

Ruby master - Feature #16478

Fold symbol tables for tracking basic operation method redefinition changes into a single ID table

01/04/2020 01:37 AM - methodmissing (Lourens Naudé)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

References <https://github.com/ruby/ruby/pull/2812>

I stumbled onto these 2 tables (`vm_opt_method_table` and `vm_opt_mid_table`) while working on <https://github.com/ruby/ruby/pull/2809> and experimented at first with converting `vm_opt_mid_table` to an ID table for alignment with other method table concerns.

The fact that it stored a stable / constant `Qtrue` value also hinted at a possibility of collapsing it into one table mapping ID -> basic operation (BOP).

The `rb_method_definition_t::original_id` seems stable to me as even when aliased, it would not change the method definition.

I see this contract used in [dtrace](#) hooks, [trace instruction](#) hooks and [code coverage](#) as well.

For example:

```
static ID
frame_func_id(const rb_control_frame_t *cfp)
{
    const rb_callable_method_entry_t *me = rb_vm_frame_method_entry(cfp);

    if (me) {
        return me->def->original_id;
    }
    else {
        return 0;
    }
}
```

```
/*!
 * The original name of the current method.
 *
 * The function returns the original name of the method even if
 * an alias of the method is called.
 * The function can also return 0 if it is not in a method. This
 * case can happen in a toplevel of a source file, for example.
 *
```

```
 * \returns the ID of the name or 0
 * \sa rb_frame_callee
 * \ingroup defmethod
 */
```

```
ID
rb_frame_this_func(void)
{
    return frame_func_id(GET_EC()->cfp);
}
```

Thoughts on the viability of relying on `rb_method_definition_t::original_id` as a contract for tracking method redefinition?

This change is mostly about refactoring and alignment of using an ID table for method ID indexed concerns. It can potentially be beneficial during for example the Rails boot sequence using only 1 table (reduced inserts because only 1 table, ID table lookups are faster) for what is usually an event with a lot of pressure on method definition. I can measure further if the direction make sense.

History

#1 - 01/04/2020 01:42 AM - methodmissing (Lourens Naudé)

- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN)

- Tracker changed from Bug to Feature

Whoopsie, changing to the Feature tracker (from Bug)