



# “Cloud Computing: Overview, Concepts and Business Deployment Scenarios”

## Bachelor's Thesis

Author: **Ivailo P. Sokolov**, Student ID 0351477

Under the supervision of:

a.o. Univ. Prof. Dr. **Rony G. Flatscher**

Institute for Management Information Systems,

**Vienna University of Economics and Business**,

Augasse 2-6, A-1090 Vienna, Austria

## Abstract

*This bachelor's thesis tries to outline and assess strategic business and technology aspects of cloud computing. Theoretical background and overview is presented on the basic underlying principles - autonomic and utility computing, software-, platform- and infrastructure as a service, as well as virtualization and grid computing. Their relation to cloud computing is explored and a case for scaling out vs. scaling up is made and scaling out of relational databases in traditional application is stressed a bottleneck. By evaluating strategic issues and weighting in business adoption pros and cons I argue that several of the cons currently outweigh many of pros. I specifically point out cost efficiency questions, vendor lock-in effects leading to operational risks to be prevailing for the majority of larger business customers that could potentially mandate their IT and computing needs from the cloud. Leading current cloud architectures are compared, esp. Google AppEngine, Microsoft Azure, Amazon EC2 as well as Hadoop/Yahoo. After presenting socio-economic and long-term implications I argue that the process of cloud business deployment will be gradual, but also that government regulations and legal aspects are also likely to slow down the process further. Ultimately, I conclude with an outlook and recommendations for companies and cloud providers.*

## Abstract (Deutsch)

*Diese Bakkalaureatsarbeit versucht die geschäftlichen und technologischen Aspekte des Cloud Computing zu hinterläuten. Der theoretische Hintergrund erläutert die zugrundeliegenden Prinzipien - autonomic und Utility Computing, Software-, Plattform-, and Infrastruktur-as-a-Service, sowohl Virtualisierung und Grid Computing. Die Beziehung zu Cloud Computing wird erklärt und ein Argument für scaling-out vs. scaling-up gebracht, wobei die Skalierung von relationalen Datenbanken als Engpass identifiziert wird. Unter Rücksichtnahme auf strategische Rahmenbedingungen und Abwägung von Vor- und Nachteilen, stellt sich heraus dass wenige Nachteile die vielen Vorteilen überscheinen. Ich gehe speziell auf Kosteffizienzfragen und Vendor lock-in Effekte, die zu operationalen Risiken für die Mehrheit der grossen Geschäftskunden führen können. Populäre Cloud Computing Architekturen werden verglichen, allen voran Google AppEngline, Microsoft Azure, Amazon EC2 sowie Hadoop/Yahoo. Nach der Darstellung sozioökonomischer und langfristiger Implikationen, komme ich zu dem Schluss, dass der Prozess der Verbreitung von Cloud Computing ein gradueller sein wird. Staatliche Regulierungen und rechtliche Fragen sind auch nicht zu unterschätzen. Ich schliesse mit einer Trendübersicht und Empfehlungen für betroffene Firmen und Cloud Providern.*



## **Keywords**

cloud computing, cloud architectures, utility computing, scalable web applications, SaaS, IaaS, PaaS, cloud computing analysis, cloud computing adoption, autonomic computing, Amazon EC2, AppEngine, Azure

## Table of contents

A list of the main topics in this bachelor's thesis:

### 1 Introduction and Motivation

1.1 Autonomic and Utility Computing

1.2 Grid vs. Cloud Computing

1.3 Cloud Computing Definitions

1.3.1 Virtualization

1.3.2 Service-oriented Architecture

1.3.3 Software as a Service

1.3.4 Infrastructure as a Service

1.3.5 Platform as a Service

### 2 Potential Business Advantages vs. Setbacks in Reality

2.1 Introduction

2.1.1 Scaling Out vs. Scaling Up

2.1.2 The Different Forms of Vendor Lock-In

2.2 Business Adoption Concerns, Standardization Issues

2.2.1 The Start-up Case vs. Big Enterprises. Marketing Aspects

2.2.2 Software Licensing Models and Open Source Usage

2.3 Cost Efficiency Issues

2.3.1 Providers Passing Down Lower Hardware Costs to  
Customers

2.3.2 Own Server Virtualization vs. Cloud Services Provision

2.3.3 Fixed/Threshold vs. Supply/Demand Price

Determination

2.3.4 Unexpected Additional Costs In Testing and  
Debugging

2.4 Privacy and Security Issues

2.4.1 Data Security – Confidentiality and Availability

2.4.2 Cloud Provider Malfeasance

2.4.3 Uptime Guarantees



- 3 Strategic Implications of Cloud Technology
  - 3.1 The Socio-Economic Perspective
  - 3.2 Regulatory and Jurisdiction Issues
- 4 Current Cloud Architecture Technologies and Platforms Comparison
  - 4.1 Amazon Web Services. Elastic Compute Cloud and Simple Storage Service
  - 4.2 Microsoft Azure Services
  - 4.3 Google App Engine. BigTable and MapReduce
  - 4.4 Hadoop and Yahoo
  - 4.5 Apple and iPhone in the Cloud
  - 4.6 Business Software Clouds - Salesforce.com
- 5 Outlook and Conclusions
- 6 Bibliography

# 1 Introduction and Motivation

Cloud computing is a very current topic and the term has gained a lot of traction being sported on advertisements all over the Internet from web space hosting providers, through data centers to virtualization software providers. Cutting through the hype of cloud computing is not an easy task as a simple web search suffices to convince that there are nearly as many definitions on what constitutes 'cloud computing' as there are players in the market seeking to gain new territory in that promising new business field. IBM, Amazon, Microsoft, Google, Yahoo and Apple among others are very active in the area of cloud computing. They either already provide cloud computing commercial solutions in one form or another, or actively sponsor research centers, pursuing development of marketable technology. Marketing rhetoric notwithstanding, the academic world has also provided meaning and definitions on what does or should cloud computing aim at and what are typical services that are expected to be encompassed by the definition of cloud computing, as evidenced for instance by the work of the RAD lab at the University of California at Berkeley or the GRIDS lab at the University of Melbourne. Other renown companies such as Oracle are eager to provide “new offerings that allow enterprises to benefit from the developments taking place in the area of Cloud Computing ” [O\_Web], yet they attempt to steer clear out of the hype and highlight that they have “redefined cloud computing to include everything that we already do”, as stated by Oracle's Larry Ellison. [O\_CEO] Furthermore, differently colored opinions on cloud computing technology from industry speakers and experts ranging from praise and optimism to critique on the viability and feasibility along with concerns on privacy, security and not least cost efficiency of the currently offered cloud computing models are available as white papers and and seem to be broadly discussed within the IT community.

Such complex technology and business models setting entails an extensive research and provides the motivation towards writing this bachelor thesis. The main goal is to “clear the air on cloud computing” and provide an unbiased and independent, albeit critical outlook of the technology. As the title of this thesis suggests its aim is to enable the reader to gain an overview of the vital aspects of cloud computing in a three-fold way: by a) providing common definitions of the important terms; b) by setting apart the advantages of the technology and the disadvantages and problems inherent to it; and c) by ultimately delivering concrete technical and business model details on popular cloud architectures, offered by the big players in the field. Special emphasis is put on the critical examination of each strategy as now more than ever in the face of the global economic crisis, companies face higher refinancing and investment costs and as any company thinking about adopting or moving to cloud computing technology would do in practice, short-to-medium term disadvantages of the technology have to be pragmatically and carefully weighted out against any hyped long-term potential efficiency achievements, be it strategic, technical or cost related.

## **1.1 Autonomic and Utility Computing**

In order to understand the vision, goals and strategy behind cloud computing, two key concepts that form its foundations need to be explained first. What seem to be the promising advantages of autonomic computing - systems that manage themselves, coupled with the flexibility and freedom of utility computing mark the core values of the business proposition offered by what is referred to as 'cloud computing'.

Autonomic computing, the term initially being introduced by IBM's Senior Vice President Paul Horn to the National Academy of Engineers at

Harvard University in 2001 [IBM\_AC\_Vision], represents a research aim towards achieving self-managing computing systems, whose components integrate effortlessly. They would do so in a biological manner as they are inspired by research in natural and social economic networks. Their primary target is by developing 'autonomic elements' to combat the ever growing complexity of integrating and interconnecting the myriad diverse software systems that still continue to emerge exponentially throughout all areas of IT. This was pinpointed as the allegedly upcoming "software complexity crisis" in a manifesto released by IBM in 2001. [IBM\_Manifesto] IBM defines four major concepts that would differentiate current computing from autonomic computing. A parallel could be drawn from these four characteristics to the desired characteristics one would want (or expect) systems deployed 'in the cloud' to possess: [IBM\_AC\_Vision]

- **Self-management** - automatic configuration of components according to high-level policies. This would assure seamless adjustment of the rest of the system
- **Self-optimization** - components strive proactively to optimize their own performance. That would account for a continuously improving efficiency of the whole system in general
- **Self-healing** - the system in general diagnoses and removes software (IBM cited even hardware) issues. Thereby the system should ideally self-repair and self-maintain to the extent possible
- **Self-protection** - the system defends itself from malicious attacks and cascading failures. IBM also cited an 'early warning' mechanism to prevent systemic failures

According to IBM, even though their own claims for such a high degree of automation might seem like science fiction, increasingly autonomic

systems in their vision would not spawn out of nowhere, but rather gradually as engineers add more and more sophisticated autonomic managers to existing humanly managed elements. However, IBM states in addition two necessary attributes to autonomic computing, that taken in the context of cloud computing seem to be problematic and relevant right now, long before the significant engineering challenges towards developing all the fancy autonomic systems are overcome. These purely organizational challenges are:

- **Privacy policies and laws** - autonomic systems must appropriately segregate and protect private data (not even remotely mentioned how)
- **Open standards** - the system must rely on such, also including its communication protocols; it cannot and shall not exist in a proprietary world, says IBM. Additional arguments are provided further on in this thesis that emerging cloud computing providers almost naturally expect to perform a 'vendor lock' into their proprietary world on their clients, which reflects both the customer as well as the cloud computing industry negatively.

Utility computing is the second key concept that one encounters in all cloud computing models. It is by no means a new concept as articulated in one form or another as early as the 1960s [McCarthy 1961, MIT; Wikipedia] [Kleinrock 1969, ARPANET][UNI\_MELB\_08] and implies that it is only natural that at some point computing power will be offered as a standardized service billed on actual usage with very limited or no upfront set-up charges. As argued later on in this thesis, inherent to this type of services is in practice the problem of negotiation and definition of a Service Level Agreement (SLA) between the cloud computing provider that promises to deliver a certain service as a utility, i.e. storage and/or processing power 'in the cloud', and the client that needs a certain level of Quality of Service (QoS) that needs to be guaranteed - monitored and enforced. [Grid\_SLA] Paramount to SLAs is the

QoS measure of guaranteed uptime. Uptime requirements are estimated to be currently anchored (notwithstanding of necessity to do so for all applications) at 99,99% in the SLAs of most enterprises (e.g. for provision of hosting, storage, etc. services), yet current cloud providers are not yet prepared to match those levels. [McKinsey\_09]

## 1.2 Grid vs. Cloud Computing

In one of their earlier white papers on the topic IBM specifically highlight the differences or rather the evolution of cloud computing over grid computing. [IBM\_CC] The term “grid computing” denotes dividing a large task into many smaller ones that run on parallel servers. [IBM\_CC] Wikipedia provides a vague, citation-less definition broadly describing grid computing as “a form of distributed computing whereby a 'virtual super computer' is composed of a cluster of networked, loosely coupled computers acting in concert” to carry out processor-intensive large tasks. [Wikipedia] RAD at Berkeley summarize shortly, without elaborating further on the separation of the two concepts, that “grid computing” suggests protocols that offer a form of shared computation over long distances, but (contrary to cloud computing) those protocols and software solutions had however not grown beyond their communities. [Berk\_ATC] According to IBM, the key advantage is that cloud computing not only is able to divide a large computational task into many smaller tasks to run on parallel servers, but could also support “non grid environments”, such as a three-tier web site architecture (i.e. separation of presentation, application logic and database, e.g. the Model-View Controller (MVC)) that runs “standard or Web 2.0 applications”. By that IBM probably stresses that large, resource intensive community websites could be built and run efficiently upon cloud architectures. Other researchers conclude that cloud computing is most likely to bring about the advantages of grid computing as “the single point of access for all the computing needs of the

customers". [UNI\_MELB\_08] Chapter 2. outlines the advantages and Chapter 3. the problems stemming from the technology.

