Extended Grid Database Model for Heterogeneous Replication in a Distributed System

Dayo Reuben Aremu[1] and Moses Kazeem Abiodun[2]
Department of Computer Science.
Unversity of Ilorin, Ilorin,
Nigeria.
Email: [1]draremu2006@gmail.com, [2]abbeykmos@yahoo.com

ABSTRACT

There is an increasing need to share resources across heterogeneous database around the world. Data are stored in various database products such as MySQL, Oracle, SQL Server and so on. Each of these databases has different database schema, which needs to be brought together as one entity to achieve heterogeneous replication. Several models have been proposed including the Publish and Subscribe, Charron Bost et al Database Replication Model, Grid Database Replication Model (GDRM), Adaptive Model and so on to achieve this. However, heterogeneous replication has not been achieved and the consistency of data is not guaranteed. This study therefore aimed at extending grid database replication model to achieve heterogeneous replication and improve consistency of data. The objectives of the study were to: (i) design an extended Replication Model to achieve heterogeneous database replication; (ii) simulate consistency of data replication with load balancing using Gridsim Toolkit; and (iii) implement a new load balancing techniques to improve cloud services to client. The Charron-Bost Database Replication Model was extended by adding consistency using Load Balancing technique. In addition, the Model was extended to achieve heterogeneous replication to bridge the gap among different database system. The framework for achieving heterogeneous replication between MySQL and Oracle database was also provided. The hybridization of Round Robin with Maximum Minimum RRMaxMin was simulated using Cloud Analyst and the performance was compared with other load balancing algorithm for response time and cost effectiveness. The findings of this study were that: The extended model enhance ability to perform heterogeneous replication in grid environment. The introduction of a new load balancing algorithm (RRMaxMin) brought about significant improvement in the cloud services. Out of all the algorithms that were evaluated, RRMaxMin performed better in terms of overall response time of 226.16ms and 89.9% cost effectiveness. The study concluded that load balancing strategy is a more reliable way to achieving heterogeneous replication in a distributed environment. The study recommended that RRMaxMin should be adopted in the cloud service to improve user satisfaction.

Keywords: *Replication, Database-Replication; MySQL-Replicator, Tungsten-Replicator.*

Reference Format:

Aremu, Dayo Reuben and Abiodun, Moses Kazeem (2019), Extended Grid Database Model for Heterogeneous Replication in a Distributed System,
Afr. J. MIS, Vol.1, Issue 3, pp. 1 - 23.

1. INTRODUCTION

Recently, replication has attracted a great deal of attention and interest especially in the community of grid users. However, the challenges to achieve replication in a heterogeneous environment is enormous because data are stored in various database products such as MySQL, Oracle, DB2 (Database2), SQL Server and so on. Each of these databases has different database schema. Heterogeneous replication works slightly differently compared to homogeneous replication, which occurs between the same databases as the native MySQL to MySQL replication. Even though homogeneous replication offers the advantages of easy replication process, easy design and easy management of replication, the major disadvantages of Homogeneous replication is that it is not easy to force homogeneous environment as most organizations uses systems that they are familiar with. Heterogeneous replication is very important to many organizations around the world and it is very good as huge data can be stored in one Global center from different data center, remote access is done using the global schema, different DBMSs may be used at each node, which brings in variety and increases experiences of the operator. The major disadvantage of heterogeneous is that it is difficult to design and manage.

,In addition, another challenge that needs to be address in a distributed networking environment is how to maintain consistency of data[1]. Several works have been done on replication model. For instance, Sharma in [2] adopted a "Publish and Subscribe" structure, where a component that capture identified data changes in the source database(s) publishes them into provisional tables. The components that apply change then take information from the provisional tables and apply them to the target databases(s) in a planned order. The drawbacks of this model are that: it does not address grid issues; it is difficult to optimize replication strategies; and there is no resource discovery mechanism.

Chen, Berry and Dantressangle in [3] came up with the model for grid database replication (GDRM) to carry out database replication in the Grid. The drawback of this model were that: it does not perform heterogeneous grid database replication; and the consistency of data was not guaranteed. Charron-Bost et al. in [4] presented a generic architecture for practical database replication which allows replication to be carried out from one database system to another, but the drawbacks of this model were that: it does not support heterogeneous replication; and the consistency of data replicated cannot be guaranteed. This study therefore presents an Extended Grid Database Replication Model for managing database replication and proposes a load balancing strategy for managing data consistency in a heterogeneous environment. The study also proposed a hybridization of the Round Robin and Maximum Minimum Algorithm in order to improve the services in the cloud. The specific objectives of the study were to: design an extended Replication Model to achieve heterogeneous database replication and improve consistency of data replicated using load balancing techniques; simulate consistency of data replication with load balancing using Gridsim Toolkit; and Implement a new load balancing technique to improve cloud services to client. The rest of the paper is organized as follows: Section 2 discussed the related work. Section 3 presented the methodology for database replication model. Section 4 presented simulation/implementation of the replication designs and discusses the results obtained from the analysis and evaluation of the model alongside the existing models. Section 5 summarizes and concluded the paper.

2. RELATED WORK

Abdul-Wahid et al., in [5] presented Adaptive Distributed Database Replication through Colonies of Pogo Ants. The model optimizes distribution of partially replicated databases in network systems which minimizes inter node communication and gives better response time and enables dynamic addition of replicas. Though the model is event driven system, but it does not synchronizes the learning process with events.

Chen Y, Berry D., and Dantressangle P. in [3] presented Transaction Based Grid Database Replication. This model used existing grid mechanism to provide metadata registry and it defines high level APIs which manages transaction based replication across multiple domains and legacy software's are moved to grid smoothly. APIs of this framework is not generic enough to cope with systems from different vendors and optimization related to memory or processing is not discussed.

Correia et al. (2007) presented GORDA: An Open Architecture for Database Replication. The Architecture provides third party replication by introducing a middleware wrapper without modifying the database server. The model produces better solutions on PostgreSQL. Some architecture however does not provide expected results when the middleware interface is implemented on them like apache derby and sequoia.

Agrawal et al., in [6] presented Asynchronous View Maintenance for VSLD Databases. This mechanism enables better implementation and maintenance of view and indexes in massive scale databases and solves complex queries. However, maintenance of complex views and index requires resources (memory and processing) and in small scale databases it does not provide better results.

Charron-Bost et al., in [4] also proposed the generic architecture for practical database replication. The architecture works with homogeneous database replication. However, it does not take care of heterogeneous replication between different database system and the consistency address in the model is not well stated.

In [7] Thomson presented Calvin: Fast Distributed Transactions for Partitioned Database Systems. It is a data replication and transaction scheduling layer that works well in distant geographical replicas and has no single point of failure providing Atomicity, Consistency, isolation and replicas are located at distant geographically areas so it does not perform well in HATs (Highly Available Transactions).

Mohammed and Mohammed, (2014) proposed An Adaptive Replication Model for Heterogeneous System. The model provides several advantages for enterprise application, a more secure and reliable data transmission. It make database replication much easier to handle. However, the Model supports only three types of database, supporting all the existing databases can be another important improvement of our prosed system. The prototype tool has fewer fields to inset data into different master and replication table.

Mohammed et al., in [8] presented A Round-based Data Replication Strategy. IPFRF is superior to PFRF in terms of average file delay per request, average file bandwidth consumption per request and percentage of files found. The model has not been tested on a real data grid. It needs to allow nodes to enter and leave the grid to achieve expected result.

Stavros S. and Angelo Sifaleras [9] proposed Binary-Tree Based Estimation of File Requests for Efficient Data Replication. The strength of BTBest is that it predicts the increasing demand of high-potential files and makes them available via replication sooner than other well-known strategies. BTBest has to be tested in a real data grid. The insertion of more options regarding the increasing or decreasing factors needs to be looked at.

