

Green Cloud Computing: Towards Optimizing Data Centre Resource Allocation

Akshat Dhingra¹, Sanchita Paul²

M.Tech Scholar, Department of Computer Science and Engineering, Birla Institute of Technology, Ranchi, India¹

Assistant Professor, Department of Computer Science and Engineering, Birla Institute of Technology, Ranchi, India²

Abstract- Cloud computing aims to offer utility based IT services to the users irrespective of their location on the basis of a pay-as-you-go model. With the increasing popularity of cloud computing in the present times, there has been a phenomenal increase in the power consumption as the data centres that host the Cloud applications consume huge amounts of energy. Therefore there is an utmost need to develop solutions that aim to save energy consumption without compromising much on the performance. Such solutions would also help reducing the costs thereby benefitting the cloud service providers. In this paper, we aim to reduce the power consumption of the data centre by continuously optimizing the resource allocation according to the current resource utilisation and for that an optimization technique called Bacterial Foraging has been used thereby improving the energy efficiency of the data centre. The results make it clearly evident that cloud computing has great potential and offers significant performance gains as well as cost savings even under dynamic workload conditions.

Keyword- Energy Efficient Data Centres, Bacterial Foraging, Resource Allocation Optimization, Power Consumption Minimization, Chemotaxis, Elimination, Modified Best Fit Decreasing.

1. INTRODUCTION

The designers have always primarily focused on improving the performance of computing systems and hence the performance has been steadily growing driven by more efficient system design and increasing density of the components described by Moore's law. Although the performance per watt ratio has been constantly rising, the total power draw by computing systems is hardly decreasing. On the contrary, it has been increasing every year that can be illustrated by the estimated average power usage across three classes of servers shown in the Table 1 in Watts/Unit.

Table 1: Power Consumption among three classes of servers

Class of Server	2000	2001	2002	2003	2004	2005	2006
Low-end	186	193	200	207	213	219	225
Mid-Range	424	457	491	524	574	625	675
High-end	5534	5832	6130	6428	6973	7651	8163

Apart from the overwhelming operating costs due to high energy consumption, another rising concern is the environmental impact in terms of carbon dioxide (CO₂) emissions caused by this high energy consumption. In 2007, the total carbon footprint of the IT industry including personal

computers, mobile phones, and telecom devices and data centres was 830 MtCO_{2e}, 2% of the estimated total emissions and this figure is expected to grow in the coming years [1]. Therefore, the reduction of power and energy consumption has become a first-order objective in the design of modern computing systems.

There are two possible solutions to make IT Systems greener: 1) improve efficiency or 2) find a plentiful supply of clean, affordable energy. As the later is still in the realms of science fiction, energy efficiency is where the main focus of research will be in the near future. IT companies are learning that cutting emissions and cutting costs naturally go together, by making systems energy efficient money may be saved automatically. Energy-Aware Computing research is attempting to address this problem. Work in this field is tackling issues ranging from reducing the amount of energy required by a single processor chip to finding the most effective means of cooling a warehouse sized data centre. Cloud Computing has the potential to have a massive impact, positive or negative, on the future carbon footprint of the IT sector. On the one hand, Cloud Computing data centres are now consuming 0.5% of all the generated electricity in the world, a figure that will continue to grow as Cloud Computing becomes widespread particularly as these systems are "always-on always-available". However, the large data centres required by clouds have the potential to provide the most efficient environments for computing. The growing popularity of cloud computing would therefore drive the cloud providers to build efficient systems in order to reduce the total cost of ownership (TOC) and hence improve their green credentials. The main aim of Energy-Aware Computing is to promote awareness of energy consumption at both software and hardware levels and hence consumes lesser amount of power.

Dhingra et. al [5] gives a brief review of the various power management schemes at the data centre level with the help of virtualization. This paper begins with a brief introduction of cloud computing, its potential and the need to make cloud computing energy aware in Section 1. Section 2, gives an introduction to a power consumption model [2] that helps predict the power consumption. In Section 3, an architectural framework of an energy aware cloud is presented and briefly explained. Section 4 presents an energy efficient resource allocation algorithm with an extensive case study to realise the benefits of the proposed solution. Section 5 concludes this paper with limitations and future directions.

