A SWARM-INSPIRED TECHNIQUE FOR SELF-ORGANIZING AND CONSOLIDATING DATA CENTRE SERVERS

IONUT ANGHEL, CRISTINA BIANCA POP, TUDOR CIOARA, IOAN SALOMIE, IULIA VARTIC *

Abstract. This paper proposes a swarm-inspired data centre self-organizing and consolidation technique which aims at reducing the power demand in data centres while ensuring the workload execution within the established performance parameters. Each data centre server is managed by an intelligent agent that implements a bird's migration-inspired behaviour to decide on the appropriate server consolidation actions. The selected actions are executed to achieve an optimal utilization of server computing resources thus lowering power demand. The data centre servers self-organize in logical clusters according to the birds V-formation self-organizing migration model. The results are promising showing that by using the proposed swarm-inspired solution, the data centre Deployed Hardware Utilization Ratio Green Indicator increases compared to the widely used Fit First consolidation algorithm. The average power saving of the proposed technique is around 40% of the power demanded by the data centre computing resources and about 16% of its total power demand including the IT facility, when comparing to OpenNebula Fit First consolidation technique. This paper is an extended version of the one published in WIMS'12 proceedings showing more details about the swarm-inspired consolidation technique and the defined algorithms.

Key words: Swarm optimization, self-organizing, data centre, server consolidation, birds V-formation.

AMS subject classifications. 68T42, 68T05, 37N25

1. Introduction. Green computing is receiving more and more interest from both data centre owners and research groups. Green IT is defined as designing, manufacturing, and using computer systems in an energy efficient manner with low or no environment impact [1]. Initiatives such as Climate Savers Computing Initiative [28] or the Green Grid [38] aim at developing methods and techniques for reducing the energy consumption in computing environments. Studies have shown that exponential growth trends of the data centres are no longer sustainable without the careful consideration of their energy efficiency [29] [31]. An important rising concern is the environmental impact in terms of carbon dioxide (CO_2) emissions caused by high energy consumption. From this point of view, CO_2 emissions of the worldwide data centres are equivalent to about half of the total airlines' CO_2 emissions and are expected to grow from 76 $MTCO_2e$ to 259 $MTCO_2e$ by 2020 [32]. Studies show that the cost of the energy required by a data centre server along its lifetime will probably exceed its hardware costs [18]. Data centre electricity consumption accounts for almost the 2% of the world production and their overall carbon emissions are greater than both Argentina and Netherlands [25]. Moreover, infrastructure systems such as cooling systems or power delivery systems supporting the IT equipment itself, consume 60% of a data centre's energy usage. All these problems led to considering energy or power efficiency as first-order objective in the design of modern computing infrastructures including data centres and clouds.

In this context, many researches from both management and technical background, are working to reduce the energy consumption of IT data centres. In general, data centre administrators are always focusing on the performance aspect when configuring the data centre servers or clusters and ignore the energy consumption. The sub-optimal utilization of computing resources is one of the main factors reported in the literature that contributes to the data centre high energy consumption [25]. To solve this problem, state of the art solutions use techniques based on virtualization and server consolidation for minimizing the number of underutilized servers in data centres [2].

The problem of organizing and using the data centre IT hardware resources in an energy/power efficient manner can be approached as an optimization problem. Nature can serve as an inspiration for solving different domain problems. Biological systems feature organized, complex and intelligent behaviour. Applying biology concepts in the field of computer science is a new evolving research direction. The main idea is to model and develop computer systems by mapping biological principles and concepts to the actual problem that needs to be solved by computer science. By inspiring from insects, birds or animals behaviour, efficient and self-managing distributed systems can be designed and modelled. Thus, the problem of organizing and using the data centre

---

*Computer Science Department, Technical University of Cluj-Napoca, Romania ({`Ionut.Anghel, Cristina.Pop, Tudor.Cioara, Ioan.Salomie`}@cs.utcluj.ro).

IT hardware resources in an energy efficient manner can be approached as an optimization problem and can be solved by applying bio-inspired concepts and heuristics. Birds and insects are both employing swarm level self-organizing behavioural strategies, which help them to live and survive by efficiently using and conserving energy. For example, penguins self-organize in swarms in case of cold temperatures and high winds by huddling themselves in concentric circular formations [22]. In these formations, the penguins which are not on the exterior side are kept warm, leading to energy conservation. The exterior penguins are the ones that face the harshest conditions and ensure the comfort of the other members of the colony, but they are continuously replaced by other penguins coming from the interior part of the formation. The swarm of birds' flying in V-formations during their migration is another self-organizing behavioural strategy for conserving energy [23]. In such formations, the leading birds (situated in the head of the V) together with the ones situated in the tips of the V-formation wings, consume a lot of energy while the other birds consume less energy [24]. In this context, by inspiring from the swarms' self-organizing behavioural strategies which have ensured the survivability of different biological species over time, energy efficient self-organizing strategies for data centres servers can be developed.

This paper proposes a server consolidation technique for reducing power consumption in data centres inspired from the birds self-organizing behaviour during migration. Each data centre server has an attached intelligent agent which implements behaviour similar with the behaviour of a bird in the V-formation. The V-formation leading birds are modelled by a cluster of active data centre servers that can accommodate the incoming workload. The birds from the start of the V-formation wings are represented by a cluster of data centre fully loaded active servers that are candidates for powering down after finishing the execution of their deployed workload. The birds in the middle of the V-formation wing are modelled by a cluster of servers containing powered down servers, while the following birds in the V-formation wing are represented by a cluster of idle servers that are candidates for passing to active states and for accommodating the incoming workload. The servers attached intelligent agents communicate and collaborate to decide on the following types of actions: resource consolidation actions (workload deployment and migration), dynamic power management actions (turn on/off server) and server placement in the appropriate V-formation cluster. This paper is an extended version of the paper [30] and shows more details about the swarm-inspired consolidation technique and the defined algorithms.

