena from a mathematical viewpoint. But as the years went by and the company grew, I had to spend more time on general management, and I could feel myself getting stupider with each passing year. I didn't like that, so I started solving little theoretical problems at night just for fun—things I could tackle on my own, since I no longer had a research group like I did when I was on the faculty at Columbia. As time went by, I realized that I was enjoying that more and more, and that I missed doing research on a full-time basis.

I had a friend, Rich Friesner, who was a chemistry professor at Columbia, and he was working on problems like protein folding and protein dynamics, among other things. Rich is a computational chemist, so a lot of what he did involved algorithms, but his training was in chemistry rather than computer science, so when we got together socially, we often talked about some of the intersections between our fields. He'd say, "You know, we have this problem: the inner loop in this code does such-and-such, and we can't do the studies we want to do because it's too slow. Do you have any ideas?"

Although I didn't understand much at that point about the chemistry and biology involved, I'd sometimes take the problem home, work on it a little bit, and try to come up with a solution that would speed up his code. In some cases, the problem would turn out to be something that any computer scientist with a decent background in algorithms would have been able to solve. After thinking about it for a little while, you'd say, "Oh, that's really just a special case of this known problem, with the following wrinkle."

One time I managed to speed up the whole computation by about a factor of 100, which was very satisfying to me. It didn't require any brilliant insight; it was just a matter of bringing a bit of computer science into a research area where there hadn't yet been all that much of it.

At a certain point, when I was approaching my 50th birthday, I felt like it was a natural time to think about what I wanted to do over the coming years. Since my graduate work at Stanford and my research at Columbia were focused in part on parallel architectures and algorithms, one of the things I spent some time thinking about was whether there might be some way to apply those sorts of technologies to one of the areas Rich had been teaching me about. After a fair amount of reading and talking to people, I found one application—the simulation of molecular dynamics—where it seemed like a massive increase in speed could, in principle, have a major impact on our understanding of biological processes at the molecular level.

It wasn't immediately clear to me whether it would actually be possible to get that sort of speedup, but it smelled like the sort of problem where there might be some nonobvious way to make it happen. The time seemed ripe from a technological viewpoint, and I just couldn't resist the impulse to see if it could be done. At that point, I started working seriously on the problem, and I found that I loved being involved again in hands-on research. That was eight years ago, and I still feel the same way.

HANRAHAN: In terms of your goals at D. E. Shaw Research, are they particularly oriented toward computational chemistry or is there a broader mission?

SHAW: The problems I'm most interested in have a biochemical or biophysical focus. There are lots of other aspects of computational chemistry that are interesting and important—nanostructures and materials science and things like that—but the applications that really drive me are biological, especially things that might lead not only to fundamental insights into molecular biological processes, but also to tools that someone might use at some point to develop lifesaving drugs more effectively.

Our particular focus is on the structure and function of biological molecules at an atomic level of detail, and not so much at the level of systems biology, where you try to identify and understand networks of interacting proteins, figure out how genetic variations affect an individual's susceptibility to various human diseases, and so forth. There are a lot of computer scientists working in the area that's commonly referred to as bioinformatics, but not nearly as many who work on problems involving the three-dimensional structures and structural changes that underlie the physical behavior of individual biological molecules. I think there's still a lot of juicy, low-hanging fruit in this area, and maybe even some important unifying principles that haven't yet been discovered.

HANRAHAN: When you mention drug discovery, do you see certain applications like that in the near-term or are you mostly trying to do pure research at this point?

SHAW: Although my long-term hope is that at least some of the things we discover might someday play a role in curing people, that's not something I expect to happen overnight. Important work is being done at all stages in the drug development pipeline, but our own focus is on basic scientific research with a relatively long time horizon, but a large potential payoff. To put this in perspective, many of the medications we use today were discovered more or less by accident, or through a brute-force process that's not based on a detailed understanding of what's going on at the molecular level. In many areas, these approaches seem to be running out of steam, which is leading researchers to focus more on targeting drugs toward specific proteins and other biological macromolecules based on an atomic-level understanding of the structure and behavior of those targets.

The techniques and technologies we've been working on are providing new tools for understanding the biology and chemistry of pharmaceutically relevant molecular systems. Although developing a new drug can take as long as 15 years, our scientific progress is occurring over a much shorter timescale, and we're already discovering things that we hope might someday be useful in the process of drug design. But I also enjoy being involved in the unraveling of biological mysteries, some of which have puzzled researchers for 40 or 50 years.

HANRAHAN: This machine you've built, Anton, is now operational. Can you tell us a little bit about that machine and the key ideas behind it?

