julia

Nathan Tippy
OCI

Julia Language highlights

- Targeting Technical/Scientific community
- High performance, JIT to LLVM
- Built in parallelism for cloud deployments
- Multiple dispatch
- Pure functions / Immutable structs
- Custom parameterized types
- Metaprogramming and Macros

Benchmarks on julialang.org

Julia (0.2)
Fortran (gcc 4.8.1)
Go (go1)
JavaScript (V8 3.7.12.22)
Python (2.7.3)
Mathematica (8.0)
R (3.0.2)
Matlab (R2012a)
Octave (3.6.4)

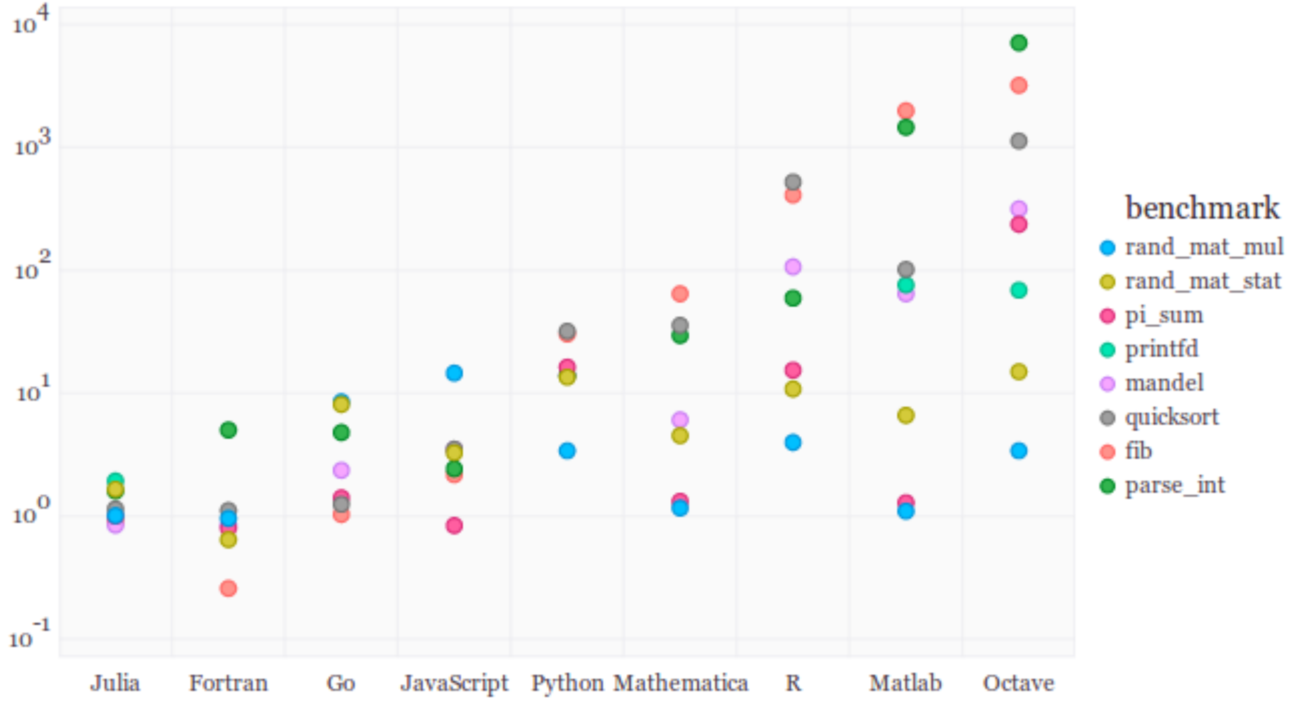

Figure: benchmark times relative to C (smaller is better, C performance = 1.0).

C compiled by gcc 4.8.1, taking best timing from all optimization levels (-O0 through -O3). C, Fortran and Julia use OpenBLAS v0.2.8. The Python implementations of `rand_mat_stat` and `rand_mat_mul` use NumPy (v1.6.1) functions; the rest are pure Python implementations. Plot created with Gadfly and IJulia from this notebook.

Getting started

- MIT License
- http://julialang.org/downloads/
  - Current Release v0.2.0
  - Use nightly build at your own risk
- Active google groups
  - julia-users
  - julia-dev
- https://github.com/JuliaLang/julia

Basics

Conventions

- Variables are lower case with _ between words
- Types are capitalized with CamelCase
- Functions are all lower case
- Mutating ‘in-place’ functions end with !

Other items of note

- Arrays are 1 based
- x^2 is x squared, a$b is a xor b
- f(x)=2x+3x^3; f2(x)=2x+3x.^3;

Standard Types

- Float
  - Float64, Float32, Float16, Char, Bool
- Int
  - Uint128, Uint64, Uint32, Uint16, Uint8
  - Int128, Int64, Int32, Int16, Int8
- Complex
  - Complex128, Complex64
- BigInt, BigFloat (GNU MPFR Lib)
- Rational
  - 2//6 == 1//3; num(2//6) == 1
- String

Standard library support for...

Linear Algebra / BLAS

Distributed Arrays / Parallel computing

Numerical Integration

Signal Processing

Combinatorics

Statistics

Multiple dispatch

- Methods called are based on the arguments and the module in use.
- Some helpful methods
  - apropos(“type”)
  - methods(function)

Enough Slides ...

On with the examples

- REPL
- BASH
- JuliaStudio (Forio)

Active work

- Static compile of Julia to exe
- Shorten startup time

Resources

http://julialang.org/

http://www.youtube.com/user/JuliaLanguage

https://github.com/JuliaLang/julia

http://docs.julialang.org/en/release-0.2/