2. POWERCONSUMPTION MODEL

The Power consumption in Complementary Metal-Oxide-Semiconductor (CMOS) circuits comprises of static and dynamic power consumption. The static power consumption is caused by leakage currents that are present in any active circuit, independently of clock rates and usage scenarios. This static power is mainly determined by the type of transistors and process technology and its reduction therefore requires improvement in low level system designs. This would hence not be the focus of this paper. To develop new policies for Dynamic Power Management and understand their impact, it is necessary to create a model of dynamic power consumption. Such a model has to be able to predict the actual value of the power consumption based on some run-time system characteristics. One of the ways to accomplish this is to utilize power monitoring capabilities that are built-in modern computer servers. This instrument provides the ability to monitor power usage of a server in real time and collect accurate statistics about the power usage. Based on this data it is possible to derive a power consumption model for a particular system. However, this approach is complex and requires collection of the statistical data for each target system. Fan et al. [3] have found strong relationship between the CPU utilization and total power consumption by a server. Hence, the idea behind the proposed model is that the power consumption by a server grows linearly with the growth of CPU utilization from the value of power consumption in the idle state up to the power consumed when the server is fully utilized. This relationship can be expressed as:

$$P(u) = P_{idle} + (P_{busy} - P_{idle}) * u \quad (1)$$

Where $P(u)$ is the estimated power consumption at a particular instant of time; P_{busy} is the power consumed when the server is fully utilized; P_{idle} is the power consumed by the idle server; and u is the CPU utilization. The CPU utilization may change over time due to variability of the workload.

3. Framework

This paper aims to optimize the VM allocation in order to reduce the energy consumption. There are four major entities in the entire framework which are illustrated in the figure below.

The components in the energy optimization layer as shown in the below are:

- **Service Analyser:** Analyses the requirements of each submitted request and takes a decision whether to accept or reject the request based on the information gathered from the VM Manager, Energy Monitor and Resource utilization.
- **Energy Monitor:** Determines the energy consumed by each of the VMs.
- **Resource Utilization:** Determines the resources consumed by each of the Virtual Machines and hence has current information on the total resource consumption and availability.
- **Optimization Function:** This component tries to minimize the energy consumption without compromising much on the performance.
- **Migration Controller:** Performs the migration operation as and when commanded by the Optimization function and requires inputs from the VM manager.

- **ON/OFF Control:** Determines what VMs to turn ON or OFF as per the requirement
- **VM Manager:** Keeps a track of each VM including what physical machine it resides on and what are its resource requirements.

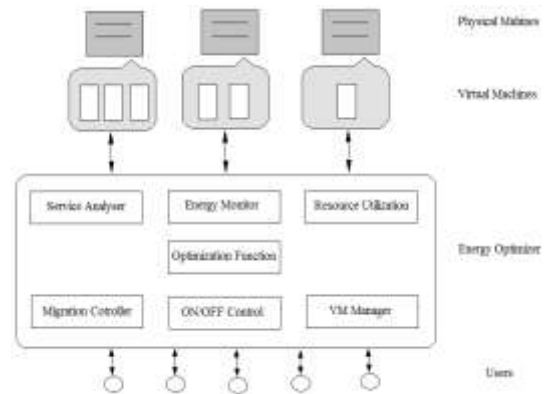


Figure 1: Framework for Energy Efficient Data Centre

4. OPTIMIZING DATA CENTRE RESOURCE ALLOCATION

The problem of optimizing the VM allocation could be divided into two steps. In the first step, VMs are placed on the hosts and in the second step the optimization of the current allocations of the VMs take place. In the first step, we allocate the VMs to these physical hosts according to Modified Best Fit Decreasing (MBFD) Algorithm proposed by Buyya et al. [2]. In this algorithm, the VMs are first sorted in a decreasing order of their CPU requirements and then each VM is allocated to a physical machine that provides least increase in the power consumption due to this allocation. The steps involved in Modified Best Fit Decreasing Algorithm are given below for better understanding.