SHAW: Anton was designed to run molecular dynamics (MD) simulations a couple orders of magnitude faster than the world's fastest supercomputers. MD simulations are conceptually pretty simple. In biological applications like the ones that interest us, the idea is to simulate the motions of large biomolecules, such as proteins or DNA, at an atomic level of detail over a simulated time period during which biologically interesting things happen in real life. The trajectories followed by these molecules are typically calculated by numerically integrating Newton's laws of motion, using a model based on classical mechanics that approximates the effects of the various types of forces that act on the atoms. During each "time step" in the integration process, the atoms move a very short distance, after which the forces are recomputed, the atoms are moved again, and so on.

HANRAHAN: And you're doing this on a femtosecond ($10^{\wedge -15}$ seconds) time scale, correct?

SHAW: That's right. Each of those time steps can only cover something on the order of a couple femtoseconds of biological time, which means it takes a really long time to observe anything of biological interest.

HANRAHAN: If you were to do that simulation with a conventional computer, such as a normal core or dual or quad core, how much biological time could you simulate in a day?

SHAW: Depending on the number of cores and other characteristics of the processor chip, you'd expect performance on the order of a few nanoseconds per day, as measured using a standard molecular system called the Joint AMBER-CHARMM Benchmark System, which represents a particular protein surrounded by water.

HANRAHAN: And what's the comparable figure for Anton?

SHAW: Our latest benchmark measurement was 16,400 nanoseconds per day on a 512-node Anton configuration, so Anton would run three or four orders of magnitude faster, and roughly two orders of magnitude faster than the fastest that can be achieved under practical conditions on supercomputers or massively parallel clusters.

The techniques and technologies we've been working on are providing new tools for understanding the biology and chemistry of pharmaceutically relevant molecular systems.

HANRAHAN: When I read the Anton paper in *Communications*,[b] it reminded me a lot of what I worked on, which were graphics chips—just in the way you're choreographing communication and keeping everything busy and all these really important tricks. Can you summarize some of the key innovations in Anton that make it so fast?

SHAW: Each node of the Anton machine is based on a specialized ASIC that contains, among other things, 32 very long, arithmetically dense, nonprogrammable pipelines designed specifically to calculate certain forces between pairs of interacting particles. Those pipelines are the main source of Anton's speed. In addition, Anton uses an algorithm that I developed in conjunction with the machine's top-level architecture to reduce the amount of data that would have to pass from one Anton ASIC to another in the course of exchanging information about the current positions of the atoms and the forces that act on them. In essence, it's a technique for parallelizing the range-limited version of the classic *N*-body problem of physics, but the key thing for our purposes is that it's highly efficient within the range of parameter values we typically encounter in simulating biological systems.

HANRAHAN: Is this the midpoint algorithm or the NT algorithm?

SHAW: The one I'm referring to is the NT algorithm, which is what runs on Anton. The midpoint algorithm is used in Desmond, an MD code that was designed to run on ordinary commodity clusters. They're both examples of what I've referred to as "neutral territory" methods. The NT method has better asymptotic properties, but the constants tend to favor the midpoint method on a typical cluster and the NT method at a higher degree of parallelism.

HANRAHAN: When I first saw your NT algorithm I was stunned by it, just because people have been thinking about this problem for so long. Force calculations between particles are such an important part of computational science, and to have made the observation that

[b] "Anton, A Special-Purpose Machine for Molecular Dynamics Simulation" by D.E. Shaw et al. was published in the July 2008 issue of *Communications of the ACM*, 91–97.

the best way to compute the interactions of two particles involves sending them both somewhere else is just amazing. I understand your proof but it still boggles me because it seems so counterintuitive.

SHAW: It is kind of weird, but if you look back through the literature, you can find various pieces of the puzzle in a number of different contexts. Although I wasn't aware of this at the time, it turns out that the idea of "meeting on neutral territory" can be found in different forms in publications dating back as far as the early 1990s, although these early approaches didn't offer an asymptotic advantage over traditional spatial decomposition methods. Later, Marc Snir independently came up with an algorithm that achieved the same asymptotic properties as mine in a different way, and he described his method in a very nice paper that appeared shortly before my own. His paper included a proof that this is the best you can do from an asymptotic viewpoint. Although the constant factors were such that his method wouldn't have performed well in the sorts of applications we're interested in, it's clear with the benefit of hindsight that the straightforward addition of certain features from the NT method would have made his algorithm work nearly as well as NT itself for that class of applications.

But the important thing from my perspective was not that the NT algorithm had certain asymptotic advantages, but that with the right kind of machine architecture it removed a key bottleneck that would otherwise have kept me from exploiting the enormous amount of application-specific arithmetic horsepower I wanted to place on each chip.

