Jade Agent Framework for Distributed Data Mining and Pattern Analysis

Darshana Patel
Research Scholar
Gujarat Technological University
Chandkheda, Ahmedabad

J. S. Shah,PhD
Gujarat Technological University
Chandkheda
Ahmedabad, Gujarat.

ABSTRACT
This paper describes the role of agent in distributed data mining, comparison of various agent frameworks and also focusses on jade agent framework as a suitable framework for distributed data mining and discovery of patterns from multi agent systems. In this paper, overview of JADE architecture and basic steps to develop agents are described. The paper also focusses on implementing association rule mining algorithm- Apriori on JADE agent framework and result has been generated.

General Terms
Mobile Agents, Distributed database, Knowledge Discovery, Multi agent systems.

Keywords
Jade Agent Framework, Distributed Data Mining, Pattern Analysis

1. INTRODUCTION
In past few years, there is an outburst of information which is increasing by leaps and bounds and to analyze this information has been a crucial task. The information is distributed and is heterogeneous, thus the role of multi agent systems and knowledge discovery has become vital. Data Mining is very useful for discovery of patterns in multi agent systems for multi agent learning, adaptation, evolution and behavior analysis. The agents support and enhance knowledge discovery process which includes data selection, extraction, preprocessing and integration. Agents monitor the data changes at distributed sites and mine dynamic patterns and merge them. Multiple agents collaborate to mine distributed data. It is also not possible to store large amount of data at single place, its integration is also complex in parallel storage systems. The availability of data sources in mobile environment highly depends on time, thus agents are best suitable for autonomy, integration, dynamic selection and scalability. Agents also provide privacy, mobility and communication in distributed database.

2. DISTRIBUTED DATA MINING
Distributed Data Mining (DDM) is a branch of data mining that offers a framework to mine distributed data. Distributed data mining originated from the need of mining over decentralized data sources regardless of their physical locations. Distributed data mining is required in two cases, 1) The data is distributed on different locations and mining process is performed and 2) Data mining, sometimes needs lots of resources to perform mining process and to distribute the load, distributed mining process is required. The mining process is carried out at the local sites and final results are aggregated at the global site [1].

Association Rule Mining is one of the most important data mining techniques which find associativity from large set of data items. ARM specially focuses on distributed architecture, which will require integration of knowledge generated from multiple data sites. [2] Association Rule mining technique generates rules on the basis of minimum support count. Apriori algorithm is most popular algorithm of Distributed Data Mining. [1] Apriori algorithm performs repeated scanning of database to generate association rules. In this research paper results has been generated on the basis of Apriori algorithm.

3. AGENTS
3.1 Agents in Distributed Data Mining
The Multi Agent systems combined with distributed data mining can create adaptable systems. Distributed data mining enhances agents learning and knowledge processing and provide capability to avoid uncertainty due to historical event analysis. Data mining techniques like Association rule mining have no equivalent techniques in agent systems. Agents when equipped with Association rule mining techniques can have the capability of learning, discovery probing and searching patterns. There are two basic reasons why agent technology is suitable in Distributed data mining. 1) Distributed data mining has the characteristics suitable for mobile agent based approach which is modular with well-defined subtasks to encapsulate different mining algorithms 2) It can reduce the communication overhead of transferring large amount of data. This mobility aspect of agents make it most suitable for distributed data mining. The problems in implementing such systems are infrastructure and architecture of the systems [3].

3.2 Types of Frameworks
Agent architectures are the key mechanisms underlying the independent components that supports efficient performance in dynamic, open and real-world environments. There are various agent development platforms available for development of different kinds of applications. This research also describes the study of a variety of agent technologies along with their advantages and disadvantages. Different mobile agent frameworks are Aglets, Voyager, JADE, TACOMA, Grasshopper, SPRINGS, Tryllians Agent Development Kit, Zeus etc.

