Green Cloud Computing: An Energy-Aware Layer in Software Architecture

Rasoul Beik
Computer Engineering Department
Islamic Azad University – Khomeinishahr Branch
Isfahan, Iran
beik@iaukhsh.ac.ir

***Abstract*— Emerging cloud computing has caused data centers consume more energy and therefore there is more CO_2 emission. Using resources more efficiently can help reducing energy consumption in data centers. There are different hardware and software solutions and technologies such as virtualization, using hardware with lower energy consumption, and implementing efficient software algorithms to optimize energy consumption. Considering new architectural approaches in softwares which are run on the cloud is one of the most important methods to achieve green cloud computing. A Software can be implemented as a set of loosely coupled services and components that can be shared between different softwares. This paper proposes an energy-aware layer in software architecture that is responsible for evaluating micro-metrics and macro-metrics of energy consumption in data centers and then makes services to migrate to hosts consuming energy more efficiently.**

Keywords-green cloud computing; software architecture; energy consumption; energy consumption metrics

I. INTRODUCTION

Cloud computing is growing; some features like convenient and on-demand access to shared resources have caused many computing services to move to the cloud, and cloud computing has to scale-up dynamically to support new demands without losing quality of service. Scale-up means more power is consumed by newly deployed resources and consequently there will be more environmental impacts and CO_2 emissions.

Therefore cloud providers need to use resources on the cloud more efficiently to reduce energy consumption. They use hardware devices consuming less energy; they use virtualization to improve resource utilization; and they design self-optimized software applications to consume energy efficiently.

Two basic metrics categories to evaluate energy consumption in clouds are micro-metrics and macro-metrics. Micro-metrics evaluate the energy consumption of resources especially on CPUs, and macro-metrics evaluate overall energy consumption of data centers.

This paper proposes a software architecture layer that is responsible for automatically evaluating the micro-metrics and macro-metrics, and provides some policies based on these metrics to evaluate energy consumption of cloud sites. This layer is responsible to move services to the sites consuming energy more efficiently. Mobile agents can be used either as a tool to gather data to evaluate metrics automatically or as a method to migrate between hosts and run services.

II. GREEN CLOUD COMPUTING

The goal of being green is reducing carbon emission that causes global warming. The most important reason of CO_2 emission is energy consumption, so reducing energy consumption not only means conserving more energy sources for the future but also means reducing CO_2 emissions. Although emerging new business processes like e-work, e-commerce, and e-learning, which are based on ICT, have caused reduced energy consumption in those fields, growing IT applications needs more ICT infrastructures which cause more energy consumption in ICT area [1].

Most current software applications being used in IT applications do not care about how to use resources; most of them must be run on separate servers rather than sharing servers with each other, because most software designers and architects worry about the conflicts between applications and losing quality of service of applications that are run on shared servers. In this situation, servers consume energy but their utilization is lower than what it can be. Additionally, in this view, adding extra hardware resources is the best way to scale-up software applications to satisfy growing user demands, because the amount of the energy consumed to deploy new hardware does not matter for most software designers and architects; so it means scaling-up most current software applications causes more energy consumption. Therefore optimized usage of resources must be considered in using IT applications instead of using extra resources.

Resource usage optimization is one of the ways being used to reduce energy consumption. Resource usage optimization not only reduces energy consumption but also helps managers to control data center costs; especially they can control server usage because servers are the most important resource in all data centers [2]. It should be considered that business customers need to access efficient applications, and most times the best way to increase efficiency is adding more resources for faster processing, but, as mentioned, it is in contrast with energy consumption optimization. The best solution to use resources more efficiently without losing application

978-1-4577-1964-6/12/$31.00 ©2012 IEEE

performance is designing architecturally efficient applications. Software architects must consider some issues like composition, factorization, functionality, dependencies, and scalability in designing new softwares to use resources more efficiently. Applications must be designed as a set of components that are able to work independently; they must be platform independent and can be configured dynamically [2]. There should be control layer in software architecture which is responsible for managing resources, allocating resources to components, and sharing resources between components to improve efficiency. Reference [3] proposes a dynamic power management layer in software architecture which is responsible for scaling-up or scaling-down resources based on resource demands and resource availability.

