



[HTTP://www.dotnetrocks.com](http://www.dotnetrocks.com)



Carl Franklin

Carl Franklin and Richard Campbell interview experts to bring you insights into .NET technology and the state of software development. More than just a dry interview show, we have fun! Original Music! Prizes! Check out what you've been missing!



Richard Campbell

Text Transcript of Show #480
(Transcription services provided by [PWOP Productions](#))



Microsoft battles HIV!
September 10, 2009
Our Sponsors



Developer
EXPRESS

[HTTP://www.devexpress.com](http://www.devexpress.com)



CoDe
component developer magazine
[HTTP://www.code-magazine.com](http://www.code-magazine.com)



RadControls
FOR ASP.NET
telerik
[HTTP://www.telerik.com/](http://www.telerik.com/)



Geoff Maciolek: The opinions and viewpoints expressed in .NET Rocks! are not necessarily those of its sponsors, or of Microsoft Corporation, its partners, or employees. .NET Rocks! is a production of Franklins.NET, which is solely responsible for its content. Franklins.NET - Training Developers to Work Smarter.

[Music]

Lawrence Ryan: Hey, Rock heads! Quit Binging your Google and listen up! It's time for another stellar episode of .NET Rocks! the Internet audio talk show for .NET developers, with Carl Franklin and Richard Campbell. This is Lawrence Ryan announcing show #480, with guests Jonathan Carlson, Bob Davidson, David Heckerman, and Carl Kadie, recorded live Monday, August 17, 2009. .NET Rocks! is brought to you by Franklins.NET - Training Developers to Work Smarter and now offering DotNetNuke video training with Chris Hammond from Engage Software on DVD, dnrTV style, order your copy now at www.franklins.net. Support is also provided by Telerik, combining the best in Windows Forms and ASP.NET controls with first class customer service, online at www.telerik.com, and by GrapeCity Data Dynamics, makers of ActiveReports.Net, simple, powerful and cost-effective reporting for Windows Forms and ASP.NET web applications, online at www.datadynamics.com. Support is also provided by CoDe Magazine, the leading independent magazine for .NET developers, online at www.code-magazine.com. And now, the man who drinks his VB.NET with a C# chaser... Wait a minute, did I F# that up? Carl Franklin.

Carl Franklin: Thank you very much and welcome back to .NET Rocks! Carl Franklin here.

Richard Campbell: Richard Campbell here.

Carl Franklin: How are you doing?

Richard Campbell: I am well, sir. Things are good. No rest for the wicked.

Carl Franklin: Hey, show 500 is coming up.

Richard Campbell: It's coming.

Carl Franklin: And we promise we won't do anything as bad as show 400.

Richard Campbell: We'll be sober.

Carl Franklin: Yeah, exactly. Yeah, show 400 was really kind of an afterthought, but for 500 we have an idea. Because we're going to be in Sweden, we're going to be in Berlin, we're going to be in Las Vegas, we're going to be in Poland, we're going to be

in Bulgaria, and Amsterdam, and in Los Angeles before show 500 airs, so we're going to talk to a lot of people and get a lot of little vignette interviews and just see what people have to say about .NET, and about .NET Rocks!, and what they're looking forward to. Also, we want to hear from you, our listeners. It's been a long time since we just had a mass call-in show.

Richard Campbell: Right.

Carl Franklin: So that's what we're going to do. So if you have a .NET Rocks! story or a quip or something that happened that we helped you with or something funny, anything like that, call us and leave a message and we'll play it on the show. Inside the United States you can call toll-free, 8774926751, outside the United States you can call 8604478832 and just leave us a message and we'll be sure to, you know, if it's funny and good.

Richard Campbell: If you qualify, you could be on .NET Rocks! episode 500.

Carl Franklin: Do you remember show 100? Show 100 was like this but we had guests call in and leave messages.

Richard Campbell: Right.

Carl Franklin: Probably we'll have some guests call too, but it was funny because we've got some guests who were like, "Okay, Carl and Richard, start recording now. Hey, Carl and Richard, this is..." and we played them just like that, but anyway... So let's get into Better Know a Framework and start the show off.

[Music]

Richard Campbell: All right.

Carl Franklin: Today, I'm going to talk about HTTP Style UriParser.

Richard Campbell: Oh.

Carl Franklin: Which is in System and this inherits System.UriParser and there's a whole bunch of URI styles of course, FTP, File, HTTP, and so this one lets you parse in HTTP URI which is essentially a URL. Right?

Richard Campbell: Right.

Carl Franklin: The important part of the UriParser base class is get components which gets the components from the URI so that will return the scheme, the host, the port, all the little pieces of the

URI and in this case an HTTP URI or URL. So there you go. So Richard, you've got an email for us?

