Abstract

[Place holder abdtract]

Contents

1 Introduction **2**

2 Background **3**
2.1 Glossary of terms . . . 3
2.2 Network . . . 3
2.2.1 The small world effect . . . 5
2.2.2 Transistivity/clustering . . . 5
2.2.3 Degree distribution . . . 5
2.2.4 Network resilience . . . 5
2.2.5 community structure . . . 6
2.3 Sparce Matrix . . . 6
2.4 Small-world model . . . 6
2.5 Data diffusion . . . 6
2.6 Basic Diffusion Models . . . 6
2.7 Breadth First Search as a matrix multiplication. . . . 7
2.8 SIR/SIS . . . 7
2.9 greedy algorithm . . . 8
2.10 R-mat . . . 8
2.11 Cache oblivious model . . . 9

3 Related works **10**

4 Methode **11**
4.1 R-mat . . . 11
4.2 Adjacency matrices . . . 11
4.3 The algorithm . . . 11

5 Result **12**

6 Discussion **13**

7 Conclution **14**

List of Figures

List of Tables

2.1 Table of usefull glossary 4

Chapter 1

Introduction

Using graphs as representation and algorithms that traverse graphs are common through many scientific domains[1]. We can use graphs to simulate how a disease would spread through a social network, we can predict how efficient viral marketing is and we can see how a new trend would spread through a community[2].To select the most influential nodes for a information diffusion through a social network is an NP-hard problem[2]. One of the common ways to select them, is by brute forcing through the network with algorithms like BFS. The algorithm is a greedy hillclimbe algorithm[CITATION NEEED]. The greedy algorithm, proposed by[KEMPE] goes through all the nodes and computes the effect on the network that nodes had. The greedy algorithm takes the k top most influential nodes as the starter node.

The problem that we will explore with this report, is if there is a way to optimize the seed-selection of the most influential nodes in a graph. The report will focus on how to use hardware to achieve an optimization to this problem. One idea is to try to use the irregular memory access that is requested during a BFS search.

Chapter 2

Background

In this chapter, we will look at the fundamental concepts and background information needed to understand the the different infections model, problem with seed-selection for social networks, and solving Breadth-first search using matrix multiplication. This chapter will contain notations that we would use throughout the report and used in the field.

2.1 Glossary of terms

2.2 Network

A *network* is a collection of *Vertices*, commonly known as nodes with *edges* connecting them together[3]. The edges serves as a connection, or a "bridge" between the nodes, while the nodes can reprecent something, or containing information. In the real world, multiple systems takes the form of networks around the world, examples are the internettt, the World Wide Web, social media like Facebook, twitter etc. There are different types of network or *graphs*. These include from *socialnetwork*, *sinformationnetworks*, *technologicalnetworks* ,and *biologicalnetwork*. Each of them have a different properties, but we will focus more about social network.

A social network is a set of people connected to each other via some form of contact or interactions[3]. The nodes are people while the edges are the connections between peoples. The social network display information regarding connection, interaction or location of a set of people.It forms patterns regarding friendships, business interactions between companies and families history/ ancestral tree. The social network is often used in social science[3]. Some noteable experiments are [**?**], which looks at the small world problem. The small world problem can be summarized as: "what is the probability that any two people, selected arbitrarily from a large population, such as that of the United States, will know each other?". [4]. This in itselfe is not that interesting, the [**?**] takes that altho person a and z does not know each other, do they have a set

