School of Informatics

Informatics Project Proposal
Benchmarking Graph Processing Systems

B138641
April 2019

Abstract

The aim of this project is to examine state-of-the-art graph processing systems and compare their performance and scalability in both local and parallel settings. More specifically, the project focuses on showing that often times scalability comes at a cost, and in many cases such an approach is not worth the time and effort. In our research we will prove this by conducting several experiments that will demonstrate that a single-core set-up outperforms executions on multi-core environments due to the parallelization overhead that incurs.

Date: Tuesday 2nd April, 2019

Tutor: Pablo León-Villagrá
Supervisor: Dr Milos Nikolic

Contents

1 Introduction

Modern computer applications and services have evolved to an extent that they have become difficult to manage. The amount of data handled in such applications is only growing while the industry and the research community have built new systems and infrastructures to be able to keep up with this growth rate. Additionally, applications and services now feature relational data, which are often expressed as graph structures, fact that makes things even more demanding and complicated.

To extract useful information and perform data analytics tasks, many different graph data processing systems have been developed. Although these systems have been extensively used in various applications, many people argue that their performance is questionable and that there is a massive overhead that incurs due to the way these systems scale.

Moreover, there is no clear indication of which system is appropriate for each occasion. For this reason, there have been many researchers focusing on creating benchmarking systems in order to evaluate the performance of such systems and observe their strengths and weaknesses.

While research on benchmarking graph processing systems has provided information about the performance of several graph processing systems, it tends to be quite specific and concerns certain aspects of these frameworks. This project was motivated by the need for a more thorough investigation that covers a broader range of cases and systems. Our research aims at showing that scalability is not always the best approach by performing various experiments under different conditions, that is, measuring performance using many different algorithms, systems and other parameters. The main goal is to demonstrate how in many cases running such tasks in single node settings is more efficient performance-wise than distributed and multi-node settings. Afterwards, if it is feasible time-wise, a second goal is to indicate how some tasks can be carried out with the use of higher-level systems, such as graph database systems.

We will now break down the problem and briefly mention some of its aspects along with how this project contributes to the domain of graph processing and why it is important that such an investigation is conducted. In Section 2 we will discuss necessary and relevant knowledge to the problem's domain. Afterwards, we consider specific parts of our research and analyse steps in Section 3. We then focus on the evaluation of our results which will be detailed in Section 4 and highlight what we expect to see in our results in Section 5. Lastly, we state a detailed report of the project's plan, its objectives and the deliverables in Section 6.

1.1 Problem Statement

While there is research on individual graph processing systems that measures performance under certain circumstances, there hasn't been extensive work that exposes a more objective benchmark for these systems' performance, such as one that features the use of a wide variety of algorithms, multiple set-ups that range from one to many cores, as well as more realistic use cases. Our work focuses on these aspects and will try to provide a more broad benchmark that will be able to better indicate if such systems are worth using and, if so, under which settings.

1.2 Research Hypothesis and Objectives

Our hypothesis is that graph processing systems do not scale as well as they claim to do for common tasks. We were able to identify these common tasks and other useful information form a survey [1]. Moreover, this survey revealed how the graph research community and the industry perform their analysis tasks, as well as what data they use, what kind of algorithms they run, and so forth; all these were very helpful for formulating our objectives.

We aim to support our hypothesis by conducting experiments that will demonstrate how the selected graph processing systems perform in single-core settings, as well as when scaled up, and hope to show through our results that for common tasks executions in single-core settings perform better.

As mentioned in Section 1, a secondary objective of our work will be to provide alternatives to approaches using less primitive systems, such as graph database systems. More specifically, graph processing has become more and more popular and has gained a lot of attention over the past few years. The people that get involved with such tasks are not necessarily researchers

of this field or experienced programmers, so the use of these primitive systems can be quite challenging. For this reason, many people use alternatives, instead, such as graph database systems or querying systems to conduct their work. Thus, examining such systems and comparing results against graph processing systems will also provide very useful information and will result to our work covering a broader range of cases, which of course also grows the size of our audience.

1.3 Timeliness and Novelty

Advances in technology nowadays have become more sophisticated and involve an extensive amount of data processing, a lot of which is expressed via graph representations. The popularity of this domain is constantly growing, given that more and more researchers and companies analyse graph data and perform graph processing tasks. Consequently, carrying out this research at this point is crucial since it will provide insights and indications as to how analyses and graph processing tasks can be performed, which will greatly impact the way researchers and developers approach their problems.

