A Novel Local Search-Based Approximation Algorithm to Optimize Virtual Machine Placement With Resource Constraints

Darshan Maheshbhai Shah[1*], *M. Vinayaka* Murthi[2], and *Anand* Kumar[3]

[1]Reva University, Bengaluru, Karnataka - 560064, India
[2]Reva University, Bengaluru, Karnataka - 560064, India
[3]M.S. Engineering College, Bengaluru, Karnataka - 560064, India

Abstract. Many problems in cloud computing are not solvable in polynomial time and only option left is to choose approximate solution instead of optimum. Virtual Machine placement is one of such problem with resource constraints in which overall objective is to optimize multiple resources of hosts during placement process. In this paper we have addressed this problem with large size NP-Hard instances and proposed novel local search-based approximation algorithm. This problem is not yet studied in the research community with NP hard instances. A new proposed algorithm is empirically evaluated with state-of-the-art techniques. and our algorithm has improved placement result by 18% in CPU utilization, 21% in resource contention and 26% in overall resource utilization for benchmark instances collected from azure private cloud data center.

1 Introduction

Cloud computing is a buzz word and available in variety of forms through services and applications. In our day-to-day life it is consumed through mobile devices to robotics and big data to Internet of things. Cloud computing is a on demand platform in which cloud provider served different services dynamically based on user needs and requests. Such need raised many open challenges for cloud providers. One of such challenges is effective resource management. Due to its on-demand platform, users raised request at any time and provider served such requests instantaneously based on service level agreement. This model known as Pay as You Go in which user has to pay only what is consumed by them.

Cloud computing provides services in various forms like Platform as a Service (PaaS), Software as a Service (SaaS) and Infrastructure as a Service (IaaS). Virtual Machine is one of the important IaaS components in which user raised various virtual machine requests as per need of different operating systems types. Later such request has been provisioned by

* Corresponding author: darshan.ms.shah@gmail.com

cloud provider using various orchestrator or scheduler available in cloud data center. Placing Virtual Machine in cloud data center is a complex process in which various resources are changing frequently consumed by different applications and services. During placement process it is highly important that correct host has been identified for the provisioning. If we generalized this problem than it is a classical assignment problem in which various jobs need to assign to various machines such a way that budget is maximized. There are various techniques available in research community like genetic algorithm, ant colony optimization, swarm optimization, various heuristics, integer programming and linear programming. All of these heuristics' techniques are lacking reliability and performance while integer and linear programming techniques are slow in performance for large size instances specially for NP hard instances. In this paper we addressed this gap by providing novel local search-based approximation algorithm for NP hard instances which solves problem efficiently and less time compare to other techniques. Its performance is evaluated on real time azure cloud data center.

All machine resources are covered in our technique which are currently missing in existing techniques where only one dimensions taken into consideration to solve the problem. Such misplacement caused high rise in consolidation and server migration in later stages of resource management. Apart from that it is important to understand the flow of virtual machine placement for an effectiveness in solution. A detailed workflow is mentioned in the following section and we have fully considered this workflow in the design of our new algorithm for integration, stability and easy maintenance.

1.1 Our contribution

This research work has contributed in following areas of the problem,

1. List down limitation of existing techniques for VM placement problem.
2. Proposed novel local search approximation algorithm for NP hard instances to optimize VM placement with resource constraints.
3. Evaluation of new algorithm focusing on various aspects of resource management like resource contention, CPU utilization, multiple resources utilization for shared resources on real time NP hard instances.

The paper has different sections as follows. Section 1 described need and importance of VM placement problem in cloud computing environment. Section 2 described various techniques available to solve VM placement problem with their limitation. Section 3 described VM placement problem along with its workflow. Section 4 described new approximation algorithm in detail. Section 5 described practical evaluation on azure cloud benchmark NP hard instances and their comparison with state-of-the-art techniques. Section 6 conclude with future work.

2 Literature Study

Various research has been conducted in past decade in area of Virtual Machine placement optimization. Different constraints are considered during optimization process to identify gaps in existing literature. We have referred different research work in line with our research problem.

