Large Scale Parallel Data Processing
Spring 2019

Graphs and MapReduce

Team Members:

1. Kousthubh Belur Sheshashyee
2. Rachna Reddy Melacheruvu

Project Overview

Perform the below two tasks:

- Explore the degree of separation between two people in the twitter data dump
- Find cycles of length K in a large graph

Degree of Separation

- Given a person in the twitter dataset, the result displays their 1st connections, 2nd connections, 3rd connections and so on based on the input(K) provided
- This also provides a way of understanding if a person can be reached through one of his/her connections on the twitter

Twitter Dataset

- The dataset is the friendship/followership network among the users of the twitter social website. The friends/followers are represented using edges and edges are directed. For example - 1,2. This means, user id “1” is following user with id “2”
- The dataset consists of 11316811 nodes and 85331846 edges in total

Algorithm

Single Source shortest path using Breadth First Search(BFS)

- Job 1: Adjacency list generation

  Ex: s a,c

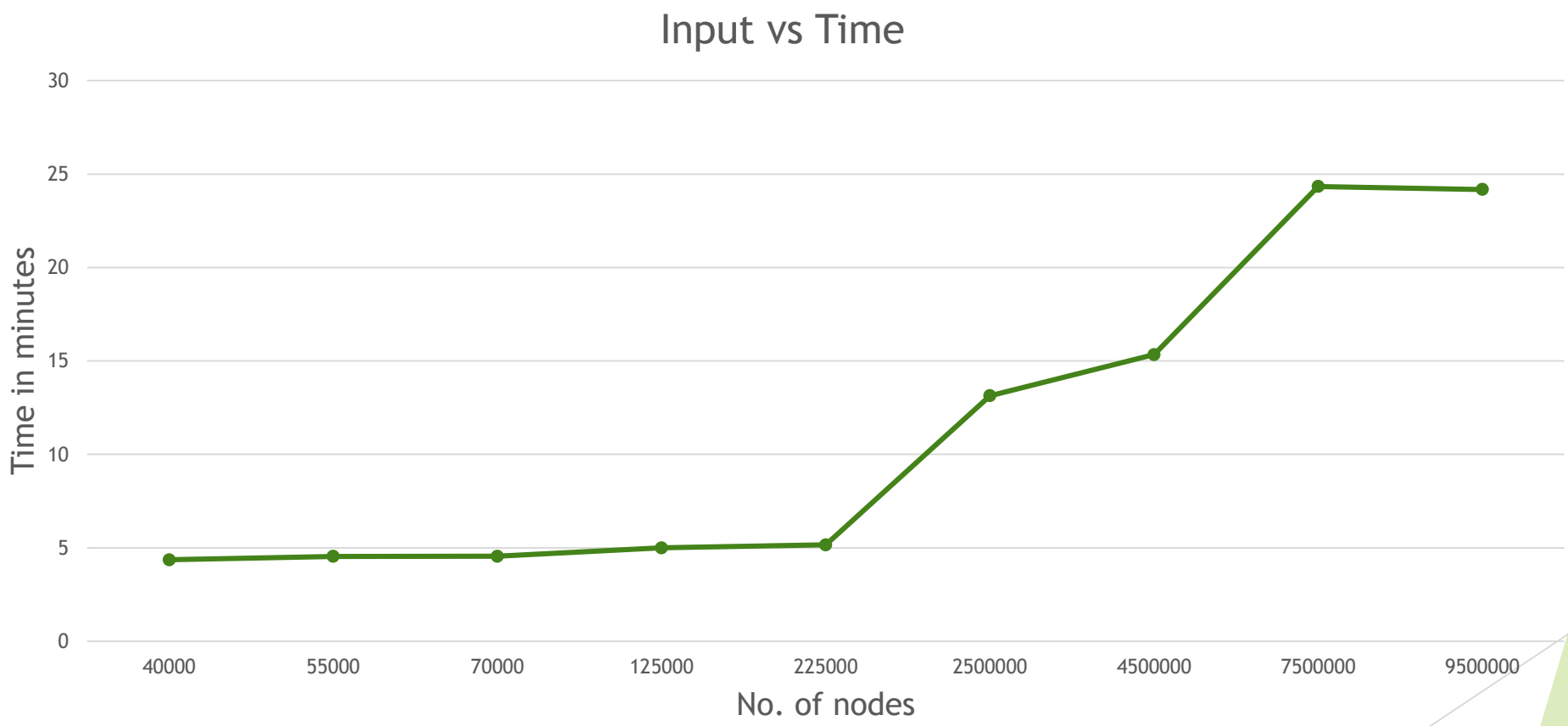
  a b,d

  b a

- Job 2: Runs BFS for K iterations

	Intermediate output		Output connections	
1st iteration:	s	a,c#s#0	a	s
	a	b,d#s#1	c	s
	b	a#null#∞		
2nd iteration:	s	a,c#s#0	b	a<-s
	a	b,d#s#1		
	b	a#a<-s#2		

Experiments



Cont…

- It can be seen from the above graph that up to 225k nodes, increase in running time is not very evident
- Once the input size increases above 250k, the running time increases exponentially
- This is because the size of intermediate output increases exponentially after a certain point which results in worker machines taking longer time to process the intermediate output files

Results

Cluster Size	Number of nodes	Time taken
Small (5 workers)	11316811 (full input size)	32min 48sec
Large (10 workers)	11316811 (full input size)	23min 52ec

Note on Speedup

- Speedup achieved = 1.37
- Theoretically running time on 11 workers should be half the running time on 5 workers
- The adjacency list is not uniform as for few of the nodes, the adjacency list is very huge
- Load on worker node is not distributed equally because of data skewness in the adjacency list

Interesting Structures in a Graph

- Find cycles of length k in a graph
- The dataset consists of 8.4M nodes and 25.2M edges with a maximum degree of 28

Algorithm

- Job 1: Adjacency list generation

  Ex: s a,c

  a b,d

  b a,s

- Job 2: Runs BFS for K iterations

Intermediate output:

1st iteration: s a,c/active:b

a b,d/active:s:b

b a,s/active:a

2nd iteration: s a,c/active:a->b

a b,d/active:b->s:a->b

b a,s/active:s->a:b->a

Experiments

Dataset #	Maximum degree	Number of edges	Comments about the run
1	40	63.5M	600sec time out error
2	13	54.1M	No cycles
3	18	42.7M	Java heap space error
4 (chosen dataset)	28	25.2M	Successful run without errors

Challenges

- Finding the appropriate dataset having the following feature:
  - Big data
  - Max degree between 15 and 30
  - Presence of cycles
- Got 600sec time out error for most of the datasets
  - Reducer task was taking too long
  - Removal of existing sub-paths in the newly explored path in each iteration
- Works best for sparse graphs compared to dense graphs where each node has a large maximum degree

Result

Cluster Size	Number of edges	Time taken
Small (5 workers)	25.2M (full input size)	59min 49sec
Large (10 workers)	25.2M (full input size)	38min 13sec

Thank You!