Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **2**, March **2017**

Design and Implement of Large-scale Social Network Analysis Platform Based on Hadoop

Di Yang*, Hua-Min Yang, Peng Wang and Song-Jiang Li

[1]School of Computer Science and Technology
Changchun University of Science and Technology
No.7186, WeiXing Street, ChaoYang District, Changchun, P.C 130022 - China
yangdi19900@163.com
*Corresponding author:yangdi19900@163.com

Received July, 2014; revised December, 2016

Abstract. *With the rapid development of social networks, Social Network Analysis (SNA) has become an important research direction. Aiming at the convenience of analyzing big data of social network, this paper designs a Hadoop based large-scale social network analysis platform (HNAP) for processing and analyzing large-scale data sets in a distributed environment. The designs of system framework, data model and processing model are introduced in detail. We also stresses on analyzing the characteristics of importing file format in preprocessing, the necessary features for an algorithm suitable for Hadoop platform and post-processing model in processing model. Finally, the rationality of the platform is verified by experiments. Our design is fully applicable to large-scale social network analysis, and takes full account of expandability in the future.*
Keywords: Social network, Social network analysis, Hadoop, HNAP

1. **Introduction.** With the rapid development of information technology, internet has entered into web2.0 era with the representative characteristics of personalization, interaction, virtuality and sociability. As a typical application of web2.0 era, Social Network Services (SNS) become popular all over the world at a very fast speed used in education, business, entertainment and many other fields [1].With the increase of social network users, a large-scale social network data is generated. How to deal with these large-scale social network data to extract further useful information for mining potential business opportunities has become an important direction of research and also a great challenge for social network analysis.

Many scientific institutions and internet companies are working to develop their own big data analysis platform [2]. Stanford University has developed a big data analysis tools SNAP [3] system which provides structured analysis and visualization analysis. The open-source environment also supports multiple compiling environment. However it is running in single-core condition which running efficiency is low in large data sets. AutoMap [4] is a text mining tool developed by CASOS at Carnegie Mellon University. It is used to enable the extraction of information from texts by using Network Text Analysis methods. AutoMap supports the extraction of several types of data from unstructured documents which includes: content analytic data, semantic network data, meta-network data and sentiment data. PEGASUS [5] designed by Carnegie Mellon University is a Peta-scale graph mining system, fully written in Java. It runs in parallel, distributed manner on top of Hadoop. It improves on the execution efficiency, but the algorithms provided by the

system are not enough to fully support the analysis. Statnet [6] of University of Washington is a suite of software packages for network analysis that implement recent advances in the statistical modeling of networks. Statnet provides a comprehensive framework for ergm-based network modeling, including tools for model estimation, model evaluation, model-based network simulation and network visualization. It runs in the R package. X-RIME [7] of IBM is a Hadoop based large-scale social analysis platform which provides structured analysis, transmission analysis, visualization analysis and efficient indexing query function. It improves on the analysis of the performance. SAE [8] of Tsinghua University is a tool for analyzing and mining large social network. The core of the analytic engine is a distributed graph database, which provides storage for the networking data. The system provides structure analysis, content analysis, communication analysis, influence analysis and user behavior analysis. Some open source projects such as PAJEK [9] and GraphLab [10] are also commonly used in social network analysis.

With the continuous expansion of network scale, the scale of user data is increasing fast. The traditional single-core processing mode has become inapplicable to process massive data. When dealing with large-scale data, it always appears insufficient of storage capacity and processing capacity which makes large-scale data analysis become more difficult. So distributed processing mode is more applicable. But existing distributed processing platforms are not all open source, and algorithms they provided are often limited on the analysis capabilities. Therefore we designed a Hadoop based Network Analysis Platform (HNAP) using distributed computing to solve the problem of insufficiency of single-core processing mode and developed HNAP algorithm libraries to solve the problem of insufficiency of analysis capacity by considering the characteristics of programs under the condition of MapReduce programing model.