Table 2.1: Table of usefull glossary

Vertex(*pl.vertices*): The fundamental unit of a network, also called a site(physics), a node (computer science), or an actor (sociology).
Edge: The line connecting two vertices. Also called a bond (physics), a link(computer science), or a tie (sociology).
Directed/undirected: An edge is directed if it runs in only one direction (such as a one-way road between two points), and undirected if it runs in both directions. Directed edges, which are sometimes called arcs, can be thought of as sporting arrows indicating their orientation. A graph is directed if all of its edges are directed. An undirected graph can be represented by a directed one having two edges between each pair of connected vertices, one in each direction.
Degree: The number of edges connected to a vertex. Note that the degree is not necessarily equal to the number of vertices adjacent to a vertex, since there may be more than one edge between any two vertices. In a few recent articles, the degree is referred to as the connectivity of a vertex, but we avoid this usage because the word connectivity already has another meaning in graph theory. A directed graph has both an in-degree and an out-degree for each vertex, which are the numbers of in-coming and out-going edges respectively.
Component: The component to which a vertex belongs is that set of vertices that can be reached from it by paths running along edges of the graph. In a directed graph a vertex has both an in-component and an out-component, which are the sets of vertices from which the vertex can be reached and which can be reached from it.
Geodesicpath: A geodesic path is the shortest path through the network from one vertex to another. Note that there may be and often is more than one geodesic path between two vertices.
Diameter: The diameter of a network is the length (in number of edges) of the longest geodesic path between any two vertices. A few authors have also used this term to mean the average geodesic distance in a graph, although strictly the two quantities are quite distinct.

of individuals $\{ b_1, b_2, ...b_n\}$ who are mutual friend or even a "chain" of such individual($a - b - c - ... - y.z$).

Some properties that networks often exhibits are the small world effect, transitivity or clustering, degree correlations, and community structure[3].

2.2.1 The small world effect

The small world effect was first demonstrated by Stanley Milgra in the 1960s during his famous letter passing experiment[5]. The experiment was about passing letters from person to person to reach a designated target with only small steps. For the published case, the chain was around six [6]. This shows us that for most pair of vertices can reach each other with a short parth. A more precise wording is that "Networks are said to show the small world effect if the value of l scales logarithmically or slower with the network size for a fixed mean degree.". [3]. We have defined l to be the mean geodesic distance between vertex pairs in a network.

2.2.2 Transistivity/clustering

Transistivity, also sometimes called clustering is the properties that shows that there is a high number of triangles in the network. A triangle in a network is if node A is connected to node B and node C, and node B and C is also connected to each other, thus creating a triangle. In social network, we say that a friend of your friend is likely to be your friend too[3]. The transistivity is often used to show *networkdensity*.

2.2.3 Degree distribution

Degree distribution is the histogram of degrees of vertex. For random graphs, the distribution would most likely be a poiison distribution or binomial. For real world network, the distribution would often have a highly rightskewed distribution. Resulting in that the distribution has a long right tail of values. The probability P_k is the probability that the new vertex v have the degree k.

2.2.4 Network resilience

For most network model, there is the need to remove nodes from the network. Removal of a node can have no effect on the network as a whole, or it can be devastating. Network resilience looks at how the network can ressist to such a removal. There are two different removal scheme, the random removal where nodes are randomly picked and removed, or the targeted removal where specific nodes are removed depending on the critteria.

The experiment mentioned in [**?**] by Albert *etal* showed that for a subset of network representing the internett and the world wide web, targeted removal had a larger impact then random removal. The targeted removal removed the highest degree nodes from the network, and the random removal removed nodes

randomly. The random removal had a minimal effect on the network, while the targeted removal had a much larger impact, the mean vertex-vertex distance increased. They proposed that the internett was highly resilient to random removal, while much more vuarneble to targeted removal.

There are other studies that proposed a different interpetation about the data found.

2.2.5 community structure

One properties that is often observed in a social network, is the community structure. The community structure is where a group of vertecies having high density of edges with each other, while having lowe density of edges to other "community". We can see an example of the community structure clearly displayed from [**?**]

2.3 Sparce Matrix

A sparce matrix is an matrix with few elements in them. The social network can often be reprecented with a sparce matrixs

2.4 Small-world model

2.5 Data diffusion

Data diffusion is looking at how information is propagated through a network or a graph. An example would be how a new Internet meme, a new trend or how a new disease is spread through a community. The process consist of a set of starter nodes that are "infected", during each time-step, there are a chance that the "infected" node would "infect" its neighbor. To start such a simulation, we would need to first pick out a set of initial "starter node". These k starter node is a set of node that in the initial time-step is infected. They will pass on the information/infection during each time-step and the information/infection will propagate through the network.

