International Journal of Advanced Culture Technology Vol.9 No.1 218-223 (2021)
DOI https://doi.org/10.17703/IJACT.2021.9.1.218

IJACT 21-3-30

Enhanced and applicable algorithm for Big-Data by Combining Sparse Auto-Encoder and Load-Balancing, ProGReGA-KF

[1]Hyunah Kim, [2]Chayoung Kim

[1, 2] College of Liberal Arts and Interdisciplinary Studies, Kyonggi University,
154-42 Gwanggyosan-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do 16227, Korea
hyuna486@kgu.ac.kr[1], kimcha0@kgu.ac.kr[2]

Abstract

Pervasive enhancement and required enforcement of the Internet of Things (IoTs) in a distributed massively multiplayer online architecture have effected in massive growth of Big-Data in terms of server over-load. There have been some previous works to overcome the overloading of server works. However, there are lack of considered methods, which is commonly applicable. Therefore, we propose a combing Sparse Auto-Encoder and Load-Balancing, which is ProGReGA for Big-Data of server loads. In the process of Sparse Auto-Encoder, when it comes to selection of the feature-pattern, the less relevant feature-pattern could be eliminated from Big-Data. In relation to Load-Balancing, the alleviated degradation of ProGReGA can take advantage of the less redundant feature-pattern. That means the most relevant of Big-Data representation can work. In the performance evaluation, we can find that the proposed method have become more approachable and stable.

Keywords: *Big-Data representation, Machine-Learning, massively multiplayer online architecture, Load-Balancing, Sparse Auto-Encoder.*

1. INTRODUCTION

In the distributed massively multi-player online architectures in the IoTs [1], there have been main uncompromising issues such as managing millions of end-users concurrently and providing consistency-guarantees and resilience for inter-mediate communications on the load-balancing network. Communications and interactions between the huge numbers of end-users might have occurred and developed considerably comparing to the number of end-users, mainly responsible for evoking Big-Data representation. There have been many previous methods for considering the applicable Big-Data involved the distributed massively multiplayer online architecture. One of the main issues of the previous ones is how can we take advantage of the historical Big-Data representation. That means the massively multiplayer online architectures in Big-Data world [2] require new criterions for Load-Balancing because the cumulative huge amount of Big-Data representation might be helpful if Big-Data representation can have demonstrated classifying the most relevant and less redundant form. Based on Big-Data representation by Machine-Learning [3], such as Sparse Auto-Encoder [4], the freer move of the end-users' Avatar in the 3-D (Dimension), the more possible the composition of Load-Balancing severs' interactions to attempt to reduce erroneous decisions such as hotspots [5], around which the end-users are more intensive than in any other

Manuscript received: February 18, 2021. / revised: February 25, 2021 / accepted: March 3, 2021
Corresponding Author: Chayoung Kim
Tel:+82-31-249-9509, Email: kimcha0@kgu.ac.kr
College of Liberal Arts and Interdisciplinary Studies, Kyonggi University, Korea

3-D world, occurring performance degradations of servers. In Machine-Learning, if the severs have communized each other in an affordable size of a group of Avatars, then they have learned each other by interactions and Big-Data representation between them have been brought into the Big-Data world. We consider the ongoing issue, how we can deliver the presentation of the server manageable powers by the accumulated Big-Data representation. For that reason, the server over-load algorithms have the capability of analyzing the feature-pattern of interactions between end-users by preprocessing and adapting the considered feature-pattern for Load-Balancing. There have been many previous Load-Balancing methods in the distributed massively multi-player online architectures for preventing the hotspots from reducing the quality of the 3-D world and considering the degradation of the inter-mediate communication over-load. However, there are not many works on that Big-Data feature-pattern representation. Therefore, our research work is for the qualification of the manageable Load-Balancing because the issues of loads and the demands of end-users in 3-D world result in Big-Data representation, making a meaningful feature-pattern.

The goal of this work is to learn the feature-pattern of Load-Balancing by Avatar's 3-D world based on the Big-Data representation gained from the historical loads accumulated. In the 3-D world, learning the over-head feature-pattern in advance can benefit Load-Balancing, preventing the degradation of the quality of the 3-D world and enable the severs balancing the huge loads. In this research, the historical loads by the suggested 3-D simulation have been exploited as the research dataset. The provided loads have the ensuing properties: the number of cells per each region, assigned to each server, and how many cells it is correspondence with. To exploit the following properties, we proposed a new approach in accordance with feature-pattern Big-Data representation. In this approach, we combine a Sparse Auto-Encoder [4] as Big-Data reconstruction for achieving a better representation of data inputs and a well-known method, ProGReGA [5]. The Sparse Auto-Encoder [4] can makes the less redundant and the most relevant of data inputs. Therefore, the preprocessed feature-pattern by the Sparse Autoencoder [4] can further seize the most relevant feature-pattern from the historical Big-Data. The proposed method is compared with the representation-free ProGReGA-KF [5] on the own simulation environment. We find out that the proposed method has become more approachable and stable than the previous ProGReGA-KF [5] with the less number of end-users' migrations between inter-mediate servers. The rest of this research is organized as follows. The related work is described in Section 2. Section 3 shows the details of the proposed method and Section 4 explains the experimental results and comparison. Finally, we conclude this work in Section 5.

