Advances in Parallel Computing Technologies and Applications
D.J. Hemanth et al. (Eds.)

doi:10.3233/APC210149

Parallel Computing Enabled Cloudd-Based IOT Applications

Abiraami T V [a,1], Maithili K [a] and Nivetha J E [a]
[a] *Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai*

Abstract. In delay-sensitive IoT applications, the acquisition and processing of data from the sensor devices to the cloud-based computing environment result in higher computational cost and inefficient system. The cloud servers dedicated to performing larger tasks are found quite difficult as IoT applications collect data frequently. Hence there is a constant retrieval and updating leads to the synchronization of data in the cloud. The potential of cloud servers and virtual machines tend to lose the ability of computing larger tasks. Hence the scope of parallel computing in cloud-based IoT applications are proposed with certain parallel shared models of computation. The Parallel Random Access Machine is introduced in cloud-based IoT applications. The parallel algorithms are designed to eliminate the conflicts encountered in the proposed model. Hence Conflicts of Concurrent Read Concurrent Write PRAM and Conflicts of Concurrent Read Exclusive Write PRAM algorithms are introduced which promotes the efficiency of cloud-based IoT applications.

Keywords. IoT, parallel computing model, parallel algorithms, Cloud computing.

1. Introduction

Internet of Things is a promising technology that provides a feasible solution to the problem being addressed in computation and communication networks. The intense smart applications of IoT that is recently been implemented in various fields such as healthcare [1,2] , agriculture, vehicular automation etc,.. are in the lead of managing the huge data being generated. The big data evolved from the applications due to frequent collection of sensor data urge the utilization of data processing techniques. For instance, the numerous data from the IoT sensor devices namely blood pressure level, heartbeat rate and similar other attributes consequently promotes efficient data organizing, managing and storing. Thus the cloud computing paradigm [3] provides the open-source infrastructure to the number of real-time applications for secure storage and management of data. The delay-sensitive IoT applications are influenced by computational cost and efficiency. To improve the throughput of the system, the simultaneous processing of data is expected. The resource allocation and scheduling of tasks to the cloud servers are certainly expected to be appropriately assigned. The challenges of latency and high performance are encountered in cloud-based IoT applications. The scope of parallel computing in cloud-

[1] Abiraami TV, Dept. CSE, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology. E-mail: abivenkat0506@gmail.com.

based IoT applications enhances the performance of the system. With the introduction of parallel computing [1] design techniques and algorithms the computational cost, energy consumption and latency can be reduced. Parallel Computing in the cloud divides the larger task into sub-tasks to the dedicated processors. The set of instructions are executed sequentially in traditional serial computing. Thus parallel computing is introduced to perform instructions simultaneously with the number of dedicated processors. The processors in the network of parallel systems [4] execute the tasks simultaneously. The inherent sequentialities in parallel computing are solved by using suitable models and algorithms. The output of a task can be an input to another task is said to be data dependence. Hence the data dependence causes the inherent sequentialities in parallel computing. The connected network's algorithm is applied to the parallel random access memory model of computation. The shared memory sub-models have resulted in conflicts for its specific operations. The proposed Conflicts of Concurrent Read Concurrent Write algorithm (CCRCW PRAM)and Concurrent Read Exclusive Write algorithm (CCREW PRAM) solves the conflicts in the parallel computing shared models of computation and promotes the entire application.

2. Related Works

The inevitable IoT applications based on raspberry pi [5] used for automation systems are designed using parallel computing platform's. The components are considered clusters in IoT servers and experimented with the Apache Hadoop framework. Based on the number of requests the IoT servers in the parallel computing framework [6–8] in apache Hadoop have been shown with better performances. The scheduling of tasks using a map-reduce algorithm in first-order logic is allocated for heterogeneous resources. The proper resource utilization and resource allocation are done with the map-reduce algorithm to process huge data through servers. The Soil Quality Analysis in a parallel computational framework [9] for agricultural risk management is observed by decision making parameters. The data collection in smart agricultural management [10, 11] is processed to distribute tomographic images. The cloud computing environment performing various operations and software services are identified with the challenges for doing parallel jobs. To tackle all these challenges the cloud computing platform [12, 13] are dealt with parallel scheduling and dynamic allocation. The cloud computing services like infrastructure, software and platform are continuously facing several challenges [3, 14] when the data gets generated are greater. The virtual machines in the cloud could manage data processing by intelligently distributing computing request. The parallel computing framework with python-based implementation resulted in studied.