Pooja and Pranati, [10] presented a round robin algorithm and describe it as an example of static algorithm and it appropriate task to the processor in no particular order. Since the time to run the task is not known in advance, some of the nodes are lightly loaded thus, some nodes get heavily loaded.

Kokilavani and Amalarethinam, [11] presented the Maximum Minimum algorithm in which the cloud manager focuses on tasks having maximum execution time after discovering task with the lowest running time. The allocated task is taken away from the list of the tasks that are to be allocated to the processor. Hence, the execution time for all other tasks is updated on that processor. This makes the algorithm to work well since the specifications are known in advance and because of its static algorithm.

3. METHODOLOGY

This section presents methodology for achieving database replication in a heterogeneous environment. The methodology extended the work of Charron-Bost in [4] by adding a consistency service using a load balancing technique to bring about data consistency in heterogeneous environment. Using virtual machines, the heterogeneous replication between MySQL and Oracle was performed. The section also presents design for the implementation of a new load balancing algorithm in the cloud services to improve the performance of cloud services around the world. The methodology presented in this section is summarized in Figure 3.1.

The proposed model extended the charron-bost model by enhancing the reflector with a consistency service using load balancing method called filter. The data is extracted using bin log from MySQL and replicated using an applier to Oracle. The proposed model extended the architecture of Charron-Bost [4] by adding a consistency service to enhance the efficiency of the system and breaking down the task for heterogeneous replication in a grid environment.

In the Master replicator, we have the Extractor Class, which is the entry point in this model. The MySQL Extractor class takes out all the transaction, written by MySQL to the binary log.

The reflector of the Charron-Bost *et al.*, 2010 was also extended to accommodate the schema change necessary for heterogeneous replication. The new extended reflector is called a filter. The functions of the filters includes:

i. EnumToStringFilter –In the binary log, MySQL database saves ENUM column category values as indexes; the filter maps the textual illustration instead, so it save suitable text values on the Oracle side, as different to worthless values.
ii. To upper case filter –The Column names are normally lower case in MySQL table; the filter changes the case to what Oracle likes.
iii. PrimaryKeyFilter –This used for UPDATE and DELETE row changes, it strip down all the key-value pairs that are unnecessary in the WHERE clause, leaving only the primary key comparison.
iv. OptimizeUpdatesFilter As MySQL binary log is excessive in the key (WHERE) part, it has also many information in the SET part of an UPDATE; for example, a table that has hundred columns and the statement updates only one, the row change event will contain all the hundred columns updated; this filter strips abundant assignments and leaves only the columns that actually changed.

The load balancing method of consistency works with the reflector to make sure that everything works accordingly. The communication agent links with the slave replica to transfer the transform data to the replica and an Applier on the other end apply the replica to the target database. This allows heterogeneous replication to be perform easily and all field in one database can be transform to the other databases system without any loss of data.

Every single transaction row is executed to Oracle. After it has gotten to the end of the task, the transaction is committed and data is available for applications linked to the Oracle instance. The good thing here is that all the phases stated above are working in parallel. Each phase is separated from the other by queues, which work like buffers. For example, MySQL binary log is usually very fast when taking out data, thus, it thrusts the data to the queue, and from which next phase is reading and so on, and thus extraction is not blocked.

3.1 Research Framework

Almost all the database software available in today's market has support for heterogeneous replication since it is now important for organizations to share valuable vast amount of information together. Open source and commercial base heterogeneous software are now abound in the market. The most common of these tools are Oracle Golden Gate and Tungsten Replicator. Based on the review of the two software, the researcher gives the framework of performing heterogeneous replication between MySQL and Oracle using Tungsten Replicator. Tungsten replicator is selected since it is free and its performance is like the oracle golden gate.

The common stages to replicate from MySQL to Oracle in a dependable setting are:

1. Extract from MySQL's binary log the latest committed transaction.
2. Cover the transaction with metadata, including Global Transaction ID.
3. Transform or Filter it to adapt to the taste of Oracle.
4. Apply it Oracle.
5. If it succeed, go back to (1), and if not, break the replication.

It is best to break the replication for data that is sensitive, if the unexpected happens. When an error comes up, you can investigate it, fix it and resume. This is the reason why step (2) comes handy. Global transaction IDs in heterogeneous replication is also present in Tungsten Replicator, hence we are aware, where it stopped and where to resume. The flowchart of the framework is as shown in Figure 3.2:

3.2 Load Balancing for Consistency

This study seeks to extend Charron-Bost's work in [4] by adding a consistency service using load balancing techniques to bring about consistency in the replicated data. The Charron-Bost et al model [4] make used of strong consistency, which brought about a lot of bottleneck in the system. Our approach of load balancing allows for a flexible approach to the consistency issues. The main objective here is to bring a corresponding solution amongst load balancing and replica consistency. The study implements the load balancing approach using quorum structure, since, this offers a better management and representation of replicas. For this, the study recommend that coteries should represent the grid. A coterie comprises all nodes having replicas of the same data, organized in a binary tree. A route from the root to the leaf node of the tree is refer to as quorum. To increase the data availability, n versions is define for each coterie node. Every single version is categorized by three variables<N, S, V>, signifying respectively, the node that makes or altered this version, the stamp which denotes the moment of the creation or the update version of the replica and lastly the value of the replica. An illustration is shown in Figure 3.3.

The example shows that for each node of the coterie, three versions are define. The consistency of the replica is founded on the write and read protocol. Replicated data is updated through a write protocol and it is demanded through a read protocol. Three states are defined for a node: Free (F), Occupied (O) and Blocked (B) as shown in Figure 3.5. A node is in Free State if all its versions are release. If a node contains at least one version locked, then it is deem that the node is occupy. A node is in a block state when every versions are locked. The likely switches of a state to another are demonstrated in Figure 3.4.

If it is assume the initial state of the node is free (F), if a request to read or to write is sent to that node, the version selected to accomplish the operation will be locked and the node passes to occupied state (O). If this node receives another request then it keeps the same state even if it still has released versions, else it moves to the blocked state (B). If a node is in a state called blocked and a version has been released, then the node returns to an occupied state. The node returns to a free state if all the locks of all versions are released.

The load balancing algorithm is hereby presented:The structured coterie and load nodes of each coterie will be the Input. The restructured coterie will be the result of processing.

```
For each coterie do
If exist node ≠ leaf and load node = 3
Then
        For each parent node do
        If (parentload > childload1) and (parentload > childload2)
        Then swap (parent, child(j)) with minimal communication cost
                Elseif (parentload > childload1)
                        Then swap(parent, child(1))
                Elseif (parentload > childload2)
                        Then swap(parent, child(2))
                Endif
        Endif
        Next
   Endif
 Endif
Next
```

3.3 Load Balancing in Cloud

Load Balancing is a process that allocates the workload among diverse nodes in the given environment such that it certifies no node in the system sits idle or loaded too much for any instant of time. The major concern of load balancing in the cloud domain is in assigning the load dynamically among the nodes in order to satisfy the user requirements and to offer maximum resource utilization by categorizing the overall available load to distinct nodes. The researcher also uses simulation to examine how load balancing algorithm can help improve the cloud services. Cloud computing is a new evolving technology in the academics and industry. Cloud Analyst is a tool that is used for development and simulation of cloud environment before actual deployment to the real world application. In the processing of cloud analyst various service broker policies and load balancing policies have been used for response of different users. This project proposed a new hybrid approach for load balancing using Round robin and Max-Min Algorithm to allocate virtual machines to different user base requests called RRMaxMin Load balancing.

The Proposed Hybrid approach will overcome the problem of Round robin that assign tasks without priority and the Max-Min that suffers from starvation where the tasks with the maximum finish time will get executed first while leaving behind the tasks having the minimum completion time. The proposed flow of work of the new hybrid algorithm is shown Figure 3.5.

