



HAL
open science

A Genetic Algorithm for Power-Aware Virtual Machine Allocation in Private Cloud

Nguyen Quang-Hung, Pham Dac Nien, Nguyen Hoai Nam, Nguyen Huynh Tuong, Nam Thoai

► **To cite this version:**

Nguyen Quang-Hung, Pham Dac Nien, Nguyen Hoai Nam, Nguyen Huynh Tuong, Nam Thoai. A Genetic Algorithm for Power-Aware Virtual Machine Allocation in Private Cloud. 1st International Conference on Information and Communication Technology (ICT-EurAsia), Mar 2013, Yogyakarta, Indonesia. pp.183-191, 10.1007/978-3-642-36818-9_19 . hal-01480228

HAL Id: hal-01480228

<https://inria.hal.science/hal-01480228>

Submitted on 1 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Genetic Algorithm for Power-Aware Virtual Machine Allocation in Private Cloud

Nguyen Quang-Hung¹, Pham Dac Nien², Nguyen Hoai Nam²,
Nguyen Huynh Tuong¹, Nam Thoai¹

Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology
268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
¹{*hungnq2, htnguyen, nam*}@*cse.hcmut.edu.vn*
²{*50801500, 50801308*}@*stu.hcmut.edu.vn*

Abstract. Energy efficiency has become an important measurement of scheduling algorithm for private cloud. The challenge is trade-off between minimizing of energy consumption and satisfying Quality of Service (QoS) (e.g. performance or resource availability on time for reservation request). We consider resource needs in context of a private cloud system to provide resources for applications in teaching and researching. In which users request computing resources for laboratory classes at start times and non-interrupted duration in some hours in prior. Many previous works are based on migrating techniques to move online virtual machines (VMs) from low utilization hosts and turn these hosts off to reduce energy consumption. However, the techniques for migration of VMs could not use in our case. In this paper, a genetic algorithm for power-aware in scheduling of resource allocation (GAPA) has been proposed to solve the static virtual machine allocation problem (SVMAP). Due to limited resources (i.e. memory) for executing simulation, we created a workload that contains a sample of one-day timetable of lab hours in our university. We evaluate the GAPA and a baseline scheduling algorithm (BFD), which sorts list of virtual machines in start time (i.e. earliest start time first) and using best-fit decreasing (i.e. least increased power consumption) algorithm, for solving the same SVMAP. As a result, the GAPA algorithm obtains total energy consumption is lower than the baseline algorithm on simulated experimentation.

1 Introduction

Cloud computing [7], which is popular with pay-as-you-go utility model, is economy driven. Saving operating costs in terms of energy consumption (Watts-Hour) for a cloud system is highly motivated for any cloud providers. Energy-efficient resource management in large-scale datacenter is still challenge [1][13][9][5]. The challenge of energy-efficient scheduling algorithm is trade-off between minimizing of energy consumption and satisfying demand resource needs on time and non-preemptive. Resource requirements depend on the applications and we are interested in virtual

computing lab, which is a cloud system to provide resources for teaching and researching.

There are many studies on energy efficient in datacenters. Some studies proposed energy efficient algorithm that are based on processor speed scaling (assumption that CPU technology supports dynamic scaling frequency and voltage (DVFS)) [1][13]. Some other studies proposed energy efficient by scheduling for VMs in virtualized datacenter [9][5]. A. Beloglazov et al. [5] presents the Modified Best-Fit Decreasing (MBFD) algorithm, which is best-fit decreasing heuristic, for power-aware VM allocation and adaptive threshold-based migration algorithms to dynamic consolidation of VM resource partitions. Gori, . et al. [9] presents score-based scheduling, which is hill-climbing algorithm, to place each VM onto which physical machine has the maximum score. However, the challenge is still remain. These previous works did not concern on satisfying demand resource needs on time (i.e. VM starts at a specified start time) and non-preemptive, in addition to both MBFD and score-based algorithms do not find an optimal solution for VM allocation problem.

In this paper, we introduce our static virtual machine allocation problem (SVMAP). To solve the SVMAP, we propose the GAPA, which is a genetic algorithm to find an optimal solution for VM allocation. On simulated experimentation, the GAPA discovers a better VM allocation (means lower energy consumption) than the baseline scheduling algorithm for solving same SVMAP.

