PyTorch/XLA SPMD

Distributed 101

Agenda

1. What’s GSPMD
2. Mesh & Partition Spec
3. Basic ideas
4. Colab example

GSPMD

1. https://arxiv.org/abs/2105.04663
2. User only express sharding intention, let compiler shard the tensor for you.
3. User don’t need to shard every tensor, compiler will propagate the sharding for the user.
4. Collective ops(all_gather, reduce_scatter etc) will be added after compilation

Mesh

1. Represent the device topology
2. Product of the Mesh shape is the total number of devices

```
num_devices = xr.global_runtime_device_count()
mesh_shape = (num_devices, 1)
device_ids = np.array(range(num_devices))
mesh = Mesh(device_ids, mesh_shape, ('data', 'model'))
```

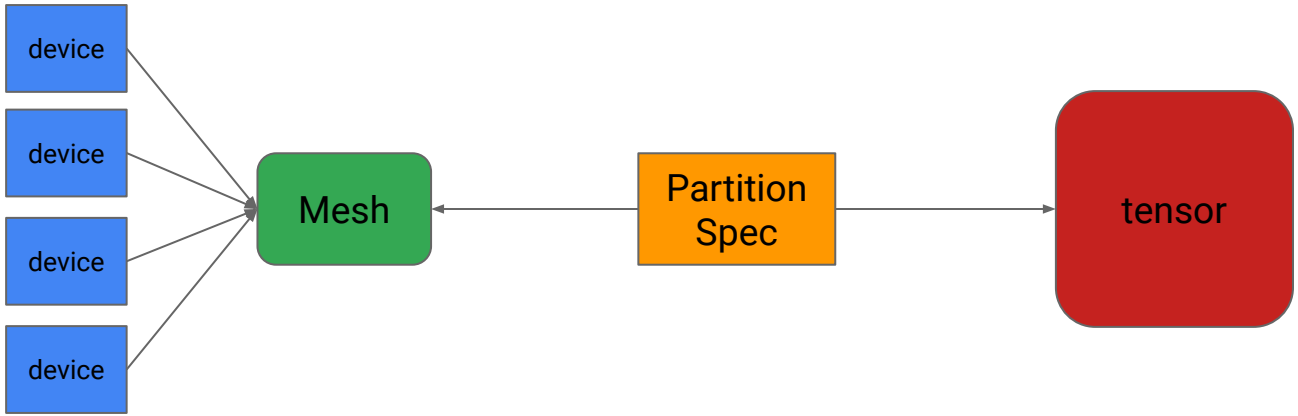
Partition Spec

1. Represent how Mesh is mapped to the tensor
2. Partition Spec has the same rank as the tensor

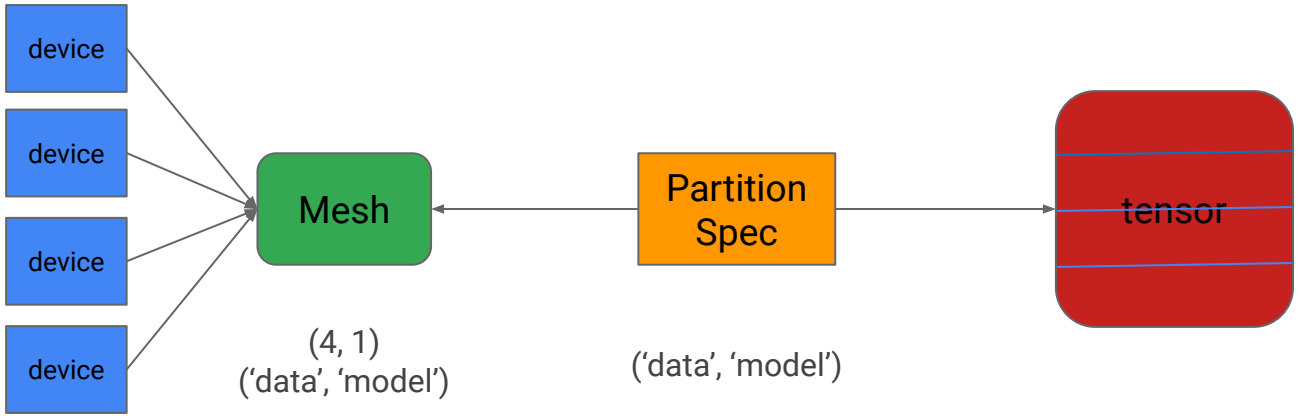
```
t = torch.randn(8, 4).to(xm.xla_device())


partition_spec = ('data', 'model')
xs.mark_sharding(t, mesh, partition_spec)
```