In the proposed framework, cloud job scheduling and load balancing policy has been used for allocation of virtual machine to different user bases requests. In this process user base has been deployed in different regions of the world that transmit their request to different datacenters allocated by third party cloud server.

Figure 3.6 shows the process of implementing the RRMaxMin, while the assigned task with minimum execution time is being processed. The task with the maximum expected time will be allocated with round robin so that it does not starve the job with minimum execution time tasks. This approach enhances the maximum minimum algorithm while the round robin algorithm will only be used on the task with maximum execution task.

Algorithm:

Input: jobs, datacentres, user base
Output: Response time, Datacentres servicing time and cost analysis
For all task in task set **do**
Calculate expected completion time
Is Task set empty?
YES: then end.
NO: then update the time for the selected resource and expected time
Else
Find tasks with minimum execution time and tasks with maximum execution time
End if
Calculate the difference between the maximum time and minimum time
If difference <= minimum then
Assign tasks with Maximum Exec time using Round Robin
Else if difference is >= maximum
Assign tasks with Minimum Exec time
End if
Remove the task from the task set
Update the time for the selected resources and expected time
End

4. IMPLEMENTATION

This section presents the implementation of the extended Grid Database Replication Model. The model was extended by adding a consistency model using load balancing techniques which was simulated using Gridsim simulator. The reflector was extended by adding a schema transformation from one database to another and was implemented using an experiment built using two Virtual Machines (VM) on an X series server, one server signifies the source server and another one signifies the target server. The section also presents the implementation of a new load balancing approach using cloud analyst. The performance of the new load balancing system was tested against the existing one.

4.1 Experimental Design

Achieving heterogeneous replication has been a tasked that previous research has been trying to implement and while some have been able to achieve it to a certain extent. This study uses the tungsten replicator to achieve it with improve performance.

To perform this experiment, two Virtual Machines (VM) was built on X series server, where one server signifies the source server and another one signifies the target server as shown in Table 4.1. After the environment is ready, the researcher performed some test scenarios under several conditions to check whether the data could replicate without any loss. The result of the test scenario can be seen in Table 4.2. It shows the test scenario and result from the experiment, the model using Tungsten Replicator can duplicate data even in situation when the network is off. This can be observed as one turn off one of the component that replicate. The data will be pooled at trial file at server 1 and continue delivery to server 2 as the network is on without data being lost.

The operation consists of following steps:

1. Installation / Building tungsten from source
2. Preparing equivalent schema for Oracle
3. Configuring Master service
4. Configuring Slave service
5. Generating worker tables (temp tables used by tungsten) for replication to be created on MySQL
6. Start the replication

Making corresponding schema for Oracle: The ddl in tungsten replicator extractor read table definitions from MySQL and create appropriate Oracle table definitions to use during replication.

Table 4.2 shows the scenarios carried out for testing the effectiveness of tungsten replicator, it shows that heterogeneous replication can be achieve easily with Tungsten replicator. The replication of data from MySQL database to Oracle database work seamlessly and without any loss of data. This result goes a long way to put the mind of organization at rest to know that no matter the database system you are running now, if there is any need to move to another database system, all your record will still be intact. Data integration across the globe is also very easy with replication in a heterogeneous environment.

4.2 Comparison between the two models

See Table 4.3.

4.3 Load Balancing In Cloud Computing Using RRMaxMin Algorithm

For the implementation of the algorithms, Cloud Analysts was used, A graphical user simulator for cloud environment. Using Eclipse Java EE IDE for Web Developers. Version: Mars.2 Release (4.5.2) Build id: 20160218-0600. This study has been able to add the new algorithm RRMaxMin to Cloud Analyst source code.
The proposed model defined six user bases and two datacenters in cloud computing environment. The user bases and datacenters were allocated in different regions for processing. Region boundaries have been defined by different colour regions.

Table 4.4 represents user base configuration that has been designed for simulation of cloud computing environment.

This table represents datacenter configuration that has been designed for simulation of cloud computing environment.

Figure 4.1 shows the configuration parameters used for this simulation. The simulation duration was set to 60 minutes, two data center has been defined and closest data center service broker policy was used. Other parameter were as described in Table 4.6.

In Figure 4.2 the load balancing used is Round Robin, the number of simultaneous user from a data center is set at 1000. The request grouping factor in data center is set to 10 and the executable instruction length per request is set to 100.

Figure 4.3 show the load balancing algorithm added to the Cloud Analyst listed among the other load balancing algorithm of Round Robin, Equally spread current execution load and throttled.
On the basis of these simulation parameters various different load balancing policies have been used for simulation of cloud computing. On the basis of these simulations various parameters have been evaluated for performance evaluation of proposed approach.

4.6 Performance Evaluation Parameters

The parameters used for performance evaluation of the proposed approach include.

(i) Over All Response Time

Over all response time has been measured for user base request completion and datacenter processing time. These parameters provide information about time taken by cloud service provider to respond to requests.

(ii) Datacenter Request servicing time
Datacenter request servicing time provide information about a single datacenter for serving requests of different user bases

(iii) Cost Analysis
Cost analysis parameters provide information about different load balancing approaches using data transmission cost and total VM cost occurred.

The Chart in figure 4.4 shows the result of the four algorithms that were tested and clearly the new algorithm performed slightly better than the Throttle algorithm in terms of overall response to time. It also shows that the new algorithm RRMaxMin is better than Equally Spread and Round Robin algorithm.

The chart in Figure 4.5 show that in Data center processing time, the throttle algorithm is better than the new algorithm just propose slightly.

Figure 4.6 chart show that in terms of cost of data transfer the new algorithm RRMaxMin fared better that others.

4.3 Discussion of Results

The result from this study revealed that:

(i) The designed extended model enhanced ability to perform heterogeneous replication in grid environment and sharing of information across different database systems without any loss of data;

(ii) By using load balancing to reconfigure the coterie, it increases the ability to get access to files. The results obtained from this approach shows that managing the replicas consistency is well-adjusted automatically following the state of each node of the coterie and a consistency index of 81% was achieved.

(iii) The introduction of a new load balancing algorithm (RRMaxMin) brought about significant improvement in the cloud services. The Processing time taken by the data center is good from both Throttled (0.61ms) and RRMaxMin (0.63), but it was worst for Round Robin (1.73ms); and

(iv) Out of all the algorithms that were evaluated, RRMaxMin performed better in terms of overall response time of 226.16ms and 89.9% cost effectiveness.

5. SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

This study presents an extended grid database replication model. The specific objectives were to: design an extended Replication Model to achieve heterogeneous database replication; simulate consistency of data replication with load balancing using Gridsim Toolkit; and implement a new load balancing techniques to improve cloud services to client.

The study extended the work of Charron-Bost et al (2010) by adding a consistency service using a load balancing strategy to bring about data consistency in heterogeneous environment. The consistency model was implemented using Gridsim simulator. Heterogeneous replication was also performed using virtual machine in a master-slave environment to replicate from MySQL to Oracle using Tungsten Replicator. Another heterogeneous environment is the cloud computing, and the study look at the ways to improve cloud services to clients by implementing a new load balancing algorithm called RRMaxMin using Cloud Analyst simulation tool. The new load balancing algorithm was compared with other algorithm like the Round Robin, Throttle and Equally Spread Algorithm.

The findings from the study show that:

(1) The designed extended model enhances ability to perform heterogeneous replication in grid environment and sharing of information across different database system s without any loss of data;

(2) The framework for achieving heterogeneous replication has been laid down and will enhance the sharing of information across different database system.

Using Virtual Machine, heterogeneous replication from MySQL to Oracle was performed which was successful without any loss of data.

(3) By using load balancing to reconfigure the coterie, it increases the ability to get access to files. The results gotten from this approach shows that managing the replicas consistency is well-adjusted automatically following the state of each node of the coterie and a consistency index of 81% was achieved.

