Agents System and Its Suitability for Distributed Applications:- A Practical Approach.

U. P. Kulkarni †, P.D.Desai. †, J.V.Vadavi †, A.R.Yardi ††

† Department of Computer Science & Engineering, S.D.M College of Engineering and Technology, Dharwad, India.
†† Dy.Director, Walchand College of Engineering and Technology, Sangli, India.

Summary

Agent systems offer significant advantages for modeling complex adaptive systems. However, they are not the solution to all complex problems. It is important to note the strengths, weaknesses and implementation issues related to agent based systems. With the advent of the Internet era and advancement of technology, the scope and the feasibility of mobile entities (code) have increased dramatically. As mobility can be deployed in many applications including e-commerce, the interest in this area of research has intensified. Over the past decade, data mining has gained an important role in analysis of large datasets and there by understanding the complex systems in almost all areas. Such datasets are often collected in a geographically distributed way, and cannot, in practice, be gathered in to single repository. Existing data mining methods for distributed data are of communication intensive. Many algorithms for data mining have been proposed for a data at a single location and some at multiple locations with improvement in terms of efficiency of algorithms as a part of quality but effectiveness of these algorithms in real time distributed environment are not addressed, as data on the web/network are distributed by very of its nature. As a consequence, both new architectures and new algorithms are needed. Our aim in this article are to state when and where exactly the mobile agents system can potentially play a significant role in complex distributed-systems by stating few quantitative measurements of the value of mobility and emphasize the distributed application focusing on distributed data mining.

Key words:
Agent, Data mining, Software Agent, Mobility.

1. Introduction

A mobile agent is a running program that can move from host to host in a network when and where it chooses [1, 2, and 3]. Mobile agents are one form of mobile code. In its simplest form, the concept of mobile code involves dynamically installing code on a remote host. In Web applications, applets and servlets are a common form of mobile code. The mobile code concept also appears in systems that extend the notion of remote procedure calls to transport the procedure to the server along with the call, remote evaluation. Mobile code promises to increase system flexibility, scalability, and reliability. To date, however, this promise has been only partially fulfilled [1]. Among the reasons for the technology's unmet potential are security concerns [2, 3, and 4] and incomplete knowledge of the possible consequences of mobile code use [1]. The agent-oriented approach is increasingly being applied in industrial applications, but it is far from as widespread as the object-oriented approach [8]. Agent based Systems are being developed using object oriented techniques, and there will be fewer agents in the systems than objects. Agents oriented views of software development complement the Object oriented software development and not to replace it. Agent oriented technologies are more appropriate than object oriented technologies for the applications that can be naturally modeled as societies of interacting autonomous entities. The foremost decision a system designer has to make is, when to use agents to solve problem. It is important to note that not all problems benefit from agent based approaches. Thus, it is crucial to identify the key characteristics which make a problem suitable for agent solutions. Agent based system closely represent how natural systems work by distributing a problem among a number of autonomous entities. Thus, if the problem cannot be effectively divided into a series of interacting sub problems or sub goals, an agent approach may not be successful. The agency approach offers the power of problem decomposition and parallelism for tackling complex problems. The autonomous nature of agents also offers flexibility in dealing with uncertain situations and localized failures. However, agent based systems require significant life support and monitoring overhead. Agent intelligence and learning issues have to deal with balancing requirements and resources, such as solution quality, memory, computation time, communication. Dealing with errors, misinformation and rogue information is an important issue with agents, especially with the use of agents in electronic commerce. System designers need to pay close attention to the authentication and data consistency issues, and incorporate evaluation mechanisms in the agent architecture. Agent based systems, like other asynchronous distributed environment, are faced with the system level issues, such as stalling, deadlock, cycling, etc. Arguably, a centralized observer or controller is the antithesis of the distributed decision making qualities of agents. Most real world agent

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

applications need some level of system monitoring in order to ensure error free operations. Agent systems can be hard to debug due to their distributed asynchronous nature. Hence special attention needs to given to diagnostic tools for agent based systems before they are deployed in the real world. The question of designing best agents for a particular task still remains unanswered. Moreover, the design of a society of agents which is ideally suited for solving a complex problem requires the use of prescriptive tools for analyzing distributed agent systems. Traditionally, simulation has been used to analyze the performance of agent based systems under different operating conditions before they are implemented in the real world. Simulation is, however, only a descriptive tool. The computational expense of exhaustive testing of all operational scenarios may be significantly high, if at all possible. The difficulty in developing prescriptive analytical tools for agent based systems is due to the non equilibrium conditions under which they operate, the difference in the information available to each agent for decision making and, more practically, the sheer size of some of the real world agent system applications. There are good arguments in support of the claim that agent technology will prove to be a valuable tool for building complex distributed systems. But as yet, these arguments are unsupported by much substantial evidence: agent technology is essentially immature and untested. With no body of experience to guide them, agent system developers tend to find themselves falling into the same traps. This article focuses on the experimental evidences of effectiveness of Mobile agents for distributed computing applications with a several experimental studies which includes: sorting, searching, data transfer and distributed data mining applications.