HANRAHAN: I think it's a great example of computer science thinking because it's a combination of a new algorithm, which is a new way of organizing the computation, as well as new hardware. Some people think all advances in computing are due to software or hardware, but I think some of the most interesting ones are where those things coevolve in some sense.

SHAW: I agree completely. The history of special-purpose machines has been marked by more failures than successes, but I suspect that the creative codesign of new architectural and algorithmic approaches could have increased, at least to some extent, the number of applications in which a sufficient speedup was achieved to outweigh the very real economies of scale associated with the use of general-purpose commodity hardware. In our case, I don't think we could have reached our performance goals either by implementing standard computational chemistry algorithms on new hardware or by using standard high-performance architectures to run new algorithms.

Our latest benchmark measurement was 16,400 nanoseconds per day on a 512-node Anton configuration, so Anton would run three or four orders of magnitude faster, and roughly two orders of magnitude faster than the fastest that can be achieved under practical conditions on supercomputers or massively parallel clusters.

HANRAHAN: I think a lot of people in computer science were very surprised by your Anton paper because they might have thought that normal computing paradigms are efficient and that there's not a lot to be gained. You read these papers where people get 5% improvements, and then you sort of blow them out of the water with this 100-fold improvement. Do you think this is just a freak thing or are there chances that we could come up with other revolutionary new ways of solving problems?

SHAW: I've been asked that before, but I'm really not sure what the answer is. I'd be surprised if this turned out to be the *only* computationally demanding application that could be addressed using the general approach we've taken to the design of a special-purpose machine. On the other hand, there are a lot of scientific problems for which our approach would clearly *not* be effective. For one thing, some problems are subject to unavoidable communication bottlenecks that would dominate any speedup that might be achieved using an application-specific chip. And in some other applications, flexibility may be important enough that a cluster-based solution, or maybe one based on the use of GPUs, would be a better choice.

One of the things I found attractive about our application from the start was that although it was nowhere close to "embarrassingly parallel," it had several characteristics that seemed to beg for the use of special-purpose hardware. First, the inner loop that accounted for a substantial majority of the time required for a biologically oriented molecular dynamics simulation was highly regular, and could be

Anton, a massively parallel machine designed to perform molecular dynamics simulations.

PHOTOGRAPH COURTESY OF DAVID E. SHAW

mapped onto silicon in an extremely area- and power-efficient way. It also turned out that these inner-loop calculations could be structured in such a way that the same data was used a number of times, and with well-localized data transfers that minimized the need to send data across the chip, much less to and from off-chip memory.

There were some parts of our application where any function-specific hardware would have been grossly underutilized or unable to take advantage of certain types of new algorithms or biophysical models that might later be discovered. Fortunately, most of those calculations aren't executed all that frequently in a typical biomolecular simulation. That made it feasible to incorporate a set of programmable on-chip processors that could be used for various calculations that fell outside the inner loop. We were also able to make use of problem-specific knowledge to provide hardware support for a specific type of inter-chip communication that was especially important in our application.

Since my own interest is in the application of molecular dynamics simulations to biological problems, I haven't been forced to think very hard about what aspects of the approach we've followed might be applicable to the codesign of special-purpose machines and algorithms for other applications. If I had to guess, I'd say that at least *some* aspects of our general approach might wind up being relevant to researchers who are looking for an insane increase in speed for certain other applications, but that hunch isn't based on anything very solid.

HANRAHAN: In the *Communications* paper, Anton was described as a computational microscope. I really liked that phrase and that the name Anton came from van Leeuwenhoek, who was one of the first microscopists.

SHAW: Part of the reason I like the metaphor of a computational microscope is that it emphasizes one of the key things that Anton *isn't*. Although we sometimes describe the machine as a "special-purpose supercomputer," its range of applicability is in practice so narrow that thinking of Anton as a computer is a bit like thinking of a microscope as a general-purpose laboratory instrument. Like the optical microscope, Anton is really no more than a specialized tool for looking at a particular class of objects that couldn't be seen before.

HANRAHAN: So now that you have this microscope, what do you want to point it at? I know you must be collaborating, and you have computational chemists and biologists at D. E. Shaw Research. Do you have some problems that you want to go after with it?

SHAW: There's a wide range of specific biological phenomena we'd like to know more about, but at this point, there's a lot we can learn by simply putting things under the microscope and seeing what's there. When Anton van Leeuwenhoek first started examining pond water and various bodily fluids, there's no way he could have predicted that he'd see what we now know to be bacteria, protozoa, and blood and sperm cells, none of which had ever been seen before. Although we have no illusions about our machine having an impact comparable to the optical microscope, the fact is that nobody has ever seen a protein move over a period of time even remotely close to what we're seeing now, so in some ways, we're just as clueless as van Leeuwenhoek was when he started looking.