- Initialize all the physical machines.
- Initialize all the VMs.
- Sort VMs in decreasing order of their CPU requirements.
- Allocate the VM to the physical machine that provides least increase in the power consumption due to this allocation.

After the initial allocation has been done, in the second step, the Bacterial Foraging Algorithm is applied to a set of solutions which are generated based on the current resource utilisation to get the most optimum solution (VM Allocation) that consumes minimum energy and offers least compromise in the performance. Bacterial Foraging Optimization Algorithm (BFOA) [4] is a new comer to the family of nature-inspired optimization algorithms. During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an E.coli bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place it tumbles frequently to find a nutrient gradient. Moving

the flagella in the counterclockwise direction helps the bacterium to swim at a very fast rate. In the above-mentioned algorithm the bacteria undergoes chemotaxis, where they like to move towards a nutrient gradient and avoid noxious environment. The four prime steps in BFOA are:

- i. **Chemotaxis:** This process simulates the movement of an E.coli cell through swimming and tumbling via flagella. Biologically an E.coli bacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose $\Theta^i(j, k, l)$ represents i-th bacterium at jth chemotactic, k-th reproductive and l-th elimination-dispersal step. $C(i)$ is the size of the step taken in the random direction specified by the tumble (run length unit). Then the mathematically, the movement of the bacterium may be represented by $\Theta^i(j+1, k, l) = \Theta^i(j, k, l) + C(i)\Delta(i) / \sqrt{(\Delta^T(i)\Delta(i))}$ where Δ indicates a vector in the random direction whose elements lie in $[-1, 1]$.
- ii. **Swarming:** Interesting group behaviour has been observed for several motile species of bacteria including E.coli and S. typhimurium, where intricate and stable spatial-temporal patterns (swarms) are formed in semisolid nutrient medium. A group of E.coli cells arrange themselves in a travelling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effector. The cells when stimulated by a high level of succinate, release an attractant aspartate, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. The cell-to-cell signalling in E. coli swarm may be represented by the following function $J_{cc}(\Theta, P(j, k, l)) = \sum_{i=1}^S J_{cc}(\Theta, \Theta^i(j, k, l)) = \sum_{i=1}^S [-d_{attractant} \exp(-w_{attractant} \sum_{m=1}^p (\Theta_m - \Theta_m^i)^2)] + \sum_{i=1}^S [-h_{repellant} \exp(-w_{repellant} \sum_{m=1}^p (\Theta_m - \Theta_m^i)^2)]$ (3) Here $J_{cc}(\Theta, P(j, k, l))$ is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function, S is the total number of bacteria, p is the number of variables to be optimized, which are present in each bacterium. $d_{attractant}$, $w_{attractant}$, $h_{repellant}$, $w_{repellant}$ are different coefficients that should be chosen properly.
- iii. **Reproduction:** The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.
- iv. **Elimination and Dispersal:** Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very

small probability while the new replacements are randomly initialized over the search space.

The optimization algorithm which is the second step that optimizes the current allocation is explained below.

4.1 Bacterial Foraging Optimization of the Current Resource Allocation

Input: Number of VMs and physical machines, minimum and maximum resource requirements of each VM.

Output: Best Solution.

Population=: All feasible solutions;

For (l=0 to N_{ed}) //Elimination-Dispersal loop.

For (k=0 to N_{re}) //Reproduction loop

For (j=0 to N_c) // Chemotaxis loop

ChemotaxisandSwim(Population, N_s); //Searches for the best solution.

For (Each Cell in Population)

If(PowerConsumed(Cell) <= PowerConsumed(Cell_{best}))

Cell_{Best} = Cell; //Solution that consumes //minimum amount of power.

End

End

End

SortByHealth(Population); //Sorting done in increasing order of power //consumption

Selected=: SelectedByHealth(Population); //Selection of the best solution.

