Energy Efficient Data Centres Within Smart Cities: IaaS and PaaS Optimizations

Corentin Dupont, Mehdi Sheikhalishahi(✉), Federico M. Facca, and Silvio Cretti

Create-Net, Trento, Italy
{cdupont,msheikhalishahi,ffacca,scretti}@create-net.org

Abstract. Data centres are power-hungry facilities that host ICT services and consume huge amount of the global electricity production. Consequently, in the last years, research trends in the field focused on mechanisms able to reduce the overall consumption of a data centre so as to reduce its energy footprint. In this paper, we argue that the data centres for city needs should be located physically in the smart city, in order to address smart city needs, and serve citizens without any latency. Furthermore, those data centres should strive to make their energy footprint greener, i.e. consume more renewable energy. We present the concept of Service Flexibility Agreement (SFA), an extension of the traditional SLA able to qualify the flexibility of applications deployed in a smart city cloud environment. In particular, we detail preliminary models able to exploit this flexibility in order to increase the ratio of renewable energy consumed. The introduction of new solutions, such as containers and platform as a service (PaaS) in cloud data-centres also opens new challenges and opportunities. We describe how the combination of PaaS and IaaS cloud layers provide the needed flexibility to support the SFA.

Keywords: Platform as a Service · Energy management · Flexibility · Scalability

1 Introduction

Data centers are one of the largest and fastest growing consumers of electricity in the United States. In 2013, U.S. data centres consumed an estimated 91 billion kilowatt-hours of electricity – enough electricity to power all the households in New York City twice over – and are on-track to reach 140 billion kilowatt-hours by 2020[1]. However, a recent study [1] showed that, while still growing, the current energy consumption of data centres (DCs) is less than previously excepted. Electricity used in US DCs in 2010 was significantly lower than predicted by the EPAs 2007 report to Congress on DCs. There is a combination of factors explaining this slow down, among which the application of new energy policies in DCs. For example, consolidation techniques to reduce the power of servers in a

[1] http://www.nrdc.org/energy/data-center-efficiency-assessment.asp.

A. Leon-Garcia et al. (Eds.): Smart City 2015, LNICST 166, pp. 408–415, 2016.
DOI: 10.1007/978-3-319-33681-7_34

DC are nowadays adopted in several cloud management solutions. As important energy consumers, it is also important that the energy management and policies of DCs prioritize the consumption of renewable energies over brown energies. However, the main problem with the utilization of renewable energies is that they are very variable in time. To adapt to such energies, we need to adapt and shift the workload of applications. This means reducing the workload when there is less renewable energies available, for example.

Beyond that, the technological landscape is changing. DCs can now host more than simple virtual machines. New "virtualization" techniques such as containers are appearing on the scene, and Platform-as-a-Service solutions are more and more used on top of Infrastructure-as-a-Service solutions. PaaS management frameworks model the architecture of applications and provide management functions to scale up and down multi-tier applications. Some frameworks allow to automatize this process: Cloudify[2], for example, provides a language for auto-scaling. This language defines Key Performance Indicators (KPIs) and thresholds that will trigger the scaling operations. For example, in the case of a 3-tier Web server application, it is possible to describe that if the latency in serving the pages goes over a certain threshold, a new front-end VM should be launched. As such, the "intelligence" of PaaS management frameworks can be easily employed to apply energy management policies taking into account application SLAs and their architecture. Following this reasoning, we believe that the combined management of PaaS and IaaS may bring new opportunities in energy policies management.

However, when it comes to the adaptation of applications workload to dynamic power budgets, we think that a piece is missing. Indeed, currently PaaS frameworks have no way to lower or postpone workload in a reasonable way when there is no renewable energy or energy is too expensive, for instance. PaaS models can be enhanced to better describe the flexibility of applications and allow to perform optimizations at smart city DC level, such as increasing the renewable energy usage.

Thus, in this paper we propose the concept of Service Flexibility Agreement (SFA). The SFA is an extended Service Level Agreement (SLA): it includes a description of the flexibility of an application. While the SLA usually defines only minimum levels of resources that an application should be guaranteed to have, in the context of flexible applications, this is not enough: for example some applications can accept to have a temporarily reduced performance or shifted activities. Similarly, some applications would benefit from a temporary increase of allocation of resources when renewable energies are available. The SFA defines a simple interface to describe this flexibility in terms of resource allocated over time, with possible deviations. Thanks to the SFA, an energy-aware PaaS framework is able to dynamically reconfigure applications or single layers (e.g. scale-up and down) to comply with a given energy budget (e.g. the amount of green energy available at a given moment in time). Finally, changes occurring at the level of the PaaS framework can be exploited by an underlying IaaS framework: the information sharing between the two improves the energy usage.

