Enhanced Method to Stream Real Time Data in IoT using Dynamic Voltage and Frequency Scaling with Memory

H. A. Hashim
Department of Computer Science
College of Computer Science and Engineering
Taibah University, Yanbu, KSA

***Abstract*—DVFS (Dynamic Voltage and Frequency Scaling) is a popular CPU (Central Processing Unit) level voltage frequency scale technology based on the application precedence. To motivate recurrence / voltage scaling as a feasible tool for energy productivity, i) basic workloads should ensure that memory recurrence scaling has an impact with insignificant degradation and (ii) that there is an enormous open door for reduction of power in this work. Therefore, if memory recurrence is that, two limiting forces on energy efficiency impact all items in an anomalous state. The competence depends on both power and runtime, because energy is the result of time and energy. The reduction in control alone will increase skills. However, further discussions at lower control work focuses are conducted, expanding operating times and energy in this way. There is a bloating edge that decreases the recurrence / voltage of memory in this way. This shows further that the recurrence of statically-scaling memory has little impact on many lower workloads because of recurrent effects only the idling of transmission interchange, part of the memory dormancy. This will be shown. Inspire in this paper, the scaling of memory recurrence will affect frame power (show a systemic model to simplify the scaling of voltage) and therefore electricity. It presents DVFS memory computing in real time. The DVFS technology is popular for measuring the frequency of voltage according to the CPU level applications. In this work, an enhanced DVFS with memory technique proposal is used to decrease energy use and improve performance at the memory level**.

***Keywords*—Dynamic voltage; frequency scaling; central processing unit**

I. Introduction

Data Streaming in most environments is very tedious and different factors has to be concentrate to keep the quality of service.

A. Big Data Stream Computing (BDSC)

BDSC, a long-lasting vision for ‘high performance computing’ and ‘high real time computing’, has opened a new age of upcoming computing due to Big Data, a dataset that is big, large, unstructured, scattered and beyond the capacity of software and devices to access, obtain, analyze, etc. [1] Stream computing system is a computer standard which reads data from hardware or software sensor collections in a stream format and computes continuous real-time data streams, where results and feedback should also be made available in a reliable data stream. A stream of data is an infinite series of datasets in the network and has more parallel streams processing a single stream at the same time [2].

B. IOT Application Data Streams

The IoT notes that innovation is a paradigm in which the inevitable numbers of sensors in the trillions will have the capacities to slowly track physical conditions, individuals and virtual substances; to deal with real as well as on-line perceptions; to carry out activities that improve the efficiency and efficacy of the environment or society's way of life and consolation [3]. Over a span of ten years, the building blocks of IoT engineering have evolved with sensor systems work and inevitable processing experience. The existing IoT inflection point has all been contributing to the ongoing development in capacity of a quick-port system (e.g. 2G/3G/4 G) and impromptu (e.g. Bluetooth) systems, affordability, smartphones and crowd-sourced data collection [4], big data analytics and the cloud-based data center platforms. IoT applications are now also present in vertical areas, such as failure management and demand control in intelligent grids, as well as sleep and exercise tracking, and health band and intelligent watch recommendations.

In particular, these streams were explored in huge scales (billions of sensors, several opportunities per second) to request transmission of large-scale computational assets through transmitted sensors. Distributed computing offers a typical stage for adapting the perceptions in all server farms conveyed and sending critical answers to the edge of the IoT framework.

Collection of IoT Data streams, interfacing, and improving these data streams with the relevant contextual metadata such as time stamps as well as location data, including too many IoT sources (such as devices linked to Internet). The collection process, as previously outlined, traditionally requires a solution to the heterogeneity of their IoT data streaming and the heterogeneity of interfaces to data formats and sources. IoT application authentication and validation of data source and origin format. Application validation. The method requires a precision, continuity and credibility validation. IoT data interoperability and convergence, which deal with IoT data streams homogenizing and uniting, capture IoT data streams, interfaces and enhances these streams with the use of relevant contextual metadata, including timelines and location data. Semantic IoT Data interoperability and fusion as previously outlined, the collection process classically requires the heterogeneity of its IoT data streams, the interfaces of data formats and data source heterogeneity. IoT application validation and data source and format validation. IoT data validation. The method requires a precision, quality and competence evaluation. The protocol Interoperability and integration of the semanticization of IoT data sources Semantic IoT data interoperability and convergence [5].

C. D-RAM

For initialization operations and bulk data copying, a sample of such wastage often happens on which a copy page is initialized or added to a value. If the processor doesn't explicitly need the initialized or copied data then the processor can substantially save energy by changing the bandwidth and time within the DRAM (with minor changes to the DRAM) [6].

