

Ruby master - Feature #14792

Multiple RubyVM in one process to make real multi-threading.

05/29/2018 05:11 AM - HfCloud (Xiangyu Shi)

Status:	Feedback	
Priority:	Normal	
Assignee:		
Target version:		
Description		
<p>This is an old problem, maybe running multiple RubyVM is a good way, which can really run ruby code in parallel in one process (Each thread runs a RubyVM) and be compatible with old codes (If some codes depend on the GIL, they can still work properly). Some other programming languages have successfully realized it (It is said that lua uses this method, and my friend has made real multi-threading javascript by running multiple ES VM, threads can share data by passing pointers (safe problem is guaranteed by the programmer)).</p> <p>Why do I have this requirement :</p> <p>Really parallel threads needed, I choose ruby as the script language in my game engine (https://github.com/sxysxy/HfEngine). And in some situations, I don't need to do too much computation, but I hope they are really running in parallel. (for example, I divide rendering and logic into 2 parts and dispatch them onto two different threads, (putting rendering and logic in one thread is really not a good design)). The logic threads (logic threads are called 'producer', there can be many logic threads) will submit rendering commands list to the rendering thread ('consumer'). This process requires to be highly real-time (maybe 3000 times per second), and multi-process will fail to make it.</p> <p>Possibility:</p> <ol style="list-style-type: none">1. Some other languages have made it. (As I noted before)2. Though there are many global functions (such as 'rb_define_method'), but GET_THREAD() can help distinguish different threads so different RubyVM. When these functions are called, they can work on specific RubyVM of the calling thread.3. It should be compatible with all existed codes. (I think). including ruby codes and ruby native extensions' codes. (because it does not demand to change any interface)4. Each RubyVM can have its own GC system.5. There can be such ruby codes to create new RubyVM on new thread: <pre>vm = RubyVM.new("a.rb") #create a new RubyVM on new thread(not run immediately) #vm.set_global_variable("\$argv", [1, 2, 3]) #wrong! this may cause trouble when GC. a = [1, 2, 3] vm.set_global_variable("\$argv", a) #right! variable 'a' is managed by the RubyVM on the main thread, new thread do not hold a reference to a vm.exec #run it #... vm.shutdown #shutdown and destroy it</pre> <p>Hope to add this feature in new ruby.</p>		
Related issues:		
Related to Ruby master - Bug #13188: Reinitialize Ruby VM.		Rejected
Related to Ruby master - Feature #7003: Please decide. MVM to be with 2.0?		Rejected 09/10/2012

History

#1 - 05/29/2018 05:18 AM - HfCloud (Xiangyu Shi)

- Description updated

#2 - 05/29/2018 05:19 AM - HfCloud (Xiangyu Shi)

- Description updated

#3 - 05/29/2018 05:23 AM - shyouhei (Shyouhei Urabe)

- Related to Bug #13188: Reinitialize Ruby VM. added

#4 - 05/29/2018 05:23 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #7003: Please decide. MVM to be with 2.0? added

#5 - 05/29/2018 05:24 AM - HfCloud (Xiangyu Shi)

- Description updated

#6 - 05/29/2018 05:26 AM - shyouhei (Shyouhei Urabe)

This is what we call the MVM feature. There has been rich amount of efforts to make it possible and still not available in a production-ready manner.

#7 - 05/29/2018 05:28 AM - HfCloud (Xiangyu Shi)

shyouhei (Shyouhei Urabe) wrote:

This is what we call the MVM feature. There has been rich amount of efforts to make it possible and still not available in a production-ready manner.

Is there any system API, which can divide threads' memory and allow them share data by passing pointers directly?

#8 - 05/29/2018 05:46 AM - shyouhei (Shyouhei Urabe)

HfCloud (Xiangyu Shi) wrote:

shyouhei (Shyouhei Urabe) wrote:

This is what we call the MVM feature. There has been rich amount of efforts to make it possible and still not available in a production-ready manner.

Is there any system API, which can divide threads' memory and allow them share data by passing pointers directly?

Not yet. Ko1 is working on this area. See also <https://www.youtube.com/watch?v=mjzmUUQWqco>

#9 - 05/29/2018 05:56 AM - HfCloud (Xiangyu Shi)

shyouhei (Shyouhei Urabe) wrote:

HfCloud (Xiangyu Shi) wrote:

shyouhei (Shyouhei Urabe) wrote:

This is what we call the MVM feature. There has been rich amount of efforts to make it possible and still not available in a production-ready manner.

Is there any system API, which can divide threads' memory and allow them share data by passing pointers directly?

Not yet. Ko1 is working on this area. See also <https://www.youtube.com/watch?v=mjzmUUQWqco>

Oh...I have seen the relative issues and know you are trying making it. Appreciate your work!

It's really a lesson indicating that don't abuse global variables :)

#10 - 05/29/2018 07:33 AM - naruse (Yui NARUSE)

- Status changed from Open to Feedback

#11 - 05/29/2018 09:12 AM - Ksec (E C)

HfCloud (Xiangyu Shi) wrote:

shyouhei (Shyouhei Urabe) wrote:

This is what we call the MVM feature. There has been rich amount of efforts to make it possible and still not available in a production-ready manner.

Is there any system API, which can divide threads' memory and allow them share data by passing pointers directly?

Slightly off topic, have you consider mruby?

#12 - 05/29/2018 03:12 PM - HfCloud (Xiangyu Shi)

Ksec (Ed Chick) wrote:

HfCloud (Xiangyu Shi) wrote:

shyouhei (Shyouhei Urabe) wrote:

This is what we call the MVM feature. There has been rich amount of efforts to make it possible and still not available in a production-ready manner.

Is there any system API, which can divide threads' memory and allow them share data by passing pointers directly?

Slightly off topic, have you consider mruby?

Sorry for so late my reply...

I do not know mruby very much, I saw the examples using it in C language, It looks nice, according to its interfaces I think it can realize multiple RubyVM(It uses a mrb_state instead of global variables and functions).

But I still have some hesitation: I have written many MRI ruby2.5 codes in my project, (some core functions are also written in ruby), native extension codes are even more. Can mruby be compatible with my codes?(in syntax and behaviour). What's more, though MRI ruby2.5 looks much heavy than mruby, MRI ruby2.5 has great performance. On my device, (Pentium E5500 & GTX650), my game engine can reach 3500fps. I know ruby's development team have made much effort to optimize the performance and it truly does well. How about mruby?