NET-Computer: Internet Computer Architecture and its Application in E-Commerce

Marcel O. Odhiambo
Dpt of Electrical and Mining Engineering
University of South Africa (UNISA)
Pretoria, South Africa
Ohangmo@unisa.ac.za

Patrice O. Umenne
Dpt of Electrical and Mining Engineering
University of South Africa (UNISA)
Pretoria, South Africa
umennpo@unisa.ac.za

Abstract— **Research in Intelligent Agents has yielded interesting results, some of which have been translated into commercial ventures. Intelligent Agents are executable software components that represent the user, perform tasks on behalf of the user and when the task terminates, the Agents send the result to the user. Intelligent Agents are best suited for the Internet: a collection of computers connected together in a world-wide computer network. Swarm and HYDRA computer architectures for Agents' execution were developed at the University of Surrey, UK in the 90s. The objective of the research was to develop a software-based computer architecture on which Agents execution could be explored. The combination of Intelligent Agents and HYDRA computer architecture gave rise to a new computer concept: the NET-Computer in which the computing resources reside on the Internet. The Internet computers form the hardware and software resources, and the user is provided with a simple interface to access the Internet and run user tasks. The Agents autonomously roam the Internet (NET-Computer) executing the tasks. A growing segment of the Internet is E-Commerce for online shopping for products and services. The Internet computing resources provide a marketplace for product suppliers and consumers alike. Consumers are looking for suppliers selling products and services, while suppliers are looking for buyers. Searching the vast amount of information available on the Internet causes a great deal of problems for both consumers and suppliers. Intelligent Agents executing on the NET-Computer can surf through the Internet and select specific information of interest to the user. The simulation results show that Intelligent Agents executing HYDRA computer architecture could be applied in E-Commerce.**

Keywords- Intelligent Agents; Internet; HYDRA computer architecture; NET-Computer; cloud computing; E-Commerce

I. INTRODUCTION

Agents' technology was developed from related works on distributed Artificial Intelligence (AI), in which researchers aimed at spreading intelligence behavior through distributed computer systems to tackle inherently distributed problems. They developed intelligent software programs (or Agents), the most advanced of which can communicate, collaborate and even learn from each other as they use their knowledge for problem solving.

Research on Intelligent Agents has been ongoing for sometime yielding interesting results. Research at University of Surrey, UK in late 90s resulted in HYDRA and Swarm computer architectures for Intelligent Agents execution. The motivation was to develop a software-based computer architecture based on compiled code (program) as opposed to interpreted code for Intelligent Agents execution. In addition there was a need to extend the developed architecture for Agents execution in parallel and distributed environments. The Agents executing on Swarm or HYDRA computer architecture traverse the network of hosts carrying the user rights or permissions and execute user tasks and, when the task terminates, the Agents carry the results back to the user.

The Internet provides an inter-connected worldwide network enabling the user to log onto and browse documents on a remote machine several thousands of miles away. Combining information processing with the Internet opens the door for all sorts of communication and information processing possibilities. Automatic banking or Automatic Teller Machines (ATM), Imaging, Remote Teaching and Telemarketing are among the many services that can be provided [1].

A growing segment of the Internet is Electronic Commerce. Consumers are looking for suppliers selling products and services on the Internet. Meanwhile, suppliers are looking for buyers to increase their market share. The vast amount of available information, however, causes a great deal of problems for both the consumer and the supplier. Often, a task executed online is not only time consuming but boring as well. Intelligent Agents are best suited for this type of task, since they can surf through the Internet selecting specific information of interest to the user [1].

The ultimate objective of Intelligent Agents is for the user to log on to the proposed NET-Computer (based on Intelligent Agents and the HYRDA computer architecture), issue task(s) to be autonomously executed, log off and then, at a later time, log on to collect the results. The Agents autonomously execute and locate items on the Internet while the user is offline and free time to carry out tasks requiring user intervention.

The simulation results show that the combination of Intelligent Agents and HYDRA computer architecture could be used to implement the NET-Computer in which the computing resources reside on the Internet and that the proposed NET-Computer architecture could be applied in E-Commerce. This paper is structured as follows: section I introduces the topic covering the objectives and the relevance of Intelligent Agents, the HYDRA Computer architecture, the NET-Computer and electronic commerce. Section II provides a definition of Intelligent Agents, their characteristics and motivation and concludes with a brief mention of architecture models. Section III provides a brief description of HYDRA computer architecture. Section IV describes the concept of the NET-Computer architecture based on a combination of Intelligent Agents and HYDRA computer architecture. Section V describes how the NET-Computer could be applied in E-commerce. Section VI provides a brief mention of cloud computing (not based on Intelligent Agents) as an indication of the trend towards Internet computing. Section VII describes the simulation of Intelligent Agents execution and section VIII concludes the paper and provides suggestions for further work.

