Introduction to Unicorn, a Rack HTTP server

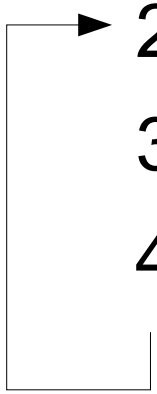
Paul Grayson, 6 June 2012

Background - Rack

- Rack is an interface between web servers and application code.
- It is all Ruby: app is accessed via a method
- Rails 3 is on Rack
- [show example Rack app]

Background - sockets

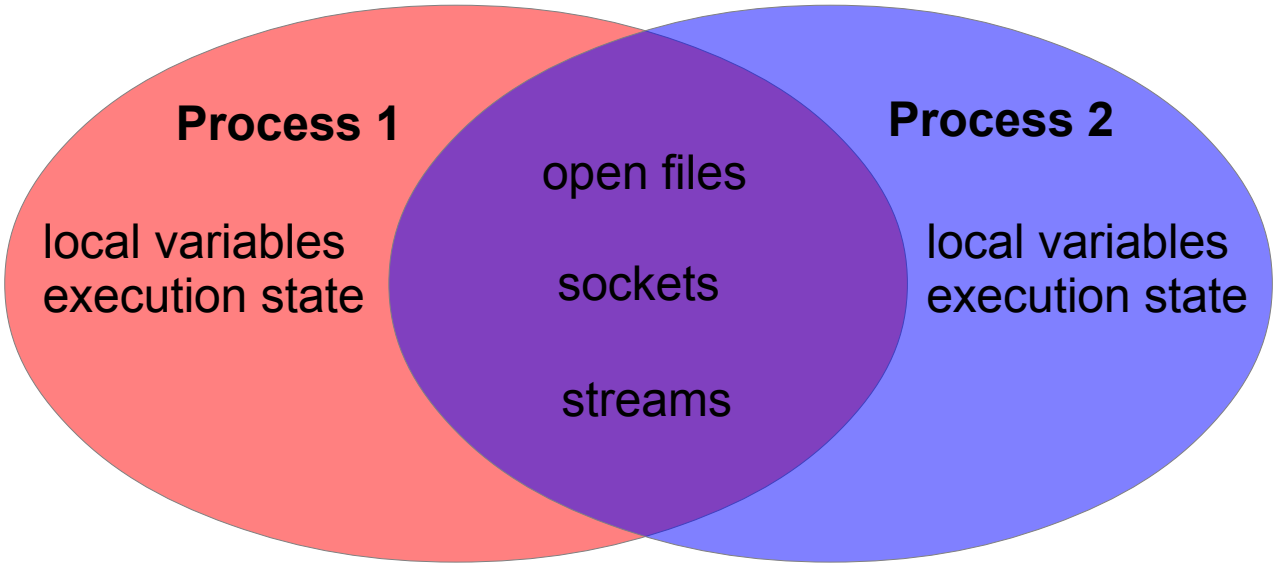
- HTTP connection life cycle
  1. Server listens on a TCP port
  2. Client connects
  3. Server accepts the connection
  4. HTTP Request/response etc.



Background - processes

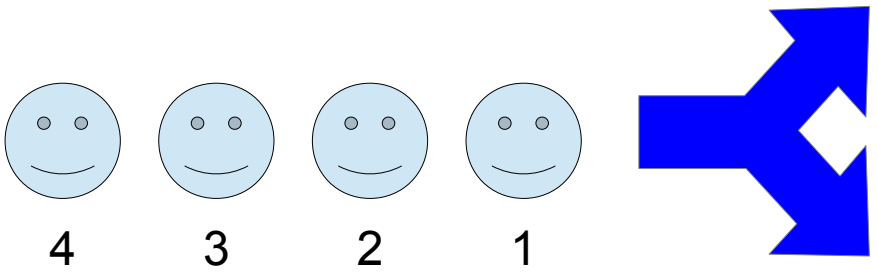
- Processes
  - Independent program copies
  - Copy-on-write saves RAM
  - Processes *may* share system resources

    ...such as sockets.



