ISSN 2278-3091

Volume 9, No.4, July – August 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse255942020.pdf

https://doi.org/10.30534/ijatcse/2020/255942020

Dynamic Resource Allocation and Memory Management Using Machine Learning for Cloud Environments

Dipak Raghunath Patil[1], **Dr. Manish Sharma**[2]

[1]Research Scholar, [2] Associate Professor

[1,2]Department of Computer Science & Engineering, School of Engineering & Technology, Suresh Gyan Vihar University, Jaipur India,

[1]patil.59712@mygyanvihar.com, [2]manish.sharma@mygyanvihar.com

ABSTRACT

In today's environment, large data has generated by various platforms, mobile computing is a kind of environment that generates such massive data in recent years. In those cases, heavy resources should be required to manage such data. This research basically proposed dynamic resource utilization using memory management with a large scale file system in the mobile environment. In mobile unnecessary generated cache or such file system generate unnecessary memory overhead of mobile device and it utilized major resources of device unnecessarily. To identify such data block and eliminate from execution will improve device performance and effective memory management. In this paper we proposed effective memory and resource management using machine learning technique and optimization. Some feature extraction techniques have been used from data block and previous pointer. System also reduces the processing overhead when system perform any transaction, the experimental analysis shows the effectiveness of proposed system with machine learning algorithms.

Key words: Machine Learning, Cloud Computing, Resource Matchmaking, Workflow Scheduling.

1. INTRODUCTION

Cloud computing allows users to scale up or scale down their services according to requirements. Cloud computing Formulates the services as a single point of access for the user, and costs are charged for each use. Cloud computing grows Innovation where a pool of assets is linked in and offers grid computing private and public networks the Application technology. Information technology is not application-oriented, and is a service-oriented approach. This does say the Virtualized Cloud consumers' tools. Cloud computing provides dynamic provisioning, such that computers can be distributed to Store data and attach or remove machines which are needed after the workload. Namely, some cloud computing systems Offered by Microsoft, Amazon, Twitter, and IBM. Cloud computing is a knowledge-only information sharing network The network and can devise requirements and their linked data can be accessed from anywhere at any moment.

In this research we proposed effective resource and memory management using machine learning for large scale file system as well as mobile platforms. The main purpose of such investigation achieves two goals in the algorithm introduced. Overheat avoidance: A physical machine's capacity must be sufficient to meet the available resources of all VM's working on it. Otherwise the physical machine is overloaded, and can direct its VM's performance to degrade. Cloud computing evolves into a computing standard, infrastructure as a service has emerged as an important idea in the computer science field. It can abstract the fundamental physical resource such as a CPU, Memory and Storage by referring to this definition and present this Virtual Resource to users in the reserved Virtual Machine. Various Virtual Machines are capable of running on a unique physical machine. Another big problem in cloud computing is the availability of resource management strategy for cloud users. There are two provisions to the cloud computing environment. The goal is to achieve an optimal resource provisioning solution that is the primarily important part of cloud computing. Formulating a best decision that takes into account the market price and the uncertainty of waiting-time to change the trade-offs between on-demand and over-subscribed prices.

2. RELATED WORK

Cloud computing has many benefits, and the many important ones are reduced costs, pre-utilization of various resources, and remote access. Reduces cloud computing costs by reducing the company's capital expenditures in renting of a third - party services physical network. We can easily do so because of the versatile nature of cloud computing gain more cloud provider's services when we need scaling up our company. The remote connectivity gives us the ability to Access load balancers at any time anywhere at. To win the full degree of benefits listed above, the services provided should be distributed in terms of capital optimally suited to cloud operating applications. The next segment addresses the value of the assets allotment.

