

ITray: Multi-agent Solution for LAN Based File Sharing

LDSB Weerasinghe^{1,#}, B Hettige¹, RPS Kathriarachchi¹, AS Karunananda¹

¹Department of Computer Science, General Sir John Kotelawala Defence University, Sri Lanka

#LDSB Weerasinghe; sidath.weerasinghe@gmail.com

Abstract— Multi-agent systems are capable of handling the complexity of the real world problems through its emerging features including coordination, communication and negotiations. Thus, Multi-agent systems can be applied to handle the environmental complexity of a computer network to achieve better performance and reduce resource wastage. For instance, some of the clients contain the huge workload when other clients are free on the network. In this situation, some task can be allocated to free clients to complete all tasks as much as possible. Further, most of the network resources are wasting for download same file in many clients. This resource wastage can be reduced by checking each client before downloading the file. This paper presents a distributed multi-agent system that can be used to reduce resource wastage of the Local Area Network. The Multi-agent system has been implemented through the MaSMT framework with selections of a managing agent and four ordinary agents, namely, file send agent, file receive agent, download agent and load balancing and dynamic scheduling agent. When the user feeds an URL to the system for download, the system communicates with other manager agents in the Local Area Network and checks whether the file is available or not. If the file is available, get that file from that client. Otherwise, the system will download the file through the file download agent. When the user feeds several URLs, the system follows previous procedure and allocates downloading tasks to other clients through the load balancing and dynamic scheduling agent when they are free. Task allocation, file sending and reserving has been done, though the agent-agent communication. The system has been successfully tested in a laboratory environment and successful results were archived when downloading numbers of large files such as video playlist.

Keywords— Multi-agent, File Sharing, MaSMT

I. INTRODUCTION

A Multi-Agent system (MAS) is a set of distributed software agents that cooperate to solve complex problems that are beyond the individual capacities each agent (Rzevski and Skobelev, 2014). Agents are very sensitive to environment and agents provide some features like act on behalf of others, autonomous, proactive, reactive, rationality, ability

to move, interact and intelligent. These features might fit for the understanding of the new generations of multi-agent-based applications.

These days Peer-to-peer (P2P) file sharing is very popular and has several characteristics such as there is no central point of control within the network, each client (node) is considered as an equal to all other connected peers, clients can easily share processing power, applications and content by direct connection between peers on the network. Cost sharing and reducing, resource aggregation, interoperability, autonomy and dynamism are some benefits in peer to peer architecture. Peers install client software (such as KaZaA (Leibowitz et al., 2003), Morpheus, eMule ("eMule-Project.net - Official eMule Homepage"), Limewire (Lewthwaite and Smith, 2008)) to share files with other peers. These all the system use to only share files with peers but they were not capable of sharing resources. The proposed system has capable of sharing resources with each peer on the network.

At the present time various types of download managers are used by computing people every day (*Download Managers A Better Downloading Experience*, 2008). The download manager is responsible for privacy, security and file management to the user and downloads any types of files. In a real environment, one person downloads the file and share with other using pen drive. Otherwise, all the people connected to the internet and try to download the same file. Since network traffic become increase and network become slow. It will affect all the peers on the network and wastage user's time. People download bulky number of files in one computer device when all other computer devices are free on the same network. These people hold huge workload in their devices because of that it makes time wastage. The networking task becomes more complex due to dynamic behavior of the network. Network status will be change time to time and network application also change frequently. Therefore another problem is difficult to schedule task in the network. These types of problems are overcome by the proposed system using multi-agent technology and the concept of file message passing, load balancing and the dynamic scheduling.

The proposed system is a java based multi-agent system developed using MaSMT framework for reducing the

network resource wastage and sharing files with one and many peers simultaneously. This system fully intelligent and automatic system consist with many functional requirements such as file send facility, file receive facility, file download facility, file searching facility over the network and can do some activates over the network. Proposed system store all peers IP addresses in the XML file and its work as agent ontology and system has five agents to accompanist the user's task.

The rest of this paper is organized as follows. Section 2 describes an overview of some existing systems. Section 3 reports brief note on the Multi-agent system and frameworks. Section 4 also gives design and implementation of the ITray. Then section 5 demonstrates how ITray works in areal environment. Finally, section 6 describe the evaluation, conclusion and the future work of the system.

II. RELATED WORKS

This section briefly describes the different approaches are used to model multi-agent systems and LAN based network applications with their technology advancements and limitations.

PViz is a multi-agent application use for visualizing peer to peer agent simulation (Giouroukis et al., 2015). As a classic software scenario for PViz, consider a researcher in the area of dis-tribute multi-agent systems who wants to use a simulator to test their newly designed protocol. PViz is developed using the Piccolo2D graphics library, which retains numerous capabilities useful for the task, specially zooming and node selection while being well combined with Java Swing.

