RAPIDS

21.12 Release

@RAPIDSai

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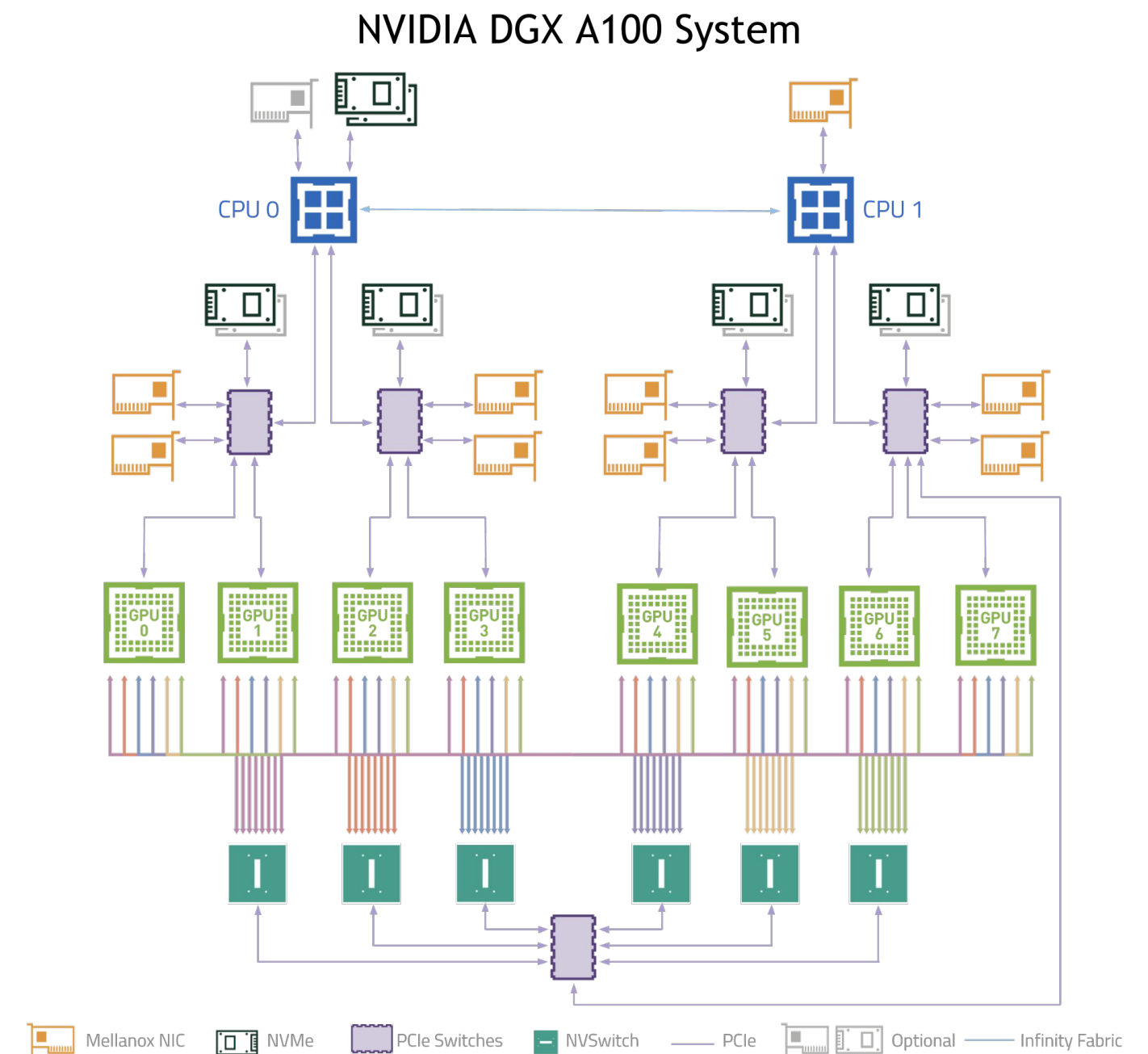
https://rapids-goai.slack.com/join

