INTELLIGENT QUERY PROCESSING FROM BIOTECHNOLOGICAL DATABASE USING CO-OPERATING AGENTS IN A SECURE CLOUD ENVIRONMENT

R. Geetha[1] and K. L. Shunmuganathan[2]
[1]College of Agriculture, Kerala Agricultural University, Vellayani, Trivandrum, India
[2]Department of Computer Science and Engineering, Dhanalakshmi Srinivasan College of Engineering, Chennai, India
E-Mail: geetharado@gmail.com

ABSTRACT

Cloud computing is a scalable and flexible environment that delivers economical computing facilities to clients based on queries. Response to query depends upon the speed of pattern matching, error correction, and resource allocation during query processing. The algorithms for query processing, sequence alignments of data from NCBI, EMBL, DDBJ and storage of recurring complex enquiries, are complicated and time consuming in the HADOOP framework. In order to obtain secure, swift and consistent response to queries, multi-agents have been introduced as a substitute for Job Tracker. They enhance the pace of query handling, allocate tasks to the Task Trackers, boost fault tolerance and increase reliability. Multi-Agents have also been deployed amid user and Agents, in the middle of Agents and between platforms to provide security measures. Agent development is done following the FIPA (Foundation for Intelligent Physical Agents) standards, using FIPA ACL (Agent Communication Language).

Keywords: multi-agent, hadoop, cloud computing, query processing, pattern matching, sentinel, FIPA ACL.

1. INTRODUCTION

Escalated online data and query traffic has resulted in the advent of cloud computing. Amazon's EC_2 (Elastic Compute Cloud), Amazon's S3 (Simple Storage Service) Google's GFS (Google File System) are a few cloud computing providers. Cloud computing has elasticity and scalability. To harness these features of the cloud for analysis of large volumes of data, the Big Data concept was introduced into the cloud using Hadoop MapReduce framework. Hadoop MapReduce and HDFS (Hadoop Distributed

File System) are the predominant and competent models in the cloud environs that improve data processing efficiency.

The biotechnological databases have an overload of data, whose analysis and understanding would be beneficial. To execute the data on these copious databases e.g. NCBI, DDBJ, EMBL; they must be emulated onto the cloud. The data is duplicated and disseminated over numerous resources. During query processing, the input queries are fragmented before transmission to the task nodes. In the current method of data replication each task requires a different completion time. It is important to consider this time difference before further allocation of task to the nodes.

When handling complex queries, essential auto correction, elimination of duplication and redundancy within queries, enhance the processing time. Also, heartbeats from slave nodes inform about the active status of each node.

Biological databases get updated regularly. So updating these emulations is also important, to ensure integrity in the results obtained. However, the present-day situation does not support repetitive updations. Again, it is imperative that the storage location of the data be secured. Security is also required between the virtual machines, amidst platforms, and during the transfer of data from/to the original database.

To arbitrate swift query processing and augment accuracy and veracity of the results obtained thereof, we introduce co-operating multi-agents in the Hadoop MapReduce.

The agents are developed using JADE (Java Agent Development Environment) in compliance with FIPA specifications in the FIPA OS.

A. Cloud Computing

This is an internet based network of remote resources to store, manage and process data in parallel. Cloud computing resources have loosely coupled architecture, are agile, have multi-tenancy, have device and location independence and maintain data in multiple sites. As a result, the system efficiency is based on query response and error tolerance [1], [2], [3].

B. Big Data and Hadoop Framework

Organizations collect large volumes of structured and unstructured data, which with appropriate analysis become a source of information to assist in decision making, improve efficiency and reduce cost. Hadoop MapReduce would be a system that can process data analytically in real-time [4-7]. Hadoop is an open-source structure for the storage of big data. It can be built up from single servers to multitudes and follows the master-slave architecture [8]. Hadoop Map Reduce is software frameworks to assist in the development of application that can parallely handle immense data on hardware clusters. Hadoop Distributed File System (HDFS) replicates query and assigns tasks to nodes, thereby easing the execution of queries in MapReduce. When a user submits a query through MapReduce application, the JobTracker in the Master section divides the job to minor tasks and sends to the scheduler, which in turn assigns tasks to each node in the TaskTracker (slave) [9], [10]. However, any small failure in the JobTracker results in a complete failure of Hadoop MapReduce services.

