

# CLOUD COMPUTING AN OVERALL VIEW

## ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΕΦΟΥΣ: ΜΙΑ ΣΥΝΟΛΙΚΗ ΘΕΩΡΗΣΗ

Ντίνας Θεόδωρος

8 January 2012

**Πανεπιστήμιο Μακεδονίας**

**University of Macedonia**

**ΔΠΜΣ Πληροφοριακά Συστήματα**

**Master Information Systems**

**Δίκτυα Υπολογιστών**

**Computer Networks**

**Καθηγητής: Α.Α. Οικονομίδης**

**Professor: A.A. Economides**

## Abstract

This survey is addressing to readers with knowledge of computers and networks and gives them an overall view to cloud computing — what it is and how it may help companies, public agencies, research organizations and institutes, researchers or developers, rethink how they deploy technology. Moreover, this survey tries to give a comprehensive answer in the most fundamental questions that a new comer would ask him self about cloud computing. First of all, there is a presentation given about the topic and the problem and afterwards the fundamental questions are answered. These basic questions are: Is cloud computing a new paradigm shift? What exactly is cloud computing and did it appear suddenly or it evolved somehow? What purposes does cloud computing serve? What is the main goal of cloud computing? Is cloud computing network computing or outsourcing? Why choose cloud computing instead of other traditional computing technologies? What is served through the cloud? Which aspects of quality of service in cloud computing is acquired from end users? How do clouds generally work? What are the main advantages of clouds? What are the main disadvantages of clouds? What types of clouds have developed? How much will it cost an enterprise/organization/developer/user to move to the cloud? Finally, there is a conclusion about the usage of clouds and some promising good ideas for future cloud computing.

## Περίληψη

Αυτή η εργασία/έρευνα απευθύνεται σε αναγνώστες με γνώσεις υπολογιστών και δικτύων και τους δίνει μία συνολική θεώρηση της υπολογιστικής νέφους—τι είναι και πώς μπορεί να βοηθήσει τις επιχειρήσεις, τους δημόσιους οργανισμούς, τους ερευνητικούς οργανισμούς και ινστιτούτα, ερευνητές και προγραμματιστές, να ξανασκεφτούν τον τρόπο με τον οποίον μπορούν να αναπτύξουν την τεχνολογία τους. Επίσης, αυτή η εργασία/έρευνα προσπαθεί να δώσει μία ολοκληρωμένη και κατανοητή απάντηση στα πιο ουσιώδη ερωτήματα που ένας νεοεισερχόμενος στον χώρο της υπολογιστικής νέφους θα αναρωτιόταν. Πρώτα απ' όλα, γίνεται μία παρουσίαση του θέματος και του προβλήματος και μετά δίνεται απάντηση στα ουσιώδη ερωτήματα. Αυτά τα βασικά ερωτήματα είναι : Αποτελεί η υπολογιστική νέφος μια νέα αλλαγή? Τι ακριβώς είναι η υπολογιστική νέφος και εμφανίστηκε ξαφνικά ή εξελίχτηκε κάπως? Ποιους σκοπούς εξυπηρετεί η υπολογιστική νέφος? Ποιος είναι ο βασικός στόχος της υπολογιστικής νέφος? Είναι η υπολογιστική νέφος υπολογιστική δικτύου ή εξωτερική ανάθεση? Γιατί να μην επιλεγούν άλλες παραδοσιακές τεχνολογίες υπολογιστικής από την υπολογιστική νέφος? Τι εξυπηρετείται στο νέφος? Ποιες πτυχές της ποιότητας προσφερομένων υπηρεσιών της υπολογιστικής νέφος απαιτούνται από τους τελικούς χρήστες της? Πώς λειτουργεί γενικά η υπολογιστική νέφος? Ποια είναι τα βασικά πλεονεκτήματα της υπολογιστικής νέφος και ποια τα μειονεκτήματα της? Ποιοι τύποι νεφών έχουν αναπτυχθεί? Πόσο θα κοστίσει σε μία εταιρία/οργανισμό/προγραμματιστή /χρήστη να μετακομίσει/κινηθεί στο νέφος? Τέλος, διατυπώνονται κάποια συμπεράσματα σχετικά με την υπολογιστική νέφος και παρουσιάζονται κάποιες καλές ιδέες σχετικά με το πολλά υποσχόμενο μέλλον της υπολογιστικής νέφος.

In the last decade, technological achievements such as sensor technology, increase of computational power, increase in available bandwidth and popularity of handheld devices created a new technological environment where it is easy for any user to provide and consume content of any form. Furthermore, building a personal Web page or starting a blog and making both searchable for the public worldwide are reality. Moreover, web applications face a much broader and geographically more diverse user base than before. Hence, the inevitable result is an overwhelming flow of data.

The problem of the management of this large scale data/data sets sizes for applications which are growing in incredible scale combined with the imbalanced computer architecture caused by the latency gap between multicore CPU's & mechanical hard disks which is also growing every year has created a new need: The need for data intensive computing, new data management approaches, new tools to manage, visualize and analyze data flood.

Additionally, a new aspect of online services has occurred: quality of service. If the network mechanisms are neglected, the applications will just have access to a best effort service, which is not enough to their requirements. Likewise, applications designed to run in the cloud must be designed to recover from failures that happen all the time. You have to assume the QoS will be poor from PaaS vendors and design for failures (Rosh 2010).

The solution to the problem of the management of this large scale data is to change accordingly computing infrastructure: computing architecture and processing mechanisms. There are two ways to achieve this. The first way is to build bigger and faster single supercomputers and the second way is to use cheaper clusters of computers (cloud computing).

Hurwitz et al. (2010, p. 8) claimed that cloud computing is the next stage in evolution of the Internet. The cloud in cloud computing provides the means through which everything — from computing power to computing infrastructure, applications, business processes to personal collaboration -can be delivered to you as a service wherever and whenever you need.