https://rapids.ai
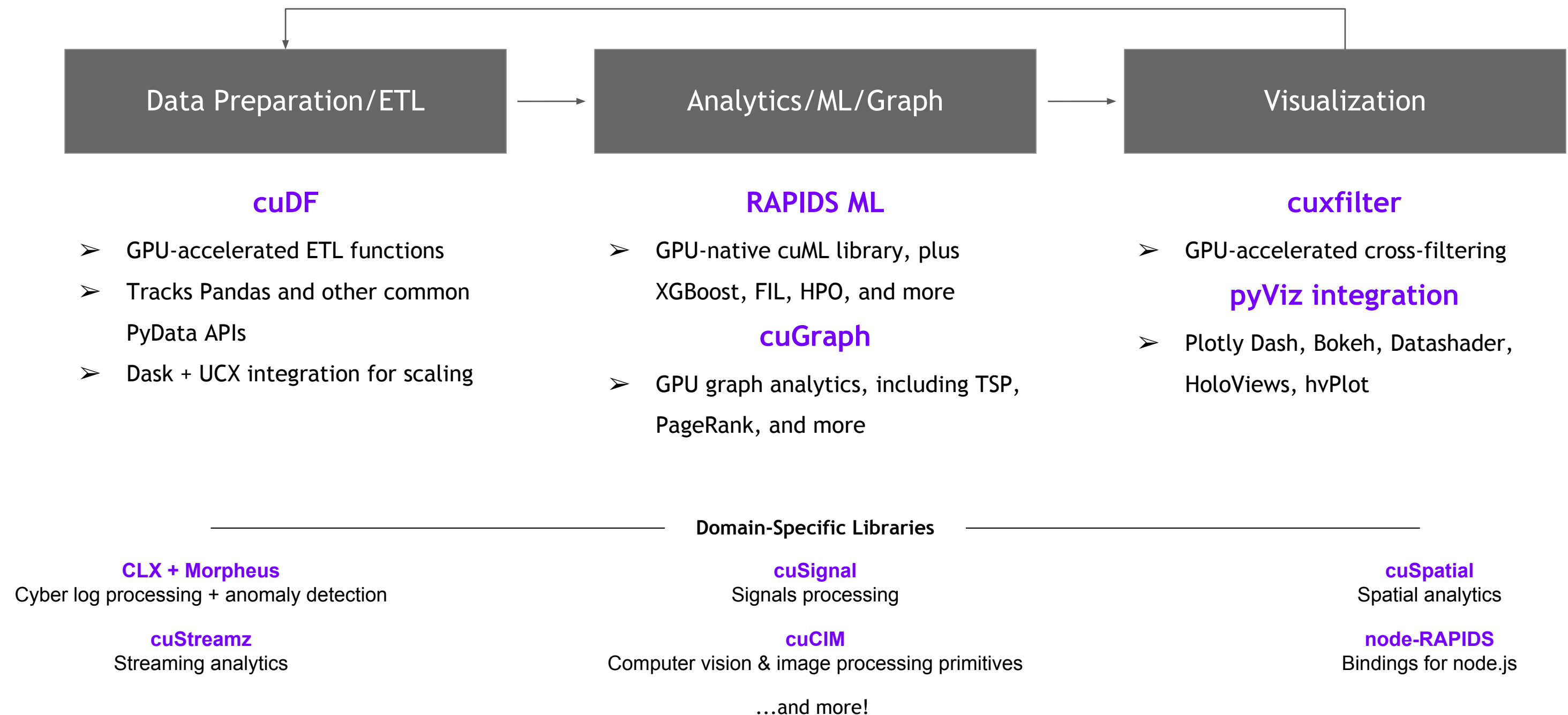
Why GPUs for Data Science?
Numerous hardware advantages

- 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!

NVIDIA DGX A100 System

Mellanox NIC   NVMe   PCIe Switches   NVSwitch   ——— PCIe   Optional   ——— Infinity Fabric

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

…and more!

cuSpatial
Spatial analytics

node-RAPIDS
Bindings for node.js

Overview of Changes: RAPIDS 21.12 Release

▸ **RAPIDS** CUDA enhanced compatibility is now supported by RAPIDS–users can now use newer CUDA binaries or runtimes without updating their CUDA driver version (450.80.02+).

