A Distributed Framework for Large-Scale Time-Dependent Graph Analysis

Sabeur Aridhi

Associate professor, LORIA/INRIA Nancy Grand Est, University of Lorraine, France

with Wissem Inoubli, Livia Almada, Ticiana L. Coelho da Silva, Gustavo Coutinho, Lucas Peres, Regis Pires Magalhaes, Jose Antonio F. de Macedo, and Engelbert Mephu Nguifo

This work is funded by the **LSTG French-Brazilian FUNCAP project**
http://projets.isima.fr/lstg/

September 18, 2017

Outline

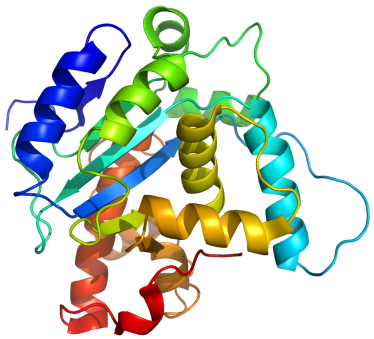
Context and motivations

Application domains

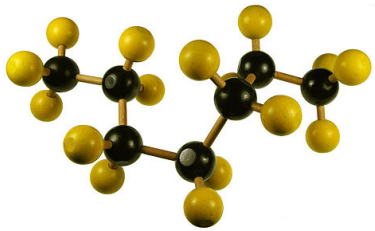
- Computer networks,
- Social networks,
- Bioinformatics,
- Chemoinformatics.

Graph representation

- Data modeling.
- Identifying relationship patterns and rules.



Protein structure



Chemical compound

Social network

Context and motivations

Availability of graph data

- Exponential growth of graph data.
- Availability of multiple graph data sources.
  - Big companies process petabytes of data everyday.

Context and motivations

Availability of graph data

- Exponential growth of graph data.
- Availability of multiple graph data sources.
  - Big companies process petabytes of data everyday.
- 3Vs of Big Data (**V**olume, **V**elocity and **V**ariety).

Context and motivations

Availability of graph data

- Exponential growth of graph data.
- Availability of multiple graph data sources.
  - Big companies process petabytes of data everyday.
- 3Vs of Big Data (**V**olume, **V**elocity and **V**ariety).

Existing works

- Many Graph Processing Frameworks:
  - Static Graph Analysis: Pregel, GraphLab, GraphX, ...
  - Dynamic Graph Analysis: BLADYG, CHRONOS, ...
  - Temporal Graph Analysis: Graphhast, ...
- Dynamic and temporal issues together !

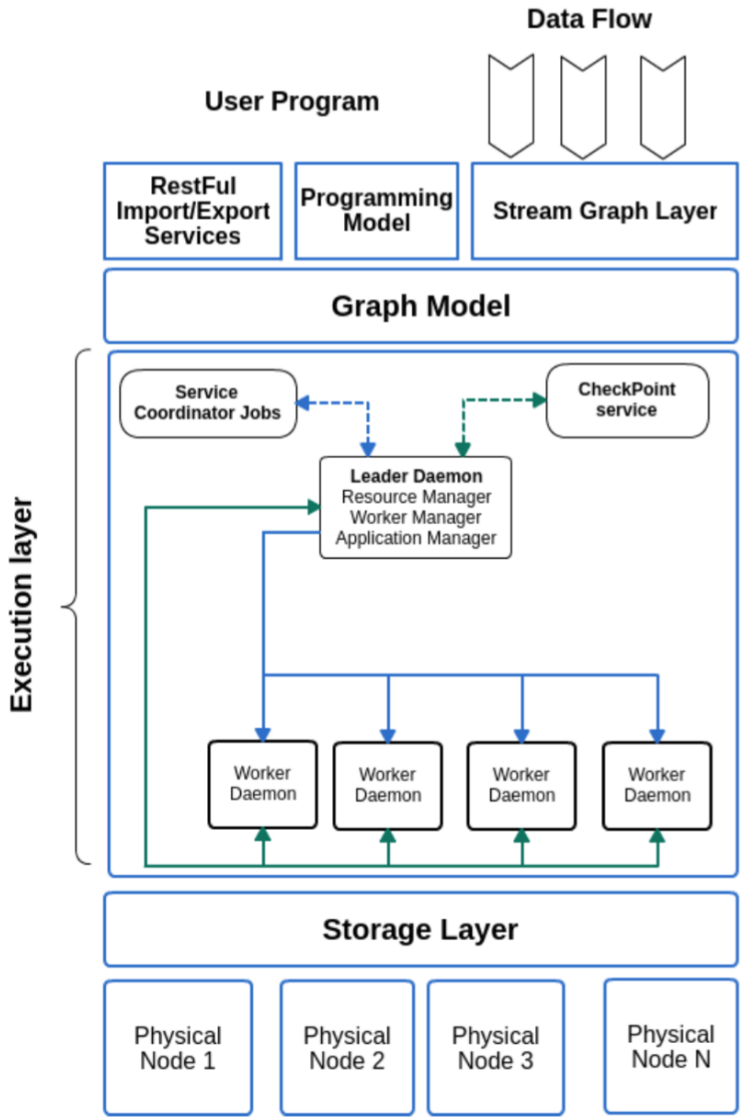
Context and motivations

Challenges

- The size of graph datasets is rapidly increasing.
- Modern graphs are become more and more dynamic.
- New needs are emerging to support properties variation of the graph over the time.
- Single machine approach is not anymore sustainable.
- In many case data are already partitioned.

Outline

Overview


Data Flow
User Program
RestFul Import/Export Services
Programming Model
Stream Graph Layer
Graph Model
Execution layer
Service Coordinator Jobs
CheckPoint service
Leader Daemon
Resource Manager
Worker Manager
Application Manager
Worker Daemon
Worker Daemon
Worker Daemon
Worker Daemon
Storage Layer
Physical Node 1
Physical Node 2
Physical Node 3
Physical Node N

How it works ?

- **Storage layer** It allows saving and persisting the considered graph. We use distributed file systems in order to store large graphs and to guarantee a fault tolerant storage solution.
- **Execution layer** The execution layer represents the core of our proposed framework. It allows scalability of processing and fault tolerance. We have implemented a checkpoint service which aims to ensure and capture the different processing states in real-time.
- **Stream Graph layer** The Stream Graph layer ensure the streaming of graph changes/updates.
- **Graph Model** Graph Model describes the data structure used by the different layers and component of the proposed framework.

How it works ?

A typical temporal/dynamic graph analysis task consists of:

- an input graph (with temporal properties);
- a set of incremental changes (edge/node insertions and/or removals);
- a sequence of (temporal) graph operations;
- an output.

Graph partitioning

Several types of predefined partitioning techniques

- In **hash partitioning**, edges are distributed across machines according to a user-defined hash function.
- In **random partitioning**, edges are distributed across machines randomly.
- In **vertex-cut**, edges are evenly distributed across machines with the goal of minimizing the number of replicated vertices.
- In **edge-cut** partitioning, the vertices of a graph are divided into disjoint clusters of nearly equal size, while the number of edges that span separated clusters is minimum.

Outline

Conclusion

At a glance

- A distributed framework for large-scale time-dependent graph analysis
  - Deals with already partitioned graphs
  - Allows many partitioning techniques on the graph
  - Can be used for many temporal graph-based applications:
    - social networks
    - transportation networks
    - ...

Prospects

Prospects

- Provide a stable implementation of the proposed framework
- Provide scripting language that helps the users to define their tasks.
- Provide graph primitives for dynamic/temporal networks.
- Study fault tolerance, networking and data communication

That's it !

Questions?