Research challenges and potential green technological applications in cloud computing

P. Sasikala

Department of Computer Science and Computer Applications, Makhanlal Chaturvedi National University of Journalism and Communication, Press Complex, Bhopal, 462011, India
E-mail: sasipravi@yahoo.com

Abstract: Cloud computing has generated a lot of interest and competition in the IT industry. It has become a scalable services delivery platform in the field of services computing. The technical foundations of it include service-oriented architecture and virtualisations of hardware and software. The goal is to share resources among the cloud service consumers, cloud partners, and cloud vendors in the cloud value chain. The technology faces several significant challenges and the current research focuses on the technical issues that arise when building and providing clouds and the implications on enterprises and users. Structured along the technical aspects on the cloud programmes, we discuss associated technologies; advances in the introduction of protocols, interfaces, and standards; techniques for modelling and building clouds; and feasibility, testing and the future arising through cloud computing. Cloud computing is a huge leap orienting towards green computing, an environmentally sustainable forever computing with an enormous bright future.

Keywords: cloud computing; definition; characteristics; services; models; standardisation; green computing.


Biographical notes: P. Sasikala is a Reader in Computer Science at Makhanlal Chaturvedi National University of Journalism and Communication, Bhopal, India. Her doctoral research was in the area of text mining and she received her PhD in Computer Science from Barkatullah University, Bhopal. She has 15 years of research cum teaching experience. She has published over 25 papers in peer reviewed international journals and conferences. Her current research interest is in the field of text mining, cloud computing and ICT.

1 Introduction

The concept of cloud computing dates back to the 1960s, when John McCarthy (2010) opined that computation may someday be organised as a public utility. Cloud computing is internet-based computing, whereby shared resources, software, and information are provided to computers and other devices on demand, like the electricity
grid (Sasikala, 2010). Cloud computing is a paradigm shift following the shift from
mainframe to client-server in the early 1980s. Cloud computing describes a new
supplement, consumption, and delivery model for IT services based on the internet, and it
typically involves over-the-internet provision of dynamically scalable and often
virtualised resources (Marks and Lozano, 2010). A key element of cloud computing is
customisation and the creation of a user-defined experience. Cloud computing users
avoid capital expenditure on hardware, software, and services when they pay a provider
only for what they use. The major cloud service providers include Salesforce, Amazon
and Google and those actively involved in cloud computing are Fujitsu, Microsoft,
Hewlett Packard, IBM and VMware (Cloud Computing Portal, 2010).

According to Google Trends, the term cloud computing started becoming popular in
2007 as shown in Figure 1. Today cloud computing has reached high popularity and has
developed into a major trend in IT. However, like with most ‘hype’ technologies,
everyone seems to have heard of cloud computing, but no one seems sure as to what it
really means and has to offer. A number of universities, vendors and government
organisations are investing in research around the topic of cloud computing. While
industry has been pushing the cloud research aspects at high pace, academic world has
only recently started working deep in it. This is evident from the sharp rise in workshops,
seminars and conferences focusing on cloud computing (ACM, 2010; IEEE, 2010;
Indicthreads.Com, 2010).

Figure 1  Searches for ‘cloud computing’ on Google.com, taken from Google Trends (see online
version for colours)

Literature survey showed that several whitepapers, manuscripts, articles, chapters, views,
etc., are available on the internet on cloud computing. They all provide an overview and
thrust in this field, and the majority of the papers were seen published in 2009 (Mccarthy,
2010; Sasikala, 2010; Marks and Lozano, 2010; Cloud Computing Portal, 2010; ACM,
2010; IEEE, 2010; Indicthreads.Com, 2010). However, a systematic complete through
comprehensive literature review of the research, challenges and opportunities in cloud
computing is missing. Hence, in this paper we intend to provide various definitions of
cloud computing with a state-of-the-art review of the academic research across the globe
in cloud computing. The paper also highlights the research challenges and potential
technological applications in cloud computing.
2 Cloud computing

2.1 The definitions

There has been much discussion in industry as to what cloud computing actually means. The term cloud computing seems to originate from computer network diagrams that represent the internet as a cloud. Most of the major IT companies and market research firms such as IBM (2009), Sun Microsystems (2009), Gartner (Plummer et al., 2008) and Forrester Research (Staten, 2008) have produced whitepapers that attempt to define the meaning of this term. These discussions are mostly coming to an end and a common definition is starting to emerge. The US National Institute of Standards and Technology (NIST) has developed a working definition that covers the commonly agreed aspects of cloud computing. The NIST defines cloud computing as, “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Mell and Grance, 2009). The NIST definition is one of the clearest and most comprehensive definitions of cloud computing and is widely referenced in all government documents and projects. This definition describes cloud computing as having five essential characteristics, three service models, and four deployment models.

2.2 The essential characteristics of cloud computing

- **On-demand self-service**: Computing resources can be acquired and used at anytime without the need for human interaction with cloud service providers. Computing resources include processing power, storage, virtual machines, etc.

- **Broad network access**: The previously mentioned resources can be accessed over a network using heterogeneous devices such as laptops or mobile phones.

