The LDBC Social Network Benchmark

Interactive Workload

http://www.ldbcouncil.org

Orri Erling
OpenLink Software, UK
oerling@openlinksw.com

Alex Averbuch
Neo Technology, Sweden
alex.averbuch@neotechnology.com

Josep Larriba-Pey
Sparsity Technologies, Spain
larri@sparsity-technologies.com

Hassan Chafi
Oracle Labs, USA
hassan.chafi@oracle.com

Andrey Gubichev
TU Munich, Germany
gubichev@in.tum.de

Arnau Prat*
Universitat Politècnica de Catalunya, Spain
aprat@ac.upc.edu

Minh-Duc Pham
VU University Amsterdam, The Netherlands
m.d.pham@vu.nl

Peter Boncz
CWI, Amsterdam, The Netherlands
boncz@cwi.nl

The LDBC Social Network Benchmark

Interactive Workload

http://www.ldbcouncil.org

Orri Erling
OpenLink Software, UK
oerling@openlinksw.com

Alex Averbuch
Neo Technology, Sweden
alex.averbuch@neotechnology.com

Josep Larriba-Pey
Sparsity Technologies, Spain
larri@sparsity-technologies.com

Hassan Chafi
Oracle Labs, USA
hassan.chafi@oracle.com

Andrey Gubichev
TU Munich, Germany
gubichev@in.tum.de

Arnau Prat*
Universitat Politècnica de Catalunya, Spain
aprat@ac.upc.edu

Minh-Duc Pham
VU University Amsterdam, The Netherlands
m.d.pham@vu.nl

Peter Boncz
CWI, Amsterdam, The Netherlands
boncz@cwi.nl

The LDBC Social Network Benchmark

Interactive Workload

http://www.ldbcouncil.org

Orri Erling
OpenLink Software, UK
oerling@openlinksw.com

Alex Averbuch
Neo Technology, Sweden
alex.averbuch@neotechnology.com

Josep Larriba-Pey
Sparsity Technologies, Spain
larri@sparsity-technologies.com

Hassan Chafi
Oracle Labs, USA
hassan.chafi@oracle.com

Andrey Gubichev
TU Munich, Germany
gubichev@in.tum.de

Arnau Prat*
Universitat Politècnica de Catalunya, Spain
aprat@ac.upc.edu

Minh-Duc Pham
VU University Amsterdam, The Netherlands
m.d.pham@vu.nl

Peter Boncz
CWI, Amsterdam, The Netherlands
boncz@cwi.nl

The LDBC Social Network Benchmark

Interactive Workload

The graph & RDF benchmark reference

http://www.ldbcouncil.org

Orri Erling
OpenLink Software, UK
oerling@openlinksw.com

Alex Averbuch
Neo Technology, Sweden
alex.averbuch@neotechnology.com

Josep Larriba-Pey
Sparsity Technologies, Spain
larri@sparsity-technologies.com

Hassan Chafi
Oracle Labs, USA
hassan.chafi@oracle.com

Andrey Gubichev
TU Munich, Germany
gubichev@in.tum.de

Arnau Prat*
Universitat Politècnica de Catalunya, Spain
aprat@ac.upc.edu

Minh-Duc Pham
VU University Amsterdam, The Netherlands
m.d.pham@vu.nl

Peter Boncz
CWI, Amsterdam, The Netherlands
boncz@cwi.nl

The LDBC Social Network Benchmark

Interactive Workload

http://www.ldbcouncil.org

Orri Erling
OpenLink Software, UK
oerling@openlinksw.com

Alex Averbuch
Neo Technology, Sweden
alex.averbuch@neotechnology.com

Josep Larriba-Pey
Sparsity Technologies, Spain
larri@sparsity-technologies.com

Hassan Chafi
Oracle Labs, USA
hassan.chafi@oracle.com

Andrey Gubichev
TU Munich, Germany
gubichev@in.tum.de

Arnau Prat*
Universitat Politècnica de Catalunya, Spain
aprat@ac.upc.edu

Minh-Duc Pham
VU University Amsterdam, The Netherlands
m.d.pham@vu.nl

Peter Boncz
CWI, Amsterdam, The Netherlands
boncz@cwi.nl

The LDBC Social Network Benchmark

Interactive Workload

http://www.ldbcouncil.org

Orri Erling
OpenLink Software, UK
oerling@openlinksw.com

Alex Averbuch
Neo Technology, Sweden
alex.averbuch@neotechnology.com

Josep Larriba-Pey
Sparsity Technologies, Spain
larri@sparsity-technologies.com

Norbert Martinez

Hassan Chafi
Oracle Labs, USA
hassan.chafi@oracle.com

Andrey Gubichev
TU Munich, Germany
gubichev@in.tum.de

Arnau Prat
Universitat Politècnica de Catalunya, Spain
aprat@ac.upc.edu

Thomas Neumann, Linnea Passing

David Dominguez

Xavier Sanchez

Minh-Duc Pham
VU University Amsterdam, The Netherlands
m.d.pham@vu.nl

Peter Boncz
CWI, Amsterdam, The Netherlands
boncz@cwi.nl

Renzo Angles (U. Talca)

SNB "Task Force" acknowledgements

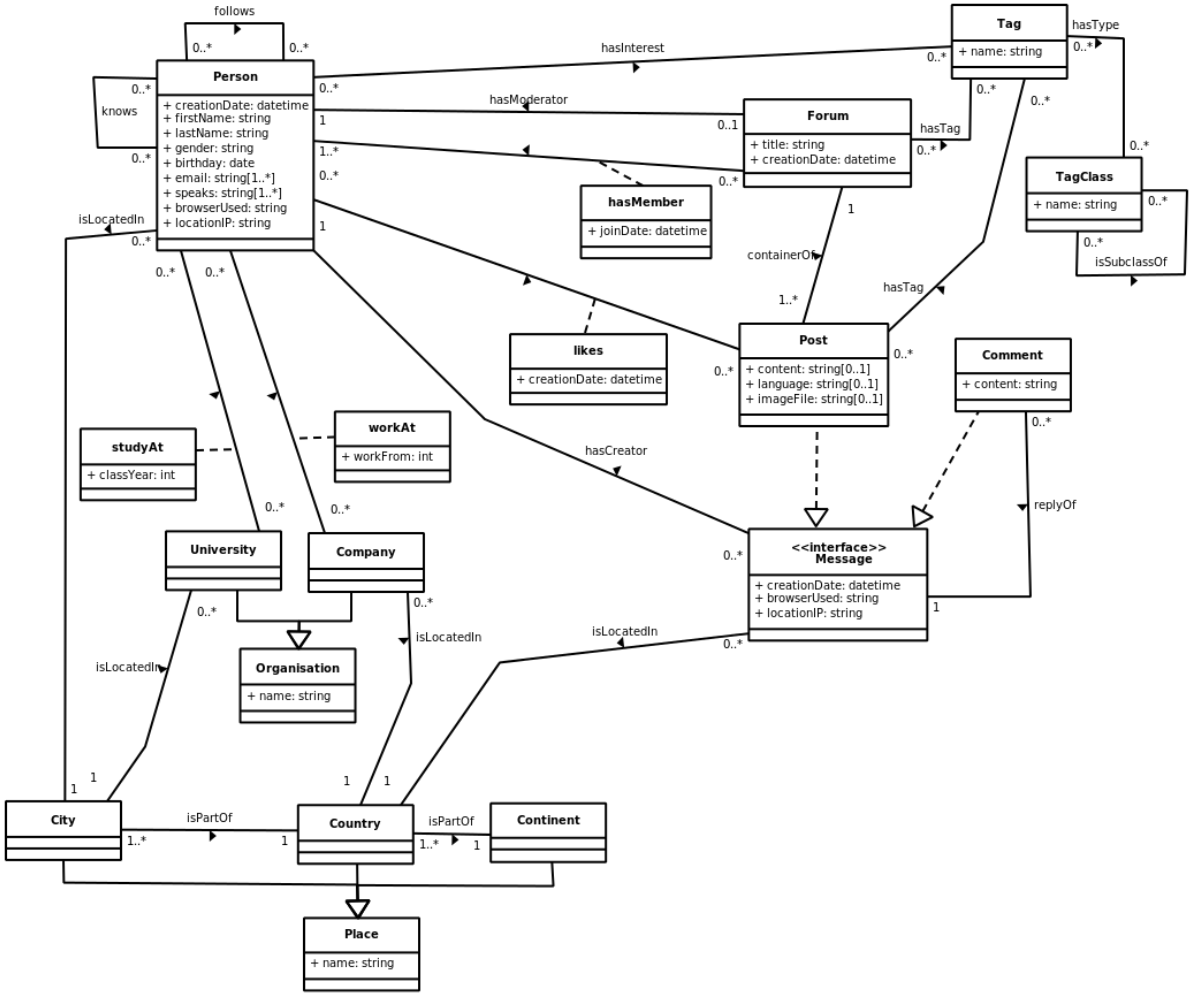
LDBC Organization (non-profit)