Is cloud computing a new paradigm shift? Sarna (2010, p. xxv) argued that cloud computing isn't a new paradigm shift but a natural development to meet needs that have been evident for more than forty years. Virtualization is the key technology that enables cloud computing. Remote hosting has developed from simply renting infrastructure to providing and maintaining standardized virtual servers that can be scaled up and down as demand fluctuates. But what exactly is cloud computing and did it appear suddenly or it evolved somehow? Computing evolved, as show in figure 1, following two ways. The first way is the distribution of computational power and the second way is through parallel processing. On one hand, the main characteristics of distributed computing are the distribution of data and workload through many computers and that the system is abstract and transparent to the user. This means that the user has the perception that he is working with one computer. In addition there is no single program and every element of the system has a different program for a certain workload. A paradigm of this architecture is client/server architecture. On the other hand, in parallel processing there is a single program for a certain workload witch is programmed from the user to run into different computers (or cpu's in the micro world). A paradigm shift of this computing is greed computing. Nowadays, both architectures are joint into what we call cloud computing.

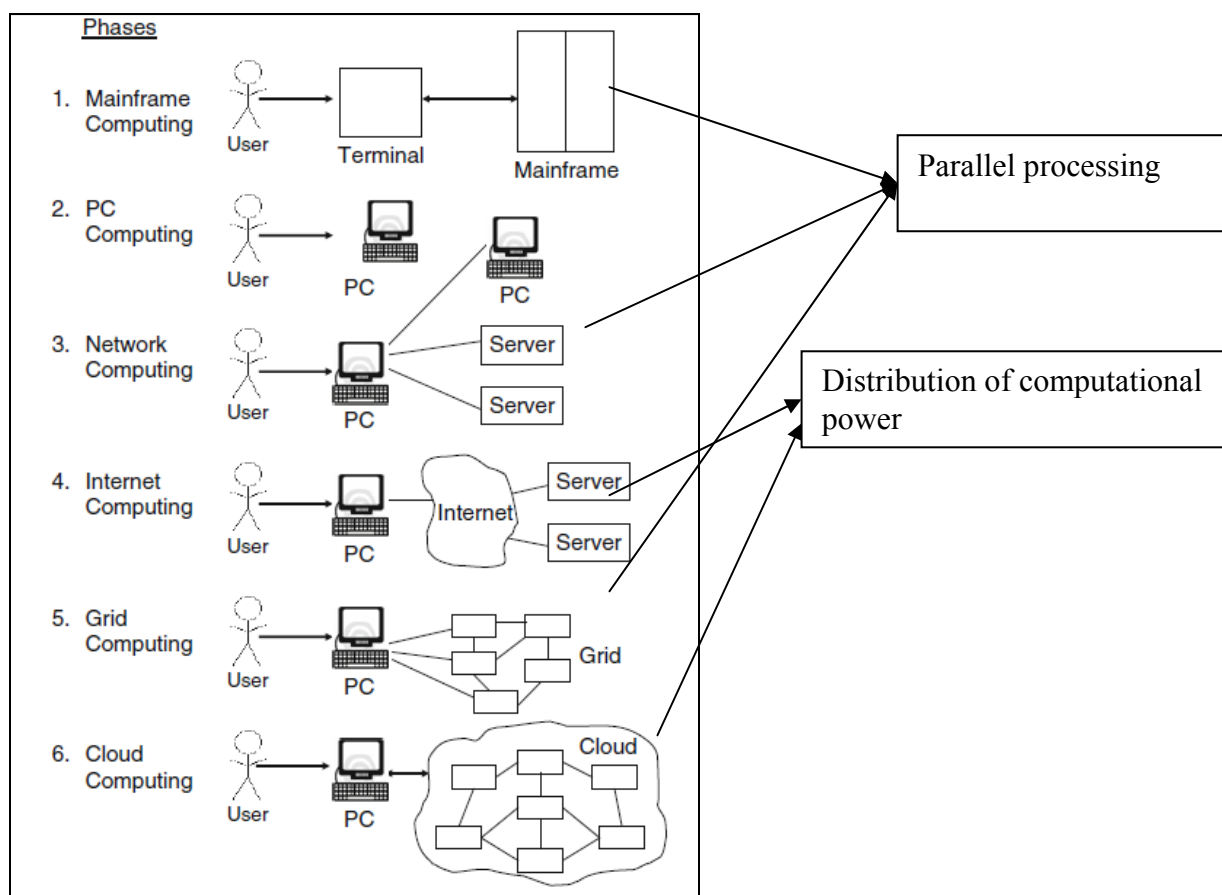


Figure 1. The evolution of computing from the poster. The poster: "Six computing paradigms" (Furht, 2010, p.4).

Just what purposes does cloud computing serve? As Figure 2 shows, according to Rajan and Jairath (2011, p.666), Cloud Computing makes a virtual pool of resources such as storage, CPU, networks and memory to fulfill the user's resource requirement and provides on demand (pay per use) hardware and software without barriers. It can be named as dynamic

computing because it provides resources when required (dynamically). Cloud Computing manages the pool of resources automatically and dynamically through software and hardware. On the bottom end it can make Internet as a desktop. As we work on desktop, Cloud Computing can be used in the same manner.