(4) The introduction of a new load balancing algorithm (RRMaxMin) brought about significant improvement in the cloud services. The processing time taken by the data center is good from both Throttle and RRMaxMin, but it was worst for Round Robin and

(5) Out of all the algorithm that were evaluated, RRMaxMin performed better in terms of overall response time of 226.16ms and 89.9% cost effectiveness.

5.2 Conclusion and Future Work

The study extends the Grid Database Replication Model by adding consistency to replicated data using a load balancing approach. The consistency of data is very important to any organization. This research work has been able to demonstrate that consistency of data replicated in Grid can be relied upon.

Heterogeneous replication between MySQL and Oracle was achieved using tungsten replicator in a virtual machine. The essence is to bridge the gap across organizations using different database systems, but needs to share data amongst themselves. This study shows that this is now possible irrespective of the database system in place. The new algorithm RRMaxMin was successfully integrated into the Cloud Analyst and used to simulate various scenarios in order to determine the effectiveness of the algorithm. The results show that there was an improved performance in terms of data processing time and cost implication. The data processing time still needs to be improved upon. The study recommended hybridization of the Round Robin Maximum Minimum for further enhance the data centre response time of the algorithm.

REFERENCES

[1] Dang, Kemian (2008). "Towards Heterogeneous Grid Database Replication", *Master of Science Computer Science,* School of Informatics University of Edinburgh.

[2] Sharma R., (2000). *Microsoft SQL Server 2000: A Guide to Enhancements and newFeatures,* Addison-Wesley Professional.

[3] Chen Y, Berry D., and Dantressangle P. (2007) *Transaction-Based Grid Database Replication*. National e-Science Centre, Edinburgh, UK.

[4] Charron-Bost, B., F. Pedone and A. Schiper, (2010). *Replication: Theory and Practice*. 1st Edn., Springer, Berlin,pp: 290.

[5] Abdul-Wahid, S., Andonie, R., Lemley J, Schwing, J and Widger J. (2007). Adaptive Distributed Database Replication Through Colonies of Pogo Ants, In *Parallel and Distributed Processing Symposium*, IPDPS.

[6] Agrawal, P., Silberstein, A., Cooper, B. F., Srivastava U and Ramakrishnan R. (2009). Asynchronous view maintenance for VLSD Databases, In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*.

[7] Thomson, A., Diamond T, Weng S. C, Ren, K, Shao, P & Abadi, D. J. (2012)."Calvin: fast distributed transactions for partitioned database systems" In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. (2012).

[8] Mohammad B, Abdallah A. E., Almakadmeh K, and Tahat N., (2016) A round based data replication strategy, *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 31–39, 2016.

[9] Stavros Souravlas and Angelo Sifaleras, (2017). Binary-Tree Based Estimation of File Requests for Efficient Data Replication, *Parallel and Distributed Systems IEEE Transactions on*, vol. 28, no. 7, pp. 1839-1852, 2017.

[10] Pooja S., Pranati M. (2013). Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing, *International Journal of Computer Science and Information Technologies*, Vol. 4 (3) , 2013, 416-419.

[11] Kokilavani T, Dr. D. I. George A. (2011). Load Balanced Min-Min Algorithm for Static Meta Task Scheduling in Grid computing‖ *International Journal of Computer Applications* Vol-20 No.2, 2011.

[12] Sachin K., Neeraj S. (2012). A Priority based Dynamic Load Balancing Approach in a Grid based Distributed Computing Network, *International Journal of Computer Applications* (0975-8887), Vo. 49, No.5, July 2012, pp. 11-13.

[13] Santhosh, B., & Manjaiah, D. (2014). An improved task scheduling algorithm based on max-min for cloud computing. *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2, Special Issue 2, pp. 84 – 88.

[14] Silberschatz, A., H.F. Korth and S. Sudarshan, (2011). *Database System Concepts*. 6th Edition, McGraw-Hill, New York, ISBN-10: 0071289593, pp: 1349.

[15] Anees Hussain and Khan M. N. A. (2014). *International Journal of Database Theory and Application* Vol.7, No.1, pp.93-102.

[16] Ratan M. and Anant J.,(2012). Ant Colony Optimization: A solution of Load Balancing in Cloud, *International Journal of Web & Semantic Technology* (IJWesT), April 2012.

[17] Xiong, C., Feng, L., & Chen, L. (2013). A new task scheduling algorithm based on an improved genetic algorithm in a cloud computing environment. *Advances in Information Sciences and Service Sciences*. Vol. 5(3), pp. 32.

