GPU-accelerated graph analytics in LynxKite

Daniel Darabos

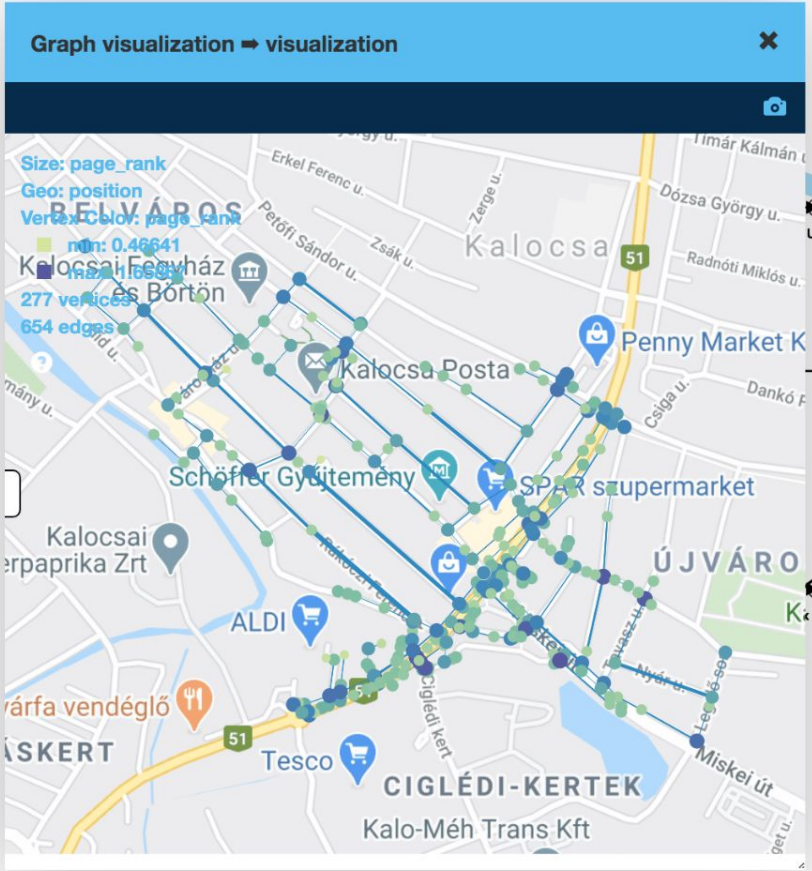
lynx kite

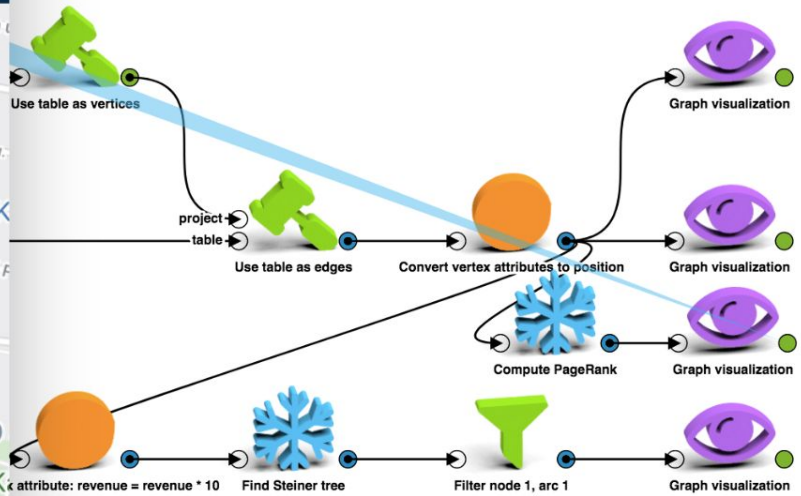
GPU-accelerated graph analytics in LynxKite

GPU-accelerated algorithms are the latest addition to the LynxKite open-source graph data science platform. With the hardware and software improvements of the last few years, GPUs have become a viable and super fast alternative to classical big data tools. In this talk I will introduce NVIDIA's RAPIDS cuGraph library, explain how we have made it a seamless part of LynxKite workflows, and show a few cool applications this has enabled.

What is LynxKite?

Fiber network optimization





What is LynxKite?

An open-source data science tool for analyzing graphs. (E.g. social networks.)

