International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-5, March 2019

Master Slave Scheduling Architecture for Data processing on Internet of things

B.Mallikarjuna, Anand Dohare, Mhd. Shahajad, Tulika

Abstract: ***In this paper proposed master slave scheduling architecture, the data transferred to the local nodes without transferring data to the cloud or high end servers. There is a need of localized map reduce framework for IoT. The mobile agents have the ability to migrate the tasks from one node to another node for data processing, Mapper and Reducer are two mobile agents are used master side, slave mobile agents are used to map the data to IoT sensor nodes for transferring data to the local nodes, without transferring data to the high end servers. The mobile agents migrates the tasks from one node to another node. The proposed framework proved that effective results using the cost function.***

Index Terms: ***Master Slave scheduling MapReduce, Mobile Agents, Internet of Things, Data processing Introduction***

I. INTRODUCTION

Scheduling in IoT is a complex task, in data processing on IoT sensors many resources are contributed [1]. In the recent trends, Internet of Things (IoT) is a primary research field [1,2]. The IoT is a paradigm which connects every object is connected to the internet, observation of data processing, there is huge amount of data is processing in terms of zetta bytes on sensor nodes[3]. The data analysis is efficient important aspect for efficient IoT management [1,2]. The emerging technologies Big data, Cloud computing , Fog computing processing the data and huge volumes of data is generated by the sensor nodes in the IoT [4]. In IoT has to improve the network connectivity to improve the data processing and network bandwidth, data transfer the network should be minimal it required efficient scheduling mechanism [5]. There is a challenging task data processing to the local nodes without processing the data to the high servers like cloud [6]. To address the data processing many company's developed efficient architectures [6], Google developed efficient scheduling mechanism for data processing for Map-Reduce model, In IoT, data processing is treated as the complex task, many researches contributed number of computation methods and some models are good enough to solve the issue as much as possible. To address the data processing [16], Google developed MapReduce model which is developed with an idea of map and reduce functions in the LISP programming language. Yahoo developed Hadoop platform which actually uses the MapReduce function for processing the large volumes of datasets [3-4]. The MapReduce framework duplicates the data to the high end servers or to the cloud platform before the data processing is initialized. This process requires huge data storage and it is more cost effectively. The data transmission from the sensor nodes to the high end server requires more cost [5].The idea behind that Map and Reduce are two major functions to process the huge volume of datasets the environment developed on LISP programming yahoo developed the same architecture on Hadoop platform [6,7].

To solve the aforementioned issue, there is a necessity to develop master slave scheduling architecture for localized Map-Reduce, it processe data to the sensor nodes in IoT environment [6]. The main aim of the proposed master slave scheduling architecture for local mobile agent framework is to implement the Map-Reduce at the sensor nodes rather than transmitting data at the high end servers or cloud and aggregate the resultant framework. The rest of the paper section 2 describe the related work, section 3 explains the proposed architecture of master slave scheduling architecture, section 4 deals with results and discussions and finally conclusion is drawn in section 5.

II. RELATED WORK

The emerging technologies bigdata, cloud computing, fog computing made huge oportunities to the business enviornment for solvind the data needs, the HADOOP environment to develop MapReduce is a new approach for data processing . In MapReduce faremw rok devides the data into different small parts and assign them in to the worker nodes, there are number of worker nodes consist of environment . The aggrigate results are collected from the worker nodes . The HDFS has own file system to implement MapReduce framework, Hadoop platform uploads the the data to process HDFS file system. This process requires more bandwidth and consumes high cost The prposed architecture , the data processing is done at the local mobile agents by deploying the code for processing the data with the utilization of localized mobile agents . The study [6] proposed the Hadoop framework consist of HDFS but it is not suitable for the high end servers. There are several studies of MapReduce framework, MATE [8] and Phoenix [9] are proposed shared memory CPUs architecture that contains MapReduce Access Patterns (MRAP) [9]. Many commercial service providers are using the MapReduce process by implementing their own file system such as Hadoop file system (HDFS), shared memory from the CPUs and Google File System (GFS).