"sponsors"

+ non-profit members (FORTH, STI2) & personal members
+ **Task Forces**, volunteers developing benchmarks
+ **TUC**: Technical User Community (6 workshops, 36 graph and RDF user case studies, 12 vendor presentations)

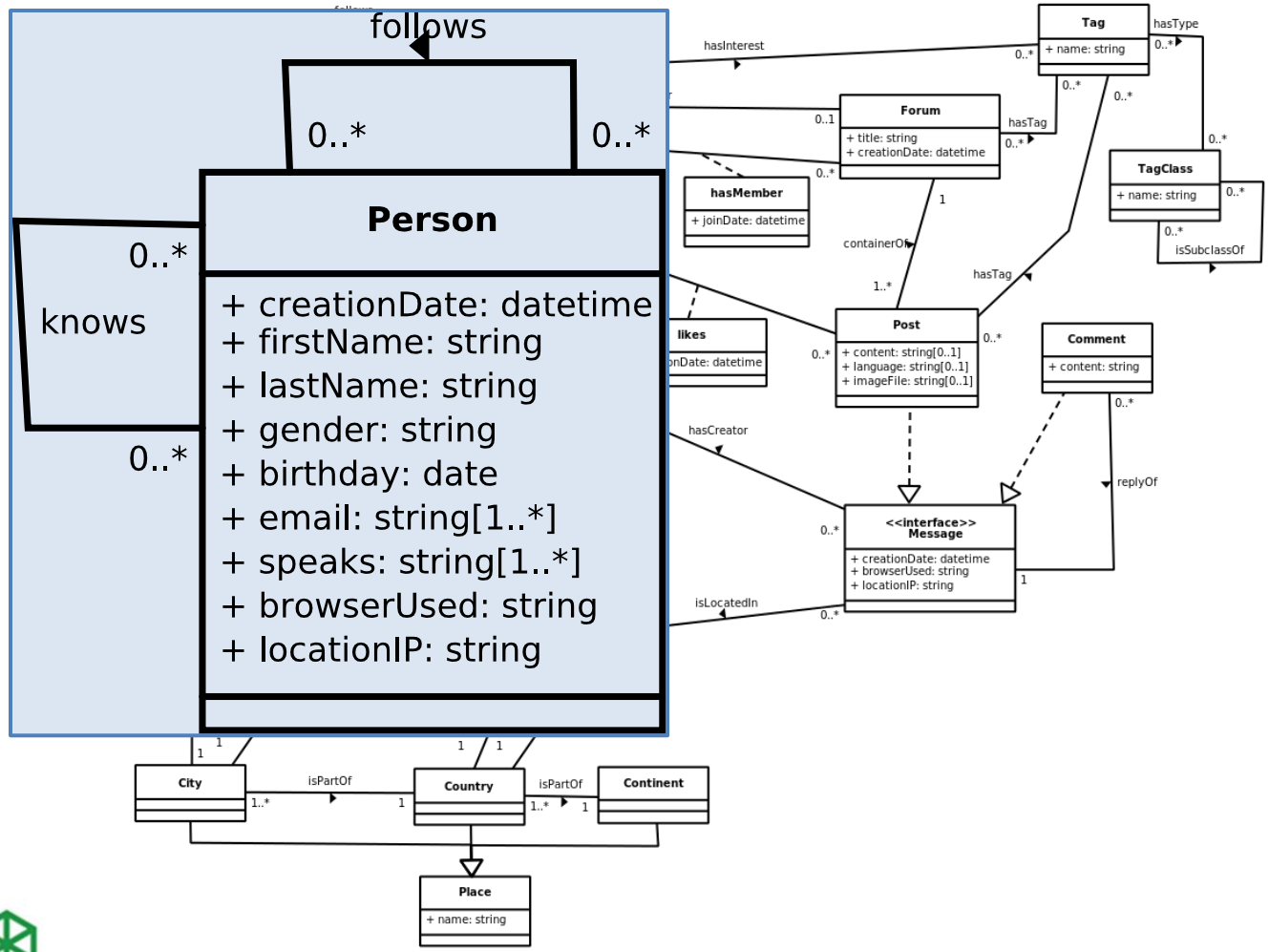
LDBC

Social Network Benchmark: schema



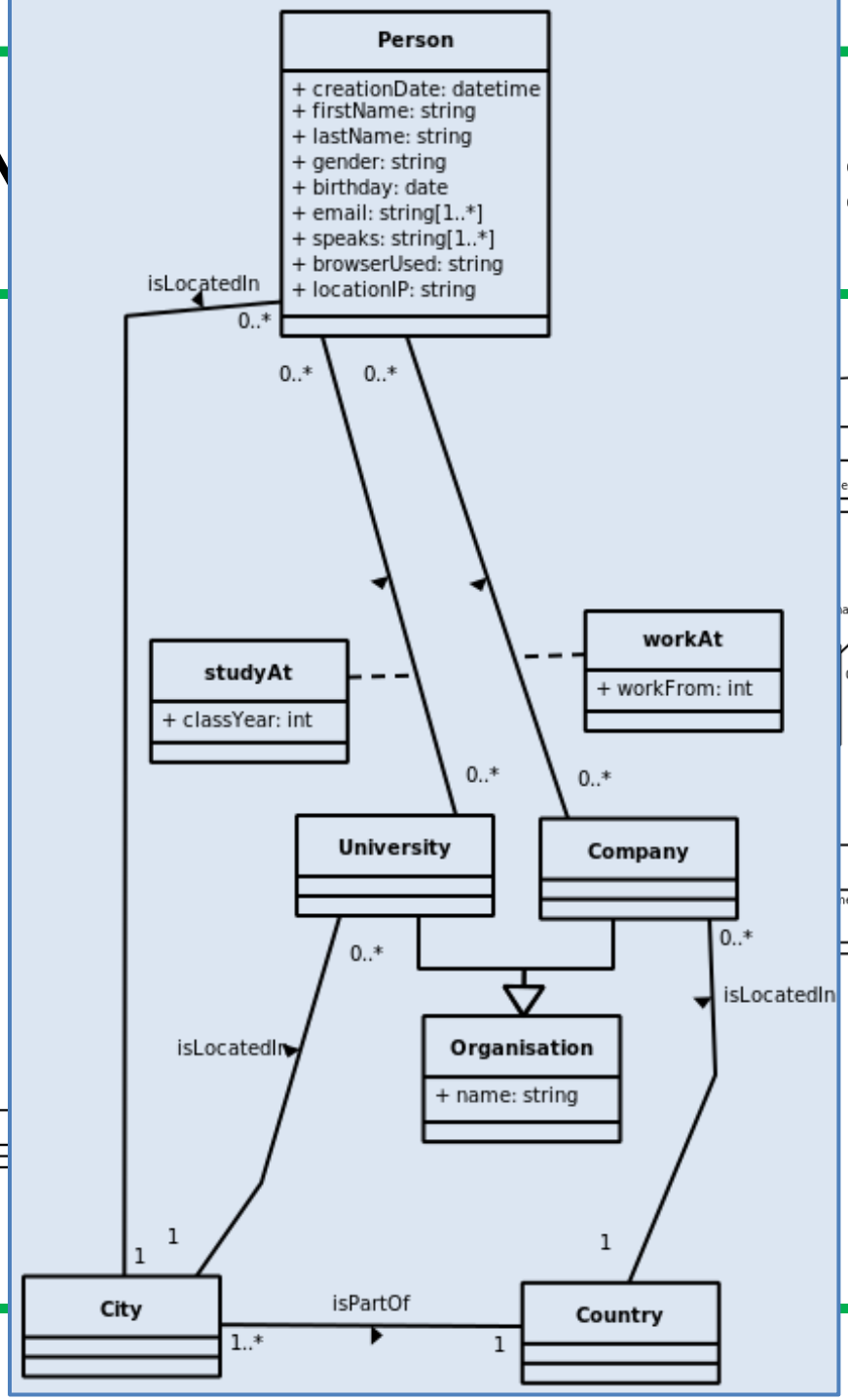
LDBC

Social Network Benchmark: schema

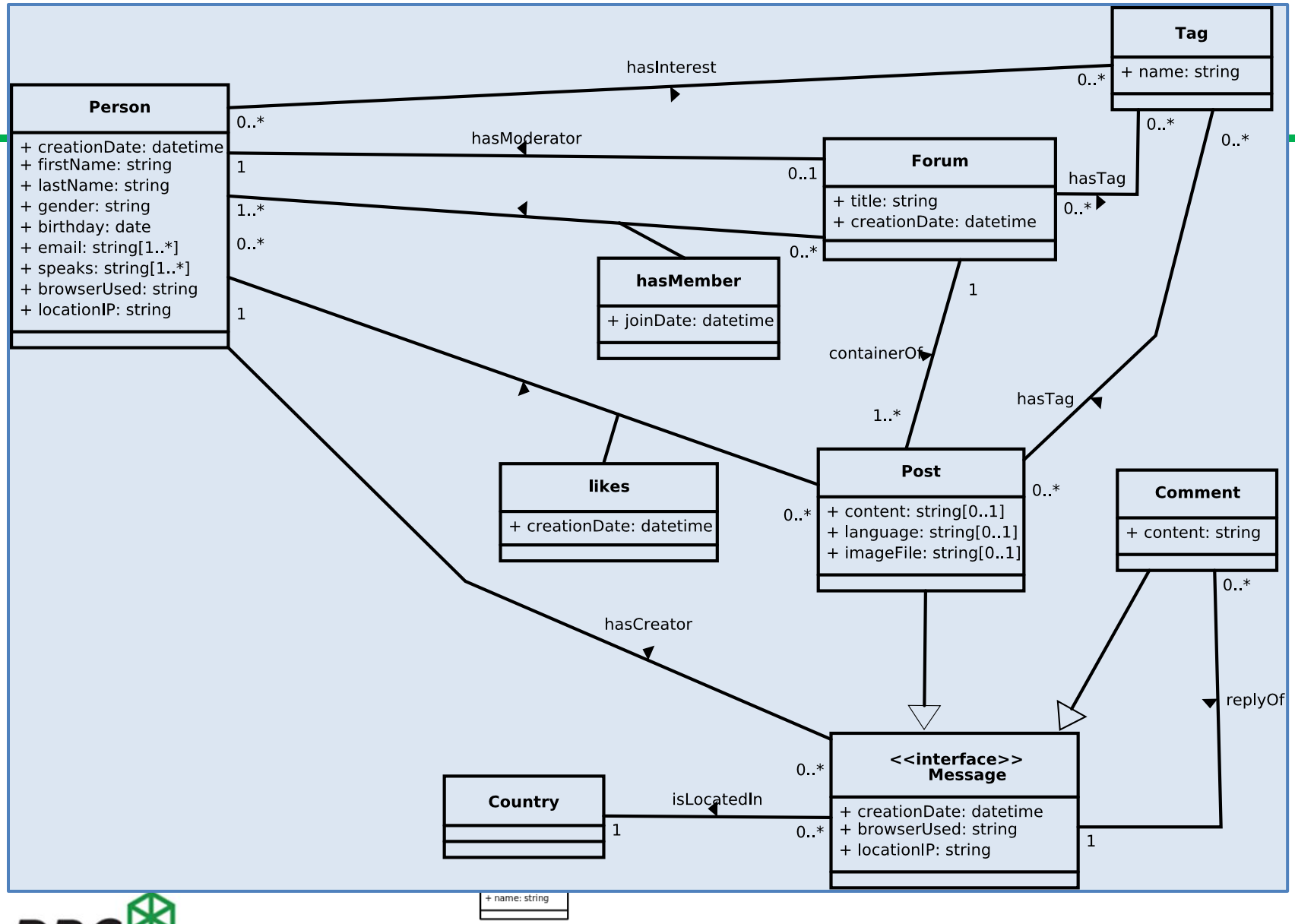


LDBC

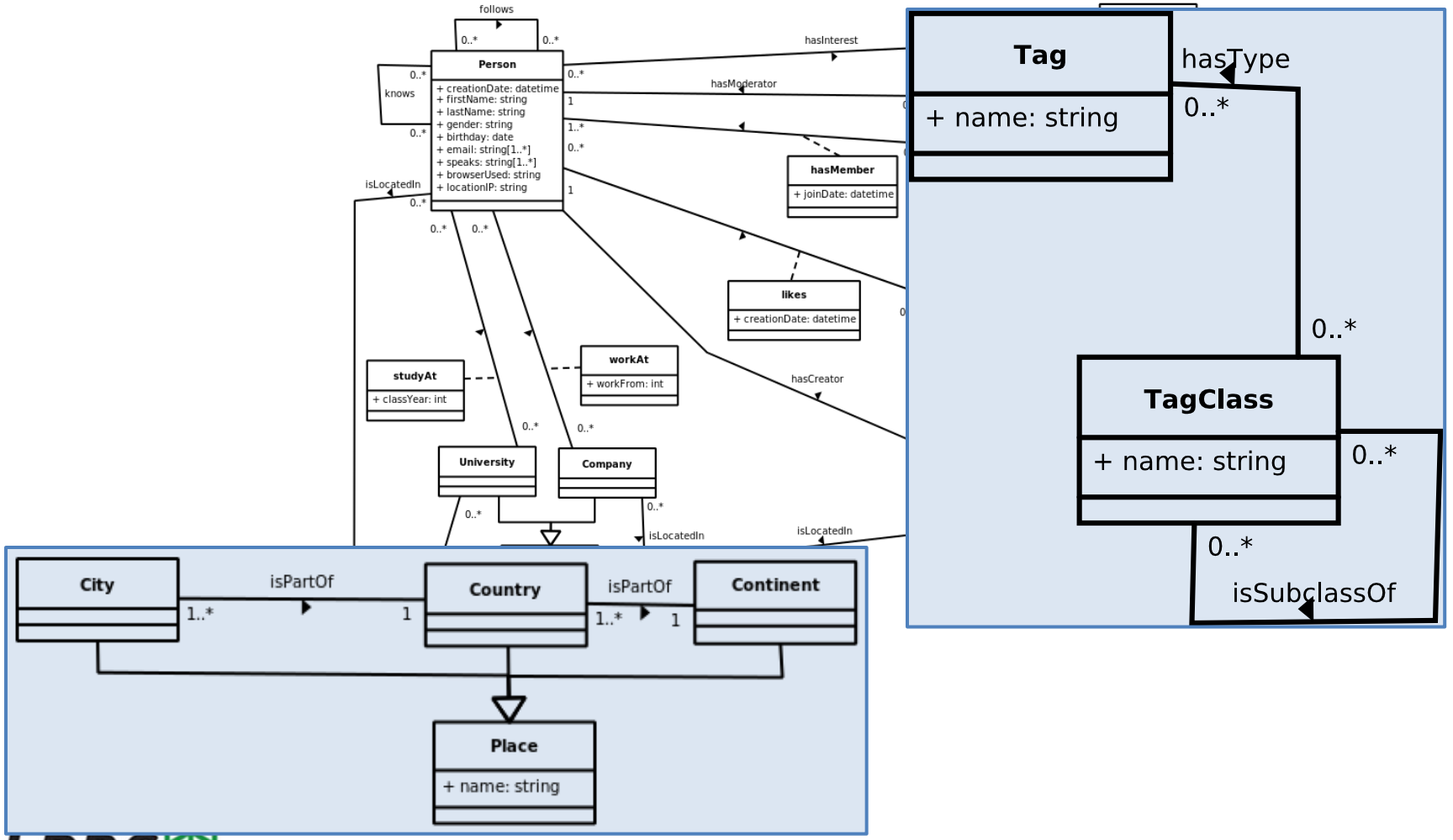
Social N : schema



LDBC


Tag
+ name: string
hasInterest
0..*
Person
+ creationDate: datetime
+ firstName: string
+ lastName: string
+ gender: string
+ birthday: date
+ email: string[1..*]
+ speaks: string[1..*]
+ browserUsed: string
+ locationIP: string
hasModerator
1
0..1
Forum
+ title: string
+ creationDate: datetime
hasTag
0..*