The rest of the paper is organized as follows: Section 2 presents related work, Section 3 introduces the swarm-inspired consolidation technique, Section 4 describes a case study and relevant evaluation results while Section 5 concludes the paper and shows future work.

2. Related Work. Organizing and using the data centre IT hardware resources in an energy/power efficient manner by applying virtualization and server consolidation techniques is a NP computational optimization problem which can only be solved by using a holistic approach.

The current approaches to resource consolidation take advantage of virtualization by proposing models to migrate the virtual machines in data centres from one server/cluster to another [20]. By virtual machine migration, the workload can be consolidated on a smaller number of physical machines allowing for servers, or even for entire operation nodes, to be completely shut down [27]. Authors reveal that when consolidation is used, an optimal solution for energy/performance trade-off can be defined. Efficient consolidation models based on the bin packing technique were proposed in [10]. Two well-known heuristics for the bin packing based consolidation, the best-fit decreasing (BFD) and the first-fit decreasing (FFD), were used [19]. To enable energy efficient consolidation, the inter-relationships between energy consumption, resource utilization, and performance of consolidated workloads must be considered [9].

In [15] self-configuring and self-optimizing agents handle user requests to dynamically allocate resources for cloud applications. Support Vector Machines and Model-Predictive Control are employed to predict and plan future virtual machines configurations while guaranteeing SLAs. The agents share a common knowledge base where adaptation rules are stored. A thermal aware workload scheduling and consolidation solution aiming to reduce the power consumption and temperatures in data centres was proposed in [12]. The simulation results show that the algorithm can significantly reduce the energy consumption with some degree of performance loss. In [25] a novel technique for controlling the data centres servers CPU allocation and consolidation based on first order Kalman filter is presented. In [14] the server consolidation problem is approached for small data centres as a constraint satisfaction problem. The authors also propose a heuristic for dealing with server consolidation

in large data centres.

Learning techniques are also used to trade-off between computing resources power consumption and performance during the allocation process [16]. In [17] the problem of dynamic server consolidation in virtualized data centres is approached by proposing the development of an energy aware run-time consolidation algorithm based on reinforcement learning. In [11] a consolidation methodology that uses machine learning to deal with uncertain information is proposed. Previous server behaviour data is used to predict and estimate the current power consumption and also to improve the scheduling and consolidation decisions.

Few state of the art approaches study and use biologically-inspired techniques for optimizing the energy/power consumption in IT systems and data centres. Since biological systems naturally tend to conserve their energy, many simple principles found in the biological systems might be used in IT power management [4]. The adoption of biological principles (e.g. decentralization, natural selection, symbiosis) in the process of designing and building services on top of server farms is proposed in [6]. A service is designed as a biological entity, equivalent to an individual bee in a bee colony that competes or collaborates for computing resources. Using natural selection principles, the services that waste energy are banned for execution. An immune-inspired method for designing applications capable of adapting to network dynamic changes is proposed in [8]. The environmental conditions are modelled as antigens, while the agents (used to design applications) are modelled as biological entities having an immune system.

In [5] the management of energy consumption in wireless sensors networks is approached using a biologically-inspired agent based solution. The agent behaviour focuses on biologically-inspired actions (e.g. pheromone emission or migration), each of them having an associated energetic cost. Through these actions, the life time and the state of each agent evolve autonomously and there is no need of a centralized control unit. Inspired by the behaviour of insects in search for a proper migration place, the authors of [7] propose a method for optimizing the energy consumption in data centres. The migrating insect is modelled by a running virtual machine, a colony of insects is modelled as a set of virtual machines running on the same server, and the candidate migration places are modelled by servers. Authors use a scout-worker migration model in which a set of agents (scouts) investigate each server to identify the appropriate one where a virtual machine can migrate. In [3], authors formulate the problem of server consolidation as an optimization problem which aims to maximize the number of servers hosting zero virtual machines, while ensuring that all available virtual machines are run. To solve this optimization problem in a decentralized and self-organizing way, authors propose a gossip-based algorithm in which each server interacts with its neighbours by means of messages containing the number of virtual machines running on it. The server that has space for extra virtual machines accepts to host them.

3. The Swarm-inspired Consolidation Technique. This section introduces the server consolidation technique inspired from bird's behaviour during migration. The goal of the technique is to reduce the data centre power consumption while ensuring the workload execution within the pre-established parameters.

3.1. Biology Inspiration. During migration, the birds organize in a V-formation which consists of a leading bird followed by two lines of birds (the V-formation wings), flying closely behind each other (see Figure 3.1).

In this formation, the birds constantly shift positions and the leading birds periodically leave their position and re-join the line further back. Such a V-shape organization reduces the drag force that each bird experiences compared to flying alone. Also, this orientation allows the birds to communicate more easily to keep the flock together, thus minimizing the possibility of losing birds during migration over long distances. The theory behind swarm of birds V-formation energy conservation was proved by researchers through monitoring flying birds' heartbeats. Also, it has been proven that a swarm of birds in V-formation can fly up to 70% further distance than one bird, with the same effort [21].

