A Comparison of Current Graph Database Models

Renzo Angles
Universidad de Talca (Chile)

3rd Int. Workshop on Graph Data Management: Techniques and applications (GDM 2012)
5 April, Washington DC, USA

Outline

1. Introduction
2. Current graph databases
3. Comparison of graph database models
4. Conclusions and future work

Outline

1. <u>Introduction</u>

2. Current graph databases

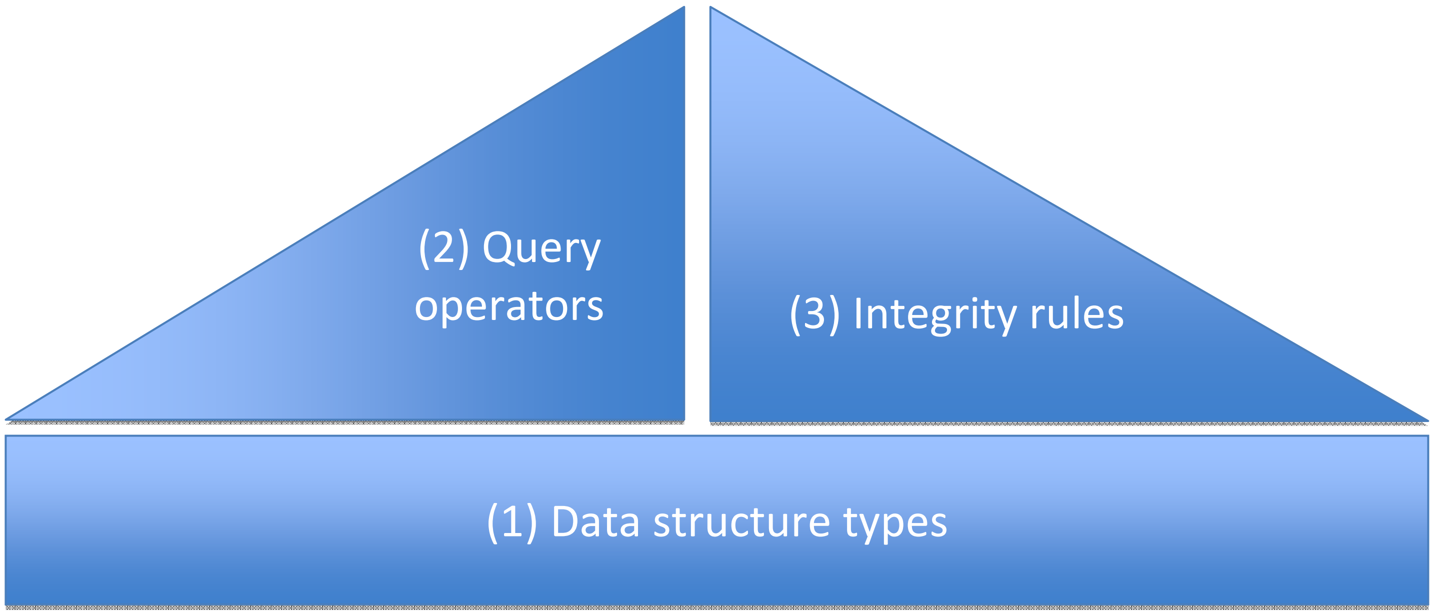
3. Comparison of graph database models

4. Conclusions and future work

Database models

- A **data model** is a collection of conceptual tools used to model real-world entities and the relationships among them [Silberschatz et al. 1996].

- A DB model consists of 3 components [Codd 1980]:



