A Swarm-inspired Data Center Consolidation Methodology

Cristina Bianca Pop, Ionut Anghel, Tudor Cioara, Ioan Salomie, Iulia Vartic
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
{cristina.pop, ionut.anghel, tudor.cioara, ioan.salomie, iulia.vartic}@cs.utcluj.ro

ABSTRACT
This paper proposes a swarm-inspired data center consolidation methodology which aims at reducing the power consumption in data centers while ensuring the workload execution within the pre-established performance parameters. Each data center server is managed by an intelligent agent that deals with its power efficiency by implementing a bird's migration-inspired behavior 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 consumption. The data center servers self-organize in logical clusters according to the birds V-formation self-organizing migration model. The results are promising showing that the swarm-inspired data center consolidation methodology optimizes the utilization ratio of the data center computing resources and achieves estimated power savings of about 16%.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *scheduling.*

I.6.5 [**Simulation and Modeling**]: Model Development – *modeling methodologies.*

General Terms
Design, Algorithms, Management.

Keywords
Swarm optimization, data center, server consolidation, birds V-formation.

1. INTRODUCTION
Over the last years the energy efficiency management of IT Processes, Systems and Data Centers has dramatically emerged as one of the most critical environmental challenges to be dealt with. As an example, worldwide data centers CO2 emissions are equivalent already to about half of the total airlines' CO2 emissions and are expected to grow from 76 MTCO2e to 259 MTCO2e by 2020 [1]. 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 [2]. Since computing demand and electricity prices rise, posing new environmental concerns, the energy consumption of IT systems and data centers is a high priority for the industry.

Currently, many researches, from both management and technical sites, are striving to reduce the energy consumption of IT data centers. In general, the data center administrators are always focused on the performance aspect when configuring the data center clusters or servers and ignore the energy/power consumption. The sub-optimal utilization of computing resources is one of the main factors reported in the literature that contributes to the data center high energy consumption [2]. One state of the art solution to the problem of data center computing resources under-utilization is resource consolidation which aims at combining the workloads that are executed on different machines (servers) so that an optimal number of computing resources are always used [3]. In this context, the problem of organizing and using the data center IT hardware resources in an energy/power efficient manner can be seen as an optimization problem.

Biology offers many clues for solving such optimization problems. The birds and insects are employing self-organizing behavioral strategies, at the group level, which help them live and survive in the harshest conditions by efficiently using and conserving energy. For example, penguins self-organize in case of cold temperatures and wind by huddling in circular formations [4]. In these formations, the penguins that 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 birds' V-formation in their migration process is another self-organizing behavioral strategy for conserving energy [5]. 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. When the leading bird gets tired, it is replaced by one of the following birds and takes a place in the V-formation wings in order to rest. In this context, by inspiring from the biological self-organizing behavioral strategies which have ensured the survivability of different biological species over time, data centers servers power efficient self-organizing strategies can be developed.

This paper proposes a server consolidation methodology for reducing power consumption in data centers inspired from the birds self-organizing behavior during migration. Each data center server has an attached intelligent agent which implements a behavior similar with the behavior of a bird in the V-formation. The V-formation leading birds are modeled by a cluster of active data center servers that can accommodate the incoming workload. The birds from the start of the V-formation wings are represented by a cluster of data center 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 modeled 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

WIMS'12, June 13-15, 2012 Craiova, Romania

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.

