

Using Migrating Agents in Designing Web Search Engines and Property Analysis of Available Platforms

Niraj Singhal¹, Ashutosh Dixit², R. P. Agarwal³, A. K. Sharma⁴

^{1,3}Faculty of Electronics, Informatics and Computer Engineering
Shobhit University, Meerut, India

^{2,4}Department of Computer Engineering
YMCA University of Science and Technology, Faridabad, India

Email : sonia_niraj@yahoo.com

Abstract

Mobile agent is a type of software agent, a composition of computer software and data. It is able to migrate from one computer to another autonomously and continue its execution on the destination computer. When very large volumes of data are stored at remote hosts, these data can be processed in the locality of the data rather than transferring over the network. The main concern is, move the computations to the data rather than data to the computations. Distributed crawling based on migrating crawling agents the process of selection and filtration of web documents can be done at web servers rather than search engine side which reduces network load caused by the web crawlers. In this paper we present advantages of mobile agents and their use in various areas. We also present use of mobile agent technology in designing web search engines, various available mobile agent technologies and an analysis of their various properties.

Keywords- Web search engine, migrating agents, agent platforms, property analysis

1. Introduction

In the distributed crawling based on migrating crawling agents or migrants [15,19,20,21], the process of selection and filtration of web documents can be done at web servers rather than search engine side which can reduce network load caused by the web crawlers [20,21]. Mobile agent [13] is a type of software agent with the feature of autonomy, social ability, learning, and most importantly, mobility. It is a composition of computer software and data, which is able to migrate (move) from one computer to another autonomously and continue its execution on the destination computer. In migrating crawling agents, the mobile code generated by search engine side transfers and executes on web servers, an environment controlled by another party.

Because of mobility of mobile agent, the security problems [10] have also become a bottleneck for development and maintenance of mobile agent technology. In this paper we present advantages and use of mobile agents in various areas. We also present use of mobile agent technology based crawling agents use in web information retrieval and also discuss various available mobile agent technologies and their comparison on various parameters.

2. Related Work

Mobility allows an agent to move, or hop, among agent platforms. Migrating agents are computational software processes capable of roaming wide area networks such as the web, interacting with foreign

hosts, gathering information on behalf of its owner and coming back having performed the duties set by its user. An agent is an autonomous entity that acts on behalf of others in an autonomous fashion. Nwana [13] identifies seven type of agents i.e. collaborative agents, interface agents, migrating agents, information agents, reactive agents, hybrid agents and smart agents. Key characteristics of mobile agents are [13,15] migration, data acquisition, route determination and communication.

The agent platform provides the computational environment in which an agent operates. The platform from which an agent originates is referred to as the home platform, and normally is the most trusted environment for an agent. One or more hosts may comprise an agent platform, and an agent platform may support multiple computational environments, or meeting places, where agents can interact. They may cooperate or communicate with other agents making the location of some of its internal objects and methods known to other agents without necessarily giving all its information away (figure 1).

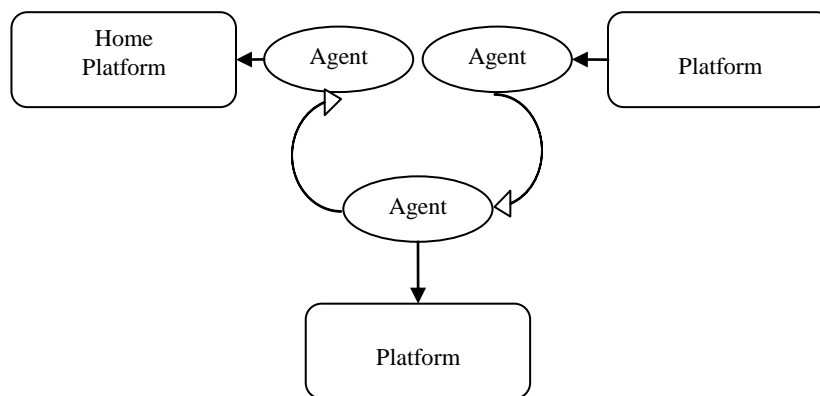


Figure 1. Working of migrating agents

The mobile agents interrogate their local environment to acquire the information, necessary to achieve their goals. This information needs to be filtered locally by the agent before it is either stored with the agent or forwarded to some receiving destination. Once an agent has finished with a network node, it must make a decision of where to move next. The ability of agents to communicate is fundamental to mobile systems. Advantages of mobile agents are [13,15,21] efficiency, bandwidth, latency, asynchronous task execution, robustness, fault tolerance, support form heterogeneous environments, support for electronic commerce, easier development paradigm and peer-to-peer communication.

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 main concern is, move the computations to the data rather than the data to the computations. By migrating to the location of the resource, a mobile agent can interact with the resource much faster than from across the network. While the agent acts on behalf of the client on a remote site, the client may perform other tasks. Instead of being online for a longer period, a mobile user may develop an agent request while being disconnected, launch the agent during a brief connection session, and receive back the agent with the result at some later time.

2.1 Applications of Mobile Agent Technology

Several fields in which mobile agent technology is of immense use are [3,4,7,8,11,12,17,19,20,31], web information retrieval, data processing, mobile computing system, network management, electronic

commerce, energy efficiency and metering, wireless multimedia sensors, grid computing and grid services, distributed data mining, multimedia, human tracking, security, intrusion detection, affective computing, climate environment and weather, e-learning, location, recommendation and semantic web services.

