Apache Arrow Tensor Arrays

A toolchain for tensor transport and storage

Rok Mihevc, Alenka Frim

Apache Arrow committers

January 24, 2025

Overview

1. Fixed Shape Tensor
2. Variable Shape Tensor
3. FixedShapeTensorArray and NumPy ndarray
4. DLPack protocol

Fixed Shape Tensor

- Type parameters: `data type` and `shape` of individual tensor elements
- First dimension of the tensor is the length of the array
- Data are stored as `FixedSizeList`
- Optional parameters: `dim_names`, `permutation`
- Elements in a fixed shape tensor extension array are stored in row-major/C-contiguous order

Fixed Shape Tensor - memory layout

type: extension<arrow.fixed_shape_tensor[value_type=int32, shape=[2,2]]>
pyarrow array: [[[1,2,3,4],[10,20,30,40],[100,200,300,400]]]

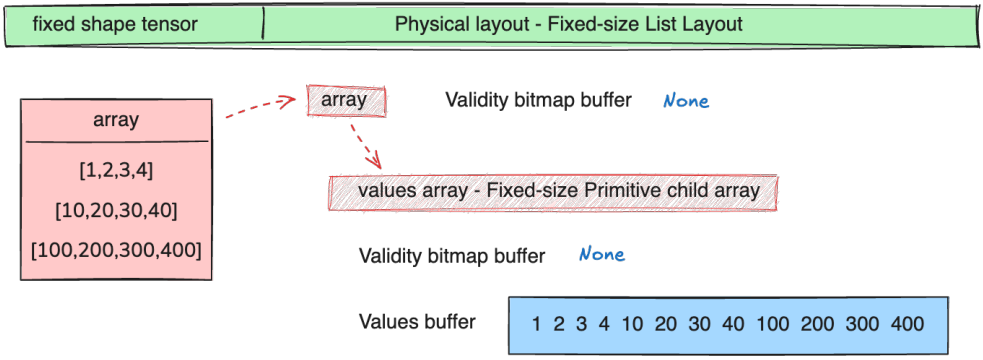

Figure: Fixed Shape Tensor memory layout

Variable Shape Tensor

- Type parameters: `data type`
- First dimension of the tensor is the length of the array
- Data are stored as `StructArray`
  - `data` is a List holding tensor elements
  - `shape` is a `FixedSizeList<int32>[ndim]`, `ndim`
- Optional parameters: `dim_names`, `permutation` and `uniform_shape`
- Elements in a fixed shape tensor extension array are stored in row-major/C-contiguous order

Variable Shape Tensor - memory layout

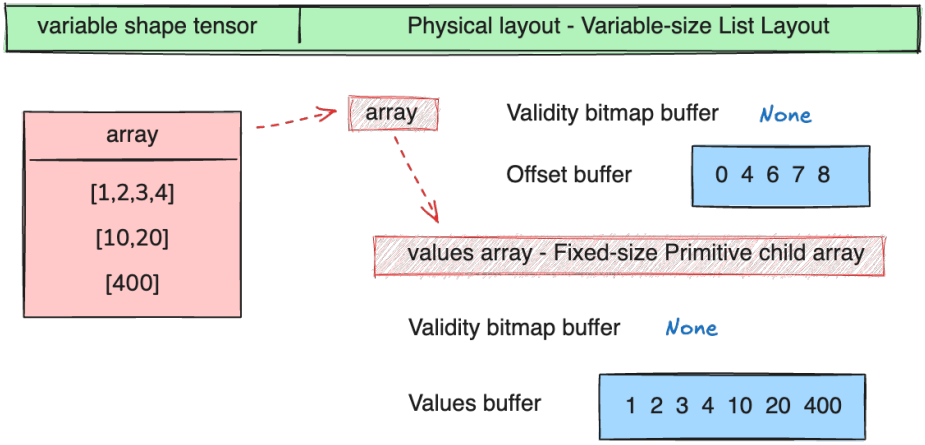

Figure: Fixed Shape Tensor memory layout

Create a FixedShapeTensorArray

Examples

```
>>> import pyarrow as pa
>>> tensor_type = pa.fixed_shape_tensor(pa.int32(), (2, 2))
>>> arr = [[1, 2, 3, 4], [10, 20, 30, 40], [100, 200, 300, 400]]
>>> storage = pa.array(arr, pa.list_(pa.int32(), 4))
>>> tensor_array = pa.ExtensionArray.from_storage(tensor_type, storage)
```

Create a FixedShapeTensorArray

Examples

```
>>> tensor_array
<pyarrow.lib.FixedShapeTensorArray object at ...>
[
  [
    1,
    2,
    3,
    4
  ],
  ...
]
```

Move to NumPy ndarray

Examples

```
>>> tensor_array.to_numpy_ndarray()
array([[[  1,   2],
        [  3,   4]],

       [[ 10,  20],
        [ 30,  40]],

       [[100, 200],
        [300, 400]]], dtype=int32)
```

Move back to PyArrow

Examples

```
>>> pa.FixedShapeTensorArray.from_numpy_ndarray(
...       tensor_array.to_numpy_ndarray()
... )
<pyarrow.lib.FixedShapeTensorArray object at ...>
[
  [
    1,
    2,
    3,
    4
  ],
  ...
]
```

DLPack protocol

- Enables device aware data interchange between array/tensor libraries
- Currently producer side of DLPack implemented for pyarrow Array
- Future plan: Implementation of producing and consuming part for `Tensor` class and `FixedShapeTensorArray.to_tensor()` method to connect `FixedShapeTensorArray` with libraries supporting DLPack (NumPy, CuPy, Tensorflow, PyTorch, JAX, MXNet, TVM, mpi4py, Paddle, etc.)

The End