Computational Astrophysics with Go

Pramod Gupta

Department of Astronomy, University of Washington

Computational Astrophysics Code

- Common pattern in computational astrophysics code:
- Many iterations over multiple loops
- Performance is important
- For example:
- N-body simulations
- Monte Carlo Radiative Transfer

Computational Astrophysics Code

- Language Choices:
- Interpreted languages: Python etc.
- Intepreted languages do runtime checks but are too slow
- Compiled Languages: C, C++, Fortran
- Common compiled languages do not do runtime checks but are fast
- Is there some other suitable language?

The Go Programming Language

- Open source language from Google
- Designed by distinguished computer scientists
- One of the designers is Ken Thompson who created UNIX
- Version 1.0 released in 2012

Go Features

- Statically typed compiled language (good performance)
- Automatic memory management (no memory leaks)
- Runtime checks (no silent index out of array bounds)

Is Go suitable for Computational Astrophysics?

- To find the answer:
- Implement Monte Carlo Radiative Transfer in Go
- For accurate results, a large number of incoming photons have to be simulated
- Hence the program has loops with a large number of iterations

Monte Carlo Radiative Transfer

- Scattering and absorption in spherical layers in an exoplanet atmosphere
- Photon travels random optical depth tau τ till it gets scattered or absorbed
- $\tau = -\log(1 - \xi)$ where ξ is a random number between 0 and 1
- Probabilty of scattering is equal to the single scattering albedo

Monte Carlo Radiative Transfer

- Use the Stokes vector (I, Q, U, V) to include the physics for polarization
- The new Stokes vector is the product of a 4×4 matrix and the old Stokes vector
- The random direction of scattering depends on the same 4×4 matrix

Monte Carlo Radiative Transfer

- At the surface of the planet, photon gets absorbed or reflected by Lambertian surface
- The photon either gets absorbed within the atmosphere, or it gets absorbed on the surface of the planet or it exits at the top of the atmosphere
- If the photon exits at the top of the atmosphere then its Stokes vector and exiting direction (θ, ϕ) is recorded

Experience with Go

- Fast compile times and fast run times
- Helpful compiler and runtime error messages
- Executable has no dependencies on installed libraries on the target machine
- Everyone's code is formatted the same way by gofmt

Experience with Go

- Readable online documentation
- Classic book "The Go Programming Language" by Donovan and Kernighan
- Large number of built-in packages including complex numbers and random numbers
- External Gonum project for additional scientific packages
- Less number of scientific packages compared to older languages

Conclusion

- Go is an excellent language for Computational Astrophysics.