Graph database model

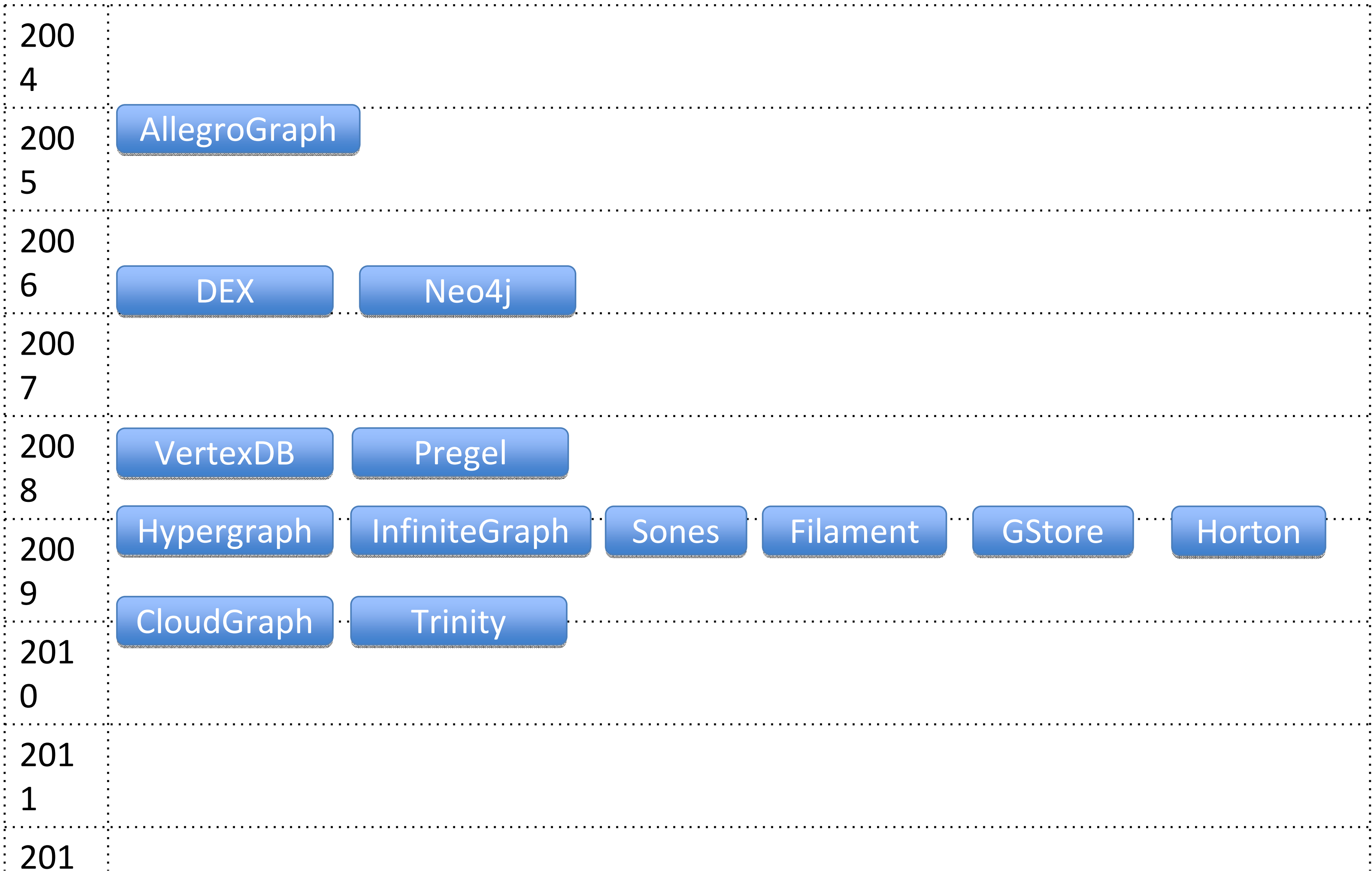
1. Graph data structures
   - Simple graphs (nodes + edges + labels + direction)
   - Generalizations: nested graphs (hypernodes), hypergraphs (hyperedges), attributed graphs
2. Graph-oriented operations
   - Simple functions (e.g., shortest-path)
   - Graph query language (operators)
   - Domain-specific queries: graph pattern mining
3. Graph integrity constraints
   - Schema-instance consistency
   - Identification of nodes, atributes and relations
   - Path constraints
   - …

Graph databases (before 2003)

1975	R&M		
…			
1984	LDM		
. . .			
1987	G-Base		
1988	O2		
1989	Tompa		
1990	GOOD	W&X	Hypernode
1991	GROOVI		
1992	GMOD	Gram	Simatic-XT
1993	PaMal	GOAL	Hy+
1994	GraphDB	DGV	Hypernode2
1995	G-Log	GGL	Hypernode3
1996	GRAS		
…			
1999	GOQL		
…			
2002	GDM		

R. Angles and C. Gutierrez.
“Survey of Graph Database Models”.
ACM Comp. Surveys, 2008.

Graph databases (after 2003)



Motivation

1. What is the most suitable graph database?
   – Empirical comparison: desirable but hard
   – Benchmark: there is not a standard one
   – The application domain is also important

2. Objective: comparison of graph data models
   • Independent of implementation
   • Easier to evaluate (vs empirical evaluation)
   • Shows (in advance) the expressive power for data modeling

Outline

1. Introduction
2. Current graph databases
3. Comparison of graph database models
4. Conclusions and future work

Current graph databases

Graph database systems

- AllegroGraph (2005): SW-oriented databases
- DEX (2007): bitmaps-based graph database
- Neo4J (2007): disk-based transactional graph database
- HyperGraphDB (2010): hypergraph-based database
- InfiniteGraph (2010): distributed-oriented system
- Sones (2010): object-oriented database

Current graph databases

Graph stores

- VertexDB (2009): key-value disk store (TokyoCabinet)
- Filament (2010): graph library on PostgreSQL
- G-Store (2010): prototype
- Redis_graph (2010): implemented on python

Work in progress

- Pregel (Google, 2009): vertex-based intrastructure for graphs
- Horton (Microsoft, 2010): transactional graph processing
- CloudGraph (2010): use MySQL as backed
- Trinity (Microsoft, 2011): RAM-based key value store

