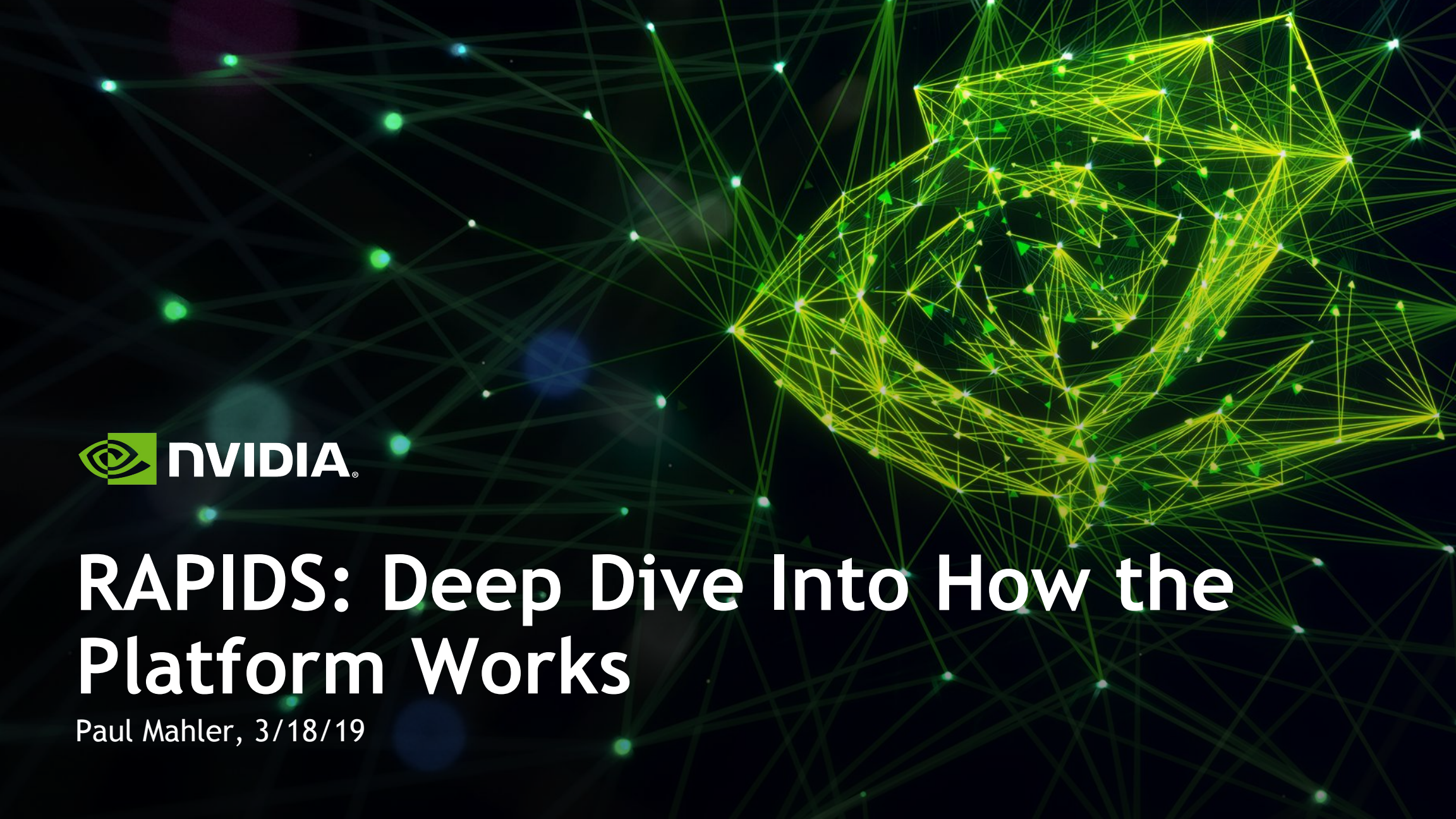


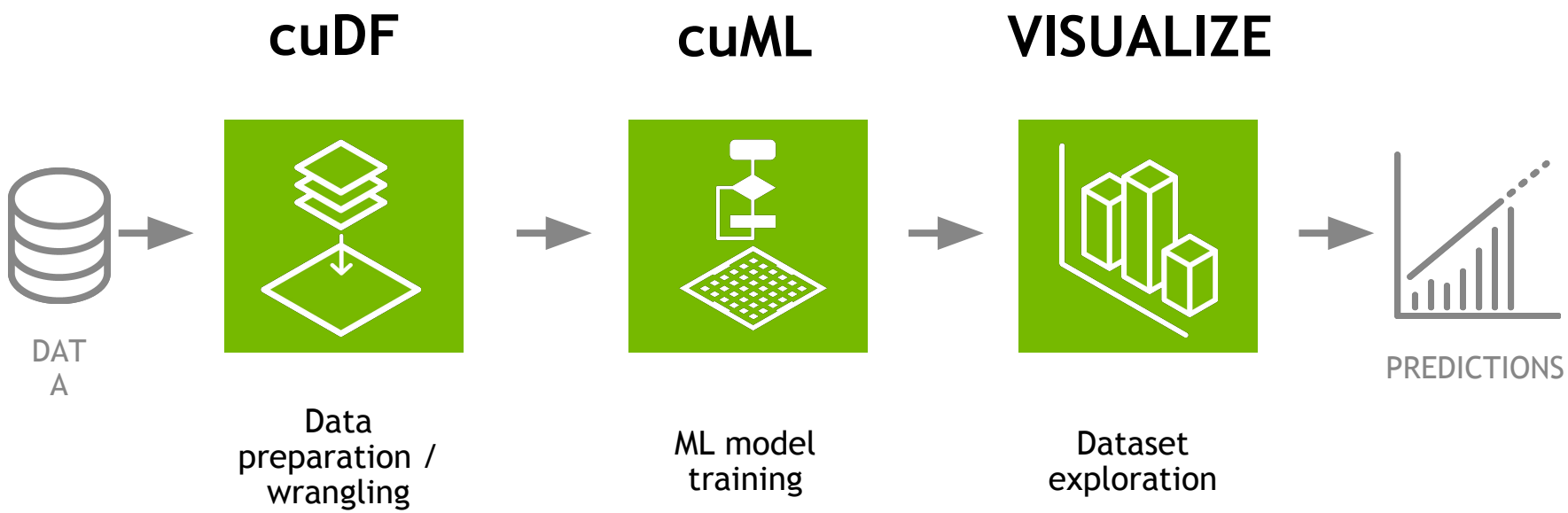
RAPIDS: Deep Dive Into How the Platform Works

Paul Mahler, 3/18/19


Introduction to
RAPIDS

DATA SCIENCE WORKFLOW WITH RAPIDS