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

2. RELATED WORK

The data center servers' energy efficiency 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 centers from one server/cluster to another [18]. 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 [19]. 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 [20]. Two well-known heuristics for the bin packing based consolidation, the best-fit decreasing (BFD) and the first-fit decreasing (FFD), were used [21]. To enable energy efficient consolidation, the inter-relationships between energy consumption, resource utilization, and performance of consolidated workloads must be considered [22]. In [23] a consolidation methodology that uses machine learning to deal with uncertain information is proposed. Previous server behavior data is used to predict and estimate the current power consumption and also to improve the scheduling and consolidation decisions. A thermal aware workload scheduling and consolidation solution aiming to reduce the power consumption and temperatures in data centers was proposed in [24]. 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 centers servers CPU allocation and consolidation based on first order Kalman filter is presented. In [26] the server consolidation problem is approached for small data centers as a constraint satisfaction problem. The authors also propose a heuristic for approaching the server consolidation in large data centers. Optimal computing resources allocation techniques for server clusters based on reinforcement learning are proposed in [27]. Learning techniques are also used to trade-off between computing resources power consumption and performance during the allocation process [28]. In [29] the problem of dynamic server consolidation in virtualized data centers is approached by proposing the development of an energy aware run-time consolidation algorithm based on reinforcement learning.

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 [7]. 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 [9]. 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. In [8] a biologically-inspired agent based approach is used to manage the energy consumption in a wireless sensors network. The agent behavior 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 behavior of insects in search for a proper migration place, the authors of [10] propose a method for optimizing the energy consumption in data centers. The migrating insect is modeled by a running virtual machine, a colony of insects is modeled as a set of virtual machines running on the same server, and the candidate migration places are modeled 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. The criteria used for choosing a suitable server where to migrate virtual machines includes the server's power consumption class and its amount of free resources. An immune-inspired method for designing applications capable of adapting to network dynamic changes is proposed in [11]. The environmental conditions are modeled as antigens, while the agents (used to design applications) are modeled as biological entities having an immune system. In [6], 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 neighbors 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 regardless their resource needs which are not considered in this approach.

3. THE SWARM-INSPIRED CONSOLIDATION METHODOLOGY

This section introduces the server consolidation methodology inspired from bird's behavior during migration. The goal of the methodology is to reduce the data center power consumption while ensuring the workload execution within the pre-established parameters.

3.1 Methodology Prerequisites

For developing a swarm-inspired consolidation methodology, the data center consolidation problem must be formalized.

The *data center workload* is formally represented as a set of virtualized tasks 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:

$$VT = [CPU_{req}, MEM_{req}, HDD_{req}] \quad (1)$$

The *data center* is modeled as a bi-dimensional array, C_{ij}, describing the data center servers and the computational resources allocation. 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 center' servers are efficiently used.

A *data center server* is defined as:

$$S_i = \{[CPU_S, MEM_S, HDD_S], state\} \quad (2)$$

where CPU_S, MEM_S and HDD_S are the server's resources current load, and *state* reflects the state of the server (ACTIVE - turned on and running virtualized tasks, IDLE - turned on without running any virtualized tasks , OFF - completely shut down). For a server, two power efficient optimal resource allocation levels are defined (see relations 3 and 4). In the case of the first optimal allocation level (relation 3) the server does not execute any workload and can be turned off, while in the case of the second optimal allocation level (relation 4), the server is active and its resources are most efficiently used (this is determined by measurements or by using the vendors' specification; usually it is around 80% of the resource utilization level).

$$S_{optimal-low} = [0\%, 0\%, 0\%] \quad (3)$$

$$S_{optimal-high} = [CPU_{opt_{load}}, MEM_{opt_{load}}, HDD_{opt_{load}}] \quad (4)$$

3.2 Data Center V-formation Organization Model

Starting from the birds' self-organization model in V-formations to save energy when migrating, we have defined a data center servers' logical self-organization model to accommodate and execute the incoming workload in energy efficient manner. Figure 1 illustrates the swarm-inspired logical self-organization model of data center servers. In our model each data center server has an attached intelligent agent (see the smiley symbol in Figure 1) which implements a behavior similar with the behavior of a bird in the V-formation defined by the swarm-inspired consolidation algorithm from Section 3.3.