Integrating cloud service provider utilization and allocation activities limited resources under the cloud setting limit for Reach cloud technology needs. It does require the kinds as well as amount of assets required in order for each application to complete a job for the customer. The allocation and distribution time of the Materials also provide the feedback for an optimal RAS. An ideal one the following requirements should be avoided according to Strategy for resource allocation. In batch and high-performance computing systems, operating systems, real-time computing and more previously cluster and cloud infrastructure management, there is much prior work in resource allocation. Most of the research, however, has centred on how to arrange tasks or jobs once their resource requirements are identified, and Mesos allows users to identify their needs for resources. Users most often do not know the specifications of the tool and find it difficult to determine the efficiency and cost of their selections. Define a computer facility with a pool of common infrastructure (CPUs, memory, power, storage space, mobile servers, etc.). That very facility can be a customer grid, a data center, a supercomputer or a multiprocessing machine running an operating system that supports dynamic resource readjustment. Assume that this facility is used asynchronously by N clients (projects) and that resources can be dynamically reassigned among the ventures, and only one project at a time should use a particular quantity. Define computer facilities with such a pool of common infrastructure (CPUs, memory, power, storage space, mobile servers, etc.). That very facility can be a customer grid, a data center, a computer simulation or a multiprocessing machine trying to run an operating system that supports resource provisioning readjustment. Assume that this facility is used asynchronously by N clients (projects) and that resources can be dynamically reassigned among the ventures, and only one project at a time should use a particular quantity. Below we study some prevailing systems which already developed by existing researchers.

A technique to recognize a drug from its image [1]. Here a person images a tablet from his smart phone. Then, their methodology scans a top quality input images and finds a list of the most appropriate pills to the image of both the query. Obtaining best performance is difficult due to factors such as lighting, phone orientation and presence of a wide range of drugs. A map of inequalities measures the horizontal distance between two images at each pixel. In [2] present a disproportion estimation model which allows an equilibrium to be struck between precision and speed. They pass the given input pair first via a feature extractor, which can measure feature maps at different resolution.

A strategy to assist drivers in locating and billing empty parking spaces depending on the precise use of the parking space. They use several "ground" cameras that are connected via WiFi. In addition, several linked "top-view" cameras that have wide-angle lenses are used to cover the entire parking lot. The ownership of all parking slots is identified from their feeds, and vehicles are tracked until they are parked [3]. Various methodologies for network traffic selection and analysis at intersections [4]. They positioned the camera at an intersection with traffic. Based on each-lane width, speed limits on each side and traffic density, they find which collecting six blocks each second is enough to track a vehicle. They map the lanes on the field of view by hand. It's a very basic program, and easy to use. In this you can click plant picture and it matches the image with the pictures in the database and gives you the most similar image. It should identify only certain plants in the database and related photos in the database [5].

There are primarily three types of nodes used to create a CNN that are in (a) Convolutional layer, (b) Pooling layer, (c) densely integrated layer and one more layer called regularization layer [6]. The Convolutional layer is the core part of CNN that performs most of the computation and intensive lifting on the inputs that includes taking a volume of input combining them by performing dot products (filters) on them and providing reflectance output level. Max pooling is normally added periodically after a time. Its role is to slowly reduce the simplification’s spatial scale which in effect reduces the number of computational parameter in the network. The Fully Connected layer is the final layer but have complete connectivity to all previous layers' installations.

According to [7] it was the first inception model that indicates a sequential stacking of CNN layers is not required. It shows that innovative layer structuring can offer greater efficiency and results. There are network pieces which operate in parallel. Essentially there is a choice of whether to provide a pool operation or a convolutional operation that makes the process parallel at each layer of ConvNet. Two of Google LeNet's key points are that it has used only 9 start-up modules and about 100 layers. Instead of using average pooling layer there are no Completely Linked layers used. The main advantage of this is that 12x less parameter are used than other networks, which is the main reason for speed and efficiency. It was the first inception model that indicates a sequential stacking of CNN layers is not required. It shows that innovative layer structuring can offer greater efficiency and results. There are network pieces which operate in parallel. Basically, every layer of ConvNet has a choice of whether to provide a pool operation or a conversion operation that makes the task parallel [8].

Implemented BPLRU: A Buffer control strategy for Enhancing Random Writing in Flash Storage [9], system proposes a new write buffer management scheme called Block Padding Least Recently Used which greatly improves flash storage random write performance. We’re testing the arrangement using trace-driven experiments and experimental implementation simulations. It information about the network 44% better performance than traditional memory management in cloud and distribution environments.

In FloWaveNet: A compositional Flow for Raw Audio [10], it explains FloWaveNet, a raw audio compression flow-based generation model. FloWaveNet only needs a single maximal loss of probability without any contractual criteria and is essentially parallel because of the flow-based conversion. The model will reproduce the raw audio effectively in real-time with a resolution equivalent to the originally WaveNet and ClariNet.

