

MAINS: MULTI-AGENT INTELLIGENT SERVICE ARCHITECTURE FOR CLOUD COMPUTING

T. Joshva Devadas¹, S. Subbulakshmi² and R. Rajaguru³

Department of Information Technology, Sethu Institute of Technology, India

E-mail: ¹call_joshva@yahoo.com, ²infoforlakshmi@gmail.com and ³rajaguru.rama@gmail.com

Abstract

Computing has been transformed to a model having commoditized services. These services are modeled similar to the utility services water and electricity. The Internet has been stunningly successful over the course of past three decades in supporting multitude of distributed applications and a wide variety of network technologies. However, its popularity has become the biggest impediment to its further growth with the handheld devices mobile and laptops. Agents are intelligent software system that works on behalf of others. Agents are incorporated in many innovative applications in order to improve the performance of the system. Agent uses its possessed knowledge to react with the system and helps to improve the performance. Agents are introduced in the cloud computing is to minimize the response time when similar request is raised from an end user in the globe. In this paper, we have introduced a Multi Agent Intelligent system (MAINS) prior to cloud service models and it was tested using sample dataset. Performance of the MAINS layer was analyzed in three aspects and the outcome of the analysis proves that MAINS Layer provides a flexible model to create cloud applications and deploying them in variety of applications.

Keywords:

Agents, Cloud, Architecture, Intelligence

1. INTRODUCTION

“Pay for what you use” is the fantabulous premise for cloud computing [1]. This can be achieved by continuously monitoring the type of query request arriving at our system in the given time interval.

Agents are task oriented with the ability to learn by themselves and they react to the situation. The agent learns by verifying its previous experience from its knowledge base and updates its knowledgebase if the agent learns anything new. Researchers make use of agents to improve the performance of the system.

Agents with learning capability are termed as intelligent processing systems that can use its characteristics to observe, monitor and react to the environment. A new discipline, called agent-based Cloud computing must be set for providing agent-based solutions founded on the design and development of software agents for improving cloud resources and service management and discovery, SLA negotiation, and service composition.

Autonomous agents can make clouds smarter in the interaction with users and more efficient in allocating processing and storage to applications. Recent innovations on agent based computing spreads over data mining, image processing, wireless sensor networks and so in. Agents are software programs that perform tasks on behalf of others.

Recent research on cloud computing introduces agents that reduce the time needed for processing and thus improves the performance of computing by not repeating those tasks that had

been already being carried out by the agents [3]. In this paper we propose an agent layer which is placed in between the end user and database. This layer is termed as Multi Agent Cloud Intelligent service (MAINS) layer that makes the language understandable to both the end user and the Cloud Infrastructure. When more number of users are accessing the cloud database storage for the first time or in the subsequent times, the proposed Cloud architecture with the use of agents works with improved performance.

Section 2 describes the existing significance of mighty cloud architecture. Section 3 describes the proposed idea of introducing an agent based their intelligent services layer that reduces the computation time of our system.

Section 4 notifies the significance of agents and the cause of inserting the MAINS layer in between the end user and cloud. Section 5 describes the implementation of agent cloud system with its challenges. Section 6 describes the quality evaluation based on its respective parameters.

2. BACKGROUND

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) described in [2]. Every service request in the existing cloud architecture is processed and sent it to the requested user as services. The main significant services of cloud are on-demand self-service, broad network access, resource pooling, and rapid elasticity respectively.

On-Demand Self-service: A consumer can unilaterally provision capabilities such as server time, network storage as needed automatically without requiring human interaction with each service provider.

Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms such as mobile phones, laptops, PDA's.

Resource Pooling: The computing resources are pooled to serve multiple consumers using a multi-tenant model with different physical and virtual resources dynamically assigned and reassigned according to the consumer demand.

Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in.

The existing cloud infrastructure architecture depicted in Fig.1 focuses on its unique service models such as, Infrastructure as a service (IaaS), Platform as a service (PaaS), and Software as a service (SaaS) and deployment models like, Public cloud, Private Cloud and Hybrid Cloud [11].