### 1.3 Cloud Computing – Definitions

In a survey conducted in July 2008, Cloud Computing Journal cites the attempts to correctly define 'cloud computing' of 21 independent experts – practitioners and academics. [CCJ\_21\_Exp] The opinions differ, but a pattern is found such that the wording in almost all explanations hovers around the keywords scalability, on-demand, pay-as-you-go, self-configuration, self-maintenance and Software as a Service. IBM takes a technical stance and considers a 'cloud' to be a pool of virtualized resources that hosts a variety of workloads, allows for a quick scale-out and deployment, provision of virtual machines to physical machines, supports redundancy and self-recovery and could also be monitored and rebalanced *in real time*. [IBM\_CC] A scientific definition is proposed by the GRIDS Lab at the University of Melbourne:

*"A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers."* [UNI\_MELB\_08]

The researchers thus emphasize that a 'cloud' is thereby not only a combination of clusters and grids, but is also extended by the implied usage of virtualization technologies such as Virtual Machines (VMs) to meet a specifically *negotiated* service quality level. This definition implies and captures two potentially problematic issues of a) the business issue of negotiating *the proper* SLA from the customer's perspective and b) of having the technical capacity to correctly account for and guarantee the service outlined in that SLA at all (resource monitoring, failure redundancy,

rebalancing of workloads, etc. from the provider's perspective). Hence, GRIDS' definition seems somewhat more neutral than Berkeley's one:

*“Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services (Software as a Service - SaaS). The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the public, we call it a Public Cloud; the service being sold is Utility Computing.” [Berk\_ATC]*

Berkeley's researchers moreover limit their definition with several additional assumptions – a) the ability to “pay-as-you-go” as the necessary billing model, implying Utility Computing as inherent to Cloud Computing and b) that users “keep their data stored safely in the infrastructure” whilst offloading their problems to the SaaS Provider that offloads his problems to the Cloud Computing Provider. UC Berkeley's definition on cloud computing is firm with regard to zero capital expenditure for taking advantage of cloud resources. This differentiation is important, due to the apparent capital expenditures associated with making private cloud environments (definition follows in the next section) publicly available. [McKinsey\_09] Berkeley's researchers propose a three way model for provision and usage of “cloud services”, that could also be seen as recursive in case of mash-up provider that is a cloud user of another platform at the same time:

*Cloud Provider → SaaS Provider / Cloud User → SaaS User*

Berkeley refrains specifically of the usage of the terminology Infrastructure/Hardware as a Service and Platform as a Service, that is commonly found in cloud computing explanations by industry experts and academics – rather Utility Computing (used here again interchangeably with Cloud Computing) is classified in three models – Computation, Storage and

Networking. In addition to IaaS and PaaS (explained later on in this chapter), some experts [Linth\_D] list the following building blocks of cloud computing:

- Storage-as-a-Service
- Database-as-a-Service
- Information-as-a-Service
- Process-as-a-Service
- Application-as-a-Service
- Integration-as-a-Service
- Security-as-a-Service
- Management/Governance-as-a-Service
- Testing-as-a-Service

In addition to delivering meanings to the above mentioned concepts later on in this chapter, theoretical background is also given on virtualization and Service-Oriented Architecture (SOA). Points of critique and drawbacks of cloud computing are dealt with in Chapter 3.

### **1.3.1 Virtualization**

In order to further clarify the term virtualization in the context of cloud computing, a short background of the concept is provided in this section with the intent to present an overview on its multiple aspects. Popek and Goldberg proposed in 1974 a now often cited definition of a Virtual Machine (VM) which is an “efficient, isolated duplicate of the real machine”. [Pop\_Gold\_74] In the 1960s it was virtualization was designed to partition mainframe hardware but gradually declined in popularity as PCs became more widespread that provided a more efficient and affordable distribution of processing power. [Turban\_08] Yet nonetheless in the recent years that has changed due to

decreasing hardware costs, increasing common hardware power and the resulting underutilization of resources (what is now referred to as commodity hardware on which cloud computing datacenters rely as their processing power source [Berk\_ATC]). [Turban\_08] Virtualization in general implies many tasks related to the abstraction of computer resources. Those may include:

- **OS/platform virtualization** - involves “aggregation and sharing of physical resources from the way in which other systems, applications or users interact with those resources”. [Turban\_08] In effect the user or user application sees an abstract, emulated version of the computer platform, e.g. a different Operating System (OS). [Wikipedia, “platform virtualization”]
- specific **hardware resource virtualization** - includes virtual memory and RAM virtualization, storage separation logical/physical (e.g. RAID platforms - redundant array of independent disks), network resources virtualization
- **software application virtualization** - refers to various emulation techniques, such as to enable cross-platform (OS) execution and usage of software. Thereby an abstraction layer is created that runs the software compiled for the other OS. This layer is defined by Popek and Goldberg, whereby system instructions are divided into three groups (privileged, control sensitive and behavior sensitive) and their theorem states that to be virtualizable the computer has to have its control sensitive instructions to be a subset of the privileged ones. [Pop\_Gold\_74] IBM provides an applied example of virtualization in their vision on possible uses in education, highlighting that applications and information is presented in a consistent way, regardless of geographical location or physical equipment of the user, both in a thin and thick client way. [IBM\_Virt\_Edu]

- **virtual application appliance (VAA)** – is a virtualized software application that decouples other software applications from the underlying OS. This is done to optimize running (especially single PC 'standard' and non-scalable) applications in a virtual infrastructure, e.g. in a cloud environment. As a result a discrete object is formed that contains all of the program's dependencies, including executables, function libraries, files/registry, configuration settings and network identity [VAA\_AppZero]
- virtualization in **computer clusters and grid computing** – it is an integral part that allows for multiple discrete computers to be joined into a larger virtual supercomputers [Bader\_D] Load-balancing clusters provide management and distribution of the computational workload, thereby improving the overall cluster performance, yet still appearing to the end user logically as a single (virtualized) machine. [Wikipedia, “Computing Clusters”]

Examples of virtualization platforms are VMware ESX and Microsoft Hyper-V. They provide the infrastructure and management tools to instantiate, run and maintain virtual machines in a reliable manner. Thereby virtualized are multiple guest OS's, network resources and the system management as well as single software applications. [F5\_Virt\_Gd] They are typically the first form of virtualization that is introduced in the data center. [F5\_Virt\_Gd] Advantages are the apparent consolidation of IT resources and the thereby resulting cost savings, immediate drawbacks include complexity and scale management challenges, even though these virtualization platforms come in the form of packaged software that only needs configuration. The managed pool of internally virtualized resources is what UC Berkeley refers to as an “private cloud”. [Berk\_ATC] To illustrate the parallel with cloud computing one needs to consider how cloud computing technology has evolved as it by no means appeared out of nowhere. Thus, it has been

suggested that 'the cloud' actually is “massive implementation of virtualization”. [J\_Hurwitz] Irving Wladawsky-Berger, Chairman of IBM Academy of Technology points out that the path from simple grids, through large virtualized systems to cloud computing and fully distributed computing is a continuous process and although that path is complicated and disruptive. In his view, SOA based protocols and encapsulated software components with clearly defined interfaces should facilitate the move from single system virtualization to systems wide virtualization possible on a massive scale. [J\_Hurwitz]

McKinsey & Company argue that large enterprises can achieve server utilization rates similar to those cloud providers are reporting to be achieving from their platforms by what they call “aggressive virtualization” of the company's existing IT resources. In their view, even “quickly achievable virtualization”, i.e. by simply adding an additional layer of virtualization leveraging more stacking and consolidation, significant overall utilization gains, as well as cost advantages over completely moving to the next, “cloud layer” are feasible for most companies. [McKinsey\_09] Section 2.3 presents arguments for those claims in detail.

### **1.3.2 Service-oriented Architecture**

Service-oriented Architecture refers to a modular design principle in software architecture. Service-orientation aims at separating individual functions into distinct units or “services”, that could be accessed, e.g. via a network, by developers to integrate them in a reusable manner in their applications. [Wikipedia, “SOA”] Paramount is the loose coupling of those services to programming languages and specific underlying platforms, i.e. the services communicate with the applications (or other services) that invoke

them via their predefined interfaces. Ideally, those should be standard, available, documented and easily implementable. IBM suggests the following guiding principles towards designing a service – it should be granular, componentized, encapsulated, leveraging existing modules, having life cycle management and complying to common industry and IT standards. [IBM\_SOA] Ultimately, SOA based applications should leverage a multitude of already developed services – purposefully designed, stateless pieces of business logic that compute specific tasks and deliver clear and usable results in return. XML/SOAP protocols are examples of commonly used for building SOA applications and utilizing web services (services accessible via HTTP protocols).

From the business perspective SOA should allow for reuse of existing investments through leverage of already bought technology, evidenced e.g. as plenty of companies are creating services extracted from existing applications to be mandated for further standardized usage company wide in the enterprise SOA. [SOA\_Dummies\_2ed] Moreover, by deploying a flexible SOA in the enterprise, existing systems could be changed more flexibly to accommodate for changing business and user needs. [Redhat\_JB\_Dell]

SOA, as an architecture design principle is a necessary ingredient towards enabling any of the cloud computing models and paradigms mentioned in this thesis for two key reasons:

- **Firstly**, the term Service-oriented Infrastructure (SOI), as defined by HP, Cisco and Capgemini to be the “virtualized IT infrastructure in an industrialized way” [Wikipedia, “SOI”] manages a multitude of services as well as SOA applications. Intel reaches to draw a parallel with autonomic computing and further enhance the understanding of the

“SOI layer” with a couple of very high-level tasks such as management of virtualization, load balancing and capacity planning, monitoring and problem diagnosis, security enforcement and utilization metering (incl. SLA compliance). [Intel\_SOI] If and when, theoretically, systems (including those of normal, non-IT enterprises) are made to be capable of such seamless encapsulation, abstraction and management of whole computing resources, moving, providing or acquiring them from the cloud would be the next logical thing to do. However, for the time being this as well as most of IBM's vision of autonomic computing, remains largely wishful thinking, yet still points towards the general trend in automating enterprise IT resources.

- **Secondly**, any software or software platform that is to be 'offered as a service' or “provided in a pay-as-you-go manner” [Berk\_ATC] should be designed along SOA principles. Customers, or software application developers would thereby theoretically design their systems to be modular or use other's modules and ideally pay only for the components they need (if they are drawing on external pieces of code). Although apparently not a new concept at all, I would argue that the trend towards cloud computing and the resulting need for more interoperable systems (as they are hosted/executed in the cloud) would naturally strengthen the case for SOA based software. According to some experts, SOA is already anchored to a point that in ten years from now there will be no market segment for SOA software, “as this would be the way things are done”. [J\_Hurwitz\_SOA]

### 1.3.3 Software as a Service

Software as a Service (SaaS), the term coined by IDC and summarized by SIIA in 2001, initially referred to the Application Service Provider (ASP) model

in general, and the shift from desktop/packaged software towards web-based, outsourced solutions in particular. [SIIA\_SaaS] The software is provided over a network, e.g. the Internet, but also VPN, and is provided to the users in a recurring fee basis. The SIIA, which “promotes the common interests” and “protects the intellectual property rights” of the traditional software companies [[www.siiia.net](http://www.siiia.net)], recognized early on that “packaged software as a separate entity may even cease to exist” and outlined the resulting new value chain from the move to SaaS provided over the Internet. SIIA highlighted the differences among the service delivery models – server-based computing (so called 'thin client'), hosted client computing (run on users' desktop), 'normal' web applications and Java applications and provided a business case example with Enterprise Resource Planning (ERP) – SaaS from SAP – mySAP.com. [SIIA\_SaaS]

UC Berkeley define cloud computing to be “the sum of SaaS and Utility Computing” without private clouds. In their view cloud computing has all the advantages of SaaS, including ease of installation and maintenance, access from anywhere and centralized control over versioning. In addition, the fact that the “data is kept safely in the infrastructure” is listed as advantage. In their view cloud computing would allow SaaS deployment, without building or provisioning a datacenter. [Berk\_ATC]

Examples of multi tenant SaaS provision from cloud managed environments are Workday.com Inc and Salesforce.com Inc. Workday offers HR and Payroll software, whereas Salesforce.com offers Customer Relationship Management (CRM) software. To highlight SaaS' advantages both companies emphasize on their websites the fact their SaaS delivery models are “the opposite of ERP” and “not software”. To mitigate the potential security question that arises, both companies eagerly outline the number of

big enterprise customers that have entrusted them with their data. Section 3.4 sets apart the privacy and security issues with placing data 'in the cloud'.

### 1.3.4 Infrastructure as a Service

Infrastructure as a Service (IaaS) refers to renting raw hardware, with SOI as the underlying principle for managing its resources. In cloud computing models those resources are virtualized and by using statistical multiplexing the “illusion of infinite resources” [Berk\_ATC] is achieved as cloud environments can scale out. In essence, an additional layer of virtualization is applied over the actually virtualized instances of OS', which are extracted from preconfigured images (e.g. GoGrid's Windows 2003 or RedHat Linux images) and set up to run as VMs with differently adjusted RAM, CPU and hard disk storage capabilities. The ability to procure or rent virtualized dedicated servers from web hosting providers is nothing new and has been around for years. But that additional level of abstraction and management of the resources that enables invocation and configuration of additional (thus to the user 'infinite') resources (e.g. new instances of servers, more RAM or more storage when needed) is what I argue to be the major key twist to be attributed to cloud computing. That additional level of abstraction may include also communication abstraction such as load balancing, i.e. evenly spreading the incoming traffic to running (not crashed) web servers (e.g. F5 hardware load balancing, offered by GoGrid) as well as abstraction of storage that allows the databases to be scalable in line with the rest of the system.