Open Source, GPU-accelerated ML Built On CUDA



WHAT IS RAPIDS?

The New GPU Data Science Pipeline

rapids.ai

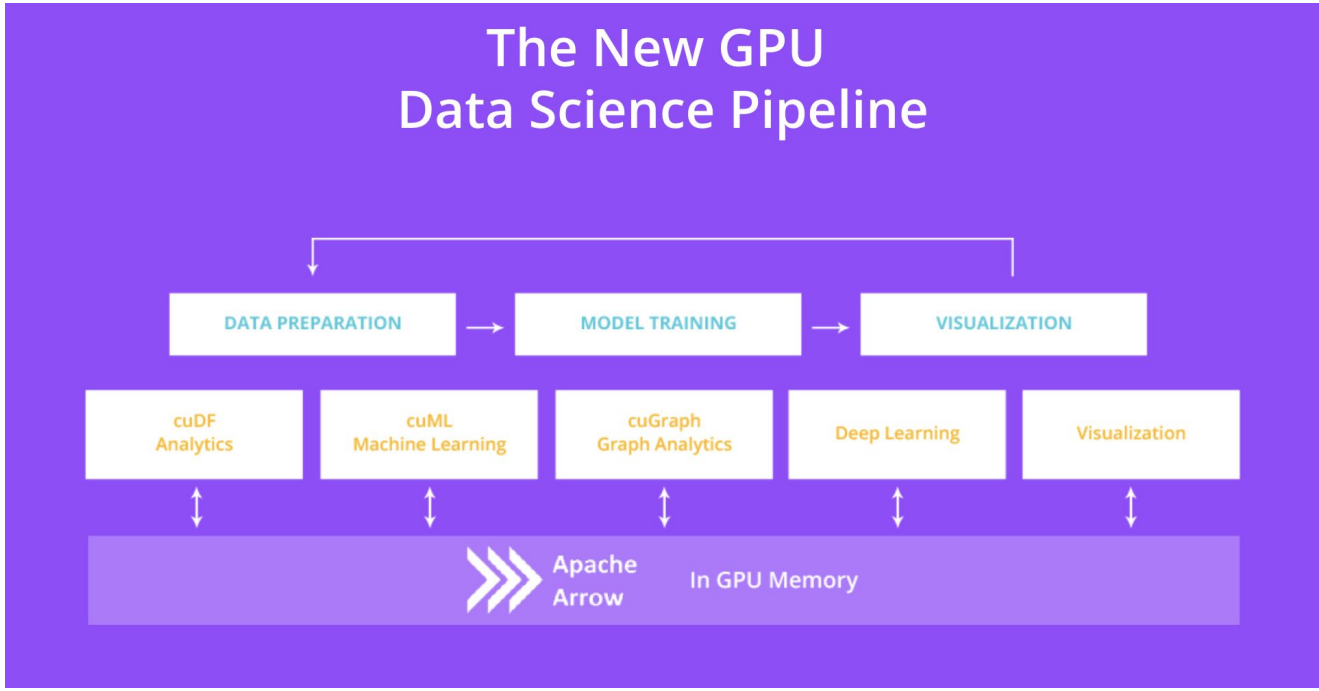
Suit of open-source, end-to-end data science tools

