What's new in GHC?

- Tons of activity
- Lots of contributors:
50 commits/week
What's new? Syntax!

- **case**

  \[ \text{f :: Maybe Bool} \rightarrow \text{Int} \]
  \[
  \text{f} (\text{Just x} \rightarrow x) \\
  \text{f} (\text{Nothing} \rightarrow \text{True})
  \]

- **multi-if**

  \[
  \text{if | x > 0 \rightarrow e1} \\
  | x == 0 \rightarrow e2 \\
  | \text{otherwise} \rightarrow e3
  \]
What's new? Types!

- Kind polymorphism and data type promotion

```haskell
data T f a = MkT (f a)
-- OLD: T :: (*->*) -> * -> *
-- NEW: T :: forall k. (k->*) -> k -> *
```

- Deferred type errors
- Holes (demo)
- Type nats (Iavor)

```haskell
append :: Vec n1 -> Vec n2 -> Vec (n1+n2)
```

- Performance improvements (and MUCH more beauty) in the constraint solver
Long standing project to overhaul the code generator

- STG (A-normal form, decorated)
  - Big, complicated, naive
  - Simple

- C-- with no procedure calls
  - x86
  - PowerPC

- C-- with procedure calls
  - C--

- New Hoopl C-representation
  - Many new optimisation opportunities
  - CPS conversion

- Lot s of cruft removal
The “New code generator”

- New optimisation opportunities
  - traditional compiler optimisations have far more opportunity before CPS conversion
  - already better in some cases: one user reported a 2x speedup
  - better calling conventions so we can pass SIMD values in registers

- Will be in 7.8.1
  - but try out a preview in 7.6.1 with “-fnew-codegen”
What's new: platforms

- Win64 version works (thanks IHG)
- Full-scale cross-compilation getting very close
What's new? Parallelism

- Changing number of virtual processors (Simon M)
- SIMD instructions (Geoff)
- Cloud Haskell (Duncan)
- DPH (Ben)
Changing `-N` at runtime

- (in 7.6.1)
- Previously you had to say `+RTS -N2` to use 2 cores, now you can say
- Call `setNumCapabilities` any time
- Useful for
  - increasing the parallelism only for the parallel sections of your program
  - no need to tell your users about `+RTS -N`
  - fork and then `setNumCapabilities` (for daemons)

```haskell
import GHC.Conc
main = do
  n <-getNumProcessors
  setNumCapabilities n
  ...
```
• Eventually, this will enable us to dynamically tune the amount of parallelism to respond to contention on the machine.

• Diagram shows running a parallel program while periodically changing -N in a cycle 1..8..1..
Status of SIMD Support in GHC

- Low-level primops for SIMD operations.
- Support for stream fusion of SIMD operations in the *vector* library.
- Transparent support for SIMD operations in DPH.
The Good News

Dot Product

- C (SIMD)
- vector (SIMD)
- DPH
- DPH (SIMD)
- DPH (Parallel Array)

Time (ms)

Threads
The Bad News...

- Register allocation not so hot.
- Not yet in HEAD.

http://hackage.haskell.org/trac/ghc/wiki/Simd
Template Haskell overhaul to provide both

- **Untyped** quotes and splices (for expressions, decls, types, and patterns). Allows explicit use of syntax constructors, and analysis of syntax.

- **Typed** quotes and splices (for expressions only, à la MetaML). Guarantees no type errors in generated code.

```
[e| reverse [True] |]  :: Q Exp

[| reverse [True] |]   :: TExp [Bool]
```
On the horizon

- Allow multiple instances of the same package to be installed. For example
  - containers-4.5 built against deepseq-1.4
  - containers-4.5 built against deepseq-1.2

- Needs Cabal changes too
- Avoids some manifestations of package hell
On the horizon

- GHC 7.8 in late 2012 or early 2013
  - New codegen
  - TH overhaul
  - Major DPH release
  - Type holes
  - SIMD
GHC is (still) hot

- Lots going on. Lots. Really a lot.
- We are having way to much fun
- You should join in. We need you.