2. **System framework.** In order to make full use of system resources to solve the problem of insufficiency of storage capacity and processing capacity under the environment of single-core, HNAP system is set up on the basis of Hadoop platform[11].

The main design ideas of HNAP: The whole system is divided into two parts, Hadoop Basic and Hadoop implement. Hadoop Basic is the basic part of the model, including HDFS, HNAP data model and MapReduce programming model. HDFS is the bottom of HNAP which provides storage for the vast amounts of data. Then, on the HDFS, data model is built to apply to analysis. Considering the characteristics of the network, data model is based on graph theory model. On top of HNAP data model, MapReduce programming model provides the ability to implement a series of applied social network analysis algorithms. HNAP implement, including HNAP Algorithm Libraries, uses Hadoop Basic to implement algorithmic analysis. Algorithm Library is the core of the whole system which determines the system capability. Then the above is the algorithm to be implemented, such as K-cores, PageRank, etc. Finally, the whole system model is constructed as shown in figure 1.

3. **Data model.** In order to make analysis easier, we design HNAP data model according to the following two principles: convenient for large-scale social network data processing and suitable for social network analysis. HNAP platform is mainly dependent on massively parallel processing capabilities of Hadoop platform. Therefore the design of data structure should be based on Hadoop massive data sets interface to meet the requirements of data processing under distributed development environment. Meanwhile, social networks are highly dynamic and relationships between the entities in the network are always developing and evolving, the connection between users is the most direct performance [12]. The representation of social networks is usually based on graph theory [13], by which we can

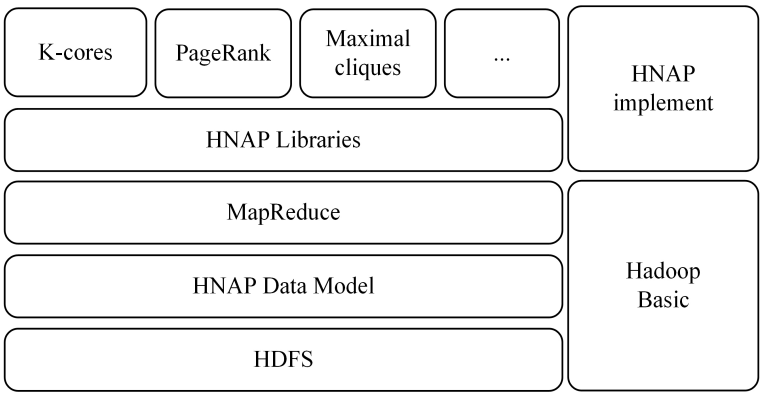

FIGURE 1. HNAP system framework

better express the relationship between the social networks of individuals, use a series of social network analysis algorithm for analysis and mining, and make analysis results more intuitive and clear. There are many storage method for graph, the most commonly used are the adjacency matrix storage and adjacency list storage method. Based on this, we design HNAP data model which supports graph theory and distributed environment. As shown in figure 2.