II. INTELLIGENT AGENTS

The definition of an Agent falls everywhere along a continuum, from simple macros in which the user enters a few parameters to truly Intelligent Agents, which demonstrate learning abilities and artificial intelligence. An Agent [2, 3] is a software entity with some degree of autonomy, which carries out operations on behalf of the user or another program, and thus, represents or has knowledge of the user's goals or wishes.

Agents are the user's personal representatives or assistants on the Internet carrying the user identity, access rights (permissions) and responsibilities. Agents act on behalf of the user at the user's request or using some agreed user protocol. Agents communicate with their peers by exchanging messages on behalf of the user [4]. While Agents can be as simple as subroutines, typically they are larger entities with some sort of persistent control (e.g. distinct control threads within a single address space, distinct processes on a single machine or separate processes on different machines).

The characteristics of an Agent can be broken down into three main dimensions: agency, intelligence and mobility [5].

- "Agency" describes the Agents' degree of independence. Agents must be able to operate on the Internet while the user is disconnected or, in any case, not interacting with the web. The most basic Agent represents the user in some way and proceeds to complete a task with that information in mind. More complex Agents can interact with other programs and data.
- "Intelligence" is the amount of learned behavior or possible reasoning capacity that an Agent may possess. At the basic level, the Agent may follow a set of rules that are pre-defined by the user.
- "Mobility" is the dimension which makes Agents useful for Internet applications, yet poses many privacy and security issues. An Agent can be static and reside on the client computer to manage and gather information on that machine. A more mobile Agent may be composed on one computer (at the client or server end) and then sent to another to gather data and then return the data to the client.

Agents are divided into three main categories: Advisory, Assistant and Internet [1].

- "Advisory Agents" don't actually carry out tasks but they offer instructions and advice to help users do their work. These Agents are experts in a particular domain, but in the beginning, they have only rudimentary knowledge of the user, user's work patterns, and user preferences.
- "Assistant Agents" are more ambitious than Advisory Agents because they often act without direct feedback from the user. While this allows them to be much more powerful, it also raises a lot of technical and social issues that are yet to be resolved.
- "Internet Agents" and their related issues are especially relevant on the Internet. Along with the explosive growth of the web, has come the demand for tools to help manage and to sift through the vast amounts of available information, and Agents (variably known as Web Crawlers, Spiders, and Robots) often fit the bill. Some Internet Agents attempt to present an integrated view of the Internet as a whole, but the most common to date are information gatherers.

Some of the currently available Agents can save the user time by performing repetitive tasks such as gathering and processing information. With the help of an intelligent Agent, the user input could decrease to the time required to fire the Agent on the Internet. The Agent can work on/offline thus freeing the user to work on other projects or more enjoyable endeavors.

A. Agents Execution on the Internet

Agents freely roam the Internet searching for relevant information or services that are of interest to the user. Agents provide the user with the information, and the user is free to act otherwise. However, a higher level of Agent sophistication involves service performing Agents which execute the specific tasks on behalf of the user e.g. "find me the nearest Chinese restaurant and book me a dinner table for two this evening". Finally there are predictive Agents that volunteer information or services to the user without explicitly being asked [3]. For example, the Agent may monitor the stock exchange and return information pertaining to a particular item which might be of interest to the user.

Agents' execution on the Internet raises a lot of issues which needs to be addressed if Agents processing is to be widely accepted. Such issues include: authentication, or how much should the user trust the Agent especially when the Agents are delegated important decisions such as making financial commitments. Security of such a transaction should be guaranteed. In traversing the network searching for information, the Agents will need access permission at every site they visit in addition to the site authenticating the identity of the user that the Agent purports to represent [6]. This is to prevent Agents masquerading as some other user's representative. The need to guard against wayward programs that install viruses, compromise host, or pilfer the database. The resources used by the Agent at each site needs to be charged to the user. These and many others are the issues which must be addressed.

B. *Agents Motivation*

Four main motivations exist for the Agent paradigm. The first important motivation is performance improvement [4]. In Figure l(a), we have the common situation where the client and the server are located on two different nodes. To carry out computation, several messages will often be transferred back and forth between the two parties. There are performance advantages in moving the client to the server side of this connection. The amount of network interactions is reduced as a result of co-locating service requesters and service providers heavily engaged in communication. This scenario is illustrated in Figure 1(b).