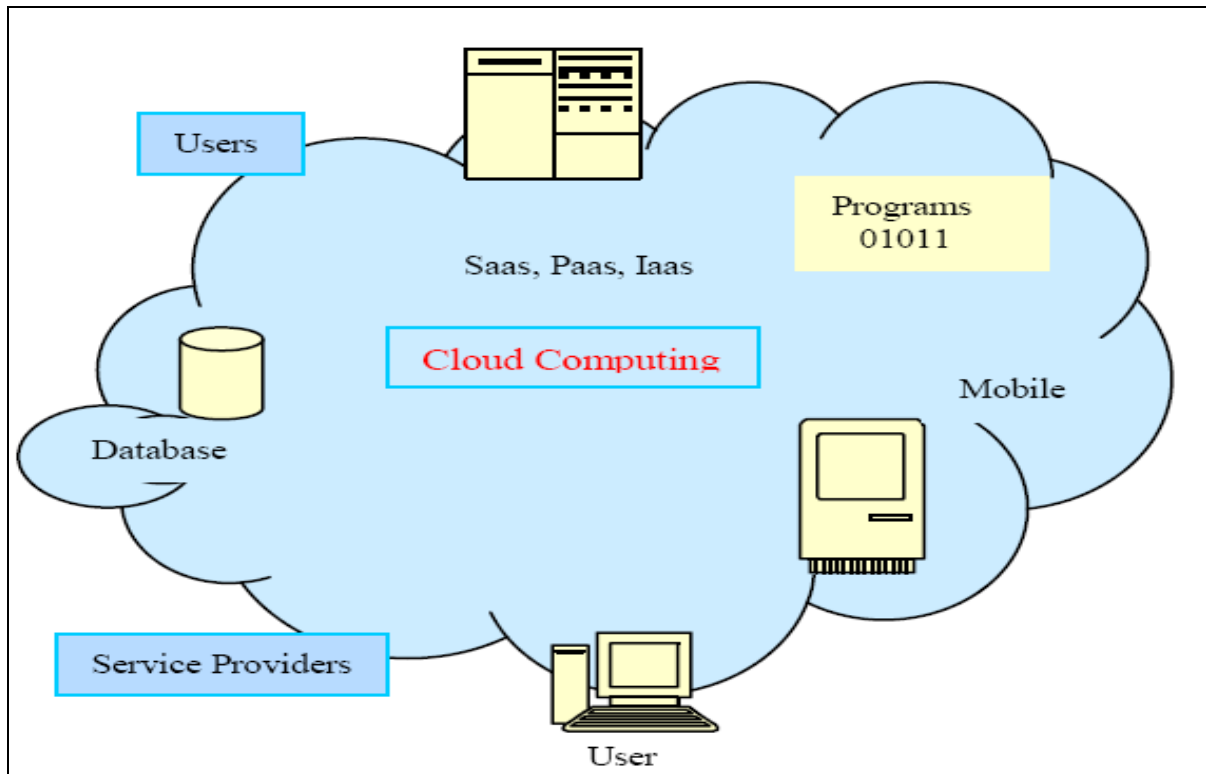


Figure 2: The poster "figure 1", ( Rajan and Jairath , 2011, p.666).

Another similar perspective is given in wikipedia as figure 3 shows.

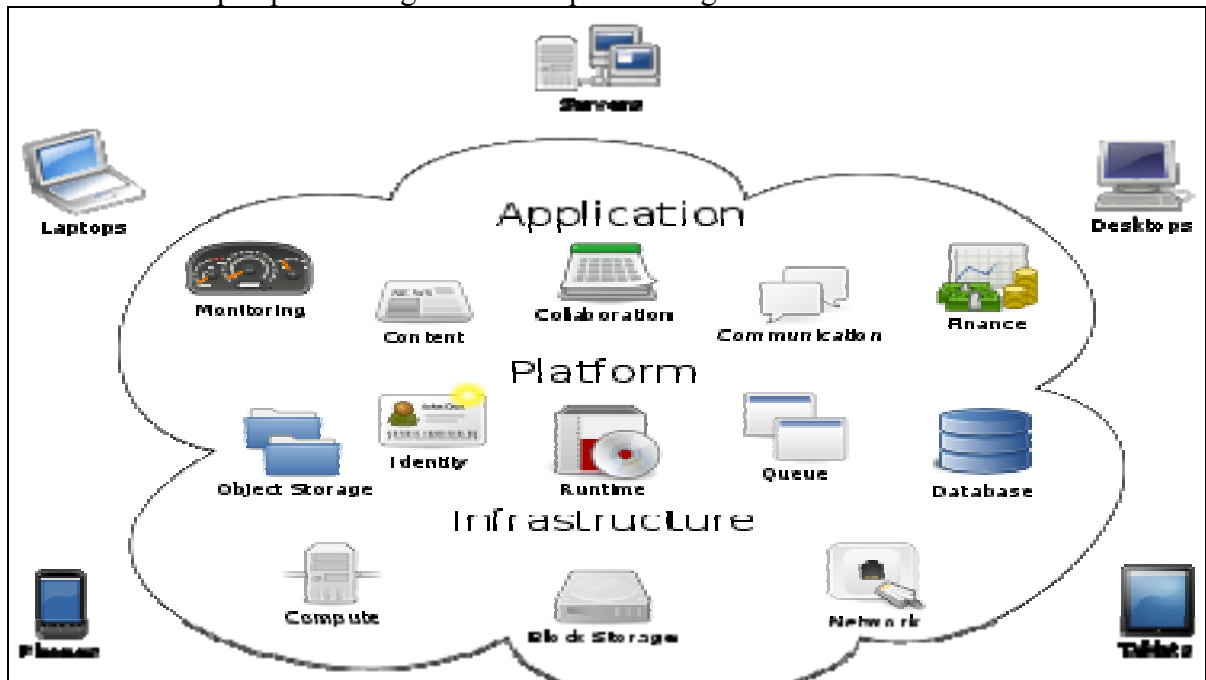


Figure 3: Cloud computing (Johnston 2009).

What is the main goal of cloud computing? As Sakr et al. claim (2011, p. 311), the main goal of cloud computing is to facilitate the job of implementing every application as a distributed, scalable and widely-accessible service on the Web with the minimum amount of effort in terms of software, CPU and network (for example make it easy for anybody to create services such as Facebook, Flickr and YouTube).

Is cloud computing network computing or outsourcing? According to Miller (2009, p.13), cloud computing isn't network computing. With network computing, applications/documents are hosted on a single company's server and accessed over the company's network. Cloud computing is a lot bigger than that. It encompasses multiple companies, multiple servers, and multiple networks. Plus, unlike network computing, cloud services and storage are accessible from anywhere in the world over an Internet connection; with network computing, access is over the company's network only. Similarly, Cloud computing also isn't traditional outsourcing, where a company farms out (subcontracts) its computing services to an outside firm. While an outsourcing firm might host a company's data or applications, those documents and programs are only accessible to the company's employees via the company's network, not to the entire world via the Internet. So, despite superficial similarities, networking computing and outsourcing are not cloud computing.