2. RELATED WORKS

The Internet of Things (IoT) [1] is gaining ground in the field of wireless telecommunications in a rapid change. The base foundation of the IoT is the prevalent presence of various objects or things — such as sensors, mobile phones, radio-frequency-identification (RFID) tags [6],—which, through unique addressing methods, can interact with each other and cope with them to attain common goals. In [7], the researchers combine the Group Method of Data Handling method based on Evolutionary Algorithm (EA-GMDH) and Phase Space Reconstruction (PSR) for predictable host over-loads. The prediction of EA-GMDH and PSR is highly related to the weight parameters, as the evolutionary method is a stochastic global search algorithm which might fall into local optima. Echo State Networks (ESN) is a recent development in the field of Recurrent Neural Network (RNN) [6] and it leads to a simple, fast and constructive method for a supervised training of RNN. The main idea of ESN is to take advantage of a RNN as a supplier of important dynamics from which the desired output, which is different from the predicted output is combined. The architecture selected from the Machine-Learning, RNN is to consider the recurrent layer as a large architecture of non-linear transformations of the input Big-Data and decouple the hyper-parameters inside and outside the one.

Unsupervised feature-pattern Machine-Learning [8] consults a class of Machine-Learning techniques [9], which developed in a rapid change since 2006, where many stages of non-linear Big-Data representation processing in the hierarchical architectures are taken advantage of feature-pattern classification. An unsupervised feature-pattern learning technology has been successfully exploited in many research areas, such as hand-written digit-images recognition, visual objects classification and nature language processing

[6]. After generating the deep neural network with unsupervised feature-pattern learning methods [e.g., Auto-Encoder [4] and Restricted Boltzmann machines (RBM) [6]], the weights of the hyper-parameters have been starting in a parameter space than if they had been randomly generated. Because the Deep Neural Network (DNN) can also be regarded to perform feature-pattern learning, since they learn Big-Data representation of the input data-sets at the hidden layers, subsequently exploited for regression at the output layer or feature-pattern classification.

One of Load-Balancing methods for distributed massively multi-player online architectures in the IoTs have been taking into account for the use of up-load band-width of the distributed servers [5]. The method in the research [5] divided into three phases and proposed different methods for the Load-Balancing phase and ProGReGA [5] presented the lowest over-head of all, while ProGReGA-KF [5] presented the second fewest migrations of end-users between distributed servers, as well as the second lower over-head and a fair distribution.

3. THE PROPOSED ALGORITHM

In this part, we describe our proposed model in Figure 2 based on Figure 1 [5]. Like the well-known method ProGReGA-KF [5], end-users interacting with each other might be connected to the same Load-Balancing server. If two Avatars of two different end-users could be distant from each other, both of the end-users might be interacting with a third Avatar between them. So, it is mandatory to consider how many pairs of end-users and which of them would be divided into the different Load-Balancing servers. The proposed approach is exploiting Big-Data representation by Sparse Auto-Encoder [4]. The over-head feature-pattern obtained from the past loads could be a criterion of the load balancing. In the proposed approach, like ProGReGA-KF [5], the main aspect in the method is focusing each local server's Big-Data information that generated the Load-Balancing process and the neighbors and accumulated Big-Data information on each sever. In this respect, we attempt to map the original Big-Data space to a new space, more suitable for maximizing the relevant feature-pattern as a good feature-pattern Big-Data representation. The Sparse Auto-Encoder [4], one method to automatically learn feature-pattern from unlabeled Big-Data. After the preprocessed feature-pattern layer by the Sparse Auto-Encoder [4], the proposed algorithm has the simple phases likewise ProGReGA [5] for Load-Balancing these local regions and partitioning for the over-head reduction.