1.4 Significance

The more we know, the better we understand and the better we decide. As such, possessing information related to various aspects of a researcher's or developer's approach for graph data processing tasks is of great significance. The reason for this is that a thorough experimentation that covers many use cases, some of which will probably be similar to the intended task, will help in the process of the decision making with regards to the chosen technologies that will be used, the resources required, and others. This will result to better approaching a problem and providing more timely and efficient solutions.

1.5 Feasibility

While our task seems to be quite feasible, we will nonetheless present some factors that may constitute an obstacle to our research or may limit it to an extent in some of its aspects.

Firstly, an important factor to take into account regards the configurations of the graph processing systems used. In detail, while the single-core approach is fairly easy to implement, the scaled versions require more attention, as they demand more resources. The difficulty we might face concerns our resources, given that as it is, we they are limited to 8 cores. Obtaining access to a cluster environment or some computational nodes will significantly help our research, although, in case such thing cannot be arranged, our work will be limited to single-core experimentation.

Afterwards, with regards to the scope of our work, we have already briefly mentioned in Section 1 and Section 1.2 that in case we have enough time we will try to provide alternatives to the means of carrying out the graph processing tasks. Therefore, another part of our work that may or may not be feasible to complete is performing the graph processing tasks by using these graph database systems.

1.6 Beneficiaries

The outcome of this work will either implicitly, or even explicitly, impact the graph research community, as well as the industry, where developers and analysts use graph processing systems to perform their data analysis tasks. In detail, given that the results will provide insight as to whether a scaled approach that involves distributed graph processing systems is needed, or if a more simple, single-core, implementation is more preferable, researchers and small to medium sized companies can benefit from possessing such information, because it can potentially result to them saving time, effort, and money.

This is due to the fact that scaled approaches demand more resources, which often come at a big cost. Additionally, setting up and managing these resources requires a lot of time and effort. Therefore, having a priori knowledge about the resources needed for a specific task, or knowing the performance of an algorithm that might be similar to the one being developed, for instance, can benefit a project's planning and budget.

2 Background

As we already mentioned in Section 1, it has been a while since the research community and companies in the industry have been dealing with immense amounts of data. Today's approaches feature the use of distributed big data processing systems, given that other solutions may be extremely slow in terms of the time needed to process the data or carry out the computational tasks. Graph computation problems account for a big partition of the different challenges that are faced. For this reason, several systems have been built that specialise in heavy graph data processing tasks. The systems that will be considered in our research (Section 3.3) can be split into two categories; distributed graph data processing systems, and graph database systems. We will now consider both of these types of systems and discuss some of their aspects that are relevant to our research.

2.1 Distributed Graph Data Processing Systems

TODO

*** such as Apache Flink Gelly [2], Apache Spark GraphX [3, 4], Apache Giraph [5], GraphLab [6], PowerGraph [7], Naiad [8], etc.

2.2 Graph Database Systems

While specialised graph data processing systems offer a solution to graph computation tasks, usually individuals have to familiarise themselves with the programming model that the system follows, as well as spend time to learn its programming interface in order to fulfill their tasks. To tackle interaction with low-level primitive systems, other systems were developed, offering a more high-level solution that didn't involve getting this technical in order to complete a graph computational task.

3 Methodology

We will now present the way by which our research will be conducted; that is, how we plan to approach the problem stated in Section 1.1, as well as the steps that will be followed in order to fulfill our project's purpose.

3.1 Approach

At first, we discuss on a high level how we aim to solve the problem. The core idea is to perform several graph data processing tasks, with a number of different datasets, using several processing systems. Given that there are numerous different systems, as well as algorithms, the number of cases to examine is extremely large. For this reason, we consider a set of each category, based on the popularity and use of each system, as well as the types of tasks carried out, both of which were retrieved from [1].

We now move on to mentioning specifics with regards to the datasets, systems and tasks that will be under consideration in Section 3.2 Section 3.3, and Section 3.4, respectively.

3.2 Datasets

Given that we aim at covering as many cases as possible, in order to provide broad and useful insights through our work, we consider several datasets with different characteristics; that is, different number of nodes, edges, edge distributions, etc. The datasets chosen are publicly available and are commonly used within the graph research field. The datasets and their characteristics that we will use to conduct our research are as follows:

Dataset	Nodes	Edges	Description
ego-Gplus [9]	107,614	13,673,453	Social circles from Google+
soc-Pokec [9]	1,632,803	30,622,564	Pokec online social network
cit-Patents [9]	3,774,768	16,518,948	Citation network among US Patents
twitter [10]	41,700,000	1,470,000,000	Twitter follower network

Table 1: Datasets considered along with their characteristics

Additionally, if needed, we might consider generating datasets with different distributions for their edges, for example, uniform, skewed, and others.

