RAPIDS

22.06 Release

@RAPIDSai

https://github.com/rapidsai

https://rapids-goai.slack.com/join

RAPIDS

https://rapids.ai

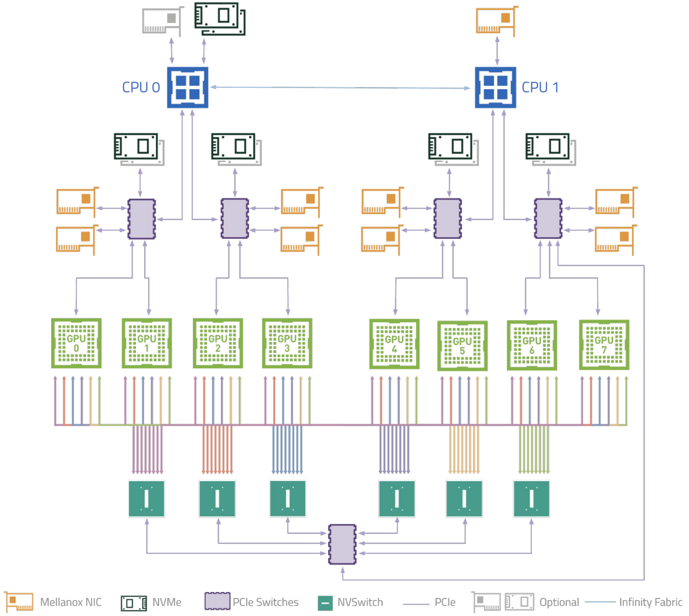
Why GPUs for Data Science?

Numerous hardware advantages

- Thousands of cores with up to ~20 TeraFlops of general purpose compute performance
- Up to 1.5 TB/s of memory bandwidth
- Hardware interconnects for up to 600 GB/s bidirectional GPU <--> GPU bandwidth
- Can scale up to 16x GPUs in a single node

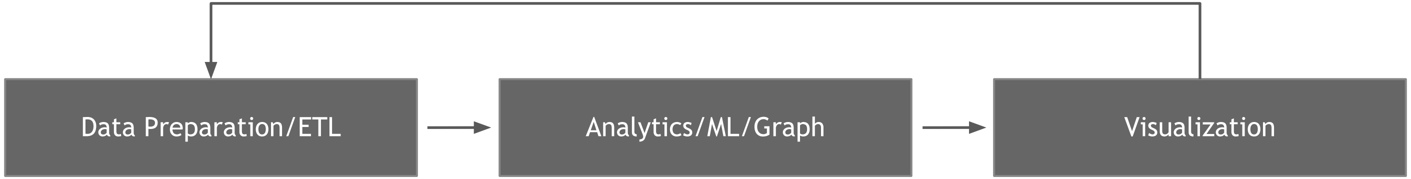
Almost never run out of compute relative to memory bandwidth!

NVIDIA DGX A100 System



What is RAPIDS?

End-to-End GPU Accelerated Data Science



Data Preparation/ETL

Analytics/ML/Graph

Visualization

cuDF

- ➢ GPU-accelerated ETL functions
- ➢ Tracks Pandas and other common PyData APIs
- ➢ Dask + UCX integration for scaling

RAPIDS ML

- ➢ GPU-native cuML library, plus XGBoost, FIL, HPO, and more

cuGraph

- ➢ GPU graph analytics, including TSP, PageRank, and more

cuxfilter

- ➢ GPU-accelerated cross-filtering

pyViz integration

- ➢ Plotly Dash, Bokeh, Datashader, HoloViews, hvPlot

Domain-Specific Libraries

CLX + Morpheus
Cyber log processing + anomaly detection

cuStreamz
Streaming analytics

cuSignal
Signals processing

cuCIM
Computer vision & image processing primitives

cuSpatial
Spatial analytics

node-RAPIDS
Bindings for node.js

...and more!

Major updates from the RAPIDS 22.06 release

- The release of a new RAPIDS Graph-as-a-Service library GaaS uses cuGraph, cuDF, and other libraries on the server to execute graph data prep and analysis on server-side GPUs. Multiple clients can connect to the server allowing different users and processes the ability to access large graph data that may not otherwise be possible

- The addition of Multi-Node Multi-GPU (MNMG) algorithms across many libraries, including cuML and cuGraph When working with massive amounts of data, it’s not uncommon to run out of room on a single GPU. We continue to expand support for Multi-Node Multi-GPU (MNMG) workloads across RAPIDS libraries

- Support for interactive debugging and profiling of memory usage with RMM Memory allocations in RAPIDS are performed by the RAPIDS Memory Manager (RMM). In the 22.06 release of RMM, users can now trigger a callback function when RAPIDS libraries allocate or free memory for interactive debugging, profiling, and a wide range of other use cases

Overview of Changes: RAPIDS 22.06 Release

- **cuDF** Addition of df.eval() for faster and more memory efficient calculations, df.applymap() for element-wise functions
- **cuML** Initial support for multi-node Logistic Regression, based on Dask-GLM, and Dask’s CuPy array support
- **cuGraph** Property Graph available on MNMG architectures; eigenvector centrality algorithm, and Triangle counting algorithm on data located across multiple GPUs
- **Dask-CUDA** Users can now trigger a callback function when RAPIDS libraries allocate or free memory for interactive debugging, profiling, and a wide range of other use cases

cuDF Updates: Deep Dive

Release 22.06

Features added in 22.06

- Decompression by “Zstandard” released for Parquet and ORC (experimental)
- Merging/Joining is 10-30% faster, due to performance optimization
- Documentation updates to user guide, new contributors, library design, C++ doxygen
- Upgrade cudf to support pandas 1.4.x versions
- Add a section to the docs that compares cuDF with Pandas
- Add cudf.DataFrame.applymap
- Implement DataFrame.eval using libcudf ASTs

Planned Upcoming Features

- Support

cuML Updates: Deep Dive

Release 22.06

Features added in 22.06

- First version of Multi-Node Multi-GPU Logistic Regression
- LabelEncoder functionality for CuPy and NumPy arrays
- Random Ball Cover support for 3D data in Nearest Neighbors
- Some other minor tidbits: sample_weight support for Ridge Regression, simplicial_set functions in UMAP
- Double precision support in FIL

Planned upcoming features

cuGraph Updates: Deep Dive

Release 22.06

Features added in 22.06

- Released the initial version of Graph-as-a-Service
- Initial version of a Multi-GPU Property Graph
- Initial release of Python wrappers around cugraph-ops aggregators (pylibcugraphops)
- New Degree Centrality algorithm
- New Multi-GPU Eigenvector Centrality algorithm
- New Multi-GPU Triangle Counting algorithm

Planned Upcoming Features

- Data Masking
- Integration with DGL

Join the Conversation

GOOGLE GROUPS

https://groups.google.com/forum/#!forum/rapidsai

DOCKER HUB

https://hub.docker.com/r/rapidsai/rapidsai

SLACK CHANNEL

https://rapids-goai.slack.com/join

STACK OVERFLOW

https://stackoverflow.com/tags/rapids

THANK YOU

@RAPIDSai

RAPIDS