Renewable Energy Aware Data Centres: The Problem of Controlling the Applications Workload

Corentin Dupont

University of Trento
Trento, Italy
corentin.dupont@unitn.it

Abstract. Data centres are powerful facilities which aim at hosting ICT services. They have huge needs in term of power supply; furthermore the current trend is to prioritize the utilization of renewable energies over brown energies. Renewable energies tend to be very variable in time (e.g. solar energy), and thus renewable energy aware algorithms tries to schedule the applications running in the data centres accordingly. However, one of the main problems is that most of the time very little information is known about the applications running in data centres. More specifically, we need to have more information about the current and planned workload of an application, and the tolerance of that application to have its workload rescheduled. In this paper, we will first survey the problem of understanding, building information about and finally controlling the load generated by applications. Secondly we will propose hints of solutions for that problem.

Keywords: Data Centre, Renewable Energy, Application Profile, Resource Management, Job Scheduling

1 Introduction

In the last decade, energy-awareness has been of great concern for researchers. Indeed, the prices for electricity are constantly getting higher and carbon emissions to the environment are increasing every year[1]. Users try to switch off the lights or put the ICT devices in sleep mode when not needed. This straightforward approach works fine for domestic appliances but more complicated ICT facilities like data centres (DCs) require, however, more sophisticated approaches to achieve the energy savings while preserving the performance. Data centres are large facilities which purpose is to host information processing and telecommunication services for scientific and/or business applications. Until recently, research on data centres has been

[1] Green Grid Consortium, http://www.thegreengrid.org

focused only on improving metrics like performance, reliability, and availability. However, due to the rise in service demands together with energy costs, the energy efficiency has now been added as a new key metric for data centres. Energy-aware strategies are beginning to be integrated inside the data centre resource manager. In practice, a Virtual Machine (VM) placement algorithm considers the data centre and the workload characteristics to place the VMs among the servers in the most efficient way, considering performance and energy consumption. This placement must be done respecting the requirements of the Service Level Agreement (SLA) existing between the data centre and its clients.

In parallel to reducing the overall energy consumption, the current trend is to foster the use of renewable energies. Renewable energies have the problem to be very variable and time-dependent: for example solar power is available only during the day, and is subject to variations due to the meteorological conditions. Thus, renewable energy aware algorithms in data centres must try to shift the workload of running applications in time, to match it with the availability (or forecasted availability) of renewable energy. This is however difficult because generally very little information is known about a running application. It is thus necessary to augment this “meta” knowledge about applications running in data centres, to allow better match-making between green energy availability and the requirement of the applications in term of energy. Secondly, having provided this meta-data, the applications must provide a way for the data centre management framework to control, at least partially, the workload to be run. In this paper we survey the problem of understanding and controlling the workload of applications, with the final objective to allow energy aware algorithms to perform better energy savings.

In the remainder of this paper, we will first perform a survey of the related works in section 2 and define the problem that need to be solved in Section 3. We will then propose a possible solution architecture and technologies in Section 4 and finally conclude in Section 5.

2 State of the Art

The problem described is transversal: it is spanning several domains such as energy aware algorithms, data centre management systems, application profiling and static property derivation from applications. We review the related works in two sub-sections: the optimization algorithms frequently used in Data Centres and how they are embedded in data centre management frameworks in a first part, and then we will survey existing techniques on automatic program property derivation, and specifically of the application profile.

