

JAVA 9: How Java 9 will change your life (<https://jaxenter.com/how-java-9-change-life-139019.html>)

Can Julia give us everything?

November 11, 2016  Gabriela Motroc

 19   8  Facebook  Google+

(<http://www.reddit.com/submissions?mini=true&url=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>) (<https://www.linkedin.com/sharing/share-offsite/?url=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>) (<https://www.facebook.com/sharer/sharer.php?u=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>) (<https://plus.google.com/share?url=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>)

This programming language compiles straight to machine code as it runs. It seems to be a valid alternative to Hadoop and it might follow in the footsteps of C and Java. Does Julia, a four-year-old programming language, have what it takes to compete with the industry's finest?



Julia - Wooden 3D rendered letters/message (http://www.shutterstock.com/pic-508320295/stock-photo-julia-wooden-3d-rendered-lettersmessage-can-be-used-for-an-online-banner-ad-or-a-print-postcard.html?src=k_hi2nhtnZgRdgmITyKCCA-1-47) image via Shutterstock

Last year, Viral Shah, one of Julia's creators, told *Venture Beatt* (<http://venturebeat.com/2015/05/18/why-the-creators-of-the-julia-programming-language-just-launched-a-startup/>) that even though big tech companies have not adopted Julia because “they tend to have their own programming languages,” the team was seeing rapid adoption in finance and financial modeling services, and also in business logistics and companies interested in high-performance computing (HPC). One year later, Keith Lubell, CTO at Berkery Noyes, a New York-based investment bank told *Waters Technology* (<http://www.waterstechnology.com/waters/feature/2476518/the-infancy-of-julia-an-inside-look-at-how-traders-and-economists-are-using-the-julia-programming-language>) that Wall Street is starting to notice Julia.

Julia's biggest fan — Federal Reserve Bank of New York

Julia has gathered quite an impressive portfolio of positive reviews (<http://juliacomputing.com/press>). If you take a look at Julia Computing, a consultancy which promises to help firms looking to use the Julia language, you will see a lot of praise with regard to this programming language's speed, high performance, and flexibility. More importantly, the Federal Reserve Bank of New York chose to run its DSGE model (dynamic stochastic general equilibrium) —used for forecasting and policy analysis— on Julia and ditch MATLAB. In the official statement (<http://libertystreeteconomics.newyorkfed.org/2015/12/the-frbny-dsge-model-meets-julia.html>) announcing the shift to Julia, FRBNY stated that even though MATLAB is “a widely used, mature programming language” which has served their purposes very well for a long time, Julia has some irresistible benefits such as:

- it is more accessible to users from academic institutions or organizations without the resources for purchasing a license
- it boasts (<http://julia-lang.org/benchmarks/>) performance as fast as that of languages like C or Fortran
- it is still simple to learn

According to FRBNY, after testing their code, they discovered that “the model estimation is about ten times faster with Julia than before, a very large improvement.”

What is Julia and why was it created?

In a blog post (<http://julia-lang.org/blog/2012/02/why-we-created-julia>) published four years ago by the creators of this programming language, Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral Shah revealed that every language language they used had a trade-off so they decided to develop something that has it all (<http://julia-lang.org/>).

We want a language that's open source, with a liberal license. We want the speed of C with the dynamism of Ruby. We want a language that's homoiconic, with true macros like Lisp, but with obvious, familiar mathematical notation like Matlab. We want something as usable for general programming as Python, as easy for statistics as R, as natural for string processing as Perl, as powerful for linear algebra as Matlab, as good at gluing programs together as the shell. Something that is dirt simple to learn, yet keeps the most serious hackers happy. We want it interactive and we want it compiled.

(Did we mention it should be as fast as C?)

While we're being demanding, we want something that provides the distributed power of Hadoop — without the kilobytes of boilerplate Java and XML; without being forced to sift through gigabytes of log files on hundreds of machines to find our bugs. We want the power without the layers of impenetrable complexity. We want to write simple scalar loops that compile down to tight machine code using just the registers on a single CPU. We want to write $A*B$ and launch a thousand computations on a thousand machines, calculating a vast matrix product together.