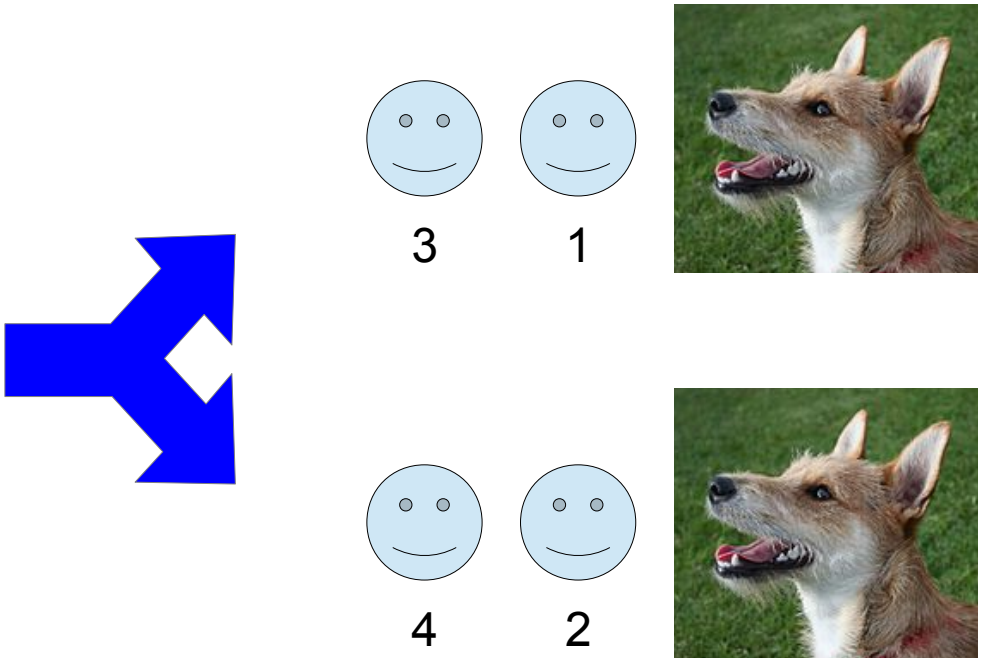
Traditional Rack servers (Mongrel, Thin)

- Multiple processes & ports
- A *proxy* distributes clients



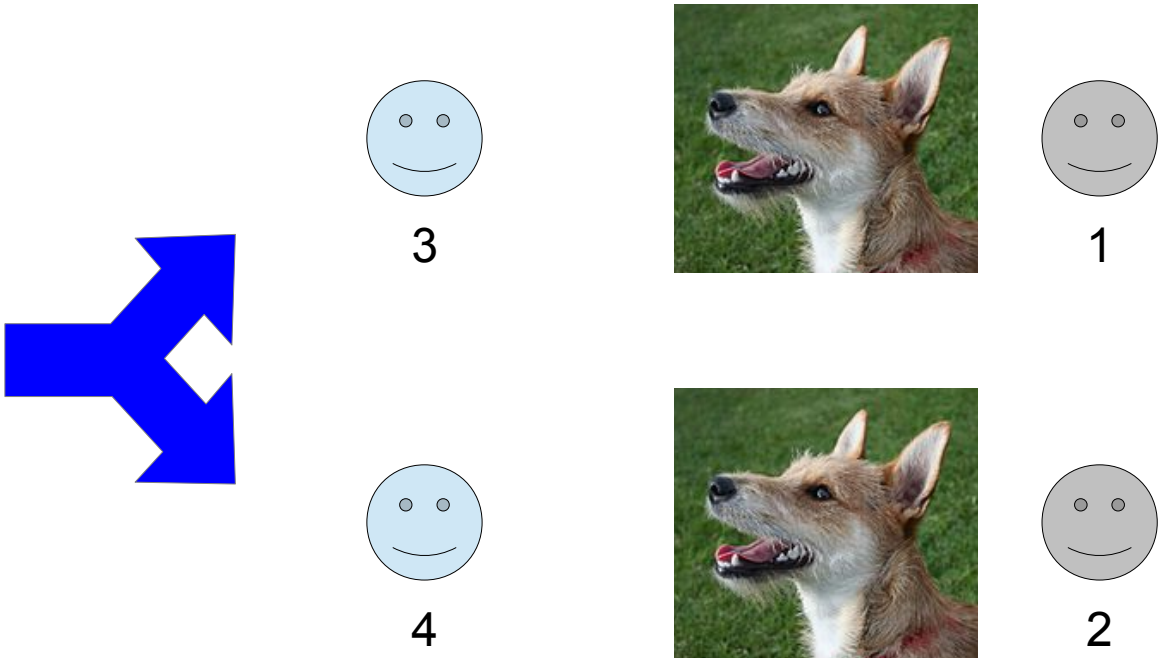
Traditional Rack servers (Mongrel, Thin)