The leading bird works the hardest due to its flying into undisturbed air. The flying of this bird optimizes the aerodynamics of the two birds following it. The two birds behind the leading bird help also reducing the drag that is experienced by the leading bird. The V-formation eases the flight of all birds through energy conservation with the cost of the bird in the lead position which works the hardest. When this bird is tired, it will leave the lead position and move into one of the lines of the V-formation from its back. As a consequence, a bird from the back of the V-formation will take the leading position, thus keeping the formation together. This self-organizing behaviour allows all birds to be leaders as well as to be in the back of the formation to save
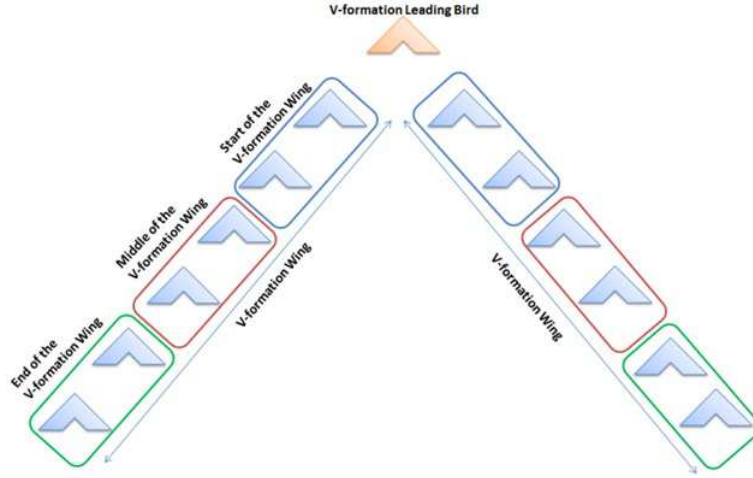
Fig. 3.1. *Birds V-formation organization during migration*

energy. The V-formation model is used as inspiration for the current approach to logically organize servers in data centres.

3.2. Concepts Definition. For developing the swarm-inspired consolidation technique, the data centre consolidation problem must be formalized.

The *data centre workload* is formally represented as a set of virtualized tasks (VTs) annotated with Resource Allocation (RA) requirements. Virtualization provides a uniform and dependency-free management of server workload tasks, while enabling facilities like virtualized task migration.

The *virtual task* is formally defined using its RA requirements for the server's processor (CPU), memory (MEM) and hard disk (HDD) computational resources as follows:

$$VT = [CPU_{req}, MEM_{req}, HDD_{req}] \tag{3.1}$$

The data centre is defined and modelled as a bi-dimensional array, C_{ij}, storing the data centre available servers and the allocation of computational resources. If the virtualized task VT_i is placed on the server S_j, then $C_{ij} = 1$, otherwise $C_{ij} = 0$.

In this context, the server consolidation problem is reduced to finding the optimal C_{ij} configuration in which the computational resources of the data centre' servers are efficiently used. Formula 3.2 shows an example of a bi-dimensional array C_{43} modelling a small data centre with three servers (S_1, S_2, S_3) that accommodate four virtual tasks (VT_1, VT_2, VT_3, VT_4). According to the C_{ij} array definition, the tasks are deployed as follows: VT_1 and VT_4 are placed on server S_1, VT_2 is placed on server S_2 and VT_3 is placed on server S_3.

$$C_{43} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{3.2}$$

A *data centre server* is defined by its state and the load level of its resources:

$$S_i = \{[CPU_S, MEM_S, HDD_S], state_S\} \tag{3.3}$$

where CPU_S, MEM_S and HDD_S are the current load of server's resources, and *state* is defined as follows:

$$state_S = \{ACTIVE, IDLE, OFF\} \tag{3.4}$$

In the ACTIVE state the server is turned on and runs virtualized tasks. In IDLE state the server is turned on without running any virtualized task while in OFF state the server is completely shut down.

For a server, two energy efficient, optimal resource allocation levels are defined: *low optimal resource allocation level* and *high optimal resource allocation level*.

The *low optimal allocation level*, $S_{optimal-low}$, is defined as the server resources load value for which the server does not execute workload:

$$S_{optimal-low} = [0\%, 0\%, 0\%] \tag{3.5}$$

$S_{optimal-low}$ is used to determine when the server does not execute any workload and can be turned off.

The *high optimal allocation level*, $S_{optimal-high}$, is defined as the server resources load value for which the server is most efficiently used.

$$S_{optimal-high} = [CPU_{opt_{load}}, MEM_{opt_{load}}, HDD_{opt_{load}}] \tag{3.6}$$

$S_{optimal-high}$ can be determined by measurements or by using the vendors' specification and is usually around 80% of the maximum server resource utilization capacity.

The *server computational resource under-utilization threshold*, T_{under}, is defined as a percentage of the server optimal configuration computational resources values, $S_{optimal-high}$ (for example [5%, 5%, 5%]).

For the proposed technique, the IT facility aspects of the data centre are not considered, the focus of the self-organizing and consolidation algorithm being on the IT computing resources.

3.3. V-formation Self-Organization Model of a Data Centre. Starting from the V-formation self-organization model of migrating birds for saving energy, a data centre servers' logical self-organization model to accommodate and execute the incoming workload in an energy efficient manner can be defined. For defining the data centre swarm-inspired self-organizing model, data centre related concepts and resources are mapped onto the birds V-formation concepts as shown in Table 3.1.

TABLE 3.1
V-formation - data centre concepts mapping

Birds V-formation organization model	V-formation self-organization model of Data Centres
Leading bird	Cluster of active servers
Birds immediately following leading bird	Cluster of fully loaded active servers
Birds in the middle of the V-formation	Cluster of turned off servers
Birds at the end of the V-formation	Cluster of idle servers

The V-formation organization allows for mapping the specific clusters of servers on each V-formation area and to self-organize server clusters inspired from the birds' behaviour and movements in the V-formation.

Figure 3.2 illustrates the swarm-inspired logical self-organization model of the data centre servers. In the model, each data centre server has an attached intelligent agent which implements behaviour similar with the behaviour of a bird in the V-formation defined by the swarm-inspired consolidation algorithm from Section 3.4.