Aglets is developed by IBM which is combination of agents and applets which provides mobility to the applet. Aglets is agent technology developed with use of java. Aglet is abstract class of java. Aglet can work as mobile agent, it traverse from one location to another location and complete the task. Aglet implements event handler operations. Aglet use single thread mechanism so two agents cannot send synchronous messages to each other at the same time otherwise it create deadlocks.

Voyager is also java based agent framework developed by Object Space. With use of voyager technology, distributed application is possible with traditional messaging or RMI. Voyegers business software is not freely available and the

agent is active for particular time period only. TACOMA Tromoso and Cornell Moving Agents is developed by Tromoso and Cornell University. In TACOMA architecture data files and code files are used. Data files store status of the agent and the code file stores actual code of the agent.

Grasshopper is developed by IKV++. Grasshopper use MASIF (Mobile Agent System Interoperability Facilities) support for developing mobile agent application. This application is telecommunication based applications. Grasshopper provides GUI support to develop agent. The Grasshopper mobile agent framework is made of various regions. Developers can get dynamic proxy benefit due to regions.

Springs is developed by DISG- Distributed Information Systems Group at the University of Zaragoza in Spain. It proposed a hierarchical infrastructure of regions but it does not support FIPA standard and does not provide graphical tools.

JADE-Java Agent Development Environment, is developed by Research & Development department of telecom Italia and latter distributed as open source under LGPL license. It is java based, follow FIPA specifications. Jade support GUI. It provides run time environment for agent development and can develop multi agent system. It has inbuilt agent mobility service and agent management system [4].

3.3 Comparison of different frameworks

The comparative analysis of different agent framework has been described in Table 1.

3.4 Jade Framework

JADE is middleware that simplify the development of applications. It is already used in different sectors like supply chain management, rescue management, tourism etc. JADE is compliant with FIPA specifications. It provides a same set of APIs that are independent from underlying network and java versions [5]. It seems to be better as compared to other frameworks. Jade is open source platform, designed purely in java and supports GUI. It provides run time environment for agent development and can develop multi agent system. It has inbuilt agent mobility service, agent management system, security features and sound agent mobility. Jade has been already used in many successful projects in various sectors. From above different features of jade it has been identified as most suitable agent architecture.

4. ARCHITECTURE OF JADE

A JADE platform is composed of agent containers that can be distributed over the network. Agents live in containers which are the Java process, that provides the JADE run-time and all the services needed for hosting and executing agents. There is a special container, called the main container, which represents the bootstrap point of a platform: it is the first container to be launched and all other containers must join to a main container by registering with it." [6].

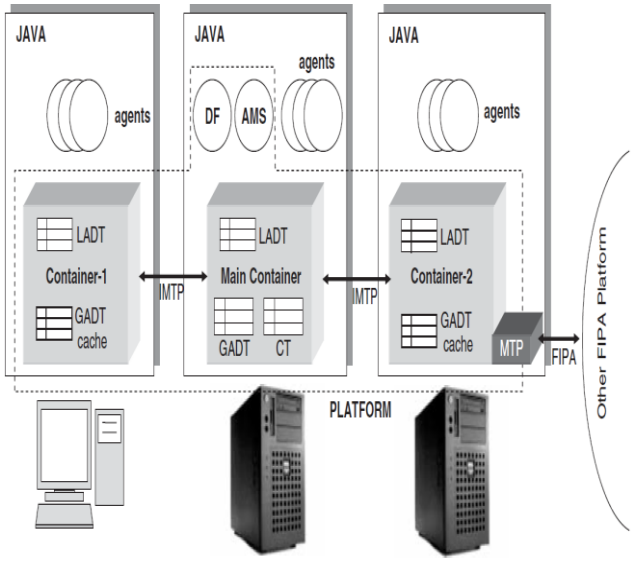

Fig 1: Main Architectural Elements of JADE [6]

The Main Container has several responsibilities like:

The Main container manages the container tables which are the registry of the objects and convey address of all container nodes of the platform. It also manages the registry of all agents known as Descriptor Table for Global Agent (GADT), including the location and status of the agents. The Main Container is the host for two agents like AMS (Agent management system) and Directory Facilitator (DF). It hosts yellow page service of the platform. Two special agents are automatically instantiated and started by JADE when the main-container is launched, whose roles are defined by the Agent Management standard of FIPA." [6].