Richard Campbell: I do indeed and this one's subjectline is "Database guarantees from a used car salesman?"

Carl Franklin: Hmm?

Richard Campbell: "Hi guys. I've just listened to show 466 and I have to support Damien's call for a database episode of .NET Rocks!"

Carl Franklin: Ooh.

Richard Campbell: "As a C# developer, I use .NET Rocks! to keep me inform on subjects that relate to my professional work but which I don't have the time to investigate thoroughly. Since I don't know if I'm ever going to work with Surface, Silverlight, or SharePoint but rely on databases ever single day, I think SQL Server or Databases in general are prime candidates for the subject of the show. I dig into the details of databases inter-working so seldomly that I forget all the subtleties that's why I would love it if you would drill these subtleties into my brain once again."

Carl Franklin: Yeah.

Richard Campbell: "Databases are very much about guarantees and whenever I start thinking about these guarantees I feel like a bad boy for relying so heavily on something that I don't quite understand."

Carl Franklin: Yup.

Richard Campbell: "How are transactions implemented? How do transaction logs work? If used correctly, do transactions really guarantee the acid properties as I would expect with the Uncertainty Principle in quantum physics and what-not. There has to be a limit somewhere."

Carl Franklin: Ooh. Quantum Physics, I knew that would come up sooner or later.

Richard Campbell: "Are distributed transactions really possible without giving up some of the acid properties to some degree?" Just on the side here, the answer is no. They really are that reliable. I've done these tests but we can talk about that later.

Carl Franklin: On another show. Hey, weren't Kim and Paul supposed to do their own podcast on SQL Server?

Richard Campbell: Supposedly.

Carl Franklin: Supposedly.

Richard Campbell: I don't know if I believe them.

Carl Franklin: Every once in a while we get an email, "Okay, we're really serious this time," and we never hear from them again.

Richard Campbell: We never hear from them again. They're busy people, what can...

Carl Franklin: They are very busy people and God bless them.

Richard Campbell: Yeah. Let me finish this email.

Carl Franklin: All right.

Richard Campbell: "Relying on these things feel very much like applying a mathematical theorem for which I've never seen the proof. For all intents and purposes, I could probably define without understanding the details but it just doesn't feel right. It kind of feels like cheating. So please, if you find it in you to share some of your wisdom on these issues mentioned above it will be greatly appreciated. I love the show. It continues to push me to expand my skillset. Cheers, Rune Ibson from Denmark."

Carl Franklin: Cheers, Rune.

Richard Campbell: Thanks, Rune.

Carl Franklin: Yeah.

Richard Campbell: We'll send you a mug.

Carl Franklin: Absolutely.

Richard Campbell: And if you've got questions, concerns, ideas for shows, things we ought to be doing or should be doing more of, send us an email, dotnetrocks@franklins.net.

Carl Franklin: Our guests today are Jonathan Carlson, Bob Davidson, David Heckerman, and Lisa Hildebrandt, and Carl Kadie who are working on a very interesting project and I'm just going to let them introduce themselves and then we'll talk about the project. So who wants to take a first crack at it?

David Heckerman: Okay. Well, I'll go first. This is David Heckerman. I've been a researcher at Microsoft for 17 years. I have a background both in Statistics and Medicine. I have an M.D. PhD hence the interest in HIV vaccine.

Carl Franklin: Okay.

Carl Kadie: This is Carl Kadie. I'm 15 years with Microsoft. I'm a research programmer in the Research division. Before that I worked on the

very first version of MSN. Academically, I have a PhD in Machine Learning but when I came to Microsoft I was working just as a line software developer then I moved to research.

Carl Franklin: Okay.

Bob Davidson: This is Bob Davidson and I'm almost a 21-year veteran of Microsoft where I've been doing internal tools for most of that time. I joined David's group about six months ago to work on the biology side of things and to try and work on the things like we're going to talk about today.

Jonathan Carlson: My name is Jonathan Carlson. I'm a researcher at Microsoft Research working in eScience group but I focus on my researches on HIV, specifically we look at how HIV adapts in the immune system. Even making the machine learning technique...

Carl Franklin: Oh, machine learning, yeah.

Jonathan Carlson: We develop graphical models and try to model how HIV adapts in the immune system.

Carl Franklin: Okay. Well, who wants to start this off and tell us how this project got started. I mean the goal is to find a vaccine for HIV, how did it start?