But, why choose cloud computing instead of other traditional computing technologies? The three main reasons which are leading to cloud computing are:

1. High storage and management costs of traditional infrastructures which is also the main disadvantage of building a single supercomputer
2. Reduced costs associated with the management of hardware and software resources which is the main advantage of creating clusters of computers.
3. Computing as a utility: businesses and users become able to access application services from anywhere in the world on demand.

On the bottom end, shifting the location of the infrastructure technology to the network creates economy of scale principles businesses and users become able to access application services from anywhere in the world on demand.

Just what is served through the cloud? According to Sakr et al. (2011, p 312) the definition specifies the following levels (figure 4) of abstraction which are referred to as service models:

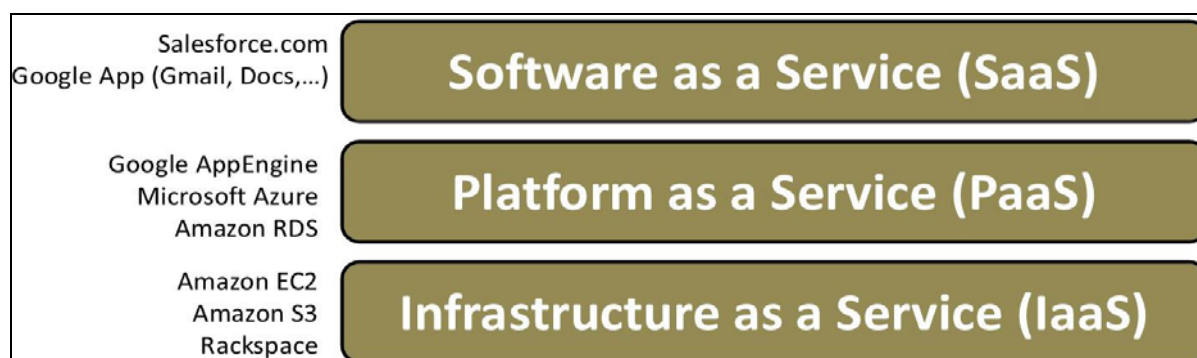


Figure 4: The poster "Models of Cloud Services" (Sakr, Liu, Batista, & Alomari, 2011, p.2).

1) Infrastructure as a Service (IaaS): Rajkumar and Priyesh (2011, p.1) say that the clients can create a virtual processing system and application environment by specifying choice of processing power, storage, network parameters, servers (often in the form of virtual machines), network bandwidth, storage, and related tools necessary to build an application environment from scratch and have control over operating. (e.g. Amazon EC210).

2) Platform as a Service (PaaS): Wu (2010, p.2) argues that cloud systems can offer an additional abstraction level: instead of supplying a virtualized infrastructure, they can provide the software platform where systems run on. The sizing of the hardware resources demanded by the execution of the services is made in a transparent manner. A well-known example is the Google Apps Engine. As Teeter & Barksdale (2008,p.11) mention about Google Apps Engine, is that users/companies can create, edit, and store massive amounts of information through the cloud with any device that has a Web browser (such as a computer or cell phone) and Internet access. The Web applications (or apps, for short) as well as the files themselves, are stored securely on powerful servers in data centers throughout the world. Users can share their information with others, including friends and coworkers, and collaborate in real time on important projects. Because the files are already online, a user simply sends a message to his or her friend that contains a link to a file, and those friends can click the link to see and contribute to the sender's great work. Using Google Apps, you don't need to send attachments back and forth or keep track of different file versions.

Finally, the maintenance, load-balancing and scale-out of the platform are done by the service provider and the developer can concentrate on the main functionalities of his application. Some known platforms per vendors are shown in table 1.

Table 1 Platform per Vendors (Sarna, 2010.p.41)

Vendors	Platform
Amazon	Amazon.com; (see <a href="http://www.amazon.com/gp/browse.html?node=201590011">www.amazon.com/gp/browse.html?node=201590011</a> )
Google	Google AppEngine12; (see <a href="http://code.google.com/appengine">http://code.google.com/appengine</a> )
Sun	Zembly.com for creating and hosting social applications and Network.com for pay-as-you-go computing
Microsoft	Azure; see <a href="http://www.microsoft.com/azure/windowsazure.mspx">http://www.microsoft.com/azure/windowsazure.mspx</a> )
IBM	IBM Cloud; see ( <a href="http://www-03.ibm.com/press/us/en/pressrelease/22613.wss">http://www-03.ibm.com/press/us/en/pressrelease/22613.wss</a> )
VMware	Eucalyptus; <i>Source:</i> <a href="http://www.eucalyptus.com/news/06-16-2010">http://www.eucalyptus.com/news/06-16-2010</a>
GoGrid, Joyent, , and others	GoGrid, Joyent, and others
Rackspace	OpenStack; a new open source effort led by Rackspace and NASA; See Sean Michael Kerner, Rackspace, NASA Partner on OpenStack Cloud Computing Install," Internet.com, July 19, 2010. Available at <a href="http://www.serverwatch.com/news/article.php/3893726/Rackspace-NASA-Partner-on-OpenStack-Cloud-Computing-Install.htm">www.serverwatch.com/news/article.php/3893726/Rackspace-NASA-Partner-on-OpenStack-Cloud-Computing-Install.htm</a> (accessed July 20, 2010). It is currently built with the Ubuntu Linux distribution and using the KVM virtualization hypervisor, is compatible with Amazon's AWS and is expected to run directly on Linux as well as be compatible with VMware, Xen or Hyper-V.
CohesiveFT	Elastic Server <i>Source:</i> <a href="http://www.cohesiveft.com/elastic/">www.cohesiveft.com/elastic/</a> wraps code in containers for better portability and management and protects from un-needed differentiation and vendor lock-in. VPN-Cubed IPsec <i>Source:</i> <a href="http://www.cohesiveft.com/vpncubed/">http://www.cohesiveft.com/vpncubed/</a> supports hybrid architectures
Zend Technologies	Zend [ <a href="http://www.zend.com">www.zend.com</a> ] and Abique for vendor independence
Open source cloud-optimized versions of Linux	Peppermint, CloudLinux, and Ubuntu Enterprise Cloud
Abique	Abique <i>Source:</i> <a href="http://www.abiquo.com/products/abiquo-overview.php">http://www.abiquo.com/products/abiquo-overview.php</a>
CA	3Tera ( <a href="http://www.3tera.com/Cloud-computing">www.3tera.com/Cloud-computing</a> ) for vendor independent

	frameworks
Elastra	Elastra ( <a href="http://www.elastra.com">www.elastra.com</a> ) to automate scalability

