Energy-aware semantic modeling in large scale infrastructures

Zhu, H.; van der Veldt, K.; Grosso, P.; Zhao, Z.; Liao, X.; de Laat, C.

Publication date
2012

Document Version
Author accepted manuscript

Published in
Work in Progress Sessions (WiP): the 2012 IEEE International Conference on Internet of Things (iThings 2012), the 2012 IEEE International Conference on Cyber, Physical and Social Computing (CPSCom 2012), the 2012 IEEE International Conference on Green Computing and Communications (GreenCom 2012)

Link to publication

Citation for published version (APA):
Zhu, H., van der Veldt, K., Grosso, P., Zhao, Z., Liao, X., & de Laat, C. (2012). Energy-aware semantic modeling in large scale infrastructures. In *Work in Progress Sessions (WiP): the 2012 IEEE International Conference on Internet of Things (iThings 2012), the 2012 IEEE International Conference on Cyber, Physical and Social Computing (CPSCom 2012), the 2012 IEEE International Conference on Green Computing and Communications (GreenCom 2012): 20-23 November 2012, Besançon, France* (pp. 11-14). IEEE.

Energy-aware semantic modeling in large scale infrastructures

Hao Zhu*†, Karel van der Veldt*,Paola Grosso*, Zhiming Zhao*,Xiangke Liao† and Cees de Laat*
* *System and Network Engineering research group*
University of Amsterdam, Amsterdam, The Netherlands
Email: { h.zhu, karel.vd.veldt, p.grosso,z.zhao, delaat } @uva.nl
† *School of Computer*
National University of Defense Technology, Changsha, China
Email:xkliao@nudt.edu.cn

***Abstract*—Including the energy profile of the computing infrastructure in the decision process for scheduling computing tasks and allocating resources is essential to improve the system energy efficiency. However, the lack of an effective model of the infrastructure energy information makes it difficult for the resource managers to provide up-to-date energy metrics of underlying environment. We modeled the relationship between energy components and infrastructure components and defined energy description elements and energy metrics using semantic web technologies, and developed a description model called Energy Description Language (EDL).**

Based on the EDL, we prototyped an basic knowledge system called Energy Knowledge Base (EKB) that provides context-based information retrieval support for applications, in order to make energy aware decisions on scheduling and resource allocation. We validated the logic and effectiveness of the energy ontology by testing the performance of the EKB.

***Keywords*-semantic web; ontology; energy knowledge; energy monitor; energy description language.**

I. Introduction

Energy efficient resource management is becoming an increasingly important requirement for distributed environments such as Grids and Clouds. Besides cost concerns, huge energy use in data centres is beginning to prompt environmental concerns in terms of greenhouse gas (GHG) emissions. Using green energy and improving the energy efficiency of the computing systems are two useful strategies. Switching nodes into low-power mode or scheduling latency-tolerant jobs onto nodes with green energy sources also provide effective mechanisms for reduction of energy cost and GHG emissions. However, all these approaches require that the management components that schedule applications and allocate resources have both the information of the run time state of the infrastructure and of the energy profile on different application characteristics. In essence, there is need for energy-aware resource management systems suited for large scale infrastructures.

The heterogeneity of resources and diversity of Service Level Agreements(SLAs) in large scale infrastructures management make information representation difficult. Semantic Web technologies can play a crucial role in describing distributed resources. They provide mechanisms to integrate different descriptions that are maintained by the various domains' owners. They also enable flexible context based resource discovery instead of traditional keyword based searches. Amarnath et al. [1] introduced a semantic component (ontology) in the conventional Grid architecture to represent the Grid metadata and to offer the required information for effective resources management. However, the concepts in their ontology are not energy-aware and they lack support for information retrieval required by an energy aware management system suitable for Grids and Clouds. Daouadji et al. [2] developed an ontology-based resources description framework for resource allocation purpose with minimal CO2 emissions. Their model is mostly applied for green path selection.

Given the limitations in existing models, we created an energy aware semantic model (the Energy Description Language) that contains the various energy attributes of resources present in distributed large scale infrastructures; we also built a basic knowledge system (the Energy Knowledge Base) that enables context-based resource discovery and information retrieval to support simple power management scenarios. In the remainder of the paper we present the EKB and the EDL, with focus on the basic architecture of the system, on the ontology and on the initial set of services available to management systems.

II. The EKB - Energy Knowledge Base

Our EKB is a resource knowledge system that profiles both the energy information and the underlying resources information. The energy information and resources information are defined by two related ontologies. The energy ontology - EDL - which we present for the first time in this article represents the energy attributes of the various resources; it also captures the relationship between energy components and infrastructure components. The infrastructure ontology used by the EKB is the INDL - Infrastructure Description Language - developed by Ghijsen et al. [3]; INDL describes infrastructure components and network topology and we chose it because of the detailed component descriptions and the high scalability of its model.

