CompatibleOne: Designing an Energy Efficient Open Source Cloud Broker

Julien Carpentier, Jean-Patrick Gelas, Laurent Lefèvre,
Maxime Morel, Olivier Mornard
INRIA - LIP Laboratory - École Normale Supérieure of Lyon
University of Lyon, France
Email: {julien.carpentier,laurent.lefevre,jean-patrick.gelas,maxime.morel}@inria.fr

Jean-Pierre Laisné
BULL
Email: Jean-Pierre.Laisne@bull.net

***Abstract*—In this paper, we present the new French Cloud management software called CompatibleOne. CompatibleOne is an open source cloud services broker *i.e.* a cloud service management software with brokering capabilities. CompatibleOne can provision, deploy and manage any type of cloud services, these services being provided by heterogeneous service providers selected according to Service Level Agreement (SLA). CompatibleOne can also federate heterogeneous resources and more precisely integrate seamlessly various cloud services. Given this approach it allows us to exploit original solutions and moreover include at design level some new paradigms like energy efficiency. This paper focuses on the current activities done in the CompatibleOne project for energy monitoring and energy efficient management of Clouds systems.**

***Index Terms*—Cloud computing; Energy efficiency; CompatibleOne.**

I. INTRODUCTION

Over the last few years, virtualization has largely proved to be an useful and efficient technology, since it solves number issues such as spacing problems in computer rooms through consolidation, and allows for a better usage of equipment. The other advantages that has come from virtualization is agility, ease of deployment of new environments, and some fault tolerance aspects.

Today, energy consumption of large scale IT systems represents a significant part of the world's consumed electricity. A lot of distributed systems are currently moving to cloud systems and frameworks. Therefore, applying green computing policy to these cloud infrastructures becomes essential.

A short presentation of first results has been previously presented in [5]; this paper focuses on recent advances in the CompatibleOne project, a french initiative for supporting next generation clouds. We present the COEES : CompatibleOne Energy Efficiency services. This multi layer framework allows the energy monitoring, reporting and exposing of physical and virtual resources. It can be embedded in classical Cloud monitoring infrastructures (like Ganglia [6]). This paper will present how such energy aware approach can be taken into account in order to design an energy efficient open source cloud broker and to finally obtain a complete Green Cloud infrastructure [7].

In this article, after an introduction to the Compatible One initiative (section II), we will focus on the energy monitoring management aspects and we will explain how those aspects are introduced in the framework at the design phase (section III). We then present how the problem of energy efficiency is vital and must be deeply integrated in the Cloud framework (section IV). In section V, we conclude this paper and present some future works.

II. THE COMPATIBLEONE INITIATIVE

CompatibleOne[1] is a research project publicly funded by French Ministery (FUI) to provide open source 'cloudware' for the creation, deployment and management of private, public or hybrid Cloud platforms. This project is led by the french Bull company. There are 11 industrial partners (Activeon, CityPassenger, enovance, Eureka, Mandriva, Nexedi, Nuxeo, Prologue and Xwiki) and two academic partners (INRIA and Institut Telecom). The CompatibleOne project participants intend to provide a complete implementation of the three major services of a Cloud : IaaS, PaaS, SaaS (Infrastructure, Platform, Software as a Service).

CompatibleOne is a cloud resource management software which affords integration of other cloud resources into the cloud services management by providing brokering capabilities. Therefore CompatibleOne is not limited to manage resources but enables utilization of any cloud service selected on criteria described in SLA.

In other words, CompatibleOne is a service delivery platform that manages cloud resources, not just infrastructure but any resource that can be available and delivered through CompatibleOne. This platform allows to combine the various services supplied by different suppliers: this way CompatibleOne provides the capability to federate resources across diverse and heterogeneous Cloud Service Providers (CSP). Thanks to CompatibleOne service management layer based on service oriented architecture, any other resource can be seen as an object. Cloud Service Providers can incorporate such solution into their service delivery platform to add seamlessly new value added services.

CompatibleOne supplies secured environment that encourages developers to create modern cloud aware stateless applications. CompatibleOne facilitates the usage of advanced data

[1] http://www.compatibleone.org/

management infrastructure as required in Big Data projects. Finally CompatibleOne permits to manage those cloud aware applications across multiple CSP with all necessary guarantee from security to governance.

