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

Wissem Inoubli[1], Livia Almada[4], Ticiana L. Coelho da Silva[4], Gustavo Coutinho[4], Lucas Peres[4], Regis Pires Magalhaes[4], Jose Antonio F. de Macedo[4], Sabeur Aridhi[2], and Engelbert Mephu Nguifo[3]

[1] University of Tunis El Manar, Faculty of sciences of Tunis, LIPAH, 1060, Tunis, Tunisia
[2] University of Lorraine, LORIA, Campus Scientifique BP 239, Vandoeuvre-les-Nancy, France
[3] LIMOS - UBP - Clermont University, BP 10125, 63173, Clermont Ferrand, France
[4] Federal University of Ceara, Computer Science Department, Brazil

Abstract. In the last few years, we have seen that many applications or computer problems are mobilized as a graph since this data structure gives a particular handling for some use cases such as social networks, bioinformatics, road networks and communication networks. Despite its importance, the graph processing remains a challenge when dealing with large graphs. In this context, several solutions and works have been proposed to support large graph processing and storage. Nevertheless, new needs are emerging to support the dynamism of the graph (Dynamic Graph) and properties variation of the graph during the time (temporal graph). In this paper, we first present the concepts of dynamic and temporal graphs. Secondly, we show some frameworks that treat static, dynamic and temporal graphs. Finally, we propose a new framework based on the limits of the frameworks study.

Big Graph Processing, Dynamic Graph, Temporal Graph, Distributed Graph

1 Introduction

Graph data structure represents an appropriate model for several applications in various domains such that social network (e.g., FaceBook, Twitter), bioinformatic [4] , road networks (e.g., Google Maps) [9] [13], social networks [12] and communication networks (e.g., cellular networks) [10]. Its interests increase especially with the development of intelligent objects: internet of objects (IoT) and machine learning with the deep-learning model that is based on the processing of graph data. Despite the fact that this data structure (graph format) shows its effectiveness for several problems and applications modeling, different linked problems arise mainly: (1) centralized machines are not able to deal with complex tasks on big graphs, (2) big graphs may change over time. Precisely, graphs can be subject to a set of structural modifications (addition and/or deletion of vertices and/or edges) and updates of some properties. The high volume of changes in dynamic graphs can be detected in big applications like Facebook whose updates exceed 86 thousands per second in 2013 [3]. Many solutions proposed to solve queries in graphs fails to represent the reality in the sense that they do not consider the temporal dependence. These tools were developed for solving queries in static graphs and can not be directly extended to answer time-dependent problems. In this paper, we present a novel framework for dynamic and temporal graph processing. The proposed framework allows defining time-dependent graph processing tasks in a distributed way. Our framework provides a high level API that allows the implementation of distributed and scalable time-dependent algorithms. Since static (non-time-dependent) graphs are a particular case of time-dependent graphs, our solution is also suitable for handling this kind of graphs. The rest of the paper is structured as follows. In Section 2, we highlight the related works on temporal and distributed graph processing. In Section 3, we describe the proposed framework and its components.In Section 4, we draw the conclusion and the future works.

2 Related works

The graph processing topic has attracted many researchers and industrial institutes [2] . In fact, it is considered as huge consumers of computing resources due to its complexity. Therefore, a single thread or a simple machine does not meet our needs if we deal with large graphs or if we aim to treat complex tasks. This challenge prompted the birth of new programming models and dedicated frameworks to build efficient distributed and parallel applications [6]. In this context, considerable efforts have been made since distributed computing needs very hard configuration like resources managers, fault tolerance, communications between machines, data partitions, and other configurations. In this section, we briefly review related works and then determinate main constraints to support dynamic and temporal graph processing effectively. Pregel [15] is a scalable and distributed framework for large graph processing. It represents an evolution of the MapReduce framework [5] to graph processing systems. It is known as the first implementation of Bulk Synchronous Parallel model which is based on vertices computing and a sequence of iterations called supersteps. Each vertex sends and receives messages to or from its neighbors. Note that in the framework or in the programming model (BSP), the communication is a primordial aspect that reduces the graph processing runtime. Giraph [7] is another distributed framework that represents a next generation of the Pregel framework. It uses a multi-threading to optimize the computing, so the MapReduce framework also inspires it. Each computing task is considered as a Job and it uses HDFS to store the input or the output data. Various works have been also conducted to support dynamic aspects of Big graphs like PowerGraph [8], Graphchi [16], and X-Stream [17] but they are limited to dynamic storage and partition. Others systems like GraphIn [19] and BLADYG [1] have been proposed to ensure dynamic graph processing. GraphIn proposes an incremental graph processing and is based on a novel programming model called IN-GAS (based on gather-apply-scatter programming paradigm). As for BLADYG, it consists of a scalable, distributed and dynamic framework for graph processing which is based on an actor model [18]. Note also that temporal graph analysis is an important aspect that must be taken into consideration in future works. In this context, Graphast [14] has been recently developed as a framework that supports such an aspect. It is addressed to ensure a temporal processing graph and to provide users with many tools for applications building on a time-dependent graph.