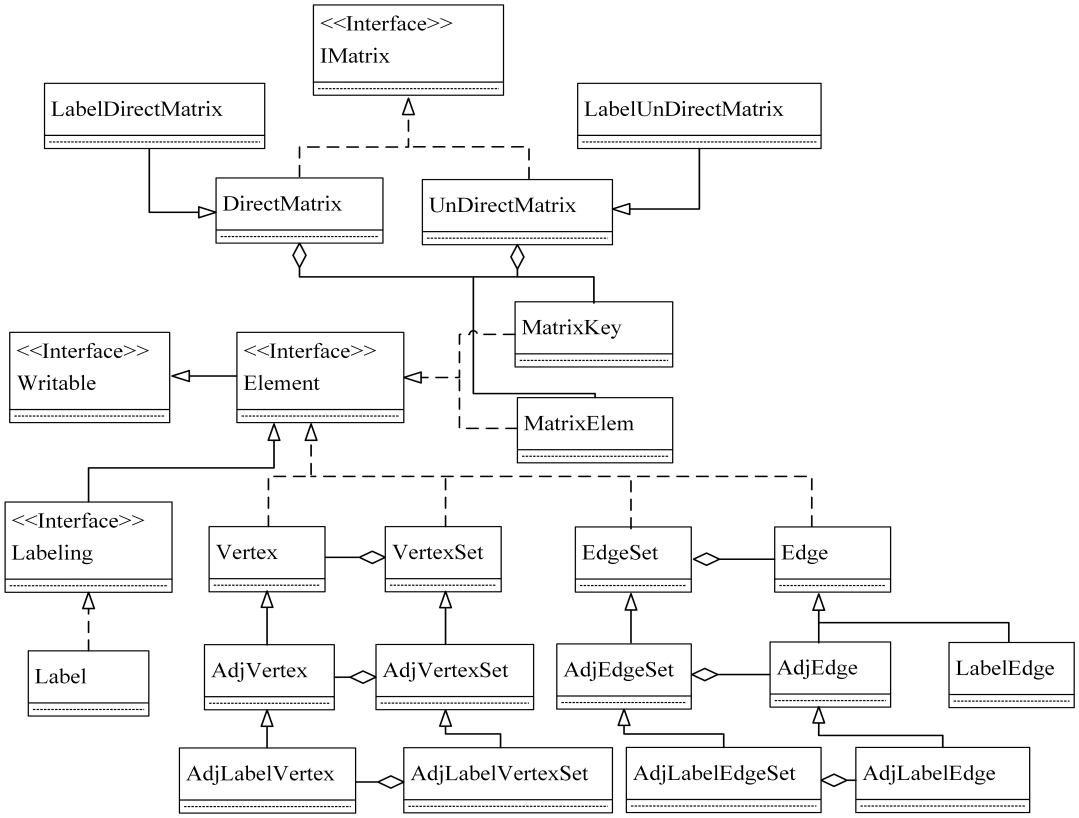

FIGURE 2. HNAP data model

In order to facilitate further extensions and add new elements, HNAP data model uses the form of polymorphic inheritance. Interface Element inherits serializable interface Writable. Other elements inheriting interface Element can be used in MapReduce programing model. Data elements are divided into three categories: label elements, adjacency list elements and adjacency matrix elements. Label elements carry additional

information of vertexes and edges, and also can be extended by interface Labeling. Adjacency list elements are divided into vertex elements and edge elements. Vertex elements are derived into AdjVertex and AdjLabelVertex through inheritance, the defined corresponding vertex collection classes AdjVertexSet and AdjLabelVertexSet are convenient for transmission and unified handling. Edge elements are also derived into AdjEdge and AdjLabelEdge through inheritance. But they both indicate only one side of an edge, unable to tell the side is a beginning vertex or ending vertex, need to combine with vertex elements. There is also a type of edge with the beginning vertex or ending vertex for a visual representation of a directed edge. At the same time, we also define the corresponding edge set for easy operation. Vertexes combining with edges constitute the visual expression of social network. Adjacency matrix consists of element MatixKey and MatrixValue which constitute DirectMatrix and UNDirectMatrix. Matrix elements can be used in conjunction with the label elements to constitute the label matrix used for data analysis. The structure of the HNAP data model is described in Table 1.