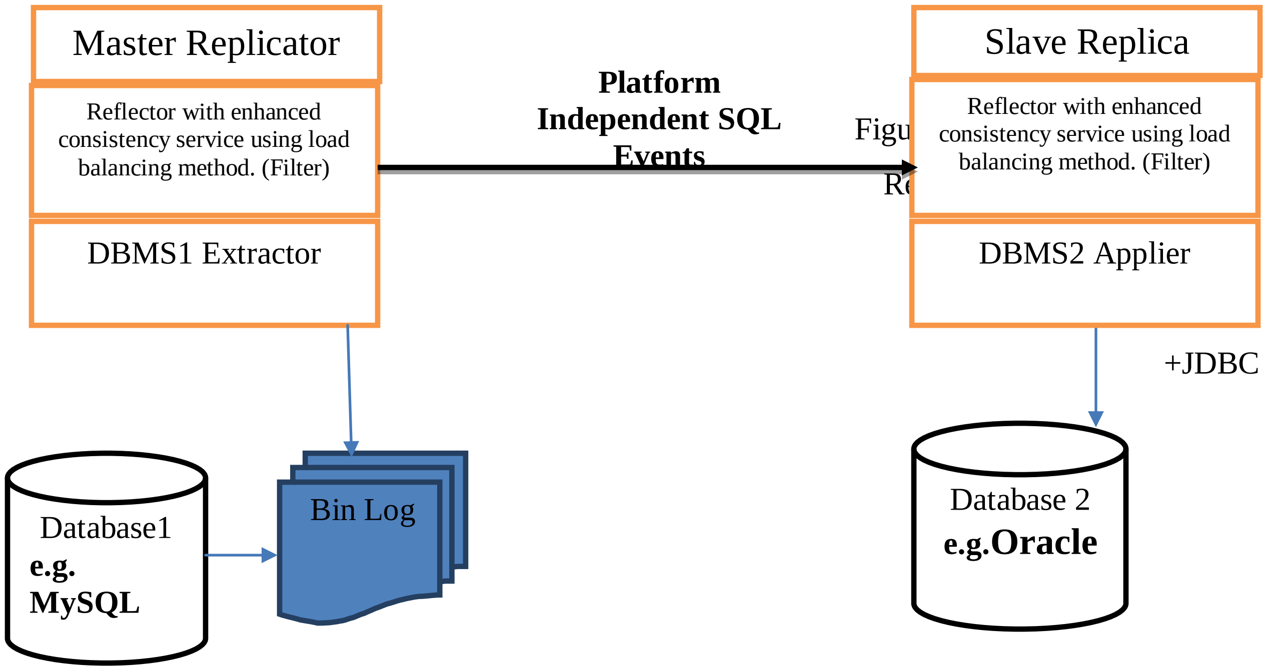


Figure 3.1: The Extended Grid Replication Model

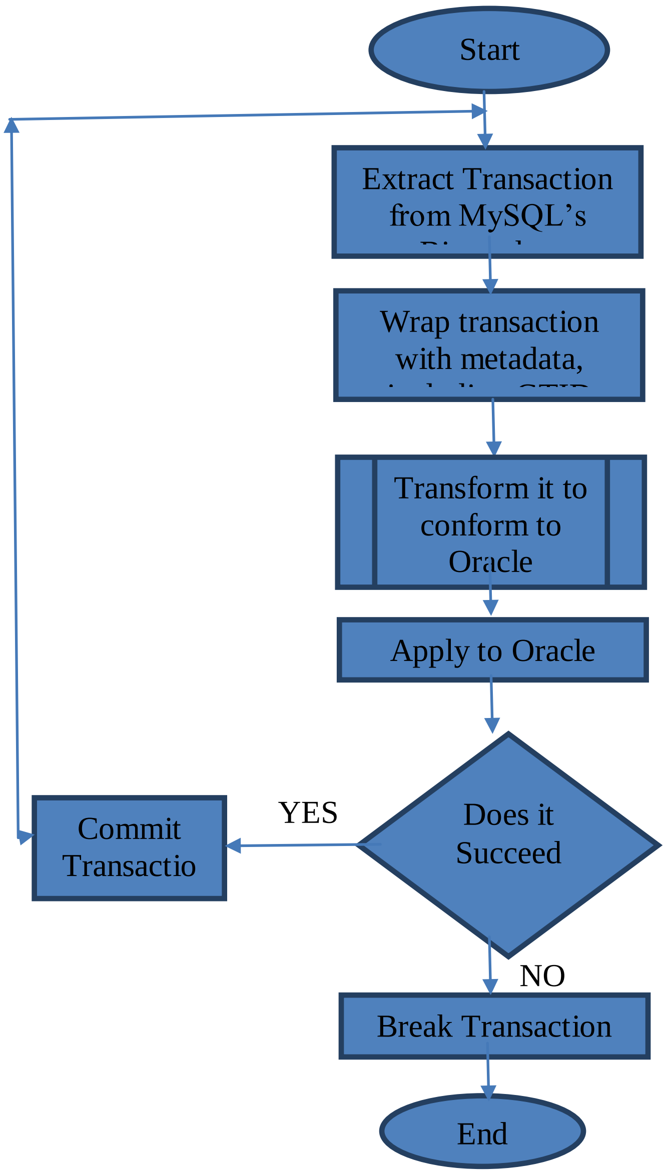


Figure3.2: Framework to Replicate from MySQL to Oracle

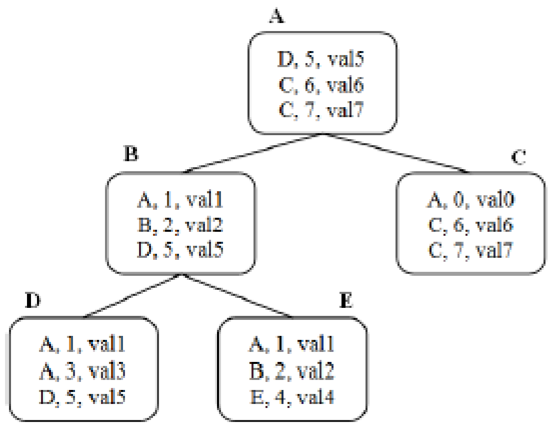


Figure 3.3: Coterie with version examples

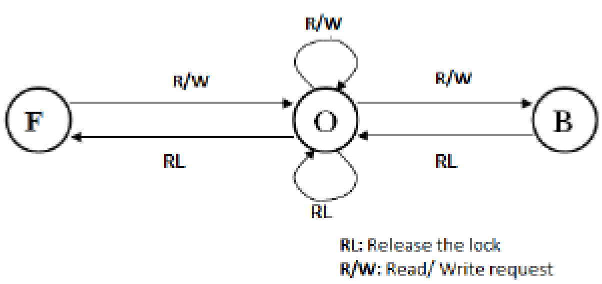


Figure 3.4 State of a node in a coterie

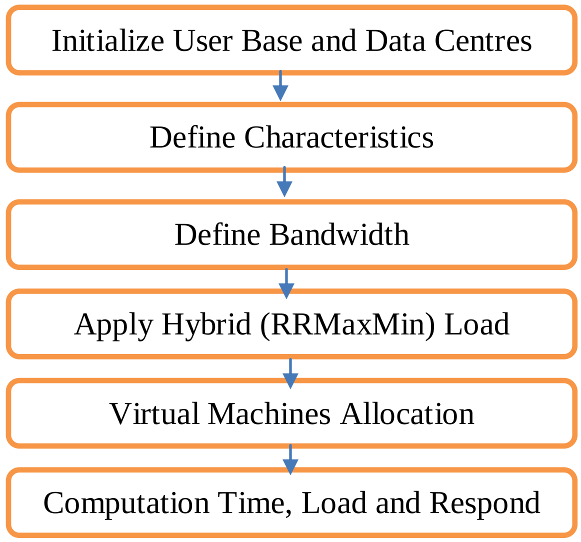


Figure 3.5: Flow step for the Hybrid Approach

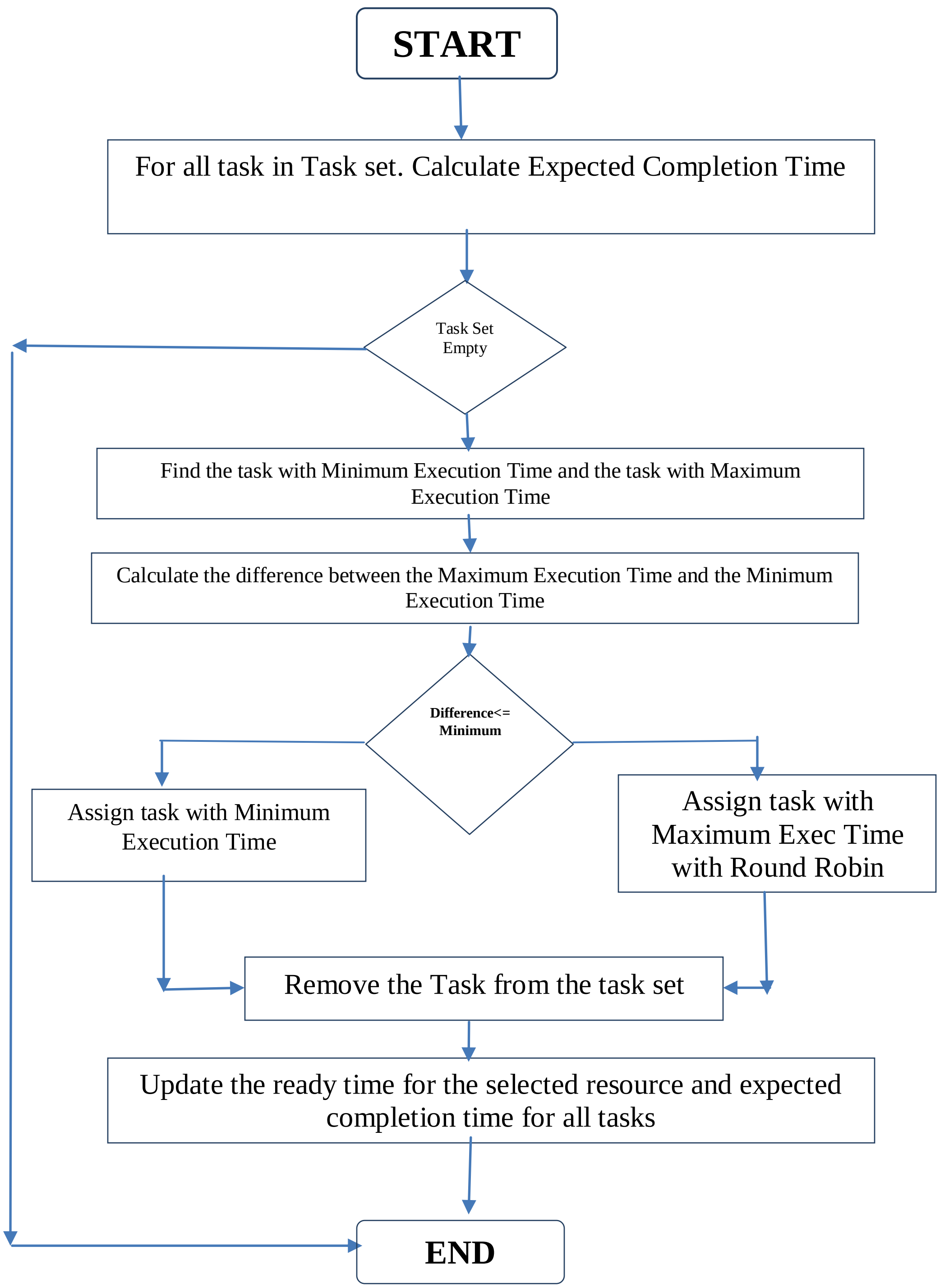


Figure 3.6: The Flowchart of the algorithm for RRMaxMin

Table 4.1: Two Virtual Machines (VM) built on X series server

Component	**Server 1**	**Server 2**
Host	Mysql-stg	Tungsten2
Role	Master	Slave
Ip	172.17.4.105	172.17.4.106
Port	3306	1526
Platform	Window server 2003 32 bit	Linux Ubuntu 32 bit
Database	MySQL 5.1	Oracle 11g
Tungsten Replicator	5.2.1	5.2.1
User	Tungsten	Accounts-places
Password	Root2345	Accounts
Environment	Intel Core i5-2410M 2.3Ghz CPU, 8 GB RAM, 160 GB HD	Intel Core i5-2410M 2.3Ghz CPU, 4 GB RAM, 160 GB HD

Table 4.2 Test Results

Scenario	**Tests Results**
Ability to transform data from MySQL to Oracle without loss of data.	Successful
There were changes in the data on the server 1 and no change on the server 2 after the components on server 2 are active, the data transactions that occur on the server 1 are replicated on the server 2	Successful
After the components on server 1 are active, the data transactions that occur on the server 2 are replicated on server 1	Successful

