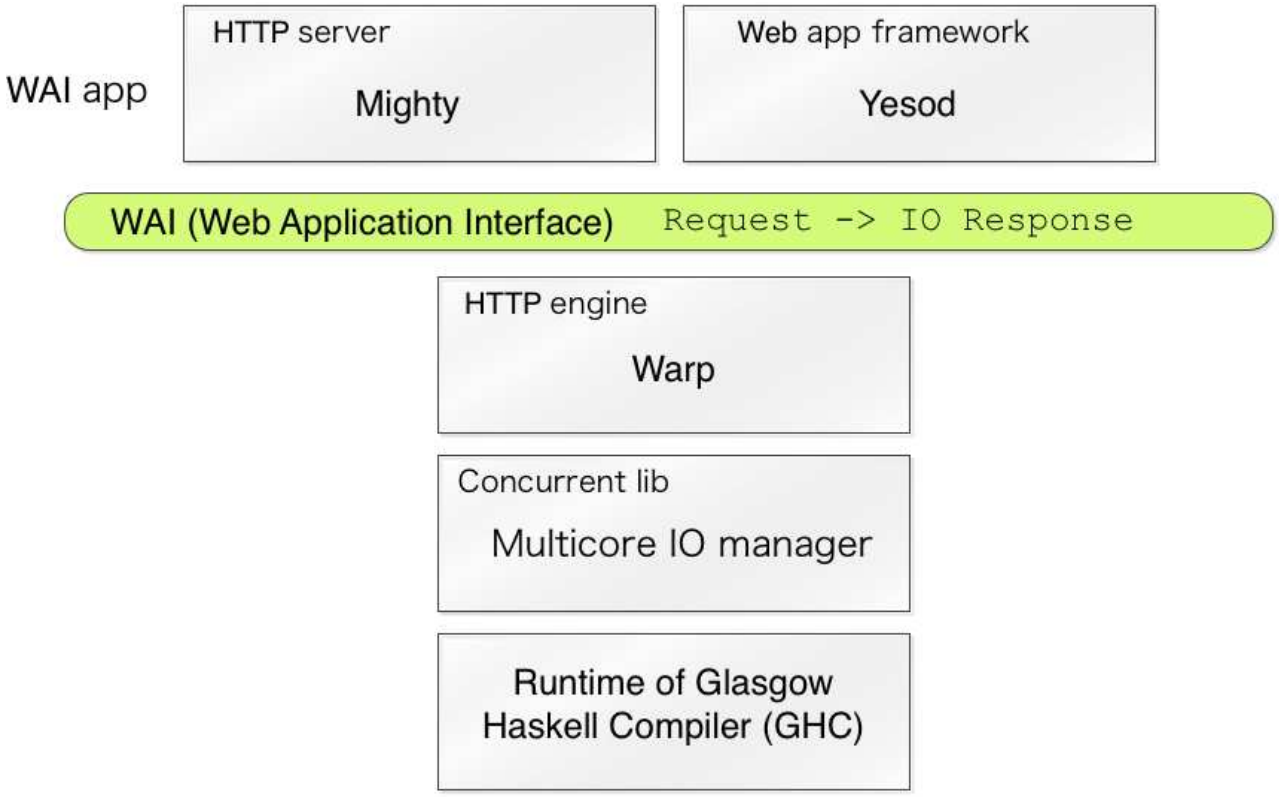
HTTP/2 in Warp

with Haskell lightweight threads

Kazu Yamamoto
@kazu_yamamoto

What is Warp?

- HTTP server library for WAI (Web Application Interface) written in Haskell

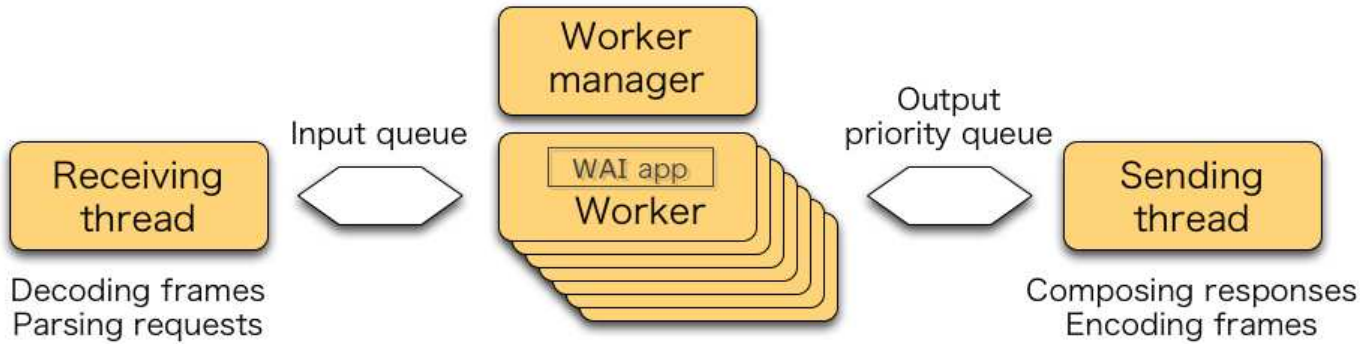


Warp distinctions

Lightweight threads	Not native threads Not event driven
Immutable data	Thread safe
Software Transactional Memory (STM)	Dead-lock free Not live-lock free

