Massively Parallel Graph Databases the future of Big Data Analytics?

by Pavel Velikhov
Director Engineering, TigerGraph

Main points of the talk

- Quick intro into Graph DMBSs
- Beating MPP RDBMSs at their own game
- Additional analytic capabilities of Graph MPP DBMSs
- Parallel Graph Algorithms on Big Data
- In-database machine learning (especially graph features and GNN)
- Conclusion

Quick breakdown of GraphDBs

There are lots of them!

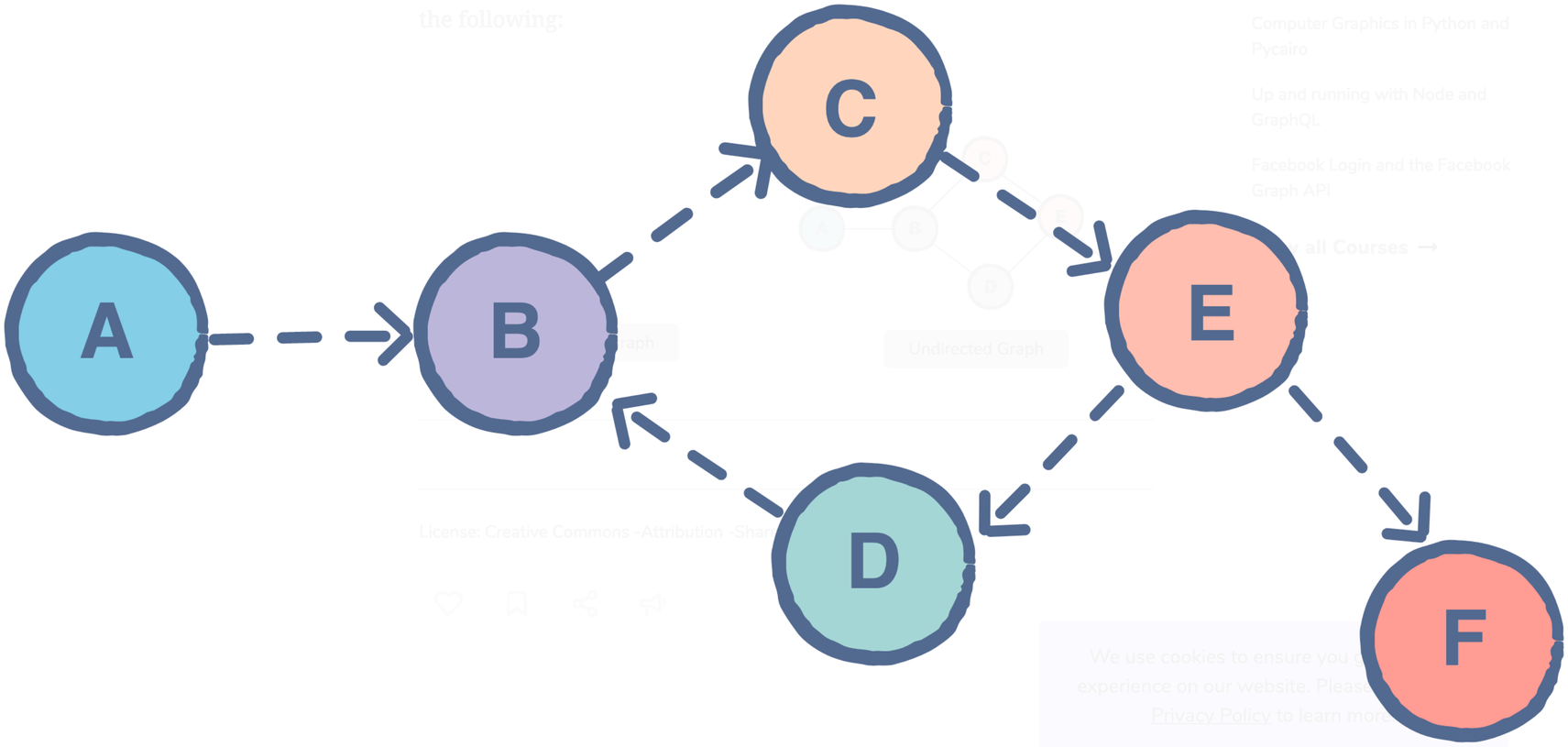
Rank			DBMS	Database Model	Score		
Jan 2022	Dec 2021	Jan 2021			Jan 2022	Dec 2021	Jan 2021
1.	1.	1.	Neo4j	Graph	58.03	0.00	+4.25
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	40.04	+0.33	+7.07
3.	3.	↑ 7.	Virtuoso	Multi-model	5.37	+0.31	+3.23
4.	4.	4.	ArangoDB	Multi-model	4.73	-0.02	-0.56
5.	5.	↓ 3.	OrientDB	Multi-model	4.56	+0.16	-0.77
6.	6.	↑ 8.	GraphDB	Multi-model	2.86	-0.02	+0.75
7.	7.	↓ 6.	Amazon Neptune	Multi-model	2.63	+0.07	+0.32
8.	8.	↓ 5.	JanusGraph	Graph	2.39	-0.02	-0.19
9.	9.	↑ 12.	TigerGraph	Graph	2.02	+0.02	+0.62
10.	10.	↑ 11.	Stardog	Multi-model	1.89	-0.04	+0.42
11.	11.	↓ 10.	Dgraph	Graph	1.51	-0.07	-0.05
12.	12.	↓ 9.	Fauna	Multi-model	1.36	-0.04	-0.55
13.	13.	↑ 14.	Giraph	Graph	1.31	-0.03	+0.18
14.	14.	↓ 13.	AllegroGraph	Multi-model	1.24	+0.02	+0.05
15.	15.	15.	Nebula Graph	Graph	1.14	0.00	+0.22
16.	16.	16.	Blazegraph	Multi-model	0.96	+0.00	+0.09
17.	17.	17.	Graph Engine	Multi-model	0.85	+0.01	+0.15
18.	18.	18.	TypeDB	Multi-model	0.76	-0.02	+0.07
19.	19.	19.	InfiniteGraph	Graph	0.47	+0.01	-0.04
20.	20.	↑ 29.	Memgraph	Graph	0.37	+0.00	+0.30
21.	21.	↑ 25.	AnzoGraph DB	Multi-model	0.33	-0.01	+0.17
22.	22.	↓ 21.	FlockDB	Graph	0.30	+0.00	-0.02
23.	↑ 25.	↓ 22.	HyperGraphDB	Graph	0.24	+0.03	-0.03
24.	24.	↑ 27.	TerminusDB	Graph, Multi-model	0.22	-0.02	+0.10
25.	↓ 23.	↓ 20.	Fluree	Graph	0.22	-0.02	-0.11
26.	↑ 28.	↑ 30.	HugeGraph	Graph	0.14	+0.02	+0.07
27.	↑ 30.	↓ 23.	GraphBase	Graph	0.13	+0.05	-0.04

