




Article

Smart Architecture Energy Management through Dynamic Bin-Packing Algorithms on Cloud

Neha Gupta ¹, Kamali Gupta ^{1,*}, Shalli Rani ^{1,*} , Deepika Koundal ^{2,*}  and Atef Zaguia ³ 

¹ Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India; neha.gupta@chitkara.edu.in

² Department of Systemics, School of Computer Science, University of Petroleum & Energy Studies, Dehradun 248007, India

³ Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. BOX 11099, Taif 21944, Saudi Arabia; zaguia.atef@tu.edu.sa

* Correspondence: kamali.singla@chitkara.edu.in (K.G.); shalli.rani@chitkara.edu.in (S.R.); dkoundal@ddn.upes.ac.in (D.K.); Tel.: +91-9896809404 (D.K.)

Abstract: Smart Home Architecture is suitable for progressive and symmetric urbanization. Data being generated in smart home appliances using internet of things should be stored in cloud where computing resources can analyze the data and generate the decisive pattern within no time. This additional requirement of storage, majorly, comprising of unfiltered data escalates requirement of host machines which carries with itself extra overhead of energy consumption; thus, extra cost has to be beard by service providers. Various static algorithms are already proposed to improve energy management of cloud data centers by reducing number of active bins. These algorithms are not able to cater to the needs of present heterogeneous requests generated in cloud machines by people of diversified work environment with adhering to the requirements of quality parameters. Therefore, the paper has proposed and implemented dynamic bin-packing approaches for smart architecture that can significantly reduce energy consumption without compromising upon makespan, resource utilization and Quality of Service (QoS) parameters. The novelty of the proposed dynamic approaches in comparison to the existing static approaches is that the proposed approach dynamically creates and dissolves virtual machines as per incoming and completed requests which is a dire need of present computing paradigms via attachment of time-frame with each virtual machine. The simulations have been performed on JAVA platform and dynamic energy utilized-best fit decreasing bin packing technique has produced better results in maximum runs.

Keywords: smart-home; internet of things; cloud computing; virtualization; bin-packing technique



Citation: Gupta, N.; Gupta, K.; Rani, S.; Koundal, D.; Zaguia, A. Smart Architecture Energy Management through Dynamic Bin-Packing Algorithms on Cloud. *Symmetry* **2021**, *13*, 2298. <https://doi.org/10.3390/sym13122298>

Academic Editor: Vijayakumar Varadarajan

Received: 26 October 2021

Accepted: 25 November 2021

Published: 2 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The invention of smart frameworks has fascinated the manual day-to-day activities and has resulted in escalation of raw digital data being generated via internet of things (web of sensors, electronic devices and software that are connected using internet for amplification of the automation process of devices) [1] and its interlinked and symmetric smart architectural devices. This unprecedented demand of technology driven service extraction has led to production of enormous data which has subsequently arisen a need to propose better energy management solutions. The most prominent adoption to internet of things (IoT) service can be witnessed with the usage of smart home appliances and has become the major contributing raw data generation source. With the rapid and positive development in society, “home” term cannot suffice the term “building with necessary appliances” but it represents the “smart and intelligent building”. Smart home comprises of appliances like Amazon Echo, Philips Hue, TP-Link HS220, Ecobee Smart Thermostat, LG Smart TV, NetGear Arlo Q and many more that are connected in symmetry via internet and can communicate with each other [2] for facilitation of automated services. The smart

appliances in smart building can be categorized as scheduled and non-scheduled [3]. Scheduled devices are thermostatic devices like iron, television, etc. are need based devices and their energy requirements are non-predictable and dynamic in nature. Non-scheduled appliances like refrigerator, lighting devices have predefined, predictable and static energy needs. The sensors connected with such smart home devices or appliances collect the raw data and transfer it to the cloud data centers via gateway. This data is further processed into patterns which are subsequently reverted to the respective appliance for responding to the desired service within limited time frame (as depicted in Figure 1). Hence, realizing the actual definition of smart appliance and making it more adaptive in making accurate decisions. Though, the concept of smart home is highly automated in its implementation and operation, still storage of generated massive data is a big challenge before service providers. The total cost of using the smart home architecture is directly proportional to the cost of energy used in storing and analyzing the raw data on cloud data centers. Mitigation of this cost highly depends upon the usage of number of active servers or bins in cloud data centres and can be efficaciously reduced by turning off extra bins, limiting the usage of extra bins for service extraction [4]. Therefore, a bin-packing algorithm comprehends best solution to easy adoption to smart architectures besides reducing the energy consumption on cloud. It is also anticipated to be a best remedy to United Nation's sustainable development goal initiative.

Perhaps, the increasing count of cloud clients on the remaining limited active server bins have deviated from the actual objective of user satisfaction and can lead to distrust of clients on cloud paradigm. This has given birth to the concept of virtualization [5] that creates virtual machine instances on physical server and provides the requisite service on the pay-per-use basis as shown in Figure 2.