Using the migrating birds V-formation, in our self-organization model the data center servers are logically grouped as follows:

- the V-formation leading birds are modeled as a cluster of servers that are in active state (marked with green in Figure 1), execute workload, and can accommodate incoming workload
- the birds that are immediately following the leading birds, in the V-formation wings, are represented by a cluster of data center fully loaded active servers (marked with yellow in Figure 1) that are candidates for powering down after finishing the execution of their deployed workload
- the birds in the middle of the V-formation are represented by a cluster of servers containing powered down servers (marked with red in Figure 1)
- the birds in the back of the V-formation are represented by a cluster of idle servers (marked with blue in Figure 1) candidates for passing to active states and for accommodating incoming workload.

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

$$V_{leading-cluster} = \{S_i | S_i.state = \ TURNED\ ON\ \wedge \exists j\ such\ that\ C_{ij} = 1\} \quad (5)$$

A *V-formation wing* is modeled by a set of three data center clusters. The *first wing cluster* contains fully loaded active servers that are candidates for powering down after finishing the execution of their deployed workload (relation 6).

$$V_{wing-first-cluster} = \{S_i | S_i.state = \ TURNED\ ON\ \wedge \exists j\ such\ that\ C_{ij} = 1 \wedge S_i \cong S_{optimal-high}\} \quad (6)$$

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

$$V_{wing-second-cluster} = \{S_i | S_i.state = \ TURNED\ OFF\ \wedge C_{ij} = 0\ \forall j\} \quad (7)$$

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 follows:

$$V_{wing-third-cluster} = \{S_i | S_i.state = \ TURNED\ ON\ \wedge C_{ij} = 0\ \forall j\} \quad (8)$$

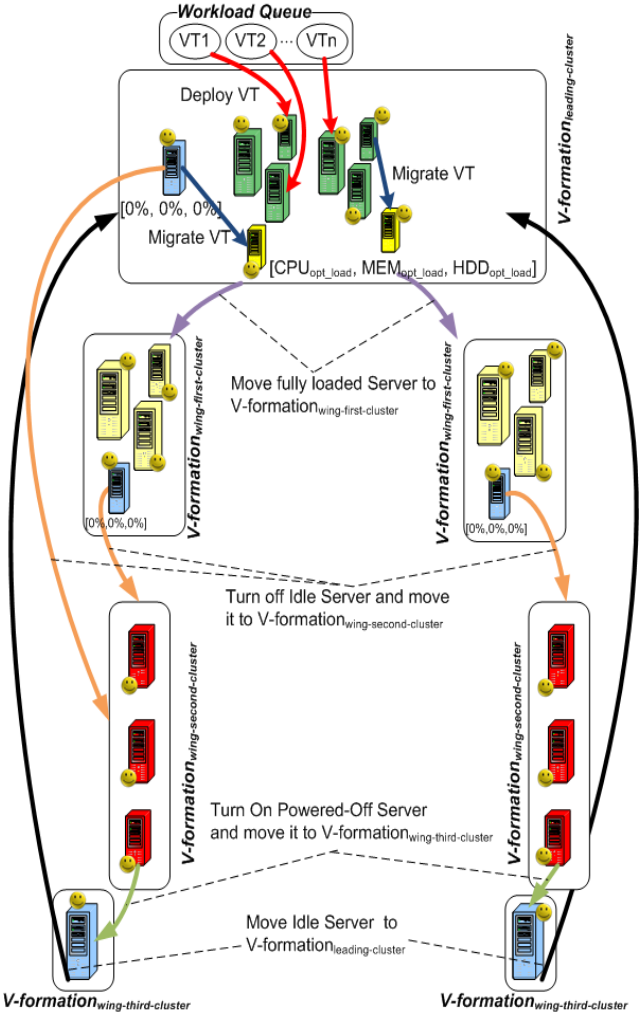

Figure 1. Servers' organization model inspired from birds migration V-formation. The arrows represent the servers' possible movement between clusters.

