ISSN:0975-9646

Role of Distributed Systems in Data Mining

Mona Shah[#1], Dr. Hiren Joshi[*2]
[#] *Research Scholar, RK University, Rajkot, India-360020*
JG College of Computer Applications, Ahmedabad,India-380054.
[*] *Associate Professor & Director (I/c), School of Computer Science,*
Dr. BabaSaheb Ambedkar Open University, Ahmedabad,India-382481.

***Abstract*-This work is an endeavour to present the background of distributed systems and its applications in data mining tasks. The role of distributed systems in data mining is ardent and indubitable. In fact with the advent of more compact and robust technological advances in terms of communication devices and range achievable, this area sounds promising with wide applicability. With ubiquitous computing, effective and uninterrupted communication becomes an integral part of any system.**

***Keywords*— Distributed Systems, Data Mining, Network, architecture**

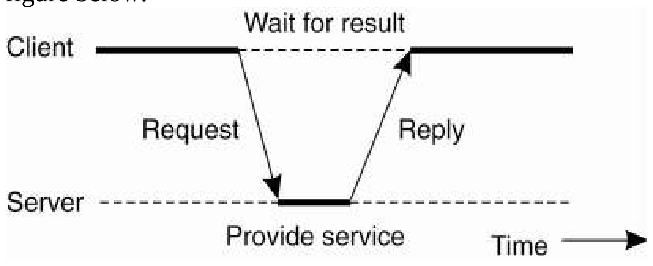
I. INTRODUCTION

Computer Networks and hence distributed systems have a quintessential accountability in every alternate and absolutely common application used globally. Practically saying standalone application or systems are hardly seen, with every modern computing device having the capability of being connected anywhere and everywhere. Applications are designed and aimed at reaching out maximum target users while exploiting networking capabilities of systems. The wireless industry brings out a variety of devices that can process different types of data. Although such devices might not have extra ordinary storage or computing capabilities, it is through the varying level of wireless connectivity, they have the power of reaching out all possible applications for distributed and mobile environments. This paper has been divided into four sections (1) Introduction (2) Backdrop of Distributed Systems (3) State-of-the Art and Related Work (4) Conclusion

II. BACKFROP OF DISTRIBUTED SYSTEMS

A distributed system comprises of a number of computers located at geographically distributed locations connected in a network communicate to share hardware, software and data with each other to accomplish a common goal. The coordination among these networked computers takes place with the help of message passing. Due to the coordination, the users perceive it as a single system. A distributed system is built on top of a network and tries to hide the existence of multiple autonomous computers. The most common examples of a distributed system are: a service oriented architecture (SOA) viz: Amazon web services which is a form of request/reply paradigm, a distributed database management system viz: online railway or air line booking system and net banking, a peer to peer application viz: skype for free audio and video connectivity or any massively multiplayer online game (MMOG) played on a variety of platforms like computers, ipads, video game consoles and smart phones. The distributed system stands apart in three significant aspects[1]: the absence of central clocks for synchronisation of processes, concurrency of components and independent failure of components. Many dimensions come into picture when it comes to variability of distributed systems namely types of processors; inter process mechanisms, timings, failure and security. Distributed systems may further be classified as (1) synchronous and asynchronous: synchronous being its natural form *per se* (2) Message passing and shared memory, which is based on the medium of communication and (3) Crash failure and Byznatine/arbitrary failures- based on the fault model.

The most evident rationale for the use of a distributed system is resource sharing although individual components might have their own - operating system, memory, resources and clock to work with and share too. Apart from resource sharing it is also the need to communicate with each other to realize a task common to all by means of sharing data and results. The way the components of a distributed system (DS) interact with each other will illustrate the architectural models. They are client server model, peer- to peer process model and spontaneous network model. There are many styles of architectural layout of a DS namely: (1) Layered (2) Object based (3) Event based (4) Shared data spaces. The layered architecture has one by one layers logically piled up above one another and the flow of request passed from the first layer (bottom most) to the last layer (top most) and similarly the reply of the request flows in the opposite direction of the request. The object based structure deploys the way of calling remote method where the interaction can happen between individual components on request and requirement. For the event based technique, the event required by one or more component is published in public and the required component can use it. In the shared data space approach, all the required methods/processes exist in a common shared pool which is contributed by different components and accessible to the members of the network. The client-server design is the most popular design existing for more than 25 years. Each member of the network who requires service will inform the server, who in turn will satisfy the request. One of the member's is a server and the rest are clients. It's a request-reply paradigm as sown in the figure below.