Algorithm: Local regions selection

```
local_group <-{R} #R is Region
local_weight <-w_t(R) #Regions' weight
local_capacity<-p(s(R)) #Power of Server of Region
average_usage<-local_weight/local_capacity
while average_usage > max(1, U_total)do # U_total s SystemUsage
    if there is any not selected region neighbor to one of local_group
    Then R<- not selected region neighbor to one of local_group, with smallest u(s(R)) #Region's resource usage
    else if there is any empty region
    then R<- empty region with highest p(s(R))
    else stop when no more regions to select
    end if
    local_weight<-local_weight+w_t(R)#Regions'weight
    local_capacity<-p(s(R)) #Power of Server of Region
    average_usage<-local_weight/local_capacity
    local_group <- local_group ∪ {R}
end while
Running the local balancing algorithm with local_group as input
```

Figure 1. The definition based on [5] for the proposed algorithm

```
Algorithm: ProGReGA with Sparse Auto-Encoder
Input: Each_Cell in all regions, C = {1, 2, … n},
Output: Region_List based on Data Reduction
1. E=Encoded(Input, activation='relu')
2. D=Decoded(E, activation='sigmoid')
3. AC=Auto-Encoder(Input, D)
4. weight_to_divide <-0
5. free_capacity <-0
6. for each region R in Region_List do
7.     weight_to_divide <- weight_to_divide + w_r(R)
8.     free_capacity <- free_capacity + p(s(R))
9.     Free all cells from R temporarily
10. end for
11. Sort Region_List in decreasing order of p(s(R))
12. for each region R in Region_List do
13.     weight_share <-weight_to_divide × p(s(R))/free_capacity
14.     while w_r(R)<weight_sharedo
15.         if there is any cell from R neighboring a free cell then
16.             R <- R U {neighbor free cell with the highest Intc(AC)}
17.         else if there is any free cell then
18.             R <- R U {the heaviest free cell}
19.         else
20.             Stop no more free cells
21.         end if
22.     end while
23. end for
```

Figure 2. The proposed Algorithm

4. PERFORMANCE EVALUATION

In this research, the proposed method, combining Sparse Auto-Encoder [4] and ProGReGA-KF [5] is compared with the well-known method ProGReGA-KF [5]. For the performance evaluations, we should consider a heterogeneous server system, simulate the proposed method on a grid-square cells, select a grid-cell with the lowest interaction in the smallest grid-cell clusters by the over-load sever and transfer or migrate the over-load of the selected server to the least over-load one. However, the well-known ProGReGA-KF [5] has forced an uneven distribution of Avatars in the virtual 3-D world environment, putting to test the Load-Balancing method. The purpose of the well-known ProGReGA-KF [5] was how to take into account of hotspot form the local regions and reduce the distributed over-head. On the other hand, in a 3-D virtual world, "sophisticated degradation" is better of the reality [5]. Therefore, in the performance evaluations, we should consider the normal 3-D world environments. As the simulation of comparisons based on Unity-3D [7] begins, they have been starting to move in accordance with the random model architecture [5]. The simulation' environment is resulting in 2-D (dimensional) spaces, divided by the grid-cell, belonging to some regions. It can be migrated into another regions for Load-Balancing. We can consider in the average weight of a local region on the criterior of proportional Load-Balancing and the number of immigrations by the defined formula in Chapter 3. Also, we can calculate how many differences based on the time for a pre-process between the proposed algorithm, combining Sparse Auto-Encoder [4] and ProGReGA [5] and the well-known ProGReGA-KF [5] in terms of Big Data sets.

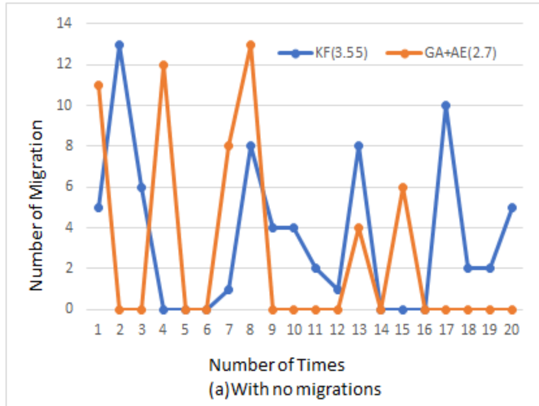

Figure 3(a). The comparison of the proposed method (A-ENN+ProGreGA) with the well-known method (ProGReGA-KF)

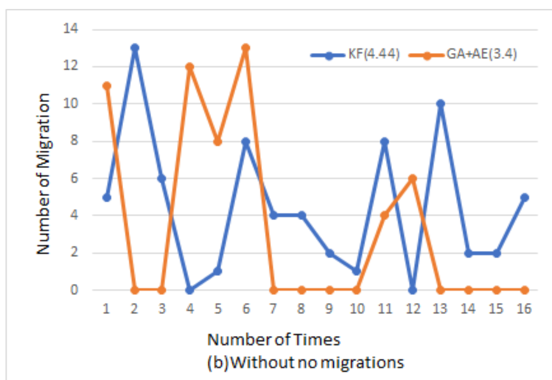

Figure 3(b). The comparison of the proposed method(A-ENN+ProGreGA) with the well-known method (ProGReGA-KF)