The Agent Management System (AMS): The task of AMS is supervision of entire platform. It is a point from where all agents can find the details to manage their life cycle. Every agent is automatically registered, at start up, with AMS by JADE itself to get valid AID.

The Directory Facilitator (DF) Agent: DF provides service of yellow pages. If any agent wants to register their service of search for the service they can used DF agent. The DF provides subscription service too, DF can notify to agent whenever a service registration or modification is made that matches some specified criteria. To provide distribute yellow page service multiple DF can be started concurrently across several domain. Here JADE architecture provides following facilities: Platform features are implemented as separated modules. Flexible support for integration of new features and modification of existing features. Easy establishment and deployment of features across a distributed platform. 'Deploy what you need' strategy to only start required features and use target-specific implementations (e.g. for mobile environments)"[6].

Table 1. Comparative Analysis of Mobile Agent Frameworks

Features	JADE	Voyager	Aglet	Springs	Tacoma	Grasshopper
Security	Strong Traditional	Secured channel	Partial	Partial	Firewall agent	Partial
Model	Procedural	Procedural	Event	Procedural	Behavioral	Procedural
Programming Language	JAVA	JAVA	JAVA	JAVA	JAVA	-
Proxies	No	Yes	Yes	Yes	No	Yes
Sync Communications	No	Yes	Yes	Yes	No	Yes
Asynchronous Communications	Yes	Yes	Yes	Yes	Yes	Yes
Messages	(FIPA)	No	Yes	Yes	Yes	Yes
Remote Calls	No	Yes	No	Yes	Yes	-
Available Download	Yes (LGPL)	Not Free	IBM PL	Yes	-	Not Free
GUI Tools	Yes	No	Some	No	Yes	Yes
Level of Activity	Very High	High	Very Low	High	Very High	None

5. IMPLEMENTING DISTRIBUTED DATAMINING ALGORITHM ON JADE

Different agents can be simply created in JADE framework. In JADE Mining algorithm can be implemented as agent.

5.1 Basic steps to develop an agent

To create jade agent is simple task in JADE. Jade.core.Agent class is extended and setup () method is implemented to create JADE agent. The task of setup method is agent initialization. In creation of jade agent instead of initialization in constructor, setup method is advisable because, at the time of construction, agent is not yet linked to the JADE runtime environment and so some methods inherited from agent class may not work properly. Every agent instance has unique id, which is used to identify individual agent. The id is referred as agent identifier and it is instance of jade.core.AID class. The getAID() method returns local agent identifier. The AID object include GUID (Globally unique name) and number of addresses. There are many methods available in AID that provides details about agent address and name. Agent terminated through doDelete() method and to perform cleanup operation, takedown() method is invoked. To pass arguments to agent getArguments () method of Agent class is used. [7]

```
public class Ddm extends Agent
{
        public void setup () {
        Object[] args = getArguments();
        String arg1 = args[0].toString();

        // Implementation code of the algorithm

        }
}
```

JADE agent performs multiple task concurrently. The task of agent is cooperative and behavior is scheduled for execution with use of action () method. There are three types of behavior of agent, one shot behavior, cyclic behavior and generic behavior. In one shot execution, the action method is invoked only once and the inbuilt class is also available for that. In cyclic behavior, the action method is invoked to perform the same tasks every time and it is never completed. In generic behavior, based on status different behavior is performed.

5.2 Implementation and Results of Mining algorithm

Apriori, Association rule mining algorithm has been implemented on JADE agent and result has been generated.
The Jade implementation includes creation of java class files for the algorithms, booting up the Jade architecture. After the Jade boots up, new agents can be created and upon execution of the agents, the results of the algorithms are generated. The agents can be created using two ways [5].