Warp HTTP/2 architecture

- Warp is written in thread programming
  - Haskell provides lightweight threads (aka green threads)



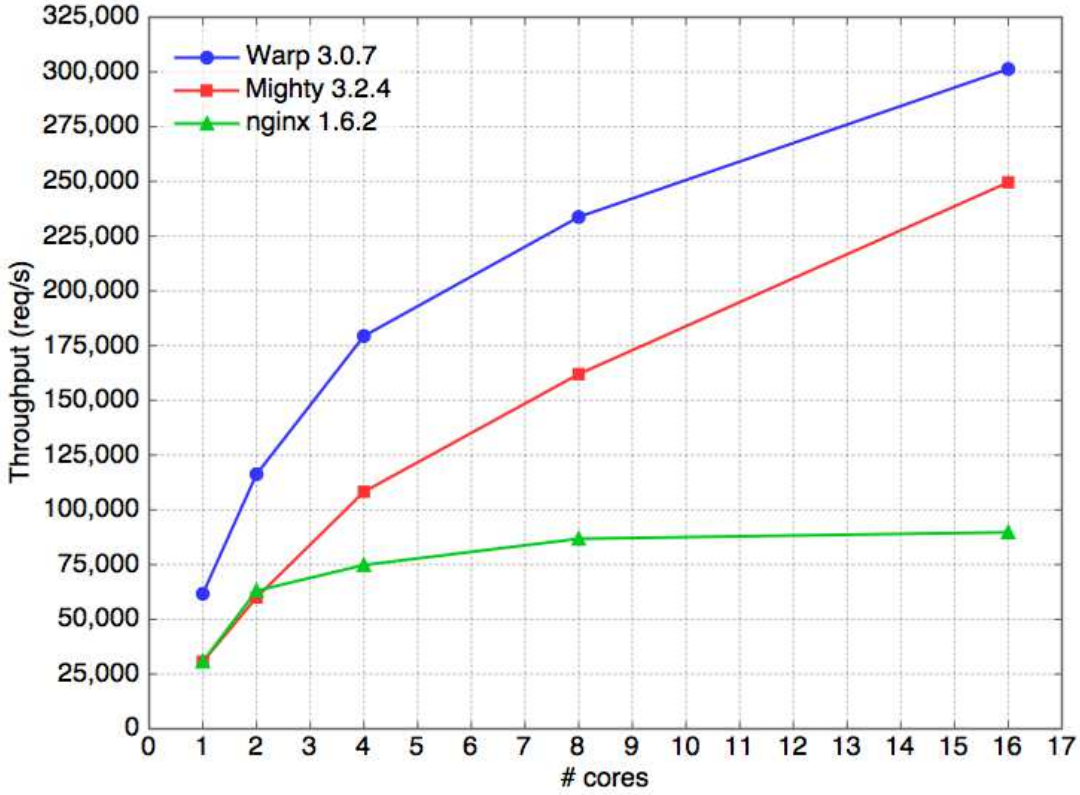
Lightweight components

- For N:M model, language support is the key

Haskell Lightweight thread	**Go** goroutine (lightweight thread)
Erlang Lightweight process	**Rust** Task (lightweight process)

