BIG DATA CLUSTERING USING GRID COMPUTING AND ANT-BASED ALGORITHM

Ku Ruhana Ku-Mahamud
Universiti Utara Malaysia, Malaysia, ruhana@uum.edu.my

ABSTRACT. Big data has the power to dramatically change the way institutes and organizations use their data. Transforming the massive amounts of data into knowledge will leverage the organizations performance to the maximum. Scientific and business organizations would benefit from utilizing big data. However, there are many challenges in dealing with big data such as storage, transfer, management and manipulation of big data. Many techniques are required to explore the hidden pattern inside the big data which have limitations in terms of hardware and software implementation. This paper presents a framework for big data clustering which utilizes grid technology and ant-based algorithm.

Keywords: big data, grid computing, ant-based clustering

INTRODUCTION

Data size has increased dramatically with the advent of today's technology in many sectors such as manufacturing, business, science and web application. Some data are structured, semi-structured while others are unstructured and mix with different types of data such as documents, records, pictures and videos (Hall, 2013). Such a huge data is defined by researchers as big data (Qin, 2012). The term big data is defined by Garlasu et al. (2013) as data that involve great volume, cannot be structured into regular database tables and are produced with great velocity. Chen et al. (2013) defined it as data sets with sizes beyond the ability of commonly used software tools to capture, manage and process within a tolerable elapsed time. Based on these definitions, it can be concluded that there are three aspects of big data namely its scale, its motion and its form. It is estimated that all the global data generated from the beginning of time until 2003 represented about 5 Exabyte and until 2012 is 2.7 Zettabytes (Garlasu et al., 2013). Many organizations start to investigate how to use and process big data to make a profit (Qin, 2012).

Resources of data are from Web applications such as Google and Facebook which produce a very big volume of data due to the big number of customers that they have. Big data can also be generated from areas such as science, finance, communication, and business. Sources of big data can be classified into human-generated data and machine-generated data. Companies such as Google, Hortonworks and Amazon tried to provide solutions for big data through the use of Map/Reduce and Hadoop (Agneeswaran, 2012). Map/Reduce is a software in distributed computing environment based on two functions called map and reduce. The functions are designed to work with a list of inputs. The map function produces an output for each item in the list while the reduce function produces a single output for the entire list. Hadoop is a software library framework for developing highly scalable distributed computing applications. The Hadoop framework handles the processing details leaving developers free to

focus on application logic. However the application of Map/Reduce and Hadoop are still in the early stages in manipulating big data.

Classification and clustering of big data is important to uncover useful information and knowledge. Traditional methods such as k-means, self-organizing maps, hierarchical clustering algorithm, partitional clustering algorithm and the expectation-maximization algorithm cannot be used for big data classification due to the limitations in the scalability of the algorithms (Das, Abraham & Konar, 2009). Hall (2013) highlighted that unsupervised clustering, fuzzy, possibilistic or probabilistic will be able to group the data. However, the algorithms scale poorly in terms of computation time as the size of the data gets large and are impractical without modification when the data exceeds the size of memory. Hardware constraints occur when processing big data such as the limitation in storage capacity and the processing speed (Garlasu et al., 2013).

Implementing big data solutions required an infrastructure which supports the scalability, distribution and management of data (Kim, 2012). Thus this study proposes grid technology to overcome the hardware limitation in term of storage space, processing power and memory capacity. For algorithm scalability, ant-based clustering algorithm is proposed. The remainder of this paper is structured as follows. Section 2 briefly reviews ant colony optimization and Section 3 presents the framework for big data using grid technology. Section 4 illustrates the ant-based clustering algorithm and the clustering integration technique is described in Section 5. Concluding remarks are presented in Section 6.

ANT COLONY OPTIMIZATION

Real ants have the ability to discover the shortest route from the nest to the food source (Dorigo & Stützle, 2004). The ants do not have an advanced vision system but they have the ability to communicate with the environment. Ants use a chemical substance called pheromones to communicate with the environment and with each other. In artificial ants, pheromone is deposited along the path where ants move looking for food. The evaporation property in artificial ants is a powerful mechanism to update the route information. The following ant will most likely select the route with richer pheromones. This mechanism will make the ant choose the shortest path. In 1992, Marco Dorigo proposed the first Ant Colony Optimization (ACO) algorithm in searching for an optimal solution in graphs to solve optimization problems such as the travelling salesman problem (TSP), job scheduling and network routing (Dorigo & Stützle, 2004).

