## Ruby master - Feature #13608

### Add TracePoint#thread

#### 05/29/2017 05:21 AM - naruse (Yui NARUSE)

<table>
<thead>
<tr>
<th>Status</th>
<th>Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee</td>
<td></td>
</tr>
<tr>
<td>Target version</td>
<td></td>
</tr>
</tbody>
</table>

### Description

Add TracePoint#thread

rb_trace_arg_t, TracePoint's internal struct, already stores the thread which the event happened at, but there's not API to fetch it. How about adding an API to get the info.

```diff
diff --git a/test/ruby/test_settracefunc.rb b/test/ruby/test_settracefunc.rb
index 53ee82a229..65dbc938ab 100644
--- a/test/ruby/test_settracefunc.rb
+++ b/test/ruby/test_settracefunc.rb
@@ -699,6 +699,23 @@
    def test_tracepoint_enabled
        assert_equal(false, trace.enabled?)
    end
+    def test_tracepoint_thread
+        trace = TracePoint.new(:call, :return){|tp|
+            next if !target_thread?
+            next if tp.path != __FILE__
+            assert_equal(Thread.current, tp.thread)
+            case tp.event
+                when :call
+                    assert_raise(RuntimeError) {tp.return_value}
+                when :return
+                    assert_equal("xyzzy", tp.return_value)
+            end
+        }
+        trace.enable{
+            foo
+        }
+    end
+    def method_test_tracepoint_return_value obj
+        obj
+    end
+end
diff --git a/vm_trace.c b/vm_trace.c
index decb2c32e4..702b84e6e3 100644
--- a/vm_trace.c
+++ b/vm_trace.c
@@ -776,6 +776,12 @@
    return trace_arg->path;
 }
+VALUE
+rb_tracearg_thread(rb_trace_arg_t *trace_arg)
+{(  +    return trace_arg->th->self;
+)}
+static void
+fill_id_and_klass(rb_trace_arg_t *trace_arg)
+{  @ -913,6 +919,15 @@ tracepoint_attr_path(VALUE tpval)
+}  /*
+ * Thread of the event
```
+ */
+static VALUE
+tracepoint_attr_thread(VALUE tpval)
+{
+    return rb_tracearg_thread(get_trace_arg());
+
+} /*
+ */

* Return the name at the definition of the method being called *
+
static VALUE
@@ -1502,6 +1517,7 @@ Init_vm_trace(void)
    rb_define_method(rb_cTracePoint, "self", tracepoint_attr_self, 0);
    rb_define_method(rb_cTracePoint, "return_value", tracepoint_attr_return_value, 0);
    rb_define_method(rb_cTracePoint, "raised_exception", tracepoint_attr_raised_exception, 0);
+    rb_define_method(rb_cTracePoint, "thread", tracepoint_attr_thread, 0);
    rb_define_singleton_method(rb_cTracePoint, "stat", tracepoint_stat_s, 0);

History
#1 - 05/29/2017 05:51 AM - ko1 (Koichi Sasada)
use case?

#2 - 05/30/2017 07:03 AM - naruse (Yui NARUSE)
ko1 (Koichi Sasada) wrote:

    use case?

In which thread the event was happened.
Just thread id or something is also OK.

#3 - 05/30/2017 09:26 AM - Eregon (Benoit Daloze)
ko1 (Koichi Sasada) wrote:

    use case?

When debugging/monitoring with TracePoint, it seems useful to know in which thread the event happens.
It's also a trivial way to implement manually #13483, although not efficiently.

#4 - 05/30/2017 11:32 AM - ko1 (Koichi Sasada)
Eregon (Benoit Daloze) wrote:

    When debugging/monitoring with TracePoint, it seems useful to know in which thread the event happens.

    Thread#current is not enough?

    It's also a trivial way to implement manually #13483, although not efficiently.

I understand this side. But this issue doesn't solve performance problem.

#5 - 09/25/2017 06:29 AM - ko1 (Koichi Sasada)
- Status changed from Open to Rejected

Thread#current should be enough.