The arrows in the figure represent the servers' possible movement between clusters. Using the V-formation self-organization model, the data centre servers are logically grouped as follows:

- V-formation leading bird is modelled as a cluster of *servers that are in active state*. The servers execute workload consisting of virtualized tasks but their high optimal allocation level, $S_{optimal-high}$ is not yet reached, thus new incoming workload may be accommodated.
- Birds that are immediately following the leading bird in the V-formation wings are represented by a *cluster of fully loaded active servers*. These servers have reached the $S_{optimal-high}$ high optimal allocation level and other virtual tasks could not be deployed on them. After finishing the execution of the already deployed workload, these servers are candidates for powering down.
- Birds in the middle of the V-formation are represented by a *cluster of powered down servers* which have reached the low optimal allocation level $S_{optimal-low}$ after executing their workload.
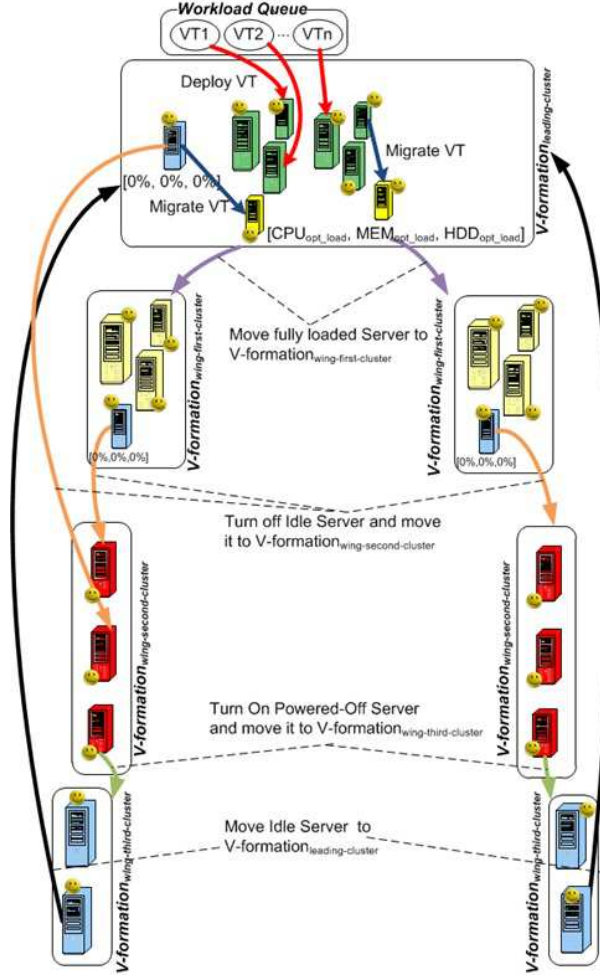
Fig. 3.2. *Data centre self-organization model inspired from birds V-formation organization during migration*

- Birds at the end of the V-formation are represented by a *cluster of idle servers* reaching low optimal allocation level, $S_{optimal-low}$. These servers are candidates for transitioning to active states and for accommodating new incoming workload when the servers from the leading clusters cannot satisfy the requirements of the incoming workload.

We formally define the *V-formation leading cluster* as:

$$V_{leading-cluster} = \{S_i | S_i.state = TURNED\ ON \land \exists\ j\ such\ that\ C_{ij} = 1\} \tag{3.7}$$

A *V-formation wing* is modelled by a set of three data centre clusters. The *first wing cluster* contains fully loaded active servers that are candidates for powering down after finishing the execution of their deployed workload:

$$\begin{aligned} V_{wing-first-cluster} &= \{S_i | S_i.state = TURNED\ ON \land \exists\ j\ such\ that \\ &\quad C_{ij} = 1 \land S_i \cong S_{optimal-high}\} \end{aligned}$$

$$\tag{3.8}$$

The *second wing cluster* contains turned off servers organized as a queue and is defined as follows:

$$V_{wing-second-cluster} = \{S_i | S_i.state = TURNED\ OFF \land C_{ij} = 0\ \forall\ j\} \tag{3.9}$$

The *third wing cluster* contains idle servers that are candidates for passing to active states. The reason behind constructing and using the V-formation wing third cluster is to make servers available on short notice to the V-formation leading cluster for accommodating new workload. This way the server wake up time delays and performance penalties are avoided. The third wing cluster is formally defined as:

$$V_{wing-third-cluster} = \{S_i | S_i.state = TURNED\ ON \wedge C_{ij} = 0\ \forall\ j\} \tag{3.10}$$

3.4. The Self-Organizing and Consolidation Algorithm. The logical movement of servers between the clusters defined by the data centre V-formation model of organization follows the birds' movements in the swarm during migration. Server's logical movement decision is taken by each server attached agent that implements the Self-Organizing and Consolidation Algorithm (see Algorithm 3.4). Each server attached agent features ticker behaviour. The agents monitor their attached server, communicate and collaborate among them to decide on the following types of actions: server placement in the appropriate V-formation cluster, resource consolidation actions (workload deployment and migration) and dynamic power management actions (turn on/off server). The algorithm takes as inputs the current monitored server state (its workload levels), a queue containing the incoming workload virtualized tasks, the values of the two power efficient server optimal resource allocation levels, $S_{optimal-low}$ and $S_{optimal-high}$, the server computational resource under-utilization threshold and the V-formation clusters, T_{under}. The algorithm is implemented as the server's agent behaviour.

[!h] Self-Organizing and Consolidation Algorithm **Input:** S - the server managed by the agent; $VT_{queue} = \{VT_1, .., VT_n\}$ - the virtualized tasks queue; $S_{optimal-low}$; $S_{optimal-high}$; T_{under}; $V_{leading-cluster}$; $V_{wing-first-cluster}$; $V_{wing-second-cluster}$; $V_{wing-third-cluster}$