Why we only consider 2 DBs

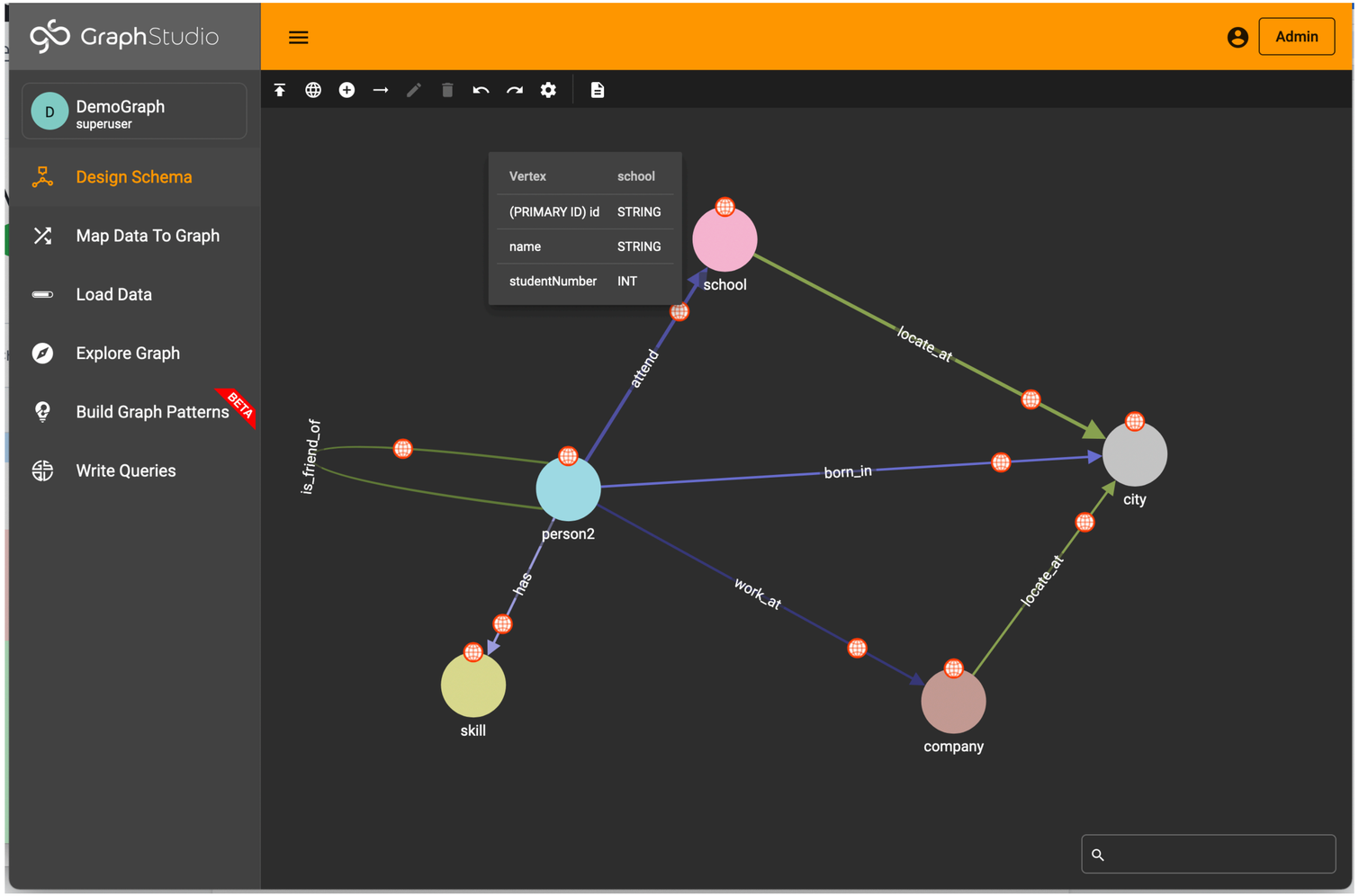
TigerGraph and Nebula

- MPP - e.g. Neo4j is really nice, but has almost zero capabilities for MPP
- Powerful query language - leading to GQL
  - JanusGraph and DGraph are MPP, but don't have a powerful QL
- Enterprise-ready
  - Lots of start-ups right now, but few mature products