Based DeepSense system [11]: the GPU-based strong deep convolutional neural architecture for mobile utility devices System introduces our early concept of DeepSense a deep convolutional neural network (CNN) based mobile GPU platform. We first examined the discrepancies amongst server-class and mobile-class GPUs for their architecture, and examined the effectiveness of different optimization techniques such as division divergence exclusion and memory capacity.

It is noted that multi-drone video-traffic can swamp the wireless spectrum conveniently. Strategies such as data compression often provide minimal benefits, because they are unaware of the properties of the system installed on drone, including video processing or monitoring. They're suggesting four approaches to reduce drone-data bandwidth use [12].

To fine-tune it in [13] over a limited training range of photographs collected from an atmospheric point of view. A machine-learning model trained for one task is used as the point of departure for a tutorial on some other task in the transition active learning. Transfer learning increases precision for aerial images without needing a large spectral data training set.

In [14] suggested a method for obstacle avoidance using the stereo camera-generated depth map and the "monocular camera" model approximately distance. Pixels with the same difference are likely to belong to the same entity that is considered to be an on-road entity. On the basis of this, their strategy searches in the vertical direction for pixels of the same difference on the depth diagram. Use of camera - based depth values removes imprecision from monocular camera in distance estimation.

To implement a "motorist-drowsiness identification" technique based on the CNN [15]. It classifies an input image of the driver's window into one of three categories: "natural" (no fatigue), "yawning" and "drowsy." Its technique consists of two main phases. In the first step, "community perceptions and orientation" is performed using "multi-task cascaded convolutional networks" They use "information deconstruction" technique [16] to further minimize latency. This method uses a network of "train" and a network of test. The train network is the broad original network that is learned from the dataset itself. The test network is a smaller and faster network, which learns technologies from the train network's "soft targets."In their configuration, the "slimmed-two-stream" network works as the network of test while the stream-two operates as the network of teachers. Both multiple-stream and two-stream trimmed down networks have the same filter sizes, but the trimmed down-two-stream network uses fewer kernels in fully connected layers. Resource scheduling and dynamic ordering with memory management using deep learning approach in [17].

Vehicles are enabled by an Internet connection that allows us to communicate on the highway with many other vehicles. The link circumstance could either be vehicle to vehicle, vehicle through point of access or point of access to point of access. Implementing MEC communities along the side of the street that allow two-way communication between both the vehicles on the move. One vehicle will coordinate with the other emerging vehicles and warn them of any potential danger or traffic jams, as well as any pedestrian and bikers present. MEC also allows for scalable, stable, and distributed computing that are compatible with the local captures [18].

Furthermore, the three-tier architecture that encompasses role model, structured cloud infrastructure and mobile network virtualization will enable healthcare advisors support their consumers, irrespective of their geographical region. MEC allows devices to obtain spectral parameters from embedded devices for patients, e.g. pulse rate, temperature, etc., and to send it to the remote server for storage, data synchronization and exchange. Health professionals who have access to system service will identify patients automatically and support them appropriately [19].

In [20] recommended a flow-based as well as time-based model of power consumption. They performed various studies in a cloud computing platform for efficient energy use, using decentralized nano network infrastructure. The authors state the power consumption of nDCs has not yet been examined. Consequently, multiple models were introduced for measuring electricity usage on both shared and unshared network equipment. The authors conclude that nDCs can result in efficiency gains if technologies, specifically IoT applications, produce and interpret data at the user's properties.

To suggested the Femto Cloud network [21] that creates a cluster of coordinated mobile fully distributed devices that could be self-configured into some kind of correlative virtual cluster. On each smart phone, a Femto Cloud processing power service is enabled to measure device computational resources and sharing power with the other portable devices, and energy details. Mobile assets are built and managed within a user profile shared in a network cluster linked to a phase transitions or control system on a wifi connection.

Incorporating MEC near smart phones with the aid of an increasing system capacity and low latency will elevate data analytics. For instance, big information can be recorded and evaluated at the closest MEC location, instead of following the usual path from a network edge to the network infrastructure. The product of the advanced analytics can now be passed on for further analysis to the core network. This scenario can also handle massive data account information from many IoT devices [22].

