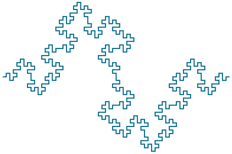
Heterogeneous computation with Cloud Haskell and GHCJS

Luite Stegeman



May 21, 2015

STATUS

Done

- GHC 7.8 and 7.10 support
- Concurrent Haskell runtime
- Official Cabal (and Hackage) support
- Improved build system and Template Haskell
- CPU and Heap profiling

In Progress

- Improved base library
  - Comprehensive JS bindings
  - JSString library
- New code generator with
  - Typed IR
  - Source maps
  - Reduced code size
  - Pluggable functionality

Planned

- GHCJSi

STATUS

Done

In Progress

- Improved base

statistical CPU profiling on node.js

```
...
2909  98.8%    0   0.0%   LazyCompile: ~Module._compile module.js:378:37
2908  98.8%    0   0.0%    Function: ~<anonymous> test.js:1:11
2899  98.5%    0   0.0%     LazyCompile: ~h$cpuProfiler.runCC test.js:27:17
1739  59.1%    0   0.0%      CostCentre: cost centre A main.hs:10:10
1168  39.7%    0   0.0%       CostCentre: cost centre B main.hs:14:3
1168  39.7%  1167  39.6%        CostCentre: cost centre C main.hs:21:9
0571  19.4%   570  19.4%       CostCentre: cost centre C main.hs:21:9
1160  39.4%    0   0.0%      CostCentre: cost centre B main.hs:14:3
0589  20.0%   582  19.8%       CostCentre: cost centre A main.hs:10:10
0006   0.2%    4   0.1%        LazyCompile: *pow native math.js:89:17
0571  19.4%   571  19.4%       CostCentre: cost centre C main.hs:21:9
0005   0.2%    0   0.0%     LazyCompile: ~<anonymous> node.js:208:48
0005   0.2%    0   0.0%      LazyCompile: ~NativeModule.require node.js:783:34
...
```

Status

Done

- GHC 7.8 and 7.10 support
- Concurrent Haskell runtime
- Official Cabal (and Hackage) support
- Improved build system and Template Haskell
- CPU and Heap profiling

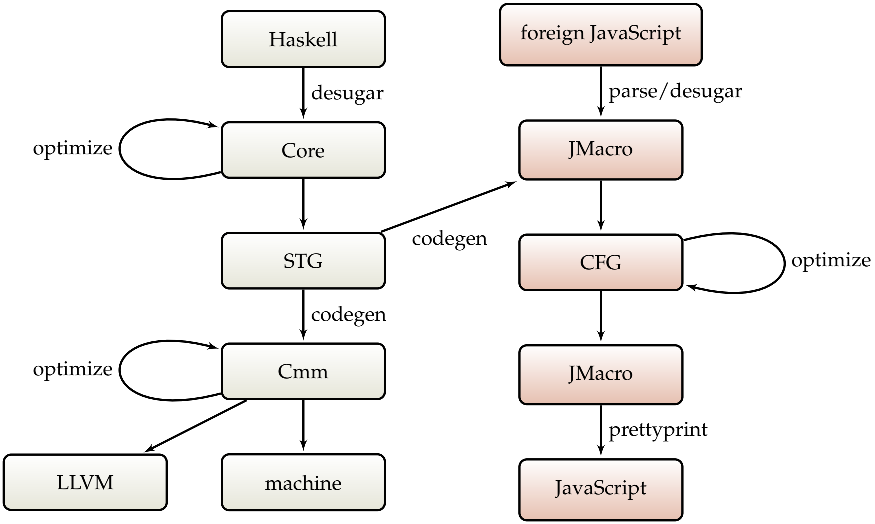
In Progress

- Improved base library
  - Comprehensive JS bindings
  - JSString library
- New code generator with
  - Typed IR
  - Source maps
  - Reduced code size
  - Pluggable functionality