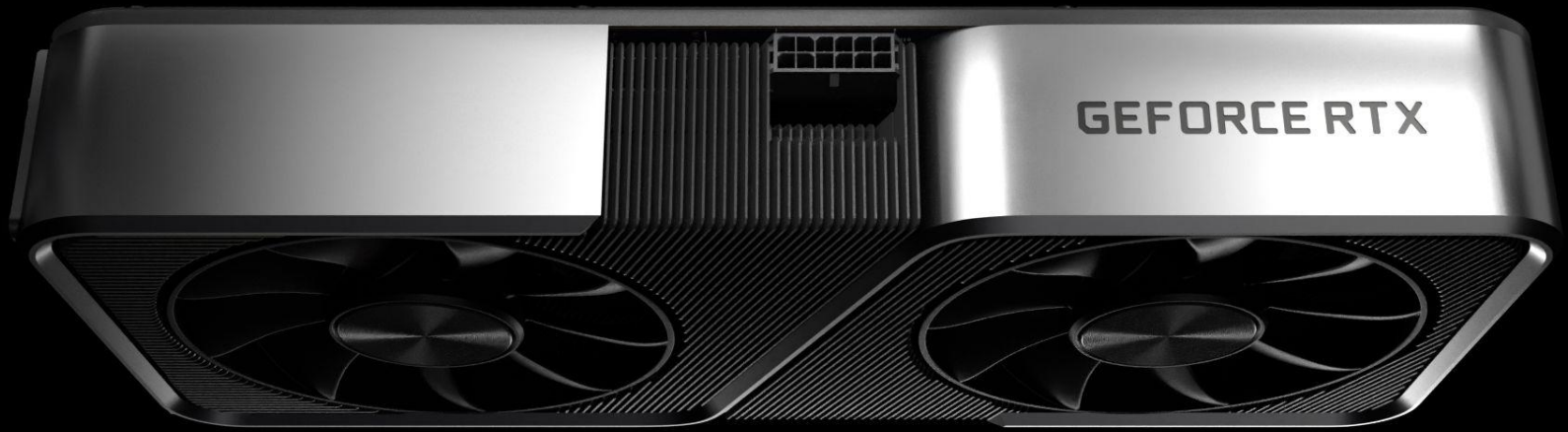
LynxKite has a Python API and a rich web UI.

LynxKite has 200+ operations from the mundane (import CSV, merge parallel edges) to the sophisticated (find optimal Steiner tree, train graph neural network).

Most of these are our own implementations, but some come from open-source libraries.

We have an enterprise version that is used by telecom companies, banks, retailers, etc.

What is a GPU?


GEFORCE RTX

What is a GPU?

Like a CPU, but fast.

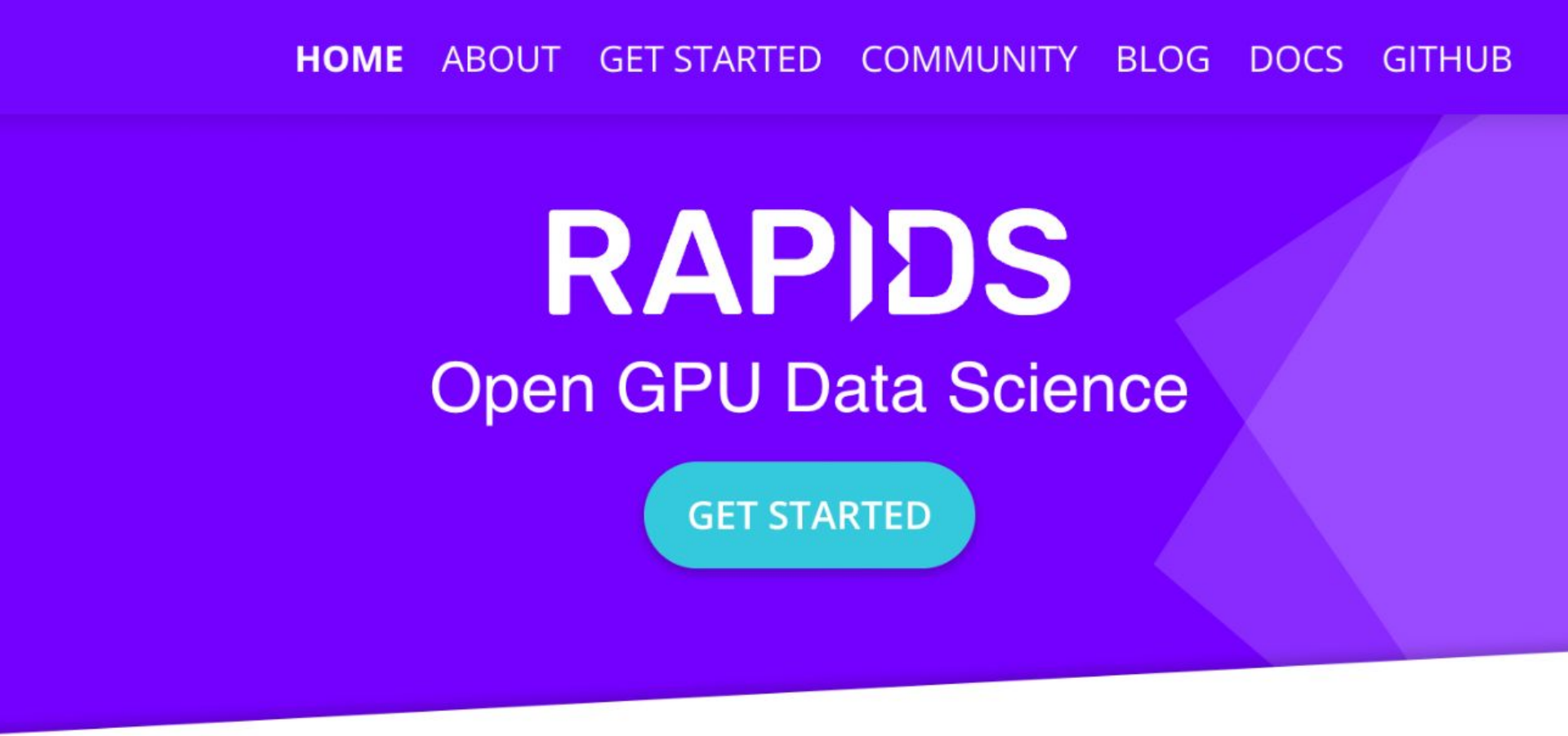
What is a GPU?