The first variant of ACO is an Ant System (AS) (Colorni, Dorigo & Maniezzo, 1991) where the pheromone trail is updated only after all ants have constructed their solutions and the pheromone quantity deposit by each ant is calculated based on the solution quality. The first improvement on the ant system called the Elitist strategy for Ant System (EAS) (Dorigo & Stützle, 2004). The improvement was done by providing strong additional reinforcement to the arcs belonging to the best tour found since the start of the algorithm. Rank-Based Ant System (AS_{rank}) is another improvement over ant system introduced by Bullnheimer, Hartl and Strauss (1999). In $AS_{rank,}$ each ant deposits an amount of pheromone that decreases with its rank. This is similar to EAS, where the best-so-far ant always deposits the largest amount of pheromone.

Max-Min Ant System (MMAS) proposed by Stützle and Hoos (1997) has four improvements over AS algorithm. First, the best-so-far solution during the execution is exploited by allowing only one ant to update the pheromone trail after iteration. The second improvement is the implementation of the limit range of pheromone trail values to the interval $[\tau_{min}, \tau_{max}]$. Third, the pheromone trails are initialized to τ_{max} in order to achieve higher

exploration of solution at the start of the algorithm. Finally, in case of stagnation or no improved solution is generated for a specific number of iterations, MMAS will reinitialize the pheromone to τ_{max}. MMAS algorithm achieves better performance than AS algorithm because of the modification over AS structure. Ant Colony System (ACS) is then introduced by Dorigo and Gambardella (1997) to improve the performance of AS in solving TSP. In the ACS algorithm, ants apply exploitation and exploration mechanisms when they select the next node to move to. ACS also applies local pheromone updates and global pheromone updates to direct the search for the next iteration. The global update is calculated based on the quality of the best solution so far while the local update applies an evaporation concept.

Ant colony optimization algorithm has the ability to hybrid with other heuristic and meta-heuristic algorithms to solve different types of NP-hard problems. Zhang, Ning and Zhang (2012) proposed a hybrid ant colony and genetic algorithm to solve vehicle scheduling problem while Dong, Fu and Xue (2012) presented ant system-assisted genetic algorithm to solve TSP. Hybrid ant colony algorithm was also used in task scheduling problem (Wei, 2012) and resource allocation (Yunxia, 2012).

Modelling classification and clustering activities as graph seek problems permit the utilization of ACO in obtaining optimal solutions. Deneubourg et al. (1990) presented the first idea on data clustering technique that can be constructed on ideas coming from ant colonies. Ants perform collaboratively in the procedure of gathering dead bodies to preserve the cleanliness of the nest where ants try to cluster all dead bodies in a specific area. Ant-based clustering is a distributed procedure and it utilized positive feedback. In artificial ants, the ants are modelled through straight forward agents that arbitrarily move in their environment.

Ant-based clustering algorithm proposed by Deneubourg et al. (1990) has been modified by Lumer and Faieta (1994) to be applied in numerical data analysis and data mining. Data elements are randomly scattered on the grid and ants move randomly on a two dimension grid. At each step, an ant is selected at random and can either pick an element only if the ant drops an element at its current location. The probability of picking an element increases with low density and decreases with the similarity of the element. The dimension of the grid is defined as such that its number of sites should exceed the number of elements by roughly an order of magnitude. In addition, the number of elements should exceed the number of ants by at least another order of magnitude.

PROPOSED FRAMEWORK FOR BIG DATA MANIPULATION

Grid technology is used for data storage and data processing while ant-based algorithm is utilized for data clustering in the proposed framework for big data manipulation as depicted in Figure 1. The flow chart starts with various data sources sending data to the databases. These databases formalize the data intensive computing which have the ability to handle high-volume data flows (Magoules, Nguyen & Yu, 2009).Due to the requirements of data intensive applications, data storage elements are required to be able to scale to large capacities with features such as reliability and availability, flexibility, manageability and security (Brzezniak et al., 2008). An open source projects called Hadoop Distributed File System and Kosmos File System could be employed to address the requirements (Verma et al., 2011).

