21.08 Release

@RAPIDSai
https://github.com/rapidsai
https://rapids-goal.slack.com/join
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!
What is RAPIDS?
End-to-End GPU Accelerated Data Science

Data Preparation/ETL

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

Analytics/ML/Graph

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

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

Visualization

- 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!
Overview of Changes: RAPIDS 21.08 Release

- **cuDF** Decimal support for CSV reader, functionality to convert dataframe to struct series; fillna added to groupby; supports multiple inputs in JSON and ORC reader; list read and write support for ORC; experimental read support for structs in ORC; experimental support for null in UDFs for Python;

- **cuML** New Bernoulli Naive Bayes model; improvements to HDBSCAN, ARIMA, FIL and Random Forest; new weighted multi-node multi-gpu KMeans algorithm; new distances added to pairwise_distances;

- **cuGraph** Doubly Compressed Sparse Row and Doubly Compressed Sparse Column support added to libcugraph; graph batching for C++; epsilon parameter added to Hungarian algorithm; continued improving graph primitives for performance; depth limit functionality on traversal algorithms; Enhanced multi-gpu scaling

- **CLX** Maintenance to existing code; working through including CLX use cases in Morpheus

- **cuCIM** Add functionality to perform morphological thinning of a binary image
cuDF Updates: Deep Dive
Release 21.08

Features added in 21.08
- Decimal data type is now supported for csv reader function in Python
- List read and write, and experimental read support for structs in ORC
  - `Read_json` supports reading multiple input files/buffers
  - `Fillna` feature added to groupby
- Experimental support for handling null in UDFs for Python
- cuDF Dataframe has new functionalities for structs: `to_struct` and `explode` methods

Planned Upcoming Features
- Expanded support for additional decimal types
- Enhanced ORC struct and map support
- Conditional equijoins support in libcudf
cuML Updates: Deep Dive
Release 21.08

Features added in 21.08

- Single-GPU implementation of Bernoulli Naive Bayes algorithm
- Added support for chebyshev, canberra, hellinger and minkowski distances for pairwise distance calculations
- Vector leaf prediction and significant improvements and optimizations to the Forest Inference Library (FIL)
- GTIL (General Tree Inference Library) for CPU inference was introduced to the FIL backend for Triton
- Multiple improvements to the new Random Forest backend to optimize memory and performance.
- Add weighted KMeans sampling for KernelSHAP
- Support for weighted sampling in the multi-node multi-gpu KMeans algorithm
- Many more model-specific improvements and bug fixes: ARIMA memory improvements, dtype conversion optimization for FIL, multiple HDBSCAN improvements ...

Planned Upcoming Features

- Categorical features support in FIL
- Support for missing observations, padding and exogenous variables for ARIMA
- Single-node single-GPU implementation of Gaussian Naive Bayes
cuGraph Updates: Deep Dive
Release 21.08

Features added in 21.08
- Doubly Compressed Sparse Row and Doubly Compressed Sparse Column support added to libcugraph
- Epsilon parameter is now supported in the Hungarian algorithm
- Random Walk updated to improve performance
- Resolve tech debt and enhance the library
- Depth limit functionality on traversal algorithms
- Addressed issues with multi-gpu scaling

Planned Upcoming Features
- Biased Random Walk
- Multi-Seed Breadth First Search
- Multi-GPU Triangle Counting
- Multi-GPU HITS
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