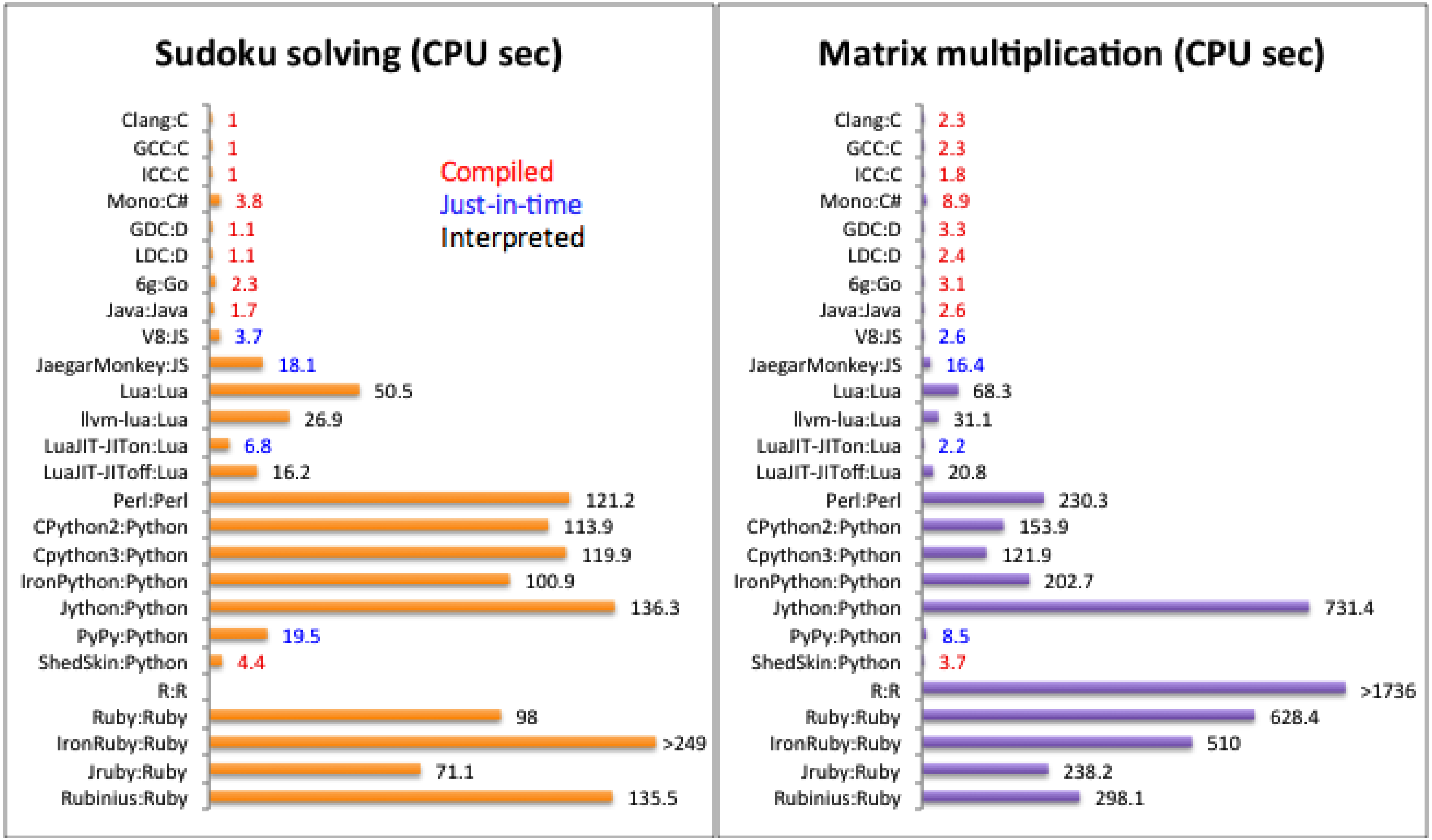
What is Python

- Python is a high-level programming language
  - easy to read and compact code
  - fast prototyping (good for research & development)
- Python is convenient for machine learning
  - reading data from files
  - training and applying models (e.g. decision trees)
  - plotting data and results

Limitation of Python

- Python is an interpreted language, and thus slower than a compiled language like C or Fortran.
- *Why* are interpreted languages slow?
  - Read and compiled by another program (interpreter) before being executed by the machine (10-100x slowdown).

Interpreted vs. Compiled Languages



Source: https://attractivechaos.github.io/plb/

Writing Fast ML Algorithms

- Use a different programming language (e.g. C or Fortran)
- Use Python bindings with low-level languages (e.g. F2PY)
- Use an optimized library that provides some useful operations (e.g. matrix multiplication) directly in Python

Numpy/Scipy

- Library that provides fast methods to manipulate arrays and perform a wide range of operations on arrays.
- Examples of array operations useful in machine learning:
  - Linear projections, distance matrices, eigenvalue decompositions
- Characteristics of Numpy/Scipy:
  - **In surface:** intuitive user interface for manipulating arrays
  - **Under the hood:** optimized code based on high performance libraries (BLAS, Lapack, etc)

Example: Matrix Multiplication

$Z = X \cdot Y$ $\quad X \in \mathbb{R}^{m \times n}$ $\quad Y \in \mathbb{R}^{n \times o}$ $\quad Z \in \mathbb{R}^{m \times o}$

Python multiplication

```
for i in range(m):
    for j in range(n):
        for k in range(o):
            Z[i,k] += X[i,j]*y[j,k]
```

Numpy multiplication

```
import numpy

Z = numpy.dot(X,Y)
```

+ Computational speedup

+ Terser syntax (no loops)

- limited set of operations implemented in Numpy