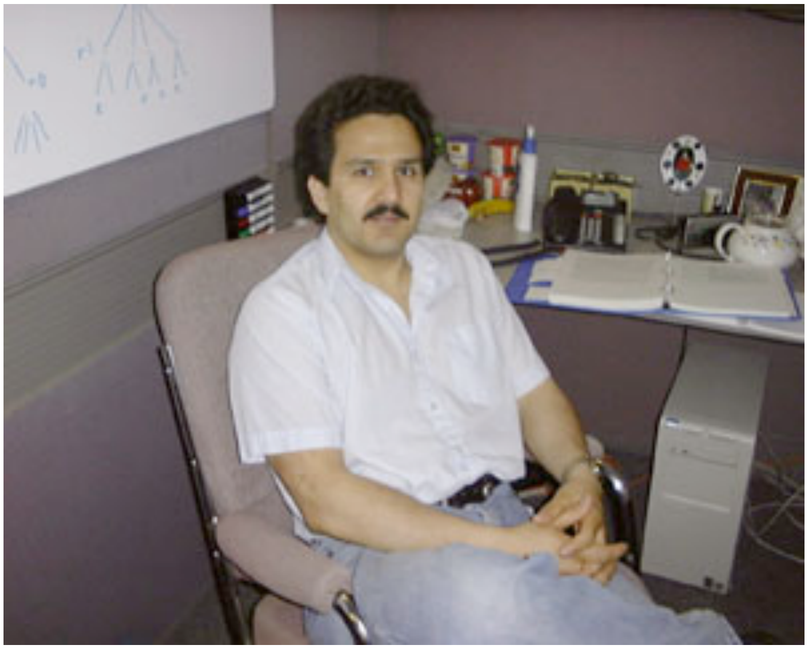
DARIUS NAQVI

I grew up in Regina, Saskatchewan. I received a B.Sc. in mathematics from the University of Regina in 1986 and an M.Sc. in mathematics from the University of Alberta in 1989. At that point, I felt I needed a break, and so I decided to work for a while before continuing my education. (As it turns out, I am either still on my break, or never going back to continue my education, depending on how one looks at it.)



For the most part, my degrees in mathematics have had an indirect effect on my career choices. What I've found is that studying mathematics causes one to be very precise and logical in one's thinking, and these are attributes that apply extremely well to programming computers.

So I did a sidestep and started working for a computer consulting company. I found that too business-oriented and not mathematical enough. After a couple of other jobs to which I had the same objection, I eventually found what I was looking for when I became a developer at a company, Myrias Computer Technologies Corp., that specialized in producing development and runtime environments for parallel programming. The employees of that company eventually became the development team of YottaYotta Inc., a company developing distributed and scalable network storage.

I found my calling

I can say that once I started software development involving parallel programming, I found my calling. For me, and possibly for many others who have studied mathematics, this is the right choice on the spectrum that ranges from applied to highly abstract. The logical thinking and ability to construct proofs that one learns when studying mathematics has come in very handy for me.

The thinking process involved in solving a difficult programming problem is indistinguishable from the thinking process involved in solving other types of mathematical problems or proving or theorems.

Courses in subjects like modern algebra and general topology teach one to visualize things that are very difficult to visualize. The type of thinking one learns in these courses helps when trying to visualize or create complicated programming constructs. There are instances where a mathematics degree applies more directly in the way I do my job. Whenever I am dealing with a difficult bit of coding, I find it very natural to construct a proof to show that my approach is correct, even if the proof is only in my head or on the back of an envelope.