Table 1. Description of HNAP data model

Element	Description
Writable	Writable is the interface Hadoop used for serialization, based on class DataInput and class DataOutput to implement serialization protocol. The object of key and value in Hadoop must implement Writable.
Element	Element is the interface of vertex elements, edge elements and other elements in graph theory, and inherits interface Writable.
Labeling	Labeling is the interface of label elements, and implements interface Element and Cloneable. It uses container to store different kinds of label information.
Label	Label implements interface Labeling, and stores label information in the form of key-value.
Vertex	Vertex is the base class of vertex in graph theory, and implements interface Element and Cloneable.
AdjVertex	AdjVertex inherits the class Vertex, represents adjacent vertex in adjacent list, and usually combines with an edge element to express a vertex and its edge information.
AdjLabelVertex	AdjLabelVertex inherits the class AdjVertex, which is a labeled vertex class.
Edge	Edge represents directed edges, and implements interface Element.
AdjEdge	AdjEdge is one side of an edge in adjacent list which can represent the beginning point or ending point, need to combine with vertex elements.
AdjLabelEdge	AdjLabelEdge inherits the class AdjEdge, which can carry additional label information.
IMatrix	IMatrix is an interface of adjacent matix.
DirectMatrix	DirectMatrix inherits interface IMatrix, and represents directed matrix.
UNDirectMatrix	UNDirectMatrix inherits interface IMatrix, represents undirected matrix.
MatrixKey	MatrixKey implements interface Element and Cloneable, and shows location information of matrix elements
MatrixValue	MatrixValue implements interface Element and Cloneable, and shows value information of matrix elements and which Matrix it belongs to.

HNAP data model used for analysis and processing large data sets mainly contains the following three characteristics:

(1). The complete definition of graph theory model. According to the characteristics of social network, HNAP data model is based on graph theory model of social network, which makes the defined data model suitable for large-scale social network analysis.

(2). Based on the Hadoop distributed file system platform. The definition of HNAP data model is based on the Hadoop distributed file system to achieve large-scale mass storage efficiency of data sets.

(3). Expandability. HNAP data model reserves interface, fully considers the future expansion needs, has the flexibility and expandability.

4. **Processing model.** According to the system framework, on HNAP data model we use MapReduce programming model to develop algorithm library based on Hadoop, then build HNAP processing model for social network analysis. Due to the different data sources, the first to do before processing is to convert importing data to unified handling data. Next, HNAP can use the algorithms in algorithm library for data analysis. Finally, HNAP process the analysis result into readable type to output, complete an analysis. Thus, HNAP processing model includes the following four parts: obtaining the original data to be analyzed, preprocessing original data for unified handling data, using the algorithms library for social network analysis, processing follow-up results to get the final results of social network analysis. The processing model is shown in figure 3.