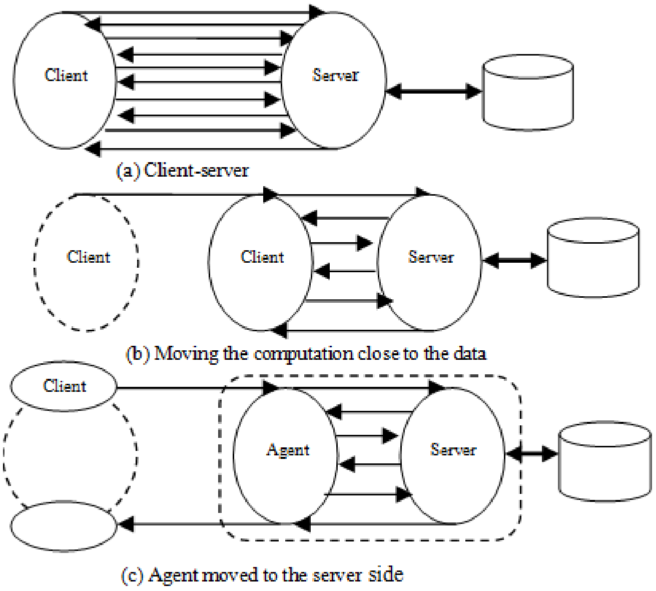
Client
Server
(a) Client-server
Client
Client
Server
(b) Moving the computation close to the data
Client
Agent
Server
(c) Agent moved to the server side

Fig. 1. Client- Server Agent Model

A second motivation for the Agent model is that it is intuitive. Agents acting more or less independently on behalf of somebody are a well-known concept. Figure l(c), represents a similar situation as in Figure l(b), but now the client itself does not move. It sends a representative to the server side to do the actual computation. The Agent will typically reduce the amount of data to be transferred over the network. In this example, the result is finally sent back to the client. A third motivation for the Agent paradigm is that communication is reduced to a local site issue. All that is needed is to move, or co-locate, Agents that wish to communicate. Locality of Agents is not hidden, but communication channels are. Finally, this model gives cleaner and simpler failure semantics. In a regular distributed system, the state between the client and the server is normally distributed. A failure half-way makes it tricky to ensure that both parties roll back their state to maintain consistency.

Alternative Agent architecture, such as server-server models shown in Figure 2, also exist. In the server model, the Agent server co-ordinates the activities of local Agents arriving from other hosts. Nevertheless, it is not the intention of this paper to carry out a comparative analysis of the advantages/disadvantages of the two agent models. The inclusion of the agent-server only serves the purpose of informing the reader of the existence of alternative agent model. A number of Agent-based systems have been implemented, tested and the simulation results obtained point to the feasibility of Agents processing. Agent systems are implemented either as interpreted program code, e.g. WAVE [7], TACOMA [8], Telescript [9-11], Agent Tcl [12], or compiled program code, e.g. Swarm [13], HYDRA [14]. In the interpreted program code version, the execution engine runs on each host, receiving, interpreting user programs (strings) and executing user commands. In the compiled program code version, the compiled program and user commands (compiled code) are uploaded and executed by the execution engine

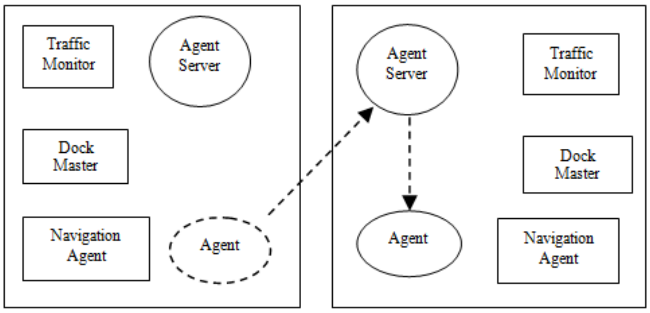

Fig. 2. Server – Server Agent model

III. HYDRA COMPUTER ARCHITECTURE

Parallel and Distributed Swarm computer architecture for Agent execution [14] is based on the Swarm computer architecture [13]. The HYDRA computer shown in Figure 3, consists of four modules: Access Point (AP) through which the user can access the HYDRA computer to issue tasks and collects the results after task termination, Routing Module (RM) responsible for routing Agents to local destinations, Processing Module (PM) is the execution engine and also the destination of an executing Agent, and Network Router (NR) that connects hosts on the network, routes Agents between the hosts and ensures message delivery to correct destinations. The Network Router and Routing Module together form the Communication Router (CR).