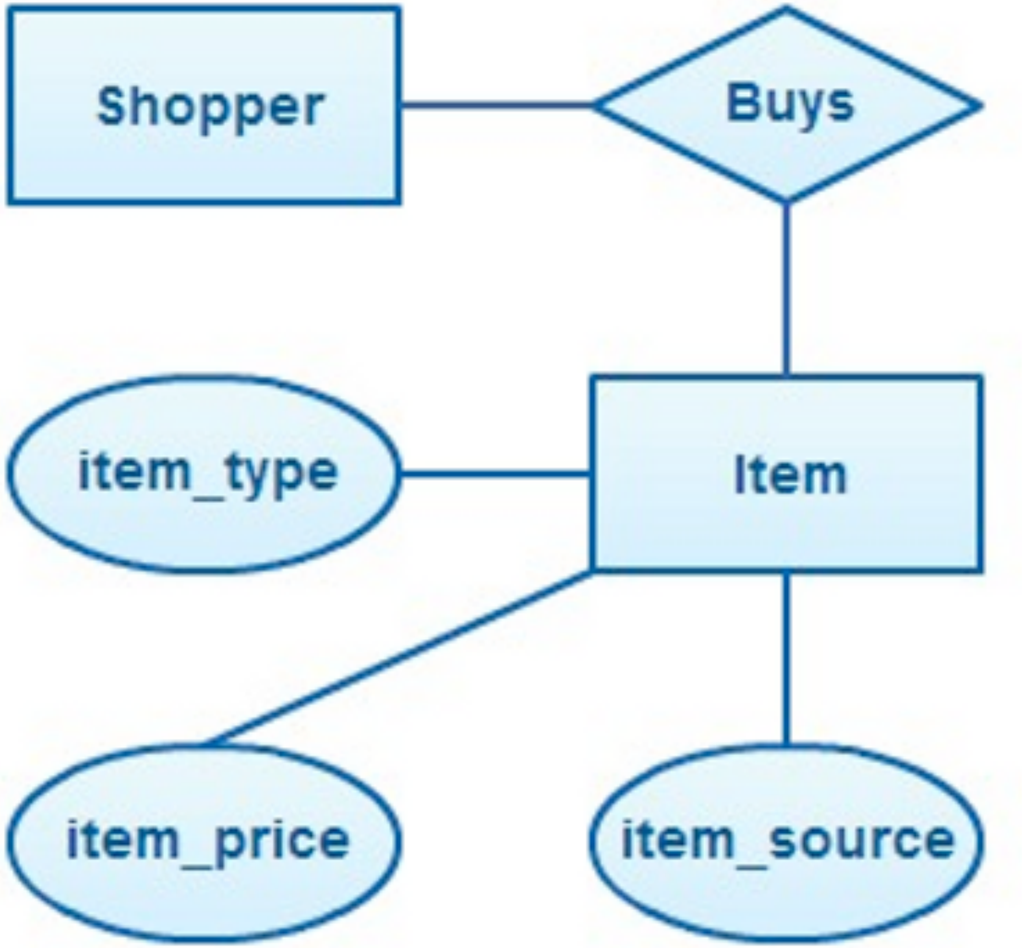
Graph Data



Example of Graph Schema



Relational data and Graphs

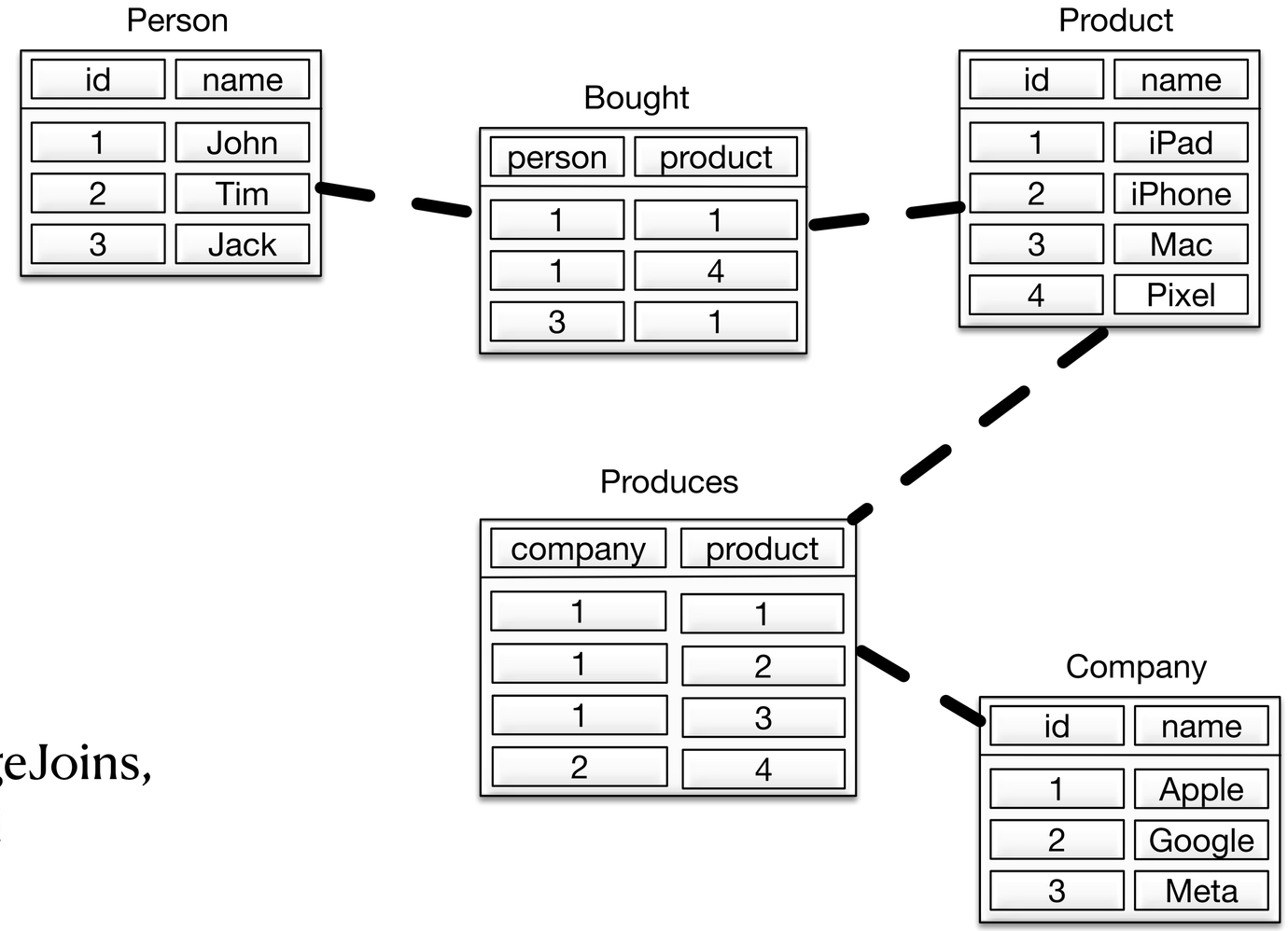


How we handle this data in RDMSs