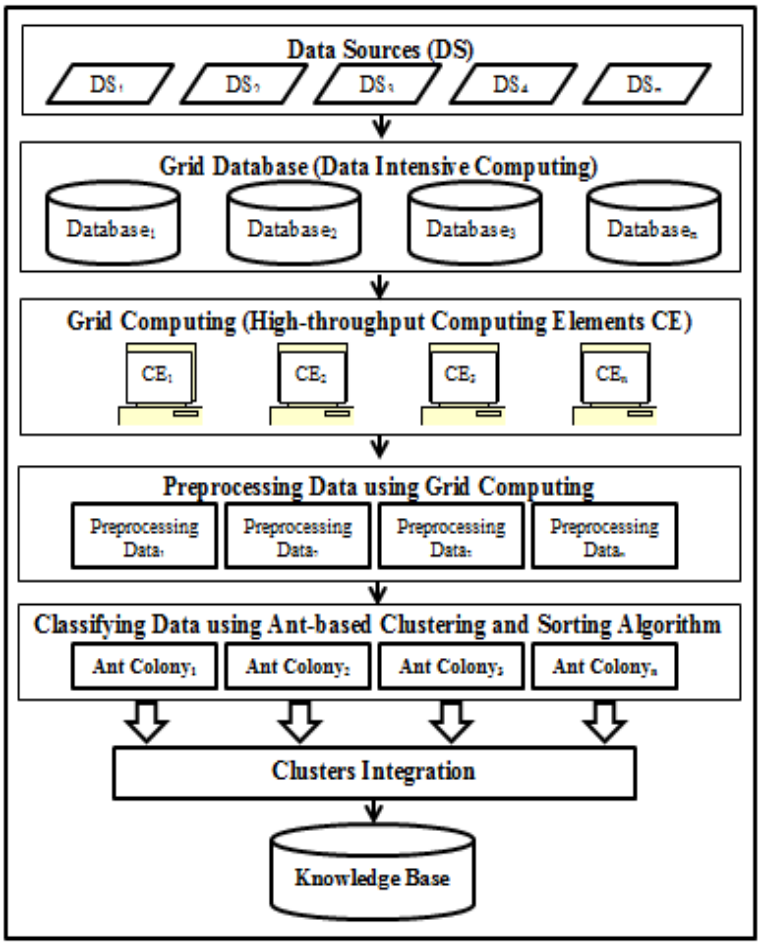

Figure 1. Grid Technology Approach for Big Data

The next layer is high-throughput computing elements which have the benefit of using unused processor cycles in the grid computing which can perform independent tasks (Magoules, Nguyen & Yu, 2009). Thus, complicated process can be divided into multiple tasks and process them by the computing elements in the grid computing environment (Hall, 2013). The divide and conquer approach will load as much data as possible to fit in the memory, cluster it and keep some type of module (cluster representation) of the data for future use. In other word, each computing element will be loaded with data from data intensive layer up to its memory capacity. For more complex grid technology scenario, the job could be divided into multiple tasks and distributed on more computing element using sophisticated resource management system or job scheduling such as co-scheduling of parallel job proposed by Madheswari and Banu (2011).

The preprocessing data stage performs data cleaning, data representation and data scaling to address outliers, missing values, inconsistent values and duplicate data. Techniques such as aggregation, sampling, dimensionality reduction, feature creation, discretization and binarization and variable transformation can be applied (Tan, Steinbach & Kumar, 2009). Clustering process can be performed once the data preprocessing is completed. Ant-based clustering algorithm is proposed as it provides a powerful nature-inspired heuristics for solving the clustering problems and this can be used in each computing element on different data set. The final layer in the framework will integrate the clusters from all computing elements and save them in knowledge base.