3 Distributed temporal and dynamic graph analysis

Based on the theoretical study of the above-mentioned frameworks, we propose a novel distributed framework for temporal and dynamic graph representation and analysis. Figure 1 presents the global architecture of the proposed framework. We mention that the proposed system aims to overcome limits of the above-mentioned frameworks and to meet all requirements for processing dynamic and temporal big graphs. As shown in Figure 1, the proposed system is divided into several layers.

Storage layer

The storage layer allows saving and persisting the considered graph. This layer has been implemented in some frameworks. However, others solutions do not implement this component or do not support very big graphs. In our framework, 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 is implemented to support efficiently all the requirements for large graph processing. Among these requirements, we cite basically the scalability of processing and the fault tolerance. In this context, we note that several solutions such as GPS and Graphast do not take into consideration some aspects such as the scalability and the distributed computing, while others like BLADYG, Pregel and GraphX ensure them. However, the latters can not guarantee fault tolerance. To handle this limit, we have implemented a checkpoint service which aims to ensure and capture the different processing states in real-time.

Stream Graph layer

We notice that the changes that occur to the graph data are mainly addition and/or deletion of nodes/edges. The Stream Graph layer ensure the streaming of graph changes/updates. To do that, we use Kafka Broker [11] as a streaming tool that ensures the continuity and the scalability of data flows.

Graph Model

Graph Model describes the data structure used by the different layers and component of the proposed framework. The Graph Model component describes the edges and the vertices of the considered graph. It also describes the graph updates. We also mention that the Graph Model layer allows to represent time dependent costs on the edges. In our model, nodes and edges can have time-dependent costs

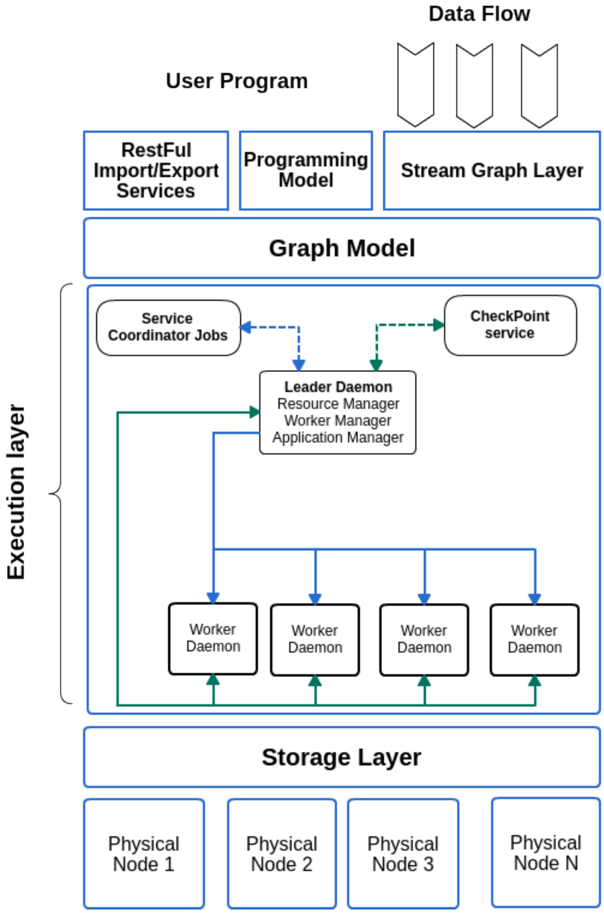

Fig. 1. Proposed Framework

as well as other properties such as the distance between nodes, geographical coordinates, and description. Temporal functions represent time-dependent costs, mapping time periods to values. The model is flexible enough to allow the addition or removal of properties. It also allows dynamic updates on property values. All the other modules of the framework interact with the Graph Model.