- We split the graph into tables
  - Vertices and edges each get a separate table
- Typically in OLAP queries over Star or Snowflake schema we reconstruct the graph via join operations
- Why is this a problem:
  - Joins are expensive
  - 10-15 joins is max what we can handle

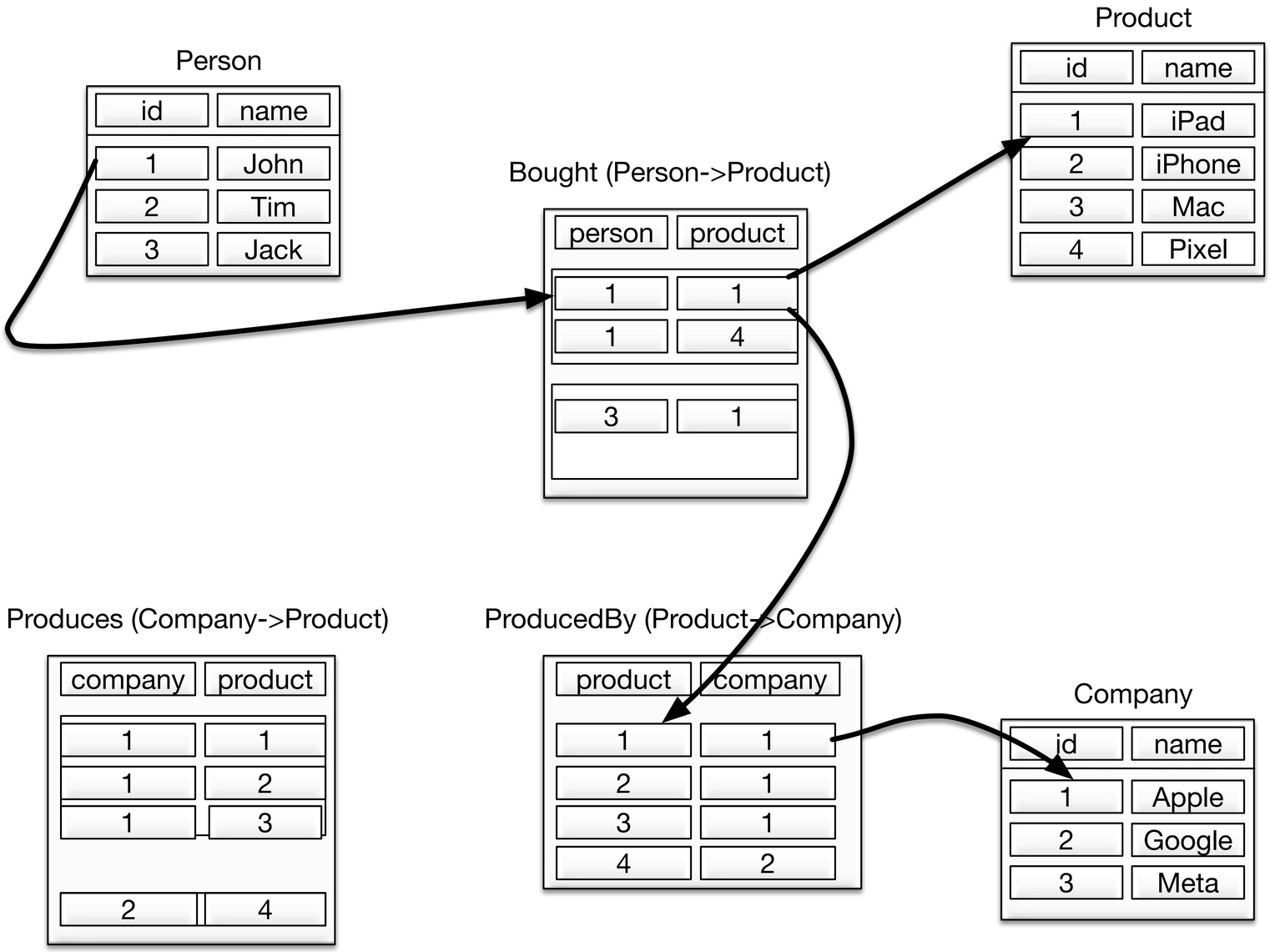
What's wrong with Joins?