The biotechnological databases like NCBI, EMBL, and DDBJ are repositories for data related to DNA, RNA and protein sequences along with pathways, gene expressions and ontologies. Analysing this multifaceted and diverse data at high speeds would provide valuable information that can enhance research [11], [12].

C. Security in the cloud

With users moving their data to the cloud environment, large volumes of data are left exposed to malicious cyber-attacks [15]. As more and more data is deployed to the cloud, the security issues are amplified. Security measures include deterrent controls which act as a fence to stop attacks, preventive controls that provide strong authentication methods, detective controls that detect an attack that has been made and corrective controls that take steps to reduce the effects of an attack. Security and privacy to access is maintained through physical and personal security. Data security involves access authentication and confidentiality [16], [17], [18].

D. Multi agents

Agents are software programs that perform tasks on behalf of a user or another program. The expertise of Agents is based on what is designated in the "action" part. The system framework generates instances of the Agent actions and links them to the Agent object. A Multi-Agent system is a society of such systems that are capable of adjusting to situational variations, interact and exchange knowledge, to achieve a common goal. This means that they are flexible, autonomous and social. Agent based cloud computing aids in augmenting these multi agent qualities. [19-30].

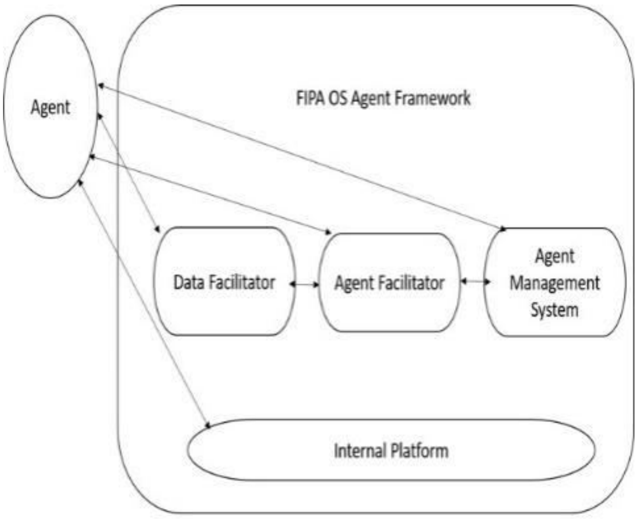

Figure-1. Agent platform.

E. Jade

Java Agent Development Environment is a robust and efficient background that follows the FIPA standards. Due to its JAVA base, it can effortlessly network with Java implementations. Jade environments are called containers. The main container in JADE has three automatically created agents called

RMA: the Remote Management Agent which handles the GUI interface

AMS: the Agent Management Service, which keeps of all agents and JADE programs

DF: the Directory Facility, which advertises agent services

The Main Container must be started first. A central Java RMI registry, keeps a record of all active containers to ensure effortless agent communication.

F. FIPA ACL

Agent interaction takes place using standards called Agent Communication Language (ACL). Foundation for Intelligent Physical Agents (FIPA) ACL is an example of communication language. It promotes message-based dialogue between agents. There are about 20 sets of ontologies called performatives, a housekeeping section and the content section. Communication using ACL eliminates the inaccuracies that transpire in conventional means. FIPA ACL actively supports insertion and deletion of services, even at run-time.

G. FIPA Open source agent platform

FIPA OS is an open agent platform that supports communication between agents using ACL. It is a Rapid Application Developer (RAD), composed of Directory facilitator (DF), which acts as the yellow pages; Agent Management System (AMS), that facilitates the Universal Description, Discovery and Integration (UDDI) for a Universal Registry of Web Services; Agent Communication Channel (ACC), which encourages inter-operability between VMs in the same platform or between different platforms; and the Internal Platform Message Transport (IPMT), that provides forwarding messages and addresses [32]. This is an abstract architecture specification that encourages iclusion of new agents as and when the need arises. The interfaces of the platform are unambiguous but loosely coupled and hence can be redefined. This openness improves the chances of implementation of the FIPA standards, agent interactions and spontaneous addition and deletion of facilities. The FIPA OS has been developed to create diverse multi-agent platforms [33] [34].

H. Query processing

Parsing and translation, optimization, translation and execution are the major steps in Query Processing [35]. Selection Query Model, where tuples are directly assigned to scores; Data Access Model, where the data are classified according to the processing techniques; Implementation Level, where the techniques are divided based on the level of integration with the DBMS. Query and Data Uncertainty, wherein query processing takes place based on the data ambiguity; Ranking Function, where query is processed depending on the scores [36-38].

