When Clouds become Green: the Green Open Cloud Architecture

Anne-Cécile ORGERIE [a] and Laurent LEFÈVRE [a]
[a] *INRIA RESO - Université de Lyon - LIP (UMR CNRS, INRIA, ENS, UCB)*
École Normale Supérieure - 46, allée d'Italie - 69364 LYON Cedex 07 - FRANCE
annececile.orgerie@ens-lyon.fr, laurent.lefevre@inria.fr,

Abstract. Virtualization solutions appear as alternative approaches for companies to consolidate their operational services on a physical infrastructure, while preserving specific functionalities inside the Cloud perimeter (e.g., security, fault tolerance, reliability). These consolidation approaches are explored to propose some energy reduction while switching OFF unused computing nodes. We study the impact of virtual machines aggregation in terms of energy consumption. Some load-balancing strategies associated with the migration of virtual machines inside the Cloud infrastructures will be showed. We will present the design of a new original energy-efficient Cloud infrastructure called Green Open Cloud.

Keywords. Energy Efficiency, Large-Scale Systems, Cloud Infrastructures, Migration

Introduction

Cloud infrastructures have recently become a center of attention. They can support dynamic operational infrastructures adapted to the requirements of distributed applications. Cloud systems provide on-demand computing power and storage, so they can perfectly fit the users' requirements. But as they reach enormous sizes in terms of equipments, energy consumption becomes one of the main challenges for large-scale integration.

This paper deals with the support of energy-efficient frameworks dedicated to Cloud architectures. Virtualization is a key feature of the Clouds, since it allows high performance, improved manageability, and fault tolerance. In this first step of our work, our infrastructure is focusing its action on this essential aspect. It also uses migration, which brings the benefit of moving workload between the virtual machines (VMs).

The remainder of this paper is organized as follows. Section 1 reviews related works. This is followed by an evaluation of the electric cost of a virtual machine in Section 2. Section 3 outlines the architecture and the components of the energy-aware Cloud infrastructure that we propose. The conclusion and future works are reviewed in Section 4.

1. Related Works

Innovative technologies have broadly contributed to the expansion of Clouds. They differ from Grids as explained in [2] and can be part of the next-generation data centers with

virtualized nodes and on-demand provisioning. Clouds are already used by numerous companies. For instance, Salesforce.com handles 54,000 companies and their 1.5 million employees via just 1,000 servers [18]. Different manufacturers, like IBM [1], also support and provide Clouds infrastructures and services for customers companies. Cloud computing is, by nature, dynamically scalable and virtualized resources are often provided as a service over the Internet [7]. This opens up wide new horizons where everything is considered as a service (infrastructure, platform, software, computing, storage). Among the other advantages of the Clouds are scalability, cost, and reliability. Cloud providers such as Amazon[1] should however face doubts on security and loss of control over sensitive data. The other main issue is the accounting.

Virtualization is the key feature of the Clouds, which allows improving the efficiency of large-scale distributed systems [15]. This technology needs powerful resource-management mechanisms [6] to benefit from live migration and suspend/resume mechanisms that allow moving a virtual machine from a host node to another one, stopping the virtual machine and starting it again later. The design of resource-management policies is a challenging (NP-hard) and dynamic problem.

Live migration [5] greatly improves the capacities and the features of Cloud environments: it facilitates fault management, load balancing, and low-level system maintenance. Migration operations imply more flexible resource management: when a virtual machine is deployed on a node, we can still move it to another one. It offers a new stage of virtualization by suppressing the concept of locality in virtualized environments.

However, this technique is complex and more difficult to use over the MAN/WAN [16] than in a cluster. IP addressing is a problem since the system should change the address of the migrated virtual machine which does not remain in the same network domain. Moreover, it impacts the performances of the virtual machines by adding a non negligible overhead [17].

With virtualization came some ideas concerning energy management [15,8]. Indeed, large-scale distributed systems are always increasing in size, and thus in power consumption. Each node can be virtualized and host several virtual machines.