All that being said, there are some biological systems and processes that we've been interested in for a while, and we're beginning to learn more about some of them now that we're able to run extremely long simulations. The one that's probably most well known is the process of protein folding, which is when a string of amino acids folds up into a three-dimensional protein. We've already started to learn some interesting things related to folding that we wouldn't have known if it hadn't been for Anton, and we're hoping to learn more over time. We've also conducted studies of a class of molecules called kinases, which play a central role in the development and treatment of cancer. And we're looking at several proteins that transfer either ions or signals through membranes in the cell. We're also using Anton to develop new algorithms and methods, and to test and improve the quality of the physical models that are used in the simulation process itself.

HANRAHAN: It seems like you're almost at a tipping point. I worked at Pixar, and one of the singular events was when computer graphics were used in *Jurassic Park.* Even bigger was *Toy Story*. Once the graphics software reached a certain maturity, and once you showed that it could be used to make a blockbuster, then it wasn't that long afterward that almost every movie included computer-generated effects. How close are we to that tipping point in structural biology? If you're able to solve some important problems in structural biology, then people might begin considering molecular dynamics simulations as part of standard practice, and then routinely deploy this approach to solving biological problems.

SHAW: That's a great analogy. Although the evidence is still what I'd characterize as preliminary, I think there's enough of it at this point to predict that MD simulations will probably be playing a much larger role within the field of structural biology a few years from now. It's hard to tell in advance whether there's a *Toy Story* around the corner, since the biggest breakthroughs in biology often turn out to be ones that would have been difficult to even characterize before they happened. It *may* be that there are some deep principles out there that are waiting to be discovered *in silico*—ones that can tell us something fundamental about some of the mechanisms nature has evolved to do what it wants to get done. It's hard to plan for a discovery like that; you just have to be in the right territory with the tools and skills you think you might need in order to recognize it when you see it.

HANRAHAN: There's no place that's more fun to be when you have a new microscope. van Leeuwenhoek must have had a good time. I've read a bunch about Robert Hooke, too, who was a contemporary of van Leeuwenhoek. He was part of the Royal Society. Every week or two, they would get together and make all these discoveries because they were looking at the world in a different way.

SHAW: I've always thought it would be great to live during a period when a lot of fundamental discoveries were being made, and a single scientist could stay on top of most of the important advances being made across the full breadth of a given discipline. It's a bit sad that we can't do that anymore—victims of our success.

HANRAHAN: But one thing that amazes me about you is the number of fields you've had an impact on. I'm trying to figure out your secret sauce. You seem to be able to bring computation to bear in problem areas that other people haven't been as successful in. Obviously you've had a huge impact on the financial industry with the work you did on modeling stocks and portfolios. And now you're doing biochemistry. How do you approach these new areas? How do you bring computers to bear on these new problems?

SHAW: I'm not sure I deserve those very kind words, but for what it's worth, I tend to generate a plentiful supply of ideas, the vast majority of which turn out to be bad ones. In some cases, they involve transplanting computational techniques from one application to another, and there's usually a good reason why the destination field isn't already using that technique. I also have a remarkable capacity to delude myself into thinking that each idea has a higher probability of working than it really does, which provides me with the motivation I need to keep working on it. And, every once in a while, I stumble on an idea that actually works.

HANRAHAN: It sounds like you have this "gene" for computing. You know algorithms, you know architecture, but yet you still are fascinated with applying them to new problems. That's what's often missing in our field. People learn the techniques of the field but they don't know how to apply them in a new problem domain.

SHAW: I love learning about new fields, but in some ways I feel like a tourist whose citizenship is computer science. I think to myself, "I'm *doing* computational finance, but I *am* a computer scientist. I'm *doing* computational biology, but I *am* a computer scientist." When we computer scientists start infiltrating a new discipline, what we bring to the table is often more than just a bag of tricks. What's sometimes referred to as "computational thinking" is leaving its mark on one field after another—and the night is still young. C

Related articles on queue.acm.org

A Conversation with Kurt Akeley and Pat Hanrahan
http://queue.acm.org/detail.cfm?id=1365496

Beyond Beowulf Clusters
Philip Papadopoulos, Greg Bruno, Mason Katz
http://queue.acm.org/detail.cfm?id=1242501

Databases of Discovery
James Ostell
http://queue.acm.org/detail.cfm?id=1059806