EnviroInfo 2013: Environmental Informatics and Renewable Energies


# Energy Efficiency in Cloud Software Architectures

Giuseppe Procaccianti, Stefano Bevini, Patricia Lago[1]

**Abstract**

Cloud-based software is often considered as providing a greener, more energy-efficient solution. At the same time, it introduces more complexity and demands for new investments in cloud services, technologies, and competencies for migration, maintenance, and evolution of the underlying software architectures. To understand better the implications of cloud software architectures on energy efficiency, in this paper we present the preliminary results of a systematic literature review that investigates what kind of software architectures for cloud service provisioning allow to achieve energy-efficient solutions.

## 1. Introduction

Cloud computing infrastructures are often described as an energy-efficient technology (Berl et al., 2010). In principle, improving data center utilization by virtualizing resources is a way to save energy. Nowadays, energy efficiency (EE) is starting to be considered as a Service-Level Objective (SLO), i.e. a specific, measurable characteristic of a service, to be described as achievement values in Service Level Agreements (SLAs)[2]. An example would be: "The energy bill of the client should be reduced by 20% in one year". We believe that cloud service providers (CSPs) could benefit from representing EE as a SLO. However, in order to offer cloud services, providers rely on very complex software architectures. It is yet unclear, and possibly unexplored, what architecture characteristics do positively or negatively influence EE, and if there are explicit or implicit reference architectures that can help ensuring that the EE exposed as SLO is delivered. Our work aims at analyzing the relation between software architectures and EE as a SLO in cloud-based service solutions. To this end, we carry out a systematic survey of the publications discussing how software architectures impact EE in cloud solutions, and analyze what types of architectures emerge from such publications. In this paper, we present our preliminary findings.

## 2. Motivation

Recently, the problem of defining SLAs for cloud services has been widely discussed among practitioners and experts of the field. Customers do not know precisely what should or shouldn't be included in an SLA regarding the provision of cloud services. On the other side, providers lack standard tools to assess and certify the level of the service they are providing, in terms of performance, reliability, serviceability etc. The same goes for EE: even if it has been proven that cloud technologies provide benefits in terms of energy savings (Berl et al., 2010), this factor is not adequately exploited as an added value for cloud service provisioning. This work aims at investigating scientific literature to explore the use of EE as a valid SLO in cloud service provisioning.

[1] VU University Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.
email: g.procaccianti@vu.nl , stefano.bevini@gmail.com , p.lago@vu.nl
[2] http://www.greenbiz.com/news/2009/01/12/energy-efficiency-new-sla , last visited on June 12th, 2013

Another perspective of our work is the relationship between software architectures and EE. Nowadays, the role of software in energy consumption is widely discussed among the scientific community, and a number of metrics for software EE has been proposed (Bozzelli, Gu and Lago, 2013). Our work tries to advance to the next step: is it possible to quantify the effects on energy consumption when adopting a certain software architecture? We will perform a literature review of software architecture studies searching for an answer to this question, and eventually identify the possible gap for future research.

## 3. Study Design and Execution

The research question (RQ) driving our study is: *What kind of software architectures for cloud service provisioning allow to define Service Level Objectives regarding energy efficiency?*

In order to answer our RQ, we followed a systematic literature review process (Kitchenham et al., 2009). We performed a preliminary analysis of the research space, and we identified 306 hits (i.e. potential related studies). Thus, we formulated a review protocol for our study, by defining a search query for interrogation of academic databases and inclusion and exclusion criteria to refine our results according to our RQ. In this way, we selected the primary studies for our research. We subsequently classified and analyzed these studies in order to extract relevant results.

### 3.1 Search Strategy

From our research question, we extracted five keywords: *"software architecture", "cloud", "service", "SLA", "energy"*. Then, in order to expand our search space, we identified relevant terms and synonyms. Finally, this resulted in the following search string:

> *"software architecture" AND cloud AND service AND "(energy OR power) efficiency" AND (SLA OR SLO OR "service level")*

We used *Google Scholar* as **data source** for our search, with a time range spanning from 2000 to 2013. The keywords specified in the search string were applied to titles, abstracts, and body of the studies, in order to include all possibly relevant studies.

