WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM

e-ISSN: 2788-5879 p-ISSN: 2788-5879

IoT sensor network data processing using the TWLGA Scheduling Algorithm and the Hadoop Cloud Platform

Mohanad Hameed Rashid[1,*] and Wisam Mohammed Abed[2]

[1]Computer Science Quality Department,Anbar Education Directorate, Ramadi, Iraq

[2]Computer Science Preparation and Training Department, Anbar Education Directorate, Ramadi, Iraq

*Corresponding Author: Mohanad Hameed Rashid

DOI: https://doi.org/10.31185/wjcm.122

Received: December 2022; Accepted: February 2023; Available online: March 2023

ABSTRACT: Monitoring environmental conditions can be done effectively with the help of the Internet of Things (IOT) sensor network. Massive data generated by IOT sensor networks presents technological hurdles in terms of storage, processing, and querying. A Hadoop cloud platform is suggested as a fix for the issue. The data processing platform makes it possible for one node's work to be shared with others employing the time and workload genetic algorithm (TWLGA), which lowers the risk of software and hardware compatibility while simultaneously increasing the efficiency of a single node. For the experiment, a Hadoop cluster platform employing the TWLGA scheduling algorithm is built, and its performance is assessed. The outcomes demonstrate that processing huge volumes of data from the IOT sensor network is acceptable for the Hadoop cloud platform .

Keywords: IOT, Sensor network, Hadoop, Cloud computing

1. INTRODUCTION

Cloud computing offers a useful means of processing resources. which breaks the network's whole computing operation into several smaller processes and distributes them across numerous available servers. The findings are provided to the user after computation and analysis. Massive data storage, rapid data analysis, and real-time processing are all features of cloud computing [1, 2]. Using a cluster of several inexpensive pieces of hardware called nodes, the distributed computing framework Hadoop can execute programs. For the purposes of monitoring air pollution, locating faults, and disaster management, Huge volumes of data are gathered, processed, and stored using a Hadoop-based architecture [3, 4].The temperature data is continually gathered, saved, and aggregated by the computers in this paper's sensor network, which consists of both fiber Bragg grating sensors and conventional sensors. The temperature data is then packaged into various sizes. Real-time sensor network data handling in the cloud has been enhanced. The investigation of a computing-based sensor network data processing platform The effectiveness of TWLGA and the Hadoop cluster platform are also being researched. and the Hadoop cluster platform are also being researched.

2. METHODOLOGY

2.1 IOT SENSOR NETWORK BASED ON THE CLOUD

A node in an IoT sensor network that is gathering and processing massive volumes of data in real time will become too busy and unable to respond in time. To prevent paralysis, data is spread from one busy node to other idle nodes utilizing cloud computing technologies. Together with one controlled node, these nodes share resources. Data processing

is therefore spread out among a large number of nodes rather than being focused on a small number of active nodes. The managed node receives the whole set of computed results. Because of this, cloud computing technology not only improves the nodes' capacity to process data in real-time but also guarantees the speedy gathering and analysis of enormous volumes of data. Cloud computing technology lessens the danger of heavy workloads since instead of using active nodes to calculate the resources, several idle nodes are used. As a result, The IOT sensor network that uses the cloud is more flexible, dependable, and controlled. The theoretical cloud computing paradigm is continuously applied to the data processing of IoT sensor networks.

The crucial technological concept Fig 1 depicts an IOT sensor network based on the cloud .The IOT sensor network shown in Figure 1 is made up of multiple smaller networks, each of which has a number of sensors. The sub networks' sensing components keep an eye on parameter changes. The convergent nodes are interconnected, and each convergent node is connected to several sub networks. The cloud computing technology connects all of these nodes to the management nodes. Instead of using busy nodes to determine the data resources, a sequence of idle nodes is used, based on input from the managed node. The control center issues orders via the internet to the monitored nodes. The IOT sensor network's analysis findings are shown on the display interface.

