ISPA 2010 - IEEE Intl. Symp. Parallel and Distributed Processing with Applications. Taipei, Taiwan. Sep 2010.

Energy Measurement for the Cloud

Yi Yu, Saleem Bhatti
School of Computer Science, University of St Andrews
St Andrews, Fife, KY16 9SX, UK
Email: {yi,saleem}@cs.st-andrews.ac.uk

***Abstract*—One of the aims of cloud-based services is to offer cost savings through *elastic* service provision. This elasticity refers to use of resources by the customer and the provision of resources by the provider. An increasingly important resource to consider is *energy* (or *power*). As cloud services are intended to be 'always on' the energy costs of cloud service provision is already significant, and will continue to rise as global energy prices continue to rise and more users make use of cloud services. While providers can invest in energy-efficient hardware, how can we make *users* (*customers*) energy-aware, and incentivise users towards energy-efficient use of cloud systems? Clearly, being able to measure actual energy usage will allow the provision of suitable feedback to users, as well as enable new energy-aware systems metrics that allow systems management policies to become responsive to energy usage. We list the requirements and described a prototype for such an energy measurement system.**

I. INTRODUCTION

Today, cloud services are widely used and their use is increasing: as a measure of the growth (and presenting the commercial incentive for the growth), it is estimated that revenue from cloud services will rise from $56B in 2009 to $150B in 2013 [3]. Meanwhile, electricity consumption is set to rise 76% from 2007-2030 and prices will continue to rise as raw fuel costs rise, as CO_2 costs are applied, and as generation prices increase [6]. Overall, the combined power usage by ICT will triple in the period 2008-2020 if "current trends go unchecked" [16]. The costs of cloud services due to energy usage will rise in the future: Microsoft has publicly stated that it believes that by 2015, "costs to operate servers will exceed the costs to purchase server hardware" [11].

A. Green Cloud (Dis)Incentives

Clearly, if the energy usage of a cloud can be monitored, there is the potential to introduce system management policies that will allow energy usage to be used in management practices. However, this is not the general practice today: no common energy measurement infrastructure exists [15].

We take the position that, in today's ICT environment, the incentives are wrong for wide-scale implementation of energy-saving policies, and useful information is not available from deployed systems to help business policy makers and system managers move towards energy efficiency.

Firstly, energy-saving equipment and energy-saving tools often add additional cost to basic ICT purchases for end users, e.g. lower power CPUs in server equipment are optional 'upgrades'. Where total cost of ownership (TCO) considerations are made for purchases, these usually do not include energy-usage estimates, and operational and/or functional requirements to the business objectives are key. Furthermore, the installed base of (legacy) equipment may not have suitable hardware for energy monitoring, let alone permit energy control. Even when money can be found for purchase and installation of energy-usage monitoring hardware (e.g. for legacy systems), there is lack of an *information model* that permits appropriate collection of energy-usage and resource-usage data for devices within an administrative domain.

Secondly, for new equipment, established vendors currently use proprietary energy-saving features to compete for sales and maintain their customer base. So, there is little incentive for the established vendors to co-operate and agree cross-platform, vendor-independent, energy-aware information models. As very few ICT environments are single-vendor provisioned, this means that there are a disparate set of devices and components, which may be energy-aware, but whose energy usage is not easily visible to systems managers in an easily accessible or consistent manner. This prevents rationalisation of systems management for greater energy efficiency.

Commercial providers such as EnergyICT[1] offer proprietary services for energy management, but these may be expensive to deploy, and may not be easily integrated into existing systems management infrastucture.

What are the correct data and incentives for business policy makers and managers? For the UK, an NCC survey report [12] makes clear some key issues:

- Only 13.4% of organisations monitor power consumption, and there is little knowledge or experience about being green.
- Legislation, current and proposed, is *not* a major incentive to be green: cost savings are the *overwhelming* incentives.