### 3.2 Primary Study Selection

#### 3.2.1 Inclusion/Exclusion Criteria

We defined two sets of criteria to identify the articles to be selected as primary studies. A candidate study must satisfy all **inclusion (I)** criteria, while not matching any **exclusion (E)** criteria. In Table 1 the list of the criteria we adopted is presented.

#### 3.2.2 Results of the selection process

Our search strategy resulted in 149 original contributions. Out of those, only 10 studies were judged as completely irrelevant basing upon their title. We then proceeded into applying our inclusion/exclusion criteria to the abstracts, which more than halved our number of candidates: 78 studies were removed. Finally, we reviewed the full-text of the remaining 61 contributions, and out of those we identified our final set of 26 primary studies. Table 2 gives an overview of the selection process.

| *Nr.* | *Description* | *Rationale* |
|---|---|---|
| I1 | A study that directly proposes software architectures, architectural styles or strategies. | We want to identify how software architectures affect EE, thus we need articles proposing software architectures that are generally reusable or applicable, or indirectly proposes them from a service provisioning perspective. |
| I2 | A study that addresses EE as a quality requirement. | We want to investigate whether EE is considered, either implicitly or explicitly, by providers or experts, as a quality attribute for cloud services. |
| I3 | A study that is developed by either of academics and practitioners. | Both academic and industrial solutions are relevant to this study. |
| I4 | A study that is published in software engineering/cloud computing field. | Software engineering is our reference field, but cloud computing research can provide us an insight on what trends are set in terms of software architectures for cloud. |
| I5 | A study that is peer-reviewed. | A peer-reviewed paper guarantees a certain level of quality and contains reasonable amount of content. |
| I6 | A study that is written in English. | For feasibility reasons papers written in other languages than English are excluded. |
| E1 | A study that does not propose software solutions for EE. | Traditionally, EE has been regarded as a hardware issue. We want to drive past this assumption and address the software impact of power consumption. |
| E2 | A study that does not imply any type of service provisioning. | We are not interested in solutions that generally increase the EE of a datacenter, without having in mind how to provide an energy-efficient service to a customer. |
| E3 | A study that does not consider EE as a primary quality characteristic. | We are not interested in studies that consider EE a secondary concern. |
| E4 | A study that does not aim at optimizing the EE of the cloud computing infrastructure. | Mobile devices often leverage cloud services by offloading computation tasks, in order to increase their battery life. Although this is an EE improvement, it is not relevant for the EE of the cloud computing infrastructure, thus we want to exclude these solutions from our study. |

Table 1.
Inclusion/Exclusion Criteria.

| ***Stages of the selection process*** | ***Removed*** | ***Result*** |
|---|---|---|
| Search results | N/A | 149 |
| Title checking | 10 | 139 |
| Abstract checking | 78 | 61 |
| Full-text checking | 35 | **26** |

Table 2.
Overview of the selection process.

### 3.2.3 Search results traceability

We recorded the reference details of the studies using JabRef [3]. For every step of the review process, a different JabRef database file was created. Moreover, we used an Excel sheet to report the matching of the inclusion/exclusion criteria and the stage of the decision (title, abstract or full-text checking) for each study.

## 3.3 Data Extraction and Analysis

We used an extraction form in order to retrieve and store relevant information about each primary study. Besides general information, the form records how EE is addressed and which architectural elements were identified in the presented solution. The extraction form is structured as follows:

- *Study Identifier*: provides a unique identifier for the study;
- *Study Title*: the publication title;
- *Study Type:* the publication type (i.e. journal article, conference article, thesis);
- *How EE is addressed:* a brief summary of how the presented solution addresses the EE of the cloud infrastructure;
- *Main architectural elements:* the main software elements of the solution.