Data mining is the new technology of discovering the meaningful information from the data repository, which is widely used in almost all domains, which includes: finance, insurance, process control, quality supervising, engineering and scientific data analysis etc. Recently, mining of databases has attracted a growing amount of attention in database communities due to its wide applicability in retail industries in improving marketing strategies. Analysis of past transaction data can provide very valuable information on customer behavior and business decisions. The amount of data stored grows twice as fast as the speed of the fastest processor available to analyze it. The problem of extracting the knowledge is harder for large datasets due to the following properties:

- Datasets are distributed geographically, so pieces of the same logical datasets are physically located far away from each other.
- Datasets are immovable in practice.

Because of these properties, algorithms to compute with large datasets cannot assume or control the partitioned structure, the sizes, and the locations of the pieces of the datasets and must take account of the latencies and bandwidth required to move data among the pieces. The analysis of large datasets has become an important tool in understanding complex systems in areas such as economics, business, science and engineering. Such datasets are often collected geographically distributed way and cannot in practice be gathered in to a single repository. Applications that work with such datasets cannot control most aspects of the data's partitioning and arrangements. So far, attention in data mining process has always focused on extracting information from data physically located at one central site and they often do not consider the resource constraints of distributed and mobile environments. Few attempts were also made in parallel data mining. However most real life applications rely on data distributed in several locations. As a consequence both new architectures and new algorithms are needed. In this paper author proposes a method that explores the capabilities of mobile agents to build an appropriate frame work and an algorithm that better suits the distributed data mining applications. It also makes the performance analysis and comparison with the existing such method.

This article is organized in to two major parts; the first part i.e. section 2 of this article addresses the agents' capabilities and its suitability in basic operations of any system. The second part i.e. section 3 of this article addresses the use of agents in building effective mechanism for distributed data mining.

2. Mobile agent paradigm.

The effectiveness of mobile agents for distributed computing applications was done as a set of experiments using IBM's AGLET, an agent based open source platform for application development and observations made are as under:

1. Agents are not suitable for data transfer as it is done in conventional systems.

2. Agents are better suited for applications where the job is dispatched to carry upon the remote data rather than bringing the data and performing the computations.

3. Mobile agents (AGLET) are not enough matured to perform distributed computing applications in specific contexts.

In experiment 1, the agent was made to travel through several host in a network of computers and round trip time (RTT) was measured. Following observations (Fig-1) were made:

- As the size of the agent increases, the network latency and RTT also increase.
- After some size of the agent, the latency and RTT vary minimally even if the size of the agent is increased further
- Agent fails to dispatch at 6291455 Bytes. The reason being the constraints of Java Virtual Machine (JVM) on the size of the object.

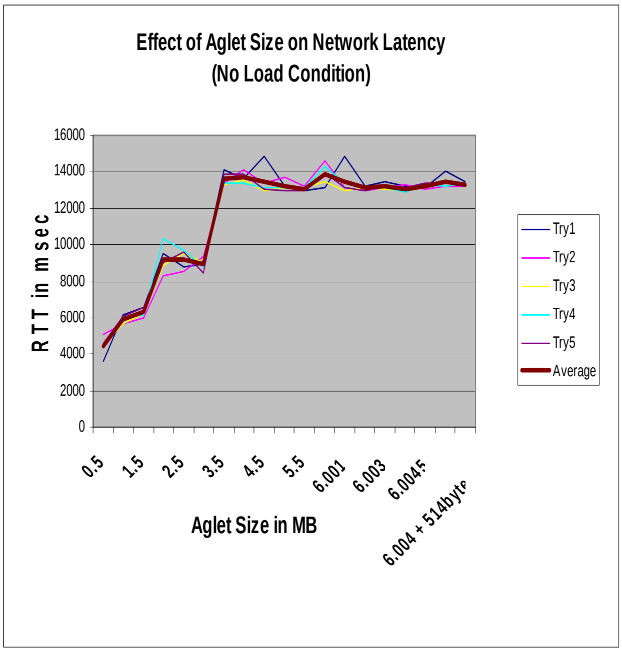


Fig-1

In experiment 2, the file of different size was transferred over a network using traditional File Transfer Protocol (FTP) and also using agent and the time was measured. Following observations (fig-2) were made:

- Agents are not suitable for huge data transfers.
- The poor performance of the mobile agents is due to the object serialization and deserialization process, in which object is marshaled and unmarshalled
- This justifies the statement that, when very large volumes of data are stored at remote hosts, these data should be processed in the locality of data rather than transferred over the network.

