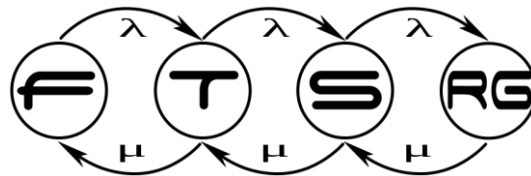


Budapest Neo4j Meetup - 2019/06/25

What makes graph queries difficult?

Gábor Szárnyas
szarnyas@mit.bme.hu

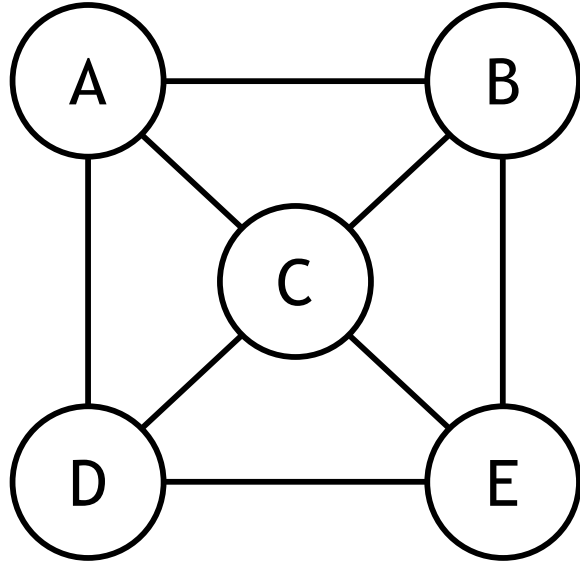


**Hungarian Academy of
Sciences**

With contributions from Petra Várhegyi and Bálint Hegyi

The property graph data model

SIMPLE GRAPH



5 people

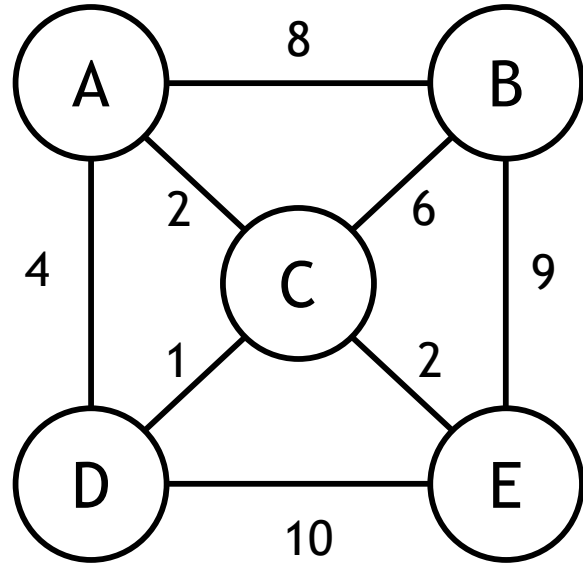
- Many of them know each other

This is a **simple graph**.

Algorithms:

- breadth-first search
- depth-first search
- PageRank
- connected components

ADD EDGE WEIGHTS



5 people

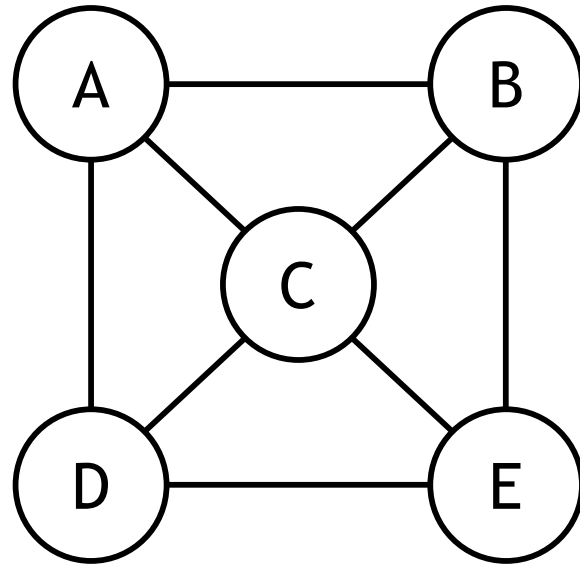
- Weight: communication cost

This is a **weighted graph**.

Algorithms:

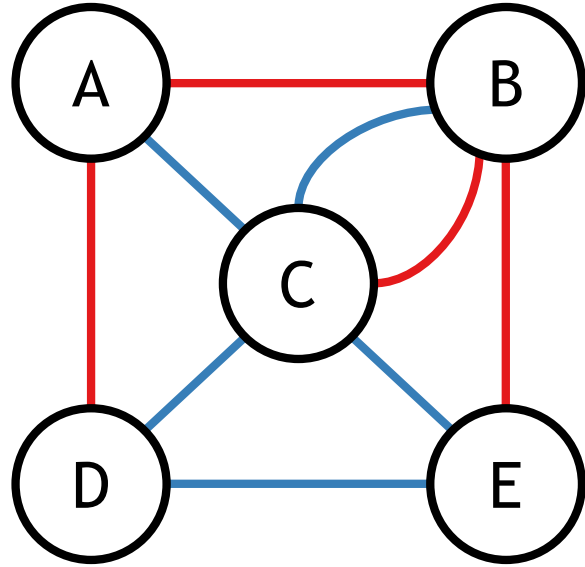
- shortest path algorithms
- max-flow

ADD EDGE TYPES



5 people

ADD EDGE TYPES



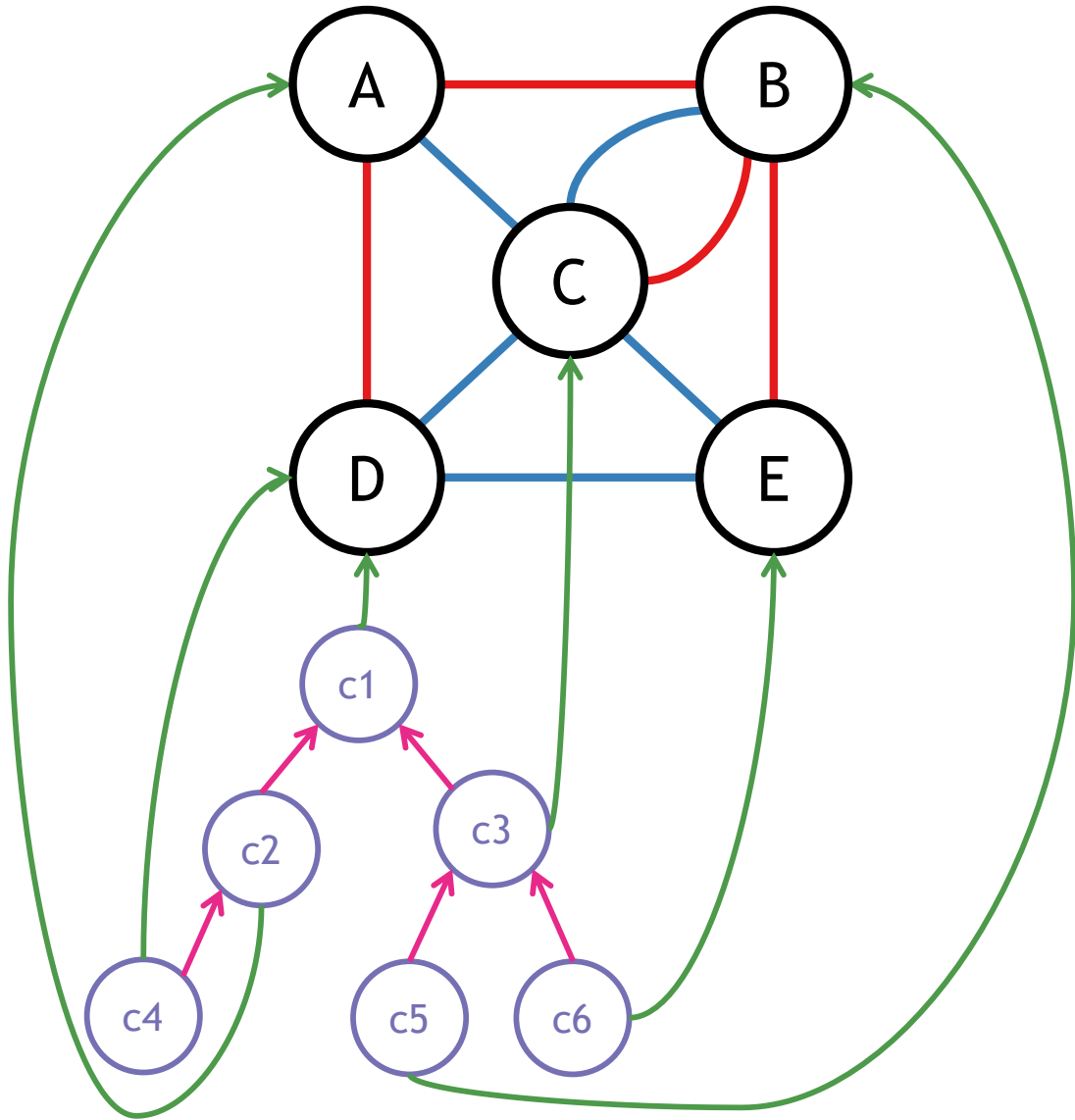
5 people

- Business partners
- Friends

Multiple edge types
but only a single node type.