For the data analysis phase, we adopted a qualitative approach, by means of a coding technique. The first step was an exploratory study of the selected contributions, in order to define a set of *codes* regarding the objects of our study, namely software architectural strategies, practices and elements. Subsequently, we reviewed again the software architectures exposed in the primary studies according to our set of codes, in order to examine the frequency of the elements we identified.

As regards the traceability of our analysis, whenever a code was identified in a primary study we annotated the corresponding section of the full-text of the contribution. In this way, the systematic mapping we performed can be verified by independent reviewers.

In section 4 we present, as preliminary results, the codes that resulted from the analysis and the corresponding primary studies. These codes can serve as a basis to classify and describe green software architectures.

## 4. Codes

During our analysis, we identified three different levels of architecture-related concepts that were used for describing software solutions for EE. Those three levels (see Figure 1) are:

- *Strategies:* a strategy can be defined as the way through which a particular software solution addresses EE;
- *Techniques:* a technique can be defined as the instantiation, or enactment, of a strategy through a specific technical approach;
- *Components:* an individual architectural component (Garlan and Shaw, 1993) that plays a defined role in the application of a technique.

[3] http://jabref.sourceforge.net/, last visited on July 2nd, 2013.

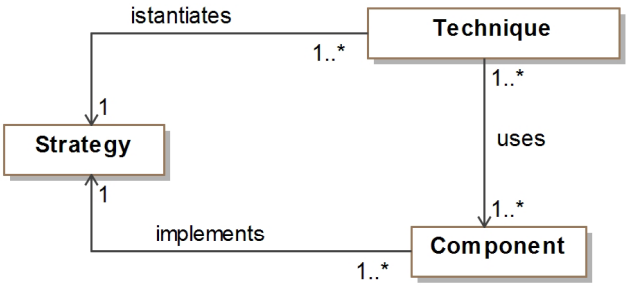


Figure 1.
Reference model for architecture-related concepts.

We identified three main strategies through which EE is addressed in the primary studies. Note that those strategies are not intended to be mutually exclusive: in fact, many solutions may adopt more than one strategy.

- *Energy Monitoring*: some components of the software architecture are devoted at monitoring the energy consumption of the IT system;
- *Self Adaptation*: some components of the software architecture enable the possibility of adapting the software behaviour in order to save energy;
- *Cloud Federation:* the software architecture comprehends the possibility to "lease" or "negotiate" the usage of cloud services from other providers basing upon energy consumption requirements.

For each strategy, we identified several techniques and components recurring in the primary studies.

In Table 3 we present the software components, along with a brief description, the strategy they typically implement and their number of occurrences in the primary studies. In Table 4 we present the architectural techniques, briefly described, in relation with the corresponding strategies and the primary studies where they were identified. In the following section, we discuss the outcome of our analysis.

## 5. Discussion

Our list of strategies, techniques and components provides initial insights in the State-of-the-Art of green software architectures for the cloud. As shown in Table 4, Self-Adaptation is the strategy aimed at EE that appears most frequently (i.e. reported 35 times), among the three strategies we have identified during this literature review. Surprisingly, Energy Monitoring is not considered an efficient strategy per se, being it presented mostly (i.e., 10 out of 11 articles) in combination with Self-Adaptation. This suggests that Energy Monitoring plays an important role in cloud software architectures that realize effective Self-Adaptation techniques. For example, we observed that Workload Scheduling or VM Consolidation techniques (see Table 4) are typically driven by Metering or other Energy Monitoring techniques.

As illustrated in Table 4, the most investigated techniques are Workload Scheduling, (identified in 18 out of 26 articles) and Consolidation (11 out of 26). The Scaling technique appears less frequently, probably because it typically requires more effort to be implemented in the application logic.