"File Synchronization and Sharing: User Practices and Challenges" research paper published by Robert and others (Capar et al., 2014). Researchers do their research by considering 16 universities and identified major problems on file synchronization and file sharing. Aboveresearch found that factors that influence users to choice away for synchronization and file sharing namely privacy, security factors, the size of data, how software is familiar and how easy to do it. Researchers gather data by conduction semi-structured interview and do quantitative analysis in four main areas; what are the issues when collaborating, how files are synchronized among devices, how are the files accessed and data separation affect synchronizations. Participants described six primary methods for maintaining access to files: using emailing files to themselves, remote mounted drive, emailing files to themselves, USB drives, carrying a notebook computer, emailing files to themselves and using a remote desktop connection. Among these methods, participants noted portability, ease-of-use, and

universality as advantages of USB drives, and several people described carrying them on their person so that they would always have access to files when they were needed. Both manual and automatic synchronization methods are used by participants. They mostly concern about files size and the number of files in the syncing process. Researchers showing that people are faced some difficulties due to file sharing and synchronization process. The proposed system has capable for sharing files easier with each and every peer at the same time. So using proposed system these problems can be solved.

Authors Babaoglu and others developed a framework for agent-based peer to peer system called Anthill which is written in Java and is based on JXTA (Babaoglu et al., 2002). An Anthill system contains a dynamic network of peer nodes with societies of adaptive agents travel through peer network, cooperating with nodes and cooperating with other agents with the purpose of solve complex problems. The goals of Anthill are to provide an environment that simplifies the design and deployment of P2P systems. Anthill includes a simulation environment to aid developers analyze and evaluate the behavior of P2P systems prior to deployment. Developers can build P2P applications simply by defining the structure of the P2P system and designing suitable ant algorithms using the Anthill API for solving the problems. The researchers clearly mention the Anthill model by using nests and ants. The researchers are implementing ants that mimic the behavior of Freenet (Clarke et al., 2000), for the purpose of comparison with the Gnutant framework (Montresor et al) and studying how the reliability, availability and performance of hash-based routing may be improved.

The technology which is developing rapidly on the web today is mobile agent technology which can be applied for file download from remote server to remote client or server. C.R Nirmal developed a mobile multi-agent for multimedia file download in an e-commerce environment (Nirmala et al., 2010). Researchers send mobile agent embedded within it the various types of file names to be downloaded from the remote server by specifying the itinerary details. This technique appears to be faster when compared to traditional client-server FTP based file download system. The researchers discovered how mobile agents can successfully apply in place of common FTP and how mobile agents can be made to accelerate the speed of downloading. JAVA orientation of mobile agents provide several benefits in Internet system programming, in which there is a need for different kinds of incorporated monitoring, information, and notification, encapsulating

artificial intelligence techniques, security and robustness (Kotz and Gray, 1999). Researcher mentions the list of advantages of mobile agent for downloading purpose. This paper introduces a practical mobile agent scenario which could be used for the purpose of the downloading of files from different hosts which is very much necessary for an e-commerce and this is not suitable for handling complexity because of less communication.

III. MULTI-AGENT TECHNOLOGY

Every complex problem can easily solve by communicating with each other. Base on that multi-agent technology builds in with the power of communication with its agents and this technology become a new trend in artificial intelligence. One agent cannot get the perfect solution for any problem but communication with other agents can get the best solution for that problem. The problem converted more complex because of dynamic behavior, interconnected with each other, distributed and uncertain. In agent, technology agents have special features such as act on behalf of others, autonomous, proactive, reactive, intelligent, ability to move and interact. Most of the researchers use development framework such as JADE (Bellifemine et al., 2008) (Java Agent Development Framework), JaCaMo (Boissier et al., 2013), SeSam (Klgl, 2003) (Shell for Simulated Agent Systems), Madkit (Gutknecht and Ferber,) (Multi-Agent Development Kit), MaSMT and etc for develop multi-agent systems. Because of it save developer's time and developer's work capacity. JADE framework is a well-known formwork for developing multi-agent systems and it's based on JAVA language. MaSMT is a light weighted JAVA based multi-agent framework (Hettige et al., 2013). Madkit is JAVA based framework provide GUI facility for the agent debugging and agent simulation purpose. JaCaMo framework supports agent development for the android platform to mobile agents.

