I J C T A, 9(41) 2016, pp. 1235-1239
© International Science Press

Parallel HITS Algorithm Implemented Using HADOOP GIRAPH Framework to resolve Big Data Problem

Hema Dubey[*], Nilay Khare[*], Alind Khare[**] and Ankur Dwivedi[*]

ABSTRACT

The Hypertext Induced Topic Search (HITS) algorithm developed by Kleinberg is a ranking technique to evaluate correlative authority and hub scores of web pages by exploring the hyperlink structure of the web. Today, the web graph has enormous size, and computing Authorities and Hub values effectively and rapidly for such massive data is truly a difficult task. In this paper, we present a parallel-distributed implementation of HITS algorithm on Hadoop framework using Giraph. The proposed work is analyzed on three standard datasets taken from Stanford University. The experimental outcomes illustrate that the proposed distributed HITS algorithm is more efficient and faster in terms of computational time than sequential HITS algorithm for colossal web graphs.

Key Words: Hypertext Induced Topic Search (HITS), Giraph, Hadoop, Bigdata;

1. INTRODUCTION

At the present era, many organizations such as google, facebook, yahoo, linkedIn etc. need to deal with terabytes of data every day. The data which is beyond the capability of a traditional database system to handle properly is termed as big data [1]. We cannot define big data based on the size of data exceeding some terabytes or petabytes rather it can be assumed that when the volume of data is larger that formerly encounter in a particular organization, then possibly we can say that organization is dealing with big data. Thus the definition of big data can vary from sector to sector. The primary problem exists in today's age is to how to handle the immense volume of data efficiently. This is known as a big data problem. Nowadays, many Industries are using customized big data processing frameworks such as Hadoop [1] MapReduce [2, 3], spark, storm, twister, etc. for parallel and distributed data computing.

Hypertext Induced Topic Search (HITS) is a popular hyperlinks structure based page ranking algorithm which processes a massive collection of web pages. Therefore, there is a need of some technology that can handle such huge set of web pages for HITS algorithm to provide a fast and efficient calculation of hubs and authorities values. Hub value is the total sum of authority values of the pages to which a page links. Authority value is defined as the sum of hub values of the pages which point to that page. HITS algorithm analyzes links or edges to rate web pages. The Rank of a page is a numeric value between 0 and 1 that signifies how important the page is on the web. The web is represented as a graph, where vertices are pages and edges are links to them. Greater authority score occurs if the page is pointed to by pages with high hub scores. A higher hub score occurs if the page points to many pages with high authority scores [4].

In this paper, we have proposed parallel-distributed HITS algorithm implemented on Hadoop Giraph framework for the purpose of efficiently dealing with big data problem. Hadoop [1] is capable of storing

[*] Maulana Azad National Institute of Technology, Bhopal, India, *E-mail: hema32150@gmail.com; nilay.khare@rediffmail.com; ankurd87@gmail.com*

[**] IIIT Delhi, India, E-mail: kharealind@gmail.com

and processing massive web graphs efficiently. Giraph is a framework built on the top of Hadoop to process huge web graphs iteratively. We have used Hadoop Giraph to program clusters of machines to perform extensive graph processing in a reliable and faster manner. We have compared the performance of both serial HITS algorithm implemented using Java versus parallel HITS algorithm implemented on Hadoop Framework.

2. BACKGROUND

2.1. Serial HITS algorithm

The Hypertext Induced Topic Search (HITS) algorithm was developed by J. Kleinberg in 1998 and is now a part of the CLEVER Searching project of the IBM Almaden Research Center [4]. A similar concept is incorporated in the Ask.com search engine. HITS is a hyperlink structure-based and query dependent algorithm. When a user fires a search query, first of all, HITS extends the list of pertinent web pages as retrieved by a searching engine, afterward, it calculates two rank scores (that is authority and hub) of the extended collection of web pages. A page with high authority value indicates that it has many in links and a page with high hub value has many out links. The HITS algorithm is an iterative method which works as follows:

1) Initially, we consider each node having a hub score and authority score of 1.
2) Calculate Authority score of node *x* by summing up the hub scores of all incoming nodes of node *x*.
3) Calculate Hub score of node *x* by summing up the Authority scores of all outgoing nodes of node *x*.
4) Normalization of scores:- each hub value is divided by the square root of the sum of squares of all hub values, and each authority value is divided by the square root of the sum of squares of all authority values.
5) Repeat steps 2, 3 and 4 until the authority and hub values start converging.

2.2. Hadoop Giraph

Hadoop is open source Apache programming system that can handle big data over a distributed environment. Hadoop supports massive data parallel processing in a fault tolerant behaviour. Hadoop provides HDFS (Hadoop Distributed Filesystem) for storing huge files with easy data access, executing on clusters of low-cost commodity hardware. Hadoop also provides MapReduce for processing colossal data.

