I've been asked a few times in RocketChat to go over how to leverage RapidsAI and Docker for both local or Compute development.

The best place to start is with Docker.

From Docker themselves:

Docker takes away repetitive, mundane configuration tasks and is used throughout the development lifecycle for fast, easy and portable application development - desktop and cloud. Docker's comprehensive end to end platform includes UIs, CLIs, APIs and security that are engineered to work together across the entire application delivery lifecycle.

What is RapidsAI?

RAPIDS

Open GPU Data Science | RAPIDS

A suite of software libraries for executing end-to-end data science completely on GPUs

The tl;dr is that RapidsAI is a pre-built set of GPU ready docker containers and libraries (including everyone's favorite XGBoost).

You can run the Rapids environment locally via conda as well, but when in rome…

What cool stuff do I get?

There are a bunch of cool GPU accelerated packages that come ready to rip, the one that first got me interested was Welcome to cuML's documentation! — cuml 0.18.0 documentation

cuML is insanely fast for most of the big computation that I've come across.

It does take some casting between numpy, pandas, cudf, and cupy but we are using Python, so casting is part of life right?

https://docs.rapids.ai/overview/latest.pdf

This is a really good overview of all the GPU accelerated features that cuML opens up.

Something else that is super snazzy is the pre-configured Dask Scheduler, even for my single GPU setup I can run a lot of stuff in parallel.

So what if I want to try it out and feel the speed for myself?

Read below:

Pre-Reqs:

- Linux
- NVidia GPU

First install Docker

Docker Documentation – 10 Mar 21

Install Docker Engine

Lists the installation methods

Then install the NVidia GPU CUDA Drivers

https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html

go through:

- Pre-Installation
- Package Manager Installation
- Driver Installation

Next download and run the basic RapidsAI Docker Container

RAPIDS

Getting Started | RAPIDS

Get started with RAPIDS using conda, docker, or from source builds.

```
docker pull rapidsai/rapidsai:0.18-cuda11.0-runtime-ubuntu20.04-py3.8 docker run --gpus all --rm -it -p 8888:8888 -p 8787:8787 -p 8786:8786 \ rapidsai/rapidsai:0.18-cuda11.0-runtime-ubuntu20.04-py3.8
```

- make sure to select the right CUDA version, OS version
- I use python 3.8 here for the walrus operator :=

Once the container boots, it will display a URL to open for the Jupyter labs instance

If you like the environment and want to make it a bit more reusable, here are a few steps to consider:

1. Make a shell script that runs the docker command
2. Get rid of the --rm
3. Add some volumes for your personal files/notebooks
4. Persist the Jupyter settings
5. More?

This is an older example of my start-rapids.sh script which shows off a bunch of the optional flags:

!/bin/sh

```
sudo docker run \ --gpus all \ -it \ -p 8888:8888 \ -p 8787:8787 \ -p 8786:8786 \ -e EXTRA_APT_PACKAGES="build-essential" \ -e EXTRA_PIP_PACKAGES="numerapi numpy" \ -e EXTRA_CONDA_PACKAGES="joblib scikit-learn torch" \ -v $(pwd):/rapids/workspace \ rapidsai/rapidsai:cuda11.0-runtime-ubuntu20.04-py3.8
```

Happy to talk more about local / cloud development environments with people. I’ve been working with AWS for many years professionally, and love helping myself/others optimize their workflows through tooling