Julia's biggest problem right now

— the community. Julia is four years old and even though it is starting to receive attention from the financial industry, it has failed to grab developers' attention. Julia's community is rather small but that is fully understandable since its rivals have had decades to loyalize users.

Waters Technology cited Lubell as saying that “there is a fear that a language with a larger user base like Python could potentially improve its performance, and then leave Julia in the dust.”

What's next for Julia

The creators of this programming language have been busy working on version 0.5 (<http://julialang.org/blog/2016/10/julia-0.5-highlights>), which was released last month. This “pivotal release introduced more transformative features than any release since the first official version —several of these features set the stage for even more to come in the lead up to Julia 1.0 (<https://www.youtube.com/watch?v=5gXmpY1kJY>).”

There are now over one thousand (<http://pkg.julialang.org/pulse.html>) registered packages but if this programming language wants to win developers' hearts, it will have to step up its game.

Be the first to share this article with your network!

 19
 
 8
  Facebook
  Google+

<https://www.reddit.com/submissions?cc=1&url=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>
<https://www.linkedin.com/sharing/share-offsite?url=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>
<https://www.facebook.com/sharer.php?u=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>
<https://plus.google.com/share?url=https://jaxenter.com/julia-general-purpose-programming-language-130136.html>

general-purpose-programming-language-130136.html |
 Julia give us everything? |
 general-purpose-programming-language-130136.html |
 general-purpose-programming-language-130136.html |
 general-purpose-programming-language-130136.html


 Author: Gabriela Motroc (@GabrielaMotroc) <https://twitter.com/GabrielaMotroc>

All Posts by Gabriela Motroc (<https://jaxenter.com/author/gabrielamotroc>)

Gabriela Motroc is an online editor for JAXenter.com. Before working at S&S Media she studied International Communication Management at The Hague University of Applied Sciences.

Recommended For You

Comments

3 Comments JAXenter

Login

Recommend Share

Sort by Best

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Nick Evgeniev · a year ago

what's the purpose of comparing pre-1.0 language to mainstream? For example I'm aware of julia since 2014 but not going to look into it till 1.0 release plus ide support (rstudio, vs, idea whatever comes first)

Reply Share

Miloslav Raus · a year ago

No chance of Python ever catchin up to Julia. Python is too dynamic / 'convenient' for it's own good (too many things are dictionaries). And even in absence of annotations, Julia - thanks to delayed compilation (harvesting type info at worst 'just in time') - produces type-specialized native code.

Reply Share



Jesse · a year ago

The keyword here is mainstream, Julia needs to achieve that before it can be adopted anywhere. Also Julia started out as a scientific data language, why on this green Earth does everyone from Python crowd think data science just use Haskell. Oh right because Python was originally setup with that mindset, but angry protest lead to the creation of languages like Julia. Idunno I'm not a programmer or even a data scientist for that matter. However if I had to choose a language it would be Java, if it can be reasoned and written without grey hair I'm certain it works.

Reply Share

Subscribe Add Disqus to your site Add Disqus Add Privacy

PDF MAGAZINE

Open Source 101: Everything you need to know about open source

free download



(http://jaxenter.com/jax-magazine)

MEET US IN LONDON!



jax
DEVOPS

The Conference for
Continuous Delivery,
Microservices, Docker
& Clouds.

April 9 – 12, 2018 | London
Expo: April 10 – 11, 2018
devops.jaxlondon.com

**Only till
December 7**
✓ Free Workshop Day
✓ Save up to £ 700

(https://devops.jaxlondon.com?utm_medium=banner&utm_source=jaxenter.com&utm_campaign=sands_om&utm_content=sidebar_ad)

FEATURED POSTS



Who will use JDK 9?
(<https://jaxenter.com/who-will-use-jdk-9-139200.html>)



The new workspace currency is open source
(<https://jaxenter.com/new-workspace-currency-open-source-139217.html>)