3. Proposed system Model

This proposed system model involves a shared memory model of computation in parallel computing. The cloud computing layer comprises the Parallel Random Access Model [3] for parallel allocation, synchronization and resource utilization. The appropriate parallel algorithms for the shared memory model of computation are proposed to enhance the performance of the cloud with parallel computing for IoT applications. The architecture of parallel computing enabled cloud-based IoT applications as shown in Figure 1.

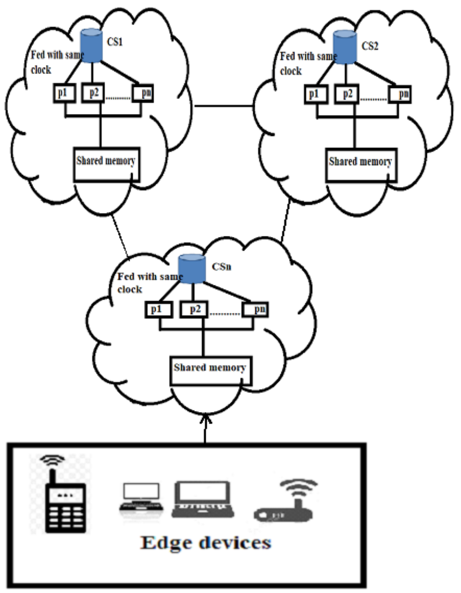

Figure 1. Parallel Computing enabled Cloud-based IoT Applications

3.1. Data Acquisition

The bottom layer of the proposed model is the edge data acquisition layer which is responsible for collecting sensor data from the edge devices. The noticeable sensor devices are temperature sensor, infrared sensors, ultrasonic sensors, etc, in numerous IoT applications updates the data often. The data can be visualized by the user in mobile applications and other web applications. IoT allows the data collected from the physical objects involved in applications could be monitored in available IoT platforms like Blynk, Thingspeak IoT and certain other platforms. The acquisition of sensor data is temporarily stored in the edge devices. The larger task of the application is derived from the organisation of data in the sensor nodes. The sensor nodes offload the compilation of task to the cloud environment.

3.2. Cloud Computing Environment

After every successful updation of sensor nodes, the data collected for the task computation will be offloaded to the cloud environment [3]. Thus the proposed parallel computing model enables the cloud to work efficiently. The shared memory concept of parallel computing makes the cloud performs a set of instructions without conflicts [15].

3.2.1. Parallel Random access machine PRAM

A random-access machine typically stores the program output in the memory. The read and write operations are handled in random access machines with unbounded sequences of registers. A Parallel Random access machine is a shared memory model which executes instructions like load, store, JMP, CMP, etc. Consider P processors are fed with the same clock. Each P is a parallel Random Access Memory machine assigned with a unique address. The machines are accessible by indexing i.e memory locations. The cloud with numerous updation request from the sensor data performs the task regardless of modifications. When the request has often initiated the control of processing those data has to be carried out in an organised approach. When parallel Random Access Machines allows read, write, jump instructions the resources can be properly handled. The upper bound for memory locations available in the parallel computing shared memory model is infinite. This model of computation with infinite memory locations faces a challenge on conflicts for specific operations. Thus the conflict occurs when there is a request for retrieval of data of the same memory location. To solve these conflicts certain operations are considered.

3.2.2. Exclusive Read Exclusive Write EREW

Exclusive Read Exclusive Write doesn't allow the processors to read or write to the same memory locations at the same time. For instance there are n processors, P_1,P_2,P_3,P_4,$P_5$$\ldots$.$P_n$. with dynamic memory locations . When there is a read and write request from the processors P_1,P_4,P_5,P_6 there comes the conflict as the processors are requesting for reading from and write to operations at the same memory locations.