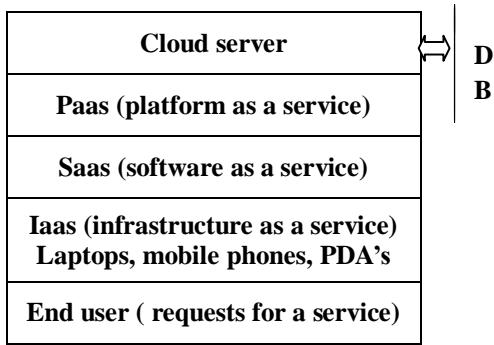


Fig.1. Existing Cloud Architecture

3. PROPOSED SYSTEM

The primary goal of cloud computing is to have Server somewhere access the information anywhere. This attractive innovation for storage had made the world to turn a while with surprise. Agent-based applications can rely on cloud computing infrastructures to access and use vast amounts of processors and data. So this approach would allow offloading the compute intensive agents to the appropriate subsets of processes and storage elements in a cloud.

The major anticipation of user in cloud is to access data from network with minimal subscription and minimal time. But some bottlenecks occur due to multiple access with cloud database for similar processing in our existing cloud infrastructure. Moreover it lacks in storing knowledge and needs improved processing.

To overcome this bottleneck problem we need to add an improved facility. Hence our present work wishes to accommodate an intelligent service layer that uses multiple agents which helps to improve the performance and reduces computation time.

3.1 NEED FOR AGENTS

Agents are introduced in the proposed system in order to make use of its characteristics such as, autonomy, pro-activity, reactivity, negotiation, learning, communication and cooperation.

The Autonomy characteristics of an agent describe the capacity to act autonomously to some degree on behalf of users or other programs also by modifying the way in which they achieve their objectives.

Pro-activity signifies the capacity of agents to pursue their own individual set goals, including by making decisions as result of internal decisions.

The re-activity characteristics enforce the agents to react to external events and stimuli and consequently adapt their behavior and make decisions to carry out their tasks.

Agents Negotiates with other agents to carry out organized conversations to achieve a degree of cooperation with other agents.

Agents are capable of learning from the environment or from the other agents, updates it in its knowledge base and use it to react with the environment during decision making with improved performance

Communication and Cooperation characteristics helps the agents to interact and communicate with other agents (in multiple agent systems), to exchange information, receive instructions and give responses and cooperate to fulfill their own goals.

3.2 INTELLIGENT LAYER MODELING

To provide knowledge we ought to model an intelligence layer. When modeling the intelligence the functional elements are identified and roles are assigned with respect to the interactions. The components are identified and mapped into architectural components and uses respective connectors to establish interactions.

The Interaction Component interacts with end user and transfers the data to the analysis agent. The Analysis Component analyzes the query given by end user. It verifies the request whether it's known or already processed. The Processing Component receives the data from analyzer and converts to the compatible query format. The Message Handler Component contains all the user assistive information for effective communication. The knowledge management Component controls and coordinates all activities in the agent infrastructure.

These architectural elements are mapped into intelligent processing elements called an agent that performs tasks such as knowledge processing, reacting with the environment, share knowledge by communicating cooperating and collaborating with agents. Thus these intelligent programmable elements are termed as,

- Interaction Agent
- Analysis Agent
- Processing Agent
- Message Handler
- Knowledge Management Agent

These five agents are introduced to formulate the Multi Agent Intelligent Service Layer (MAINS). This layer is introduced in the cloud architecture to reduce the processing time, assist the end user and react intelligently using its knowledgebase. All the agents present in this layer carries out the work concurrently which thereby improves the performance of the system consumed at minimal time.

Hence the incorporation of MAINS layer in cloud would allow offloading the compute intensive agents to the appropriate subsets of processes and storage elements in a Cloud.

3.3 MAINS ARCHITECTURE

Basically cloud computing is described based on the service model. Thus the components of MAINS architecture is modeled based on the services rendered by the components. This section describes the services of the MAINS architectural components in detail.

3.3.1 Interaction Agent Service:

Interaction agent (IA) accepts the service request from the end user and forwards the received service request to the analyzer. Once the data is processed, it accepts the processed information from analyzer and submits the same to end user. It can prepare interaction dialogues with association of MHA.

When a task is initiated by the user then this agent initiates all other agents present in the system (One in turn instantiates other threads which achieves concurrent activeness) that makes all agents all time ready.