All the virtual machines created on any physical host share its computing resources. Virtualization not only reduces the total energy but also helps in proper resource utilization. However, it has increased the task of system administrator to handle many virtual machines on one server. The virtualization concept is primarily being implemented to reduce the number of active servers to save energy consumption and can be well achieved by employing the techniques of bin-packing. Therefore, the term that is coined as bin-packing packs virtual machines into active bins and delivers the usage-based services to multiple clients simultaneously. The process is inclined towards minimizing number of active bins to reduce energy consumption. Following heuristic based bin-packing algorithms are prevalent in the cloud paradigm [6]:

- First-Fit Algorithm—It assigns the job to the first available virtual machine (VM) on physical host which can provide the required computing capacity.
- Best-Fit Algorithm—It assigns the job to the VM with minimum computing capacity which can provide the required computing capacity.
- Worst-Fit Algorithm—It assigns the job to the VM with maximum computing capacity which can provide the required computing capacity.
- First-Fit Decreasing (FFD), Best-Fit Decreasing (BFD) and Worst-Fit Decreasing (WFD) algorithms—These algorithms arrange all virtual machines in decreasing order of computing capacity before assigning the job.

The research work carried out primarily focuses on identifying and exploring solutions in the domain of bin-packing for leveraging the benefits it offers in the paradigm of cloud computing, primarily, energy mitigation. Therefore, following are the underlying contributions of this research work:

- Extensive survey has been conducted in direction of bin-packing.
- The dynamic algorithms are proposed that can cater to the heterogeneous service fulfillment considering the today's client requirements.

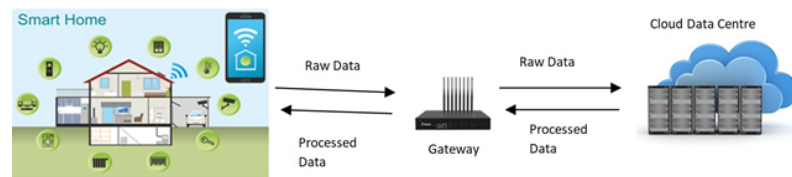


Figure 1. Process flow of cloud based smart home architecture.

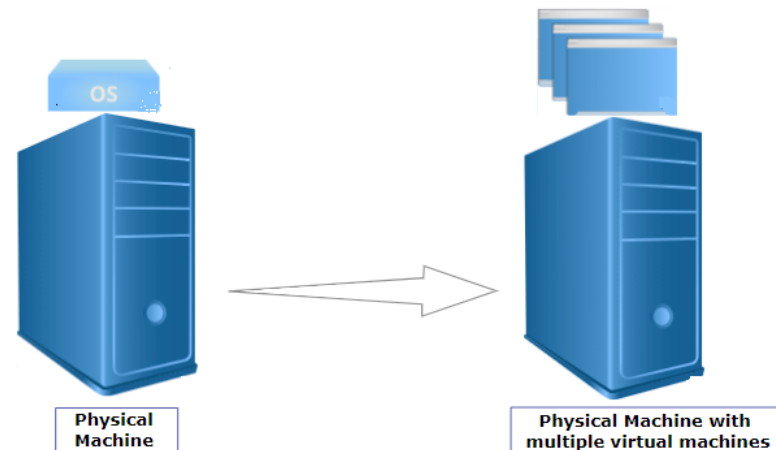


Figure 2. Virtualization in cloud computing.

- Workflow defining the processing of data generated from IoT devices till it reaches the cloud and application of energy mitigation effects is proposed and discussed.
- For adjudging the performance of the proposed system, makespan and resource utilization parameters have been calculated.

The arrangement of contribution in the paper is as follows: Section 2 presents the related works done in area of bin packing techniques and smart home architecture. Section 3 contains the proposed algorithms to implement bin-packing technique using best fit decreasing concept. Section 4 represents the results and discussion details about proposed approaches and Section 5 contains the concluding remarks and future scope. The extensive survey conducted is presented in next section.

2. Related Works

The detailed literature review on bin-packing algorithms in cloud computing and energy management in smart home appliances is mentioned below:

Madumathi, R. et al. [7] proposed “Drip Based Resource Allocation” for assigning user job to the virtual machine using bin-packing technique on cloudSim simulator. It has reduced the number of active physical servers along with higher resource utilization as compared with First-Come-First-Serve (FCFS) and Round Robin Algorithms. Gbaguidi, F.A.R. et al. [8] proposed VM placement algorithm using bin-packing algorithm to use minimum number of active machines. The algorithm was compared with the brute force method. VMware hypervisor ESXi 5.5 was used for virtualization. Kumaraswamy, S. et al. [9] addressed the virtual machine placement issue. Three algorithms were designed “Approximation algorithm”, “Polynomial Time Approximation Scheme (PTAS) algorithm” and “Randomized algorithm”. Berndt, S. et al. [10] considered fully dynamic bin-packing problem to minimize number of servers and amount of repacking. The term “migration factor” was defined as total size of repacked items divided by size of an arriving or departing item. Gupta, N. et al. [11] discussed the use of virtualization and VM migration in cloud computing. The paper compared the existing bin-packing algorithms and concluded that “EUBFD-Energy utilized BFD Packing Algorithm” is more efficient in reducing the number of physical servers and hence, the total energy consumption of the system. Karwayun, R. et al. [12] proposed dynamic resource

allocation scheme for cloud computing which includes first-fit bin-packing for allocating VM, VM migration scheduler and energy consumption calculator.