Level API

To facilitate the development of user applications and to ensure a high level of abstraction that allow the exploitation of this framework in several applications, we propose an API level. It implements a programming interface that helps the user to build several applications. We also mention that the level API allow the implementation of efficient algorithms / queries on the graph model.

RestFul Import/Export Service

In this component, we propose a RestFul module which seeks to ensure the compatibility with other systems or applications and to manipulate data efficiently. This layer is responsible for importing/exporting data from given data sources and formats, like GML, GraphML, and GraphSON,to build the Graph Model. New data import/export services are easily added to the framework by the implementation of a standard interface.

4 Conclusion

In this paper we present the architecture of a novel distributed framework for dynamic and temporal graph processing. The proposed framework provides various tools for graph processing like graph storage, graph streaming and programming interfaces that allow the users to build concurrent and distributed time-dependent graph applications. In future works, we aim to evaluate this framework and compare it with existing systems in real world applications.

References

1. Aridhi, S., Montresor, A., Velegrakis, Y.: BLADYG: A graph processing framework for large dynamic graphs. Big Data Research (in press) (2017)
2. Aridhi, S., Nguifo, E.M.: Big graph mining: Frameworks and techniques. Big Data Research 6, 1–10 (2016)
3. Armstrong, T.G., Ponnekanti, V., Borthakur, D., Callaghan, M.: Linkbench: a database benchmark based on the facebook social graph. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. pp. 1185–1196. ACM (2013)
4. Candau, S., Bastide, J., Delsanti, M.: Structural, elastic, and dynamic properties of swollen polymer networks. Polymer Networks pp. 27–71 (1982)
5. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
6. Dhifli, W., Aridhi, S., Nguifo, E.M.: MR-SimLab: Scalable subgraph selection with label similarity for big data. Information Systems 69, 155–163 (2017)
7. Elser, B., Montresor, A.: An evaluation study of bigdata frameworks for graph processing. In: Big Data, 2013 IEEE International Conference on. pp. 60–67. IEEE (2013)
8. Gonzalez, J.E., Low, Y., Gu, H., Bickson, D., Guestrin, C.: Powergraph: Distributed graph-parallel computation on natural graphs. In: OSDI. vol. 12, p. 2 (2012)
9. Han, B., Liu, L., Omiecinski, E.: Neat: Road network aware trajectory clustering. In: Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on. pp. 142–151. IEEE (2012)
10. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: Proceedings of the 6th annual international conference on Mobile computing and networking. pp. 56–67. ACM (2000)
11. Kreps, J., Narkhede, N., Rao, J., et al.: Kafka: A distributed messaging system for log processing. In: Proceedings of the NetDB. pp. 1–7 (2011)
12. Lee, P., Lakshmanan, L.V., Milios, E.E.: Incremental cluster evolution tracking from highly dynamic network data. In: Data Engineering (ICDE), 2014 IEEE 30th International Conference on. pp. 3–14. IEEE (2014)

13. Li, X., Han, J., Lee, J.G., Gonzalez, H.: Traffic density-based discovery of hot routes in road networks. Advances in Spatial and Temporal Databases pp. 441–459 (2007)
14. Magalhães, R.P., Coutinho, G., Macêdo, J., Ferreira, C., Cruz, L., Nascimento, M.: Graphast: an extensible framework for building applications on time-dependent networks. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems. p. 93. ACM (2015)
15. Malewicz, G., Austern, M.H., Bik, A.J., Dehnert, J.C., Horn, I., Leiser, N., Czajkowski, G.: Pregel: a system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. pp. 135–146. ACM (2010)
16. Moon, S., Lee, J.G., Kang, M., Choy, M., Lee, J.w.: Parallel community detection on large graphs with mapreduce and graphchi. Data & Knowledge Engineering 104, 17–31 (2016)
17. Roy, A., Mihailovic, I., Zwaenepoel, W.: X-stream: Edge-centric graph processing using streaming partitions. In: Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles. pp. 472–488. ACM (2013)
18. Scholliers, C., Tanter, É., De Meuter, W.: Parallel actor monitors: Disentangling task-level parallelism from data partitioning in the actor model. Science of Computer Programming 80, 52–64 (2014)
19. Sengupta, D., Sundaram, N., Zhu, X., Willke, T.L., Young, J., Wolf, M., Schwan, K.: Graphin: An online high performance incremental graph processing framework. In: Euro-Par. pp. 319–333 (2016)