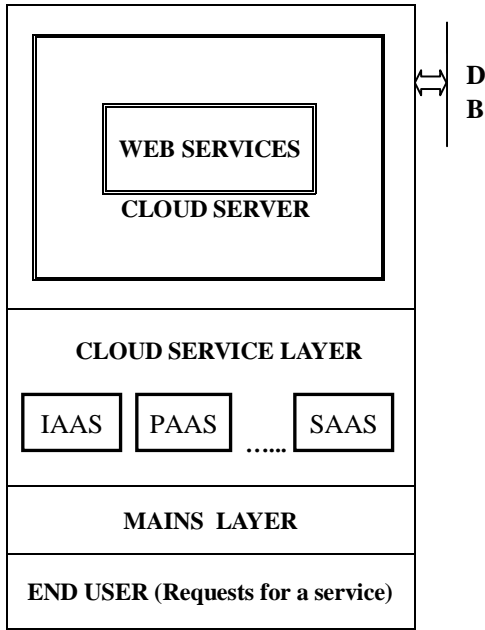


Fig.2. Proposed Intelligent Cloud Architecture

3.3.2 Analysis Agent Service:

Analyzes whether the service requested by end user is already known or new and is done in two step fashion. It verifies whether the request arrived from the end user is already processed, if so then retrieves the data from its possessed knowledge. Otherwise if the data is new to our MAINS then it forwards the service request to the processing agents for service. Metadata consists of information like who, how the data was queried, created, accessed.

3.3.3 Processing Agent Service:

Receive the metadata information from analyzer agent. The metadata format is converted to the required query format and sent to cloud service layer for further processing. Receives the processed data from cloud service layer and updates KB, then the processed information is continued by forwarded to IA.

3.3.4 Message Handling Agent Service:

Contain user assistive information needed for agents to interact and users to respond. Whenever the agents are in need of establishing communication with end user or with other agents it prepares appropriate dialogue from Data Dictionary (DD) or from its Knowledge Base (KB).

3.3.5 Knowledge Management Agent Service:

Whenever knowledge sharing is needed knowledge management agent assists all the other agents with the required knowledge. All the user log information is present in knowledge management agent.

The Knowledge Management Agent control and coordinate with the other agents using its possessed knowledge. Since all the agents have its in built learning capability knowledge

management agent learns to update its KB from the cloud by validating and verifying its recent updates.

If the request is a known request then this agent shadows the following scenario. Similar user request arriving our agent-cloud infrastructure is based on local or regional interest. When the knowledge management agent receives a request from the analysis agent, it verifies its knowledgebase to ensure that the request has been already processed or not by verifying the user log information.

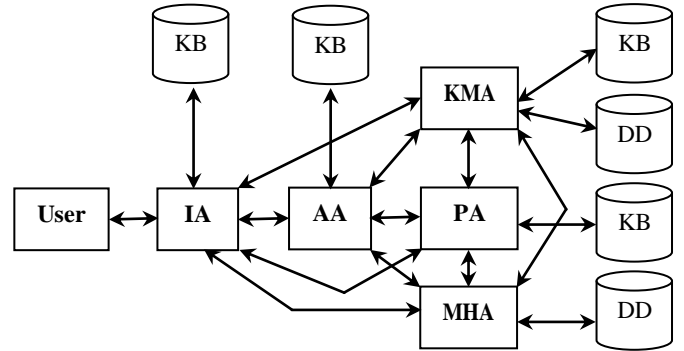


Fig.3. MAINS Architecture

If it is the novice request then the knowledge management agent forwards the request to the Analysis Agent for further processing. Otherwise knowledge management agent verifies the user log information with the user request from its knowledge base and extracts the relevant information from it, and forwards the extracted information to the IA for further processing.

3.3.6 Design Challenges of MAINS Layer:

Challenges in defining, describing, decomposing and allocating the roles among the available agents are considered as a major issue. It requires defining the language and protocols needed for interaction and communication. Describing the agent behavior in making decision, taking actions and formulating the coordination, agent states, process associated with action, plans and knowledge are considered as some of the other important issues.

4. IMPLEMENTATION

Advances in data collection and storage have allowed researches to create massive, complex and heterogeneous database which have stymied traditional methods of data analysis referred in [3]. This led to the analytical tools that often combine techniques from variety of fields such as Statistics, Computer Science and Mathematics to extract meaningful knowledge from data. Weka 3.6 is a machine learning tool which contains collection of visualization tools and algorithms for data processing.