Output: Self-organizing and consolidation actions
begin
  foreach $TICK$ **do**
    if $(S \in V_{leading-cluster})$ **then**
      Task_Migration$(S, V_{leading-cluster}, T_{under}, S_{optimal-high})$
      NotifyAgents$(V_{leading-cluster}, Migration)$
      if $(\textbf{Distance}(S, S_{optimal-high}) < T_{under})$ **then**
        Move$(S, V_{wing-first-cluster})$
      end if
      if $(\textbf{Distance}(S, S_{optimal-low}) < T_{under})$ **then**
        Move$(\textbf{Turn\_Off\_Server}(S), V_{wing-second-cluster})$
      end if
      if $(VTqueue \neq \emptyset)$ **then**
        $VT = \textbf{Dequeue}(VTqueue)$
        if $(\textbf{Task\_Deployment}(S, S_{optimal-high}, T_{under}, VT))$**then**
          NotifyAgents$(V_{leading-cluster}, Deployment)$
          if $(\textbf{Distance}(S, S_{optimal-high}) < T_{under})$ **then**
            Move$(S, V_{wing-first-cluster})$
            NotifyAgents$(V_{wing-third-cluster}, IdleToActive)$
          end if
        end if
      end if
    end if
    if $(S \in V_{wing-first-cluster})$ **then**
      if$(\textbf{Distance}(S, S_{optimal-low}) < T_{under})$ **then**
        Move$(\textbf{Turn\_Off\_Server}(S), V_{wing-second-cluster})$
      end if
    end if
    if $(S \in V_{wing-second-cluster})$ **then**
      if $(\textbf{ReceivedNotification}(TurnOFFToIdle))$ **then**
        Move$(\textbf{Turn\_ON\_Server}(S), V_{wing-third-cluster})$

        end if
      end if
      if $(S \in V_{wing-third-cluster})$ **then**
        if (**ReceivedNotification**($IdleToActive$)) **then**
          Move($S, V_{leading-cluster}$)
          NotifyAgents($V_{wing-second-cluster}, TurnOFFToIdle$)
        end if
      end if
    end foreach
  end

On every tick, if the server is part of the $V_{leading-cluster}$ cluster, its attached agent will try to consolidate the server workload by using task migration (lines 5-7). If after task migration a $V_{leading-cluster}$ of servers becomes fully loaded, its attached agent will decide to move it to the $V_{wing-first-cluster}$ (lines 8-10). Otherwise, if a server becomes empty, its attached agent decides to turn it off and moves it to the $V_{wing-second-cluster}$ (lines 11-13).

As long as there is an incoming workload, the attached agents of the $V_{leading-cluster}$ will try to properly deploy the virtualized tasks on the cluster of servers. If a server from $V_{leading-cluster}$ can accommodate the current virtual task, then the task is deployed and the attached agent signals the task deployment to the attached agents of $V_{leading-cluster}$ (see lines 14-17). After deployment, the server load level is tested and if it is fully loaded then it is shifted to the $V_{wing-first-cluster}$ cluster and a notification that a server must be moved from idle state to active state is sent to the agents attached to $V_{wing-third-cluster}$ servers (see lines 18-21). Following this notification, an agent attached to a server part of the $V_{wing-third-cluster}$ wakes up a server form idle state (the server is randomly chosen) and moves it to the $V_{leading-cluster}$ to accommodate the incoming workload (lines 35-40). At the same time, a notification is sent to the $V_{wing-second-cluster}$ to signal that a turned-off server must be powered-on and put in idle state (line 38) to replace the server that was moved in the $V_{leading-cluster}$.

If the server is part of the $V_{wing-first-cluster}$, its attached agent periodically evaluates if the server deployed workload has finished its execution (see line 26). If this is the case, the server is turned off and moved to the $V_{wing-second-cluster}$ containing turned off servers (see line 27). For the $V_{wing-second-cluster}$, if a server attached agent receives a turn off to idle notification (see line 31) the server is turned on and moved to $V_{wing-third-cluster}$ (line 32).

To identify the appropriate servers from the V-formation leading cluster on which a task can be *deployed* the following steps are taken: (i) a representation of the servers after virtualized tasks deployment is constructed (see Algorithm 3.4, line 5) and (ii) the degree of similarity between the configuration of the servers after deployment and the optimal one is calculated (see Algorithm 3.4, line 6).

[!h] Task Deployment Algorithm **Input:** S; $S_{optimal-high}$; T_{under}; $VT_j = [CPU_{reqj}, MEM_{reqj}, HDD_{reqj}]$
Output: Boolean value signalling VT_j correct deployment
begin
  $bool = false$
  $S_{new} = $ **Server_Load_After_Deploy**(S, VT_j)
  $d = $ **Distance**($S_{new}, S_{optimal-high}$)
  if ($d < T_{under}$) **then**
    $bool = $ **Deploy**(VT_j, S)
  end if
  return $bool$
end

To calculate the degree of similarity of server configurations the following formula is used:

$$d(S_i, S_j) = [|CPU_{S_i} - CPU_{S_j}|, |MEM_{S_i} - MEM_{S_j}|, |HDD_{S_i} - HDD_{S_j}|] \tag{3.11}$$

The server, for which the distance is the smallest (all three term inequalities should be satisfied), is selected to accommodate the task (see Algorithm 3.4 lines 7-9).

For migrating virtualized tasks, the data centre servers are analysed to identify the servers that can accommodate virtual tasks from the server subject of migration. Firstly the algorithm tests if the server computing resources are inefficiently utilized (see Algorithm 3.4, line 5). This situation appears mainly because the workload virtualized tasks have different execution times and consequently, some of them will last longer than others. If the distance between the server computational resources utilization values and the optimal configuration values ($S_{optimal-high}$) drops below a specific value (for example 35%) then the virtualized tasks deployed on the server are iterated for migration (see Algorithm 3.4, lines 6-12).

[!h] Task Migration Algorithm **Input**: S; $V_{leading-cluster}$; T_{under}; $S_{optimal-high}$;

Output: Virtual tasks are migrated

begin

   $bool = false$

   if (**Distance**$(S, S_{optimal-high}) < [35\%, 35\%, 35\%]$) **then**

     foreach VT_j *deployed on S* **do**

       foreach $S_i \neq S \in V_{leading-cluster}$ **do**

         if (!$bool$) **then**

           $bool = $**Task_Deployment**$(S_i, S_{optimal-high}, T_{under}, VT_j)$

         end if

       end foreach

     end foreach

   end if

end

4. Case Study and Results. This section presents and discusses the results obtained by using the proposed swarm-inspired self-organizing and consolidation technique.