The most notable IaaS cloud offerings appear to be Amazon's (e.g. Elastic Compute Cloud (EC2), part of Amazon Web Services and covered in Section 4.1). GoGrid's Cloud Hosting/Storage (they advertise themselves being

the first to provide a web Graphical User Interface (GUI) to managing cloud environments) as well as Mosso's Cloud Servers/Sites/Files seem to be major contenders in the IaaS category. Billing models include monthly subscription plans or 'pay-as-you-go' billing in addition to data transfer - 'server RAM hours' (GoGrid), 'server hours' (Mosso) and separate cloud storage billing.

### 1.3.5 Platform as a Service

In the context of cloud computing, Platform as a Service (PaaS) represents an intersection between IaaS and SaaS. It is a form of SaaS that represents a platform provided to serve as the infrastructure for development and running of new applications in the cloud. Benefits include the ability to build and deploy scalable web applications without the cost and complexity of procuring server and setting them up. [Grd\_17\_04] Prominent examples of PaaS are Google AppEngine (outlined separately in Section 4.3) and Salesforce.com's business software development's platform - Force.com. According UC Berkeley's classification those application domain specific platforms (RAD refrains from using the term PaaS) are “at the other extreme spectrum” (vs. IaaS explained in 1.3.4 above) and are not suitable for general purpose computing due to their proprietary database and constraints and stateless-only computation tier.

Microsoft Azure Services on the other hand is a platform that supports general purpose computing by enabling applications written in a .NET language (C#, VB.NET, J#) to run in the cloud environment managed by the underlying Azure OS. The user has access to some functions that could be integrated in their application code to better take advantage of the automatic scalability properties of the distributed cloud environment (including fail over capabilities), but the user has no control whatsoever on the underlying

Windows Azure OS [Berk\_ATC]. Windows Azure acting as the managing OS, will provide automatic load balancing, geo-replication, application life cycle management and many additional features such as SQL Services, .NET Services, Share Point Services etc [MS\_Azure\_FAQ]. However, the platform is highly proprietary and its drawbacks are discussed in Sections 2.2 and 4.2.

## 2 Potential Business Advantages vs. Setbacks in Reality

This chapter aims at dismantling the pros and cons of cloud computing, while retaining a pragmatical and independent point of view. Advantages and pros spread faster as the technology gains traction – prominent examples are outlined in the introductory section. The goal is however to critically set apart the following, more unpopular key aspects:

- marketing claims for *future potentials* vs. current *technical capabilities*
- business models for which cloud computing *makes sense* vs. those for which it does not – arguably, the *majority* of IT spending
- different types of *vendor lock-in* effects – explained and weighted
- security issues to which more concern should be paid

### 2.1 Introduction

"Cloud computing is... the user-friendly version of grid computing."

Trevor Doerksen [CCJ\_21\_Exp]

"It's stupidity. It's worse than stupidity: it's a marketing hype campaign"

Richard Stallman [Guardian\_Sep08]

*"Cloud computing is the same old client-server computing we've known for years, except pretending to be intoxicatingly new and different and liberating. Today's so-called cloud isn't really a cloud at all. It's a bunch of corporate dirigibles painted to look like clouds."*

Peter Lucas, Joseph Ballay, Ralph Lombreglia [Maya\_WrCl]

To introduce my pros and cons analysis I will first look at the pros. Cloud computing implies, inter alios, the effectiveness of resource usage and scalability of grid computing. Grid computing architectures are not easy to set up as they imply complexities of all sorts - middleware and network configurations among others. However, as grids are 'taken to the next level', here are some of the often quotes pros from the business perspective of companies to move to cloud computing (cons are discussed extensively further on in this chapter):

- **metering of standardized resources usage** based on actual consumption - utility computing and pay-as-you-go models are introduced to charge the customer for hardware usage, be it server-RAM-hours, gigabyte-storage-hours, CPU-hours, etc. Thus, in addition to the currently spread standardized-server-configuration-hours (for renting a dedicated server from a datacenter) and network bandwidth usage (GB of data transferred), more flexibility is introduced as resources are relinquished after no longer being needed
- **elasticity** - scalability and load-balancing of the server resources are built-in. Thereby short-term automatic provision, enabling invocation of additional resources is paramount. The benefits of this could be enormous to companies that experience frequent and significant changes in computing or storage needs.
  - **Service unavailability** and therefore lost-customer-costs are avoided as all potential computing needs/server requests are possible to be met. A classic example may include social networks that receive a sudden surge in popularity ("victim of own success"), a web shop

- during peak pre-holiday times, but also a news or company website (e.g. an airline) that, due to critical events receives an overwhelming amount of traffic that requires more than the planned/available computing resources, in order for all of the requests to be served
- The ability to perform **large batch jobs** (e.g. a very resource intensive calculation, a data mining or business intelligence task) in very short periods of time to the benefit of faster business decisions. Should a company decide to outsource such a seldom required, but very resource-intensive task once required, arguably drawing onto additional processing power from the cloud could be very cost efficient. Google's MapReduce or Hadoop are examples of such technology that supports extracting results of distributed computational tasks over multiple nodes with petabytes of data (see Chapter 4)
  - The ability to run large parallel tasks in **parallel processing pipelines** – including document processing (e.g. text file generation from scanned images text recognition), image and video/audio processing (cumbersome format conversions), indexing, log analysis, data mining. In addition, technical tasks such as automated unit testing and parallel deployment testing on different configurations are suitable. [AWS\_Varia]
  - **no capital expenditure** on hardware (as well as software) that performs the computing needs. These are the fixed costs associated with one time purchases of IT infrastructure that are amortized over time. They are converted to operating expenses for renting the resources of the cloud provider. UC Berkeley states that cloud providers would be able to leverage economies of scale and procure hardware at 1/5 to 1/7 the prices of medium-sized datacenters (having up to 10000 computers) [Berk\_ATC]
  - **uncomplicated deployment** as well as availability of autonomic management features that lead to easier and less costlier maintenance,

i.e. less personnel costs of the cloud provider for managing a given pool of server resources (e.g. administrators per 1000 servers), thus the ability to offer the resources at lower prices.

All of the above listed pros lead to *faster time-to-market* as well as *lower specific project costs* related to the implementation of a given software solution in a cloud rather than a traditional internal IT department or datacenter. However, as argued further on in this chapter, those benefits are easily pinpointed if one were to “create the next Facebook” or “the next YouTube”, but largely questionable if one were to move their on-premises or own datacenter existing computing resources to the cloud.

In IBMs view, [ESN\_CC] security is also increased and they also cite a combination of public and private clouds, that share the same infrastructure but the clouds are isolated through own firewalls. According to IBM, “if you had one small cloud, it can never be as efficient as a huge pool of IT resources.” [ESN\_CC] Once of the most controversial advantages, ensuring the **physical security of the data** is also mentioned by IBM, as theft or damage of on-premises hardware (by breaking and entering) is potentially left out.

Moreover, in UC Berkeley's view, a whole new aspect to cloud computing is the “illusion of infinite resources” [Berk\_ATC]. However, largely marketing terminology, “infinite resources” arguably refers only to the arbitrary addition/removal of additional computational nodes, i.e. VMs/servers and does not imply infinite storage or network scalability. As I argue in this paper, the scalability effects provided by that additional layer of virtualization over the pool of virtualized resources (see Section 1.3.4) are responsible for the marketing hype surrounding cloud computing and the further level of abstraction over the underlying resources is its technology-enabler.

### 2.1.1 Scaling Out vs. Scaling Up

The scaling up vs. scaling out debate is a classic one. However, in the context of cloud computing it needs to be revisited because a) there are different strategies to solving problems related to achieving application scalability and b) not every piece of application or computing resource scales with the same success. This relates directly to cloud computing, as the advertised performance for “infinite scalability” looks very different if applied to a traditional, relational database (found in most web applications) vs. a distributed, large scale non-relational, multi-dimensional database such as Google's BigTable. Achieving scalability is generally possible through:

- **Vertical scalability**, or *scaling-up* - refers to adding more hardware capacity - additional memory blocks, a faster CPU, a bigger/faster hard disk, etc. Generally, if one invests or rents two times faster hardware (e.g. more memory and more CPUs), the system is scaled up, but the outcome effect (e.g. number of requests/users served in a web-application) is not or rarely linear. Scaling-up is especially efficient for relational database tiers, due to the large shared memory space, many dependent threads and the tightly-coupled internal structure. [Davis\_Java] The major server vendors continue to invest heavily in building bigger and more powerful shared-memory servers [IBM\_Scale]
- **Horizontal scalability**, or *scaling-out* - refers to adding additional nodes (PCs, servers) of similar capacity (CPU, RAM), connected in a cluster of small machines via a network. This solution of adding “commodity hardware” [Berk\_ATC] is particularly effective for building high-throughput web-centric applications. [IBM\_Scale] The web-tier, e.g. a social networking website is able to take advantage of this approach,

because such web applications a) use small non-shared memory space, b) have many independent threads, c) have a loosely-coupled external structure and d) run possibly on many OS's. [Davis\_Java] Google, Yahoo, eBay and Amazon all rely such cluster architecture for their computational needs. IBM concludes in a study (search of unstructured data) that a scale-out system is four times faster than a similarly priced scale-up system but at a convenience and management costs (multiple system images). [IBM\_Scale]

However, as far as cloud computing goes, it must be taken into account that:

- A “scale-out-in-a-box” solution, i.e. multiple virtualized instances of an application running concurrently within a single operation system could have performance advantages similar to scale-out systems, but without the additional system images overhead. Chapter 2.3.2 presents arguments why more aggressive internal virtualization of the on-premises IT resources – seeking similar purposes – could prove to be more cost-efficient than provisioning computing power from a cloud provider.
- Moving traditional (non-cloud-hosted) web applications to the cloud, e.g. websites that use popular relational databases, may provide limited technical performance gains in contrast to what is theoretically possible, advertised by the cloud provider, and nowhere close to what is achieved by Google/Facebook/Amazon themselves. Of course, LAMP (Linux, Apache, MySQL, PHP/Python) is also an option for placement in the cloud. [IBM\_CC] The problem is however, that relational databases do not scale out very well in distributed platforms because performing UPDATE-statements requires a lot of synchronization work when a record is stored over different machines; not without the complexity and needed expertise to set up database clusters and program the applications/database accordingly (e.g. to properly handle tables stored

over multiple machines). In fact, arguably most web applications of smaller community/presentational websites do not need relational databases at all, as many of them do not use transactional features (e.g. a simple photo sharing website). Transactional features (i.e. beginning and committing transactions of multiple data manipulations - a series of UPDATE, INSERT, DELETE, etc. SQL statements that could be rolled back) are though absolutely necessary in real time applications - e.g. in banking and in e-commerce. Relational DBMS, open source and proprietary are a well developed domain of IT and are built with an optimal mix of flexibility, performance and scalability and have inherently strong consistency through normalized data; hence popular for most web and business applications. Relational databases scale well too, but usually (i.e. unless additionally configured) only when that scaling happens on a single server node. [Bain\_09] Database clustering techniques are used to scale distributed databases [Vers07] - arguably as good as any cloud database, yet they could be considered more complex to configure and manage for non-professional and non-experienced database administrators and programmers - something against the 'one-click, on-demand, for-everyone' cloud marketing paradigm. Thus, because of its simplicity the alternative - a special (largely proprietary), schemaless, column(attribute)-oriented, key/value storage model - Amazon's SimpleDB or Google's BigTable/Datastore or similar (open source Hadoop HBase) technologies for effectively handling large amount (up to petabytes) of data dispersed among different physical/virtual machines allows for leveraging most of the hyped possibilities of *the cloud*. This means that *existing web-applications* that would like to move to the cloud and, whose owners/programmers do not have the knowledge and expertise to set up scalable relational DBMS, will need to be *rewritten* to use these new storage models. Even more so:

- the Application Programming Interfaces (API)s of the abovementioned storage models are proprietary, non-standard, and to programmers; even if a model is taken and implemented, the API would probably be changed soon [Maya\_WrCl] because
- creating an arbitrarily scalable, durable and meaningful in query richness, performance and availability persistent storage model is still an open reserach question, as conceded by RAD at UC Berkeley.

### 2.1.2 The Different Forms of Vendor Lock-In

One of the most significant aspects of any of the cloud architectures currently out there is their *proprietary nature* - be it IaaS, PaaS or any combination of both. Cloud computing is a relatively new approach towards leveraging virtualization and the fact that major players see and define the term differently could be attributed to the large competitiveness issues that arise as the technology matures and becomes widely adopted.

Companies such as Google and Amazon largely owe their rapid growth to the innovative virtualization/grids/"clouds" they have developed internally - for their own needs - and now have those resources and infrastructure in excess. Thus, from their point of view, providing that excess power to third parties and leveraging on their successful Internet-scale hosting/platform solutions might seem like a very profitable idea both for them and their potential clients/users. However, other big Internet companies seem to have reached that point too - e.g. Microsoft and Yahoo and they would also like that market share.

Thus, a customer is left to choose *one* from the *many* competing, yet different concepts, deploy their business there and stick with it. The resulting fierce competition leads any cloud architecture provider to try to lock in as

many clients, users, testers, developers and vendors in their technology as possible. It is at the time being obvious that until the technology has matured and the market is settled/saturated, that the least those cloud providers would like to do is to cooperate with each other, instead of compete with each other. Competition will, of course, prove positive for the development of cloud computing technology in the mid/long run. However, it will be negative in the short run for any customer stuck with a cloud implementation from a provider platform that is temporarily offline/unsuccessful/bankrupt for many reasons - their data is locked-in and their non-standard APIs are not reusable elsewhere.