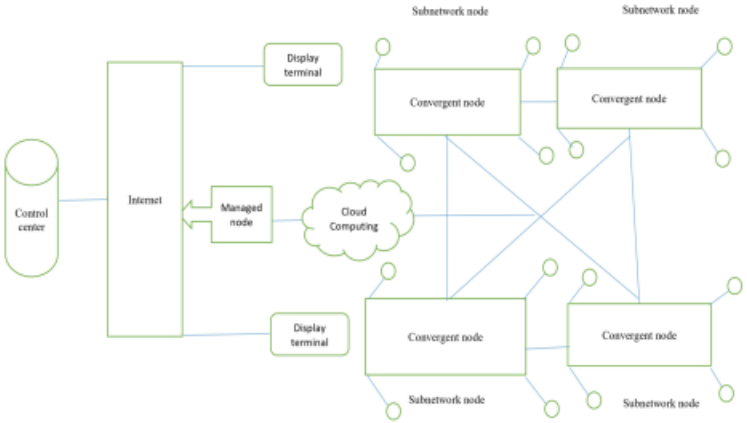

FIGURE 1. The cloud-based Internet of Things sensor network

2.2 THE GENETIC ALGORITHM FOR TIME AND WORKLOAD

The scheduler completes the precise job assignments on the hardware and software processors that make up a Hadoop cluster. Hadoop's default scheduler is First in, First out, or FIFO. The FIFO queue-form scheduling method is very straightforward and unable to satisfy more sophisticated demands. The machines in the experiment use Hadoop to build a cluster. Various setups, levels of processing power, and workloads mean that certain jobs will be incorrectly allocated to nodes with high workloads. TWLGA, which takes into account the twin restrictions of time and workload, is presented as a solution to this issue and allows Job completion and logical homework [5, 6]. Chromosome coding [7], node workload [8], fitness function [9], crossover, and variation [10] operations are scheduled by the TWLGA.

2.3 CODING ON CHROMOSOMES

In data processing, resource-intensive tasks are indirectly encoded. Once the number of task slices and the resource number of the matching task slices have been established, the chromosomal coding is complete.After the chromosomes have been decoded, the task-resource tables are obtained. Third, the amount of time it will take for each resource to perform all of the tasks assigned to it may be calculated using the expected time to complete (ETC) matrix. The period of time needed for resources to finish each subtask that was given to them is

$$\text{EachResourceTime} = \sum_{j=1}^{n} \text{time}\,(i, j), i \in [1, m] \quad (1)$$

Where time (i, j) represents the duration and n the number of subtasks assigned to each resource needed for each resource to accomplish the jth work. Since all tasks are computed concurrently in cloud computing, the task's final completion time may be determined by the resource with the longest running duration.

$$\text{JopFinalTime}\,(J) = \text{MAX}(\,\text{EachResourceTime}\,(i)) \quad (2)$$

2.4 WORKLOAD OF NODES

Client computers, master nodes, and slave nodes are the three primary categories of computer roles in a Hadoop cluster. The client computer's job is to provide data to MapReduce and load it into the cluster. The two main functional components of the master node are its large data storage capacity and its ability to do concurrent calculations on the data. The bulk of computers are made up of slave nodes, which are responsible for doing all of the grunt work associated with data storage and computing. Each slave has a task tracker daemon running that interacts with the master node and follows its instructions. along with a data node that processes data. a variety of things. The burden of each slave node is affected by several things, such as CPU consumption, memory use, and network resources. Just four factors are often considered in a Hadoop cluster. Use the formula below to define the workload of the nodes.

$$\text{WorkLoud}\ (i) = w_{cpu} \times \mu_{cpu} + w_{me} \times \mu_{me} + w_{di\ sk} \times \mu_{di\ sk} + w_{nr} \times \mu_{nr} \tag{3}$$

Where w_{cpu}, w_{me}, w_{disk}, and w_{nr} stand for the contribution of each node's Overall performance is determined by the CPU, memory, storage, and network resources. The figures show how much of the available CPU, memory, disk, and network resources are being used. m_{cpu}, m_{me}, m_{disk}, and m_{nr}, and so forth. In the experiment, data from the sensor network is collected and stored by the master computer and slave computers, but certain slave nodes are overworked while others are largely inactive. The upshot is that the master computer divides the burden of one node among the others, By offering compatibility support, one node may function more effectively while reducing the chance of hardware and software failure.