Table 4.3 Comparison between the two models

CHARRON-BOST MODEL	**NEW MODEL BASED ON TUNGSTEN REPLICATOR**
Very efficient in performing homogeneous replication	Built for Heterogeneous replication
Uses a strong consistency approach	Uses a load balancing consistency approach.
Uses load balance method at the beginning of the model	Uses load balancing to enforce consistency of the replicated data
Can be used for different homogeneous database system	Can be used for different heterogeneous database system

Table 4.4: User base configuration

Name	Regions	Request/Hr	Size	Peak/Hr	Peak Users	Normal Users
UB1	0	600	100	1-3	1500	100
UB2	1	1600	1000	3-5	500	1000
UB3	2	2000	1000	5-7	500	1000
UB4	3	1000	1000	7-9	1600	600
UB5	4	1000	1000	9-11	2000	700
UB6	5	7000	300	11-12	2000	800

Table 4.5: Datacenter Configuration

Name	R	Cost Vm/hr	Memory Cost/s	Data Transfer/Gb	Speed	H/w Units
DC1	0	0.1	0.5	0.1	10000	2
DC2	4	0.1	0.8	0.1	10000	3

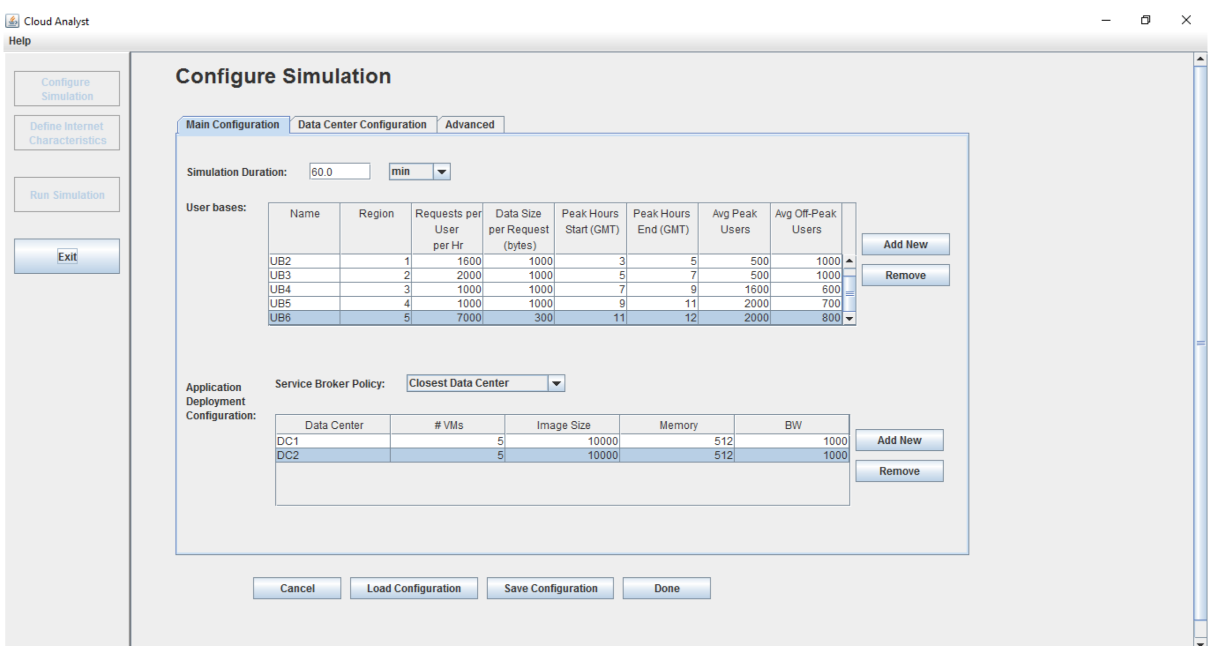


Figure 4.1: The Cloud Analyst Configuration Page

Table 4.6: Other simulation parameter

Parameter	Value
User Grouping factor in User Base	1000
Request Grouping Factor	10
Executable instruction length/request	100
Load Balancing Policy	Round Robin, Equally Spread Current Execution, Throttle and RRMaxMin
Simulation Duration	60.0 min
VM Image Size	10000
VM Memory	512
VM Bandwidth	1000
Data Center Architecture	X86
Data Center OS	Linux
Data Center VMM	Xen