David Heckerman: This is David Heckerman. We were doing sort of a straightforward application statistics in computer science. We were building things like spam filters and data mining tools as part of Microsoft products and I have this background and this general interest in medicine, and at one point it became clear that the sort of tools that we're developing might be applicable to these very challenging problems such vaccine for HIV. So we kind of dabble with that a bit and had a little bit of success, and Bill Gates actually looked at our work and was very interested in it. He helped us, introduced us in terms of working on HIV throughout the world actually and we started talking with them and we realized that we could help them a bit in processing and analyzing their data finding new interesting signals in their data to help build this HIV vaccine.

Carl Franklin: So by machine learning, are you talking about neural networks or not exclusively or other techniques as well?

David Heckerman: Yeah, machine learning is very broad. It basically means statistics with a computational benefit to it and so there are all sorts of machine learning techniques. The ones that we practice the most go by the name Graphical Model. You could consider neural method a special case.

Carl Franklin: So this is a project that started at Microsoft?

David Heckerman: Yes.

Carl Franklin: Wow.

David Heckerman: We're in this group called Microsoft Research. There are about 800 of us worldwide.

Carl Franklin: Sure.

David Heckerman: There are a lot of different things as you might guess with 800 people working in research. There are a lot of different things that we're doing and there's certainly room for some of us to think about how Computer Science and Machine Learning and Statistics can help in the areas of biology.

Carl Franklin: Specifically, you were talking before we started recording about how HIV is a special case because it mutates so quickly and that's one of the reasons why scientists haven't been able to nail down a vaccine. How is what you're working on attacking that problem?

David Heckerman: Well, what we're trying to do is figure out just how HIV mutates when it gets inside of you. It's not completely free to mutate any which way. HIV is not going to mutate into a zebra or something like that.

Carl Franklin: Right.

David Heckerman: It's constrained. So we're trying to, through computer science technique and a lot of data gathered by our collaborators, trying to figure out exactly what the constraints on those mutations are and if we can figure that out then we can build a vaccine that will teach our immune system to keep one step ahead of the virus.

Carl Franklin: So you're looking for patterns. Are you looking at genetic data?

David Heckerman: Absolutely.

Carl Franklin: Yeah, you're looking at gene sequences and how they change. Are you also looking at the -- I can imagine millions of factors that come into play as possible effectors of those mutations.

David Heckerman: Absolutely and Jonathan, why don't you -- Jonathan has been doing a lot of great work in this area in bringing all these different factors.

Carl Franklin: Jonathan.

Jonathan Carlson: Yeah, so we're focusing on onset of human proteins, a class of proteins called the HLA protein and these proteins are known to help the immune system identify which cells have been infected with HIV. The way it does that is it actually recognizes and binds the short sequences of HIV viral protein in fragments and presents those fragments to the immune system. So what we're working at is there are patterns to how HIV mutates to different versions of the HLA protein.

Richard Campbell: The idea being there's only so many combinations, and in theory once you know all those combinations you can build something that will cover the full spectrum.

Jonathan Carlson: Exactly and one of the interesting properties of the HLA protein is that they're highly diverse in the human population so there maybe a thousand or so variations of this protein...

Carl Franklin: Wow.

Jonathan Carlson: And each person has sets of this protein and so for each person there's a very specific subset of HIV fragments and there are means of being capable to target them.

Richard Campbell: Okay.

Carl Franklin: Are these the things that cause the virus to mutate itself? Are these the only things that cause the virus to mutate?

David Heckerman: They certainly are not the only things and I want to be clear they're not causing the virus to mutate. To mutate the virus, it mutates randomly.

Carl Franklin: Randomly.

David Heckerman: Then there's the question of which of those mutations are selective for the process in natural selection.

Carl Franklin: Oh I see.

David Heckerman: There are some random mutations that will be beneficial to the virus, and some of them will be costly to the virus. Whether they're beneficial or costly is somewhat a function of how the virus protein structured itself and somewhat a function of how that protein interacts with the human immune system.

Carl Franklin: I see.

David Heckerman: So the HLA proteins are a major force of interaction between any instance and the virus.

Carl Franklin: So based on the HLA proteins that the particular person has, they would be more or less likely for the HIV virus to mutate to specific set.

Bob Davidson: So to be successful in certain areas, so if your HLA recognizes one and successfully attacks it and kills that variation of the HIV virus, then the virus is no longer successful so what the virus is trying to do is to mutate where the HLA is not going to recognize it. The human side is trying to make sure that you can recognize foreign bodies and encapsulate them and present them to the rest of the immune system to shut it down.

Carl Franklin: Oh okay.

Bob Davidson: So it's a hide-and-seek game that's going on if you will where the AIDS virus, as it goes to this random mutations, is trying to -- and were imputing attempt here, what I mean is its more random than attempt, but is successful when he can hide and not be found and is able to more readily infect other cells and to reproduce itself.