If the chunk-size of the agent is increased, agents perform better than FTP as shown in fig-3.

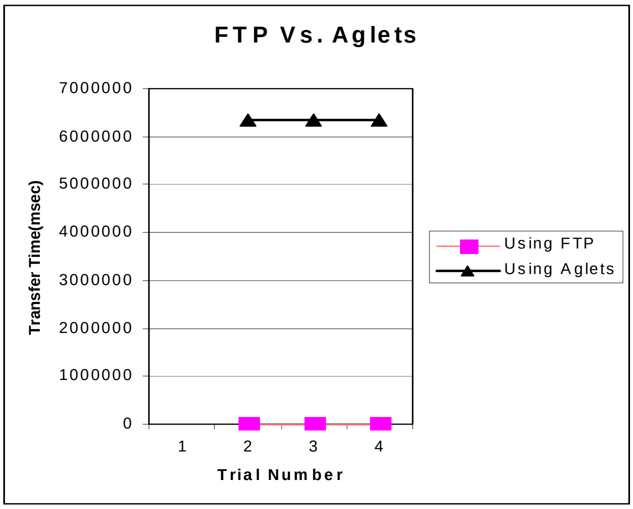


Fig-2

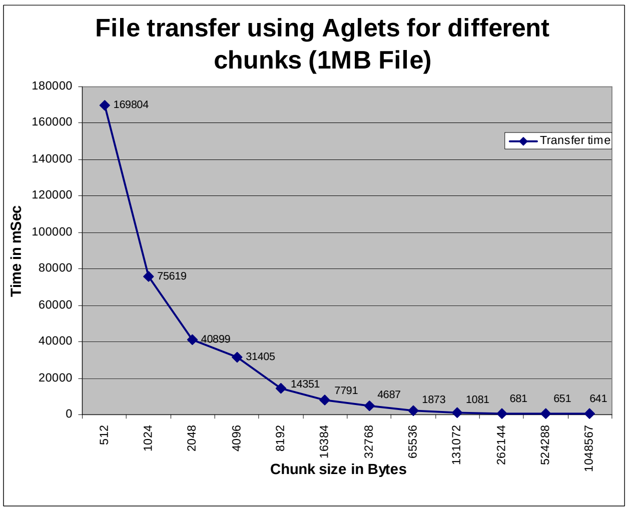


Fig-3

In experiment 3, the suitability of application of mobile agents in applications that involves huge computations was evaluated.

Case I: refers for the agent goes to the remote data, performs computations and brings the result back and Case II: refers for the agent brings the remote data, performs computations locally and obtains the result. Following observation (fig-4) was made:

- When very large volumes of data are stored at remote hosts, these data should be processed in the locality of the data rather than transferred over the network. The motto is simple: move the computations to the data, rather than the data to the computations

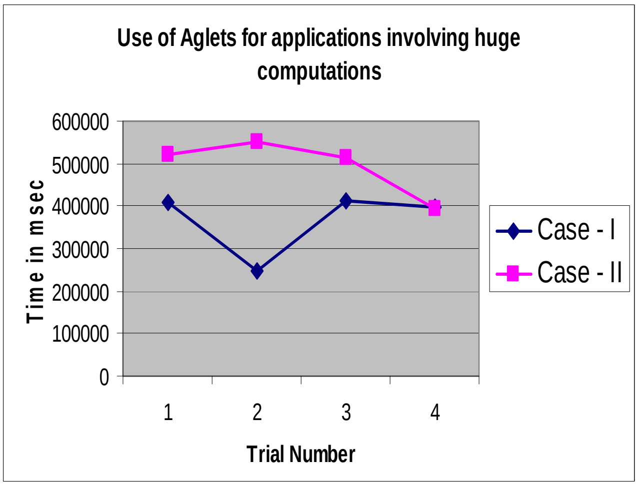


Fig-4-a (File Size: 1 Lakh elements)

In experiment 4, investigation of the effectiveness of using a mobile agent in searching for a key element was done and performance behavior was compared with the traditional searching approach. Following observations (fig-5) were made:

- Mobile agent technology is a viable and suitable approach for applications that involve distributed computing
- Parallel execution of agents for searching shows a uniform performance in execution time.

In experiment 5, investigation about the effectiveness of using a mobile agent in sorting a huge data file (centrally available and was distributed to various machines using agents to use their computing resource) was done and its performance behavior was compared with the traditional sorting approach. Following observations were made:

- As the data subsets are dispatched to remote nodes for sorting, the marshalling and unmarshalling takes time.