A nice example of Platform-as-a Service (PaaS) cloud computing is shown in Figure 5 where the PaaS provides Integrated Development Environment (IDE) including data security, backup and recovery, application hosting, and scalable architecture.

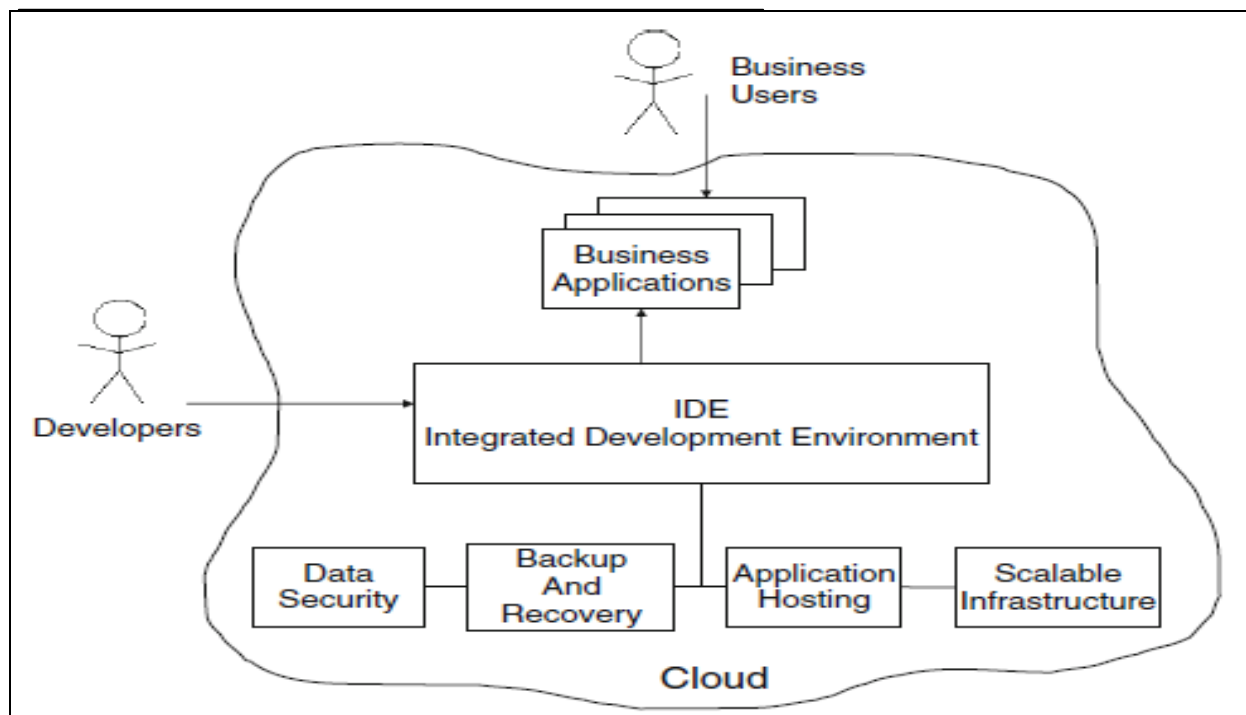


Figure 5: The poster “The concept of Platform-as-a-Service” (Furht, 2010, p.6)

Some words about scripting languages of Paas:

Sarna (2010.p.55) claims that much enterprise development is not done with traditional compiled programming languages, but with scripting languages. To start, it is helpful to distinguish scripting languages from compiled languages, as illustrated in Table 2.

Table 2 : The table “Where scripts are located and executed” Sarna (2010.p.56) .

Method	Where script code is located	Where script code is executed
CGI*	In files in the CGI-BIN directory on the server.	On the server.
PHP, ColdFusion, ASP	Embedded in the HTML document.	On the server.
Javascript	Embedded in the HTML document.	On the user's PC, by browser.
Java**	In files on server.	On user's computer, by browser.

\* Perl and CGI are NOT the same. CGI programs, or scripts, can be written in a variety of computer languages, including C. CGI is the process by which scripts are run. However, Perl is the most commonly used language for writing CGI scripts.

\*\* Java is a compiled language.



Microsoft Visual Studio and Other Development Environments: According to Sarna (2010.p.56) today, most new (full-bodied) applications are developed for the enterprise either using Microsoft's .Net architecture with Visual Studio or in the C++, Java or Python languages using other tools. Simpler projects or parts of projects may be coded in PHP or PERL (CGI scripts), and run in a J2EE container, if necessary, so they can integrate well with Java; they may also be coded in JavaScript. Some developers are fanatical about C# and Ruby. Considerable in-house development is done with cascading style sheets (CSSs) and ASP.Net, a Web framework. ASP.Net applications need to be hosted on a Windows hosting provider; there are many. Programs developing with .Net tools will (after recompilation) run most directly in the Microsoft Azure cloud platform. They also will run in the cloud on a Windows virtual server. C++ produced with Microsoft tools is, of course, compatible with that produced by other C++ compilers. Visual Studio is probably the most popular enterprise development tool. The latest version, Visual Studio 2010 is a single development environment for Windows, Web, Cloud, Office and Sharepoint development. Built-in languages include C/C++ (using Visual C++), VB.NET (using Visual Basic .NET), C# (via Visual C#), and F#. Support for other languages such as M, Python, and Ruby, among others, is available via language services installed separately. It also supports XML/XSLT, HTML/ XHTML, JavaScript, and CSS. Visual Studio 2010 is packaged in a range of editions for different needs.