2 Problem Formulation

2.1 Terminology, notation

We describe notation that is used in this paper as following:

- VM_i : the i-th virtual machine
- M_j : the j-th physical machine
- ts_i : start time of the VM_i
- pe_i : number of processing elements (e.g. cores) of the VM_i
- PE_j : number of processing elements (e.g. cores) of the M_j
- $mips_i$: total required MIPS (Millions Instruction Per Seconds) of the VM_i
- $MIPS_j$: total capacity MIPS (Millions Instruction Per Seconds) of the M_j
- d_i : duration time of the VM_i , units in seconds
- $P_j(t)$: power consumption (Watts) of a physical machine M_j
- $r_j(t)$: set of indexes of virtual machines that is allocated on the M_j at time t

2.2 Power consumption model

In this section, we introduce factors to model the power consumption of single physical machine. Power consumption (Watts) of a physical machine is sum of total power of all components in the machine. In [8], they

estimated power consumption of a typical server (with 2x CPU, 4x memory, 1x hard disk drive, 2x PCI slots, 1x mainboard, 1x fan) in peak power (Watts) spends on main components such as CPU (38%), memory (17%), hard disk drive (6%), PCI slots (23%), mainboard (12%), fan (5%). Some papers [8] [4] [6] [5] prove that there exists a power model between power and resource utilization (e.g. CPU utilization). We assume that power consumption of a physical machine ($P(\cdot)$) is linear relationship between power and resource utilization (e.g. CPU utilization) as [8][4][6][5]. The total power consumption of a single physical server ($P(\cdot)$) is:

$$P(U_{cpu}) = P_{idle} + (P_{max} - P_{idle})U_{cpu}$$

$$U_{cpu}(t) = \sum_{c=1}^{PE_j} \sum_{i \in r_j(t)} \frac{mips_{i,c}}{MIPS_{j,c}}$$

In which:

- $U_{cpu}(t)$: CPU utilization of the physical machine at time t , $0 \leq U_{cpu}(t) \leq 1$
- P_{idle} : the power consumption (Watt) of the physical machine in idle, e.g. 0% CPU utilization
- P_{max} : the maximum power consumption (Watt) of the physical machine in full load, e.g. 100% CPU utilization
- $mips_{i,c}$: requested MIPS of the c -th processing element (PE) of the VM_i
- $MIPS_{j,c}$: Total MIPS of the c -th processing element (PE) on the physical machine M_j

The number of MIPS that a virtual machine requests can be changed by its running application. Therefore, the utilization of the machine may also change over time due to application. We link the utilization with the time t . We re-write the total power consumption of a single physical server ($P(\cdot)$) with $U_{cpu}(t)$ as:

$$P(U_{cpu}(t)) = P_{idle} + (P_{max} - P_{idle})U_{cpu}(t)$$

and total energy consumption of the physical machine (E) in period time $[t_0, t_1]$ is defined by:

$$E = \int_{t_0}^{t_1} P(U_{cpu}(t))dt$$

2.3 Static Virtual Machine Allocation Problem (SVMAP)

Given a set of n virtual machines $\{VM_i(pe_i, mips_i, ts_i, d_i) | i = 1, \dots, n\}$ to be placed on a set of m physical parallel machines $\{M_j(PE_j, MIPS_j) | j = 1, \dots, m\}$. Each virtual machine VM_i requires pe_i processing elements and total of $mips_i$ MIPS, and the VM_i will be started at time (ts_i) and finished at time $(ts_i + d_i)$ without neither preemption nor migration in its duration (d_i) . We do not limit resource type on CPU. We can extend for

Algorithm 1: GAPA Algorithm

Start: Create an initial population randomly for s chromosomes (with s is population size)
Fitness: Calculate evaluation value of each chromosome respectively in given population.
New population: Create a new population by carrying out follows the steps:
Selection: Choose the two individual parents from current population based on value of evaluation.
Crossover: By using crossover probability, we create new children via modifying chromosome of parents.
Mutation: With mutation probability, we will mutate at some position on chromosome.
Accepting: Currently, new children will be a part of the next generation.
Replace: Go to the next generation by assigning the current generation to the next generation.
Test: If stop condition is satisfied then this algorithm is stopped and returns individual has the highest evaluation value. Otherwise, go to next step.
Loop: Go back the Fitness step.

other resource types such as memory, disk space, network bandwidth, etc.

We assume that every physical machine M_j can host any virtual machine, and its power consumption model ($P_j(t)$) is proportional to resource utilization at a time t , e.g. power consumption has a linear relationship with resource utilization (e.g. CPU utilization) [8][2][5].