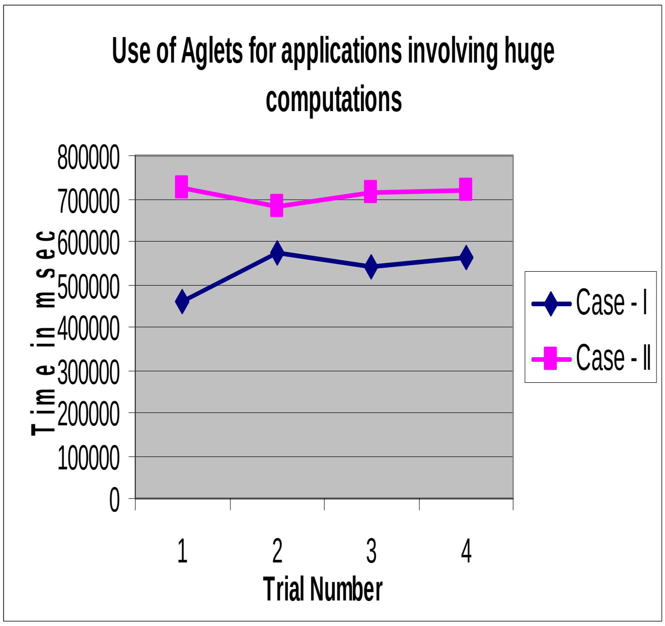


Fig-4-b(File Size: 2 Lakh elements)

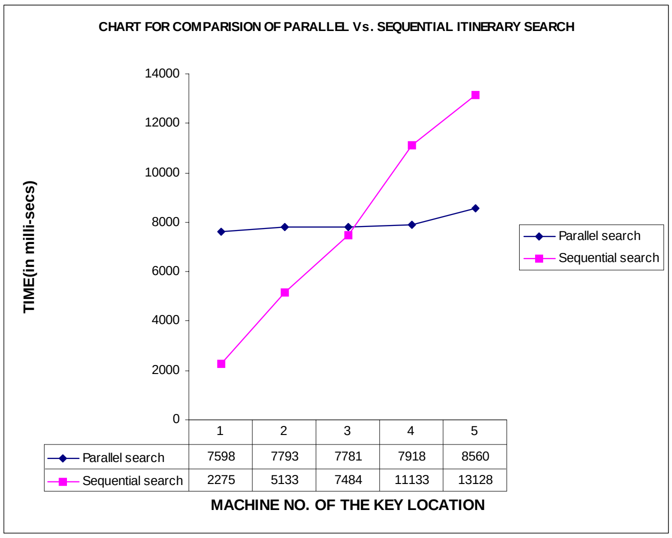


Fig-5

- Hence we observe that distributed sorting using agents are not a viable approach as seen in fig-6.

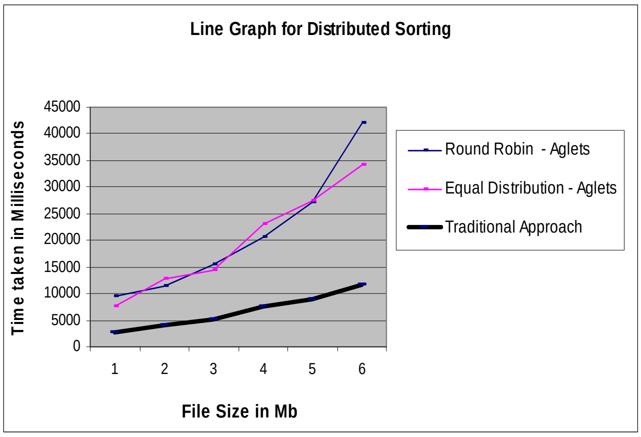


Fig-6

In experiment 6, effectiveness of algorithm for mining association rules in real time distributed environment were addressed, as data on the web/network is distributed by very of its nature. Following observation was made:

- Mobile agents are better than traditional approaches as seen in fig-7.
- When compared to the existing applications of similar kind this proposed approach proves to be one of the best and robust methods to handle distributed data and mine global association rules in faster way.

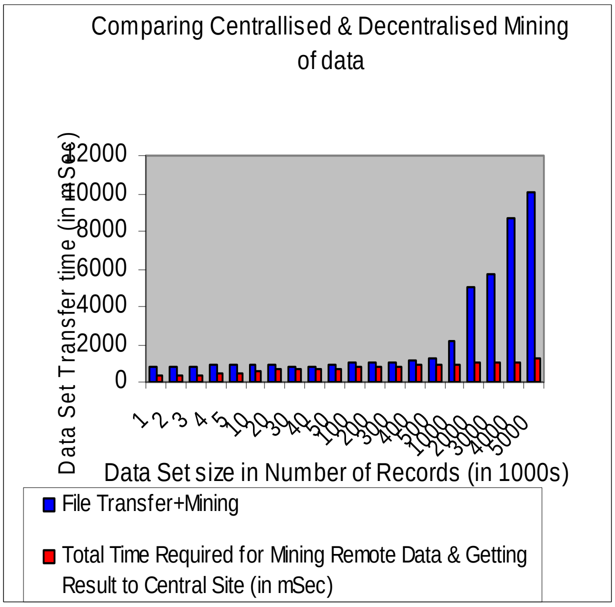


Fig-7.

