ISSN: 2393-8528

IJICSE
Open Access

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 2 Issue 6; November-December-2015; Page No. 04-06

# A Comparative Study of Graph Processing Tools: Pregel & Apache Giraph

**Richa Singh[1], Revathy Nair[2], Mohit Vats[3]**

[1,2,] Research Scholar, Dept.of Computer Science & IT ,SGVU,Jaipur

[3]Assistant Professor,Dept.of Computer Science & IT,SGVU,Jaipur

**ARTICLE INFO**

Received: 17 Nov. 2015
Accepted 21 Dec. 2015

**Corresponding Author:**

**Richa Singh**

Research Scholar, Dept.of Computer Science & IT, SGVU, Jaipur

**ABSTRACT**

Most of the practical computing problems are concerned with large graphs. The analysis of large graph provides valuable insights for social networking and web companies in content ranking. A large number of graph processing systems have been developed and evaluated. These graphs have a scale of billions of vertices and trillions of edges. The different frameworks for network analysis are–Apache Giraph, Neo4j, Apache Giraph++, Apache Flink, Pregel etc. These all frameworks are suited for the distributed network and based on the bulk synchronization algorithm. In this paper, we describe the two network analysis tools-pregel and apache giraph.

**Keywords:** Pregel, Apache Giraph, bulk synchronization algorithm, distributed computing.



## I. INTRODUCTION

The graphs can confine data dependencies in a network and the processing of graph now become a key component in the various applications such as social networking sites (facebook, twitter, google etc.) and web based search. The increase in the popularity in the social network site is proved by a large number of users in a short period of time. Due to the increase in the accessibility of the internet, provides a user 24/7 online presence and give confidence to build them strong interconnection relationships. Now-a-days, social networking sites become a tool for choice of connecting people. The complex networks are analyzed through the graphs and there are various graph processing tools have been developed. The algorithms include the computation of shortest path, clustering and the variation in the page rank in the network.

The graph provides a fundamental model of entities and the connections that specifying the relationships between them. A graph represents a data set which is defined by a set of *n* vertices V and a set of m edges E. The main task of analyzing a network is to describe a structure as a whole or identifying the important elements. The social site can produce an erroneous in the network data which cause exceed in the memory storage of a single computer, results in the distributed system. A distributed system consists of a set of self-sufficient processors having their own memory but seems to be a single consistent system to a user. The distributed systems are scalable as any number of processors can be added to it. There are many programming models have been introduced for the distributed system, some are the general –purpose (MapReduce) and some are graph focus (Pregel, Apache Giraph, Giraph++, Apache Flinketc). The graph focused frameworks are fundamentally based on the bulk synchronization algorithm. These frameworks are concerned of the scaling of the processing crosswise the machines. These frameworks based on two properties of graph processing applications:

- Most of the applications progress in an iteratively manner, update the data in the rounds till a fix point is reached.
- The computations in each iteration are done at the vertex level independently so that each vertex can be processed individually parallel.

The bulk synchronization processing arranged the computations in the synchronous steps which are known as “supersteps”, bordered by a global synchronization barrier. In this, vertex has a local state but there is no shared state and the communication process at the synchronous barrier is completed via a message passing.

## II. LITERATURE REVIEW

Ching, A, et al., “One Trillion Edges: Graph Processing at Facebook-Scale” focused on the real –world applications and the performance of the large scale

applications. They described how the graph processing frameworks support the workload of BSP based applications like Facebook. This paper also describes the improvements done in the Apache Giraph which enables one trillion edges to a graph. They proposed a technique which includes the splitting of supersteps and composable computations, permitted to them a pool of potential applications. They shared how the graph data can be scheduled for the pipeline computations.

Malewicz, G., et al., "Pregel: A System for Large-Scale Graph Processing" discussed a model which is more efficient, scalable and fault-tolerant implementation on the group of thousands of computers. They investigated the techniques to increase a graph scale and synchronicity of model which contributes to avoid the cost to wait frequently inter-step barriers. The pregel is designed for where communications done mainly over the edges and they were not focused to change. They believed that the system designed by them is sufficiently abstract and flexible.

Han, M., et al., "An Experimental Comparison of Pregel-like Graph Processing System" compared the system within the Pregel-like graph processing system and identified that the system performs well with the Giraph and GraphLab. They evaluate that the Giraph 1.0.0's needs a considerable improvements. They comparedthe system under four graph processing system –Giraph, GPS, Mizan and GraphLab on five datasets and four different algorithms: PageRank, SSSP, WCC and DMST. They found that the synchronous mode of Giraph and GraphLab performs all-around performance; GPS stand out at memory efficiency. The study found that the synchronous mode of Giraph, GraphLab and GPS do better than Mizan in all the experiments.

Staudt, C and Meyerhenke, H, "Complex Network Analysis on Distributed System- an Empirical Comparison" described and compared the graph processing algorithms for the distributed system. They considerd the four frameworks- Apache Giraph ,GraphLab, Giraph++ and Apache Flink .

Leskovec, J and Faloutsos, C, "Samling for Large Graph" explained the techniques used in graph processing, exploiting many algorithms for efficiently meeting large graphs. Their work covered many important aspects such as crawling for social network sites.

Kumar, R., "Online Social Networks: Modeling and Mining" focused mainly on the enormous structure of graph. He defined a generative model to describe the progression of the network and also defined the techniques which are used to verify the reliability of the model.

Li, J. and Power, R., "Piccolo: building fast, distributed programs with partitioned tables" implements the distributed graph calculations on the partitioned tables. Lumsdaine, A., et al., "Challenges in Parallel Graph Processing" compared a parallel System on x86-64 BGL (Boost Graph Library), cluster of 200 processors to the BlueGene implementations, achieved 0.43 seconds for a graph of 4 billion vertices and 20 billion edges. They found that the system goes worst performance above 32 processors.