5.2.1 Starting agents using command line

In jade framework to start an agent through command line arguments following command is used. In following command arguments are also included.

Java jade. Boot –gui –agents dap: examples.Apriori.apriori_opt (flu1.csv, 100)

5.2.2 Starting agents using Agent Management Console

JADE framework provides console support to create an agent also. The output of the algorithm can be viewed on the console.

5.2.3 Creation of Agent using Apriori Algorithm

The new agent can be added to the platform using Agent Management console. The class file of the algorithm is added using the package and can be selected during creation of agent. After the main container is initialized, other containers can also be launched on different host within the platform.

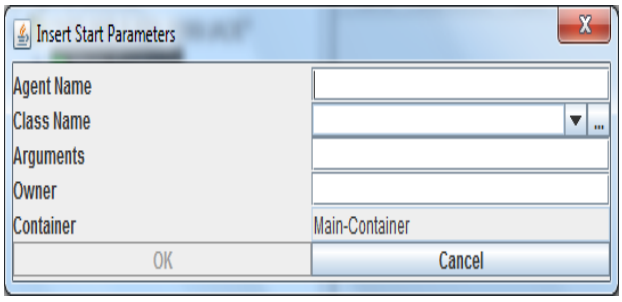

Fig 2: Creating Jade Agent (Insert Start Parameters)

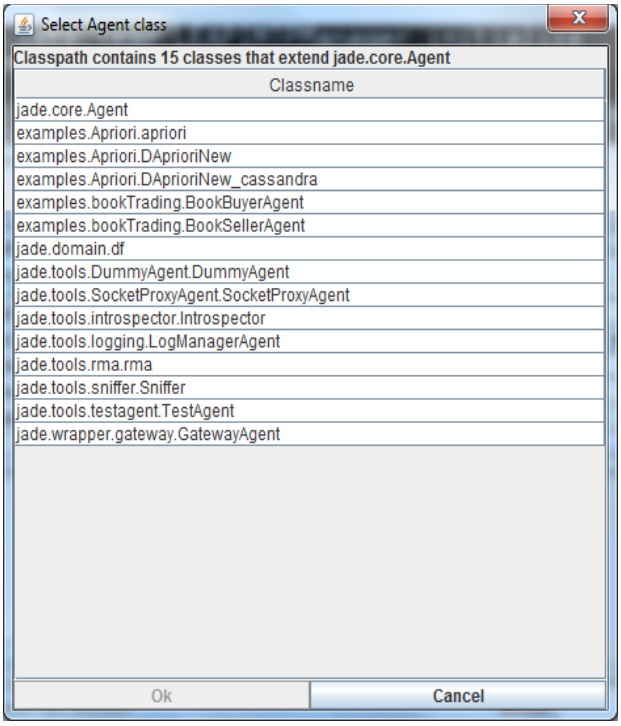

Fig 3: Creating Jade Agent (Select Agent Class)

Selecting the Agent Class: The files in above figure are created using java and the compiled class files of the algorithms are added to the examples package. This class files are used to create the java agents using Jade.

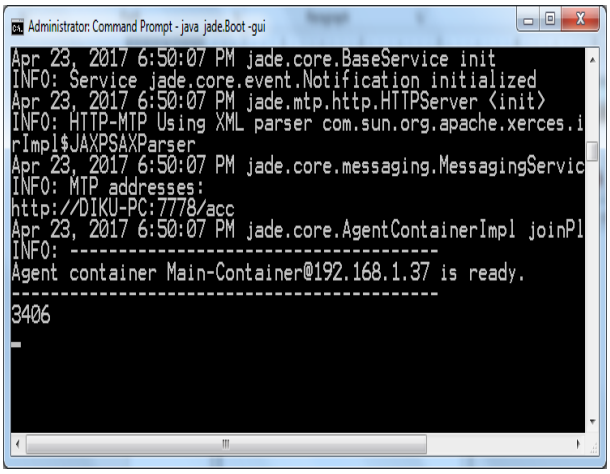

Fig 4: Output of Apriori Algorithm

The output of the Apriori algorithm is generated as a text file containing the pair of item sets and the corresponding count which is greater than the minimum support count.