The section 3 of this article describes the effective methodology of implementing this distributed data mining using mobile agents.

3. Agents' based distributed data mining.

Several Distributed Data Mining (DDM) systems using client server model, agent-based model and grid-based architectures including the parallel mining on clusters were proposed. DDM system based on client server model includes: Decision Center [9], IntelliMiner [10]. All these are based on the technologies like: CORBA, DCOM, EJB, RMI, JDBC etc. Agent-based DDM [11] includes JAM, BODHI, and DAME. Section II of this paper highlights the capabilities of Mobile Agents in building the various distributed applications, particularly focusing on the Distributed Data Mining. But the issues of consolidating the knowledge collected from various distributed sites are not being addressed.

These existing approaches for DDM suffer from one or more of the following limitations [11].

- Lack of coordination among the distributed sites while generating the local knowledge, affects the quality of Global knowledge.
- Non-flexibility in addition of new algorithm to its knowledge base.
- Capability to dynamically discover data sites based on user requirements.

The paper [12] proposes the parallel algorithm for mining frequent item sets, which uses two steps with some scope for further improvements. This article suggests the improvements over the method proposed by You-Lin Ruan et al [12] and uses the advantages of mobile agents [13] over client server based approaches in terms of better bandwidth usage and network latency.
To overcome all these, a mobile agent based DDM has been proposed. Here knowledge from distributed sites are extracted in the form of association rules. Based on the types of values, the association rules can be classified into two categories:

- Boolean Association Rules : Keyboard ➔Mouse [Support = 6%, confidence = 70%]
- Quantitative Association Rules: (Age = 26 …30) ➔(Cars =1, 2) [Support = 3%, confidence = 36%]

Generally the DDM process consists of following steps:

1. Generate the knowledge locally at each distributed sites.
2. Integrate the local distributed knowledge to get global knowledge.
3. Quality check on the global model.

The notations used in the methodology proposed are as below:

- DB →Database.
- D →Number of Transaction.
- n →Number of sites (S1, S2, … Sn).
- DBi →Distributed Data sets at Si , DB =U DBi, i= 1 to n.
- X.Sup →Support count of a X at DB –Global.
- X.Supi →Support count of a X at DBi –Local.
- Minsup →Minimum support threshold.
- GFI →Global Frequent Item Set.
- CGFI →Candidate Global Frequent Item Set.
- X →Global Frequent Item iff, X.Sup >= minsup * D.
- X →Local Frequent Item iff, X.supi >= minsup * Di.
- LFI i →Local Frequent Item set at site-i.
- PGFI →Possible Global Frequent Item Sets- These are item sets at sites-i, which are not part of LFI i, but by adding these count at central place conver CGFI to GFI.

The method proposed by You-lin Ruan et al[12] which minimize scans of the database and data is easy to use and update. Moreover it makes each processor to process independently and decrease the number of candidate global frequent item sets according to the relation between local frequent item sets and global frequent item sets. It basically uses 2 steps as shown in figure-8.

1. Mining Local Frequent Item sets (LFI) at each site in parallel and send them to central site to calculate Global Frequent Item sets.
2. Central site calculates the Candidate Global Frequent Item sets-CGFI and send them to all sites. Each local site computes the count of such CGFI and sends them back to central site to complete the process of finding GFI.

DDM Algorithm-1:

Input : Distributed-Data-Set DBi i=1 to n, minsup, Distributed sites' address.
Output : Global Frequent Item set – GFI

1. Send Mining Agent (MA) to all sites with support value.
2. Each Cooperative MAi, Computes LFI i in parallel.
3. Send LFIi (i=1 to n) to central site.
4. Compute GFI & CGFI: GFI=∩ LFI I, CGFI= Ù LFI i - ∩ LFI I, i=1 to n.
5. Add an item set to a GFI, if its count in frequent sites is greater or equal to minimum support value.

```
For all X ε CGFI do
{
    If Σ i= 1 to n   X.Sup i >= Minsup * D    Then
    {
        GFI = GFI Ù  {X};
        CGFI = CGFI - {X};
    }
}
```

6. Send CGFI to each site Si , i=1 to n.
7. Compute counts of any item set X .

```
        (X ε CGFI) in infrequent sites.
        for i = 1 to n do
        {
          Search X at site Si;
          Get X.Supi and send to central site.
        }
```

8. Send count of each CGFI to central site.
9. At Central site: Compute Global Counts of each item set in CGFI

```
For all X ε CGFI do
{
  If X.Sup = Σ X.Sup I, I= 1 to n >= Minsup * D then
  {
    GFI = GFI Ù  {X}
  }
}
```

Proposed Algorithm:

The proposed algorithm is an improvement over above DDM-1 approach. The basic objective is to reduce the time required to compute GFI.The proposed algorithm performs two tasks parallel.

