

JULIA E AS SUAS FUNCIONALIDADES

EDUARDA TATIANE CAETANO CHAGAS

2016

Resumo

Tal relatório possui como objetivo primordial o estudo da relação existente entre a programação voltada para meios científicos e a linguagem Julia, que consiste em uma síntese das principais linguagens do ramo, mostrando como suas propriedades e características facilitam determinado processo.

Deste modo, introduzimos os principais problemas tratados por intermédio da computação científica, que constitui o elemento base de nosso estudo, pois análise da linguagem foi realizada fundamentando-se em quais requisitos básicos esta deveria possuir para suprir as necessidades do setor.

Logo após foi exposto seus principais atributos e o que a destaca em meio a tantas outras. Também foi desmonstrado como realizar o primeiro contato com a programação em tal linguagem, facilitando a introdução do leitor nesta nova jornada.

Por último foi verificado quais efeitos as características de tal ferramenta iriam realizar sobre a computação científica, tendo em vista que esta dedica-se a formulações de padrões e experimentos matemáticos.

Sumário

1	METODOLOGIA	2
2	INTRODUÇÃO	3
3	OBJETIVO GERAL	4
3.1	OBJETIVOS ESPECÍFICOS	4
4	DESENVOLVIMENTO	5
4.1	PROBLEMATIZAÇÃO	5
4.2	SURGIMENTO	5
4.3	CARACTERÍSTICAS DIFERENCIAIS	5
4.4	VANTAGENS EM RELAÇÃO AS DEMAIS	6
4.5	ITERAGINDO COM JULIA	7
4.5.1	SEÇÃO INTERATIVA	7
4.5.2	SEÇÃO ESTÁTICA	8
5	RESULTADOS E CONCLUSÕES	9

Capítulo 1

METODOLOGIA

A metodologia consiste em um processo de pesquisa teórica, onde foi realizada a análise de documentações e artigos relacionados a temática, e por meio da investigação e dedução foi desenvolvido tal relatório, com o objetivo de determinar a relação existente entre a nova linguagem de programação Julia e o ramo da programação científica.

Após a procura em sites relacionados com tecnologias e programação, encontraram-se um conjunto de depoimentos a cerca de experiências posteriores referentes a programação utilizando determinado processo, assim como também análises e comparações elaboradas por profissionais da área estudada.

Sendo estudados as seguintes publicações destes sites:

- http://julia-pt-br.readthedocs.org/pt_BR/latest/manual/introduction.html
- <https://e17aeternus.wordpress.com/2015/02/26/algumas-impressoes-sobre-a-linguagem-julia/>
- <http://www.cienciaedados.com/tag/julia/>
- <http://gizmodo.uol.com.br/julia-linguagem-programacao/>

foi fundamentando-se o conceito de tal relação (Julia e a computação científica) e como esta ocorria.

Por meio da análise da própria documentação da linguagem tornou-se conhecimento sobre suas verdadeiras finalidades que esta proporciona e as opções que esta passa a fornecer a seus usuários.

E por final foram expostas os resultados e conclusões obtidos por meio do estudo desta relação biunívoca entre a linguagem e a ciência.

Capítulo 2

INTRODUÇÃO

A computação científica baseia-se no processo de análise e resolução de problemas científicos, através da construção de modelos e técnicas matemáticas utilizando meios computacionais. Possuindo uma ampla área de atuação, podemos encontrá-la no processo de reconhecimento de padrões, simulação computacional, modelagem financeira, física matemática, mecânica computacional e outras.

Dentre os seus problemas mais recorrentes destacam-se: simulação numérica, que trata do projeto de um modelo computacional de um sistema real; e análise de dados, que por sua vez consiste no estudo dos dados de um problema e a identificação de padrões existentes.

Por intermédio de linguagens de programação dinâmicas e de alto desempenho, se faz possível o estudo da computação científica, sendo por meio destas programados os softwares necessários para a efetuação de seus projetos.

Neste relatório apresentamos a linguagem Julia, que trata-se de um novo meio de realização dos cálculos e técnicas da programação científica, expondo suas principais características, semelhanças e diferenças entre as demais linguagens específicas da área e como iniciar o processo de programação em tal linguagem mostrando como utilizar sua seção interativa, como compilar e executar o seu primeiro código.