3.3 Graph Processing Systems

Initially, we aim at examining the graph processing systems stated below. Note that they are mentioned in order of interest, that is, ... ***

- Timely Dataflow *** add citation
- GraphLab *** add citation
- Apache Flink (Gelly) [2]

- Apache Spark [3]
- Apache Giraph [5]

The basis for our selection of graph processing systems is their popularity [1] and ***.

Afterwards, a secondary goal is to also carry out our tasks using graph database systems. In case we have time to consider this category of systems, we hope to test at least one of the following systems. Again, we order the systems based on how interested we are in testing them ***.

- EmptyHeaded [11] [12]
- Neo4j [13]
- OrientDB [14] (maybe)
- ArangoDB [15] (maybe)

3.4 Algorithms

Similarly to the choice of graph processing systems, our choice of algorithms is based on the popularity of the type of task [1]. As such, we will consider the following types of computations:

*** Add examples for each type of algorithm

- Finding connected components
- Neighborhood queries
- Finding short or shortest paths
- Subgraph matching
- Ranking and centrality scores
- Aggregations
- Reachability queries

3.5 Environment Settings

We aim at providing two types of environment settings for our experimentations. Firstly, all experiments will be executed on a single-core setting. Afterwards, depending on the resources, experiments will also be executed in scaled environments, which will consist of multiple nodes. The scaling factor cannot be determined yet since it is dependent on the amount of resources that we will have in our availability.

3.6 Limitations

Currently,

4 Evaluation

After having completed all steps and tasks mentioned in Section 3,we will carry on to analysing and interpreting our results.

TODO.

Question(s):

- Provide exact evaluation metrics now?
- Should I focus on throughput as well or just **latency**?

5 Expected Outcomes

We will now review what we hope to achieve through our research and discuss how our contribution will impact the graph research community, as well as companies or individual developers that perform graph data processing tasks.

TODO.

6 Research Plan, Milestones and Deliverables

TODO.

References

[1] Siddhartha Sahu, Amine Mhedhbi, Semih Salihoglu, Jimmy Lin, and M. Tamer Özsu. The ubiquity of large graphs and surprising challenges of graph processing. *Proc. VLDB Endow.*, 11(4):420–431, December 2017.

[2] Apache Flink. `https://flink.apache.org/`. Accessed: 2019-04-01.

[3] Apache Spark Graph X. `https://spark.apache.org/graphx/`. Accessed: 2019-04-01.

[4] Joseph E. Gonzalez, Reynold S. Xin, Ankur Dave, Daniel Crankshaw, Michael J. Franklin, and Ion Stoica. Graphx: Graph processing in a distributed dataflow framework. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 599–613, Broomfield, CO, 2014. USENIX Association.

[5] Apache Giraph. `https://giraph.apache.org/`. Accessed: 2019-04-01.

[6] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M. Hellerstein. Distributed graphlab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5(8):716–727, April 2012.

[7] Joseph E. Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, and Carlos Guestrin. Powergraph: Distributed graph-parallel computation on natural graphs. In *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pages 17–30, Hollywood, CA, 2012. USENIX.

[8] Derek G. Murray, Frank McSherry, Rebecca Isaacs, Michael Isard, Paul Barham, and Martín Abadi. Naiad: A timely dataflow system. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, SOSP '13, pages 439–455, New York, NY, USA, 2013. ACM.

[9] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600, New York, NY, USA, 2010. ACM.

[11] Christopher R. Aberger, Andrew Lamb, Susan Tu, Andres Nötzli, Kunle Olukotun, and Christopher Ré. Emptyheaded: A relational engine for graph processing. *ACM Trans. Database Syst.*, 42(4):20:1–20:44, October 2017.

[12] EmptyHeaded. `https://github.com/HazyResearch/EmptyHeaded`. Accessed: 2019-04-01.

[13] Neo4j. `https://neo4j.com/`. Accessed: 2019-04-01.

[14] OrientDB. `https://orientdb.com/`. Accessed: 2019-04-01.

[15] ArangoDB. `https://www.arangodb.com/`. Accessed: 2019-04-01.