Arguably, a company that moves its applications and data to a cloud computing provider would suffer much more operational risk costs than it would when merely switching the company's inter/intranet web site to a different hosting provider or even when choosing other vendor's software. I would compare the business's dependency on the deliverability, sustainability, future pricing outlook, support and maintenance capabilities of the cloud provider to those when choosing an Enterprise Resource Planning (ERP) system. For instance, should a big company need to extract their data and applications from one cloud provider and move to another (having presumably different APIs), the **costs** associated with manual data manipulation, data transfer (bandwidth) and software re-programming could easily outweigh any cost savings that the cloud infrastructure presents over own datacenter virtualization.

Until standards for interoperability between cloud providers exist, any business that adopts cloud computing has a disproportionate disadvantage and as I argue in my concluding outlook, many serious IT departments would not "move to the cloud", for that reason alone, no matter how compelling the advantages (see outlook - Chapter 5). In addition, serious business continuity concerns could be raised due to lock-in effects, as cloud architecture

providers could be subject to police investigations and thus uptime could be seriously disrupted (see Chapter 3).

## 2.2 Business Adoption Concerns, Standardization Issues

The vendor lock-in situation, described above in Section 2.1, is even clearer understood if platforms such as Google's AppEngine or Microsoft Azure Services are used, for which the application code should be separately developed and is different than other, 'normal' web applications (e.g. Java, PHP, etc. based). To illustrate this, I take a look at Microsoft's Azure Platform and argue that it will further exacerbate the lock-in effects, as a plethora of companies developing .NET applications are expected to recommend and deliver even more Azure-specific programs to their clients, effectively locking-in even more clients into Microsoft's solution.

It is fundamental for companies that would adopt Azure-based SaaS to understand that Windows Azure OS will run only at Microsoft data centers, as Microsoft does “not envision selling Windows Azure for on-premises deployment”, due to the alleged complex structure and features of their multi-tenant global datacenter. [InfoWeek\_Az] The intention is for Microsoft to provide an “on-demand” vs an “on premises” platform that will allow for scale out SaaS applications. [Chappell\_Az]

David Chappell, a paid consultant for Microsoft Azure's publicity, draws an easy comparison with normal proprietary, off-the-shelf software:

*“Windows Azure runs on machines in Microsoft data centers. Rather than providing software that Microsoft customers can install and run themselves on their own computers, Windows Azure is a service: Customers use it to run applications and store data on Internet-accessible machines owned by Microsoft. Those applications might provide services to businesses, to consumers, or both.”*

Microsoft has also stated that they will offer their SaaS applications on the Azure Platform too. As a side effect software will become cheaper for the customers but Microsoft's margins are expected to get significantly lower than their off-the-shelf software products [Ray Ozzie, Microsoft's Chief Architect - June 14, 2009]. Regardless of that, users will not only be locked-in to using closed source applications, but also be locked-in to using a single available entity to execute and deliver them.

The same applies to Google or any other PaaS cloud provider. Therefore it could be argued that in case the proprietary pioneer cloud providers gain enough critical mass of users to dominate before any industry standards, APIs and interoperability definitions are in place to assure competition and redundancy, cloud computing could prove to be even dangerous and destructive. IT-strategy consultancy Maya Design [Maya\_WrCl] draws a parallel with the financial sector meltdown, arguing that radical experiments (evidenced by moving large IT departments to the cloud) should not be performed by non-redundant entities, but cloud computing should rather develop in a decentralized (Peer-To-Peer), parallel and standardized environment. In a June 11<sup>th</sup> entry in RAD at UC Berkeley's blog [RAD\_blog], peer-to-peer is argued to be unsuitable for cloud computing, as the lack of a centralized administrative entity makes it hard to ensure high levels of availability and performance and the low connectivity between the nodes is deemed inappropriate for data intensive applications.

Thus, to summarize it could be concluded that despite security and privacy issues (outlined in the next sections), cloud lock-in thanks to lack of standardization is the biggest business adoption concern. In my view, when the technology matures enough and more cloud providers integrate common protocols and APIs for interoperability, the different IaaS sub-models will be

able to emerge separately as viable options for outsourcing of certain IT resources, e.g. storage-as-a-service, security-as-a-service, testing-as-a-service, etc. Only then will a serious company be able to diversify its cloud vendors and thus procure flexibly only the scalable services it needs in a less-risky, more redundant manner.

This trend is recognized by Sun with the announcement of the Sun Open Cloud Platform. Sun Microsystems will try to position itself as the pioneer of open and interoperable cloud ecosystems. Its core APIs are also released under the Creative Commons license (not all but “some rights reserved”). David Douglas, Senior VP at Sun states their conservative vision (compared to Microsoft's), explaining that they envision coexistence of public clouds and private clouds behind firewall, thus a hybrid cloud environment. [Douglas\_Sun] “Surge computing” could also spur in effect - only unmanageable tasks go from the private to the public cloud [Berk\_ATC]. Arguably, even the other way round, companies selling their own free compute resources to the could could be possible in an P2P manner if private and public clouds are technically interoperable.

Furthermore, other - niche companies - such as ParaScale, Inc - a company developing cloud storage software for both public and private deployment shares the same view how the market will develop and also relies on fully open standards and protocols. [[www.ParaScale.com](http://www.ParaScale.com)] Microsoft's management acknowledges that the “cloud market” is now immature and at some point interoperability standards should/will be implemented, highlight that they are already using open protocols SOAP, XML and REST in Azure but yet contradict themselves emphasizing that the “creation of interoperability principles and processes should not be a vendor-dominated process”. [S\_Martin\_MSFT] The vendors however have their incentives not to do actively pursue that goal. The Hadoop project must be mentioned here as an example

of the first prominent community effort to produce a standardized, cloud API environment that could be deployed at any commodity hardware provider and which, if deployed at enough datacenters, could eventually drive prices down and free customers from lock-in effects (see Chapter 4.4).

### 2.2.1 The Start-up Case vs. Big Enterprises. Marketing Aspects

An example of how a business profits from cloud computing economies of scale and elasticity is the already turned classic case of a small start-up companies, whose whole IT architecture – storage, network and computing power is provisioned from the cloud. Thereby the company is able to organically grow and meet all of the demand for its website, by scaling out in the public cloud and not having to commit any capital resources to procure server hardware. McKinsey & Co. gives examples of small businesses – SaaS and “Web 2.0” community websites such as ShareThis.com, SmugMug.com, unfuddle.com, JungleDisk.com and 37signals.com to which cloud computing is especially compelling. [McKinsey\_09] Currently, most cloud computing customers are small businesses and IBM cites even innovation through development of personal hobbies to be potentially enhanced by cloud computing technology (due to the factor zero start-up costs). [IBM\_CC]

Therefore, in a scenario where a web application, possibly optimized for scalability (scalable storage/non-relational database), is run at a cloud provider instead at a regular, non-scalable on demand datacenter, the benefits of the cloud really pay off. Often advertised is how a web site could become “the next Facebook” or “the next YouTube”. However, although the advantages for a startup web site (I would not call it “startup business” yet) are clear, I do not consider this as a proper case for marketing “cloud computing” in general. It is because the broad term *cloud computing* is narrowed down to

this niche meaning. It is obvious that in comparison to the “startups” rhetoric, found in prominent cloud computing white papers such as IBM's and UC Berkeley's, less marketing efforts are being exercised towards advertising cloud computing for bigger enterprises. This is very significant, as cloud computing targets diverse e-business and e-commerce activities, and naturally the market for moving existing companies' websites to *the cloud* is far much bigger than that of new startup websites, e.g. new social networks.

In my view, the reason why bigger companies are intentionally left out of most marketing materials, is simply because cloud computing is not - or not yet - that compelling to those that have *already invested* in extensive IT infrastructure for plenty of reasons. By infrastructure here I mean not only hardware - servers/clusters/grids, high bandwidth networks, but also software - ERP systems, data warehouses and database management systems, communication software and perhaps most importantly existing web applications interconnected with rest of the infrastructure. There are pragmatic reasons would for the time being hinder rapid cloud computing adoption at companies sized medium and above. McKinsey lists four reasons:

- **Financial** - cost efficiency issues are raised as large companies need massive amounts of computing at their disposal - the viability of one-price-fits-all models, the risks of further price increases and operational risks associated with one-entity dependence (see Chapter 2.3)
- **Technical** - many applications need to be re-engineered to fully absorb cloud technology potential as not all aspects of the IT architecture, including e-commerce web sites, are proportionally scalable. Security and reliability issues are raised in addition (see Chapter 2.4)

- **Operational** – even after the technological issues are resolved, business implementations and perceptions of the newly gained IT flexibility have to be properly managed
- **Organizational** – the operational alignments would impact the organizational structure – both IT and business units, as IT supply and demand will function differently

In addition, I would augment the cons that big companies face adopting cloud computing with the following points:

- **advancement of virtualization** technology and internal cloud technology would enable companies that already possess large hardware inventories to leverage their usage further and extend their usable lifetime simultaneously. Companies such as VMWare and Citrix as well as IBM and Sun are already under way to sell them the necessary soft- and hardware. Hence, these companies will be able to extract even more utilization from their existing infrastructures and effectively be able to scale the demand for their websites accordingly (or better perform large batch-jobs)
- **less fluctuation in computing demand** – resulting from the advancement of virtualization and better internal utilization. In effect, the *importance of infinite elasticity* might not prove sufficient to most companies, to warrant relying on public clouds
- **cloud computing PaaS** such as Microsoft Azure Services and even more limited (due to higher abstraction level) ones such Google AppEngine could very possibly not gain widespread usage at all due to their data security and lock-in restraints. However, they could serve as a proof-of-concept and prompt organizations that recognize the possibilities of highly scalable virtualized solutions and try to align their **internal**

**infrastructure** to *emulate* these models as much as possible. To illustrate, whole industries might never touch Microsoft Azure-based products that run and store data at Microsoft's servers, even though they might be using Microsoft software on-premises for decades. Banking and finance, government and public services, are such sectors

RAD lab's researchers [Berk\_ATC] argue that computing capacity elasticity is valuable to established companies as well and cite Target.com (Target is U.S. second largest retailer chain) to rely on Amazon Web Services customer for their e-commerce activities. I argue that this is a non-relevant startup vs. big company cloud computing example as **a)** Amazon is a strategic partner of Target since 2002, providing them with variable computing power services long before AWS and cloud computing [Wikipedia, Target], **b)** online sales make a small portion of Target's total revenue (strategic importance – even if the website is offline for several hours, local stores would generate revenue and **c)** Target's main computing power needs comes from within its internal IT infrastructure – from its head office ERP system to every local cash register PC, rather than their e-commerce website. Hence, rather than to convince the benefits of utility computing for the entire IT department, this example perfectly illustrates how cloud computing is useful for only *certain aspects* of its needs – namely scalability of the web site's front-end.

To summarize, the larger the company in organizational terms, the more IT infrastructure it has and the less fluctuation in computing power demand it experiences, the more likely that this company will stay out of mainstream public cloud computing adoption and the more the company will tend to make use of its internal resources. Therefore, adoption of certain SaaS solutions or use of some cloud services for scalability will not mean entire IT departments' resources will be procured from the cloud.

### 2.2.2 Software Licensing Models and Open Source Usage

Open source software has been critical to the recent growth of both SaaS and IaaS cloud computing (in contrast to proprietary PaaS vendors). Open source software has gained additional usage for IaaS providers such as Amazon Web Services or Rackspace/Mosso. They rely on Linux/Unix OS and open source databases (MySQL) for their lower cost cloud computing offerings, although Microsoft Windows/SQL Server based solutions are also available (albeit at a higher cost due to proprietary licensing fees). IBM's cloud computing architecture is also based on open source Linux and XEN (virtualization software) in addition to their own Tivoli, DB2 and Websphere server software. [IBM\_CC] Moreover Sun Microsystems also relies on “best in world-class open source technologies like Java, MySQL, OpenSolaris and Open Storage to deliver breakthrough efficiencies in cost and scale” for their Sun Open Cloud Platform. [Douglas\_Sun]

However, some experts believe that many SaaS vendors may run into licensing conflicts as they develop spin-offs and derivative works of open source software for the purposes of cloud computing. [NW\_SaaS] The General Public License (GPL) requires derivative works to be made available to the community too to ensure that freedoms are preserved. [Wikipedia, GPL] The problem is that such SaaS vendors will try to impose SaaS usage fees that are not compatible with GPL and hence be subject to license violation. It is however not clear, whether commercial SaaS applications where no binaries are not distributed will be subjected to licensing problems, provided that they are not licensed under the Affero-GPL (a version of GPL covering the SaaS “loophole” [Wikipedia, Affero\_GPL]). In my opinion though, as cloud computing attracts more attention it is inevitable that open standards, SOA-based protocols and entire platforms for cloud architectures (and services) get developed as open source projects and are made available to the community.

The focus would move to those new large scale community projects rather than long tail, non GPL-compliant open source SaaS derivatives.