Gua, L. et al. [13] addressed the resource allocation strategies in cloud and proposed “Multi-dimensional cloud resource dynamic allocation model (MDCRA)” based on vector bin packing problem. “MinTotal Dynamic Bin Packing problem” was explored in [14] explored that aims to minimize the total cost of the servers. They also proposed “Hybrid First Fit” algorithm which uses the different size of objects. Ren, R. et al. [15] solved the “MinUsageTime DBP problem” and established a new improved upper bound. They concluded that first fit is best suited for the mentioned problem. Kamali, S. [16] proposed “HarmonicMix” which used live migration to improve packing of user job on servers. The algorithm was compared with “Bin Packing with Variable Sized Items (VISBP)” to prove its effectiveness. Gupta, K. et al. [17] presented the algorithms used for virtual machine migration techniques. The paper also compared these algorithms in terms of energy consumption.

A new architecture “micro controller based smart home energy management system (SHEMS) architecture” was proposed by [18] whose purpose is to switch off the devices when they are not used. They also suggested that use of renewable resources can improve energy utilization. TAĀTAN, M. [19] presented “Smart Energy Management (SEM) system”, based on NodeMCU and Android for decreasing energy use by monitoring the electrical devices time to time and report the suitable time of appliance usage. Singh, P.P. et al. [20] discussed the importance of electrical appliance energy management. The paper also discussed the future scope of smart home architecture along with data security and energy consumption. Artificial intelligence role in IoT was also explained.

An approach “smart home energy management system (HEMS)” was studied in [21] which can monitor the current usage of electrical appliances and their history records as well through meta-heuristics algorithms. The scheme can decrease the energy consumption. Al-Ali, A.R. et al. [22] combined the uses of IoT and Big data analysis. The paper proposed Energy Management System where each home appliance is connected with IoT device that acts like data acquisition module and transfer data to server for processing.

Ali, S.A. et al. [23] presented the review of energy consumption issues and probable solutions that can happen in cloud data centers. The paper analyzed various energy efficient techniques like VM selection, VM placement, VM allocation etc. with their findings. The various issues with dynamic energy management in cloud data centers were well explained in [24]. The paper suggested using some hybrid solution that can combine hardware and resource management concepts to affect the energy consumption.

Arshad, R. et al. [25] discussed the energy requirement by IoT devices, their drawbacks and use of Green-IoT in the area. The paper also discussed the strategies for designing energy efficient data centers, energy efficient transmission of data from sensors and design of energy efficient policies. Kanwal, S. [26] presented “Genetic Algorithm (GA) based leader election (GLEA)” approach to analyze the nodes in cloud data centers and found the approach to be memory efficient with minimum execution time. A hybrid approach “Integration of VMs Auto-Scaling with Live Migration” was proposed in [27] to combine auto-scaling and live-migration to calculate the total cost of dynamic VMs. This approach is also able to predict the VM workload for better performance.

As discovered from literature review, the cloud environment can process the raw data to derive useful conclusions for problem at hand and it is imperative to reduce the enormous heat exhibited from the devices. Among the various techniques used to reduce the energy consumption and thus, the cost of using cloud environment, bin packing techniques are majorly used. So next section explicates the proposed bin-packing approaches.

3. Proposed Algorithms

Envisioned with the aim to offer better solutions that can mitigate energy consumption, dynamic approaches are proposed and has been discussed in this section. The proposed approaches have following characteristics:

- After a fixed interval of time, new virtual machines can be created and presented for allotment on basis of new job requests.
- Each virtual machine is instantly created by hypervisor for one job request.
- Each incoming request has a time-frame associated with it that fixes up the life frame of the virtual machine and the VM gets dissolved once the request gets completely executed on the server making space for another incoming request.
- After each run, energy is computed.
- Resource Utilization and Makespan are calculated once all the virtual machines get dissolved and all requests are serviced.
- Inputs to all the approaches are CPU capacity and power requirement of each server, CPU capacity and execution time frame for each virtual machine.

The terminologies used in algorithms are shown in Table 1.

Table 1. List of Abbreviations.

Abbreviation	Definition
P_{max}	Maximum power capacity of server i
CPU_i	Computing capacity of server i
CPU_{v_j}	Computing capacity of VM j
$U_i(F,t)$	Computing capacity utilization factor of server i as a function of placement and time
$P_i(F,t)$	Utilized power of server i as a function of placement and time
TVM_j	Time taken by jth VM on server for execution
Res_util	Average Resource utilization of data centre
T	Fixed interval of time, here, $T = 10$ s
TVM_max_i	Maximum time used by any VM in ith run

The mathematical illustration of the used equations is as follows:

- Computation of Utilization Factor [17]: The utilization factor refers to the utilization of servers and it is calculated by dividing the CPU capacity used by all VMs running on server and total CPU capacity of server. The formula of utilization factor for server i is computed as a function of placement 'F' and time 't' as shown in below:

$$U_i(F, t) = (CPU_util_i) / (CPU_i) \quad (1)$$

CPU_util_i is the total CPU capacity used by all VMs on i^{th} server and CPU_i is the CPU capacity of the server.

- Computation of Power Consumption [17]: It refers to the energy consumed by N servers in data centers and is calculated for one server at a time and gets summed up for computing total power consumption of a data centre and is computed after VM allotment 'F' at a specific time 't' as shown below:

$$Total_power_consumption = \sum_{i=1}^N P_i(F, t) \quad (2)$$

where,

$$P_i(F, t) = 0.7 * P_{max} + (0.3 * P_{max} * U_i(F, t)) \quad (3)$$

Power consumption of a server is calculated by using maximum power requirement of server and utilization factor of the server.

