Why Use Tables and Graphs for Knowledge Discovery

JOHN FEO

NORTHWEST INSTITUTE FOR ADVANCED COMPUTING
PACIFIC NORTHWEST NATIONAL LABORATORY

HPC USER FORUM
SEPTEMBER, 2016
Mary called her sister Sally to discuss buying her 6-year daughter a pony for Christmas.

1) Mary called Sally
2) Mary has a sister named Sally
3) Sally has a sister named Mary
4) Either Mary or Sally has a daughter
5) The daughter is 6 years old
6) Mary wants to buy a pony

Sally rented Joe’s condo in Hawaii for a two week vacation. She paid $1200 rent.

1) Sally traveled to Hawaii
2) Sally vacationed in Hawaii
3) Joe owns a condo
4) Joe’s condo is in Hawaii
5) Sally rented Joe’s condo
6) Joe rented his condo for $600 per week

<table>
<thead>
<tr>
<th>NAME</th>
<th>SIBLING</th>
<th>CHILD</th>
<th>AGE</th>
<th>CALLED</th>
<th>FUTURE PURCHASES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Sally</td>
<td>?</td>
<td>?</td>
<td>Sally</td>
<td>Pony</td>
</tr>
<tr>
<td>Sally</td>
<td>Mary</td>
<td>?</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mary called her sister Sally to discuss buying her 6-year daughter a pony for Christmas.

1) Mary called Sally
2) Mary has a sister named Sally
3) Sally has a sister named Mary
4) Either Mary or Sally has a daughter
5) The daughter is 6 years old
6) Mary wants to buy a pony

Sally rented Joe’s condo in Hawaii for a two week vacation. She paid $1200 rent.

1) Sally traveled to Hawaii
2) Sally vacationed in Hawaii
3) Joe owns a condo
4) Joe’s condo is in Hawaii
5) Sally rented Joe’s condo
6) Joe rented his condo for $600 per week
Why graphs? Good for finding patterns

**Witty Worm**

**Distributed DoS Smurf Attack**

---

*Some String*

refersPatientTo

?Pr1 : XXX

?Np1

?Pt1

:YYY

:YYY

?Pr2

:YYY

:YYY

?Pt2

:YYY

?Np2
Why not graphs?

<table>
<thead>
<tr>
<th>id</th>
<th>srcaddr</th>
<th>dstaddr</th>
<th>protocol</th>
<th>srcport</th>
<th>dstport</th>
<th>start</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62.175.55.106</td>
<td>131.215.122.25</td>
<td>6</td>
<td>25</td>
<td>5641</td>
<td>5682</td>
<td>9084</td>
</tr>
<tr>
<td>2</td>
<td>67.72.170.48</td>
<td>131.215.122.25</td>
<td>6</td>
<td>57947</td>
<td>3724</td>
<td>17364</td>
<td>24602</td>
</tr>
<tr>
<td>3</td>
<td>67.72.170.48</td>
<td>131.215.122.25</td>
<td>6</td>
<td>60373</td>
<td>3724</td>
<td>28494</td>
<td>31684</td>
</tr>
<tr>
<td>4</td>
<td>67.72.170.48</td>
<td>179.137.146.48</td>
<td>6</td>
<td>80</td>
<td>64845</td>
<td>41239</td>
<td>43766</td>
</tr>
<tr>
<td>5</td>
<td>67.72.170.48</td>
<td>131.215.122.25</td>
<td>17</td>
<td>58020</td>
<td>9050</td>
<td>45516</td>
<td>46136</td>
</tr>
</tbody>
</table>

Edge attributes

Product A
producer
quantity
price

Vendor A
Vendor B
Vendor C
country
country
country

USA
UK
So, may be we need ...

- Table and graph data structures
- Table operations and graph methods
- Relational and graph/knowledge databases
- New query languages? No, just a few extensions
- Disruptive platforms? No, just integrate new features
- New workflows? No, just new options
Some attributes are edges

GraQL – adding a graph view to SQL

- Extensions to declare vertices and edges
- Extensions to describe graph patterns
- Extensions to return query results as a graph

```
create vertex ProductVtx(id) from table Product
create vertex ProducerVtx(id) from table Producer
create vertex TypeVtx(id) from table Type
create vertex FeatureVtx(id) from table Feature
```

```
create edge feature with vertices (ProductVtx, FeatureVtx) from table ProductFeature
where ProductFeature.product = ProductVtx.id and ProductFeatures.feature = FeatureVtx.id
```

A graph view

1. **TypeVtx**: subclass
2. **FeatureVtx**: feature
3. **ProductVtx**: type
4. **OfferVtx**: product
5. **VendorVtx**: vendor
6. **PersonVtx**: reviewer

Connections:
- **ProducerVtx** produces **ProductVtx**
- **ReviewVtx** reviews **ProductVtx**
- **VendorVtx** offers **ProductVtx**
A graph makes it easy to find patterns