The objective scheduling is minimizing energy consumption in fulfillment of maximum requirements of n VMs.

2.4 The GAPA Algorithm

The GAPA, which is a kind of Genetic Algorithm (GA), solves the SVMAP. The GAPA performs steps as in the Algorithm 1.

In the GAPA, we use a tree structure to encode chromosome of an individual. This structure has three levels:

Level 1: Consist of a root node that does not have significant meaning.

Level 2: Consist of a collection of nodes that represent set of physical machines.

Level 3: Consist of a collection of nodes that represent set of virtual machines.

With above representation, each instance of tree structure will show that an allocation of a collection of virtual machines onto a collection of physical machines. The fitness function will calculate evaluation value of each chromosome as in the Algorithm 2.

Algorithm 2: Construct fitness function

```
powerOfDatacenter := 0
For each host  $\in$  collection of hosts do
    utilizationMips := host.getUtilizationOfCpu()
    powerOfHost := getPower (host, utilizationMips)
    powerOfDatacenter := powerOf Datacenter + powerOfHost
End For
Evaluation value (chromosome) := 1.0 / powerOfDatacenter
```

3 Experimental study

3.1 Scenarios

We consider on resource allocation for virtual machines (VMs) in private cloud that belongs to a college or university. In a university, a private cloud is built to provide computing resource for needs in teaching and researching. In the cloud, we deploy installing software and operating system (e.g. Windows, Linux, etc.) for practicing lab hours in virtual machine images (i.e. disk images) and the virtual machine images are stored in some file servers. A user can start, stop and access VM to run their tasks. We consider two needs as following:

- i A student can start a VM to do his homework.
- ii A lecturer can request a schedule to start a group of identical VMs for his/her students on lab hours at specified start time and in prior. The lab hours requires that the group of VMs will start on time and continue in spanning some time slots (e.g. 90 minutes).
- iii A researcher can start a group of identical VMs to run his/her parallel application.

3.2 Workload and simulated cluster

We use workload from one-day of our university's schedule for laboratory hours on six classes in the Table 1. The workload is simulated by total of 211 VMs and 100 physical machines (hosts).

We consider there are two kind of servers in our simulated virtualized datacenter, which includes two power consumption models of two power model of the IBM server x3250 (1 x [Xeon X3470 2933 MHz, 4 cores], 8GB) and another power model of the Dell Inc. PowerEdge R620 (1 x [Intel Xeon E5-2660 2.2 GHz, 16 cores], 24 GB) server with 16 cores in the Table 2. The baseline scheduling algorithm (BFD), which sorts list of virtual machines in start time (i.e. earliest start time first) and using best-fit decreasing (i.e. least increased power consumption, for example MBFD [5]), will use four IBM servers to allocate for 16 VMs (each VM requests single processing element). Our GAPPA can finds a better VM

allocation (lesser energy consumption) than the minimum increase of power consumption (best-fit decrease) heuristic in our experiments. In this example, our GAPA will choose one Dell server to allocate these 16 VMs. As a result, our GAPA consumes less total energy than the BFD does.

Table 1. Workload of a university’s one-day schedule

Day	Subject	Class ID	Group ID	Students	Lab. Time	Duration (sec.)
6	506007	CT10QUEE	QT01	5	456	8100
6	501129	CT11QUEE	QT01	5	123	8100
6	501133	DUTHINH6	DT04	35	123	8100
6	501133	DUTHINH5	DT01	45	456	8100
6	501133	DUTHINH5	DT02	45	456	8100
6	501133	DUTHINH6	DT05	35	123	8100
6	501133	DUTHINH6	DT06	41	123	8100

3.3 Experiments

Table 2. Two power models of (i) the IBM server x3250 (1 x [Xeon X3470 2933 MHz, 4 cores], 8GB) [16] and (ii) the Dell Inc. PowerEdge R620 (1 x [Intel Xeon E5-2660 2.2 GHz, 16 cores], 24 GB) [15]

Utilization	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
IBM x3250	41.6	46.7	52.3	57.9	65.4	73.0	80.7	89.5	99.6	105.0	113.0
Dell R620	56.1	79.3	89.6	102.0	121.0	132.0	149.0	171.0	195.0	225.0	263.0

We show results from the experiments in the Table 3 and Figure 1. We use a popular simulated software for a virtualized datacenter is the CloudSim [14][6] to simulate our virtualized datacenter and the workload. The GAPA is a VM allocation algorithm that is developed and integrated into the CloudSim version 3.0.