- **Resource pooling**: Cloud service providers pool their resources that are then shared by multiple users. This is referred to as multi-tenancy where for example a physical server may host several virtual machines belonging to different users.

- **Rapid elasticity**: A user can quickly acquire more resources from the cloud by scaling out. They can scale back in by releasing those resources once they are no longer required.

- **Measured service**: Resource usage is metered using appropriate metrics such as monitoring storage usage, CPU hours, bandwidth usage, etc.

The above characteristics apply to all clouds but each cloud provides users with services at a different level of abstraction, which is referred to as a service model in the NIST definition.
2.3 The three most common service models of cloud computing

- **Software as a service (SaaS):** This is where users simply make use of a web browser to access software that others have developed and offer as a service over the web. At the SaaS level, users do not have control or access to the underlying infrastructure being used to host the software. Salesforce’s customer relationship management software (http://www.salesforce.com/uk/crm/products.jsp) and Google Docs (http://docs.google.com) are popular examples that use the SaaS model of cloud computing.

- **Platform as a service (PaaS):** This is where applications are developed using a set of programming languages and tools that are supported by the PaaS provider. PaaS provides users with a high level of abstraction that allows them to focus on developing their applications and not worry about the underlying infrastructure. Just like the SaaS model, users do not have control or access to the underlying infrastructure being used to host their applications at the PaaS level. Google App Engine (http://code.google.com/appengine) and Microsoft Azure (http://www.microsoft.com/windowsazure/) are popular PaaS examples.

- **Infrastructure as a service (IaaS):** This is where users acquire computing resources such as processing power, memory and storage from an IaaS provider and use the resources to deploy and run their applications. In contrast to the PaaS model, the IaaS model is a low level of abstraction that allows users to access the underlying infrastructure through the use of virtual machines. IaaS gives users more flexibility than PaaS as it allows the user to deploy any software stack on top of the operating system. However, flexibility comes with a cost and users are responsible for updating and patching the operating system at the IaaS level. Amazon Web Services EC2 and S3 (http://aws.amazon.com/) are popular IaaS examples.

Erdogmus (2009) described SaaS as the core concept behind cloud computing, suggesting that it does not matter whether the software being delivered is infrastructure, platform or application, it is all software in the end. Although this is true to some extent, it nevertheless helps to distinguish between the types of service being delivered as they have different abstraction levels. The service models described in the NIST definition are deployed in clouds, but there are different types of clouds depending on who owns and uses them. This is referred to as a cloud deployment model in the NIST definition.

2.4 The four common models of cloud computing

- **Private cloud:** A cloud that is used exclusively by one organisation. The cloud may be operated by the organisation itself or a third party. The St. Andrews Cloud Computing Co-laboratory (http://www.cs.st-andrews.ac.uk/stace) and Concur Technologies (Lemos, 2009) are example organisations that have private clouds.

- **Public cloud:** A cloud that can be used (for a fee) by the general public. Public clouds require significant investment and are usually owned by large corporations such as Microsoft, Google or Amazon.
• **Community cloud**: A cloud that is shared by several organisations and is usually setup for their specific requirements. The Open Cirrus cloud test bed could be regarded as a community cloud that aims to support research in cloud computing (Open Cirrus™, 2010).

• **Hybrid cloud**: A cloud that is setup using a mixture of the above three deployment models. Each cloud in a hybrid cloud could be independently managed but applications and data would be allowed to move across the hybrid cloud. Hybrid clouds allow cloud bursting to take place, which is where a private cloud can burst-out to a public cloud when it requires more resources.

Figure 2 provides an overview of the common deployment and service models in cloud computing, where the three service models could be deployed on top of any of the four deployment models.

**Figure 2** Cloud computing deployment and service models (see online version for colours)

Others such as Vaquero et al. (2009) and Youseff et al. (2008) concur with the NIST definition to a significant extent. For example, Vaquero et al. studied 22 definitions of cloud computing and proposed the definition as: “Clouds are a large pool of easily usable and accessible virtualised resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing also for optimum resource utilisation. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the infrastructure provider by means of customised SLAs”. This definition includes three of the five characteristics of cloud computing described by NIST, namely resource pooling, rapid elasticity and measured service but fails to mention on-demand self-service and broad network access. Youseff et al. (2008) described a five-layer stack that can be used to classify cloud services; they use composability as their methodology where each service is composed of other services. The five layers are applications, software environment, software infrastructure, software kernel, and hardware. This is similar to the SaaS, PaaS and IaaS service models described in the NIST definition and only differs in the lower two layers, namely the software kernel and hardware layers. Grid and cluster computing systems such as Globus and Condor are examples of cloud services that fall into the software kernel layer, and ultra large-scale data centres as designed in IBM’s Kittyhawk project (Appavoo et al., 2008) are examples of hardware layer services (Youseff et al., 2008). However, these are not convincing examples of cloud services as they do not have the essential characteristics of cloud computing as described in the NIST definition,
therefore we feel that the two extra layers used by Youseff et al. could reasonably be seen as unnecessary when describing cloud computing.

