A bit about myself related to graphs

• Long standing interest in graphs since 2003
  • Two books on graphs
  • 19/50+ publications on graphs (2700+ citations)

• Current: Principal Scientist Manager @ GSL
  • Working on graph projects with Azure Data and Liquid Team @ LinkedIn

• Past: Principal Research Staff Member @ IBM Research
  • The tech lead for IBM Db2 Graph product
Market Research

- The global graph analytics market in 2022 was valued at $1.14B
- Global market for graph databases will grow at projected CAGR of 34.8% during forecasted period (2023-2028) and reach $6.9B by 2028.
Customer Use Cases

- Finance
- Insurance
- Healthcare
- Security
- Retail
- Energy
- Power
- Manufacturing
- Supply chains
- Transportation
Graph Workloads

Graph Queries (Graph OLTP)
- low-latency graph traversal and pattern matching
  - e.g. neighbors of a vertex, shortest path between two vertices

Graph Algorithms (Graph OLAP)
- iterative, long running, graph processing
  - e.g. Pagerank, community detection

New Trend: Graph ML/AI
- GNN
  - e.g. node embedding, graph embedding
  - vector
Graph Models

Resource Description Framework (RDF)
• Directed, edge-labeled graph (subject-predicate-object triple)
• Application
  • Knowledge representation & inference
  • Semantic Web

Property Graphs (PG)
• Directed, vertex-labeled, and edge-labeled graph with properties on each vertex/edge
• Application
  • Graph traversals/pattern matching
  • Path/graph analytics

Most graph databases support PG model

[Diagram showing RDF and PG models with patient 1, disease 1, disease 2, diagnosis 1, and diabetes relationships]
Graph languages/Interfaces

Graph OLTP

- SPARQL for RDF graphs
- Chaos for PG graphs
  - Gremlin (imperative)
  - Supported by ~30 graph vendors
  - openCypher (declarative)
  - Supported by ~10 graph vendors
- ISO Standard efforts: GQL and SQL/PGQ (declarative)

Need a few years to settle down!

Choose a standard/popular language and avoid inventing new ones!

Graph OLAP

- API/DSL + built-in algorithms
  - Most support Pregel-like API

https://www.gqlstandards.org/existing-languages
Graph Technology Landscape 2022
## Competitive Landscape

| Company                        | Deployment                  | Graph Model | Query Language       | Visualization tools | Transaction       | Graph OLAP                                                                 | Scale-Out |
|--------------------------------|-----------------------------|-------------|----------------------|--------------------|-------------------|---------------------------------------------------------------------------|-----------|
| **Graph Only Companies**       |                             |             |                      |                    |                   |                                                                           |           |
| TigerGraph                     | On-prem / AWS, Azure, GCP   | PG          | GSQL                 | Graph Studio       | ACID              | GSQL, 23 built-in algorithms                                              | Yes       |
| Neo4J                          | On-prem / AWS, Azure, GCP   | PG          | Cypher               | Studio             | Non-repeatable reads may occur | Pregel API, 48 built-in algorithms (including Graph ML) | Yes*      |
| DataStax Enterprise Graph      | On-prem / AWS, Azure, GCP   | PG          | Gremlin              | Studio             | Row-level (Cassandra)  | SparkGraphComputer API                                                    | Yes       |
| Databricks GraphX & GraphFrames| On-prem / AWS, Azure, GCP   | PG          | Motif Finding DSL    | -                  | -                 | Pregel API, 7 built-in algorithms                                          | Yes       |
| **Data Companies**             |                             |             |                      |                    |                   |                                                                           |           |
| Amazon Neptune                 | AWS                         | PG, RDF     | Gremlin, SPARQL      | Neptune Workbench  | ACID              |                                                                           |           |
| Microsoft SQL Graph            | On-prem / Azure             | PG          | SQL Extension        | Power BI plugin, 3rd party tools | ACID | Python/R scripts via Machine Learning Services | Yes* (Read-Only Queries) |
| Microsoft Cosmos DB Graph      | Azure                       | PG          | Gremlin              | Azure Portal, 3rd party tools | -                |                                                                           | Yes       |
| Oracle Spatial and Graph       | On-prem / OCI AWS, Azure, GCP | PG, RDF    | PGQL, SPARQL         | Graph Studio       | ACID              | Green Marl DSL, 50+ built-in algorithms (including Graph ML)              | Yes       |
| IBM Db2 Graph                  | On-prem / CP4D              | PG          | Gremlin              | Graph UI           | ACID              |                                                                           | Yes       |
Graph Solution Space

Native Graph DB
- Everything from scratch
- Pros: performance
- Cons: high engineering cost

Hybrid Graph DB
- Graph engine + existing backend store
- Pros: faster development, leverage backend store
- Cons: performance

Native Graph DB
- Neo4j
- TigerGraph

Hybrid Graph DB
- AWS Neptune
- IBM Db2 Graph
- Oracle Spatial & Graph
- DataStax Graph
- Microsoft SQL Graph
- Microsoft Cosmos DB Graph
Graph Solution Space

Graph-only DB
• Only support graph workload (con)

Converged DB (Multi Model)

Hybrid Graph DB
• Support poly query languages/APIs on the shared data (pro)

AWS Neptune

Native Graph DB
Neo4j
TigerGraph

IBM Db2 Graph
Oracle Spatial & Graph
DataStax Graph
Microsoft SQL Graph
Microsoft Cosmos DB Graph

Hybrid Graph DB

Converged DB
Advantage of Converged DB solution

Poly languages/APIs on shared data
- View the data in the way that is needed!
- No data transfer or transformation cost
- *If graph queries on original data* (no schema change, no secondary copy)
  - No disturbance of existing applications
  - Transaction updates are visible to graph analysis in real time

Leverage of existing backend data store
- Transaction support
- Access control
- Compliance to audits and regulations
- Temporal support
- Scalability
- HA & DR
Graph Benchmarks

- Graph500 Benchmark
- HPC Scalable Graph Analysis Benchmark
- LinkBench
- Open Graph Benchmark
- **LDBC Benchmarks (most comprehensive)**
  - LDBC-SNB (used by TigerGraph and Neo4j)
  - LDBC Graphalytics
  - LDBC SPB
  - FinBench

*All future performance studies should adopt LDBC benchmarks!*
Opportunities and Directions

Growing market for graph databases (CAGR 34.8%, $6.9B by 2027)

Graph-only vendors are currently leading
- Strength: performance and algorithm support
- Weakness: Data import/export is a bottleneck for end-to-end scenarios

Major cloud vendors are investing in graph space
- Advantage: they own the whole stack, including the source of truth
Recommendation for Researchers (more practical impact)

• Use widely-adopted graph models, languages, and benchmarks

• Practical challenges that industry faces:
  • Multi-tenancy and access control
  • Security and compliance
  • End-to-end pipelines with mixed graph and non-graph workloads
  • Dynamic graphs
Questions & Suggestions?

THANK YOU!