Can't we fix it?



No Join strategy efficiently
solves this problem:
i.e. you can try HashJoins, MergeJoins,
the result is still very expensive !

How MPP Graph DBMSs beat relational OLAP at their own game



But...

Its not just joins, its subqueries

- You could have a subquery with group-by and aggregates
- Its quite common
- GraphDBs separate aggregation and grouping from vertex and edge processing (Accumulators in TigerGraph)

How does it look like in TigerGraph

A crash course in GSQL

```
SumAccum<FLOAT> @cSales, @pSales;


SELECT c
FROM   Customer:c - (Bought->:b) - Product:p
ACCUM  float rev = b.quant*(1-b.disc)*p.price,
       c.@cSales += rev,
       p.@pSales += rev
```

What about n-way relations

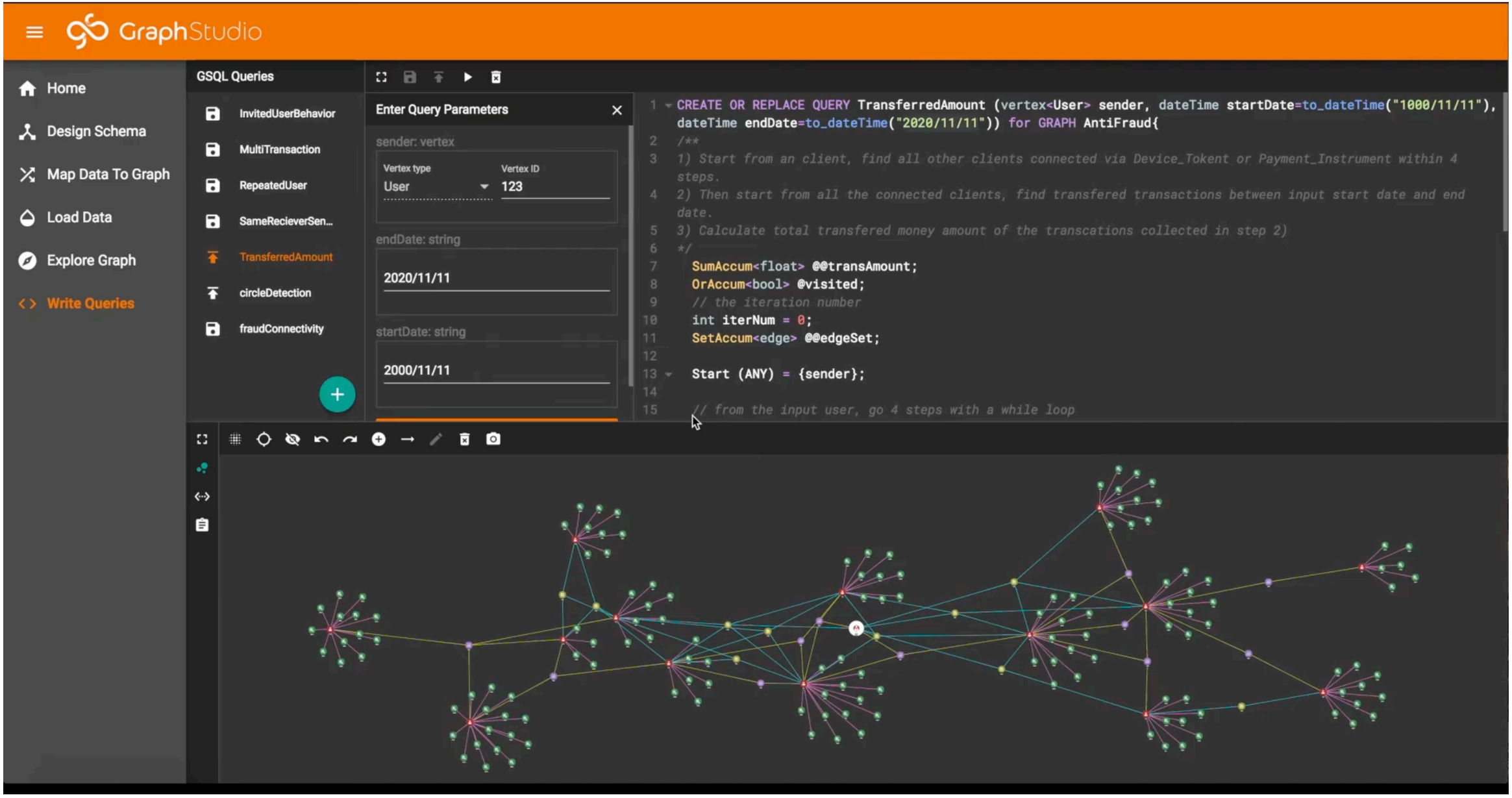
Can you handle them?