It is useful to think of a cloud as a collection of hardware and software that runs in a data centre and enables the cloud computing model (Armbrust et al., 2009). According to Erdogmus, the benefits of cloud computing are scalability, reliability, security, ease of deployment, and ease of management for customers, traded off against worries of trust, privacy, availability, performance, ownership, and supplier persistence (Erdogmus, 2009).

Although there are still many internet forum and blog discussions on what cloud computing is and is not, the NIST definition seems to have captured the commonly agreed aspects of cloud computing that are mentioned in most of the academic papers published in this area. However, cloud computing is still in its infancy and as acknowledged by the authors Mell and Grance (2009), this and any definition is likely to evolve in the future as new developments in cloud computing are explored. The current two-page NIST definition of cloud computing could be nicely summarised using Joe Weinman’s retro-fitted CLOUD acronym that describes a cloud as a common, location-independent, online utility provisioned on demand (Weinman, 2008).

3 Associated technologies

Further, we review the researches that describes technological aspects of research in cloud computing. Voas and Zhang (2009) identified cloud computing as the next computing paradigm that follows on from mainframes, PCs, networked computing, the internet and grid computing. These developments are likely to have similarly profound effects as the move from mainframes to PCs had on the ways in which software was developed and deployed. One of the reasons that prevented grid computing from being widely used was the lack of virtualisation that resulted in jobs being dependant on the underlying infrastructure. This often resulted in unnecessary complexity that had an effect on wider adoption (Vouk, 2008). Ian Foster, who was one of the pioneers of grid computing, compared cloud computing with grid computing and concluded that although the details and technologies of the two are different, their vision is essentially the same (Foster et al., 2008). This vision is to provide computing as a utility in the same way that other public utilities such as electricity, water, gas and internet are provided. In fact the dream of utility computing has been around since the 1960s and advocated by the likes of John McCarthy and Douglas Parkhill. For example, the influential mainframe operating system multics had a number of design goals that are remarkably similar to the aims of current cloud computing providers. These design goals included remote terminal access, continuous operational provision (inspired by electricity and telephone services), scalability, reliable file systems that users trust to store their only copy of files, information sharing controls, and an ability to support different programming environments (Corbató et al., 1972). Therefore it is unsurprising that many people compare cloud computing to mainframe computing. However, it should be noted that although many of the ideas are the same, the user experience of cloud computing is almost completely the opposite of mainframe computing. Mainframe computing limited people’s freedom by restricting them to a very rigid environment; cloud computing expands their freedom by giving them access to a variety of resources and services in a self service manner.
Foster et al. (2008) compare and contrast cloud computing with grid computing. They believe cloud computing is an evolved version of grid computing, in such a way that it answers the new requirements of today’s time, takes into account the expensiveness of running clusters, and the existence of low cost virtualisation. IT has greatly evolved in the last 15 years since grid computing was invented, and at present it is on a much larger scale that enables fundamentally different approaches. Foster et al. see similarities between the two concepts in their vision and architecture, see a relation between the concepts in some fields as in the programming model (“MapReduce is only yet another parallel programming model”) and application model (but clouds are not appropriate for HPC applications that require special interconnects for efficient multi core scaling), and they explain fundamental differences in the business model, security, resource management, and abstractions. Foster et al. find that in many of these fields there is scope for both the cloud and grid research communities to learn from each other’s findings, and highlight the need for open protocols in the cloud, something grid computing adopted in its early days. Finally, Foster et al. believe that neither the electric nor computing grid of the future will look like the traditional electric grid. Instead, for both grids they see a mix of micro productions (alternative energy or grid computing) and large utilities (large power plants or data centres).

In market-oriented cloud computing, a follow on work from their market-oriented grid computing and market-oriented utility computing papers, Buyya et al. (2008) describe their work on market-oriented resource allocation and their Aneka resource broker: In the case of limited availability of resources, not all service requests will be of equal importance, and a resource broker will regulate the supply and demand of resources at market equilibrium. A batch job for example might be preferably processed when the resource value is low, while a critical live service request would need to be processed at any price. Aneka, commercialised through Manjrsoft, is a service broker that mediates between consumers and providers by buying capacities from the provider and subleasing them to the consumers. However, such resource trading requires the availability of ubiquitous cloud platforms with limited resources, and is in contrast to the desire for simple pricing models.

As cloud computing delivers IT as a service, cloud researchers can also learn from service-oriented architecture (SOA). In fact, the first paper that introduced PaaS (Chang et al., 2006) described PaaS as an artifact of combining infrastructure provisioning with the principles of SaaS and SOA. Since then, no academic work has been published in the field of PaaS. We have to take our to-date understanding of PaaS from the current developments in industry, in particular from the two major vendors, Force.com and from Google App Engine. Sedayao (2008) built a monitoring tool using SOA services and principles, and describe their experience from building a robust distributed application consisting of unreliable parts and the implication for cloud computing. As design goal for distributed computing scenarios such as cloud computing they propose, “like routers in a network, any service using other cloud services needs to validate input and have hold down periods before determining that a service is down” (Sedayao, 2008). Zhang and Zhou (2009) analyse convergence from SOA and virtualisation for cloud computing and present seven architectural principles and derive ten interconnected architectural modules. These build the foundation for their IBM cloud usage model, which is proposed as cloud computing open architecture (CCOA). Vouk (2008) described cloud computing from a SOA perspective and talked about the virtual computing laboratory (VCL) as an implementation of a cloud. VCL is an “open source implementation of a secure
production-level on-demand utility and service-oriented technology for wide-area access to solutions based on virtualised resources, including computational, storage and software resources” (Voas and Zhang, 2009). In this respect, VCL could be categorised as an IaaS layer service.