- Computation of Resource Utilization [28]: It refers to total time the servers have been utilized efficiently in the data centre and is computed as shown below:

$$Res_util = \sum_{i=1}^N Res_util_i \quad (4)$$

where,

$$Res_{util_i} = \frac{\sum_{j=1}^M TVM_j}{makespan_i * M} \quad (5)$$

M is the number of VMs executing on i th server

- Computation of Makespan [28]: It refers to time frame between submission of 1st job entering the system till completion of last job and is computed by taking maximum time by any VM in each run R and makespan on server i is calculated as shown below:

$$makespan_i = \sum_{j=1}^R TVM_max_j \quad (6)$$

The mathematical equations have been used by following proposed approaches for calculation of energy consumption:

Dynamic Low Perturbation Bin Packing (DLPBP)—In inception, the input is taken for N servers and M virtual machines and hypervisor creates virtual machines. Then, the approach of DLPBP gets initialized and makes allotment of M VMs onto N servers. Subsequently, after every T seconds, new job arrives into the system and energy is calculated for current requests being processed into the system. Once, the execution of all incoming requests is completed, resource utilization and makespan are calculated for computing the overall data center's performance. The approach does not consider any order for virtual machines and servers as a pre-step. The Algorithm 1 depiction for the proposed approach is as follows:

Algorithm 1 DLPBP

1. Input CPU capacity and Power requirement for N Servers.
 2. **Repeat steps 3 to 5** until no VM is left for execution.
 3. Input CPU capacity and execution time frame for M Virtual Machines.
 4. Assign VM to server on basis of CPU value of VM and server and create a Map <CPU of server, <CPU value of VM, Time value of VM>>.
 5. Compute energy/power utilized by data center for each run, overall resource utilization and makespan as $\sum_{i=1}^N (P_i(F, t), Res_{util_i}, Makespan_i)$.
-

Dynamic Best Fit Decreasing (DBFD)—The input is taken for N servers and M virtual machines. Servers are rearranged in ascending order of CPU capacity and virtual machines are rearranged in descending order of CPU capacity as a pre-step before allotment of M VMs onto N servers. For calculation of energy consumption of each server, it checks for incoming requests in each run after every T seconds. After completion of all runs, when no VM request is pending, resource utilization and makespan are calculated to know about overall data centers performance. The Algorithm 2 approach is presented below:

Algorithm 2 DBFD

1. Input CPU capacity and Power requirement for N Servers.
 2. Servers are arranged in ascending order of CPU capacity.
 3. Virtual machines are arranged in descending order of CPU capacity.
 4. **Repeat steps 5 to 7** until no VM is left for execution.
 5. Input CPU capacity and execution time frame for M Virtual Machines.
 6. Assign VM to server on basis of CPU value of VM and server and create a Map <CPU of server, <CPU value of VM, Time value of VM>>.
 7. Compute energy/power utilized by data center for each run, overall resource utilization and makespan as $\sum_{i=1}^N (P_i(F, t), Res_{util_i}, Makespan_i)$.
-

Dynamic Power and Computing Capacity Aware Best Fit Decreasing (DPCA-BFD)—In comparison to the earlier depicted approaches, this heuristic considers the power re-

quirement of machines as well. In inception, the input is taken for N servers and M virtual machines. Servers are arranged in ascending order of ratio of CPU capacity/power requirement and virtual machines are arranged in descending order of CPU capacity and makes allotment of M VMs onto N servers and calculation of makespan, energy and resource utilization is done. The Algorithm 3 is as follows:

Algorithm 3 DPCA-BFD

1. Input CPU capacity and Power requirement for N Servers.
 2. Servers are arranged in ascending order of ratio of CPU capacity/power requirement.
 3. Virtual machines are arranged in descending order of CPU capacity.
 4. **Repeat steps 5 to 7** until no VM is left for execution.
 5. Input CPU capacity and execution time frame for M Virtual Machines.
 6. Assign VM to server on basis of CPU value of VM and server and create a Map <CPU of server, <CPU value of VM, Time value of VM>>.
 7. Compute energy/power utilized by data center for each run, overall resource utilization and makespan as $\sum_{i=1}^N (P_i(F, t), Res_util_i, Makespan_i)$.
-

Dynamic Energy-Utilized Best Fit Decreasing (DEU-BFD)—The input is taken for N servers and M virtual machines. Servers are arranged in ascending order of power requirement and virtual machines are arranged in ascending order of CPU capacity and makes allotment of M VMs onto N servers. Subsequently, makespan, energy and resource utilization are computed. The Algorithm 4 is as follows:

Algorithm 4 DEU-BFD

1. Input CPU capacity and Power requirement for N Servers.
 2. Servers are arranged in ascending order of power requirement.
 3. Virtual machines are arranged in ascending order of CPU capacity.
 4. **Repeat steps 5 to 7** until no VM is left for execution.
 5. Input CPU capacity and execution time frame for M Virtual Machines.
 6. Assign VM to server on basis of CPU value of VM and server and create a Map < CPU of server, <CPU value of VM, Time value of VM>>.
 7. Compute energy/power utilized by data center for each run, overall resource utilization and makespan as $\sum_{i=1}^N (P_i(F, t), Res_util_i, Makespan_i)$.
-

The proposed approaches compute energy at the cloud level for the incoming requests that are generated by various clients via service providers. The basis of data collection for processing are the various devices that are deployed at user's end where massive and raw data is being generated and redirected to the cloud environment. The complete flow of this data redirection specifically for a smart home is shown in Figure 3, wherein the data originating from the sensor devices reaches to the cloud through gateway and is returned in the form of processed information by cloud data centres.