Adhoc networks are class of decentralized networks which won't rely upon existing foundation. Vehicular adhoc networks are just versatile adhoc organizes however rapid nodes. There are various network applications which have distinctive resource requirements which will be provided resources by accessible with the networks. This exploration study gives a review of grouping of traffic types, resource requirements of various applications and proficiency of various steering conventions in Portable adhoc systems and Vehicular adhoc systems more than each other [23].

3. RESEARCH METHODOLOGY

We have built Tensor Flow to be considerably more versatile than DistBelief, while maintaining its ability to meet the demands of Google's workloads for training machine development. TensorFlow offers a basic abstraction of explode-based programming allowing users to deploy applications on distributed clusters, local workspaces, mobile devices and customized transistors. A high-level scripting interface (Figure 1 shows) allows users to experiment with the creation of dataflow graphs various interface architectures and algorithms for optimisation any changes to the main framework. In this clause, we In brief, illustrate the main design principles of TensorFlow:

3.1 Dataflow Graphs of Primitive Operators:

TensorFlow and DistBelief both use control flow simulation for their models, but the most notable distinction is that a DistBelief model consists of very few complex "layers," while the equivalent TensorFlow model represents specific mathematical operations (such as matrix multiplication, convolution, etc.) as nodes in the computation network.

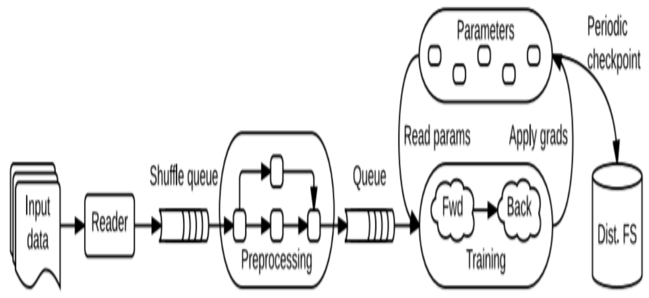

Figure 1: A schematic TensorFlow dataflow graph for a training pipeline, containing subgraphs for reading input data, pre-processing, training, and check pointing state.

3.2 Deferred Execution:

A standard TensorFlow implementation has two main regions: the first phase describes the program (e.g., the neural network to be trained and the update rules) as a graphical data flow graph with placeholders for state-representing input data and variables; and the second phase implements an improved program version on the collection of available devices. By deferring the implementation until the entire program is usable, TensorFlow can customize the execution process through the use of global computational information.

3.3 Popular Specification for Homogenous Accelerators:

In addition to general-purpose devices such as multicore CPUs and GPUs, deep learning special purpose accelerators will achieve major increase in performance and power savings. At Google, our colleagues developed the Tensor Processing Unit (TPU) explicitly for machine learning; TPUs yield an order of magnitude of performance-per-watt improvements comparison to state-of-the-art alternative technology.

The principal implication of these concepts is that there is no such thing as a parameter server in Tensor Flow. We deploy Tensor Flow on a cluster as a collection of tasks (named processes that can connect over a network) that each export the same graph implementation API and include one or more instruments

4. PROPOSED SYSTEM DESIGN

Mobile devices will also be time-sensitive in exercise; our proposed model therefore extends inferential time from seconds to several minutes in time. Showing the relevance from proposed architecture, here we present a few real-worlds Cases about smart phone applications. First, facial recognition Download, e.g., enable screen instantly optimization algorithm reduces mobile apps expand the consumer interface incrementally Limited resource on the back end (energy and memory) Overloads. Second, system allows for deep usefulness Systems operating on top of low-end IoT apps. Thirdly: the system is constructed with the time-insensitive Systems, e.g. naming of songs, this is what makes us more significant technique recommended. Obtained as one, these Results indicate scope for proposed systems concept at the top Applicable here is the machine learning system various platforms.

