**Indian Institute of Technology, Patna**

**Large Scale Graph Processing: Neo4j Vs Apache Giraph Vs Hadoop-MapReduce**

**(Survey Report)**

**Nishant M Gandhi**

**M.Tech. CSE**

**IIT Patna**

Contents

[1. Introduction 2](#_Toc383251867)

[2. Graph Processing Platforms 3](#_Toc383251868)

[a. Hadoop-MapReduce 3](#_Toc383251869)

[b. Giraph 4](#_Toc383251870)

[c. Neo4j 4](#_Toc383251871)

[3. Analysis of Platforms 5](#_Toc383251872)

[a. Hadoop-MapReduce 5](#_Toc383251873)

[b. Giraph 5](#_Toc383251874)

[c. Neo4j 5](#_Toc383251875)

[4. Conclusion 5](#_Toc383251876)

[5. References 6](#_Toc383251877)

# Introduction

Today we are living in era of big data. From social media to scientific experiments, from computer to mobile devices, generate huge amount of data every day. Storing and processing this data is also the challenge now a day. There are so many real life problems, which can be solved with use of this generated big data. Many of these problems related with big data can be mapped to graph problems.

Many solutions have been created to process large scale data. One of the most popular is [4] Hadoop with its [2] MapReduce programming platform. The lack of a programming model dedicated for graph was addressed by Google with [3] Pregel. The Pregel uses Bulk Synchronization Parallel model for graph processing. The open source version of Pregel is [1] Giraph. Another platform is [6] Neo4j which is graph database processing platform.

In this document, we will try to understand these three platforms and their pros & cons.

# Graph Processing Platforms

There are many platforms available for large scale graph processing. However we are considering only these three platforms because all three contain very different programming models to process graph.

* [6]**Neo4J**: desktop platform, NoSQL, graph database, version
* [2]**Hadoop-MapReduce**: cluster platform, generic large-scale data processing platform
* [1]**Giraph**: cluster platform, large-scale graph processing specialized platform

## a. Hadoop-MapReduce

[2]Hadoop is an open-source platform for storing & computing huge amount of data. Hadoop has been used widely in many data analytics applications. It uses MapReduce programming model. Hadoop’s MapReduce programming model is inspired by functional programming’s Map & Reduce functions. The MapReduce programming model process input data and divides it based on key/value pairs.

Data used by [4] Hadoop is stored in the Hadoop Distributed File System (HDFS). HDFS is not a part of Hadoop, although it is being used by it and the platform will not work without HDFS. Datasets which are stored in the HDFS are divided into N blocks of similar size. Each of these blocks is used as an input for Mapper.

[8]Hadoop’s programming model has low performance and high resource consumption for iterative graph algorithms, because of programming model which require multiple map-reduce cycle. For example, for iterative graph traversing algorithms Hadoop would often need to store and load the entire graph structure during each iteration, to transfer data between the map and reduce processes through the disk-intensive HDFS, and to run an convergence-checking iterations as an additional job.

## b. Giraph

[1]Giraph is an open-source, graph specific distributed system platform. Giraph uses the Pregel programming model, which is a vertex-centric programming abstraction that adapts the Bulk Synchronous Parallel (BSP) model. A BSP computation proceeds in a series of global Supersteps. Within each Superstep, active vertices execute the same user defined compute function and create & deliver inter-vertex messages. Barriers ensure synchronization between vertex computations. Once there are no messages to process and all vertices vote to halt.

[8]Giraph utilizes the design of Hadoop, from which it leverages only Map phase. The single biggest difference between Hadoop & Giraph is the fact that Giraph is in-memory which speedup job execution. For fault-tolerance, Giraph uses periodic checkpoints. To co-ordinate Superstep execution, it uses [5]ZooKeeper.

## c. Neo4j

Neo4j is one of the popular open-source NoSQL graph database implemented in java. Neo4j stores data in graphs rather than in tables. Every stored graph in Neo4j consists of relationships and vertices annotated with properties. Neo4j can execute graph-processing algorithms efficiently on just one machine, because of its optimization techniques that favor response time. [8]Neo4j uses a two-level, main-memory caching mechanism to improve its performance. The file buffer caches the storage file data in the same format as it is stored on the durable storage media. The object buffer caches vertices and relationships in a format that is optimized for high traversal speeds and transactional writes.