Furthermore, usage of VAAs (see Chapter 1.3.1) is likely to increase as companies, especially those bought into Microsoft software are likely to experiment with setting up their license-bound applications in cloud environments. GoGrid.com is advertising AppZero to be “*the only way to move Windows' apps without breaking Microsoft's licensing*”; it is sold per server instance and price is given upon request. AppZero's ROI calculator [[www.appzero.com](http://www.appzero.com)] estimates enormous savings (due to saved licensing fees and servers running) and AppZero's price appears to be around \$500 per server license.

## 2.3 Cost Efficiency Issues

As mentioned in previous chapters, the financial viability of current cloud computing offerings could be challenged, if bigger enterprises are to procure their computing needs from the cloud rather than at own datacenters. Although experts (IDC - [ESN\_CC]) emphasize that the current economic downturn the appeal of the cost advantages of cloud computing are “greatly magnified”, I would argue exactly the opposite. If we assume that new investments in hardware and software are significantly lowered during the recession, this would mean that small/startups are more likely to choose cloud computing (as they zero front-up expenditures), whereas bigger companies that already have server infrastructures are more likely to try to further leverage their existing capacity (thus not having to spend additionally to move to the cloud).

Cloud computing models' obvious advantages of usage based pricing and no upfront could be augmented by the transference of *underprovisioning* (loss of customers)/*overprovisioning* (costly excessive capacity) risks [Berk\_ATC] due to elasticity. Although true, in the recession, UC Berkeley's argument for capacity underprovisioning is questionable – e.g. e-commerce companies are less likely to face such high spikes in demand. Arguably, this argument alone does not suffice for paying higher premiums for utility computing, rather than buying and depreciating (writing off taxes) for own server hardware. This should hold for many companies, whose expected computing demand could be predicted (and are not growing like “the next Facebook”). In my view, the *transference of risk* cost **advantages** could be overshadowed by *dependence risks disadvantages*, associated with a) potential future fees/price increases and b) potential costs that would arise from changing to another provider due to structural/functional (API) differences. This is what Richard Stallman meant by arguing that “cloud computing is a trap”, that could cost companies more and more over time as they are lock-in. [Guardian\_Sep\_08] It must be mentioned though, that due to *cost associativity*, [Berk\_ATC] i.e. the ability to run say 1000 hours worth of CPU time in 1 hour without paying additional premium will inevitably evidence usage, especially for companies that have large batch-processing tasks (e.g. movie animation and special effects studios) as they will be able to effectively compete with large companies that have the edge of owning the grid clusters for performing these tasks on their own.

### 2.3.1 Providers Passing Down Lower Hardware Costs to Customers

UC Berkeley's key argument for the emergence of cloud computing is the creation of large scale commodity-server datacenters, built close to electric power plants. These datacenters highly leverage economies of scale of cheaper electricity as cooling and power infrastructure roughly account to

40% of total costs. [Pwr\_Std] A large datacenter (50,000 servers) vs. a medium one (1,000 servers) vs. has an average factor of 3 to 7 times less costs to be able to buy server hardware, provide a monthly GByte of storage, MBit of bandwidth and ultimately also have lower labor costs for the administration of a single server (1,000 servers per administrator vs. 140 in a medium datacenter). [Berk\_ATC] Labor costs sum up only to 4-5% of total costs however. [Pwr\_Std] While certainly true that those cost advantages would allow for lower end user prices, it does not necessarily imply that cloud providers will immediately be willing to pass down those savings over to the simple end user without charging large premiums ("skim-the-cream" pricing). Effectively, for a startup company this means the ability to provision scalable computing capacity at low prices, but for a large enterprise the "one-that-price-fits-all" policy actually means being overcharged than if procured from the own datacenter.

Amazon's announcement of capacity reservations/pre-paid instances is likely to be followed by other cloud providers, as this enables them to make better planning of their own resources, save costs and be able to offer longer-term commitments at lower prices. [DCH\_Apr\_08]

### **2.3.2 Own Server Virtualization vs. Cloud Services Provision**

In order to estimate the financial viability of moving to the cloud, the Total Cost of Ownership (TCO) of the assets involved needs to be considered. Conclusions could be drawn from two competing research analysis. Firstly, RAD at UC Berkeley [Berk\_ATC] argues that due to economies of scale utility cloud computing has all grounds to be cheaper than a data center; yet partly irrelevant figures how prices of CPU/hour and GB transfer have deflated from 2003 to 2008 are presented. Their analysis is biased towards the future potential of cloud architectures, rather than what is currently available (they

are primarily sponsored by Google, Amazon, Microsoft and Sun [adlab.cs.berkeley.edu]). They argue that utility-based cloud computing is universally cheaper for everyone, yet in their side notes they conclude that Amazon's WAN-bandwidth prices are comparable to what a medium-sized company (implying it is cheaper for a large one) might get elsewhere. Data transfer bottlenecks however are a problem in reality as large amounts of data could need to be shifted, if in fact most of the computing is done in the cloud. In a second comparison study, McKinsey & Co. [McKinsey\_09] assessed the economics of Amazon EC2 VM instances vs. the TCO of a typical datacenter (\$45/month, 10% server utilization and common U.S. electricity prices). The study is strictly based on current and not future prospects of the technology and is somewhat biased towards traditional datacenters, as it was presented at the Uptime Institute, which pursues the interests of the data center industry [www.uptimeinstitute.org]. In my opinion, the following conclusions could be drawn:

- **VM instance size** determines the break even point - the larger the required on-demand instance size (real compute equivalents of VM core units, e.g. 1,2GHz Xeon CPU), the costlier it is for large enterprises in comparison to the TCO of a typical data center. Clearly, the TCO of large, on-demand EC2 instances is uneconomical (up to \$180/month in some cases according to McKinsey) compared to a dedicated server at a typical datacenter
- **Small instances** are thus suitable for getting some additional capacity or executing batch tasks and are especially economical, for Linux-based VMs. In case a company has some specific business applications where computing demand is varying, running them in small on-demand EC2 instances is cheaper than running them at a fixed server in a data center
- **Pre-paid instances**, offered by Amazon have in fact lower TCO than typical datacenter offerings (<\$45), however only for Linux systems.

Procuring Windows-based on.-demand instances from Amazon is costlier than from a datacenter, even if allocated via pre-pay agreements. Clearly, OS licensing fees that are charged by Microsoft still do not allow enough flexibility for on-demand Windows server VMs to be economical for the cloud. Arguably, Microsoft will have no interest to lower these OS fees, if it prefers for customer to choose the Azure Platform for their SQL Server, SharePoint, Exchange, etc. needs

- **Significant labor costs**, primarily associated with hardware facilities, DB and IT administrative tasks could be saved if a company moves its whole IT infrastructure to the cloud. However, because the overall TCO and non-labor costs of IaaS cloud providers - e.g. Amazon EC2 - are much higher, they outweigh by far labor costs savings. In this relation, a far fetched, yet related argument is provided by Nick Carr as he describes the potential labor redundancies associated with advancement of autonomic, utility and cloud computing technologies to prove very destructive for the middle class, as many manual jobs will disappear (see Chapter 3)

McKinsey concludes that Amazon EC2 TCO must come down substantially for outsourcing a complete data center to become economical. It is proposed instead that companies try to leverage further their existing infrastructure, which as I argue would seem more likely as it is less risky in the current recession. Virtualization technology - XEN, VMware, Citrix would enable companies to achieve higher server utilization rates without additional excessive hardware expenditures. If we assume the typical server utilization rate to be 10%, according to McKinsey 18% are "quickly achievable" whereas "aggressive virtualization" would yield up to 250% improvement, i.e. up to 35% server utilization rates. Although McKinsey's figures might be wrong, new virtualization technology will in my opinion drive companies to build "internal clouds" and they may further standardize and optimize their

hardware to be easily virtualizable or restrain from buying expensive new “non-common” hardware (e.g. the demand for powerful shared-memory servers is declining [IBM\_Scale]). The emergence of internal clouds in addition to public cloud services is also expected by Sun (see Chapter 2.2), however internal “private” clouds are not true “cloud computing” as defined by UC Berkeley as they still require a company to possess/acquire the hardware necessary, i.e. no zero capital expenditures are feasible. Public clouds and cloud services on the other hand, as advocated by RAD at UC Berkeley arguably have the potential to achieve even better utilization rates (e.g. more than 35%). The use of specially conceived platforms (PaaS) that have higher abstraction levels (and are not IaaS - e.g. Google AppEngine) could however result in lower end-user costs if enough competition and enough convergence between the platforms becomes reality. The cost reduction would come from the higher server utilization rates associated with the fact that a) scalability functions in such PaaS platforms are built-in - they are by nature scalable (albeit need software written specially for them) and 2) demand in a public cloud varies widely which enables greater flexibility if eventually enough tenants share the costs.

### 2.3.3 Fixed/Threshold vs. Supply/Demand Price Determination

Current cloud computing pricing is either *usage-based* (“pay-as-you-go” models, compelling to small companies) or *pre-paid* (useful for large companies when known in advance that greater capacities need to be allocated in the cloud). However, if we assume that public clouds really do take off - utility computing becomes mainstream and small/mid-sized companies procure their computing processing needs mainly from the cloud, standardization of APIs, convergence and interchangeability of cloud providers is realized - then ultimately new pricing models will emerge. In this case, as any readily available commodity, utility computing would be

tradeable and priced according to perfect competition market principles - no cloud provider will alone have the power to change its prices significantly. Based on users' QoS requirements (and SLA-agreements) different "quality" classes of the computing commodity may emerge which will be priced similarly and which customers would be able to choose from.

Research at the University of Melbourne [UNI\_MELB\_08] outlines possible aspects of a market-based cloud architecture. It concludes that cloud providers will have to differentiate service requests based on their importance and thus system-centric server resource management on their part will not be feasible. Instead, in a market-oriented cloud architecture computing resources will have to be managed and supplied at market equilibrium. This will provide incentives for both cloud computing producers and consumers. Essentially, QoS/SLA-class would be the defining property and resources will be allocated and priced according to that.

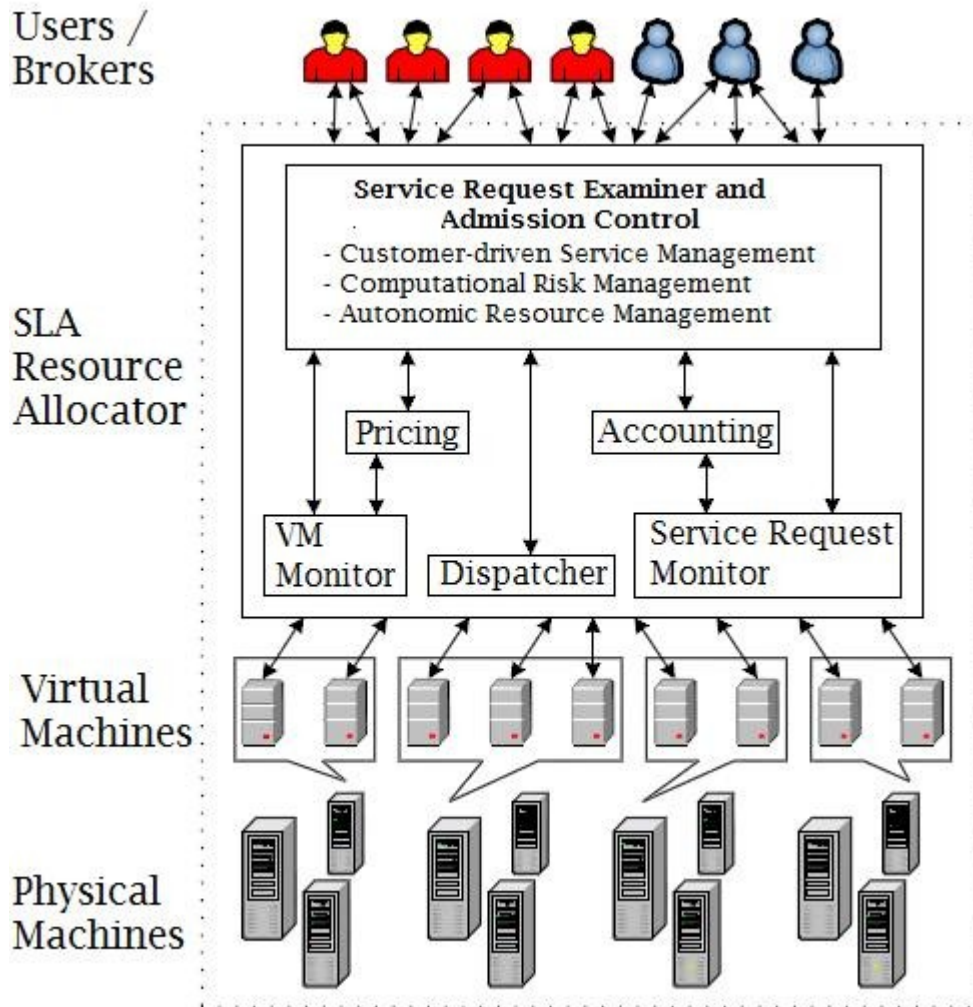


Figure 1: Market-driven cloud computing architecture [UNI\_MELB\_08]

The cloud architecture in Fig. 1 is proposed to consist of four main entities that have to be in place for the market-based principle to be applied. The **users** (or other users who act as brokers) initiate requests for data processing from the cloud. The **SLA resource allocator** is the interface between the users and the cloud provider. It is comprised of multitude of autonomic management features that enable the cloud to supply the requested commodity computing power. Firstly, this current resource usage monitoring - workload allocation (VM Monitor), deny/allow mechanism for incoming requests, a dispatcher that directs accepted requests to allocated resources. Secondly, flexible pricing algorithms are needed that charge

customers based on the current availability situation supply/demand of resources as well as the level of QoS the customer needs (guaranteed capacities, redundancies, backups, network speeds etc.). Needed is also a cost accounting module that in addition contains historical usage analysis that is used by the SLA resource allocator to make better resource allocation decisions. Ultimately, isolated **VMs** of different specifications (CPU, storage, redundancy, etc.) are started and run on the **physical machines** in the datacenter.