So, at the same time, virtualization addresses the limitations in cooling and power delivery. This leads to the design of new techniques of energy management in virtualized systems. In [10], the authors propose a management system that uses different power management policies on the virtualized resources of each virtual machine to globally control power consumption.

The emerging concept of consolidation is directly linked with energy management in Clouds. The consolidation techniques aim to manage the jobs and combine them on the physical nodes. These techniques can be used to optimize energy usage [14].

2. Energy Cost of Virtual Machines

Cloud computing seems to be a promising solution to the increasing demand of computing power needed by more and more complex applications. We have seen that virtualization is promoted by several researches to decrease the energy consumption of large-scale distributed systems. However, the studies often lack real values for the electric consumption of virtualized infrastructures.

[1] `http://aws.amazon.com`

Our experimental Cloud consists of HP Proliant 85 G2 Servers (2.2 GHz, 2 dual core CPUs per node) with XenServer 5.0[2] installed on them.

Each Cloud node is linked to an external wattmeter that logs the power consumption of the node every second.

The measurement precision of our experimental platform is 0.125 watts and the maximum frequency is one measure per second.

On Xen, the VM live migration consists in transferring its memory image from the host Cloud node to the new one. If the VM is running, it is stopped for a period of time during the copy of the last memory pages (the ones that are often modified).

So when a migration occurs, the end of the job which is in the migrated VM is delayed by a certain amount of time (the period during which the VM is stopped): we denote that time T_m. This time does not include the whole duration of the migration process. Indeed, we have the same phenomenon as in the previous paragraph: competition at the hypervisor level. If several migrations are required at the same time on the same node, they are queued and processed one by one by the hypervisor (this can also be influenced by the network bandwidth).

On Figure 1 we have launched six *cpuburn*[3], one by one in six different VMs on Cloud node 1. The first starts at $t = 10$; we see that consumption increases to reach 219 watts. Then the second starts and consumption reaches 230 watts. The third starts and the node consumes 242 watts. The fourth leads to 253 watts. The appearance of the fifth and the sixth jobs does not increase consumption.

Indeed, since the jobs are CPU intensive (*cpuburn* uses 100 % of a CPU's capacity) and since there are only four cores on the node (2 dual core CPUs), they are fully used with the first four VMs. The fifth VM is free in terms of energy cost because it should share resources that would have been fully used without it too.

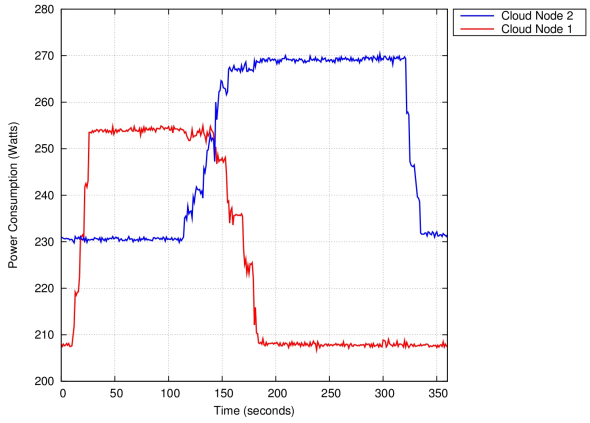

Figure 1. Migration of virtual machines

[2]XenServer is a cloud-proven virtualization platform that delivers the critical features of live migration and centralized multi-server management (`http://citrix.com/English/ps2/products/product.asp?contentID=683148`).

[3]cpuburn is a software designed to apply a high load to the processor (`http://pages.sbcglobal.net/redelm/`).

Each *cpuburn* job lasts 300 seconds (Figure 1). At $t = 110$, we launch the migration of the 6 VMs from Cloud node 1 to Cloud node 2. The migration requires sustained attention from the hypervisor that should copy the memory pages and send them to the new host node. For this reason, it cannot handle 6 migrations at the same time, they are done one by one.

The competition occurs and we see with the power consumption of Cloud node 2 that the VMs arrived one by one. The consumption of Cloud node 1 begins to decrease during the migration of the third VM. At that time, only three VMs are still running on the node.

Each job ends 5 seconds late, this is T_m. The competition that occurs during the migration request does not affect the jobs running on the last migrated VMs more than the others, since they are still running while waiting for a migration.