[2] http://www.gigaspaces.com/cloudify-cloud-orchestration/overview.

With the aim of addressing climate change, increasing the EUs energy security and strengthening its competitiveness, the EU has created the set of targets known as 20–20–20 for its climate and energy policies. Due to the trend of further urbanization, Smart Cities are one of the key enablers of the 2020 targets. It is expected that the urbanization concentrates the energy consumption in the city and Smart Cities are one way to decrease and optimize this consumption and therefore increase the energy efficiency. Therefore it will need new processes and innovation concepts at city level.

For some years there was a tendency to move DCs outside the cities, to locations where space is not an issue and the electricity infrastructure is not challenged by other large consumers. Furthermore, the selected locations had a cold climate (i.e. Finland, Iceland), in order to save on the cooling expenses. However, lately we see the trend of DCs moving back into the vicinity of cities because of issues like network latency and due to the increased density of racks and modularity of whole DC sites. DCs that are separate departments inside a company can be integrated into the smart city energy management. This allows a better control for the Smart City in terms of management, energy consumption, renewables, data security/privacy and governance.

This data is more and more processed in the Cloud. So DCs located in the city near the base station means more responsiveness and less losses. Similarly, the concept of "cloud edge" or mini-cloud located near a base station or a sensor network in a smart city where a sensor network is deployed can benefit from this approach.

2 Service Flexibility Agreement

In current PaaS architectures, the framework grants a certain flexibility to applications. For example, an application can ask the PaaS framework to spawn more or less VMs or Linux containers[3] according to its needs. However, the flexibility is entirely controlled by the application and/or the application owner. The intuition behind SFA is to delegate some of the flexibility control to the PaaS framework while still guaranteeing the end user satisfaction. With respect to a traditional SLA, the SFA adds a few new dimensions: the possibility for the required resources to vary in time, plus the possibility to qualify violations of the required performance.

As shown in Listing 1 and also represented graphically in Fig. 1, for each time frame of a day the SFA defines a recommended business performance (RecoBP, in red in the figure). The business performance is one of the KPIs of the application. For example, for a Web server it is the number of pages served per minutes, for a video transcoding service it will be how many videos can be transcoded per minutes.

We then define a concept called "Happy points", noted "H". This is an abstraction of the end-user satisfaction. An application having zero Happy points

[3] http://docs.cloudfoundry.org/concepts/architecture/warden.html.

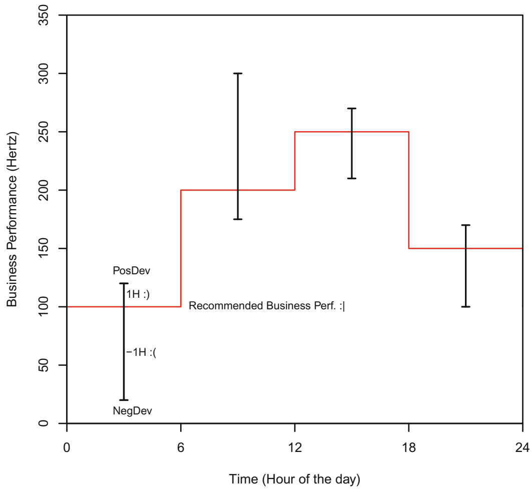

Fig. 1. Service Flexibility Agreement representation

means that the end user is reasonably satisfied. An application being allocated exactly the number of resources corresponding to the RecoBP collects 0 Happy points. This situation corresponds to the traditional SLA threshold. The positive and negative deviations (PosDev and NegDev) declared in the SFA are then the way that each application "reacts" to being given more or less resources than the RecoBP. Indeed, some applications such as video transcoding can benefit from receiving temporarily more resources because they can process more videos and thus make their end user happier. This kind of application reacts linearly to the amount of resources it is allocated. On the other hand, an application such as a web server typically have a "turning point" in their relation between performance and resources. Indeed, giving them less that a certain level of resources (such as the number of front-end VMs) will start to augment the latency in delivering web pages, thus making the end-user unhappy. On the contrary, giving them more than that level of resources will not have any perceptible impact on the end user, as the latency is already small. This is represented by the vertical deviation bars in Fig. 1: the length of the bar represents the amount of Happy points that the application will win/lose when given more/less performance than the recoBP, respectively. In this example, at 10 O'clock, if the PaaS framework allocates the resource corresponding to the recommended business performance of 200 Hz, the application will collect 0 Happy points. If the application gets 300 Hz, it will get 1H, and one more Happy point for each 100 Hz above that. Conversely, if it gets less than the recommended 200 Hz, it will loose Happy point at the rate of one happy point per 25 Hz.

```
SFA:
  - Time: 00:00 to 05:59
    RecoBP: 100 Hz
    PosDev: 20 Hz/H
    NegDev: 80 Hz/H
  - Time: 06:00 to 11:59
    RecoBP: 200 Hz
    PosDev: 100 Hz/H
    NegDev: 25 Hz/H

WorkingModes:
  - WMName: WM1
    actuator: 'cf scale myApp -i 3'
    defaultPower: '300 W'
    maxBusinessPerf: '100 Hz'
  - WMName: WM2
    actuator: 'cf scale myApp -i 5'
    defaultPower: '500 W'
    maxBusinessPerf: '150 Hz'
```