CompatibleOne offers PaaS management capabilities similar to Google Apps or Microsoft Azure but with freedom and openness the developers deserve. PaaS is seen as a simple module which is not deeply welded into the entire infrastructure service management. CompatibleOne open source platform helps developers to manage the run time environment of their choice to develop their application using the CSP of their choice and avoid lock-in. With CompatibleOne, providers have the capability to choose a platform environment and deliver it to their customers.

CompatibleOne is next generation of cloud service management software providing a full run time environment that actually a CSP or an IT department can integrate through run time environment rather than service delivery interface.

With CompatibleOne, complex IT environments can be managed with a software programming model because it's much structured, much easier to manage, and it is easier to maintain. And because the scripting is just getting too much now: An automated efficient reliable infrastructure cannot be managed with scripts all other the place. Hence to have automation imposed by the complex software environment of the future, we need to have a structured approach for managing resources. Therefore CompatibleOne takes advantage of software development methodologies for managing IT environment.

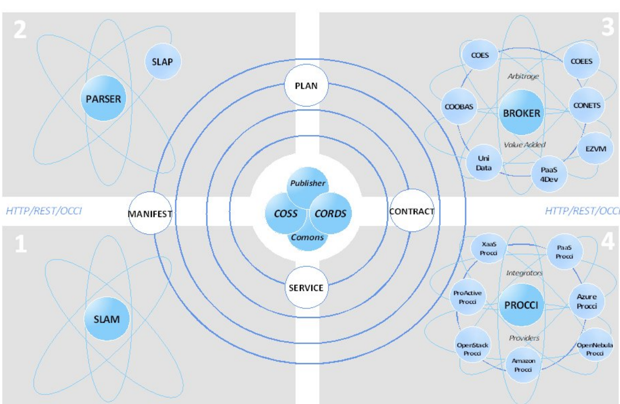

Fig. 1. CompatibleOne global architecture : ACCORDS platform

Following the 4 quadrants of ACCORDS platform (figure 1) :

- Requirements for the provisioning of cloud resources are described using the CompatibleOne Request Description Schema (CORDS) and are submitted to the system in the form of an XML document called the MANIFEST.
- This document is validated and processed by the ACCORDS Parser and results in the production of a fully qualified resource provisioning PLAN. This provisioning plan describes in precise details the operations to be performed for the construction and delivery of the cloud application configuration.
- This provisioning plan may be used at any time for the provisioning of the cloud resource configuration as described by the manifest. This provisioning operation is performed by the ACCORDS Broker working in co-operation with the placement components, COES, and provisioning components, PROCCI, of the platform. The role of placement is that of selecting not only the most appropriate provisioning platform type but also the commercial partner for the provisioning of resources. With an aim to satisfying the needs of various usage, the powerful and flexible algorithms of the placement engine allow their decisions to be taken respecting technical, financial, commercial, geographical, performance and quality of services based criteria
- In the provisioning corner, heterogeneous provider platforms will be engaged, as required, for the deployment of the applications and hardware required to satisfy the configuration as described by the manifest. When working with predetermined quotas, that will have been negotiated in advance, failure of any particular provider or partner to deliver will be fed back to the placement engine for selection of alternative providers and allowing for not only fail over management but also for the real time assessment of quality both operational and commercial of the involved parties.

These major operational components are interconnected with and through a collection of service components in a very loosely coupled and consequently flexible fashion. This allows support for the different usage scenarios to be easily achieved by the integration or replacement of certain operation specific components. Each operational concept, and consequently component, of the platform may be implemented in the form of an individual standalone service management platform and consequently offers for unlimited scalability. The generic provisioning interface, provided by the components comprising the ACCORDS PROCCI, allows for extension of the platform by the dynamic and real time addition of heterogeneous provisioning components provisioned as input manifests. This possibility may be used to spawn subsequent instances of the platform itself in order to meet increased needs for provisioning in a flexible and elastic fashion.

This framework can be deployed by companies who have doubt about other solutions and it is simple to use. One feature of CompatibleOne is to unify, integrate and hide differences with existing platforms. As CompatibleOne considers these energy consumption issues at the design stage, it aims to provide an efficient solution to this problem.

III. COMPATIBLEONE ENERGY MONITORING FRAMEWORK

Adapted energy monitoring to virtualized infrastructures is a pretty new and complex subject. Our work is divided between a constant monitoring of technological development around green IT and cloud computing and implementation of the CompatibleOne energy modules. We develop a system to collect energy consumption information that enables access to information on the energy consumed by physical elements used in the cloud, as well as the evaluation of the energy bill, and the environmental impact.