3.2.3. Exclusive Read Concurrent Write ERCW

Exclusive Read Concurrent Write enables the write operation to the same memory locations at the same time from any two processors. But no two processors can request to read from the same memory locations.

3.2.4. Concurrent Read Exclusive Write CREW

Concurrent Read Exclusive Write allows read operation from two processors at the same memory location and same time but no two processors can write to the same memory locations at the same time.

3.2.5. Concurrent Read Concurrent Write CRCW

Concurrent Read Concurrent Write allows processors to read from and write to the same memory locations at the same time. When the number of processors is allowed to write at the same memory locations at the same time the concurrent write has to be constrained. Concurrent write conflicts are constrained by the submodels such as Priority CRCW,Arbitrary CRCW and Common CRCW.

4. Problem Consideration

Consider the cloud servers C_1,C_2,$C_3$$\ldots$$C_n$ are allocated for performing tasks T_1,T_2,$T_3$$\ldots$.$T_n$ in the parallel computing environment. The task can be completed in total time T_1(n) where n is the size of the task. With p processors, the task can be completed in T_p(n). Thus the total execution time can be calculated from T_1(n)/ T_p(n). But p may be impossible to accomplish the task given from the applications as there is an inherent sequentiality. To overcome this the parallel computing random machines are proposed that avoid data dependence thus all processors execute the task simultaneously with the shared memory model of computation.

The concurrent conflicts found in parallel Random Access machine can be addressed by the following parallel algorithms. The parallel algorithms serve the purpose of cloud-based either solved using the depth-first search or breadth-first search. The following are the CCRCW PRAM algorithm and CCREW PRAM algorithm to find the components of the connected graphs to manage the conflicts encountered in parallel models.

4.1. Conflicts Of Concurrent Read Concurrent Write Pram – CCRCW Pram Algorithm

The undirected graph containing the number of vertices and edges are solved to the conflicts detected in parallel machines. Consider an undirected graph G = (V, E). It is necessary to find the parent pointer of the graph for every vertex. The vertex in a graph is chosen to represent a component. It is also important to find out the star graph i.e. the tree of level 2. To do that the graph is represented in an adjacency list. It is assumed that there is the number of processors to compute the number of data stored in a linked list. When multiple processors are requesting access in the same memory location it is noted that any one of the processors will gain the rights to do an operation. The steps involved in the CCRW PRAM algorithm are as follows

1. In the initialisation step, for every vertex v in V, the vertex is defined to its parent accordingly.
2. The parent pointer can be pointed to itself and every vertex will have a unique parent.
3. Hooking, pointer jumping and contracting are the essential steps followed in this algorithm to reduce the possibility of conflicts.
4. The height of the tree is reduced by executing pointer jumping to every vertex.
5. If a tree has more than two levels grandchildren will find its root node by its parent.

Algorithm 1 CCRCW PRAM

Input: undirected graph G= (V, E) **Output**: connected component on CRCW PRAM

```
1: for uε v do
2:     Assign p(v)=v
3:     for (u,v) ε E do
4:         edge(u,v)⟵ edge(v,u)
5:         if p(v)¡p(p(v)) then
6:             p(p(u)= p(v)
7:     for each (u,v) do
8:         if  p(u)= p(p(u) then
9:             p(u)== p(v)
```

4.2. Conflicts Of Concurrent Read Exclusive Write Pram Algorithm

To find the connected components on concurrent read exclusive write the CCREW PRAM algorithm is proposed. The undirected graph with the number of vertices and edges are stored in adjacency list representation with twin pointers i.e. I and j respectively. The neighbours and children of the vertices are to be chosen and if a vertex is not found with children then it arbitrarily chooses the neighbouring child. Edge plugging, Renaming and merge sort are important functions implemented in this algorithm. The steps followed in the CCREW PRAM algorithm are as follows.