B. Motivation

We take the position that providing a detailed energy-usage measurement infrastructure can benefit cloud providers:

- Direct costs savings: by allowing management policies to make use of energy usage information, system management policies can become adaptive to energy usage.
- Incentivising customers: if energy usage information can be integrated into customer service level agreements (SLAs) then it is possible to offer even more flexibility in costs to users: if a customer is willing to allow more

---

[1] EnergyICT http://www.energyict.com/

energy-efficient systems operation (perhaps, for example, by some reduced performance), the cost gain due to energy savings could be split with the customer.

There already exist commercial solutions for adapting system response to energy usage[2], as well as on-going research into flexible SLA provisioning through cloud federation [2]. So, if wide-scale energy measurement can be enabled and reflected in the customer SLA and integrated into systems management policy, then there is a win-win situation for cloud providers and cloud customers.

For the remainder of this paper: Section II describes the requirements of energy measurement for the cloud; we present the architecture of our proposed *Scaleable Energy Monitor* in Section III; a prototype as proof-of-concept is described in Section IV; Section V discusses two possible applications of our work for cloud providers; Section VI briefly describes the existing energy measurement metrics used for data-centres and the challenges for energy-aware deployments; Section VII concludes the paper and lists future work.

II. REQUIREMENTS

From our discussion above, we propose that it is beneficial to have energy measurement within the cloud. We chose to pose the following questions in order to help us in determining and justifying our requirements:

1) How can we gather energy information on a system-wide basis, at scale, including heterogeneous devices and infrastructure within a cloud?
2) What metrics and Key Performance Indicators (KPIs) are suitable for use in system management policies and SLAs?
3) What effects are there on system operations and performance when energy information is included into system management policies?
4) How can we provide feedback to customers (users) in order that they can have confidence in the operation of management polices and SLAs that incorporate energy-usage measurements?

These are complex research questions: we address in detail in this paper only the first issue, and provide some discussion of the other issues. Rather than take a top-down approach (e.g. as in [15]) or a tightly-coupled approach (e.g. as in [4]), we choose to take a very practical, loosely-coupled, bottom-up approach to our provision of energy usage measurement. This will allow maximal flexibility for enabling energy measurement in existing heterogeneous systems, as well as new systems. We are developing an architecture through an iterative refinement process, informed by the development of a prototype in parallel. For this prototype, we are currently concentrating on six issues which would be important for a widely applicable monitoring and management system. Firstly, as a simple set of guiding principles:

- *Legacy*: some legacy equipment may not have native power monitoring capability so we need to be able to integrate external devices and sensors for energy measurement.
- *Vendor-independence*: for new equipment, established vendors currently use proprietary energy-saving features, so we need a vendor-independent information model.
- *Scale*: as there may be potentially many different systems, sub-systems and components to manage, including support infrastructure (such as cooling systems), we need to have an energy measurement system that can scale to large numbers of monitored units.

From a technical and pragmatic view-point, our work so far has considered the following issues:

- *Identity*: we need to identify resources in a systematic manner, so that energy usage can be linked to specific systems, sub-systems or components. There are several possible systems that could be used for naming in this context, e.g. SNMP[3]/ASN.1[4] object identifiers (OIDs) or Universally Unique Identifiers (UUIDs) [8].
- *Heterogeneity*: different sensors already exist and need to be incorporated into the measurement system. Local installations may have local communication constraints that must be overcome to permit energy measurements to be made visible beyond a device or sensor.
- *Integration*: rather than insist that existing systems management practices be completely re-oriented towards energy usage, we require that energy information should be integrated into existing management and monitoring systems (e.g. *nagios*[5], SNMP systems, Eucalyptus [13]). Creating completely new and separate architecture for energy management would raise barriers to integration of energy information into existing systems management capability, and so inhibit systems energy awareness.

With these issues in mind, we describe the functionality and design of our current architecture and prototype for energy measurement for the cloud.

III. ARCHITECTURE

Our *Scaleable Energy Monitor (SEM)* is being refined with direct experience from our prototype (Section IV). We present them separately in order to describe the main philosophy of our intent, especially with respect to the principles of Legacy, Heterogeneity and Scale described in Section II.

