ISSN 2278-9723
DOI:10.31838/ijccts/08.01.01

Research Article

Workstation Cluster's Hadoop Distributed File System Simulation and Modeling

IBRAHIM R
Electronics Engineering Department, University College of Science and Technology, Khan Younis, Palestine
E-mail: engreme@yahoo.com
Received: 25.11.19, Revised: 27.12.19, Accepted: 15.01.20

ABSTRACT

On desktop PCs, the capacity of the hard disks is increasing. Therefore, this study simulated and modeled Hadoop distributed file systems on a workstation cluster. The main aim was to determine how effective and feasible it would be to exploit idle computational storage. Architecturally, the proposed model was that which relied on HDFS (Hadoop Distributed File System). Also, CPN tools were used during model implementation. Hence, the tools constituted CPN ML programming language and Colored Petri Dish Nets. To ensure that the availability of the workstations was characterized within the model, the data collection process occurred in a computer lab for about 40 days. Indeed, findings demonstrated that the HDFS's source code could be modified and the number of replicas specified to ensure that the process of reading and writing files is reliable, even in the wake of failures that could arise when computers in the cluster of workstations are turned on and off.
Keywords: cluster of workstations, Colored Petri Nets, Hadoop, distributed file system, simulation, modeling

INTRODUCTION

In the contemporary world, hard disks have increased in the storage capacity, such that their storage comes in terabytes, rather than gigabytes [1]. As a result, some unused space exists in most of the institutions' workstations [2]. With the resultant state reflecting some form of resource wastage or underutilization, there has been the establishment of distributed file systems, which combine software and hardware in the available storage capacity's network [3]. Of importance to note, however, is that a challenge comes when it reaches a level of designing the systems, especially because any failure in hard disk operation could prompt the shutdown of the desktop computers. The disk failure could also result in applications stoppage and network collapse [4]; hence, unreliable service provision to the end-users or customers [5]. The aim of this study was to determine how feasible it would be to simulate and model how an HDFS-based distributed file system would behave in large workstation clusters.

METHODOLOGY

The initial step involves the use of colored Petri nets to model the HDFS in workstation clusters. Also, this modeling involved the use of CPN tools or modeling language, especially because the CPN approach is graphical oriented, combining its strength with the programming language in CPN ML towards system verification, simulation, specification, and design.

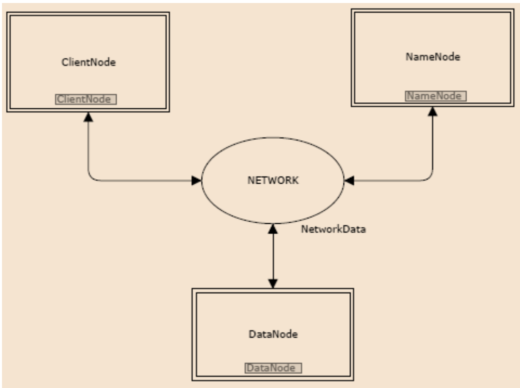

Figure 1: The use of colored Petri nets to model the HDFS in workstation clusters.

The next step constituted the ClientNode modeling process. Indeed, the role of the ClientNode was expected to involve a random generation of requests to write and read files for users. Furthermore, he ClientNode was expected to play the role of stimulating user interface programs interacting with Hadoop file systems.

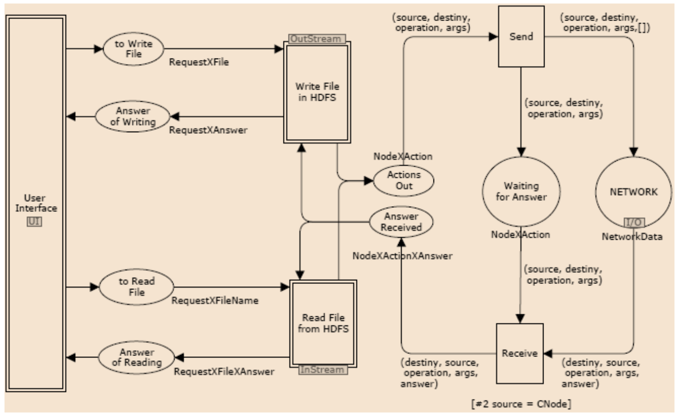

Figure 2: The ClientNode modeling process

The methodological approach proceeded to focus on the namenode modeling. Indeed, the role of the namenode was expected to constitute the server's logic implementation, with the role of the server's logic lying in the File System Namespace management. For the case of the FSN color set, this study developed it in the form of a record with the collected metrics, the list of replicated blocks, the priority queue of the blocks requiring replication, the cluster's all DataNodes' bidimensional map, the BlockInfo list, and the FileInfo list.

