Datalog for the Web 2.0: the case of Social Network Data Management

follower(Luke, X)
close-friends(Danny, X)
close-friends(Matthew, X), profile-picture(X, P)
conversation(X, [President, Obama, Dalai, Lama])

Danilo Montesi, Matteo Magnani
University of Bologna, Dept. of Computer Science
montesi@cs.unibo.it --- matteo.magnani@cs.unibo.it

Research project partially supported by Telecom Italia
Social Network Sites are Large Databases...

• ...with some peculiar features.
  – Global phenomenon, with specific sites expressing cultural diversities:
    • Facebook (>300,000,000 users)
    • QQ (>350,000,000 users).
  – Contain relevant information:
    • Terroristic attack, Mumbai, 2008.
    • Twitter revolution, Iran, 2009.
  – Enable relevant applications:
    • Politics, marketing, ...
Some figures

**Who will be my friend?**
Social-networking sites, total unique visitors
October 2009, m

- Facebook: 430.2
- Windows Live
- MySpace
- Baidu
- Twitter
- Orkut
- Hi5
- QQ
- LinkedIn
- deviantART

Source: comScore

**Sociable types**
Average time spent on social-networking sites
October 2009, hours per user

- Australia
- Britain
- Italy
- United States
- South Korea
- Spain
- Brazil
- Germany
- France
- Japan

Source: Nielsen

**In friends we trust**
Global consumer trust in advertising, Q1 2009
% of respondents

- Friends' recommendations
- Brand websites
- Consumer forums
- Editorial content
- Brand sponsorships
- Television
- Newspapers
- Magazines
- Radio
- Billboards

Source: Nielsen
Social databases

- To support social network sites, relational databases are enough.
- To enable *advanced applications*, we need specific data models and query languages:
  - EXAMPLE 1: Real-time conversation-aware search engines.
  - Should consider user popularity, networks of messages on related topics, ...
  - EXAMPLE 2: Message propagation prediction.
  - Should consider strength of relationships, probability of replying, ...
- Social data structures are often based on networks.

→ Datalog: good candidate.
  - Clean semantics.
  - Recursive queries.

- However, social databases and applications require a number of features.
- In this talk we examine the requirements of a Social Query Language (SocQL), to encourage a discussion on the features Datalog should have to be used in this real and relevant context.
Our case study: Friendfeed

- Recently acquired by Facebook.
- Aggregates social content from Facebook, Twitter, Blogs.
- Allows microblogging, but also complex conversations.
  → contains data representative of different social behaviors.
- Small (about 5 million messages per week)
  → good for a case study.
- Public APIs.
  → most of data accessible.
Data extraction

• API monitored from Sep. 6, 2009 to Sep. 19, 2009.

• Stored all entry IDs.

• Retrieved all associated comments and likes, in XML, and exported to CSV files.

• Retrieved the network of followers (friends), starting from active users and reconstructing the graph of all connections.
Social data model

**Entry** *(PostID, PostedBy, Timestamp, Text, Language)*

**Comment** *(PostID, EntryRef, PostedBy, Timestamp, Text, Language)*

**Like** *(User, EntryRef, Timestamp)*

**User** *(ID, Name, Description)*

**UserLink** *(Follower, Followed)*
Social data model

**Entry**(*PostID*, PostedBy, Timestamp, Text, Language)
**Comment**(*PostID*, EntryRef, PostedBy, Timestamp, Text, Language)
**Like**(*User*, EntryRef, Timestamp)
**User**(*ID*, Name, Description)
**UserLink**(*Follower*, *Followed*)

- Network of users.
Social data model

Entry \( (\text{PostID}, \text{PostedBy}, \text{Timestamp}, \text{Text}, \text{Language}) \)
Comment \( (\text{PostID}, \text{EntryRef}, \text{PostedBy}, \text{Timestamp}, \text{Text}, \text{Language}) \)
Like \( (\text{User}, \text{EntryRef}, \text{Timestamp}) \)
User \( (\text{ID}, \text{Name}, \text{Description}) \)
UserLink \( (\text{Follower}, \text{Followed}) \)

- Network of users.
- Unstructured content.
Social data model

- Network of users.
- Unstructured content.
- Chains of messages (conversations).