Planned

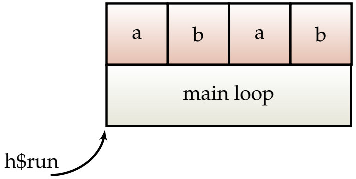
- GHCJSi

GHCJS PIPELINE



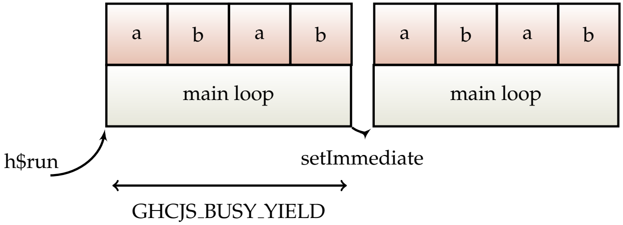
GHCJS CONCURRENCY

Asynchronous scheduling



GHCJS CONCURRENCY

Asynchronous scheduling



GHCJS CONCURRENCY

Asynchronous scheduling

constants in shims/src/thread.js

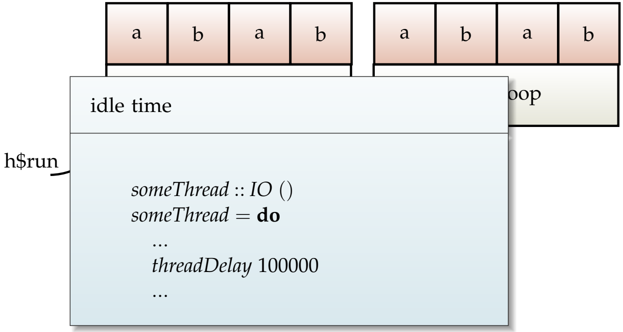
```
// preempt threads after the scheduling quantum (ms)
#ifndef GHCJS_SCHED_QUANTUM
#define GHCJS_SCHED_QUANTUM 25
#endif

// check sched quantum after 10*GHCJS_SCHED_CHECK calls
#ifndef GHCJS_SCHED_CHECK
#define GHCJS_SCHED_CHECK 1000
#endif

// yield to js after running haskell for GHCJS_BUSY_YIELD ms
#ifndef GHCJS_BUSY_YIELD
#define GHCJS_BUSY_YIELD 500
#endif
```

GHCJS CONCURRENCY

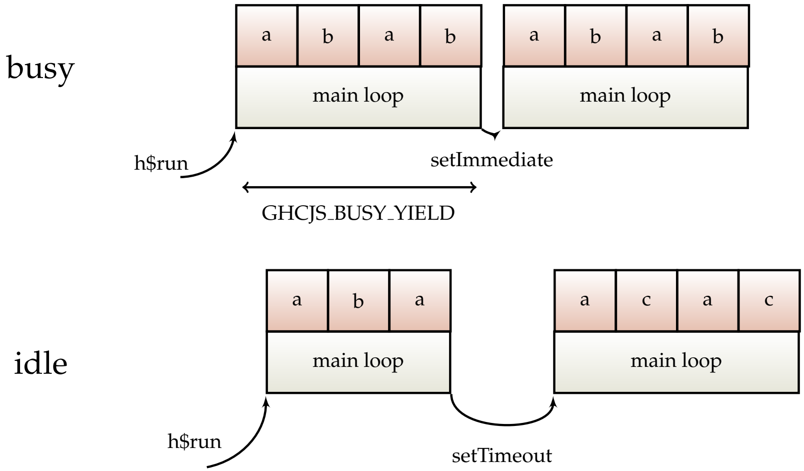
Asynchronous scheduling



```
someThread :: IO ()
someThread = do
  ...
  threadDelay 100000
  ...
```

GHCJS CONCURRENCY

Asynchronous scheduling



GHCJS CONCURRENCY

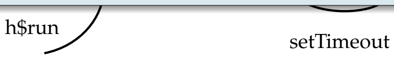
Asynchronous scheduling

interruptible foreign imports

```
import System.Timeout

foreign import javascript interruptible
  "window.setTimeout(function() { $c(f($1)); }, 1000*Math.random());"
  js_wait :: Int → IO Int

test :: Int → Int → IO ()
test x y = do
  r ← timeout x (js_wait y)
  case r of
    Nothing → putStrLn "timeout"
    Just n → putStrLn ("got: " ⧺ show n)
```



GHCJS Concurrency

Synchronous scheduling

event handling

```
myButton.addEventListener('click', function(event) {
 doSomething();
 event.stopPropagation();
});
```

GHCJS Concurrency

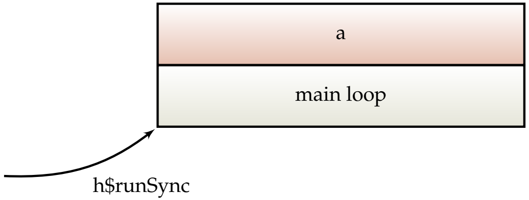
Synchronous scheduling

animations

```
requestAnimationframe(function() {
 drawAnimationFrame();
});
```