3.3 The Consolidation Algorithm

The servers' logical movement between the clusters defined by the data center V-formation organization model follows the birds' movements in the swarm during migration. The server logical movement decision is taken by each server attached agent using the *Swarm-inspired Consolidation Algorithm.* Each server attached agent features a ticker behavior. They monitor their

attached server, communicate and collaborate 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 *Swarm-inspired Consolidation Algorithm* takes as inputs the current monitored server state (its workload levels and data center cluster membership), a queue containing the incoming workload virtualized tasks and the two power efficient optimal resource allocation levels for a server (defined in sub-section 3.1).

Swarm-inspired Consolidation Algorithm

Input: S – the server managed by the inteligent agent;
$VT_{queue} = \{VT_1, ..., VT_n\}$ – the virtualized tasks queue;
$S_{optimal\text{-}low}$, $S_{optimal\text{-}high}$.
Observation: the algorithm is implemented as the agent's behaviour
begin

1. **foreach** *TICK* **do**
2. **if** ($S \in V_{leading\text{-}cluster}$)
3. **Task_Migration**(S, $V_{leading\text{-}cluster}$, T_{under}, $S_{optimal\text{-}high}$,$S_{optimal\text{-}low}$)
4. **NotifyAgents** ($V_{leading\text{-}cluster}$, *Migration*)
5. **if** (**Distance**(S, $S_{optimal\text{-}high}$)< T) **then**
6. **Move**(S, $V_{wing\text{-}first\text{-}cluster}$)
7. **if** (**Distance**(S, $S_{optimal\text{-}low}$)< T) **then**
8. **Move**(S, $V_{wing\text{-}second\text{-}cluster}$)
9. **if** (VT_{queue} != {}) **then**
10. VT = **Dequeue**(VT_{queue})
11. **if** (**Task_Deployment**(S, $S_{optimal\text{-}high}$, VT)) **then**
12. **NotifyAgents**($V_{leading\text{-}cluster}$, *Deployment*)
13. **if** (**Distance**(S,$S_{optimal\text{-}high}$)< T) **then Move**(S,$V_{wing\text{-}first\text{-}cluster}$)
14. **else NotifyAgents** ($V_{wing\text{-}third\text{-}cluster}$, *IdleToActive*)
15. **if** ($S \in V_{wing\text{-}first\text{-}cluster}$) **then**
16. **if** (**Distance**(S, $S_{optimal\text{-}low}$)< T) **then**
17. **Move**(**Turn_Off_Server**(S), $V_{wing\text{-}second\text{-}cluster}$)
18. **if** ($S \in V_{wing\text{-}second\text{-}cluster}$) **then**
19. **if (ReceivedNotification (***TurnOFFToIdle***) then**
20. **Move**(**Turn_ON_Server**(S), $V_{wing\text{-}third\text{-}cluster}$)
21. **if** ($S \in V_{wing\text{-}third\text{-}cluster}$) **then**
22. **if (ReceivedNotification (***IdleToActive***)) then**
23. **Move**(S, $V_{leading\text{-}cluster}$)
24. **NotifyAgents** ($V_{wing\text{-}second\text{-}cluster}$, *TurnOFFToIdle*)
25. **end foreach**

end

On every tick if the server is part of the $V_{leading\text{-}cluster}$ cluster its attached agent will first try to consolidate the server workload by using task migration (lines 2-3). If after task migration a $V_{leading\text{-}cluster}$ server becomes fully loaded, its attached agent will decide to shift it to the $V_{wing\text{-}first\text{-}cluster}$ cluster (lines 5-6). Otherwise, if a server becomes empty, its attached agent decides to turn it off and shifts it to the $V_{wing\text{-}second\text{-}cluster}$ cluster (lines 7-8).