The sensors collect the raw data in symmetric and systematic form and forward it to the gateway on the edge of the smart home architecture. The gateway transfers this data to cloud data centers for further analysis. The server in cloud data center analyzes the pattern of data and proposed dynamic approaches compute energy consumption, resource utilization and makespan to know about the overall performance of data centres. The decisions made based on symmetric patterns of raw data is transferred back to the gateway which is further sent to smart home appliances for providing response to the user.

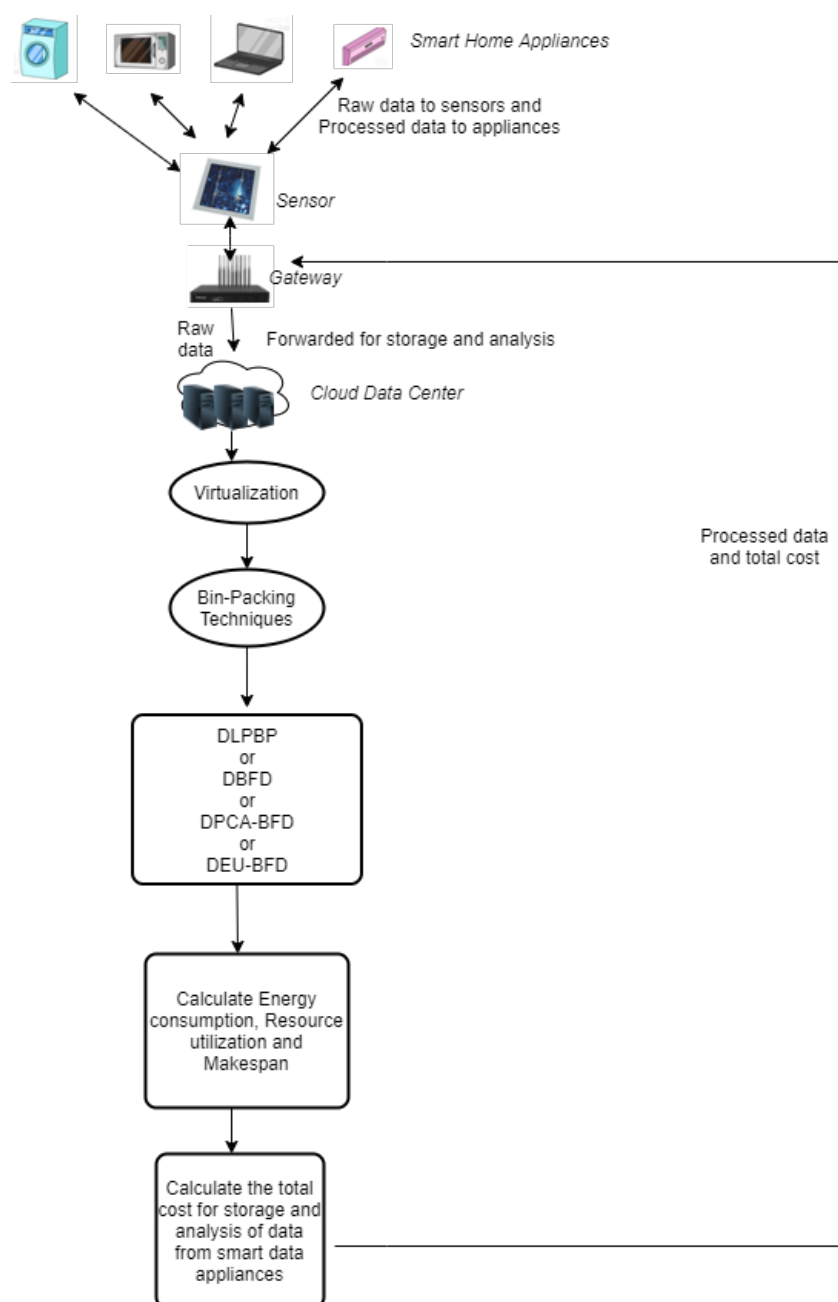


Figure 3. Complete flow of data between smart home appliances and cloud computing.

4. Results and Discussion

The static bin-packing algorithms are converted into dynamic to handle intermittent job request by the customers. The algorithms calculate the energy consumption, resource utilization and makespan for completing all the job requests on cloud data centers. The total cost of processing the data on cloud affects proportionally the total cost of using smart home architecture.

To calculate the results of algorithm, simulation is done on JAVA programming language. The initial values for simulation include minimum memory requirement of virtual machine as 4 MB. The configuration values of virtual machines and servers are: maximum computing capacity of a server is 5000, maximum computing capacity of a virtual machine is 1500 and maximum power capacity of a server is 450 joules. This configuration is constant for all proposed approaches. For the validity of algorithms, simulation runs for three different scenarios, have been carried out. The three different scenarios are: (1) 10 servers and 10 virtual machines, (2) 15 servers and 24 virtual machines and (3) 20 servers and 54 vir-

tual machines. For all the scenarios, CPU capacity and power capacity of servers are taken from user. CPU capacity requirement and time duration needed for execution of virtual machines are taken from user. After rearranging servers and virtual machines according to algorithms, energy consumption, resource utilization and makespan are calculated.