Capítulo 3

OBJETIVO GERAL

Transmitir as principais vantagens da utilização de Julia como linguagem base de um projeto científico, através da análise das suas principais características e das diferenças entre as demais linguagens da área, facilitando assim o começo do processo de inserção neste novo universo.

3.1 OBJETIVOS ESPECÍFICOS

- Estimular a utilização da linguagem Julia em procedimentos científicos;
- Ampliar os conhecimentos a cerca desta nova linguagem de programação.

Capítulo 4

DESENVOLVIMENTO

4.1 PROBLEMATIZAÇÃO

Com o surgimento de problemas cada vez mais complexos no mundo científico, se faz cada vez mais necessário a implementação da automatização de determinados procedimentos. A realização do cálculo de séries temporais com milhares de elementos, baseadas no mapa logístico, por exemplo transforma-se em algo extremamente mais simples e prático com o adjunto das técnicas de programação.

Em meio a linguagens com alto desempenho, extremamente rápidas e voltadas exclusivamente para solucionar problemas científicos, encontramos Julia, uma linguagem relativa nova, criada em 2012, e que vem ganhando cada vez mais adeptos, devido a seu modo inovador de reunir as melhores características de uma série de outras linguagens adjacentes.

4.2 SURGIMENTO

Desenvolvida por Jeff Bezanson, Stefan Karpinski, Viral B. Shah e Alan Edelman, Julia surgiu desde o seu princípio com o objetivo de facilitar o processo de programação dos profissionais do campo científico.

Ao se deparar com a necessidade de utilização de quatro linguagens diferentes de programação para a construção de uma única ferramenta de simulação para redes, Stefan Karpinski defrontou-se com um problema típico na vida de inúmeros programadores: a inexistência de uma única linguagem capaz de abranger a maior partes das funcionalidades exigidas pelos seus softwares.

Partindo de tal pressuposto, a partir do ano de 2009 a linguagem começou a ser desenvolvida, sendo a primeira versão do código aberto divulgada em 2012.

Mesmo possuindo como premissa a necessidade de ser uma linguagem extremamente abrangente, possuindo utilizações em desenvolvimento web e em programação em geral, esta encontra-se voltada para a área de Ciência de dados, isto é, colaborando imensamente nos requisitos dos projetos de computação numérica e científica.

Percebendo que uma das grandes carências da área tratava-se da inexistência de uma linguagem de programação dinâmica de alta performance e alto nível, pois esta seria em sua boa parte utilizadas por cientistas em suas pesquisas e não somente por programadores, escreveram nas linguagens C, C++ e Scheme, Julia que segundo os seus próprios desenvolvedores anseia criar uma combinação de uso, força e eficiência.

4.3 CARACTERÍSTICAS DIFERENCIAIS

Diante da existência de dezenas de linguagens disponíveis, tornam-se dificultosos os processos de escolha para o nosso projeto. Entretanto, é a partir das suas características predominantes que determinamos a nossa preferência.

Em Julia as características que as definem e diferenciam são:

- Bom desempenho: Um dos principais fatores que colaboram o seu alto desempenho é a utilização da compilação Just-in-time. Com a compilação Just-in-time, também denominada de Tradução dinâmica, se faz possível a compilação de uma determinado programa em tempo real, pois esta utiliza o processo de tradução de um código, antes e durante a execução deste. Ela possui não apenas um tradutor, mas também uma máquina virtual responsável por transformar o código em Julia para Bitcode, e só assim transformá-lo por meio do tradutor, na linguagem de máquina específica. Outro fator importante refere-se a aplicação da estrutura LLVM (Low Level Virtual Machine), que trata-se de uma infraestrutura para compiladores que objetiva otimizar os programas em termos de compilação e execução do código.
- Livre e Open source: Desenvolvida com o apoio do MIT, segundo seus criadores a finalidade é a criação de uma linguagem liberal, de livre acesso a todos, facilitando cada vez mais o seu uso.
- Multiparadigma: Suporta mais de um paradigma da programação, possibilitando o programador a atuar em diversos estilos, atuando na que mais lhe agrada, seja na programação imperativa, funcional ou orientada a objetos.
- Programação paralela e distribuída: Além de apenas tal requisitos torná-la uma linguagem mais interessante e de ampla utilização, esta destaca-se como uma das mais rápidas do setor, assim como citou Karpinski, um de seus criadores:

Pode-se lançar 100 processos em Julia, executados em máquinas diferentes e obter resultados dessas máquinas. Esse tipo de coisas tende a ser um trabalho tedioso em Java, mas em Julia é relativamente simples.

