DLI Accelerated Data Science Teaching Kit

Lecture 13.1 - Using Dask and UCX with RAPIDS

DASK

Introduction

A flexible library for parallel computing in Python.

DASK comprises of two major parts:

- **Dynamic task scheduling**

  - Optimized for interactive computational workloads.

- **"Big Data" collections**

  - Like parallel arrays, dataframes, and lists that extend common interfaces like NumPy, Pandas, or Python iterators to larger-than-memory or distributed environments

 - Run on top of dynamic task scheduler

DASK

Why DASK?

- **Familiar**: Provides parallelized NumPy array and Pandas DataFrame objects

- **Flexible**: Provides a task scheduling interface for more custom workloads and integration with other projects.

- **Native**: Enables distributed computing in pure Python with access to the PyData stack.

- **Fast**: Operates with low overhead, low latency, and minimal serialization necessary for fast numerical algorithms

- **Scales up**: Runs resiliently on clusters with 1000s of cores

- **Scales down**: Trivial to set up and run on a laptop in a single process

- **Responsive**: Designed with interactive computing in mind, it provides rapid feedback and diagnostics to aid humans
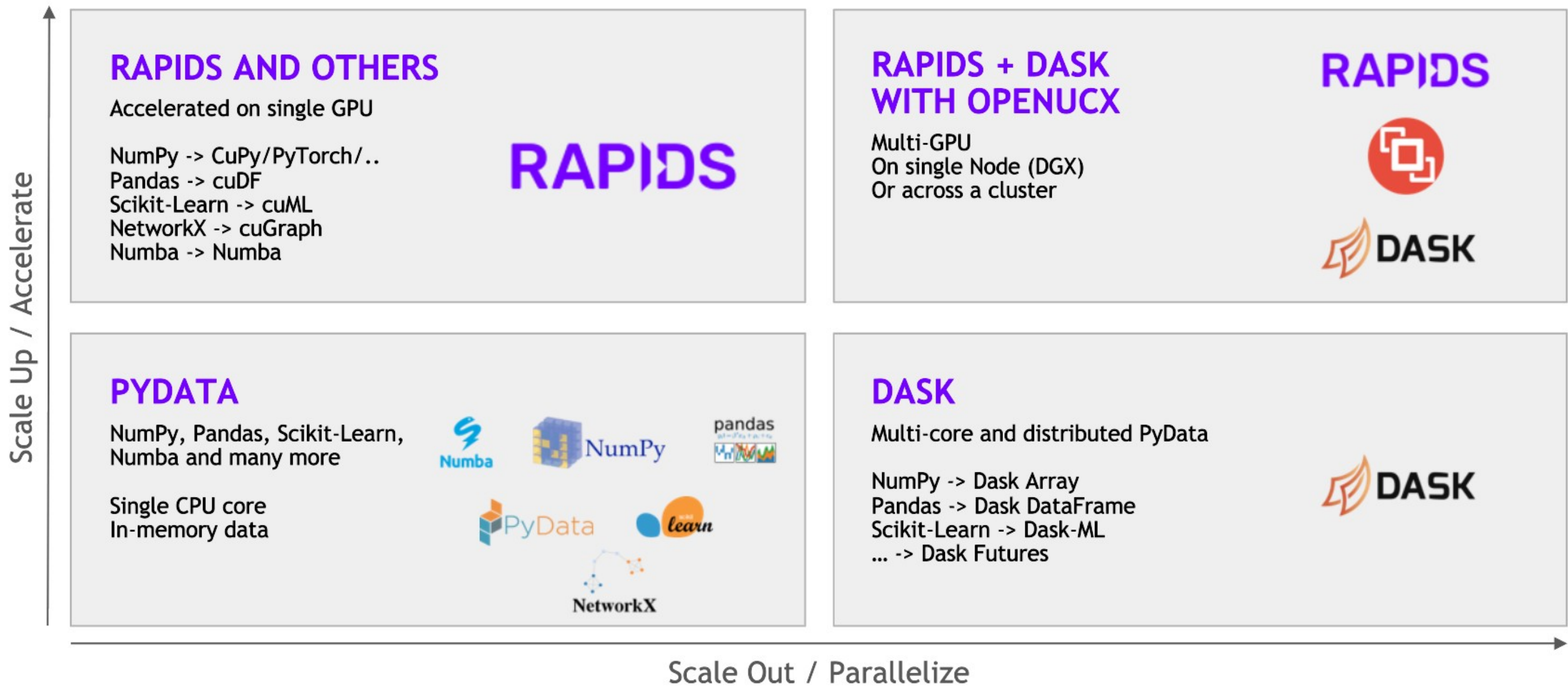
DASK and RAPIDS