2.5 FITNESS FUNCTION

The chromosomes' random coding is created through random initialization; specifically, Under the suppositions Suppose there are P chromosomes in total, each of which has a length of R, and that there are R computing resources, N sub-tasks, and P starting populations, and the number of genes are picked at random in [1,R]. As a consequence of the algorithm's effectiveness, the speed at which the task is completed, and the appropriateness of node allocation are all directly impacted by the fitness function selection. It is evident from the research above that JobFinalTime (J) is the fitness function based on GA that takes into account running time.

$$P_{\text{time}} = \frac{1}{\text{JopfinalTime (J)}} \tag{4}$$

However, because just the task running time is taken into consideration in the fitness function and not the workload component, the fitness function is improved.

$$\text{Optimal (J)} = \frac{1}{\text{Jopfinaltime}\ (J) \times (1 + \ \text{WorkLoad}\ (i))} \tag{5}$$

By considering both workload and task scheduling, the issue of tasks being allocated to nodes with a high burden is eliminated.

2.6 CROSSING AND VARIATION

As the fundamental building block of the whole algorithm and the factor determining the viability of the subsequent operations, the purpose of crossover is to produce distinct people via the crossover transformation of the genes. The capacity to search locally may be improved by the mutation operation, which can preserve and increase species variety. To accomplish adaptive adjustment, the TWLGA enhances the cross variation probability formula. The enhanced probability formula looks like this:

$$P_C = \frac{P_{C1} - P_{C2}}{P_{C1}} \times \frac{f_0 - f!}{f_0 - f^-} \tag{6}$$

$$P_m = \frac{P_{m1-}p_{m2}}{p_{m1}} \times f_0 - f'/f_0 - f^- \tag{7}$$

$\mathcal{F}$ Optimum is the population's maximum fitness, $\mathcal{F}$ is the crossover individual's maximum fitness, and f is the population's"average fitness" , where $\mathcal{F}$ is the mutant person's fitness. The enhancement of the P and P's formula for calculating the sum has resulted in a considerable boost in computation efficiency since it can now be successfully schedule tasks, execute them in less time overall, and take into consideration the burden of the nodes.

3. THE RESULTS OF AN EXPERIMENT AND A TEST

To test the functioning of a TWLGA-based cloud-based data processing system, the Hadoop cluster platform was developed. Figure 2 depicts the experimental configuration; there are five computers in the Hadoop cluster, one of which serves the first with one acting the primary computer, with the others serving as slave computers. Before installing Hadoop software, the names of five computers are altered, making the names of the five machines master. slave1, slave2, slave3 and slave4 separately in the catalog /etc/hostname of each computer. Then the hostname and the IP address are added to the configuration file in the catalog /etc/hosts, so each node computer can be recognized and accessed each other.

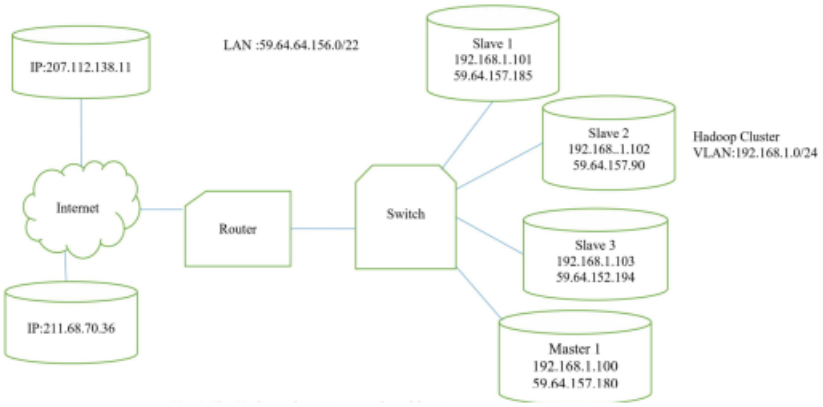

FIGURE 2. The Hadoop cluster's architecture

Each node's system of operations is Ubuntu 12.04.3, and the Hadoop software is version 1.0.2. The worksheets used to store the sensor data on the Hadoop cluster serve as Hbase's database. The cloud computing facility has Hbase installed with version 0.94.10 of the database. The master machine, which is deployed via the myeclipse program, has a web server called Tomcat that is utilized in the cluster. wavelength , year, month, day , and observation time (hour and minute). They are all represented in the fiber sensor source data format. in Table 1 So on. The temperature is calculated using information on wavelengths. There is a lot of information, and the structure is confusing. We just need to eliminate that piece of the data based on demand as Users are only interested in the year, month, time, and associated temperature. Since there are hundreds of data files, processing huge files is a benefit of Hadoop. Small files have very poor processing efficiency, so it is not recommended to handle them immediately. In the experiment, all files from the same year are combined using Hadoop's file merging feature. This creates a single huge file for every year, from which the valuable data is extracted using Map Reduce, which makes the most of Hadoop's capacity for handling enormous files.