The CR uses an interconnection scheme to connect hosts in the network, assign host numbers to host names and creates communication channels among the hosts. The hosts in the network are grouped into domains, and the domains interconnected together form a distributed network. The NR then starts the RM running at each host, creates a connection point for Access Point (user interface) at each host and then, enters a continuous loop monitoring network hosts for user connection. During start up, the CR, starts the routing module at each host, establishes a listening end-point at each host and monitor message arrival at the end-points. The CR identifies both local and remote destinations to reliably deliver messages to these destinations. The Communication Router (CR) reliably delivers messages to the hosts.

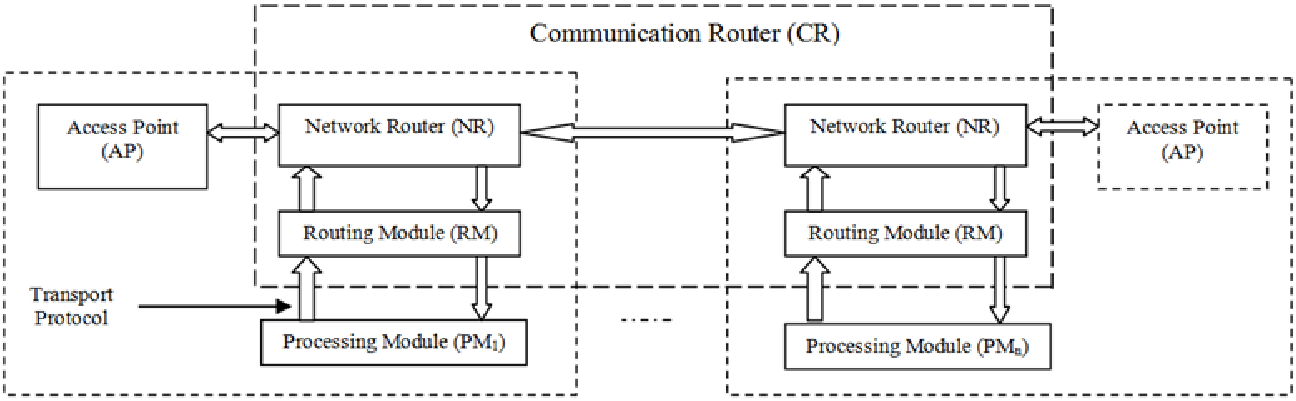

Fig. 3. Configuration of the NET-Computer

Agents navigate a network of hosts using hosts' addresses (PM number or hostname), executing a task, and when the task terminates, the Agents collects the result and sends them to the user. The HYDRA computer is a software implementation coded in C, running in the UNIX/Linux environment using the TCP/IP transport protocol. The HYDRA computing environment consists of network hosts connected together by an interconnection scheme to provide a distributed computer system.

The HYDRA computer prototype adopts a SPMD (Single Program Multiple Data) approach. At the beginning of task execution, the program is broadcasted to all hosts in the network running the prototype. The Agents then navigate the network carrying only the minimum status information comprising of task identification, destination address, status word, instruction pointer and/or data.

The objective was to develop and implement a parallel and distributed computer architecture on which Agents execution or characteristics can be simulated. HYDRA computer prototype was simulated on a network of hosts running other applications and being used by other users. Hence, performance analysis/comparison was not an objective. The main objective of the simulation was to verify that Agents executing on a HYDRA computer prototype could navigate the network to reach their destination using host address/name, execute the task and after task termination, collect the results and send them to the user. In essence, a computer prototype on which Agenst characteristics: Intelligence, Mobolity and Agency can be demonstarted

IV. THE NET-COMPUTER SYSTEM

The motivation for developing the NET-Computer is to shift the computing resources to the Internet and provide users a simple interface to access the Internet computing resources. The concept of Internet computing [15] is based on Intelligent Agents traversing the Internet executing user tasks.

Current estimates indicate that:

- Nearly 90% of the users run document processing applications.
- In an 8-hour office working day, effective computer usage is about 40% (when the computer is actively in use). In the home environment, this is less than 20%.
- The ordinary user uses less than 10% of the computing resources e.g. processor speed etc (except for scientific or data processing applications).

From the above statistics, it can rightly be argued that there is no point in providing the user with a powerful machine which spends most of its time idle. Thus, it would be worthwhile to concentrate the computing resources on the Internet and provide the user with a simple interface to access the computing resources. The user interface can be as simple as a pager, Ipad, tablet, personal organizer, etc.