A. MaSMT Framework

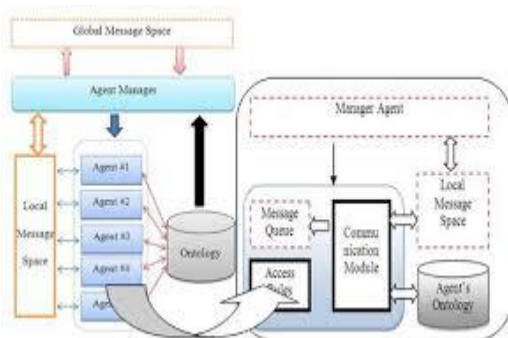


Figure1. MaSMT Framework Architecture

MaSMT is a light weighted multi-agent framework developed using JAVA language for JAVA environment and figure 1 shows the architecture of the MaSMT framework.

MaSMT has two types of agents namely manager agent and ordinary agent (Hettige et al., 2016). There is special agent called root agent which is a type of manager agent used to manage manager agents. This framework mainly designed for machine translation but it will improve as a general framework for all kind of multi-agent application and it is open source framework ("masmt - Browse Files at SourceForge.net,").

IV. DESIGN AND IMPLEMENTATION

This section briefly describes design and implementation of the Multi-agent system named ITray. ITray is specially designed for file transfer within the network, file download facility and dynamic scheduling with load balancing. ITray consists of one manager agent, five ordinary agents and ontology.

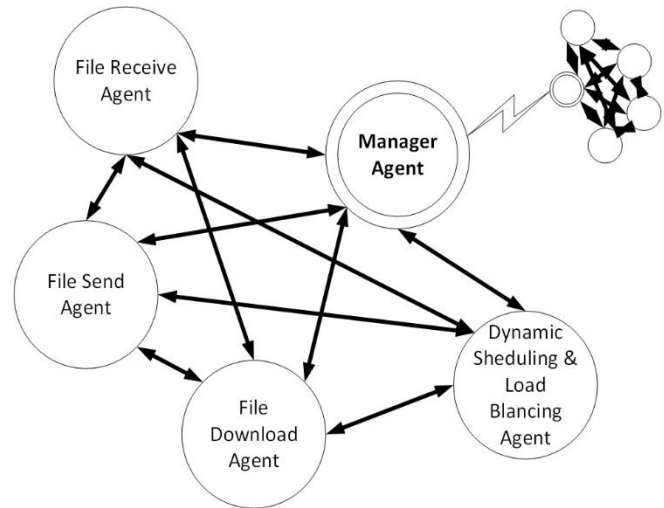


Figure2. ITray Agent Architecture.

Manager consists of server socket and that server always on when the application is in the tray. Manager agent can communicate with other manager agent in different computer devices within a network via java sockets. Manager contains GUI part and agent invokes other ordinary agents when do the appropriate task. Figure 2 show that the agent architecture of the ITray.

The GUI system interacts with the user and the ITray. GUI is build using java swing and user can give input and take output through the GUI. GUI is a part of manager agent and by clicking buttons the user can invoke several agents to do user's task. The user can drag and drop files to send to the other peers and can add download list to the GUI. The interface is built very user-friendly and affordances to the user.

Manager can invoke file sent agent which is an ordinary type agent in MaSMT framework by passing the message to

the send agent and that message contains IP address of the receiver agent and file location of the send file. According to this message, agent does their tasks. Agent can work with parallelism, can send a large number of huge file simultaneously without any interference. File send agent send file via java socket with a specific port.

Manager agent can invoke file receive agent which is an ordinary agent in MaSMT framework. Agent contains server socket powered by java. When manager invokes file receive agent; the agent listens specific port to get send the file. First, get the file name and after that get the file. This agent can get all kind of file types such as PDF, JPGE, EXE, RAR and etc. and can receive any size of files.

Download agent is an ordinary agent in MaSMT framework and manager agent can invoke agent by passing the message to that agent. Download agent waiting for a message and that message contains URL (Uniform Resource Identifier) of the file to be download which is entered by the user. If your assign several URL to the download agent, this agent can download the file in simultaneously with downgrading any performance. This agent makes parallelism to the system and can download binary files and any other kinds of files such as web pages and etc. Download agent file download speed depends on the network speed which is connected to the computer device.

ITray has an XML-based ontology which is used to store knowledge. In ITray ontology is used to store peers IP address of the network. When new user come to the network ontology is shared with that new user. The manager has a special capability to find an available agent of the LAN. The user can see the online peers on the network via GUI. Dynamic scheduling and load balancing agent use this ontology knowledge to distribute download URL load with other peers in the network by scheduling the task to each peer. This agent assigns URL to another computer device online agent to download that URL and send back download files to the send agent. For this assign

task agent considers several facts such as online agents, the number of URLs and workload of the agent.