We define SEM with three functional components: the *Agent*, the *Collector*, and the *Relay*. The Agent and Collector process and/or transform energy-related information; the Relay assists with communication functions. Figure 1 describes an example instance of the architecture. The *Agent* hides the current heterogeneity of the raw energy information measured from the *real resource (RR)* (i.e. device, computer, etc). An Agent provides the energy usage information from the RR, and allows energy-related control actions from the Collector (ultimately from an Application) to be applied to the RR. The

---

[2] VMware vSphere http://www.vmware.com/products/drs/

[3] http://www.snmplink.org/

[4] http://www.asn1.org/

[5] http://www.nagios.org/

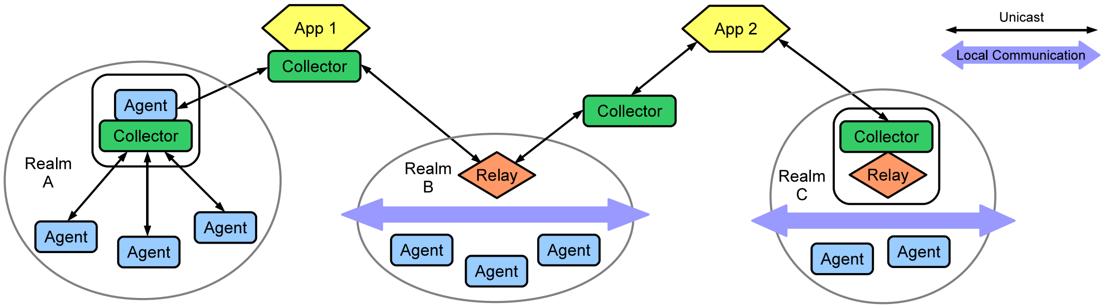

Fig. 1. An example of the architecture showing different scenarios and also the use of co-located functions to form hybrid units.

Agent could be embedded into the RR, but could also be loosely-coupled, e.g. a daemon on a computer which connects to the RR device or energy sensor via a serial line. Agents are organised in *Realms*, which are conveniently-defined (administrative or technical) domains that are organised within the context of the system being monitored.

The *Agent* identifies the RRs for which it provides energy information. Note that the energy usage information presented could be a single value for a single device, but could also be aggregated or summarised. For example, in Figure 1, for Realm A, there are three single Agents, and a hybrid *Agent/Collector*. The single Agents could be embedded each onto a blade or line-card in a chassis, and the Agent in the hybrid unit represents the energy usage of the whole chassis. Agents may poll the RR and cache energy information (with a local timestamp), or may fetch the information on demand, as queries arrive from a Controller. An Agent cache is recommended in case a Collector is not present or inactive, but is not required.

The *Collector* function has three roles: (i) to collect energy usage information from Agents; (ii) to control and configure Agents; and (iii) to pass on to the Agent energy-related actions to invoke upon the RR. A Collector will cache information received from Agents. Control and configuration actions on Agents include: start/stop polling the RR; change polling intervals; and send power management *actions* to the RR, e.g. go to standby, sleep, power down, etc.

The *Relay* hides the heterogeneity in communication and offers scaling benefits for communication. It is used to provide a gateway facility for communications because, for example; the Agent(s) are in a Realm such as a private network, where direct communication to the Collector is not permitted; or, the Agent(s) are in a Realm which uses an underlying technology that does not support the Internet Protocol. The Relay is expected to be deployed on a node that is able to communicate with both Agents and the Collector directly. (We expect Agents and Relays will be defined by end users or developers as required for different equipment and infrastructure.)

These functional components may be combined to form hybrid components, permitting scaling for the information model and for communication. For example, an *Agent/Collector* hybrid in Figure 1, Realm A, collects energy information from several Agents, aggregates or summarises the collected data, and provides access to a higher level Collector via a single Agent: this Agent would have a higher level abstraction, e.g. energy usage for a whole room or a whole department. In Figure 1 Realm C, a *Collector/Relay* hybrid collects energy information from a number of Agents using a local communication mechanism, e.g. a collection of serial links.