The EKB system provides query services that expose the information on the resources and their energy related metrics. Schedulers can invoke these services and take decisions on how to schedule jobs onto discovered resources or how to control energy attributes of resources.

There are several scenarios where an energy-aware management system would make use of our EKB. These scenarios differ by targets and mechanisms of management, and we enumerate here a few:

- *Green path search*: search network paths only containing resources supplied by green energy sources, or search network paths with GHG emissions lower than a threshold.
- *Low power resource selection*: select resources with low power characteristics, *e.g.* solid state disks, low-power processors or energy efficient Ethernet switches when sending a large file from source database to destination node.
- *PowerNap [4]*: identify configurable capabilities of resources to minimise power draw in nap state. The PowerNap is an approach to rapidly switch the whole system into or out of low-power state called nap when activity steps in or out the idle period.
- *Heterogeneous resource discovery*: discover suitable resources meeting energy efficient requirements . For instance, to run I/O intensive jobs, low-power and low-performance resources should be allocated for the energy efficient purpose.
- *Peak power management*: track the maximum power the resource consumes to judge whether beyond the upper bound of power consumption.

These scenarios allow infrastructure managers to reach the target of less emission or energy cost only by relying on resource discovery done by the EKB. In more general scenarios of power management an additional sophisticated power model that allows to predict and evaluate scheduling algorithms is required. Nevertheless, also in these more complex cases, the EKB can aid the prediction of the power characteristics and it can empower the scheduling algorithms to proceed in a consistent and effective manner.

A. System Design

Our knowledge system consists of three main components: a Resource Description Module, a Resources Description Framework (RDF) Repository and Query APIs. The Resource Description Module uses INDL and EDL to describe all the necessary concepts and properties. The RDF Repository stores the RDF triples that contain all the templates and all the instances created by the Resource Description Module. We also define a series of query service APIs in the remote server that can supply information to the clients in the various scenarios.

Management systems or schedulers obtain resource requirement from the users for application execution, and then they act as clients to perform querying of suitable resource discovery. Fig. 1 shows a complete query flow. At first, the client sends a query to the remote server through the defined APIs. The query can be translated into an RDF query on the server side and submitted to the repository. If the instances in repository are not up to date, the repository asks for data update. After receiving the new energy information and resource information from the underlying database, it creates the appropriate instances and fills in the RDF repository. At last, the result can be returned to the client with the latest data. The management system can then use this information to make power-aware scheduling.

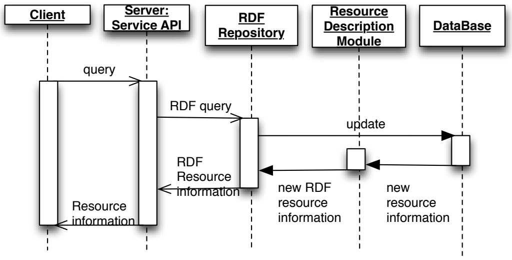

Figure 1: The sequence diagram of processing a query

III. Energy Description Language - EDL

The EDL [1] is the energy ontology we developed; the EDL represents the energy attributes of resources in distributed computing infrastructure and the relationships among them. The EDL allows to instantiate the various energy metrics and their related attributes in a uniform and highly descriptive manner.

We designed the EDL so that it can support the scenarios we enumerated in the preceding section. To support the first scenario, the EDL must include the definitions of the energy sources needed for the reduction of the GHG emissions. In the scenario of peak power management, the power is saved by restricting the upper bound of power consumption. To this purpose the EDL must be able to provide both information on the maximum power used by resources and on the SLA requirements. Finally the EDL must define concepts like capabilities and energy metrics to support the other outlined scenarios.

The EDL can also be used to support more complex scenarios than the ones we sketched out, whenever this is required by the clients of the EKB, *e.g.* schedulers, workflow planners or management systems.

The EDL model is shown in Fig. 2; it contains three main classes: *EnergyMetric*, *MonitorComponent* and *ResourceEnergyDescription*.

