A Cloud Platform for Classification and Resource Management of Complex Electromagnetic Problems

Andreas Kapsalis[1], Panagiotis Kasnesis[1], Panagiotis C. Theofanopoulos[2], Panagiotis K. Gkonis[1], Christos S. Lavranos[2], Dimitra I. Kaklamani[1], Iakovos S. Venieris[1] and George A. Kyriacou[2]
[1]*Intelligent Communications and Broadband Networks Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece*
[2]*Democritus University of Thrace, Department of Electrical & Computer Engineering, Xanthi, Greece*

Keywords: Cloud Computing, Machine Learning, Resource Allocation, SVM, Ant Colony Optimization, Eigenanalysis, Finite Difference.

Abstract: Most scientific applications tend to have a very resource demanding nature and the simulation of such scientific problems often requires a prohibitive amount of time to complete. Distributed computing offers a solution by segmenting the application into smaller processes and allocating them to a cluster of workers. This model was widely followed by Grid Computing. However, Cloud Computing emerges as a strong alternative by offering reliable solutions for resource demanding applications and workflows that are of scientific nature. In this paper we propose a Cloud Platform that supports the simulation of complex electromagnetic problems and incorporates classification (SVM) and resource allocation (Ant Colony Optimization) methods for the effective management of these simulations.

1 INTRODUCTION

The simulation of complex electromagnetic problems was always a task that required both great amount of time and resources. Even with the advancement of conventional computer technology, the execution of complex algorithms such as Monte-Carlo simulations -especially when dealing with realistic scenarios (e.g. large amount of users) - creates a resource demanding process and even if the host environment has an adequate amount of resources to successfully execute it, it usually takes a prohibitive amount of time for the user to receive the results of his or her simulation. Of course modern programming languages (even the high level ones) nowadays provide sophisticated libraries that allow for parallel programming paradigms that take advantage of multicore computers. What is more, the solution of distributed computing allows a cluster of computers in a private network to share the load by executing different parts of the application code.

Parallel and distributed computing has been a solution for complex scientific simulations. Public grid computing platforms offer to user communities from various scientific domains the opportunity to submit large resource-demanding simulations that promise to yield the desired results in an acceptable amount of time. However, grid computing platforms impose certain restrictions to users. That happens mainly because simulations are submitted directly to the physical machines for execution, and as a result the user code must be compatible with the host operating system. Physical machines might be in the situation of hosting processes for different simulations from different users. This creates a form of a single point of failure where an error (e.g. memory leak) might jeopardize the rest of executing processes.

On the other hand over the last few years Cloud Computing has been a rapidly emerging alternative, as it can also provide a very solid solution when dealing with scientific applications. Cloud Platforms that utilize Cloud Infrastructures (IaaS tier) can easily allocate a cluster of Virtual Machines for the execution of simulations. Each Virtual Machine or cluster of Virtual Machines is dedicated to the execution of a specific user simulation at each given time. Moreover, because Virtual Machines are created by using images or snapshots of operating systems, the user has the opportunity to request an

Kapsalis, A., Kasnesis, P., Theofanopoulos, P., Gkonis, P., Lavranos, C., Kaklamani, D., Venieris, I. and Kyriacou, G..
A Cloud Platform for Classification and Resource Management of Complex Electromagnetic Problems.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 388-393
ISBN: 978-989-758-158-8

environment specifically "tailored" for his needs; he doesn't have to change his code in that case.

Having the aforementioned –as well as many other significant advantages of Cloud Computing- in mind, in this paper we propose a platform that utilizes Cloud Infrastructures in order to provide a solution for complex resource demanding electromagnetic simulation problems. The platform accepts user simulation requests, decides the amount of resources that will be granted for the execution of the simulation application, creates the requested Virtual Machines, collects monitoring data and stores results in a persistent storage space. Two crucial features of the platform will be discussed in detail in later sections; the Machine Learning process which automatically figures the resource demand of a given simulation, and the process that allocates the newly-created Virtual Machines to hosts, which is targeted towards performance and energy efficiency. The rest of the paper is organized as follows; in Section 2 we present some notable related work, and in Section 3 we present the platform architecture. In Sections 4 and 5 we describe the Machine Learning and VM Allocation strategies that are adopted in the platform. In Section 6 we describe the nature of the Electromagnetic Problems and finally in Section 7 we provide some useful conclusions and ideas for future work.