Built on CUDA

Pandas-like API for data cleaning and transformation

Scikit-learn-like API

A unifying framework for GPU data science



“CLASSIC” MACHINE LEARNING

The daily work of most data scientists

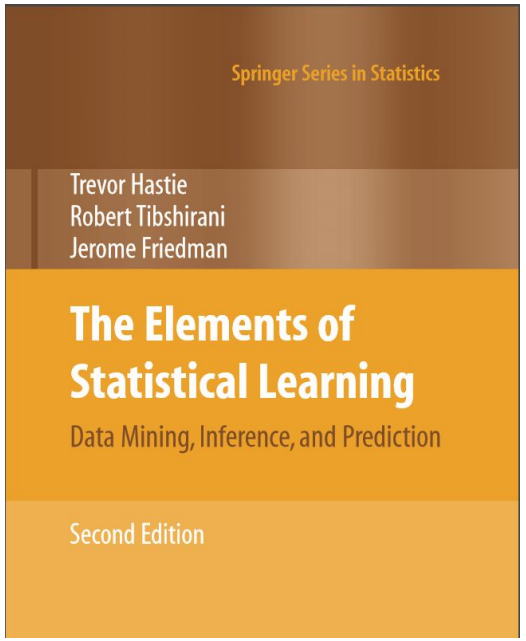
Comprehensible to average data scientists and analysts