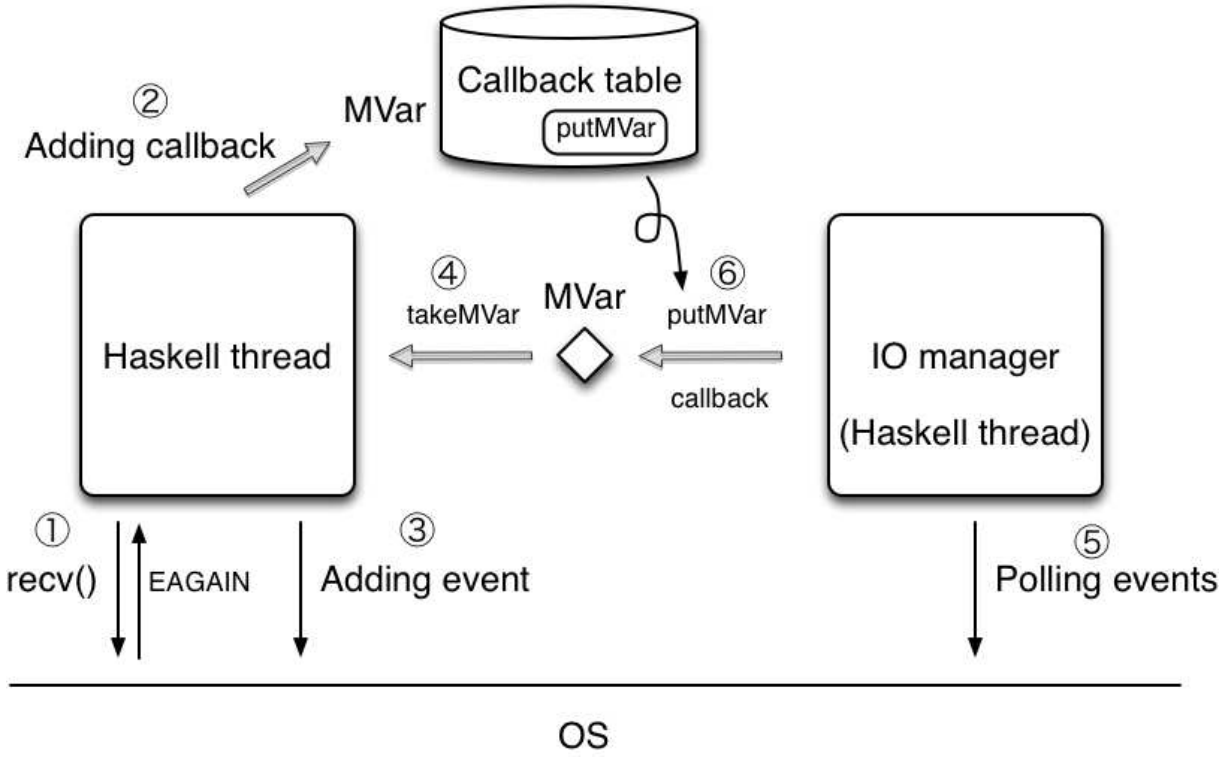
Scaling on multicores

- Haskell lightweight threads scales
- A simple throughput benchmark of HTTP/1.1



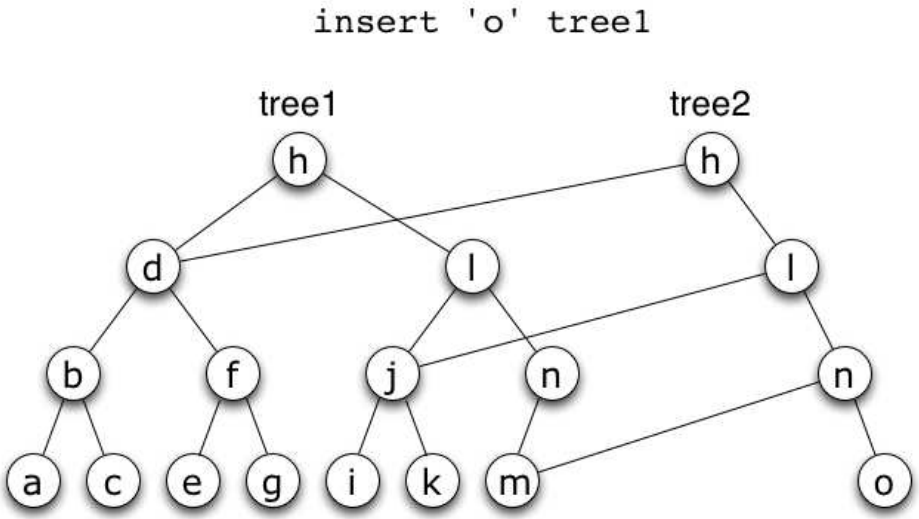
Thread programming on event driven

- Haskell programmers use thread programming
  - Haskell threads are logically blocked
- GHC runtime uses event driven programming
  - GHC runtime are not physically blocked



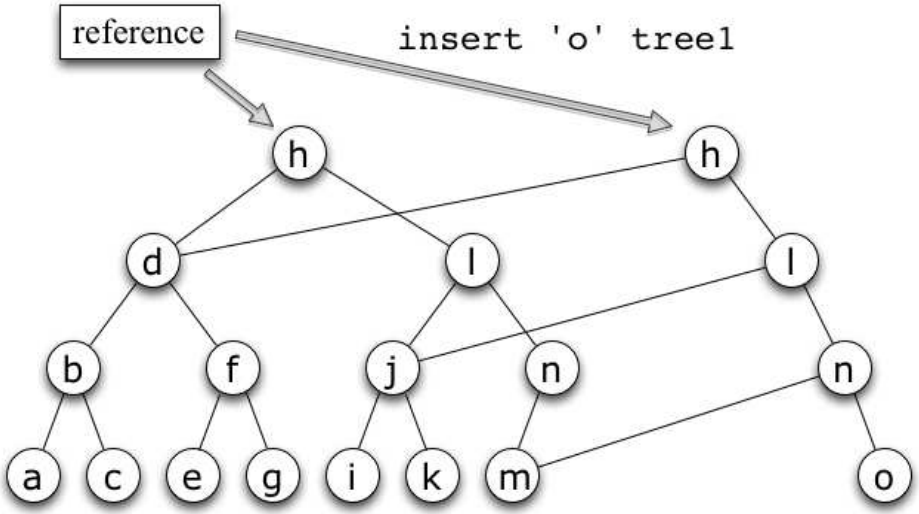
Immutable data

- Haskell is a functional programming language
  - Functional programming = programming with immutable data
- Immutable data is essentially thread-safe!
  - Useful for highly concurrent system



Software Transactional Memory

- Immutable data can be a pseudo mutable data with a reference



- Dead-lock free if used with STM references
  - STM turns multiple locks to a single