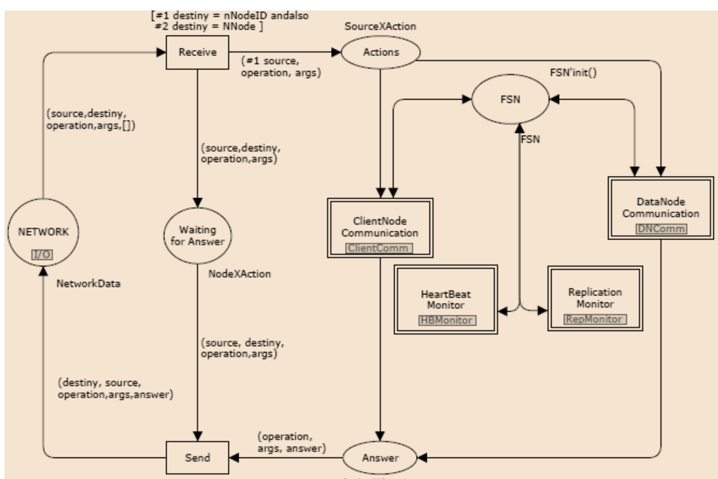

Figure 3: The namenode modeling

The final stage involved DataNode modeling. In this study, the development of the DataNode module was designed in such a way that it would receive and send messages with answers and actions to other nodes. An example was that in which, if the ClientNode send messages to the module, these messages would permit the writing of new or the reading of store-blocks.

Regarding considerations concerning the availability of workstations, the selected HDFS would split files to form blocks, which would, eventually, be stored in the workstation clusters' various nodes.

RESULTS AND DISCUSSION

In the findings, the main objective was to assess the modeled system's behavior. Indeed, several simulations were performed. With the methodological approach that was adopted described above, there was a configuration of several parameters. Examples of these variables included probabilities of hard disk erasing in case of a reset of the workstation, the maximum and minimum values required for the random generation of the file sizes, the size of the blocks, and the maximum and minimum values required for the random generation of the DataNodes' initial free space. Other parameters included the end hour, the initial hour, the number of hosts for each rack, the number of racks, the number of data centers, the number of requests to be read, the number of requests to be written, and the replication factor, with fault tolerance gained after gradually increasing the replication factor.

The figure below demonstrates the simulation outcomes that were obtained. Particularly, the figure shows the proportion of the request for reading and writing files answered successfully, especially with an increase in the replication factor. From the findings, it is evident that there was an increase in the successful answers of reading. However, the successful answers of writing decreased.

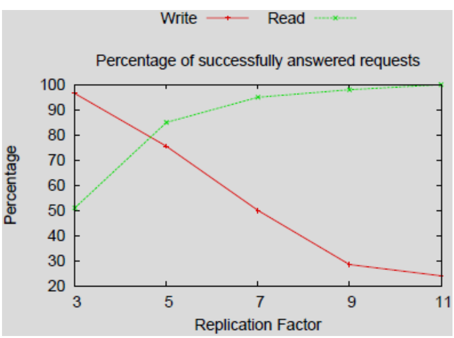

Figure 4: The rate of satisfactory responses

This study established further that there was a decrease in the writing requests' reliability due to the HDFS source code-implemented constraint. The constraint arose in such a way that the ClientNode rejected the pipeline that the NameNode had sent, especially if there was interruption in the pipeline and also if one or more DataNodes were unavailable. In such scenarios, ClientNodes would go back to the NameNodes to establish new replica location lists for the blocks. With an increase in the pipeline length causing an increase in the probability of establishing not-connectable pipelines, this trend would explain why the percentage of the writing process was low.