The complexity of the above described architecture implies highly sophisticated software and the technical realization of some of its autonomic features (risk management tactics, cloud provider price optimization etc.) is highly questionable. In my view, the University of Melbourne approach is a viable solution (from the customer's perspective) only when standardized software that could be purchased by any datacenter or even better an open source solution that could be modified and run by anyone is developed. If only a handful of cloud providers are able to acquire or develop such highly-autonomic, self-contained piece of cloud architecture software, then the prices they would offer to their clients will contain high premiums.

Furthermore, even when cloud computing matures to an extent single cloud providers are able to offer the above described services in an interchangeable manner, it is still not clear how a global market/exchange of interconnected cloud computing providers will function. Although research projects already have come up with the market structures/algorithms for resource trading (VM-based resource slices such as PlanetLab [UNI\_MELB\_08]) I would argue that such exchanges are less likely to become reality in the near future for several reasons besides those mentioned by privacy, security and regulatory issues, mentioned by the University of Melbourne:

- in order to be created, a global computing utility clearing system/cloud provider directory would need a **business case**. Plenty of efforts will be needed to regulate and standardize the participant auctioneers that will act as the cloud providers. So long a clear business case for the exchange does not exist, funding its establishment will be difficult, even initiated by market leaders
- in order for companies to be willing to procure computing from largely unknown providers in the cloud **based on the resulting lower price alone** and despite all concerns, this price will need to go enormously down (which also puts pressure on any business case the exchange platform may have). The argument that not only would the price be lower, but also a lot more computing resources would be available in a global market (i.e. the amount of simultaneous computing a company can request) is not relevant for a single company as of course no company has the need for that much resources. This could be a factor only if a whole country would export computing power as it exports electricity, but then arguably the regulatory/security issues would prevail over the lower price advantage, which would again invalidate the need for a global cloud computing marketplace

#### 2.3.4 Unexpected Additional Costs In Testing and Debugging

Developing or changing software for cloud architectures will be subject to increased costs for a variety of reasons. Notwithstanding any of the costs incurred by having to rewrite existing (e.g. non-scalable) software, the fact that cloud architectures are not on-premises poses additional difficulties across the whole software lifecycle and especially during the phases of testing and bug fixing. Consequently, maintaining a complete on-premises test

environment is complex, cumbersome and could even be not fully possible as well (e.g. Microsoft's Azure OS will not be deployable outside Microsoft datacenters). Virtualization solutions for SOA testing that handle those problems are e.g. provided by iTKO, Inc. The iTKO Lisa Suite is proprietary software for unit-testing across distributed, heterogeneous, multi-tier applications, along with the ability to virtualize dependent application behavior to eliminate the “wires hanging out” (system-to-system dependencies inherent to SOA) of the cloud. [[www.itko.com/products](http://www.itko.com/products)] Essentially, this software virtualizes multiple applications on multiple operating systems on a single server and performs continuous predefined unit, system and performance checks (Java/J2EE/JMS/.NET services/objects are supported). Prices of the Lisa Suite are not quoted on iTKO's website and in any case acquiring and setting up this testing solution may be costly for smaller companies.

Furthermore, substantial additional usage fees in the “production” cloud environment could break testing budgets. [CC\_Testing] Having to pay for computing resources for such non-productive tasks should be taken into account even if an on-premises testing environment is employed. CPU time, storage and network bandwidth could be needed if meaningful (e.g. real data) tests or simulations are done in the cloud. This is a stark contrast to running and tweaking unlimited test jobs in the own datacenter. Moreover, when other cloud services (e.g. third party's web services) are relied upon, testing and monitoring dependent modules and APIs should be done before and after deployment to the cloud. [CC\_Testing] The aforementioned concerns, albeit technical of nature could ultimately present cost pressure challenges and time-to-market bottlenecks for companies who wish to move their existing IT operations in the cloud.

## 2.4 Privacy and Security Issues

Shared infrastructure scares many enterprise customers. [Linth\_D] Placing enterprise data in a public cloud is a serious concern and companies wary about their sensitive data logically question the ability of public cloud computing providers to provide the same level of security as their own datacenters. Depending on the type of cloud computing used (IaaS or PaaS) and the level of abstraction (OS-level vs. platform vs. application level) different security issues arise in public clouds. The cloud provider is responsible for the physical security of the machines, for ensuring that VM instances are running isolated from one another (i.e. crashes and software exploits of one system do not affect the others) as well as for setting up firewalls to protect the VM machines from the network. However, higher level cloud services such as Google AppEngine and platforms like Azure are also responsible for their application-level security and clients have less control controlling it. In addition, downtimes, outright data losses in storage services and risks of cloud provider malfeasance are further threats to be weighted when a company considers public cloud services usage.

### 2.4.1 Data Security – Confidentiality and Availability

VMs have shown vulnerabilities to certain kinds of memory attacks (UC Berkeley points to research at Princeton that shows how Java and .NET VMs could be hijacked by inducing memory errors [VM\_MIT]). Even though physical access to the PC running the VM is a prerequisite, I argue that private clouds are generally more secure, as availability of the physical machines and full administrative rights are at the company's disposal. Arguably, it is much more likely that in case a bug is found (or proactively with malicious attacks) problems arise that allow VM users to access other users' VM instances or storage data. Naturally, such problems exist in large datacenters too, yet the

implications of ultra large scale failures given hundreds of thousands of potential cloud users sharing the same infrastructure could be devastating. Debugging such distributed such developed, widely distributed systems may later be very difficult, as some errors could not be reproduced in smaller, test configurations. [Berk\_ATC] Still, companies should spend additionally to ensure that their data and applications are as secure as possible in the cloud. Encrypted all data sent to the cloud may be an option to ensure security, yet this may have implications on costs for developing/configuring applications appropriately.

#### **2.4.2 Cloud Provider Malfeasance**

Cloud provider malfeasance refers to the operational counterparty risk associated with misuse, data theft or malicious altering of confidential customer data by the cloud service vendor. The cloud provider is the ultimate administrative entity and is able to effortlessly spy (including to log, analyze etc.) on VMs and storage data of the underlying instances in the cloud. This is particularly relevant for large enterprises, although one can argue that such large companies would move to the cloud only parts of their compute/storage tasks and will have separate confidentiality agreements with the cloud provider. Yet the sheer amount of business (and even technologically) sensitive data that cloud providers are able to trace and potentially take advantage of must be considered, given that large public cloud providers envision massive future usage and entire IT-departments allocated in the cloud. Moreover, involuntary or accidental data exposure could also occur (e.g. Amazon's S3 outage in 2006 when users could see other user's data [McKinsey\_09]).

### 2.4.3 Uptime Guarantees

Six Amazon (EC2 compute and S3 storage) downtimes lasting 1.5 – 8 hours as well as Google AppEngine and Gmail outages are often quoted. Although uptimes from 99,9 – 99,95% are less than what *most companies* require, it is not clear whether *most companies* really need to set their SLA higher than 99,99% (assumed monthly percentage availability required by a large enterprise from a typical/own datacenter). [McKinsey\_09][Berk\_ATC] It must be noted however, that so long cloud providers are not interchangeable, no matter how diversified Amazon's or any other company's cloud system is (e.g. with redundant servers located in different countries), it is potentially a 'single source of failure'. However, the fact that cloud architectures scale on-demand is an **advantage** [Berk\_ATC] when guarding against a Distributed Denial-of-Service (DdoS) attack, where it is arguably much more expensive for the attacker to keep up with the attack, as most of the incoming requests could be handled successfully.

To illustrate the threats with an example let's consider GoGrid.com, a competitor of Amazon Web Services that claims offering the first 100% uptime SLA for cloud services (IaaS). Yet, a detailed study reveals that this refers only to compute (CPU and RAM) uptimes and not to persistent storage availability. Moreover, the “10,000%” guaranteed implies reimbursing a client with credit for 100-fold the outage delay (e.g. 7 minutes outage = 7 hours free credit), yet “no credit will exceed one hundred percent (100%) of Customer's fees for the Service feature in question for the then-current billing month” (effectively limiting GoGrid's liability to the maximum of losing that customer's revenue for a *month*, even though the customer may have incurred huge losses due to the outage).

*"...Data retrieval issues caused by problems connecting to the Service, including without limitation problems on the Internet, do not constitute Failures and so are not covered by this SLA. Under no circumstances will GoGrid be responsible for the restoration of any data to cloud storage or for the loss of any data..."*

"... GoGrid will make **reasonable efforts** to insure that server storage is "persistent..."

Source: GoGrid's "100% Uptime 10,000% Guaranteed" SLA [[www.gogrid.com/legal/sla.php](http://www.gogrid.com/legal/sla.php)]

Clearly, this level of service/risk is acceptable for small and even medium companies, however it could not be reasonably expected that large organizations where business continuity is paramount would be ready to move strategic compute/storage tasks to the cloud under these circumstances. On the other hand, Google is allegedly able to lose 30% of its servers and still incur zero data loss. [Dr. Nusser (IBM), WU-NM VK3] However to make such a high level of redundancy possible, firstly non-relational databases (e.g. BigTable) should be used and arguably customers should be able to have a failover cloud provider interconnected with their current provider, something currently not feasible.

## 3 Strategic Implications of Cloud Technology

This chapter I focus on discussing higher level viewpoints on cloud technology. I present strategic socio-economic and regulatory/legal issues that although not immediately visible on a company micro level are likely to impact the development and deployment of cloud architectures.

### 3.1 The Socio-Economic Perspective

On the socio-economic perspective cloud computing is viewed by enthusiastic futurists as Nicholas G. Carr as a “paradigm shift” and a “revolution”. In his book - 'The Big Switch' he argues that “the diminishing strategic importance of IT departments” and “computing turning into a utility” is likely to have huge implications on the world's economy and society. He holds that due to higher levels of automation salaries of high paid IT/administrative jobs are deflating and this retranslates to endangering of the middle class (created by industrialization due to electricity available as utility). Although an engaging reading I regard his comments as contradictory thus essentially false. Firstly, I would take IBM's view that automation has always been the source of progress and that it inevitably produces complexity as a byproduct. [IBM\_AC\_Vision] Yet, the current state of complexity in the IT infrastructure threatens to undermine the benefits of IT due to arguably being close to our human limits to handle. Thus, I would take a libertarian stance ala Friedman and suggest that any worker being laid-off or made redundant due to utility computing making manual tasks automated, did not lose his job due to automation holding him back from contributing to the GDP, **but** rather made him/her available for the economy to perform a *more productive task* instead. Moreover, cloud computing is widely recognized not to be a revolutionary but evolutionary technology (e.g. Oracle's CEO claims that cloud computing features will not prompt them to change any wording in their ads).

### 3.2 Regulatory and Jurisdiction Issues

Still, according to The Economist [23<sup>rd</sup> Oct 2008] cloud computing is “the ultimate form of globalization” and references to white papers by SAP and Google handed to EU and US regulators that are quoted to highlight the dramatic economic effects of the “future Internet” (the cloud). Mainly, issues of jurisdiction and law enforcement are discussed as a company procuring compute power from one country, cloud storage from another and SaaS from

anywhere faces serious problems in case the company has to comply with customer data boundary regulations and is being audited or any of its cloud services providers goes out of business, is subpoenaed or even maliciously misuses that company's data in another jurisdiction. US's PATRIOT act allows extraction of cloud users' data without their prior consent or knowledge and the danger of a cloud provider being subpoenaed endangers a company that its data be "vacuumed up" in an investigation, even though the company is not *per se* being subpoenaed. [Linth\_D] Summarized, among the legal problems related to clouds are issues including: access, reliability, security, data confidentiality and privacy, liability, upholding of intellectual property rights, ownership of data, data portability/fungibility and auditability. [Jaeger\_P]

I agree with The Economist's assessment that data will inevitably get more globalized and move to jurisdictions where compute power is cheaper or legal oversight laxer. In addition, this may work both ways as datacenters for critical services (e.g. clearing systems such as SWIFT, Clearstream, etc.) will be built to stay in their designated jurisdictions but much of the business's requirements for computing may move to datacenters located where electricity is cheap (geothermal power), broadband connectivity is abundant and cooling is not problematic e.g. in Iceland or Greenland. [ESN\_CC] Yet it remains to be seen how far will governments react with laws favoring protectionism or globalization of compute resource usage from abroad, in case SaaS and cloud services gain the critical mass needed to establish thorough regulation. Governments are realizing the underpricing of risk witnessed by Lehman Brothers'/AIG's collapses and will in my opinion regulate heavily any single entity that gains that critical mass of control in case of massive cloud computing adoption by businesses of various sizes.