Among the components of a Cloud architecture, we have decided to focus on virtualization, which appears as the main technology used in these architectures. Our future works will include algorithms employing other Cloud components, like accounting, pricing, admission control, and scheduling. We also use migration to dynamically unbalance the load between the Cloud nodes in order to shut down some nodes, and thus to save energy.

3. Green Open Clouds

Some previous work on operational large-scale systems show that they are not utilized at their full capacity [9]. Resources (computing, storage, and network) are not used in a constant way by applications and users of large-scale distributed systems (e.g., clusters, grids, clouds). Some inactivity periods can be observed, monitored, and predicted. During these periods, some energy-aware frameworks can reduce energy consumption at a global level.

By generalizing the EARI framework (Energy Aware Resource Infrastructure) proposed for energy-efficient experimental Grids [12,13], we designed the Green Open Cloud architecture (GOC): an energy-aware framework for Clouds (Fig. 2). The described solution supports the "do the same for less" approach, dealing with efficient ON/OFF models combined with prediction solutions.

3.1. Green Open Cloud Architecture

The GOC architecture supports the following facilities:

- switching OFF unused computing, networking, and storage resources;
- predicting computing resources usage in order to switch ON the nodes which are required in a near future;
- aggregating some reservations to avoid frequent ON/OFF cycles;
- green policies which allow users to specify their requests in terms of energy targets.

The GOC infrastructure is added to the usual resource manager of the Cloud as an overlay. We do not modify the job scheduling policies (for example) in order to be adaptable to all the resource managers (such as Eucalyptus [11]).

The GOC infrastructure embeds (Figure 2): a set of electrical monitoring sensors providing dynamic and precise measurements of energy consumption, an energy data collector, a trusted proxy for supporting the network presence of switched-OFF cloud nodes, and an energy-aware resource manager and scheduler.

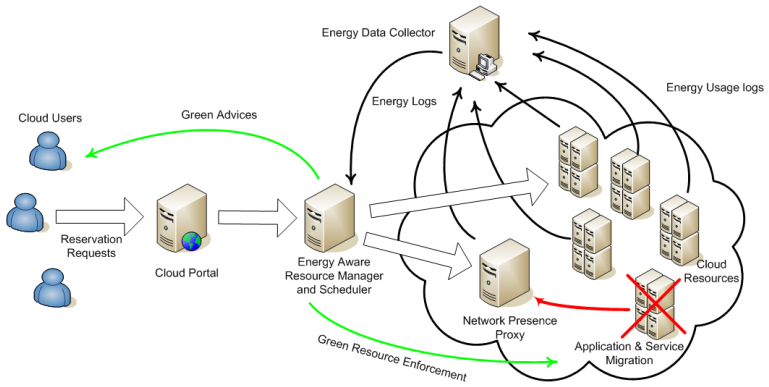

Figure 2. The Green Open Cloud (GOC) Infrastructure

The electrical sensors are connected to each node and they send their consumption measurements to the energy collector. The resource manager has access to this collector and requests for the energy logs when needed. It sends them to the Cloud portal in a well presented manner so that users can see them and see their impact on the power consumption of the nodes. The Cloud portal is responsible for providing some web services to the users and it is the access point for the outside.

The first idea to save energy is to switch off unused nodes because, as we have seen in Section 2, an idle node consumes a lot. We will thus develop our prediction algorithms, which aim to switch on the nodes when required. The energy-aware resource manager makes the decisions concerning the shutdown and the boot of the nodes (green resource enforcement).

The energy-aware resource manager also provides users with "green" advice in order to increase their energy awareness. This consists in proposing several solutions to the user when he submits a job: running it now if possible, or running it later and aggregate it with others (on the same nodes), or allowing migration decisions for running jobs. These green policies would increase resource sharing and so decrease the energy consumption of the Cloud. If the job is urgent, it can still be run immediately if there are available nodes.

3.2. GOC Resource Manager Architecture

The resource manager is the key element that concentrates the energy-efficient functionalities in the GOC architecture. Figure 3 presents the detailed architecture and features of this resources manager, including its interactions with the other components. The green boxes shows the energy-efficient parts.