- Sistema sofisticado: Contando com uma linguagem de altíssimo nível, seu uso torna-se simples, agradável e descomplicado, facilitando a inclusão de usuários que almejam a automação de seus projetos ou apenas possuir uma maior visualização de seus cálculos, por meio de histogramas, mesmo não possuindo grandes conhecimentos na área da computação.

4.4 VANTAGENS EM RELAÇÃO AS DEMAIS

Elaborada a partir de características notáveis de outras linguagens, Julia possui um grande semelhança entre elas. Como mencionado pelos seus desenvolvedores:

Queremos a velocidade do C com o dinamismo do Ruby. Queremos uma linguagem que é homoicônica, com macros verdadeiros como Lisp, mas com uma notação matemática familiar como o Matlab. Queremos algo usável para a programação geral como Python, fácil de usar para estatística como R, natural para processamento de strings como Perl, poderoso para álgebra linear como Matlab, bom para juntar programas como Shell. Algo extremamente simples para aprender, e que ainda deixe os hackers mais sérios felizes. Queremos isso interativamente e queremos isso compilado.

Logo, torna-se perceptível que Julia não ofereceu algo muito inovador para o mundo da programação, referindo-se a sua sintaxe, este apenas reuniu os melhores atributos de cada linguagem, fornecendo ao usuário algo mais coerente em relação aos diversos ramos da computação.

Assim, as linguagens que possuem uma maior similaridade com esta, seja na sua estrutura sintática, no desempenho ou na área de atuação, são:

- Python: Possuindo grandes semelhanças sintáticas, Python e Julia possuem maiores semelhanças que desigualdades, sendo as mais significativas:
 - Ambas atuam diretamente na computação científica;
 - Possuem alto desempenho, embora Julia ultrapasse neste requisito;
 - São semelhantes quanto a sintaxe e ao design, mas usando Julia é possível elaborar algoritmos relativamente mais sofisticados;

- Podem ser conectadas a linguagens de alto nível, seja diretamente, seja utilizando ferramentas já existentes;
- Ambas possuem uma interface em relação a computação paralela e distribuída;
- Lisp: Inspirando-se em vários de seus dialetos como Dylan e Common Lisp, Julia não somente explorou algumas das características de tal linguagem, como também utilizou-se como inspiração, sendo construída a partir de suas heranças. Logo torna-se fácil perceber algumas de suas maiores similaridades, como: - Possuem macros de configurações semelhantes, isto é, possuem regras semelhantes quanto a especificação de como uma certa entrada deve ser mapeada para uma substituição de sequência de saída; - Possuem algumas expressões semelhantes, logo podemos afirmar que ocorreu uma certa herança sintática de Julia em relação a Lisp.
- Matlab: Esta linguagem possui um grande poder computacional em relação a experimentações e utilização de álgebra linear numérica, sendo deste modo, extremamente importante na computação e análise numérica. Possuindo como objetivo central a monopolização no meio científico, Julia então manteu a eloquência e facilidade do Matlab. No entanto, foi capaz de ultrapassar as suas limitações em relação a programação em âmbito geral, sendo seu desempenho muitas vezes comparado ao adquirido somente com a utilização das linguagens estáticas.
- R: Sua maior contribuição trata-se da disponibilização de ferramentas necessárias para o manuseamento de dados tabulares, facilitando a implementação de banco de dados. Logo com a aquisição de tal propriedade Julia consegue assim atrair também a atenção dos usuários desta ampla tecnologia, expandindo assim sua área de atuação, mesmo mantendo fielmente seu público-alvo.