Current graph databases

Related DB technologies

- Web-oriented DBs: InfoGrid , FlockDB
- Document-oriented DBs: OrientDB
- Triple stores (RDF DBs): 4Store, Virtuoso, Bigdata
- Distributed graph processing: Angrapa, Apache Hama, Giraph, GoldenOrb, Phoebus, KDT, Signal Collect, HipG

Outline

1. Introduction
2. Current graph databases
3. <u>Comparison of graph database models</u>
4. Conclusions and future work

Data storing features

Graph Database	Main memory	External memory	Backend Storage	Indexes
AllegroGraph	•	•		•
DEX	•	•		•
Filament	•		•	
G-Store		•		
HyperGraphDB	•	•	•	•
InfiniteGraph		•		•
Neo4j	•	•		•
Sones	•			•
vertexDB		•	•	

- **Backed:** RDB (filament), BerkeleyDB (HypergraphDB), TokyoCabinet (vertexDB)
- **Indexing**: nodes, relations, atributes, triples
- **Data formats**: GraphML, graphViz, N-Triples, RDF-XML,
- **ACID (partial support)**: Allegro, HypergraphDB, Infinite, Neo4j

Operation and manipulation features

Graph Database	Data Definition Language	Data Manipulat. Language	Query Language	API	GUI
AllegroGraph	•	•	•	•	•
DEX				•	
Filament				•	
G-Store	•		•	•	
HyperGraphDB				•	
InfiniteGraph				•	
Neo4j				•	
Sones	•	•	•	•	•
vertexDB				•	

- **Languages:** SPARQL 1.0(.1), GraphQL, (Sones)
- **Why is an API the main approach?**

  does it facilities the development of applications?

Graph data structures

	Graphs				Nodes		Edges		
Graph Database	Simple graphs	Hypergraphs	Nested graphs	Attributed graphs	Node labeled	Node attribution	Directed	Edge labeled	Edge attribution
AllegroGraph	•				•		•	•	
DEX				•	•	•	•	•	•
Filament	•				•		•	•	
G-Store	•				•		•	•	
HyperGraphDB		•			•		•	•	
InfiniteGraph				•	•	•	•	•	•
Neo4j				•	•	•	•	•	•
Sones		•		•	•	•	•	•	•
vertexDB	•				•		•	•	

– **Node/edge atribution**: implementation considerations

Schema and instance representation

	Schema			Instance					
Graph Database	Node types	Property types	Relation types	Object nodes	Value nodes	Complex nodes	Object relations	Simple relations	Complex relations
AllegroGraph					•			•	
DEX	•		•	•	•		•	•	
Filament					•			•	
G-Store					•			•	
HyperGraphDB	•		•		•			•	•
InfiniteGraph	•		•	•	•		•	•	
Neo4j				•	•		•	•	
Sones					•			•	•
vertexDB					•			•	

- Node/edge identification: ObjectsI-Ds vs Values
- Support for complex relations (hyperedges = n-ary relations)

Query features

	Type			Use		
Graph Database	Query Lang.	API	Graphical Q. L.	Retrieval	Reasoning	Analysis
AllegroGraph	○	●	●	●	●	●
DEX		●		●		●
Filament		●		●		
G-Store	●			●		
HyperGraphDB		●		●		
InfiniteGraph		●		●		
Neo4j	○	●		●		
Sones	●		●	●		●
vertexDB		●		●		

- **Declarative QL:** SPARQL, Prolog, Lisp, GraphQL
- **Reasoning**: RDF(S), OWL
- **Analysis**: Social Networking, graph statistics

Integrity constraints

Graph Database	Types checking	Node/edge identity	Referential integrity	Cardinality checking	Functional dependency	Graph pattern constrains
DEX	•	•	•			
HyperGraphDB	•	•				
InfiniteGraph	•	•				
Sones		•		•		

- Most oriented to be schema-less
- Graph integrity constraints: theoretical interest

Support for essential graph queries

	Adjacency		Reachability				
Graph Database	Node/edge adjacency	k-neighborhood	Fixed-length paths	Regular simple paths	Shortest path	Pattern matching	Summarization
Allegro	●		●			●	
DEX	●		●	●	●	●	
Filament	●		●			●	
G-Store	●		●	●	●	●	
HyperGraph	●					●	
Infinite	●		●	●	●	●	
Neo4j	●		●	●	●	●	
Sones	●					●	
vertexDB	●		●	●		●	

Outline

1. Introduction

2. Current graph databases

3. Comparison of graph database models

4. <u>Conclusions and future work</u>

Conclusions

General balance of graph database models

- **Data structures**:
  - Several types of graph structures
  - Good expressive power for data modeling
- Query features:
  - Restricted to provide APIs
  - Good support for essential graph queries
  - Lack of graph query languages
- **Integrity constraints:**
  - Basic notions of restrictions
  - Oriented to be schema-less

Future work

- Empirical evaluation of graph databases
- Development of a Benchmark for GDBs
- Comparison with other database technologies (in particular RDF databases)

Thanks for your attention!

Questions?