IJETIE
International Journal of Emerging Technology and Innovative Engineering
Volume 2, Issue 5, May 2016 (ISSN: 2394 – 6598)

Date of Publication: 17.05.2016

ANALYSIS EFFICIENY OF GRAPH COMPUTATION BASED ON PARTITIONING TECHNIQUES

S.Suganya
PG Scholar, Dept of CSE
P.S.R Engineering college
Sivakasi, Tamilnadu
Suganyapavi9@gmail.com

Dr. K. RubaSoundar
Prof and head of CSE
P.S.R Engineering college
Sivakasi, Tamilnadu
rubasoundar@psr.edu.in

D. ArunShunmugam
Asst Prof, Department of CSE
P.S.R Engineering college
Sivakasi, Tamilnadu
arunshunmugam@psr.edu.in

ABSTRACT

Graph computation system performance is generally affected by the quality of graph partition. The balance factor and small edge cut ratio is used to measure the quality of graph partition. A balance graph partition with small edge cut ratio is generally preferred since it cut down the expensive network communication cost. Even so, according to the study of the Giraph, the execution over well partitioned graph strength is even two times worse than simple random partitions. This system only optimize for simple partition strategies and cannot efficiently handle the increasing workload of local message processing when high quality graph partition with the encryption process is used. In this paper, we provide a new processor and dynamic concurrency control models. The new message processor at the same time processes local and remote message in a unified way, the dynamic model adaptively adjust the concurrency of the processor based on the online statics.

Keywords-Graph computation, graph partition, message processing.

I. INTRODUCTION

Data mining deal with extraction of required information from large volume of the data such as medical research, customer relationship management, file management. The large volume of individual information are collected and analyzed with the help of data mining. The objective of graph computation is to partition information to guide parallel processing resource allocation, and improve the computation performance.

Graph computing performance splits the original graph into several sub graphs, such that these sub graphs are of about the same size and there are few edges between separated sub graphs and partition with high quality.

In particular, web graphs, social networks and other interactive networks in bioinformatics contain large amount of data. The large amounts of data are process by using parallel graph computation with message processing.

II. RELATED WORK

The field of graph computation has been advances in recent years because of the increasing the message processing

concurrency.

S. J. Alexander et.al [2], proposeda framework for large-scale graph decomposition and inference.Factorization technique that relies on partitioning a graph to minimize the number of neighboring vertices rather than edges across partitions based on a streaming algorithm.

N. Backman et.al [3], Stream processing applications run continuously and have varying load. Cloud infrastructures present an attractive option to meet these fluctuating computational demandsfor end-to-end latency.

P. Boldi et.al [4], continue the line of research on graph compression started, but they focus to the compression of social networks in a proper senseused for a long time to compress web graphs.

Z. Khayyat et.al [5], Pregel was in recent times introduced as a scalable graph mining system that can provide significant performance improvements over traditional Map Reduce implementations.

I. Stanton et,al [6], Extracting knowledge by performing computations on graphs is becoming increasingly challenging as graphs grow in size.Using our streaming partitioning methods.

S. Ghemawat et.al [8], Map Reduce is a user interface design model and an associated execution for processing and generating large data sets. Users specify a map function that processes a key pair to generate a set of intermediate key pairs, and a reduce function that merges all middle values connected with the same intermediate key.

III. SYSTEM DESCRIPTION

Using Partition aware graph engine is designed to support different graph partition qualities and maintain high performance.

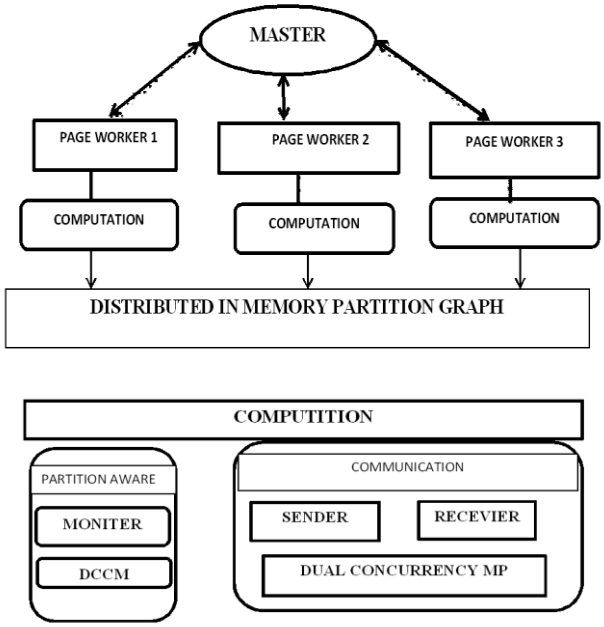


***Fig 1*: System Description**

Graph computation follows the master-worker paradigm. The computing graph is partitioned and distributive stored among workers' memory.

The master is responsible for aggregating global statistics and coordinating global synchronization

Each worker perform the computation with two modules:- the communication module and partition Aware modules.

The communication modules they perform communication between the sender and receiver with dual concurrency message processing.

The partition aware module contains two key components: a monitor and a Dynamic Concurrency Control Model. The monitor is used to maintain necessary metrics and provide this information to the DCCM.