As long as there is an incoming workload, the $V_{leading\text{-}cluster}$ cluster servers attached agents will try to properly deploy the virtualized tasks on the cluster servers. If a server from $V_{leading\text{-}cluster}$ can accommodate the current virtual tasks, then the task is deployed on it and its attached agent signals the task deployment to the other $V_{leading\text{-}cluster}$ servers attached agents (see lines 9-12). After deployment, the server load level is tested and if it is fully loaded, then it is shifted to the $V_{wing\text{-}first\text{-}cluster}$ cluster (see line 13). If no server which can accommodate the current virtual tasks is found on the $V_{leading\text{-}cluster}$ cluster, then a notification that a server must be moved from idle state to active state is sent to the agents attached to $V_{wing\text{-}third\text{-}cluster}$ servers (see line 14). Following this notification, the agents attached to a server part of the $V_{wing\text{-}third\text{-}cluster}$ cluster wake a server form idle state and shift it to the $V_{leading\text{-}cluster}$ to accommodate the incoming workload (lines 21-23). Also a notification is sent to the $V_{wing\text{-}second\text{-}cluster}$ agents to put a server in idle state (line 24).

If the server is part of the $V_{wing\text{-}first\text{-}cluster}$, its attached agent periodically evaluates if the server deployed workload has finished its execution (see line 16). If this is the case, the server is turned off and moved to the $V_{wing\text{-}second\text{-}cluster}$ (see line 17). The servers part of the $V_{wing\text{-}second\text{-}cluster}$ are turned off servers. If a server attached agent receives a turn to idle notification (see line 19) the server is turned on and moved to $V_{wing\text{-}third\text{-}cluster}$ (line 20).

To identify the appropriate servers from the V-formation leading cluster on which a server's deployed virtual tasks can be migrated, all the servers are analyzed, a description of the servers state after virtualized tasks deployment is constructed, and the similarity between the obtained servers and the optimal active servers configuration ($S_{optimal\text{-}high}$, $S_{optimal\text{-}low}$) is calculated. To calculate the degree of similarity, a three dimensional space having the server's main computational resources as axis, is defined (see Figure 2).

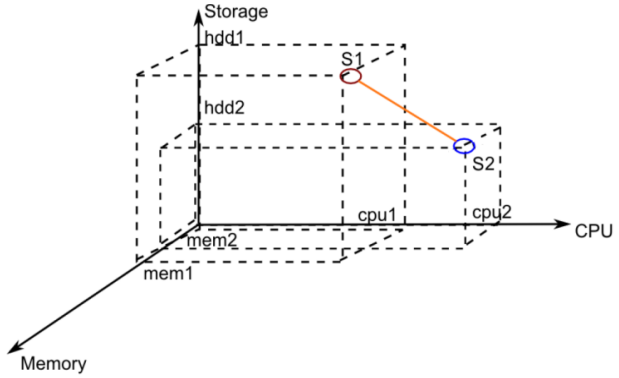

Figure 2. Computing the similarity degree for two servers S1 and S2

The similarity degree between two servers load values in the three dimensional space is evaluated using the Manhattan distance between the two servers as in relation 9.

$$\Delta = d(S_i, S_j) = |CPU_{S_i} - CPU_{Sj}| + |MEM_{S_i} - MEM_{Sj}| + |HDD_{S_i} - HDD_{Sj}| \quad (9)$$

4. CASE STUDY AND RESULTS

Due to costs, management and security constraints, it is very difficult to deploy and test the swarm-inspired consolidation methodology in a large real data center. Thus for testing purposes we have simulated a data center with 2000 servers as IT computing resources. The IT facility aspects of the data center are not the subject of our consolidation methodology thus they are ignored during testing. Each simulated data center server models the characteristics of a real server and is represented as an object having as main attributes the server CPU, MEM and HDD capabilities. The servers attached agents which implement the consolidation algorithm presented in Section 3 are developed using the JADE framework [17]. The real data center server used as a model has the following hardware configuration: CPU - Intel(R) i7 870 2.93GHz , MEM - 6GB DDR3 and HDD - 750GB.