1..*
0..*
hasMember
+ joinDate: datetime
containerOf
1
1..*
hasTag
likes
+ creationDate: datetime
Post
+ content: string[0..1]
+ language: string[0..1]
+ imageFile: string[0..1]
Comment
+ content: string
hasCreator
replyOf
<<interface>>
Message
+ creationDate: datetime
+ browserUsed: string
+ locationIP: string
Country
isLocatedIn
+ name: string

Social Network Benchmark: schema



LDBC

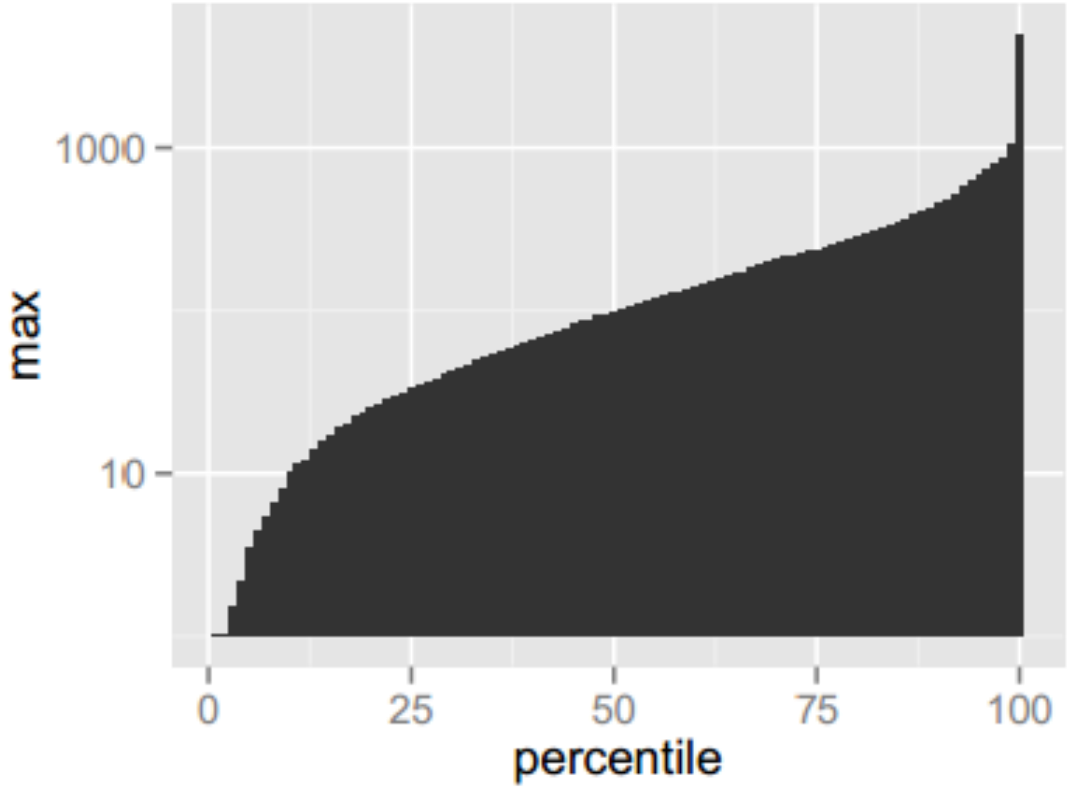
DATAGEN: social network generator

advanced generation of:

- network structure
  - **Power law** distributions, small diameter

Friendship Degree Distribution

- Based on “Anatomy of Facebook” blogpost (2013)
- Diameter increases logarithmically with scale factor
  - New:
    function has been
    made pluggable



LDBC

DATAGEN: social network generator

advanced generation of:

- network structure
  - Power law distributions, small diameter
- property values
  - **realistic**, **correlated** value distributions

LDBC

Realistic Correlated Value Distributions

- Person.firstname correlates with Person.location
  - Values taken from **DBpedia**
- Many other correlations and dependencies..
  - e.g. university depends on location

Person.location =<Germany>

Name	Number
Karl	215
Hans	190
Wolfgang	174
Fritz	159
Rudolf	159
Walter	150
Franz	115
Paul	109
Otto	99
Wilhelm	74

Person.location =<China>

Name	Number
Yang	961
Chen	929
Wei	887
Lei	789
Jun	779
Jie	778
Li	562
Hao	533
Lin	456
Peng	448

- In forum discussions, people read DBpedia articles to each other (= correlation between message text and discussion topic)
  - Topic = DBpedia article title
  - Text = one sentence of the article