In this, DRAM chip can be detected internally on a large data set. The core DRAM structure can be exploited and a page copy can be done in the DRAM without any data being fetched from the DRAM chip, as shown in recent work. If the source and destination page are within the same DRAM subarray, the results show a page copy that is improved by more than a demand in magnitude, which is important to reduce the energy by ~74 times and no bandwidth wastage for a DRAM data bus [7].

The main idea is that 1) deactivated by the source row content of the knowledge amplifier, 2) by the source row, which offers very small hardware costs, which is smaller than 0.03% of DRAM chip space, and 3) straight activates the destination row, allowing the information amplifiers to push its contents through. The main idea is to deactivate the source row.

A few definitions must first be made to clarify the system of fall-off workload. The main idea is to enforce the use of the memory bandwidth, then change the memory frequency / voltage, to reduce the power with minimal DVFS memory loss as in Fig. 1.

The main contributions of our research work include:

- Evaluate the scaling of memory frequency / voltage to increase energy efficiency and reduce storage energy.
- Observation the frequency-dependent part of the memory system power has a control algorithm, which reduces the memory frequency and reduces the performance impacts.
- The important point is that the lowering of memory frequency does not significantly change memory access latency when bandwidth is used in low capacity.
- The propose control algorithm increases the memory frequency when the usage reaches a threshold, reducing the output effect, by monitoring the memory bandwidth utilization.

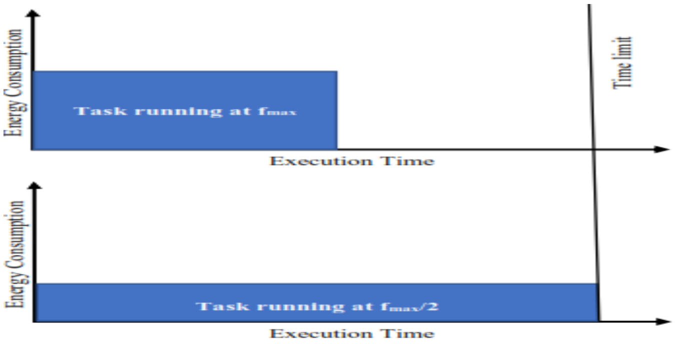

Fig. 1. Dynamic Voltage Frequency Scaling.

II. Related Work

Big Data Stream Computing is running in the memory entirely. Apply DVFS technology to the massive cluster of knowledge and the scientific evidence that gives Sun et al workloads for every on / off chip doling [8]. The duty to perform a job is defined as the entire CPI of all covers towards pathway of undertaking. The workload depends on various component parameters, such as the number of log jam on-chip as a result of data / control dependency or the missed branch requirement, and log jam cycle off-chip control as a consequence of I / D storage errors or I / D TLB failures. The CPU is located in the middle of an off-chip until the memory is needed. In the middle of an off-chip, the CPU clock cycles are therefore lost [9].

The use of DVFS in Baskiyar's main memory [10] is yet another interesting way to reduce energy consumption in DRAM. David et al [11] note that using a lower memory bandwidth does not substantially change memory access latency by reducing memory voltage / frequency. Comparatively current works have shown the importance of saving energy both in real simulated systems from the Sharifi and Shahrivari [12] systems, by adjusting the memory frequency and voltage based on predicted use of memory bandwidth. DVFS memory level can successfully allow dynamic heterogeneity in DRAM channels, leading to new mechanisms for optimization and personalization. The study of more fine-grained methods of management of power within the DRAM chips and the idle and active modes of low power are also very promising. In addition to the valuable space on the chip cache, the existing systems energy consumption considerably waste DRAM time and bandwidth by moving data from main memory to processors caches, often inappropriately.

This constraint on the application of data streaming in real time has been applied to new methods for data mining. The Sudipto [13] project variations in the k-means grouping algorithm. In Sudipto (2019), the authors present the HP Stream, a method for the grouping of streaming data. Yixin Chen and Li Tu [14] understand that the grouping process is distributed online to a variable that periodically stores and collects a detailed overview statistics, also an offline element which is only used for summary statistics. Yixin Chen and Li Tu [15] present density grouping methods for streaming data.

In an ever-growing environment, a classification process may involve in real time model testing and construction. Pedro Domingos and Geoff Hulten [16] present decision tree-based methods which include only the data stream via a single pass. The on-demand classification method in Charu et al (2016) suggests dynamically selecting tools from the correct window of previous training information to be used for classification building. The problem of routine data-streaming pattern mining is investigated. The loss counting methodologies and sampling methods for holding approximate counts over a window slide with a space with restricted access are available in Gurmeet Singh Manku and Rajeev Motwani [17].