Figure 3 shows the one of important results that the proposed method has some advantages with regard to pre-process of Big-Data sets. With the reasonable size of Big-Data sets, Load-Balancing can be possible to be handled during the 3-D world sessions. With respect to Big-Data sets, pre-process becomes significantly important. If the outputs of pre-process are given in a reasonable time period to the Load-Balancing management, still having information useful and meaningful, then Load-Balancing is served considerably to end-users during 3-D world session without letting the end-users know. In accordance with the Load-Balancing pre-process depends on the specific 3-D world such as real-time virtual 3-D in which an end-user constantly migrates between Load-Balancing servers, causing delay and hinder of the interaction between end-users, the proposed algorithm can be regarded as a more stable and achievable method. The proposed pre-process can handle the important representation from Big-Data by managing and simplifying in the proportion to the reasonable size of the accumulated Big-Data sets.

Moreover, we notice that it is considerably scalable than the well-known ProGREGA-KF [5] in terms of Big-Data representation of Sparse Auto-Encoder [4]. In comparing Figure 3(a) and Figure 3(b), we can figure out that in terms of no migrations, the proposed method is much more realistic. The reason is that the Load-Balancing part of the proposed method follows the development of ProGReGA. In the evaluation of the performance in [5], ProGReGA was mentioned that it had the lowest over-head as it was designed to create the most realistic local regions, causing the number of migrations meaningful.

5. CONCLUSION

We suggest that the method combining Sparse Auto-Encoder [4], one of the most used Big-Data representation, and ProGREGA [5], the most well-known Load-Balancing method for the distributed massively multi-player online architectures in the IoTs. Big-Data information acquired from the past huge loads of Load-Balancing servers based on the demands of end-users are getting formed into Big-Data sets. The proposed method attempts to overcome consuming a huge amount of pre-process time in terms of Big-Data sets. In accordance with an agent' learning, the feature-pattern in advance can be benefited by Load-Balancing, the proposed method can take advantage of these mentioned properties by Big-Data representation of Sparse Auto-Encoder [4]. The proposed method is compared with the representation-free one of ProGReGA-KF [5]. The proposed pre-process can handle the representation from Big-Data sets by managing and simplifying in the proportion to the manageable size of accumulated Big-Data sets. Because pre-process is given in a manageable and reasonable time to Load-Balancing, managing the balance as the reaction of Load-Balancing servers is served considerably to end-users during 3-D world session without letting the end-users know. Therefore, the proposed method has become more manageable and scalable than the representation-free one ProGReGA-KF [5] in terms of Big-Data sets [2]. If it is designed precisely to create the more realistic local regions, causing the number of immigrations manageable, the proposed algorithm can be considered more realistic and achievable.

REFERENCES

[1] D. Giusto, A. Iera, G. Morabito, L. Atzori (eds) (2010) The Internet of Things. Springer, New York. ISBN 978-1-4419-1673-0, DOI 10.1007/978-1-4419-1674-7

[2] IBM, Bringing Big Data to the Enterprise. *http://www-01.ibm.com/software/data/bigdata/*. Retrieved January 21, 2020

[3] P. Wei, Xin B. (2020) An integrated autoencoder-based filter for sparse big data, Journal of Control and Decision, Taylor & Fransis, (2020) DOI: 10.1080/23307706.2020.1759466

[4] Andrew Ng, "Sparse Autoencoder", https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf. Retrieved January 21, 2021

[5] C. Eduardo, B. Bezerra, C. Fernando, and R. Geyer, "A load balancing scheme for massively multiplayer online games", Multimed. Tools Appl. (2009) Vol. 45, pp.263–289, DOI 10.1007/s11042-009-0302-z.

[6] RNN, LSTM, https://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/, Retrieved December 21, 2020

[7] Unity3D, *https://unity3d.com/kr/*. Retrieved December 20, 2020

[8] Ian Goodfellow, "Generative Adversarial Networks, NIPS 2016 Tutorial, 2016. Retrieved January 21, 2021

[9] Q. Yang, Y. Zhou, Y. Yu, J. Yuan, X. Xing, S. Du, "Multi-step-ahead host load prediction using autoencoder and echo state networks in cloud", J Supercomput (2015) Vol.71, pp.3037–3053, DOI 10.1007/s11227-015-1426-8.