4.1. Test Case Description. Due to costs, management and security constraints, it is very difficult to deploy and test the proposed solution in a large, real data centre. Therefore, for evaluating the swarm inspired data centre self-organizing technique, a data centre with 2000 servers was simulated. Each simulated server models the characteristics of a real server and is represented as an object having as main attributes the CPU, memory and hard disk capabilities. The real server used as a model for the simulated server has the following hardware configuration: CPU - Intel(R) i7 870 2.93GHz, memory - 6GB DDR3 and hard disk - 750GB. The servers attached agents which implement the data centre self-organizing and consolidation algorithm are developed using the JADE framework [37].

To test the power saving capabilities of the swarm-inspired technique, a random workload that needs to be accommodated in the simulated data centre has been generated. The workload consists of groups of virtual tasks with different requirements, arriving sequentially, during a time interval of one hour. Each virtual task is described by its request for server resources, as defined in Formula 3.1.

It is assumed that the data centre is managed by OpenNebula, the state of the art middleware for managing virtualized data centre clouds [33]. OpenNebula uses a server consolidation algorithm based on Fit First approach [34] (which places the incoming workload on the first server in which it will fit). Two scenarios are considered for the test case: (1) the workload is accommodated in the simulated data centre by the default OpenNebula Fit First consolidation algorithm and (2) the proposed self-organizing and consolidating technique is in charge, replacing the OpenNebula algorithm.

For the first scenario, the Fit First server consolidation algorithm which deploys the new coming tasks on the first server found that can accommodate it, also overprovisions the computing resources to handle peak load values. In this case, there will always be servers in the data centre that are powered-on without executing workload.

In the second scenario, using the proposed data centre self-organizing and consolidating technique, the number of up and running servers that accommodate the workload tasks is variable. If the size of the workload increases over the total capacity of the $V_{leading-cluster}$, then the intelligent agents collaborate to adjust the number of servers in the cluster in order to be able to distribute and deploy all of the workload virtual tasks. If after the task deployment, there are servers in the $V_{leading-cluster}$ which don't execute any workload, the attached agents decide to power off these servers to save energy.

To assess the energy consumption improvement, a one hour workload that needs to be deployed in the data centre has been generated. In the evaluation process two aspects are considered for both scenarios when measuring the energy efficiency of the self-organizing and consolidating technique: (i) computing the Green Performance Indicator (GPI), DH-UR and (ii) estimating the average power consumption.

4.2. DH-UR Metric Evaluation. The Deployed Hardware Utilization Ratio (DH-UR) [26] is a Green Performance Indicator which measures the number of IT computing resources (servers) that consume energy without doing any actual work (see Formula 4.1). In other words, it measures the energy required by the idle systems, or the amount of wasted energy.

$$DH - UR = \frac{Number\ of\ Servers\ Running\ Applications}{Number\ of\ Servers\ Up\ and\ Running} \tag{4.1}$$

The value of this metric for a green data centre should be close to 1, meaning that the up and running servers are actually processing tasks. The DH-UR values for the simulated data centre have been calculated for both consolidation algorithms. Table 4.1 shows the average DH-UR values highlighting the fact that the proposed swarm-inspired approach improves the utilization of server resources and at the same time optimizes the energy efficiency. It can be noticed that the Fit First approach keeps servers under-utilized leading to a lower average value for DH-UR indicator.

TABLE 4.1
Average DH-UR metric values for the two scenarios

Algorithm	Average DH-UR Value
Swarm-inspired data centre self-organizing and consolidating algorithm	0.98
Fit First server consolidation algorithm	0.73

Figure 4.1 presents the DH-UR value evolution for the two test cases. It can be noticed that when using the swarm-inspired technique, the DH-UR value is close to 1 for all test period, while for the OpenNebula default Fit First algorithm it barely jumps over 0.8. The latter value reflects the case of today's real data centres.
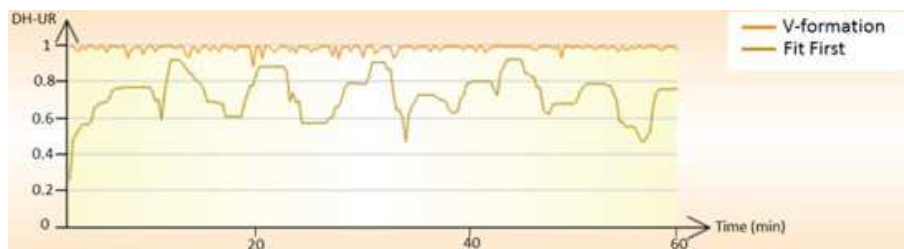

FIG. 4.1. *DH-UR Green Indicator for the two test cases*

4.3. Power Consumption Evaluation. For power consumption estimation, the same simulated data centre with 2000 homogenous servers has been used. Each simulated server models the characteristics of a real server: CPU - Intel(R) i7 870 2.93GHz, memory - 6GB DDR3 and hard disk - 750GB.

To estimate the data centre power efficiency, the instant power consumption of the real server has been measured using a power meter, in three situations: (i) the server is up and running and does not execute any tasks (Idle Power Mode), (ii) the server is up and running and executes workload (Working Power Mode) and (iii) the server is up and running and fully loaded (Full Load Power Mode). The server power consumption was measured using the ISO-TECH IPM3005 [35] power meter and the obtained results are listed in Table 4.2.