2.1 Energy Aware Optimization Algorithms in DC Management Frameworks

Virtualization was and still is the main feature for saving energy in data centres. It allows to right-size the largest culprits of energy over-consumption which are underutilized servers. In typical environments these machines sit idle almost all of the time and consume a significant amount of energy that is not needed in practice to satisfy the requirements of the applications. In a virtualized data centre, applications are embedded into virtual machines (VMs) that are consolidated on a reduced amount to servers to increase their utilization then lower the overall energy utilization. In addition, live migration allows re-arranging the VMs placement without significant disruption to the underlying services, to be able to react to fluctuating resources requirements. There is an abundant literature on energy aware algorithms for data centres. To lower the energy consumption while fulfilling performance requirements, the authors of [1] propose a flexible and energy-aware framework for the (re)allocation of virtual machines (VMs) in a data centre. The framework, being independent from the data centre management system, computes and enacts an energy aware placement of virtual machines based on constraints expressed through service level agreements (SLA). This problem is known to be NP-hard [2] with a large solution space. The framework's flexibility is achieved by decoupling the expressed constraints from the algorithms using the Constraint Programming paradigm and programming language. The problem of consolidating and rearranging the allocation of virtual machines in a data centre in an energy efficient manner is described in [3]. In the heuristic proposed in [3], the algorithm computes, for each VM to be moved, the appropriate hosting server that leads to minimizing the current overall power consumption of a data centre. This is similar to the First Fit Decreasing (FFD) algorithm which has been used in previous works [4][5], with the addition of power-awareness for choosing the server. The limitation of these frameworks comes from the fact that it has only a macroscopic view of the "size" of the workloads, which is often mapped to the reserved size of the VMs. For example, Amazon EC3 defines only a few VM size (i.e. M1 small, M1 medium...). This prevents a fine consolidation of the VMs, as resources have to be reserved for each instance.

2.2 Advanced Distributed Computing Paradigms

Hadoop[2] is a very popular open source solution for distributing workload across multiple nodes, implementing the MapReduce [6] paradigm. In this paradigm, a computation over a set of data is broken into a "map" function and a "reduce" function. The map function can be performed simultaneously and independently over different sub-sets of data. This independence of the computations allows for massive distribution of the computation on many computers. In a second step, the result of the map functions is collected and the final result is computed by the reduce function. GreenHadoop [7] uses Hadoop as a data-processing framework, and makes prediction about the availability of solar power. These predictions are used to schedule time-bound computations in order to maximize the green energy consumption of the data

[2] Hadoop: http://hadoop.apache.org/

centre. However, Hadoop and the solutions based on it have an inherent limitation: the computations must be expressed in term of a map and a reduce functions, which is a great constraint. This is a very restrictive programming paradigm and furthermore not every computation can be expressed in such a way. GreenSlot [8] tries to schedule jobs in Supercomputing data centres in order to maximize the use of renewable energy. It uses historical data and weather forecasts to determine the best moment of day to run a particular job. However, GreenHadoop and GreenSlot suffer from the fact that, in both case, the user have to describe the workload/job characteristics (notably its time-bound constraints). This information is at best very coarse grained, and sometime not available at all.

Still in the domain of distributed computing, the newly introduced serializable delimited continuations of Scala may allow to overcome the limitations of MapReduce. Scala is a general purpose programming language designed to express common programming patterns in a concise, elegant, and type-safe way. It smoothly integrates features of object-oriented and functional languages. Swarm[3] is a framework for Scala allowing the creation of applications which can scale transparently through the continuation-based approach mentioned before. Like Map-Reduce, Swarm follows the maxim "move the computation, not the data". Swarm takes the concept much further, allowing it to be applied to almost any computation, not just those that can be broken down into map and reduce operations. However, Swarm is still at a very early stage. A “GreenSwarm”, taking into account renewable energy availability in a way akin to that of GreenHadoop, would undoubtedly be an interesting development for this technology. In the same line of thinking, Cloud Haskell [9] is a domain-specific language (DSL) for distributed computing developed in Haskell (a purely functional general purpose language). It also uses serialized function closures for transmitting computations across the network.

As already mentioned before, Amazon Elastic Compute Cloud (EC2) is defining several sizes of VMs that their clients can pick up. However this level of granularity does not allow a fine consolidation of the VMs, and does not give a lot of information on the profile of the applications running, and specifically their planned variation of load in time. The building of an application (or workload) profile is addressed briefly in [10]. Historical values are used to consolidate a profile for the workload. However, no standard is described allowing the transmission and reuse of this profile. It would be of course very interesting to build this application profile automatically or semi-automatically, for example by analysing the source code of the application. Property derivation from programs and property proving is very close to theorem checking in mathematics: there is a one to one correspondence between a program and a proof (called the Curry-Howard correspondence). In [11], the authors implement a tool called HipSpec able to automatically derive and prove properties of functional programs.

