Linear-Memory GC-Root Scanning

ROSS TATE

Frequent Request by GC Languages

Provide a means to scan the stack for GC roots in linear memory

Why?

Without scanning, you need to maintain a shadow stack of GC roots

- Requires runtime infrastructure for GC shadow stack(s)
- Increases binary size due to frequent extend/update/retract instructions
- Hinders run-time performance due to extra (slow) operations
- Most effort wasted because GC runs infrequently

With scanning, most effort is only done when actually needed

- Though there is background overhead for maintaining key invariants

Challenges

Provide *efficient* access to i32/i64 values on the stack that represent GC roots *without* limiting optimizations of locals

Prevent arbitrary access to *other* applications' i32/i64 values on the stack

Make no assumptions about how an application implements its GC

Illustrative Example

RUNTIME CODE

```
(memory ...)
(local-mark $gc_root i32)
(func $mark_gc_root (param $gc_root i32)
   ... ;; instructions for the GC's gc-root-marking process
)
(func $scan_for_gc_roots
   (enumerate-marked-locals $gc_root $repeat_with_next_marked_local
      (call $mark_gc_root) ;; the value of the marked local is on the stack
      (br $repeat_with_next_marked_local)
   end)
)
```

New tag for marking i32 locals

Like loop, but maintains a pointer into the stack, and executes the body with the next $gc_root-tagged local

EXAMPLE FUNCTION

```
(func $example_method_implementation (param $this_pointer i32)
   (local $array_index i32) (local $array_pointer i32)
   (marked-locals $gc_root $this_pointer $array_pointer ;; not $array_index
      ... ;; instruction[illegible]lementing method body
   )
)
```

Indicates that the i32 locals $this_pointer and $array_pointer (but not $array_index) should be marked with $gc_root

Extension: Moving GC

(local-mark $gc_root *mutable* i32)

- The label in enumerate_marked_locals takes the new i32 value to replace the tagged local with

Extension: Concurrent GC

fiber.enumerate_marked_locals $gc_root instr* end : [fiberref] -> []

- Like enumerate_marked_locals, except body is run with tagged locals in the given fiber.

Questions?

Poll for Phase 1

Interested Languages

C#

Erlang

Go

Julia

Racket/Scheme