The workload that the simulated data center needs to accommodate is randomly generated and it consists of groups of virtual tasks arriving sequentially. Each virtual task is described by its request for the data center server resources (CPU, MEM and HDD). To test the power saving capabilities of our swarm-inspired methodology we have used the same workload (see Figure 3) on the simulated data center for two test cases scenarios and compared the power savings results.

Figure 3. The test case generated workload intensity

In the first scenario, the simulated data center is organized and managed according to our swarm-inspired methodology while in the second one, the data center is managed by an OpenNebula-like management algorithm. OpenNebula [12] is the state of the art middleware for managing virtualized data center clouds and its server consolidation algorithm is based on a Fit-First approach.

Using our swarm-inspired methodology the number of servers that are used to accommodate the workload tasks varies throughout the simulation. If the size of the workload increases over the total capacity of the V-formation leading cluster, then the intelligent agents collaborate to adjust the number of servers in the cluster in order to be able to receive all of the workload virtual tasks. If a number of servers from the $V_{leading\text{-}cluster}$ do not execute any workload after deployment, the attached agents decide to shut down these servers to conserve energy. On the other hand, the OpenNebula Fit-First server consolidation algorithm over provisions computing resources to handle peak load values. In this case there will always be servers in the data center that are powered-on without executing workload, and the deployment of the virtual tasks is done on the principle of choosing the first fit. This means that when a new virtual task needs to be deployed, it will be deployed on the first found server that can accept it.

The difference between the two consolidation approaches, highlighted above is reflected by calculating the simulated data center Deployed Hardware Utilization Ratio (DH-UR) [13] when executing the same workload. The DH-UR is a Green Performance Indicator which measures the number of IT computing resources (servers) that consume power without doing any actual work (see relation 10).

$$DH-UR = Number\ of\ Servers\ running\ application\ /\ Number\ of\ Servers\ up\ and\ running \quad (10)$$

The value of this metric for a Green data center should be close to 1, meaning that the servers that are up and running are actually processing some tasks.

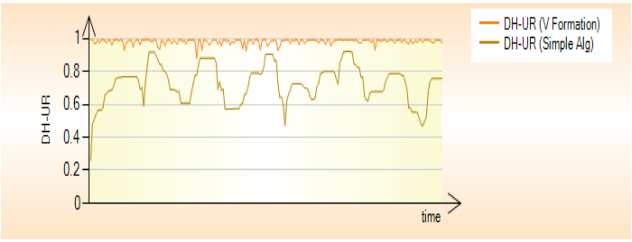

Figure 4. DH-UR metric evolution for the two test cases

Figure 4 presents the calculated DH-UR value for the two described test cases. It can be noticed that the DH-UR metric for the configuration based on our swarm-inspired algorithm is close to 1 all the time, while the same metric when running the fit first simple algorithm barely jumps over 0.8 (which is the case of today real data centers). The average DH-UR values are shown in Table 1.

Table 1. Average values for DH-UR metric

Management Algorithm	Average DH-UR Value
Swarm-inspired	0.98
OpenNebula Fit First	0.73

To estimate the data center power efficiency we have *measured using a power meter* the power consumption of the test case real data center server (its hardware configuration was presented in the beginning of the section) 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 MI2392 – Power Q Plus [14] power meter and the obtained results are listed in Table 2.

Table 2. Data center server states measured power consumption

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

The whole data center power consumption was estimated by defining and using the following metric:

$$PW(SC) = NoIdleServers * PowerIdle + NoWorkingServers * PowerWorking + NoFullLoadServers * PowerFullLoad + PW_{overhead} \quad (11)$$

where $NoIdleServers$ represents the number of data center servers that are in Idle Power Mode, $NoWorkingServers$ represents the number of servers that are in Working Power Mode, $NoFullLoadServers$ represents 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 using the same power meter* the power consumption of each management action defined by our methodology. The following types of management actions are measured: (1) deploy workload virtual task on a $V_{leading\text{-}cluster}$ server, (2) migrate a task between two servers from the $V_{leading\text{-}cluster}$, (3) hibernate server when completing its workload execution and (4) wake up a server to accommodate incoming workload. The actions measured power consumption overhead is listed in Table 3. The servers' movement actions between different V-formation clusters are logical actions and do not consume any supplementary power.