TABLE 4.2
The data centre server states and their associated instant power consumption

Power Mode	Load Level	Instant Power Consumption
Idle	0-10%	70W
Working	10-80%	100W
Full Load	80-100%	130W

For estimating the whole data centre power demand, the following metric has been defined and used:

$$
\begin{aligned}
PW(DC) \;=\; & \#IdleServers * PowerIdle+ \\
& \#WorkingServers * PowerWorking+ \\
& \#FullLoadServers * PowerFullLoad+ \\
& PW_{overhead}
\end{aligned}
$$

(4.2)

where $\#IdleServers$ is the number of data centre servers that are in Idle Power Mode, $\#WorkingServers$ represents the number of servers that are in Working Power Mode, $\#FullLoadServers$ is the number of servers in Full Load Power Mode, while $PW_{overhead}$ is the power overhead induced by the algorithms' management operations.

$PW_{overhead}$ is estimated by measuring the instant power consumption of each management action defined by the swarm inspired technique. The following types of management actions are measured: (1) deploy workload virtual task on a leading cluster server, (2) migrate a task between two servers from the $V_{leading-cluster}$, (3) hibernate server when completing its workload execution and (4) wake up a server to accommodate incoming workload.

The measured overhead for each action (together with the software tools used for enforcing the actions) is listed in Table 4.3. Apart from the OpenNebula middleware, SSH and WakeOnLan software tools are used to turn off and on the servers. The movement actions between different V-formation clusters are logical actions and do not consume any extra energy.

TABLE 4.3
Power consumption overhead of the swarm-inspired technique actions

Action	Involved Tools	Instant Power Consumption
Deploy	OpenNebula	10W
Migrate	OpenNebula	20W
Turn off Server	SSH	100W
Wake up Server	WakeOnLan	100W

The overall power consumption overhead of the swarm-inspired technique is estimated as follows:

$$
\begin{aligned}
PW_{overhead} \;=\; & \#TasksDeployed * PowerDeploy+ \\
& \#TasksMigrated * PowerMigrate+ \\
& \#ServerTurnedOn * PowerWakeUp+ \\
& \#ServerTurnedOff * PowerTurnOff
\end{aligned}
$$

(4.3)

where $\#TasksDeployed$ represents the number of workload tasks that were deployed, $\#TasksMigrated$ is the number of migrated workload tasks, $\#ServerTurnedOn$ is the number of servers that were turned on, while $\#ServerTurnedOff$ represents the number of turned off servers.
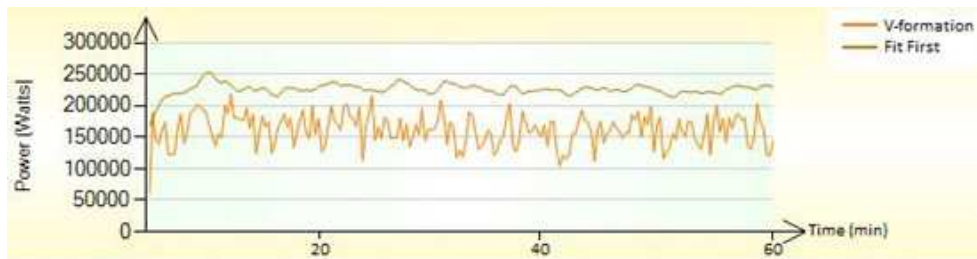
FIG. 4.2. *Power consumption evolution for the two test cases*

Figure 4.2 shows the variation of the estimated power consumption for the simulated data centre when accommodating a randomly generated workload with the two consolidation approaches alternatively activated.

When using the proposed swarm-inspired self-organizing and consolidating approach, only the servers that execute workload ($V_{leading-cluster}$ and $V_{wing-first-cluster}$) are up and running and consuming energy . The number of $V_{wing-first-cluster}$ idle servers is low and a high number of servers are kept in turned off state in the $V_{wing-third-cluster}$. The higher power demand of the Fit First consolidation algorithm is due to the high number of idle servers that are not executing tasks while demanding 70 Watts instant power.

The calculated average power demand values for the two tested scenarios are given in Table 4.4. The average values were computed for 2000 servers in the data centre and for one hour time interval.

TABLE 4.4
Average power consumption

Algorithm	Average Power Demand
Swarm-inspired data centre self-organizing and consolidating algorithm	161 670 W
Fit First server consolidation algorithm	226 322 W

As a result, an average power saving of around 40% has been estimated for the swarm-inspired approach compared with the OpenNebula default Fit First consolidation algorithm. Even though this percentage is really high it should be noted that the data centre IT facility aspect was neglected during simulation. In a real data centre, the IT facilities usually demand about 60% of the total [28]. This means that the power saving of 40% of the proposed solution represents about 16% from the total power demand of a data centre.

5. Conclusions. In this paper a swarm-inspired data centre consolidation technique which aims at increasing the data centre computing resources utilization ratio and implicitly its power efficiency is proposed. The results are promising showing that using the swarm-inspired technique the data centre Deployed Hardware Utilization Ratio increases with about 34%. The average power saving is around 40% from the power consumed by the data centre computing resources and about 16% from its total power consumption including the IT facility.

REFERENCES

[1] S. MURUGESAN, *Harnessing Green IT: Principles and Practices*, IEEE IT Professional, (2008) pp. 24-33.
[2] N. E. JERGER, D. VANTREASE, AND M. LIPASTI, *An Evaluation of Server Consolidation Workloads for Multi-Core Designs*, Proceedings of the IEEE International Symposium on Workload Characterization, (2007), pp. 47-56.

[3] M. Marzolla, O. Babaoglu and F. Panzieri, *Server Consolidation in Clouds through Gossiping*, Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks, (2011), pp. 1-6.

[4] B. W. Verdaasdonk, H. Koopman and F. van der Helm, *Energy efficient walking with central pattern generators: from passive dynamic walking to biologically inspired control*, Biological Cybernetics, Springer-Verlag, Volume 101, Issue 1 (2011), pp. 49-61.