[3] Swarm: http://www.scala-lang.org/node/3485

3 The problem: How to Understand the Load of Applications?

A great challenge of using efficiently renewable energies in data centres is to be able to schedule correctly the applications workload. Indeed, as shown on Figure 1, the availability of renewable energy can have a great variation in time, with comparison to brown energies. To increase the use of renewable energies with respect to brown energies, it is necessary to shift in time the workload of some applications in the data centre. This shows the importance of being able to know the workload an application will have to run at a certain point of time, to understand under what conditions it can be shifted or delayed, and *in fine* to schedule it correctly.

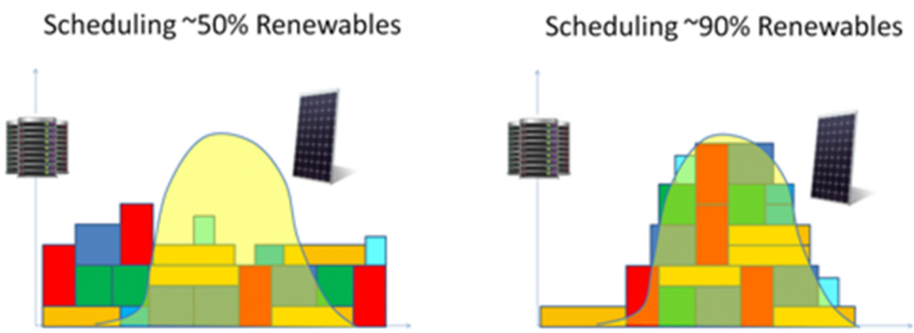

Figure 1. Matching the application workload with the availability of the renewable energies

Yet, currently most of the applications running in data centres are unaware of their self workload: they are unable to predict how much computing power they will require and when. Furthermore, the time scale at which computational workload in data centres vary can be very different from the time scale at which renewable energy may vary. For example, renewable energies may have seasonal variations, like solar energy availability, while load in data centres can have short timed variations, like hourly or daily variations.

It is of crucial importance for many ICT domains to be able to know what an algorithm contained in an application will need in term of computing resources, and yet there are little developments in this field. In data centres, the knowledge of the requirements of an application in terms of resources is still “meta-knowledge”, i.e. the knowledge of the data centre operators. It is the role of the data centre operator to provision sufficient resources for an application, and this provision is often done in a static way. For example, in data centres, database indexing maintenance operations are usually performed at night, to minimize the impact on the overall performance. However, in a data centre using primarily solar power, it would be interesting to shift this task during the lunch break, when the sun is shining. The knowledge that this particular task, “database indexing”, can cope with a 12 hour shift, and that it takes approximately half an hour, belongs to the operator’s knowledge. It is very coarse grained and subjective knowledge.

The optimization algorithms used in data centres typically needs to know the profile of applications in order to minimize a utility function (for example the energy consumption). A heuristic widely used for data centre VM management is bin-repacking, and among meta-heuristics we can count Constraint Programming, Genetic Algorithm or Simulated Annealing. For example a multi-dimensional bin-repacking will help consolidate the VMs on a part of the servers and permit switching off another part of the servers, thus saving energy. However this algorithm needs to know the profile of the applications in term of memory consumption and CPU demand in order to map those parameters to the abstract dimensions of the objects to be packed.

This advocate the need for:

1) a standardized format and protocol for applications to advertise in real-time their own needs in term of resources, including possible performance trade-offs and uncertainty ranges (this format is part of the so-called “application profile”),
2) a library and programming methodology to ease the extraction of application profiles from the application source code itself, at run-time or compile-time,
3) a data centre management framework and algorithms able to read the application profiles and use them to consolidate and schedule the applications on the servers in the appropriate way, in order to minimize a given utility function,
4) a library and programming methodology to allow an external process to control the application load to some extent.

The problem of deriving an application profile is very accurate in many load optimization and prediction problems. More generally, automatic derivation of program macroscopic properties is a topic that has a great number of applications in the ICT field, especially with the emergence of Cloud computing. For example, a smart phone application might want to “off-load” part of its workload to the Cloud, or on the contrary, a service running in the Cloud might be relocated in the local device for performance reasons. These application or service migrations must be controlled by a decision framework, which must know the exact applications profile in order to take the right migrations decisions.