The NET-COMPUTER [16] is a collection of Internet computers (servers) running user programs on demand. Both the hardware and software resources reside on the Internet. The user does not need to be aware of the exact location of the resources. The HYDRA computer architecture attached to the Internet servers shown in Figure 4(a) receives and executes the Agents. The Internet servers are connected together to form the Internet computing resources as shown in Figure 4(b). The user interface to the NET-Computer is a simple device with less complexity in hardware and software since the computing resources reside on the Internet.

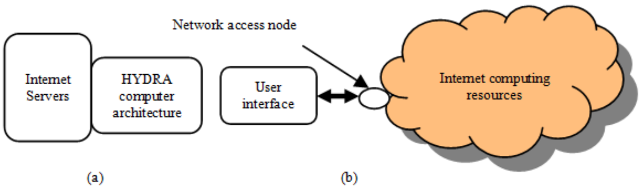

Fig. 4. Configuration of the NET-COMPUTER

V. CLOUD COMPUTING

Cloud computing [17] is all the rage. "It's become the phrase du jour," says Gartner senior analyst Ben Pring, echoing many of his peers. The problem is that (as with Web 2.0) everyone seems to have a different definition.

As a metaphor for the Internet, "the cloud" is a familiar cliché, but when combined with "computing", the meaning gets bigger and fuzzier. Some analysts and vendors define cloud computing narrowly as an updated version of utility computing: basically virtual servers [17] available over the Internet. Others go very broad, arguing anything outside the firewall is "in the cloud", including conventional outsourcing.

Cloud computing comes into focus only when one considers what IT always needs: a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing encompasses any subscription-based or pay-per-use service that extends IT's existing capabilities.

Cloud computing is at an early stage, with a motley crew of providers, large and small, delivering a slew of cloud-based services, from full-blown applications to storage services and spam filtering. Yes, utility-style infrastructure providers are part of the mix, but so are SaaS (Software as a Service) providers. Today, for the most part, IT must plug into cloud-based services individually, but cloud computing aggregators and integrators are already emerging.

Today, with such cloud-based interconnection seldom in evidence, cloud computing might be more accurately described as "sky computing", with many isolated clouds of services which IT customers must plug into individually. On the other hand, as virtualization permeate the enterprise, the idea of loosely coupled services running on an agile, scalable infrastructure should eventually make every enterprise a node in the cloud. It's a long-running trend with a far-out horizon. But among big metatrends, cloud computing is the hardest one to argue with. Cloud computing is thus an example of Internet computing though not based on Intelligent Agents. The inclusion of cloud computing is to provide an indication of the move towards Internet computing.

VI. APPLICATION OF THE NET-COMPUTER IN E-COMMERCE

A growing segment of the Internet is Electronic Commerce or E-Commerce [18]. Consumers are looking for suppliers selling products and services on the Internet. Meanwhile, suppliers are looking for buyers to increase their market share. The vast amount of available information causes a great deal of problems for both ends. Searching, a task executed online, is not only time consuming but boring as well. Intelligent Agents are best suited for this type of task. A typical campus Internet Infrastructure shown in Figure 5 is a collection of communication devices and networks of varying types and configurations connected together to form one network: the “Internet”. The tower and the high speed disk are used as a data warehouse for suppliers’ products and consumers’ needs and services. The remaining devices on the network provide user interface to the network clients. Intelligent Agents executing on such network will surf through the information on the Internet, resulting in the selection of specific information of interest to the user.

Intelligent Agents provide a mechanism for conducting E-Commerce. Using Intelligent Agents, the user composes the task offline, logs to the Internet to issue the task and logs off. At a later time the user can log to the Internet to collect the results of the task execution.

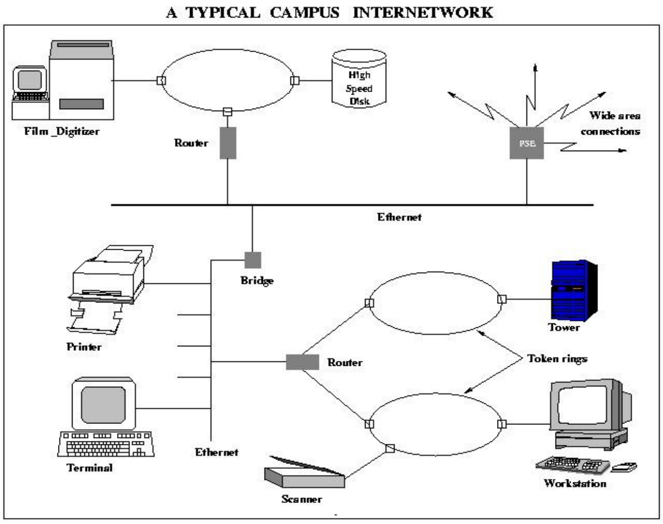

Fig. 5. Typical Internet Infrastructure

VII. SIMULATION TEST

The NET-Computer was simulated in a network of three hosts: ainur, kira and yavanna connected as shown in Figure 6. The hosts connected to networks A, B and C shown in Figure 7 are located on different floors of the same building and run UNIX operating systems.