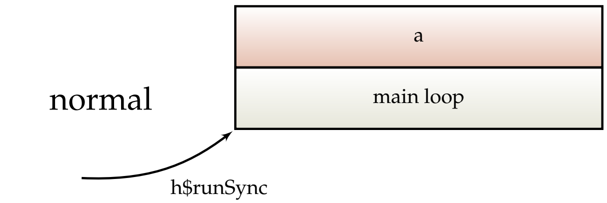
GHCJS CONCURRENCY

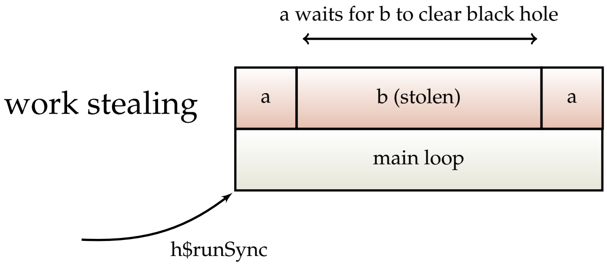
Synchronous scheduling



GHCJS Concurrency

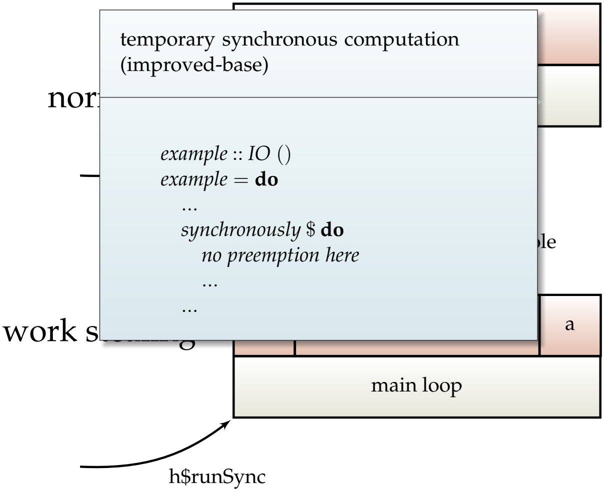
Synchronous scheduling





GHCJS CONCURRENCY

Synchronous scheduling



temporary synchronous computation (improved-base)

```
example :: IO ()
example = do
  ...
  synchronously $ do
    no preemption here
    ...
  ...
```

SHARING COMPUTATIONS

locations

- client
- web worker
- server
- other:
  - compute nodes
  - webgl / opengl (accelerate)

considerations

- compute power
- network latency
- data availability

SHARING C

locations

- client
- web
- serve
- other
  - c
  - v
    (

ower

ency

bility

shared code in a Cabal project

```
library
 exposed-modules:   Primes
 build-depends:    base >=4.7 && <4.9
 hs-source-dirs:    src

executable example-primes
 if impl(ghcjs)
  main-is:         Main.hs
  build-depends:    primes,
                    base >=4.7 && <4.9,
                    ghcjs-dom
  hs-source-dirs:    client
 else
  main-is:         Main.hs
  build-depends:    primes,
                    base      >= 4.7 && < 4.9,
                    warp      >= 3.0 && < 3.1,
                    ...
  hs-source-dirs:    server
```

Sharing Computations

locations

- client
- web
- serve
- other
  - compute nodes
  - webgl / opengl (accelerate)

The Async library

race computeOnClient sendRequestToServer

ions

ute power

rk latency

data availability

CLOUD HASKELL

- Lightweight exports
- Process control
  - Server
  - Web Worker
  - Other nodes
- Serialization (binary)

CLOUD HASKELL

- Light
- Proce
  - S
  - W
  - C
- Seria

Cloud Haskell

$$
\begin{aligned}
&\mathit{fibonacci} :: \mathit{Integer} \to \mathit{Integer}\\
&\mathit{fibonacci}\ 0 = 0\\
&\mathit{fibonacci}\ 1 = 1\\
&\mathit{fibonacci}\ n = \mathit{fibonacci}\ (n-2) + \mathit{fibonacci}\ (n-1)\\
\\
&\mathit{remotable}\ [\text{`}\mathit{fibonacci}]
\end{aligned}
$$

CLOUD HASKELL

- Lightweight exports
- Process control
  - Server
  - Web Worker
  - Other nodes
- Serialization (binary)

GHCJS support

- GHC 7.10 StaticPointers
- Transport and job control unfinished

HANDS ON!

- Install GHCJS
  - Build from source
    - length build process
    - follow README carefully
    - use linux for best results
    - GHC 7.8.4 recommended
  - Try-reflex nix environment
  - Preinstalled VM image (USB sticks available)
- Get example code
  - USB sticks

HANDS ON!

- Install GHCJS

Example

```
$ cat hello.hs
main = putStrLn "Hello, world"
$ ghcjs -o hello hello.hs
[1 of 1] Compiling Main          ( hello.hs, hello.js_o )
Linking hello.jsexe (Main)
$ node hello.jsexe/all.js
Hello, world
```

- (

HANDS ON!

- Install GHCJS
  - Build from source
    - length build process
    - follow README carefully
    - use linux for best results
    - GHC 7.8.4 recommended
  - Try-reflex nix environment
  - Preinstalled VM image (USB sticks available)
- Get example code
  - USB sticks