The input training data set considers one hundred and thirty two training samples and is given to weka tool that uses SQL database to store all the information. With 10 fold cross validation the given dataset was analyzed to compute the performance metrics accuracy, precision and QoS. The following algorithm describes the learning process of the Knowledge Management Agent [10].

End user access the intelligent cloud architecture for a specific service request to check the edibility of mushroom.

Interaction agent of MAINS layer receives the user request and initiates all the agents present in the MAINS layer and forwards the request to the analyzer agent for further processing. Analyzer agent verifies the request by analyzing the given request is present in its knowledgebase by enquiring knowledge management agent.

The knowledge management agent verifies its knowledgebase, if the requested edibility of mushroom is already requested to this intelligent cloud architecture is present then Knowledge management agent extracts the data from knowledgebase and forwards it to processing agent for further processing.

The Processing agent converts the metadata received from the analyzer agent into desired query format. The converted query is forwarded to the cloud service layer. Cloud service layer makes use of the appropriate service which in turn avail the web services offered by the cloud server to complete the task.

The processed information from the cloud service layer is given to processing agent. This agent in turn updates the processed information in knowledge base of knowledge management agent and forwards the processed outcome to interaction agent. Interaction agent in turn sends it to the end user.

4.1 ALGORITHM

```

Procedure verify-validate ()
{
  ReadUserRequest ()
    //Receive data from end user
  Mapping ()
    // Map the user request with the user
    // Log information from
    // Knowledgebase (KB)
  IF USRReq Exists {Present in KB}
    Extract the relevant information from
    Knowledgebase
    Forwards the extracted information to
    Interaction Agent
  ELSE // First time
    Register the user request as user log
    Information in KB
    Forwards the new request to AA for
    Further processing
  END IF
}
    
```

4.2 DATA SET INFORMATION

To implement the MAINS layer in the cloud architecture data samples are taken from mushroom database of UCI repository. This data set includes descriptions of hypothetical samples corresponding to 22 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. To select an edible mushroom the dataset considers twenty attributes and one hundred and thirty two

samples. The edibility of mushroom can be interpreted with the attributes present in Table.1.

The Cap shape, size, color, odor, gill spacing and some of the vital attributes considered for training the data set referred in Table.3. According to the combination of these attributes edibility, moderate and poisonous stages of mushrooms are taken into account.

Table.1. Sample Training Dataset

Sl. No.	Cap Shape	Cap Surf	Cap Color	Bruises	Odor	Gill Spacing	Gill Size	Justify
1	b	f	n	t	a	c	n	Use immediately
2	c	g	b	f	l	w	b	Care to be taken
3	x	y	c	f	c	d	n	Do not use
4	f	s	g	t	y	d	b	Under observation
5	k	f	r	t	f	c	n	Do not use
6	s	g	p	f	p	w	b	Never Use
7	c	f	b	t	f	w	b	Usage restricted
8	s	y	b	t	a	w	n	Under spoilage
9	c	y	b	t	a	w	n	Do not use
10	k	y	b	t	f	w	n	Spoiled

4.3 PERFORMANCE IMPROVEMENT

When knowledge management agent receives a new user request, the energy spent by the knowledge management agent is determined by analyzing the computation time needed for processing the request [13]. It is done by using the energy coefficient formula as specified in Eq.(1).

4.3.1 Energy Co-Efficient Calculation at First Time:

The energy consumed at first time by the agent cloud infrastructure can be calculated by the following Eq.(1)

$$CE_{NT} = (T_{tetn} - T_{tetc})/T_{tetn} \tag{1}$$

where, $0 \leq CE_{NT} < 1$. Characterizing energy at normal time (CE_{NT}) will not be equal to1, because the total elapsed time of cloud will not be zero. Moreover CE_{NT} cannot be negative because time taken by cloud to process the request will not be greater than the total processing time in network.

$$T_{tetn} - \text{Total elapsed time in network}$$

$$T_{tetc} - \text{Total elapsed time in cloud}$$

4.3.2 Energy Co-Efficient Calculation at Subsequent Time:

The energy consumed at subsequent time by the agent cloud infrastructure can be calculated by the following Eq.(2)

$$AE_{ST} = (T_{tetc} - T_{teta}) / T_{teta} \tag{2}$$

where, T_{teta} – Total elapsed time in agent

$$T_{tetc} - \text{Total elapsed time in cloud}$$

$$AE_{ST} - \text{Agent Energy at subsequent time}$$

The performance of the first and subsequent search concludes the following.