- How common are these in business analytics
- You can break them up into binary relations

Extra Analytics Capabilities

What else do Graph DBs offer?

- Can consider variable-length paths
- Can process and analyse such paths
- Output of a query can be a graph
  - Very nice for interactive analytics


GraphStudio
Home
Design Schema
Map Data To Graph
Load Data
Explore Graph
Write Queries
GSQL Queries
InvitedUserBehavior
MultiTransaction
RepeatedUser
SameRecieverSen...
TransferredAmount
circleDetection
fraudConnectivity
Enter Query Parameters
sender: vertex
Vertex type
User
Vertex ID
123
endDate: string
2020/11/11
startDate: string
2000/11/11
CREATE OR REPLACE QUERY TransferredAmount (vertex<User> sender, dateTime startDate=to_dateTime("1000/11/11"),
dateTime endDate=to_dateTime("2020/11/11")) for GRAPH AntiFraud{
/**
1) Start from an client, find all other clients connected via Device_Tokent or Payment_Instrument within 4
steps.
2) Then start from all the connected clients, find transfered transactions between input start date and end
date.
3) Calculate total transfered money amount of the transcations collected in step 2)
*/
SumAccum<float> @@transAmount;
OrAccum<bool> @visited;
// the iteration number
int iterNum = 0;
SetAccum<edge> @@edgeSet;
Start (ANY) = {sender};
// from the input user, go 4 steps with a while loop

Why not switch right now?

- That’s the eventual plan
- Some challenges still remain:
  - Technology is still young, we are not able to scale to PB Data Lakes just yet
  - All data analysts learned SQL
  - Still no standard Graph Query Language

Extra capabilities of Graph DBs

We are not done yet!

- Graph algorithms on Big Data!
- Machine Learning features:
  - More complex features
  - Graphlets and Embeddings
- In-database Machine Learning
  - Real-time scoring
  - GNNs