IV. IMPLEMENTATION

This section describes the implementation of the proposed work. The proposed system implemented with the following modules:

- Map reduce
- encryption
- File downloading

Map reduce

Map reducing program is composed of map() and reduce() method. The map() method that perform filtering and sorting . The reduce() method that perform a summary operation . The mapreduce framework is used for distributed servers, running the various task in parallel, managing all communication. Its is mainly used for providing redundancy and fault tolerances.

The dataflow of map reduce are

- An input reader
- An map function
- An partition function
- An compare function
- An reduce function
- An output reader

The map reduce is perform with node transferring process in file by using routing algorithm with their energy.

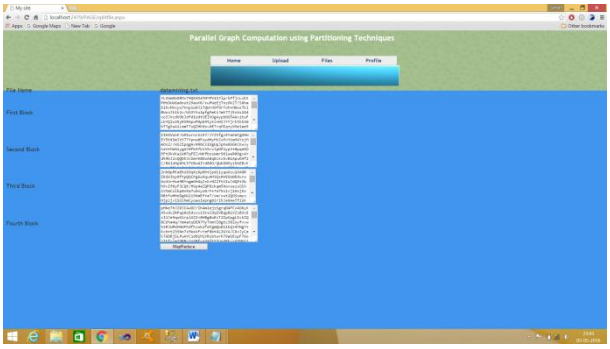
Encryption:-

Encryption process is performing by using the RSA cryptography techniques. The RSA algorithm involves four major steps:

- Key generation
- Key distribution
- Encryption
- Decryption

In this paper the encryption process is perform on uploaded file. During the file uploading the uploaded file are split by using wordchunk and its size. Total chunksize of the file is splitting and stored into separate textbox.

The data in the textbox are stored in encrypted format with private and public key.



File downloading

The file downloading process is performing after file accepting. The admin can able to downloaded needed file.

V. RESULT

This section describes the result and analysis of proposed method work

Table 1–cost for diff partition scheme

Partition Scheme	Localc ost	Remote Cost	Overall Cost
RANDOM	01	8	33
LDG1	0.5	4	30
LDG2	0.5	4	30
LDG3	0.5	3.5	30
LDG4	0.3	3	29
METIS	0.2	1.5	27

Table I describes the partition cost for different partition scheme based on the PAGE. It shows that the comparisons of cost for partition the data on different partition qualities.

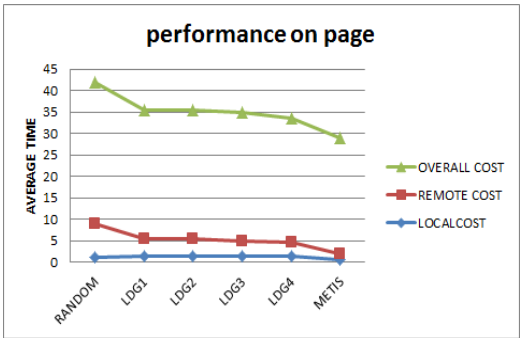


***Fig 3*- graphical representation of performance on PAGE.**

VI. CONCLUSION

In existing system it identified the partition unaware problem in currentgraph computation systems and its severe drawbacks for efficient parallel large scale graphs processing. To address this problem, In proposed system a partition aware graph computation engine named PAGE that monitors three high-level key running metrics and dynamically adjusts the system configurations. In the adjusting model, we elaborated two heuristic rules to effectively extract the system characters and generate proper parameters. We have successfully implemented a prototype system and conducted extensive experiments to prove that PAGE is an efficient and general parallel graph computation engine.

VII. REFERENCES

[1] S. Yingxia, Y. Junjie, C. Bin, and M. Lin, "Page: A partition aware graph computation engine," IEEE transactions on knowledge and data engineering, vol. 27, no. 2, february 2015, pp. 823–828.

[2] A. Amr, S. Nino, N. Shravan, J. Vanja, and S. J. Alexander,"Distributed large scale natural graph factorization," in Proc. 22Nd Int. Conf. World Wide Web, 2013, pp. 37–48.

[3] N. Backman, R. Fonseca, and U. C .etintemel, "Managing parallelism for stream processing in the cloud," in Proc. 1st Int. Workshop Hot Topics Cloud Data Process., 2012, pp. 1:1–1:5

[4] P. Boldi, M. Rosa, M. Santini, and S. Vigna, "Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks," in Proc. 20th Int. Conf. WWW, 2011,pp. 587–596

[5] Z. Khayyat, K. Awara, A. Alonazi, H. Jamjoom, D. Williams, and P.Kalnis, "Mizan: A system for dynamic load balancing in large-scale graph processing," in Proc. 8th ACM Eur. Conf. Comput. Syst.,2013, pp. 169–182

[6] I. Stanton and G. Kliot, "Streaming graph partitioning for large distributed graphs," in Proc. 18th ACM SIGKDD Int. Conf. Knowl.Discovery Data Mining, 2012, pp. 1222–1230.

[7] S. Zechao and Y. Jeffrey Xu, "Catch the wind:Graph workload balancing on cloud," in Proc. IEEE 29th Int. Conf. Data Eng., 2013, pp. 553–564.

[8] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in Proc. Operating Syst. Des. Implementation,2004, pp. 107–113

[9] S. Ma, Y. Cao, J. Huai, and T. Wo, "Distributed graph pattern matching,"inProc.21stInt.Conf.WorldWideWeb,2012,pp.949–958.

[10] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, "S4: Distributed stream computing platform," in Proc. IEEE Int. Conf. Data Mining Workshops, 2010, pp. 170–177.