Listing 1: SFA example

We further define the various Working Modes (WM) of an application. A WM corresponds to a set of resources allocated by the PaaS to an application. We associate to each WM its typical power consumption and the maximum business performance it can offer. In practice, a WM corresponds to a number of VMs or Linux Containers, each running instances of the application. Using the SFA, it is now possible to compute the number of Happy points provided by each WM for each time slot.

3 Interaction Among IaaS and PaaS Layers

In novel PaaS frameworks, application scaling is performed by launching more containers. Each container is an instance, or worker, of the application. Containers run in a VM, controlled by an underlying IaaS framework. To save energy, those VMs are traditionally consolidated on a part of the servers of the DC, which permits to switch off unused servers and thus save energy. Using the SFA, it is now possible to predict the amount of resource that an application will need, together with the possible deviations. This will allow to optimize VMs by consolidating containers in them and at a second level optimize servers by consolidating the VMs. VMs and containers movements should be minimized to preserve the energy efficiency. However to achieve this we need to enhance the interaction between the IaaS and PaaS. There are essentially two types of information that need to be exchanged:

- VMs grouping
- VMs life-time

The PaaS layer has a certain degree of knowledge about the architecture of deployed applications. If an application is composed of several containers forming the different layers, it is beneficial to keep them together on the same VMs as much as possible, because they will probably have the same life cycle. Those containers will probably exchange a lot of information among them. Furthermore, they will be switched off together when the application is terminated. This justifies to keep them together on the same VM or group of VMs as much as possible.

Knowing the VM life-time (an estimated duration that the VM will be kept running before being switched off) is important for the IaaS layer when optimizing the energy consumption of a DC through VM consolidation. Indeed, migrating a VM to consolidate it is an investment, and if the VM is about to be terminated soon, this investment is lost. In PaaS environments, the VM life-time is determined by the life-time of the underlying containers. This life-time should be calculated by the PaaS framework and transmitted to the IaaS framework. Of course, the IaaS layer shouldn't be aware of the applications running in the DC, as it is not its role and would furthermore break the separation of concerns between the IaaS and the PaaS. The sharing of VMs grouping and VMs life-time does not break the separation of concerns between IaaS and PaaS, because they are expressed in terms of the IaaS VMs only.

4 Design and Optimization

The SFA is used in the communication between an application and the underlying PaaS framework (while the SLA, on the contrary, is used for the communication between the DC and its clients), as shown in Fig. 2.

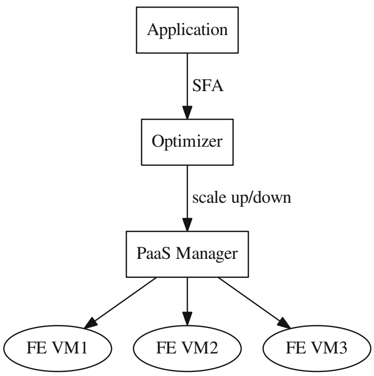

Fig. 2. SFA block diagram

In practice, we add a component in the Cloud Foundry stack called the "Optimizer". This component accepts commands from the application load balancer. Instead of just scale-up/scale-down commands as it is the case currently, we include a recommended scaling level, together with a positive deviation and a negative deviation, as described in Sect. 2. This will let the Optimizer to optimize the full PaaS layer according to multiple criteria: (1) the energy consumption in the DC, (2) the global happiness of the applications (as the sum of all happy points granted to applications) and finally (3) the usage of renewable energies. Using the SFA, the Optimizer is able to find the correct trade-offs that will allow it to comply which energy budgets while maximizing the overall end-user happiness in the DC.

5 Related Works

VM consolidation approaches to reduce energy consumption at IaaS layer have been explored in many recent papers [2–6].

