Temporal Graph Analytics with GRADOOP

Christopher Rost and Kevin Gomez
Leipzig University
About the speakers and the team

Kevin, PhD Student
Christopher, PhD Student
Prof. Dr. Erhard Rahm, Database Chair
Timo, M.Sc. Student
Lucas, B.Sc. Student
We’re hiring!

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

- Visit us at www.scads.de
- Open positions and projects (Developer and PhD Students)
- Various topics:
  - Machine Learning (Graph-ML, Privacy)
  - Image Recognition
  - Blockchain (Distributed-Ledger)
  - Graph Analytics
Motivation
“Graphs are everywhere”
“Graphs are everywhere”
“Graphs are everywhere”
“Graphs are everywhere”

- **Anna**: friendOf Mike, friendOf James, friendOf Dutch, friendOf Bob, friendOf Dutch
- **Mike**: friendOf John, friendOf Anna
- **James**: friendOf Anna
- **Dutch**: friendOf Dutch, friendOf Dutch
- **Bob**: friendOf Bob, friendOf Sheldon, friendOf Raj
- **Sheldon**: friendOf Sheldon
- **Raj**: friendOf Sheldon
- **John**: friendOf Anna
- **Chris**: friendOf Bob
- **Sally**: friendOf Bob
- **Arthur**: friendOf Jack
- **Jack**: friendOf Arthur

**Anna's attributes**: age: 29, sex: f, yob: 1991
“Graphs are everywhere”
“A open-source framework and research platform for efficient, distributed and domain independent management and analytics of heterogeneous and temporal graph data.”
Architecture

Graph Analytical Language (GrALa)

EPGM

TPGM

Distributed Operator Implementation

Apache Flink (Distributed Dataflow Engine)

Apache Accumulo

Apache HBase

HDFS / YARN

Java 8

45k LOC

ALv 2.0

Graph Analytical Language (GrALa)
Graphs and collections of graphs

LG3; label:"ACDC"; [user:3]

LG1; label:"Ramstein"; [user:6, admin:"Bob"]

LG2; label:"Metallica"; [user:4, admin:"Arthur", avgAge:34]
Operators

Unary Operators:
- Aggregation
- Pattern Matching
- Transformation
- Grouping
- Subgraph
- Sampling
- Layouting
- Call

Binary Operators:
- Combination
- Overlap
- Exclusion
- Equality

Algorithms:
- Flink Gelly Library
  - Adaptive Partitioning
  - Page Rank
  - Connected Components
  - Label Propagation

Graph Collection:
- Subgraph
- Transformation
- Pattern Matching
- Select
- Distinct
- Apply
- Reduce
- Call

Logic Group:
- Subgraph
- Transformation
- Pattern Matching
- Select
- Distinct
- Apply
- Reduce
- Call

Frequent Subgraph
Temporal extension
Temporal extension
Temporal extension
Temporal extension

LG3; label:"ACDC"; [user:3]
val_to: 2025
tx_from: 2005
tx_to: undef.

LG1; label:"Ramstein"; [user:6, admin:"Bob"]
val_from: 2006
val_to: undef
tx_from: 2006
tx_to: undef.

val_from: undef,
val_to: undef,
tx_from: 2011
tx_to: undef.

val_from: 2011
val_to: 2019
tx_from: 2011
tx_to: undef.
Time dependent operators

Operators

Unary
- Aggregation
- Pattern Matching
- Transformation
- Grouping
- Subgraph
- Sampling
- Layouting
- Call

Binary
- Combination
- Overlap
- Exclusion
- Equality

Algorithms

- Flink Gelly Library
- Adaptive Partitioning
- Page Rank
- Connected Components
- Label Propagation

Logical Graph

Graph Collection

Subgraph
- Transformation
- Pattern Matching
- Select
- Distinct
- Apply
- Reduce
- Call

Subgraph
- Transformation
- Pattern Matching
- Select
- Distinct
- Apply
- Reduce
- Call

