Hack for HipHop

Julien Verlaguet (Facebook)

HipHop Team
What is PHP good at?

• PHP features:
  – Very fast installation time
  – A lot of libraries
  – Easy to learn
  – Scales well (avoids concurrency problems)

• But do we really care?
PHP: a FAST feedback loop
PHP: the challenges at scale

• Performance (runtime):
  – At this scale, 1% matters!
  – Hard to optimize

• Development:
  – Refactoring is difficult
  – Bugs are caught at runtime
  – Tooling is primitive
Scaling PHP runtime: HipHop

- PHP
  - Interpreter (Sandboxes)
  - Compiler C++ (Production)
Scaling PHP runtime: HHVM
Scaling PHP development: Hack

- HACK
- PHP
- JIT Compiler (HHVM)
HACK
(or Hack for HipHop)
Hack

• A **statically typed** language for HHVM
• Compatible with PHP:
  – Interoperates with no overhead
  – Same representation at runtime
• Evolved from PHP:
  – If you know PHP, you know Hack!
• Designed for **incremental adoption**:
  – Gradual typing
Hack Type System

• What must be annotated?
  – Class members
  – Function parameters
  – Return types

• What is inferred?
  – All the rest

• Annotating is an incremental process
Hack Types

• Nullable: ?int, ?MyClassName
• Tuples: (int, bool, X)
• Closures: (function(int): int)
• Collections: Vector<int>, Map<string, int>
• Generics: A<T>, foo<T>(T $x): T
• Constraints: foo<T as A>(T $x): T
• Type aliasing: [new]type t = ...
• Extensible records: shape(‘x’ => int)
DEM0

Snake Oil 2.0
HACK INTERNALS
Working at scale: Hack

- We knew we wanted an IDE from day one
- Big code base
- The solution, a server:
  - The server type-checks all the files
  - Keeps track of the dependencies
  - Recomputes types when something changed
Working at scale: The Constraints

- Auto-complete requires very low latency
- Users use version control (e.g., switching between branches)
- We must use a reasonable amount of RAM
- We must have a reasonable initialization time
- Must be **stable**
Hack is written in Ocaml!

• OCaml was a good choice:
  – Ideal for symbolic computation
  – Excellent performance
  – Can be compiled to JS
  – Interoperates well with C

• The challenge:
  – The runtime doesn’t support multicore
Hack architecture

- MASTER
- Worker1
- ...
- WorkerN
- SHARED HEAP

C

Arrows indicate communication between the components.
OCaml at Scale

- IPC:
  - Pipes, sockets etc ...
  - Caching layers to avoid deserialization cost
  - Carefully crafted lock free data structures (C code)

- Garbage collection:
  - Workers keep a small heap
  - Shared memory is compacted by the master

- OCaml makes you think hard about shared objects:
  - And that’s a good thing! ;-)
Questions?