LDBC

DATAGEN: social network generator

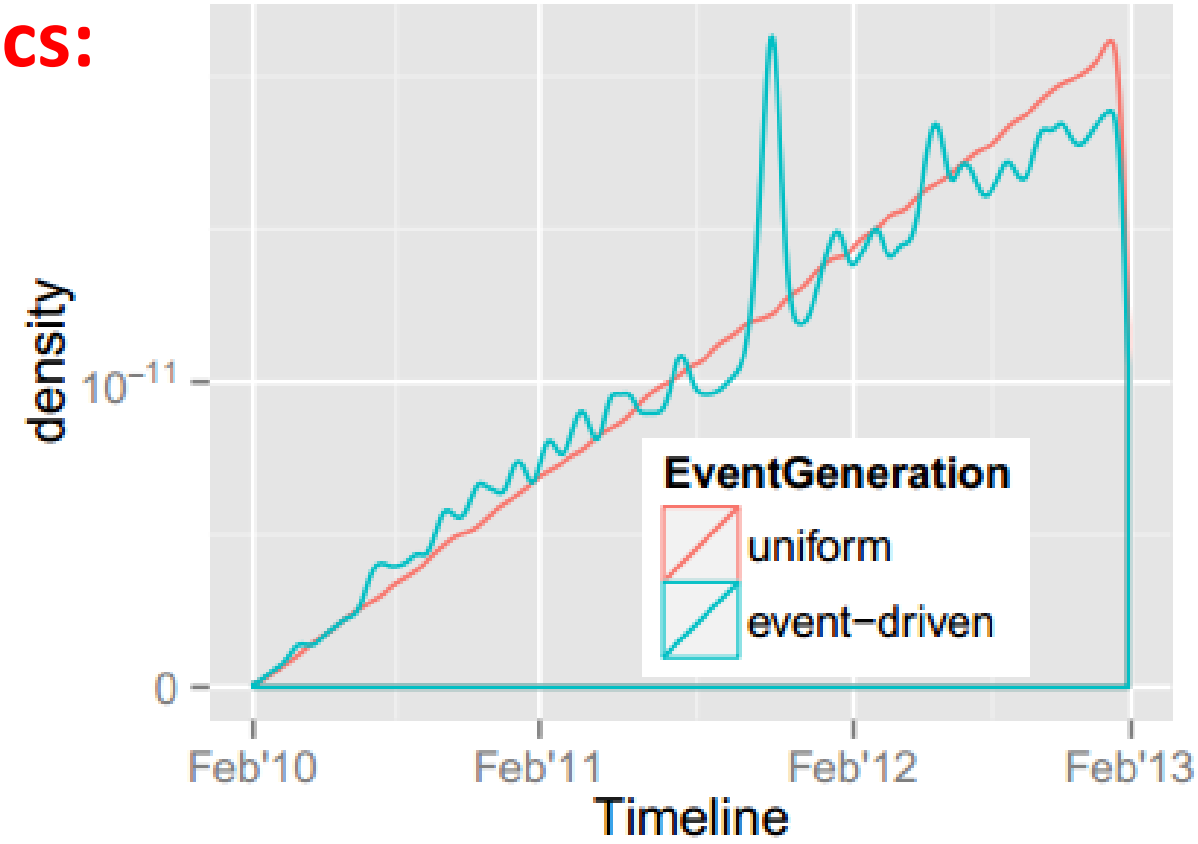
advanced generation of:

- network structure
  - Power law distributions, small diameter
- property values
  - realistic, correlated value distributions
  - **temporal correlations** / “flash mobs”

LDBC

Temporal Effects (Flash Mobs)

- Forum posts generation spikes in time **for certain topics:**



LDBC

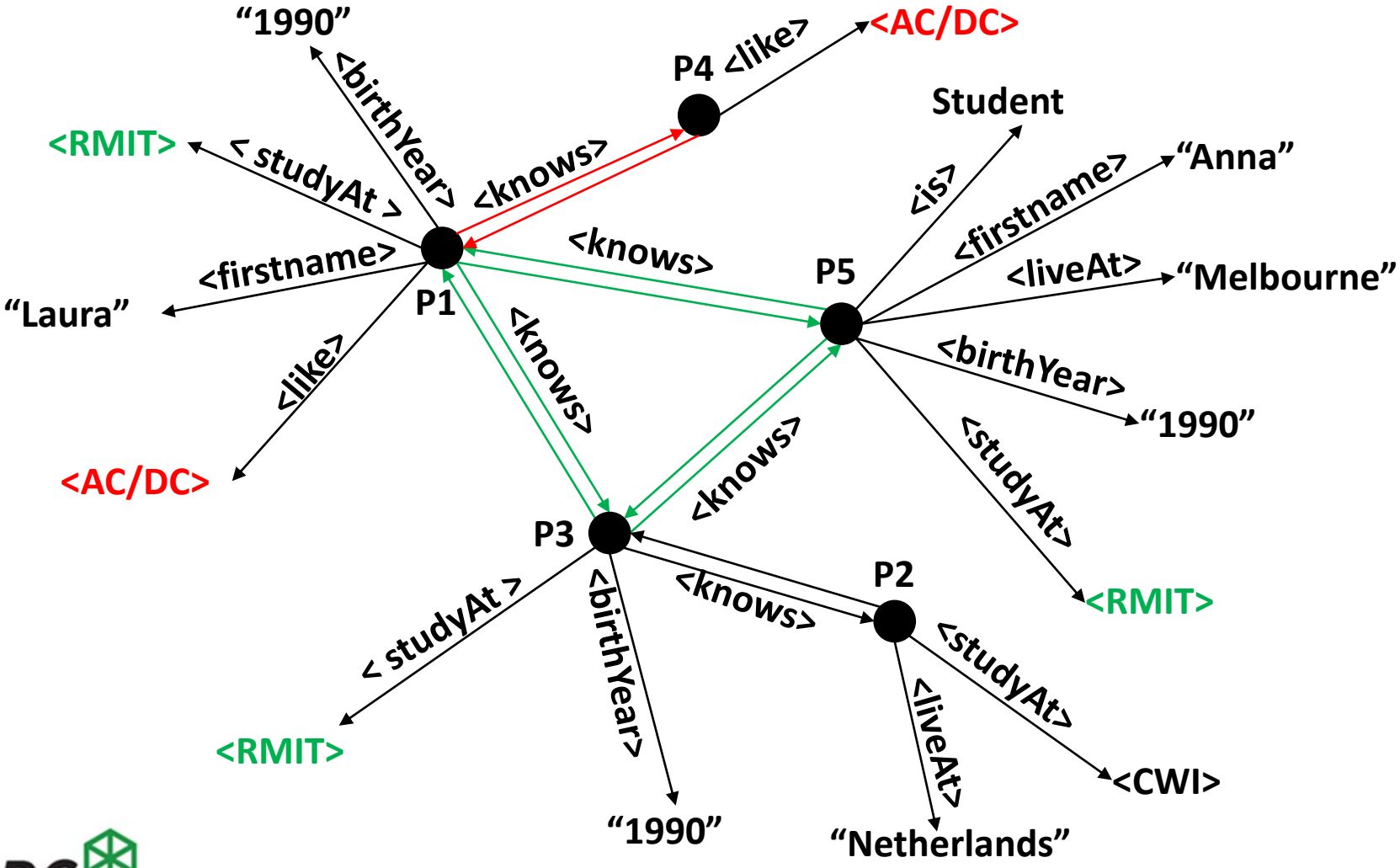
DATAGEN: social network generator

advanced generation of:

- network structure
  - Power law distributions, small diameter
- property values
  - realistic, correlated value distributions
  - temporal correlations / “flash mobs”
- **correlations between values and structure**
  - 2 correlation “dimensions”: location & interests

LDBC

“RMIT alumni stay in touch”
“Metalheads rock together”



LDBC

DATAGEN: Scaling

- Scale Factor (SF) is the size of the CSV input data in GB
- Some Virtuoso SQL stats at SF=30:

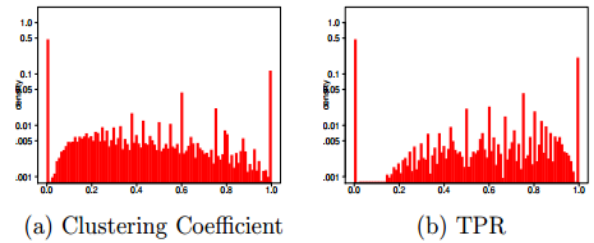
SFs	Number of entities (x 1000000)					
	Nodes	Edges	Persons	Friends	Messages	Forums
30	99.4	655.4	0.18	14.2	97.4	1.8
100	317.7	2154.9	0.50	46.6	312.1	5.0
300	907.6	6292.5	1.25	136.2	893.7	12.6
1000	2930.7	20704.6	3.60	447.2	2890.9	36.1