The energy computation values for all scenarios in all algorithms is shown in Table 2 and percentage reduction is shown in Figure 4. Besides, energy computation, other factors like resource utilization and makespan affects the performance of a data centre and thus, are imperative to be considered. The resource utilization factor [28] is how the resources are well utilized for processing of data on cloud data centers. The higher the resource utilization factor, higher is the efficiency of resources. All the proposed bin-packing algorithms are evaluated in terms of resource utilization factor and results are presented in Table 3 and percentage increment is shown in Figure 5. The makespan is other parameter which acknowledges the utilization of servers by calculating the total working time of a server i.e., the time from first virtual machine allotment till the last virtual machine completion. Table 4 depicts the calculated makespan for all bin-packing algorithms for performed simulations and Figure 6 shows the percentage reduction.

Table 2. Energy consumption (joules).

	DLPBP	DBFD	DPCA-BFD	DEU-BFD
10 Servers and 10 VMs	1400.2	1567.21	1117.25	1117.11
15 Servers and 24 VMs	2607.06	2686.12	2155.93	2176.77
20 Servers and 54 VMs	4436.45	4561.68	4051.38	4029.41

Table 3. Resource utilization.

	DLPBP	DBFD	DPCA-BFD	DEU-BFD
10 Servers and 10 VMs	4.5	4.8	2.5	4.9
15 Servers and 24 VMs	8.2	7.9	5.6	8.6
20 Servers and 54 VMs	13.3	14.3	12.9	15.7

Table 4. Makespan.

	DLPBP	DBFD	DPCA-BFD	DEU-BFD
10 Servers and 10 VMs	390	440	250	350
15 Servers and 24 VMs	765	691	532	584
20 Servers and 54 VMs	1437	1361	1111	1194

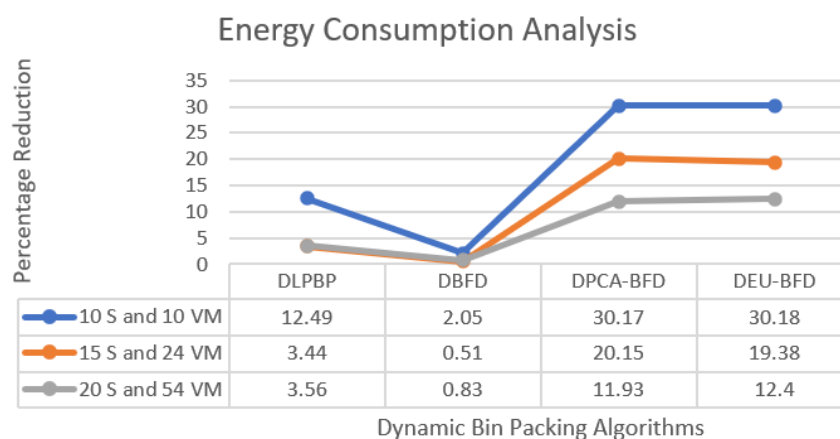


Figure 4. Energy consumption.

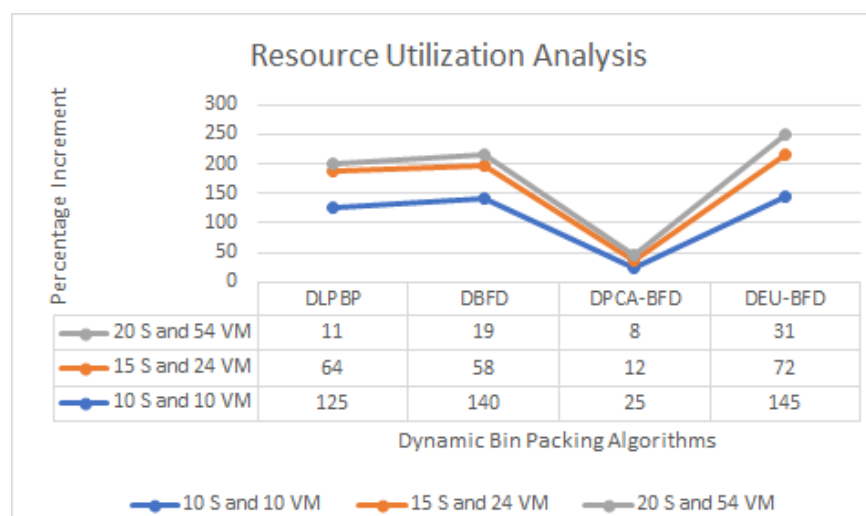


Figure 5. Resource utilization.

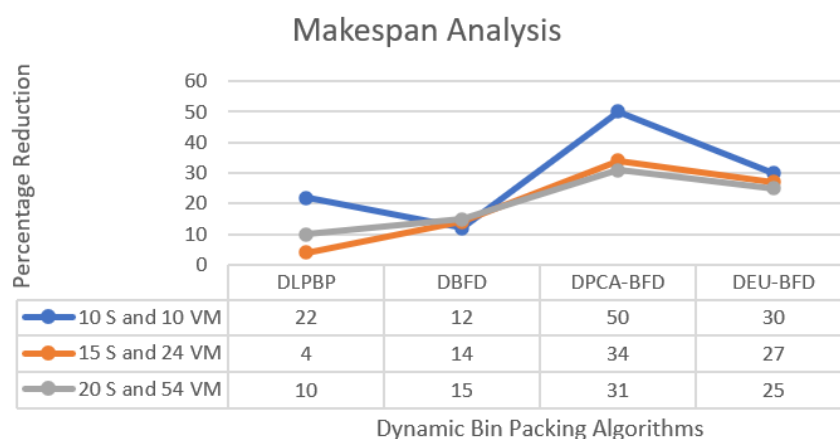


Figure 6. Makespan.