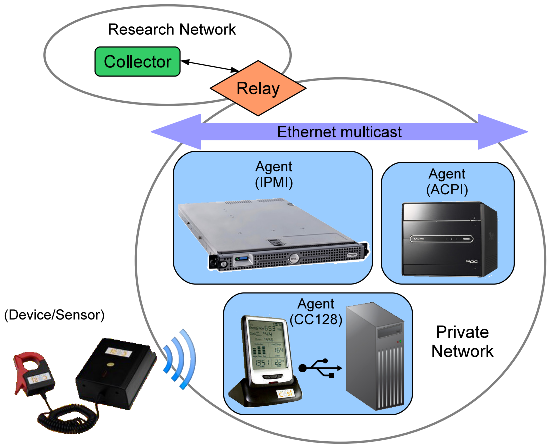

Fig. 2. Prototype Agents implemented: Envi CC128 consumer device, IPMI, and ACPI.

IV. PROTOTYPE

For our prototype instantiation of the architecture (see Figure 2), our goal was to show a proof-of-concept. We have chosen to use UUIDs [8] for identification, e.g. Agent ID (AID) and Relay ID (RID), as they are unique, platform-independent, and can be mapped to both OIDs and Uniform Resource Names (URNs)[6]. The UUIDs are currently used in a simple message forwarding system through their inclusion in data and control messages. We currently implement three different types of Agents. Each Agent has a cache, and prefetches and caches information from the RR using a controllable polling interval. Our Agents run on separate computers and gather energy information from different hardware interfaces:

- *Envi CC128*: This consumer device provides access to readings from multiple power sensors (passive monitoring

---

[6] We choose specifically *not* to overload IP addresses, so that we can integrate non-IP environments.

Fig. 3. Output of a simple WWW graphing application.

only), which it can identify individually[7]. It uses a proprietary radio protocol to communicate with the power sensors and connects to a computer using USB.

- *ACPI*: The Advanced Configuration & Power Interface (ACPI) [5] is widely supported by user-facing computing systems, e.g. laptops, and desktops. It allows power usage reporting and some power management functions.
- *IPMI*: With similar but more advanced functionality than ACPI, the Intelligent Platform Management Interface (IPMI) [7] is widely supported by high-end computing systems, e.g. servers.

Since our Agent instances are in an experimental private LAN and do not have direct access to an external network, a Relay is deployed as a gateway. As it forwards messages between the Agents and the Collector, a Relay has its RID added to or removed from the ID list within relevant messages automatically (see Section IV-B). The RID(s) within a message are (i) used by the Collector to group Agents; and (ii) used by the Relay to filter control messages sent to the Agents for which it acts, i.e. to discard irrelevant control messages. In our prototype case, IP multicast is used to allow efficient, lightweight, local communication with the Agents.

The Collector receives energy-usage information from all Agents and caches them in a local database. The user is able to (re-)configure the polling interval of a chosen Agent by specifying the AID, through the Collector.

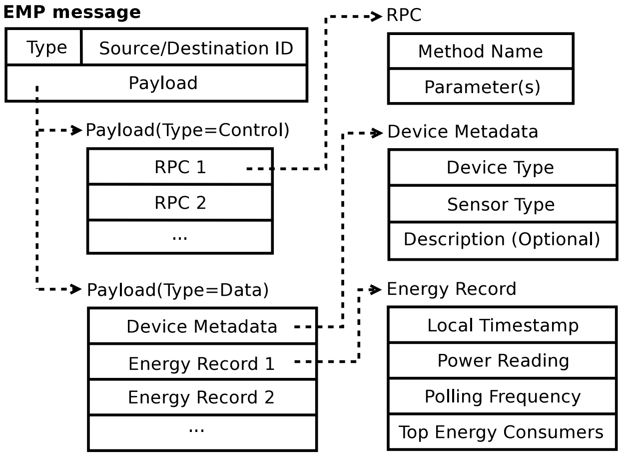

Fig. 4. Prototype EMP message structure.

We chose for our Application a simple WWW graphing tool, whose output is publicly available[8]. A snapshot is shown in Figure 3. The Application simply accesses the Collector's database and plots the numerical information using an openly available graphing utility, *RRDtool*[9]. Indeed, the information could be used in many ways, e.g. it could be an application that uploads energy information to *Pachube*[10] or *Google PowerMeter*[11], or a backend to an SNMP system, or an information feed to a cloud management system such as Eucalyptus [13], or some proprietary decision management function.