On simulated experimentation, we have total energy consumptions of both the BFD and the GAPA algorithms are 16.858KWh and average of 13.007KWh respectively. We conclude that the energy consumption of the BFD algorithm is higher than the energy consumption of GAPA algorithm is approximately 130%. In case of the GAPA, these GAPA use the probability mutation is 0.01 and size of population is 10, number of generations is {500, 1000}, probability of crossover is {0.25, 0.5, 0.75}.

4 Related works

B. Sotomayor et al. [12] proposed a lease-based model and First-Come-First-Serve (FCFS) and backfilling algorithms to schedule best-effort,

Table 3. Total energy consumption (KWh) of running: (i) earliest start time first with best-fit decreasing (BFD); (ii) GAPA algorithms. These GAPA use the probability mutation of 0.01 and size of population of 10. N/A means not available

Algorithms	VMs	Hosts	GA's Generations	GA's Prob. of Crossover	Energy (KWh)	BFD/GAPA
BFD	211	100	N/A	N/A	16.858	1
GAPA_P10-G500_C25	211	100	500	0.25	13.007	1.296
GAPA_P10-G500_C50	211	100	500	0.50	13.007	1.296
GAPA_P10-G500_C75	211	100	500	0.75	13.007	1.296
GAPA_P10-G1000_C25	211	100	1000	0.25	13.007	1.296
GAPA_P10-G1000_C50	211	100	1000	0.50	13.007	1.296
GAPA_P10-G1000_C75	211	100	1000	0.75	13.007	1.296

immediate and advanced reservation jobs. The FCFS and backfilling algorithms consider only performance metric (e.g. waiting time, slow-down). To maximize performance, these scheduling algorithms tend to choose free load servers (i.e. highest-ranking scores) when allocates a new lease. Therefore, a lease with single VM can be allocated on big, multi-core physical machine. This way could be waste energy, both of the FCFS and backfilling does not consider on the energy efficiency.

S. Albers et al. [1] reviewed some energy efficient algorithms which are used to minimize flow time by changing processor speed adapt to job size. G. Laszewski et al. [13] proposed scheduling heuristics and to present application experience for reducing power consumption of parallel tasks in a cluster with the Dynamic Voltage Frequency Scaling (DVFS) technique. We did not use the DVFS technique to reduce energy consumption on datacenter.

Some studies [9][3][5] proposed algorithms to solve the virtual machine allocation in private cloud to minimize energy consumption. A. Beloglazov et al. [3][5] presented a best-fit decreasing heuristic on VM allocation, named MBFD, and VM migration policies under adaptive thresholds. The MBFD tends to allocate a VM to such as active physical machine that would take the minimum increase of power consumption (i.e. the MBFD prefers a physical machine with minimum power increasing). However, the MBFD cannot find an optimal allocation for all VMs. In our simulation, for example, the GAPA can find a better VM allocation (lesser energy consumption) than the minimum increase of power consumption (best-fit decrease) heuristic in our experiments. In this example, our GAPA will choose one Dell server to allocate these 16 VMs. As a result, our GAPA consumes less total energy than the best-fit heuristic does.

Another study on allocation of VMs [9] developed a score-based allocation method to calculate scores matrix of allocations of m VMs to n physical machines. A score is sum of many factors such as power

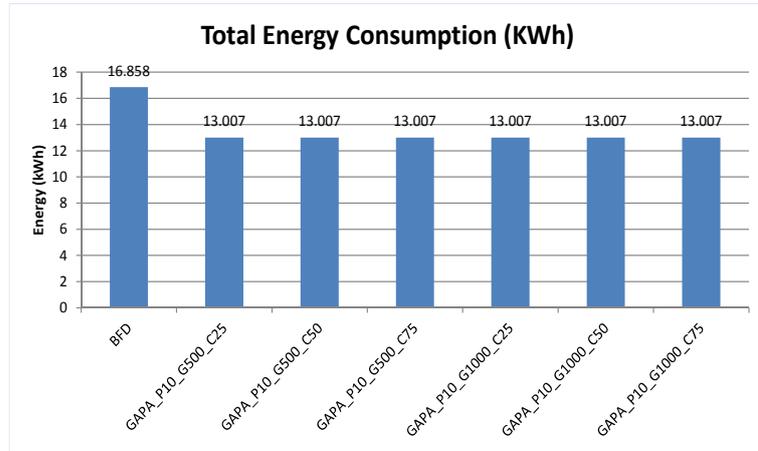


Fig. 1. The total energy consumption (KWh) for earliest start time first with best-fit decrease (BFD), GAPA algorithms

consumption, hardware and software fulfillment, resource requirement. These studies are only suitable for service allocation, in which each VM will execute a long running, persistent application. We consider each user job has a limited duration time. In addition to, our GAPA can find an optimal schedule for the static VM allocation problem on single objective is minimum energy consumption.