3) Software as a Service (SaaS): Refers to special-purpose software made available through the Internet (e.g. Sales-Force13). Therefore, it does not require each end-user to manually download, install, configure, run or use the software applications on their own computing environments. Some services per provider are given in table 3.

Table 3: The table "Cloud service providers and their offerings", Sarna (2010.p.42)

Product	Services	Link
Strike Iron	Address verification, sales tax calculation, SMS and e-mail alerts	<a href="http://www.strikeiron.com">www.strikeiron.com</a>
Xignite	Financial data feeds	<a href="http://www.xignite.com/">www.xignite.com/</a>
Google Maps	Embedded maps	<a href="http://goo.gl/uD1N">http://goo.gl/uD1N</a>
Google Voice	Place calls, send SMS, download voicemails/ recorded messages	<a href="http://goo.gl/xcWl">http://goo.gl/xcWl</a>
Salesforce.com	Sales management, service management	<a href="http://www.salesforce.com/">www.salesforce.com/</a>
Force.com	Cloud development platform	<a href="http://www.salesforce.com/platform">www.salesforce.com/platform</a>
Yahoo BOSS	Open search Web services platform.	<a href="http://goo.gl/6pTx">http://goo.gl/6pTx</a>
Bloomberg	Financial data	<a href="http://goo.gl/8as1">http://goo.gl/8as1</a>
USPS	Address information, tracking	<a href="http://goo.gl/ckNg">http://goo.gl/ckNg</a>
NetSuite	Financials, CRM, inventory, and e-commerce	<a href="http://www.netsuite.com">www.netsuite.com</a>
RightNow Technologies	Customer experience across the Web, social networks, and contact center	<a href="http://www.rightnow.com/">www.rightnow.com/</a>

According to (Furht, 2010, p.6) there are three categories of cloud services, as illustrated in Figure 6. Figure 6a shows the cloud service SaaS, where the entire application is running in the cloud. The client contains a simple browser to access the application. A well-known example of SaaS is salesforce.com. Figure 6b illustrates another type of cloud services, where the application runs on the client; however it accesses useful functions and services provided in the cloud. An example of this type of cloud services on the desktop is Apple's iTunes.

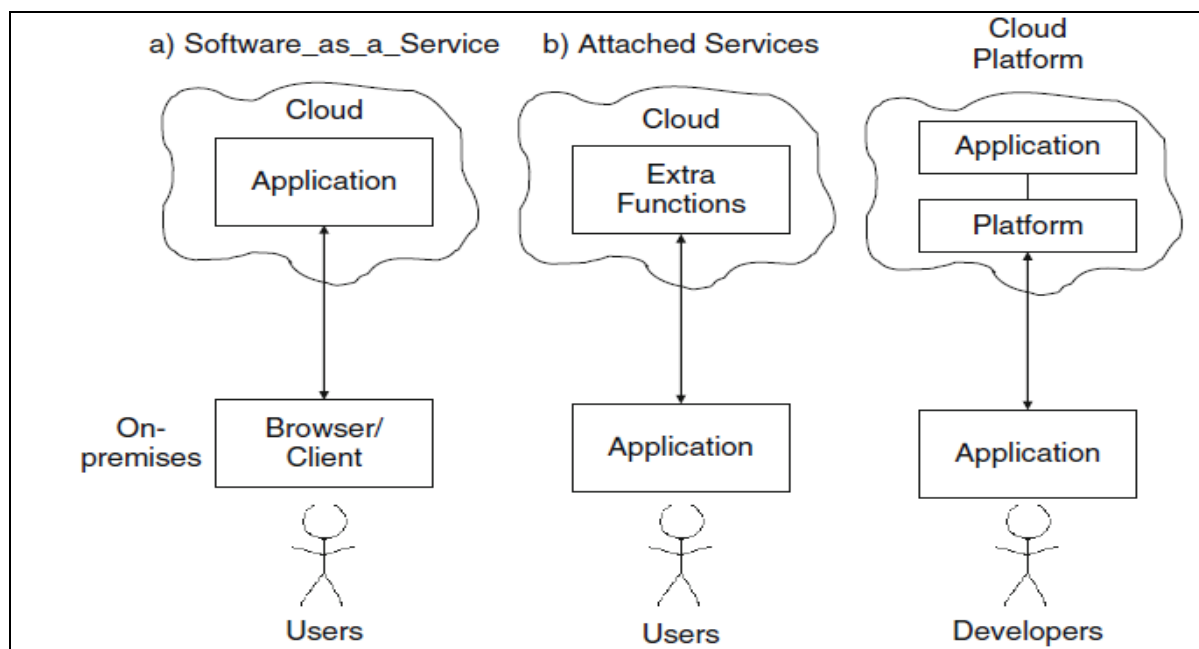


Figure 6: The poster “The categories of cloud” (Furht, 2010, p.6)

Which aspects of quality of service in cloud computing is acquired from end users? As Sakr et al. (2011, p. 315) argues, the main objectives of a cloud which are called ACID guarantees are:

1. Availability:
  - a. Users can read and write data at any time without ever being blocked.
  - b. The response times are (virtually) constant and do not depend on the number of concurrent users, the size of the database or any other system parameter
  - c. Users do not need to worry about backups. If components fail, it is the responsibility of the service provider to replace them and make the data available using replicas in the meantime), scalability(storage provided is virtually infinite and the network links are virtually capable to quickly transfer any quantity of data between the available servers)
  - d. Reliability in case of site failure. They must be always accessible even on the occasions where there is a network failure or a whole datacenter has gone offline.
2. Scalability: They must be able to support very large databases with very high request rates at very low latency. They should be able to take on new tenants or handle growing tenants without much effort beyond that of adding more hardware. In particular, the system must be able to automatically redistribute data to take advantage of the new hardware.
3. Elasticity: They must be able to satisfy changing application requirements in both directions (scaling up or scaling down). Moreover, the system must be able to gracefully respond to these changing requirements and quickly recover to its steady state.