We need to research and develop the algorithms, methodologies and tools to make applications “self aware” to a certain extent. In practice very seldom applications are able to know “what” workload they have to perform and when it needs to be done. They just “do it”, in some sense. There is no internal representation of this workload, or when there is one (like in Hadoop or some database management framework), it is not general enough and standardized.

4 A Possible Architecture

We present an architecture for the communication between the different actors of a data centre producing and using the application profile.

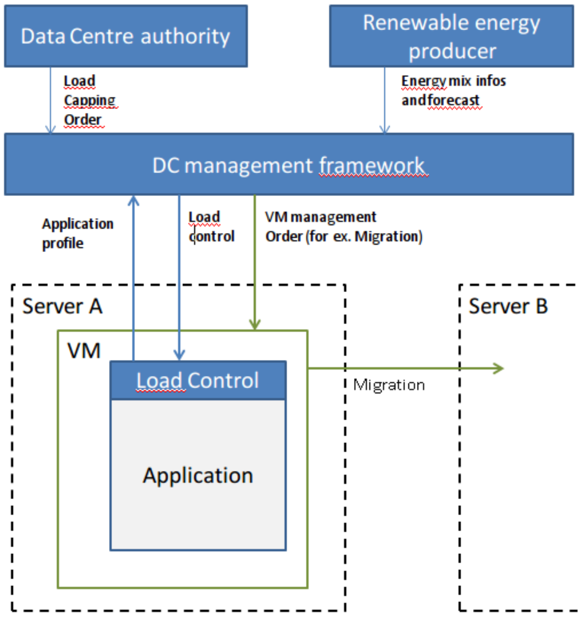

Figure 2. Architecture overview

In Figure 2, the DC authority can force the DC management framework to respect some energy consumption limitations and give possible trade-offs in term of performance. At the same time the renewable energy producer will give details on the energy mix available, and the forecasts for the next days. The DC management framework will then collect the application profiles from the running applications. The DC management framework will ask to the applications, according to the information contained in these profiles, to respect some load conditions and time restrictions in order to globally match the workload of the data centre with current and forecasted renewable energy availability. Additionally, at a more macroscopic level the traditional VM and server management can happen: VMs can be migrated in order to consolidate them on a part of the servers and the freed off servers can be switched off to save energy.

4.1 Representing the Application Workload

Applications running on desktop operating systems (for ex. Linux, Windows or Mac OS) typically communicate very few to no information to their host operating system when launched. For example, a multi-threaded application will ask the creation of its threads to the operating system, thus letting it know the number of threads. The idea is to increase this amount of information in the case of applications running in data centres, to let the operating system and subsequently the data centre management framework build an appropriate and up to date application profile. Based on this profile, the load balancer and scheduling system of the data centre will be able to make better optimizations of the workload.

The granularity of the virtualization in a Cloud data centre is at the level of the Operating System: the Virtual Machines are containing one OS, and this is what is benchmarked and possibly moved. This level is still macroscopic: usually a server contains around 10 VMs. However, the hosted OS may contain many applications. Better energy savings could possibly be achieved if this level of granularity was lower. We also need a way to let the application externalize some of its workload. A bit like in Hadoop, the application could let to an external process the task to distribute and run a part of its workload. Recent advances in parallel processing may let us perform efficient workload allocation (see Section 2). If the application embeds its own scheduler and doesn't externalize its workload, it should nevertheless be able to receive orders or advises on the best moments to run its workload from the DC management framework. In that respect, currently, the applications offer very little cooperation with the operating system. In multi-task systems, the application will let the OS pre-empt it. But it will not let the OS know if a computationally intensive activity needs to be done mandatorily in the next hour, for example.