Table	**Size (MB)**	**Largest Index (MB)**
post	76815	ps_content (41697)
likes	23645	l_creationdate (11308)
forum_person	9343	fp_creationdate (5957)

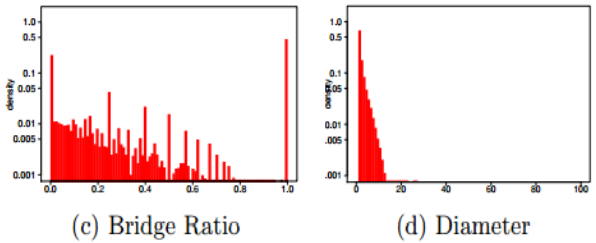
LDBC

DATAGEN: Graph Characteristics

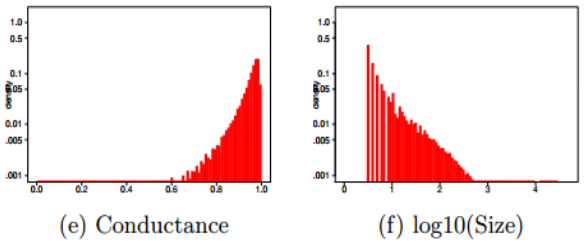
Livejournal



(a) Clustering Coefficient (b) TPR

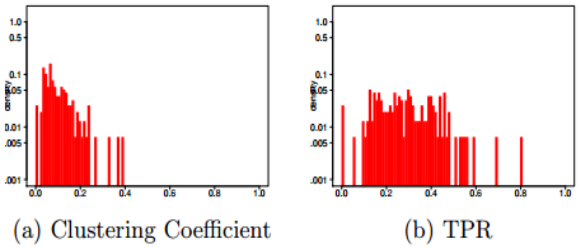


(c) Bridge Ratio (d) Diameter

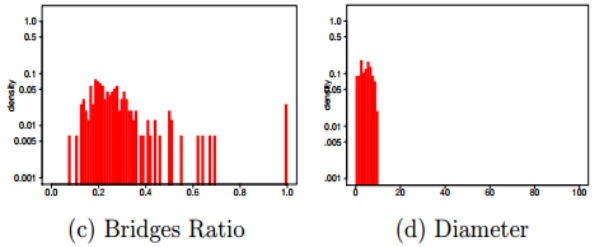


(e) Conductance (f) log10(Size)

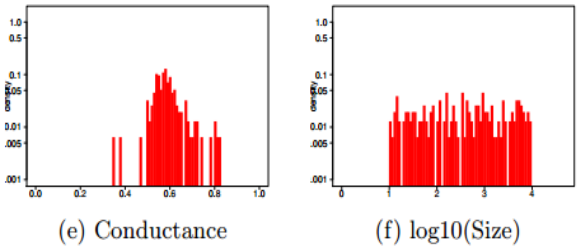
LFR3 (synthetic)



(a) Clustering Coefficient (b) TPR

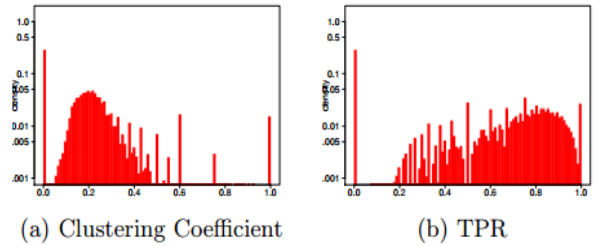


(c) Bridges Ratio (d) Diameter

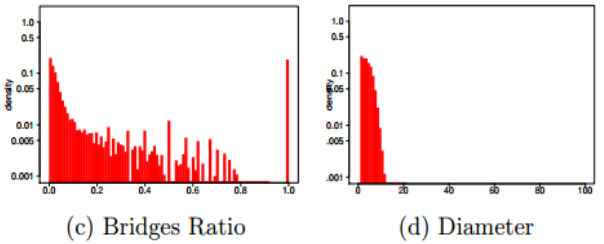


(e) Conductance (f) log10(Size)

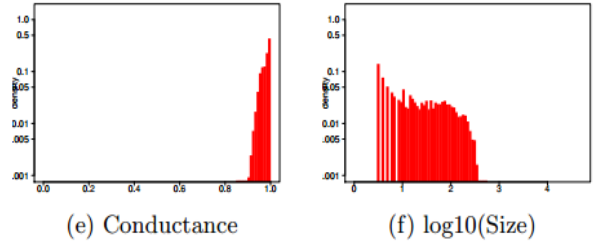
LDBC DATAGEN



(a) Clustering Coefficient (b) TPR



(c) Bridges Ratio (d) Diameter



(e) Conductance (f) log10(Size)

GRADES2014 “How community-like is the structure of synthetically generated graphs” - Arnau Prat(DAMA-UPC); David Domínguez-Sal (Sparsity Technologies)

LDBC

Benchmark Workloads

- **Interactive**: tests throughput running short queries while consistently handling concurrent updates
  - *Show all photos posted by my friends that I was tagged in*

Topic of this SIGMOD paper

- **Business Intelligence**: consists of complex structured queries for analyzing online behavior
  - *Influential people the topic of open source development?*

Draft queries available on ldbcouncil.org website (deliverable D2.2.4) & github

- **Graph Analytics**: tests the functionality and scalability on most of the data as a single operation
  - *PageRank, Shortest Path(s), Community Detection*

LDBC

GRADES2015 “Graphalytics: A Big Data Benchmark for Graph-Processing Platforms ” - Mihai Capota, Tim Hegeman, Alexandru Iosup(TU Delft); Arnau Prat (UPC), Orri Erling (OpenLink Technologies), Peter Boncz (CWI)

Workloads by system

System	Interactive	Business Intelligence	Graph Analytics
Graph databases	Yes	Yes	Maybe
Graph programming frameworks	-	Yes	Yes
RDF databases	Yes	Yes	-
Relational databases	Yes	Yes	Maybe, by keeping state in temporary tables, and using the functional features of PL-SQL
NoSQL Key-value	Maybe	Maybe	-
NoSQL MapReduce	-	Maybe	Yes

LDBC

Interactive Workload

MapReduce-base data generation

- Generate 3 years of network activity for a certain amount of persons
  - 33 months of data ➔ bulk load
  - 3 months of data ➔ insert queries
- Scalable (SF1000 in one hour on 10 small compute nodes)
  - can also be used without a cluster (pseudo-distributed)

During data generation, we perform Parameter Curation to derive suitable parameters for the complex-read-only query set

LDBC

Database Benchmark Design

Desirable properties:

- Relevant. ➔ "Choke Points"
- Representative.
- Understandable.
- Economical.
- Accepted.
- Scalable.
- Portable.
- Fair.
- Evolvable.
- Public.

Jim Gray (1991) *The Benchmark Handbook for Database and Transaction Processing Systems*

Dina Bitton, David J. DeWitt, Carolyn Turbyfill (1993) *Benchmarking Database Systems: A Systematic Approach*

Multiple TPCTC papers, e.g.
Karl Huppler (2009) *The Art of Building a Good Benchmark*

LDBC

Benchmark Design with Choke Points

Choke-Point = well-chosen difficulty in the workload

- "difficulties in the workloads"
  - arise from Data (distribs)+Query+Workload
  - there may be different technical solutions to address the choke point
    - or, there may not yet exist optimizations

➔lot's of research opportunities!