The user's access portal is directly linked with the admission control module that is responsible for the security. Then, the job acceptance module determines if the user's

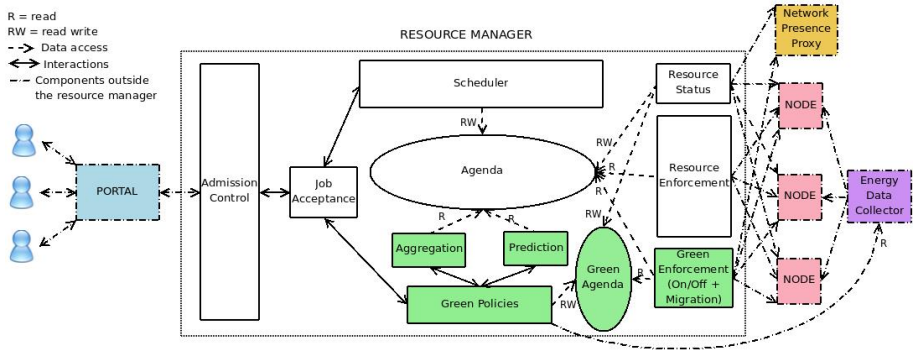

Figure 3. Architecture of the GOC Resource Manager

submission is acceptable according to management policies (for example, the system administrator can put a limit on the resources for a single user). If the submission is accepted, it is sent to the scheduler and the green policies module. The scheduler looks at the agenda to see if the submission can be put into this agenda (enough resources at the date wished by the user). According to the green policies defined by the admin and by using aggregation, the green policies module computes other possible slots for this job which are more energy efficient (the job will consume less energy because it will be aggregated with other ones).

The answers of the scheduler and the green policies module are sent back to the user who picks out one solution between the one he has submitted and the energy-efficient solutions proposed by the green policies module. Afterward, the solution chosen by the user is returned to the scheduler which puts it into the agenda.

At the end of each reservation, if there is totally or partially free nodes (with few VMs), the green policies use prediction to anticipate the next use of the freed resources. If they will be used in a short time, we do not switch them off or migrate their remaining VMs. We switch them off if they are totally free. If they are partially free and if their jobs will not end in a short time, we try to migrate their VMs on other nodes to minimize the number of nodes that are powered on. Otherwise, if they are partially free and if their jobs will end in a short time, we do not change anything. It will indeed cost more energy to migrate the VMs for such a short time. This process is described in Algorithm 1.

Algorithm 1 At the end of each job

ForEach totally or partially free resource M **Do**
  Predict the next use of M.
  If M will be used in less than T_s **Then**
    Nothing to do for M.
  Else
    If M is totally free **Then**
      Switch off M.
    Else
      If the job(s) of M will end in more than T_m **Then**
        Try to migrate the remaining VMs of M on other partially free resources to minimize the number of used resources.
      Else
        Nothing to do for M.

The green policies module is in charge of taking the on/off and migration decisions, then it inscribes it in the green agenda, which is read by the green enforcement module. The latter module is in charge of switching the resources on an off and migrating the VMs. This part is totally transparent for the non-green modules. Indeed, they have no access to the green agenda and the presence proxy is informed when a node is switched off and so can answer for it. The green enforcement module has access to the agenda in order to switch on the resources at the beginning of a job.

The resource enforcement module launchs the jobs and creates and installs the VMs of the users. It reads the agenda to know the reservations features (start time, VM configuration). It ensures that the user will not take more resources than he is allowed to.

The resource status module checks if the nodes are dead (not working properly). If the node has been switched off by the green enforcement module, the presence proxy answers instead. If a node is dead, the module writes it in the agenda.

As we switch off unused nodes, they do not answer to the Cloud resource manager. So they can be considered dead (not usable). This is a problem that we solve by using a trusted proxy. When we switch off a Cloud node, we migrate its basic services (such as ping or heartbeat services for example) on this proxy, which will answer for the node when asked by the resource manager. The key issue is to ensure the security of the infrastructure and to avoid the intrusion of malicious nodes. Our trust delegation model is described in [4].

