gpu-buffer-interface Documentation

Release 0.1

NumFOCUS

July 20, 2013

CONTENTS

Contents:

CHAPTER

ONE

OPENCL/CUDA BUFFER INTERFACE

1.1 Python interface

This document specifies an object as implementing the buffer interface by having a number of attributes and methods. If an implementation chooses to implement the buffer interface as a temporary view, the way to obtain the object implementing the `BufferInterface` is implementation-defined. The user may call `BufferInterface.release()` to explicitly release any resources associated with the view.

class **`BufferInterface`**

`buffer`

OpenCL: An object that has a `int_ptr` attribute which is a `int` representing the `cl_mem` pointer.

CUDA: An object that is castable to an `int` representing the numerical value of the CUDA device pointer.

`offset`

OpenCL: Relative offset from the beginning of :attr‘buffer‘ to the start of the array data.

CUDA: Always zero.

Todo

We should also define a way to access a host pointer for the relevant memory area, if available.

`dtype`

Todo

What type system? `numpy.dtype`? lltypes?

`shape`

A shape tuple, just like :attr‘numpy.ndarray.shape‘.

`strides`

A strides tuple, just like :attr‘numpy.ndarray.strides‘.

`strides`

A strides tuple, just like :attr‘numpy.ndarray.strides‘.

`release``()`

If *self* is a proxy to the actual object holding the data then either garbage collection or an explicit *release* call will free the resources occupied by the proxy. Otherwise, this method is a no-op.

Note: I don't think it's appropriate for the interface to mandate the use of `pyopencl` or `pycuda`. It's of course intended that the interface play well with the two, but I believe this is about specifying an interface, not an implementation. That's how I arrived at the spec above.

1.2 CPython interface

Todo

I would suggest that the CPython-specific interface be as minimal as possible, just providing a hatch to get to the low-level C interface.

1.3 C interface

Todo

Frédéric/Arnaud–could you paste in the interface that your array object obeys as a starting point?

CHAPTER

TWO

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*