Giraph [6, 7] is a massive graph handling model that uses the iterative methodology to work on hundreds of machines. Giraph processes huge graphs by using Hadoop MapReduce. Companies for example PayPal and Facebook uses Giraph framework for processing colossal social graphs with trillions of links. Giraph was encouraged by Pregel [5], the graph processing architecture developed at Google. Pregel provides huge scale graph processing API for executing iterative graph algorithms. Giraph has greatly extended the basic Pregel model with new functionality such as master computation, shared aggregators, edge-oriented input, out-of-core computation, composable computation, and many more features [5].

Giraph is based on master-slave architecture [6], in which a host (may be a physical or virtualized server) is a slave node or we can say worker node that carries out all computations and keeps data in the HDFS. Huge graphs are first partitioned and then distributed over different slave nodes. Giraph utilizes BSP (Bulk Synchronous Parallel) model that iteratively executes a chain of supersteps. Each superstep consists of message exchanging phase succeeded by aggregation phase. Hence, all the slave nodes can communicate with each other in parallel by sending small messages. After exchanging messages, aggregation is performed in order to update the properties of nodes and edges; this is known to be one superstep.

Giraph framework is fault tolerant, easy to program,and can process graphs of massive scale.Giraph is very efficient as compared to MapReduce framework because Giraph loads the whole graph in the memory only once for all supersteps whereas MapReduce has to load the graph again and again for each superstep

3. PROPOSED PARALLEL HITS ALGORITHM

3.1. Parallel HITS Algorithm Implemented Using HADOOP GIRAPH Framework

The proposed algorithm focuses on resolving the big data problem for HITS algorithm. Since HITS algorithm requires rapid and efficient processing of massive web graphs, so there is a need of parallel and distributed graph processing technique. In this paper, we have implemented HITS algorithm on Hadoop framework using Giraph. The input web graph is distributed over the Hadoop cluster so as to perform the parallel computation of hubs and authority scores. Since HITS is an iterative algorithm, therefore it stops when the hubs and authorities values start converging. In Giraph, nodes present within the graph communicates with each other by sending messages. So after completion of an iteration, the hubs and authorities values are shared with the relevant nodes. An iteration is called as superstep. The term superstep is taken from BSP programming model [7]. Each super step is composed of three phases:

1. Local computation phase, where each processor performs calculations using data values accumulated in local memory and make communication requests for instance reading and writing remote memory.
2. Global communication phase, in which, data is exchanged between the nodes based on the requests issued during the local computation phase.
3. Barrier synchronization phase, where all nodes wait until each node completed the global communication phase and the data is ready to be available for next superstep.

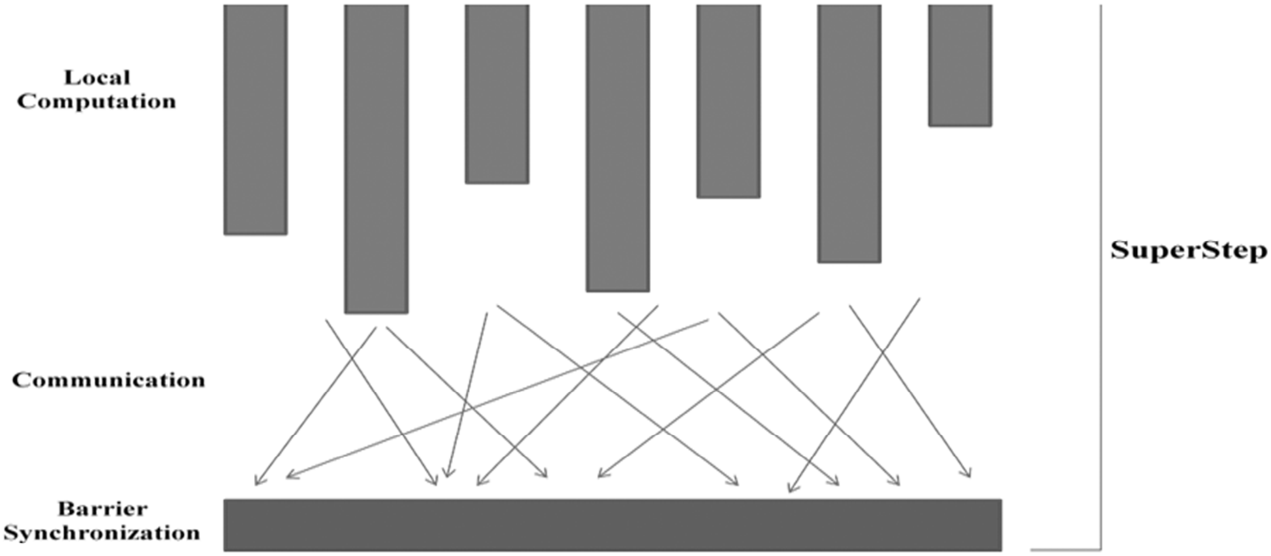

Figure 1: BSP (Bulk Synchronous Parallel) model [7]