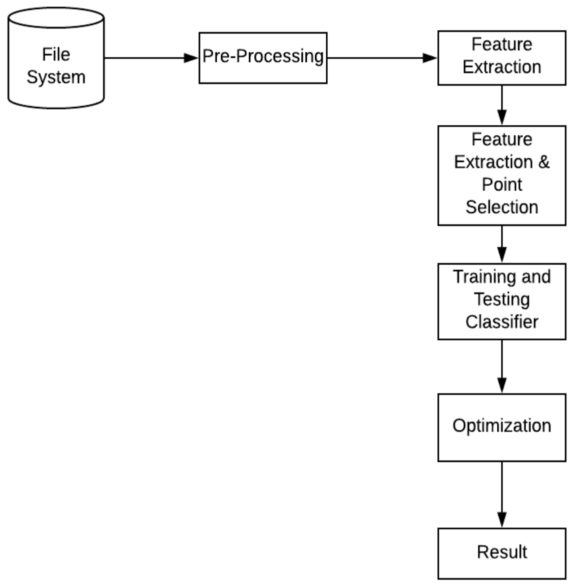
File System
Pre-Processing
Feature Extraction
Feature Extraction & Point Selection
Training and Testing Classifier
Optimization
Result

Figure 2: Proposed system design

The above figure 2 shows system overview of execution process, we basically discussed about the resource utilization and automatic memory management for devices. This technique is similar to garbage collector which is already introduced in [5], [6], [9] the main objective behind those researched to eliminate unutilized files like cache files. The proposed system work with similar approach using optimization techniques, that reduces the device overhead as well as resource allocation simultaneously. Each block has as three processes by respective machine learning algorithms and selects the features accordingly, the optimization algorithm checks the criteria if the cache is free then it will not be executed otherwise optimize the entire memory block respectively. The effectiveness of the evaluation algorithm based on requires time for optimization with feature selection. When the system is executed with respect to the file system, accuracy depends on unique feature selection and unnecessary data elimination. A proposed strategy can perform both things effectively.

5. ALGORITHM DESIGN

The various features extraction techniques has used to extract data from input data block or file systems, but after consideration those features system need to optimized entire data. The below algorithm works for memory destructor as well as block optimization respectively.

Algorithm : Memory Distractor for Optimization

Input : All Data block Db[], Memory address MA[Db[i]], memory threshold Max_Size

```
Step 1 : For each read DB[i] from Db
Step 2 : if MemUnit::isActive() is false then
            *p = MemUnit::setMemActive();
            MemUnit::setCachObj(*p);
            Repeat
            read data;
Step3:Repeat this procedure till
            size_of(*p) >  data_size;
Step 4 : if size_of(free_cache) < max_size then
            MemUnit::setMemStatic();
Step 5 : return true ;
```

Output : True if memory block has optimized else false

6. RESULTS AND DISCUSSIONS

The implementation of proposed system has been developed in python 3.7 with TensorFlow lib framework. Initially some real file system has given to generate background knowledge of module and another time pointer has given with some cache files. In two consecutive intervals system automatically release some unutilized cache file using optimization algorithm which performance shows in below graph.

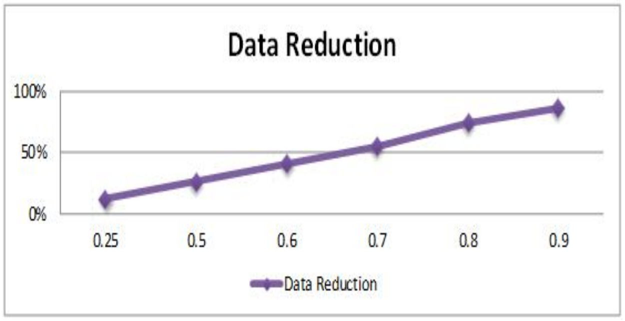

Figure 3: Data reduction using various thresholds.

The above figure 3 shows data normalization using various thresholds, the basic objective behind data reduction to eliminate the redundant data. The different thresholds remove number of percentage redundant data from existing collected data which is demonstrated in below.

