Julia Vision

Jeff Bezanson, Stefan Karpinski, Viral B. Shah & Alan Edelman

Text Processing

Was once a niche language application

- **niche languages:** SNOBOL, sed, awk, early Perl (before 5)
- **general purpose:** Python, Perl 5, Ruby

“Gang of Forty”

Matlab Maple Mathematica SciPy SciLab IDL R
Octave S-PLUS SAS J APL Maxima Mathcad
Axiom Sage Lush Ch LabView O-Matrix PV-WAVE
Igor Pro OriginLab FreeMat Yorick GAUSS MuPad
Genius SciRuby Ox Stata JLab Magma Euler Rlab
Speakeasy GDL Nickle gretl ana Torch7

Mathematical Programming

“Technical computing” is still a niche

- Fortran – the first compiled language, targeted this niche

Can we make …

- a general purpose language
- powerful enough to subsume this niche?

  [power = performance / effort]

Two Language Problem

People love dynamic environments

- for data analysis and exploration
- but dynamism and performance are at odds

A standard compromise:

- high-level logic in convenient, dynamic language (Matlab, Python, R)
- performance-critical code in static, low-level language (C, C++, Fortran)

Creates a huge barrier to development

- continually breaking the abstraction barrier indicates a poor abstraction

Example: Creating a Basic Numeric Type

We’re going to create a **modular integer type** that is

- fully functional
- fully integrated
- fast & compact

… in 11 lines of code.

Built in = User-Defined

In most languages, numeric types are built in

- defined in low-level language
- new ones can’t be added

In Julia, numeric types are defined in Julia code

- they just happen to be defined before you start
- user-defined types are just as good

    equally fast, equally compact, fully first-class citizens

Hacking the Core is Easy

Newcomers have added major pieces:

- BitArrays
- SubArrays
- Distributions
- DataFrames

What requires major surgery in many systems is easy

- changing core arithmetic behaviors (e.g. overflow, promotion)
- adding new “core” types: arithmetic types, string types, etc.

Community

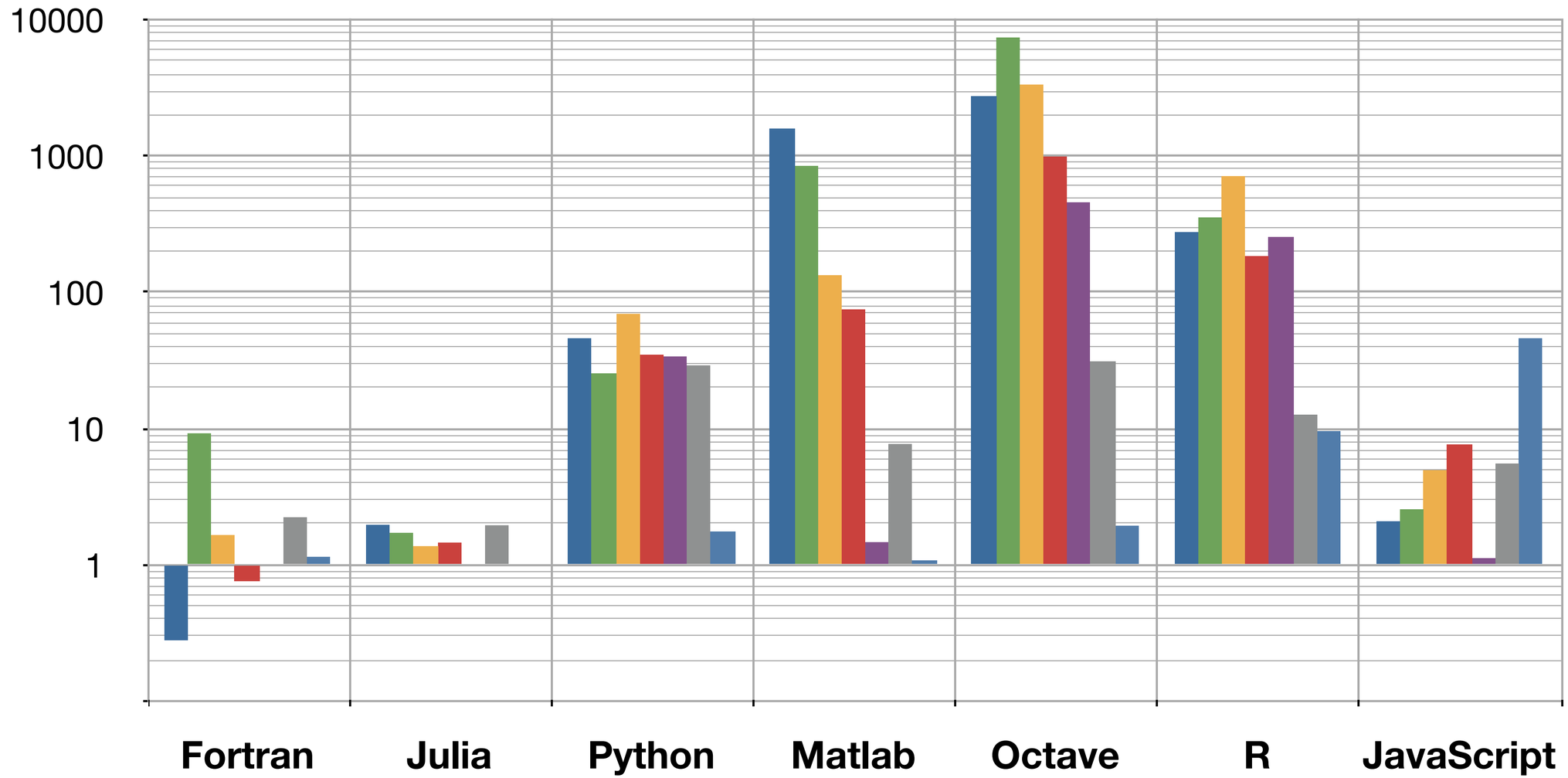
100+ contributors to base Julia

750+ mailing list subscribers

60+ packages

ArgParse Benchmark Cairo Calculus Calendar Clustering Color Compose
DataFrames Debug DecisionTree Distributions Example FastaRead FileFind
FITSIO Gadfly GetC GLM GLUT Graphs Grid HDF5 HTTP HypothesisTests
ICU IniFile Iterators Ito JSON kNN Languages LM Loss MAT MCMC Mongrel2
Named NHST ODBC OpenGL Optim Options PatternDispatch PLX Profile
ProjectTemplate RDatasets Resampling Rif SDL Sims Sound TextAnalysis
TextWrap Thyme Tk Trie UTF16 Winston ZMQ

Obligatory Performance Slide



execution time relative to C++ (lower is better)