Wang et al [18] suggest PW Join, is a three-operation algorithm designed to add binary windows to activities that are value-driven and can be found in streaming data. In Daoud [19] authors suggest GrubJoin, a multi-purpose, adaptive window data stream connection to efficiently complete CPU load flaking associations.

III. Existing Method

The existing method there is no concentration of memory. In BDSC platforms, DVFS-level fonts are accessible. It increases the energy for small businesses. The equally recent needs of the fundamental association of memory are threefold. To begin with, diverse innovation: an innovation that is quite adaptable to resources, cost and constraints, as defined by the past, is a new requirement for the outcome. Because DRAM, which continually stretches over 100-nm to 30-nm from innovation hubs, the need for more mobile innovation did not stand out. From now on, DRAM has not been done as much as the 30 nm hub with the enormous device or circuit scaling errands.

Secondly, the QS execution and accuracy in a common simple memory setting has been similarly forgiven for a new requirement. As one center system was a memory and the current limit and transmitting ability were a very small measure of an asset sharing, a much less prevalent or apparent need for planned implementation was present. Today's top methodological constraints have become increasingly common with more specialists / centers sharing memory frames on a chip and the aggregate costs of storage execution, reasonableness of memory, organizational union and methods to control memory impedance. Thirdly, the technique produced for the fundamental memory framework requires significantly higher data transmission capacity / control / energy effectiveness.

Propelled energy productivity to allow the frameworks of data transmission capacities, power and electricity is far more flexible where shared principle memory is shared between various operators, and it can allow new applications in all areas where processes are used. This is no innovative precondition at the moment, it is conceivably an additional cutting edge technology constraint which has not remained the usual calculation of cost, limitations and implementation.

A. Drawbacks

The techniques used by the DVFS are used to degrade and *measure* processor operating frequency and input voltage. It is a dynamic approach to reducing the use of energy. The lower the power state, the more prominent Baskiyar (2010) points out the energy reserve funds. Nevertheless, the execution time is longer. The time to complete the execution is littler and the task is completed before the date for which the errand remains running at higher express capacity. At the moment that the company is completed, the part is run in a stand-off gear, some amount of energy is also required. When the business continues to run in a small power express, the execution time is longer but not past the mark. When an assignment is completed in time, only less energy is required.

B. Proposed Method Dvfs with Memory Technique

Add frequency scaling algorithms for the energy efficiency boost. It reduces the application's energy consumption and improves application efficiency. The software basis is the frequency-based control system. The control strategy of data streams is displayed in Algorithm. The computing platform in large data is managed when IoT application data is being computed. It is therefore difficult to look for additional loopholes to overcome this problem. In Stream Computing Storm platform is open to meet the full requirements of the current data streaming dimension.

Round Robin is used by default in this scheduling strategy. The data circulation is not adequate at the point slow processing and the power use very many failures to this end. Data generated by computers. In the Storm platform shown in Fig. 2 the mechanism runs internally.

Three different types of data sensing devices (2) communication on request, Billard et al (1993), (3) event driven communication from Meng et al (2014). In the real time of computing the data streams, three different types of data sensing devices are available in IoT. It requires more computational power to process this type of data. Modifying the timetable approach to suit the various data sets offers highly effective management of data. The algorithm updated notice Algorithm 5.1 which calculates the time complexity of the algorithm. For further analysis, steps are identified with a double slash in the algorithm itself. Three strategies are used for the process mention in algorithm. Secondly, the DVFS referred to in Sveur et al (2010) solutions to energy reduction refine their stream maps. Secondly, there is hot-swapping technique to reprogram your worker nodes online. The third is to assign one and two approaches to performance scaling.

```
Input: Real-Time IoT streaming data
Output: Optimized Solution with frequency based task allocation
Step 1: Bandwidth required data stream
Step 2: Bandwidth available to process task
Step 3: Average Bandwidth thread assigning task
Step 4: if $B$andwidth < $Tf$b  then
Step 5: set memory frequency to $f$b
Step 6: else if $Tf$b = $B$andwidth < $Tf$c
then set memory frequency to $f$2
Step 7: else if $Tf$c = $B$andwidth then
Step 8: set memory frequency to maximum frequency $f$a
Step 9: end if
Step 10: end while
```

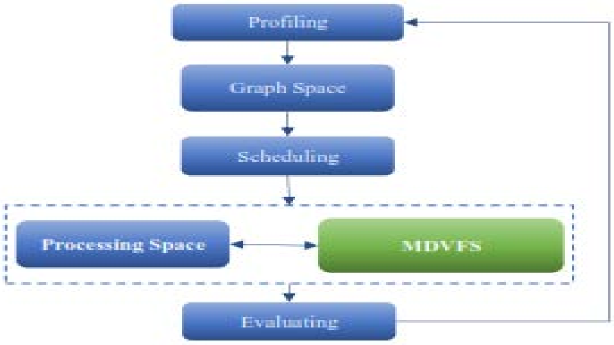

Fig. 2. Internal Work Flow of Storm.