BIG DATA CLUSTERING USING ANT COLONY

Meta-heuristic algorithms such as ant-based clustering algorithm show very promising performance in data mining (Das, Abraham & Konar, 2009; Pancerz, Lewicki & Tadeusiewicz, 2012). The problem of finding the right number of clusters is considered as an NP-hard problem (Das, Abraham & Konar, 2009). Therefore, meta-heuristic algorithms can be applied as clustering algorithm in solving NP-hard problem. Ant-based clustering algorithm is inspired by the real ant colonies when they cluster the corpses and larvae sorting. Figure 2 depicts the pseudo-code for ant-based clustering approach in big data.

```
1: Begin
2: Initialization phase
3: Randomly scatter all data on the grid
4: While (termination condition not met) do
5:     Each ant randomly picks up one data item
6:     Each ant randomly placed on the grid
7:     For each ant (i=1, …, n) do
8:             While (ant[i] carries item)
9:                     ant[i]:= move randomly on the grid
10:                    if (ant[i] decide to drop item) do
11:                            ant[i]:= drop item
12:            End while
13:    End for
14: End while
15: End
```

Figure 2. Ant-based Clustering Algorithm Pseudo-Code

The algorithm's basic principle focuses on agents where the agents represent the ants that randomly move around in their environment which is a squared grid with periodic boundary conditions. While ants wandering around in their environment, they pick up the data item that are either isolated or surrounded by dissimilar ones. The picked item will be transported and dropped by ants to form a group with a similar neighborhood items base on similarity and density of data items. The probability of picking an element increases with low density and decreases with the similarity of the element. The idea behind this type of aggregation pheromone is the attraction between data items and artificial ants. Small clusters of data items grow by attracting ants to deposit more items. Therefore, this positive feedback leads to the accumulation of larger clusters.

The probabilities for any ant to pick and drop an item in improving the quality of the clustering are (Handl, Knowles & Dorigo, 2006):

$$P^*_{pick}(i) = \begin{cases} 1.0 & if\ f^*(i) \leq 1.0 \\ \frac{1}{f^*(i)^2} & else \end{cases} \quad (1)$$

$$P^*_{drop}(i) = \begin{cases} 1.0 & if\ f^*(i) \geq 1.0 \\ f^*(i)^4 & else, \end{cases} \quad (2)$$

where $f^*(i)$ is a modified version of Lumer and Faieta (1994) neighborhood function given by:

$$f^*(i) = \begin{cases} \frac{1}{\sigma^2}\sum_j (1 - \frac{d(i,j)}{\alpha}), & if \ (f^*(i) > 0 \wedge \forall j \ (1 - \frac{d(i,j)}{\alpha} > 0 \\ 0, & otherwise \end{cases} \quad (3)$$

The value of α is randomly selected from the interval [0, 1]. The modified $f^*(i)$ function has two important features. First, similar to the original neighborhood function $f(i)$, the division by the neighborhood size σ^2 penalizes empty grid cells which produce a tight clustering. Secondly, the additional constraint $\forall j \ (1 - \frac{d(i,j)}{\alpha} > 0$ serves the purpose of heavily penalizing high dissimilarities which significantly improve spatial separation between clusters (Handl, Knowles & Dorigo, 2006).

CLUSTER INTEGRATION

An approach called cluster ensembles proposed by Strehl and Ghosh (2002) can be used to combine multiple partitions of a set of objects into a single consolidated cluster. Three effective and efficient techniques to obtain high-quality combiners which are known as "consensus functions" will be applied. The first combiner focuses on the similarity measurement in the partitions and then re-clusters the objects.

The second combiner is based on hyper graph partitioning and the third technique collapses groups of cluster into meta-cluster which then competes for each object to determine the combined clustering. The combiner examines only the cluster label but not the original features. In other words the combiner works with the output from any algorithms that were used to obtain these clusters. Figure 3 illustrates the cluster ensemble model where X is denoted as a set of features, ϕ represents the clustering algorithm, λ represents the cluster and Γ is the consensus function (combiner).