4.5 ITERAGINDO COM JULIA

Após baixar e realizar o processo de instalação da linguagem Julia, seguindo as instruções fornecidas em <http://julialang.org/downloads/>, é possível relacionar-se com seu sistema tanto de modo estático, quanto interativo.

Caso o objetivo seja aprender tanto a linguagem, como o funcionamento de certa função, o meio mais recomendado seria a utilização do seu modo interativo, também conhecido como read-eval-print, ou simplesmente pela sua abreviação repl. Entretanto, o modo estático é o meio mais indicado no processo de compilação de códigos.

4.5.1 SEÇÃO INTERATIVA

Ao utilizar a linguagem, percebe-se que esta inicia automaticamente em seu modo interativo, aguardando apenas algum contato por parte do usuário. Neste momento é possível recrear-se com tal funcionalidade. Para que a linguagem exerça a funcionalidade de calculadora é necessário apenas:

- Digitar a expressão desejada;
- Pressionar o botão ENTER;

Após tais passos é possível visualizar logo abaixo o seu resultado final.

Entretanto para que o resultado de determinada expressão não seja visualizado é preciso apenas adicionar o ponto e vírgula em seu final. Desse modo o seu valor é depositado na variável `ans`, que possui a função de armazenar o resultado da última expressão divulgada, tendo este sido mostrado ou não. Também é possível manipular a variável `ans`, podendo realizar qualquer tipo de operação com esta, considerando-a como uma simples variável.

4.5.2 SEÇÃO ESTÁTICA

Para sair da seção interativa é necessário apenas:

- Pressionar a tecla `CONTROL`;
- Pressionar a tecla `d`;

ou

- Digitar `quit()`;

Assim, a seção estática é iniciada, podendo agora apenas compilar códigos presentes em outros arquivos.

No entanto, também é possível compilar e executar o código em uma seção interativa. Para que este procedimento seja realizado é preciso apenas digitar `include("file.jl")`, sendo `file.jl` o nome do arquivo em Julia que se deseja compilar e executar. Tal arquivo pode ser editado e escrito em qualquer editor de texto, no entanto existe a sua própria IDE, a chamada `JUNO`.

Para compilar e executar um determinado arquivo em uma seção estática, ou não-interativa é preciso apenas realizar uma chamada `Julia` e repassar como argumento o nome do arquivo desejado. Desse modo após o processo de compilação o código é executado. Logo, é necessário apenas escrever o seguinte:

- `$ julia file.jl`

Caso o programa necessite de alguma variável logo no início de sua execução é possível repassá-la no momento em que o arquivo é chamado, isto é:

- `$ julia file.jl arg1,arg2,...`

Assim os argumentos serão passados para o programa por meio da variável global `args`.

Capítulo 5

RESULTADOS E CONCLUSÕES

Tendo em vista os aspectos relatados, deduzimos que diante da grande variedade de linguagens disponíveis Julia torna-se a opção mais satisfatória e completa, pois possui uma estrutura definida exatamente para determinada área de conhecimento.

Após a análise dos fundamentos tratados na computação científica, expomos as propriedades presentes em tal linguagem que facilitam o processo de resolução de cálculos e procedimentos da área. Enquanto que as diferenças entre as demais demonstram o porquê da linguagem ser a mais recomendada para trabalhos técnicos.

Logo, levamos a acreditar que diante do observado Julia trata-se da linguagem não só mais prática, como também mais preparada para tais atividades, como por exemplo a análise de séries temporais, devido ao alto desempenho e a sua extensa bibliotecas de funções matemáticas.

Referências Bibliográficas

- [1] Algumas impressões sobre a linguagem julia, disponível em. <https://e17aeternus.wordpress.com/2015/02/26/algumas-impressoes-sobre-a-linguagem-julia/>.
- [2] A ambiciosa linguagem de programação que quer substituir python, r e matlab, disponível em. <http://gizmodo.uol.com.br/julia-linguagem-programacao/>.
- [3] Julia, a princesinha da ciência de dado, disponível em. <http://www.cienciaedados.com/tag/julia/>.
- [4] Manual de julia documentação, disponível em. http://julia-pt-br.readthedocs.org/pt_BR/latest/manual/introduction.html.