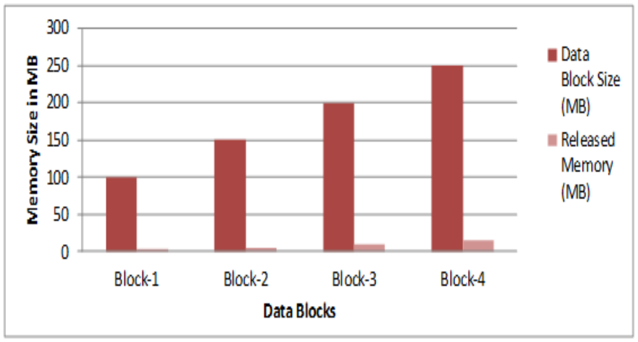

Figure 4 : Memory optimization using proposed system

The above figure 4 shows memory release by optimized using proposed system, four different data block has given input to system and evaluate the experiment analysis. According to this experiment system around release 2.50% unutilized memory of entire block.

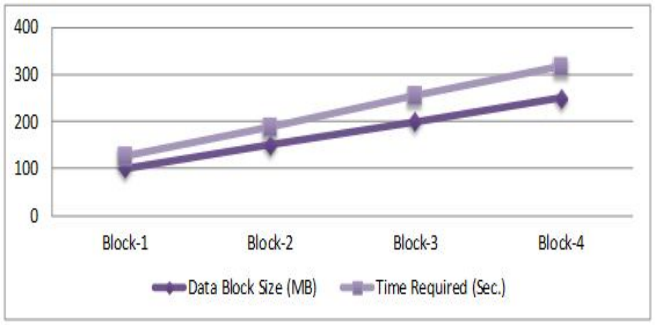

Figure 5: Time required in seconds for data processing using proposed techniques

The figure 5 shows time required for each block optimization by proposed algorithm, each block contains different memory size, so it takes time accordingly, it also change based on deployed configuring environment. Finally we conclude using the proposed technique system can provides effective results like grid computing based memory and energy management of machine or device.

6. CONCLUSION

This research we introduces automatic resource and memory management which is a novel platform enables deep mobile resource allocation in various file systems. This system does away with machine output Uses memory and time between. To that end, we proposed iterative mechanism of inference through Function block execution leading to no accuracy Loss and is environmentally friendly resource restriction. This architecture applied with some inbuilt machine learning frameworks, experiments investigate performance of system, Good extensibility, and original expertise of machine learning. We also evaluate the system overheads and limitations of Current system. We are going to serve as future Designing new technologies to make it work better Models and Structures for research. System also improves the speed of device execution and provides effective memory management using machine learning techniques.

REFERENCES

[1] H. Lee, J. Kim, D. Yang, J.-H. Kim. **Embedded Real-Time Fall Detection using Deep Learning for Elderly Care**, 31st *Conference on Neural Information Processing Systems (NIPS),* 2017, pp. 1-5. arXiv: 1711.11200.

[2] T. Abtahi, C. Shea, A. Kulkarni, T. Mohsenin. **Accelerating Convolutional Neural Network with FFT on Embedded Hardware**, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* ISSN: 1557-9999 Vol. 26, no. 9, pp. 1737–1749, 2018.

[3] Viola, Paul & Jones, Michael. **Rapid Object Detection using a Boosted Cascade of Simple Features**, *IEEE Conf Comput Vis Pattern Recognit,* 2001. 1. I-511. 10.1109/CVPR.2001.990517.

[4] Y. Sun, X. Liu, M. Yuan, L. Ren, J. Wang, Z. Chen. **Automatic in-trap Pest Detection using Deep Learning for Pheromone-based Dendroctonus Valens Monitoring**, *Biosyst. Eng.* Volume 176, pp.140–150, 2018.

[5] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa. **SDMEC: Software Defined System for Mobile Edge Computing**, *IEEE International Conference on Cloud Engineering Workshop (IC2EW),* 2016, pp. 88–93.

[6] Andrej Karpathy. **Convolutional Neural Networks for Visual Recognition.** http://cs231n.github.io

[7] Adit Deshpande. **The 9 Deep Learning Papers You Need To Know About** (Understanding CNNs Part 3) https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html

[8] Calvin Ng, Alvin Chua, **Training of a deep learning algorithm for quad copter gesture recognition,** *International Journal of Advanced Trends in Computer Science and Engineering,* Volume 9, No.1, pp. 211-216, 2020. doi: 10.30534/ijatcse/2020/32912020.