V. ITRAY IN ACTION

This section briefly describes how ITray in works on particular LAN. The ITray intelligent application has been developed as a Java application and adapt to run on any environment such as Windows, Mac and Linux. When ITray is opened, it's in the system tray and user can click the ITray logo then can see the GUI of the ITray. Figure 3 shows the GUI of ITray.

For first time user for the LAN that user needs to request peers list from one peer in the network. Peers share their ontology knowledge with other peers. The user can check how many peers are the online on the LAN by using ITray very easier and ITray display the peer's IP address on the GUI by using XML-based ontology of the ITray. The user can drag one or more files and drop to the send basket of the ITray GUI for send that files to the other online peer or peers. User drag and drop and types of files and any size of files. ITray provides features to send files to specific peer or send files to all online peers on the LAN. After user's chosen, send agent is invoked by manager agent via local message passing to send agent. Send agent does their task according to the message. Other machine receiver agent gets the file and saves that received file in their download location.

When user copies a web URL it automatically pastes to the ITray. ITray always listens to the changes of the system clipboard and get the copied URL to the ITray's GUI. When the user clicks the download button, then ITray send that URL to all other online peers in the LAN. Other peer's manager agent get the URL and check that file is available with their system and if that file is available to invoke send agent and send that file. If that file is not available with other peers then it will add to download queue. After that manager invoke the download agent via sending a message to the download agent for download that URL. Download queue is become increase ITray's manager agent invoke dynamic scheduling and load balancing agent. Then distribute the URL to the other peers download the file and send it back to the sender. This agent dynamically schedules the task to the other agent system over the network. Other computer device agent system gets the URL and invokes their download agent by passing the message for download the URL file. After downloaded that file then it sends to the agent which sends the URL to the agent. This all process is done by ITray automatically by communicating with each agent and using intelligent mechanism.

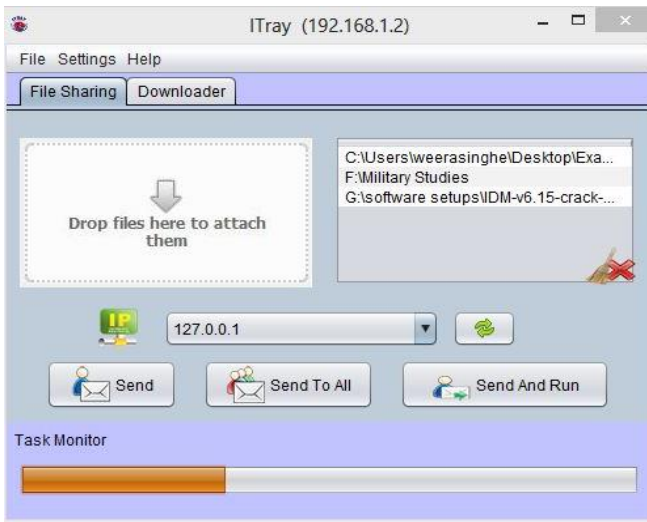


Figure 3. ITray GUI

VI. CONCLUSION AND FURTHER WORKS

This paper presented research of the design and implementation of the intelligent file sharing and communication application name ITray. ITray has a very complex communication with agents and it is a distributed over the network so that it designed as a multi - agent system using the MaSMT development framework. The ITray consist with the one manager agent as a core system and have four ordinary agents namely file send agent, file receive agent, download agent and dynamic scheduling and load balancing agent. ITray is developed using Java language and agent-agent communication done via FIPA Agent Communication Language (ACL) and multi-agent system – system communication is done via Java socket.

peer’s. These calculated statistics illustrate figure 4 graph. The graph shows that ITray is very efficiently share resources and download files within a short time. When ITray put into the real environment, some firewalls and virus guards blocked ITray some features. Some firewalls not route send files, they blocked and sometimes their not route all the data packets in some networks. Some virus guard’s blocks download feature of the ITray. These are the problems faced by the ITray in arealenvironment. For the future work, this system will improve to share all the resources on the computer within the LAN.

REFERENCES

Babaoglu, O., Meling, H., Montresor, A., 2002. Anthill: A framework for the development of agent-based peer-to-peer systems, in: Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE, pp. 15–22.

Bellifemine, F., Caire, G., Poggi, A., Rimassa, G., 2008. JADE: A software framework for developing multi-agent applications. Lessons learned. Inf. Softw. Technol. 50, 10–21. doi:10.1016/j.infsof.2007.10.008

Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A., 2013. Multi-agent Oriented Programming with JaCaMo. Sci Comput Program 78, 747–761. doi:10.1016/j.scico.2011.10.004

Browser Statistics [WWW Document], n.d. URL http://www.w3schools.com/browsers/browsers_stats.asp (accessed 7.23.16).