[5] P. Boonma and J. Suzuki, *Biologically-inspired Adaptive Power Management for Wireless Sensor Networks*, Handbook of Wireless Mesh & Sensor Networking, Chapter 3.4.8, McGraw-Hill (2008), pp. 190 - 202.

[6] P. Champrasert and J. Suzuki, *SymbioticSphere: A Biologically-Inspired Autonomic Architecture for Self-Adaptive and Self-Healing Server Farms*, Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks, (2006), pp. 469-474.

[7] D. Barbagallo, E. Di Nitto, D. J. Dubois and R. Mirandola, *A Bio-Inspired Algorithm for Energy Optimization in a Self-organizing Data Centre*, Self-Organizing Architectures - First International Workshop, SOAR 2009, Cambridge, UK, September 14, 2009, Revised Selected and Invited Papers, LNCS, Volume 6090, (2010), pp. 127-151.

[8] C. Lee, H. Wada and J. Suzuki, *Towards a Biologically-inspired Architecture for Self-regulatory and Evolvable Network Applications*, Advances in Biologically Inspired Information Systems: Models, Methods and Tools, Studies in Computational Intelligence Volume 69, (2007), pp. 21-45.

[9] J. Torres, et al., *Tailoring Resources: The Energy Efficient Consolidation Strategy Goes Beyond Virtualization*, International Conference on Autonomic Computing, (2008), pp. 197 - 198.

[10] Y. Ajiro and A. Tanaka, *Improving Packing Algorithms for Server Consolidation, Proceedings of the Computer Measurement Group's*, (2007).

[11] J. Berral, et al., *Towards energy-aware scheduling in data centres using machine learning*, Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, (2010) pp. 215-224.

[12] L. Wang, et al., *Towards Thermal Aware Workload Scheduling in a Data Centre*, Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks, (2009) pp. 116-122.

[13] E. Kalyvianaki, and T. Charalambous, *On Dynamic Resource Provisioning for Consolidated Servers in Virtualized Data Centres*, Proceedings of the 8th Int. Workshop on Performability Modeling of Computer and Communication Systems (PMCCS-8), (2007).

[14] B. Speitkamp and M. Bichler, *A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centres*, IEEE Transactions on services computing, Volume 3, Issue 4, (2010).

[15] O. Niehorster and A. Brinkmann, *Autonomic Resource Management Handling Delayed Configuration Effects* Third IEEE International Conference on Coud Computing Technology and Science, pp. 138 - 145, 2011.

[16] J. O. Kephart, et al., *Coordinating Multiple Autonomic Managers to Achieve Specified Power-Performance Tradeoffs*, International Conference on Autonomic Computing, (2007).

[17] T. Cioara, I. Anghel, I. Salomie, G. Copil, D. Moldovan and B. Pernici, *A context aware self-adapting algorithm for managing the energy efficiency of IT service centres*, Ubiquitous Computing and Communication Journal, Volume 6 No. 1, (2011).

[18] L. Barroso, *The Price of Performance*, ACM Queue, Volume 3, Issue 7, (2005), pp. 48-53.

[19] R. J. Anderson, E. W. Mayr and M. Warmuth, *Parallel Approximation Algorithms for Bin Packing*, Stanford University, 1988.

[20] M. Poniatowski, *Foundations Of Green IT: Consolidation, Virtualization, Efficiency, and ROI in the Data Centre*, Prentice Hall, 2009.

[21] G. H. Schueller, and S. K. Schueller, *Animal Migration (Animal Behaviour)*, Chelsea House Pub (Library), 2009.

[22] B. Davis and K. Davis, *Marvels of Creation: Breathtaking Birds*, Master Books, 2006.

[23] C. L. Henderson, *Birds in Flight - The Art and Science of How Birds Fly*, Voyageur Press, 2008.

[24] P. Berthold, *Bird Migration - A General Survey*, Second Edition - Oxford University Press, 2001.

[25] J. M. Kaplan, W. Forrest and N. Kindler, *Revoluzioning Data Centre Energy Efficiency*, McKinsey&Company, 2008. Available online at: http://www.mckinsey.com/clientservice/bto/pointofview/pdf/Revolutionizing_Data_Centre_Efficiency.pdf

[26] J. Stanley, K. Brill and J. Koomey, *Four Metrics Define Data Centre Greenness*, Uptime Institute, white paper, Available online at: http://uptimeinstitute.org.

[27] S. Srikantaiah, A. Kansal and F. Zhao, *Energy Aware Consolidation for Cloud Computing*, Microsoft Research, 2009. Available online at: http://research.microsoft.com/pubs/75408/srikantaiah_hotpower08.pdf

[28] Climate Savers Computing Initiative, http://www.thegreengrid.org/Home/about-the-green-grid/TGGCSCI.aspx

[29] K. Brill, *Data Centre Energy Efficiency and Productivity*, The Uptime Institute, Whitepaper, 2007.

[30] C. B. Pop, I. Anghel, T. Cioara, I. Salomie, I. Vartic, *A Swarm-inspired Data Center Consolidation Methodology*, International Conference on Web Intelligence, Mining and Semantics, Article No. 41, 2012.

[31] U.S. Environmental Protection Agency, ENERGY STAR Program, Report to Congress on Server and Data Centre Energy Efficiency, Public Law 109-431, 2007.

[32] SMART 2020: Enabling the low carbon economy in the information age, Report by The Climate Group on behalf of the Global eSustainability Initiative (GeSI), 2008.

[33] OpenNebula, the open source toolkit for cloud computing, http://opennebula.org/.

[34] OpenNebula scheduler, http://opennebula.org/documentation:archives:rel3.0:schg

[35] ISO-TECH IPM3005, http://www.iso-techonline.com/products/iso-tech-electrical-installation-testers.html

[36] GAMES FP7 Research Project, http://www.green-datacentres.eu/.

[37] Jade, Java Agent Development Framework, http://jade.tilab.com.

[38] The Green Grid, http://www.thegreengrid.org/