**SELECT THE TOP 10 PRODUCTS MOST SIMILAR TO A SPECIFIC PRODUCT, RATED BY THE COUNT OF FEATURES THEY HAVE IN COMMON**

```
select y.id
from graph ProductVtx(id = %Product%) -feature-> FeatureVtx()
   <-feature- foreach y: ProductVtx(id != %Product%)
into table T1

select top 10
   id, count(*) as sameFeatures
from table T1
   group by id
   order by count(*) desc

SELECT ?otherProduct ?sameFeatures {
   ?otherProduct type Product .
   FILTER(?otherProduct != %Product%)
   {SELECT ?otherProduct (count(?otherFeature) as ?sameFeatures)
      {%Product% productFeature ?feature .
      ?otherProduct productFeature ?otherFeature .
      FILTER(?feature = ?otherFeature)
      }
      Group By ?otherProduct
   }
} }
Order By desc(?sameFeatures) ?otherProduct
Limit 10
```
More complex definitions and queries

```plaintext
create vertex server(ipAddr)
from table Netflow
with ip = srcaddr or ip = dstaddr

create edge comm with vertices (server as src, server as dst)
from table Netflow
with attributes (id, protocol, srcport, dstport, size, start, end)
where src.ipAddr = srcaddr and dst.ipAddr = dstaddr

from graph
( server -comm(protocol = 1) -> foreach ip1: server -comm(protocol = 1) ->
  server -comm(protocol = 1) -> foreach ip3: server -comm(protocol = 1) -> ip1
 )
and
( ip1 -comm(protocol = 1) -> server -comm(protocol = 1) -> ip3
 )
and
( ip1 -comm(protocol = 1) -> server -comm(protocol = 1) -> ip3
 )

-x->   out-edge
<--x-- in-edge
~x-->  conditional edge
|--x--> no edge
```
Graph Engine for Multithreaded Systems (GEMS)

Provide an in-memory property-graph database capable of doing deep analytic query processing over multi-terabyte datasets with human-interactive response times on commodity computing parts

Focus on

- Blending relational and graph data representations, methodologies, algorithms, and queries
- Providing an application programming interface (API) for custom applications using embedded graph databases
- Supporting data feeds that supply data in batches (e.g., hourly)
- Serving a range of user profiles: application programmer, subject matter expert analyst
GEMS 2.0

Scalable, in-memory hybrid search engine for pattern discovery


Parallel and Distributed Algorithms and data Structures:
- Table to store tuples
- Index (Graph) for fast neighbours access
- Dictionary for encoding
- CSV and RDF parser

Global Memory and Threading runtime system:
- Runtime library implemented in C
- Requirements: MPI, Linux, x86
  - Single node uses Intel TBB
- Partitioned Global Address Space (PGAS)
- Parallel loops program structure (*parFor*)
- Massive user-level asynchronous tasks
- Software multi-threading to hide latency
- Message aggregation

Morari, et.al. Scaling Irregular Applications through Data Aggregation and Software Multithreading. IPDPS 2014: 1126-1135
NOUS: Incremental Knowledge Graphs

- Incrementally construct KGs from streaming data and multiple sources
- Support advanced trending and explanatory questions
- Automatically generate hypotheses to guide users in information discovery
- Web-based, interactive UI

NOUS features

- Continuous pattern (or rule) discovery, search, and reasoning
- Algorithms that monitor emerging “solution sketches” or “pathways”
- Integrates machine-learning models (e.g., LDA, RNN) to steer the search

- Detecting new entities and relationships
- Distant supervision based relationship discovery
- Zero-shot learning for discovering new entities

- Dynamic checking for verification
- Learn and maintain a Dynamic Rule-base to accept or reject facts
- Use ML to assign confidence

- Combine structure and attributes
- Track provenance
United States government crack an iPhone that belonged to a gunman in the San Bernardino

Apple repair the particular iPhone hole that the government hacked.

Federal officials specify the procedure used to open the iPhone.

Federal officials deny to specify the procedure used to open the iPhone.

Jay Kaplan, chief executive of Synack and a former NSA analyst, says Apple has to earn the trust of Apple’s customers,”

F.B.I. cracks Mr. Farook’s

LegbaCore, which previously found and fixed flaws for Apple. found and fixed flaws for Apple.

Built on top of Stanford CoreNLP and OpenIE

Adds a layer with heuristics to minimize the noise in triples
Post processing raw triples

- Entity Disambiguation
- Event Detection
- Relationship Discovery
Navigating data

Find news about autonomous drones

What are their capabilities?

What are their software components?

Ardupilot: An Open Source award winning platform
Navigating data (cont.)

Which vendors are mentioned with Ardupilot?

Where is Arduino?

Who sells Arduino?
Summary

- Big data is changing our world

- Winning platforms will implement a variety of methods and structures

- PNNL has developed an in-memory, scalable table-graph engine capable of supporting a trillion attributed vertices and edges

- ... and extended SQL to define graph views and queries

- ... and built a knowledge graph capable of processing two million web crawls and analyzing trends, connecting dots, and presenting hypotheses
Contacts

- GraQL – Daniel Chavaria, daniel.chavaria@pnnl.gov
- GEMS – David Haglin, david.haglin@pnnl.gov
- NOUS – Sutanay Choudhury, sutanay.choudhury@pnnl.gov