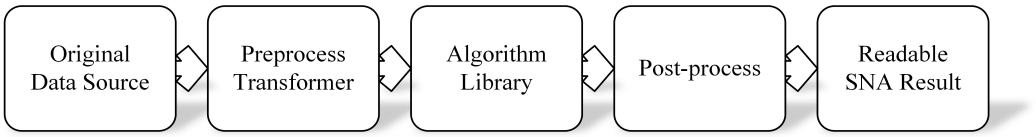

FIGURE 3. HNAP processing model

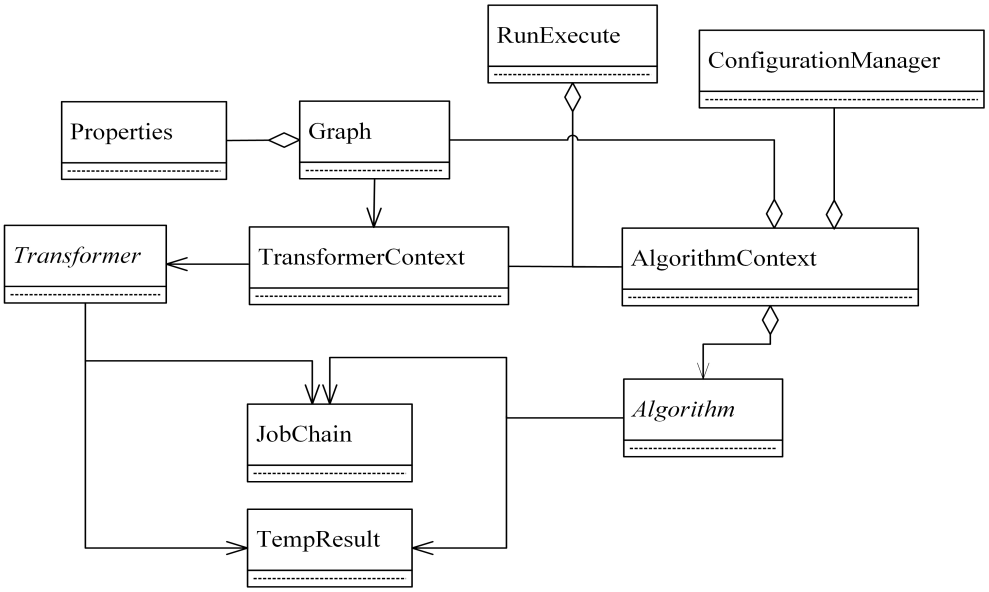

FIGURE 4. HNAP processing model class

Figure 4 shows system master class diagram. The core of master controlling program is class Algorithm which is an abstract class inherited by the actual analysis algorithms for polymorphism. Its convenient for further extensions of analysis algorithm. The design adopts the Strategy pattern, makes a practical algorithm to configure class AlgorithmContext which maintains a reference of an object of Algorithm. Class ConfigurationManager provides support to the algorithm execution parameters. Abstract class Transformer is a base class derived into different transformer classes. They convert the raw data into a system-defined data type for easy operation. It also adopts the Strategy pattern, makes a practical algorithm to configure class TransformerContext which maintains a reference of an object of Transformer. Considering a MapReduce of analysis algorithm or data

transformation may need to be executed multiple times to get the results, we design class JobChain to control MapReduce execution order. The provisional results that generated by program execution are maintained by class TempResult. Class RunExecute is the entrance of the program with the method of reflection to call the corresponding data conversion or analysis algorithm. The classes of the HNAP processing model are described in Table 2

Table 2. Description of HNAP processing model

Class	Description
Algorithm	Algorithm is an abstract class, implemented by practical analysis algorithms
AlgorithmContext	AlgorithmContext is configured by practical analysis algorithms, which maintains a reference of an object of Algorithm
ConfigurationManager	ConfigurationManager provides support to the algorithm execution parameters
Transformer	Transformer is an abstract class, implemented by practical transformation algorithms
TransformerContext	TransformerContext is configured by practical transformation algorithms, which maintains a reference of an object of Transformer
JobChain	JobChain is used to control MapReduce execution order
TempResult	TempResult is used to maintain provisional results generated by program execution
RunExecute	RunExecute is the entrance of the system

4.1. **Preprocessing.** As the data source is different, the original data format is different. Therefore, the purpose of the preprocessing is to specify data format so that the importing data can be processed by the program uniformly. Preprocessing is divided into two steps: text transformation and data type transformation. Text transformation converts the raw data into a unified format files, then data type transformation converts unified format files into the appropriate data type files stored in HDFS. In HNAP data model, combined with social network graph theory model, we store data in the form of the adjacency list and adjacency matrix. Therefore, we do adjacency list format conversion of the raw data as shown in figure 5. Text transformation shall be finished according to different source files, this part shall be maintained by programmers depending on actual condition. After unifying source file, we can use the program for a unified data type transformation into form of Vertex and Edge or adjacency matrix consisted of MatrixKey and MatrixValue.

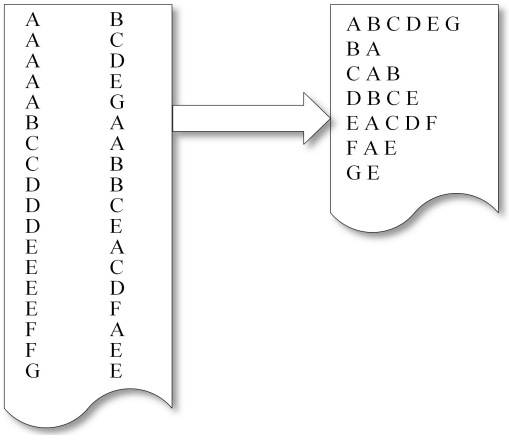

Figure 5. Text transformation

4.2. **HNAP algorithm library.** The algorithm library is the core of the whole system, and determines the size and strength of system analysis ability. Algorithm library depends on the MapReduce programming model provided by Hadoop. The problem can be solved by MapReduce has a common characteristic: a job can be decomposed into several sub jobs, and these sub jobs are relatively independent, the job will be solved after processing these sub jobs in parallel [14]. We set PageRank [15] algorithm as an example to show the characteristic above. This problem can be decomposed into three jobs of MapReduce: completion of raw data processing, iterative calculation of PageRank value, outputting final results. The three jobs are shown in figure 6.