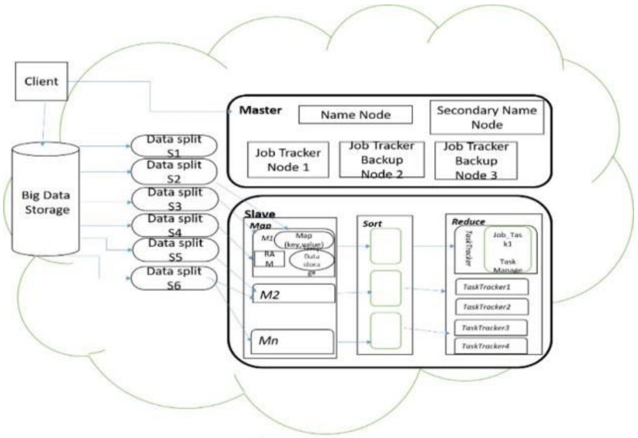

Fig 2: Query Processing

2. EXISTING SYSTEM

The existing work in the area of Query processing with the help of Co-operating Multi-Agents uses Hadoop MapReduce and FIPA OS in the cloud environment. To overcome the problem of the Job Tracker as a single point of failure, Multi-Agents are introduced in place of JobTracker. These aid in processing and storage of queries and reduce redundant query operations. The Agents introduced are Pattern Matching Agent, to check if it is a frequent query, Decomposition Agent for auto-correction, normalization and transformation of complex queries into simple sub queries, Scheduling Agent used to allocate sub-queries to the task tracker based on the execution time, Provisioning Agent for identification of the execution plan with least time, and Recovery Agent that receives heartbeats from the slave nodes to ensure of their active status [38].

3. PROPOSED SYSTEM

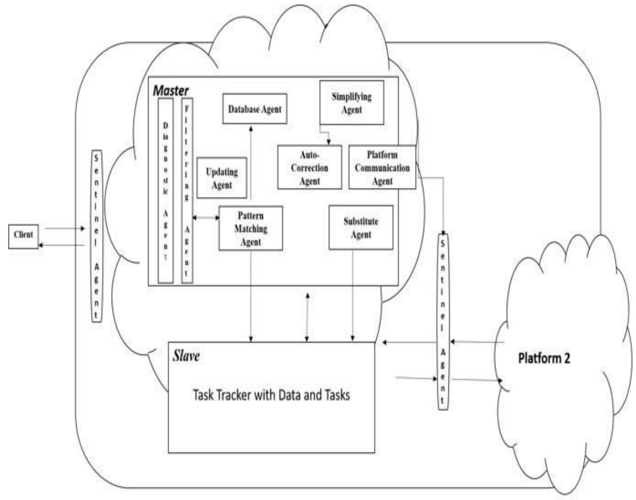

Figure-3. Proposed system architecture.

Use of the Hadoop framework with MapReduce, HDFS and the Agents in the Cloud has definitely reduced processing time. Data bank NCBI, DDBJ, EMBL are uploaded onto the cloud. The new system purports co-operating multi-agents for the purpose of

- Auto-correction of input query
- Identification and elimination of the word redundancy
- Query breakdown into simpler forms
- Identification of malicious query
- Storage of malicious query for future references
- Identification of IP address that sends malicious query and quarantine such addresses after suitable warnings
- Store malicious IP addresses for future references
- Stowage of recurrent enquiries
- Recent query match with those in the stockpile
- Provide security during query transfer from the biotechnological databank to the cloud
- Provide security to the data stored within the cloud
- Provide security during access between two cloud platforms
- Provide security between virtual machines

Auto-Correction Agent, Simplifying Agent, Filtering Agent, Diagnostic Agent, Pattern-Matching Agent, Sentinel Agent, Database Agent, Platform Communication Agent. These are the Agents that are to be included in the new system.

Auto-correction agent - This agent checks for simple syntax errors in the query input by the user and auto corrects the erroneous SQL terms with appropriate and standard vocabulary.