TPCTC 2013: Erling,Boncz,Neumann www.cwi.nl/~boncz ➔ Publications
"TPC-H Analyzed: Hidden Messages and Lessons Learned from an Influential Benchmark"

LDBC

SNB Interactive Workload: Complex Read Query Set

Q1. *Extract description of friends with a given name* Given a person's `firstName`, return up to 20 people with the same first name, sorted by increasing distance (max 3) from a given `person`, and for people within the same distance sorted by last name. Results should include the list of workplaces and places of study.

Q2. *Find the newest 20 posts and comments from your friends.* Given a start `Person`, find (most recent) Posts and Comments from all of that Person's friends, that were created before (and including) a given `Date`. Return the top 20 Posts/Comments, and the Person that created each of them. Sort results descending by creation date, and then ascending by Post identifier.

Q3. *Friends within 2 steps that have recently traveled to countries X and Y.* Find friends and friends of friends of a given `Person` who have made a post or a comment in the foreign `CountryX` and `CountryY` within a specified period of `DurationInDays` after a `startDate`. Return top 20 Persons, sorted descending by total number of posts.

Q4. *New Topics.* Given a start `Person`, find the top 10 most popular Tags (by total number of posts with the tag) that are attached to Posts that were created by that Person's friends. Only include Tags that were attached to Posts created within a given `time interval`, and that were never attached to Posts created before this interval.

Q5. *New groups.* Given a start `Person`, find the top 20 Forums which that Person's friends and friends of friends became members of after a given `Date`. Sort results descending by the number of Posts in each Forum that were created by any of these Persons.

Q6. *Tag co-occurrence.* Given a start `Person` and some `Tag`, find the other Tags that occur together with this Tag on Posts that were created by start Person's friends and friends of friends. Return top 10 Tags, sorted descending by the count of Posts that were created by these Persons, which contain both this Tag and the given Tag.

Q7. *Recent likes.* For the specified `Person` get the most recent likes of any of the person's posts, and the latency between the corresponding post and the like. Flag Likes from outside the direct connections. Return top 20 Likes, ordered descending by creation date of the like.

Q7. *Recent likes.* For the specified `Person` get the most recent likes of any of the person's posts, and the latency between the corresponding post and the like. Flag Likes from outside the direct connections. Return top 20 Likes, ordered descending by creation date of the like.

Q8. *Most recent replies.* This query retrieves the 20 most recent reply comments to all the posts and comments of `Person`, ordered descending by creation date.

Q9. *Latest Posts.* Find the most recent 20 posts and comments from all friends, or friends-of-friends of `Person`, but created before a `Date`. Return posts, their creators and creation dates, sort descending by creation date.

Q10. *Friend recommendation.* Find a friend of a friend who posts much about the interests of `Person` and little about topics that are not in the interests of the user. The search is restricted by the candidate's `horoscopeSign`. Returns 10 Persons for whom the difference between the total number of their posts about the interests of the specified user and the total number of their posts that are not in the interests of the user, is as large as possible. Sort the result descending by this difference.

Q11. *Job referral.* Find a friend of the specified `Person`, or a friend of her friend (excluding the specified person), who has long worked in a company in a specified `Country`. Sort ascending by start date, and then ascending by person identifier. Top 10 result should be shown.

Q12. *Expert Search.* Find friends of a `Person` who have replied the most to posts with a tag in a given `TagCategory`. Count the number of these reply Comments, and collect the Tags that were attached to the Posts they replied to. Return top 20 persons, sorted descending by number of replies.

Q13. *Single shortest path.* Given `PersonX` and `PersonY` , find the shortest path between them in the subgraph induced by the

Q14. *Weighted paths.* Given `PersonX` and `PersonY` , find all weighted paths of the shortest length between them in the subgraph induced by the Knows relationship. The weight of the path takes into consideration amount of Posts/Comments exchanged.

Choke-Point: shortest paths

Q14. *Weighted paths.* Given `PersonX` and `PersonY` , find all weighted paths of the shortest length between them in the subgraph induced by the Knows relationship. The weight of the path takes into consideration amount of Posts/Comments exchanged.

- compute weights over a **recursive forum traversal**
  - on the fly, or
  - materialized, but then maintain them under updates
- **compute shortest paths** using these weights in the friends graph

SNB Interactive Workload: Complex Read Query Set

Q1. *Extract description of friends with a given name* Given a person's `firstName`, return up to 20 people with the same first name, sorted by increasing distance (max 3) from a given `person`, and for people within the same distance sorted by last name. Results should include the list of workplaces and places of study.

Q2. *Find the newest 20 posts and comments from your friends.* Given a start `Person`, find (most recent) Posts and Comments from all of that Person's friends, that were created before (and including) a given `Date`. Return the top 20 Posts/Comments, and the Person that created each of them. Sort results descending by

Q3. *Friends within 2 steps that have recently traveled to countries X and Y.* Find friends and friends of friends of a given `Person` who have made a post or a comment in the foreign `CountryX` and `CountryY` within a specified period of `DurationInDays` after a `startDate`. Return top 20 Persons, sorted descending by total number of posts.

Q4. *New Topics.* Given a start `Person`, find the top 10 most popular Tags (by total number of posts with the tag) that are attached to Posts that were created by that Person's friends. Only include Tags that were attached to Posts created within a given `time interval`, and that were never attached to Posts created before this interval.

Q5. *New groups.* Given a start `Person`, find the top 20 Forums which that Person's friends and friends of friends became members of after a given `Date`. Sort results descending by the number of Posts in each Forum that were created by any of these Persons.

Q6. *Tag co-occurrence.* Given a start `Person` and some `Tag`, find the other Tags that occur together with this Tag on Posts that were created by start Person's friends and friends of friends. Return top 10 Tags, sorted descending by the count of Posts that were created by these Persons, which contain both this Tag and the given Tag.

Q7. *Recent likes.* For the specified `Person` get the most recent likes of any of the person's posts, and the latency between the corresponding post and the like. Flag Likes from outside the direct connections. Return top 20 Likes, ordered descending by creation date of the like.

Q7. *Recent likes.* For the specified `Person` get the most recent likes of any of the person's posts, and the latency between the corresponding post and the like. Flag Likes from outside the direct connections. Return top 20 Likes, ordered descending by creation date of the like.

Q8. *Most recent replies.* This query retrieves the 20 most recent reply comments to all the posts and comments of `Person`, ordered descending by creation date.

Q9. *Latest Posts.* Find the most recent 20 posts and comments from all friends, or friends-of-friends of `Person`, but created before a `Date`. Return posts, their creators and creation dates, sort descending by creation date.

Q10. *Friend recommendation.* Find a friend of a friend who posts much about the interests of `Person` and little about topics that are not in the interests of the user. The search is restricted by the candidate's `horoscopeSign`. Returns 10 Persons for whom the difference between the total number of their posts about the interests of the specified user and the total number of their posts that are not in the interests of the user, is as large as possible. Sort the result descending by this difference.

Q11. *Job referral.* Find a friend of the specified `Person`, or a friend of her friend (excluding the specified person), who has long worked in a company in a specified `Country`. Sort ascending by start date, and then ascending by person identifier. Top 10 result should be shown.

Q12. *Expert Search.* Find friends of a `Person` who have replied the most to posts with a tag in a given `TagCategory`. Count the number of these reply Comments, and collect the Tags that were attached to the Posts they replied to. Return top 20 persons, sorted descending by number of replies.

Q13. *Single shortest path.* Given `PersonX` and `PersonY` , find the shortest path between them in the subgraph induced by the

Q14. *Weighted paths.* Given `PersonX` and `PersonY` , find all weighted paths of the shortest length between them in the subgraph induced by the Knows relationship. The weight of the path takes into consideration amount of Posts/Comments exchanged.

Choke-Point: **outdegree correlation**

Q3. *Friends within 2 steps that recently traveled to countries X and Y.* Find top 20 friends and friends of friends of a given `Person` who have made a post or a comment in the foreign `CountryX` and `CountryY` within a specified period of `DurationInDays` after a `startDate`. Sorted results descending by total number of posts.

