

Ruby master - Feature #13715

[PATCH] avoid garbage from Symbol#to_s in interpolation

07/04/2017 10:04 PM - normalperson (Eric Wong)

| | |
|---|--------|
| Status: | Open |
| Priority: | Normal |
| Assignee: | |
| Target version: | |
| Description | |
| <pre>"ruby -e 'p GC.stat(:total_allocated_objects)'" goes from 70199 to 69540 allocated objects when loading RubyGems from a clean install.</pre> | |
| <p>The increased VM size slows down the whileloop2 and vm2_dstr case slightly, but string interpolation often consists of non-strings. The addition of inline cache helps integer cases slightly, and the intended Symbol optimization gives a major improvement.</p> | |
| <pre>speedup relative to trunk name built ----- -----: loop_whileloop2 0.984 vm2_dstr* 0.991 vm2_dstr_digit* 1.167 vm2_dstr_int* 1.120 vm2_dstr_nil* 1.181 vm2_dstr_sym* 1.663</pre> | |
| <p>Digits (0-9), Integers, and perhaps true/false/nil may be optimized in the future.</p> | |
| <pre>* vm_eval.c (rb_vm_call0_body): new function exports vm_call0_body * vm_inshelper.c (vm_tostring): new function * insns.def (tostring): call vm_tostring with ci + cc * compile.c (iseq_compile_each0): adjust tostring insn compile * benchmark/bm_vm2_dstr_digit.rb: new benchmark * benchmark/bm_vm2_dstr_int.rb: ditto * benchmark/bm_vm2_dstr_nil.rb: ditto * benchmark/bm_vm2_dstr_sym.rb: ditto</pre> | |

History

#1 - 07/13/2017 02:48 AM - ko1 (Koichi Sasada)

```
VALUE rb_vm_call0_body(rb_thread_t *th, struct rb_calling_info *calling,
```

You don't need to expose vm_call0_body() because vm_eval.c and vm_inshelper.c are included in vm.c.

```
if (RB_TYPE_P(recv, T_SYMBOL)) {
  vm_search_method(ci, cc, recv);
```

It seems we can use vm_method_cfunc_is().

```
calling.block_handler = VM_BLOCK_HANDLER_NONE;
calling argc = 0;
calling.recv = recv;
val = rb_vm_call0_body(th, &calling, ci, cc, 0);
return RB_TYPE_P(val, T_STRING) ? val : rb_any_to_s(recv);
```

How about to call rb_obj_as_string() directly? I understand that you want to reuse method search results, but code will be simplified.

Or make new function to call method with given ci, cc?

#2 - 07/13/2017 08:11 AM - normalperson (Eric Wong)

ko1@atdot.net wrote:

```
VALUE rb_vm_call0_body(rb_thread_t *th, struct rb_calling_info *calling,
```

You don't need to expose `vm_call0_body()` because `vm_eval.c` and `vm_inshelper.c` are included in `vm.c`.

Ah, thanks. I just added a prototype for `vm_call0_body`.

```
if (RB_TYPE_P(recv, T_SYMBOL)) {  
    vm_search_method(ci, cc, recv);
```

It seems we can use `vm_method_cfunc_is()`.

Right, I expanded the function in my original patch it since I wanted to make it obvious that `cc` is populated regardless of function match. I am using a comment instead, now.

```
calling.block_handler = VM_BLOCK_HANDLER_NONE;  
calling argc = 0;  
calling.recv = recv;  
val = rb_vm_call0_body(th, &calling, ci, cc, 0);  
return RB_TYPE_P(val, T_STRING) ? val : rb_any_to_s(recv);
```

How about to call `rb_obj_as_string()` directly? I understand that you want to reuse method search results, but code will be simplified.

I think the >10% improvement for non-symbol `dstr` benchmarks is worth the complexity, especially since we populate `cc` for `vm_method_cfunc_is` anyways.

Or make new function to call method with given `ci`, `cc`?

I'm not sure what you mean, `vm_call0_body` is insufficient?

Anyways, v2 patch here:

<https://80x24.org/spew/20170713075445.25252-1-e@80x24.org/raw>

Files

| | | | |
|--|---------|------------|--------------------------|
| 0001-avoid-garbage-from-Symbol-to_s-in-interpolation.patch | 6.32 KB | 07/04/2017 | normalperson (Eric Wong) |
|--|---------|------------|--------------------------|