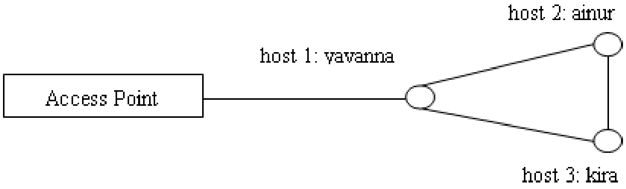

Fig. 6. Simulation test network

The objective of the simulation test is to test the following Agents characteristics:

- Mobility: Agents navigate the network and reach destination address (host).
- Agency or task execution: Agents carry out useful work on behalf of the user.
- Intelligence: Agents follow a set of rules predefined by user.

In addition, the user is at liberty to configure and select the hosts making up the HYDRA computer system as explained in section III. At the beginning of execution, a node file containing node distribution (PM number and links to other nodes in the network) is prepared for each PM. At each PM a unique number is stored which is accessed by Agents visiting the node. The aim here is to demonstrate the ability of the

Agents to access a stored data item. The unique number is stored in the persistent workspace 3 (PW3) which is accessible and can be modified by Agents belonging to the same task. The unique number is retrieved by the Agents and used in a simple addition routine.

In the test routine, the Agents load a value of 0000001Ahex to the Agent's workspace 2 (AW2), the Agents then navigate the network to node 1 in all PMs. On reaching its destination, the Agents retrieve the number stored at PW3 and add it five times to the number carried by the Agents in AW2 and store the result in PW3 as shown in Table I. The Agents then navigate through the network to node 1 in PM3 where each Agent adds the contents of AW2 to the contents of PW3 in PM3 and stores the result in AW3. The final result is stored in PW4 in PM3 as shown in Table II.

The test program is described as follows:

AW2, AW3, AW4 - Agent variables in the Agent workspace.
PW3, PW4 - persistent workspace (nodal variables)
01 Assign a value to AW2
02 Insert destinations PM (PM1-3) address in the destination list (DL).
03 Insert destination node address (node 1) in the destination list (DL).
04 Spawn to destination address in the destination list.
05 Load a value from persistent workspace PW3 to AW3.
06 Add contents of AW3 to contents AW2 and store the result in AW2.
07 Add one (1) to contents of AW4.
08 Test if contents of AW4 equal 5.
09 If contents of AW4 are less than 5, go back to 06.
10 Store the contents of AW2 in persistent workspace PW3.
11 Store the value 0 in persistent workspace PW4.
12 Insert destination PM (PM3) address in the destination list.
13 Insert destination node (node 1) address in the destination list.
14 Spawn to the address in the destination list.
15 Load a value from persistent workspace PW3 to AW3.
16 Add contents of AW3 to contents AW2 and store the result in AW3.
17 Load a value from persistent workspace PW4 to AW4.
18 Add contents of AW3 to contents AW4 and store the result in AW3.
19 Store the contents of AW3 to persistent workspace PW4
20 Terminate.

21 CONCLUSION

During the simulation test, the Agents carrying a data item traversed the network using destination addresses (PM number). At each visited PM, the Agents retrieved the data item stored at the site and added the data to the data item carried by the Agent, executing the addition loop five times and stored the final result at the visited site. At the end of the addition, the Agent retrieved the final result. The Agent then traversed the network to PM3. At the final destination in PM3, the Agent retrieved the stored data, added it to the data carried in the Agents workspace and stored the result at PM3. The results are presented below in Tables I and II.