In addition, there are energy efficient solutions based on scaling operations (scale up/down application) based on applications performance metrics. Although these proposals reduce the energy footprint of applications and cloud infrastructures, they do not model the applications performance trend to finely define a trade-off between applications Quality of Service and energy footprint.

Autonomic Computing has been exploited in the design of cloud computing architectures in order to devise autonomic loops aiming at providing coordinated actions among cloud layers for efficiency measures, turning each layer of the cloud stack more autonomous, adaptable and aware of the runtime environment [7,8].

In order to reach a global and efficient state due to conflicting objectives, autonomic loops need to be synchronized. In [7], authors proposed a generic model to synchronize and coordinate autonomic loops in cloud computing stack. The feasibility and scalability of their approach is evaluated via simulation-based experiments on the interaction of several self-adaptive applications with a common self-adaptive infrastructure.

In addition to elasticity, scalability is another major advantage introduced by the cloud paradigm. In [9], automated application scalability in cloud environments are presented. Authors highlight challenges that will likely be addressed in new research efforts and present an ideal scalable cloud system.

[10] proposes a co-design of IaaS and PaaS layers for energy optimization in the cloud. The paper outlines the design of interfaces to enable cross-layer cooperation in clouds. This position paper claims that significant energy gains could be obtained by creating a cooperation API between the IaaS layer and the PaaS layer. Authors discuss two complementary approaches for establishing such cooperation: cross-layer information sharing, and cross-layer coordination.

6 Conclusion and Future Work

The PaaS Cloud paradigm and the corresponding frameworks are currently developing at high-speed within smart cities. As shown in this paper, they provide new opportunities for energy optimization. We presented the SFA, a concept able to describe the flexibility of applications within smart cities. This format includes the trade-off between the amount of resources allocated to a PaaS scalable application, the resulting performance and the corresponding user experience, in a simple way. This allows the PaaS framework to perform multi-criteria optimizations such as lowering the global energy consumption of the data centre in a smart city, using more renewable energies and increasing the application user satisfaction. As future work, we will design the proposed API and components within the CloudFoundry framework.

Acknowledgments. This work has been carried out within the European Project DC4Cities (FP7-ICT-2013.6.2).

References

1. Koomey, J.: Growth in Data Center Electricity Use 2005 to 2010. Analytics Press, Oakland, 1 August 2010 (2011)
2. Cardosa, M., Korupolu, M.R., Singh, A.: Shares and utilities based power consolidation in virtualized server environments. In: Proceedings of the 11th IFIP/IEEE Integrated Network Management (IM 2009), Long Island, NY, USA, June 2009
3. Schröder, K., Nebel, W.: Behavioral model for cloud aware load and power management. In: Proceedings of HotTopiCS 2013, 2013 International Workshop on Hot Topics in Cloud Services, pp. 19–26, ACM, May 2013
4. Hermenier, F., Lorca, X., Menaud, J.-M., Muller, G., Lawall, J.: Entropy: a consolidation manager for clusters. In: Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE 2009, pp. 41–50. ACM (2009)
5. Sheikhalishahi, M., Wallace, R.M., Grandinetti, L., Vazquez-Poletti, J.L., Guerriero, F.: A multi-capacity queuing mechanism in multi-dimensional resource scheduling. In: Pop, F., Potop-Butucaru, M. (eds.) ARMS-CC 2014. LNCS, vol. 8907, pp. 9–25. Springer, Heidelberg (2014)
6. Dupont, C., Hermenier, F., Schulze, T., Basmadjian, R., Somov, A., Giuliani, G.: Plug4green: a flexible energy-aware vm manager to fit data centre particularities. Ad Hoc Netw. **25**, 505–519 (2015)
7. Alvares de Oliveira Jr., F., Sharrock, R., Ledoux, T.: Synchronization of multiple autonomic control loops: application to cloud computing. In: Sirjani, M. (ed.) COORDINATION 2012. LNCS, vol. 7274, pp. 29–43. Springer, Heidelberg (2012)
8. de Oliveira, F.A., Ledoux, T., Sharrock, R.: A framework for the coordination of multiple autonomic managers in cloud environments. In: 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), pp. 179–188, September 2013
9. Vaquero, L.M., Rodero-Merino, L., Buyya, R.: Dynamically scaling applications in the cloud. SIGCOMM Comput. Commun. Rev. **41**(1), 45–52 (2011)
10. Carpen-Amarie, A., Dib, D., Orgerie, A.-C, Pierre, G.: Towards energy-aware IaaS-PaaS co-design. In: SMARTGREENS 2014, pp. 203–208 (2014)