Capar, R., Vardell, E., Brennan, K., 2014. File Synchronization and Sharing: User Practices and Challenges. Seattle, USA.

Clarke, I., Sandberg, O., Wiley, B., Hong, T., 2000. Freenet: A distributed anonymous information storage and retrieval system, in: ICSI Workshop. Berkeley.

eMule-Project.net - Official eMule Homepage. Downloads, Help, Docu, News... [WWW Document], n.d. URL <http://www.emule-project.net/home/perl/general.cgi?l=1> (accessed 3.20.16).

Giouroukis, D., Platis, N., Tryfonopoulos, C., 2015. PViz: Visualising P2P Multi-Agent Simulations, in: Proceedings of the 2015 International Conference on Autonomous Agents

3 Peers Network Avarage Download Time

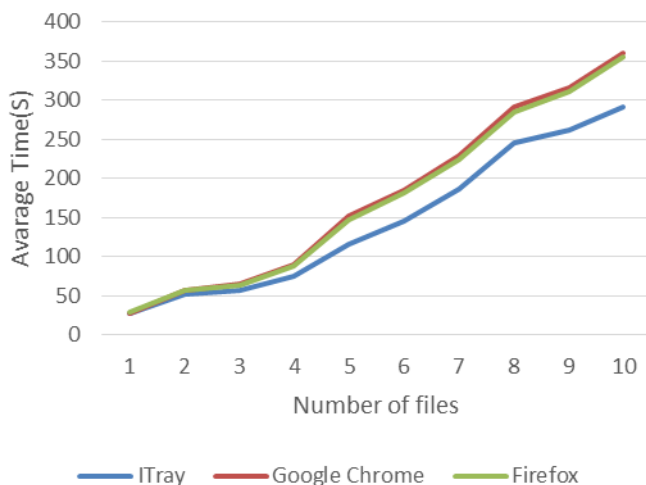


Figure 4.Graph of ITray Results

The ITray has been successfully tested in laboratory environments with different networks and a different number of machines. A lot of people use their web browsers like Google Chrome, Mozilla Firefox and Internet Explorer to download their files. For testing purpose, Google Chrome and Mozilla Firefox browsers are used. Because most of the people used these two web browsers (“Browser Statistics,”). Different types and different size of 10 files are used and calculate the average time to download these 10 files on different machines. These files are download using Google Chrome, Mozilla Firefox and ITray on different days in different time slots with different networks and get the average download time. ITray has used 3 peers’ network and share resources among these 3

and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1945–1946.

Gutknecht, O., Ferber, J., n.d. The MADKIT Agent Platform Architecture.

Hettige, B., Karunananda, A.S., Rzevski, G., 2016. A multi-agent solution for managing complexity in english to sinhala machine translation. *Int. J. Des. Nat. Ecodynamics* 11, 88–96. doi:10.2495/DNE-V11-N2-88-96

Hettige, B., Karunananda, A.S., Rzevski, G., 2013. MaSMT: A Multi-agent System Development Framework for English-Sinhala Machine Translation. *Int. J. Comput. Linguist. Nat. Lang. Process. IJCLNLP* 2, 411–416.

Klgl, F., 2003. The Multi-Agent Simulation Environment SeSAM.

Kotz, D., Gray, R., 1999. Mobile Agents and the Future of the Internet, in: ACM. pp. 7–13.

Leibowitz, N., Ripeanu, M., Wierzbicki, A., 2003. Deconstructing the Kazaa network. *IEEE Comput. Soc*, pp. 112–120. doi:10.1109/WIAPP.2003.1210295

Lewthwaite, J., Smith, V., 2008. Limewire examinations. *Digit. Investig.* 5, S96–S104. doi:10.1016/j.diin.2008.05.017

masmt - Browse Files at SourceForge.net [WWW Document], n.d. URL <https://sourceforge.net/projects/masmt/files/?source=navbar> (accessed 5.31.16).

Montresor, A., Glu, O.B., Meling, H., n.d. Gnutant: Free-Text Searching in Peer-to-Peer Systems. ResearchGate.

Nirmala, C.R., Ramaswamy, V., Jyothi, N.M., 2010. Multi Mobile Agents for Multimedia File Download in E-Commerce Environment. *Int. J. Comput. Appl.* 4, 9–14.

Rzevski, G., Skobelev, P., 2014. Managing Complexity.

White Paper: Download Managers A Better Downloading Experience, 2008. . Conceiva Pty. Ltd.