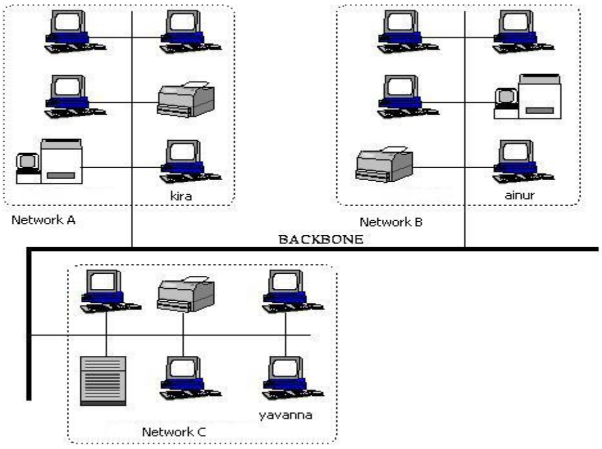

Fig. 7. The NET-COMPUTER execution network

TABLE I. AGENTS ADD CONTENTS OF AW2 TO THE CONTENTS OF PW3

PM Number	PM1	PM2	PM3	Loop count
Value in PW3	90	120	150	
Value in AW2	0000001A	0000001A	0000001A	
Sub total in AW2	000000AA	0000013A	0000016A	1
Sub total in AW2	0000014A	0000025A	000003BA	2
Sub total in AW2	000001DA	0000037A	0000051A	3
Sub total in AW2	0000027A	0000049A	0000066A	4
Sub total in AW2	0000031A	000005BA	000007BA	5
Total in PW3	0000031A	000005BA	000007BA	

TABLE II. AGENTS ADD CONTENTS OF AW2 TO THE CONTENTS OF PW3 AND STORE THE RESULTS IN PW4 IN PM3

PM1	PM2	PM3	Sub Total	Total in PM3
0000031A (AW2)	-	000007BA (PW3)	00000ADE (AW3)	00000ADE (PW4)
	000005BA (AW2)	000007BA (PW3	00000D85 (AW3	00001974 (PW4)
	-	000007BA (PW3) - 000007BA (AW2)	00000F55 (PW3)	000029C9 (PW4)

The simulation tests were carried out to test Agents ability to traverse the network using destination (PM) addresses, access a data item stored at a site, manipulate the data item and store the final result at the site or communicate the data item (result) to a different location. In the proposed NET-Computer, the computing resources are distributed over a wide geographical area. Thus, Agents navigated the network to reach their execution environment and the user at the end of the task execution to deliver the results.

The hosts used in the simulation test are connected to physically separated networks. We can therefore conclude that the Agents traversed the network to reach their destinations. Thus, confirming Agents navigation ability in the HYDRA computer architecture. The simple addition exercise demonstrated that Agents can be used to carry out some useful work such as manipulating the user program. The ability of the Agents to access stored information (data), manipulate the data and carry the data to different locations within the network, demonstrates the Agents' ability to collect/access, modify data and communicate the same. The simulation test results have demonstrated the feasibility of the NET-Computer based on HYDRA computer architecture and Intelligent Agents. However, more research work is needed to adapt the HYDRA architecture for deployment on the Internet. The simulation results confirmed Agents characteristics of mobility (traversing the network to reach destination host and user), Agency (carrying out addition loop on behalf of the user) and Intelligence (accessing stored information/date).

A. Application of the Research

Searching for a service on the Internet is a difficult task given the vast amount of information available; this is further complicated by the duplication of sites offering the same service. The ability to execute parallel search on the Internet and to receive the results that matches the user interest is a very welcome help. The characteristics of Agents autonomously roaming the network with no central control best suits applications such as relational and network databases, processing in graph-like structures, intelligent and mobile communication amongst others. The simulation results show that the HYDRA computer prototype could be used to implement distributed and parallel computer based LANS, WANs and the Internet, in which the computing resources residual on the Net side. Agent-based processing could be a major development on the Internet supporting electronic commerce/shopping and information processing amongst others.

B. Implication of the Research

The Internet consisting of a collection of computers interconnected together has made it possible for service providers to electronically avail services to the users. For example, a number of organizations have sprung up offering electronic mail and Internet related services. Users of these services log on to the mail servers to both send and receive mail and then log off. The user terminal serves as a simple input/output device through which the user can compose their mail offline, log on to the mail server to send/receive mail then log off. In the HYDRA computer prototype, the Access Point serves as the input/output device. The Processing Modules are akin to the Internet computers. It is thus feasible to build a distributed computer system based on a HYDRA computer prototype and Intelligent Agents. The computing power of such a computer system resides with the Internet computers and the user is provided with a simple input/output terminal with minimal computing power.

C. Future Studies and Recommendations

Security and fault tolerance are desirable and important characteristics in distributed systems however, these were not taken into account in the HYDRA prototype. The security of the site (host) and Agents must be guaranteed before a guest Agent is permitted to execute at a site. Fault-tolerance would enable the system to operate (though down-graded) in the presence of a fault or to be brought down gracefully.