## 4 Current Cloud Architecture Technologies and Platforms Comparison

This chapter presents a functional overview of popular cloud computing architectures. Background is given on the technical aspects of each architecture and parallels are drawn with the previous three chapters, whilst also discussing business concerns.

As an introduction to the comparison overview, the platforms need to firstly be categorized based on their level of abstraction - whether the cloud architecture is IaaS, PaaS or a combination of both:

- cloud instances on **hardware level (IaaS)** are provided by Amazon EC2, GoGrid and Mosso Servers. Thereby only a handful of APIs are present on top for setting up the instances. This allows for running on-premises applications and common relational databases, which due to aforementioned scalability issues have only limited potential for elasticity. Alternatively, Amazon provides SimpleDB for building scalable storage programs, yet due to its proprietary nature it is not widely adopted and requires significant application reengineering
- **hybrid IaaS/PaaS** public clouds such as Microsoft Azure (and to an extent Sun Grid) allow for more API calls to native scalability functions. However, what goes on beneath the surface is largely abstracted (and mostly unknown, in Microsoft's case). In Microsoft Azure, programmers are able to plan in for elasticity features in their applications (by deliberately using built-in functions where applicable) and allow for some level of automatic cloud failover/scalability optimization when running their application at Microsoft's datacenter/cloud, although 'normal' .NET applications are still possible to be executed, albeit

without utilizing fully the platform's abilities (some applications may actually even run slower)

- **cloud service PaaS** such as Google AppEngine or SalesForce/Force.com are proprietary, domain-specific PaaS/SaaS (SaaS because they are delivered as a web application) that are arbitrarily programmable only to a limited extent and for specific business tasks. Google (much like Amazon) made parts of their own algorithms for highly elastic computing and storage publicly available, whereas the business development platform Force.com allows programmers using their Apex and even non-programmers using the visual GUI Visualforce to simply program customizable extensions to their main SaaS application SalesForce.com

Generally, when comparing platforms I suggest the reader to consider storage and compute capabilities differently. My research shows that most marketing materials, and cloud product descriptions do not actively force that differentiation between the compute and the storage cloud. Even some white papers vaguely mention the topic, yet this differentiation is highly important for business considerations. Should a company's marketing or management get excited about 'moving IT to the cloud' they need to separate their storage from their computational needs in order to decide for a proper cloud architecture. Table 1 below presents some technical and functional aspects of five different cloud computing paradigms:

	Amazon Web Services	Mosso Cloud Servers	Windows Azure Services	Google AppEngine	Salesforce.com, Force.com
(X)-a-a-S Type	IaaS	IaaS	IaaS/PaaS	PaaS	PaaS/SaaS
Application runs in a visible VM	X	X	X	-	-
Provides administrative access to VM	X	X	-	-	-
Can create own OS installations	X	-	-	-	-
Service Type	Compute, Storage (S3)	Compute, Storage	Compute, Storage, Framework	Compute, Web Application	Web Application
Virtualization	OS Level running XEN	OS Level running XEN	OS Level	Application Container	Application Container
Programming Framework	Linux-based Amazon Machine Image (AMI)	N/A, TBA	MS .NET Languages	Python or Java	Apex (script) or Visualforce (GUI)
Runs normal on-premises applications	X	X	-	-	-
Suitable for creating moderately scalable applications	X	X	X	-	-
Suitable for creating very scalable applications	X	-	X	X	X
Suitable for running parallel processing applications	X	-	X	-	-

Table 1: Functionality comparison of different cloud architectures

Source: [D\_Ch2],[UNI\_MELB\_08] and own research

## 4.1 Amazon Web Services. Elastic Compute Cloud, Simple Storage Service and SimpleDB

Amazon Web Services is a brand of remote computing web services offered by Amazon Inc. Throughout this thesis Elastic Compute Cloud (EC2) was used to exemplify a hardware-close OS-level XEN virtualization cloud architecture that is priced by hourly usage of EC2 instance units (VMs of different hardware capabilities - a multiple of ~1,2 GHz Opteron/Xeon processors) as well data transfer fees. The largest instance is equivalent to 8 EC2-units, totals 15GB of RAM and 1690GB of hard disk space and supports 64-bit platforms. Linux, Sun OpenSolaris Microsoft Windows Server 2003 are supported, however instances running Windows cost more per hour due to licensing fees. Eucalyptus, an open source project now supports standardized integration of Amazon's APIs into Linux distributions for the purposes of building clouds.

Amazon's Simple Storage Service (S3) is an online storage service that allows users to store unlimited amounts of data leveraging Amazon's own e-commerce infrastructure with pricing between \$0.120 and \$0.150 per GB monthly (European prices are ~20% higher). [<http://aws.amazon.com/s3>] Additionally, data transfer charges (almost identical per GB/month storage pricing) and two types of HTTP requests (POST, PUT, etc. and GET separately) are billed separately. S3 leverages REST and SOAP protocols but also provides the BitTorrent P2P protocol to lower costs for high scale distribution. It has a relatively low uptime guarantee of 99.9% anchored in the S3 SLA.

[<http://aws.amazon.com/s3-sla/>] Amazon SimpleDB is a distributed database for storing augmented key/blob structured data which scales automatically. A blob (Binary Large Object) is a schemaless unstructured data with varying contents. SimpleDB query execution time is limited to 5sec and operates through with a simplified SQL-like API. Still marked as "beta", SimpleDB is priced based on machine hours, data transfer and storage utilization and can

be integrated with S3 and EC2. In conjunction with an EC2 instance S3 and SimpleDB are useful for creating very scalable small applications benefiting from elastic BLOB-storage. [D\_Ch2]

## 4.2 Microsoft Azure Services

As already mentioned in Chapter 2.2 Microsoft will offer the Azure Services runtime platform for executing .NET applications in the cloud and will thereby sell hosting (compute) accounts and storage accounts itself. Authorization will be done via Windows Live ID. [Chappell\_Az] A *fabric layer* will be present to abstract the VMs from application instances and it will assign automatically more instances/computing resources for elasticity purposes, as well as provide basic application failover/restart capabilities. The fabric layer will not allow developers to control the OS or the VMs directly thus I would categorize it in the PaaS class. VMs will run either in a *web role* or a *worker role*. A web role denotes starting an ASP.NET/IIS web application that handles network (HTTP) requests, whereas a worker role does not use the IIS (Microsoft's web server) and represents a batch background job started from a queue that can only have outgoing connections (to write results after a job is executed) and will be possible to be realized in any .NET language (C#, VB.NET, J# but also Ruby and Java using SDKs). Azure will store data in *tables* (of relational nature) and *blobs* which will be held in *containers* assigned to each customer account. SQL Data Services (a restricted view of SQL Server [Berk\_ATC]) will manage the containers stored at different Microsoft datacenters and provide access to the storage data (see [Chappell\_Az]).

## 4.3 Google App Engine. BigTable and MapReduce

Google App Engine, also discussed in previous chapters, is a service offered by Google Inc to enable user-made applications to run on Google's own infrastructure via a large set of proprietary APIs. AppEngine is purely a PaaS that supports running restricted versions of Java and Python code with a 30 sec timeout and read-only file system capabilities. Application execution can only be invoked via HTTP requests. Storage is done in Datastore - a schemaless blob that is strongly consistent and that can be queried with very simple one-column `WHERE` clauses. AppEngine's built-in services include using the URL-fetching and mail-broadcasting services Google claims to use themselves as well as simple memory caching, scheduled tasks (cron jobs) and image manipulation functions for performing background batch jobs. Google accounts are required for setting up the applications and could also be used instead of programming user modules for user authentication. AppEngine supports limited free usage and over a certain quota (500MB storage and 5mio requests per month) is priced per hourly CPU time, GB/month of storage/transfer and mails sent.

To understand how Google is able to deliver those services along with its whole set of projects I will introduce BigTable and MapReduce briefly - the technologies behind the vast amount of petabytes of data stored at Google's server and the ability to perform ultra fast searches over it. BigTable is a distributed, column-oriented, multi-dimensional sorted map that is able to run on thousands of physical machines and allow for extremely high consistency. Data is replicated on multiple machines so a hard disk failing of a given machine has no effect on bringing the whole system down, which potentially allows full consistency even if 30% of Google's servers were to fail at once (see [BigT] for details). MapReduce on the other hand is the programming model and implementation of processing BigTable's huge datasets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that

merges all intermediate values associated with the same intermediate key. Google's index of the WWW is regenerated using MapReduce and it allows for many parallel processing tasks in distributed applications. The map/reduce paradigm has since been implemented in many other projects. Please refer to [MapR] for a detailed description of it.

#### 4.4 Hadoop and Yahoo

Hadoop is the free and open source implementation of Google's primary infrastructure technologies - BigTable, MapReduce and Google's distributed file system (GFS). It is a top-level Apache Software Foundation project implemented in Java that consists of a map/reduce engine, a distributed file system (HDFS), a job/task tracker and the column-oriented HBase schemaless database similar to BigTable. [<http://wiki.apache.org/hadoop>] Yahoo is the main contributor to the project and have recruited Doug Cutting (the original inventor) to lead Yahoo's cloud computing Hadoop efforts. [YahooH] In June 2009 Yahoo released their own distribution of Hadoop and they claim to run the largest Hadoop clusters worldwide. Hadoop is not only responsible for generating Yahoo's search index, but also employed by nearly any cloud computing stakeholder - Facebook, IBM and Rackspace (owners of Mosso Clouds) and many others. However, a closer look at Hadoop's site reveals that most of those companies use Hadoop for a limited number of background batch processing tasks - mostly analytics - log/click/ads analyses. The NY Times used Hadoop running on Amazon EC2 instances for a very large batch processing tasks (TIFF → PDF conversion) that cost under \$300 and converted all scanned articles from 1851-1922 to be made public domain. [NYT\_HDP] This clearly shows that after the platform gets developed to a mature level that enables it to be installed at any cloud provider - much like the LAMP (Linux-Apache-MySQL-PHP/Python) stack, the proprietary Microsoft/Google

platform offerings could be challenged with an open system that provides similar advantages without locking the customers to a specific provider.

#### **4.5 Apple and iPhone in the Cloud**

Apple is not a usual follower of industry fashion, but in the context of cloud computing it employed a clouds architecture for storing user synchronization data (e-mail, contacts, calendars). MobileMe is provided as SaaS against a \$99/year subscription charge. [Maya\_WrCl] However, the service experienced outages between 16-28 July 2008 that prompted users' emails to be lost and data not to be retrieved. [Newsweek08] This incident is to stress that real-time cloud architectures are apparently struggling with concurrency and consistency problems and are still highly exposed to partial outage at least and information loss and leakage at worst, and to emphasize that the results in reputation and revenue loss due to disgruntled customers are substantial. This applies in fact to all web-based services and not only the paid ones - e.g. also Gmail, Google Docs and other SaaS whose data is stored and updated in cloud architectures, spread on many servers running not yet fully researched distributed programming data manipulation techniques, are prone to problems that are not at all anticipated by the customers.

#### **4.6 Business Software Clouds – Salesforce.com**

SalesForce.com is a CRM software vendor founded in 1999 by Oracle executives. [Wikipedia] It delivers CRM SaaS using monthly subscription plans per user (from \$9 for the Group to \$250 for the Unlimited version). SalesForce.com, much like Amazon and Google utilizes a serious internal grid computing architecture and because new customers are allocated within this infrastructure it enables them to market their CRM as *software in the cloud*.

Co-branded with Force.com, customers are able to go beyond the delivered CRM and write their own (web) applications, that are however able to run only against Force.com's database and are written in Apex Code – Salesforce's own programming language. Apex Code's syntax is Java-like and allows for integration with web services APIs written in other languages. An isolated sandbox environment is available that serves as a test/development platform. The Visualforce GUI application/workflow designer is provided in addition to enable code free creation of web UI using Force.com's (and Apex Code) components and mashups integration. The advantage of Force.com's applications is the ability to swiftly scale out, but the limitation of not being able to run any other applications as well as the need to Salesforce's proprietary database makes this not a real show case of cloud computing in my view. Force.com's PaaS is licensed for \$50-75 per user per month with limitations on database objects and storage space. Salesforce.com has however grown significantly and was added in Sep 2008 to the S&P500 index (after Freddie Mac and Fannie Mae).