Carl Franklin: Yeah.

Bob Davidson: That's the hide-and-seek game that's going on.

Carl Franklin: Okay. Tell us a little bit about the machine learning process and how that is helping, how that works.

Jonathan Carlson: The basic idea is to try to build the statistical model of how evolution works, and what we can do there as David mentioned earlier, the ideal graphical model. The idea is that you can draw out the patterns that you think are happening and then test statistically whether those models fit the data. So for example, we have a model that says that HIV is randomly mutating throughout the course of history until it infects a specific individual; we actually get to feed the players of this cat and mouse game, this hide-and-seek game that Bob was talking about and then we look for correlations in those individuals. The idea then is that if we see that everybody with a certain HLA protein has specific mutation or is more likely to have a specific mutation with an HIV, then that mutation is likely to help HIV adapt from that particular HLA protein.

Carl Franklin: So it sounds like you're working to produce more than one strain of this vaccine. It sounds like you're going to have many. Is that true?

Jonathan Carlson: Yes, absolutely. That's one of the challenges with HIV. Because it mutates so rapidly, it looks like we may have to design something that is more specific for people with specific HLA proteins and so of course one of the goal is to find mutations that are common across HLA proteins or across very broad subset of HLA proteins.

Carl Franklin: Sure. So I can imagine that you would have sort of a base level if you want, say, a more general vaccine and then some more specific vaccines base on the HLAs that person has.

Jonathan Carlson: Yeah, to tell exactly where that is going to go but that's only a possibility.

Bob Davidson: In terms of just the economics of you'd like it to be a small number as possible.

Carl Franklin: Sure.

Bob Davidson: And if you can have it so that there is a critical protein that you recognize that we don't recognize now, then that maybe an interesting point as well. I mean, as Jonathan, said there are so many opportunities still available that we're learning so much that it's hard to say what direction its going to end up going.

Carl Kadie: This is Carl Kadie. One of the tools that we have after we find some of these candidate sequences, there's a tool called create epitome and with it you give it a list of the little protein fragments and how much of the population each one covers and then it works to try to create longer and longer sequences of protein so that it confines one that's not too long but still covers as much of the population as you can. That's kind of a straightforward computer science optimization with a little bit of a trick, but we don't think we can practically cover everything so we're trying to cover as much as we can with not going too long without making the vaccine too long.

Carl Franklin: How far long in the process are you?

David Heckerman: We are at the stage now where we have some what we think are reasonable designs for vaccine and now I think we want to get a little bit more data and verify those designs, but there's a good chance that we'll now want to move to the next phase which is testing which is very tough...

Carl Franklin: I was going to say that. Are there possible side effects from a vaccine such as this, things like...?

David Heckerman: Yeah. Generally, first thing you want to test for is safety and then once you know the

things are safe then you move on to efficacy, or sometimes you test for safety and efficacy. Generally, what you do is you get volunteers who are at high risk of getting HIV and who are willing to take the vaccine and then they get the vaccine and you watch what happens. Watching it work takes a long time.

Carl Franklin: Yeah.

Richard Campbell: Is the software you're working on also going to help on the testing side of things as well as you start doing these trials?

David Heckerman: To analyze the results of these trials, we want to apply statistical methods and occasionally some of these fancier statistical machines learning sort of things that we do and be helpful, but I think most of our effort is on the design side as oppose to the evaluation side.

Carl Franklin: This portion of .NET Rocks! is brought to you by our good friends at Telerik without whom this show would not exist. No doubt you bump into testing tasks now and then in your work and we can bet writing functional test is not your favorite thing. It's difficult. It takes ages and the results could be dubious. Well, get ready to start liking it, thanks to Telerik. With the just launched WebAii testing framework, building web automation tests is a breeze. Enjoy codebase automation of advance ASP.NET AJAX and Silverlight apps. Write a single test and have it executed against multiple browsers at once. Benefit from rich API LINQ support, integration Visual Studio unit testing NUnit, xUnit, and MbUnit, not to mention the free wrappers for a Telerik RadControl for ASP.NET AJAX and Silverlight as shipped with Telerik's new testing tool. Surely one of its best features, WebAii testing framework which is developed by ArtOfTest, is absolutely free. If you're already hooked on WebAii testing framework, you can start using it right away. Go to www.telerik.com for more info. And hey, make sure you thank them for supporting .NET Rocks!

Richard Campbell: So what sort of test are you in the process of doing that you're analyzing the data from? This is blood sample work from a large collection of people?

David Heckerman: Yeah. For example, we can get people that are infected with HIV and we actually know how badly or how well their immune system is fighting HIV, and then we can measure what parts of HIV their immune system is attacking and what's the correlation between where the attacks are taking place and how well the attack are doing.