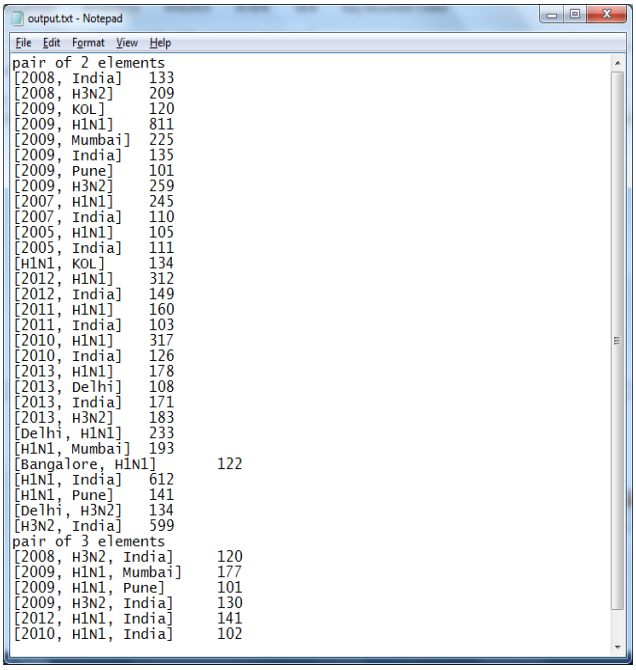

Fig 5: Output of Apriori Algorithm (Text File)

6. CONCLUSION

Agent framework plays key role for agent development. Myriad agent frameworks are available, this paper provides overview of different frameworks and comparative analysis of it. From different frameworks, JADE has been identified as suitable framework due to support for FIPA specifications, GUI and java environment. This paper describes architectural overview of JADE and the simple steps to develop an Apriori data mining agent in JADE framework. Agent paradigm is identified as effective area in distributed environment but the major limitation of agent is security. In future work, it should be focused and after resolving problem of security, the agents

can be used more extensively, thereby many effective results can be generated in distributed data mining.

7. ACKNOWLEDGMENT

We would like to express our very great appreciation to significant advisors and contributors for their valuable and constructive suggestions during the planning and development of this research work. Their willingness to give time so generously has been very much appreciated. The authors would like to thank them for their helpful comments to improve this paper.

8. REFERENCES

[1] Shichao Zhang, Xindong Wu, Chengqi Zhang, "Multi-Database Mining",IEEE Computational Intelligence Bulletin, June 2003, Vol 2, No.1

[2] O. Folorunso, A. S.Sodiya, G. O. Ogunleye and A. O. Ogunde, "*A Review of Some Issues and Challenges in Current Agent Based Distributed Association Rule Mining*", Medwell Journals, vol. 10, no. 2,pp.84-95,2011

[3] Longbing Cao, University of Technology, Sydney Vladimir Gorodetsky, St. Petersburg Institute for Informatics and Automation, Russian Academy of Sciences Pericles A. Mitkas, Aristotle University of Thessaloniki , Agent Mining: The Synergy

[4] Mats Persson, "Mobile Agent Architectures", Scientific Report, Division of Command and Control Warfare Technology, ISSN 1104-9154, January 16, 2001 [Online]. Available: https://www.lysator.liu.se/~matpe/publish/agark.pdf. [Accessed: 14- Jun- 2016].

[5] "Jade Site | Java Agent DEvelopment Framework", Jade.tilab.com, 2017. [Online]. Available: http://jade.tilab.com. [Accessed: 23- Nov- 2016].Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

[6] Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood, Developing multi agent systems with JADE, ISBN: 978-0-470-05747-6 February 2007Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[7] Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood, Developing multi agent systems with JADE, ISBN: 978-0-470-05747-6 February 2007