On the other hand, in the traditional approach to software, an enterprise has to own everything – network infrastructure, hardware resources, and software resources – to deploy software and run it efficiently, but in this approach there are some drawbacks especially wasting energy. Emerging cloud computing has changed this approach. Cloud computing means service delivery over the Internet. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are three types of services delivered by cloud computing; providers provide these services elastically and based on user demands [4]. In cloud service, clients are not the owner of hardware and software infrastructure; they access software, storage, and computing resources but they do not pay for infrastructure costs [5]. There is a shared pool of resources that can be accessed based on on-demand user request. Rapid provisioning and releasing of resources with the least management effort or service provider interaction are some appealing features of cloud computing. It provides better access to software, computing facilities, and data through the network rather than using premises software and hosting [5]. Resource pooling in cloud computing points that users do not know about the location of resources requested by them; there is a pool of shared resources and each one can be assigned to user's request by cloud [5]. Based on the users' perspective, elasticity is one of the most attractive features of cloud computing [5]; cloud resources can be rapidly scaled-out and scaled-in, so users would not worry about availability of resources. It helps enterprises to scale their IT services as their businesses grow in a cost effective manner. Enterprises do not need to worry about cost of hardware, software, or other IT resources when their businesses are growing fast; they just pay for the services on the cloud by their demands. On the other hand, because providers can provide services for more enterprises simultaneously, they can manage resources more efficiently and therefore overall energy consumption will be optimized. They can share hardware resources, like servers, between more applications, they can design software components that can be shared between applications, and they can scale-up and scale-down resources based on user demands [4].

Virtualization and delivery of computing as services are the most important features of cloud computing [6]. Virtualization is an unreal computing environment created through the hypervisor. The hypervisor is a software which can manage several varied computing platforms running concurrently. Each client has its own unique copy of the selected platform and hypervisor passes control to a specific client instance based on demands [5]. Virtualization is an attractive tool for applications to use fewer resources and to utilize servers more effectively [1][2]. As mentioned in [7], there are four levels in the green maturity model for virtualization. Level 0 points to traditional design and may cause low efficiency. Level 1 introduces the idea of sharing applications; it is based on client/server and N-tier technologies. In level 2 data center virtualization is introduced; dedicated hardware infrastructures are eliminated and virtualized servers and storages are used. Last level, level 3, is cloud virtualization; too many organizations share the same infrastructure. Cloud virtualization can improve energy consumption by reducing infrastructure resources; it can be more efficient if new features, like resource sharing, are considered in software architecture.

As mentioned above, Software as a Service (SaaS) is one of the service delivery methods in cloud computing. SaaS is a software delivery model which provides business functionality over the Internet or intranet as a service [8]. Reference [5] points that cloud software service is an application, modeled as Software as a Service, that can be accessed through the Internet.

These kinds of applications are different from single function applications; they are composed of several components that maybe reside on different Internet locations. Cloud service architecture is responsible for orchestrating components to achieve applications that can be shared between users to provide more effectiveness and efficiency with respect to traditional applications. Necessity, usability, reliability, and scalability are the most important features of cloud service.

III. THE ENERGY-AWARE SOFTWARE LAYER

There are different solutions to reduce energy consumption to achieve green cloud computing. Two main solution categories are hardware solutions and software solutions.

Hardware solutions try to design and manufacture devices consuming less energy without losing their performance. Reference [1] mentions some mechanisms to powering down processors and reducing energy consumption in memory. Running servers at a level lower than their utilization in data centers is one of the main causes of consuming energy inefficiently. Energy efficient hardware and power minimization in servers and networks are suggested solutions to cure that problem [6].