Population=: Selected;

End

End

ChemotaxisandSwim (Population)

Begin

For (Each cell in Population)

{Determine the current resource usage and the power consumed by the cell; PowerConsumed (Cell);

1. Determine the random feasible VM to be migrated also considering the upper and lower utilisation thresholds;

2. Determine the power consumed if the VM is migrated (NewPower);

3. If (NewPower < PowerConsumed(Cell))

Make necessary modifications in the cell and store; }

End

Here, the parameters N_c , N_{re} , and N_{ed} represent the number of Chemotaxis, Reproduction and Elimination steps respectively.

The values of N_c , N_{re} , N_{ed} depends on the level of optimization that is desired and may vary in different scenarios. Hence, it would be interesting to analyse the result with different values of N_c . A 'Cell' represents each possible solution and 'Cell_{Best}' is the best solution after each chemotaxis step. 'PowerConsumed' is a function that determines the power consumed by each Cell. ' N_s ' represents the total number of swim steps.

The flowchart for resource allocation optimization using bacterial foraging is shown below.

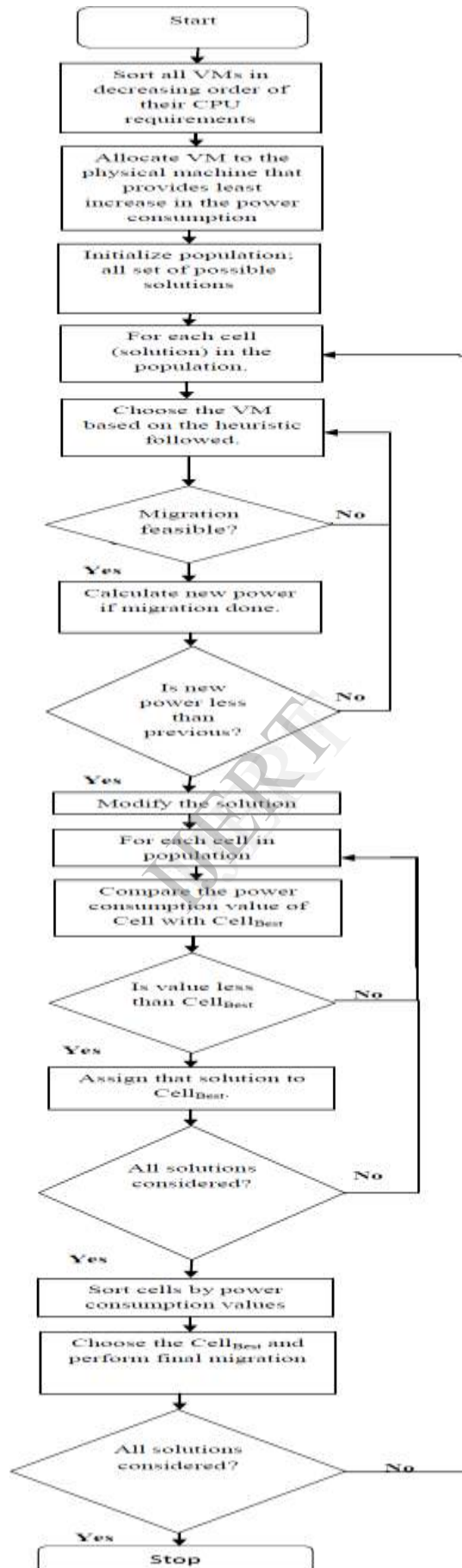


Figure 2: Flowchart showing the optimization process

Case Study

For better understandability and to check the correctness of the algorithm, a dry run of the algorithm is presented and the outputs are compared to an otherwise Non-Power Aware Policy wherein the allocations are static.

We assume three physical machines whose details are summarised below in the table.

Table II: Physical Machines

Physical machine	CPU	RAM	Hard Disk	P_{idle} (in Watts)	P_{busy} (in Watts)
P1	1 GHz	2 GB	1 TB	175	250
P2	2 GHz	4 GB	2 TB	210	300
P3	3 GHz	6 GB	3 TB	245	350

We also assume three VMs whose requirements are summarised below in the table.