Like a CPU, but fast.

But cannot run Python code.

Only manufactured by NVIDIA.*

What is NVIDIA RAPIDS?


HOME ABOUT GET STARTED COMMUNITY BLOG DOCS GITHUB
RAPIDS
Open GPU Data Science
GET STARTED

What is NVIDIA RAPIDS?

Pandas that runs on GPUs.

Also Numpy, scikit-learn, SciPy Signal, crossfilter, scikit-image, NetworkX.

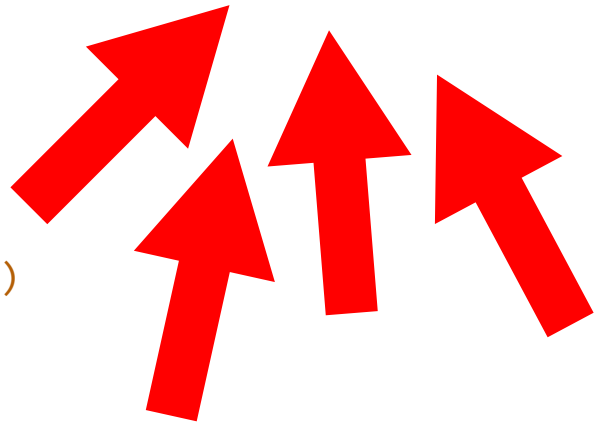
CuPy cuML cuSignal cuxfilter cuCIM cuGraph