1. Local sites send LFIs to central site and also to all their neighbors.
2. Calculation of GFI/CGFI at central site and counts of CGFI at local sites is done as a overlapped operation. That is, local sites need not wait for central site to send CGFI. Thus total time taken is reduced drastically.

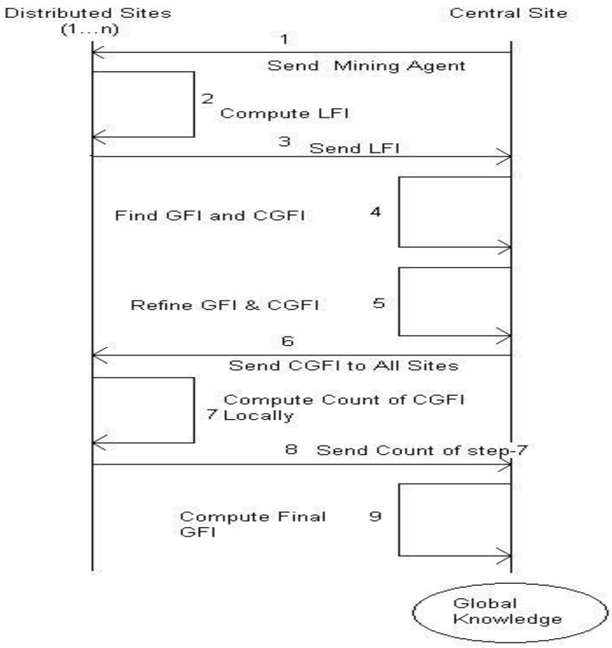


Fig-8

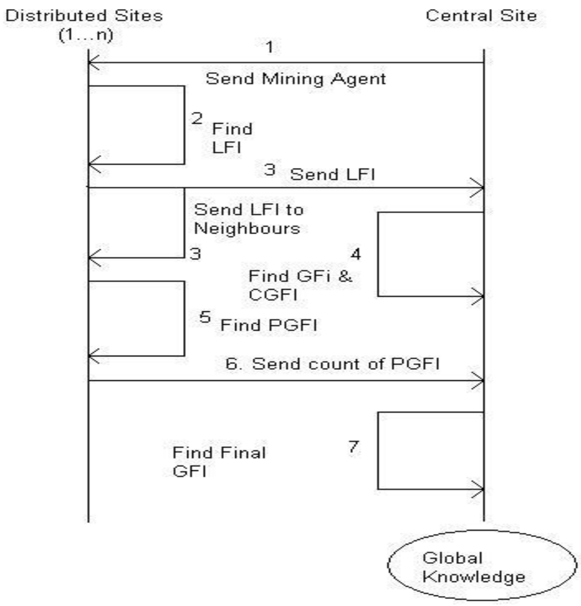


Fig-9

DDM Algorithm-2:

Input : Distributed-Data-Set DBi, i=1 to n, Minsup.
Output : Global Frequent Item set – GFI.

1. Send Mining Agent (MA) to all sites:
   For I= 1 to n do
   {
   MA.send (Location = I, S=Support, Addresses of all distributed sites);
   }
2. Each Cooperative MAi Computes LFI I in parallel.
3. Send LFI to central site and also to all its neighbors.
4. Compute GFI & CGFI at central site.
   $GFI = \cap LFI_{i \;\; i=1 \text{ to } n}$;
   $CGFI = \grave{U} LFI_i - \cap LFI_I$, $I = 1$ to n.
5. Calculate PGFI and their count at each site.
   $PGFI_{j=\text{all sites}}$ = All Item Sets at site-j $\cap LFIi_{\; i=1 \text{ to } n, \; i<>j}$,
6. Send count of PGFIi, i=1 to n to central site from each infrequent site.
   Note: Step-4 and Step-5, 6 are performed in parallel.
7. Calculate GFI at central site using PGFI count.
   For all X ε CGFI do
   {
   If X.Sup = Σ X.Sup i,i= 1 to n >= MinSup * D then
   {
   GFI = GFI Ù {X}
   }
   }

Note: step-6 of DDM algorithm-1 is totally eliminated in DDM algorithm-2.

Performance Measurement:

The proposed algorithm uses the basic approach of finding GFI as proposed by You-Lin Ruan et al, including the terminologies used and adds overlapped activity, thereby reducing the time taken to mine geographically distributed data.