In CompatibleOne context, energy monitoring provides an additional SLA criterion for Broker choice in virtual machine placement, a precise consumption billing and energy efficiency for Cloud Service Provider with live energy monitoring of their platforms.

An abstraction of energy collect is available through HTTP REST primitive. According to energy usage, some modules need an important precision and others just an average consumption for energy usage prediction or define Cloud Service Provider a ”green” reputation score.

A. Probes specifications

For information gathering, we first target physical probes because the system needs to get precise energy logs every couple of seconds.

We studied and used different industrial probes in terms of precision and measures frequency abilities. We list in this section the main wattmeters equipments we use in the CompatibleOne experience :

1) Eaton ePDU: [1] (Figure 2)
Voltage : 230 V
Current : 16 A
Outlets : C13, C19
Interface : serial, ethernet (SNMP)
Measure frequency : 1 value every 5 seconds
Measure precision : 1 W

Fig. 2. Eaton ePDU

2) Schleifenbauer ePDU: [3] (Figure 3)
Voltage : 230 V
Current : 16 A
Outlets : C13, C19
Interface : ethernet (SNMP, Modbus, MySQL)
Measure frequency : 1 value every 3 seconds
Measure precision : below 0.1 W

Fig. 3. Schleifenbauer PDU

3) Dell iDRAC6 - IPMI: [4] (Figures 4 and 6)
Measures : internal sensors (Dell proprietary sensors)
Interface : IPMI
Measure frequency : 1 value every 5 seconds
Measure precision : 7 W

Fig. 4. IPMI add-on card

4) Wattmetre OmegaWatt: [2] (Figure 5)
Interface: serial only (RS-232)
Measure frequency :1 value per second
Measure precision: 0.125 W

Fig. 5. Wattmetre (OmegaWatt)

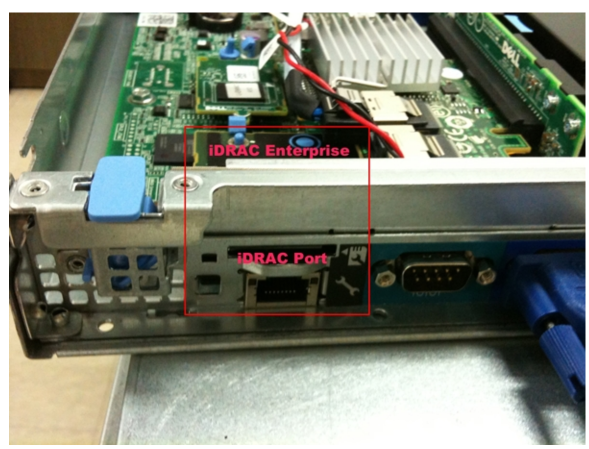

Fig. 6. IPMI add-on card embedded in server

B. Probes and Collector daemons

Eaton and Schleifenbauer managed Power Distribution Unit (PDU) are two of our targeted choices as we get all information we need like instant consumption, power factor, efficiency through Simple Network Management Protocol (SNMP). Intelligent Platform Management Interface (IPMI) energy collection through dedicated add-on cards is also included into our solution because it is currently the most deployed and embedded solution in infrastructures with energy monitoring facilities (see Table I).

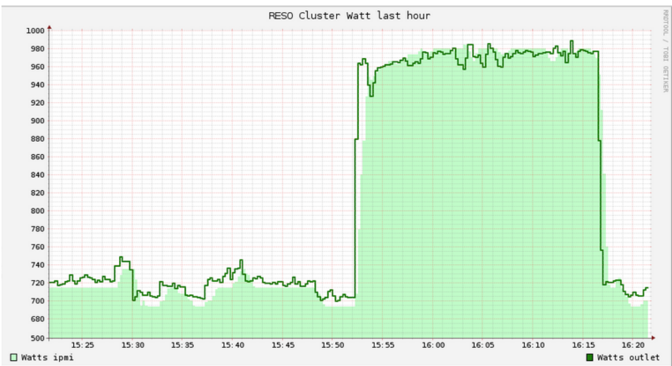

Fig. 7. Precision benchmark with IPMI and Outlets measures

However, precision with IPMI is below outlet measurements as shown on this energy consumption graphic where dark green curve represent outlet measure and light green zone represents measurements collected with IPMI (figure 7).

C. Monitoring System & Implementation choices for COEES

In this section, we present the design of COEES : Compatible One Energy Efficiency Services.