Manuscript published on 30 March 2019.
*Correspondence Author(s)
B. Mallikarjuna, School of Computer Science & Engineering, Galgotias University, Greater Noida, Uttar Pradeshg , India.
Anand Dohare, Research Scholar, Computer Science & Engineering, SHUATS University, Allahabad, India.
Mhd. Shahajad, Computer Science & Engineering, Galgotias University, Greater Noida, Uttar Pradeshg , India.
Tulika, Computer Science & Engineering, SHUATS university, Allahabad, India.

For instance, Hadoop platform uploads the data from the current environment to the HDFS file system before processing it.

This process requires more bandwidth and consumes high cost. In our approach, the data processing is done at the local nodes by deploying the code for processing the data with the utilization of localized data. In [6], the author presented the Hadoop framework for embedded computers. The Hadoop file system is not suitable for the high end servers. Therefore, the model doesn't serve good. Twister [10,11] and Hadoop [6] proposed difeerent models are store the data they are used for temporary files instead of using the Key Value Storage (KVS). The MISCO framework [12] proposed for data processing in mobile devices using HTTP. These models store the data in temporary files instead of using the key value storage (KVS).In [12], the Misco framework was developed for data processing in mobile devices using HTTP.

The authors in [12] proposed a Map-Reduce framework for heterogeneous devices. The study [10,11,12] map reduce framework for homogeneous devices and heterogeneous devices

III. MASTER SLAVE SCHEDULING ARCHITECTURE

In IoT environment is entirely different from Hadoop environment, the Map-Reduce process is implemented in Hadoop environment, In IoT environment consist of sensor nodes, processing the data and aggregate the data at the sonsor nodes, The following requirements have been made in IoT environment as shown in Figure 1.Mobile agents are process the tasks and makes effective scheduling on IoT sensor nodes. There are two types of mobile agents in the proposed framework, mapper agent and reducer agent. The mapper agent consists of master agent and slave agent. All slave agent having the responsible for transferring the data to the IoT sensor nodes and limited bandwidth.

In the proposed framework, mobile agents are introduced to achieve the requirements presented in the previous section. The functionality of mobile agents is to carry the tasks and process the tasks at sensor nodes and finally to obtain the results from the sensor nodes. In the original MapReduce model, we have mapper and reducer classes to support the data processing task. In the same way, the proposed framework also has the mapper and reducer. The mobile agents have many advantages compared to the previous ones. The mobile agents can select one or more nodes according to the result and performs migration. This type of practice is very useful in unstable network. The mobile agent does not follow the centralized approach and they operated in decentralized manner. This is crucial to process the data stored in the sensor nodes to maintain scalability. The mobile agents can store and retrieve the data from the heap memory of the senor nodes. Therefore, it is easy for the developers to use the data without knowing the file system like Hadoop.

IoT Sensor Nodes: In IoT environment sensor nodes low memory when compared to the high end servers. In IoT environment does not provision of file system hence data should processed local nodes with limited bandwidth. *Limited Bandwidth:* All sensor nodes in IoT environment is mapped with limited bandwidth and all sensor nodes ability to process the data.

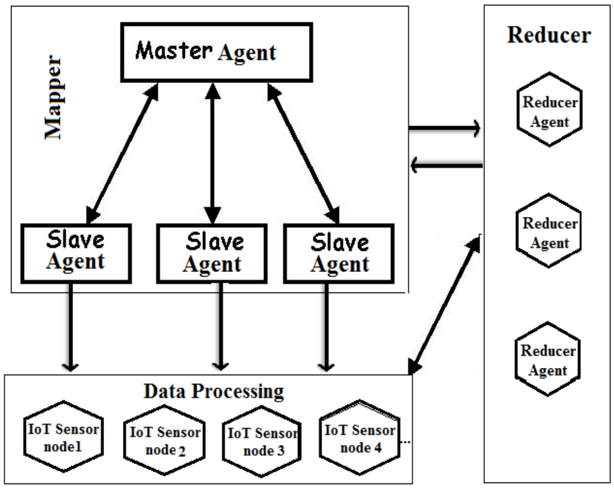

Figure 1: Master Slave Scheduling Architecture

The major advantage of Master slave scheduling architecture is parllel execution and data processing at IoT sensor nodes. The following Master slave architecture is parllel execution, it process the data all nodes parllelly and execute the data each node at separately. IoT senosr nodes process the data to all nodes without transfering data to high end servers like cloud.