2.6 Basic Diffusion Models

When we talk about information propagation, we can look at how medical and technological innovations is spread through a social network. We can simulate those kind of behavior with different diffusion models. There are two basic diffusion models used to simulate the propagation of information through a network[2], the *linearthresholdmodel* and the *independentcascademodel*[2].

This process can be simulated by looking at a social network and how information propagates through the network. We look at each node in the graph as a person, they can be either active, or inactive. The activation of a node

depends on which diffusion model we choose. A active node is "infected", while the inactive is the "healthy" ones. The activation of each nodes is dependent on which model we pick.

The linear threshold model uses a threshold θ_v between the interval [0,1], which represent the faction of $v's$ neighbors that need to be active to activate node v. The Linear Threshold Model activates the current node v when the weight $b_{v,w} > \theta$ from its neighbor w outweighs the θ_v. This is more in line with the situation were each person have a chance to adopt to a new trend if exposed to the trend enough time by his close friends. An example would be a product promoted on social network like twitter and Facebook. The user will adopt the new trend if he is exposed to it from enough friends or idols.

The independent cascade model changes states with a probability $p_{v,w}$, where v is current node, and w is it's neighbor. During each propagation, the node v have a $p_{v,w}$ chance to change state if the neighbor w changed state. If during time-step t a node v changed state, its neighbor w would have a $p_{v,w}$ chance to change state in the next time step. An example here would be spread of a disease. The current node$v$ have a chance(θ_v) to infect its neighbor

2.7 Breadth First Search as a matrix multiplication.

By looking at the connectivity matrix, we can see that a connectivity matrix is a graph represented in a matrix format. The BFS can be achieved by looking at the BFS operations as a matrix multiplication [7]. If we look at the graph as a connectivity matrix, we can apply matrix multiplication to generate the breadth first search.

Breadth first search is an tree traversal algorithm. BFS start at the root node r, or any arbitary node in a n tree, and stores all its child node in an *queue*. The algorithm then takes the first node from the queue, v_1 and stores all the child node to v_1 in the back of the queue, this process continous until the queue is empty and all the nodes have been iterated over. From the psaudocode we can see:

We can also look at how an breadth first seaerch can be executed by appllying sparce matrix multiplication. We can see that teh adjacency matrix is an sparse matrix

2.8 SIR/SIS

There are two different epidemic model that we will be looking at, the *SIR* (susceptible, infected, removed,) and the SIS mode(susceptible, infected, susceptible).Both is used to simulate how a disease or an epidemic can spread through the general population, in this case, we can use this

The SIR model stands for susceptible, infected and removed. The node would have three different states, the susceptible state mean that the node is

Algorithm 1 Breadth First Search

```
1: dist[∀v ∈ V] = −1; currentQ, nextQ = ∅
2: step = 0; dist[root] = step
3: ENQUEUE(nextQ,root)
4: while nextQ ≠ ∅ do
5:     currentQ = newxtQ; nextQ = ∅
6:     step = step + 1
7:     while currentQ ≠ ∅ do
8:         u = DEQUEUE(currentQ)
9:         for v ∈ Adj[u] do
10:            if dist[v] == −1 then
11:                dist[v] = step
12:                ENQUEUE(nextQ, v)
   return dist
```

susceptible to the disease, or in this example, change state. The infected state is the state were the node v is infected. The state Removed/Recovered, is the state were the node v have the disease removed, or have recover from the infection. In this model, the Removed/Recovered state is not susceptible to the infection again.

The SIS stands for susceptible, infected and susceptible. Unlike the SIR this model can reinfect the recovered nodes. This model is used for simulation of outbreak of disease.

2.9 greedy algorithm

The greedy algorithm was proposed by [KEMPE]. The algorithm is a greedy hillclimbing algorithm, the algorithm fins the starter set of nodes S by iterating through all the vertex in V and calculate the total amount of spread. The spread is saved and the k most influential nodes would be choosen. @@@INSERT PSAUDOCODE HERE@@@@@@

Another popular algorithm is the degree algorithm. Unlike the greedy algorithm, the degree sort all the node acoring to their degree distribution. The algorithm picks the top k nodes according to the degree distribution.