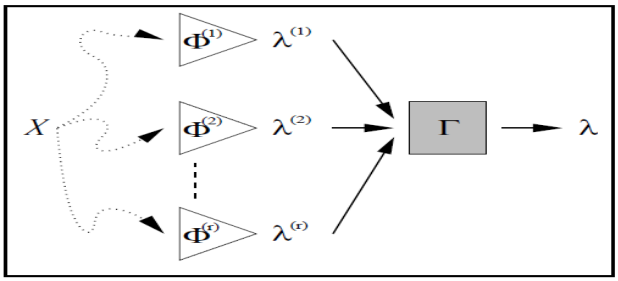

Figure 3. Cluster Ensemble Model (Strehl & Ghosh, 2002)

Clusters ensemble approach by Yang and Kamel (2006) which incorporates multi-ant colonies algorithm for clustering is suggested. The approach consists of two parts. The first part focuses on several independent and heterogeneous ant colonies where each uses ant-based clustering algorithm. In the second part, a queen ant agent (also called master) aggregates the output clusters from each ant colony using a hyper graph model which is proposed by Strehl and Ghosh (2002).

Each ant colony works in parallel and produced clusters and sent them to the queen ant agent. The queen ant agent combines the clusters to update and broadcast the similarity matrix and then the procedure is iterated.

The approach using a queen ant agent to aggregate the clusters is very suitable to be adopted. The output clusters from ant-based clustering algorithm will be sent to the queen ant agent for combination process using aggregation with hyper graph model. Based on Strehl and Ghosh (2002) and Yang and Kamel (2006) implementation, the first step is to transform the

output clusters label into a suitable hyper graph representation. A hyper graph consists of vertices and hyper edges. The regular graph edge connects exactly two vertices. The hyper graph is a generalization of an edge in that it can connect any set of vertices. The queen ant agent will construct a new similarity matrix to combine the clusters based on the similarity.

The advantage of using queen ant as an agent is that the computing of the new similarity matrix will be done centrally by the queen ant agent rather than letting all the colonies exchange information locally.

CONCLUSION

A framework for the clustering of big data using grid computing and ant colony algorithm has been proposed. The grid concept is to enable the storage of data in distributed databases across a wide geographical area while ant-based algorithm is for the clustering of big data. Ant-based algorithm has many advantages to be used in big data mining because it has the ability to scale with the size of the data set, prior knowledge of the number of expected clusters is not needed and easy to integrate with clusters ensemble model. Big data analysis opens the door for many research areas and one of the most important areas is the data security.

ACKNOWLEDGMENTS

The author wishes to thank the Ministry of Higher Education Malaysia for funding this study under the Fundamental Research Grant Scheme, S/O codes 12377 and 11980, RIMC, Universiti Utara Malaysia, Kedah for the administration of this study.

REFERENCES

Agneeswaran, V. S. (2012). Big-data – theoretical, engineering and analytics perspective. In S. Srinivasa & V. Bhatnagar (Eds.), *Big Data Analytics SE – 2*Berlin, Germany: Springer-Verlag, 7678, 8–15.

Brzezniak, M., Meyer, N., Flouris, M., Lachaiz, R. & Bilas, A. (2008). Analysis of grid storage element architectures: high-end fiber-channel vs. emerging cluster-based networked storage. In M. Brzezniak, N. Meyer, M. Flouris, R. Lachaiz & A. Bilas (Eds.), *Grid middleware and services SE* – 13, US: Springer, 187–201.

Bullnheimer, B., Hartl, R. F. & Strauss, C. (1999). A new rank-based version of the ant system: a computational study. *Central European for Operations Research and Economics*, 7(1), 25 – 38.

Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S. & Zhou, X. (2013). Big data challenge: a data management perspective. *Frontiers of Computer Science*, *7*(2), 157–164.

Colorni, A., Dorigo, M. & Maniezzo, V. (1991). Distributed optimization by ant colonies. *Proceedings of the 1st European Conference on Artificial Life*. 134 – 142.

Das, S., Abraham, A. & Konar, A. (2009). Metaheuristic pattern clustering – an overview. *Metaheuristic Clustering SE – 1*, Berlin, Germany: Springer-Verlag, 178, 1–62.

Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. & Chretien, L. (1990). The dynamics of collective sorting robot like ants and ant like robots. *Proceedings of the 1st International Conference on Simulation of Adaptive Behaviour: From Animals to Animats,* 356–363.

Dong, G., Fu, X. & Xue, H. (2012). An ant system-assisted genetic algorithm for solving the traveling salesman problem. *International Journal of Advancement in Computing Technology, 4*(5), 165-171.

Dorigo, M. & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53 – 66.

Dorigo, M. & Stutzle, T. (2004). *Ant colony optimization*. Cambridge, England: MIT Press.

Garlasu, D., Sandulescu, V., Halcu, I., Neculoiu, G., Grigoriu, O., Marinescu, M. & Marinescu, V. (2013). A big data implementation based on grid computing. *Proceedings of the 11th International Conference on Roedunet*, 1–4.

Hall, L.O. (2013). Exploring big data with scalable soft clustering. In R. Kruse, M. R. Berthold, C. Moewes, M.Á. Gil, P. Grzegorzewski & O. Hryniewicz (Eds.), *Synergies of Soft Computing and Statistics for Intelligent Data Analysis SE – 2*, Berlin, Germany: Springer-Verlag. 190, 11–15.

Handl, J., Knowles, J. & Dorigo, M. (2006). Ant-based clustering and topographic mapping, *Artificial Life*, 12(1), 35-61.

Kim, B. (2012). A classifier for big data. In G. Lee, D. Howard, D. Ślęzak & Y. Hong (Eds.), *Convergence and Hybrid Information Technology SE – 63*, Berlin: Germany: Springer-Verlag, 310, 505–512.

Lumer, E. & Faieta, B. (1994). Diversity and adaptation in populations of clustering ants. *Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, 3, 501-508.

Madheswari, A.N. & Banu, R.S.D.W. (2011). Communication aware co-scheduling for parallel job scheduling in cluster computing. In A. Abraham, J. Lloret Mauri, J. Buford, J. Suzuki & S. Thampi (Eds.), *Advances in Computing and Communications SE - 56*, Berlin, Germany: Springer, 191, 545–554.

Magoules, F., Nguyen, T.-M.-H. & Yu, L. (2009). *Grid resource management : Towards virtual and services compliant grid computing*. Boca Raton: CRC Press.

Pancerz, K., Lewicki, A. & Tadeusiewicz, R. (2012). Ant based clustering of two-class sets with well categorized objects. In S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo & R. Yager (Eds.), *Advances in Computational Intelligence SE – 25*, Berlin, Germany: Springer-Verlag, 299, 241–250.

Qin, X. (2012). Making use of the big data: next generation of algorithm trading. In J. Lei, F. Wang, H. Deng & D. Miao (Eds.), *Artificial Intelligence and Computational Intelligence SE – 5*, Berlin, Germany: Springer-Verlag , 7530, 34–41.

Strehl, A. & Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research, 3*, 583–617.

Stützle, T. & Hoos, H. (1997). MAX-MIN ant system and local search for the traveling salesman problem. *Proceedings of the International Conference on Evolutionary Computation.* 309 – 314.

Tan, P.-N., Steinbach, M. & Kumar, V. (2009). *Introduction to data mining*. New Delhi: Pearson Education.

Verma, A., Venkataraman, S., Caesar, M. & Campbell, R. (2011). Scalable storage for data-intensive computing. In B. Furht & A. Escalante (Eds.), *Handbook of Data Intensive Computing SE* - 4, New York, US: Springer**,** 109–127.

Wei. X. (2012). Study of ant colony hybrid algorithm in grid task scheduling. *Advance in Information Science and Service Sciences, 4*(5), 325-331.

Yang, Y. & Kamel, M. S. (2006). An aggregated clustering approach using multi-ant colonies algorithms. *Pattern Recognition, 39* (7), 1278–1289.

Yunxia, Z. (2012). Study on the resource allocation algorithm based on ant colony optimization. *Journal of Convergence Information Technology, 7*(16), 214-223.

Zhang, S. Ning, T. & Zhang, Z. (2012). A new hybrid ant colony algorithm for solving vehicle scheduling problem. *International Journal of Advancement in Computing Technology*, *4*(5), 17-23.