[1] The EDL ontology is available at:https://bitbucket.org/hzhu/edl

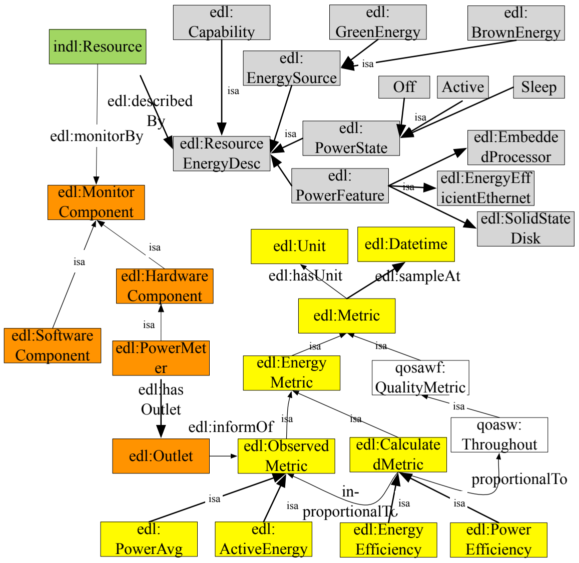

Figure 2: The Energy Description Language - EDL and its three main parts: the EnergyMetric class and subclasses (in yellow), the MonitorComponent class and its subclasses (in orange) and the ResourceEnergyDescription and it subclasses (in gray)

A. *EnergyMetric*

The *EnergyMetric* contributes to the power management from two aspects. As energy constraints constrain the design of modern large scale computer systems, energy metrics are created for SLAs representation. Furthermore, energy metrics are the measurement intended to quantify energy-related properties of resources.

The *EnergyMetric* class defines two types of energy metrics, the *ObservedMetric* and the *CalculatedMetric*, distinguished by the way they are obtained. We get the value of the former directly from accessing a Power Distribute Unit (PDU), while the value of a *CalculatedMetric* is obtained by numerical calculations on the value of an *ObservedMetric* and of a *PerformanceMetric*. The *CalculatedMetric* correlates with *ObservedMetric* and *PerformanceMetric* in the relationship of *InverselyProportional* and *Proportional* respectively. The *PerformanceMetric* objects reuse partially the QoS attributes defined in [5].

Each *EnergyMetric* object has an *energyValue* modeled by *datatype* properties with *xsd:double* type value. Each *EnergyMetric* object is associated with a *Unit* instance according to its physical quantity. In many cases a numerical value alone cannot be understood without its unit type. The *PowerFactor* is the ratio of the real power flowing to the apparent power. The *ActiveEnergy* and the *PowerAvg* represent the energy consumption and the average real power in the sampled time. We also define the *PowerCapping* to represent the maximum power resource consumes. The *EnergyEfficiency* and *PowerEfficiency* depend on the performance of the resources monitored. If the resource is a networking device, *Throughout* may be a suitable performance metric for it. Performance metrics are used to evaluate the energy efficiency of the resources by calculating the performance per unit of energy consumption. The *PowerEfficiency* is a measure of the rate of computation or transmission that can be processed by a computer for every Watt of power consumed. For instance, the Green500 List ranks supercomputers in the Top500 list in terms of FLOPS per watt. Comparably, the *EnergyEfficiency* measures the number of operations or the bytes of data transmission for every joule of energy consumed. The *CalculatedMetric* can be extended by introducing the metrics of the whole infrastructure like Power Usage Effectiveness (PUE).

Although efficiency seems more useful, absolute metrics in the *ObservedMetric* are essential. Energy efficiency can be improved by enhancing the performance even if resources continue to consume large amounts of absolute power. Resources in the idle state can not adequately characterised by just efficiency under load but can be measured by *ActiveEnergy* and *PowerAvg*.

B. *MonitorComponent*

The *MonitorComponent* monitors the value of the metrics. The *PowerMeter* object represents the energy monitor device, usually a PDU. The PDU has a list of outlets that monitor different resources. Each specified *Outlet* instance is responsible for providing the observed energy metrics of the attached node resource. In this sense resources and energy metrics are connected by outlets: we can know which value of energy metric is produced by which resource by reasoning, and vice versa. The *Datetime* class defines the sampling time interval for each observed energy measurement. The *SoftwareComponent* monitors the performance attributes that are not directly available from the hardware monitor.

C. *Resource Energy Description*

The power management system requires the energy descriptions for resource configuration and allocation. Elements in the *ResourceEnergyDesc* contain information on the energy source, on the configurable capabilities the resource has and on how to set them. The *EnergySource* class defines the kind of energy source used by the resources in the distributed system, *e.g.* wind, solar or gas. Each energy class has a corresponding price and amount of GHG emissions per joule; these can be used to calculate the GHG emissions associated to running the resources under a specified scheduling method. The running state of a resource is determined by the *PowerState* class. A management system should in fact have the knowledge on whether the resource is in *Off*, *Sleep* or *Active* state. The *PowerFeature* class indicates which low power components the resources has. Some resources are made up of low power processors like Atom, SSD storage and Energy Efficient Ethernet supporting IEEE 802.3az. The *Capability* tells us the available capabilities of the resource that we can adjust when switching into low power (nap) mode.