- i. $AE_{ST} \leq CE_{NT}$
- ii. If $CE_{NT} - AE_{ST} = k$, total waiting time of the user in between the fundamental report and the modified report. Here, $0 < k < 1$.

It is clear that the agent based search minimizes the computation time enormously for subsequent search. To prove the derived concept this paper includes a new MAINS layer to the cloud architecture.

5. PERFORMANCE ANALYSIS

The best rating of a technology is ever accepted by its performance evaluation. We have analyzed our proposed work in three aspects based on QoS parameters performance evaluation, ROC curve analysis and accuracy vs instance. The QoS parameter performance evaluation was conducted in two dimensions namely first and subsequent time of query processing. The outcome of analysis was depicted in Table.2 and Fig.3.

Table.2. Performance Evaluation

Time/Attributes	CPU Usage	Processing Time	Efficiency	Energy (Power)
First Time	3	4	5	0.7
Subsequent Time	1.5	3	7	0.3

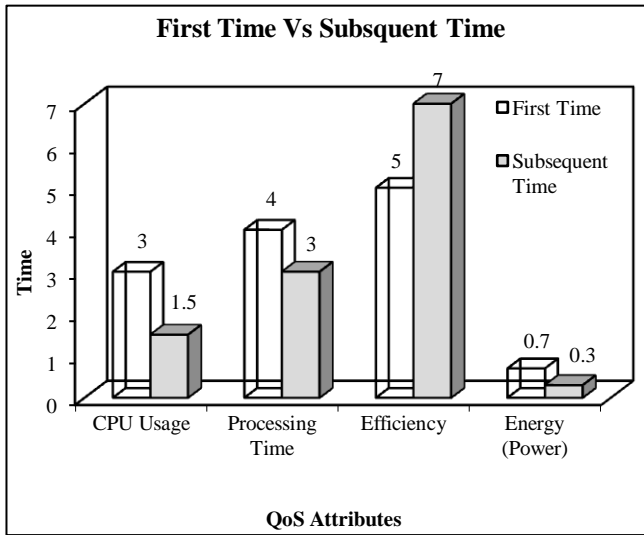


Fig.3. First Time vs. Subsequent Time

Table.4. ROC Prediction

Algorithm	25 Instances	50 Instances	75 Instances	100 Instances
ID3	0.73	0.88	0.83	0.88
C4.5	0.77	0.82	0.84	0.86
NB-TREE	0.81	0.89	0.88	0.97

Table.5. Accuracy Prediction

Algorithm / Instance	25 Instances	50 Instances	75 Instances	100 Instances
ID3	88	72	82	79
C4.5	96	78	86	80
NB-TREE	96	74	89	88

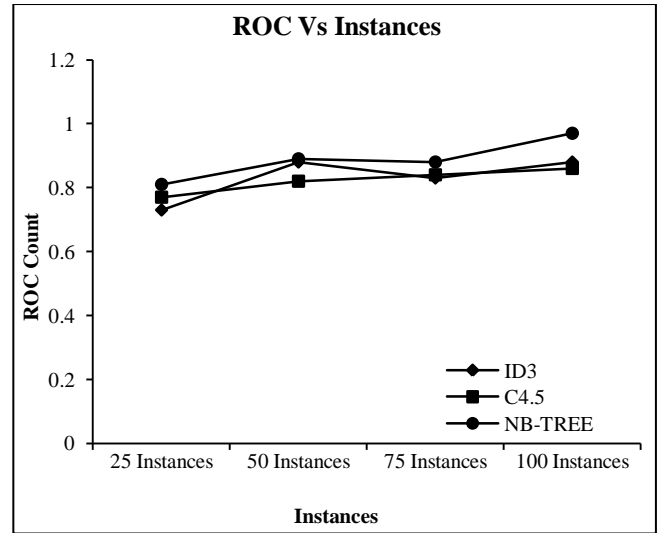


Fig.4. Roc Curve

The ROC (Receiver Operating Characteristics) curves evaluate the performance of the learning algorithm based on the two characteristics True Positive Rate (TPR) and False Positive Rate (FPR) of the given data set. Since the value of the decision variable is of Boolean valued function, decision tree based machine learning algorithms are found to be more suitable [9].