C. Evaluating the Performance Time of a Task

The application of the DVFS technique to the huge group of gushing information and the scientific confirmation of every on / off chip doling process referred to by Sun et al (2015). The workload of an order is described as the total of all covers of the CPI to the stream of path. Task workloads depend on different component parameters such as on-board data / control reliability stoppage number and off-off-board log jam period checks due to I / D shop missing or I / D TLB missing data / control depending on branch missing forecasting. In the middle of an off-chip the CPU returns until the memory trade has been requested.

CPU clock cycles are therefore adjusted in the middle of an off-chip. This must first provide a few examples in order to clarify the rotary workload structure.

Sun et al (2015) shows Energy Reduction by graph construction methods, where Storm is a spelling structure and it is unbelievably interesting in relation to recently suggested package planning systems. The energy reductions are given in response time and high energy calculations.

It also generates a DAG of centers which enables the record to move between centers. It is ideal to vary from Hadoop MapReduce from Bhandarkar (2011) [20], storm and spark streaming (spouting structure rendered over Spark). All consider a DAG of activities covering the retaliation of spouts, but then imperceptibly process the DAG in particular. Storm defines the DAG of the center and sets out separate centers for each activity of the DAG. Streak Streaming does not reassign, but using Spark's basic instruments to delegate commands to available resources vigilantly. This gives various kinds of performance properties.

D. Enhanced Data Stream Adapting Dvfs with Memory

For BDSC implementations, the in-memory state is always in use. This uses the same memory capacity at all measurement speeds of data streams. Control frequency levels and improve performance for low and high task topologies.

BDSC for IoT workload analysis evaluates factors such as latency, jitter, performance, CPU and the use of memory. When running the program, three separate frequency ranges show that the transmission capacities will change as much as the static recurrence decrease in low data transfer capacity benchmarks, when retaining the remarkable decrease in capability (due to the extended runtime and the aggregation swells along these lines). The Bandwidth approach (0.5, 2) decreases memory control by 6,15W (11,41W maximum) over SPEC CPU 2006, for 0.18% normal (1.72% maximum) lull. All things considered, over the SPEC CPU 2006. If the data transfer capacity (i.e. with transmitting capacity generally less than 1.2 GB / s per channel) is sorted on one side of the gcc, all considerations of the lower memory frame control by 9.81W are taken into account.

1) Dram structure background: This is important when the standard 1363MHz memory structure is 65.1W. The bandwidth (0.5, 2) in the entire setting gives 2.43% normal (maximum 5.25%) change in energy efficiency. It is important to note that I reductions in control and thus improvements in energy productivity due to memory DVFS that give the approach CPU-based DVFS a similar degree of power as in Table I.

A device arrangement composed of a DIMM has a rank. In a rank, the different banks provide free DRAM storage items with relevant translators and intelligence speakers on each unit. Both parties share the I / O (collectors and drivers) equipment to the edge of DDR transport. Each group is a cell lattice, divided into columns. The column support can be used at any time to keep the individual dynamic line (or page). An initial call puts a line in the wire, which enables read / compose instructions to enter segments in the line.

A pre-charge charge returns information to the display and is ready for the subsequent execution. Itemized memory process representations can be found in detailed data sheets. Various strategies are the way a memory controller handles these charges. The page-open and page-shut solutions are two common methods. Page-open retains the last line dynamic (page) in the line pad. It allows the following access to a similar line, as no precise information is necessary. Page-shut plays a pre-load when the load is complete. Despite the fact that this brings outline hits, it decreases idleness in a missed line, because no preload is necessary, only an actuate. Take note that in this paper a page shut strategy is embraced. This is informed by the vision of multi-center frames as in Table II.