Richard Campbell: I mean, HIV has really run rampant in Africa. What's the sampling like? Is there an African focus or is this a worldwide research plan?

David Heckerman: We're working with collaborators around the world. Definitely some of those people are working with folks in Africa, South Africa, in particular around the area of Durban. Our collaborators are Philip Goulder and Bruce Walker. Philip is at Oxford. Bruce is at Harvard. But we're working with other collaborators, as well as people base in the United States, people base in Australia, people base in other areas of Africa.

Richard Campbell: Just to be clear, you guys aren't actually doing the blood sample collection. You're just gathering data from all the different folks that are doing it to do the analysis?

David Heckerman: Right.

Richard Campbell: Okay.

Carl Franklin: Now, this is as you said a vaccine. So this is a preventative measure. Can it also be effective on people who already have the virus?

David Heckerman: It maybe. This is an area that's very interesting to us and obviously a lot of other HIV researchers. We're not sure. Some of the tests that we're going to be doing in the coming months will help us determine that.

Carl Franklin: What kind of technology are you using? I can imagine there's maybe an OLAP cubes somewhere?

Carl Kadie: Some of the initial data is stored in SQL. A lot of it comes from our collaborators just in hand coded Excel files. Most of the coding we're doing is in C# and we're creating new algorithms to learn not so much using -- we're using a lot of the .NET technology but kind of the analysis parts are the new thing that we're contributing and so as you can imagine this is computationally intensive as well.

Carl Franklin: Right.

Carl Kadie: So there's a lot of work to do. You know, you start on the desktop and then you need to scale out and so we're using clusters and then we're also experimenting with some of the work on Azure and how far we can scale out.

Carl Franklin: I was going to say maybe the peer method might work really well for this if it's pure number crunching. You know, sort of like the SETI@Home model.

Bob Davidson: Partially the problem with that is that we have confidential information here and so where we can send information or we can't sent information is somewhat problematic.

Carl Franklin: Sure.

Carl Kadie: We're fortunate to have access to a 6,000-node HPC Cluster.

Carl Franklin: Oh.

Carl Kadie: So until we outgrow that if we outgrow it, that has the convenience of just being upstairs and having all the machines be identical and easy to program.

Carl Franklin: That's crazy.

Richard Campbell: Only 6,000 nodes.

Bob Davidson: Only 6,000. We have to share them with other projects.

Carl Kadie: Once in a while.

Richard Campbell: Can you talk a little bit about the programming side of the HPC part? Like breaking, this sort of work up and define pieces. Is that really difficult?

Carl Kadie: So most of the problems we're working on are what some people call embarrassingly parallel, but I'd like to think of it as being delightfully parallel. The problems are easy, generally easy to split up into 1,000 or 10,000 or last week I was running something that split up into 100,000 parts and the program knows that the work is in the 100,000 parts and it knows that it's suppose to do part, say 220, and then it just writes its output which on a lot of these programs is a single text line for each of the work items which they might be on -- the total work might be in the millions or up to a billion, and then there's usually one final pass we call tabulation that we can do on a desktop.

Richard Campbell: Which is really sort of the synthesizing of all those different units back together again?

Carl Kadie: Right. A lot of times each line represents one statistical test, but because we're literally doing millions or perhaps a billion statistical tests the chance of having coincidental result is very high. So that last step is to try to find out what's likely to be really significant, not just apparently significant.

Richard Campbell: That sounds like a fairly specific scientific terminology too apparently versus really.

Carl Kadie: Yeah. Typical scientific test is something significant if there is only five chances in a hundred, that it happened by chance.

Richard Campbell: Right.

Carl Kadie: But of course if you're testing a million hypothesis, that's a lot of things that are going to be said to be significant and a lot of our results go back to the lab where they can be tested and we really don't want to give them 50,000 things to test.

Richard Campbell: In the end, the service you're providing more than anything is to call down the sheer volume of data into something that's relevant, that they'll be testing the things that matter.

Carl Kadie: That's correct. A lot of results will say here's a list of a hundred, say, of these little protein segments that we think would be useful for vaccine. Of these hundred, we think half of them are real. We can't tell you which half, but a hundred is not that many and you can go test it in the test tube.

Richard Campbell: Right.

Carl Franklin: Yeah.

Richard Campbell: And this is about using computers to cut down the amount of test tube time.

Carl Kadie: Exactly.

Richard Campbell: Am I reading this correctly? Are all the tools that you're working with here are essentially available online? The computational biology tools are on CodePlex?