In software solutions, virtualization is one of the best technologies for consuming energy efficiently in data centers. Additionally, designers and architects try to deploy applications which use hardware resources more efficiently. Developing energy-aware scheduling and smart traffic routing algorithm are some proposed solutions [1][6]. There are different types of devices that are used in a network infrastructure of a cloud, and how much energy each type of device consumes depends on user behavior and the type of the applications which are run on a cloud and communicating with each other. Reference [6] classifies applications in three types, models the amount of energy consumed by each type, and determines conditions to reduce energy consumption. Keeping quality of service at an acceptable level while computing services are growing on the cloud is a problem that leads to deploy new software architectures which can optimize themselves dynamically [9]. Cloud computing caused enterprises do not need to own everything; they can rent what they need instead of buying and owning them. Softwares should be deployed and run on shared resources; they cannot be tied to a specific server; they should have a minimum dependency to hardware and the location of hardware. Software architects design softwares that are composed of components as services which can be run independently. In addition to be loosely coupled, these components can interact with each other, can be run on separate locations, and can be shared between different applications.

A. Berl et al in [1] mentions that:

> *"In a business environment based on cloud computing, workflows that run over many sites will tend to be popular. Thus, developing methods that map the workflow onto resources under the constraint of energy optimization becomes a central problem of great value and novelty."*

And in another paragraph this is added that:

> *"Energy-related problems have to be solved according to defined policies without needing human interaction."*

Defined policies can be based on metrics determining energy consumption. These metrics can be categorized as micro-metrics and macro-metrics. Micro-metrics are power-efficiency metrics like EPI (Energy per Instruction), performance/watt, MIPS/watt, and IPS/watt in microprocessors [10]; they can be used to determine the amount of consumed energy when a service is run by a CPU. Less energy consumed to run a service by a CPU, better energy-aware is that CPU.

In data centers, overheads like cooling devices consume energy which is generally wasted. Macro-metrics are metrics which are defined to determine overall energy consumption in data centers. For example, PUE (Power Usage Effectiveness) can be used to benchmark the energy consumption efficiency in data centers [7]. Some factors like location of data center, source of energy consumed by data center, and cost of energy affect these kinds of metrics.

Based on defined micro-metrics and macro-metrics, an energy-aware layer is responsible for locating the best site where a service can be run with respect to consuming energy efficiently. Fig. 1 shows overall architecture and the position of energy-aware layer to run a service. In this manner, the application requests for running a service; then energy-aware layer, which evaluates micro-metrics and macro-metrics of different sites and knows about energy consumption and resource usage level of each site, locates the best site to run the service and finally moves service to selected site; the result of running the service will be sent to the application.

Mobile agents have an essential role in this issue. A mobile agent is an independent program which is able to move between hosts. It has an identity and state and like a process contains threads. Bandwidth requirements reduction, mediocre high latency effects, and robustness in networks are some