In a recently work, J. Kolodziej et al. [10] presents evolutionary algorithms for energy management. None of these solutions solves same our SVMAP problem.

5 Conclusions and Future works

In a conclusion, a genetic algorithm can apply to the static virtual machine allocation problem (SVMAP) and brings benefit in minimize total energy consumption of computing servers. On simulation with workload of one-day lab hours in university, the energy consumption of the baseline scheduling algorithm (BFD) algorithm is higher than the energy consumption of GAPA algorithm is approximately 130%. Disadvantage of the GAPA algorithm is longer computational time than the baseline scheduling algorithm.

In the future work, we concern methodology to reduce computational time of the GAPA. We also concern some other constraints, e.g. deadline of jobs. We also study on migration policies and history-based allocation algorithms.

References

1. Albers, S. and Fujiwara, H.: Energy-efficient algorithms. ACM Review, Vol. 53, No. 5, (2010) pp. 86–96, doi: 10.1145/1735223.1735245

2. Barroso, L.A. and Hözl, U.: The Case for Energy-Proportional Computing, Vol. 40, pp. 33-37. ACM (2007), doi: 10.1109/MC.2007.443
3. Beloglazov, A. and Buyya, R.: Energy Efficient Resource Management in Virtualized Cloud Data Centers, Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 826-831. (2010) doi: 10.1109/CCGRID.2010.46
4. Beloglazov, A. and Buyya, R.: Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of VMs in Cloud Data Centers, ACM (2010)
5. Beloglazov, Abawajy, A., J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing, FGCS, Vol. 28, no. 5, pp. 755-768, (2012). DOI: 10.1016/j.future.2011.04.017
6. Beloglazov, A. and Buyya, R.: Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers, CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE, Concurrency Computat.: Pract. Exper., pp. 1-24, (2011), doi: 10.1002/cpe
7. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, FGCS, Vol. 25, No. 6 , pp. 599-616. (2009) doi: 10.1016/j.future.2008.12.001
8. Fan, X., Weber, W.-D., Barroso, L. A.: Power provisioning for a warehouse-sized computer, Proceedings of the 34th annual international symposium on Computer architecture, pp. 13-23. ACM (2007) doi: 10.1145/1273440.1250665
9. Goiri, ., Juli, F., Nou, R., Berral, J., Guitart, J., Torres, J.: Energy-aware Scheduling in Virtualized Datacenters, in IEEE International Conference on Cluster Computing (CLUSTER 2010), (2010), pp. 58-67.
10. J. Kolodziej et al. (Eds.): A Taxonomy of Evolutionary Inspired Solutions for Energy Management in Green Computing : Problems and Resolution Methods, Advances in IntelligentModelling and Simulation, SCI 422, pp. 215–233.
11. Sotomayor, B., Keahey, K., Foster, I.: Combining batch execution and leasing using virtual machines, Proceedings of the 17th international symposium on High performance distributed computing - HPDC '08, pp. 87-96. ACM (2008) doi: 10.1145/1383422.1383434
12. Sotomayor, B.: Provisioning Computational Resources Using Virtual Machines and Leases, PhD Thesis submitted to The University of Chicago, US, (2010)
13. Laszewski, G. V., Wang, L., Younge, A. J., He, X.: Power-aware scheduling of virtual machines in DVFS-enabled clusters, 2009 IEEE International Conference on Cluster Computing and Workshops, pp. 368–377. (2009) doi: 10.1109/CLUSTR.2009.5289182
14. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.

15. SPECpower ssj2008 results for Dell Inc. PowerEdge R620 (Intel Xeon E5-2660, 2.2 GHz). http://www.spec.org/power_ssj2008/results/res2012q2/power_ssj2008-20120417-00451.html. Last accessed: Nov. 29, 2012
16. SPECpower ssj2008 results for IBM x3250 (1 x [Xeon X3470 2933 MHz, 4 cores], 8GB). http://www.spec.org/power_ssj2008/results/res2009q4/power_ssj2008-20091104-00213.html. Last accessed: Nov. 29, 2012