



Search for

Go

InfoWorld **Julia** seeks to be the C for scientists

Developers seek to provide one dynamic language for high productivity, performance

By Paul Krill

Editor at Large, InfoWorld

APR 18, 2012

Julia is a new programming language being developed for building technical applications. In its early stages, Julia has been used in such applications as image analysis and linear algebra research. The language is the brainchild of developer Stefan Karpinski, along with Jeff Bezanson and Viral Shah. InfoWorld Editor at Large Paul Krill recently got the lowdown on Julia during an interview with Karpinski.

InfoWorld: When did you develop Julia, and what was the main intention in developing it?

Karpinski: We started talking in August 2009. Viral put Jeff and me in contact, and we started talking about our frustrations with technical computing languages. Primarily, one of them being that you just have to use so many different languages for different things because they're all good at one thing but not good at everything. You end up having to switch languages a lot.

[Discover the power of PowerShell: PowerShell intro for Windows Server admins • PowerShell intro for Exchange admins • Essential PowerShell scripts for security admins • All about PowerShell providers and modules. | Keep up with hot topics in programming with InfoWorld's App Dev Report newsletter.]

[Check out InfoWorld's recent interviews with the founders of [C++](#) and [Python](#). | Subscribe to [InfoWorld's Developer World newsletter](#) for the latest news on software development.]

InfoWorld: When you say technical computing, to what type of applications are you

specifically referring?

InfoWorld

...ry, but it's pretty much anything that involves a lot of number-crunching. In my own background, I've done a lot of linear algebra but a fair amount of statistics as well. The tool of choice for linear algebra tends to be Matlab. The tool of choice for statistics tends to be R, and I've used both of those a great deal. But they're not really interchangeable. If you want to do statistics in Matlab, it's frustrating. If you want to do linear algebra in R, it's frustrating.

InfoWorld: So you developed Julia with the intent to make it easier to build technical applications?

[Learn Java from beginning concepts to advanced design patterns in this comprehensive 12-part course!]

Karpinski: Yes. The idea is that it should be extremely high productivity. To that end, it's a dynamic language, so it's relatively easy to program, and it's got a very simple programming model. But it has extremely high performance, which cuts out [the need for] a third language [C], which is often [used] to get performance in any of these other languages. I should also mention NumPy, which is a contender for these areas. For Matlab, R, and NumPy, for all of these options, you need to at some point drop down into C to get performance. One of our goals explicitly is to have sufficiently good performance in Julia that you'd never have to drop down into C.

InfoWorld: The Julia Web page says Julia is designed for cloud computing and parallelism. What's the intent there?

Karpinski: The idea is that a lot of number-crunching these days ends up being done in the cloud environment, where you can allocate resources on demand. Traditional parallel computing for technical problems often use MPI [Message Passing Interface], a very, very established and popular way of doing large-scale parallel applications. But it doesn't have any facility for elasticity. If you want to add processors midcomputation, that can't really be done.

It also has limited functionality for recovery from faults, and that's a huge problem in

cloud computing because resources are not dedicated hardware, they're not big
InfoWorld that have extremely high reliability. They're virtual machine
here, and it's not uncommon for a resource to just disappear.

InfoWorld: How does Julia overcome these issues with cloud computing?

Karpinski: Well, these aren't solved problems. These are things that we're still working on, but we have a simpler model for building a large parallel applications via a global distributed address space. This means that you can hold a reference to an object that lives on another machine participating in a computation. You can manipulate references easily and pass them around between machines, making it easy to keep track of what's being computed where. Also, you can add machines in midcomputation. We haven't yet built the facility for removing machines midcomputation because we thought this was less of a pressing issue. But if you realize partway through a computation that you want more machines because you want the computation to go faster, you can do that.

InfoWorld: Would you use Julia for big data applications, data analysis, that sort of application?


Karpinski: It depends on what you mean by big data. "Big data" is a very overloaded term at this point. But I think a lot of, for example, predictive analysis, modeling problems are often linear algebra problems, and graph analysis problems are also something that Julia would be good at tackling. Certain kinds of applications are probably better addressed by a traditional technology like Hadoop. If you have a lot of things to count, Hadoop is perfect. If you want to compute a large matrix factorization, Hadoop is absolutely atrocious. That's the kind of place where Julia would be a stellar choice.

InfoWorld: You're not going to use Julia to develop, say, an accounting application or a mobile application, would you?

Karpinski: We actually have plans to develop a compiler. And once you have a compiler, you could write Julia code for embedded and mobile applications.

InfoWorld: When would that compiler be ready?

Karpinski: Probably sometime in the next year.

 InfoWorld: The Web page says, "Julia's LLVM-based just-in-time compiler combined with it to approach and often match the performance of C, C++." What was your intent in approaching or matching the performance of C, C++?

Karpinski: C, C++, and I should also add Fortran, because it's another old compiled language and is known for its performance. Those are the gold standard, that's how you measure performance. If you can match or beat C++, you've made it.

InfoWorld: What's the exact status of Julia at this point?

Karpinski: It's a pre-1.0. We're shooting for 1.0 soon. [Version] 1.0 is a big commitment. To me, at least, and to the other core developers, it means that we're committing to a stable API.

InfoWorld: And this is available under an MIT license, correct?

Karpinski: Yes. The core is available under MIT, which is very permissive.

InfoWorld: Why the name, Julia?

Karpinski: That's everybody's favorite question. There's no good reason, really. It just seemed like a pretty name.

This article, "[New Julia language seeks to be the C for scientists](#)," was originally published at [InfoWorld.com](#). Follow the [latest developments in business technology news](#) and get a digest of the key stories each day in the [InfoWorld Daily newsletter](#). For the latest developments in business technology news, follow [InfoWorld.com on Twitter](#).

Paul Krill is an editor at large at InfoWorld, whose coverage focuses on application development.

Follow     

YOU MIGHT LIKE

Ads by Revcontent



InfoWorld

**Gut Health
with One**

Gundry MD

**Almost Every
Mac User**

MacKeeper

**Small Bitcoin
Investment
That's Making**

LifeDaily

**Bengaluru:
The Bitcoin
Casino That's**

LifeDaily

**New Site Finds
the Cheapest
Flights in**

FlightFinder

**These
Perfectly
Timed Photos**

bradofo.com/blog/

**He
Transformed
His Gut with**

Gundry MD

**The Bitcoin
Casino That's
Making**

LifeDaily

**If You Haven't
Invested at
Least \$10 into**

LifeDaily

**Indians Are
Going Crazy
With This**

LifeDaily