This is an **edge-typed graph**.

ADD NODE AND EDGE TYPES



5 people ●

■ Business partners

■ Friends

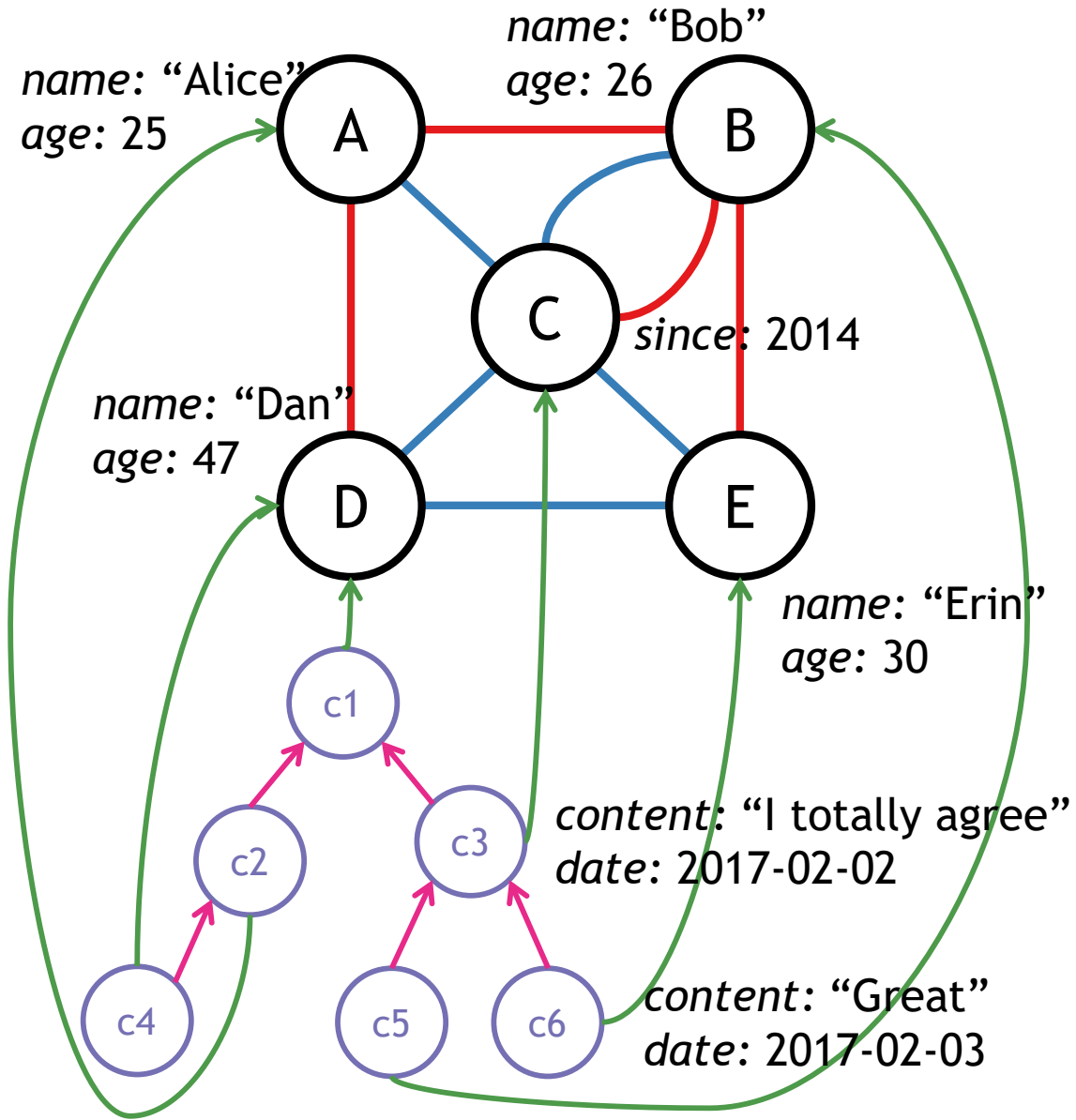
6 comments ●

■ Replying to another comment

■ Authored by a given person

This is a **typed graph**.

ADD PROPERTIES



5 people ● - *name, age*

■ Business partners

■ Friends - *since*

6 comments ● - *content, date*

■ Replying to another comment

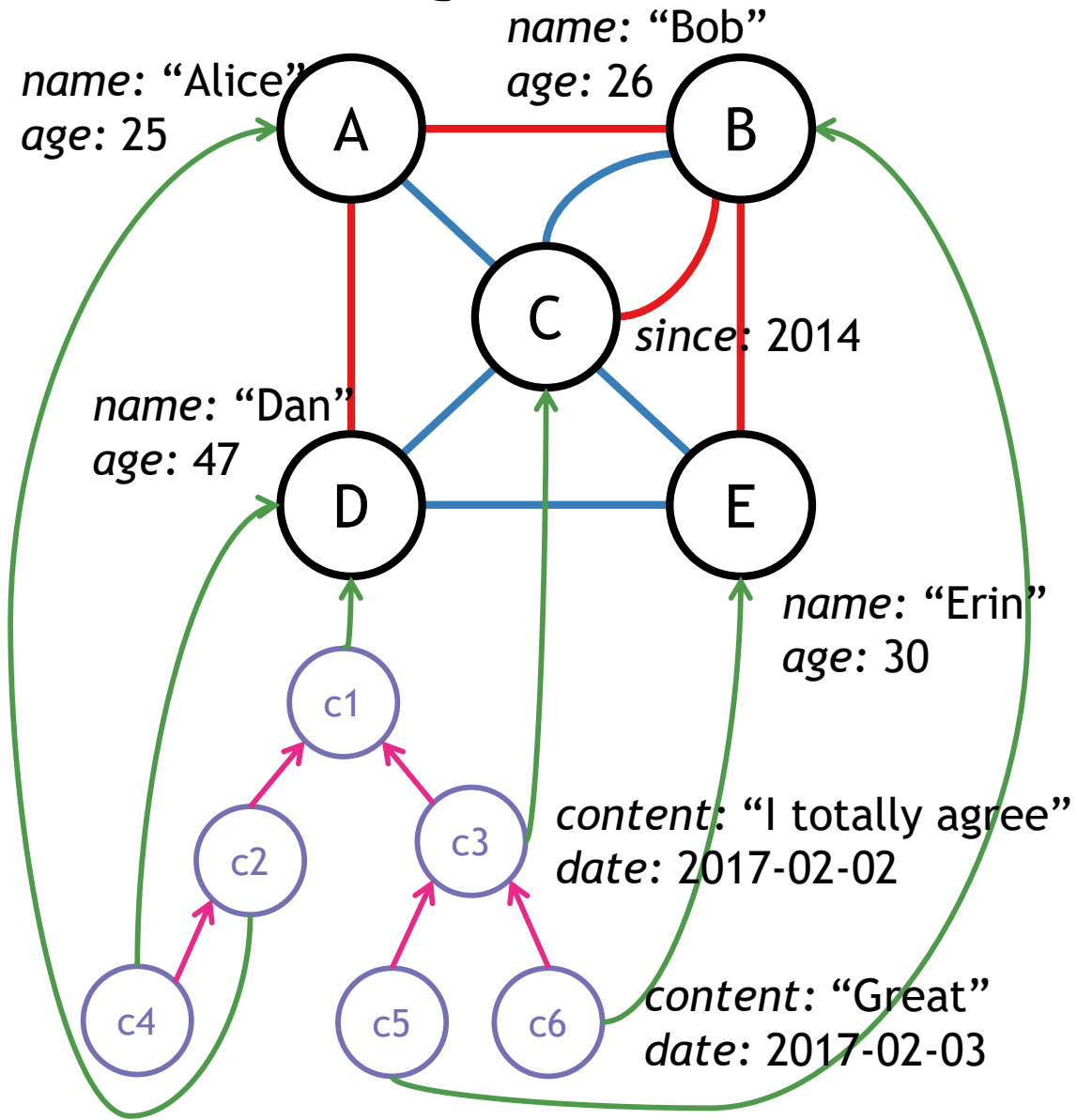
■ Authored by a given person

This is a **property graph**.

Similar to object-oriented data.