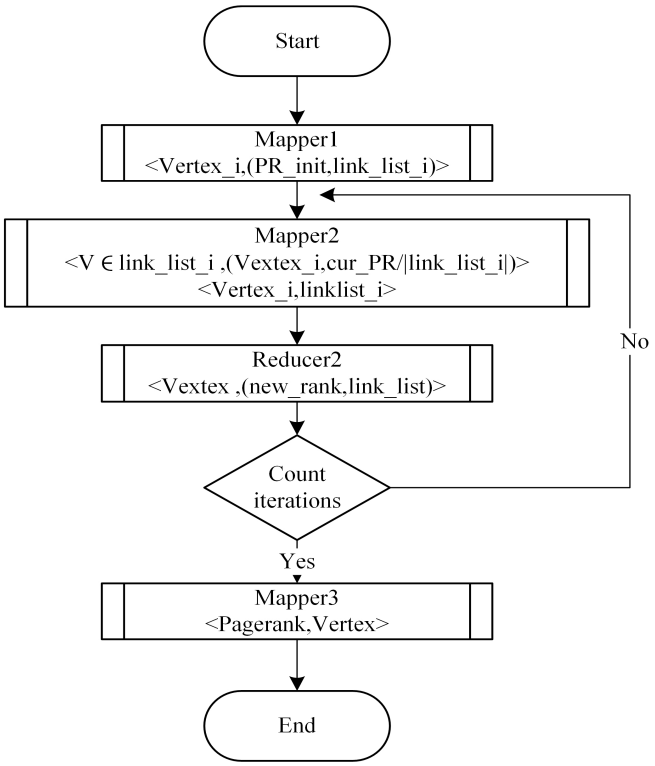

FIGURE 6. PageRank MapReduce flow diagram

In job 1, Mapper1 is used to deal with the raw data, which converts raw data into handling data format. In job 2, after processing raw data, Mapper2 is used to calculate transition probability of each node. Reducer2 calculates new pagerank value of each node. Job2 is performed iteratively. It won't stop until it reaches the final iterative times. In job 3, Mapper3 will get the final pagerank values of the iterative calculation results and output PageRank values from large to small order, it doesnt need Reducer. As can be seen from the algorithm implementation process, every job can process in parallel. Internal dependency between jobs only exists in that input data of the post job depends on the output of the previous job. Therefore, algorithm suitable for programming using MapReduce programming model must have relative independence in execution steps. This should be taken into consideration when designing algorithm library.

4.3. **Post-processing.** After the data is analyzed by the algorithm library, system produces a series of provisional results which are binary files under Hadoop platform, and have poor readability and treatability. We cant visually see the results of the analysis. Its not convenient to process other statistical analysis on data in later development. So we develop Post-processing model to convert binary file into readable, understandable

and processed text format for continuously statistical analysis. FFigure 7 plans for Post-processing model class diagram. Abstract class Postprocess is a base class for subsequent processing algorithm to expand down, to complete algorithms such as node statistics, collection statistics, finding objects and so on. Class RunPostprocess is the entrance to the subsequent processing procedures.