- Multiple processes & ports
- A *proxy* distributes clients



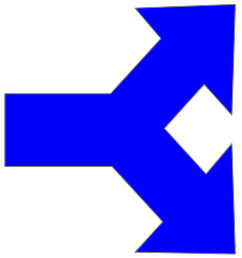
Traditional Rack servers (Mongrel, Thin)

- Multiple processes & ports
- A *proxy* distributes clients



Traditional Rack servers (Mongrel, Thin)

- Multiple processes & ports
- A *proxy* distributes clients

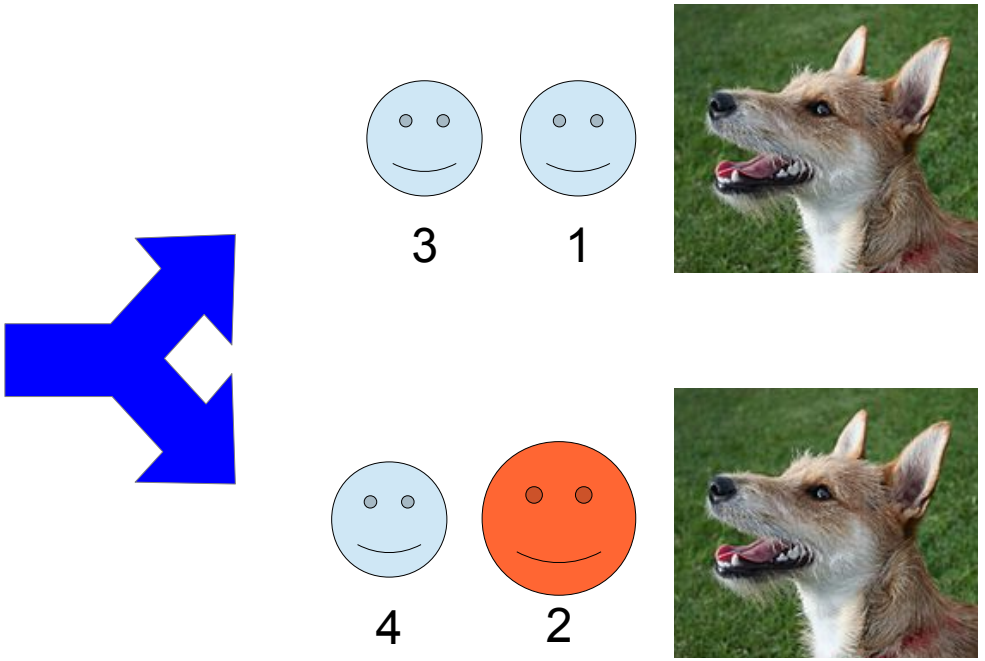


3 1

4 2

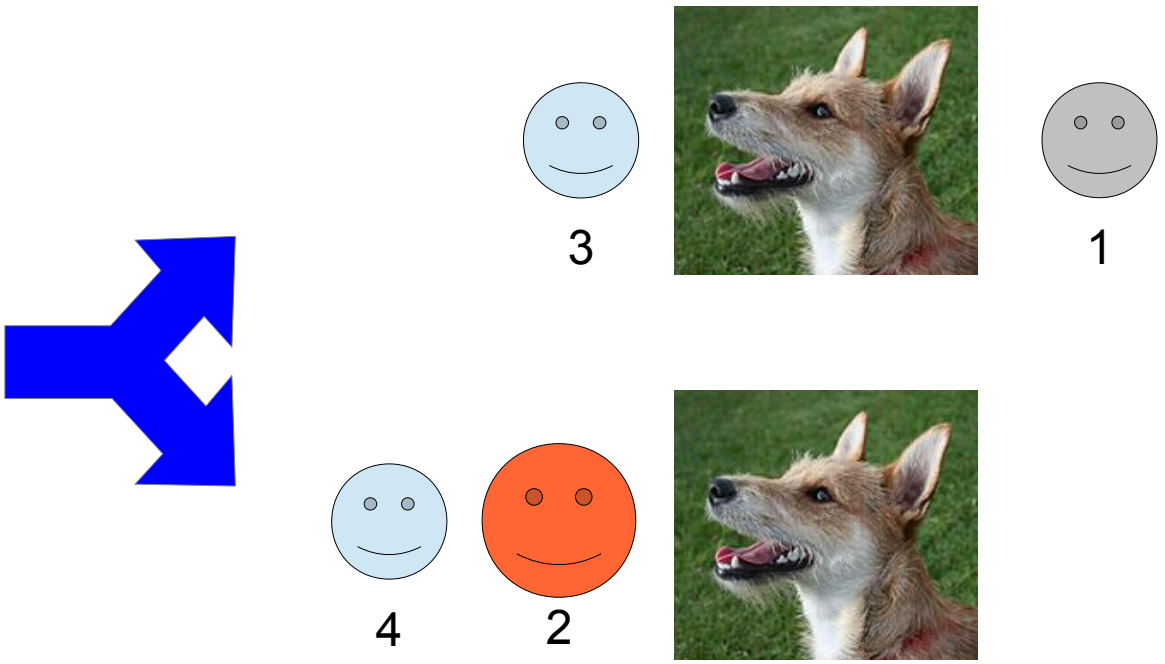
Traditional Rack servers (Mongrel, Thin)

Slow requests will slow down other clients!