TABLE I. ENERGY PER OPERATION ON DRAM WITH FREQUENCY OF 850MHZ AND 1060MHZ THROUGH BASELINE FREQUENCY IS 1363MHZ

Command	Energy (nJ) @ 1363MHz	Energy (nJ) @ 850MHz	Energy (nJ) @ 1060MHz
Write (array)	22	22	21
Read (array)	17.5	17.3	12.5
Write I/O (1 DIMM/channel)	4.2	7	3.2
Read I/O (1 DIMM/channel)	1	1.7	2
Activate +Pre-charge(page open-close)	24.9	25.1	21.3
I/O additional termination (2 DIMMs/channel)	12.3	20.9	15.2
Average energy/write, page-close policy, 2 DIMMs/channel	60.9	72	49.8
Average energy/read, page-close policy, 2 DIMMs/channel	55.2	64.8	41.6

TABLE II. BACKGROUND POWER OPERATION: EVALUATING ALL THE PARAMETERS OF THE POWER CONSUMPTION VALUES

Power-down state	Exit latency to read Command	Power @ 1363	PIL, Out. Clk	IBT, ODT	DLL	Clk. Tree	Input Buf	Self Refresh
Self-Refresh	512 tCK	0.6W	0	0	0	0	0	1
Self-Refresh	tMRd+tXPDLL	0.98W	1	0	0	0	0	1
Pre-Charge Slow Down	tMRD+tXPDLL	1.35W	1	0	0	0	0	0
Charge Slow Down	tXPDLL	1.64W	1	1	1	0	0	0
Pre-Charge fast power	tXP+tRCD	2.79W	1	1	1	0	0	0
Active Paower	tXP	3.38W	1	1	1	0	0	0
Pre-Charge Standby	tRCD	4.76W	1	1	1	1	1	0
Active standby	0	5.46W	1	1	1	1	1	0

Fig. 3, the power consumption of various segments in the DRAM system is depicted as the basis for this work.

2) I/O Power: This portion of equipment power includes information cushions, read / write handles, DLL (delay bolted circles), drivers for the information transport and logic of control and is consumed when the DRAM is still sat (not shut down) or when the order is executed effectively.

I / O capacity is subordinate to memory recurrence: with less recurrence, it declines. The I / O control field, due to the dynamic control execution of transport scales, has a circuitous effect while the energy effect of recurring scales is taken into account, which has been stated below. In memory shutdown, I / O power is reduced.

3) Registered power: A registered DIMM consists of clock and order / address line information / produce registers; enlist control includes these parts, as well as associated justification and a stage bolted cycle (PLL). Like I / O control, enlist power is recognized as recurring subordinate with the transport interface. It also correlates with low-control countries.

4) Transmission power: Termination Power: Finally, at present day DRAM device incorporate On-Die Termination (ODT) to appropriately end the transport amid dynamic operation. End power is scattered in on kick the bucket resistive components and is changed in accordance with transport electrical attributes, contingent upon DIMM number. With per channel 2 DIMMs, DDR3 end power can achieve 1.8-2.2W for each DIMM.

E. Experimental Setup

In this model, the new proposed frequency control algorithm was introduced based on DVFS memory level policies to boost energy efficiency through the introduction of a Storm device step and accurate tests. Creates a Storm platform simulation environment which has been built in full parallel fault tolerance, distributed with Storm0.10.0 the latest version software. A 4 core Intel 13 processor 2.00 GHz 64 Bit processor, 16 GB of memory, and 1 Gbps network connectivity is built for virtual machines. 4 core 2 machines with an external 10 TB storage capacity are connected to each other.

The Linux server runs on each individual machine (Ubuntu server version 14.01). The following component of software usually is designed and used in conjunction with Java 1.8, zoo 3.4.0, python 3.0. In addition, all updated scheduling techniques refer to the replacement of the existing energy-efficient traffic scheduling plan for the IScheduler in Storm network. Observing the StormUI output. The average loading time for tuple metrics is used.

In this method the default time function (Storm) was used to monitor each tuple's processing time. StormUI can collect this information, but it was an average 10-minute display. This method took averages of 1 minute in this testing and implementation, which gave us far better accuracy in real-time estimates of results. Ubuntu Linux uses the NTP protocol specifications to synchronize worker nodes for the duration of experiments. Rest of all test values listed in Table III.

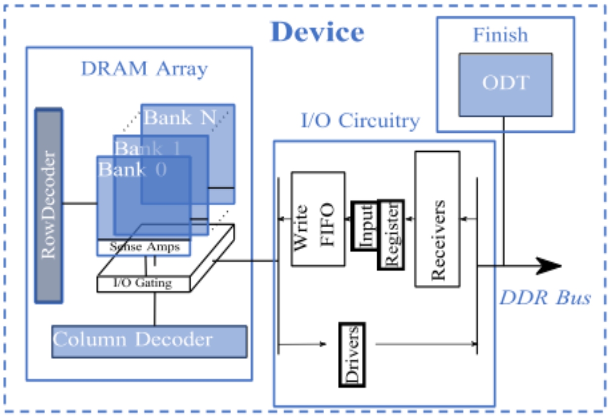

Fig. 3. General Overview of DRAM Device Structure.