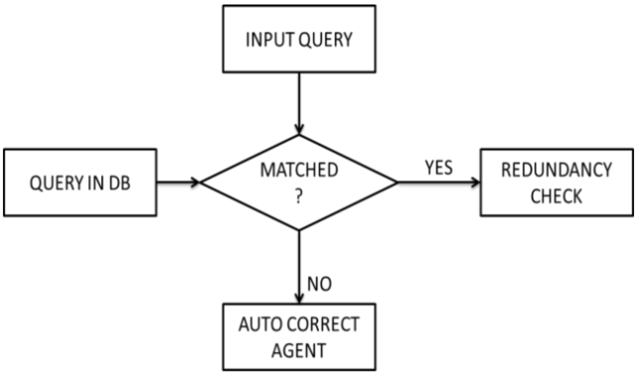

Figure-4. Flowchart of auto-correction agent.

Simplifying agent - Queries can be simple and nested. The Simplifying Agent breaks up the complex query into simple queries. Whenever a redundancy within a query is identified, This Agent rewrites the query after eliminating the redundant parts.

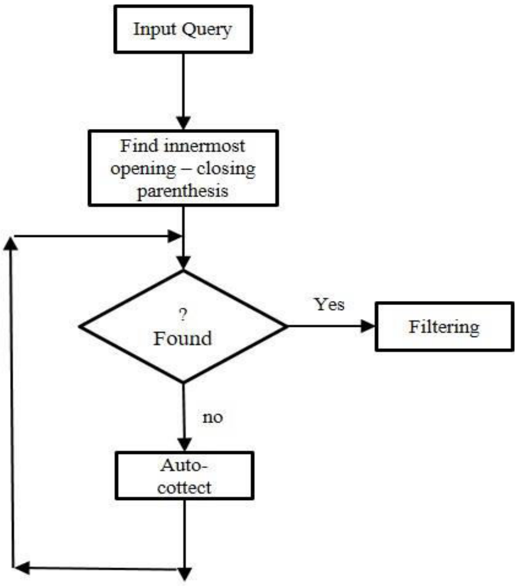

Fig. 5 Flowchart of Simplifying Agent

Filtering agent - This agent inspects previously quarantined IP addresses and prevents their admission into the cloud. It also examines detained requests so that such malicious queries maybe eliminated immediately.

Diagnostic agent - The Diagnostic Agent examines the queries for superfluous, malevolent appearances. Such queries are quarantined, their IP addresses identified and deposited for upcoming allusions.

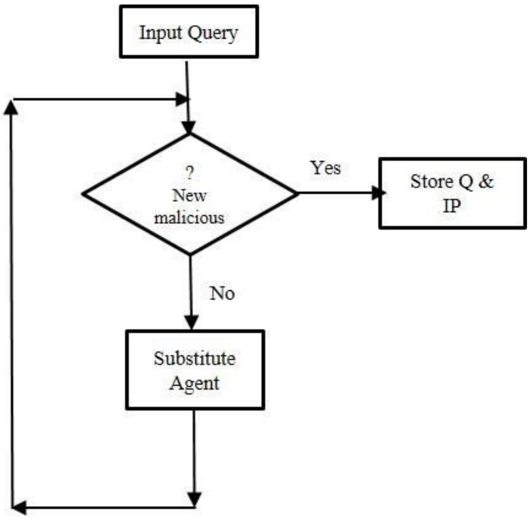

Fig. 6 Flowchart of Diagnostic Agent

Pattern matching agent - When a query is encountered for the first time, it is scheduled to the task node. But when the query recurs, the pattern match of the query is done with those stored in the database. If the current query matches with existing ones, then instead of processing again, the previous results are retrieved from the database. Query matching also differentiates between the safe query and the unsafe queries.

Database agent - A query that comes in for the first time is deposited instantly into the database. However, when it reappears, it is not stored. But the result for the previous such query search is given. This Agent helps in manipulation of data in the database and maintains an execution plan.

Substitute agent - This Agent creates back up plan in case of unexpected occurrences. It also maintains the current status of each node, possesses a list of all failed nodes, keeps a record of all task miscarriages and retains the location of backup data.