2 RELATED WORK

Autonomous Virtual Machine allocation is quite a new aspect. The idea of applying Machine Learning techniques in VM allocation over the cloud is investigated the last five years. This approach has the advantage of adapting the cloud services dynamically according to the Service Layer Agreement (SLA) between the client (user) and the provider (cloud).

M. Macias et al. (2011 and 2012) classify the clients' policies for SLA negotiation and allocation. The clients classification is based on the client's affinity with the provider and the Quality of Service (QoS) the client are willing to acquire. Afterwards the provider applies Machine Learning techniques (G. Reig, 2010) to predict future jobs and confirm if the offered job can be executed. J.Rao et al. (2009) propose VCONF, a RL approach to automate the VM configuration process. VCONF is based on model-based RL algorithms that generate policies learned from iterations with the environment. X. Dutreilh et al. (2011) propose a more sophisticated model free RL approach using appropriate initialization of the weights for the early stages and convergence speedups applied in order to avoid the slow convergence and optimal policy discovery in the early phases of learning. L. Chimakurthi et al. (2011) introduce an energy efficient mechanism capable of allocating the cloud resources according to the given SLA using the Ant Colony algorithm.

Another approach (A. Quiroz, 2009) uses decentralized online clustering to categorize the incoming tasks and optimize the provisioning of the resources. They also present a model-based approach to calculate the application service time given the derived provisioning in order to tackle the inaccurate resource requirements. A rule-based approach is followed by M. Maurer et al. (2012) in an attempt to reduce the SLA violations by monitoring the workload itself. They introduce the notion of workload volatility (WV), determine a function to calculate it and dynamically classify workload into WV classes (LOW, MEDIUM, MEDIUM HIGH and HIGH WV). C.J. Huang et al. (2013) propose a system that optimizes the resource allocation in cloud computing. The two main components of their system is the application service prediction module that uses Support Vector Regression (SVR) to calculate the response time and the resource allocation optimization module that is based on Genetic Algorithm (GA) and redistributes the resources according to the current status of the installed VMs.

3 PLATFORM ARCHITECTURE

The architecture of the proposed platform is based on a multi-agent system. For each simulation request that is received by the platform a group of agents is created that handles the lifecycle of the simulation until its completion. The platform entry point is the Supervisor service that creates the aforementioned set of agents. There are four agents (Profiler, Planner, Client, and Monitor) that perform a distinct set of tasks during the simulation lifecycle. The system is event or message driven; each agent performs a task when it receives a message from a peer or supervising entity. Below we describe in detail the tasks that each entity of the platform is responsible for. An overview of the platform architecture is shown in fig. 1.

3.1 Supervisor

The Supervisor service is responsible for receiving simulation requests by the users. It performs some checks according to the user SLA and once it accepts the request, the Supervisor service communicates with the underlying agent system and requests for the

creation of four agents that will be dedicated to carry out the execution of the user simulation.

3.2 Profiler

The Profiler is responsible on making decisions based on historical data and user SLAs about the amount of resource that are to be allocated into a VM or VMs. The Profiler incorporates the SVM (Support Vector Machine) supervised machine learning method. The algorithm predicts the appropriate amount of resources that are necessary by the simulation in order to be executed in the required amount of time according to the user SLA and the infrastructure characteristics.

3.3 Planner

Another integral part of the platform is the agent that receives the decision output from the Profiles and is responsible for creating an allocation plan for the resources. For this the Planner agent uses an Ant Colony Optimization algorithm variant that is targeted towards energy efficiency. The ACO variant is described in detail in Section 5.

3.4 Client

The Client agent receives the allocation plan and its sole responsibility is to communicate with the service stack of the IaaS (Infrastructure as a Service) in order to create or destroy the requested VMs into the pre-selected virtual hosts.

