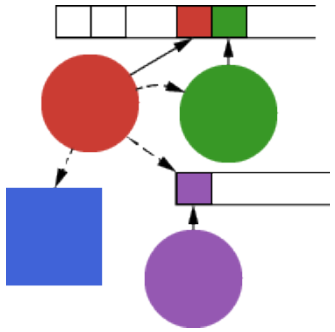


Actors.jl

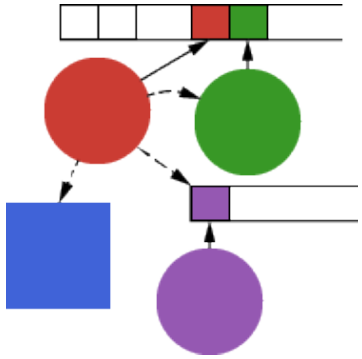
Concurrent computing in Julia with actors.

Paul Bayer, 2020-01-27

What are actors?

(according to the Actor model)

Actors.jl



An actor ... in response to a message it receives, can concurrently:

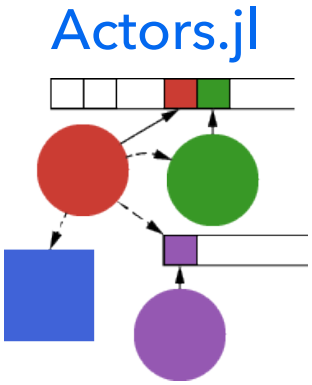
- send a finite number of messages to other actors;
- create a finite number of new actors;
- designate the behavior to be used for the next message it receives

and actors …

- come in systems (“one actor is no actor”),
- can be used to model concurrency.

How are actors implemented?

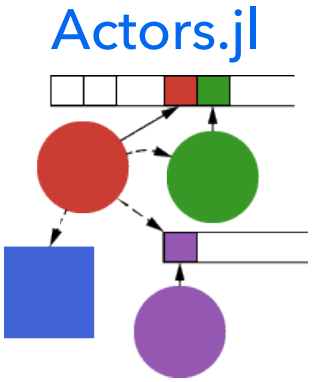
… to provide a higher level of abstraction for parallelism



- they are Julia **Tasks** that
  - **persist** in a loop,
  - listen to a **Channel**,
  - have **Functions** as **mutable** behaviors,
  - with **local** variables/state/acquaintances,
  - execute their **behavior** when a **message** arrives,
  - follow a message **protocol**.

What actors are good for?

How can they complement parallel computing in Julia



Actors are **concurrent objects**, they can

- **interact** with tasks, users or other actors via messages,
- **represent** variables, atomic operations, event handlers, state machines …
- fulfill **special duties**, e.g. as supervisors, guards, firewalls, generic servers …
- **compose** as groups, hierarchies, frameworks …
- make **concurrency easier** to understand and reason about and therefore
- **complement** Julia's features for concurrent and parallel programming.