Hence, we have considered the machine learning algorithms ID3, C4.5 and NB-TREE. These algorithms are used to analyze the performance metrics ROC and Accuracy. The outcome of the analysis was depicted in Table.4, Table.5, Fig.4 and Fig.5 respectively. We observe that NB-TREE found to be more suitable for the given data set.

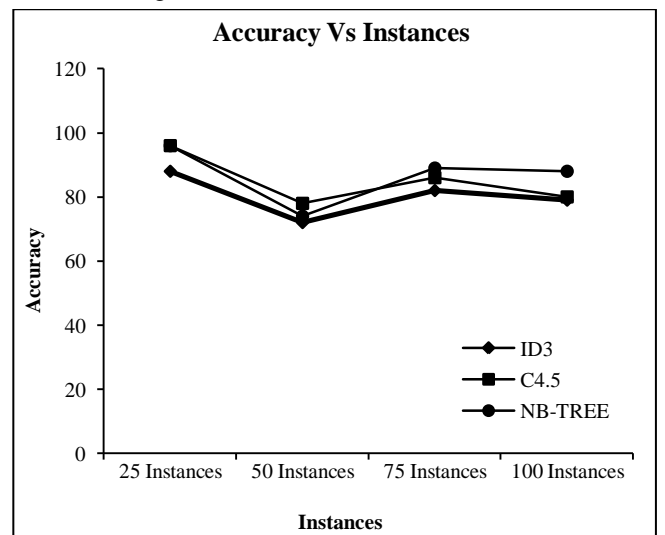


Fig.5. Accuracy Curve

Table.3. Sample Test Data

			Sunken = s	Knobbed = k	Flat = f	Convex = x	Conical = c	Bell = b	Cap shape
					Smooth = s	Scaly = y	Grooves = g	Fibrous = f	Cap Surface
Red = e	Purple = u	Pink = p		Green = r	Gray = g	Cinnamon = c	Buff = b	Brown = n	Cap Color
							No = f	Bruises = t	Bruises
Pungent = p	None = n	Musty = m		Foul = f	Fishy = y	Creosote = c	Anise = l	Almond = a	Odor
				Notched = n		Free = f	Descending = d	Attached = a	Gill Attachment
							Distant = d	Close = c	Gill Spacing
							Narrow = n	Broad = b	Gill Size
Pink = p	Orange = o	Green = r		Gray = g	Chocolate = h	Buff = b	Brown = n	Black = k	Gill Color
							Tapering = t	Enlarging = e	Stalk Shape
	Missing=?	Rooted = r		Rhizomorphs = z	Equal = e	Cup = u	Club = c	Bulbous = b	Stalk Root
	Pink = p	Orange = o		Green = r	Gray = g	Chocolate = h	Buff = b	Brown = n	Stalk Color
Cinnamon =	Pink = p	Orange = o		Green = r	Gray = g	Chocolate = h	Buff = b	Brown = n	Stalk Color Below
					Pink = p	Orange = o	Green = r	White = w	Stalk Shape Above
				White = w	Pink = p	Orange = o	Green = r	Gray = g	Stalk Surface Above
				White = w	Pink = p	Orange = o	Green = r	Gray = g	Stalk Surface Below
							Universal = u	Partial = p	Veil Type
								Brown = b	Veil Color
						Two = t	One = o	None = n	Ring Number
Zone = z	Sheathing = s	Pendant = p		None = n	Large = l	Flaring = f	Evanescent = e	Cobwebby = c	Ring Type
	Wood = d	Waste = w		Urban = u	Paths = p	Meadows = m	Leaves = l	Grasses = g	Habitat
		Solitary = y		Several = v	Scattered = s	Numerous = n	Clustered = c	Abundant = a	Population

6. CONCLUSION

Agent based cloud infrastructure is described in terms of rationality and perfection. Rationality maximizes the expected performance and perfection maximizes the actual performance. Performance is based on response time.