There can be various amendments to the client server model. It can be with more than one server or existence of proxy server interacting as middleware between the client and the actual server from where the services are fetched. A "push-model" is also prevalent where the server invokes the client and finally the mobile agents where the program migrates between computers and performs task on behalf of someone. On other hand, the peer to peer design, each node acts in dual role: client and server. Each node contributes and/or offers either data/ program to all others. The scalability issues and single point failure cannot paralyse the system are the comprehensible returns. In case of spontaneous networks, the users who can carry different devices like mobiles, laptops, PDAs between different network environments can make the most of local and remote services while on the move.

The classification of the models according to the properties that are common to all the architecture models is identified as fundamental models which have interaction models, failure models and security models. The interactions model reflects the delays in messages and processes and the ways to handle them. The synchronous and asynchronous distributed systems where in the former has a time bound limit for execution of each step in a process, while in latter case the same thing can be arbitrarily long. Many problems can be overcome with synchronous systems. Often in real life it is seen that asynchronous systems are controlled with timeout limits. A failure model focuses on the criteria in which failures may co-occur in a DS. Once the failure is detected measure are taken to prevent it or at least minimize the effect of failure. The omission failure has several cases. To name a few, the process may halt and remains undetected, the message is sent in the outgoing buffer but never reaches the other end, the message is sent but is not put into outgoing buffer ever and a message in the process incoming buffer is never received. The arbitrary failure which is also known as Byzntanine failure is a form in which desired processes may not occur or undesired ones might take place. The process or channel exhibits arbitrary behaviour. The timing failure model occurs in synchronous DS where the client does not receive response in a specified time interval.

III. State Of Art And Work In Few Domains

MuthuKrishnan and Rutgers[2] address the issue of challenges faced in the data analysis of the advertisements (ads) generated on the internet. Different parameters are involved in the mentioned process and knowledge from multiple domains becomes essential. How are the ads are decided, who decides it, which slots are allotted, how does the bidding take place? The problem lies in analysis of the advertisement on the internet in a way that they rely on real time data collection and in addition to that the publishers and the marketers generate new data and distort existing data for malicious motives. The real time data collection task in the current problem which involves large amount of continuous data (VLDB) is inherently taxing in terms of efficacy and robustness. It requires strong and proved network and powerful distributed system working over different regions or domains which will fetch the end result.

Kargupta, Sarkar and Gilligan[3] talk about the performance of data mining systems designed for commercial fleets. Minefleet, a software for commercial fleet owners, models, benchmarks, monitors and analyses high throughput data streams regarding vehicle health, emissions, driver behavior, fuel-consumption, and fleet characteristics. Then those are sent to remote server over wide wireless networks, which aid the fleet managers. Minefleet is one such successful mobile and distributed data stream mining application. The performance of a vehicle connected to the software Minefleet is done onboard and it sends the results of the onboard analysis to the server over wireless network. Since the analytics happens at the vehicle onboard, there is significant reduction in the cost of wireless communication and yet mining gets all the possible data. MineFleet also tenders different distributed data mining capabilities for detecting fleet-level patterns across the different vehicles in the fleet.

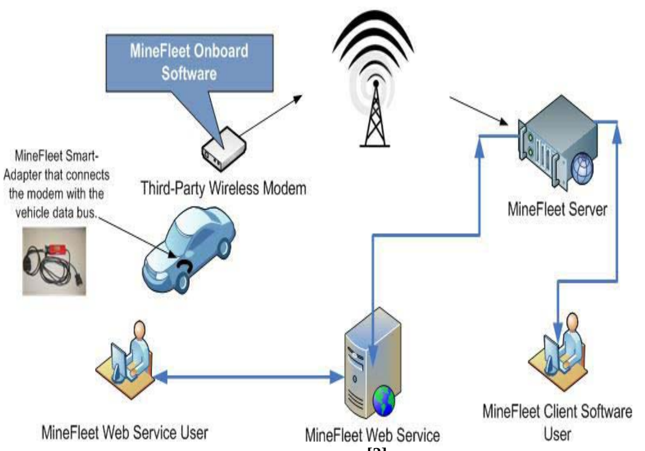

MineFleet Onboard Software
MineFleet Smart-Adapter that connects the modem with the vehicle data bus.
Third-Party Wireless Modem
MineFleet Server
MineFleet Web Service User
MineFleet Web Service
MineFleet Client Software User

Figure 1 [2]

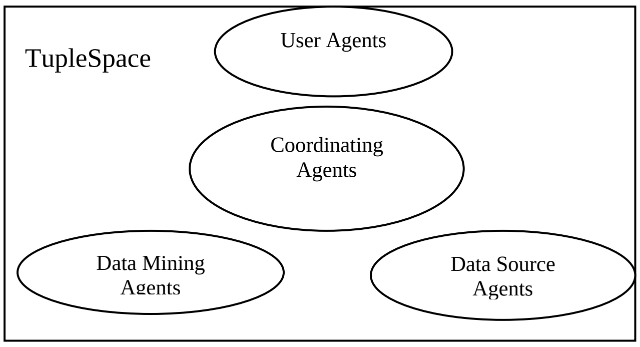
The diagram above is the architecture of Minefleet software where a minefleet client software user has onboard device which does the job of data management and mining task in real time. If there is any unusual pattern detected it is reported to the Minefleet Server. The Minefleet server receives analytics from different vehicles, manages and it stores the data received from various fleet in a RDBMS. The web services components offer some smart API functions for analytics which can be integrated with third party applications.