Table 1. The format of the source data

Year	Month	Day	Hour	Minute	Wavelength
2021	04	02	09	31	154575
2021	04	02	09	31	154578
2021	04	02	09	32	154576
2021	04	02	09	32	154580
2021	.....	.....	.....	....	.....
2021	04	02	12	21	154594
2021	04	02	12	21	154594

3.1 HADOOP I/O PERFORMANCE EVALUATION

The Hadoop I/O performance test is essential since enhancing reaction times and the data processing system's top priority is processing power. Hadoop 1.0.2.jar is used to evaluate Hadoop's reading and writing I/O performance. The Hadoop Distributed File System's reading and writing performance is evaluated using the MapReduce task (HDFS). To gauge HDFS I/O performance, It reads and writes 10 512M files. The test results are shown in the table.

Table 2 shows that the Hadoop cluster has a reading speed of 114.21"Mb/sec", This is around 30 times more quickly than the rate of writing. 4.15 Mb/sec. Due to its suitability for one-time writing and multiple-time reading, the Hadoop cluster is generally utilized for reading activities.

Table 2. shows the results of an evaluation of HDFS I/O performance

(Symbol)	(Read)	(Write)
The number of files	10	10
processed Mbytes in total	5120	5120
Throughput(Mb/sec)	17.73	3.50
I/O rate on average (Mb/sec)	114.21	4.15
standard deviation for I/O rate	120.75	1.93
Test execution period (sec)	147.09	313.40

3.2 MAPREDUCE PERFORMANCE EVALUATION

For parallel and distributed applications with massive amounts of data, Map Reduce is a key programming paradigm. Additionally, Hadoop is a related open-source Map Reduce implementation. The five test files for the Map Reduce performance have a combined size of 160MB, 320MB, 640MB, 1.3GB, and 2.6GB. The experimental test results are shown in the table after the five files have been arranged and tallied on one, two, and three nodes, respectively.

Table 3. The Map Reduce performance test

Symbol	160Mb	320Mb	640Mb	1,3Gb	2,6Gb
(1)node sec	30	65	409	549	1214
(2)node sec	39	91	360	488	1055
(3)node sec	56	117	353	454	902

According to Table 3 the more nodes a tiny quantity of data contains, the slower the computation pace is. The multi-node system exhibits superiority as the volume of data grows. The time spent on network transmission is particularly crucial since when Since Map Reduce works with tiny data sets, each node's data must be read. Its usefulness for big data is demonstrated through processing enormous volumes of data.

3.3 PERFORMANCE TEST FOR HBASE

HBase's performance is evaluated using Yahoo's generic performance testing tool. When there are "1", "50", "100", "1000", or "5000" threads, The testing for writing time, data throughput, and reading time is all carried out separately. Figures 3, 4**,** and 5 exhibit the test findings. The HBase performance test when writing data is shown in Figures 3 and 4. As the number of threads rises, the writing time and data throughput alter in opposition to one another. The reading time for data in Fig 5 of the HBase performance test declines exponentially before leveling off as the number of threads rises. Whether writing or reading data, the throughput is between 2300 and 2900, and there hasn't been much of an overall difference regardless of whether there are more threads processing data or not. HBase can therefore continue to handle data quickly even when there is considerable concurrency.