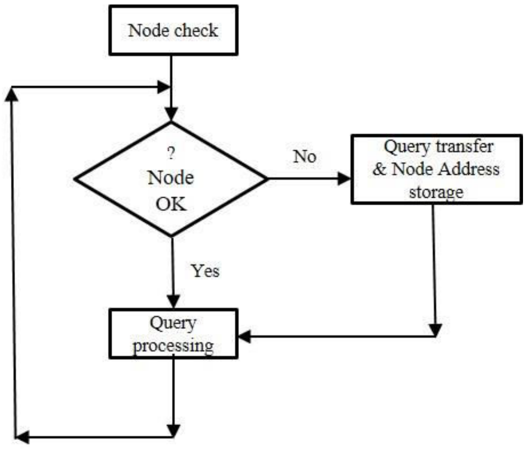

Fig. 7 Flowchart of Substitute Agent

Sentinel agent - These agents will be placed between virtual machines within the same cloud, between platforms and also at the storage location on the cloud. They continuously monitor the cloud for malicious endeavours and strikes. And when the strikes do occur, they take remedial actions to circumvent the attacks.

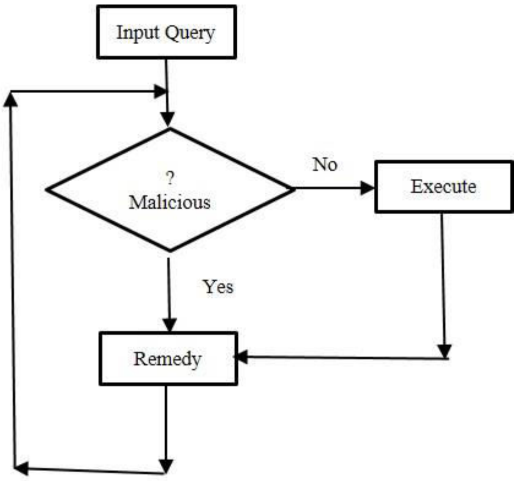

Fig.8 : Flowchart of Sentinel Agent

Platform communication agent - This agent obtains the IP address of the platform to which the query is to be forwarded in case of query transfer to another multi-agent on another platform.

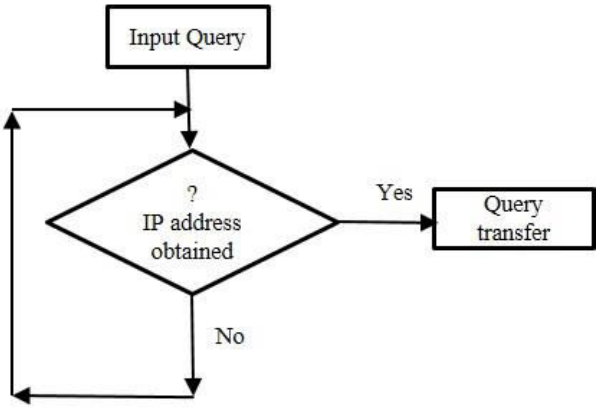

Fig.9 : Flowchart of Platform Communication Agent

Updating agent - This agent will ensure that the copy of the data in the cloud is updated such that integrity of the result is retained.

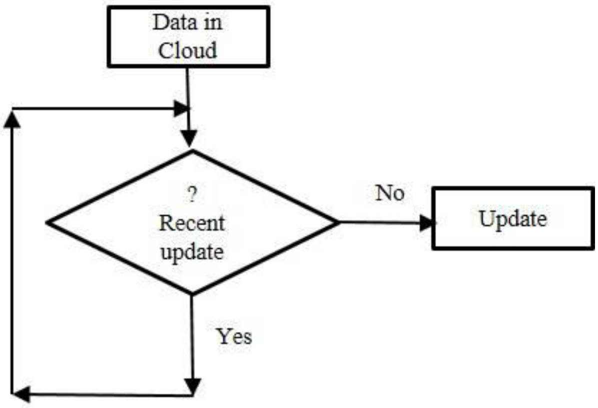

Fig.10 : Flowchart of Updating Agent

The Multi-Agent System is tested for defence susceptibilities by SQL injection attacks, where reprehensible SQL statements are inserted for execution. Invincibility tests are also done on the system to identify where Metasploit Framework Security tool of the Kali Linux platform, is used to examine security vulnerabilities, perform penetration testing and IDS signature development.

4. RESULTS AND DISCUSSIONS

The Jade.Jar is imported in netbeans. These files will be used to take advantage of intelligent agents. These will be then extended in the normal java file and its methods and attributes will be used as such. The agents that are created will be under main container. This will act as the dummy and they will show all the node names and its IP address on which it is running.