Jumnal & Kumar [1] have used cluster aware approach using crew search algorithm to identify right hosts for optimum allocation. Authors also focused on consolidation process post VMs placement. All the tests were performed on cloudsim simulator with random test data. Two performance criteria were used to measure performance of algorithm, quality of service and service level agreement. Cohen et al. [2] have adopted parallel processing mechanism to optimize VM placement result. A new dynamic randomized algorithm APSR is proposed which focused on describing overhead in communication and reducing deployment attempts. Algorithm was tested on simulated data rather than actual benchmark

results. Overall focus only on one part of VM placement other important constrains like resource constraints were neglected in this approach. Flores et al. [3] have focused on various policy available in cloud data center for resource management. A new algorithm named Policy Aware Algorithm (PAL) is proposed which focus to minimized total communication cost. These policies are different from provider to provider so this approach is very limited for one cloud provider. There is a need to have more generalized policy defined in algorithm so that approach becomes more generic and feasible for more than one cloud provider. Alam et al. [4] have focused on cloud reliability and performance with multi objective optimization. VM placement problem is modeled as integer programming problem and also included network delay in consideration. Mathematical model has evaluated on simulated data and lacking more rigorous evaluation. Also, integer programming is only suitable for instances were all request data is available in advance. This is practically not feasible in all scenarios. Refer to Hassan et al. [5], a novel approach adopted to optimize resources in the cloud data center. Authors have created architecture to accommodate various algorithms known as resource management pipeline approach which used several key performance indicators such as power, network traffic and service level agreement. It has used real-time testbed for algorithm evaluation. A proactive smoothing has improved unnecessary migration by approximately 16 to 49% with this approach. Yavari et al. [6] have focused on energy optimization to maximize resource management. Two heuristics algorithms HET-VC and FET-VC were proposed. Total six parameters are investigated for evaluation. All the empirical evaluation was conducted on simulated data using cloudsim. The authors have considered wide variety of parameters but approach was limited with energy consumption and tested only on random data.

Feng et al. [7] have considered load balancing objective to optimize VM Placement. A new algorithm PMVLP has been proposed to maximize Physical machine workload management. The limitation of this approach is that it is based on greedy approach to solve the problem and all the test conducted on simulation using random cloud data. Gao et al. [8] have focused on heat recirculation for energy and resource optimization. A simulated annealing-based algorithm named SABA has been proposed to lower overall energy consumption in the cloud. The algorithm is more improvised version of existing Simulated algorithms by adding variations in distribution and initial iteration to find solution. It is tested on the related algorithms with random simulated data. Wang et al. [9] have focused on virtual network functions to optimize resource management used in the virtual machine allocation. It has also targeted that latency in the network should also reduce by minimizing the cost has to pay for the leasing machines. A new scheme VPS is prosed by addressing public cloud networks for dynamic request. Scheme focused on rebooting machine time and modeled solution as multi-threaded. Solution is modeled as mixed integer programming to get more actuary in the performance. This approach is focused more on network related optimization but in the machine allocation there are also other factors considered like dependency between various machines, order of request and multiple resources utilization.

3 Virtual machine placement problem

Virtual Machine placement problem with resource constraints is a combinatorial optimization problem in which virtual machine are assigned to various hosts such a way that all requests occupied in minimum hosts and all corresponding resources available on hosts are maximized. The host machine contains various types of resources like CPU, memory, network bandwidth and storage. All these resources are used based on different applications and services deployed by end users over the various virtual machines. In below diagram if we represent host with two resources then it creates two-dimensional space in which each resource represents as vectors. Total utilization is measured by checking total capacity vector in which all consumed resources must be aligned to diagonal of the machine. As stated in Figure 1, each resource occupied space and should be aligned to the diagonal of the machine dimensions. This is only for the two resources but in real scenario machines have many more

such resources and it get more complex to handle. An efficient resource allocation technique should utilize all resources in another terms, allocation should be as near as to the diagonal of machine.

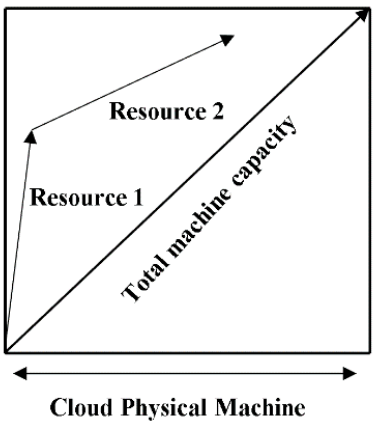

Fig. 1: Virtual machine placement problem with multiple resources

Virtual machine placement is a multi-facet process which goes through various stages during placement process. In Figure 2, it is shown that it validates request, check capacity within the cloud environment, configure virtual machine and maintain the details in the storage. Existing algorithms only focus on configuring machine and missed to considered overall flow of placement. We have addressed this in our optimization strategy and included in the algorithm design.