Table 3. Swarm-inspired methodology actions measured power consumption overhead

Management Action	Power Consumption
Deploy	10W
Migrate	20W
Hibernate Server	100W
Wake up Server	100W

The overall power consumption overhead induced by our swarm-inspired methodology is estimated as follows:

$$PW_{overhead} = NoTasksDeployed * PowerDeploy + NoTasksMigrated * PowerMigrate + NoServerTurnedOn * PowerWakeUp + NoServerTurnedOff * PowerTurnOff \quad (12)$$

where $NoTasksDeployed$ represents the number of workload tasks that were deployed, $NoTasksMigrated$ is the number of migrated workload tasks, $NoServerTurnedOn$ is the number of servers that were turned on, while $NoServerTurnedOff$ represents the number of turned off servers.

Figure 5 shows the estimated power consumption of the simulated data center when accommodating the same workload, in both test case scenarios.

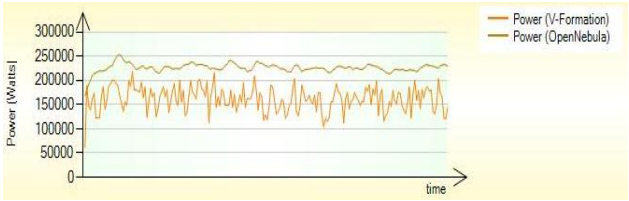

Figure 5. Power consumption evolution for the two test cases

The power consumption difference is due to the fact that when running the OpenNebula Fit-First consolidation algorithm, there is a high number of idle servers that are not executing tasks but consume 70 Watts each. When using our swarm-inspired approach, only the servers that are executing workload ($V_{leading-cluster}$ and $V_{wing-first-cluster}$) are up consuming power. The number of $V_{wing-first-cluster}$ idle servers is lower and a higher number of servers are kept in turned off state in the $V_{wing-third-cluster}$.

The average values for the power consumption in the two configurations can be found in Table 4. The average power saving is around 40%. Even though this percentage is really high it should be noted that the data center IT facility aspect was neglected during simulation. In a data center the IT facilities usually consume about 60% from the total power consumption [15]. This means that our solution's power saving of 40% represents about 16% from the total power consumption of a data center.

Table 4. Average power consumption

Management Algorithm	Average Power Consumption
Swarm-inspired	161670W
OpenNebula Fit-First	226322W

5. CONCLUSIONS

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

6. ACKNOWLEDGMENTS

This work has been partially supported by the GAMES project [16] and has been partly funded by the European Commission IST activity of the 7th Framework Program (contract number ICT-248514). This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work.

7. REFERENCES

[1] Green IT, Energy efficiency: the silent killer & enabler - data centres, semis & the smart grid. 2009. *Societe Generale Report*. Available online at: http://www.samsung.com/global/business/semiconductor/products/dram/DDR3/swf/download_file/green_it/SGCrossAssetResearch_Green_IT.pdf.

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

[3] Jerger, N. E., Vantrease, D. and Lipasti, M. 2007. An Evaluation of Server Consolidation Workloads for Multi-Core Designs. *Proceedings of the IEEE International Symposium on Workload Characterization*. DOI = http://dx.doi.org/10.1109/IISWC.2007.4362180

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

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

[6] Marzolla, M., Babaoglu, O., Panzieri, F. 2011. Server Consolidation in Clouds through Gossiping. *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 1-6*. DOI= http://dx.doi.org/10.1109/WoWMoM.2011.5986483