3.3. Queue and Predictions

As we have seen in Section 2, there is competition between the VMs and between the jobs. Another level of competition appears in the resource manager. The Cloud portal transmits the user's jobs to the resource manager which schedules them on the different Cloud nodes. But it treats user requests one by one. So the requests are first queued.

A key feature of our infrastructure consists in switching off unused nodes. But, we need to boot them before a burst of jobs. Otherwise, if there are no more available host nodes, we should switch on the sleeping nodes when the requests arrive and we will delay them by the booting time.

Our prediction algorithms are based on the average values of the last submissions' inter-arrival times. When we have idle nodes (because they have finished their jobs or because we have migrated their VMs to free them), we need to know if we can switch them off. So we have defined a period of time denoted T_s.

This time is such that the node consumes as much power when it is idle as when we switch it off and on again during that time. So T_s is defined by:

$$T_s = \frac{E_s - P_{OFF}(\delta_{ON \to OFF} + \delta_{OFF \to ON}) + E_{ON \to OFF} + E_{OFF \to ON}}{P_{idle} - P_{OFF}}$$

where P_{idle} is the idle consumption of the node (power in watts), P_{OFF} the power consumption when the node is off (power in watts), $\delta_{ON \to OFF}$ the duration of the node's shutdown (in seconds), $\delta_{OFF \to ON}$ the duration of the node's boot, $E_{ON \to OFF}$ the energy consumed to switch off the node (in Joules), $E_{OFF \to ON}$ the energy consumed to switch on the node (in Joules), and E_s an energy threshold (a few Joules) that represents the amount of energy we save by switching the node off and on during T_s.

When a node is free, we predict when the next job will occur. If it is in less than T_s seconds and if this node is required for this predicted job, we leave it on. Otherwise we switch it off. Our prediction is computed as follows: at a given time t, the submission times of the $(n+1)$ previous jobs are denoted $t_0, \ldots, t_n$ where t_n is the most recent. So the next predicted job submission t' is:

$$t' = t + 1/n[(t_1 - t_0) + \cdots + (t_n - t_{n-1})] + t\_feedback = t + 1/n[t_n - t_0] + t\_feedback$$

where $t\_feedback$ is a feedback computed with the previous predictions: it represents an average of the errors made by computing the previous few predictions. The error is the difference between the true value and the predicted one.

We have seen in [12] that even with a small n (5 for example) we can obtain good results (70% of good predictions on experimental Grid traces). This prediction model is simple but it doesn't need a lot of disk accesses and is really fast to compute. These are crucial features for real-time infrastructures.

3.4. Comparison between GOC and EARI

GOC infrastructure is an adaptation of our Energy-Aware Reservation Infrastructure [13] that deals with Grids environments. This infrastructure is quite different from our previous work. Indeed, in EARI, we only handle the reservations: each user that wants to submit a job should precise its length in time, its size in number of nodes, and a wanted start time. With GOC, we just need the size in terms of number of VMs and there is no reservation nor agenda. For this reason, it greatly modifies the specifications of the elementary entity: the job in our case, the reservation for EARI. It implies different management algorithms for the jobs (to act live and not in the future) and different prediction algorithms since we want to predict the next submission and not the next reservation (in EARI, the user does a submission for a reservation that is in the future).

In both infrastructures, we guarantee the performance: we do not impact on the resources wanted by the user nor on the end time if the user does not agree.

We have validated EARI by using real usage traces of an experimental Grid. It is not possible with GOC since we do not have access to Cloud usage logs, so the GOC infrastructure is validated with some usage scenarios.

4. Conclusion and Perspectives

This paper presents our first step in designing a Green Open Cloud architecture by taking into account the energy usage of virtualization frameworks.