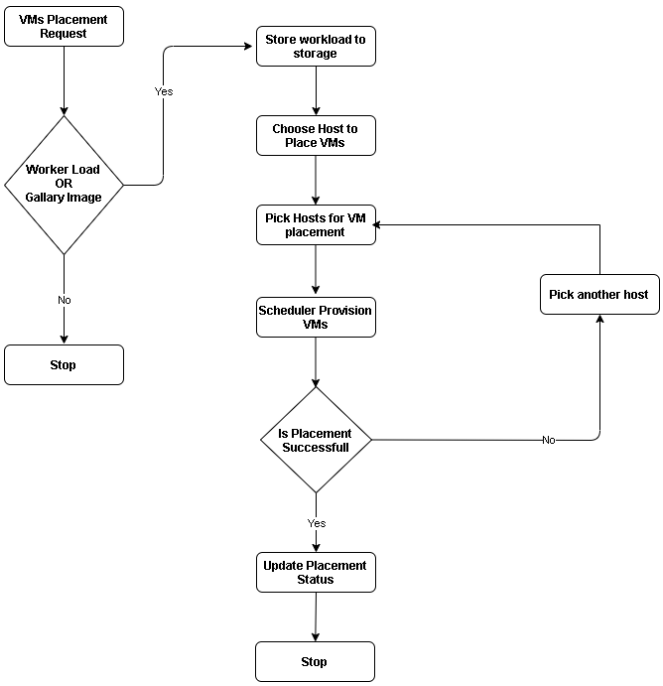

Fig. 2: Virtual machine placement problem flow diagram

4 Novel Local Search Approximation Algorithm

Our novel local search algorithm is divided into two main parts. First part is to build structure to create virtual placement in which each machine is presented as resource cube and second is applying double swapping search technique to find global optimum by searching among various local optimum. First part of our algorithm framework focused on building structure for the search, initializing variables and also perform preprocessing of the data by separating out them in the various sets. From Figure 3, the diagram of our algorithm framework in which algorithm transform its searching process from phase one to three. Various slots are allocated as matrix form and then in second phase search algorithm performed swapping which actually search between various sets to derive better optimum solution within local optimum. The third phase is more like consolidating between sets and finalized allocation of virtual machines over selected hosts for provision.

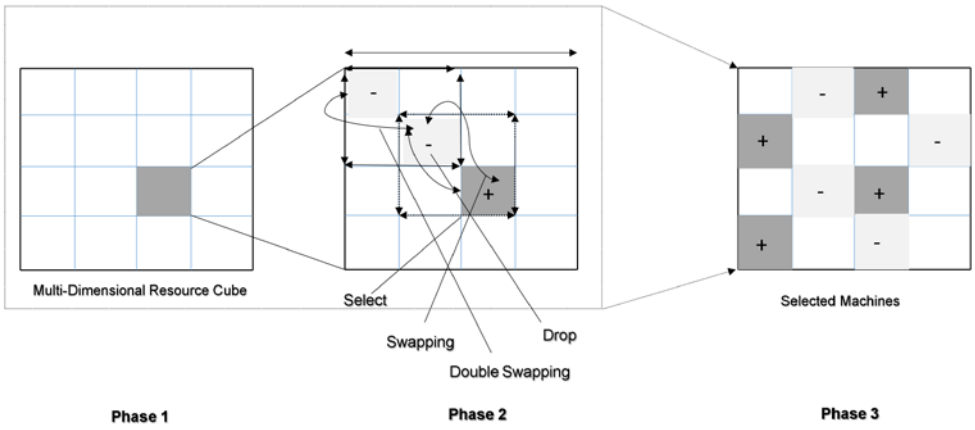

Fig. 3: Novel local search approximation algorithm phases

4.1 Double swapping

Swapping is a process in which values are swap between corresponding values during search process. If new value has better solution than previous one then it is replaced with much higher value. This process keeps running until all such element sets are evaluated. Due to extensive searching, this process is lengthy and time consuming and even in some cases exponential. To avoid this drawback, we have adopted double swapping approach in which multiple elements of the instance sets are exchanged during search option rather than single element. While forming a structure in the first phase, various sets are allocated based on the instances type, their relationships and corresponding usage. During this process, link has been established between sets and common associated sets and then all grouped into main three category sets. Each set internally belonged to parent, ulti-parent and global parent sets.
The same is explained in Figure 4.
Parent: Each element has parent. This association is established based on elements type and similarity in resources. For example, memory intensive machines fall under same host.
Ulti-Parent: Collection of parent belongs to similar group. For example, machines from the racks on same resource group.
Global Parent: Collection of ulti-parent sets belongs to same region. For example, machines fall under same area code and region.

