Building a web app in D

Lio Lunesu

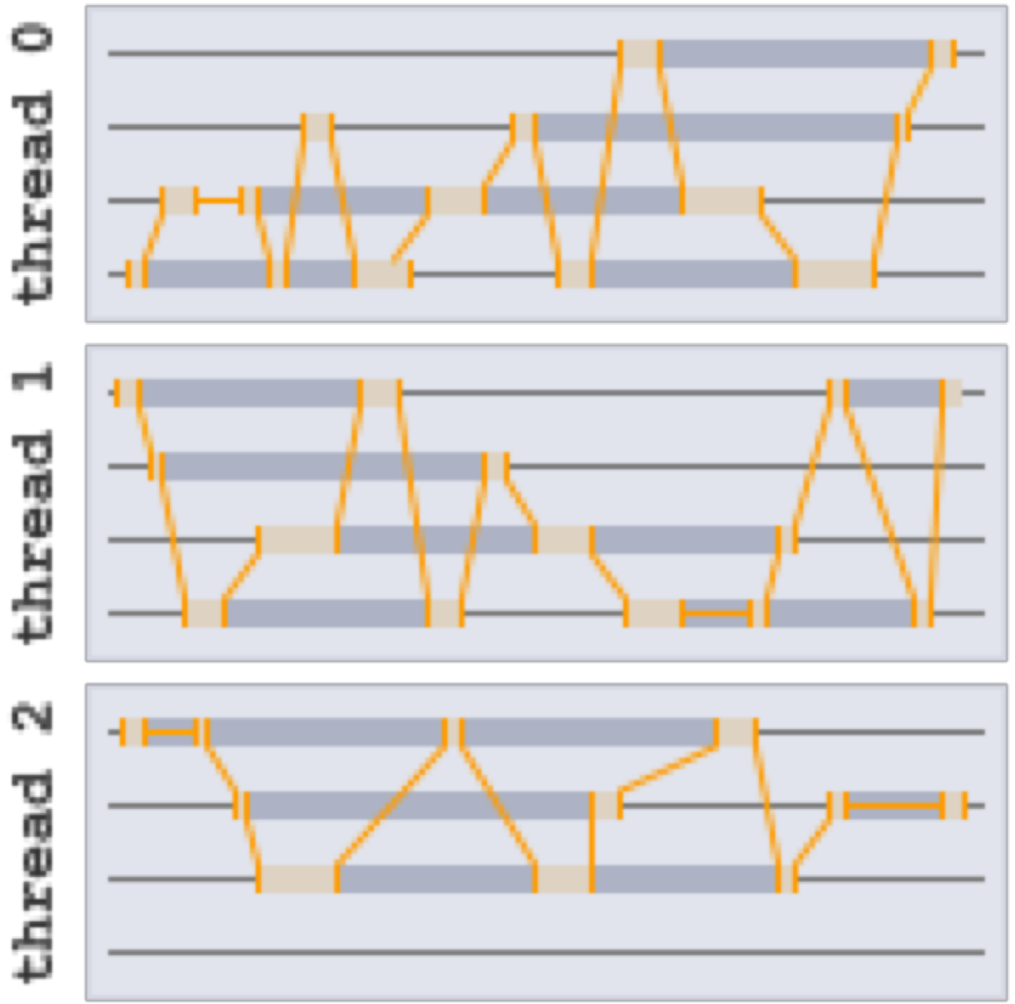
What's D

- Multi-platform **statically compiled** systems language
- Community driven, **Open Source**
- **Familiar** JavaScript **syntax** and convenience
- "**dub**" module management
- **Multi-paradigm**: imperative, functional, OO, DBC, TDD
- Best-in-class **generative programming**
- **Familiar toolsets**: GDC (GNU), LDC (LLVM)
- Compiler enforced memory **safety**, thread safety

What’s Vibe.D

- High-level declarative **REST** and **web application framework**
- **Asynchronous I/O** for maximum speed and minimum memory usage
- **Compile-time "Diet" templates** for unparalleled dynamic page speed
- Compiled to **native machine code**
- **Multi-threading** and integrated **load-balancing***
- **Fiber based** blocking programming model for concise and intuitive development
- Full support for **exception based** error handling

Fiber based **pseudo-blocking**




```
listenTCP(1337, (conn) {
  conn.write("Hello!\n");
  conn.write("Your respons
  conn.write(conn.readLine
  conn.write("Bye!\n");
});

auto c = connectTCP("127.0
c.write("How are you\n");
writeln("Server response:
writeln(c.readAllUTF8());
```

Getting Started

```
cd $(mktemp -d)

nix-shell -p dub -p dmd

dub init -t vibe.d

dub
```

Basic Template

dub.json	Dub app manifest with all dependencies
shell.nix	Pull in dmd, dub, and make
.envrc	”use nix” lets **direnv** add all tools to $PATH
Makefile	Quickly switch between local test and deployment
dockerize	Tool that copies all dependencies to output dir (uses ldd)
Dockerfile	Multi-stage: Build-stage: **FROM dlanguage/dmd AS build-stage** Run-time stage: **FROM scratch**

Demo?

Resources

https://github.com/lionello/vibeweb

noun.ly/skirt