Frequent Subgraph
# Time dependent operators

<table>
<thead>
<tr>
<th>Operators</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unary</strong></td>
<td><strong>Flink Gelly Library</strong></td>
</tr>
<tr>
<td>Subgraph</td>
<td>Adaptive Partitioning</td>
</tr>
<tr>
<td>Sampling</td>
<td>Page Rank</td>
</tr>
<tr>
<td>Layouting</td>
<td>Connected Components</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Label Propagation</td>
</tr>
<tr>
<td>Difference</td>
<td>Frequent Subgraph</td>
</tr>
<tr>
<td>Grouping</td>
<td></td>
</tr>
<tr>
<td>Pattern Match</td>
<td></td>
</tr>
<tr>
<td>Call</td>
<td></td>
</tr>
</tbody>
</table>

| **Binary** | |
| Combination | |
| Overlap | |
| Exclusion | |
| Equality | |

| **(Temp-)Logical Graph** | |
| Subgraph | |
| Transformation | |
| Pattern Matching | |
| Select | |
| Distinct | |
| Apply | |
| Reduce | |
| Call | |

| **(Temp-)Graph Collection** | |
| Subgraph | |
| Transformation | |
| Pattern Matching | |
| Select | |
| Distinct | |
| Apply | |
| Reduce | |
| Call | |
Use case: spread of airborne pathogens

**Virus X**
- Symptoms: 3\textsuperscript{rd} eye growing
- Transmission: if contact $> 5$ min
- Incubation period: 5 time units

**Contact tracking**
Sensors capturing who is close to whom at what time.
\{empID1, empID2, t\_from, t\_to\}

In case of an infection, which hospital services are at risk of contracting the virus X?
TemporalGraph contacts = mySource.getTemporalGraph();
Breaking news:

Employee of oncology infected by VirusX.

Detected at $t_{20}$. 
Breaking news:

Employee of oncology infected by VirusX.

Detected at $t_{20}$.

**Virus X**

Symptoms: 3rd eye growing

Transmission: if contact $>$ 5 min

Incubation period: 5 time units
TemporalGraph contacts = mySource.getTemporalGraph();

contacts = contacts.snapshot(new FromTo(t15, t20));

Virus X
Symptoms: 3\textsuperscript{rd} eye growing
Transmission: if contact > 5 min
Incubation period: 5 time units
TemporalGraph contacts = mySource.getTemporalGraph();

contacts = contacts.snapshot(new FromTo(t15, t20));

**Virus X**

Symptoms: 3rd eye growing

Transmission: if contact > 5 min

Incubation period: 5 time units
TemporalGraph contacts = mySource.getTemporalGraph();

contacts = contacts.snapshot(new FromTo(t15, t20));

contacts = contacts.groupBy(
  (v → v['srv']),
  [],
  (e → e.getLabel()),
  [new MaxDuration()]);

**Virus X**

Symptoms: 3\textsuperscript{rd} eye growing

Transmission: if contact > 5 min

Incubation period: 5 time units
Virus X
Symptoms: 3rd eye growing
Transmission: if contact > 5 min
Incubation period: 5 time units
Virus X
Symptoms: 3rd eye growing
Transmission: if contact > 5 min
Incubation period: 5 time units

TemporalGraph contacts = mySource.getTemporalGraph();

contacts = contacts.snapshot(new FromTo(t15, t20));

contacts = contacts.groupby(
  (v -> v['srv']),
  [],
  (e -> e.getLabel()),
  [new MaxDuration()]);

contact
maxDur: 40 min

contact
maxDur: 124 min

contact
maxDur: 9 min

contact
maxDur: 122 min

contact
maxDur: 2 min

contact
maxDur: 146 min

contact
maxDur: 2 min

contact
maxDur: 9 min

contact
maxDur: 146 min

$t_0 \ t_1 \ t_5 \ ... \ t_{10} \ t_{15} \ t_{20}$

Time
Breaking news:

Employees of oncology and surgery quarantined because of VirusX.

TemporalGraph contacts = mySource.getTemporalGraph();
contacts = contacts.snapshot(new FromTo(t15, t20));
contacts = contacts.groupBy((v -> v['srv']), [], (e -> e.getLabel()), [new MaxDuration()]);
Conclusion

- Distributed graph analysis system
- Temporal property graph model
  - Bitemporal support
  - Logical graphs and graph collections
  - Composing operators and algorithms
- Declarative workflow creation

- Visit Gradoop: http://gradoop.com
- Read our Wiki: https://github.com/dbs-leipzig/gradoop/wiki
- Start: https://github.com/dbs-leipzig/gradoop/wiki/Getting-started
- Try the examples: https://github.com/dbs-leipzig/gradoop/wiki/Examples