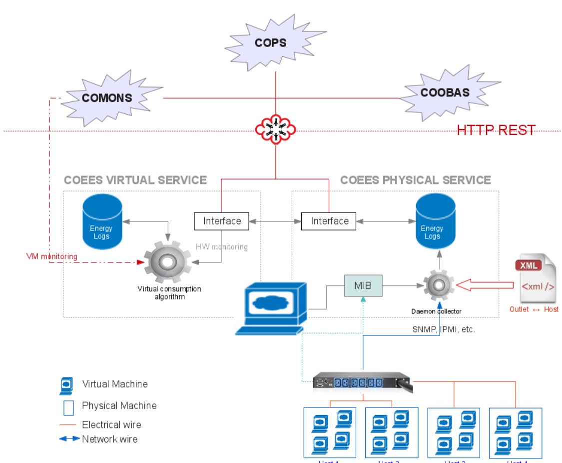

Fig. 8. Energy Monitoring Modules Architecture

1) General Architecture: On figure 8 we highlight the necessity to split hardware monitoring and virtual machine monitoring provisioned on-demand through Cloud Service Provider IaaS system like OpenStack or OpenNebula.

Indeed, physical host consumption comes directly from probes through daemon collector request because we only need a mapping between physical host and outlet and network topology is pretty stable. Virtual host life is based on user needs and we need to keep a trace of virtual host user, physical host consumption and context (CPU usage, IO, etc.) to estimate virtual host consumption.

COEES physical and virtual services are both executed on a specific virtual machine hosted by Cloud Service Provider in order to reach each probes and collect data. This virtual machine is deployed after CompatibleOne platform is launched if COEES service is running. Monitoring systems are time-based and large scale so it is crucial to use common time to link events with timestamp, so we choose POSIX time (ISO8601).

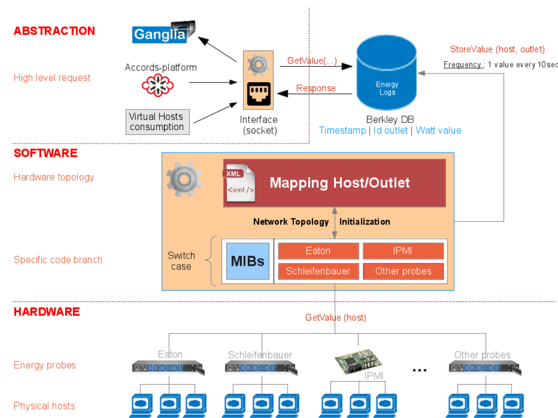

Fig. 9. COEES Software Architecture

2) Software Architecture: Energy monitoring system in CompatibleOne is designed on different layers (Figure 9). These layers provide modularity and abstraction.

First layer concerns the monitored physical equipments. Just above we deploy the energy monitoring probes, which can represent an heterogeneous set of sensors (PDU or IPMI add-on cards which come from different vendors).

So in the next layer, we instantiate a collecting daemon which can handle all these probes. Its aim is energy consumption values gathering from the probes below. This daemon must run on a physical machine reachable by Cloud Service Provider in order to reach probes through network. (local network or Virtual Private Network (VPN))

Provider administrators must fill a XML (eXtensible Markup Language) file describing energy monitoring topology: mapping between machines (hostname or ip) and outlets (see an example on Fig. 12). For each machine, this mapping gives us the information on which probe it is plugged and all information needed to communicate with this probe. Each type of probes could own different access protocols. For example, some PDUs use SNMP protocol. Each vendor use its specific Management Information Base (MIB), so we need to know the Object Identifier (OID) name to have the required value, like watts. (Figures 10-11)

TABLE I
ENERGY MONITORING SYSTEMS

Product	Interface	Pros	Cons
OmegaWatt	Serial	known probes (GRID'5000 experimental platform)	Not scalable
Eaton	SNMP v1	Outlet measure integration, precision	-
Schleifenbauer	SNMP v1	Outlet measure, integration, precision	-
IPMI Addon card	IPMI Interface	Generic in most one rack server	precision

Some PDU do not implement directly watt value and we have to compute it using voltage, current and power factor value $P_{Watt} = U(volt) * I(ampere) * Pf(powerfactor)$

So, using all these mapping informations, we can use our generic function "GetValue(host)" which will handle all the specificities of the probes below. Next, we store the data retrieved from the probes using a function StoreValue(host, outlet) " in a Berkeley DB (Berkeley DataBase) using this structure :

```
Timestamp | id_outlet | watt value
```

As database in Berkeley DB is a simple file it can be reopened by the next layer of the system which handles the high level requests. This is done using another software daemon. The goals of this layer are to answer high level request or to export data to other services like exposition to Ganglia or virtual machine consumption service. High level request are coming from the other modules of CompatibleOne. For example elasticity or billing services which will ask about energy consumption. The communication is made using an Open Cloud Computing Interface (OCCI).