A. Energy Management Protocol

We define an *Energy Management Protocol (EMP)*, which allows the querying and retrieval of energy-usage information, as well as remote power management. Currently, for ease, the EMP is defined as a set of XML messages, lending itself to easy integration and interpretation within other applications (WWW applications as well as existing management applications). We can, of course, easily transform the XML messages (e.g. using XML schema-based transforms or stylesheets), or replace XML messages with a more compact format when we consider the design and architecture to be reasonably mature.

As Figure 4 illustrates, there are two types of EMP messages: (i) *control* messages, sent from Collectors to Agents; (ii) *data* messages, sent from Agents to Collectors.

A *control* message could carry more than one remote procedure calls (RPCs) targeting (i) a specific Agent by AID; or (ii) all Agents by a universal broadcast ID; or (iii) a specific cluster of Agents, for example by RID.

A *data* message includes an AID, *Device Metadata*, and a number of energy records. *Device Metadata* is (i) the type of device – computer, lighting, air conditioner (or *cluster* – meaning the record is aggregated from a group of devices); (ii) sensor type – IPMI, ACPI, consumer power meter, etc; and (iii)

---

[7] http://www.currentcost.com/product-cc128.html

[8] http://mist.cs.st-andrews.ac.uk/power.html

[9] http://oss.oetiker.ch/rrdtool

[10] http://www.pachube.com

[11] http://www.google.org/powermeter

some optional description of the device. The *Energy Record* includes (i) a local timestamp; (ii) the actual power reading; (iii) the current polling frequency/interval; and (iv) additional information such as top *N* energy consumers. Depending on the actual type of the Agent and its underlying RRs, this latter information could be: process IDs or names of interfaces of the highest power consuming processes/devices on a computer; AIDs of the highest power consuming devices within a cluster; or, for the example of a larger-scale system with a hierarchy of Agents (hybrid units), the highest power consuming departments within a university.

For hierarchical energy monitoring using Agent/Collector hybrids (A/C), we expect each A/C hides its underlying Agents' metadata from the Collector at the next level (i.e. 'above') by default, but is able to provide information about individual energy consumers (Agents) upon request.

B. Minimising Manual Configuration

Considering the possibilities of large-scale Agent deployments, we intend to minimise the required manual configuration. For our prototype, we have chosen to use separate multicast IP addresses for upstream communication (Agent to Collector, via zero or more Relays) and downstream communication (Collector to Agent, via zero or more Relays). We have arranged that the Agents, Relays and Collectors can easily discover the relevant communication configuration from a multicast signalling channel. Some manual configuration will be required with suitable information regarding Realms, and of course any device-, sensor- or RR-specific information. However, the intention is that the system is self-organised.

V. APPLICATIONS AND DISCUSSION

A. Improving Conventional Cloud Management

Conventional elastic cloud management consists of dynamic voltage and frequency scaling (DVFS), sleep (on/off) scheduling, virtual machine (VM) management, and cooling management, in order to maintain a stable, yet highly utilised system [10]. However, even with very little workload, such as 10% CPU utilisation, any of today's high-availability rack-mount servers would consume more than 50% of their peak power consumption [1]. We are working on a system model for individual devices that is summarised in Figure 5: in our view, the energy consumption and system workload have a relationship that is partly near-linear, partly near-exponential, and partly workload-independent. When a computer is carrying out its optimal workload (at the point marked $x\%$), any small addition of workload could lead to dramatic increase in power consumption, thus raising operational expenditure (OPEX) non-proportionally. We are currently investigating the efficacy of this model based on experiments using deterministic, synthetic workloads. We would like to integrate power-awareness (including energy-usage of infrastructure such as cooling) provided by SEM into current cloud management policies, and observe the performance of various policies. Ultimately, our aim is (i) to allow cloud providers to find the most reliable, yet economical cloud management policies for their own specific clouds; and (ii) to permit cloud providers to integrate power usage into user SLAs.