TABLE III. EXPERIMENTAL VALUE ASSIGNING

S. NO	Bounds	Values
1	Estimation period and monitoring load	40 sec.
2	Estimation of coefficient (α)	0.5
3	Period of Schedule fetching $p(s_f)$	20 sec.
4	Period of Schedule generation $p(s_g)$	400 sec.
5	Each Experiment's Running Time E_{RT}	15000 sec.
6	No. of worker node available	15

IV. Results and Discussions

The BDSC evaluation of factors like latency, jitter, speed, CPU, and memory use for the IoT workload discussion. During implementation three different frequency ranges are established and demonstrate that the data transfer strategy can improve both energy productivity and static recurrence, while retaining a strategic distance from the broad performance declines, for lower transmission capacity thresholds (due to extended runtime and along these lengths). Normal framework control decreases: decreases are critical for low data transmission, compared with time spent at lower frequencies. Generally speaking, bandwidth (0.5, 2) decreased memory power by 6.15W (11.42W maximum) (DC control) for a standard (1,71% maximum) log jam by 0.18% over SPEC CPU 2006. In this method, the memory system power is reduced by overall 9.62W by including benchmarks and on one side of the gcc when sorted by data transfer capacity (thus with a transmission capacity usually below 1,12GB / s per channel). This is important when the standard 1363MHz memory frame power is 65.1W. In an entire network environment, bandwidth (0.5, 2) gives a standard change in energy efficiency of 2.43 percent (maximum 5.23 percent). It is important to note that I control reduction and changes in energy efficiency as a result of DVFS memory will give CPU-based DVFS comparative power losses and (ii) reduction accompanies negligible performance decreases.

Fig. 4 seems by all accounts, to be troublesome for undertakings that arrangement a high throughput, perhaps illustrative the memory spent by messages holding up in line somewhat than spent by the assignment objective itself, not operational power. As such, the power the device expands paying little respect to which or what number of charges it is executing, subordinate just on its present state and recurrence. Current DDR3 device bolsters an assortment of shut down states to spare power when idle. So as to evaluate DRAM control amid a framework's execution, must measure the power use in each state, and after that ascertain normal foundation control weighted when spent in each state.

With a specific end goal to comprehend the possible for voltage scaling in open DDR3 device, tests performed on 8 DIMMs in proposed assessment framework while physically controlling the memory voltage controller yield. The outcomes appear in Fig. 5. At 1363, 1060, and 850MHz individually, watched normal least constant voltages of 1.173V, 1.203V and 1.280V separately, in addition, a greatest over the 8 DIMMs of 1.24V, 1.27V and 1.35V distinctly. In this fashion, gather the required supply voltage lessens with recurrence. Afterward, in alluded from Benkhelifa et al (2014), will display control diminishment by conservatively expecting Vdd of 1.35V, 1.425V and 1.5V. Least Stable Memory Voltage as a Function of Frequency Distribution is shown in Fig. 5. Take note of that these voltages are well over the base stable voltages for the tried DIMMs.

By fluctuating the memory frequency and thus the highest transfer capacity, the memory failure seen with customer workloads is modified. As memory idleness expands, an off-demand center is less ready to cover up the inertness and slow down time, resulting in lower execution. Eventually, this effect is based on the quality of the application. Nevertheless, as mentioned below, know the data workloads in SPEC CPU2006, which includes CPUs and severe memory applications. This is because a transmission capacity-based scaling device with legally defined edges is shifted to a greater recurrence, with a lower dormancy and a higher immersion level, when tailing delays are obviously remarkable.

The SPEC-CPU2006 operating memory at 850 MHz and 1060 MHz (corruption by 1363 MHz) is shown in Fig. 6. Standard usage of normal transfer speeds is sorted for benchmarks. After all, metrics with higher transfer pattern rate experience more execution corruption at lower memory rates, provided that the data transmission of the measuring instrument is equal to or beyond the lower recurrence of the pinnacle transfer rate. As illustrated in Fig. 6, dormance increases significantly as the most extreme data transmission methodologies for use. These benchmarks show that genuine data transfer capacity has decreased at lower memory frequencies due to memory throughput limitations. As the previous area became apparent, benchmarks with the lower data transfer demand are generally tolerant for lower memory recurrence. This can minimize normal power by reducing the memory recurrence if it has no impact; because the runtime is not impaired or because energy efficiency can improve in less favorable conditions.