Napper and Bientinesi (2009) ran an experiment to compare the potential performance of Amazon’s cloud computing with the performance of the most powerful, purpose build, high performance computers (HPC) in the top 500 list in terms of solving scientific calculations using the LINPACK benchmark. They found that the performance of individual nodes in the cloud is similar to those in HPC, but that there is a severe loss in performance when using multiple nodes, although the used benchmark was expected to scale linearly. The AMD instances scaled significantly better than the Intel instances, but the cost for the computations were equivalent with both types. As the performance achieved decreased exponentially in the cloud and only linearly in HPC systems, Napper and Bientinesi (2009) conclude that despite the vast availability of resources in cloud computing, these offerings are not able to compete with the supercomputers in the top 500 list for scientific computations.

In a non-peer-reviewed summary of keynote speeches for a workshop on distributed systems Birman et al. (2009) express that the distributed systems research agenda is quite different to the cloud agenda. They argue that while technologies from distributed systems are relevant for cloud computing, they are no longer central aspects of research. As example they list strong synchronisation and consistency as ongoing research topics from distributed systems. In cloud computing they remain relevant, but as the overarching design goal in the cloud is scalability, the search is now for decoupling and thus avoiding synchronisation, rather than improving synchronisation technologies. Birman et al. 2009) come to a cloud research agenda comprising four directions: managing the existing compute power and the loads present in the data centre; developing stable large scale event notification platforms and management technologies; improving virtualisation technology; and understanding how to work efficiently with a large number of low end and faulty components.

Cloud computing has been compared to several related fields of research. This section has shown that the cloud computing research programmes differ from the programmes in related fields, but that there are several findings in related research communities the research community can benefit from. We have also seen, that practitioners in distributed computing, grid computing, and SOA have joined the cloud community and proposed goals for research based on the background of their field.

4 Standards and interfaces

Cloud computing seeks to be a utility delivered in a similar as way electricity is delivered. Due to the higher complexity involved in delivering IT resources, open standards are necessary that enable an open market of providing and consuming resources. Currently, each vendor develops its own solution and avoids too much openness, to tie consumers in to their services and make it hard for them to switch to competitors. However, to new adopters the fear of vendor lock-in presents a barrier to cloud adoption and increases the required trust. There are three groups currently working on standards for cloud computing: The cloud computing interoperability forum (http://www.cloudforum.org), the open cloud consortium
Research challenges and potential green technological applications

(http://www.opencloudconsortium.org), and the DMTF open cloud standards incubator (http://www.dmtf.org/about/cloud-incubator). There is also a document called the open cloud manifesto (http://www.opencloudmanifesto.org), in which various stakeholders express why open standards will benefit cloud computing. In literature, Grossman (2009) points out that the current state of standards and interoperability in cloud computing is similar to the early internet era where each organisation had its own network and data transfer was difficult. This changed with the introduction of TCP and other internet standards. However, these standards were initially resisted by vendors just as standardisation attempts in cloud computing are being resisted by some vendors.

Keahey et al. (2009) looked into the difficulties of developing standards and summarised the main goals of achieving interoperability between different IaaS providers as being machine-image compatibility, contextualisation compatibility and API-level compatibility. Image compatibility is an issue as there are multiple incompatible virtualisation implementations such as the Xen, KVM, and VMWare hypervisors. When users want to move entire VMs between different IaaS providers, from the technological point of view this can only work when both providers use the same form of virtualisation. Contextualisation compatibility problems exist because different IaaS providers use different methods of customising the context of VMs, for example setting the operating system’s username and password for access after deployment must be done in different ways. Finally, there are no widely agreed APIs between different IaaS providers that can be used to manage virtual infrastructures and access VMs. For machine image or VM compatibility there is an ongoing attempt to create an open standard called the open virtual machine format (OVF). At the API-level, for PaaS AppScale (http://code.google.com/p/appscale), an open source effort to re-implement the interfaces of Google App engine, is aiming to become a standard, and for IaaS management, Amazon EC2’s APIs are quickly becoming a de-facto standard, popularised through their open source re-implementation eucalyptus.

