DOI: 10.36297/vw.jeit.v1i1.8

VW Electronics and Information Technology, Volume: 1, Issue: 1, 04-11 (2019)

Towards Decoupling Computation in BSP Communication

Ibad ur Rahman[1,ϕ]
Yaseen Muahmmad[2,ϕ]
Shahzad Ahmed[3]
Muhammad Hanif [4,*]

[1]Militry College of Signals, NUST, Pakistan. ibaduet@gmail.com
[2]Univeristy of Engineering and Technology, Peshawar, Pakistan. yaseen@live.com
[3]Department of Electronics and Computer Engineering, Hanyang University, Seoul, South Korea shahzad1@hanyang.ac.kr
[4]Division of Computer Science and Engineering, Hanyang University, Seoul, South Korea honeykhan@hanyang.ac.kr

ϕ Authors contributed equally to this work
*Corresponding author email: honeykhan@hanyang.ac.kr

Received: Sept. 01, 2019
Accepted: Sept. 21, 2019
Published online: Sept. 23, 2019

Abstract: In recent times, the Bulk Synchronous parallel (BSP) model, which splits the underlying running programs into multiple super-steps, has gain popularity in distributed graph processing systems. However, high number of message and data exchange in each super-step will create a long epoch of time, often called as network overhead or communication delays. In this paper, we add an additional step in the synchronization phase through decoupling the synchronization in to local and global synchronization. This approach move some of the data and messages exchange over the network to locally optimized pointers for the next super-step. We prototyped the proposed system and the preliminary evaluation shows decoupled mechanism outperforms baseline system on the average. The system help eliminate the unnecessary communication between the instances and localize some of the DAG nodes for the next super-step.

Keywords: Communication, BSP, Hama, Distributed Computing.

1. Introduction

In the past decade, there has been exceptional data growth due the innovation in social networking sites, scientific experiments, government records, ecommerce sites, and sensors & radar network [1]. This large amount of data leads to the introduction of terminologies like "big data" and "large scale data" to identify data that cannot be captured, curated, processed, or managed by available traditional tools rationally [2]. Big data has some main features as commonly called as five "Vs" of big data: Velocity, Volume, Value, Variety, and Veracity [3].

Velocity states the speed at which massive amounts of data is being generated, collected, and analyzed. Every day the number of emails, twitter messages, photos, video clips, etc. increases at lighting speeds around the world. Every second of every day data is increasing. Volume refers to the farfetched amounts of data generated each second from social media, cell phones, cars, credit cards, M2M sensors, photographs, video, etc. The immense amounts of data have become so large in fact that we can no longer store and analyze data using traditional database technology. Value referring to the worth of the data being extracted. Having boundless amounts of data is one thing, but unless it can be turned into value, it is useless. While there is a clear link between data and insights, this does not always mean there is value in Big Data. Variety is the different types of data we can now use. Data today looks very different than data from the past. We no longer just have structured data (name, phone number, address, financials, etc) that fits nice and neatly into a data table. Today's data is unstructured. In fact, 80% of all the world's data fits into this category, including photos, video sequences, social media updates, etc. Veracity is the quality or trustworthiness of the data. Just how accurate is all this data? For example, think about all the Twitter posts with hash tags, abbreviations, typos, etc., and the reliability and accuracy of all that content. Gleaning loads and loads of data is of no use if the quality or trustworthiness is not accurate.

To handle such volume of data, a rich set of tools has been developed as it require a lot of computing power and memory for processing. As super computers are not a feasible option, therefore, distributed computing have been adapted as an alternative. However, large hardware clusters foist further challenges needs to addressed appropriately [4][5][6]. To tackle the technological challenge of providing mechanism for storage, knowledge retrieval, and big data manipulation, emergence of big data processing frameworks are increasing rapidly[7].

Apache Hama [8] is one of the open source bulk synchronous parallel (BSP) computing framework. It is distributed computing framework with the ability to perform tasks on matrices, graphs, machine learning [11], and networking algorithm. In Hama, all the communication is done in synchronization phase. The data transfer in synchronization phase may sometime becomes the bottleneck for the system. Also, the local massages take some part of the network, leading to a network overhead for the whole Hama cluster. In this paper, we present decoupled synchronization phase for the hama framework so that it efficiently communicate with the nodes and gain significant performance in the process. The rest of the paper is section 2 presents motivation for the work, section 3 shows the details of the system architecture and a use-case scenario, and section 4 concludes the paper.

2. Motivation

As described in figure 1, Hama framework follows bulk synchronous parallel [9] computing model to perform varied immense computational tasks on matrices, graphs, machine learning, network, and deep learning algorithms. In framework like Hama, which runs on BSP models, a parallel program executes transversely a set of virtual processors and runs as a series of parallel super-steps, which are separated by barrier synchronization[10]. The overall computation in a BSP based solution includes various super-steps and each super-step is unruffled to the following phases:

- Local Computation: Each process executes local computation using local data values and issues communication requests for remote memory read and write actions.
- Global Communication: Processes interchange their locally produced data according to the requests made during the local computation phase.
- Barrier Synchronization: Guarantees completion of all communiqué operations and makes formerly exchanged data available to processes for consumption in the next super-step.

In frameworks like this, even the local data and messages exchange in the barrier synchronization phase transfer through the network, which make it harder for efficient usage of available resources. It also becomes bottleneck in shared clusters, where the network is shared among different application. The worse case condition is when different network intensive applications compete for the network available resources in shared environment, leading to failure of super-steps and sometime the whole application.