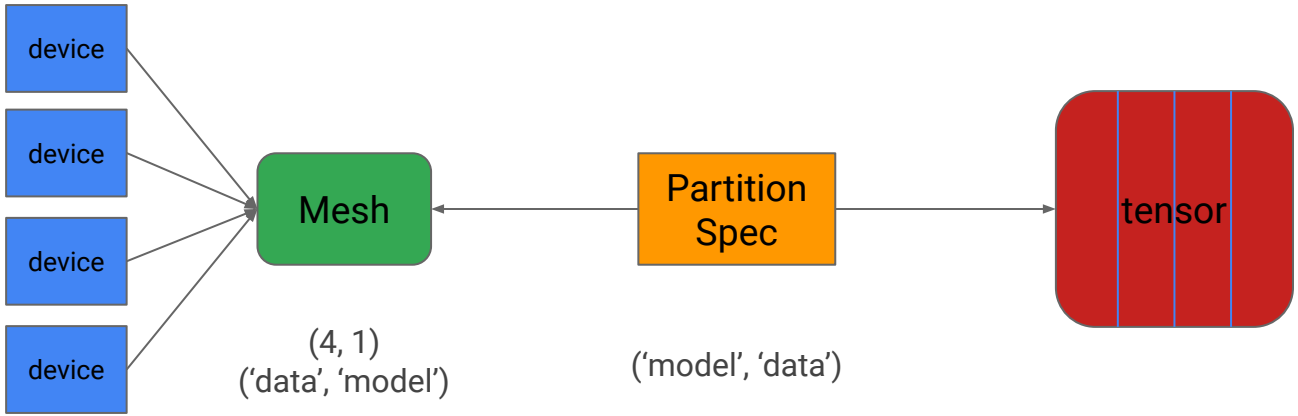
Basic ideas



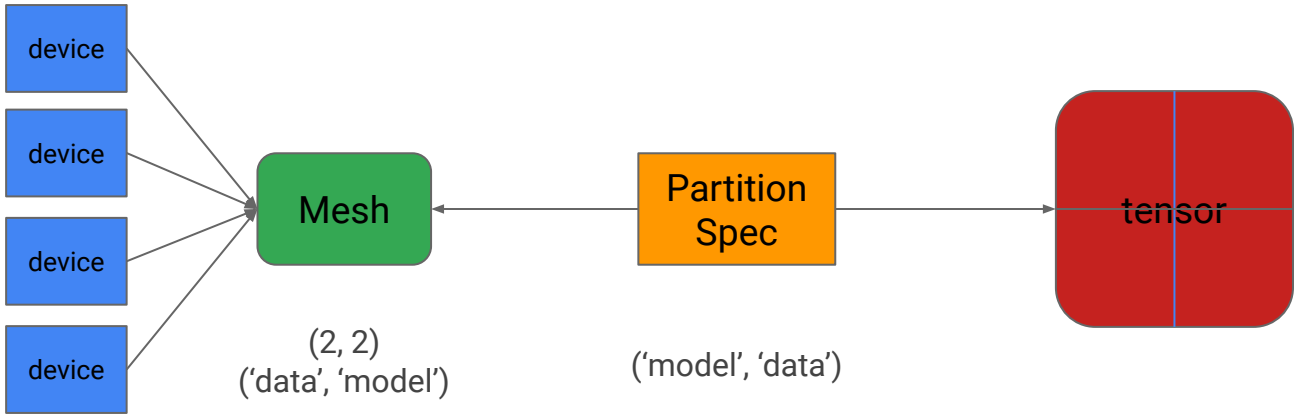
Basic ideas


device
device
device
device
Mesh
(4, 1)
('data', 'model')
Partition Spec
('data', 'model')
tensor

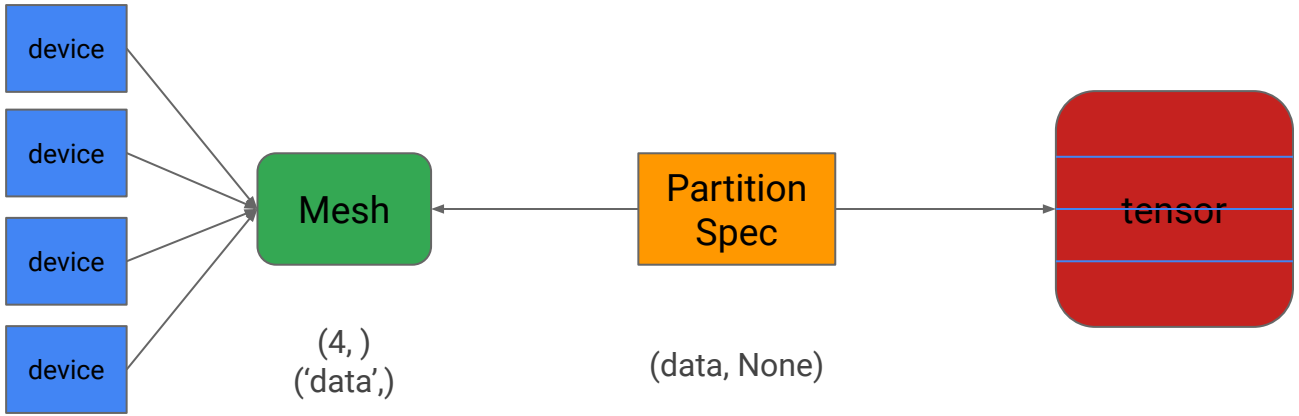
Basic ideas


device
device
device
device
Mesh
(4, 1)
('data', 'model')
Partition Spec
('model', 'data')
tensor

Basic ideas


device
device
device
device
Mesh
(2, 2)
('data', 'model')
Partition Spec
('model', 'data')
tensor

Basic ideas


device
device
device
device
Mesh
(4,)
('data',)
Partition Spec
(data, None)
tensor