Based on the EARI model, we have experimentally measured and analyzed the electric cost of migrating virtual machines. We have proposed original software frameworks able to reduce the energy usage of Cloud infrastructure. These frameworks embed load balancing solutions with migration facilities and an ON/OFF infrastructure associated with prediction mechanisms. The GOC architecture is customizable, to include any green policy. For example, an admin can increase energy awareness by using a green accounting (the users who accepts to delay their jobs can be rewarded). The GOC architecture

can be adapted to any Cloud infrastructure: we are currently implementing and experimenting our GOC architecture with Eucalyptus [11] in large-scale distributed systems (i.e. the Grid5000 platform [3]) to fully validate it.

References

[1] G. Boss, P. Malladi, D. Quan, L. Legregni, and H. Hall. Cloud Computing. Technical report, IBM, 8 October 2007.

[2] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems*, 25(6):599–616, June 2009.

[3] F. Cappello et al. Grid'5000: A large scale, reconfigurable, controlable and monitorable grid platform. In *6th IEEE/ACM International Workshop on Grid Computing, Grid'2005*, Seattle, Washington, USA, Nov. 2005.

[4] Georges Da-Costa, Jean-Patrick Gelas, Yiannis Georgiou, Laurent Lefèvre, Anne-Cécile Orgerie, Jean-Marc Pierson, Olivier Richard, and Kamal Sharma. The green-net framework: Energy efficiency in large scale distributed systems. In *HPPAC 2009 : High Performance Power Aware Computing Workshop in conjunction with IPDPS 2009*, Roma, Italy, May 2009.

[5] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, pages 273–286, Berkeley, CA, USA, 2005. USENIX Association.

[6] Laura Grit, David Irwin, Aydan Yumerefendi, and Jeff Chase. Virtual machine hosting for networked clusters: Building the foundations for "autonomic" orchestration. In *VTDC '06: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing*, page 7, Washington, DC, USA, 2006. IEEE Computer Society.

[7] Brian Hayes. Cloud computing. *Communication of the ACM*, 51(7):9–11, 2008.

[8] Fabien Hermenier, Nicolas Loriant, and Jean-Marc Menaud. Power management in grid computing with xen. In *XEN in HPC Cluster and Grid Computing Environments (XHPC06)*, number 4331 in LNCS, pages 407–416, Sorento, Italy, 2006. Springer Verlag.

[9] A. Iosup, C. Dumitrescu, D. Epema, Hui Li, and L. Wolters. How are real grids used? the analysis of four grid traces and its implications. In *7th IEEE/ACM International Conference on Grid Computing*, September 2006.

[10] Ripal Nathuji and Karsten Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. In *SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 265–278, New York, NY, USA, 2007. ACM.

[11] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The eucalyptus open-source cloud-computing system. In *Proceedings of Cloud Computing and Its Applications*, Chicago, Illinois, USA, October 2008.

[12] Anne-Cécile Orgerie, Laurent Lefèvre, and Jean-Patrick Gelas. Chasing gaps between bursts : Towards energy efficient large scale experimental grids. In *PDCAT 2008 : The Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dunedin, New Zealand, December 2008.

[13] Anne-Cécile Orgerie, Laurent Lefèvre, and Jean-Patrick Gelas. Save watts in your grid: Green strategies for energy-aware framework in large scale distributed systems. In *14th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, Melbourne, Australia, December 2008.

[14] Shekhar Srikantaiah, Aman Kansal, and Feng Zhao. Energy aware consolidation for cloud computing. In *Proceedings of HotPower '08 Workshop on Power Aware Computing and Systems*. USENIX, December 2008.

[15] Richard Talaber, Tom Brey, and Larry Lamers. Using Virtualization to Improve Data Center Efficiency. Technical report, The Green Grid, 2009.

[16] Franco Travostino, Paul Daspit, Leon Gommans, Chetan Jog, Cees de Laat, Joe Mambretti, Inder Monga, Bas van Oudenaarde, Satish Raghunath, and Phil Yonghui Wang. Seamless live migration of virtual machines over the man/wan. *Future Gener. Comput. Syst.*, 22(8):901–907, 2006.

[17] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya. Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. Technical report, Technical Report, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 5 April 2009.

[18] Business Week. With Sun, IBM Aims for Cloud Computing Heights, 26 March 2009 http://www.businessweek.com/magazine/content/09_14/b4125034196164.htm?chan=magazine+channel_news.