The intelligent cloud architecture flourishes with qualified deployment of MAINS layer which reduces the total processing time with improved performance by the analysis of ROC and accuracy curve prediction. This work may further be deployed and extended to serve the telecom industry, social networks and in Big Data Analytics where the data ranges from 10 to 150 TB.

REFERENCES

- [1] Rajkumar Buyya, et al., "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Generation Computer Systems*, Vol. 25, No. 6, pp. 599-616, 2009.
- [2] Cloud Security Alliance, "Security best practices for cloud computing", 2010.
- [3] Stephen D. Bay, Dennis Kibler, Michael J. Pazzani and Pandhraic Smyth, "The UCI KDD Archive of larger data steps for data mining Research and Experimentation", *ACM SIGKDD*, Vol. 2, No. 2, pp. 14-18, 2000.
- [4] Fujitsu, "Fujitsu Cloud Solutions", 2014. Available at: <http://www.fujitsu.com/global/news/pr/archives/month/2012/2012071801.html>.
- [5] E-people, "The people's voice is the voice of heaven", Anti-Corruption and Civil Rights Commission. Available at: <http://www.epeople.go.kr/jsp/user/on/eng/intro01.jsp>.
- [6] H. Adkisson, "The technology chase", 2005. Available at: <http://www.iathink.com/2005/11/index.html>.
- [7] J. Allaire, "Building New Business Models on the Web", SYS-CON Media Inc., 2009.
- [8] R. Alt, R. Heutschi and H. Osterle, "Webservices: Hype odor losung? Outtasking statt outsourcing von Geschäftsprozessen", *New Management*, No. 1/2, pp. 63-70, 2003.
- [9] G. Arumugam and T. Joshva Devadas, "Text database cleaning by filling the Missing values using Object Oriented Intelligent Multi-Agent System Data Cleaning Architecture", *International Journal of Computer Science and Information Technologies*, Vol. 1, No. 5, pp. 454-464, 2010.
- [10] S. Subbulakshmi, B. Praveen kumar, Raj Aravind and Sathish Kumar, "Worldwide Credentials of Individual by DNA Pattern Using Cloud Computing", *International Journal of Computer Science and Technology*, Vol. 4, No. 3, 2013.
- [11] S. Subbulakshmi and B. Praveen Kumar, "Web Mashup Integration End users using QoS and Proficient Assessment", *International Journal of Computer Applications*, Vol. 75, No. 5, pp. 1-5, 2013.
- [12] Sheng-Yuan Yang, Dong-Liang Lee, Kune-Yao Chen and Chun-Liang Hsu, "Energy-Saving Information Multi-agent System with Web Services for Cloud Computing", *Security-Enriched Urban Computing and Smart Grid Communications in Computer and Information Science*, Vol. 223, pp. 222-233, 2011.
- [13] S.Y. Yang, H.C. Chiang and K.J. Wu, "Developing an Intelligent Energy-saving Information Processing and Decision Supporting System", *Proceedings of Symposium on Constructing Industrial and Academic Park of Green Energy Science and Technology and Intelligent Energy-saving Techniques and Project Achievement Lunching Ceremony*, Taipei, Taiwan, pp. 41-47, 2010.
- [14] T. Anderson, L. Peterson, S. Shenker and J. Turner, "Overcoming the Internet impasse through virtualization", *Computer*, Vol. 38, No. 4, pp. 34-41, 2005.
- [15] J. Turner and D. Taylor, "Diversifying the internet", *Proceedings of the IEEE Global Telecommunications Conference*, Vol. 2, pp. 760-766, 2005.
- [16] N. Feamster, L. Gao and J. Rexford, "How to lease the internet in your spare time", *SIGCOMM Computer Communication Review*, Vol. 37, No. 1, pp. 61-64, 2007.
- [17] N.M.M.K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges", *IEEE Communications Magazine*, Vol. 47, No. 7, pp. 20-26, 2009.
- [18] N.M. Mosharaf Kabir Chowdhury and Raouf Boutaba, "A survey of network virtualization", *Computer Networks*, Vol. 54, pp. 862-876, 2010.
- [19] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145, 2011.
- [20] M. Armbrust, et al., "A view of cloud computing", *Communications of the ACM*, Vol. 53, No. 4, pp. 50-58, 2010.