Kulkarni, M., et al., "Optimistic parallelism requires abstraction" explained the irregular calculations, based on the optimistic set of iterations. In this case, to eliminate the data races when a condition of multiple iterators access the same data, the users have to provide the commutative semantics and undo operations between the different function calls. As a result, the user must take care of data sharing and avoidance of deadlock.

## III. NETWORK ANALYSIS TOOLS

### THE PREGEL

The network analysis tools are based on the BSPi.e. Bulk Synchronous parallel which deal with the problem of parallelizing jobs across the multiple workers for scalability. The BSP is mainly a vertex-centric in which a vertex can have active or inactive state. The calculation consists of series of supersteps and the synchronizations done at the superstep barrier. The problem with the BSP is that fast workers have to wait for the slow ones.

The first implementation of BSP is Pregel, provides the basic API for Graph processing algorithms. The basic of Pregel paradigm is characterized as 'think like a vertex'. The graph computation is defined by what each vertex has to compute. The edges are the communication channels which are used to transfer the result of one vertex to another and the edges are not counted in the computations. The computation is done in the series of supersteps. At each step, a user-defined function is executed at a vertex, a transmission of message and a vertex can change its state either active or inactive state. The message is transferred to its neighbor or the vertex whose id is known. Each superstep is ended with the synchronization barrier ensures that message is transmitted to the following supersteps. A vertex may be cause to halt the superstep and can resume when it receives a message.

To reduce a communication overhead, the Pregel preserves data locality. The data locality can be achieved by the computations performed on locally stored data. At the starting of program input graph is loaded once and all the computations performed in-

memory. Consequently, the Pregel supports only graphs that fit into the memory.

Pregel works on the master/slave model in which a machine acts like a master while others like workers. The master machine has a responsibility to partition the graph into subgraphs and allocates each partition a worker. The Pregel is planned for the Google cluster architecture. The architecture schedules jobs to optimize resource allocation which involve the killing of instances and the moving to the other locations.

The applications of Pregel are: PageRank, shortest path and Bipartite Matching.

- PageRank which is used to determining the importance of the document. This is based on the number of references to the document.
- Shortest path focus on the single-source shortest path in which algorithm finds a shortest path between the source vertexes to every other vertex.
- Bipartite Matching involves the two different set of vertices with edges only between the set. The result is a subset of edges with no common endpoints.

### THE APACHE GIRAPH

The Apache Giraphis an iterative graph processing system which is planned for the scaling of the thousands of the machines. It is capable of processing a trillion of edges. It is presently used at the facebook for the processing and analyzing the social graph. It was mainly inspired by the Pregel. Giraph extended the Pregel by adding the functionality such as master computation, edge-oriented input, shared-aggregators, out of-core component, composable computations etc.
The input model of Giraph is vertex-centric. It requires a data set which is mainly relative to the vertices and the edges. The jobs in the giraph run in a non-preemptible FIFO manner. The jobs are queued up into the pool where they wait until all the jobs get all the resources and execute.

The Apache Giraph is an open source implementation of the programming model which is written in the JAVA and uses ApacheHadoop for the execution of the vertices. The Apache Giraph is less efficient in some cases but provides an out-of-core mechanism that enables to handle the large data sets. It is the only framework that supports the twitter graph.

## IV. RESULT &CONCLUSION

The Apache Giraphsupports the different data structures for the vertex and adjacency lists. These include shared aggregators which prevents the bottleneck at the master and the decrease the overhead of Java garbage collection. The Giraph 1.0.0 is a user base such as facebook and also gone under multiple optimizations.

The Pregel is a distributed framework providing a natural API for graph processing to the users. It manages the distribution of the details invisibly, message passing and the fault tolerance. It implements a stateful model rather than dataflow model. It uses an open message approach and not replicate theremote values locally.

## V. REFERENCES


1. Catanese, S. et al. “Crawling Facebook for Social Network Analysis Purposes” In Proceedings of the 1st Workshop on Mining the Future Internet, pages 14–19, 2010
2. Ching, A et al. “One Trillion Edges: Graph Processing at Facebook-Scale”
3. Staudt, C et al., “Complex Network Analysis on Distributed System- An Empirical Comparison”Conference Paper July 2015
4. Schelter, S., “Large Scale Graph Processing with Apache Giraph” Invited talk at GameDuell Berlin 29th May 2012
5. Hâncean, M., “A Brief Overview of Social Network Analysis and its Current State within Romanian Sociology” International Review of Social ResearchVolume 3, Issue 3, October 2013, 5-11
6. G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: A system for large-scale graph processing. In SIGMOD, pages 135–146, 2010.
7. Han, M et al., “An Experimental Comparison of Pregel-like Graph Processing System”
8. Leskovec, J and Faloutsos.,C ., “Sampling from large graphs” In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 631–636. ACM, 2006.
9. Power, R and Piccolo,L ., “Building fast, distributed programs with partitioned tables” In Proceedings of the 9th USENIX conference on Operating systems design and implementation, OSDI’10, pages 1–14, Berkeley, CA, USA, 2010. USENIX Association.
10. Gregor, D andLumsdaine, A.,“The Parallel BGL: A generic library for distributed graph computations.”In Parallel Object-Oriented Scientific Computing (POOSC), 07/2005 2005.Accepted.
11. Kumar, R and Novak, J “Structure and evolution of online social networks” Link Mining: Models, Algorithms, and Applications, pages 337–357, 2010.