1. Find the smallest neighbour of the vertex
2. For every vertex find out its children if not choose arbitrarily.
3. Every parent pointer is a directed version.
4. Take the adjacency list of the vertex and combine it into the adjacency list representation which is a doubly-linked list.
5. Reduce the tree into super vertex by performing pointer jumping;
6. Rename the edges and perform merge sort to find out the connected components.

Algorithm 2 CCREW PRAM

Input: undirected graph G= (V, E) Output: connected component on CREW PRAM

```
1:  for uεv do
2:      G=smallest neighbour(U¡V)
3:      if G==true then
4:          Assign P(v)=u
5:      for vεV do
6:          P(v)←— [p(v),v]
7:      for each (u,v) do
8:          if p(v)=p(p(v)) then
9:              [u,v]=r(u),r(v)]
10:         paramergesort(val,n,data)
11:         Data = mergesort(data)
12:     for d=1 to n do
13:         Data= paramerge(val,d,data)
14:     New ←—data
```

5. Experiments And Evaluation

The data for IoT application is experimented by implementing both conventional and proposed model. The number of resources offloaded to the cloud environment computes the task and calculates the total execution time in sec. Figure 2 illustrates that the total execution time is improved in the proposed model than the conventional serial computing.

The proposed parallel random-access machine model of computation performs concurrent retrieval and updation of data obtained from edge devices of IoT applications. Figure 3 explains that the CRCW PRAM model found conflicts when the number of processors requesting to access the same memory location. When two processors accessing the same memory location at the same time are executed by the proposed algorithm the machine more processors computed in lesser time. Similarly, exclusive write operations are allowed to only one processor. The arbitrary CCCRW PRAM model implemented in Figure 4 explains that a randomly chosen processor performs IoT data updating to the

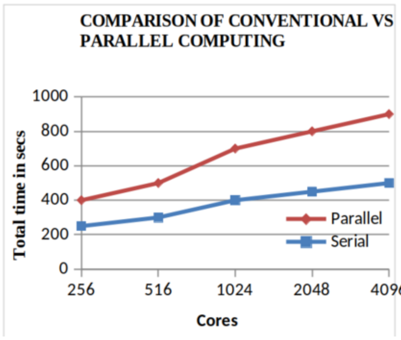

Figure 2. Comparison of conventional vs parallel computing

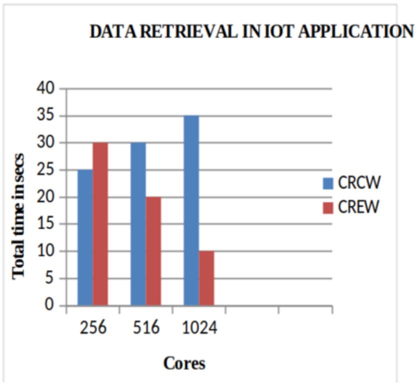

Figure 3. Data retrieval in IoT Applications

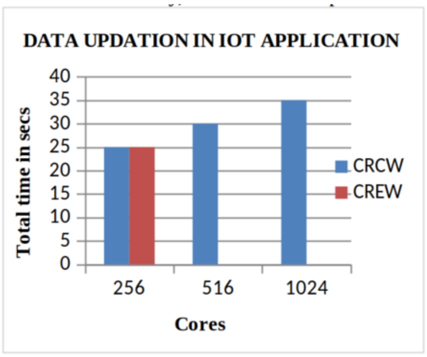

Figure 4. Data updation in IoT application for different data values

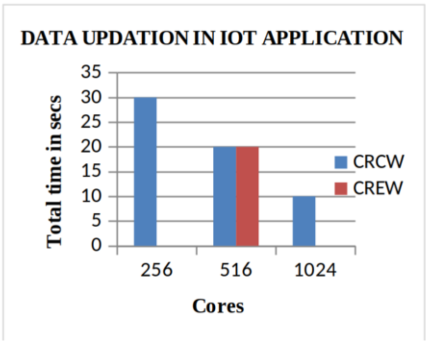

Figure 5. Data updation in IoT application for similar data values

cloud. Similarly, the CCREW PRAM algorithm executes the exclusive updation. When the number of processors requests for accessing the same dataset updation all the processors are allowed to access simultaneously and the time consumption is accordingly computed with the number of processors.