Bob Davidson: That's correct. We have internal versions so not everything is up to the minute online, but basically when a scientific paper is published about the work, then the code is open source and put on CodePlex and will often try to make a Web version or a Silverlight version available too.

Richard Campbell: Wow.

Carl Franklin: Yeah, I'll say.

Richard Campbell: And in just sharing all these information. You've also been working on the malaria and Hep C as well?

David Heckerman: Yeah. The problem with Hep C is very similar to HIV. Hep C is a virus. It mutates very rapidly, about as rapidly as HIV, and it's not as

lethal as HIV but it's certainly causing a lot of problems, and it's not as prevalent as HIV but it certainly affects a lot of people.

Richard Campbell: It did seem to me, and maybe this is just falls memory, that the two ailments sort of popped up around the same time in the '80s.

David Heckerman: HIV definitely got started in early '80s. I don't remember when Hepatitis C was -- you know, it was coming out but if your memory says it was early '80s, then that's quite possible. I don't know that they're related. They're fairly very different viruses, but I don't think there's any correlation there or possible connection.

Richard Campbell: It's interesting that they have similar mutation behavior. I mean, is that a very common behavior?

David Heckerman: No. These viruses, there are only a handful of these viruses that mutates...

Richard Campbell: Fairly rare.

Carl Franklin: Is this being done, like has this been done before this particular type of analysis that you're doing and having so much success with, and if not, why not? Is it a matter of funding? Is it a matter of smarts or technology or...?

David Heckerman: I think it's been about, what, 25 years since HIV vaccine research really kick in the gear and originally people thought, "Well, it's a vaccine. We'll go with the various standard techniques for building a vaccine," and it just takes a lot of time to exhaust the low hanging fruit.

Carl Franklin: Okay.

David Heckerman: And so now we're at the stage where people are going, huh. You know, all of the ways that people have tried to build a vaccine to test are just not working. We've got to get creative here, and so we're part of this wave where people are trying some very unusual things to develop a vaccine.

Carl Franklin: Yeah.

David Heckerman: And then there is the computational issue. I mean, we really do use these 6,000 nodes. Unfortunately, we use a lot of electrical power to solve this problem and this computational power is a test that have been around that long.

Bob Davidson: I mean, an example, one of the problems we're working on a couple of weeks ago that Carl did a really good job of optimizing are initial estimates from a hundred years of computation time.

Carl Kadie: Yup, that's right.

Bob Davidson: And Carl was able to improve it. But even so, we got it down on the cluster and it was still weeks of running on the cluster after much optimization on his part and again the cluster is not small.

Richard Campbell: You got it from a hundred years to a couple of weeks so don't be too upset.

Carl Franklin: Yeah.

Bob Davidson: Well, we went from a hundred CPU years to 14 CPU years which on a thousand processors took just under a week.

Richard Campbell: Oh, nice.

Carl Franklin: Wow.

Bob Davidson: I think one of the things that has changed is statisticians' always kind of knew the idea of some of the techniques we're using, but they thought of them as kind of not practical. Because like one thing we do the control randomness, it's we just run up many, many cases that we know are random just to test them against the cases that we're hoping aren't random. So people always knew that would kind of work theoretically but everyone thought that was just kind of a crazy waste of computation, but times have changed and now it's a sensible use of computation.

Richard Campbell: Well, the fact is we have a lot of horsepower to use and pretty good tools to utilize it.

Jonathan Carlson: One of the things that this points out is that when you have statistics that started off because you didn't have the computational power you need to do clever tricks to not measure something then the computational power increases and your data awareness and data collection and your data gathering power increases, statistics are there to validate things but you can start to measure things and you can start to really count things when you do this. So as more and more power comes online, more and more direct measurement. If you look at the environment and other places where people are measuring impacts, the number of data points you've got to look at using count, and now you want to infer and get the statistics out of all that raw data, there's going to be a ton of cool heavy duty computational problems coming at us.

Carl Franklin: Let me ask you this. Are you guys using LINQ or using any sort of functional programming languages? F# perhaps?

Carl Kadie: Yeah. We're using C# and within C# -- soon as LINQ became available we switched over to it in a lot of places and a lot of the new code uses it even more. A lot of our code does what LINQ does. It enumerates some things, maybe one batch of things and inside that another batch. It might do some selection. Sometimes it does groupings, and a lot of times one thing pipelines into another. So LINQ has been a very nice way to express some of the algorithms.

Carl Franklin: How about on the functional programming side?

Carl Kadie: We haven't been doing functional programming beyond what is convenient to do in C#, LINQ in passing, Lambda Expressions, and that sort of things and that has worked out pretty well for us. It gives us a nice language to code quickly in C#, and by making parts of the code functional it does makes it easier to run things on multiple processors on one machine which is a nice thing to do if you're waiting for your results to come back.