Graph processing: Queries and analytics

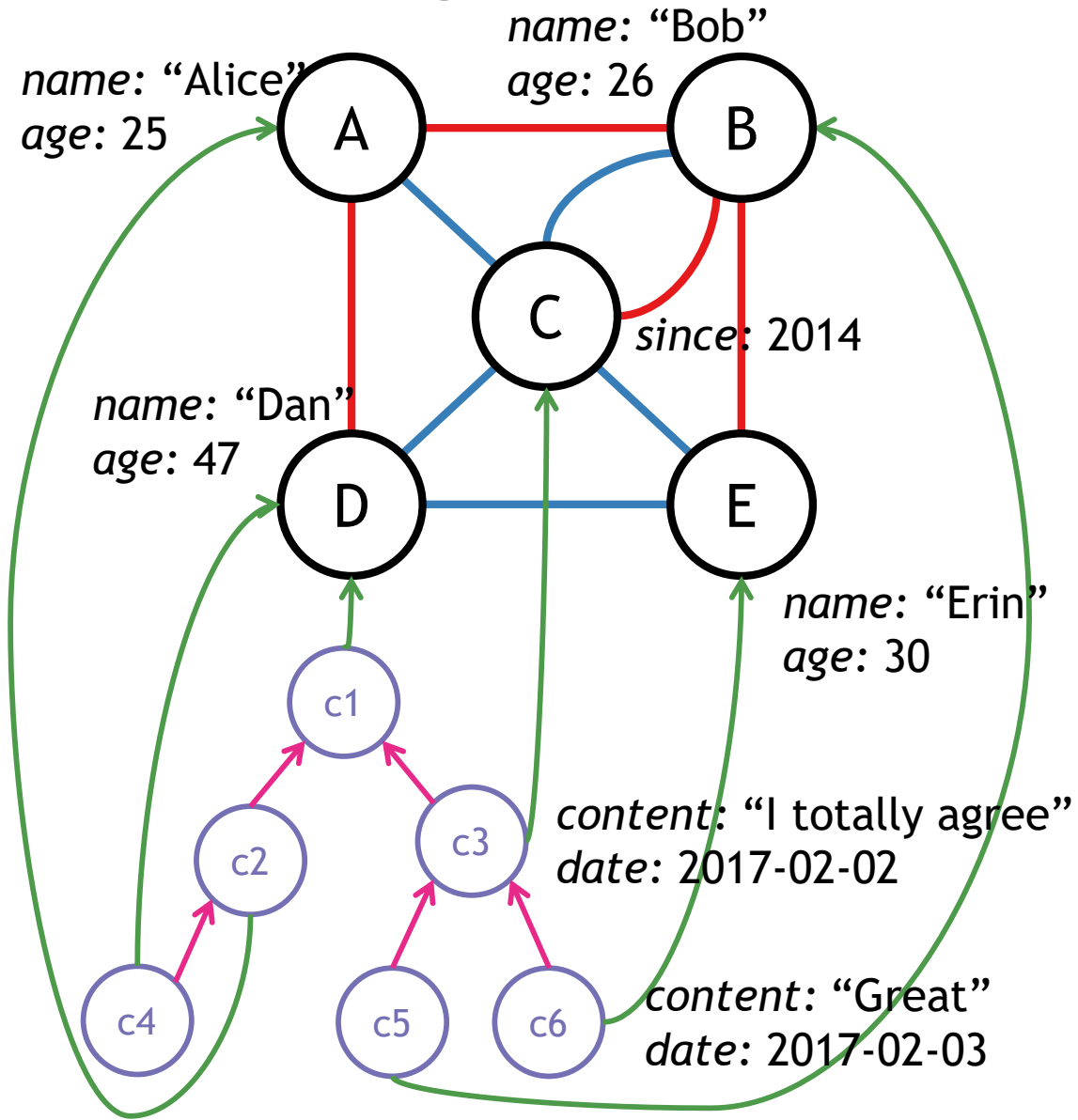
GRAPH QUERIES: LOCAL



Local graph query:
Return “Dan” and his comments.

Well-researched topic.
Typical execution times are low.

GRAPH QUERIES: GLOBAL



Global graph query:

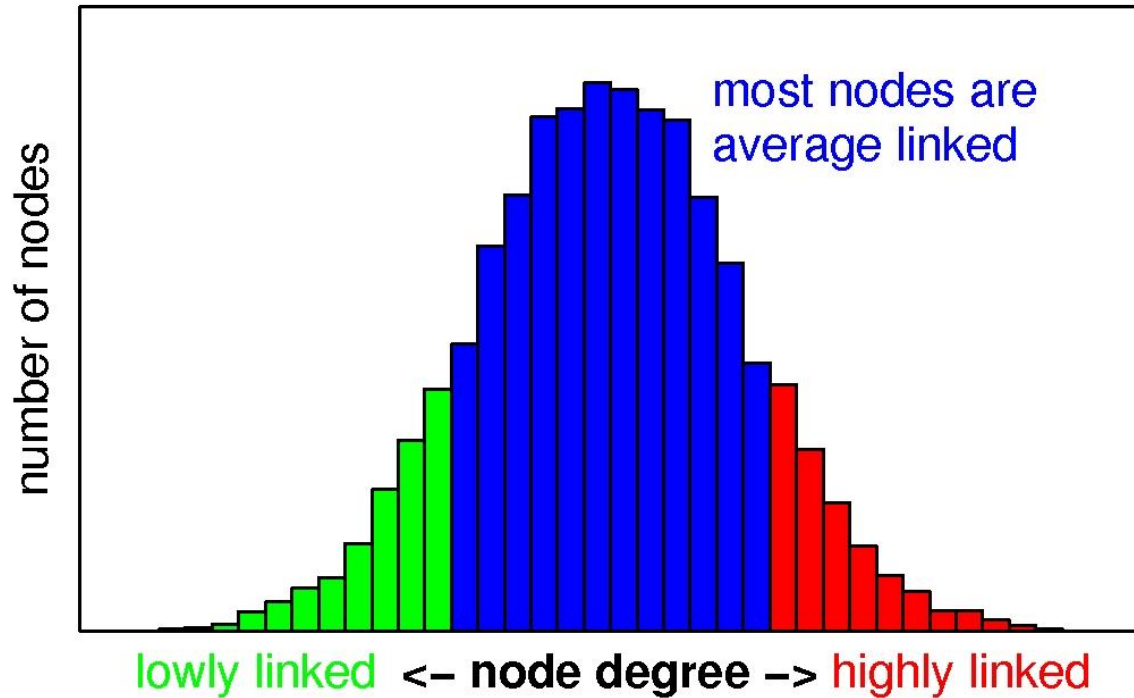
Find people who had no interaction with "Cecil" through any comments, neither replying nor receiving a reply.

The result is „Alice”.

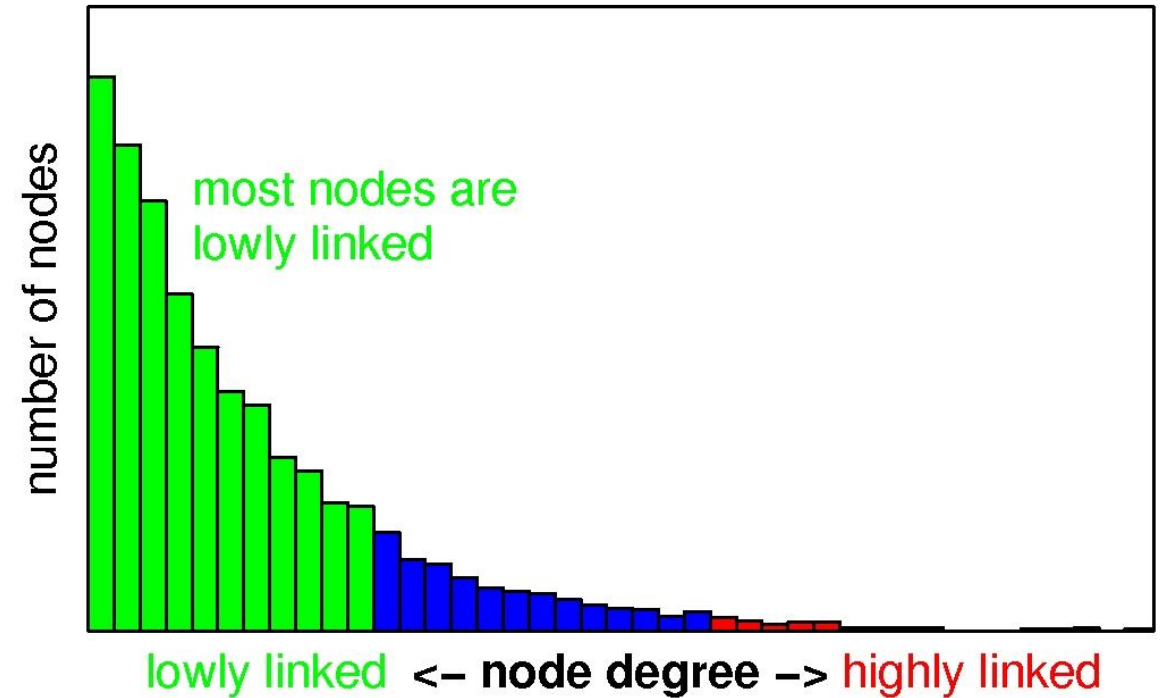
Typical execution times are high.