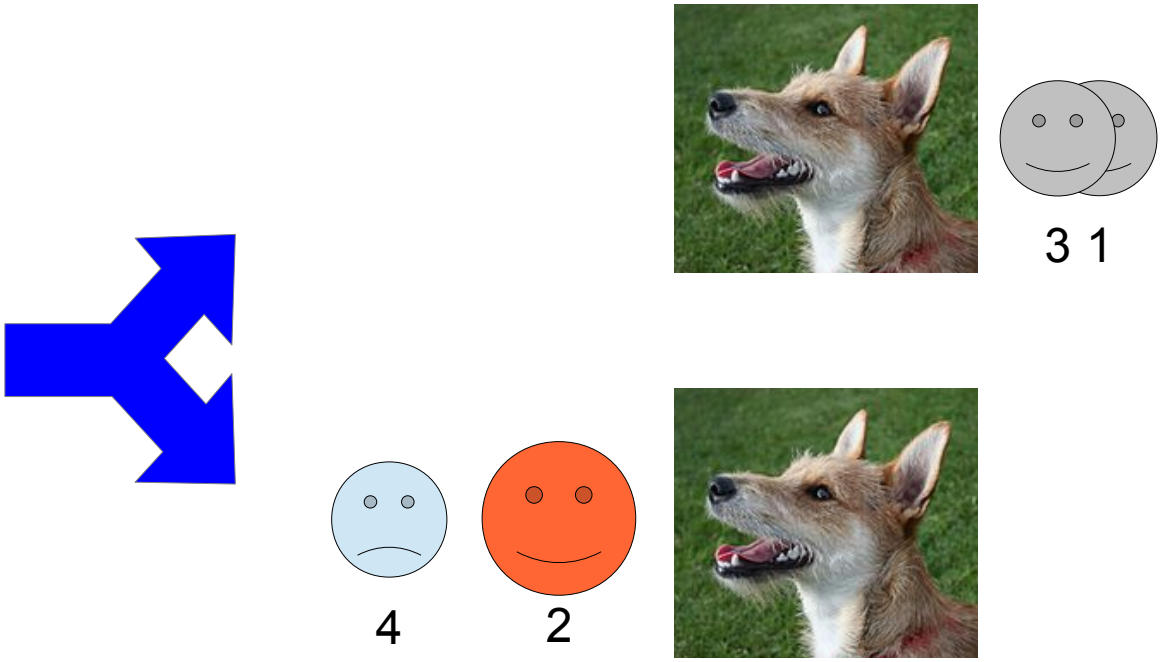
Traditional Rack servers (Mongrel, Thin)

Slow requests will slow down other clients!



Traditional Rack servers (Mongrel, Thin)

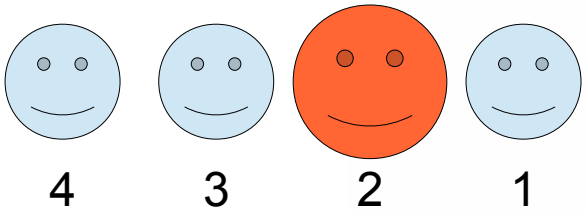
Slow requests will slow down other clients!



[show thin example]