To achieve the writing process' good readability therefore, especially with chaotic environments leading to the aforementioned constraint, there was a need for relaxation. In chaotic environments, there tends to be a random change in the status of machines. To respond to this compromised state, the proposed model was modified in such a way that rather than pipeline rejection in situations where one or more DataNodes were out of service, the ClientNodes would accept them on the condition that the DataNodes available for block copying exceed 1, which was the specified threshold in the simulation study. With a replication strategy developed in HDFS, this modified procedure did not affect the reading process significantly. Following the aforementioned modification, there was an increase in the rate of writing requests' successful responses. At the same time, this implementation of the modified model did not pose a significant impact on the reading process.

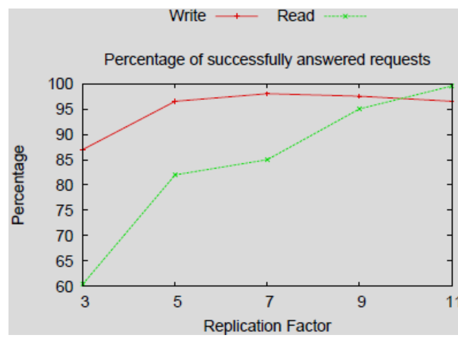

Figure 5: Outcomes after implementing the modified model

CONCLUSION

In summary, the main purpose of this investigation was to establish a Colored Petri Dish framework. This model was poised to test and explore new configurations, as well as some of the other strategies exhibiting different features from those that the HDFS system implements. In the proposed model, the study involved the simulation. The purpose of the simulation activity lay in the criticality of assessing the system's behavior, given the workstation's clusters. With uniform and binomial distribution functions incorporated, there was the generation of the idle time, given an hour's number of powered-on computers. From the findings, the simulation outcomes demonstrated that the critical constraint determined the pipeline rejection or acceptance, with the NameNode returning the rejected pipeline. Given the dedicated machines' clusters, this framework was found to work well. However, if responses exceeding 95% were to be achieved, which would imply a high percentage and satisfactory responses, the proposed model would require modification in which the critical constraint would be relaxed. Upon this modification, specific findings demonstrates that the proposed model is better placed to perform well relative to the provision of satisfactory responses for the requests of reading and writing files in NDDE (non-dedicated distributed environment) settings. Also, the superiority of the proposed modified framework was confirmed upon being simulated in chaotic environments relative to the availability of workstations. In the modeled system, however, a good quality service would be achieved if the replication of the data blocks is done several times.

REFERENCES

1. Ratzer, A.V., Wells, L., Lassen, H.M., Laursen, M., Frank, J., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K.: Cpn tools for editing, simulating, and analysing coloured petri nets. In: Applications and Theory of Petri Nets 2003: 24th International Conference, ICATPN 2003, Springer Verlag (2003) 450-462
2. Borthakur, D.: The Hadoop Distributed File System: Architecture and Design. The Apache Software Foundation. (2007)
3. Novaes, R.C., Roisenberg, P., Scheer, R., Northeet, C., Jornada, J.H., Cirne, W.: Non-dedicated distributed environment: A solution for safe and continuous exploitation of idle cycles. In: In Proceedings of the Workshop on Adaptive Grid Middleware. (2003) 107-115
4. Anderson, D.P., Fedak, G.: The computational and storage potential of volunteer computing. In: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid. CCGRID '06, Washington, DC, USA, IEEE Computer Society (2006) 73-80
5. Bolosky, W.J., Douceur, J.R., Ely, D., Theimer, M.: Feasibility of a serverless distributed _le system deployed on an existing set of desktop pcs. SIGMETRICS Perform. Eval. Rev. 28(1) (June 2000) 34-43
6. Zhang, W., Liu, H., Al-Shabrawey, M., Caldwell, R., Caldwell, R.Inflammation and diabetic retinal microvascular complications(2011) Journal of Cardiovascular Disease Research, 2 (2), pp. 96-103. DOI: 10.4103/0975-3583.83035