[9] Kim H, Ahn S. **BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage**, *In FAST '08: 6th USENIX Conference on File and Storage Technologies,* Volume 8, 2008, pp 239-252.

[10] Kim S, Lee SG, Song J, Yoon S. **FloWaveNet: A Generative Flow for Raw Audio**, *36th International Conference on Machine Learning,* PMLR, 2019, arXiv:1811.02155,.

[11] Huynh LN, Balan RK, Lee Y. **Deepsense: A gpu-based Deep Convolutional Neural Network Framework on Commodity Mobile Devices**. *In Proceedings of the 2016 Workshop on Wearable Systems and Applications* 2016 Jun 30 pp. 25-30, ACM.

[12] J. Kaster, J. Patrick and H. S. Clouse. **Convolutional Neural Networks on Small Unmanned Aerial Systems**, *IEEE National Aerospace and Electronics Conference (NAECON),* 2017, pp. 149-154, doi: 10.1109/NAECON.2017.8268760.

[13] Cavigelli, Lukas &Degen, Philippe &Benini, Luca. **CBinfer: Change-Based Inference for Convolutional Neural Networks on Video Data**, pp.1-8. 2017, 10.1145/3131885.3131906.

[14] C. Wang , Y. Wang , Y. Han , L. Song , Z. Quan , J. Li , X. Li. **CNN-Based Object Detection Solutions for Embedded Heterogeneous Multicore SOCs**, *in: Design Au- tomation Conference (ASP-DAC), 22nd Asia and South Pacific*, 2017, pp. 105–110.

[15] Paolucci, Pier & Ammendola, Roberto & Biagioni, Andrea & Frezza, Ottorino & Lo Cicero, Francesca & Lonardo, Alessandro & Martinelli, Michele & Pastorelli, Elena & Simula, Francesco & Vicini, Piero. **Power, Energy and Speed of Embedded and Server Multi-Cores applied to Distributed Simulation of Spiking Neural Networks: ARM in NVIDIA Tegra vs Intel Xeon quad-cores,** 2015. arXiv:.1505.03015.

[16] Cai, Lile & Barneche, Anne-Maelle & Herbout, Arthur & Foo, Chuan & Lin, Jie & Chandrasekhar, Vijay & Aly, Mohamed. **TEA-DNN: the Quest for Time-Energy-Accuracy Co-optimized Deep Neural Networks**, pp.1-6, 2019, doi: 10.1109/ISLPED.2019.8824934.

[17] Dipak Raghunath Patil and Rajesh Purohit, Dynamic **Resource Allocation and Memory Management using Deep Convolutional Neural Network**, International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-9 Issue-2, pp 608-612, 2019. doi: 10.35940/ijeat.A9961.129219.

[18] S. K. Datta, C. Bonnet, and J. Haerri, **Fog Computing Architecture to Enable Consumer Centric Internet of Things Services**, *International Symposium on Consumer Electronics (ISCE). IEEE,* 2015, pp. 1–2.

[19] Stantchev, Vladimir & Barnawi, Ahmed & Ghulam Muhammad, Sarfaraz & Schubert, Johannes & Tamm, Gerrit, **Smart Items, Fog and Cloud Computing as Enablers of Servitization in Healthcare.** *Sensors & Transducers.* Vol.185, pp.121 – 128, 2015.

[20] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, **Fog Computing May Help to Save Energy in Cloud Computing**, *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[21] K. Habak, M. Ammar, K. A. Harras and E. Zegura, **Femto Clouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge**, *IEEE 8th International Conference on Cloud Computing*, New York, NY, ISSN: 2159-6190, 2015, pp. 9-16.
doi: 10.1109/CLOUD.2015.12.

[22] A. Taherkordi, F. Eliassen, and G. Horn, **From IOT Big Data to IOT Big Services**, *SAC '17: Proceedings of the Symposium on Applied Computing,* April 2017 pp. 485–491, 2017.
doi: 10.1145/3019612.3019700.

[23] Pooja Sharma, Ajay Kaul, M L Garg, **Resource Provisioning for various network applications in adhoc networks,** *International Journal of Advanced Trends in Computer Science and Engineering,* Volume 8, No.5, pp. 2653-2659, 2019.
doi: 10.30534/ijatcse/2019/119852019.