GRAPH ANALYTICS: NETWORK SCIENCE

random networks

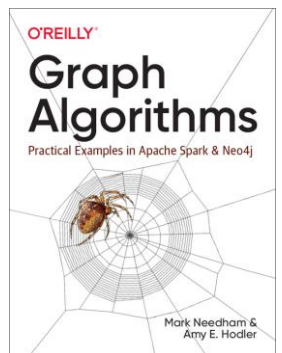


real networks (power-law, scale-free)

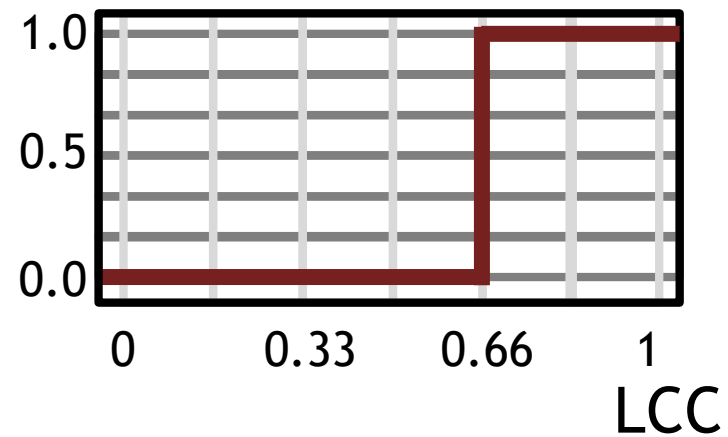
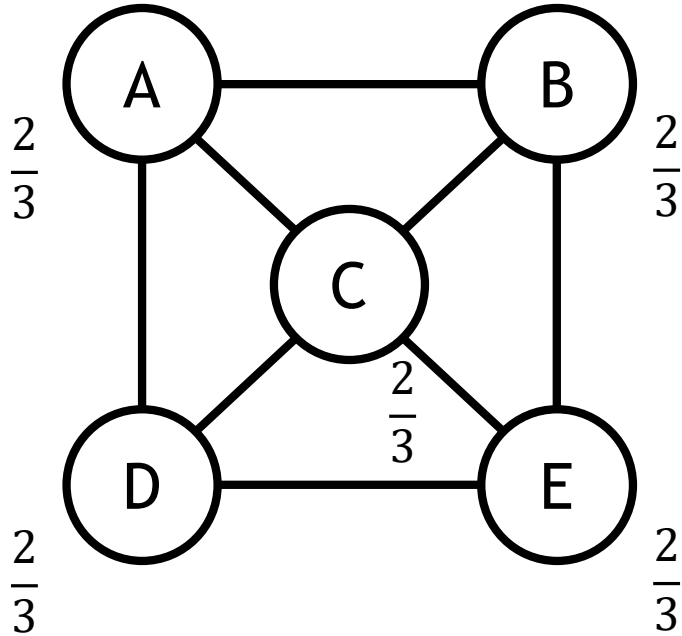


www.network-science.org

- Studies the *structure of graphs*
- Pioneered by László Barabási-Albert et al.
- Degree distributions, clustering coefficient, etc.



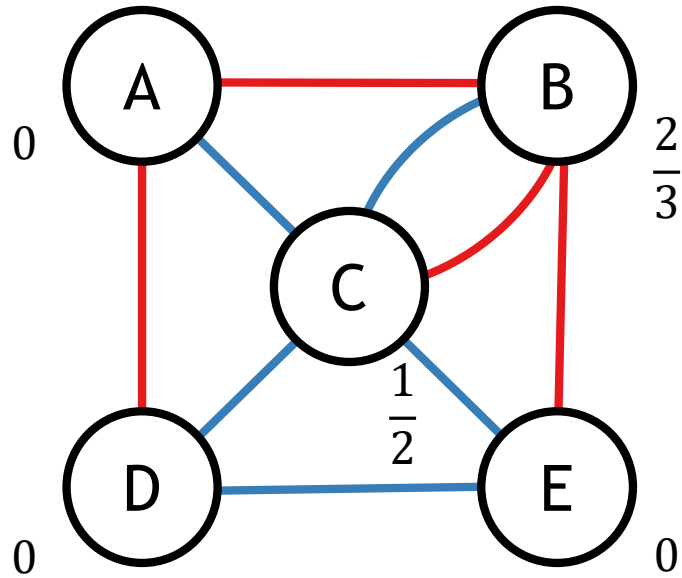
LOCAL CLUSTERING COEFFICIENT



$$\text{LCC}(v) = \frac{\text{Number of triangles containing } v}{\text{Number of neighbors of } v \text{ choose } 2}$$

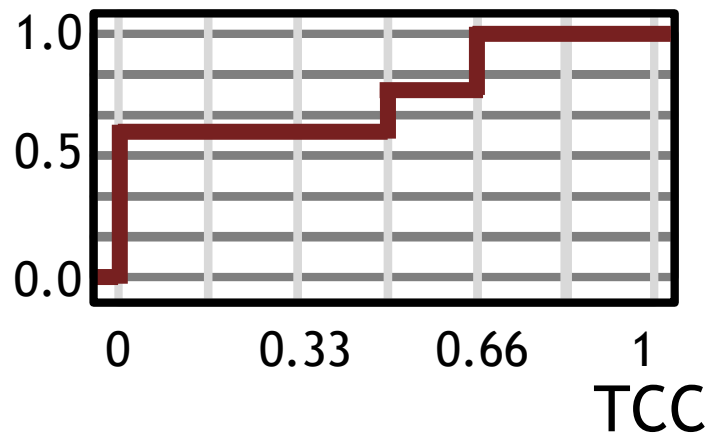
The empirical cumulative distribution function does not present much useful information in this case.

TYPED CLUSTERING COEFFICIENT



$$\text{TCC}(v) = \frac{\text{Red Triangles} + \text{Blue Triangles}}{\text{Red Stars} + \text{Blue Stars}}$$

The diagram illustrates the formula for the Typed Clustering Coefficient (TCC) for a node v . The numerator shows two types of triangles: one with two red edges and one blue edge, and another with one red edge and two blue edges. The denominator shows two types of stars: one with two red edges and one with two blue edges.

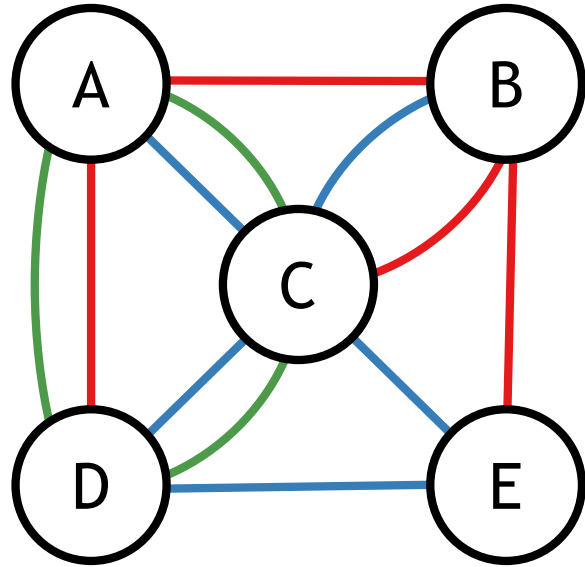


More information

High combinatorial complexity:

- t types $\rightarrow t \times (t - 1)$ triangles
- $\mathcal{O}(t^2)$ steps

TYPED CLUSTERING COEFFICIENT



■ Business partners

■ Friends

■ Family member

3 types \rightarrow 6 triangles

$$\text{TCC}(v) = \frac{\begin{array}{c} \text{triangle with } v \text{ and red edges} \\ + \\ \text{triangle with } v \text{ and red edges} \\ + \\ \text{triangle with } v \text{ and blue edges} \\ + \\ \text{triangle with } v \text{ and blue edges} \\ + \\ \text{triangle with } v \text{ and green edges} \\ + \\ \text{triangle with } v \text{ and green edges} \end{array}}{\begin{array}{c} \text{star with } v \text{ and red edges} \\ + \\ \text{star with } v \text{ and blue edges} \\ + \\ \text{star with } v \text{ and green edges} \end{array}}$$