In energy consumption analysis graph, reduction in energy is presented in a form of percentage. Out of three scenarios, DEU-BFD generates highest percentage reduction in 2 scenarios and DPCA-BFD in 1 scenario. In resource utilization analysis graph, increment in the utilization of server is presented in form of percentage. In all scenarios, DEU-BFD is utilizing the server better than other algorithms. In makespan analysis graph, reduction in makespan of server is presented in form of percentage. In all scenarios, DPCA-BFD is giving highest reduction in active time period of server. So the overall performance of DEU-BFD and DPCA-BFD is more effective than DLPBP and DBFD in respect to three parameters-energy consumption, resource utilization and makespan.

The existing algorithms for bin packing techniques are static [17] and does not fit in current heterogeneous and dynamic environment where job request can arrive at cloud data centers at any time. This has arisen a need to propose solutions that can cater to ever-lasting need of clients of service extraction in different time-zones without considering any geographical disparities. Thus, the present study has proposed dynamic mechanisms that offers following advantages over the static approaches:

- The proposed dynamic approaches has associated a heterogeneous execution time frame with each virtual machine corresponding to each incoming request so that the virtual machine can exit the system once the job is completed which can make the server room available for another incoming request, thus, can lead to better utilization of the available server computing resources in real-time. The absence of

such mechanism in static approaches has otherwise forced the system to turn on new server machines in case new jobs have popped in once all jobs have been executed.

- The system permits entry of new job requests after a fixed interval of time. Each time a user issues a new request, a new virtual machine is created on demand and can be allocated to the available server machine as the dynamic proposed heuristic provisions flexibility of handling requests at different time-zones, thus reduces the problem of starvation upto some extent.

The summary of new features added in dynamic algorithms is presented in Table 5.

Table 5. Comparison between static and proposed dynamic algorithms.

Existing Static Algorithms	Proposed Dynamic Algorithms
Hypervisor has to dissolve the virtual machine on completion.	Algorithm itself dissolve the virtual machine by regularly checking the execution time period.
Hypervisor has to take care of new job requests entry and calling of algorithm for the execution of the same.	Algorithm itself asks for new job request after a specified time period.

5. Conclusions

Smart home architecture is the future of architecture design. The collaboration of smart home appliances with cloud computing can make data storage and analysis, an effortless and symmetric task. The raw data generated by smart home appliances need to be analyzed rapidly to get prompt response by them. If the task of storage and processing of data is shifted on cloud, then, the total cost of using smart home architecture can be minimized. As this increasing data requires huge resource management, so, it poses a threat of increased energy consumption. Therefore, envisioned with the aim to mitigate the effects of this ever-increasing energy needs of data centre, this paper proposes dynamic bin packing solutions for reducing energy consumption in a data centre capable of accepting massive data from smart architectures. The already existing and standard static bin packing are modified to get more scope of improvement. The main difference between existing and proposed algorithm is that the existing algorithms were static and were not according to current heterogeneous environment. All proposed heuristics are significant in reducing energy consumption considering the present heterogeneous service demands posed by various clients but in maximum runs, DEU-BFD is consuming less energy and is better utilizing the available resources and DPCA-BFD is generating better results in comparison to others in terms of makespan. Thus, the proposed approaches are scalable and suitable for smart building architecture. In future scope, an energy efficient switcher can be developed that can switch between algorithms to give best results in form of energy consumption. The switcher can choose the dynamic algorithm which allot the VM with minimum energy and finally implement the chosen one.

Author Contributions: Conceptualization, N.G. and K.G.; methodology, S.R.; software, D.K.; validation, N.G. and K.G.; formal analysis, A.Z.; investigation, N.G.; resources, K.G.; data curation, S.R.; writing—original draft preparation, N.G.; writing—review and editing, K.G. and D.K.; visualization, N.G.; supervision, K.G.; project administration, S.R.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Taif university Researchers Supporting Project Number (TURSP-2020/114), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Acknowledgments: This work was supported by Taif university Researchers Supporting Project Number (TURSP-2020/114), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: Authors do not have any conflict of interest.

