



CuPy

BIRON Victoire / JARDEL Jean / METRAL-BOFFOD Benjamin



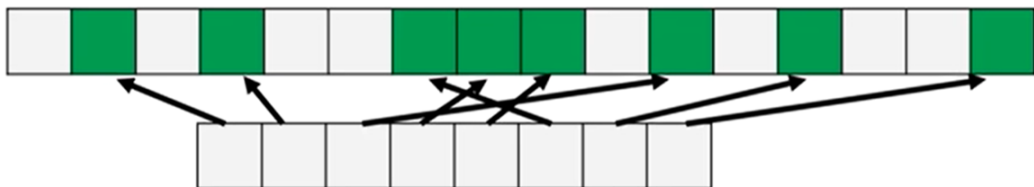
CuPy

- CuPy is an open-source NumPy for NVIDIA GPU
- Python users can easily write CPU/GPU code
- Existing NumPy code can be accelerated thanks to GPU and CUDA libraries



Why is NumPy Faster? - Contiguous Memory

Lists

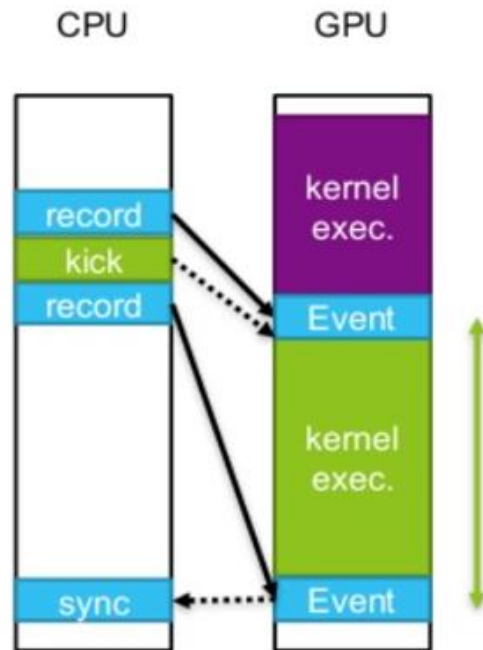


NumPy



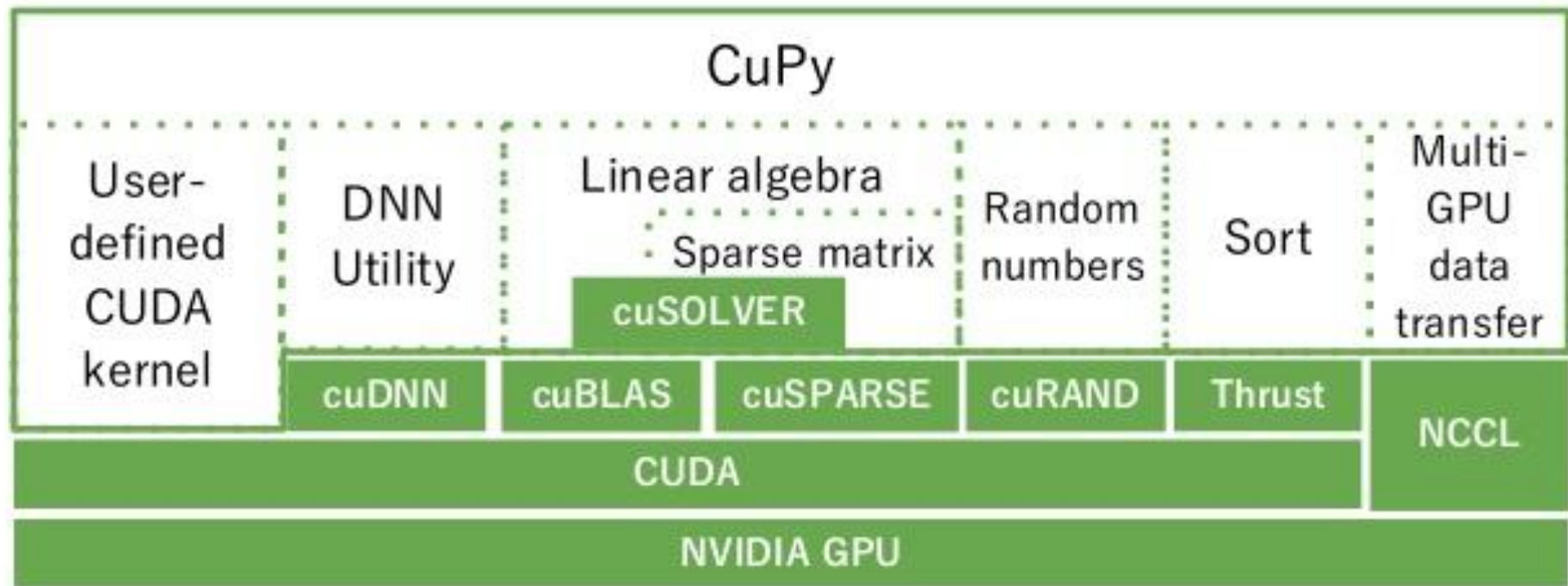
Benefits:

- SIMD Vector Processing
- Effective Cache Utilization





Inside CuPy



Requirement to install CuPy

Pip install cupy require you to install several other packages to make it run :

cuBLAS, cuDNN, cuRand, cuSolver, cuSPARSE, cuFFT and NCCL

- Required : numpy (python 3.6)
- Upgrade pip
- Install CUDA v11 on nvidia website

- `pip install numpy`
- `python -m pip install -U setuptools`
`pip`
- <https://developer.nvidia.com/cuda-downloads>

Finalise CuPy installation

Several Packages installed:

- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.1\libnvvp
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.1\bin
- C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common

Pre-built binaries!

\$ pip install cupy-cuda80	(Binary Package for CUDA 8.0)
\$ pip install cupy-cuda90	(Binary Package for CUDA 9.0)
\$ pip install cupy-cuda91	(Binary Package for CUDA 9.1)
\$ pip install cupy-cuda92	(Binary Package for CUDA 9.2)
\$ pip install cupy-cuda100	(Binary Package for CUDA 10.0)

cuDNN and NCCL included!

Add environment path:

Name: *CUDA_PATH*

Directory: *C:\Program Files\NVIDIA
GPU Computing
Toolkit\CUDA\v11.1*

CuPy is...

a library to provide NumPy-compatible features with GPU



CuPy

```
import cupy as cp
x_gpu = cp.zeros((10,))
W_gpu = cp.zeros((10, 5))
y_gpu = cp.dot(x_gpu,
W_gpu)
```

```
y_gpu = cp.asarray(y_cpu)
```



NumPy

```
import numpy as np
X_cpu = np.zeros((10,))
W_cpu = np.zeros((10, 5))
y_cpu = np.dot(x_cpu,
W_cpu)
```

```
y_cpu = cp.asnumpy(y_gpu)
```