Scaling up and Scaling out

Image Credit: NVIDIA
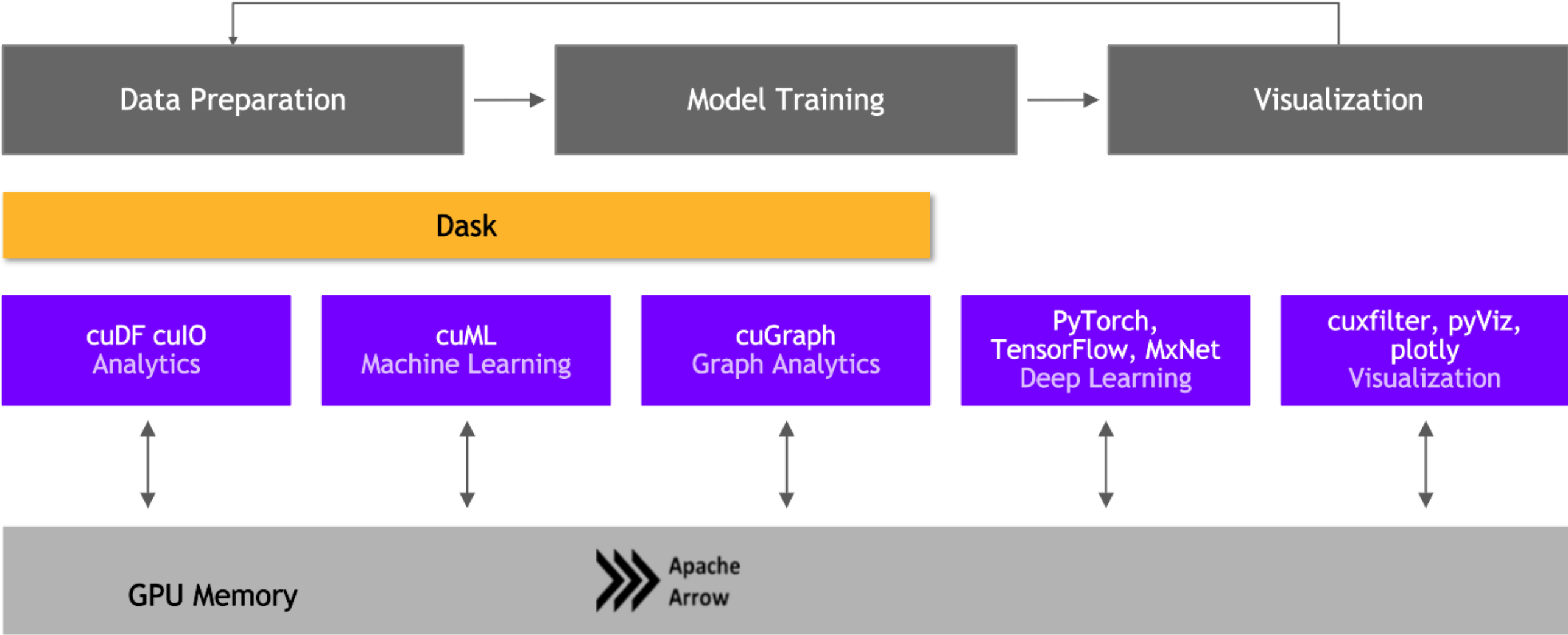
DASK and RAPIDS

Scaling up and Scaling out

DASK + **RAPIDS** = Scalable distributed performance

Scale Up / Accelerate (vertical axis)

RAPIDS AND OTHERS
Accelerated on single GPU

NumPy -> CuPy/PyTorch/..
Pandas -> cuDF
Scikit-Learn -> cuML
NetworkX -> cuGraph
Numba -> Numba

RAPIDS + DASK WITH OPENUCX
Multi-GPU
On single Node (DGX)
Or across a cluster

PYDATA
NumPy, Pandas, Scikit-Learn, Numba and many more

Single CPU core
In-memory data

DASK
Multi-core and distributed PyData

NumPy -> Dask Array
Pandas -> Dask DataFrame
Scikit-Learn -> Dask-ML
... -> Dask Futures

Scale Out / Parallelize (horizontal axis)

Image Credit: NVIDIA

UCX

Introduction

- Unified Communication X is a framework that facilitates an easy and efficient way to construct widely used high-performance computing (HPC) protocols

    - Example protocols: MPI tag matching, RMA operations, rendezvous protocols, stream, fragmentation, remote atomic operations, etc.

- Accelerated networking library designed for low-latency, high-bandwidth transfers for host and GPU device memory objects
- UCX-Py is the python interface for UCX which is a high-level library that is easy for users to interact with

UCX with RAPIDS

Improved performance

- Leverage UCX-Py to handle communication bottleneck for distributed computation

- UCX-Py allows RAPIDS to use hardware interconnects like NVLink and InfiniBand

- Applications using DASK can easily make use of UCX with small code changes:

  - Instead of the default TCP protocol, use UCX protocol!

- To enable to UCX protocol in applications using Dask, set:

  - `cluster = LocalCUDACluster(protocol= "ucx", enable_tcp_over_ucx= True)`

- Efficient passing of single message passing GPU objects between endpoints

DLI Accelerated Data Science Teaching Kit

Thank You