3.5 Monitor

Once the requested resources are created and are accessible, the Monitor agent performs periodic checks on the VMs that host the user's simulation. The Monitor collects data regarding the resource utilization of each VM as well as the resource usage of the simulation process of each VM.

3.6 Datastore

The datastore may not be a part of the agent system, but it is a highly important part of the platform as it holds monitoring data that are used for profiling of user simulations. The datastore uses a NoSQL document database. Due to the unstructured and loosely coupled form of stored data NoSQL databases provide better write/read times especially when clients query for large chunks of data.

3.7 User Dashboard

The user dashboard is a Web UI that allows the user initiate new simulations as well as review his or hers on-going or completed ones. It displays status notifications on screen in real time fashion and provides also information logs regarding the simulations that have complete successfully or unsuccessfully. The user can also download the results of the simulation into his computer.

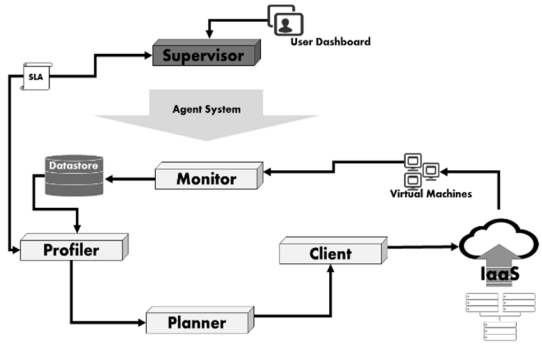


Figure 1: Overview of the Platform Architecture.

4 PROFILER

In our approach we attempt to categorize the aforementioned computational tasks (Section 6) requested by the users into classes. Each computational task will be mapped to the generation of a finite number of VMs with the proper attributes (CPU and Memory) to handle the user's request. As a result we propose a supervised machine learning classification approach to achieve the autonomous VM allocation in a cloud computing architecture. The most appropriate method for supervised classification for this is the Support Vector Machine (SVM) using the Gaussian Kernel. For a number of training examples m, the regularized SVM cost function ($J(\theta)$) of a data set $D = \{(x^{(i)}, y^{(i)}),\ i \in m\}$ is given by:

$$J(\theta) = C\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(x^{(i)}) + \left(1-y^{(i)}\right)\log h_\theta(x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2 \quad (1)$$

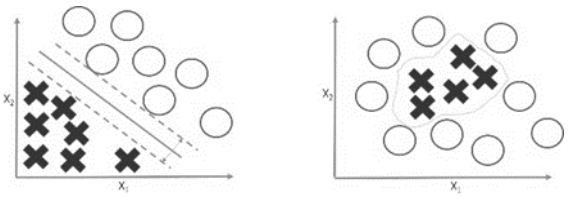
$$h_\theta(x^{(i)}) = \frac{1}{1+e^{-\theta^T x^{(i)}}} \quad (2)$$

where n denotes the number of features, θ denotes the weight of a feature, $h_\theta(x)$ depicts the hypothesis value that equals the sigmoid (logistic) function and C depicts the regularization term. A huge advantage

of SVM is that it finds a hypothesis *h* that assures the lowest true error, Structural Risk Minimization principle (V. N. Vapnik, 1995). The Quality of Experience (QoE) is a major issue for the users and SVM provides a more "safe" distance (fig. 2) from the decision boundary for each example, which reduces the classification errors that other classifiers with "soft" margin produce. What is more, the Gaussian Kernel the ability to generate non-linear decision boundaries (fig. 3) using linear classification methods and Gaussian kernel function enables the classification of data that have no obvious fixed-dimensional vector space representation (A. Ben-Hur, 2010). The Gaussian Kernel function on two samples (*x*, *x'*) is given by:

$$K(x, x') = exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \quad (3)$$

where $\|x - x'\|^2$ denotes the squared Euclidean distance between two feature vectors and σ^2 depicts the variance.



a) b)

Figure 2: a) Margin Classifier b) Non-linear decision boundary.