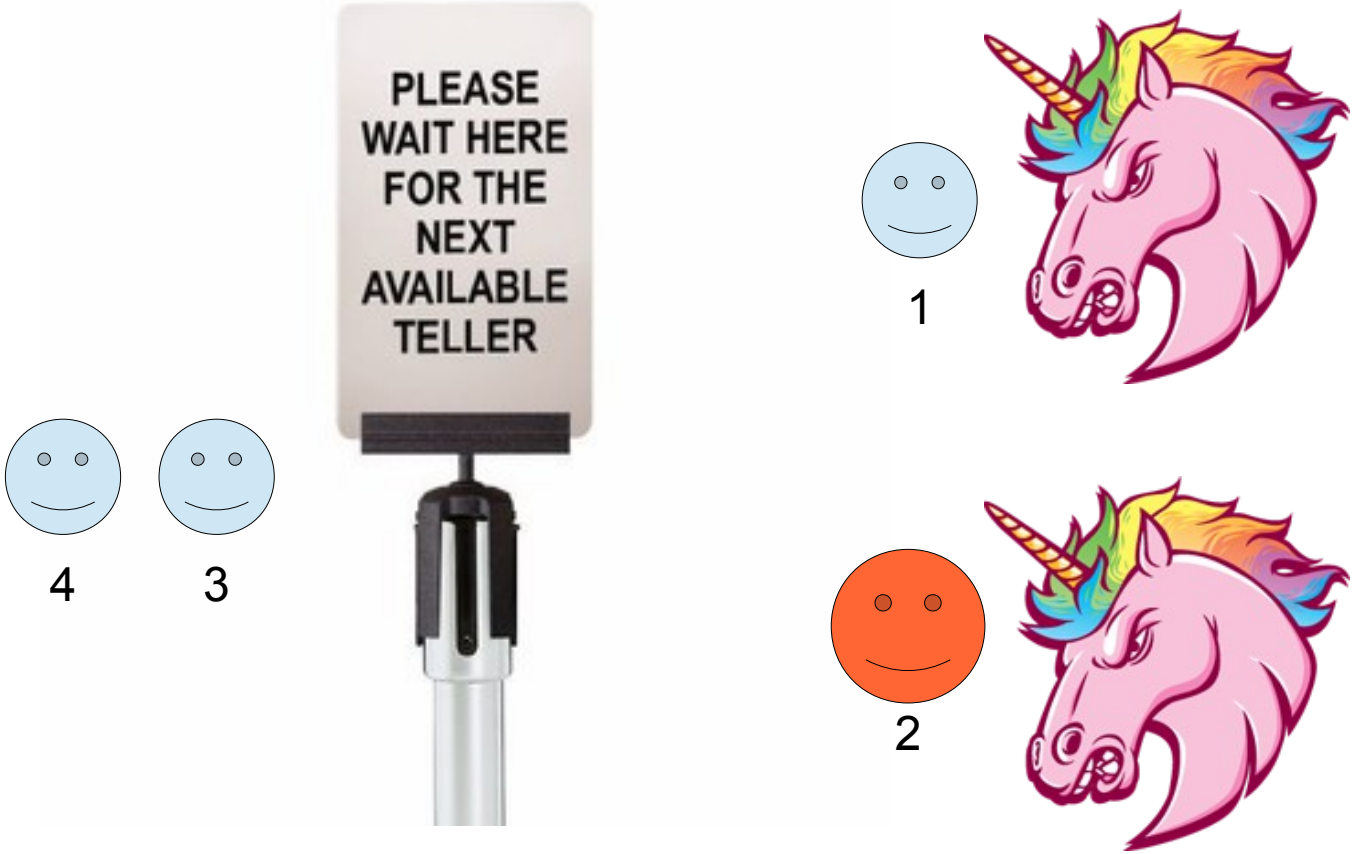
Unicorn

- Multiple processes, ONE shared socket
- Unicorns accept connections when idle



Unicorn

- Multiple processes, ONE shared socket
- Unicorns accept connections when idle



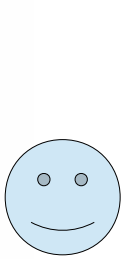
Unicorn

- Multiple processes, ONE shared socket
- Unicorns accept connections when idle

Unicorn

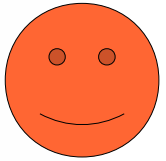
- Multiple processes, ONE shared socket
- Unicorns accept connections when idle

4



3

1



2

Unicorn

- Multiple processes, ONE shared socket
- Unicorns accept connections when idle

4


2

Unicorn

- Multiple processes, ONE shared socket
- Unicorns accept connections when idle

Unicorn

- Multiple processes, ONE shared socket
- Unicorns accept connections when idle

[show unicorn example]

Unicorn – painless restarts with no downtime

- Based on obscure UNIX signals:

  USR2: start new copy

  WINCH: gracefully stop workers

  QUIT: graceful shutdown

  - Lots of other signals, e.g. reopening logs
- Write your own scripts to do it
- [show example]

Unicorn does not solve all problems

- Use something in front of ruby for
  - slow clients
  - static files
- Unicorn will occasionally die, so monitor it

  (because it is Ruby?)
- Rails still uses a lot of memory

Conclusion: a decent server setup