In superstep one, initialize each vertex of a web graph with authority score and hub score to be one. At the beginning of an iteration, each vertex sends its authority and hub scores to all its outgoing vertices. Then Authority score of a node is calculated by summing up the hub scores of each node that points to it and Hub score of a node is calculated by summing up the authority scores of each node that it points to. In each superstep, after computing authority and hub scores, both these scores are normalized. These supersteps are repeated until the values of authorities and hubs start to get converged.

4. EXPERIMENTS AND RESULTS

We have implemented serial HITS algorithm using Java programming language. The proposed work is implemented using Giraph on Hadoop framework. Both the serial and parallel-distributed HITS algorithms are experimented on three standard social networks datasets obtain from Stanford University [8]. The names of three datasets are Amazon0302, Amazon0312 and WikiTalk. The description of these datasets is shown in table 1. We have performed the experiments on a cluster of five machines configured with Hadoop version 2.7.1. The operating system used is ubuntu 14.0 LTS. Each system has 4 GB RAM and 320 GB hard disk. The version of Giraph used is 1.1.0.

Table 1
Description of Data sets [8]

Dataset		*Type*	*Description*	*Nodes*	*Edges*
1	Amazon0302	Directed Graph	Amazon product co-purchasing network from March 2, 2003	262111	1234877
2	Amazon0312	Directed Graph	Amazon product co-purchasing network from March 12, 2003	400727	3200440
3	WikiTalk	Directed Graph	Wikipedia talk (communication) network	2394385	5021410

The experiments are performed for 100 iterations for both serial and parallel-distributed HITS algorithm. The execution (running) time of both serial HITS algorithm and parallel-distributed HITS algorithm for all three standard datasets is shown in table 2.

Table 2
Running times of Serial CPU based HITS and proposed HITS on Hadoop Giraph and also speed up of proposed algorithm for three datasets

Dataset		*Execution Time of Serial HITS algorithm (in milliseconds)*	*Execution Time of Proposed Algorithm (Parallel-Distributed HITS on Hadoop-Giraph) (in milliseconds)*	*Speed-up*
1	Amazon0302	24637	15984	1.54
2	Amazon0312	56535	45638	1.24
3	WikiTalk	194239	140763	1.38

As it is seen from the table 2, the proposed HITS algorithm on Hadoop using Giraph reduces the time complexity of sequential HITS algorithm. For Amazon0302 dataset, the proposed algorithm has a speed up of about 1.54. Thus we can say the speed of computing hubs and authorities scores on Hadoop increases by 65% for dataset 1. For Amazon0312 dataset, the proposed work has a speed up of 1.24 which means that the speed of computing HITS scores increases by 81%. Finally, for dataset WikiTalk, there is a speed up of 1.38 for proposed algorithm, which indicates that the speed of computing HITS values increases by 72.5%.

Hence, we can conclude from the table 2 that the proposed HITS algorithm implemented on Hadoop is more efficient in terms of execution time as compared to serial HITS algorithm.

5. CONCLUSION

This paper aims at reducing the execution time for calculation of hubs and authorities values especially when the web graph is of very large size. The proposed parallel-distributed HITS algorithm implemented using Apache Giraph enhances the performance of HITS scores computation by 65% to 81%. We perform experiments on Hadoop Giraph framework and Java programming model using standard datasets taken from Stanford University. We conclude that the proposed HITS algorithm presents improved results in terms of execution time as compared to sequential CPU based HITS algorithm if the size of input web graph is very large. The experimental results show that the proposed algorithm has a speed up of about 1.2 to 1.5.

REFERENCES

[1] *http://hadoop.apache.org/*

[2] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In OSDI, pages 137–150, 2004.

[3] Dean J., Ghemawat S.: Mapreduce: Simplified data processing on large clusters. In: OSDI 2004.

[4] Kleinberg, J. “Authoritative Sources in a Hyperlinked Environment” Journal of the ACM, Vol. 46, No. 5, September 1999, pp. 604–632.

[5] Malewicz, Grzegorz, et al. “Pregel: a system for large-scale graph processing.” Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.

[6] Sebastian Schelter, Invited talk at GameDuell Berlin on “Large Scale Graph Processing with Apache Giraph-IBM”, 29th May 2012.

[7] Westin Denver, “Apache Giraph - The Linux Foundation”, April 7-9, 2014.

[8] “Stanford Large Network Dataset Collection”, Snap.stanford.edu, 2016. Available: https://snap.stanford.edu/data/. [Accessed: 10- oct- 2016].