A team of 12 member [4] come up with a fresh idea with a system for data mining system in distributed environment and named it as **F**(fast)**I**(integrated)**U**(user friendly)-Miner. The key features of this system are cross language program integration, compatibility to accommodate data mining tasks in heterogeneous environments. The system architecture comprises of a user interface layer, task and system management layer and resource abstraction layer. The user interface layer has task configuration and execution module which is meant for workflow oriented task configuration. The workflow is denoted as a directed graph where each node is a specific data mining algorithm and the edges show the relationship for data dependency among the algorithms. The user has to configure his data mining algorithm requirement through this GUI interface. The second module of user interface layer allows the user to import a new algorithm written in any language as long as its runtime is supported by backend servers which is a list of current enabled technologies ranging from Hadoop, Java, Python, shell, C and C++. The system monitoring module of user interface takes care of tracking the status of submitted jobs and scrutinizes the resource utilization which is transparent to the users. Task management and system management are modules of the system management layer. The task management is concerned with the user selecting the appropriate algorithm or a combination from the algorithm library which is then added to the list of scheduling tasks The job manager keeps track of the running status of a executed job. In the resource abstraction layer, each server storage is shared by all tenant workers.

Giuseppe Di Fatta and Giancarlo Fortino [5] present the idea of multi-agent system framework for a distributed data mining based on a peer to peer model. The framework adopts a dynamic load balancing policy that is particularly suitable for irregular search algorithms. The architecture has two kinds of nodes: (i) the processing node (2) the coordinating node. There is only coordinating node which plays the role of observing the work of remaining processing nodes with the help of generating statistics. The processing nodes perform local level mining, exchange data and partial results with other nodes. The network performance, capabilities become a strong and highlighted feature in this model when it comes to exchange of data and/or results among participating nodes or collecting statistics by the observer node.

Manuel, Wesley and Henrick[6] introduced a distributed data mining approach appropriate for grid computing environments. Two levels of data mining models are generated. One at local level for each site in the distributed system and from all the local model is generated the global model. The solution obtained is more robust and reliable in case the classifiers are available from the training data set. Because of the available classifiers the process of incremental learning takes place which as a result improves the performance of the global model. In the other case, where the classifiers are not provided, the system derives them from the training data. The problem arises is of identifying the significance of each local classifier in the global model. They come up with six different strategies to construct a global model from local model and found one of them to be superior in terms of accuracy.

Andrew Wong and Nick Cercone[7] have given the implementation of the multi-agent system (MAS) proposed earlier by Ni and Chang[8] exploring the real power of distributed computing for mining stock data. The tuplespace paradigm has been used here. MAS has the power of explicit divisions of labor and independence and processing multiple different requests at the same time. The concept of Blackboard shared memory has been used for creation of distributed MAS. The system breaks down the data mining tasks into elementary levels and this division allows for agents to be specially crafted to perform one task to the best of their ability. Different agents are available like controlling agent, data mining agent, user agent, coordinating agent and data source agent. The agents communicate with each other by passing messages. Each agent has specific role. The user agent places his request to find the moving average of a stock which is forwarded to the tuplespace. In the tuplespace, one or more agents are always listening to such requests of which one of them will pick it up and remove the request from the tuplespace. The 'take 'method' will read and remove the tuple from the tuplespace and the 'write' method will place a tuple into the tuplespace. The data source agent is responsible for retrieving stock information from Yahoo finance. The coordinating agent is responsible for administering the tuplespace. The message between the agents in the tuplespace has request/reply syntax. There can be more than one tuplespace also. The data mining agent are used to calculate the moving average of any stock. This type of multiagent system has a solution to get extended in providing solutions for long term, short term, combinational computation and frequent and short computations. The figure [6] below is the representation of a multi-agent system.