References

- Hassan, R.; Qamar, F.; Hasan, M.K.; Aman, A.H.M.; Ahmed, A.S. Internet of things and its applications: A comprehensive survey. *Symmetry* **2020**, *12*, 1–29. [\[CrossRef\]](#)
- Mocrii, D.; Chen, Y.; Musilek, P. IoT-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet Thing* **2018**, *1–2*, 81–98. [\[CrossRef\]](#)
- Zamanloo, S.; Askarian Abyaneh, H.; Nafisi, H.; Azizi, M. Optimal two-level active and reactive energy management of residential appliances in smart homes. *Sustain. Cities Soc.* **2021**, *71*, 102972. [\[CrossRef\]](#)
- Savu, L. Cloud computing: Deployment models, delivery models, risks and research challanges. In Proceedings of the 2011 International Conference on Computer and Management, CAMAN 2011, Sydney, Australia, 2–4 May 2011. [\[CrossRef\]](#)
- Xing, Y.; Zhan, Y. Virtualization and cloud computing. *Lect. Notes Electr. Eng.* **2012**, *143*, 305–312. [\[CrossRef\]](#)
- Kumaraswamy, S.; Nair, M.K. Bin packing algorithms for virtual machine placement in cloud computing: A review. *Int. J. Electr. Comput. Eng.* **2019**, *9*, 512–524. [\[CrossRef\]](#)
- Madhumathi, R.; Radhakrishnan, R.; Balagopalan, A.S. Dynamic resource allocation in cloud using bin-packing technique. In Proceedings of the ICACCS 2015—Proceedings of the 2nd International Conference on Advanced Computing and Communication Systems, Coimbatore, India, 5–7 January 2015; pp. 5–8. [\[CrossRef\]](#)
- Gbaguidi, F.A.R.; Boumerdassi, S.; Ezin, E.C. Adapted Bin Packing Algorithm for Virtuals Machines Placement into Datacenters. *Int. Conf. Cloud Comput.* **2017**, 69–80. [\[CrossRef\]](#)
- Kumaraswamy, S.; Nair, M.K. Virtual machine placement in distributed cloud centres using bin packing algorithm. *Int. J. Grid Util. Comput.* **2018**, *9*, 396–403. [\[CrossRef\]](#)
- Berndt, S.; Jansen, K.; Klein, K.M. Fully dynamic bin packing revisited. *Math. Program.* **2020**, *179*, 109–155. [\[CrossRef\]](#)
- Gupta, N.; Gupta, K.; Singh, A.; Kaur, A.; Gupta, D. Cloud computing: A walk through in models, challenges, and energy solutions. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, 1022. [\[CrossRef\]](#)
- Karwayun, R. A Dynamic Energy efficient Resource Allocation Scheme for Heterogeneous clouds using Bin-Packing Heuristics. *Algorithms* **2018**, *9*, 15–24.
- Guo, L.; Du, P.; Razaque, A.; Almiani, M.; Al-Rahayfeh, A. Energy saving and maximize utilization cloud resources allocation via online multi-dimensional vector bin packing. In Proceedings of the 2018 5th International Conference on Software Defined Systems, SDS 2018, Barcelona, Spain, 23–26 April 2018; pp. 160–165. [\[CrossRef\]](#)
- Li, Y.; Tang, X.; Cai, W. Dynamic Bin Packing for On-Demand Cloud Resource Allocation. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 157–170. [\[CrossRef\]](#)
- Ren, R.; Tang, X.; Li, Y.; Cai, W. Competitiveness of dynamic bin packing for online cloud server allocation. *IEEE/ACM Trans. Netw.* **2017**, *25*, 1324–1331. [\[CrossRef\]](#)
- Kamali, S. Efficient bin packing algorithms for resource provisioning in the cloud. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2016; Volume 9511, pp. 84–98. [\[CrossRef\]](#)
- Gupta, K.; Katiyar, V. Energy Aware Virtual Machine Migration Techniques for Cloud Environment. *Int. J. Comput. Appl.* **2016**, *141*. [\[CrossRef\]](#)
- Chouaib, B.; Lakhdar, D.; Lokmane, Z. Smart home energy management system architecture using IoT. *ACM Int. Conf. Proc. Ser.* **2019**. [\[CrossRef\]](#)
- Taştan, M. Internet of things based smart energy management for smart home. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 2781–2798. [\[CrossRef\]](#)
- Singh, P.P.; Khosla, P.K.; Mittal, M. *Energy Conservation in IoT-Based Smart Home and Its Automation*; Springer: Singapore, 2019; Volume 206, pp. 155–177. [\[CrossRef\]](#)
- Yang, T.Y.; Yang, C.S.; Sung, T.W. An Intelligent Energy Management Scheme with Monitoring and Scheduling Approach for IoT Applications in Smart Home. In Proceedings of the 2015 3rd International Conference on Robot, Vision and Signal Processing, RVSP 2015, Kaohsiung, China, 18–20 November 2015; pp. 216–219. [\[CrossRef\]](#)
- Zualkernan, I.A.; Rashid, M.; Gupta, R.; Alikarar, M. Smart Home Big Data. *IEEE Trans. Consum. Electron.* **2017**, *63*, 426–434.
- Ali, S.A.; Affan, M.; Alam, M. A Study of Efficient Energy Management Techniques for Cloud Computing Environment. *arXiv* **2018**, arXiv:1810.07458.
- T, V.; S, M.S.B. Dynamic Energy Management In Cloud Data Centers: A Survey. *Int. J. Cloud Comput. Serv. Archit.* **2013**, *3*, 13–26. [\[CrossRef\]](#)
- Arshad, R.; Zahoor, S.; Shah, M.A.; Wahid, A.; Yu, H. Green IoT: An investigation on energy saving practices for 2020 and beyond. *IEEE Access* **2017**, *5*, 15667–15681. [\[CrossRef\]](#)
- Kanwal, S.; Iqbal, Z.; Irtaza, A.; Ali, R.; Siddique, K. A genetic based leader election algorithm for IoT cloud data processing. *Comput. Mater. Contin.* **2021**, *68*, 2469–2486. [\[CrossRef\]](#)
- Aldossary, M. A hybrid approach for performance and energy-based cost prediction in clouds. *Comput. Mater. Contin.* **2021**, *68*, 3531–3562. [\[CrossRef\]](#)

-
28. Gupta, K.; Katiyar, V. Deadline-credit aware heuristic for dynamic resource provisioning in virtualised cloud environment. *Int. J. Internet Technol. Secur. Trans.* **2019**, *9*, 390–403. [[CrossRef](#)]