Figure 4 shows the detail view how this sets construct hierarchy during search operation. It helps to identify sets and reduce overall search operation in swapping phase.

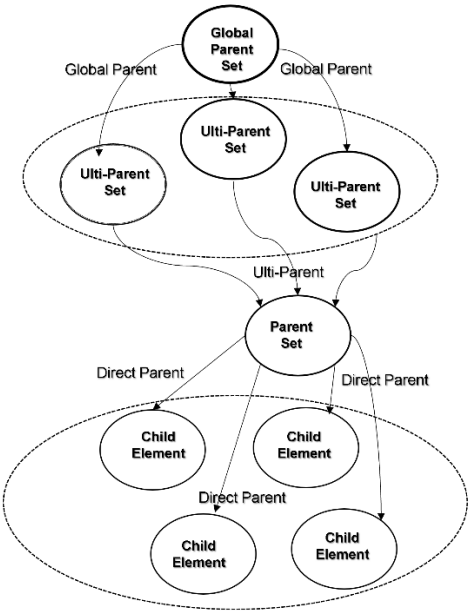

Fig. 4: Novel local search approximation algorithm internal search structure

Below is the main sub routine of local search approximation algorithm in virtual machine configuration.

Algorithm: Novel Local Search Approximation Algorithm

```
Input: hostsList, vmList
Output: VM Placement
function ManageVMPlacement ()
    if k ≤ 1 or S ≤ 1 then              // invalid data or request then return
        return {}
    end if
    // initialize variables
    X ← Hosts;
    Y ← ∅;
    Z ← ∅;
    PQ ← ∅;
    temp ← ∅;
    initialize {Yi, Zi};
    for each VM in VMList do
        find Xi covering related VMs    // find suitable machine using search
        updateStructure (Xi); // update structure with value
        //perform search in sets to better result
        while j < 0 do
            PQ = PQ – X;
            Temp = VMs covered by Xi
              while k < Z do
                  if Zk has satisfied capacity && Zk has more VMs than in set then
                      X = Zk
                  end if
              end while
        end while
      temp = VMs covered by Xi;
      Y = X;
      Z = Y;
    end for
end function
```

5 Empirical evaluation

The algorithm is evaluated on real time azure cloud instances collected from the cloud data center [10-12]. All client machines are installed with Linux and Windows operating systems having 1TB storage and 32GB RAM. While Windows Server and Linux are installed on servers. Azure cloud platform is configured with Azure Pack a well-known private cloud platform. Servers are configured with 1TB storage space and 128GB and 256GB RAM respectively.

5.1 CPU intensive workload

To test this scenario two servers with IBM and Dell are used having with 128 GB RAM. This ensures the overall computation power required for multiple applications are installed on both servers. All these applications are using multiple threading for the operations. During evaluation we have observed that Virtual machines installed on these servers remained constant for our new algorithm due to efficient allocation. While genetic algorithm and ant colony algorithm has increased the CPU utilization up to 90% in almost all instances.