To do so, simply choose a fixed edge to relay for all recurrence movements. The calculation of controllers takes place occasionally in established ages and measures the normal use of data transfer for the last age. In view of this calculation, it chooses the recurrence of the comparative memory. This pragmatism is the product of equation 1 of tests, two sides have to be indicated: the shift between 850 and 1060 MHz and a shift between 1060 MHz and 1363 MHz. Assess two edge settings: Bandwidth (0.5,1) moves to 1066 and 1363MHz at 0.5GB / s and 1.0GB / s per channel, separately, and Bandwidth (0.5,2) moves at 0.5GB / s and 2.0GB / s. Both thresholds are mild because they are under the knees in Fig. 5. Or perhaps, because of normal transmitting capacity estimates per differential, they are picked in Fig. 5. Such parameter choices have a negligible impact on the implementation of the proposed model performance.

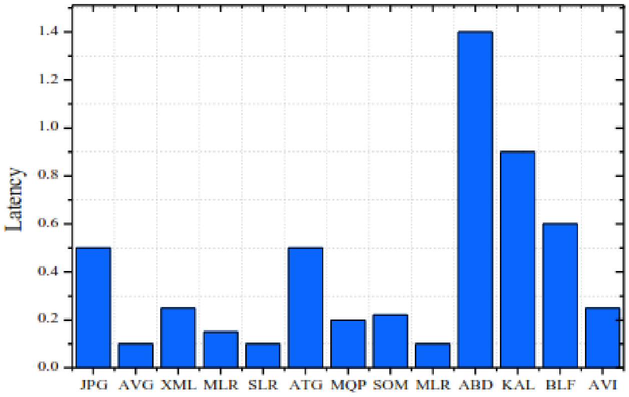

Fig. 4. Performance Analysis of IoT’s Application Data Streams Applying MDVFS.

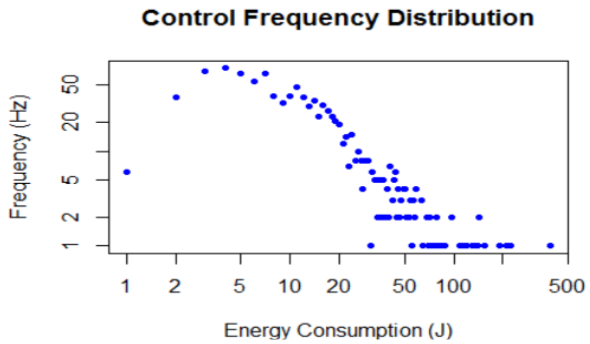

Fig. 5. Least Stable Memory Voltage as a Function of Frequency Distribution.

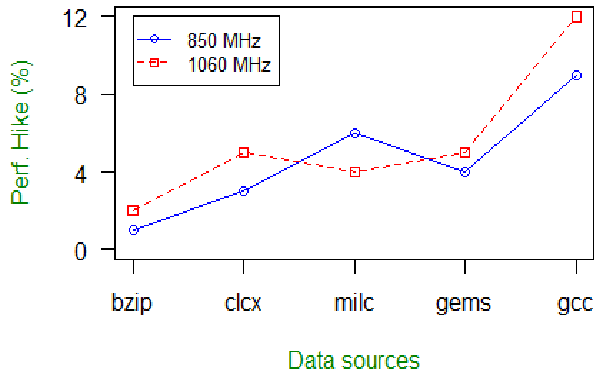

Fig. 6. Performance Varying as Per Memory Bandwidth Range from 1060MHz and 850MHz to Baseline 1363MHz.

V. Conclusion

The main barriers in memory recurrence scaling were presented in this approach and a fundamental assessment was made using a clear and natural calculation. More work remains to be completed, however. Initially there is a simple framework and a large plan space to determine and predict the impact of memory recurrence changes and to foresee the future effect.

In this paper, the proposed model is to evaluate the scaling of memory frequency / voltage to increase energy efficiency and reduce storage energy. Beginning with the observation that the frequency-dependent part of the memory system power has a control algorithm, which reduces the memory frequency and reduces the performance impacts. The important point is that the lowering of memory frequency does not significantly change memory access latency when bandwidth is used in low capacity. The proposed control algorithm increases the memory frequency when the usage reaches a threshold, reducing the output effect, by monitoring the memory bandwidth utilization. In this method, DVFS can be an efficient technology of energy efficiency, with particular when memory bandwidth is low.

References

[1] Fox R. Griffith A. Joseph R. Katz A. Konwinski G. Lee D. Patterson A. Rabkin I. Stoica "Above the clouds: A Berkeley view of cloud computing" Rep. UCB/EECS vol. 28 2009.