2.1.1 Data Processing

Intensive data processing is a graphic application that has to process images to search for a particular one. Let us suppose that all images are stored in a centralised database, which different clients can access to. In a mobile agent approach a processing agent is sent from the user site to the image server, where it carries out processing. This approach needs only two connections, one to send the processing agent, and one to take it back the results. If a large number of images are to be analysed, this approach permits to save bandwidth in a significant way [12].

2.1.2 Distributed Data Mining

Distributed data mining is a complex system which is related on the distribution of resources over the network as well as data mining processes. Mobile agent plays an important role in distributed data mining structure in e-commerce environment based on Web services. In the context of data in heterogeneous environment, mobile agents can resolve problems of integration and communication [3,7,8,17,31].

2.1.3 Electronic Commerce

The commercial activity is a significant part of the network infrastructure allowing an open market of the services. Mobile agent can be implemented in e-commerce to search and to filter information of interest from electronic markets. The mobile agent survives even if failure rate is more than 80% however for higher failure rate performance degraded significantly. The advantage of adopting mobile agents for M-commerce is to scale up to large, dynamic world market places distributed over the Internet and to ease the access and participation of mobile users [3].

2.1.4 Energy Efficiency and Metering

Mobile agent technology plays an important role in energy metering and control. Mobile agent technology can be used for controlling power of distributed generations in an isolated micro grid, for adapting knowledge retrieving in the context of mobile grid, and developing a multi-agent system to enable the function beyond the capabilities of any singular mobile agent in the system which perform particular task [3].

2.1.5 Grid Computing and Grid Services

Mobile agent technology has also been used to develop agent based Grid services. The services are delivered using a standard interface which is compliant with the common services. The agent can migrate from the Grid node to any nodes outside the Grid by the http default transmission protocol. A Geographic Information Grid System (GIGS) based on mobile agent can be designed that provide services and improve sharing distributed resources [3].

2.1.6 Human Tracking

Mobile agent paradigm can also be used to enhance video monitoring system in the automatic human tracking system. The mobile agent utilizes the algorithm of determination of neighbour video cameras to pursue the human efficiently. Since e-learning systems don't consider the emotional intelligence in the context of instruction, an emotional intelligent e-learning system can be designed based on mobile agents [4].

2.1.7 Mobile Computing System

The wide spread of notebook computers and personal digital assistants (PDA) points out the need of new paradigms for their applications. Mobile agents can be used to model applications that reside on a PDA and use resources spread over the network [12].

2.1.8 Multimedia

The mobile agents roam in the network to search relevant images and mark the relevant path in order to guide to the interesting sites. The agents can be used to find the sites containing the relevant images. Mobile Agent can also be used to create location dependent multimedia services for mobile devices. The action performed by the mobile agent would be configuration and reconfiguration, communication, downloading multimedia to mobile device and Quality of service handling. [3].

2.1.9 Network Management

When network administrators have more than one node to manage, use of mobile agents reduces the network bandwidth occupation. One of the most significant use of mobile agents applications is the management of commercial telecommunication networks. Currently if an optical cable in the WAN is accidentally cut, the time required to locate the problem may extend ridiculous because of the slow response of the network itself. The mobile agents can also be used to collect information of all mobile nodes in order to reduce the network delay and the overhead of control messages for routing [3].

2.1.10 Web Information Retrieval

The Internet and the web have given access to a large amount of information, but due to the wide spread of such information the search for a particular topic can be very hard, implying a waste of time and bandwidth. Search engines facilitate the retrieving of web pages containing given words, but they present several problems, particularly in maintaining a large amount of data and in specifying query. An agent-based solution provides one or more agents that visit web servers searching for interesting pages. It saves bandwidth because the only needed communications are those to send and receive the agent [4,12,19,20].

2.1.11 Wireless Multimedia Sensors

Mobile agent improves the efficiency of transmitting agent by using the publish/subscribe mechanism to build interest route. Mobile agent can improve the reliability of the entire application. The mobile agents can also be used in image processing applications over wireless sensor networks, where multiple hops may exist between target source nodes, and the sensory data packets may not be aggregated efficiently [3].

3. Web Search Engines Based On Mobile Agents

A search engine (figure 2) is a coordinated set of programs that is able to read every searchable page on the web, create an index of the information it finds, compare that information to a user's search request (i.e. query), and finally return the results back to the user. It is a searchable database which collects information from web pages on the Internet, indexes the information and then stores the result in a huge database where from it can be searched quickly. Search engines operate as a link between web users and web documents. Without search engines, this vast source of information in web pages remain veiled for us. A general web search engine has three parts i.e. Crawler, Indexer and Query engine; as shown in figure 2.

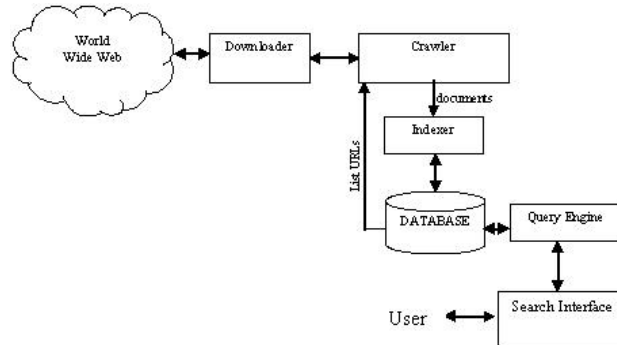


Figure 2. General architecture of a search engine

The web crawler is a module that searches the web pages from the web world. These are small programs that peruse the web on the search engine's behalf, and follow links to reach different pages. Starting with a set of seed URLs, crawlers extract URLs appearing in the retrieved pages, and store pages in a repository database. The indexer extracts all the uncommon words from each page and records the URL where each word has occurred. The result is stored in a large table containing URLs; pointing to pages in the repository where a given word occurs. The query engine is responsible for receiving and filling search requests from users. It relies on the indexes and on the repository. Because of the web's size and the fact that users typically only enter one or two keywords, result sets are usually very large.