Algorithm 1: Mster Slave Scheduling Algorithm

Input: Mapper Agent, Slave Agent

Output: Aggregated resultant data

Begin

Step1: The master node contains Mapper agent and slave, the mapper agent contains many number of salve agents.

Step 2: The slave agent allocates the tasks to sensor nodes, the salve agent migrate one master node one or more sensor nodes

Step 3: The slave agents computes the data gathered by the sensor nodes.

Step 4: After the data computation is completed, the slave agents migrates to the reducer node where the final result has to be aggregates.

Step 5: The slave agents utilizes the inter agent Communication for sending the results to the reducer.

End

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The maste slave scheduling architectue having many advantages when compared with the Hadoop architecture, The mobile agenets can select one or more IoT sensor nodes for allocating tasks for data processing . The proposed approach contains long term benefits for huge data transmission while

IV. RESULTS AND DISCUSSIONS

The proposed environment is tested by using the JDK 1.8 and the framework environment GUI was designed by the mobile agents, the framework is evaluated by mobile agent duplication cost function, the environemnt was created by 8 nodes has been implimented in dual core processor by using Giga Ethernet. The Figure 2 shows the cost function for agent duplication, compare the agent size and cost for agent duplication it is measured in terms of milliseconds, there are two callback methods are invoked for agent migration.

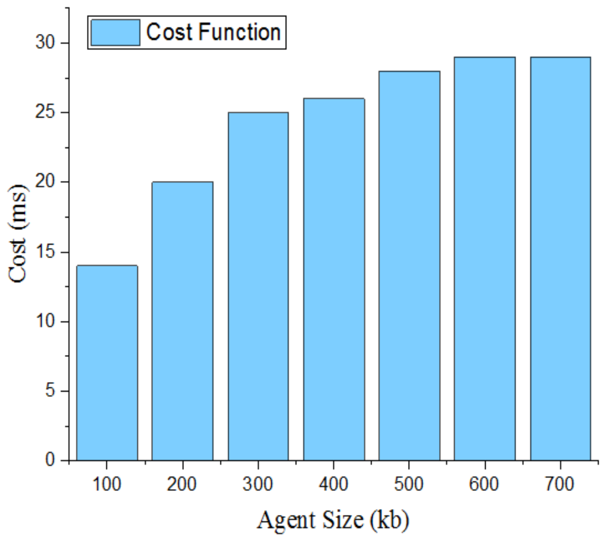

Figure 2: Aggeregate csot function

The following Figure 3 describes cost migration between two nodes, the migration cost involves the process of preparing the agents, opening the TCP connection, mobile agents are migrating from one node to another node , unpack the the agents for security verification. While mobile agents seize is increasing therefor cost function is also increasing.

The proposed framework is tested is tested with the Hadoop model, the Hadoop model is tested with dual core processor it contains 2 GB RAM and it copies the data 34 MB/S, the data transferred 2.0 GB. The Hadoop environment is created by 8 mobile agents , the data transmission is recorded for total cost 100 seconds on sensor nodes to the HDFS. The proposed environment the data need to be transffrred nodes are reduced 15 MB.