F. Battiston et al.:

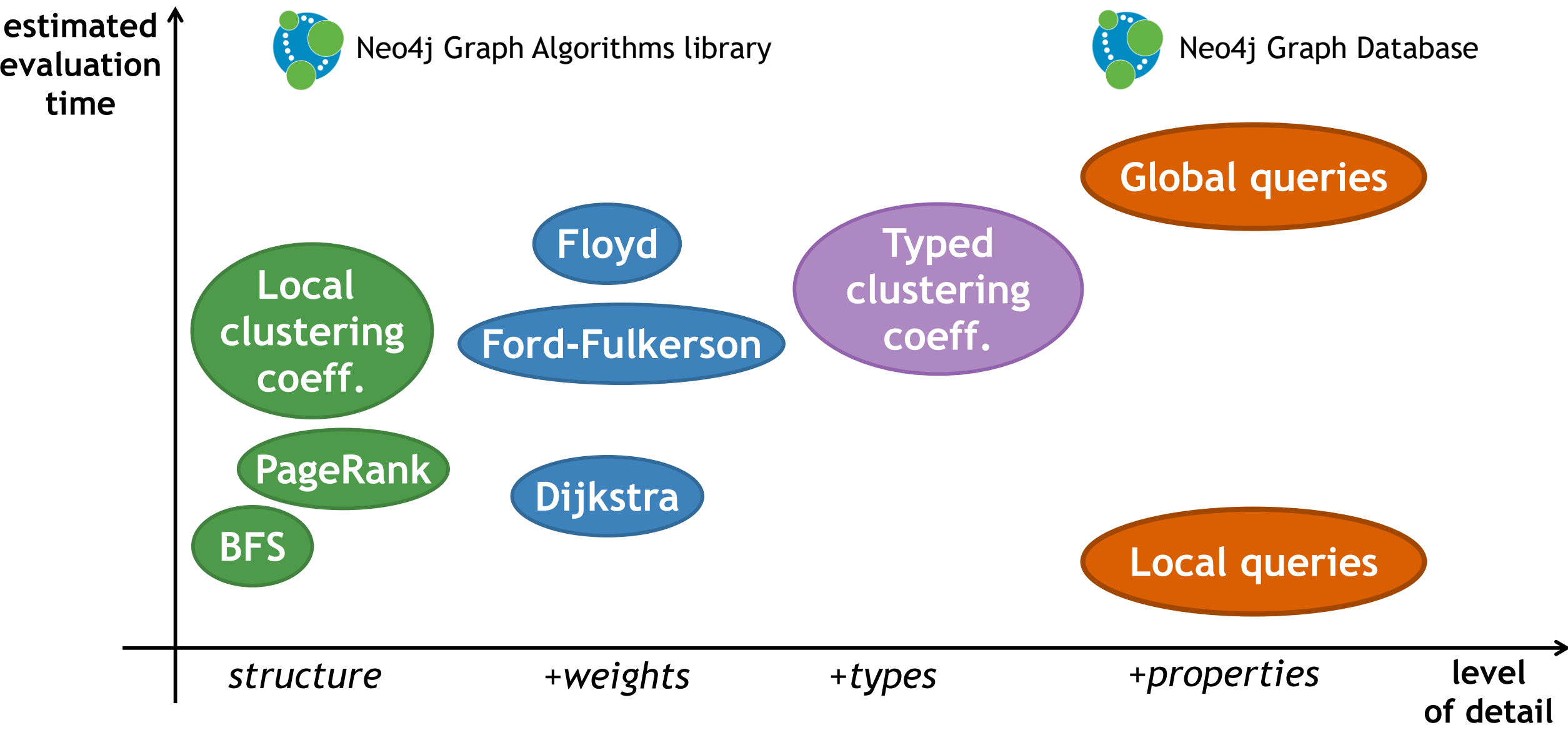
Structural measures for multiplex networks,
Physical Review E, 2014



Petra Várhegyi:

Multidimensional Graph Analytics,
Master's thesis, 2018

GRAPH PROCESSING TECHNIQUES AND LANGUAGES



GRAPH PROCESSING TECHNIQUES AND LANGUAGES

estimated
evaluation
time



Neo4j Graph Algorithms library



Neo4j Graph Database

Local
clustering
coeff.

Floyd

Ford-Full

Typ

coeff.

Global queries

Pr

Dijkstra

Local queries

evaluation time

structure

+weights

+types

+properties

level
of detail

Graph processing tools and challenges

GRAPH PROCESSING CHALLENGES / STRUCTURE

connectedness

the “curse of connectedness”

computer
architectures

data structures contemporary computer architectures are good at processing are linear and simple hierarchical structures, such as *Lists*, *Stacks*, or *Trees*

caching and
parallelization

a massive amount of random data access is required [...] poor performance since the CPU cache is not in effect for most of the time. [...] parallelism is difficult to extract because of the unstructured nature of graphs.



B. Shao, Y. Li, H. Wang, H. Xia (Microsoft Research):
Trinity Graph Engine and its Applications,
IEEE Data Engineering Bulletin 2017

GRAPH PROCESSING CHALLENGES / PROPERTIES

topology

existing graph query methods [...] focus on the topological structure of graphs and few have considered attributed graphs.

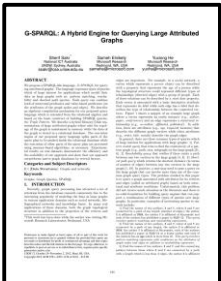
properties

applications of large graph databases would involve querying the graph data (attributes) in addition to the graph topology.

complex optimization

answering queries that involve predicates on the attributes of the graphs in addition to the topological structure [...] makes evaluation and optimization more complex.

S. Sakr, S. Elnikety, Y. He (Microsoft Research):
G-SPARQL: A Hybrid Engine for Querying Large Attributed Graphs,
CIKM 2012



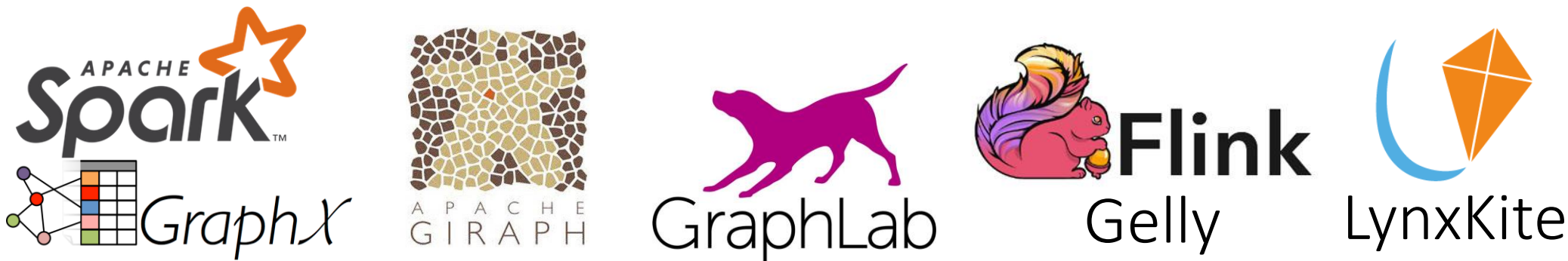
GRAPH PROCESSING TOOLS

Currently, there is a strong distinction between graph query and analytical tools - this might change in the future.

graph
queries



graph
analytics



János Szendi-Varga (GraphAware):
Graph Technology Landscape 2019

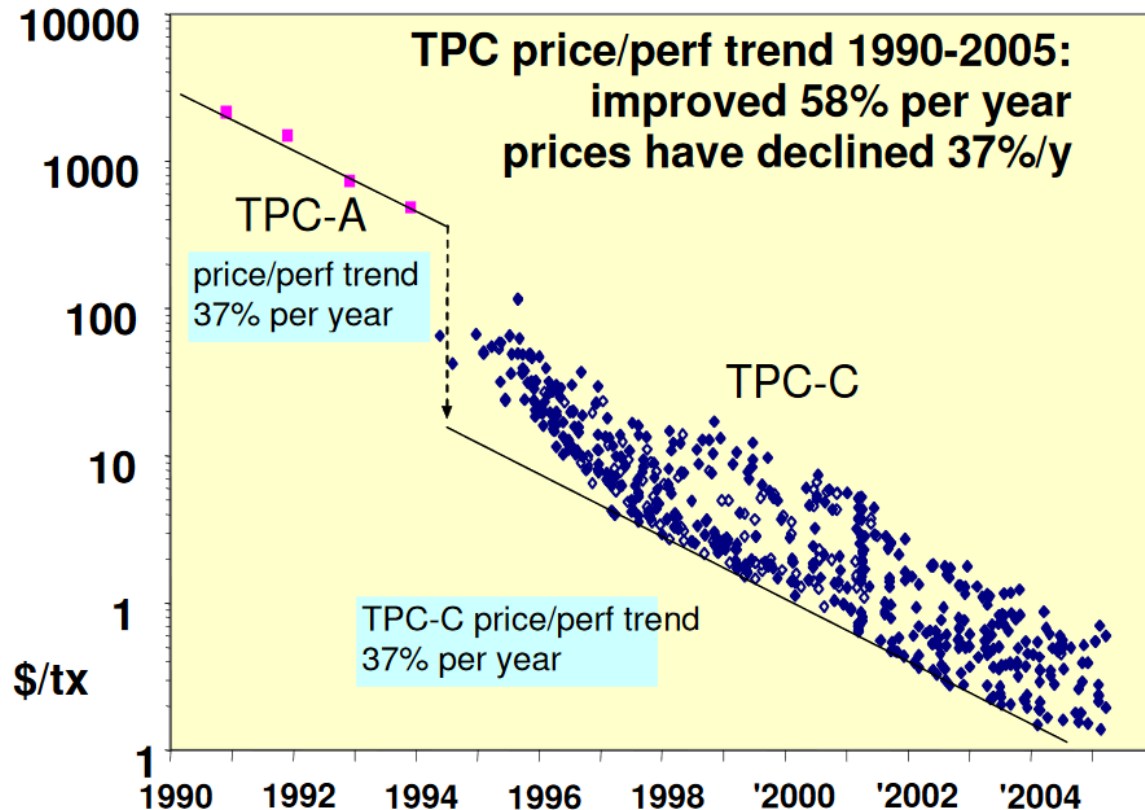


 Neo4j Graph Algorithms library

Benchmarks: Defining a common understanding

TRANSACTION PROCESSING PERFORMANCE COUNCIL (1988-)