In the distributed crawling with migrating agents approach [4,15,19,20], agents allow packaging a conversation and dispatching it to a destination host where the interactions can take place locally. Migrating agents are also useful when it comes to reducing the flow of raw data in the network. 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 main concern is, move the computations to the data rather than the data to the computations. By migrating to the location of the resource, a mobile agent can interact with the resource much faster than from across the network that reduces network traffic also.

The distributed crawling with migrating agents approach (figure 3) uses a crawler manager at the search engine site, that deposes migrating crawlers to the web servers with a list of URLs of respective web servers. The migrating crawler, on reaching a server crawls the pages, select the best of the pages for its collection and comes back to the search engine with the collection. It reduces unnecessary overhead of being the unnecessary pages to the search engine site. The size of the collection can further be reduced by filtering the required specialized web pages and even compressing them.

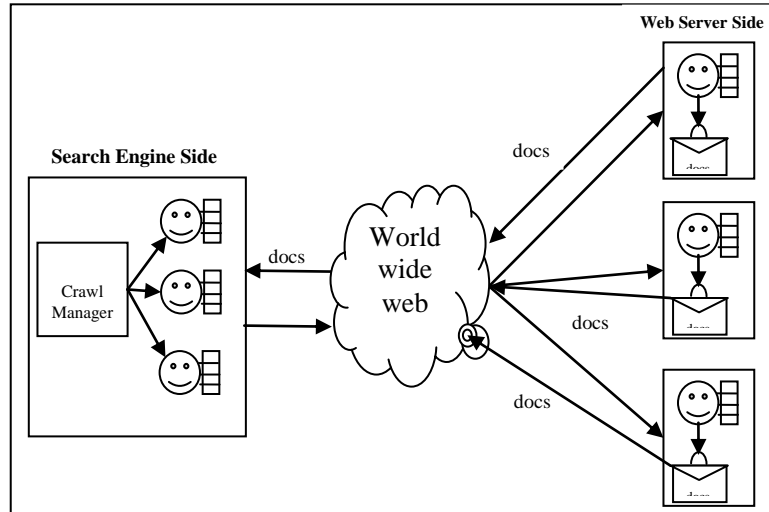


Figure 3. Search Engine based on migrating agents

4. Mobile Agent Systems

Various mobile agent systems available are Agent Building and Learning Environment, Agent TCL system, Aglets, Anchor Toolkit system, ARA, Concordia, Grasshopper, HTTP-Based Infrastructure for mobile agents, JADE, MAF, Mole, PIAX, SensorWare, SMARD, SPRINGS, TACOMA, TAgent, Tracy, Trinity, Tryllian and Voyager. These systems differ in their goals, motivations, and implementations but they all by and large provide common functionalities that support migration of agents, programming languages, communication between agents and various forms of security. Various studies [2,10,12,18,23,25,30] have been done representing development, use, advantages and disadvantages of mobile agents. Now, let us have a look on all these mobile agent development environments.

4.1 Agent Building and Learning Environment

The Agent Building and Learning Environment (ABLE) [43] project started in early 1999 at the IBM T.J. Watson research laboratory. The aim was to produce a fast, re-usable and scalable toolkit for creating intelligent software applications. ABLE is a Java-based framework, component library, and productivity toolkit for building intelligent agents that can use machine learning and reasoning. It is designed to be used by applications involved in autonomic computing, data mining, forecasting, planning, retail and resource balancing etc. It provides a distributed platform allowing agents to be configured, run and managed across different physical systems. Its framework provides a set of Java interfaces and base classes used to build a library of JavaBeans called AbleBeans. The library includes AbleBeans for data transformation and scaling, for rule-based inferencing and for machine learning techniques.

4.2 Agent TCL system

The Agent TCL system [23,26,30,46] is a model for supporting mobile agents developed at Dartmouth College. Agent TCL appears to be the most flexible architecture. It supports state oriented migration, multiple languages and networking protocol. It also supports security mechanisms but it lacks in constraining agent execution because of which resources are directly accessible via TCL and freely available for use by agents. It is inefficient than other interpreted languages. It lacks object-oriented features, thus making it difficult to write and debug large scripts.

TCL is inefficient as compared to other interpreted languages and is ten thousand times slower than optimized C. Its non-object-oriented nature provided no code modularization aside from procedures, thus making it difficult to write and debug large scripts. It lacked facilities for capturing the internal state of an executing script, which are essential for providing transparent migration at arbitrary points and lack of constraining agent execution. Resources, directly accessible via TCL are freely available for use by agents. Even with these drawbacks, Agent TCL appears to be a decent system.