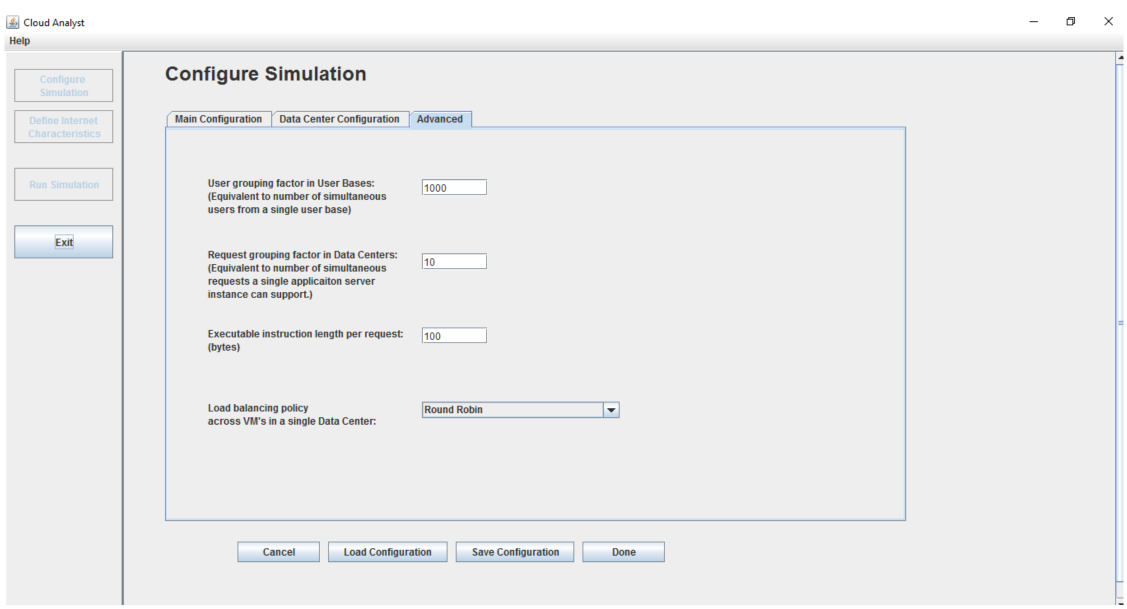
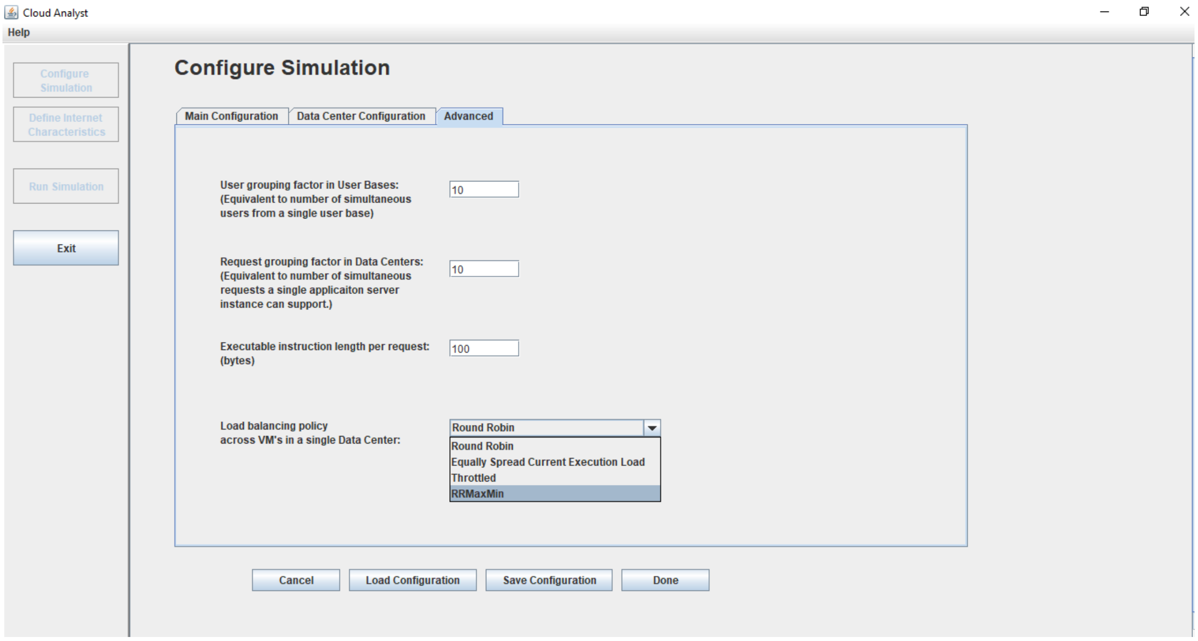


Figure 4.2: The Advanced Configuration Page



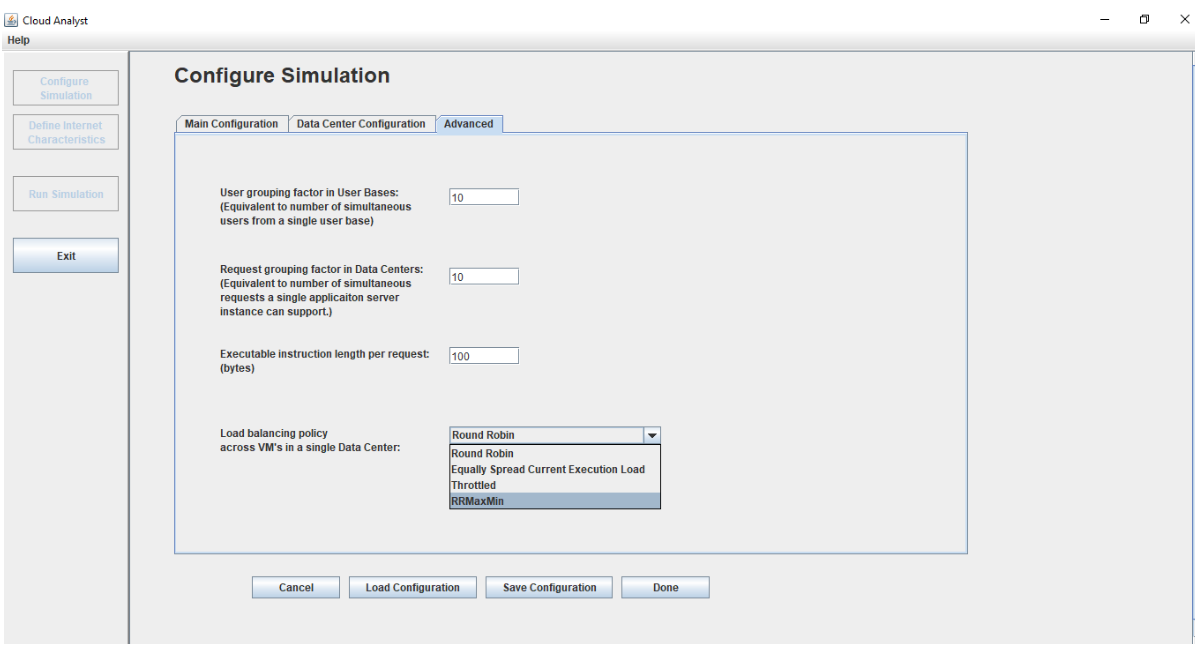


Figure 4.3: The Advanced Configuration Page showing the added RRMaxMin Algorithm

Table 4.7: Overall response time using different policies

Parameter	Round Robin	Equally Spread	Throttle	RRMaxMin
Total Response Time	227.19	227.09	226.18	226.16
Data Center Processing Time	1.47	1.38	0.49	0.56

Table 4.8: Data center request servicing time using different policies

Data Center	Round Robin	Equally Spread	Throttle	RRMaxMin
DC1	1.45	1.36	0.49	0.56
DC2	1.76	1.61	0.61	0.63

Table 4.9: Total Data transfer cost using different policies

Cost	Round Robin	Equally Spread	Throttle	RRMaxMin
DC1	562.36	562.36	562.36	196.56
DC2	67.05	67.05	67.05	23.53