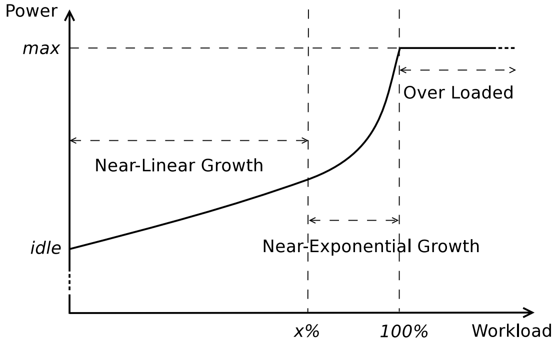

Fig. 5. Power vs Workload Assumption

B. Extended Power-Saving for the Cloud

Dynamically predicting users' demand and only providing necessary resources is a natural way to conserve power and increase resource utilisation. However, for modern Internet applications and services, one user request may trigger hundreds or even thousands of servers of the service provider for short period, e.g. a large data-processing task using *map-reduce*. As a result, high bursts of resource demand, and hence energy consumption, are expected, and the magnitudes depend on the actual computational load [10]. To cope with such 'spikes', resource over-provisioning is a current solution: the service providers risk losing money with low utilisation, and also risk customer dissatisfaction (and perhaps loss of customers) if sudden increases in service requests cannot be satisfied.

On the other hand, cloud services subscribers normally are not aware of the energy consumptions and/or related costs and potential savings of the services they buy. In this case, cloud providers, who potentially "have a detailed knowledge of the overall and host-level energy consumption of their facilities" [4], could make their customers not only aware of their energy consumption, but also fully accountable for it by offering discounted service charges when aggressive energy-saving actions are likely to *temporarily* have a negative impact on user's perceived QoS by some pre-agreed and acceptable margin. This appears to be a win-win solution for both cloud providers and cloud customers, with respect to costs.

One method to estimate how much discount to offer is for the service providers to accurately capture history of energy usage and resource usage (including SLA violations by the provider, for example), and estimate the amount of energy saved by ignoring or smoothing the demand spikes of service requests. A proposed solution of an accounting and billing architecture for federated cloud infrastructures [2] includes within its *Accounting Layer*, an *SLA Violation Assessment* component to pass any data regarding SLA violations to the billing layer. Such a component could be an important part of an energy-aware based SLA and charging scheme.

However, addressing SLA violations is not sufficient for enabling energy-aware SLAs and charging. Due to dynamic resource allocation, it is currently difficult to isolate resource usage for a single user [4]. We believe a comprehensive energy monitoring and power management integration as a contributor to meeting this challenge.

VI. ENERGY-AWARE METRICS AND POLICY

When detailed power measurements are made available, we then need to build system-wide metrics and KPIs of both system workload and energy consumptions to (i) make power management decisions, and (ii) estimate savings by applying such power management decisions. The GreenGrid consortium [11] define two related metrics – Power Usage Effectiveness (PUE) and Datacenter Efficiency (DCE).

However, these metrics rate data-centres as a whole, and are only applicable to our discussion of Section V-A. To enable our energy-aware SLAs and charging scheme described in Section V-B, we would eventually need *per user* metrics, that can be aggregated/accumulated over time, and used for accounting in terms of individual SLAs. This would allow us to apply energy-cost based savings to users. Currently, no such mechanism or metrics exist for this purpose, and the challenge is to design something that is practical and can be applied in useful timescales, e.g. monthly bills by a cloud provider to a customer. We are currently examining this issue.

Previous work has already presented a comprehensive method to estimate the TCO for cloud computing [2]. Regarding the OPEX of cloud infrastructures, the authors pointed out that the power consumption of ICT equipment and their cooling costs have direct relationship [9] [14]. That is to say, as the energy consumption of computing systems reduces, less heat is generated, hence cooling cost also reduces. As a result, when we measure and estimate the total OPEX savings by applying advanced power management policies, cooling systems and the IT equipment shall be considered in our new metrics, as this would show correctly the overall benefit in energy-usage and cost savings.