4.3 Aglets

Aglets [23,25,32] developed at the IBM Tokio Research Laboratory is completely made in Java, granting a high portability of the agents and the platform both. An aglet is a Java agent, able to autonomously and spontaneously move from one host to another. Aglets includes both, a complete Java mobile agent platform with a stand-alone server, and a library that allows developer to build mobile agents and to embed the Aglets technology in their applications. Aglets follows an applet like developing paradigm and user only needs to define a few methods to implement the behaviour of agents. Aglets is secure because it is developed in Java and is compliant to the Java2Security Manager. Aglets is flexible, since it allows user to extend the platform in order to implement new functionalities.

Its disadvantage is that the proxies it provides are not dynamic proxies i.e. they cannot be used after the agent they point to moves to another place, therefore the programmer must obtain himself an updated proxy, if needed, before using it. As every agent is assigned a single thread, the programmer must avoid the execution of long-running tasks: otherwise, that would prevent agent events such as incoming messages from being considered. The platform does not support remote calls to agents or assigning them user-friendly identifiers.

4.4 Anchor Toolkit system

The Anchor toolkit [2,29] handles the transmission and secure management of mobile agents in a heterogeneous distributed computing environment. The toolkit protects the agents being dispatched between hosts through encrypted channels. A mobile agent's host platform is required to sign the agent's persistent state before dispatching the agent to the next platform. The signed persistent state can be used later to detect potential problems with the agent's state. The architecture of the Anchor toolkit consists of an Agent Viewer, Agent APIs, Anchor Server, Anchor Security Manager, Anchor Class Loader, Secure Agent Transfer Protocol handler, Anchor Java Naming and Directory Interface and Anchor Java Native Interface.

4.5 ARA

Ara [4,23,30] is a platform for the portable and secured execution of mobile agents in heterogeneous networks. The Ara platform provides facilities for access to system resources and agent communication under the characteristic security and portability requirements for mobile agents in heterogeneous networks. In Ara, mobile agents are programmed in some interpreted language and executed within an interpreter for this language, using a special run-time system for agents, called the Core; the central part of an Ara system. The core mediates any access from an application agent to the host system or to another agent for security and portability. Ara is concerned with designing of secure and portable execution of mobile agents.

Ara's security model is flexible as domains of protected resources can be dynamically created in the form of places, and that the admission of agents to such a domain, as well as their actual rights at that place, can be controlled in a fine grained manner down to individual agents and resources.

4.6 Concordia

Concordia [9,16,23,30] is a full-featured framework developed at Mitsubishi Electric Information Technology Center America. Concordia consists of a set of components that provides services such as communication, administration, persistent storage, security etc. In Concordia agent communication is held either through asynchronous distributed events or collaboration. Asynchronous distributed events are events that agents receive via the Event Manager component. The agent determines the type of events an agent receives when it first registers with the Event Manager. The Event Manager can forward events to an agent even after it migrates to another system.

Its security model provides support for two types of protection i.e. protection of agents from being tampered with, and protection of server resources from unauthorized access. Concordia uses encryption to protect agents during transfer and, to protect resources on each server, Concordia relies on its Security Manager component to manage resource protection. The Security Manager authenticates each agent by verifying its identity. If the identity matches, then the agent is able to access the resource.

4.7 Grasshopper

Grasshopper [6,25] is a mobile agent platform that is an integration of the traditional client/server paradigm and mobile agent technology. It is developed compliant to the first mobile agent standard of the Object Management Group (OMG) [41]. A Grasshopper system can be composed of different regions. It provides agent developers with interesting features, including a graphical user interface to manage agents, agencies, and regions. By defining regions, the developer can benefit from dynamic proxies.

Various services provided by a Grasshopper core agency are Communication Service, Management Service, Persistence Service, Registration Service, Security Service and Transport service. Communication service is responsible for all remote interactions that take place between the distributed components of Grasshopper. The Grasshopper persistence service enables the storage of agents and places on a persistent medium. Grasshopper supports two security mechanisms i.e. external and internal security, where External security protects remote interactions between the distributed Grasshopper components and Internal security protects agency resources from unauthorised access by agents.

4.8 HTTP based Infrastructure for Mobile Agents

The HTTP based infrastructure for mobile agents [1,10] is a project being developed at the Goethe University in Germany. It is designed to provide a low-level infrastructure to support agent mobility and communication through the use of HTTP. Its tasks include accepting agents, creating an appropriate runtime environment for agents to execute within, supervising the execution of agents and terminating agents if required.

In addition to this, the agent server must also organise the transfer of mobile agents to other hosts, manage communication between agents and their users, and perform authentication and access validation. Agents are transferred as encapsulated Multipurpose Internet Mail Extension (MIME) documents in a stateless manner, by posting them to a special URL that is managed by an agent server. Upon receiving this agent, the agent server parses the code of the mobile agent and determines whether it is acceptable.

4.9 JADE

Java Agent DEvelopment Framework (JADE) [2,3,5,23,25,33] is a java software framework that allows implementation of multi-agent systems. It is a middleware that aims at supporting the development of applications that address this evolution as its fundamental is the peer-to-peer intelligent agent approach. It allows the development of systems of peers able to work in a proactive way, according to usage rules given by owners, to communicate and negotiate with others, directly and regardless to their role and position and to coordinate in order to solve complex problems in a distributed way.

An agent is composed of different concurrent and nonpreemptive behaviors which can be added dynamically. The main disadvantage is that mobility is not a key element in JADE. So, it focuses on other functionalities relevant to the development of multiagent systems. The JADE built-in Agent Mobility Service supports mobility among containers within the same JADE platform.