5 PLANNER

The proposed platform uses the output of the Machine Learning Classification process and performs the next crucial task which is the resource allocation (in the form of VMs) into physical hosts. The goal of the Planner agent is to pack VMs that are created for a specific simulation into the same host. That is profitable for two reasons; the first is that the simulation will experience lower network overhead especially when popular distributed computing frameworks (e.g. MPI, MapReduce) use ssh as their main communication protocol. The second reason is that the allocation should be also driven towards energy efficiency regarding the physical infrastructure, as inactive hosts can be put into energy saving mode. Given the fact that the allocation problem is of NP-Hard nature, we have chosen to use the meta-heuristic Ant-Colony Optimization algorithm (ACO). The ACO algorithm models a multi-agent system that imitates the behavior of real life ant colonies when searching for food sources. It was originally developed by Marco Dorigo et al. (1999). The ACO algorithm shows better results towards workload performance and energy efficiency than other greedy algorithms such as the First-Fit Decreasing (FFD) (Feller et al., 2011). Our approach regarding the implementation of the algorithm takes the following assumptions into account; a) the resource demand by the Virtual Machines is static and b) the resource vector contains the CPU cores and ram in GB. A full formalization and description of the ACO algorithm is provided in the work by Feller et al. (2011).

In the proposed variant let B denote the set of physical hosts with size n, and I the set of VMs with size m that the algorithm will try to allocate into hosts. Also, let $\overrightarrow{C_v}$ denote the resource capacity vector (2-dimensional in our case) of a physical host $v, v \in B$. Similarly, let $\vec{r_i}$ denote the resource demand vector of a VM $i, i \in I$. As mentioned earlier this ACO variant is targeted towards energy efficiency by trying to pack as many VMs to the same physical host. Thus, the objective function the algorithm tries to minimize is given by the following:

$$f(y) = \sum_{v=0}^{n-1} y_v \quad (4)$$

Where y_v is equal 1 if a physical host has allocated VMs, and 0 otherwise. The objective function is subject to the following constraints; a) each VM i is allocated to exactly one physical host and b) the maximum resource capacity for each of the physical hosts is not exceeded. An integral characteristic of the ACO algorithm is the way each ant-agent is making decisions on which physical host will allocate a VM into. The probability to choose the i, v pair is a combination of the pheromone trail, which is a mechanism for encouraging other ants to follow the same path, and a heuristic information. The probability is given by the following:

$$p_{i,v} = \frac{[\tau_{i,v}]^a \times [v_{i,v}]^\beta}{\sum_{u \in N_u}[\tau_{u,v}]^a \times [v_{u,v}]^\beta} \quad (5)$$

Where $\tau_{i,v}$ is the amount of pheromone trail for the specific VM - physical host pair, $v_{i,v}$ is the heuristic information for that same pair and N_u is the set of all VMs that are not assigned into any host and ensure that the resource capacity constraint of the physical host is not violated when taking also into account the current load assigned to it. The above can be show in eq. 6:

$$N_v := i \quad (6)$$

$$s.t$$

$$\sum_{j=0}^{n-1} x_{i,j} = 0 \quad (7)$$

$$and \ \overrightarrow{b_v} + \vec{r_i} \leq \overrightarrow{C_v} \quad (8)$$

Where $\overrightarrow{b_v}$ denotes the current resource load for physical host v. Another characteristic that is worth mentioning is the pheromone evaporation mechanism which is triggered after the end of each iteration of the algorithm and is given by the following:

$$\tau_{i,v} \coloneqq (1-p) \times \tau_{i,v} + \Delta_{\tau_{i,v}}^{best} \quad (9)$$

Where p is the pheromone evaporation rate and $\Delta_{\tau_{i,v}}^{best}$ is a bonus to the best VM – physical host pair of each iteration.

