GPU Accelerated Property Graph in Python

What is a Property Graph?

- <picture>
- Unlike a traditional graph which strictly models some relationship between vertices using edges, a property graph allows for arbitrary data (usually a set of key:value pairs) to be associated with each vertex and edge.
- Property Graphs are very useful for modeling real-world data that has complex relationships or interdependencies.
- <graphic showing a PG vs. a table>?

Use Cases

- GNNs
- Cyber
- Visualization
  - Standard graph viz use cases (maps, social networks, circuit layout, etc.) that require additional information about each element for rendering (color, size, shape, etc.)

GPU Acceleration

- Queries/subgraph extraction (if supported) can be faster
- GPU-based algorithms can use edge props for weights w/o data xfer
- Future GPU-based algos may use additional properties (besides edge weights)
- Keep some or all data on GPU for other parts of a larger workflow
  - GNNs

Challenges with GPU Acceleration

- Memory size
- Host/device data xfer
- <mention advantages of multi-GPU with RAPIDS>

Comparison to Graph Databases

- Most graph databases are not GPU-accelerated, and often optimized for queries and not analytics
- Training, larger analytic workflows at scale result in lots of query overhead