4.10 MAF

Mobile Agent Framework (MAF) [3,23,34] is a Python research prototype that provides a set of primitives to facilitate the development of distributed mobile agent. It is a research prototype which carries the several goals in mind, it aims to provide a set of primitives to facilitate the development of distributed mobile agent; second, it strives to meet the application requirement from distributed sensor field which is to provide a light-weight, self-organized and secure agent platform, although this framework itself is primarily developed using Python language, it should provide a mechanism to be able to incorporate and integrate effortlessly with a variety of "foreign agents" written in other languages such as C and C++.

4.11 Mole

Mole [23,30] is the first mobile agent system that has been developed in the Java language and provides a stable environment for the development and usage of mobile agents in the area of distributed applications. There is strong migration and weak migration. Mole supports asynchronous communication using an even driven model. In the event driven model, depending on these events, internal rules, state information and timeout intervals, output events are generated, that in turn may be the input for other synchronization objects. Mole allows synchronous and asynchronous messages among agents along with RPC type communication.

It uses a Sandbox security model in which service agents have access to system resource, providing controlled, secure abstractions of these resources inside the agent system. Moreover, service agents may offer access to legacy software, using the native code interface offered by Java. This does not cause any security problems, because the service agents are immobile and may be started only by the administrator of the location. User agents may only communicate with other agents and have no direct access to system resources.

4.12 PIAX

P2P Interactive Agent eXtensions (PIAX) [3,23,42] is an open source framework that integrates mobile agents paradigm and P2P structured overlay network. It allows to build a scalable and efficient federated system in a large-scale distributed environment where various kinds of data and processes are located in each device. Overlay network enables pervasive devices to communicate with each other efficiently, while agent platform on the overlay network encourages the devices to cooperate with other devices.

PIAX is a Java class library that integrates mobile agent platform and P2P structured overlay network. The features of PIAX are powerful P2P agent mechanism with simple APIs, built-in versatile P2P overlay

networks, multi-overlay mechanism for plug-in independent P2P overlay networks, unified APIs for discovering peers and messaging between agents.

4.13 SensorWare

SensorWare [3,23,35] is an implementation of mobile agent environment for wireless sensor network. Its scripting language is based on Tcl, and scripts can move their code and data from node to node, autonomously. The distributed algorithms are realized as control scripts that are autonomously replicated or migrated in the proper sensor.

SensorWare abstracts the run-time environment of a sensor node using a set of services, and a scripting language to form scripts out of these services. These scripts perform certain tasks when executing in a node. Scripts can also move their code and data from node to node, autonomously. So, distributed algorithms are realized as control scripts that are autonomously populated in the proper sensor nodes after a triggering user injection

4.14 SMARD

Secure Mobile Agent Rapid Development (SMARD) [3,23,37] is a development environment oriented programming of mobile agents based applications. Using a graphical interface, programmers can design, build and launch mobile agents which are intended to run on JADE platforms.

4.15 SPRINGS

SPRINGS [23,25,27,36] developed by the Distributed Information Systems Group at the University of Zaragoza in Spain, focuses on scalability and reliability in scenarios with a moderate and high number of mobile agents. Its development has been inspired by the features of other popular platforms, such as Voyager and Grasshopper. It proposes a hierarchical architecture based on regions.

It provides full location transparency for movements so that the programmer does not need to specify network addresses but just the name of the destination, and for calls, through the use of dynamic proxies. The main disadvantage of SPRINGS is perhaps that it does not support agent communication and does not provide sophisticated security mechanisms. Despite it is easy to use, it does not offer any graphical tool to the user.

4.16 TACOMA

Tromoso And Cornell Moving Agents (TACOMA) [3,21,23,30,44] is being developed by the University of Tromoso and Cornell. TACOMA considers agents, either stationary or mobile, to be computational unit of the system. An *agent* in TACOMA is a piece of code that can be installed and executed on a remote computer. Such an *agent* may explicitly migrate to other hosts in the network during execution. An agent needs to store code and data for future computations and must be able to carry this information around when it migrates, and later retrieves it.

A TACOMA agent executing on one host moves to another host by using TCP to communicate with TACOMA software at the destination host. The current versions of Tacoma provide support for agents written in C, C++, ML, Perl, Python, Scheme, and Visual Basic. The TACOMA platform has also been ported to new operating system architectures, in particular Windows NT, Windows CE and the PalmOS.

4.17 TAgent

Travel Agent (TAgent) [3,45] is a Java based platform which allows to develop mobile agent. These mobile agents can act on behalf of their owner without the requirement for the user to interact. It provides an easy service extending an easy agent development platform and a secure design of the agents.

TAgents project aims to build an open source mobile agent system. It's main focus is on extendibility and flexibility. The TAgents platform was designed to be easy to extend. The basic platform can be extended with all kinds of service from database access up to the free choice of the communication protocol. Easy of an agent or system agent. The basic agent class comes with a rich feature set and any other service can easily be accessed through the service providing mechanism.

4.18 Tracy

Tracy [23,25,40] was developed at the University of Jena in Germany and has a plugin-oriented architecture. Plugins are the software components that can be added dynamically to a running agency, if required, in order to provide high-level services such as inter-agent communication, migration, security etc. Tracy agencies are lightweight and extensible. The platform also offers several migration strategies for agents. A key disadvantage of Tracy is that it does not support remote communications between agents, an agent must travel to the agency where another agent is running in order to communicate with it.