6. Conclusion

The proposed parallel computing shared model of computation improves the performance of IoT applications in terms of efficiency and lower computational cost. The workload of the cloud is managed as the number of processors available to execute the task simultaneously with the proposed parallel computing algorithms. The CCRCW PRAM and CCREW PRAM algorithms are implemented in a way to reduce the total computation of data from edge devices in IoT applications. Further, the energy consumption of the cloud can be reduced in future research.

References

[1] Sukesh B, Venkatesh K, Srinivas LN. A Custom Cluster Design With Raspberry Pi for Parallel Programming and Deployment of Private Cloud. Role of Edge Analytics in Sustainable Smart City Development: Challenges and Solutions. 2020 Jul 15:273-88.

[2] Sai Ramesh L, Sundar SS, Selvakumar K, Sabena S. Tracking of Wearable IoT Devices Through WAP Using Intelligent Rule-Based Location Aware Approach. Journal of Information & Knowledge Management. 2021 Feb 29;20(supp01):2140005.

[3] Fang A, Cui L, Zhang Z, Chen C, Sheng Z. A Parallel Computing Framework for Cloud Services. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) 2020 Aug 25 (pp. 832-835). IEEE.

[4] Tyagi N, Rana A, Kansal V. Load Distribution Challenges with Virtual Computing. Intelligent Computing in Engineering. Advances in Intelligent Systems and Computing. 2020 Apr 9;1125:51-6.

[5] Nugroho S, Widiyanto A. Designing parallel computing using raspberry pi clusters for IoT servers on apache Hadoop. In Journal of Physics: Conference Series 2020 Apr 1 (Vol. 1517, No. 1, p. 012070). IOP Publishing.

[6] Yang X, Nazir S, Khan HU, Shafiq M, Mukhtar N. Parallel computing for efficient and intelligent industrial Internet of Health Things: an overview. Complexity. 2021 Jan 23;2021:1-11.

[7] Dafir Z, Lamari Y, Slaoui SC. A survey on parallel clustering algorithms for big data. Artificial Intelligence Review. 2021 Apr;54(4):2411-43.

[8] Ianni M, Masciari E, Mazzeo GM, Mezzanzanica M, Zaniolo C. Fast and effective Big Data exploration by clustering. Future Generation Computer Systems. 2020 Jan 1;102:84-94.

[9] Pereira MF, Cruvinel PE, Alves GM, Beraldo JM. Parallel Computational Structure and Semantics for Soil Quality Analysis Based on LoRa and Apache Spark. In 2020 IEEE 14th International Conference on Semantic Computing (ICSC) 2020 Feb 3 (pp. 332-336). IEEE.

[10] Shi J, Shun J. Parallel algorithms for butterfly computations. In Symposium on Algorithmic Principles of Computer Systems 2020 (pp. 16-30). Society for Industrial and Applied Mathematics.

[11] Reshma R, Sathiyavathi V, Sindhu T, Selvakumar K, SaiRamesh L. IoT based Classification Techniques for Soil Content Analysis and Crop Yield Prediction. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) 2020 Oct 7 (pp. 156-160). IEEE.

[12] Ding L, Wang Z, Wang X, Wu D. Security information transmission algorithms for IoT based on cloud computing. Computer Communications. 2020 Apr 1;155:32-9.

[13] Xhafa F, Sangaiah AK. Advances in Edge Computing: Massive Parallel Processing and Applications. IOS Press; 2020 Mar 10.

[14] Luo Y. Environmental cost control of coal industry based on cloud computing and machine learning. Arabian Journal of Geosciences. 2021 Jun;14(12):1-6.

[15] Branitskiy A, Kotenko I, Saenko IB. Applying machine learning and parallel data processing for attack detection in iot. IEEE Transactions on Emerging Topics in Computing. 2020 Jul 1:1-12