Carl Franklin: PLINK also?

Carl Kadie: Exactly. We're using the parallel extension library in PLINK to run on typically eight processors when we're running on a desktop and it's very nice to get the results back in 10 to 12 minutes instead of in an hour.

Carl Franklin: That's great.

Bob Davidson: And this does lead to one thing that is not directly related here, but the work that we're doing within the group here, we're also leveraging -- and there's not an announcement yet but the Microsoft Biology Foundation, we will have another library of code that people can use to take this code and use it. Really even though we have open source, the original solutions that we've come up with, there's a lot of people that want to do a lot more exploration and ask lots more questions and having a library there for them to do the manipulation and to do the data interrogation is on a roadmap.

Richard Campbell: It also seems to me that you're really exercising these concepts of massive parallel computing so no matter what project is working on going forward this is great code to reference on how to parallelize work.

Carl Kadie: Yes. For these kinds of jobs that started on the desktop and then you want to move on to the cluster...

Richard Campbell: Right.

Carl Kadie: We've developed our own libraries that work on top of the HPC libraries that make it especially convenient to kind of add a flag and it runs on the desktop and then you start to copy your results back. But we've been automating more and more of the process, and within our own tools we've been finding that we're kind of gaining leverage by having a shared -- a library we're using among all our tools and those become available as we put things on CodePlex.

Richard Campbell: So really it's about there's a big difference between running on eight cores and running on 6000, and you try to make that fairly transparent?

Bob Davidson: I just want to make sure, as Carl said we're working with a lot of very delightfully parallel programs, and so again we run the same program across the different subsets of the data across those nodes. But to run on the desktop or to run on the node in a cluster, we actually run the same code for 99% of the code. When we start looking at I want to really split parallelization in non-embarrassingly or non-delightfully parallel situations, then we're going to have a harder time just like everybody else. Writing good, efficient parallel code that makes efficient use of computation, communication, and storage is still a hard problem in many spaces.

Carl Franklin: Hey, I just want to give a shout out real quick to our friends at Data Dynamics who make ActiveReports.NET among other awesome things. ActiveReports.NET is great because it allows you to just build your reports with the Easy Editor, embed them right in your application, provide PDF and HTML output, give your end-users a Report Editor, royalty free of course, a great Access report upsizing Wizard and all this for a price that isn't going to break the bank. ActiveReports.NET from Data Dynamics, go check it out now at datadynamics.com.

Richard Campbell: So then we start pulling this concept of Azure. Now that you're starting to feel like 6,000 nodes is a bit tight, do you think the Azure guys can give you more room?

Jonathan Carlson: Yes. We actually have an Azure implementation of the code up and running right now. It's not scaled out as far as we have with the cluster. We see slightly different behaviors. When you design a cluster, you design a cluster for high performance so we use bleeding edge processors. We have good IO, we have high speed interconnect. When you design Azure for massive scale out and power management, you don't use necessarily the latest and greatest Intel processor so you see slight different performance characteristics on the individual node bases but you have much more

opportunities to scale up. So it's really interesting when I look at the difference between the two in how they play together, but yeah, we expect to scale up to Azure as well.

Richard Campbell: Yeah. Correct me if I'm wrong, but this HPC machines, they have these really high speed connections between the machines so they're literally like bus-level speed machines from machine to machine.

Carl Franklin: Backplane.

Jonathan Carlson: Yes, some machines do. The cluster that we're on is not Fibre Channel everywhere to backplane, backplane. But we do have high speed networking, it's all co-located, you're not going through a lot of switches. So it's all sitting there in one lab. Again, for our situation we're not communications bound.

Carl Franklin: Okay.

Richard Campbell: Right.

Jonathan Carlson: We take a little bit of data from the dataset on each node. We process it, we crunch, crunch, crunch and then we write a little bit of data out that we then go back and validate and summarize and say here are the interesting points.

Carl Franklin: So you're spending more time crunching numbers than you are at transmitting data back and forth.

Jonathan Carlson: Correct.

Bob Davidson: One interesting thing that we think Azure give us is not just scaling up with more processors like we get with the HPC, but also making it -- we're hoping it will make it much more convenient to have multiple users. Right now, when we want to run a collaborators' job, they end up mailing or FTP-ing the dataset to us and we have to go through some manual steps to run the job for them and send the results back. So we think just the bookkeeping we're hoping will be so much easy on Azure where perhaps people can be given their own accounts and just their own computational resources.

Richard Campbell: Well, and you use what you need.

Carl Kadie: Because really a cluster is just a big desktop computer that one person owns.