Higher level of interpretability

Solutions for unlabeled data

Techniques such as regression and decision trees, clustering

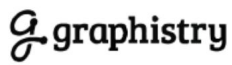
Scikit-learn



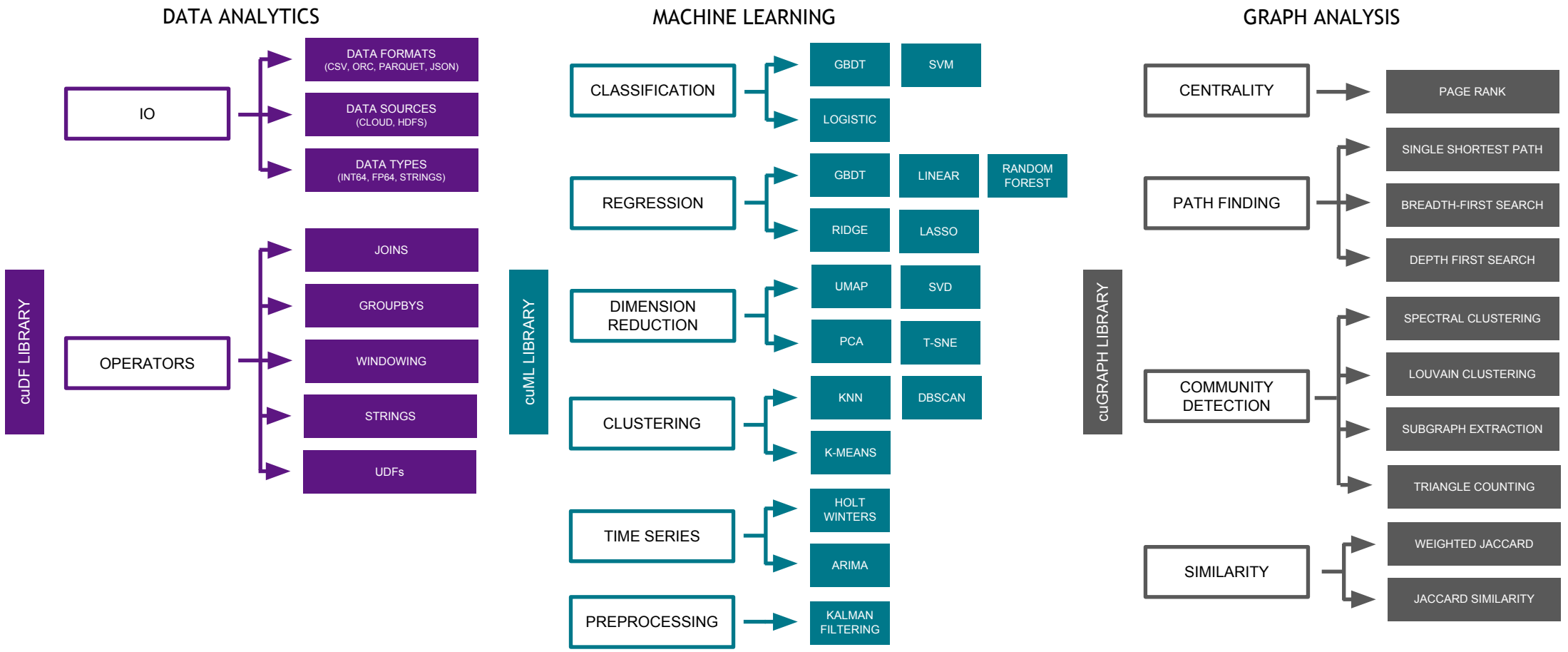
Ecosystem Partners

Community Contributors

Ecosystem Partners



RAPIDS ROADMAP



UP TO 5-15X SPEEDUP

UP TO 10-20X SPEEDUP

UP TO 100-500X SPEEDUP

RAPIDS PREREQUISITES

See more at rapids.ai

- NVIDIA Pascal™ GPU architecture or better
- CUDA 9.2 or 10.0 compatible NVIDIA driver
- Ubuntu 16.04 or 18.04

- Docker CE v18+
- nvidia-docker v2+

Get RAPIDS

RAPIDS is available as conda or pip packages, docker images, and from source builds. Use the tool below to select your preferred method, packages, and environment to install RAPIDS. Certain combinations may not be possible and are dimmed automatically. Be sure to review the prerequisites section for more details about requirements to use RAPIDS.

	☑ Preferred		Beta	
Method	**Conda**	Docker	Pip	Source
Packages	cuDF	cuML	**cuDF & cuML**	
Linux	**Ubuntu 16.04**	Ubuntu 18.04	CentOS 7	
Python	**Python 3.6**		Python 3.7	
CUDA	**CUDA 9.2**		CUDA 10.0	

Command

```
conda install -c nvidia -c rapidsai -c pytorch -c numba -c conda-forge \
    -c defaults cudf=0.5 cuml=0.5 python=3.6
```

COPY COMMAND

CONDA DETAILS

GETTING STARTED RESOURCES

Rapids.ai

cuDF Documentation: https://rapidsai.github.io/projects/cudf/en/latest/

cuML Documentation: https://rapidsai.github.io/projects/cuml/en/latest/

Github: https://github.com/RAPIDSai

Twitter: @rapidsai

Architecture

RMM Memory Pool Allocation

https://github.com/rapidsai/rmm

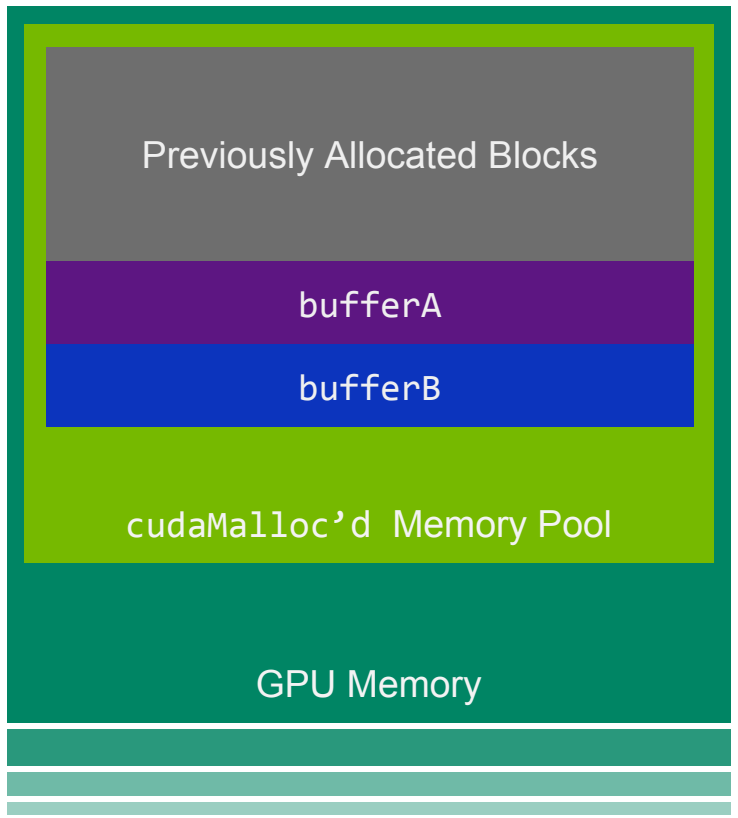
Use large `cudaMalloc` allocation as memory pool