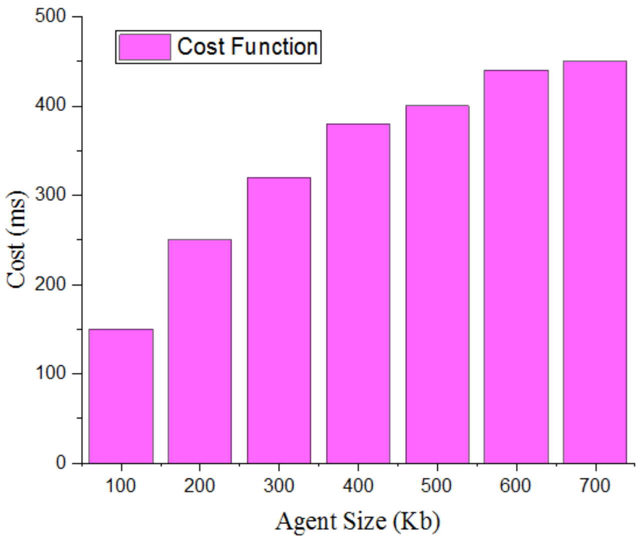

Figure 3: Agent Migration versus cost function

The proposed framework is tested is tested with the Hadoop model, the Hadoop model is tested with dual core processor it contains 2 GB RAM and it copies the data 34 MB/S, the data transferred 2.0 GB. The Hadoop environment is created by 8 mobile agents , the data transmission is recorded for total cost 100 seconds on sensor nodes to the HDFS. The proposed environment the data need to be transffrred nodes are reduced 15 MB. The proposed model is effeceint for IoT to avoid unnecessary data processing.

V. CONCLUSION

This paper proposed the effective scheduling mechanism on MapReduce framework for IoT. The proposed model is unique in their nature because of performing the data processing at the nodes instead of transferring the data to the cloud or high end servers. The traditional MapReduce model is enhanced with the mobile agents. The mobile agents have the ability to migrate from one node to another node for data processing, Mapper and Reducer are used for agent duplication and result aggregation. The proposed framework is useful for the developers to define application with any knowledge of mobile agents. In the future, this model is enhanced with scheduling mechanisms for distributing the tasks for sensor nodes.

REFERENCES

1. L. Atzori, A. Iera, and G. Morabito, "The Internet ofThings: A Survey," Comput.Netw., vol. 54, no. 15,pp. 2787–2805, Oct. 2010.
2. J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs,C. Roxburgh, and A. H. Byers, "Big data: The Next Frontierfor Innovation, Competition, and Productivity," McKinseyGlobal Institute, May 2011.
3. Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst: "HaLoop: Efficient Iterative Data Processing on Large Clusters," Proceedingsof the VLDB Endowment, Vol. 3, No. 1, 2010.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

4. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli: "Fog computingand its role in the internet of things", in Proceedings ofMCC workshop on Mobile Cloud Computing (MCC '12), pp.13-16, ACM Press, 2012.
5. B.Mallikarjuna, P.Venkata Krishna " OLB: Nature Inspired Approach for Load Balancing of Tasks in Cloud computing" in Cybernetics Information Technology, Vol. 15, No. 4, PP.138-148.
6. J. Dean and S. Ghemawat: "MapReduce: simplified data processingon large clusters," in Proceedings of the 6th conferenceon Symposium on Opearting Systems Design andImplementation(OSDI'04), 2004.
7. Y. Jung, R. Neill, L. P. Carloni: "A broadband embeddedcomputing system for MapReduce utilizing Hadoop", in Proceedingsof IEEE 4th International Conference on Cloud ComputingTechnology and Science (CloudCom), pp.1-9, 2012.
8. W. Jiang, V.T. Ravi, G. Agrawal: "A Map-Reduce System withan Alternate API for Multi-Core Environments," in Proceedingsof 10th IEEE/ACM International Symposium on Cluster,Cloud, and Grid Computing, 2010.
9. J. Talbot, R. M. Yoo, and C. Kozyrakis: "Phoenix++: modularMapReduce for shared-memory systems," in Proceedings of2nd international workshop on MapReduce and its applications(MapReduce '11). ACM Press, 2011.
10. S. Sehrish, G. Mackey, J. Wang, and J. Bent, "MRAP: ANovel MapReduce-based Framework to Support HPC AnalyticsApplications with Access Patterns," in Proceedings of HighPerformance Distribute Computing (HPDC 2010), 2010.
11. J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.H. Bae, J.Qiu, and G. Fox: "Twister: a runtime for iterative MapReduce,"in Proceedings of the 19th ACM International Symposium onHigh Performance Distributed Computing (HPDC '10). ACM,2010.
12. A. Dou, D. Gunopulos, V. Kalogeraki, T. Mielikainen, andV. H. Tuulos: "Misco: a MapReduce framework for mobilesystems," in Proceedings of 3rd International Conference onPervasive Techechnology Related to Assistive Environments,pp.1-32, Jun. 2010.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)