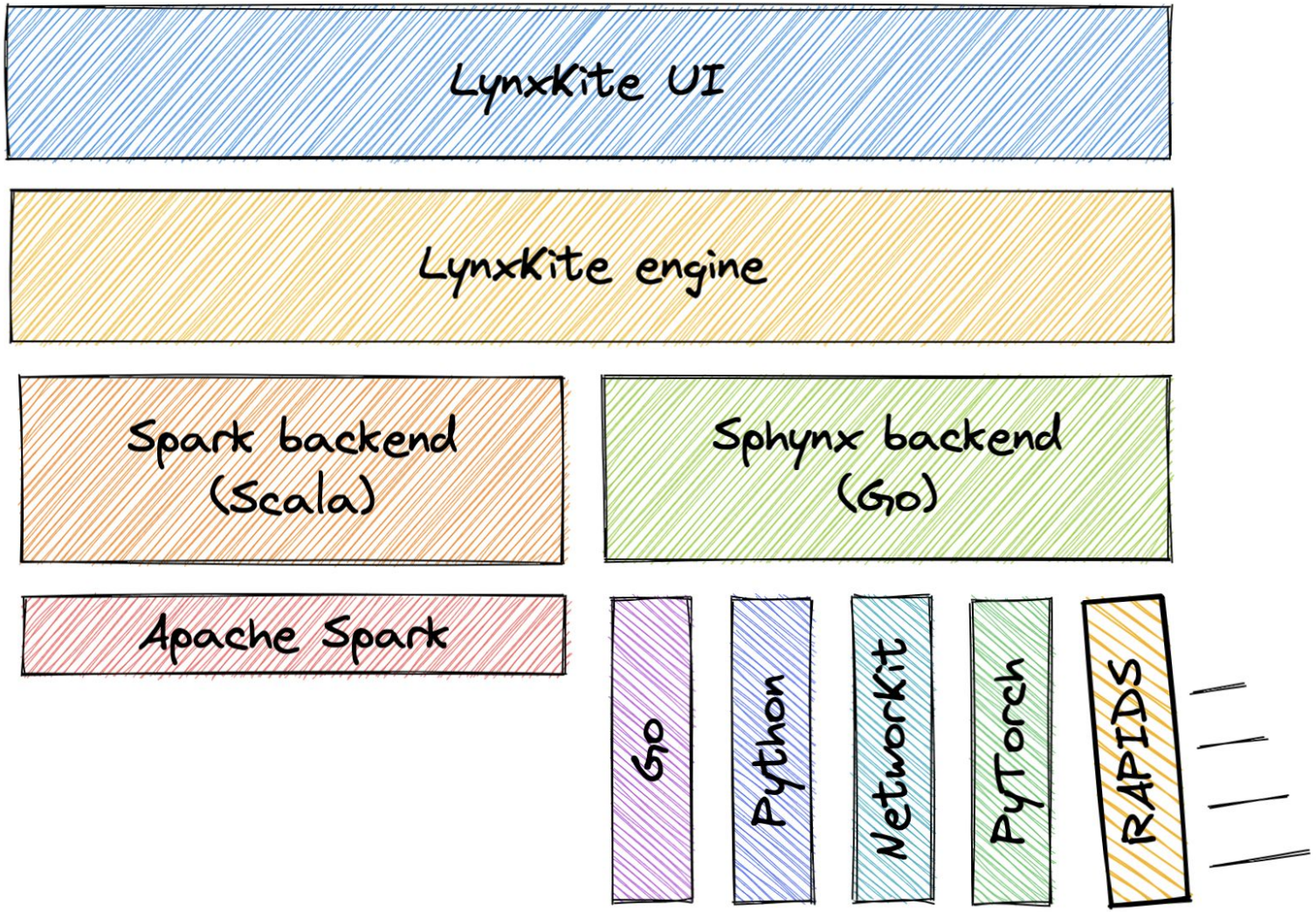
What is NVIDIA RAPIDS?

Pandas that runs on GPUs.

Also Numpy, scikit-learn, SciPy Signal, crossfilter, scikit-image, **NetworkX**.

```
import cudf
import cugraph
gdf = cudf.read_csv(
  'edges.csv',
  dtype=['int32', 'int32'], names=['src_id', 'dst_id'])
G = cugraph.Graph()
G.from_cudf_edgelist(
  gdf, source='src_id', destination='dst_id')
bc = cugraph.betweenness_centrality(G)
bc.to_csv(bc.csv', index=False)
```




LynxKite UI
LynxKite engine
Spark backend
(Scala)
Sphynx backend
(Go)
Apache Spark
Go
Python
NetworKit
PyTorch
RAPIDS

LynxKite 5.0

Released this week.

PageRank, connected components, betweenness and Katz centrality, the Louvain method, k-core decomposition, and a 2D layout algorithm are GPU-accelerated.

New faster storage layer for graph stuff.

New Python API to allow mixing LynxKite and raw PySpark stuff.