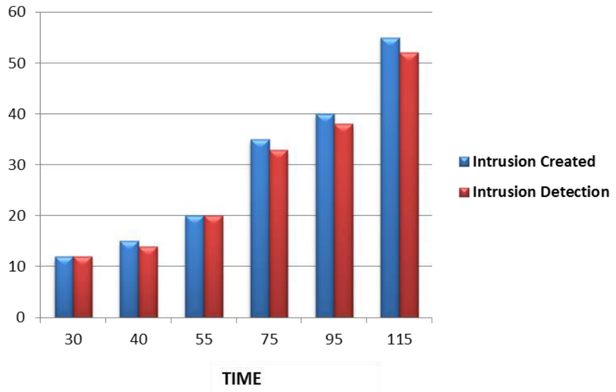

Figure-11. Comparison of intrusion creation and intrusion detection vs time.

The query which is entered as input will be either executed or rejected based on its security. More number of queries will be inserted which contains few intrusions. All the intrusions are identified efficiently. In Figure-4, the comparison between Intrusion Creation and detection versus time.

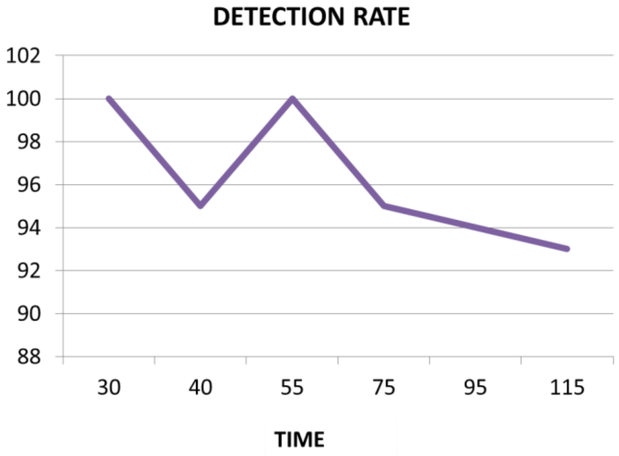

Figure-12. Comparison of detection rate vs time.

With the results in the Figure-4, the percentage of the detection rate is found and shown clearly in Figure-5. The network density i.e. the number of queries are also important case of denial of services. If more number of queries enter then the number of queries that are denied wil also increases as shown in the Figure-6.

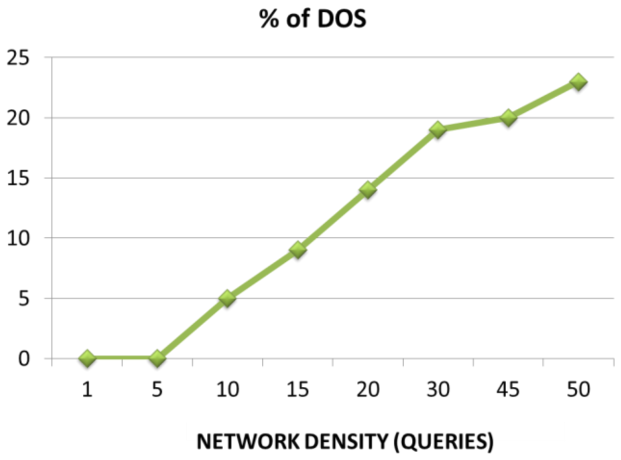

Figure-13. Comparison of network density VS % of DOS.

5. CONCLUSIONS

Hadoop in the cloud is a low investment, which can be used to access and store considerable volume of data resources and perform batch query processes in the vicinity of the data. Introduction of multi-agents has enhanced potency and scalability, which means that appropriate distribution of power and accountability ensures that the task is rescheduled to other task nodes instead of bringing the Job Tracker to a standstill. Being decentralised systems, they do not suffer from the "single-point of failure" issue faced by the Job-Tracker in Hadoop. The modularity of multi-agent systems warrant addition of future agents to improve competence. They have increased the processing speed, thereby improving the pace of query handling and allocation of tasks to the Task Trackers. Advanced fault tolerance and improved reliability are other factors that have been propelled. Security has been introduced between user and Agents. The improvisation of the existing system is based on overall improvement in the speed of performance.

REFERENCES

[1] H. Yang and M. Tate. 2009. Where Are We at Cloud Computing? A Descriptive Literature Review. 20th Australasian Conference on Information Systems, Melbourne.