The applications need to be more "self-aware": they need to have an internal representation of their own processes and workload. In other terms, they need to know "what" they have to do, and not just "how" to do it. As already mentioned before, applications are usually working in a "procedural" way: they describe a step by step method to solve a problem, but do not have a more macroscopic representation of this problem. The idea is to include in the application profile what the applications need to do and when, to a certain extend and with a certain granularity. For example, in the case of the application indexing a database, the application needs to "know" that this process can be delayed, and needs to transmit that to the DC Management System.

4.2 Extracting the Application Profiles

To extract automatically the application profile, a possibility is to perform static analysis of the application source code itself in order to infer the complexity of the algorithms. This is in the general case impossible to perform mechanically, because of theoretical limitations (namely the indecidability of the halting problem [12]). However, the general idea would be to provide a library to allow the programmer to "instrument" the code with meta-data about the various algorithms complexity. This meta-data would be composable, in the same way that simple algorithms may be composed to build a more complex algorithm. This would allow retrieving, at a higher level, the complexity of algorithms and giving hints on the maximum and average resources needed by the application. This information will be collected by the management framework of the data centre and allow fine-grained allocation and optimization, using the optimization algorithms mentioned before.

A second way one could establish dynamic properties of programs, is to use a theorem prover. With such a tool, a programmer could bundle his program with the proof that certain properties such as a maximum load hold true. Proving properties of programs is much easier in functional programming because it is semantically simpler than imperative programming and thus, easier to reason about.

5 Conclusion

In this paper we presented the problem that faces the algorithms aiming at fostering a better utilization of renewable energy in data centres: the lack of information about the running applications at a macroscopic level. We need a way to automatically or semi-automatically derive an application profile, to collect it and exploit it at data centre level, with the aim to allow renewable energy algorithms to perform better. Indeed those application profiles are necessary as an input for optimization methods typically used in data centres. The information contained in the application profile will be used to compute the allocation and scheduling of the applications. We presented a possible architecture and technologies able to create this application profile, to transmit it to the data centre management framework, and finally to control accordingly the applications workload in order to better utilize renewable energies.

Acknowledgments

The authors would like to thanks the reviewers for their useful comments.

6 References

[1] C. Dupont, G. Giuliani, F. Hermenier, T. Schulze, and A. Somov. An energy aware framework for virtual machine placement in cloud federated data centres. In Proceedings of the third International Conference of Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), May 2012

[2] E. Lawler. Recent results in the theory of machine scheduling. In Mathematical Programming: The State of the Art. 1983. Springer-Verlag, Berlin, Germany.

[3] D.-M. Quan, R. Basmadjian, H. De Meer, R. Lent, T. Mahmoodi, D. Sannelli, F. Mezza, and C. Dupont. Energy efficient resource allocation strategy for cloud data centres. In Proceedings of the 26th International Symposium on Computer and information Sciences (London, UK, September 26-28, 2011). ISCIS'11.Springer, 133-141.

[4] N. Bobroff, A. Kochut, and K. Beaty. Dynamic placement of virtual machines for managing SLA violations. Integrated Network

Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on, pages 119–128, May 2007.
[5] T. Wood, P.J. Shenoy, A. Venkataramani, and M.S. Yousif. Black-box and gray-box strategies for virtual machine migration. In Proceedings of the 4th ACM/USENIX Symposium on Networked Systems Design and Implementation (Cambridge, MA, USA). 2007. USENIX Association, Berkeley, CA, USA, 17-17.
[6] MapReduce: Simplified Data Processing on Large Clusters, Jeffrey Dean and Sanjay Ghemawat
[7] Inigo Goiri, Kien Le Thu, D. Nguyen, Jordi Guitart, Jordi Torres, Ricardo Bianchini, GreenHadoop: Leveraging Green Energy in Data-Processing Frameworks
[8] Goiri et al. GreenSlot: Scheduling Energy Consumption in Green Datacenters. In Supercomputing, 2011.
[9] Jeff Epstein, Andrew P. Black, Simon Peyton-Jones, Towards Haskell in the Cloud
[10] Richard Talaber e al. Using Virtualization to Improve Data Centre Efficiency, The Green Grid consortium, 2009
[11] Koen Claessen, Moa Johansson, Nicholas Smallbone, Dan Rosen, HipSpec: Automating Inductive Proofs of Program Properties
[12] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Halting_problem.html