Neo4j processes graphs by traversing all vertices, with the use of either the BFS or DFS traversal algorithm. To start graph traversal a program has to define a special reference vertex. This vertex is not a part of the original graph, but an additional artificial vertex which is add to the graph structure and act as a starting point of the graph traversal. All graph operations are performed as ACID transactions.

# Analysis of Platforms

The performance analyses of these platforms have been done several times but here I am using two materials and their results to write this section of report. The one is M.S. theses report of Marcin Biczak batch of 2013 from Delft University of Technology. Another one is report titled [8] “How well do Graph-processing platform performs?”

From these two materials, some important finding comes out which are as listed below. [8]However, the performance of all platforms is stable and largest variance around 10%.

## Hadoop-MapReduce

* + 1. [7]Hadoop-MapReduce performs worst in any graph algorithm then other platforms.
    2. [7]Multi-iteration algorithms suffer from additional performance penalties.

## Giraph

* + 1. [8]Giraph process graph in-memory and realize dynamic computation mechanism by which only selected vertices will be processed in all iterations of algorithms. That reduces computation time.
    2. [7]For large amounts of messages or big datasets, Giraph can lead to crashes due to lack of memory.

## Neo4j

* + 1. [8]Limited by the resource of single machine, the performance of Neo4j becomes significantly worst when the graph exceeds the memory capacity.
    2. [7]Neo4j was designed as a single machine dataset. To achieve multi scale, users of Neo4j have to implement communication between these machines as well as manage partitioning, consistency etc. It require significant amount of additional work beside the application implementations.
    3. [7]Two-level cache allows Neo4j to achieve excellent hot-cache execution times, especially when graph data accessed by the algorithm fits in cache.
    4. [8]The data ingestion time of Neo4j matches closely the characteristics of the graph. Overall, data ingestion takes much longer for Neo4j than HDFS.

# Conclusion

Based on survey, we can reach to following conclusion.

Modern computers can handle most of smaller or sparser graph databases. However, once the dataset size significantly increases or if the graph is dense, the execution time increases significantly. For this reason, single machine based graph processing platforms cannot compete with distributed system.

We have considered two graph-processing frameworks (Giraph, Neo4j) and a generic data-processing platform (Hadoop). The platforms which focus on processing graph dataset achieve significant performance advantages over generic platforms in most of cases. A Hadoop does not maintain the relations between data and treats every vertex as a disjoint, which other platforms have to and pay a performance penalty for it. Thus for certain datasets Hadoop can achieve better performance than the graph-processing platforms.

There are two significant factors for the large-scale graph-processing platforms: the programming model and platform design. The Pregel performs much better than MapReduce for iterative algorithms. Giraph has limitation that it performs in memory computation, which limitation is not for Hadoop-mapreduce. [7]The Neo4j has achieved good performance for smaller or sparser dataset on single system. It has very good documentation and hence easy to learn. [7]The Hadoop-mapreduce considered slowest platform of all evaluated platforms but Neo4j’s performance for the large or dense dataset is lower than that of Hadoop-mapreduce. [7]The Giraph platform, which represents distributed large-scale graph processing platforms, was the fastest platform in all the test experiment made by Marcin Biczak.

# References

1. Apache Software Foundation, “Giraph.” http://giraph.apache.org
2. J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” *Comm. ACM, vol. 51, no. 1,*2008, pp. 107–112.
3. Pregel: a system for large-scale graph processing - "abstract". G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. In Proceedings of the 28th ACM symposium on Principles of distributed computing, PODC '09, pages 6-6,New York, NY, USA, 2009. ACM.
4. Apache Software Foundation, “Hadoop” Website, 2011.http://hadoop.apache.org
5. Apache Software Foundation, “Zookeeper”.Website,2010. <http://zookeeper.apache.org>
6. Neo Technology, http://www.neo4j.org
7. LudoGraph: a Sampling Capable Cloud-Based System for Large-Scale Graph Processing Based on the Pregel programming model, Marcin Biczak, Masters of Science Thesis,Delft University of Technology Year 2013
8. Y. Guo, M. Biczak, A. Varbanescu, A. Isoup, C. Martella, and T. Willke, “How well do graph-processing platforms perform? an empirical performance evaluation and analysis: Extended report,” tech. rep., Delft University of Technology, 2013.