Many standard specifications for benchmarking certain aspects of relational DBs



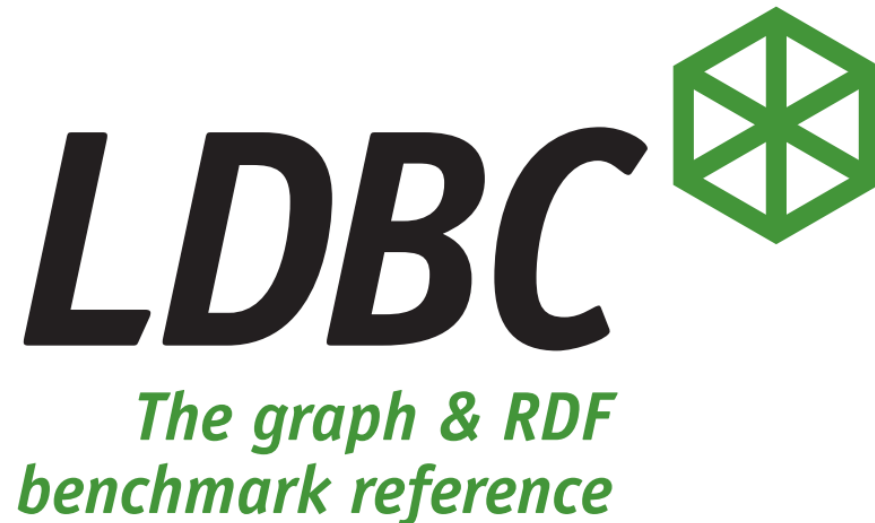
Transaction Processing - OLTP	TPC-C		
	TPC-E		
Decision Support	TPC-H	Top Ten	By Perf.
	TPC-DS		
	TPC-DI	Most Recent Ten Published	By Price/Perf.
Virtualization	TPC-VMS	All Results	
	TPCx-V	Advanced Sort	
	TPCx-HCI		
Big Data	TPCx-HS V1		
	TPCx-HS V2		
	TPCx-BB		
IoT	TPCx-IoT		
Common Specifications	TPC-Energy		
	TPC-Pricing		

TPC™

LINKED DATA BENCHMARK COUNCIL (2012-)

LDBC is a non-profit organization dedicated to establishing benchmarks, benchmark practices and benchmark results for graph data management software.

LDBC's Social Network Benchmark is an industrial and academic initiative, formed by principal actors in the field of graph-like data management.



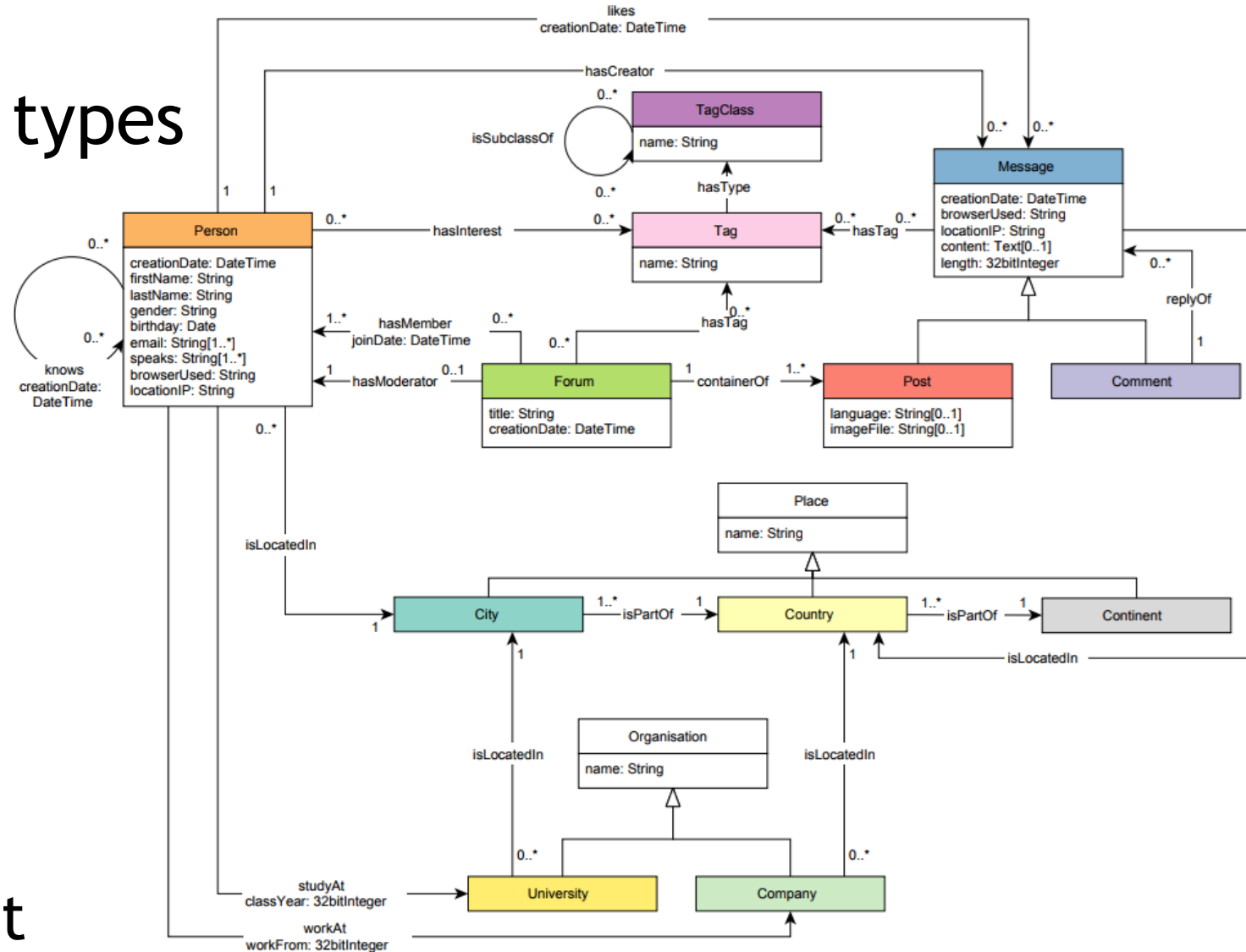
LDBC SOCIAL NETWORK BENCHMARK

Complex graph schema

14 node types, many edge types

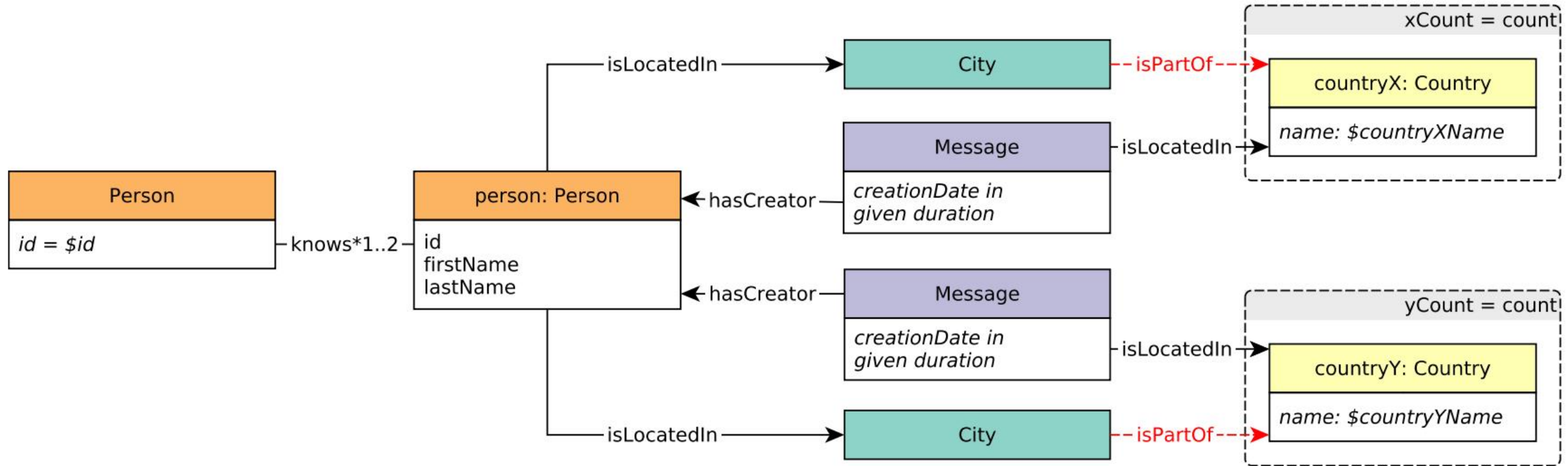
Subgraphs

- Network of persons
 - Comments
 - TagClasses
- Fixed depth trees
 - City < Country < Continent