Table III: Virtual Machines

Virtual Machine	CPU	RAM	Hard Disk
V1	500 MHz	1 GB	250 MB
V2	1 GHz	2 GB	125 MB
V3	700 MHz	1 GB	512 MB
V4	250 MHz	1 GB	100 MB
V5	200 MHz	2 GB	100 MB

The VM allocation would proceed in the following manner:

1. Allocation of VMs according to Modified Best Fit Decreasing Algorithm [2].
- 1.1. The VMs are first sorted in a decreasing order of their CPU requirements shown in the table above. Hence we get the following order.

1. V2
2. V3
3. V1
4. V4
5. V5

- 1.2. Now, each VM is allocated to a physical machine that provides least increase in the power consumption due to this allocation. Hence, we get the following allocation of the VMs to the physical machines (refer to the tables above for specifications).

Virtual Machine	Physical Machine
V1	P3
V2	P3
V3	P3
V4	P3
V5	P2

Hence from the above allocation we notice that only two physical machines are required to fulfil the resource demands. So, the machine P1 could be conveniently turned-off thereby leading to power savings. Therefore, the power consumption

under this allocation calculated with the help of Equation (1) is 549.75 W.

2. Application of Bacterial Foraging Optimization Technique to optimize the current VM allocation. This algorithm would also consider the upper and lower utilisation thresholds (60 % and 10% respectively) and try to keep the total utilisation of all the physical machine resources between these thresholds. The aim is to preserve free resources to prevent SLA violation due to consolidation in cases when utilisation by VMs increases. Also, in the optimisation algorithm, we set the number of Chemotaxis, Reproduction and the Elimination steps (N_c , N_{re} , N_{ed}) to be 1.

- 2.1 At a particular instance we assume that the VMs are using only 50% of the allocated resources. The algorithm then tries to calculate the free resources on the machines and tries to migrate the VM that causes reduction in the power consumption and does not violate the upper threshold value of resource utilisation after migration. Therefore the total resource consumption on P3 is :

CPU: 1225 MHz

RAM: 2.5 GB

Hard Disk: 493.5 MB

- 2.2 We notice that if V5 is migrated to P3, the threshold value of 60% will still not be crossed.

- 2.3 Hence, the best solution after a set of calculations done during the Chemotaxis and Swimstep with the help of the Equation (1) is given below.

Virtual Machine	Physical Machine
V1	P3
V2	P3
V3	P3
V4	P3
V5	P3

In the above allocation, we notice that the current requirements could be conveniently fulfilled with the help of only one physical machine P3. Hence, all the other machines could be switched off. The total power consumption under this allocation is 268.19 Watts, which is much lower than the one in previous allocation.

5. LIMITATIONS AND FUTURE DIRECTIONS

A limitation in the proposed solution is that the requirements should be known in advance for initial allocation to take place. This information may not be known in certain scenarios. Also, the cost of migration has not been considered as it is assumed that all the machines are located in the same location. This however may not always be the case. In future we would like to simulate a cloud computing environment and then implement our algorithm. It would also be interesting to analyse the results if the algorithm is implemented in a real scenario. As a part of the future work we would also suggest exploring some hybrid optimization techniques that harvest the benefits of Bacterial Foraging with Genetic Algorithms or Greedy Knapsack.

REFERENCES

1. E. Farnworth and J.C. Castilla-rubio, "SMART 2020: Enabling the low carbon economy in the information age," Group.
2. R. Buyya, A. Beloglazov, and J. Abawajy, —Energy-Efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges, I in Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), Las Vegas, USA, July 12-15, 2010.
3. X. Fan, W. D. Weber, and L. A. Barroso, —Power provisioning for a warehouse-sized computer, I in Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA 2007). ACM New York, NY, USA, 2007, pp. 13–23.
4. Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham, Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications.
5. Akshat Dhingra, Sanchita Paul, "A Survey of Energy Efficient Data Centres in a Cloud Computing Environment", in International Journal of Advanced Research in Computer and Communication Engineering, Volume 2, Issue 10, October 2013, pp. 4033-4041.

IJERT