Eucalyptus is an open-source software package that can be used to build IaaS clouds from computer clusters (Nurmi et al., 2008). Eucalyptus emulates the proprietary Amazon EC2 SOAP and query interface, and thus an IaaS infrastructure set up using eucalyptus can be controlled with the same tools and software that is used for EC2. The open source nature of eucalyptus gives the community a useful research tool to experiment with IaaS provisioning. The initial version of eucalyptus used Xen as hypervisor for virtual machines, but since the publication of that version, support for further hypervisors has been added, in particular for the newly popular KVM hypervisor (http://www.linux-kvm.org). Eucalyptus has a hierarchical design that makes it reasonably easy to predict its performance. However, for very large data centres this centralised design might not scale particularly well, hence Nurmi et al. recommend it for typical settings in present in academia. Although eucalyptus just re-implemented the Amazon EC2 interfaces, to date it is one of the most fundamental contributions by the research community towards standards in cloud computing, although only a few other providers use these interface APIs yet. But, for reasons such as fault tolerance or performance, or freedom from lock-in, consumers may wish to use multiple cloud providers. In the ABSENCE of open standards, or when attempts at providing open interface standards like eucalyptus are not followed by some providers, there will be heterogeneous interfaces. Dodda et al. (2009) address the problem of managing cloud resources with such heterogeneous access, by proposing a generic interface to the specific interface presented by individual cloud providers. They use their interface to an interface
to compare the performance of Amazon EC2’s query and SOAP interface, and find that the average response time for the SOAP interface was nearly double that of the query interface. These results emphasise the importance of selecting the interface through which resources from a given provider are managed. In a similar effort, Harmer et al. (2009) present a cloud resource interface that hides the details of individual APIs to allow provider agnostic resource usage. They present the interface to create a new instance at Amazon EC2, at Flexiscale (http://www.flexiscale.com), and at a provider of on-demand non-virtualised servers called NewServers (http://www.newservers.com), and implemented an abstraction layer for these APIs. The solution from Harmer et al. goes beyond hiding API details and contains functionality to compensate for loss of core infrastructure in scenarios where multiple providers are used.

Cloud computing can benefit from standardised API interfaces as generic tools that manage cloud infrastructures can be developed for all offerings. For IaaS there are developments towards standards and eucalyptus is looking to become the de-facto standard. For PaaS and SaaS stakeholders need to join the standardisation groups to work towards it. Achieving standardised APIs appears to be rather politically than technically challenging, hence there seems to be little space for academic involvement. However, standardised interfaces alone do not suffice to prevent vendor lock-in. For an open cloud, there is a need for protocols and software artefacts that allow interoperability to unlock more of the potential benefits from cloud computing.

5 Cloud interoperability and novel protocols

The next steps from compatible and standardised interfaces towards utility provisioning are universal open and standard protocols that allow interoperability between clouds and enable the use of different offerings for different use cases. Bernstein et al. (2009) describe an in-depth overview of the technological research agenda and open questions for interoperability in the cloud. They are looking for ways of allowing cloud services to interoperate with other clouds and highlight many goals and challenges, such as that cloud services should be able to implicitly use others through some form of library without the need to explicitly reference them, e.g., with their domain name and port. The collection of protocols inside and in-between the clouds that solve interoperability in the cloud are termed intercloud protocols. The intercloud protocol research agenda is made up of several areas: addressing, naming identity and trust, presence and messaging, virtual machines, multicast, time synchronisation, and reliable application transport. For cloud computing, each of these areas contains several issues. In addressing for example, the research problem is that there is the limited address space in IPv4 and that its successor IPv6 might be an inappropriate approach in a large and highly virtualised environment, as the cloud, due to its static addressing scheme: Bernstein et al criticise that IP addresses traditionally embody network locations for routing purposes and identity information, but in the cloud context identifiers should allow the objects to move into different subnets dynamically. This problem of static addresses is addressed by Ohlman et al. (2009). They recommend the usage of networking of information (NetInf) for cloud computing systems. Unlike URLs which are location-dependent, NetInf uses a location-independent model of naming objects, and offers an API that hides the dynamics of object locations and network topologies. Ohlman et al. demonstrate how this can ease management in the cloud, where the design desires transparency of location.
Further desired interoperability developments are listed by Matthews et al. (2009), who propose virtual machine contracts (VMC) as an attempt at standardising VM protection and security settings, and are working on adding VMCs to the OVF as extension to the metadata. Even in data centres under automated control and management, it is necessary to have customised settings for security and protection, such as firewall rules and bandwidth allowances for individual VMs. Today, these settings usually require the virtual appliance designer to manually communicate to both the person deploying the VM and to the administrators of the system, and the settings are specified and communicated in a site-specific non-portable format. In addition to allowing automated data centre management and cloud interoperability, Matthews et al. list several use cases for VMCs: Support for examining migration of enterprise data centres to the cloud; setting bounds on resource consumption and allowing capacity planning; detecting compromised VMs by comparing the VMs behaviour to the specified resource consumption estimates; specifying virtual network access control; specifying rules that ensure regulatory compliance and ease auditing compliance; and supporting disaster recovery as the required infrastructure elements will be known without having to instantiate a copy of the VM on a recovery cluster. Another piece of work that looks into cloud interoperability is Lim et al.’s (2009) feedback control service for scaling in the cloud. Lim et al. say scaling choices must be under control of the users, in order to have control over spending and to be able to work towards maximising return on investments. Thus, feedback control systems that make scaling decisions need to be decoupled from the cloud provider. In experiments using CPU utilisation as threshold for scaling choices, the best results were found when coarse grained ranges where specified as desired states. Lim et al. intend to consider further sensors such as application level metrics of queue lengths or response times for scaling choices in their future work, and as open research question they ask how much internal cloud knowledge such a controller would minimally need in order to be effective, and how much control needs to be exposed by the cloud for this.