Routing tables as currently implemented are based on immediate neighborhood (network hosts directly connected to each other). It would be infeasible to implement such routing tables on the Internet, considering the vast number of connected computers. The network could be partitioned into sections, each with a unique number or address. Gateway hosts would then be assigned to handle a section of the network depending on the assigned address or number.

The program executed in the HYDRA computer prototype is written in Swarm instruction (assembler language) which is very tedious, time consuming and prone to errors. Having a translator to translate programs from and to high-level language would make writing of the code user friendly and easier to spot and correct errors.

Moreover, the delays experienced when sending Agents and during execution of user tasks and the fact that the participating hosts run other programs as well, have not been taken into account in the HYDRA computer architecture. These and other matters will be the subject of further investigation in the next phase of the research. Further, the simulation was run with the user connected online. The next phase of the research will investigate the feasibility of the user to log to the NET-Computer, issue a task and log off. At a later time, the user can log to the NET-Computer to collect the results of the task execution. Through cloud or sky computing technologies, the Internet computing concepts described in [16] are beginning to become a reality.

The combination of Intelligent Agents and the NET-Computer might just provide the mechanism for furthering Electronic Commerce on the Internet. However, a lot of work still needs to be done to make Agents processing on the Internet feasible. Issues such as security, accessibility, fault-tolerance, authentication, charges for consumed resources etc still need to be adequately addressed.

References

[1] K. Indermaur, "Baby Steps: They may not fulfill our dreams yet, but agents and smart software are beginning to help us find information and do our jobs more effectively", BYTE, Vol. 20, No. 3, pp. 97-104, 1995

[2] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Upper Saddle River, New Jersey: Prentice Hall, 2003

[3] Intelligent Agents. MIT Encyclopedia

[4] M. R. Genesereth, S. P. Ketchpel, "Software Agents", Comms. of the ACM, Vol. 37, No.7, pp. 48-54. 1994.

[5] N. Jennings, M. Wooldridge, "Software Agents", IEE Review, Vol. 42, No. 1, pp. 17-20, 1996

[6] P. Wayner, "Free Agents: A new generation of light-weight, multithreaded operating environments provide security and interoperability for agent developers", BYTE, Vol. 20, No. 3, pp. 105-114, 1995

[7] P. S. Sapaty, P. M. Borst, An Overview of WAVE Language and Systems for Distributed Processing in Open Networks, University of Surrey, Department of Electronic and Electrical Engineering, Technical report, 1994.

[8] R. Renesse, D. Johansen, F. Scheneider, An Introduction to the TACOMA Distributed System Version 1.0, Technical report: University of Tromso (Norway) and Cornell University (USA), June, 1995.

[9] J. E. White, "Telescript Technology: The Foundation for the Electronic Marketplace", General Magic White Paper. General Magic Inc., 1994

[10] J. E. White, "Telescript Technology: Scene from the Electronic Marketplace", General Magic White Paper. General Magic Inc, 1996

[11] J. E. White, "Telescript Technology: Mobile Agents", General Magic White Paper. General Magic Inc., 1996

[12] R. S. Gray, Agent Tel: A Transportable Agent Systems, Department of Computer Science, Dartmouth College, 1995.

[13] L. De Errico, Agent Based Distributed Parallel Processing, Dissertation for the degree of Doctorate of Philosophy, Department of Electrical and Electronic Engineering, University of Surrey, 1996.

[14] M. O. Odhiambo, "HYDRA: Parallel and Distributed Swarm Computer Architecture", ROVPIA '99" International Conference on Robotics, Vision and Parallel Processing for Automation, Ipoh, Malaysia, 1999.

[15] M. O. Odhiambo, "Intelligent Agents: The Advent of the Net-Computer", Kenya Engineer. A Journal of the Institution of Engineers of Kenya, 1999.

[16] M. O. Odhiambo, "NET-COMPUTER: Internet Computer Architecture Based on Intelligent Agents", WORLDCOMP'11: The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing (Internet Computing (ICOMP'11), Las Vegas, USA, 2011.

[17] E. Knorr, G. Gruman, "What cloud computing really means", Inforworld, http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031

[18] M. O. Odhiambo, P. Umenne, P. A. Rayolla. "Application of Intelligent Agents in E-Commerce", The 7th Annual International Conference, Knowledge Creation and Dissemination for the Realization of Millennium Development Goals, Moi University, Eldoret, Kenya, 2011.