Shanmugam and Mohamed MA[9] propose a solution for distributed mobile systems. The mobile system considers object models which define a method that allows mobile devices to participate effortlessly in mining process. It permits to cache the location based date sets and retrieve the knowledge from those data sets to solve the location management problem. There is an issue under address that the data mining in distributed mobile systems suffer from repetitive scanning of large data set located at the mining server. The trade-off between finding the location of a mobile device and performing the mining in a timely manner prevails. A large dataset is exchanged and hence the overhead cost of network, communication and time becomes a bottleneck. The concept of surrogate object is used which is in itself an architecture which allows mobile devices to participate easily in computing and communication. The obvious advantages of surrogate object are better speed of data access; mobile location is stored and can be accessed as a lookup table in a static network which will solve the location management problem and delivering appropriate information to the mobile device depending upon the current location of the device and connectivity constraints by serving as data collection agent.

The surrogate object based data mining (SODM) model has a fixed mining server, a mobile device that generates the data to be mined, a mobile support station (MSS), a surrogate object (SO) created on MSS to act on behalf of mobile device and lastly a data provided that generates the data to be mined for non-contextual applications. The process begins with the mobile device sending the context and location dependent data to be analysed. The surrogate objects is allocated using object table which is decided when the mobile device joins the mobile environment. The data collection manage(DCM) of the respective MSS collects the data and the data mining manager (DMM)passes this data set and SODM algorithm to the respective surrogate object. The data mining task is then preformed and the results sent to the mobile device. The experiment has proven the concept of surrogate object giving better results in terms of reducing mining delay

Cheung, Zhang, Wong, Liu, Luo and Tong[10] adopt a solution for service oriented distributed data mining (DDM) through Business Process Execution language for Web services (BPEL4WS). Local data abstraction and global data analysis is the highlight of the model. It aims at reducing the overall complexity of the mining process yet support adaptive data mining inherently. Service oriented architecture (SOA) is a huge welcome because of its potential to focus on services rather than interfacing details. Secondly it is easily extendible and modifiable for new DDM applications wherein one needs to control the flow of newly added services.

IV. CONCLUSION

A general distributed data mining framework can enable and accelerate the deployment of practical solutions. The fast paced technology in terms of wireless communications establishes a strong platform for more and more distributed systems and its viability will be seen as sound and gullible. Distributed systems can accomplish the task of choosing from a pool of solutions to achieve the result and choosing the best possible solution according to the requirement. Application specific methods can be offered for the execution of a task or partition of a task in two subtasks.

REFERENCES

[1] http://en.wikipedia.org/wiki/Distributed_computing.

[2] Muthukrishnan S., Rutgers U., and Google Inc," Data Management and
Mining in Internet Ad Systems", Proceedings of the VLDB Endowment, Vol. 3, No. 2 Copyright 2010 VLDB Endowment

[3] Kargupta Hillol, Sarkar Kakali, Gilligan Michael: MineFleet®*,"An Overview of a Widely Adopted Distributed Vehicle Performance Data Mining System", *KDD'10,* July 25–28, 2010, Washington, DC, USA. Copyright 2010 ACM

[4] Zeng Chunqiu, Jiang Yexi, Zheng Li, Jingxuan Li, Lei Li, Hongtai Li, Chao Shen, WubaiZhou, Tao Li, Bing Duan, Ming Lei, Pengnian Wang, "FIU-Miner: A Fast, Integrated and User-Friendly System for Data Mining in Distributed Environment", *KDD'13,* August 11–14, 2013, Chicago, Illinois, USA. Copyright 2013 ACM

[5] Giuseppe Di Fatta, Giancarlo Fortino, "A Customizable Multi-Agent System for Distributed Data Mining", *SAC'07,* March 11-15, 2007,Seoul,Korea Copyright 2007 ACM.

[6] Santos Mantos, Mathew Wesley, Santos Henrique,"Grid Data Mining by means of Learning Classifier Systems and Distributed Model Induction" *GECCO'11,* July 12–16, 2011, Dublin , Ireland. Copyright 2011 ACM

[7] Andrew Wong, Nick Cercone, "A distributed multi-agent System implementation of the human friendly MAS for mining stock data", 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.

[8] J. Ni and Zhang C,"A human-friendly mas for mining stock data", In WI- IATW '06: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology pages 19–22, Washington,DC, USA, 2006. IEEE Computer Society.

[9] Shanmugam Ravimaran, Maluk Mohamed.M.A,"Surrogate Object Based Data Mining for Distributed Mobile Systems", 5- 7 December, 2011, Ho Chi Minh City, Vietnam..Copyright 011 ACM.

[10] William K. Cheung, Xiao-Feng Zhang, Ho-Fai Wong, and Jiming Liu, Zong –Wei Luoand Frank C.H. Tong," Service Oriented Distributed Data Mining", JULY. AUGUST2006 Published by the IEEE Computer Society 1089-7801/06/$20.00 © 2006 IEEE, IEEE INTERNET COMPUTING