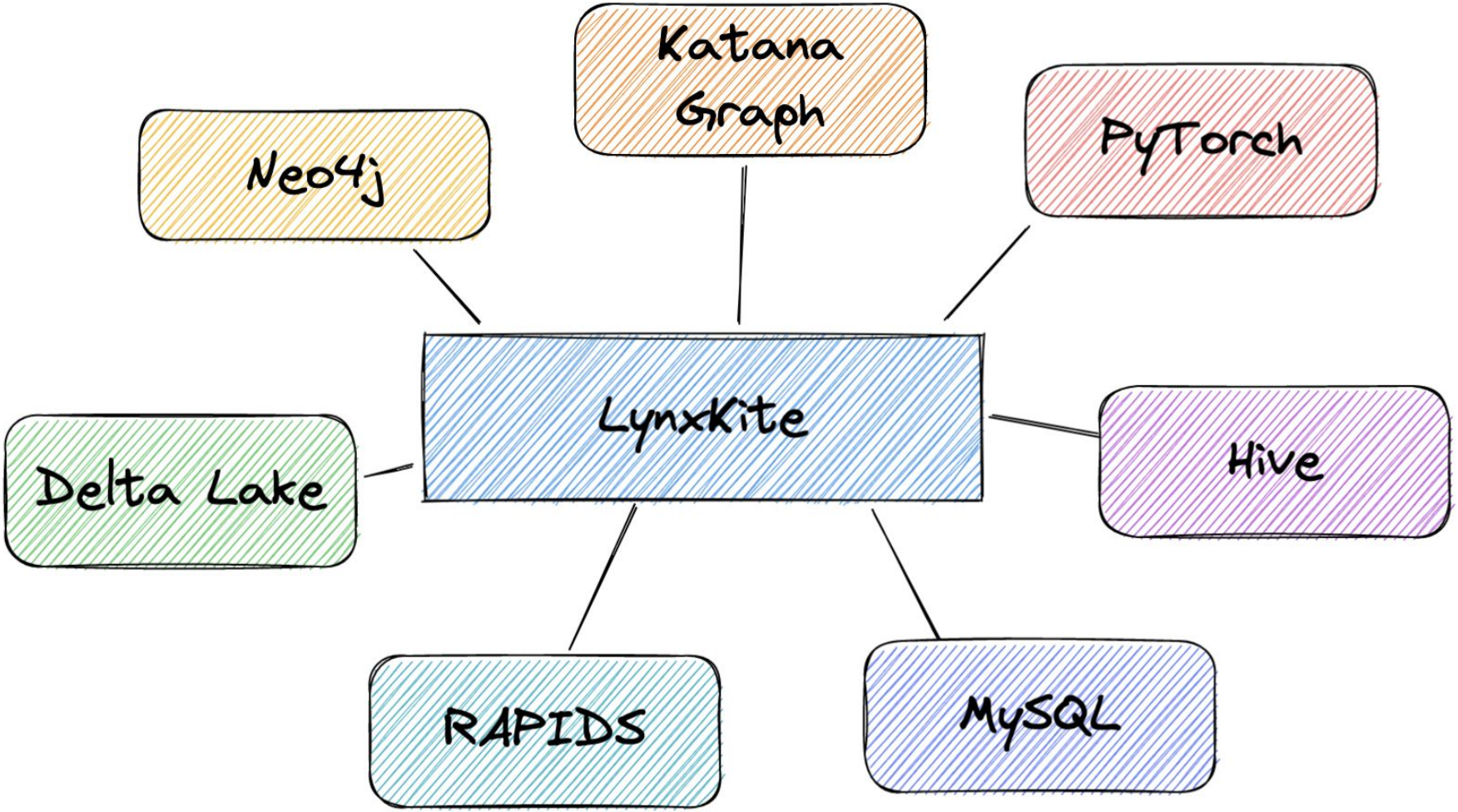
Backend preference is now configurable.

Getting started

```
docker run \
  -p 2200:2200 \
  -e KITE_ENABLE_CUDA=yes \
  --gpus all \
  --name lynxkite lynxkite/lynxkite:latest-cuda
```

How cuGraph works: In CUDA We Thrust

```
auto val_first = thrust::make_zip_iterator(
  thrust::make_tuple(*personalization_vertices, *personalization_values));
thrust::for_each(
  handle.get_thrust_policy(),
  val_first,
  val_first + *personalization_vector_size,
  [vertex_partition, pageranks, dangling_sum, personalization_sum, alpha] __device__(
    auto val) {
    auto v     = thrust::get<0>(val);
    auto value = thrust::get<1>(val);
    *(pageranks + vertex_partition.local_vertex_partition_offset_from_vertex_nocheck(v)) +=
      (dangling_sum * alpha + static_cast<result_t>(1.0 - alpha)) *
      (value / personalization_sum);
  });
```


Katana Graph
Neo4j
PyTorch
LynxKite
Delta Lake
Hive
RAPIDS
MySQL

LynxKite roadmap

Databricks integration.

Import from Gremlin sources.

Import and export for Google BigQuery.

Import complex graphs from data catalogs.

Streaming processing.

Visualization UI revamp.

cuGraph roadmap

All 32 algorithms to support MNMG.

More algorithms.

Better property graph support.

Better GNN support.

Better scaling for 1000+ GPUs.


lynxkite
WE'RE HIRING


lynxkite
WE'RE HIRING

twitter.com/DanielDarabos lynxkite.com rapids.ai