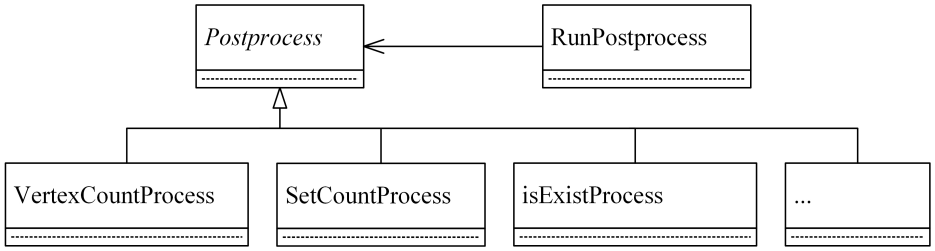

FIGURE 7. Post-process class diagram

5. **Experiment.**

5.1. **Hardware and software configuration.** According to the different function, HNAP system platform is divided into two parts: the underlying Hadoop cluster and HNAP algorithm implement. System integration environment Hadoop cluster consists of 3 machines, the configurations:

Hardware environment

- Master (NameNode): 16G memory, 500G hard drive, IP: 192.168.99.233
- Slave1 (DataNode): 8G memory, 500G hard drive, IP: 192.168.99.230
- Slave2 (DataNode): 8G memory, 500G hard drive, IP: 192.168.99.231

Software environment

- OS: Ubuntu12.04; JDK: JDK1.7; Hadoop: 2.6.0

5.2. **Experimental design.**

- Step 1: converting raw data to experimental data, the file format conversion is as mentioned in section 4.1.
- Step 2: following the algorithm to write the corresponding code.
- Step 3: starting the Hadoop platform.
- Step 4: uploading the processed data file to the HDFS.
- Step 5: running algorithm package, entering the appropriate parameters in the command, executing program.
- Step 6: saving the results in output path after the execution of the program.
- Step 7: analysing the experimental results.

5.3. **Analysis of experimental results.** We took the algorithm K-means as an example to illustrate the steps of HNAP. First we got American college football data set and political blogs data set; next programed the K-means algorithm; then started the Hadoop platform, ran the K-means package, outputted the experimental results. The parameters and results are shown as follows:

5.3.1. *American college football data set.*

- Clustering parameters: dimension=113, clustering number=2, maximum iteration number =10.
- Running time: 62578ms.
- Operating environment: 1 NameNode and 2 DataNodes.
- Figure 8 shows the result of running the College football American data set, Grey and black dots represented two kinds of points. For this data set with only 113 nodes, the time gap between usages of serial computation and parallel computation was not large. Although the results had a gap with the results of classic algorithm, in the case of K-means iteration number is 10, it could still be clustered effectively.

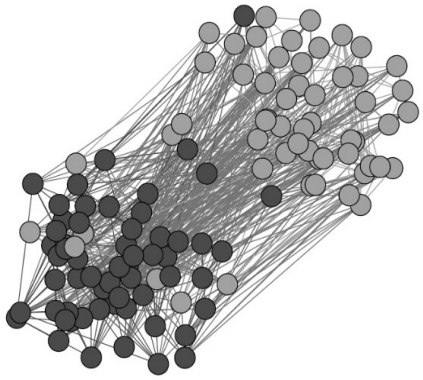

FIGURE 8. American college football data set result

5.3.2. *Political blogs data set.*

- Clustering parameters: dimension=1490, clustering number=2, maximum iteration number =10.
- Running time: 176542ms.
- Operating environment: 1 NameNode and 2 DataNodes.
- Figure 9 shows the Political blogs data set operation results. Because there were too many nodes, we could only see the results of node clustering, border relationship was not obvious. But it could also be seen that the parallel algorithm was effective for the data set. Therefore, HNAP we had designed had the ability to analyze big data.