6 ELECTROMAGNETIC PROBLEMS

6.1 Principal Component Analysis in MIMO-WCDMA Networks

The first electromagnetic problem performs the calculation of the appropriate transmission vectors in MIMO-WCDMA networks that improve Signal to Noise Ratio (SNR), where diversity combining transmission mode is assumed. In order to reduce overall complexity, Principal Component Analysis (PCA) is employed at the reception, and only the terms that contribute to a predefined signal energy are taken into account. Even so, however, in cases of multiuser/multipath scenarios, computational load can be significantly increased.

In particular, when deploying PCA in MIMO-WCDMA networks, it is assumed that all received data for a specific user after processing can be stacked as an $m \times S$ matrix (denoted as X throughout the rest of this section), where m is the number of independent observations and S the number of samples per observation (Shlens, 2005). In our case,$m = L$ where is the number of multipath components. The covariance matrix of X will be given by:

$$C_X = \frac{1}{S} X X^H \quad (10)$$

and the primary eigenvalues and eigenvectors can be directly calculated from eigenvalue analysis of matrix C_X. Afterwards, an iterative optimization approach is followed, and all transmit weight vectors can be calculated (Gkonis et al., 2015). Note however that as the number of active users increases, eigenvalue analysis should be performed at each user separately. Moreover, the dimensions of matrix X are directly related to the number of active multipath components, while processing complexity also depends on the number of transmit and receive antennas.

6.2 Eigenanalysis based on a 2-D FDFD Method

The second case of electromagnetic problems is the eigenanalysis of open-periodic electromagnetic structures. The eigenanalysis of electromagnetic structures is used to enlighten all the hidden physical properties exhibited by each structure. The following eigenanalysis is exploiting a 2-D Curvilinear FDFD method, (Lavranos et al., 2009, Theofanopoulos et al. 2014 and Lavranos et al., 2014).

The eigenanalysis starts from Maxwell's curl equations in the frequency domain and after some algebraic manipulations the following linear eigen problems occur:

$$\begin{pmatrix} A_{ee} & A_{eh} \\ A_{he} & A_{hh} \end{pmatrix} \begin{pmatrix} E_t \\ H_t \end{pmatrix} = j\beta \begin{pmatrix} E_t \\ H_t \end{pmatrix} \quad (11)$$

$$(C_m)(F)(C_e)(N) \begin{bmatrix} D_t \\ D_3 \end{bmatrix} = \omega^2 \begin{bmatrix} D_t \\ D_3 \end{bmatrix} \quad (12)$$

The eigenproblem given in eq. 11 is called 'β-formulation' and the independent variable is the eigenfrequency ω, while the eigenvalue is the complex propagation constant β. On the other hand, the eigenproblem of eq. 12 is called 'ω-formulation' and the independent variable is the propagation constant β and the eigenvalue is the eigenfrequency ω. So, in order to reveal all the modes exhibited by the structure, an independent eigenproblem has to be solved for every ω or β in a given range. The size of the eigenproblems depends on the level of meshing detail.

The nature of the eigenproblems reveals the urgency towards a form of parallelization or distributed execution. The proposed parallelization technique relies on the independent nature of each eigenproblem. So, exploiting the independence of the eigenproblems, an assignment to different machines is expected. The estimated analysis time is reduced depending on the number of the machines-workers available.

7 CONCLUSIONS AND FUTURE WORK

In this paper we described a platform that takes

advantage of Cloud Computing Infrastructures in order to apply classification and resource management methods for complex and resource heavy electromagnetic simulations. The proposed platform requires no special configuration from the user regarding his or her application code. Also, depending on the selected simulation the platform can predict the required resource demand in order to allow it to complete in the desired user time, according of course the SLA. Also, the platform uses an energy-efficient ACO variant for the allocation of the VMs into physical hosts that can also achieve low network overhead.

A series of next steps include the collection of test results against the proposed platform and the simulation of the described electromagnetic problems in various set ups. Furthermore, we would like to address the issue of dynamically reconfiguring the allocation plan according to the continuous changing resource demand of the electromagnetic simulations. Finally, we intend also to test a number of allocation algorithms and machine learning methods against the ones that are already being used in our platform.