As regards our analysis of software components, in Table 3 we can observe how the most frequent components are related to the Self Adaptation strategy, namely Internal SLA Violation Checker and Workload Scheduler (14 occurrences), VM Allocator (13 occurrences) and Adaptation Engine (11 occurrences). Note the presence of two different components dedicated to detect possible SLA violations. This strengthens our impression that a major challenge in cloud service provisioning, is to find the best trade-off between EE and SLA fulfilment.

| Strategies | Components | Description | # |
|---|---|---|---|
| Energy Monitoring | Energy Dashboard | A graphical interface component of the software architecture that provides users with software energy consumption information. | 4 |
| | Energy Database | A component of the software architecture devoted to storing energy consumption information. | 5 |
| | Energy Indicators | Components of the software architecture devoted to "rate" or classify software behaviour, or to provide real-time metrics upon energy consumption. | 4 |
| | Energy Collectors | Components of the software architecture devoted to retrieve and collect energy information from hardware or software sensors. | 6 |
| | Energy Communication Bus | Component of the software architecture devoted to provide a common interface between the collectors and the energy database. | 3 |
| | Energy Model | A component of the software architecture devoted to estimate or predict the power consumption of a software application in real-time. | 6 |
| | Energy Monitor | A component of the software architecture devoted to monitor the energy consumption of (a part of) the software system. | 4 |
| Self Adaptation | Adaptation Engine | A component of the software architecture devoted to find an optimal solution to an objective function modelling the EE of the system. | 11 |
| | Workload Scheduler | A component of the software architecture able to define, schedule and assign workloads to computational units. | 14 |
| | Scale unit | A defined set of IT resources that represents a unit of growth. | 1 |
| | Queue | One or more software components of the software architecture devoted to organize items (services, VMs, jobs) in different orders of priority according to energy consumption. | 3 |
| | VM Allocator | In virtualized environments, a component able to migrate and displace VMs on servers. | 13 |
| | SLA Violation Checker (Internal) | A component of the software architecture devoted to check and ensure the fulfilment of SLAs (NOTE: in this case the checker evaluates the violation of internal services towards external clients). | 14 |
| Cloud Federation | Energy Broker | A component of the software architecture devoted to providing access to energy efficient services. | 3 |
| | Energy Orchestrator | A component of a SOA able to switch services in case of relevant differences in their EE. | 3 |
| | Green Services Directory | A listing of all available services with energy consumption information. | 1 |
| | SLA Violation Checker (External) | A component of the software architecture devoted to check and ensure the fulfilment of SLAs (NOTE: in this case the checker evaluates the violation of external services towards internal clients). | 2 |

Table 3.
Software architectural components for EE.

| Strategies | Techniques | Description | Reference studies | # |
|---|---|---|---|---|
| Energy Monitoring | Static classification | Energy classification of software based upon the power consumption specifications of the hardware components | (Alessandro, 2010)(Curtis, 2008)(Götz et al., 2011) | 3 |
| | Metering | Power consumption real-time monitoring through external power meters | (Carpentier et al., 2012)(Curtis, 2008)(Forell, Milojicic and Talwar, 2011)(Katsaros et al., 2012)(Noureddine, Rouvoy and Seinturier, 2011)(Xu et al., 2012) | 6 |
| | Modelling | Power consumption on-line estimation using predictive models | (De Oliveira Jr, Ledoux and others, 2010) (Dougherty, 2011)(Dupont et al., 2012)(Noureddine, Rouvoy and Seinturier, 2011) | 4 |
| Self Adaptation | Scaling | Software is able to scale down in case of low requests or usage, to save energy | (Chacin Martìnez and others, 2011)(Rogers and Homann, 2008) (Dougherty, 2011) (Harman et al., 2012)(Katsaros et al., 2012)(Xu et al., 2011) | 6 |
| | Consolidation | In virtualization scenarios, the possibility to regroup VMs sparse among many servers, to reduce the number of active machines | (Carpentier et al., 2012)(Rogers and Homann, 2008)(Dupont et al., 2012) (Harman et al., 2012)(Katsaros et al., 2012)(Kounev, 2011)(Noureddine, Rouvoy and Seinturier, 2012) (Sekhar, Jeba and Durga, 2012) (Xiong, Han and Vandenberg, 2012) (Xu et al., 2011) (Xu et al., 2012) | 11 |
| | Workload Scheduling | Some components of the software architecture are devoted to manage and schedule the workload of the computational units | (Alessandro, 2010) (Chacin Martìnez and others, 2011)(De Oliveira Jr, Ledoux and others, 2010) (Dougherty, 2011)(Forell, Milojicic and Talwar, 2011)(Götz et al., 2011) (Harman et al., 2012) (Hedwig et al., 2009)(Huber, Brosig and Kounev, 2011)(Katsaros et al., 2012)(Kounev, 2011)(Lu, Varman and Doshi, 2011)(Sekhar, Jeba and Durga, 2012) (Sevalnev, 2012)(Xiong, Han and Vandenberg, 2012)(Xu et al., 2012)(Xu et al., 2011) | 18 |
| Cloud Federation | Energy Brokering | The software architecture exposes their services together with their energy consumption information | (Forell, Milojicic and Talwar, 2011) (Gholamhosseinian and Khalifeh, 2012)(Villegas et al., 2012) | 3 |
| | Service Adaptation | Switching iso-functional services depending on their energy consumption | (Forell, Milojicic and Talwar, 2011)(Villegas et al., 2012)(Wu et al., 2012)(Xiong, Han and Vandenberg, 2012) | 4 |

