UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Incremental Algorithm for large Netwotks

Project management (GEP)
Deliverable 1: Context and scope of the project

Pol Forner Gomez

Thesis supervisor: Gerard Escudero Bakx (Department of Computer Science)
Thesis co-supervisor: Edelmira Pasarella Sanchez (Department of Computer Science)
GEP tutor: Joan Sardà Ferrer
Degree: Bachelor's Degree in Informatics Engineering (Computing)
February, 2024

Contents

Context

This work is a bachelor degree of a computer engineering degree, specialization in Computing. The degree is done in the Facultad d'Informatica de Barcelona (FIB) of Universitat Politàcnica de Catalunya (UPC) and is directed by Gerard Escudero Bakx and supervised by Edelmira Pasarella Sanchez.

This work is based on the master thesis made by Juan Pablo Royo Sales in 2021 [1] and the objective is to improve the algorithm that he made. And that is why I would like to introduce some concepts that are important to understand the basics of the Juan Pablo work. Here, only will be explained the concepts that are important to my work, summarizing and adding some notes to Juan Pablo's work.

1.1 Introduction: Concepts

Nowadays, the amount of data that has been generated is huge and more important, it is growing every day. Sensors, social networks, and other sources generate data that needs to be processed and analyzed to extract useful information. The main problem is that when we try to process this data we can not use the traditional methods we use with small amount of data, because the time and resources needed are too high. To solve this problem, we need to develop new algorithms and techniques that can deal with.

Streaming

Besides this, the amount of data normally it can not be stored and it comes in what is called a data stream. A data stream is a sequence of data that made available over time, meaning that we can not store all the data and we need to compute it in time. For example, a sensor of temperature that sends the temperature every second, a traffic camera that registers all car plates that pass in front of it or just a social network that generates a huge amount of data every second.
This kind of data mentioned before needs to be processed and sometimes the data never ends, so we can not wait to finish to give a result and we must give results along the way.

Incremental algorithms

Here is where incremental algorithms come in.[2] Incremental algorithms give us the ability to obtain results from subsets of data and then update the results before finishing the whole data. This is very useful, because some problems do not need to be solved with all the data or maybe we are not interested in the final result. Recovering a previous example, if we are interested in which models of cars drive in certain road, we can use the camera that registers the car plates to get the answer. It is stupid to wait until the end of data to give the answer (also because it never ends), so we can give a result when we check it. In conclusion, incremental algorithms could be a good approach to solve some problems.

Paralelism

One of the most important techniques for dealing with time problem is parallel computing or parallelism. Parallelism allows us to divide the work and process it in different machines concurrently, reducing the time needed to process the data. When we try to fight against this huge data problems, we must find a solution that can be parallelized given that modern machines have multiple cores and we can use them to process the data.
When we put together streaming and parallelism, we can distinguish two computational models:

- **Data Parallelism**
  This model splits the data and processes it in parallel. All the computations that perform some action over a subset of data, do not have any dependency with other parallel computations. This model has the advantage that it can implement stateless algorithms, allowing to split and process the data into different machines without contextual information. Nonetheless, this model has the disadvantage that when we need to be aware of the context, it is penalized.

- **Pipeline Parallelism:**
  This model splits the computation in different stages and each stage takes the result of the previous stage to make the computation. The parallelization is done by parallelizing the stages. The main advantage is that stages are non-blocking, meaning that we do not need to process all data to execute the next stage. This allows us to make incremental algorithms. In spite of that, the main disadvantage is that one stage could be the bottleneck of the pipeline and delaying all the process.

Dynamic Pipeline Paradigm

Now that we talked about these 3 concepts: incremental algorithms, streaming and parallelism, we can introduce the next concept that we are going to be working in this project.
The dynamic pipeline Paradigm is a Pipeline Parallelism model "*based on a one-dimensional and unidirectional chain of stages connected by means of channels synchronized by data availability*". [1, Page 9, 2.2] This chain is called Dynamic Pipeline and it can grow and shrink. ...

1.2 Problem to be solved

In his work, Juan Pablo implemented an incremental algorithm using dynamic pipeline paradigm to resolve a graph problem: finding bitriangles in bipartite graphs. He decided to implement the algorithm using the functional programming language Haskell, and because of the no existence of one, he created a framework to implement the dynamic pipeline paradigm. Here is where my project comes in, I will use his framework to implement an easier problem to understand the dynamic pipeline paradigm and then I will try to improve the algorithm that he made.

As said before, the algorithm to improve finds bitriangles in bipartite graphs and here i will not explain the problem as I consider that is not important. If you are interested in the problem, you can read Juan Pablo's work [1, Page 5, 1.1].

The easier problem that I chose for learning about the Haskell framework is the word counting problem. As easy as it sounds, the problem does not need further explanation: we just need to count the number of occurrences of each word in a dataset.