ACKNOWLEDGEMENTS

This research has been co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: **THALES**. Investing in knowledge society through the European Social Fund.

REFERENCES

Shlens, J., (2005, December). A tutorial on principal component analysis.

Gkonis, P., K., Kapsalis, A., Zekios, K., Kaklamani, D., I., Venieris, I., S., Chrysomallis, M., Kyriakou, G., (2015, April). On the Performance of Spatial Multiplexing in MIMO-WCDMA Networks with Principal Component Analysis at the Reception. *CD-ROM in Procs. EuCAP 2015*, Lisbon, Portugal.

Lavranos, C., S., Kyriacou, G., A., (2009, March). Eigenvalue analysis of curved waveguides employing an orthogonal curvilinear frequency domain finite difference method, *IEEE Microwave Theory and Techniques*.

Theofanopoulos, P., C., Lavranos, C., S., Sahalos, J., N. and Kyriacou, G., A., (2014, April). Backward Wave Eigenanalysis of a Tuneable Two-Dimensional Array of Wires Covered with Magnetized Ferrite *in Proc. EuCAP 2014,* The Hague, The Netherlands.

Lavranos, C., S., Theofanopoulos, P., C., Vafiades, E., Sahalos, J., N. and Kyriacou, G., A., (2014, November). Eigenanalysis of Tuneable Two-Dimensional Array of Wires or Strips Embedded in Magnetized Ferrite, *in the Proc. LAPC 2014*, Loughborough, UK.

Dorigo, M., Di Caro, G., & Gambardella, L. (1999, April). Ant Algorithms for discrete optimization. *Artificial Life*, 137-172.

Feller, E., Rilling, L., & Morin, C. (2011). Energy-Aware Ant Colony Based Workload Placement in Clouds. *[Research Report] RR-7622*.

Macias, M., Guitart, J., (2012). Client Classification Policies for SLA Enforcement in Shared Cloud Datacenters. *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*.

Macias, M., Guitart, J., (2011, December). Client Classification Policies for SLA Negotiation and Allocation in Shared Cloud Datacenters, *in Proc. GECON 2011.*

Reig, G., Alonso, J., and Guitart, J., (2010, July). Prediction of job resource requirements for deadline schedulers to manage high-level SLAs on the cloud, *in 9th IEEE Intl. Symp. on Network Computing and Applications*, Cambridge, MA, USA, 162–167.

Rao, J., Bu, X., Xu, C., Wang, L., Yin, G., (2009). VCONF: A Reinforcement Learning Approach to Virtual Machines Auto-configuration, *in Proc. ICAC '09*.

Dutreilh, X., Kirgizov, S., Melekhova, O., Malenfant, J., Rivierre, N., and Truck, I., (2011). Using Reinforcement Learning for Autonomic Resource Allocation in Clouds: Towards a Fully Automated Workflow, *Seventh International Conference on Autonomic and Autonomous Systems*.

Chimakurthi, L., and Kumar S D, M., (2011). Power Efficient Resource Allocation for Clouds Using Ant Colony Framework, *International Conference on Computer, Communication and Electrical Technology*.

Quiroz, A., Kim, H., Parashar, M., Gnanasambandam, N., and Sharma, N., (2009). Towards Autonomic Workload Provisioning for Enterprise Grids and Clouds, *Proc. IEEE/ACM 10th Int',l Conf. Grid Computing*, 50-57.

Maurer, M., Brandic, I., and Sakellariou, R., (2012). Self-adaptive and resource efficient SLA enactment for cloud computing infrastructures, *5th International Conference on Cloud Computing (CLOUD).*

Huang, C., Wang, Y., Guan, C., Chen, H., and Jian, J., (2013). Applications of Machine Learning to Resource Management in Cloud Computing, *International Journal of Modeling and Optimization,* 2, 148-152.

Vapnik, V., N., (1995). The Nature of Statistical Learning Theory. Springer, New York.

Ben-Hur, A., and Weston, J., (2010). A User's Guide to Support Vector Machines, *Data Mining Techniques for the Life Sciences*, 609, 223-239.