4.19 Trinity mobile agent framework

Trinity [3,38] is mobile agent based framework that allows to create for wide range of mobile agent types. It allows rapid development of non-complex agents. It is mobile agent framework for creation of wide range of mobile agent types. The goal is to provide a tool for rapid development of non-complex agents. Trinity mobile agent framework is an open source software that perform Intelligent Agents tasks. It's free for both personal and commercial use, thus the perfect choice for those that want an alternative for intelligent agents programs.

4.20 Tryllian

Tryllian [23,25,47] developed by the homonym company in 2001 is based on a sensing-reasoning action mechanism. It allows programmers to define a reactive and proactive behavior of agents. It proposes a taskbased programming model and communication among agents is achieved through message passing and in accordance with the FIPA [39] standard. It also provides a persistency service. The main disadvantage of Tryllian is that it does not offer location transparency.

In addition, its task based and asynchronous model could be difficult to use, due to its differences with the classical procedural programming. The use of a single thread per agent could be inefficient and a limitation for the programmer. Tryllian provides a large set of configuration options, which could be overwhelming. Finally, it does not offer facilities for synchronous communication or conventional method invocation.

4.21 Voyager

Voyager [2,3,23,25,30] is a java agent-enhanced Object Request Broker (ORB) created by ObjectSpace Company. Its goals are to provide programmer to create state of the art distributed programs quickly and easily while providing a lot of flexibility and extensibility for the products that are being created with the voyager system. Voyager supports RMI, DCOM, and CORBA architecture to provide stationary client server applications, which makes this system very flexible. Voyager uses regular java syntax to create remote objects and move them between applications. Agents are special type of objects in voyager

applications and are simply remote objects. These objects can exist outside of the local application's address space.

Voyager provides location transparency through forwarding chains of proxies. Voyager is an interesting platform, with several functionalities, which eases the development of distributed systems. A key disadvantage of Voyager is that it is a commercial product not available for free, which could prevent many researchers from using it in favor of other alternatives available. The forwarding chain mechanism used to track mobile agents could also be inefficient as the whole chain must be traversed in order to locate an agent, and weak as a single broken link makes the agent unreachable.

5. Property Analysis

Trillo et al [] provides a qualitative comparison among mobile agent platforms, Outgrattes [] provides a comparative study of mobile agent functions in different application areas and also on mobile agent platforms characteristics, Adnan et al [] provides pros and cons of various mobile agent systems and Arti et al [] provides an analysis Comparison of Various Toolkits. Here we provide an analysis of properties of various available agent platforms.

Table 1. Analysis of properties of various agent platforms

S. No.	Mobile Agent	Company	Properties
1	Agent Building & Learning Environment	IBM T.J. Watson research laboratory, 1999	Java Framework, Rule based programming Provides a distributed platform
3	Agent TCL	Dartmouth College, 1997	Java based programming Provided as third party software Flexible architecture support for Java, Good support for migration Simple communication facilities Lack execution constraints Inefficient, very much slower than optimized C Lacks OOP features, Provide no code modularization Difficult to write and debug large scripts Good security mechanisms
3	Aglets	IBM Tokio Research Laboratory, 1997	Free, open source, Java based Easy to develop Agent based applications Applet like development paradigm Build around single thread model Flexible, poor secured, provides basic security mechanisms Synchronous/ asynchronous messages support Does not support dynamic proxies weak agent mobility
4	Anchor Toolkit system	Lawrence Berkley National Laboratory USA	Java based Strongly secure management of mobile agents Works for heterogeneous distributed computing environment Weak agent mobility
5	ARA	University of Kaiserslautern, Germany	Runs on UNIX platforms Can run agents concurrently using a fast thread package Can clone themselves , duplicating their internal state Migration can be routed over a wireless link Agents can checkpoint their internal stat Need better error handling for agents failing remotely Mobile agents transmitted using TCP only

			Lacks sufficient security mechanisms
6	Concordia	Mitsubishi Electric Information Technology Center, America	Java based, provides flexible agent mobility Support for agent persistence and recovery Reliable guarantee of agent transfer Agent Communication and good security mechanisms
7	Grasshopper	GMD Focus IKV++, 1999	Java based Integration of client server and mobile agent technology Mobile agent system interoperability facility Provides basic internal and external security Not available any more
8	HTTP-Based Infrastructure for Mobile Agents	Gotha University, Germany	Support Agent mobility and communication Agents are transferred as MIME documents
9	JADE	Telecom Italia Lab, 1998	Java based Implementation of multi-agent system Facilitates the development of distributed mobile agents FIPA compliant agent platform Free/open source, Mobility is not key element No proxies Has good security mechanisms (JASS)
10	MAF		Python/ C/ C++ based Able to incorporate and integrate with foreign agents
11	Mole	1995	Java, Run on any java VM Allows good communication among Agents with the concept sessions Transparent to user, Low cost Lacks synchronous communication and execution constraints Does not support strong migration, Coding is not easy Good security
12	PIAX		Java based Open source framework Allows to build a scalable and efficient federated system Ubiquitous computing and Multi overlay mechanism
13	SensorWare		TCL/TK based Implementation of mobile agent environment for wireless sensor network
14	SMARD		Borland Delphi and Java based Design, build and launch mobile agent using graphical interface applications run on JADE platforms
15	SPRINGS	Distributed Information System Group University of Zaragoza, Spain	Easy to use, scalable and reliable Binds location transparency Does not support agent communication using FIPA standards Does not provide sophisticate security mechanisms Does not offer any graphical tool for user
16	TACOMA		TCL/TK, Perl, Python, and C based Requires programmer to explicitly capture state information before migration Agents can not be preempted by the TACOMA system, only stopped or aborted. Pure binary data not very well supported by Tacoma. Rudimentary security mechanisms
17	TAgent		Java, build open source mobile agent system Allows extensibility environment and secured environment Compliant with MASIF standards
18	Tracy	University of	Plug-in oriented architecture, Lightweight and extensible