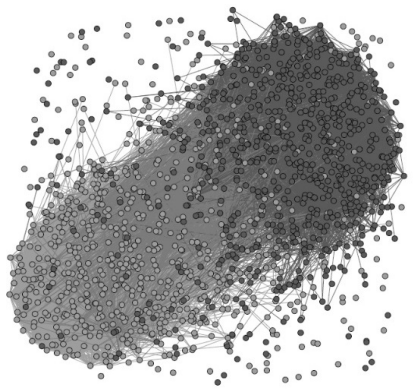

FIGURE 9. Political blogs data set result

6. **Summary.** The vigorous development of the social network has injected new vitality for the Internet. At the same time, huge amounts of data generated in social network also provides conditions for the social network analysis to dig deep information. The existing social network analysis tools are mostly depends on the algorithm itself, in the face of massive data often show inadequate. Because of this, this paper hopes that based on the existing algorithms to build a more efficient analysis platform on the processing performance. This paper plans the structure of HNAP system, the design of the data model and processing model. HNAP is built on Hadoop platform, depends on its distributed storage and processing capacity to improve efficiency in the implementation. It also provides a new platform for improvement of social network analysis algorithms. The system is fully considered the design of the future demand of expansion and applicability. Next work is to improve the algorithm library and optimize the system and algorithm gradually.

Acknowledgment. The authors would like to thank all the editors and all the referees for their useful comments and contributions for the improvement of this paper.This research work was supported by the Key Program for Science and Technology Development of Jilin Province of China under Grant No. 20150204036GX.

REFERENCES

[1] Y. M. Cheng, A. Isman, Teresa Chang-hui Hsu and Tien-Hsin Hsin, A Study on Windows Live Social Network Integrated into Information Instruction in Elementary Schools, *[J]. Journal of Computers and Applied Science Education*, vol. 1, no. 1, pp. 26-44, 2014.

[2] J. Tang, W. Chen, Deep analytics and mining for big social data, *[J]. Chinese Journal*, vol. 60, no. 5-6, pp.509, 2015.

[3] SNAP home page: http://snap.stanford.edu/snap/doc.html.

[4] J. Diesner, K. M. Carley, AutoMap 1.2 C Extract, analyze, represent, and compare mental models from texts, *Carnegie Mellon University, School of Computer Science, Institute for Software Research, Technical Report*, CMU-ISRI-04-100, 2004

[5] U. Kang, C. E. Tsourakakis, C. Faloutsos, PEGASUS, A Peta-Scale Graph Mining System Implementation and Observations, *[J]. IEEE International Conference on Data Mining, IEEE Computer Society* pp. 229-238, 2010.

[6] Handcock M S, Hunter D R, Butts C T, et al. statnet: Software Tools for the Representation, Visualization, Analysis and Simulation of Network Data, *[J]. Journal of Statistical Software*, vol. 24, no. 1, pp. 1548-7660, 2008.

[7] W. Xue, J. W. Shi, B. Yang X-RIME: Cloud-Based Large Scale Social Network Analysis,*Proc. in IEEE International Conference on Services Computing*, SCC 2010, Miami, Florida, Usa, pp. 506-513, July. 2010

[8] SAE home page: https://github.com/actnet/saedb.

[9] B. S. Dohleman, Exploratory social network analysis with Pajek, *[M]// Exploratory social network analysis with Pajek /. Cambridge University Press*, pp. 605-606, 2006.

[10] Y. Low,D. Bickson, J. Gonzalez, et al., Distributed GraphLab: a framework for machine learning and data mining in the cloud, *Proceedings of the Vldb Endowment*, vol. 5, no. 8, pp. 716-727, 2012.

[11] T. White, D. Cutting, Hadoop : the definitive guide, *[J.] Oreilly Media Inc Gravenstein Highway North*, vo. 215, no. 11, pp. 1 - 4, 2010.

[12] M. Liu, J. F. Guo and X. Luo, Link Prediction Based on the Similarity of Transmission Nodes of Multiple Paths in Weighted Social Networks, *[J.], Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 4, pp. 771-780

[13] N R. Malik, Graph theory with applications to engineering and computer science, *[J]. Proceedings of the IEEE*, vol. 63, no. 10, pp. 1533-1534, 1994.

[14] X. C. Dong, in Hadoop Internals: In-Depth Study of MapReduce, *China Machine Press, Beijing*, pp. 29-32, 2013.

[15] H. T. Haveliwala, Topic-sensitive PageRank, *[C.] International Conference on World Wide Web. ACM*, pp. 16-23, 2002.