Notations used for timing analysis are:

- Ts – Time required to send 'minsup' from main site to all distributed sites.
- Tc-LFI – Time required to calculate LFI at all distribute sites.
- Ts-LFI-m– Time required to send LFI from all distributed sites to central site.
- Ts-LFI-n – Time required to send LFI from all distributed sites to their neighbors.
- Tc-C/GFI-m –Time required to find GFI and CGFI at central site.
- Ts-CGFI-ds – Time required to send CGFI from central site to all distributed sites.
- Ts-c-CGFI-m – Time required to find count of

CGFI at each distributed sites and send to central site.

- Ts-PGFI-m – Time required to find and send PGFI and its count to central site.
- Tco-P/CGFI – Time required to convert CGFI to GFI using count of PGFI Received from all distributed sites.
- T1 & T2 – Time required to find GFI at central site using existing and proposed methods respectively.

Total time required for calculating GFI using DDM algorithm-1 is:
T1 = Ts + Tc-LFI +Ts-LFI-m + Tc-C/GFI-m + Ts-CGFI-ds + Ts-c-CGFI-m + Tco-P/CGFI

The proposed DDM algorithm-2 does not use Ts-CGFI-ds and Ts-c-CGFI-m as central site is not sending CGFI. Local sites get this information from its neighbors. Thereby communication time is reduced, which results in reduction of total time to find GFI. Thus total time taken by this DDM algorithm-2 is,

T2 = Ts + Tc-LFI + Ts-LFI-m + Tc-C/GFI-m + Tco-P/CGFI

It is clear that T2<T1 due to less synchronization steps and parallelism, which is also evident from the experimental results conducted in Local Area Network, as shown in figure-10. Figure-11 shows the GUI of the system implemented.

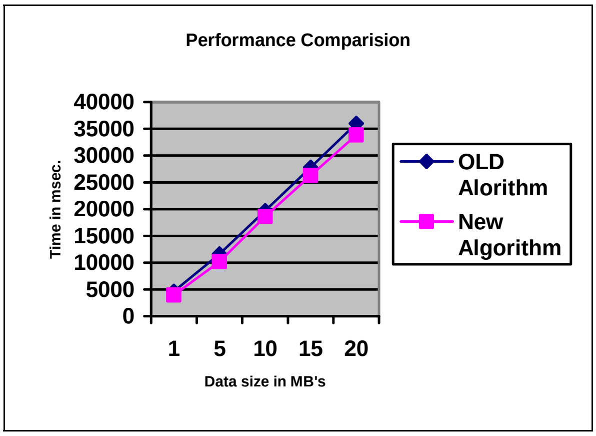


Fig-10

4. Conclusion.

The experiments conducted indicate that mobile agents are not suitable for data transfer as it is done in conventional systems. They are better suited for applications where the job is dispatched to carry upon the remote data rather than bringing the data and performing the computations. This paper proposes the solutions to the issue of knowledge consolidation in distributed data mining with less communication overhead. This is possible with mobile agents due to minimum information exchange by overlapped operations, thereby improving the efficiency of distributed data mining. Percentage of saving in processing time increases with the distance between the central site and distributed business sites. Still there is need to investigate the security and traceability aspects of mobile agents to make them a feasible approach for building distributed applications. There is also a need to investigate the applicability of mobile agents paradigm to autonomic system design and in building context aware services in ubiquitous computing applications. Authors are working on these proposed future works.

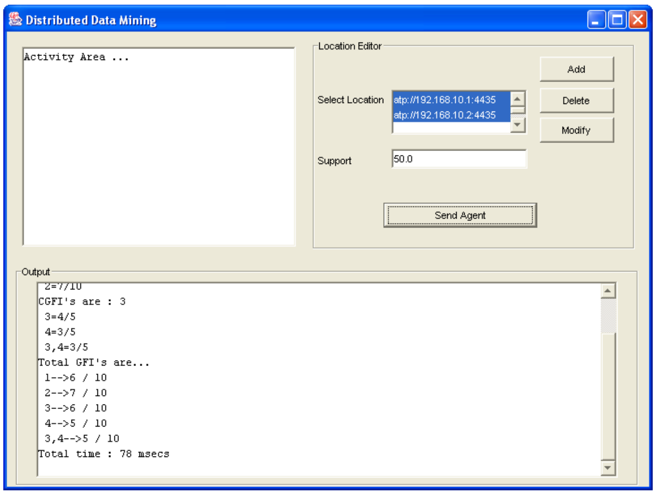


Fig-11

Acknowledgments

Authors thanks all the research associates of PG and UG courses of the Vishveshwaraiya Technological University, Belgaum for their contribution in implementing and experimentally verifying the design concepts proposed.

References

[1] David Kotz, Robert Gray, and Daniela Rus, "Future Directions for Mobile Agent Research", Dartmouth College.-IEEE Distributed System Online-August 2002.