Sun et al. (2007) have looked into the integration of SaaS services. They argue that complete or full-blown solutions are too costly and hard to configure, so that real applications will require an integration of multiple individual SaaS products. Further, they see a need for SaaS products to be seemlessly integrated with user’s existing in-house applications. Sun et al. split the functional requirements of an integration process in user interface integration, process integration and data integration, while they classify the key non-functional requirements as security, privacy, billing, and QoS reporting. They then propose SaaS-DL as an extension of Web Service Definition Language (WSDL), and introduce a reference architecture and prototype for a SaaS integration framework. In a case study they found integrating functional requirements possible even using existing SOA techniques, but they note that most SaaS providers do not provide programmatic interfaces to retrieve QoS and billing information which would be necessary to satisfy non-functional integration requirements.

Mikkilineni and Sarathy (2009) compared the evolution of cloud computing with intelligent network infrastructure in telecommunications and proposed a virtual resource mediation layer (VRML) to support interoperability between public and private clouds. VRML is an abstraction layer that sits on top of the IaaS layer and allows applications to access CPU, memory, bandwidth and storage depending on needs. The paper fell short of providing any technical details of how such a layer could be implemented, given the APIs used by different IaaS providers are incompatible and disclose only limited information.
about the real hardware. As mentioned by Grossman (2009), vendors are currently resisting standardisation attempts, which make the implementation of such abstraction layers a difficult task.

While much of the research work around cloud interfaces is taking concrete shape, most research on intercloud communication and resource sharing is still focused on defining the research questions and comes without even initial empirical results. This is perhaps to be expected as cloud computing is a relatively new field of research, despite the fact that both the general distributed computing field and the attempt to deliver IT resources as a utility to the consumer has been a goal of research for many decades. So far, a rich intercloud research programme has been stated by Bernstein et al. (2009), and it is likely that the search for interoperability will remain a challenging question in cloud computing for a while.

6 Building clouds

In this section we describe work that helps building cloud offerings. This requires management software, hardware provision, simulators to evaluate the design, and evaluating management choices. Sotomayor et al. (2009) presents two tools for managing cloud infrastructures: OpenNebula, a virtual infrastructure manager, and Haizea, a resource lease manager. To manage the virtual infrastructure, OpenNebula provides a unified view of virtual resources regardless of the underlying virtualisation platform, manages the full lifecycle of the VMs, and supportconfigurable resource allocation policies including policies for times when the demand exceeds the available resources. Sotomayor et al. argue that in private and hybrid clouds resources will be limited, in the sense that situations will occur where the demand cannot be met, and that requests for resources will have to be prioritised, queued, pre-reserved, deployed to external clouds, or even rejected. They propose advance reservations to have resources available to serve higher prioritised requests that are expected to be shortly arriving. This can be solved with resource lease managers such as the proposed Haizea, something like a futures market for cloud computing resources, which pre-empts resource usage and puts in place advance resource reservations, so that highly prioritised demand can be served promptly. Haizea can act as a scheduling backend for OpenNebula, and together they advance other virtual infrastructure managers by giving the functionality to scale out to external clouds, and providing support for scheduling groups of VMs, such that either the entire group of VMs are provided resources or no member of the group. In combination they can provide resources by best-effort, as done by Amazon EC2, by immediate provision, as done by eucalyptus, and in addition using advance reservations.

Song et al. (2009) have extended IBM data centre management software to be able to deal with cloud-scale data centres, by using a hierarchical set up of management servers instead of a central one. As even simple tasks such as discovering systems or collecting inventory can overwhelm a single management server when the number of managed components or endpoints increases, they partition the endpoints to balance the management workload. Song et al. chose a hierarchical distribution of management components, as a centralised topology will in any possible implementation result in bottlenecks, and because P2P structuring exhibits complexities that are not easy to understand. For resilience, the management components have backup servers which are notified with the changes from the original server. Once this notification no longer
arrives, the backup server will replace the original server’s task until it comes back to operation. In a case study, Song et al. show that this solution scales ‘almost linearly’ to 2,048 managed endpoints with eight managing servers. However, cloud-scale solutions might need to manage a number of virtual machines that is one or two orders of magnitude larger, and in the future will become even larger. It is left for future work to test if the solution will be feasible and scale for such numbers of managed endpoints.

Vishwanath et al. (2009) describe the provision of shipping containers that contain building blocks for data centres. The containers described are not serviced over their lifecycle, but allow for graceful failure of components until performance degrades below a certain threshold and the entire container gets replaced. To achieve this, Vishwanath et al. start with over-provisioning the demand at the start, or by putting cold nodes into the container which are not powered on once there is demand due to failure in some of the other components. This work aims at supporting the design of shipping containers with respect to costs, performance, and reliability. For reliability, Markov chains are used to calculate the expected mean time to failure over the lifecycle. For performance and cost, these Markov chains are extended into Markov reward models. These happen under the assumption of exponential failure times, and need to be evaluated against real data. The shipping containers could be used for selling private clouds in a box.