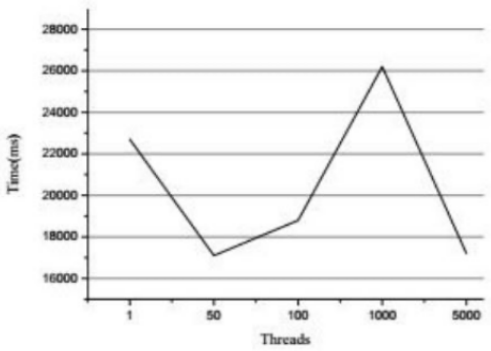

FIGURE 3. The writing time change with different threads

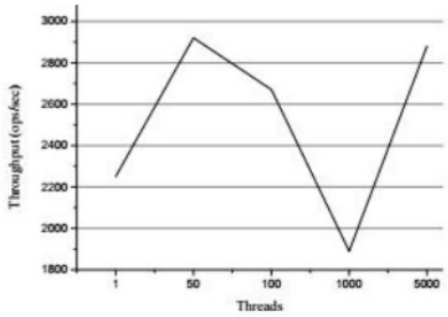

FIGURE 4. **The throughput change with different threads**

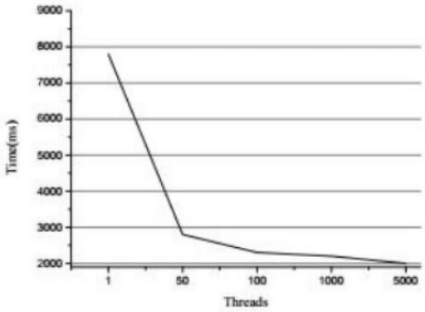

FIGURE 5. **The reading time change with different threads**

4. CONCLUSIONS

It is suggested in this study to process IOT sensor network data using the The TWLGA scheduling algorithm and the Hadoop cloud platform Huge volumes of data are processed using cloud computing technology with TWLGA, which also helps to avoid the IOT sensor network. from becoming paralyzed in order to increase platform performance. One node's duty is split across several. This increases a single node's effectiveness while lowering the network's potential risk. The performance of a Hadoop cluster platform is then constructed and assessed. The findings demonstrate that managing massive volumes of data from IOT sensor networks is feasible using the Hadoop cluster architecture.

FUNDING

None

ACKNOWLEDGEMENTS

The authors would like to thank the Ministry of Education/Directorate of Anbar Education, Total Quality Department, and Preparation and Training Department for their facilities, which helped improve the quality of this work. [11, 12]

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] P. Liu, "Cloud Computing, Electronic Industry Press Beijing," 2010.
[2] H. Yedidsion, S. Ashur, and A. Banik, "Sensor network topology design and analysis for efficient data gathering by a mobile mule," *Algorithmica*, vol. 82, no. 10, pp. 2784–2808, 2020.
[3] E. Suganya and C. Rajan, "An adaboost-modified classifier using particle swarm optimization and stochastic diffusion search in wireless IoT networks," *Wireless Networks*, vol. 4, pp. 1–13, 2020.

[4] K. Preethi and R. Tamilarasan, "Monitoring of air pollution to establish optimal less polluted path by utilizing wireless sensor network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 6375–6386, 2020.
[5] P. Yadav, "Workload Analysis in a Grid Computing Environment: A Genetic Approach," *International Journal of Computer Applications*, vol. 93, no. 16, pp. 26–29, 2014.
[6] H. Aziza and S. Krichen, "A hybrid genetic algorithm for scientific workflow scheduling in cloud environment," *Neural Computing and Applications*, vol. 32, no. 12, pp. 1–16, 2020.
[7] A. Tuncer and M. Yildirim, "Chromosome Coding Methods in Genetic Algorithm for Path Planning of Mobile Robots," *Computer and information Sciences* **II**, pp. 377–383, 2011.
[8] L. Li, M. Guo, and L. Ma, "Online Workload Allocation via Fog-Fog-Cloud Cooperation to Reduce IoT Task Service Delay," *Sensors*, vol. 19, no. 18, pp. 3830–3830, 2019.
[9] S. Shah, M. Rashid, and M. Arif, "Estimating WCET using prediction models to compute fitness function of a genetic algorithm," *Real-Time Syst*, vol. 56, pp. 28–63, 2020.
[10] Z. Zhou, F. Li, and H. Zhu, "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments," *Neural Comput & Applic*, vol. 32, pp. 1531–1541, 2020.
[11] D. Bandyopadhyay and J. Sen, "Internet of Things: Applications and Challenges in Technology and Standardization," *Wireless Pers Commun*, vol. 58, no. 1, pp. 49–69, 2011.
[12] R. Roman, "Key Management Systems for Sensor Networks in the Context of the Internet of Things," *Computers & Electrical Eng*, vol. 37, no. 2, pp. 147–159, 2011.