Custom memory management in pool

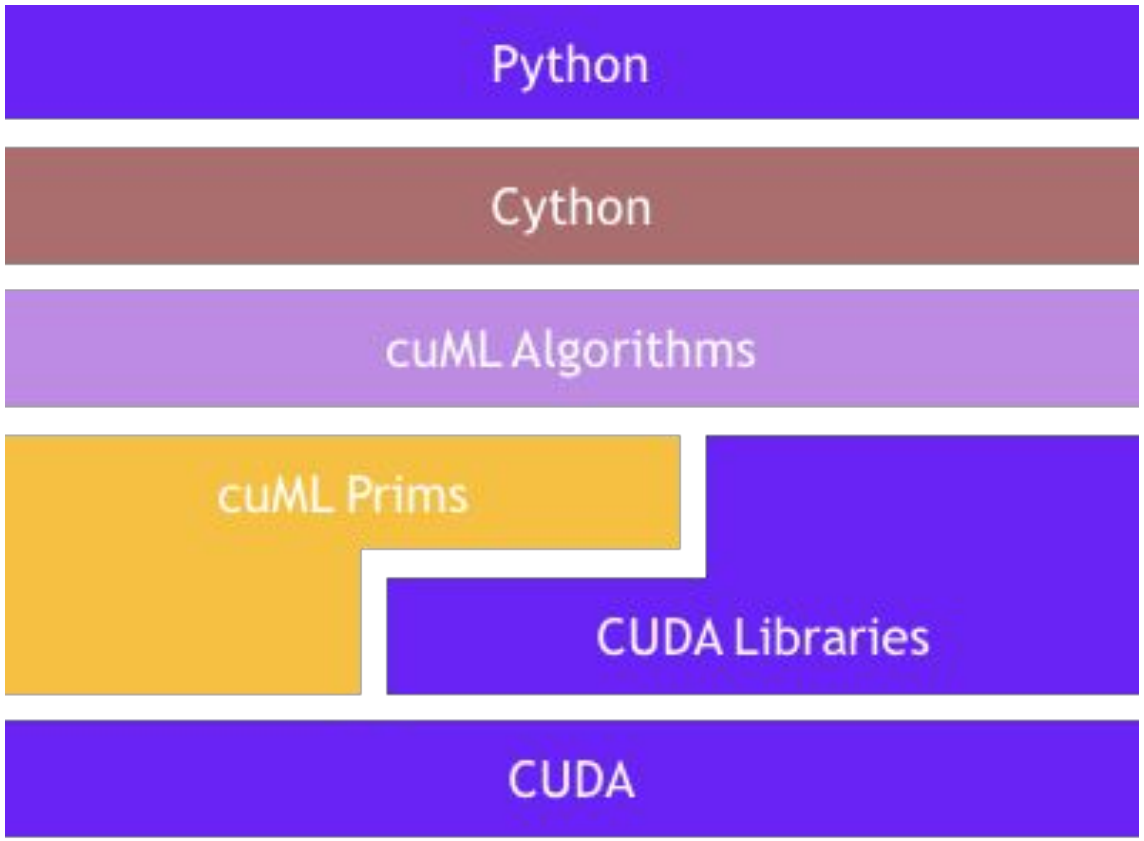
Streams enable asynchronous malloc/free

RMM currently uses CNMem as it's Sub-allocator
https://github.com/NVIDIA/cnmem

RMM is standalone and free to use in your own projects!



cuML architecture




Let’s Dive into the
Tutorial!

Getting GCP Set Up

Get GCP IP address

ssh pydata@{IP}
Password: gtc2019

conda activate rapids

Get the data: wget -v -O black_friday.zip -L https://goo.gl/3EYV8r (if you don't have wget, you can install it on mac via homebrew)

Download Jupyter Notebook wget -v -O gtc_tutorial_student.ipynb -L https://bit.ly/2Ht8hLe

jupyter-notebook --allow-root --ip=0.0.0.0 --port 8888 --no-browser --NotebookApp.token="

THANK YOU TO GOOGLE CLOUD PLATFORM

Kubeflow also has a RAPIDS container!

Google kindly donated the instances for this tutorial at GTC SJ 2019!

Paul Mahler
@realpaulmahler

NVIDIA