Entry(`PostID`, PostedBy, Timestamp, Text, Language)
Comment(`PostID`, `EntryRef`, PostedBy, Timestamp, Text, Language)
Like(`User`, `EntryRef`, Timestamp)
User(`ID`, Name, Description)
UserLink(`Follower`, `Followed`)

.partytonight?
.Yep!Mary?
Social data model

- **Entry**\((PostID, PostedBy, Timestamp, Text, Language)\)
- **Comment**\((PostID, EntryRef, PostedBy, Timestamp, Text, Language)\)
- **Like**\((User, EntryRef, Timestamp)\)
- **User**\((ID, Name, Description)\)
- **UserLink**\((Follower, Followed)\)

- Network of users.
- Unstructured content.
- Chains of messages (conversations).
- Transitive connections.
Social data model

Entry\( (\text{PostID}, \text{PostedBy}, \text{Timestamp}, \text{Text}, \text{Language}) \)
Comment\( (\text{PostID}, \text{EntryRef}, \text{PostedBy}, \text{Timestamp}, \text{Text}, \text{Language}) \)
Like\( (\text{User}, \text{EntryRef}, \text{Timestamp}) \)
User\( (\text{ID, Name, Description}) \)
UserLink\( (\text{Follower, Followed}) \)

- Network of users.
- Unstructured content.
- Chains of messages (conversations).
- Transitive connections.
- Implicit references (through data).
Data requirements:

size

- Order of $10^{6-7}$ records per week.
- About 500,000 users.
- About 1 GB (text only) per week.
- In addition to multimedia content.
- (Friendfeed has 1/1000 users w.r.t. Facebook...).
- Datalog implementations should be based on existing relational technologies and systems.
Data requirements: structured, semi-structured and unstructured

- Relations.
- Text and multimedia.
- Multiple graphs: users, messages.
- Graphs are the main motivation behind a Datalog-based social query language.
- However, atoms can be complex (more later on this...).
- Timestamps.
Data requirements: labeled and weighted edges

- Different kinds of edges.
  - Different behaviors.
  - Different relationships.

Weights to indicate properties necessary to analyze the data and make predictions.

Strength of connection.

Probability of replying (w.r.t. topic).
Data requirements: labeled and weighted edges

- Different kinds of edges.
  - Different behaviors.
  - Different relationships.
- Weights to indicate properties necessary to analyze the data and make predictions.
  - Strength of connection.
  - Probability of replying (w.r.t. topic).
Operational requirements: recursive traversal

• Many properties do not regard just direct connections.
  – E.G., user popularity is important to rank messages and conversations.
  – A user with even a single follower who is very popular will be popular as well (her messages will be seen by many users when commented).

• Datalog already supports this kind of queries.
  – Who may see a message posted by Bob?
Operational requirements: aggregation

- Queries may involve sub-graphs.
- E.G., the network of MY friends.
  - Return the probability that a message on technology will be commented.
- Aggregate functions are needed to compute these queries.
- Aggregations should work on numbers, and also edge types.
  - How many connections of each kind (follower, followed, ...) do I have?
Operational requirements: keyword search and conversation retrieval

• Content is important.
  – However, a lot of content can (sometimes) be considered noise: :-), :-D, lol, rotfl, ok, ciao...
  – Often messages are short, and thus different from documents in traditional IR.

• User's relevance is important as well in ranking.

• The same people exchanging the same message at different times may have different rankings.
  – Return all conversations about Berlusconi, ordered by their relevance.
Operational requirements: data analysis and exploratory queries

- A lot of basic information is not explicitly exposed by the data model.
- Vague queries based on data analysis capabilities.
  - Return all my friends, grouped by their degree of friendship.
  - (involves graph clustering)
  - May return: (Bob, Mary [strong], John [weak])
- In addition, exploratory and interactive queries, typical of semistructured data models.
  - Return my followers, then show me their names. Select some of them, and expand their connections...
<table>
<thead>
<tr>
<th>Data Model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Large data sets</strong></td>
</tr>
<tr>
<td><strong>Relations</strong></td>
</tr>
<tr>
<td><strong>Data graphs</strong></td>
</tr>
<tr>
<td><strong>Arc weights</strong></td>
</tr>
<tr>
<td><strong>Arc labels</strong></td>
</tr>
<tr>
<td><strong>Operations</strong></td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Recursive traversal</td>
</tr>
<tr>
<td>Aggregates (weights)</td>
</tr>
<tr>
<td>Aggregates (labels)</td>
</tr>
<tr>
<td>Keyword search</td>
</tr>
<tr>
<td>Graph ranking</td>
</tr>
<tr>
<td>Data Analysis</td>
</tr>
</tbody>
</table>
Conclusion

• We have identified the requirements of a generic query language for Social Web 2.0 applications (SocQL).

• Studying a real and complex social application.

• Datalog is a potential candidate, because of its clean representation of recursive queries.

• But it should provide additional capabilities.

• A big opportunity to leave academia...
Datalog for the Web 2.0: the case of Social Network Data Management

follower(Luke, X)
close-friends(Danny, X)
close-friends(Matthew, X), profile-picture(X, P)
conversation(X, [President, Obama, Dalai, Lama])

Danilo Montesi, Matteo Magnani
University of Bologna, Dept. of Computer Science
montesi @ cs.unibo.it --- matteo.magnani @ cs.unibo.it

Research project partially supported by Telecom Italia