

KLINT FINLEY BUSINESS 02.03.14 06:30 AM

OUT IN THE OPEN: MAN CREATES ONE PROGRAMMING LANGUAGE TO RULE THEM ALL



Stefan Karpinski, the co-creator of Julia. PHOTO: BRENDAN ADAMSON

STEFAN KARPINSKI WAS building a software tool that could simulate the behavior of wireless networks, and his code was a complete mess. But it wasn't his fault.

As a computer science grad student with years of industry experience under his belt, Karpinski was far from a programming novice. He knew how to build software. The problem was that in order to build his network simulation tool, he needed four different programming languages. No single language was suited to the task at hand, but using four languages complicated everything from writing the code to debugging it and patching it.

>Dubbed Julia, it provides an early glimpse into what programming languages might look like in the not too distant future

It's a common problem for programmers as well as mathematicians, researchers and data scientists. So Karpinski set out to solve it. He and several other computer scientists are building a new language they hope will be suited to practically any task. Dubbed [Julia](#), it

too-distant future.

Today's languages were each designed with different goals in mind. Matlab was built for matrix calculations, and it's great at linear algebra. The R language is meant for statistics. Ruby and Python are good general purpose languages, beloved by web developers because they make coding faster and easier. But they don't run as quickly as languages like C and Java. What we need, Karpinski realized after struggling to build his network simulation tool, is a single language that does everything well.

At one point, he vented his frustrations to Viral Shah, a fellow grad student at the University of California Santa Barbara. Shah introduced him to a computer scientist named Jeff Bezanson. It so happened that Bezanson had recently made a study of language design, and had come to the conclusion that the tradeoffs inherent in most languages were avoidable. "It became clear that a lot of it had been designed haphazardly," Bezanson says. "If you started from the beginning, you could recreate the things that people liked about those languages without so many of the problems."

Julia co-creator Jeff Bezanson.

PHOTO: BRIAN CHAN

Soon the team was building their dream language. MIT, where Bezanson is a graduate student, became an anchor for the project, with much of the work being done within computer scientist and mathematician Alan Edelman's research group. But development of the language remained completely distributed. "Jeff and I didn't actually meet until we'd been working on it for over a year, and Viral was in India the entire time," Karpinski says. "So the whole language was designed over email."

Together they fashioned a general purpose programming language that was also suited to advanced mathematics and statistics and could run at speeds rivaling C, the granddaddy of the programming world.

Programmers often use tools that translate slower languages like Ruby and Python into faster languages like Java or C. But that faster code must also be translated – or compiled, in programmer lingo – into code that the machine can understand. That adds more complexity and room for error.

Julia is different in that it doesn't need an intermediary step. Using [LLVM](#), a compiler developed by University of Illinois at Urbana-Champaign and enhanced by the likes of Apple and Google, Karpinski and company built the language so that it compiles straight to machine code on the fly, as it runs.

alternative to Hadoop, the widely used data crunching system developed at Yahoo and Facebook – at least in some cases. Hadoop lets you take a large problem, break it up into many small problems, and spread them across hundreds of machines. Julia is also designed parallelism.

“We never expected it to be a drop-in replacement for Hadoop, but we wanted to be able to do some of the stuff you do in Hadoop,” Karpinski says. “You can startup 100 Julia processes and run them on different machines, and fetch the results from other machines. That kind of thing tends to be tedious work in Java, but in Julia, it’s relatively straight forward.”

The first public version of Julia was released in early 2012. Many were skeptical about the need for yet another programming language, but enough people shared the frustrations of its creators that it has now begun to catch on with scientists.

Julia co-creator Viral Shah.

PHOTO: VIRAL SHAH

That said, it isn’t for everyone. Bezanson says it’s not exactly ideal for building desktop applications or operating systems, and though you can use it for web programming, it’s better suited to technical computing. But it’s still evolving, and according to Jonah Bloch-Johnson, a climate scientist at the University of Chicago who has been experimenting with Julia, it’s more robust than he expected. He says most of what he needs is already available in the language, and some of the code libraries, he adds, are better than what he can get from a seasoned language like Python.

Even if Julia never displaces the more popular languages – or if something better comes along – the team believes it’s changing the way people think about language design. It’s showing the world that one language can give you everything.

“People have assumed that we need both fast and slow languages,” Bezanson says. “I happen to believe that we don’t need slow languages.”

#ENTERPRISE #PROGRAMMING

[VIEW COMMENTS](#)

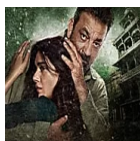
SPONSORED STORIES

POWERED BY OUTBRAIN



CRICINFO

Agarkar: India had two guys with the right temperament



AMAZON PRIME VIDEO

Sanjay Dutt Back in Action in Bhoomi. Watch Now!



BUYNUTRALYFE.COM

Are you frustrated with Hair loss ? This Ayurvedic Pill will solve your problem for sure



WWW.M.10DAYSHAIRDIL.COM

Getting your hair back was never so cheap! Get this magic lotion now at a steal deal price!



ICICI BANK

Get loan disbursal within 24 hours of application

MORE BUSINESS

VIRAL MEDIA

How Bored Panda Survived Facebook's Clickbait Purge

ERIN GRIFFITH

ARTIFICIAL INTELLIGENCE

Old Technique Could Put Artificial Intelligence in Hearing Aids

TOM SIMONITE

BUSINESS

Mapping the Future: Cartography Stages a Comeback

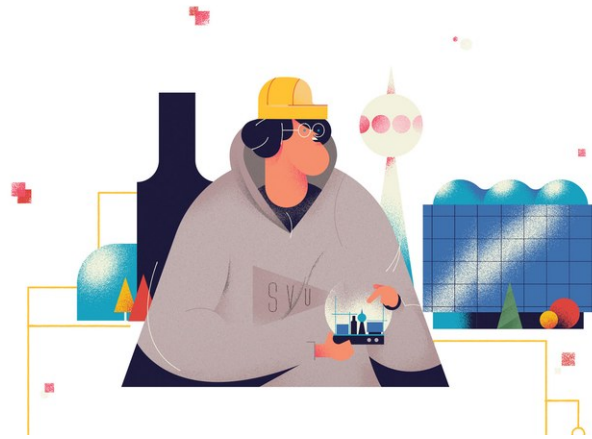
AUTOMATION

When Robots Invade the Kitchen

REBECCA FLINT MARX

Artificial Media: Not Every Message Has to Go Viral

CLIVE THOMPSON



CONSTRUCTION TECH

How Silicon Valley is Cracking the Construction Code

JAKE BULLINGER

GET OUR NEWSLETTER

WIRED's biggest stories delivered to your inbox.

Enter your email

SUBMIT

WE'RE ON PINTEREST

FOLLOW

LOGIN	SUBSCRIBE
ADVERTISE	SITE MAP
PRESS CENTER	FAQ
ACCESSIBILITY HELP	CUSTOMER CARE
CONTACT US	SECUREDROP
T-SHIRT COLLECTION	NEWSLETTER
WIRED STAFF	JOBS

RSS

CNMN Collection

Use of this site constitutes acceptance of our [user agreement](#) (effective 3/21/12) and [privacy policy](#) (effective 3/21/12). [Affiliate link policy](#). [Your California privacy rights](#). The material on this site may not be reproduced, distributed, transmitted, cached or