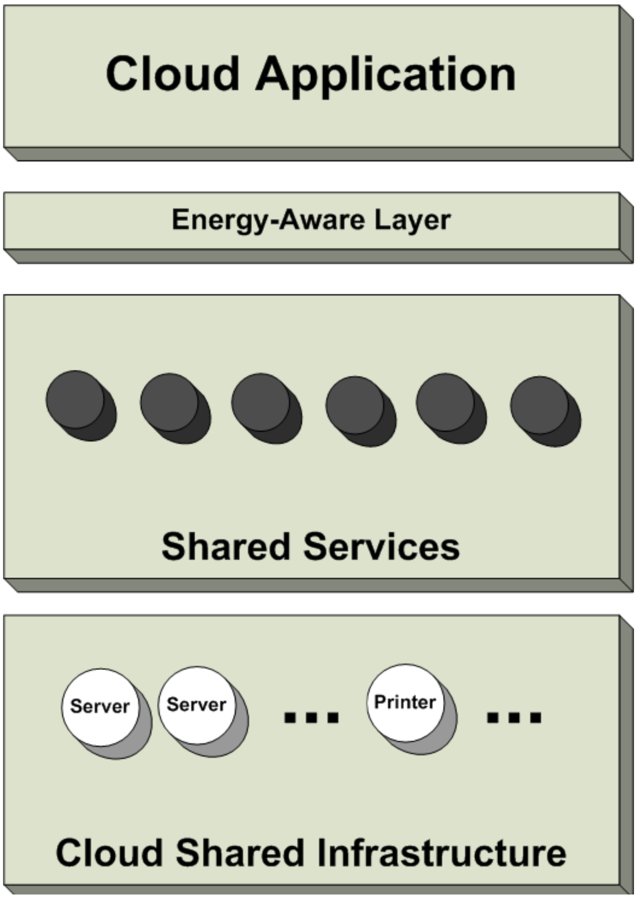

Figure 1. Energy-Aware Layer in Software Architecture

properties of mobile agents [11]. They can be used to measure or specify value of each metric that can be determined automatically. Additionally they can be used to move services to the hosts consuming energy more efficiently. Reference [11] proposes a model of implementing web services using mobile agents, so web service can have dynamic features of mobile agents. In this model, web service requests are mapped to mobile agents which can migrate to other hosts, run on them, and finally return the results to the origin server. As described in [11], the agent service does not need to be installed on hosts; it is installed at the first migrate to the specific host.

IV. Conclusion and Future Works

Although there are different solutions to achieve green cloud computing and consume energy efficiently in data centers, software architects must design softwares that are more aware of energy consumption. Based on some energy consumption policies, which can be determined automatically, resources – especially servers – must be shared between softwares. Micro-metrics and macro-metrics can be evaluated automatically to determine energy consumption in data centers. An energy-aware layer in software architecture can be used to evaluate these metrics.

The following tasks should be accomplished in order to implement mentioned layer:

1. Defining micro-metrics and macro-metrics; and specifying which one can be determined automatically.
2. Weighting micro and macro metrics based on their effects on energy consumption.
3. Using above achievements to specify sites that are more energy efficient.
4. Designing and deploying the energy-aware service layer which is responsible to locate the best hosts consuming energy efficiently and move services and components to them.

REFERENCES

[1] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, "Energy-efficient cloud computing", Comput. J., vol. 53, pp. 1045–1051, September 2010. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxp080

[2] D. Rogers and U. Homann, "Application patterns for green IT", The Architecture Journal, vol. 18, pp. 16–21. [Online]. Available: http://www.architecturejournal.net

[3] G. Rangaraj and R. Bahsoon, "Green software architectures: A market-based approach", The Second International Workshop on Software Research and Climate Change (WSRCC), in affiliation with the ACM/IEEE 32nd International Conference on Software Engineering (ICSE), Cape Town, South Africa, 2010.

[4] A. Spellmann, R. Gimarc, and M. Preston, "Leveraging the cloud for green IT: Predicting the energy, cost and performance of cloud computing", Proceedings of Computer Measurement Group Conference, Dallas, USA, December 2009.

[5] J. Katzan, Harry, "Cloud fotware service: Concepts, technology, economics", Service Science, vol. 1, pp. 256–269, 2009.

[6] F.-S. Chu, K.-C. Chen, and C.-M. Cheng, "Toward green cloud computing", Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication, ser. ICUIMC '11. New York, NY, USA: ACM, 2011, pp. 31:1–31:5. [Online]. Available: http://doi.acm.org/10.1145/1968613.1968651

[7] K. Farncis and P. Richardson, "Green maturity model for vritualization", The Architecture Journal, vol. 18, pp. 9–15. [Online]. Available: http://www.architecturejournal.net

[8] Z. Pervez, S. Lee, and Y.-K. Lee, "Multi-tenant, secure, load disseminated SaaS architecture", Proceedings of the 12th international conference on Advanced communication technology, ser. ICACT'10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 214–219. [Online]. Available: http://portal.acm.org/citation.cfm?id=1831508.1831555

[9] R. Bahsoon, "A framework for dynamic self-optimization of power and dependability requirements in green cloud architectures", Proceedings of the 4th European conference on Software architecture, ser. ECSA'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 510–514. [Online]. Available: http://dl.acm.org/citation.cfm?id=1887899.1887955

[10] E. Grochowski and M. Annavaram, "Energy per instruction trends in intel microprocessors", Technology @ Intel Magazine, pp. 1–8, March 2006.

[11] D. Cooney and P. Roe, "Mobile agents make for flexible web services", Proceedings of The Ninth Australian World Wide Web Conference, 2003.