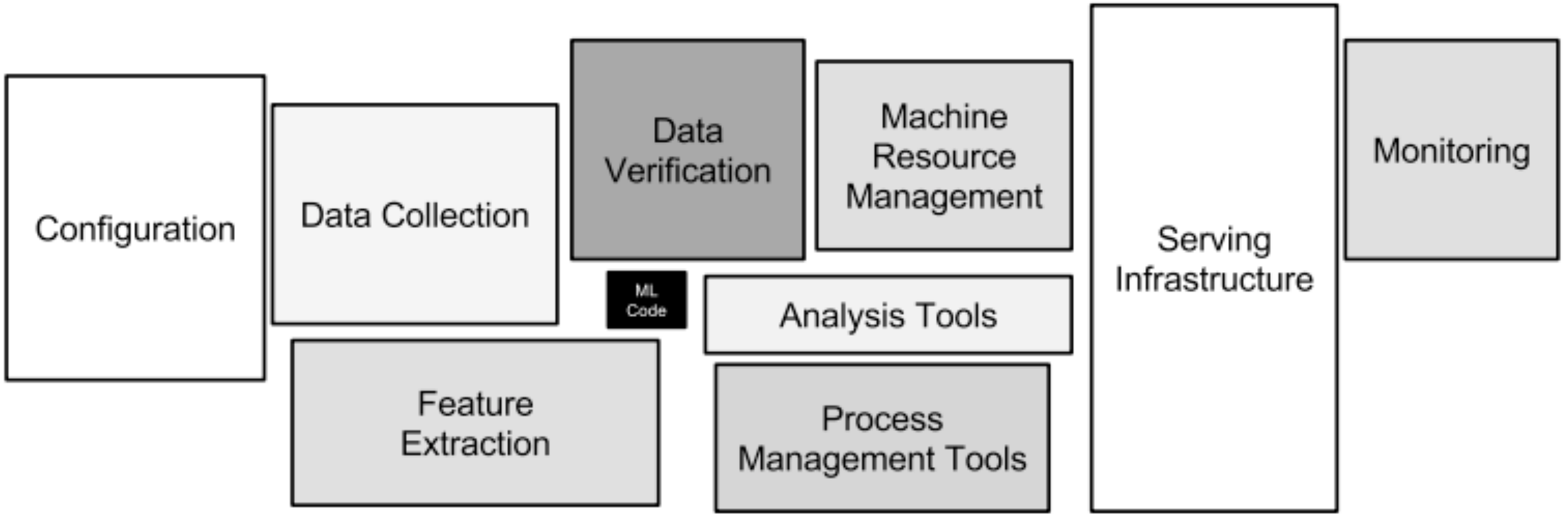
Graph Algorithms on Big Data

Has this been done before?

- Sure: PageRank on MapReduce, some algorithms in GraphX
- But no real in-database Graph Algorithms capability before
  - Techically there is MADLib, but its pretty limited and not MPP
- Advantages of in-database algorithms

In-database ML

- How do we usually work with ML and a database:
  - Download a sample of data into Python notebook
  - Extract features, build a model
  - Deploy a model into a separate service
  - Build a separate data pipeline for this service
  - Optionally retrain and redeploy the model



Hidden Technical Debt in Machine Learning Systems, NeurIPS 2015

How in-database ML solves this

- More options:
  - Also download a sample from a Graph Database
  - Train a model within a Graph Database
- Deploy a model within Graph Database
  - No need to replicate data pipeline into a separate service
  - Many options on how to perform real-time scoring
    - E.g. query-based, tigger-based, etc

Conclusions

- Beats RDMSs on their own turf (might not last that long)
- New analytic capabilities (variable length paths)
- Parallel Graph Algorithms on Big Data
- Better way to train and deploy models for Graph Data

Peeking into the future of MPP Graph Databases

- The systems are quite new - a few years in the industry
- Everybody has their own query language
- GQL standard by ISO is coming in 2023
- TigerGraph doesn’t scale yet to PB Graph Data, but this is changing fast!
- TigerGraph is closed-source proprietary, problems for countries like Russia
- Open-source alternative is Nebula: much less mature though

Some important references

- Vertex-centric parallel computation of SQL queries
  - https://dl.acm.org/doi/10.1145/3448016.3457314
  - Shows how joins are performed in TigerGraph and provides benchmark data on TPC-DS
- TigerGraph LDBC Benchmark results
  - https://www.tigergraph.com/benchmark/

Useful links

- TigerGraph web site: https://www.tigergraph.com/
- Nebula web site: https://nebula-graph.io/
- Demo of interactive analytics: https://testdrive.tigergraph.com/app/home
- Free TigerGraph Cloud version: https://www.tigergraph.com/cloud/
- Free Enterprise Version (Docker): https://info.tigergraph.com/enterprise-free
- Graph + AI Summit: https://www.tigergraph.com/graphaisummit/