IV. Query Service

We designed several query services in the EKB; these match the scenarios we enumerated in II. Our queries allow a client to obtain the following information:

1) the available resources driven by specified energy sources, resources with the low power feature and resource with other description; *e.g.* query parameters could be like (CPU = 2GHz, RAM = 1GB, EnergySource = Solar) and (Switch = Energy Efficient Ethernet).
2) the energy state of any given resources or discover the resources at a range of value; *e.g.* this service can return nodes that consume average power less than 200W with parameters (CPU = 2GHz, RAM = 1GB, PowerAvg $\leq$ 200W).
3) the time when a specified resource is in an given energy state; *e.g.* the query parameter could be (hostname = ”TWIN1”, PowerAvg $\leq$ 200W, Datetime = last 24 hours).

We deployed the EKB on one server of the DAS-4 cluster located at the University of Amsterdam [6]. For testing purposes, we assume that the DAS-4 nodes and the networking devices in this cluster are powered by different kinds of energy sources. The energy consumption of the nodes and the network devices is monitored in real-time and stored as time series in RRD database files. The EKB processes the data from the RRDs and stores the results into a Sesame RDF repository.

In order to test whether the performance and functionalities of the EKB, we made three trial queries to simulate the behaviour of a possible EKB client:

- Query1 is to search which resources use green source ”Solar”;
- Query2 is to query which resources are running with average power in the range from 100W to 200W;
- Query3 is a request of when node ”TWIN3” uses active energy in the range (0, 400) Joule with one-minute time slot.

We evaluated the response time of the different supported queries by varying the database size, *i.e.* the number of triples in Sesame. Fig. 3 shows the average value, and the standard deviation obtained in 20 subsequent runs.

We see that three types of queries have different response time curves. *Query2* needs to search all the resources in the RDF repository, so the number of triples impacts its performance significantly. Conversely, the response time of *Query1* is nearly flat because there it is not influenced by the number of instances. The response time of Query3 increases slowly with the number of triples. The performance of all queries is acceptable with less 500 ms respond time.

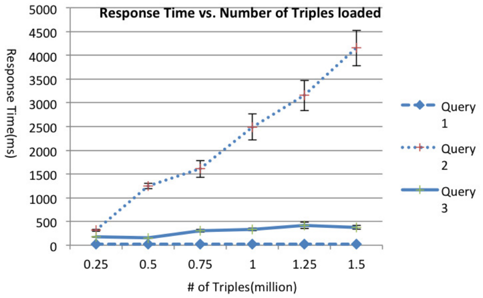

Figure 3: Query Response Times for Various-size Triple-Stores

V. Conclusions and future work

We built an energy information model for large scale infrastructures called the EDL, that provides a consistent semantic description of various energy components; the EDL relies on the descriptions of the infrastructure components provided by the INDL model. We also built an Energy Knowledge Base system, the EKB, that provides support basic power-aware resource management. In the future, we intend to develop more complex queries support in the EKB; we also intend to integrate the EKB with a full-fledged power model to perform flexible and complex power management.

Acknowledgment

This work was supported by NWO through the GreenClouds project, by the Dutch national program COMMIT and by the NSFC No.61272483.

References

[1] B. R. Amarnath and T. S. Somasundaram, “Ontology-based Grid Resource Management,” *Softw. Pract. Exper.*, vol. 39, pp. 1419–1438, 2009.

[2] A. Daouadji and K.-K. Nguyen, “Ontology-Based Resource Description and Discovery Framework for Low Carbon Grid Networks,” in *2010 First IEEE International Conference on Smart Grid Communications*, Oct. 2010, pp. 477–482.

[3] M. Ghijsen, J. V. D. Ham, P. Grosso, and C. D. Laat, “Towards an Infrastructure Description Language for Modeling Computing Infrastructures,” in *10th IEEE International Symposium on Parallel and Distributed Processing with Applications(ISPA)*, Madrid, 2012, pp. 207–214.

[4] D. Meisner, “PowerNap : Eliminating Server Idle Power,” in *ASPLOS09*, Washington,DC, 2009.

[5] Z. Zhao, P. Grosso, R. Koning, J. van der Ham, and C. de Laat, “An agent based planner for including network qos in scientific workflows,” in *MCSIT 2010*, oct. 2010, pp. 231 –238.

[6] The Distributed ASCI Supercomputer 4. [Online]. Available: http://www.cs.vu.nl/das4/home.shtml