1.3 Stakeholders

This project is co-supervised by Edelmira Pasarella Sanchez, who is a researcher and supervised Juan Pablo work. She and her team developed the algorithm so they are the main stakeholders of this project. Also the director of this project, Gerard Escudero Bakx, is the one who proposed me to do this project because he was interested in the Haskell framework made by Juan Pablo. So Gerard is also a stakeholder of this project.

Apart from these direct stakeholders, this project will help to add knowledge to the community about the dynamic pipeline paradigm code examples and more important, will add more code and knowledge to the world about the Haskell framework of Juan Pablo.

Justification

Well first we need to ask ourselves if is it worth or necessary to improve the algorithm. As I do not really know well about the algorithm, I talked with Edelmira and discuss some points where we can improve the algorithm. Their team has done some improvements since Juan Pablo's work, and also tell me about some weak points of the algorithm. So yes, I think that is worth to improve the algorithm.

With my actual knowledge of the algorithm, I can not tell if it is better to improve the algorithm or to make a new one. But another time Edelmira told me that the algorithm was good and that it was worth to improve it.

Another point to take into account if it is worth to use the Haskell framework made by Juan Pablo. This may not be clear now, because this framework is unique and there are no other examples of it. This work can be very useful for testing the framework and extracting a conclusion about this.

Scope

3.1 Objectives

The main objective of this work is to improve the incremental algorithm made by Juan Pablo for find bitriangles in large networks. This project also have another important objective: make an implementation using the Haskell framework of the word counting problem. To achieve these two objectives, I will be following the next sub-objectives:

First algorithm: word counting

Research and Learning

This first sub-objective goal is to learn the basics for complete the objective. This will be the main topics to research and learn:

- Reseach in dinamic pipeline paradigm to understand the concept.
- Refresh the basics of Haskell, because i have not used it for a long time.
- Estudy Juan Pablo Haskell framework that he created and used in his work.

Code scaffold

Like all coding projects, we need to create a good scaffold before start coding, special in this project. This is because we do not know anything about dynamic pipeline paradigm and we will need to think all the agents and stage.

Coding

All left will be to code the algorithm.

Second algorithm: finding bitriangles

Research and Learning

Here I will need a different way to learn and research as the concepts will probably be more complex. But I can take advantage of Edelmira help to guide me. This will be the main topics to research and learn:

- Research the basics of the algorithm for finding bitriangles in bipartite graphs.
- Understand the algorithm of Edelmira and her team.
- Understand the implementation of Juan Pablo.

Find weak points

Here I will need to find the weak points of the algorithm and try to improve them. As I will not have a lot of time, Edelmira and Gerard will be key to guide me in this part. I have already spoken with Edelmira and she has given me some ideas to improve the algorithm.

3.2 Potential obstacles and risks

Like all projects, you always have to take into account the potential obstacles and risks that can appear. Here are the principal ones that I have identified:

3.2.1 Time Limit

I will say that this is one of the most important risks, because as there is a deadline to finish the project, an inconvenience can make to not finish the project. Despite this, I this that I have done a good objective planning and I will be able to redirect the project if any problem appears.

3.2.2 Dynamic pipeline framework

I did not use and examined the framework and a problem that worries me is that the framework has some bugs or is not well implemented. This can potentially make me lose a lot of time but as I check, Pablo did a good work documenting it so I trust that I will not have a lot of problems.

Methodology and rigor

4.1 Methodology

For this project, I have one main issue: I'm a intern in a company in the mornings and I am also doing some subjects of the degree. This means that I have only few moments in the week to work in the project, in concrete, Mondays and Tuesday in the afternoon. For this reason, I decided that I will be doing a simplified version of the Scrum methodology.[3]

In this simplified version, I will be using my 2 days available to full work and the project and then I will use the weekends to make the planning and review. The idea is to use the start of the weekend to review the work done and check if it is need for more time to finish the goals. After that, I will document all the work done to maintain and check the progress and planning of the whole project. At the end of the weekend, I will make the planning for the next week having in mind the possible problems that I could have during the week. This periodical system is called sprints and I this is the best way to work for my situation.

Also, to be able to obtain external feedback, I will be schelude meetings with Gerard no less that once every 2 weeks.

4.2 Project monitoring and validation

As good programming practices, I will be using git [4] to control the versions of the code and I will be using a Trello board to control the tasks and the progress of the project. This will secure me to, in one part by using git to have always a backup of the code and a practical way to a previous versions. And in the other part, by using Trello to have a good control of the tasks and the progress of the project, because of the methodology that I chose to use and the good results that I have had in the past using it.

Bibliography

[1] E. Pasarella and M.-E. Vidal, "FACULTAT d'INFORMÀTICA DE BARCELONA (FIB) UNIVERSITAT,"

[2] A. M. Sharp, *Incremental algorithms: solving problems in a changing world*. Citeseer, 2007.

[3] "What is scrum? — scrum.org." (), [Online]. Available: `https://www.scrum.org/resources/what-scrum-module` (visited on 02/27/2024).

[4] "Git." (), [Online]. Available: `https://git-scm.com/` (visited on 02/27/2024).