## 5 Outlook and Conclusions

Cloud computing is undoubtedly still work in progress – both from a technical and business perspective. Although projects like Hadoop attempt to bring about a platform that is provider-independent, the lack of open standards and the abundance of proprietary APIs that each provider actively tries to bestow upon its users is still a major setback to wider scale adoption in my opinion. Clearly put, my conclusion is that non-IT industry businesses' IT departments are not yet justified to be moved to cloud architectures, and if so only for very specific business tasks and with great caution. Yet, execution of batch jobs/parallel processing tasks and smaller online businesses running only pure web applications seem to be a nice fit, regardless of being locked in

with a specific cloud provider. Listed below are my concluding thoughts, listed arbitrarily that also relate to certain vendors and technologies:

- Google AppEngine is likely to serve as a showcase for cloud computing. Companies may realize the advantages of scaling their web applications as they look up to Google as a technology leader and try to see how they can benefit from that using other ways. AppEngine in its current form is not likely to be something other than a niche in that sense
- I expect non-platform, pure IaaS cloud providers to somehow offer standardized APIs in the future. Hadoop or not, they will try to figure out how to consolidate and leverage common technologies for the purposes of providing an alternative for lock-in wary Amazon avoiders
- on the hardware side, cloud based memory architectures are likely to grow in popularity and be offered by providers to instances as an additional premium perk. Twitter.com reportedly stores much of its data in RAM instead on hard disks and is thus able to restart in 2 minutes (see [MemArc])
- the global financial crisis is likely to affect decisions for cloud technology and contrary to popular belief, not in a good way, at least in short-to-medium term. Companies will not massively invest in uncertain technologies no matter how promising they are, even though the . This is no ordinary crisis and as with cloud computing risk aversion could be amplified more than potential savings, and rightly so
- many of Microsoft's business products, that would normally be shipped as local on-premises applications will be tried to be delivered in the Azure cloud as SaaS. Whether Microsoft will be successful at that remains a question how fast companies adopt cloud solutions at all, but as the Prague Cloud Computing Expo in May concluded (reported on [www.sys-com.com](http://www.sys-com.com)), Microsoft is not likely to be a winner at this time
- a long-term if public clouds and P2P clouds become reality companies selling their own free resources from their private to the public could be

feasible, but encryption, security and availability issues are now the technical reality. In addition, reluctance to promote P2P by large vendors is likely to prolong the process even further

- bigger companies, but also cloud computing providers should focus on leveraging more virtualization technology - VAAs and application virtualization in particular. Citrix (for Microsoft platforms) and VMWare offer products that if successfully implemented in extend the lifespan of the company's IT investments while still allowing for the much hyped scalability and high server utilization rates
- data intensive batch processing tasks (see Chapter 4.4 with NY Times' PDF article conversion) can and should be outsourced to the cloud. Denying the obvious advantages of cloud computing even in its today's form is also not a good decision, especially for non business critical tasks
- hardware-makers may be torn between supplying cloud providers or becoming providers themselves. Dell seems to want to be a cloud supplier whereas Sun Microsystems to be a provider. HP and IBM, are likely to try to do both (see [Econ2])

Ultimately, legal and regulatory issues are likely to be a decisive factor. If and when major cloud computing adoption takes place, governments will need to step in and regulate in one form or another either the cloud providers or the cloud users or both and rightly so.

## 6 Bibliography

Below is a list of the literature and online sources referenced herein:

**[IBM\_AC\_Vision]** Kephart J.; Chess, David M.; "The Vision of Autonomic Computing", published by the IEEE computer Society, Jan 2003

**[Berk\_ATC]** Armbrust M., Fox A. et al; Above the Clouds: A Berkeley View of Cloud Computing; <http://radlab.cs.berkeley.edu/>; Feb 10, 2009

**[IBM\_Manifesto]** IBM; "Autonomic Computing: IBM's Perspective on the State of Information Technology"; <http://www-1.ibm.com/industries/government/doc/content/resource/thought/278606109.html>

**[ESN\_CC]** Erenben C.; Cloud Computing: The Economic Imperative; IBM eSchool News; [www.ibm.com/education](http://www.ibm.com/education), Mar 2009

**[VAA\_AppZero]** AppZero Inc.; Jumping on the Cloud-Wagon? Pack your server applications for the trip; <http://www.appzero.com/files/downloads/whitepaper>; Mar 2009

**[AWS\_Varia]** Varia, J.; Cloud Architectures; <http://jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf>; July 2008

**[O\_CEO]** Farber D.; "Oracle's Ellison Nails Cloud Computing," in CNET "Outside the Lines" [http://news.cnet.com/8301-13953\\_3-10052188-80.html](http://news.cnet.com/8301-13953_3-10052188-80.html), retrieved 14<sup>th</sup> May 2009

**[UNI\_MELB\_08]** Buyya R., Yeo C.S., Venugopal S.; Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities, Keynote Paper, Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Sept. 25-27, 2008, Dalian, China

**[O\_Web]** Oracle Technology Network, Cloud Computing Center, <http://www.oracle.com/technology/tech/cloud/index.html>, retrieved 14<sup>th</sup> May 2009

**[Grid\_SLA]** Gridipedia - The European Grid Marketplace, Service Level Agreements; <http://www.gridipedia.eu/sla-article.html>, retrieved 15<sup>th</sup> May 2009

**[IBM\_CC]** Boss G., Malladi P., Quan D., Legregni L., Hall H.; IBM on Cloud Computing; High Performance On-Demand Solutions, IBM; 8<sup>th</sup> Oct 2007

- [CCJ\_21\_Exp]** Twenty-One Experts Define Cloud Computing; Cloud Computing Journal, July 2008; <http://cloudcomputing.systems.com/node/612375/print>
- [Pop\_Gold\_74]** Popek G.J., Goldberg R.P. (1974); Formal Requirements for Virtualizable Third Generation Architectures. Comm ACM 17(7): 412-421
- [Turban\_08]** Turban E.; Electronic Commerce A Managerial Perspective; Prentice-Hall 2008; [http://wps.prenhall.com/wps/media/objects/5073/5195381/pdf/Online\\_Chapter\\_19.pdf](http://wps.prenhall.com/wps/media/objects/5073/5195381/pdf/Online_Chapter_19.pdf)
- [IBM\_Virt\_Edu]** IBM; Virtualization in education"; <http://www-03.ibm.com/industries/education/doc/content/resource/thought/3371378110.html>
- [Bader\_D]** Bader D., Pennington R.;"Cluster Computing: Applications"; <http://www.cc.gatech.edu/~bader/papers/ijhpca.pdf>
- [F5\_Virt\_Gd]** Strategic Guides to Virtualization; F5 Networks Inc.; <http://info.f5.com/virtualizationguides/>; 2009
- [J\_Hurwitz]** Hurwitz J; When Does the Data Center Become the Cloud?; Mar 28, 2009; <http://jshurwitz.wordpress.com/2008/03/28/when-does-the-data-center-become-the-cloud/>
- [Redhat\_JB\_Dell]** JBOSS and DELL just work: The Modular SOA Environment; Red Hat Middleware LLC 2009; [http://www.computerworld.com/pdfs/Redhat\\_JB\\_Dell\\_justwork\\_WP.pdf](http://www.computerworld.com/pdfs/Redhat_JB_Dell_justwork_WP.pdf)
- [SOA\_Dummies\_2ed]** Hurwitz J. et al; Service-oriented Architecture for Dummies, 2ed; ISBN 978-0470376843; Jan 2009
- [J\_Hurwitz\_SOA]** Hurwitz J.;Yes, Virginia, There Is SOA!; <http://jshurwitz.wordpress.com/2009/02/09/yes-virginia-there-is-a-soa/>; Feb 9<sup>th</sup> 2009
- [IBM\_SOA]** Balzer Y; Improve your SOA project plans; IBM; <http://www.ibm.com/developerworks/webservices/library/ws-improvesoa/>; 2004
- [Intel\_SOI]** Intel Corporation; Service orchestration of Intel-based platforms under a service-oriented infrastructure; Intel Technology Journal, Vol. 10, Issue 04; Nov 9<sup>th</sup>, 2006

**[SIIA\_SaaS]** The Software & Information Industry Association; Software as a Service: Strategic Backgrounder; <http://www.siiia.net/estore/ssb-01.pdf>; Feb 2001

**[Grd\_17\_04]** Schofield J; Google angles for business users with 'platform as a service'; The Guardian; <http://www.guardian.co.uk/technology/2008/apr/17/google.software>; Apr 17<sup>th</sup> 2008

**[Douglas\_Sun]** Geelan J.; A World of Many Clouds; <http://cloudcomputing.sys-con.com/node/902342>; Cloud Computing Journal; Apr 1, 2009

**[MS\_Azure\_FAQ]** Microsoft Azure Services FAQ; <http://www.microsoft.com/azure/faq.aspx>; retrieved May 10<sup>th</sup> 2009

**[Chappell\_Az]** Chappell D; Introducing Windows Azure; David Chappell & Associates; [http://www.davidchappell.com/writing/white\\_papers/Introducing\\_Windows\\_Azure\\_v1-Chappell.pdf](http://www.davidchappell.com/writing/white_papers/Introducing_Windows_Azure_v1-Chappell.pdf); Mar 2008

**[InfoWeek\_AZ]** InformationWeek; Microsoft Nixes Private Azure Clouds; <http://www.informationweek.com/news/windows/operatingsystems/showArticle.jhtml?articleID=216300168>; retrieved May 15<sup>th</sup> 2009

**[Davis\_Java]** Davis M.; Scaling up vs. Scaling Out; [http://weblogs.java.net/blog/malcolmdavis/archive/2006/07/scale\\_up\\_vs\\_sca.html](http://weblogs.java.net/blog/malcolmdavis/archive/2006/07/scale_up_vs_sca.html); July 2006

**[IBM\_Scale]** Maged M.; Moreira, J.; Shiloach, D.; Wisniewski R.; Scale-up x Scale-out: A Case Study using Nutch/Lucene; IBM Thomas J. Watson Research Center; IEEE 1-4244-0910-1/07; 2007

**[Linth\_D]** Geelan J.; Cloud Storage: How Can Enterprises Build Secure Private Clouds; <http://cloudcomputing.sys-con.com/node/830646>; Cloud Computing Journal; Feb 29, 2009

**[S\_Martin\_MSFT]** Martin S.; Moving Toward an Open Process on Cloud Computing Interoperability; <http://blogs.msdn.com/steveamar/>

[archive/2009/03/26/moving-toward-an-open-process-on-cloud-computing-interoperability.aspx](#); Mar 26, 2009

[NW\_SaaS] Brodtkin J.; Open source fuels the growth of cloud computing SaaS; <http://www.networkworld.com/news/2008/072808-open-source-cloud-computing.html>; Aug 28, 2008

[Guardian\_Sep08] Johnson B.; Cloud computing is a trap, warns GNU founder Richard Stallman; <http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman>; 29 Sep 2008

[Pwr\_Std] Hamilton J.; Cost of Power in Large Datacenters; <http://perspectives.mvdirona.com/2008/11/28/CostOfPowerInLargeScaleDataCenters.aspx>; 28 Nov 2008

[Bain\_09] Bain, T.; Is the Relational Database Doomed?; [http://www.readwriteweb.com/archives/is\\_the\\_relational\\_database\\_doomed.php](http://www.readwriteweb.com/archives/is_the_relational_database_doomed.php); Feb 12, 2009

[Maya\_WrCI] Lucas P., Ballay J., Lombreglia R.; The Wrong Cloud; [http://www.maya.com/file\\_download/126/The%20Wrong%20Cloud.pdf](http://www.maya.com/file_download/126/The%20Wrong%20Cloud.pdf); Mar 2009

[CC\_Testing] Michelsen J., English, J.; Demystifying Cloud-Based Testing and the Related Benefits; iTKO Inc, [www.itko.com](http://www.itko.com); June 2009

[VM\_MT] Govindavajhala S., Appel A.; Using Memory Errors to Attack a Virtual Machine; 2003 IEEE Symposium on Security and Privacy, pp. 154-165, May 2003

[Vers07] Versant GmbH; Database Scalability and Clustering; [http://www.versant.com/developer/resources/objectdatabase/whitepapers/vsnt\\_whitepaper\\_scalability\\_clustering.pdf](http://www.versant.com/developer/resources/objectdatabase/whitepapers/vsnt_whitepaper_scalability_clustering.pdf); 2007

[MemArc] Hoff T.; Are Cloud Based Memory Architectures the Next Big Thing?; <http://highscalability.com/are-cloud-based-memory-architectures-next-big-thing>; 17 Mar 2009

[DCH\_Apr\_08] Chappell D.; Cloud Platforms Today: A Perspective; <http://www.davidchappell.com/CloudPlatformsToday--APerspective--Chappell.pdf>; 18 Apr 2009

**[NYT\_HDP]** Gottfrid D.; Self-service, Prorated Super Computing Fun!;

<http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/>; Nov 1 2007

**[RAD\_blog]** RAD Lab at UC Berkeley; Above the Clouds; A Berkeley View of Cloud Computing; <http://berkeleyclouds.blogspot.com/>; June 2009

**[Jaeger\_P]** Jaeger P., Lin J., Grimes J., Simmons S.; Where is the cloud? Geography, economics, environment and jurisdiction in cloud computing; First Monday Journal, Vol. 14, Num. 5; 4 May 2009

**[Econ2]** The Economist; Corporate IT - Highs and Lows; Oct 23<sup>rd</sup> 2009

**[BigT]** Chang F. et al.; Bigtable: A Distributed Storage System for Structured Data; Google, OSDI 2006; <https://learn.wu.ac.at/dotlrn/classes/ubn/0988.09s/xowiki/download/file/bigtable.pdf>

**[MapR]** Dean J., Ghemawat S.; MapReduce: Simplified Data Processing on Large Clusters; Google, OSDI 2004; <https://learn.wu.ac.at/dotlrn/classes/ubn/0988.09s/xowiki/download/file/mapreduce.pdf>

**[YahooH]** Hadoop and Distributed Computing at Yahoo!; <http://developer.yahoo.com/hadoop/>; June 2009

**[Newsweek08]** Choney S.; A Dark Cloud; <http://www.newsweek.com/id/151088>; Newsweek Magazine; Aug 6<sup>th</sup> 2008