VII. CONCLUSION AND FUTURE WORK

In this paper we presented an architecture for an Scaleable Energy Monitor (SEM) solution that is capable of monitoring large-scale cloud infrastructure consisting of heterogeneous equipment. Scaleable, widely-deployed energy monitoring capability will enable energy-usage information to be integrated into existing cloud management policies and practises. We have presented a prototype that will allow incorporation of legacy systems, many different types of devices and sensors, and also deal with local constraints in communications.

We have also proposed making SLAs energy-aware and enabling charging schemes that allow extended power-saving by temporary, yet controllable, perturbations of SLAs, which could financially benefit both cloud providers and customers.

In the future, we plan to refine our SEM by implementing more sophisticated prototypes, as well as building a vendor-independent information model. Security and access control issues will be treated with high priority, as management policies may result in control actions that change the behaviour or configuration of systems based on. We will integrate SEM with *nagios* for quick deployment, and then look to integration with a cloud management system such as Eucalyptus. We intend to release our software on an open source licence to be used by the research community.

REFERENCES

[1] Gong Chen, Wenbo He, Jie Liu, Suman Nath, Leonidas Rigas, Lin Xiao, and Feng Zhao. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. In *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 337–350, Berkeley, CA, USA, 2008. USENIX Association.

[2] E. Elmroth, F.G. Marquez, D. Henriksson, and D.P. Ferrera. Accounting and billing for federated cloud infrastructures. *Grid and Cooperative Computing, 2009. GCC '09. Eighth International Conference on*, pages 268 –275, aug. 2009.

[3] Gartner. Press release for "forecast: Sizing the cloud; understanding the opportunities in cloud services". http://www.gartner.com/it/page.jsp?id=920712, March 2009.

[4] M. Gomez, D. Perales, and E.J. Torres. An energy-aware design and reporting tool for on-demand service infrastructures. *Grid Computing, 2009 10th IEEE/ACM International Conference on*, pages 209 –216, oct. 2009.

[5] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba. Advanced Configuration & Power Interface. http://www.acpi.info/.

[6] IEA (International Energy Agency). WORLD ENERGY OUTLOOK 2009 FACT SHEET. http://www.worldenergyoutlook.org/docs/weo2009/fact_sheets_WEO_2009.pdf, November 2009.

[7] Intel. Intelligent Platform Management Interface. http://www.intel.com/design/servers/ipmi/.

[8] P. Leach, M. Mealling, and R. Salz. A Universally Unique IDentifier (UUID) URN Namespace. *Internet RFCs*, RFC 4122, July 2005.

[9] Xinhui Li, Ying Li, Tiancheng Liu, Jie Qiu, and Fengchun Wang. The method and tool of cost analysis for cloud computing. *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on*, pages 93 –100, sept. 2009.

[10] Jie Liu, Feng Zhao, Xue Liu, and Wenbo He. Challenges towards elastic power management in internet data centers. *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on*, pages 65 –72, june 2009.

[11] Microsoft. The Green Grid: The Quest for Energy Efficient IT. http://www.microsoft.com/environment/our_commitment/articles/green_grid.aspx, 2009.

[12] National Computer Centre (NCC). The green IT paradox. Available from http://www.ncc.co.uk/, 2008.

[13] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. *Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on*, pages 124 –131, may 2009.

[14] Neil Rasmussen. Calculating Total Cooling Requirements for Data Centers. American Power Conversion white paper No.25, 2003.

[15] T. Singh and P.K. Vara. Smart metering the clouds. *Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on*, pages 66 –71, 29 2009-july 1 2009.

[16] SMART 2020, for The Climate Group. SMART 2020: Enabling the low carbon economy in the information age. http://www.gesi.org/index.php?article_id=210, Jun 2008.

APPENDIX

- Image of Dell PowerEdge server taken from: http://www.productwiki.com/dell-poweredge-1950-iii
- Image of Shuttle PC taken from: http://uk.shuttle.com/products.jsp
- Image of Envi CC128 taken from: http://www.electricity-monitor.com/
- Image of power sensor taken from: http://www.frequencycast.co.uk/currentcost.html