		Jena, Germany 2005	Offers migration strategies for agents Does not support remote communication between agents Level of activity is low, no support for remote communication
19	Trinity		.NET based Provides a tool for rapid development of non complex agents Open source software for both personal and commercial use
20	Tryllian	Homonym company, 2001	Allows reactive and proactive behavior of agents Provides persistency service Does not offer location transparency
21	Voyager	Object Space, 1997	Java, .NET, C++ based, Support for RMI, DCOM, CORBA No special syntax, Simple communication Good support for transaction, messaging system is not good. Commercial product, only support java. Location transparency Weak agent mobility and Inefficient forwarding chain mechanisms Weak security managers

6. Conclusion

Mobile agent is very useful when very large volumes of data are stored at remote hosts. It is able to migrate from one computer to another autonomously and continue its execution on the destination computer. In this paper we presented the use of mobile agent technology in designing web search engines. We also discuss various available mobile agent technologies and an analysis of their properties. It is also evident that the development of mobile agents must lie in solutions to the maintain integrity, security and protection of mobile agents and the platforms, deciding route of mobile agent movement, inter-mobile agent-system communication and collaborations, monitoring the flow of information, execution of a running mobile agent. It needs to improve more the characteristics of mobile agents, as there is no single mobile agent framework that can be alternative to all of the functionality supported by mobile agents.

References

- [1] A. Lingnau, O. Drobnik, and P. Domel, "A HTTP-Based Infrastructure for Mobile Agents", in Proceedings of the Fourth International World Wide Web Conference, Boston, USA, 1995.
- [2] Aarti Singh, Dimple Juneja, A. K. Sharma, "Agent Development Toolkits", International Journal of Advancements in Technology, ISSN 0976-4860, Vol. 2, No. 1, pp. 158-164, 2011.
- [3] Abdelkader Outtagarts, "Mobile Agent-based Applications : a Survey", IJCSNS International Journal of Computer Science and Network Security, Vol. 9 No. 11, pp. 331-339, November 2009.
- [4] Ashutosh Dixit, Jitendra Kumar Seth and A. K. Sharma, "A migrating parallel exponential crawling approach to search engine" International Journal of Computer Science & Emerging Technologies (IJCSNET), Vol. 1, Issue 2, August 2010.
- [5] Bellifemine F., Caire G., Poggi A. and Rimassa G., "JADE: A white Paper", Available at <http://exp.telecomitalialab.com>, exp, Vol. 3, No. 3, September 2003.
- [6] C. B`aumer and T. Magedanz, "Grasshopper-a mobile agent platform for active telecommunication," in Third International Workshop Intelligent Agents for Telecommunication Applications (IATA'99), Stockholm, Sweden, pp. 19-32, August 1999.
- [7] C. Moemeng, V. Gorodetsky, Z. Zuo, Y. Yang and C. Zhan, "Agent-Based Distributed Data Mining: A Survey", Data Mining and Multiagent Integration, Springer Science Business Media, LLC 2009.
- [8] Dany B. Lange, "Mobile Objects and Mobile Agents : the Future of Distributed Computing," Proceedings of the European Conference on Object-Oriented Programming, 1998.

