

Ruby - Feature #11518

Queue enhancement - promote! and promote_all!

09/09/2015 03:04 PM - jonathanscruz (Jonathan Cruz)

Status:	Assigned	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
Description		
<p>I'm submitting a patch to enhance the Queue class by adding the methods Queue#promote! and Queue#promote_all!. These methods require a block that accepts queue elements. Elements for which the block returns a truthy value are moved to the 'front' of the queue. Queue#promote! only applies to the first such element and Queue#promote_all! applies to all matching elements (preserving their relative order).</p> <p>Motivation for this enhancement: Our project has several worker threads working on long-running work units in a queue, trying to find a 'match'. The queue is pre-sorted with the most likely matches at the front and least likely at the back. However, there are cases where as we work on some elements, we gain new data that certain elements are more likely to match than we originally thought. We need a way to promote these elements to the front of the queue.</p>		

History

#1 - 09/12/2015 09:19 AM - funny_falcon (Yura Sokolov)

First: you forgot to lock queue. Multithreading will bite you.

Second: if you want custom behaviour why don't you make custom structure?
Till 2.0 Queue were implemented in ruby. If your tasks are heavy, then you won't notice difference.

#2 - 09/17/2015 08:40 PM - jonathanscruz (Jonathan Cruz)

Yura Sokolov wrote:

First: you forgot to lock queue. Multithreading will bite you.

Second: if you want custom behaviour why don't you make custom structure?
Till 2.0 Queue were implemented in ruby. If your tasks are heavy, then you won't notice difference.

First: You're right. My C skills are a bit rusty and I wasn't sure how to do that. Then I think I just forgot about it before submitting this. I'll update the patch, but if you could point me to an example of locking, it'll save me some time.

Second: That is exactly what we did. We recreated the Ruby implementation of Queue in our project and then subclassed that to add #promote! and #promote_all! methods (as well as the conditional version of #pop in <https://bugs.ruby-lang.org/issues/11517>). That works fine for our needs. However, this will eventually cause us to become out of synch with the core Ruby version of Queue. We would rather not have to remember to update our implementation of Queue every time we upgrade Ruby.

We can get by without this, but it would be nice to expose some limited methods to change the order of the queue. Obviously, you cannot expose the underlying array object and remain threadsafe, but as it is, that array is completely inaccessible even by child classes.

#3 - 04/03/2024 03:50 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

Files

ruby_queue_promote.patch	2.82 KB	09/09/2015	jonathanscruz (Jonathan Cruz)
--------------------------	---------	------------	-------------------------------