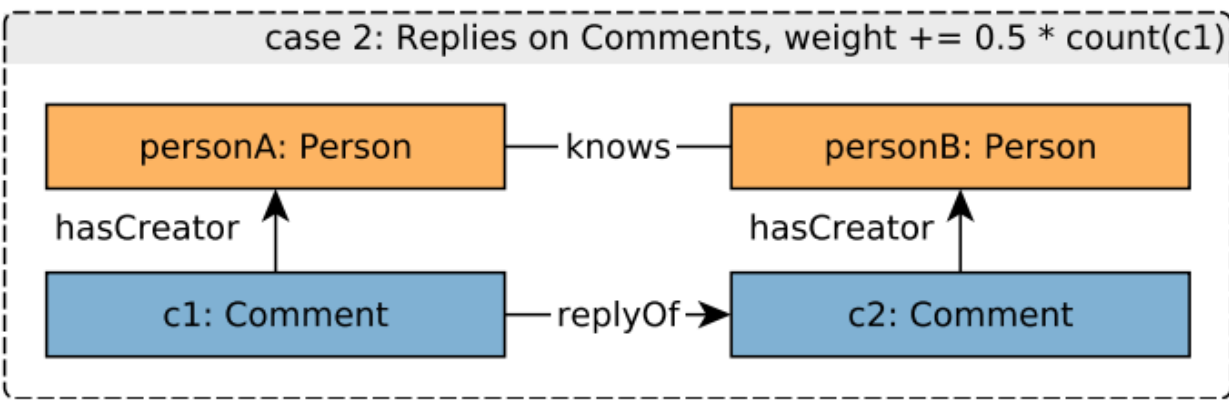
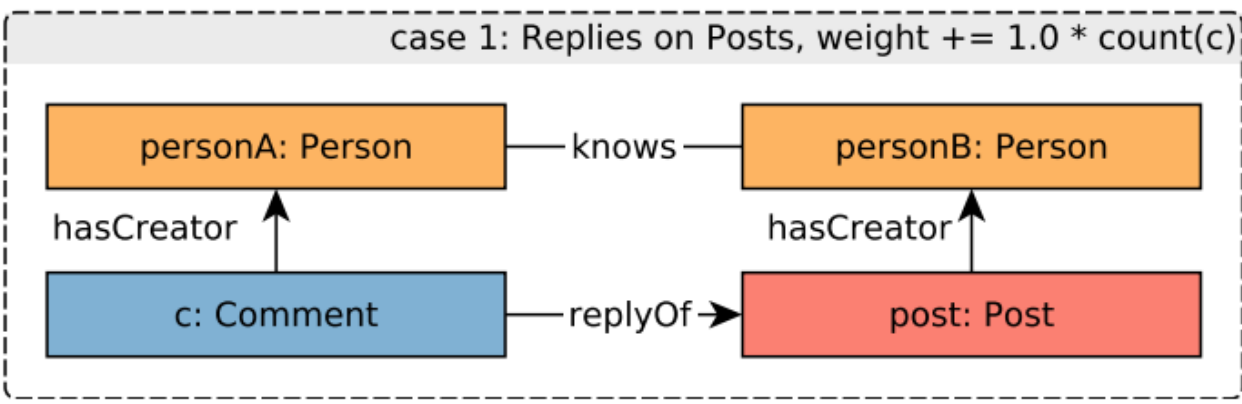
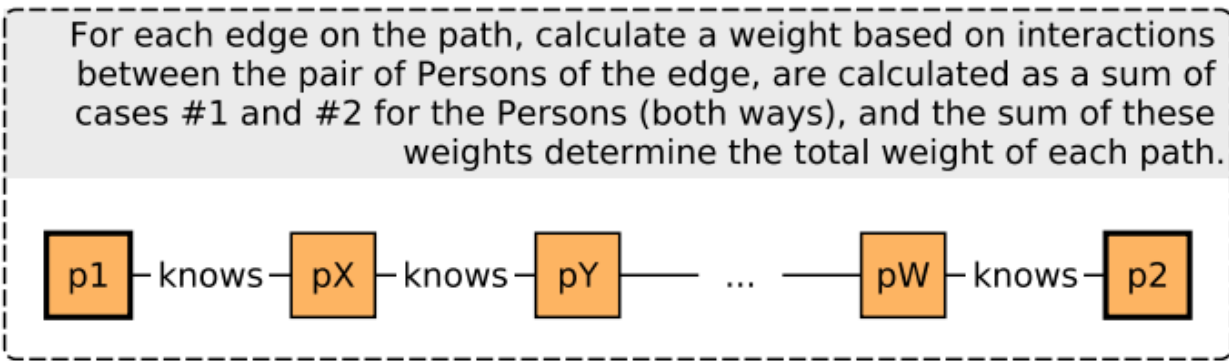
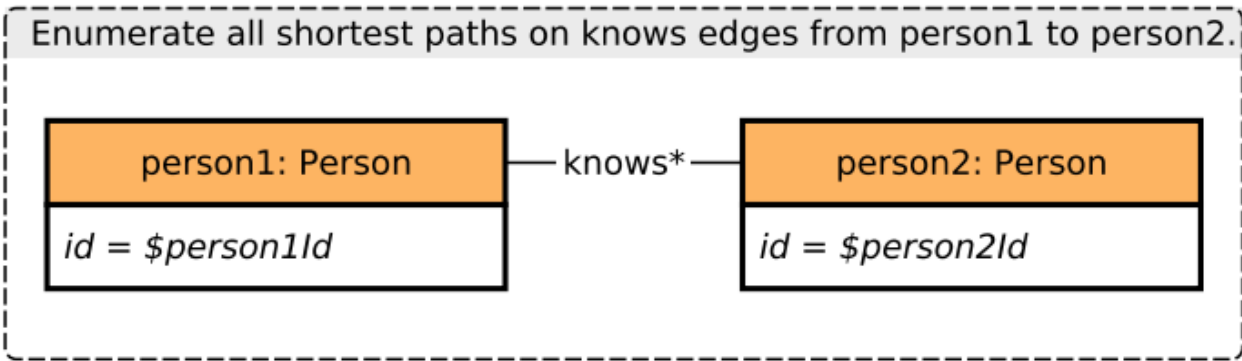
LDBC INTERACTIVE Q3

Friends and friends of friends that have been to countries X and Y



LDBC INTERACTIVE Q14

Trusted connection paths



[illegible][illegible][illegible]

The location of a feature is identified by the location of the feature's node in the tree. The following table shows the location of the feature's node in the tree.

feature	node	location
name	name	1
age	age	2
sex	sex	3
height	height	4
weight	weight	5
hair	hair	6
eyes	eyes	7
skin	skin	8
teeth	teeth	9
lips	lips	10
nose	nose	11
ears	ears	12
mouth	mouth	13
neck	neck	14
head	head	15
body	body	16
limbs	limbs	17
feet	feet	18
hands	hands	19
face	face	20
features	features	21

[illegible]

- Count messages (messageCount).
- Count Bytes (totalBytes) for all messages.
- Count Characters (totalChars) only for text messages.

The scores is calculated according to the following formula: $score = messageCount + 2 * totalBytes + 3 * totalChars$.

params

	type
messageCount	32-bit integer
totalBytes	32-bit integer
totalChars	32-bit integer
messageCount	32-bit integer
score	32-bit integer

result

score	32-bit integer
-------	----------------

sort

score	totalBytes
1.2	1.2

list


CPUs 1, 2, 3, 4, 2

Given a `log` field of `Person` (assumed that user created a `Message` instance) with the given `log`. For each of those `Person` records compute their “authority score” as follows:

- The “authority score” is the ratio of “popularity score” of the `Person` (assumed that they are the `Post`’s “Message instance”) with the given `log`.
- A `Person`’s “popularity score” is defined as the total number of likes on all their `Message` (instance).

person

	yes	string
name	person-01	66-01-2018
	person-02	66-02-2018
age	person-03	66-03-2018
	person-04	66-04-2018
	person-05	66-05-2018
end	100	
CPU	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	

[illegible][illegible]

Chapter 3. Business Intelligence Worksheet	5.5. SQL Query Basics
SQL Fundamentals	
memory	SQL Fundamentals
file	Control Program for a Tag
100	
101	
102	
103	
104	
105	
106	
107	
108	
109	
110	
111	
112	
113	
114	
115	
116	
117	
118	
119	
120	
121	
122	
123	
124	
125	
126	
127	
128	
129	
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	
143	
144	
145	
146	
147	
148	
149	
150	
151	
152	
153	
154	
155	
156	
157	
158	
159	
160	
161	
162	
163	
164	
165	
166	
167	
168	
169	
170	
171	
172	
173	
174	
175	
176	
177	
178	
179	
180	
181	
182	
183	
184	
185	
186	
187	
188	
189	
190	
191	
192	
193	
194	
195	
196	
197	
198	
199	
200	
201	
202	
203	
204	
205	
206	
207	
208	
209	
210	
211	
212	
213	
214	
215	
216	
217	
218	
219	
220	
221	
222	
223	
224	
225	
226	
227	
228	
229	
230	
231	
232	
233	
234	
235	
236	
237	
238	
239	
240	
241	
242	
243	
244	
245	
246	
247	
248	
249	
250	
251	
252	
253	
254	
255	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
272	
273	
274	
275	
276	
277	
278	
279	
280	
281	
282	
283	
284	
285	
286	
287	
288	
289	
290	
291	
292	
293	
294	
295	
296	
297	
298	
299	
300	
301	
302	
303	
304	
305	
306	
307	
308	
309	
310	
311	
312	
313	
314	
315	
316	
317	
318	
319	

[illegible]