[2] John Zachary, University of south Caralina- "Protecting Mobile Code in the Wild:" IEEE Internet Computing, March-April 2003- pp 78.

[3] R.R.Brooks, "Mobile Code Paradigms and Security Issues "Clemson University –IEEE Internet Computing, May-June 2004.

[4] A. Rubin and D. Geer, "Mobile code Security ", IEEE Internet Computing, NOV-Dec 1998, pp 30-34.

[5] M Wooldridge and N R. Jennings, "Software Engineering with Agents", IEEE Internet Computing, May-June 1999.

[6] Amund Tveit, "A survey of Agent-Oriented Software Engineering"- , Norwegian University of Science and Technology-May 8, 2001.

[7] Danny B Lange and Mitsuru Oshima, " Seven good reasons for mobile agents", Communication of the ACM, 42(3):88- 89, March 1999.

[8] T. Middelkoop A. Deshmukh, "Caution! Agent Based Systems in Operation"- University of Massachusetts at Amherst Department of Mechanical and Industrial Engineering.

[9] Chattratichat,J., Darlington, J , et al, " An Architecture for Distributed Enterprise Data Mining", in proceeding of the 7th Intl., Conf., on High Performance Computing and Networking(HPCN Europe 99, Amsterdam, The Netherlands, Springer Verlag LNCS, 1593, pp573-583

[10] Parthsarthy,S., and Subramanian, R., (1999), " Facilitating Data mining on a Network of Works Stations:", AAAI Press.

[11] Wu-Shan Jian, Ji-Huiyu, "Distributed Data Mining on The Grid", Proceeding of the Fourth International Conference on Machine Learning and Cybermetics, 18-21 August 2005, IEEE, pp 2010-2014.

[12] You-Lin Ruan, Gan Liu, Qing-Hua Li, "Parallel Algorithm for Mining Frequent Item sets", Proceeding of the Fourth International Conference on Machine Learning and Cybermetics, 18-21 August 2005, IEEE, pp 2118-2121

[13] U.P.Kulkarni, A.R.Yardi, et al, "Exploring the capabilities of Mobile Agents in Distributed Data Mining", Proceeding of the Tenth International Database Engineering & Applications Symposium- IDEA'06, IEEE-06.

[14] Ying-Wu Fang, et al , " Study on Algorithms of Parallel and Distributed Data Mining Calculating Process", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 18-21 August, IEEE-2005.

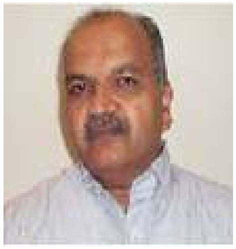
[15] William K. Cheung, et al, "Service-Oriented Distributed Data Mining", IEEE Internet Computing, July-August-2006.

[16] Wu-Shan Jiang, et al, " Distributed Data Mining On Grid", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 18-21 August, IEEE-2005.

[17] Yun-Lan Wang, et al , " Mobile Agent Based Distributed and Incremental Techniques for Association Rules", Proceedings of the Second International Conference on Machine Learning and Cybernetics, 2-5 November-2003, IEEE-2003.

[18] Matthew Eric Otey, Srinivasan Parthasarathy, " Parallel and Distributed Methods for Incremental Frequent Item set Mining", IEEE Transaction on Systems, Man and Cybernetics- Part B, Cybernetics, Vol-34, No-6, Dec 2004.

U.P.Kulkarni received the B.E. and M.E. degrees in Computer Science & Engineering from Karnataka University and Bharatiyar University in 1989 and 1991, respectively and PhD from Shivaji University in the year 2007. He is now with SDM College of Engineering & Technology as Professor and Chief Nodal Officer. He is the member of various technical bodies. His interest includes Autonomic Computing, Pervasive / Ubiquitous Applications and Software Agents.



A R Yardi is currently Professor in Electronics Department and Dy. Director of Walchand Engineering College, Sangli. He is currently guiding students for PhD in electronics & guided over 30 Postgraduate students. He has administrative experience at various levels like In charge Principal, Head of Department, and Chairman Board of studies at universities and he is member of several professional bodies

Padmashree D. Desai is currently working as a Asst. Prof. in the Department of Computer Science and Engineering, B.V.B College of Engineering and Tech, Hubli. She has a teaching experience of eighteen years. She is also involved in activities such as Effective teaching Curriculum Design and Development. She holds M.Tech in Computer Science and Engg. and involved in Research work. Her area of interest includes Data Mining, Analysis of Data, and Knowledge Management.

J.V.Vadavi is currently a Assistant Professor in the Department of Computer Science & Engineering Department of SDMCET. He is pursuing his PhD in the area of Autonomic Computing. His interest includes distributed applications, pervasive & ubiquitous systems and he is member of several professional bodies.