[2] P. Mell and G. Timothy. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology, U.S. Department of Commerce. Special Publication800-145.

[3] L.M. Vaquero, L. Rodero-Merino and J. Caceres.2009. A Break in the Clouds: Towards a Cloud Definition. ACM SIGCOMM Computer Communication Review, 39(1). Electronic Magazine, article available at https://www.semanticscholar.org/paper/A-break-in-the-clouds-towards-a-cloud-definition-Gonzalez-Rodero-Merino/2b9b4a646c5b86b0fbf0e18cd3bd0f52e06fa980/pdf.

[4] G. Johnu *et al*. 2014. Hadoop MapReduce for Mobile Clouds. IEEE Transactions on Cloud Computing. 3(1).

[5] A.T.H. Ibrahim, Y. Ibrar, B.A. Nor, M. Salimah, G. Abdullah and U.K. Samee. 2015. The Rise Of Big Data On Cloud Computing: Review And Open Research Issues. Information Systems. 47: 98-115, Elsevier.

[6] R.Rajiv *et al*. 2015. Recent Advances in Autonomic Provisioning of Big Data Applications on Clouds. IEEE Transactions On Cloud Computing. 3(2).

[7] A. Zakia, A.R.C. Mohammad and David Malone. 2016. Greener Data Exchange in the Cloud: A Coding based Optimization for Big Data Processing. IEEE Journal on Selected Areas in Communications. 34(5).

[8] L. Wang, J. Taoc, R. Ranjand , H. Martenc, A. Streit, J. Chene and D. Chena. 2013. G-Hadoop: MapreduceAcross Distributed Data Centers For Data-Intensive Computing. In International Journal Future Generation Computer Systems, Grid Computing and eScience. 29(3): 739-750.

[9] Q. Zhang, R. David, M. F. Zhani, Y. Yang and R. Boutaba. PRISM: Fine-Grained Resource-Aware Scheduling for MapReduce. 3(2).

[10] B. Daniella, V. Honoriu, D. Ancuta, B. Rares. 2012. Agent Based Patient-Scheduling System. IEEE.

[11] R. C. Taylor. 2010. An Overview of the Hadoop/MapReduce/Hbase framework and its Current Applications in Bioinformatics. Proceedings from the 11th Annual Bioinformatics Open Source Conference (BOSC) 2010, Boston, MA, USA, 9-10 July 2010. Open Access. http://www.biomedcentral.com/1471-2105/11/512/S1.

[12] [12] H. Kashyap, H. A. Ahmed, N. Hoque, S. Roy and K.B. Dhruba. 2016. Big Data Analytics in Bioinformatics: architectures, techniques, tools and issues. Springer online publication.

[13] B. Feng, X. Ma *et al*. 2016. An Efficient Protocol with Bidirectional Verification for storage security in Clouds. Accepted for publication IEEE.

[14] D.A.B. Fernandes, L. F.B. Soares *et al*. 2013. Security Issues in Cloud Environmenrt: a Survey. Special Issue paper, Springer, 28 September 2013.

[15] E. B. Fernandez, R. Monge and K. Hashizume. 2015. Building Security Reference Architecture for Clouds. Springer.

[16] G. Zhao, C. Rong *et al*. 2010. Reference Deployment Models for eliminating user concerns on cloud security. Springer Science +Business media, J. Supercomput, 17 June 2010.

[17] K. Beckers, I. Côté *et al.* 2013. A Pattern-based Method for establishing a Cloud-Specific Information Security Management System. Requirements Engineering for Security, Privacy and Services in Cloud environments, Springer.

[18] P. Larsen, S. Brunthaler and M. Franz.2014. Security through Diversity: Ae we there yet? Copublished by IEEE Computer and Reliability Societies. Vol. 14.

[19] M.H. F. Zarandi, F. K. Azad. 2013. A type2 fuzzy multi-agent based system for scheduling of steel production. IEEE.

[20] [20] K. Kravari, N. Bassiliades. 2015. A Survey of Agent Platforms. Journal of Artificial Societies and Social Simulation. 18(1): 11, http://jasss.soc.surrey.ac.uk/18/1/11. html DOI: 10.18564 /jasss.2661.

[21] M. Al-Ayyoub, Y. Jararweh, D. Mustafa and A. Qutaibah. "Multi-Agent Based Dynamic resource prvisioning and Monitoring for Cloud Computing Systems Infrastructure". Springer online publication 24 March 2015.