IV. INTERFACING WITH CLOUD MONITORING SYSTEMS

A. Presentation & Usage

Ganglia [6], open-source project that grew out of the University of California, is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters (figure 13). It leverages widely used technologies such as XML for data representation, eXternal Data Representation (XDR) for compact, portable data transport, and Round Robin Database tool (RRDtool) for data storage and visualization.

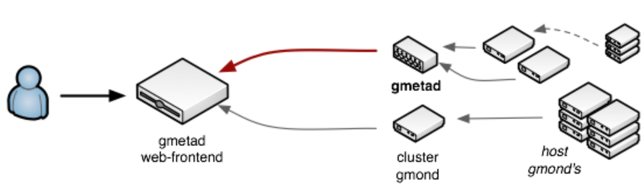

Fig. 13. Ganglia Monitoring System architecture

In our context, we use it as management and verification tool for energy data stored in database concerning virtualized host in cloud environment. This tool ensure possibility for an administrator to display aggregated values per days, weeks or months on single-node or a whole cluster (Figures 14-15).

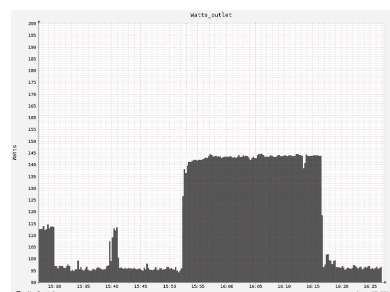

Fig. 14. Single-node watt consumption with Ganglia

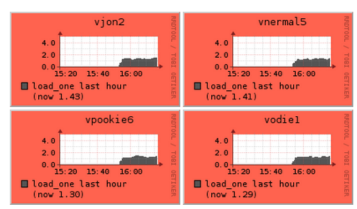

Fig. 15. Multi-node watt consumption with Ganglia

B. Technical operation

The Ganglia Metric Client (gmetric) announces a metric value to all Ganglia Monitoring Daemons (gmonds) that are listening on the cluster multicast channel. Gmetric has the ability to spoof metrics for a host that is not running gmond. This is particularly useful in our case because we need to collect only one value on different probes on the same host who run CompatibleOne Energy Efficiency Services. Thanks to this process, energy data is stored in ganglia as if all hosts give themselves energy values.

Last part consists of some frontend modifications in order to integrate new metric and associated graphs in PHP (Hypertext Preprocessor) script and HTML (HyperText Markup Language) pages.

```
U = snmpget -v 1 -c public 192.168.0.203 actualVoltageO.6.1
SPGW-MIB::actualVoltageO.6.1 = INTEGER: 232.8
I = snmpget -v 1 -c public 192.168.0.203 actualCurrentO.6.1
SPGW-MIB::actualCurrentO.6.1 = INTEGER: 0.4
Pf = snmpget -v 1 -c public 192.168.0.203 powerFactorO.6.1
SPGW-MIB::powerFactorO.6.1 = INTEGER: 86.9 %

Watt value = 80.82
```

Fig. 10. Schleifenbauer SNMP example

```
U = snmpget -v 1 -c public 192.168.0.202 outletVoltage.19
SNMPv2-SMI::outletVoltage.19 = INTEGER: 247
I = snmpget -v 1 -c public 192.168.0.202 outletCurrent.19
SNMPv2-SMI::outletCurrent.19 = INTEGER: 0.481
Pf = snmpget -v 1 -c public 192.168.0.202 outletPowerFactor.19
SNMPv2-SMI::outletPowerFactor.19 = INTEGER: 80 %

Watt value = 95.04
```