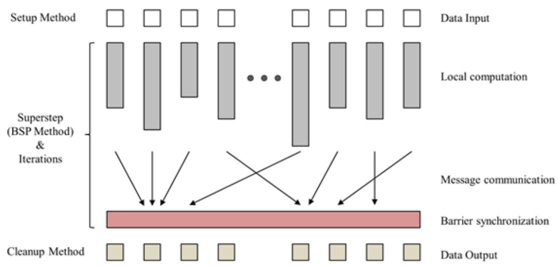

Figure 1: Hama Architecture: Bulk synchronous parallel programming based model to execute program as set of parallel processes in each superstep.

3. Proposed System

This work tries to decouple the barrier synchronization phase into two mutually independent phases of Local Synchronization and Global Synchronization as shown in figure 2. Local Synchronization create and save the result locally in the node for the next super-step to be utilized. While Global Synchronization is as the same as the default system, to synchronize all the communication and computation for the current super-step before moving forward to the next super-step. This kind of the decoupling help to partition the neighboring nodes of the graph into the same instance of the cluster, leading to network traffic reduction, increase locality of tasks due to better partitioning, and optimize resource utilization in the process.

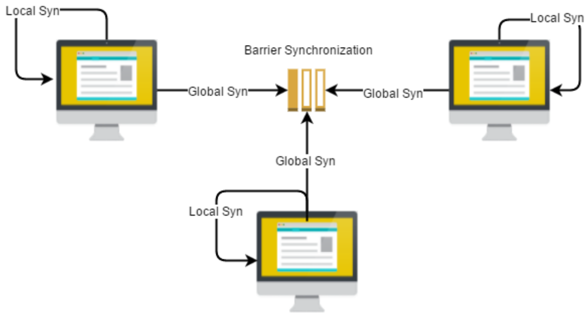

Figure 2: Modified BSP with Local and Global Synchronization phases running in parallel to optimize resource utilization.

4. Experimental setup and results

In this study, to evaluate the performance of the proposed system, we have designed and execute certain set of experiments with different variations of data and its distribution. Our preliminary simulation results shows that in case of the increase in the data locally on each node, the network overhead reduces over the course of time span. In case of the number of instances running increases, the network overhead increases with it, i.e. both of them are directly proportional to each other as we can see in table 1. It also, shows that the decoupled-Hama coup with the situation and adapt according to the changes in the network based on the condition of the cluster at run time and was able to decrease the network overhead accordingly. These results are the average of 10 fold experiments. We plan to purse the area and do more experiments and implement real world use-cases scenarios on a real test-bed.

Table 1: *Network overhead with the increase of instances in a shared cluster environment*

# of Instances	Hama	Decoupled-Hama
2	60 %	51 %
4	63%	54%
6	71%	62%
8	75%	67%
10	83%	69%

5. Conclusion and further work

This paper aims to mitigate the communication overhead in the BSP-based distributed graphs processing systems. We designed a decoupled version of the apache Hama to optimize the resource utilization through local synchronization phase inserted into the current BSP model. Preliminary evaluation shows decoupled-Hama outperforms baseline system on the average.

Author Contributions: I. R. and Y. M. designed the experiment and the algorithm, S.A. tested the optimized version of the system, I. R. and Y. M. carried out experiments, M. H. supervised the overall research work.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S. Ahmed, F. Khan, A. Ghaffar, F. Hussain, and S. H. Cho, "Finger-counting-based gesture recognition within cars using impulse radar with convolutional neural network," *Sensors (Switzerland)*, 2019.

[2] I. Anagnostopoulos, S. Zeadally, and E. Exposito, "Handling big data: research challenges and future directions," *J. Supercomput.*, 2016.

[3] V. Shobana and N. Kumar, "Big data - A review," *Int. J. Appl. Eng. Res.*, 2015.

[4] Gartner, "Gartner Says Solving ' Big Data ' Challenge Involves More Than Just Managing Volumes of Data," *Gartner*, 2011. .

[5] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of 'big data' on cloud computing: Review and open research issues," *Information Systems*. 2015.

[6] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," in *Mobile Networks and Applications*, 2014.

[7] F. H. Gebara, H. P. Hofstee, and K. J. Nowka, "Second-generation big data systems," *Computer (Long. Beach. Calif).*, 2015.

[8] S. Seo, E. J. Yoon, J. Kim, S. Jin, J. S. Kim, and S. Maeng, "HAMA: An efficient matrix computation with the MapReduce framework," *Proc. - 2nd IEEE Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2010*, pp. 721–726, 2010.

[9] L. G. Valiant, "A Bridging Model for Parallel Computation," *Commun. ACM*, 1990.

[10] Sachin Ghai, "Large scale graph processing with Apache Hama," 2015. [Online]. Available: http://www.hadoopsphere.com/2015/06/large-scale-graph-processing-with.html. [Accessed: 25-Aug-2019].

[11] Ahmed, S., Jo, H. I., & Jeon, J. Y. (2018, January). Classification of human sounds using support vector machine with psychoacoustic data. In 25th International Congress on Sound and Vibration 2018: Hiroshima Calling, ICSV 2018 (pp. 4595-4599). International Institute of Acoustics and Vibration, IIAV.