[7] Verdaasdonk, B. W., Koopman, H., van der Helm, F. 2009. Energy efficient walking with central pattern generators: from passive dynamic walking to biologically inspired control, *Biological Cybernetics*, Springer-Verlag, Volume 101, Issue 1, pp. 49-61. DOI = http://dx.doi.org/10.1007/s00422-009-0316-7

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

[9] Champrasert, P., Suzuki, J. 2006. 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*. DOI = http://dx.doi.org/10.1109/WOWMOM.2006.105

[10] Barbagallo, D., Di Nitto, E., Dubois, D. J., Mirandola, R. 2010. A Bio-Inspired Algorithm for Energy Optimization in a Self-organizing Data Center, *First International*

Conference on Self-Organizing Architectures, Revised Selected and Invited Papers, LNCS, Volume 6090, pp. 127-151.

[11] Lee, C., Wada, H., Suzuki, J. 2007. Towards a Biologically-inspired Architecture for Self-regulatory and Evolvable Network Applications, *Advances in Biologically Inspired Information Systems: Models, Methods and Tools*, Chapter 2, pp. 21 - 45, Springer.

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

[13] Stanley, J., Brill, K. and Koomey, J. 2009. Four Metrics Define Data Center Greenness, *Uptime Institute, white paper*, Available online at: http://uptimeinstitute.org.

[14] MI 2392 PowerQPlus Power Meter, http://www.metrel.si/products/power-quality-analysis/mi-2392-powerq-plus.html.

[15] Petschke, B. 2008. State of the Art Energy Efficient Data Centre Air Conditioning. *Stulz GmbH, white paper*, Available online at: www.stulz.com.

[16] GAMES FP7 Research Project, http://www.green-datacenters.eu/.

[17] Jade, Java Agent DEvelopment Framework, http://jade.tilab.com.

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

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

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

[21] Anderson, R. J.,Mayr, E. andWarmuth, M., 1988. Parallel Approximation Algorithms for Bin Packing, *Stanford University*.

[22] Torres, J., Carrera, D. et al., 2008. Tailoring Resources: The Energy Efficient Consolidation Strategy Goes Beyond Virtualization, *International Conference on Autonomic Computing*, pp. 197 - 198, ISBN: 978-0-7695-3175-5. DOI = http://dx.doi.org/10.1109/ICAC.2008.11

[23] Berral, J., Goiri, I., Nou, R., et al., 2010. Towards energy-aware scheduling in data centers using machine learning, *Int'l Conf. on Energy-Efficient Computing and Networking (e-Energy 2010)*. DOI = http://dx.doi.org/10.1145/1791314.1791349

[24] L. Wang, G. Laszewskiy, J. Dayaly, et al., 2009. Towards Thermal Aware Workload Scheduling in a Data Center, *In Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pp. 116-122, ISBN: 978-0-7695-3908-9. DOI = http://dx.doi.org/10.1109/I-SPAN.2009.22

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

[26] B. Speitkamp and M. Bichler, 2010. A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers, *IEEE Transactions on services computing*, Vol. 3 (4). DOI = http://dx.doi.org/10.1109/TSC.2010.25

[27] Tesauro, G., Jong, N. K., Das, R., Bennani, M. N., 2007. On the use of hybrid reinforcement learning for autonomic resource allocation, *Cluster Computing* 10(3): 287-299. DOI = http://dx.doi.org/10.1007/s10586-007-0035-6

[28] Kephart, J. O., Chan, H., Das, R., Levine, D. W. et al., 2007. Coordinating Multiple Autonomic Managers to Achieve Specified Power-Performance Tradeoffs, *International Conference on Autonomic Computing*. DOI = http://dx.doi.org/10.1109/ICAC.2007.12

[29] Cioara, T., Anghel, I., Salomie, I., Copil, G., Moldovan, D. and Pernici, B., 2011. A context aware self-adapting algorithm for managing the energy efficiency of IT service centres, *Ubiquitous Computing and Communication Journal*, Volume 6 No. 1, ISSN Online 1992-8424.