The last one is the random algorithm. The random algorithm just pick a random starter node.

2.10 R-mat

One problem during graph analysation and calculation is finding suiteble graphs to analyse. to generate graphs with desirede properties is not easdy to do. One solution proposed p [8] is to use the /"recursive matrix /" or R-mat model. The R-mat model generates graph with only a few parameters, the graph will naturally have the small world propertie and follows the laws of normal graphs,

and have a quick generation speed. The R-mat models goal is to generate graphs that matches the degree distribution, exhibits a " community " structure and have a small diameter and mathces other criteria.[**?**].

The algorithm to generate such a recursive matrix is as follow: The idea is to partition the adjacency matrix into four equally sized part branded A,B,C,D. Each new edge is "Droped" onto the adjacency matrix. Wich section the edge would be placed on is choosen randomly. Each section have a 'probability of a, b, c, d, and $a+b+c+d=1$. After a section is choosen, the partion that was choosen is partitioned again. This continoues until the choosen section is a 1x1 square and the edge is dropped there.

@@ INSET PICTURE OF PARTITION HERE!

From the algorithm, we can see that the R-mat generator is capable to generate graphs with total numbers of node $V = 2^x$. Since the algorithm partitioned the matrix into four part.

2.11 Cache oblivious model

[REWRITE THIS PART] Chache oblivious have examples where cache oblivious is better for, sparse matrix multiplication. for a tree iteration algorithm, better.

Chapter 3

Related works

RElated work, there are multiple people working on this kind of problem. One of the more well known is kemnp, where he tries to solve how to maximize the spread.

Other solution include the hybrid FPGA-CPU arcitecture, which reduces teh computationtime for a bfs over a small world graph by XX%[**?**]

Chapter 4

Methode

For this report, we created a python simulation to simulate the data diffusion and the seed selection. We utilized the Pycx libraries to create the GUI, and applied the R-mat generator to create the different network.

4.1 R-mat

The R-mat generater generates sociall network with the community structure. The different probability for the four partition is :$A = 0.57$,$B = 0.19$,$C = 0.19$,$D = 1 - A - B - C = 0.05$. These different probability was used by GRAPH500[CITATION NEEDED]. The R-mat generated three different adjacency matrices.

4.2 Adjacency matrices

For this report, four different sized adjacency matrix was created. One of the restriction to the R-mat generator is that the size of the adjacency matrix have to be 2^n. This resulted in that our adjacency matrix was of the size, $128 \times 128, 512 \times 512 and 1024 \times 1024$.

4.3 The algorithm

The simulation creates an social network by reading from the adjacency matrix.

Chapter 5

Result

From the several runs we got the results from all the run. The

Chapter 6

Discussion

One paper showed how implementing a parallalized bfs algorithm on a distributed system reduced the communication times by a factor of 3.5, compared to a common vertex based approach[9]. In scientific application, ,graph based computation is pervasived and often data-intensive, this especially for distributed systems.

ISNERT INFORMATION

It would be interesting to see how the parallelized BFS would improve the computation time for our simulation.

Chapter 7

Conclution

As we have seen here the

Bibliography

[1] Magnus Jahre Yaman Umuroglu, Donn Morrison. Hybrid breadth-first search on a single-chip fpga-cpuheterogeneous platform. 2015.

[2] Éva Tardos David Kampe, Jon Klein. Maximizing the spread of influence through a social network. 2003.

[3] M. E. J. Newman. *The structure and function of complex networks*. 2003.

[4] Stanley Milgram Jeffrey Travers. An experimental study of the small world problem. 1969.

[5] Stanley Milgram. The small-world problem. 1967.

[6] Stanley Milgram Fjeffrey Travers. An experimental study of the small world problem. 1969.

[7] John Gilbert Jeremy Kepner. Graph algorithms in the language of linear algebra. 2011.

[8] Christos Faloutsosx Deepayan Chakrabartiy, Yiping Zhanz. R-mat: A recursive model for graph mining. 2004.

[9] Kamesh Madduri Aydib Buluc. Parallel breadth-first search on distributed memory systems. 2011.