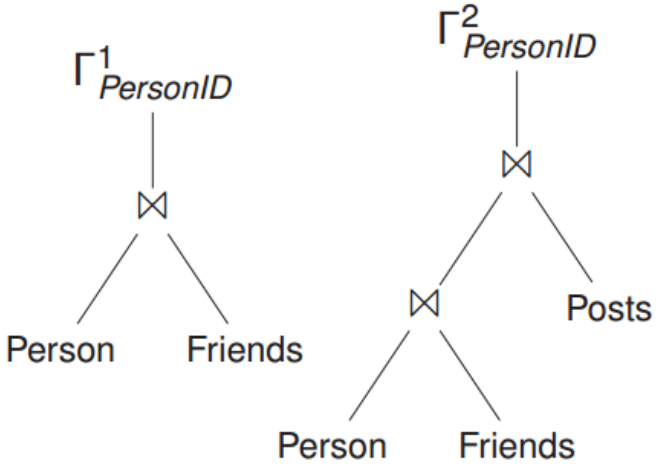
- Travel is correlated with location
  - People travel more often to nearby countries
- Outdegree after `(countryX,countryY)` selection varies a lot
  - **(Australia,NZ)**: high outdegree ("join hit ratio")
  - **(Australia,Belgium)**: low outdegree ← *different query plan, or navigation strategy likely wins*

LDBC

Parameter Curation

- Example: Q3
  - Problem: value correlations cause very large variance
  - Solution: data mine for **stable** parameter **equivalence classes**

TPCTC2014 “Parameter Curation for Benchmark Queries” Andrey Gubichev (TUM) & Peter Boncz (CWI)



PersonID	$\bowtie_1$	$\bowtie_2$
…	…	…
1542	60	99
1673	60	102
7511	60	103
958	60	120
1367	61	101
…	…	…

- form sliding windows of rows
- pick sub-window with the smallest variance in the next column

LDBC

Query Mix & Metric

Query Mix

- Insert queries (~10% of time):

  ➔ challenge: **execute parallel but respect data dependencies in the graph**

- Read-only Complex Queries (~50% of time)

  ➔ challenge: **generate query parameters with stable query behavior**

  Parameter Curation to find "equivalence classes" in parameters

- Simple Read-only Queries (~40% of time)
  - Retrieve Post / Retrieve Person Profile

Metric

- **Acceleration Factor** (AF) that can be sustained (+ AF/$ weighted by cost)
  - with 99th percentile of query latency within maximal query time

LDBC

SNB Query Driver

- Dependency-aware parallel query generation
  - **Problem**: friends graph is non-partitionable, but imposes ordering constraints.

    Could cause large checking overhead, impeding driver parallelism.
  - **Solution**: Window-based checking approach for keeping driver threads roughly synchronized on a global timestamp.

    Is helped by DATAGEN properties that ensure there is a minimal latency between certain dependencies (e.g. entering the network and making friends, or posting on a new friend's forum). This minimal latency provides synchronization headroom.

Summary

- LDBC
  - Graph and RDF benchmark council
  - Choke-point driven benchmark design (user+system expert involvement)
- Social Network Benchmark
  - Advanced social network generator
    - skewed distributions, power laws, value/structure correlations, flash mobs
  - 3 workloads: Interactive (←focus of this paper), BI, Analytics
    - Interactive Query Mix & Metrics
    - Parallel Query Driver that respects dependencies efficiently
    - Parameter Curation for stable results

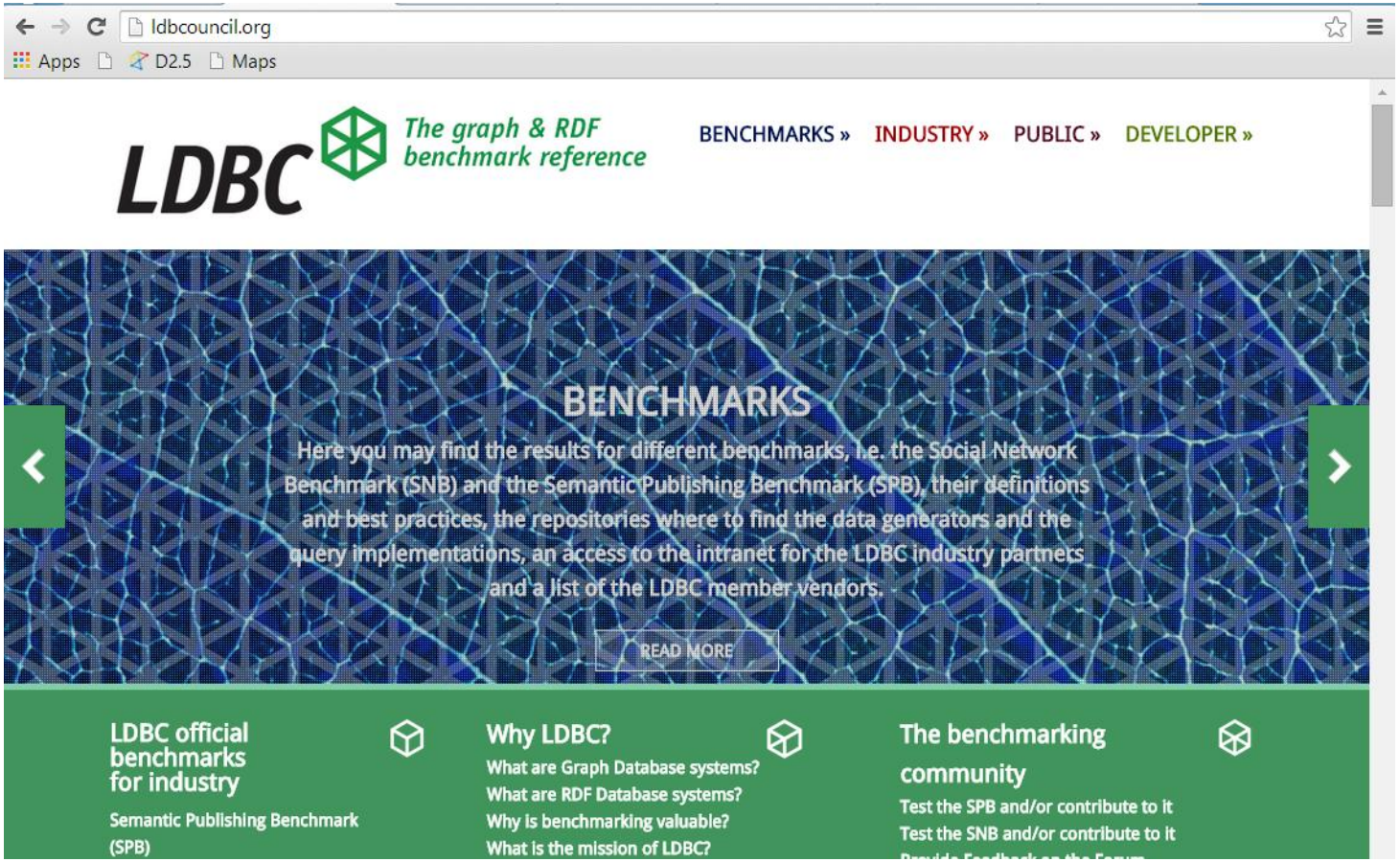
7th LDBC Technical User Community meeting
November 9+10 2015, IBM TJ Watson (NJ)

LDBC

Questions?

http://www.ldbcouncil.org http://github.com/ldbc

Blogs
Specifications
Early Result FDRs
Videos of TUC talks
Developer info
Code, Issue Tracking



LDBC