Table 4.
Software architectural techniques for EE.

## 6. Conclusions

EE is a primary issue for CSPs, as data centers are major power consumers. In this context, both the hardware configuration and the software architecture of the cloud computing infrastructure must be carefully designed in order to accommodate power consumption constraints. In this work, we started analyzing the state-of-the-art of energy efficient cloud computing software solutions. Referring to the RQ we addressed during this work, although we were not yet able to identify full-fledged architectural solutions that consider EE as a SLO, this preliminary study resulted in a list of software architectural strategies, techniques and components to address EE. Through this list, CSPs can identify software solutions for EE, suitable to be implemented into their existing platforms, or to be used as a reference, when designing a service provisioning architecture from scratch.

This is a first step towards green software architectures. Future works will be devoted to further analyze our results in order to extract reusable software solutions (i.e., design or architectural patterns) for designing energy efficient software systems.

## Acknowledgements

Europe invests in your future via the European Fund for Regional Development.

## Bibliography

Alessandro, M. (2010) *Non-Functional Requirements Over Dynamic Infrastructure Services*, Master Thesis: Università degli Studi Roma Tre.

Berl, A., Gelenbe, E., Di Girolamo, M., Giuliani, G., De Meer, H., Dang, M.Q. and Pentikousis, K. (2010) 'Energy-efficient cloud computing', *The Computer Journal*, vol. 53, no. 7, pp. 1045-1051.

Bozzelli, P., Gu, Q. and Lago, P. (2013) *A systematic literature review on green software metrics*, Technical Report: VU University Amsterdam.

Carpentier, J., Gelas, J.-P., Lefevre, L., Morel, M., Mornard, O. and Laisne, J.-P. (2012) 'CompatibleOne: Designing an Energy Efficient Open Source Cloud Broker', Cloud and Green Computing (CGC), 2012 Second International Conference on, 199-205.

Chacin Martìnez, P.J. and others (2011) *A Middleware framework for self-adaptive large scale distributed services*, PhD Thesis: Universitat Politècnica de Catalunya.

Curtis, L. (2008) 'Environmentally Sustainable Infrastructure Design', *The Architecture Journal*, vol. 18, pp. 2-8.

De Oliveira Jr, F.G.A., Ledoux, T. and others (2010) 'Self-optimisation of the energy footprint in Service-Oriented Architectures', Proceedings of the 1st Workshop on Green Computing, 4-9.

Dougherty, B.P. (2011) *Configuration and Deployment Derivation Strategies for Distributed Real-time and Embedded Systems*, PhD Thesis: Vanderbilt University.

Dupont, C., Giuliani, G., Hermenier, F., Schulze, T. and Somov, A. (2012) 'An energy aware framework for virtual machine placement in cloud federated data centres', Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on, 1-10.

Forell, T., Milojicic, D. and Talwar, V. (2011) 'Cloud management: Challenges and opportunities', Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on, 881-889.

