Randomized Algorithm for shape pattern counting on GraphX

independent project proposal

Supervisor: Prof.Ke YI, Student: GENG XU(20405672)

September 5, 2016

Introduction and Scope

Counting the number of specific shapes is a fundamental problem in graph analysis. Intuitively, such an algorithm has a time complexity of $O(n^m)$ where m is the number of edges in the shape. This project aims to solve the problem by two approaches:

- Instead of exact counting, a Monte Carlo algorithm based on random sampling is designed to provide a theoretically reliable solution.
- GraphX takes the advantage of Spark and provide convenient programming interfaces for deploying such an algorithm on computing clusters.

This project will implement such an algorithm on GraphX (scala-Spark). Experiment and analysis will be carried out for coming up with performance measurements, including accuracy and speed up ratio.

Algorithm Design

input: graph, number of edges in pattern, sample rate
output: number of pattern

```
randCount(graph, n, rate):
        vertexSet<-graph.vertices.sample(rate)
        degreeGraph<-graph.mapEdges(e<-src.degree)
        for(1 to n):
                vertexSet.foreach{v=>nextStep=random pick 1 neighbor and validate}
                vertexSet.aggregateMsg(only if triplet.dstId==nextStep
                        p=p*1/edge.attr
                        sendMsg: (pathStartId,path,p,#ofiteration,nextStep),
                        rcvMsg: (a,b)=>a++b
                )
                updateGraph
        for end
        graph.vertices.filter(v=>v.vertexId==msg.pathStartId).sum(1/p)
end

validate:
        !path.contains(picked) || (#ofiteration==n && picked==pathStartId)
```

Measurements and Experiment

Some experiments are designed to evaluate the performance of the algorithm:

Group No.	algorithm	Mode	sample rate	Measurement
1	Exact counting	cluster	1	time, counts
2	Random counting	local	1	time
3	Random counting	cluster	1	time, counts
4	Random counting	cluster	varying	time, counts

- Speed up ratio due to randomization: Group 1 vs Group 3
- Speed up ratio due to parallel computing: Group 2 vs Group 3
- sample rate - speed up/accuracy: Group 1 vs Group 3,4

Deliverables

This is a 3-credit bearing project supervised by Prof. Ke YI. The final deliverable will be runnable Spark jar and report for performance details.

Code repository: https://github.com/PhilipGeng/RandCounter

Reference

F. Li, B. Wu, K. Yi, and Z. Zhao, Wander Join, Proceedings of the 2016 International Conference on Management of Data - SIGMOD '16, 2016.

R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, GraphX, First International Workshop on Graph Data Management Experiences and Systems - GRADES '13, 2013.