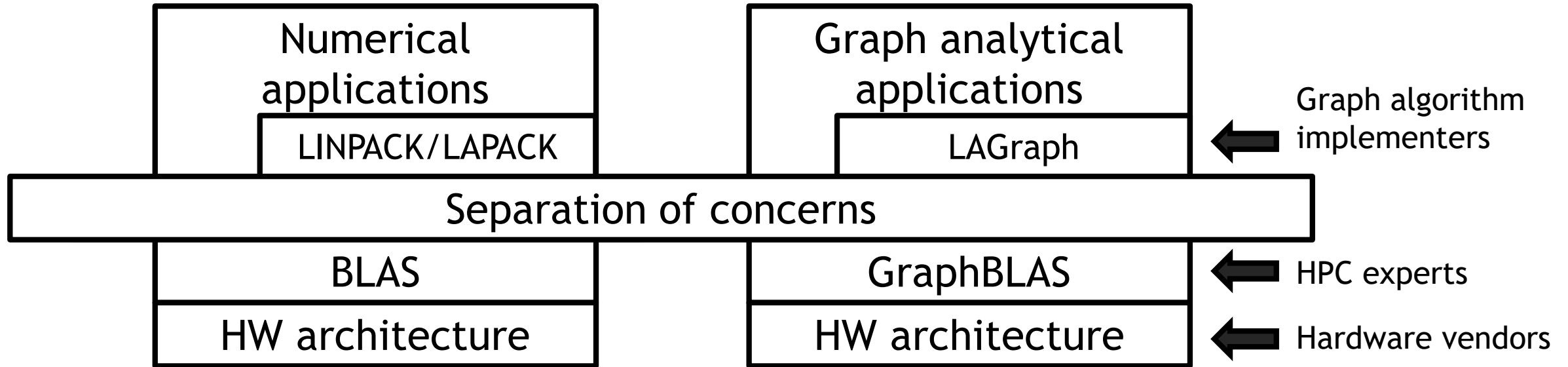
Chapter 3. Business Process Workflow		5.1. Read Query												
RF / read / F2														
query	RF / read / F2													
title	Reading Process													
	<pre> graph TD start([start]) --> read([read]) read --> process([process]) process --> readAgain([read again]) process --> end([end]) readAgain --> read cancel([cancel]) --> end </pre>													
patterns	Find all Message created after a given one (includes, that created once that a given one is finished)													
parameters	<table border="1"> <thead> <tr> <th>name</th><th>type</th></tr> </thead> <tbody> <tr> <td>messageId</td><td>string</td></tr> </tbody> </table>	name	type	messageId	string									
name	type													
messageId	string													
result	<table border="1"> <thead> <tr> <th>name</th><th>type</th><th>description</th></tr> </thead> <tbody> <tr> <td>messageId</td><td>string</td><td>The first name of the Paul's created</td></tr> <tr> <td>messageCreatedTime</td><td>String</td><td>The last name of the Paul's created</td></tr> <tr> <td>messageLastTime</td><td>String</td><td>The number of days that was created</td></tr> </tbody> </table>	name	type	description	messageId	string	The first name of the Paul's created	messageCreatedTime	String	The last name of the Paul's created	messageLastTime	String	The number of days that was created	
name	type	description												
messageId	string	The first name of the Paul's created												
messageCreatedTime	String	The last name of the Paul's created												
messageLastTime	String	The number of days that was created												
api	<table border="1"> <thead> <tr> <th>name</th><th>type</th></tr> </thead> <tbody> <tr> <td>message</td><td>string</td></tr> </tbody> </table>	name	type	message	string									
name	type													
message	string													
GUI	1, 2, 3, 5, 6, 8, 9													

[illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible]

GraphBLAS: **A unified theory built on linear algebra**

THE GRAPHBLAS APPROACH

- GraphBLAS is an effort to define standard building blocks for graph algorithms in the language of linear algebra
- 1979: BLAS (Basic Linear Algebra Subprograms)
- 2013: GraphBLAS
- Key idea: *separation of concerns*



S. McMillan: Research review @ CMU, 2015
Graph algorithms on future architectures



Tim Mattson et al.: LAGraph,
GrAPL @ IPDPS 2019

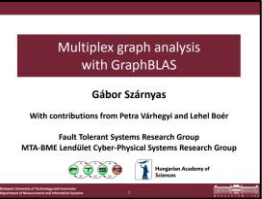
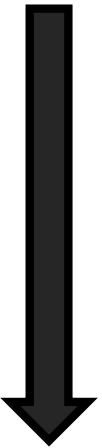
PARALLELIZATION ON SKEWED DISTRIBUTIONS

Using multiple processing units require load balancing.

Very difficult to implement for real graphs.

This work is in progress and improvements are expected.

Dataset	Algorithm	Single thread	Multi thread	Ratio
graph500-22	LCC	3083.494	261.193	11.80
datagen-7_9-fb	LCC	938.139	213.783	4.38
datagen-7_6-fb	LCC	431.211	101.173	4.26
datagen-7_5-fb	LCC	336.583	80.637	4.17
datagen-7_7-zf	LCC	183.948	138.022	1.33
datagen-7_8-zf	LCC	234.518	176.558	1.32



Gábor Szárnyas: *Multiplex graph analytics with GraphBLAS*, FOSDEM 2019



Bálint Hegyi: *Benchmarking scalable graph query techniques*, Master's thesis, 2019

Summary

SUMMARY: CHALLENGES IN GRAPH PROCESSING

No consensus on a unifying theory:

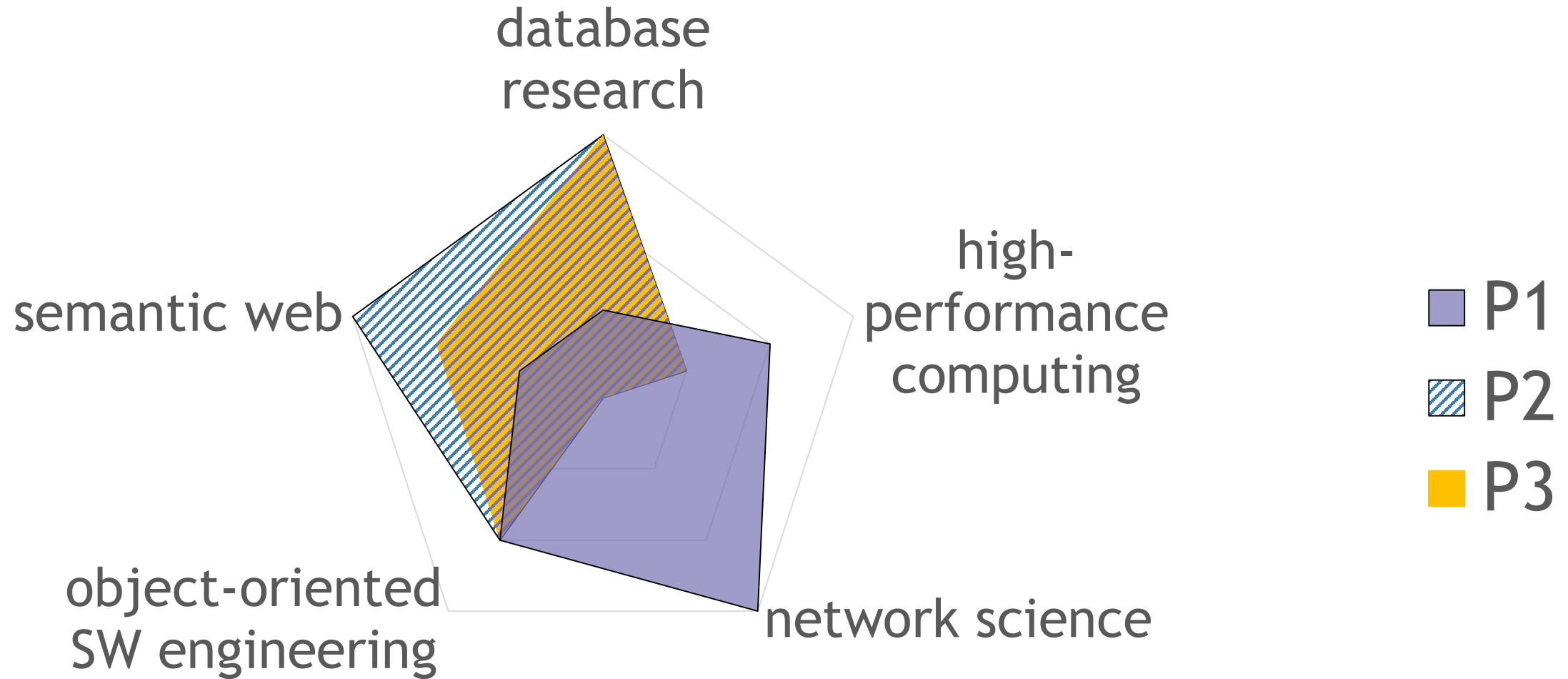
- Relational algebra?
- Linear algebra?

Performance:

- Many random access operations
- Difficult to cache
- Difficult to parallalize
- Handling propertes introduces even more complexity

Many open research and implementation challenges.

CONTRIBUTIONS IN MY PHD DISSERTATION



Gábor Szárnyas:

Query, Analysis, and Benchmarking Techniques for Evolving Property Graphs of Software Systems,
PhD dissertation, 2019