[22] [22] M. Owliya, M. Saadat, G. G. Jules, M. Goharian and R. Anane. "Agent-Based Interation Protocols and Topologies for Manufacturing Task Allocation". IEEE Transactions On Systems, Man and Cybernetics Systems. Vol. 40 No. 1, January 2013.

[23] [23] P. M. Papazoglou, D.A. Karras, R.C. Papademetriou. 2011. On Multi-Agent Based Modelling and Control in Large Scale Wireless Communication Systems for Improved Resource Allocation Performance. Springer published online.

[24] R. Olfati-Saber, J. Alex Fax and R. M. Murray. 2007. Consenses and Co-operation in Networked Multi-Agent Systems. Proceedings of the IEEE. 95(1): 215-233.

[25] A. Sara and M. Nasrollah. 2012. A Bilateral Negotiation Strategy for Grid Scheduling. 6th International Symposium on Telecommunications. IEEE.

[26] S.W. Scott, W.B. Robert and H.N. Douglas. 2005. Holonic Job Shop Scheduling Using a Multi-Agent System. Published by the IEEE Computer Society Volume 5. IEEE Intelligent Systems.

[27] S. Tharunya, A. Manuel and K.L. Shunmuganathan. 2016. A Multi-Agent Based Intelligent Query Processing System for Hadoop with Foundation for Intelligent Physical Agents-OS Using Cooperating Agent in cloud Environment. Journal of Computational and Theoretical Nanoscience. 13: 1-6.

[28] T. Y. Ma and G. Phillippe. 2013. A Hybrid Algorithm For Fast Learning Individual Daily Activity Plans For Multi-Agent Transportation Simulation. IEEE Conference and Tecnologies and Applications of Artificial Intelligence.

[29] W. Van Der and M. Woolridge. 2007. Multi-Agent Systems. Handbook of Knowledge Representation. Elsevier B.V.

[30] Z. Guessoum. 2004. Adaptive Agents and Multi-Agent Systems. IEEE Computer Society. 5(7).

[31] S. Poslad, P. Buckle and R. Hadingham. The FIPA OS Agent Platform: Open Source for Open Standards, http://fipa-os.sourceforge.net/ docs/ papers/FIPAOS.pdf.

[32] Z. Jing, H.Xiangmei and M.Xiaofeng. ESQP: An Efficient SQL Query Processing for Cloud Data Management.

[33] Foundation for Intelligent Physical Agents FIPA Abstract Architecture Specification. 1996-2002 Foundation for Intelligent Physical Agents, Geneva, Switzerland, http://www.fipa.org/, 11.5.

[34] O. Marin, P. Sens, J. Briot and Z.a Guessoum. 2001. Towards Adaptive Fault Tolerance for Distributed Multi-Agent Systems. Proceedings ERSADS, www-apa.lip6.fr.

[35] J. Zhao, Xiangmei, X. Meng. 2010. ESQP: An Efficient SQL Query Processing for Cloud Data Management. Cloud DB 10,October 2010.cmv

[36] M. F. Husain, J. McGlothlin, L. Khan, B. Thuraisingham. 20111. Scalable Complex Query Processing Over Large Semantic Web Data Using Cloud. IEEE 4th International Conference on Cloud Computing.

[37] K. K. Kumar, T.M. SanthiSri, V. Vamshipriya. 2015. Introduction to Techniques of Query Processing and Optimization. International Journal of Innovative Research in Advanced Engineering IJIRAE), 2(3). ISSN: 2349-2163; www.ijirae.com.

[38] M. Divya, K.L. Shunmuganathan. 2016. A Role of Multi-Agents in Intelligent Query Processing: Deploying Co-operative Agent using FIPA - OS in Cloud. IEEE.

[39] C. J. Chae, K.N. Choi, K. Choi. 2015. Information Interoperability System Using Multi-Agent with Security. Wireless Pers Communication, Springer.

[40] C. Dou, D. Yue, J. M. Guerrero. 2016. Multi-Agent System based Event-triggered Hybrid Controls for High Security Hybrid Energy Generation Systems. Accepted for publication. IEEE Transactions on Industrial Informatics.

[41] S. Bijani, D. Robertson. 2014. A Review of Attacks and Security Approaches in Open Multi-Agent Systems. 42(4): 607-636.