- [9] David Wong, Noemi Paciorek, Tom Walsh, Joe DiCelie, Mike Young, and Bill Peet, "Concordia: An Infrastructure for Collaborating Mobile Agents", Mitsubishi Electric ITA, Horizon Systems Laboratory.
- [10] Deepika, Ashutosh Dixit, A.K. Sharma, Niraj Singhal, "Security issues in Distributed Crawling based on Mobile Agents", in proceedings of 5th International IT Conference on "Downtrend Challenges in IT" DCIT-2009, conducted by PCTE, Ludhiana, Punjab, India, May 22, 2009.
- [11] Eid Mohamad, Artail Hassan, Kayssi Ayman and Chehab Ali, "Trends in Mobile Agent Applications" Journal of Research and Practice in Information Technology, Vol. 37, No. 4, pp. 323-351, November 2005.
- [12] G. Cabri, L. Leonardi and F. Zambonelli, "Mobile Agent Technology : Current Trends And Perspectives," Congresso Annuale AICA 98, Napoli, November, 1998.
- [13] H. S. Nwana, "Software Agents: An Overview," Knowledge Engineering Review, Cambridge University Press, Vol. 11, No. 3, pp.1-40, 1996.
- [14] Holger Peine and Torsten Stolpmann, "The Architecture of the Ara Platform for Mobile Agents", in proceedings of the first International Workshop on Mobile Agents, Berlin, Germany, 1997.
- [15] Jingsheng Zhao, Jiapeng Zhang, and Haiping Fang, "A Study of Mobile Agent-based Information Searching System", Proceedings of 2009 International Workshop on Information Security and Application (IWISA 2009), Qingdao, China, November 21-22, 2009, ISBN: 978-952-5726-06-0, 2009.
- [16] "Mobile Agent Computing", A White Paper, Mitsubishi Electric ITA, Horizon Systems Laboratory, January 1998.
- [17] Moemeng, C.; Gorodetsky, V.; Zuo, Z.; Yang, Y. and Zhang, C., "Agent-Based Distributed Data Mining: A Survey", Data Mining and Multiagent Integration, Springer Science Business Media, LLC 2009.
- [18] Nguyen G., Dang T. T., Hluchy L., Laclavik M., Balogh Z. and Budinska I., "Agent Platform Evaluation and Comparison", Published by Institute of informatics, Slovak Academy of Sciences, Pellucid 5FP IST - 2001-34519, June 2002.
- [19] Niraj Singhal, R. P. Agarwal, Ashutosh Dixit, A. K. Sharma, "Information Retrieval from the Web and Application of Migrating Crawler", published in IEEEExplore 978-0-7695-4587-5/11 © 2011 IEEE, DOI 10.1109/CICN.2011.99. pp. 476-480, 2011.
- [20] Niraj Singhal, R. P. Agarwal, Ashutosh Dixit, A. K. Sharma, "Reducing Network Traffic and Managing Volatile Web Contents Using Migrating Crawlers with Table of Variable Information", published in World Applied Sciences Journal, ISSN 1818-4952, Vol.19, No. 5, DOI: 10.5829/idosi.wasj.2012.19.05.1083, pp. 666-673, IDOSI Publications, 2012.
- [21] Nils Peter Sudmann, "TACOMA- Fundamental Abstractions supporting Agent Computing In A Distributed Environment", Department of Computer Science, University of Tromsø, Norway, November 1996.
- [22] Picco P.G., "Mobile agents: An Introduction", published in Microprocessors and Microsystems by Elsevier Sciences, vol. 25, pp. 65-74, 2001.
- [23] Prakash V. Rajguru, Sushant B. Deshmukh, "Analysis of Mobile Agent", Journal of Global Research in Computer Science, Vol. 2, No. 11, pp. 6-10, November 2011.
- [24] Glitho, R.H; Olougouna, E. and Pierre, S., "Mobile agents and their use for information retrieval: a brief overview and an elaborate case study", Ericsson Res., Montreal, Que IEEE, Jan/Feb 2002, Volume: 16, issue: 1, ISSN: 0890-8044, INSPEC Accession Number: 7165883, pp. 34-41, 2002.
- [25] Raquel Trillo, Sergio Ilarri and Eduardo Mena, "Comparison and Performance Evaluation of Mobile Agent Platforms", in proceeding of the Third International Conference on Autonomic and Autonomous Systems, ICAS '07, DOI: 10.1109/CONIELECOMP.2007.66, 2007.
- [26] Robert S. Gray, "Agent TCL: A transportable agent system", Proceedings of the CIKM Workshop on Intelligent Information Agents, Fourth International Conference on Information and Knowledge Management (CIKM 95), Baltimore, Maryland, December 1995.
- [27] S. Ilarri, R. Trillo, and E. Mena, "SPRINGS: A scalable platform for highly mobile agents in distributed computing environments", in Fourth International WWMM 2006 Workshop on Mobile Distributed Computing (MDC'06), Niagara Falls/Buffalo, New York, USA. IEEE Computer Society, June 2006.
- [28] Sharma, A.K.; Chauhan, Naresh and Goel, Amit, "An Agent based Crawler for Management of Volatile Information on World Wide Web", Proc. of National Conference on Communication & Computational techniques, Dehradun, February 2006.

- [29] Srilekha Mudumbai, Abdeliah Essiari and William Johnston, "Anchor Toolkit: A Secure Mobile Agent System," Proceedings of Mobile Agents '99 Conference, October 1999.
- [30] Syed Adnan, John Datuin, Pavana Yalamanchili, "A Survey of Mobile Agent Systems", CSE 221 Final Project June, 2000.
- [31] Yubao, M. and Renyuan, D., "Mobile Agent technology and Its Application in Distributed Data Mining", First International Workshop on Database Technology and Applications, 2009.
- [32] <http://aglets.sourceforge.net/> (accessed on October 1, 2012)
- [33] <http://jade.tilab.com/> (accessed on September 12, 2012)
- [34] <http://maf.sourceforge.net/> (accessed on October 3, 2012)
- [35] <http://sensorware.sourceforge.net/> (accessed on September 5, 2012)
- [36] <http://sid.cps.unizar.es/SPRINGS> (accessed on October 3, 2012)
- [37] <http://smard.sourceforge.net/> (accessed on October 1, 2012)
- [38] <http://trinitymaf.sourceforge.net/> (accessed on October 1, 2012)
- [39] <http://www.fipa.org/> (accessed on October 3, 2012)
- [40] <http://www.mobile-agents.org> (accessed on October 2, 2012)
- [41] <http://www.omg.org> (accessed on September 5, 2012)
- [42] <http://www.piax.org/en/> (accessed on September 5, 2012)
- [43] <http://www.research.ibm.com/able/> (accessed on October 3, 2012)
- [44] <http://www.tacoma.cs.uit.no/> (accessed on September 6, 2012)
- [45] <http://www.tagents.org/> (accessed on October 3, 2012)
- [46] <http://www.tcl.tk/doc/> (accessed on October 2, 2012)
- [47] <http://www.tryllian.org>, (accessed on October 2, 2012)