

Ruby master - Feature #13767

add something like python's buffer protocol to share memory between different narray like classes

07/25/2017 03:56 PM - dsisnero (Dominic Sisneros)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>In order for ruby to be used in more scientific or machine learning applications, it will be necessary to be able to use more memory efficient data structures. Python has a concept called Buffer Protocol that allows different representations to utilize the memory without copying the data with memory views. This is a proposal to add something similar to ruby as a c-api</p> <p>https://jakevdp.github.io/blog/2014/05/05/introduction-to-the-python-buffer-protocol/</p> <p>The Python buffer protocol, also known in the community as PEP 3118, is a framework in which Python objects can expose raw byte arrays to other Python objects. This can be extremely useful for scientific computing, where we often use packages such as NumPy to efficiently store and manipulate large arrays of data. Using the buffer protocol, we can let multiple objects efficiently manipulate views of the same data buffers, without having to make copies of the often large datasets.</p> <p>Here, for example, we'll use Python's built-in array object to create an array:</p> <pre>In [1]: import array A = array.array('i', range(10)) A Out[1]: array('i', [0, 1, 2, 3, 4, 5, 6, 7, 8, 9])</pre> <p>Note that the array object is different than a Python list, in that it stores the data as a contiguous block of integers. For this reason, the data are stored much more compactly than a list, and certain operations can be performed much more quickly.</p> <p>array objects by themselves are not particularly useful, but a similar type of object can be found in the numpy array. Because both Python's array and NumPy's ndarray objects implement the buffer protocol, it's possible to seamlessly pass data between them using views – that is, without the need to copy the raw data:</p>	
Related issues:	
Related to Ruby master - Feature #14722: python's buffer protocol clone	Closed

Associated revisions

Revision 56012d2f - 09/25/2020 12:16 PM - Kenta Murata

NEWS.md: Add memory view entry

The memory view interface added at 890bc2cdde is experimental new C-API set. This feature permits extension libraries to share a memory area that contains such a numerical array and a bitmap image. This is designed by referring to Python's buffer protocol.

[[Feature #13767]]

[[Feature #14722]]

History

#1 - 07/26/2017 05:29 PM - shevegen (Robert A. Heiler)

I am not sure I understand the proposal, also partially due to python's confusing naming scheme such as List and arrays. The API is also abysmal ... `array.array()`.

I have nothing at all against anything that makes ruby more useful in scientific applications though. Perhaps we can get some ruby people that work on the sci-ruby and similar project to comment here. <https://github.com/SciRuby>

I think that actually one reason or ruby to not be used as much as python is that python has more developers in the scientific world. I have seen lots of people writing C++ software and then adding of course only one language, which tends to be python. This may have been perl in the past but python seemed to have gained the foothold some years ago.

#2 - 10/10/2017 02:23 PM - dsisnero (Dominic Sisneros)

The naming theme is not what is important. What is important is to have an api to share memory buffers between libraries.

SEE: <https://jeffknupp.com/blog/2017/09/15/python-is-the-fastest-growing-programming-language-due-to-a-feature-youve-never-heard-of/>

Python's Buffer Protocol: The #1 Reason Python Is The Fastest Growing Programming Language Today

The buffer protocol was (and still is) an extremely low-level API for direct manipulation of memory buffers by other libraries. These are buffers created and used by the interpreter to store certain types of data (initially, primarily "array-like" structures where the type and size of data was known ahead of time) in contiguous memory.

The primary motivation for providing such an API is to eliminate the need to copy data when only reading, clarify ownership semantics of the buffer, and to store the data in contiguous memory (even in the case of multi-dimensional data structures), where read access is extremely fast. Those "other libraries" that would make use of the API would almost certainly be written in C and highly performance sensitive. The new protocol meant that if I create a NumPy array of ints, other libraries can directly access the underlying memory buffer rather than requiring indirection or, worse, copying of that data before it can be used.

#3 - 06/17/2020 07:38 AM - mrkn (Kenta Murata)

- Related to Feature #14722: python's buffer protocol clone added

#4 - 09/25/2020 12:06 PM - mrkn (Kenta Murata)

- Status changed from Open to Closed

Fixed by <https://github.com/ruby/ruby/commit/890bc2cdde4097390f3b71dfeaa36dd92ee0afe2>

#5 - 09/27/2020 10:12 AM - Eregon (Benoit Daloze)

[mrkn \(Kenta Murata\)](#) Is the native memory layout introduced in Ruby compatible with Python?

If it is/was it might be useful to e.g. be able to access a NumPy array from Ruby without copying (e.g. in PyCall.rb)