[2] Q. Zhang L. Cheng R. Boutaba Cloud computing: state-of-the-art and research challenges Journal of Internet Services and Applications vol. 1 pp. 7-18 2010.

[3] A. Thusoo J.S. Sarma N. Jain S. Zheng P. Chakka Z. Ning S. Antony L. Hao R. Murthy "Hivea petabyte scale data warehouse using Hadoop" Proceedings of IEEE 26th International Conference on Data Engineering (ICDE) pp. 996-1005 2010.

[4] M. Ezhilarasi and V. Krishnaveni , "An Optimal Solution To Minimize The Energy Consumption in Wireless Sensor Networks,"International Journal of Pure and Applied Mathematics", Vol. 119, Issue 10, (2018), pp. 829-844

[5] S. Chen Y. Sun D. Ulas K.L. Huang P. Sinha G. Liang X. Liu N. B. Shroff "When Queueing Meets Coding: Optimal-Latency Data Retrieving Scheme in Storage Clouds" IEEE INFOCOM pp. 1042-1050 2014.

[6] G. Liang U. Kozat FAST CLOUD: Pushing the Envelope on Delay Performance of Cloud Storage with Coding IEEE/ACM Trans. Networking vol. 22 no. 6 pp. 2012-2025 Nov 2013.

[7] S. L. Garfinkel "An evaluation of Amazons grid computing services: EC 2 S3 and SQS" in Tech. Rep. Harvard University 2007.

[8] V. Alieksieiev O. Gaiduchok "About the problem of data losses in real-time IoT based monitoring systems" Proceedings of International Scientific Conference "Mathematical Modeling" (Borovets Bulgaria December 13-16 2017) STUME "Industry 4.0" Sofia Bulgaria Year I vol. 1/1 pp. 10-11 2017.

[9] M. Ezhilarasi V. Krishnaveni "A Survey on Wireless Sensor Network: Energy and Lifetime Perspective" Taga Journal vol. 14 pp. 3099-3113 ISSN 1748-0345.

[10] Nagarajan, M. "A New Approach to Improve Life Time Using Energy Based Routing in Wireless Sensor Network. International Journal of Science and Research (IJSR), 2012

[11] V. Alieksieiev G. Ivasyk V. Pabyrivskyi N. Pabyrivska Big data aggregation algorithm for storing obsolete data" Proceedings of International Scientific Conference "High Technologies. Business. Society 2018" (Borovets Bulgaria March 1-15 2018) STUME "Industry 4.0" Sofia Bulgaria Year II iss. 1 (3) vol. I "High Technologies pp. 113-115 2018.

[12] N. V. Myasnikova M. P. Beresten M. P. Stroganov "Approximation of multi extremum functions and its applications to technical systems" Herald of higher education institutions. Volga region. Engineering sciences vol. 18 no. 2 pp. 113-119 2011.

[13] M. Nagarajan and S. Karthikeyan, "A New Approach to Increase the Life Time and Efficiency of Wireless Sensor Network", IEEE International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME), (2012), pp. 231-235

[14] H. Sundmaeker P. Guillemin P. Friess S. Woelfflé "Vision and challenges for realising the internet of things" Cluster of European Research Projects on the Internet of Things European Commision 2010.

[15] M Shanmugapriya, Nagarajan Munusamy, "An Approach to Increase Energy and Life Time Using Power Control Optimization in Wireless Sensor Networks", International Journal of Pure and Applied Mathematics, Volume 119 Issue.15, 2018, pp.1171-1181, ISSN: 1314-339

[16] M. Kovatsch S. Mayer B. Ostermaier "Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things" IMIS. IEEE pp. 751-756 2012.

[17] Ezhilarasi, M. & Krishnaveni, V. "An evolutionary multipath energy-efficient routing protocol (EMEER) for network lifetime enhancement in wireless sensor networks" Soft Computing, (2019). https://doi.org/10.1007/s00500-019-03928-1

[18] D. O. Olguin P. A. Gloor A. S. Pentland "Wearable sensors for pervasive healthcare management" Proc. IEEE. The 3rd International Conference on Pervasive Computing Technologies for Healthcare PervasiveHealth 2009 pp. 1-4 Apr. 2009.

[19] L. Gnanaprasanambikai and N. Munusamy, "Survey of genetic algorithm effectiveness in intrusion detection," 2017 International Conference on Intelligent Computing and Control (I2C2), Coimbatore, 2017, pp.1-5. doi: 10.1109/I2C2.2017.8321877

[20] R. C. Taylor "An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics" BMC Bioinformatics vol. 11 no. 12 2016