Sriram (2009) discusses some of the issues with scaling the size of data centres used to provide cloud computing services. He presents the development and initial results of a simulation tool for predicting the performance of cloud computing data centres which incorporates normal failures, failures that occur frequently due to the sheer number of components and the expected average lifecycle of each component and that are treated as the normal case rather than as an exception. Sriram shows that for small data centres and small failure rates the middleware protocol does not play a role, but for large data centres distributed middleware protocols scale better. CloudSim, another modelling and simulation toolkit has been proposed by Buyya et al. (2009). CloudSim simulates the performance of consumer applications executed in the cloud. The topology contains a resource broker and the data centres where the application is executed. The simulator can then estimate the performance overhead of the cloud solution. CloudSim is built on top of a grid computing simulator (GridSim) and looks at the scheduling of the execution application, and the impact of virtualisation on the application’s performance.

Abdelsalam et al. (2009) seek to optimise change management strategies, which are necessary for updates and maintenance, for low energy consumption of a data centre. However, this work simply derives the actual load from the service level agreements (SLA) negotiated with current customers. Abdelsalam then show that the number of servers currently required is proportional to the load, and identifies the number of idle servers as those available after all SLAs are fulfilled on a minimum set of servers. These are suggested as candidates for pending change management requests. One of the key aspects of cloud computing is elasticity, however, which will make it difficult to estimate the load from the SLAs in place. It is a challenge to develop such placement algorithms that the existing load can always be shrunk to a subset of the available servers while still fulfilling all SLAs, and cost factors will seek to minimise idle servers. Further work is necessary that takes these requirements into account and develops guidelines for both saving energy consumption and enabling seamless change management in cloud data centres.

In summary, several projects research into the way future clouds can be built. The papers discussed in this section differ too much to conclude with a single research
direction in which academic world is heading when looking into building future clouds. In fact, it seems there are many more research directions we will be facing when it comes to building new cloud facilities. All papers in this section for example, looked only at IaaS level clouds. To date, no paper could be found that describes technologies for building clouds at another level.

7 Cloud computing: feasibility, testing and the future

In this paper we have so far presented work that seeks to advance the technology of cloud computing. We end this by presenting new technologies and use cases that become possible through the use of cloud computing. Chun and Maniatis (2009) describe one such use-case, where cloud computing enables a technology which otherwise would not be possible: to overcome hardware limitations and enable more powerful applications on smart phones, they use external resources. This is done by partially off-loading execution from the smart phone and using cloud resources. But, Chun and Maniatis also include laptops or desktops near the phone in their ‘cloud’ because of the network latency for phones. Depending on the use case, their model offloads entire computations or parts thereof, and only has the remainder executed locally.

Another use-case that becomes feasible and affordable through the use of cloud computing is large-scale non-functional requirements testing, as described by Ganon and Zilbershtein (2009). They tested network management systems for systems where much of the functionality is in the endpoints, such as in voice over IP software. They discuss the advantages and disadvantages of cloud-based testing over testing against real elements or a simulator, and describe how a cloud-based test setup can be created using agents that are deployed into the cloud and with the use of cloud elasticity. Further, implications of using the cloud for this setup are evaluated, such as security, safety of intellectual property or software export restrictions, and solutions to tasks such as creating setups that emulate problems including noisy or delayed network connections are presented. Ganon and Zilbershtein reach the conclusion that there are significant benefits of using cloud-based testing, although it cannot completely replace traditional testing against real managed endpoints. They round up their insightful paper with a use case of a test scenario that was carried out on Amazon’s cloud and resulted in improvements to the software that could not have been highlighted with other feasible forms of testing, and with disclosing the costs occurred to carry out the cloud-based test. Matthew and Spraetz (2009) also looked at testing in cloud computing. They explained an effort to automate testing for SaaS providers along the example of Salesforce’s Apex test framework. Because consumers can use the Force.com PaaS offering to customise their business solutions into the CRM system using an entire Java-like programming language, it becomes unfeasible to test all possible states of the CRM beforehand. Instead, a test framework is provided, that allows users to specify regression tests, which can be carried out before every update to the SaaS offering. This is crucial because in a SaaS world there is no choice of version. Once an update is rolled out it is effective for all users.

In a cloud that offers IaaS, the number of VMs and thus instances of operating systems that need to be managed increases significantly. To avoid having to deploy software and updates into each virtual machine, and to avoid lengthy installation processes, entire so called ‘virtual appliances’ will be managed. This means, in cloud computing the operating system will no longer be viewed separated from the applications
deployed, but rather both will be deployed and maintained jointly. For service providers this means, they now have the ability to offer a virtual appliance, as functional disc image, instead of having to create lengthy installation procedures to guarantee compatibility with other applications in the VM. Wilson (2009) describes coronary, a software configuration management tool for virtual appliances. Coronary takes the idea of incremental updates from configuration management software such as CVS or subversion, and uses this technology to manage virtual appliances over their lifecycle. Wilson discusses the new requirements of version control when used for virtual appliances, and how coronary handles them.