▸ **RAPIDS+Dask** Support for spilling on demand; Deprecation of support for UCX version less than 1.11.1

▸ **cuDF** 128 bit Decimal support; Groupby diff; Map support for ORC writer; String now has support for istitle; mixed numeric type support in to_dlpack; Resample function added to cuDF; pandas-like UDF interface with DataFrame and Series.apply;

▸ **cuML** New LinearSVC and LinearSVR models; Dask-SQL support for cuML models; exogenous variable support in ARIMA; KL divergence exposed in TSNE;

▸ **cuGraph** K-core for undirected graphs for single and multiple GPUs; graph primitive filter for edges; transpose_edgelist and symmetrize_edgelist and count number of self loops and multi-edges in a graph in libcugraph; Improved performance for NetworkX inputs; Initial implementation of Multi GPU HITS;

▸ **Dask-SQL** Experimental GPU support for Dask-SQL, bringing the power of RAPIDS to SQL

cuDF Updates: Deep Dive
Release 21.12

Features added in 21.12

▸ 128 bit Decimal support in libcuDF
▸ *Diff* is now supported for groupby
▸ Map support for ORC writer
▸ Support for Series.str.*istitle*
▸ pandas-like UDF interface with DataFrame and Series.apply
▸ Handling of mixed numeric types supported in *to_dlpack*
▸ *Resample* function added to cuDF will allow users to resample time series data
▸ cuDF's *Grouper* function now supports grouping via time frequency

Planned Upcoming Features

▸ Expanded support for 128 bit decimal types
▸ Nested type support for JSON reader
▸ Decimal support for Dask cuDF Parquet reader
▸ Refactored hash join implementation

cuML Updates: Deep Dive
Release 21.12

Features added in 21.12

▸ Support for LinearSVM using QN solvers, *LinearSVC* and *LinearSVR* models

▸ Ability to use Dask-SQL with cuML models

▸ *ARIMA* now supports exogenous variables

▸ KL divergence exposed in *TSNE*

Planned Upcoming Features

▸ Multiple improvements to T-Distributed Stochastic Neighbor Embedding algorithm to improve accuracy

▸ Enhancements to Random Forest Algorithm

▸ Symbolic classification and regression models
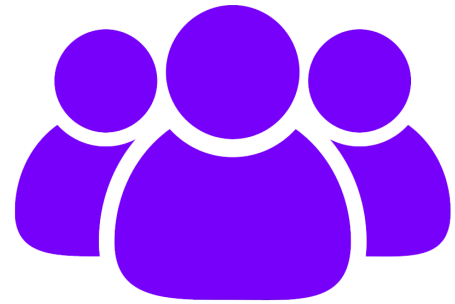
cuGraph Updates: Deep Dive
Release 21.12

Features added in 21.12

- Implemented K-core for undirected graphs for Single and Multi GPUs
- Initial implementation of Multi GPU HITS
- Initial version of node2vec in C++/CUDA for graph sampling, Python wrapper coming soon
- New graph filter for the edges, extract_if_e, in libcugraph
- Implemented transpose_edgelist and symmetrize_edgelist in libcugraph
- Added the ability to count number of self loops and multi-edges in a graph, in libcugraph
- Improved performance when a NetworkX graph is passed in

Planned Upcoming Features
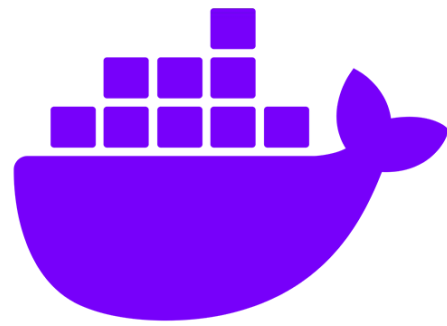
- *Multi-GPU graph primitives for Triangle Counting*
- *More algorithms poerted to using primitives and scaling to Multi GPUs*
- *Ability for user to write custom algorithms using the graph primitives*
- *Multi GPU neighborhood sampling*
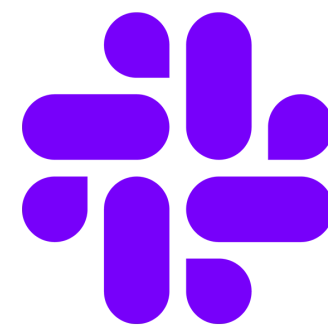- *Property Graphs*

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