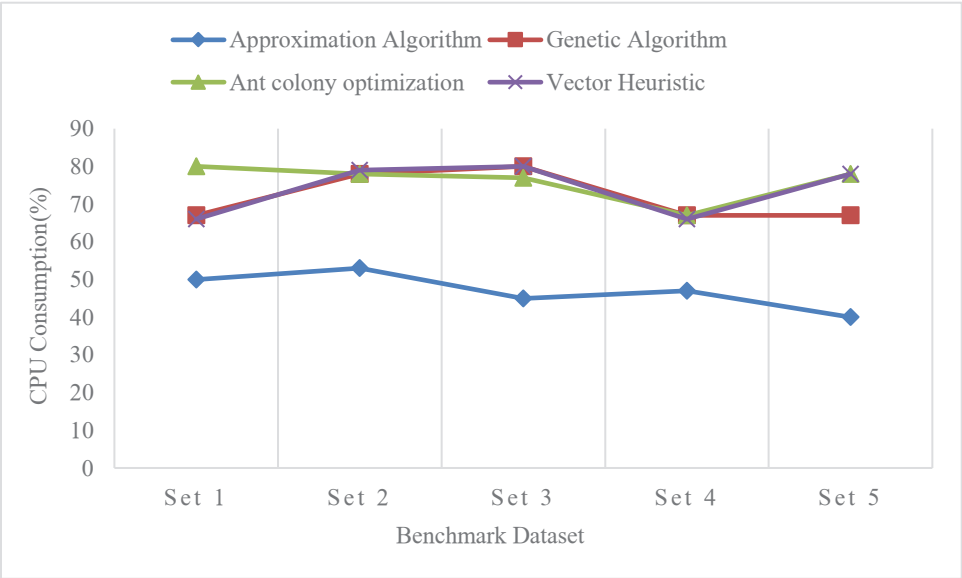

Fig. 5: CPU intensive benchmark instances

5.2 Resource contention workload

Resource contention happened when shared resources are access by more than one resource. This scenario is very common and very challenging for cloud provider to manage it well because there is no clear indication when it will happen. To test our algorithm, we have used azure cloud testbed for evaluation in which we have collected from past six-month data. In this evaluation we have found that during allocation if any of the machine used by more than two services or applications, our new approximation algorithm skips it for allocation. This has provided a boost in our utilization as all such machines are already in use by different service. But in case of other state of the art techniques, such resources are still considered for placement.

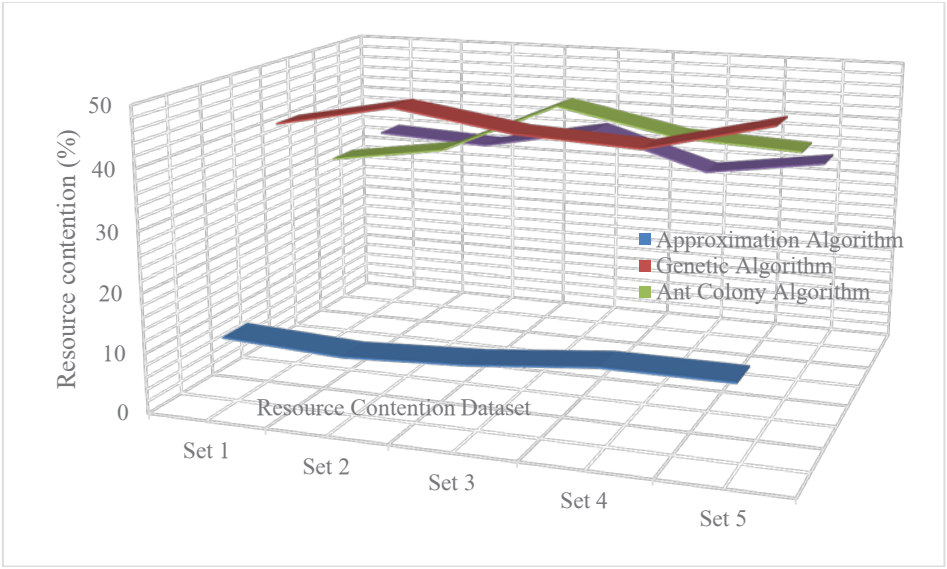

Fig. 6: Resource contention benchmark instances

5.3 Multiple resources workload

Each machine has multiple resources like CPU, memory, network bandwidth, I/O operations. In this evaluation we have taken machine where all types are used. This evaluation helps to evaluate overall utilization of the machine when different applications and services are used within virtual machine.