Cloud computing could benefit from the functionality modelling issues studied in service computing, and the context-sensitivity issues studied in pervasive computing (Mei et al., 2008). However, it is difficult to talk about cloud computing without having a particular abstraction layer in mind. The lack of scalable storage, performance unpredictability and data transfer bottlenecks are also obstacles that could limit the growth of cloud computing. These obstacles present a number of new research opportunities in cloud computing.

8 Green cloud computing

Thrust of computing was initially on faster analysis and speedier calculation and solving of more complex problems. But in the recent past another focus has got immense importance and that is green computing (http://en.wikipedia.org/wiki/Green_computing). Green computing is the practice of using computing resources efficiently. The goals are to reduce the use of hazardous materials, maximise energy efficiency during the product’s lifetime, and promote recyclability or biodegradability of defunct products and factory waste. Such practices include the implementation of energy-efficient central processing units, servers and peripherals as well as reduced resource consumption and proper disposal of electronic waste. In 1992, the US Environmental Protection Agency launched Energy Star, a voluntary labelling programme which is designed to promote and recognise energy-efficiency in monitors, climate control equipment, and other technologies. This resulted in the widespread adoption of sleep mode among consumer electronics. The term ‘green computing’ was coined shortly after the energy star programme began. It also strives to achieve economic viability and improved system performance and use, while lasting for a long time by our social and ethical responsibilities. It refers to environmentally sustainable forever computing. Cloud computing with all these unique features will certainly raise a new horizon of energy efficiency and e-waste minimisation, and will become the major contributor for green computing. Cloud computing is significantly greener than its traditional IT operations alternatives.

However, data centres hosting cloud applications may still consume huge amounts of energy, contributing to high operational costs and carbon footprints to the environment. Therefore, we need to work on green cloud computing solutions that can not only save energy for the environment but also reduce operational costs. We must have better vision, design, model of architectural elements for energy-efficient management of cloud computing environments. Green cloud computing is envisioned to achieve not only efficient processing and utilisation of computing infrastructure, but also minimise energy consumption. This is essential for ensuring that the future growth of cloud computing is
sustainable. Otherwise, cloud computing with increasingly pervasive front-end client devices interacting with back-end data centres will cause an enormous escalation of energy usage. To address this problem, data centre resources need to be managed in an energy-efficient manner to drive green cloud computing. In particular, cloud resources need to be allocated not only to satisfy quality of service requirements specified by users via SLA, but also to reduce energy usage. However, to support green cloud computing, providers also need to minimise the energy consumption of cloud infrastructure, while enforcing service delivery. Rising energy cost is a highly potential threat as it increases the total cost of ownership and reduces the return on investment of cloud infrastructure setup by providers. However, the current state-of-the-art in cloud infrastructure has no or limited consideration for supporting energy-aware service allocation that meets QoS needs of consumers and minimises energy costs to maximise return on investment.

9 Conclusions

This paper discussed the research academic world has pursued to advance the technological aspects of cloud computing, and highlighted the resulting directions of research facing the academic community. In this way the various projects were set in context, and the research programmes followed by and facing academic world was presented. This review showed that there are several ways in which the cloud research community can learn from related communities, and has shown there is interest in academic circles for describing these similarities. Further, there have been attempts at building unified APIs to access clouds which seem to be more politically than technically challenging. Then, the perhaps clearest research plan was presented towards interoperability in the cloud and the challenges that need to be overcome. Finally, both for building clouds and presenting feasibility in the cloud, the research efforts were shown to be very diverse, making it hard to suggest in which way academic circles will be moving. This paper reviewed the technical aspects of research in cloud computing with a mention about green computing, as consumers are increasingly becoming conscious about the environment.

References

Research challenges and potential green technological applications


datacenter and cloud computing environments’, ACDC ’09: Proceedings of the 1st Workshop

Mccarthy, J. (2010) available at
November 2010).

thoughts on research issues’, Asia-Pacific Services Computing Conference, 2008, APSCC ’08,
pp.464–469, IEEE.


Technologies: Infrastructures for Collaborative Enterprises, 2009, WETICE ’09, 18th IEEE
International Workshops, pp.57–62.

Napper, J. and Bientinesi, P. (2009) ‘Can cloud computing reach the top 500?’, UCHPC-MAW ’09:
Proceedings of the Combined Workshops on Unconventional High Performance Computing

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L. and Zagorodnov, D.
Computing and its Applications.

cloud computing’, Enabling Technologies: Infrastructures for Collaborative Enterprises,
2009, WETICE ’09, 18th IEEE International Workshops, pp.78–83.


Computing: Defining and Describing an Emerging Phenomenon.

Computing, in press.

service oriented architecture principles and cloud computing infrastructure’, Iiwas ’08: Proceedings of the 10th International Conference on Information Integration and Web-based
Applications & Services, pp.417–421.

cloud computing’, Parallel & Distributed Processing, 2009, IPDPS 2009, IEEE International
Symposium, pp.1–8.


Integration Perspective.


Voas, J. and Zhang, J. (2009) ‘Cloud computing: new wine or just a new bottle?’, IT Professional,
Vol. 11, No. 2, pp.15–17.