4. Performance: On public cloud computing platforms, pricing is structured in a way such that one pays only for what one uses, so the vendor price increases linearly with the requisite storage, network bandwidth, and compute power. Hence, the system performance has a direct effect on its costs. Thus, efficient system performance is a crucial requirement to save money.
5. Multitenancy: They must be able to support many applications (tenants) on the same hardware and software infrastructure. However, the performance of these tenants must be isolated from each another. Adding a new tenant should require little or no effort beyond that of ensuring that enough system capacity has been provisioned for the new load.
6. Load and Tenant Balancing: They must be able to automatically move load between servers so that most of the hardware resources are effectively utilized and to avoid any resource overloading situations.
7. Fault Tolerance: For transactional workloads, a fault tolerant cloud data management system needs to be able to recover from a failure without losing any data or updates from recently committed transactions. Moreover, it needs to successfully commit transactions and make progress on a workload even in the face of worker node failures. For analytical workloads, a fault tolerant cloud data management system should not need to restart a query if one of the nodes involved in query processing fails.
8. Ability to run in a heterogeneous environment: On cloud computing platforms, there is a strong trend towards increasing the number of nodes that participate in query execution. It is nearly impossible to get homogeneous performance across hundreds or thousands of compute nodes. Part failures that do not cause complete node failure, but result in degraded hardware performance become more common at scale. A cloud data management system should be designed to run in a heterogeneous environment and must take appropriate measures to prevent performance degrading due to parallel processing on distributed nodes.
9. Flexible query interface: They should support both SQL and non-SQL interface languages (e.g MapReduce).

But how do clouds generally work? The corner stone of any application or information system is its data base. Therefore, a slight knowledge of data management system in the cloud is acquired especially if customers decide to recode their applications in order to deploy them in the cloud.

According to Abadi (2009, p.6) there have been developed two classes of software solutions in cloud data analysis: MapReduce-like software, and commercially available shared-nothing parallel databases. MapReduce and related software such as the open source Hadoop , useful extensions , and Microsoft's Dryad/SCOPE stack are all designed to automate the parallelization of large scale data analysis workloads. Second, the commercially available shared-nothing parallel databases, such as Teradata, Netezza, IBM DB2, Greenplum, DATAlegro, Vertica, and Aster Data, that already holds a reasonable market share for on-premises large scale data analysis. DB2, Greenplum, Vertica, and Aster Data are perhaps the most natural fit since they sell software-only products that could theoretically run in the data centers hosted by cloud computing providers. Vertica already markets a version of its product designed to run in Amazon's cloud.

As Sakr et al. (2011, p. 322) claim, these two techniques of data management systems in the cloud must fulfill certain properties which are listed bellow in order to achieve high quality of service (see also table 4 on how several cloud database systems hold on these properties):

1. Consistency: All records are the same in all replicas. It is achieved by using synchronous communications. However, we cannot practically use synchronous communications in view of partial failures (see bullet point N°3).
2. Availability: All replicas can accept updates or inserts. it is highly important for cloud-based applications to be always available and accept update requests of data and at the same time cannot block the updates even while they read the same data for scalability reasons.
3. Tolerance to Partitions: The system still functions when distributed replicas cannot talk to each other. To deal with partial failures of the system, cloud-based applications do not work in synchronous communications which halt the overall system because of the lifetime synchronization among the transaction participants.

In general, it is hard to maintain all three in the face of data replication over large geographic distances. The CAP theorem shows that a shared-data system can only choose at most two out of three properties: The ‘C’ (consistency) part is typically compromised to yield reasonable system availability. Hence, most of the cloud data management overcomes the difficulties of distributed replication by relaxing the guarantees of the system. In particular, they implement various forms of weaker consistency models (e.g. eventual consistency, timeline consistency, session consistency) so that all replicas do not have to agree on the same value of a data item at every moment of time. Therefore, transactional data management applications (e.g. banking, stock trading, supply chain management) which rely on the ACID guarantees that databases provide, tend to be fairly write-intensive or require microsecond precision are less obvious candidates for the cloud environment until the cost and latency of wide-area data transfer decrease significantly.

Table 4: The table “DESIGN DECISIONS OF CLOUD STORAGE SYSTEMS”,Sakr et al. (2011, p. 322)

System	Data Model	Query Interface	Consistency	CAP Options	License
BigTable	Column Families	Low-Level API	Eventually Consistent	CP	Internal at Google
Google AppEng	Objects Store	Python API - GQL	Strictly Consistent	CP	Commercial
PNUTS	Key-Value Store	Low-Level API	Timeline Consistent	AP	Internal at Yahoo
Dynamo	Key-Value Store	Low-Level API	Eventually Consistent	AP	Internal at Amazon
S3	Large Objects Store	Low-Level API	Eventually Consistent	AP	Commercial
SimpleDB	Key-Value Store	Low-Level API	Eventually Consistent	AP	Commercial
RDS	Relational Store	SQL	Strictly Consistent	CA	Commercial
SQL Azure	Relational Store	SQL	Strictly Consistent	CA	Commercial
Cassandra	Column Families	Low-Level API	Eventually Consistent	AP	Open source - Apache
Hypertable	Multi-dimensional Table	Low-Level API, HQL	Eventually Consistent	AP	Open source - GNU
CouchDB	Document-Oriented Store	Low-Level API	Optimistically Consistent	AP	Open source - Apache

What are the main advantages of clouds? Grobauer et al. (2011,p.52) and Sakr et al. (2011, p. 312) give us a clue about the main positive characteristics of clouds:

- 1) On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service’s provider.
- 2) Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

- 3) Resource pooling. Computing resources used to provide the cloud service are realized using a homogeneous infrastructure that's shared between all service users, processing, memory, network bandwidth, virtual networks and virtual machines.
- 4) Rapid elasticity. Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- 5) Measured Service. Resource/service usage is constantly metered, supporting optimization of resource usage, usage reporting to the customer, and pay-as-you-go business models.
- 6) Code reuse. As Sarna (2010, p. xxviii) claims, Service-Oriented Architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration. SOA separates functions into distinct units, or services, comprised of unassociated, loosely coupled units of functionality that have no calls to each other embedded in them. Developers make them accessible over a network in order to allow users to combine and reuse them in the production of applications. SOA, Web 2.0, and SOA 2.0 promote code reuse in a cloud environment. Calls to cloud-provided services, such as Google's AJAX APIs, also let you implement rich, dynamic Web sites entirely in JavaScript and HTML.