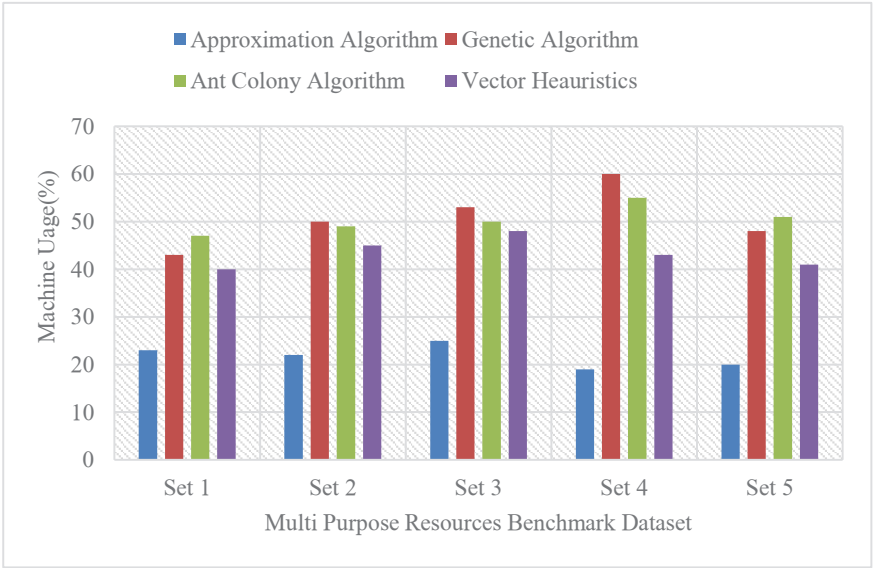

Fig. 7: Multiple resources benchmark instances

5.4 Measure significance of empirical test

We have conducted Wilcoxon Signed rank test to signify our empirical evaluation. The significance level $\alpha = 0.01$ through-out the test. Figure 8 shows overall test result with different algorithms. Here p-value indicates the probability of null hypothesis against the alternative hypothesis to the hypothesis being tested. For all the test, p-value are less than

the α and it means that there is a significant improvement in the result.

Comparison by Machine usage	p-value
Usage (Approximation Algorithm) < Usage (Genetic algorithm)	$9.8 \times 10{-}4$
Usage (Approximation Algorithm) < Usage (Ant colony algorithm)	$9.8 \times 10{-}4$
Usage (Approximation Algorithm) < Usage (Vector heuristics)	$9.8 \times 10{-}4$

Fig. 8: Wilcoxon signed rank test result

We have conducted various types of scenarios to test the performance of algorithm with other existing algorithms. Allof the testing has been performed on benchmark cloud instances. Wilcoxon Signed Rank test shown precision in our algorithm results by surpassing all existing techniques.

6 Conclusion and Future Work

In this paper we have proposed new local search-based approximation algorithm. Various existing techniques were evaluated with their limitation which helps to design better placement algorithm. We have also described our approximation algorithm various phases and its internal search structure which describes how elements are inter-related and connected with different elements in form of sets. Our empirical evaluation has shown that our proposed algorithm has surpassed all other heuristics technique. In future we are planned to use cache mechanism to improvise search performance in real time so that we can reduce iteration in search and also optimized memory allocation for storing large amount of data.

References

1. A. Jumnal, D.S.M. Kumar, ICISC, 266 (2020).
2. I. Cohen, G. Einziger, M. Goldstein, Y. Sa'ar, G. Scalosub, E. Waisbard, INFOCOM WKSHPS 2020, 1298 (2020)
3. H. Flores, V. Tran, B. Tang, IEEE INFOCOM 2020, 2549 (2020).
4. A.B.M.B. Alam, T. Halabi, A. Haque, M. Zulkernine, 2020 IEEE ICC, 1 (2020).
5. H.A. Hassan, A.I. Maiyza, W.M. Sheta, J. Cloud Comp. **9**, 63 (2020).
6. M. Yavari, A.G. Rahbar, M. Fathi, J. Cloud Comp. B, 13 (2019).
7. H. Feng, Y. Deng, Y. Zhou, 2020 Des. Auto. Test Euro.Conf. Exhib. 626 (2020).
8. T. Gao, X. Li, Y. Wu, W. Zou, S. Huang, IEEE Trans. Comm. **68**, 4946 (2020).
9. S.N. Wang, H.X. Gu, G. Wu, Proc. IEEE 8th Int. Conf. Netw. Archit. Stor. 331 (2013)
10. S. Anoep, C. Dumitrescu, D. Epema, A. Iosup, M. Jan, H. Li, L.Wolters, The Grid Workloads Archive: Bitbrains. http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains. Accessed 20 Mar 2018
11. https://docs.microsoft.com/en-us/azure/virtual-machines/windows/compute-benchmark-scores
12. R. Aljamal, A. El-Mousa, F. Jubair, 11th Int. Conf. Inf. Comm. Sys. 382 (2020).