Garlan, D. and Shaw, M. (1993) 'An introduction to software architecture', *Advances in software engineering and knowledge engineering*, vol. 1, pp. 1-40.

Gholamhosseinian, A. and Khalifeh, A. (2012) *Cloud Computing and Sustainability: Energy Efficiency Aspects*, PhD Thesis: Halmstad University.

Götz, S., Wilke, C., Cech, S. and Assmann, U. (2011) 'Runtime Variability Management for Energy-efficient Software by Contract Negotiation', Proceedings of the 6th International Workshop Models@run.time (MRT 2011).

Harman, M., Lakhotiaa, K., Singerb, J., Whiteb, D.R. and Yooa, S. (2012) 'Cloud Engineering is Search Based Optimization too', *Journal of Systems and Software*, to appear.

Hedwig, M., Malkowski, S., Neumann, D., Parekh, J., Pu, C. and Sahai, A. (2009) *Taming energy costs of large enterprise systems through adaptive provisioning*, PhD Thesis: Albert-Ludwigs-Universität Freiburg.

Huber, N., Brosig, F. and Kounev, S. (2011) 'Model-based self-adaptive resource allocation in virtualized environments', Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 90-99.

Katsaros, G., Subirats, J., Oriol Fitò, J., Guitart, J., Gilet, P. and Espling, D. (2012) 'A service framework for energy-aware monitoring and VM management in Clouds', *Future Generation Computer Systems*, vol. InPress, p. InPress.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J. and Linkman, S. (2009) 'Systematic literature reviews in software engineering--a systematic literature review', *Information and software technology*, vol. 51, no. 1, pp. 7-15.

Kounev, S. (2011) 'Self-Aware Software and Systems Engineering: A Vision and Research Roadmap', *GI Softwaretechnik-Trends*, vol. 31 (4), pp. 21-25.

Lu, L., Varman, P.J. and Doshi, K. (2011) 'Decomposing Workload Bursts for Efficient Storage Resource Management', *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 5, pp. 860-873.

Noureddine, A., Rouvoy, R. and Seinturier, L. (2011) 'Supporting energy-driven adaptations in distributed environments', Proceedings of the 1st Workshop on Middleware and Architectures for Autonomic and Sustainable Computing, 13-18.

Noureddine, A., Rouvoy, R. and Seinturier, L. (2012) 'A review of middleware approaches for energy management in distributed environments', *Software: Practice and Experience*, vol. 00, pp. 1-30.

Rogers, D. and Homann, U. (2008) 'Application patterns for green IT', *The Architecture Journal*, vol. 18, pp. 16-21.

Sekhar, J., Jeba, G. and Durga, S. (2012) 'A Survey on Energy Efficient Server Consolidation Through VM Live Migration', *International Journal of Advances in Engineering \& Technology*, vol. 5 (1), pp. 515-525.

Sevalnev, M. (2012) *Applying queueing theory to computing cluster energy optimization*, Master Thesis: Aalto University.

Villegas, D., Bobroff, N., Rodero, I., Delgado, J., Liu, Y., Devarakonda, A., Fong, L., Masoud Sadjadi, S. and Parashar, M. (2012) 'Cloud federation in a layered service model', *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330-1344.

Wu, Z., Cotterell, M.E., Qin, S., Beach, A. and Santiago, G. (2012) 'Towards a Cloud Infrastructure for Energy Informatics', Energy Informatics. Sprouts: Working Papers on Information Systems.

Xiong, N., Han, W. and Vandenberg, A. (2012) 'Green cloud computing schemes based on networks: a survey', *Communications, IET*, vol. 6, no. 18, pp. 3294-3300.

Xu, L., Tan, G., Zhang, X. and Zhou, J. (2011) 'Energy aware cloud application management in private cloud data center', Cloud and Service Computing (CSC), 2011 International Conference on, 274-279.

Xu, L., Tan, G., Zhang, X. and Zhou, J. (2012) 'A BDI agent-based approach for Cloud Application autonomic management', Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, 574-577.