Fig. 11. Eaton SNMP example

```
<cluster name="RESO">
     <probe model="schleifenbauer" ip="192.168.0.203">
      <mib filename="33_0_FR_SPGW_MIB.mib">
        <field name="voltage"       node="actualVoltageO"      type="integer"/>
        <field name="current"       node="actualCurrentO"      type="integer"/>
        <field name="powerfactor"   node="powerFactorO"    type="integer"/>
      </mib>
<hosts>
     <host ip="10.5.5.8" outlet="1"/>
     <host ip="10.5.5.9" outlet="2"/>
</hosts>
     </probe>
     <probe model="eaton" ip="192.168.0.202">
<mib filename="EATON.mib">
<field name="voltage"      node="outletVoltage"           type="integer"/>
<field name="current"      node="outletCurrent"           type="integer"/>
<field name="powerfactor"   node="outletPowerFactor"    type="integer"/>
</mib>
<hosts>
     <host ip="10.5.5.10" outlet="19"/>
     <host ip="10.5.5.11" outlet="20"/>
</hosts>
</probe>
</cluster>
```

Fig. 12. Outlet/Hosts mapping XML

C. Benchmark example

Ganglia allowed us to observe virtual host consolidation experimentation with a simple case where five virtual hosts run on a single physical host (blue thick circle) and on the other hand five virtual hosts run on five different physical hosts (red round) (figure 16). Each virtual machine embeds and runs a CPU intensive application in order to visualize the impact on energy consumption. In our case, all physical hosts are kept running, so idle energy consumption isn't considered and saved. As we can see, COEES allow designers to observe their energy efficiency improvements when consolidating some virtual hosts on the same physical resources.

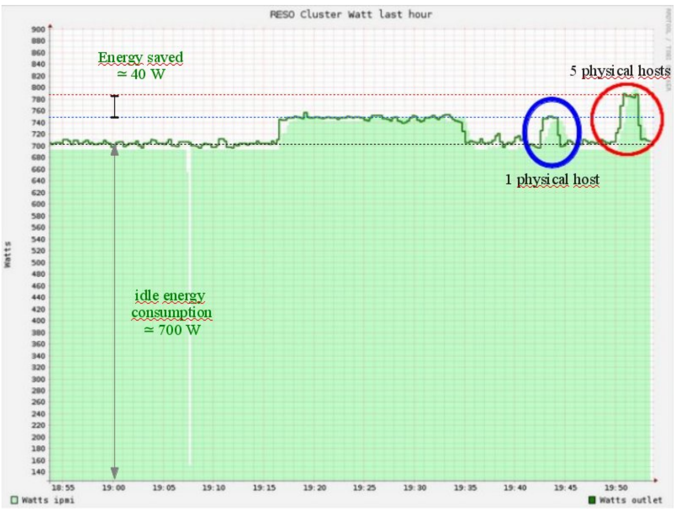

Fig. 16. Ganglia virtual host energy consolidation experiment

V. Conclusion

This paper presents briefly the design of the CompatibleOne Energy Efficiency services and the first experimental results obtained from the CompatibleOne energy monitoring framework. Primarily targeted to monitor Cloud physical resources and to support physical wattmeters through some heterogeneous requesting frameworks (SNMP, IPMI..); COEES will also deal with virtual resources energy consumption. The CompatibleOne approach relies on open source technology and will be available for research communities and companies involved in cloud development. The proposed approach will allow Green Clouds designers to develop their own frameworks of energy aware and energy efficient cloud components.

VI. Acknowledgments

This work is supported by the CompatibleOne project which is co-funded by the *French Direction Générale de la Compétitivité, de l'Industrie et des Services, Fonds Unique Interministériel, Région Ile de France, Conseil Général des Yvelines* and *Mairie de Paris*.

References

[1] Eaton epdu specifications. http://powerquality.eaton.com/ePDU-PW104MA0UC34.aspx?CX=53.

[2] Omegawatt wattmetre specifications. http://mv.omegawatt.fr/en/index.php.

[3] Schleifenbauer epdu specifications. http://www.schleifenbauer.eu/en/in-rack-pdu_tech-specs.htm.

[4] Brad Lawrence and John Jenne. Dell energy smart architecture and power management adoption. Technical report, Dell, November 2011.

[5] Laurent Lefevre, Olivier Mornard, Jean-Patrick Gelas, and Maxime Morel. Monitoring energy consumption in clouds: the compatibleone experience. Poster in Cloud and Green Computing conference, Sydney, Australia, December 2011.

[6] Matthew L. Massie, Brent N. Chun, and David E. Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(5-6):817–840, 2004.

[7] Anne-Cécile Orgerie, Marcos Dias de Assuncao, and Laurent Lefèvre. *Energy Aware Clouds*, chapter Chapter on book "Grids, Clouds and Virtualization" - Massimo Cafaro and Giovanni Aloisio editors, pages 145–170. Springer Book, October 2010. ISBN : 978-0-85729-048-9.