Richard Campbell: Yeah. The great thing about being in a cloud is all that doesn't matter now. You just harness what you need when you need it, and in theory base on the size of the set you could really

decide I need this ready in a day, I need it ready in a week, I need it ready in an hour and you scale accordingly.

Bob Davidson: Yeah, the cost model, the value model, all that comes into play in what's possible and I don't think we have the answers to those questions yet of what makes the most sense but we're definitely experimenting with what it looks like in Azure and how does it play. If you really want an answer quickly, have you pre-provisioned all your machines? How do you do it? As you look at where they are geographically, you still have to move. I mean, there is still data that has to be moved, it's just not completely critical path to everything else that has to happen.

Richard Campbell: Right. They're not emailing it to you anymore so that's good.

Jonathan Carlson: Yup. If we can have the customer do the self-service aspect of it and, you know, it's still cleaning up some interfaces. As with a lot of research code, there's a difference between research and production but dealing here with mostly researches so if you look at the difference between what is available already and what's going on, there's a different level, expectation level on what the policy is.

Richard Campbell: Right.

Jonathan Carlson: I think we're in pretty good shape.

Richard Campbell: Well, and there's also a difference between building an app for a particular project and building it for a lot of people's projects.

Jonathan Carlson: Yeah, if you have one customer or you have a lot of customers, they're all trying to solve the same thing, you can standardize it and you can make it move forward versus there's one guy and his data comes in this way and he's the only guy that has ever going to run it then who does the customization?

Carl Franklin: So do you guys have any guesses as to when you think you may have an effective vaccine ready?

Carl Kadie: When we will have dispatched AIDS?

Carl Franklin: Yes. Guess?

Carl Kadie: Friday.

Carl Franklin: I get it. Moo.

Carl Kadie: I have no idea which Frida, but Friday.

Richard Campbell: It will be a Friday for sure.

Carl Franklin: Definitely.

David Heckerman: Definitely the great limiting step is the testing.

Carl Franklin: Yeah.

Richard Campbell: In some ways we're not even at the hard part yet. Once they make a vaccine and actually proving it works, that's not easy.

David Heckerman: Right, because of the waiting.

Richard Campbell: Yeah. There's just time involved in that.

David Heckerman: The way you don't prove a vaccine works is you give people a vaccine and then you give them HIV. That's not a good way to test a vaccine.

Richard Campbell: Not a good way, no.

Bob Davidson: It's not a good way and you can't really use the love numbers of HIV. You can't go give the vaccine to thousands let's say you do the safety testing first.

Richard Campbell: Yeah.

Carl Franklin: Right.

Bob Davidson: There are steps that have to be followed that just will take time.

Richard Campbell: Yeah, no two ways around it.

Carl Franklin: It's great though that you've taken such great steps. I mean, what can I say. There are millions of people out there who are thanking you.

Bob Davidson: We hope.

Richard Campbell: Other projects? I mean, I know you've taken on a good one here but I've got to think there's some other things that this can be applied to.

Carl Kadie: David.

David Heckerman: Well, certainly Hepatitis C, and it is being applied to Hepatitis C, those are the two big challenges. One other possible area of application is it turns out that at least in South Africa the main disease or the disease that's mostly to kill you once



you get HIV is tuberculosis. In section, it's rampant down there and when your immune system is weakened by HIV it's much easier to be killed by that disease. So one of our collaborators, Bruce Walker, is now setting up a whole new lab down there in South Africa and we hope to collaborate with him on that project as well.

Richard Campbell: Excellent, and TB sounds like it's under control except it really isn't.

David Heckerman: If you're otherwise healthy and if you get the normal kind of TB, there are drugs that can take care of it. But down where there is a lot of infection and weakened immune system, these bad forms of TB has propped up that are much more likely to kill you even if you get treated.

Carl Franklin: Well, we're coming up to the end of the show. Is there any last minute things that you guys want to say before we sign off?

David Heckerman: .NET rocks.

Carl Franklin: Yes it does and it may cure AIDS someday. That's great. Thank you, guys, for the great work you're doing.

David Heckerman: Thank you.

Carl Kadie: Thank you.

Bob Davidson: You're welcome. Have a great day.

Carl Franklin: And we'll see you next time on .NET Rocks!

[Music]

Carl Franklin: .NET Rocks! is recorded and produced by PWOP Productions, providing professional audio, audio mastering, video, post production, and podcasting services, online at www.pwop.com. .NET Rocks! is a production of Franklins.NET, training developers to work smarter and offering custom onsite classes in Microsoft development technology with expert developers, online at www.franklins.net. For more .NET Rocks! episodes and to subscribe to the podcast feeds, go to our website at www.dotnetrocks.com.