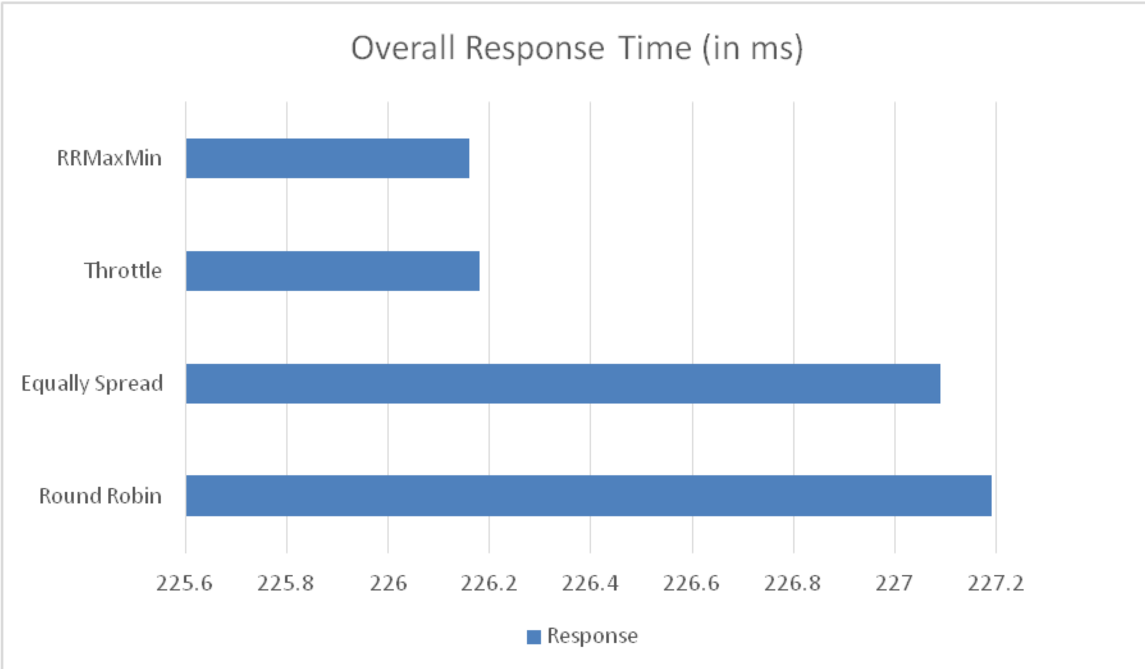


Figure 4.4: Overall Response Time of Load Balancing Algorithms

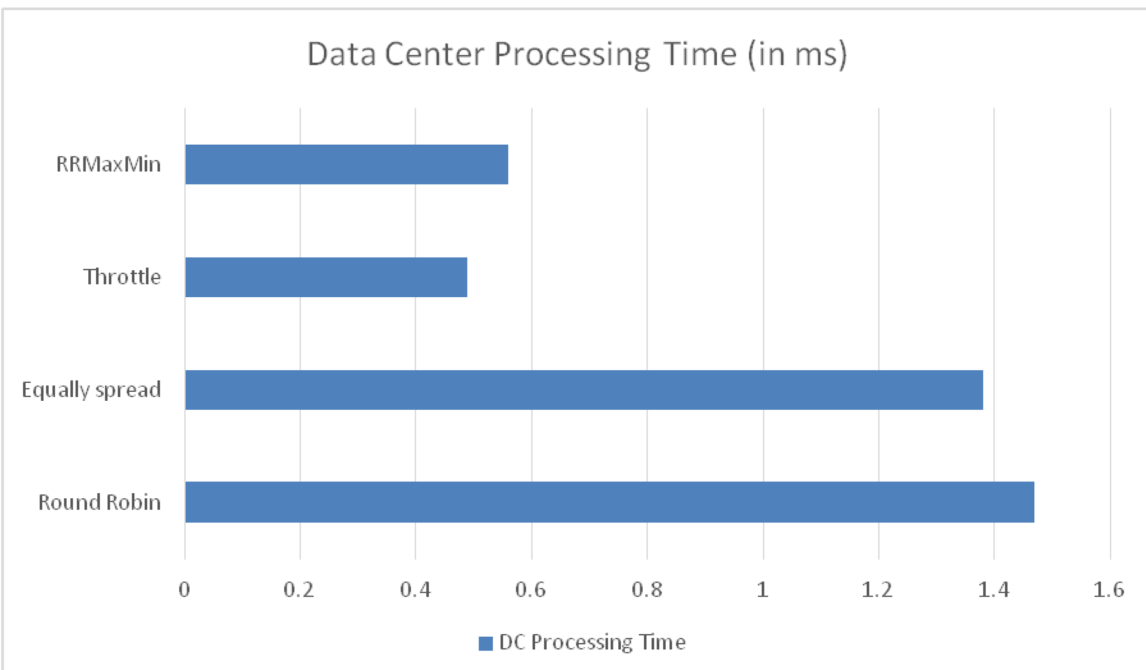


Figure 4.5: Processing Time of Data Center using Load Balancing Algorithms

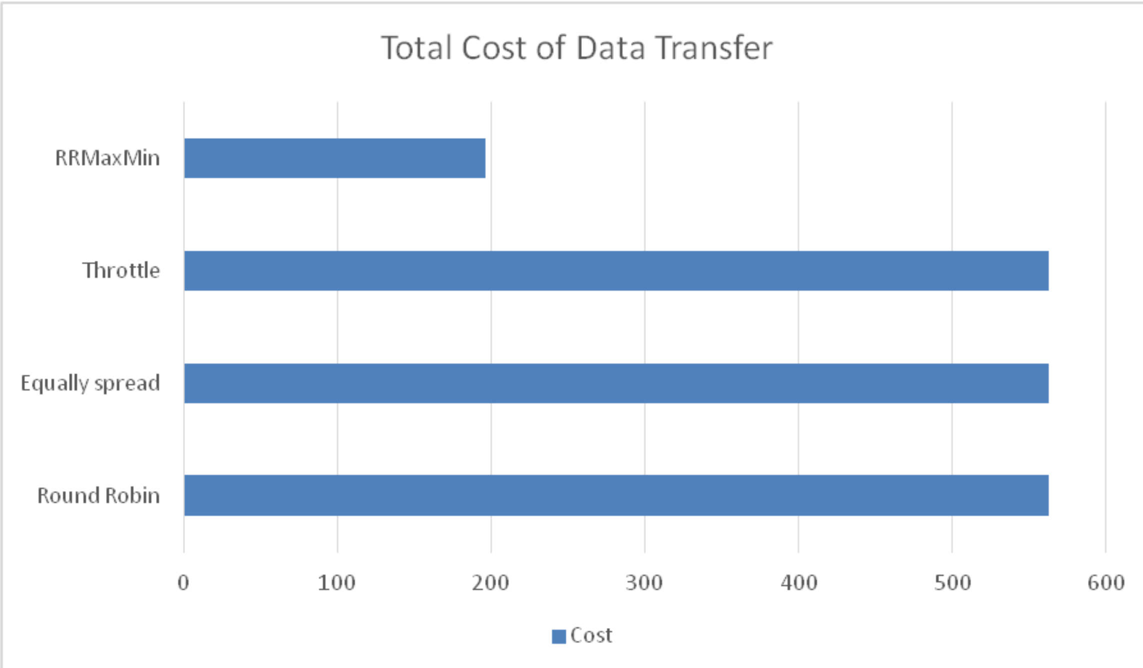


Figure 4.6 Total Cost of Data transfer of Load Balancing Algorithms

PROFILE OF AUTHORS

Dr Aremu Dayo Reuben holds a doctoral (Ph.D.) degree in Computer Science from University of Zululand, KwaDlangezwa, South Africa, Master of Science (M.Sc.) degree in Mathematics with specialization in Computer Science and Bachelor of Science (B.Sc.) degree in Mathematics, both from University of Ilorin, Ilorin, Nigeria. He started his working career in November, 1989 as Education Officer II under Kwara State Education Management Board, Ilorin, Senior Programmer at the Department of Mathematics/Computer Science University of Ilorin in the year 2000 and presently a Senior Lecturer at the University of Ilorin, Ilorin, Kwara State, Nigeria. He is a member of Nigeria Computer Society (NCS), and Computer Professional of Nigeria (CPN). As a Computer Scientist, his teaching and research interest are Software Engineering, Data Structure and Algorithm, Grid Computing, and Information Security. His current research focus is Big Data Analytics. He has published some papers in both Local and International journals in his area of research interest. He has supervised several Undergraduate, Master, and Doctoral research students, and taught many computer Science courses at both Undergraduate and Postgraduate levels during the past decade of his teaching career. He is married with children.

Mr. Abiodun Moses Kazeem holds a Bachelor's Degree (B.Sc. (ed.)) in Physics from the University of Lagos, Master of Science (M.Sc.) degree In Computer Science from University of Ilorin and presently rounding up his Ph.D. programme at the University of Ilorin, Ilorin, Nigeria. His research interest are Software Engineering, Data Structure and Algorithm, Grid Computing, Cloud Computing and Big Data Analytics. He has published some papers in his area of research interest.