But aren't there any disadvantages in clouds? Several authors including Sakr et al. (2011, p. 315) found that deploying data-intensive applications on cloud environment is not a trivial or straightforward task and argued a list of obstacles to the growth of cloud computing applications as follows:

1. Availability of a Service: In principle, a distributed system is a system that operates robustly over a wide network. A particular feature of network computing is that the network links can potentially disappear. Organizations worry about whether cloud computing services will have adequate availability. High availability is one of the most challenging goals because even the slightest outage can have significant financial consequences and impacts customer trust.
2. Data Confidentiality: In general, moving data off premises increases the number of potential security risks and appropriate precautions must be made. Transactional databases typically contain the complete set of operational data needed to power mission-critical business processes. This data includes detail at the lowest granularity, and often includes sensitive information such as customer data or credit card numbers. Therefore, unless such sensitive data is encrypted using a key that is not located at the host, the data may be accessed by a third party without the customer's knowledge. The main goal of *Security* is to limit access only to those approved, to let those with approved see and/or modify only the data they are entitled to see and no other data, and to ensure that no one can requisition resources beyond their budget. However, as Sarna (2010, p. xxviii) argues many commercial service providers have better tools and facilities for ensuring RAS than do their clients. ISO 27001 and SAS 70 are two recognized standards designed for independently ensuring that third parties handling data have sufficient controls in place. These standards have been adapted for cloud security. The Cloud Security Alliance has been developing cloud-specific standards that will further improve on such standards. CloudAudit is developing an open, extensible, and secure interface that allows cloud computing providers to expose Audit, Assertion, Assessment, and Assurance (A6) information for cloud infrastructure (IaaS), platform (PaaS), and application (SaaS) services to authorized clients.
3. Data Lock-In: Customers have two choices of migrating to the Cloud: recode or to just make the existing applications work in cloud. APIs for cloud computing have not

been, yet, subject of active standardization. Thus, customers cannot easily extract their data and programs from one site to run on another. As Velte et al. (2010,p.32) point out, the customer might pay a provider to use an application, but once they do, they may be unable to port that application to a new vendor. Or, it might be possible to move to a new vendor, but the old vendor might charge a hefty moving fee. The concerns about the difficulties of extracting data from the cloud is preventing some organizations from adopting cloud computing. Data Transfer Bottlenecks: Cloud users and cloud providers have to think about the implications of placement and traffic at every level of the system if they want to minimize costs.

4. Application Parallelization: Computing power is elastic but only if workload is parallelizable. Getting additional computational resources is not as simple as just upgrading to a bigger and more powerful machine on the fly. However, the additional resources are typically obtained by allocating additional server instances to a task.
5. Shared-Nothing Architecture: Data management applications designed to run on top of cloud environment should follow a shared-nothing architecture where each node is independent and self-sufficient and there is no single point of contention across the system. Most of transactional data management systems do not typically use a shared-nothing architecture.
6. Performance Unpredictability: Many HPC applications (High-performance computing) need to ensure that all the threads of a program are running simultaneously. However, today's virtual machines and operating systems do not provide this service.
7. Application Debugging in Large-Scale Distributed Systems: A challenging aspect in cloud computing programming is the removal of errors in these very large scale distributed systems. A common occurrence is that these bugs cannot be reproduced in smaller configurations, so the debugging must occur at the same scale as that in the production datacenters.

Another important aspect in cloud computing is the suitability of approaches and mechanisms to support certain classes of applications and end users. Specifically, as Sakr et al. (2011, p. 315) claimed, knowing the characteristics of the applications and how they manage the data, it is possible to modify the network mechanisms in order to optimize the utilization of the cloud. For example, to make possible the load and tenant balancing, it is necessary to guarantee that the migration of the applications will take a time which will not affect the quality of the service provided to the user of these migrated applications. This time is directly related to network metrics like available bandwidth and delay. A flexible query interface can also influence the network configuration of the cloud since it can require the parallelism of some tasks, which needs the efficient transfer of data from one source to several destinations.

What types of clouds have developed? According to Furht, (2010, p.7) there are three types of cloud computing, as figure 6 illustrates, depending on who owns and uses them:

- 1) Private clouds : Private cloud (or internal cloud) refers to cloud computing on private networks. Private clouds are built for the exclusive use of one client, providing full control over data, security, and quality of service. Private clouds can be built and managed by a company's own IT organization or by a cloud provider.
- 2) Public clouds. In the public cloud (or external cloud) computing resources are dynamically provisioned over the Internet via Web applications or Web services from an off-site third-party provider. Public clouds are run by third parties, and applications from different customers are likely to be mixed together on the cloud's servers, storage systems, and networks.

- 3) Hybrid clouds. A hybrid cloud environment combines multiple public and private cloud models. Hybrid clouds introduce the complexity of determining how to distribute applications across both a public and private cloud.

As Sarna (2010,p.xxv) claims, private (often on premise) clouds can provide increased utilization compared with deploying, installing, and maintaining traditional farms of servers deployed on a taskper-server basis. Public clouds offer increased efficiency, but are perceived as being less secure. Newer hybrid solutions, such as IBM Cloudburst, Eucalyptus, and Windows AppFabric provide a consistent development experience for easy migration, redundancy, and scalability.

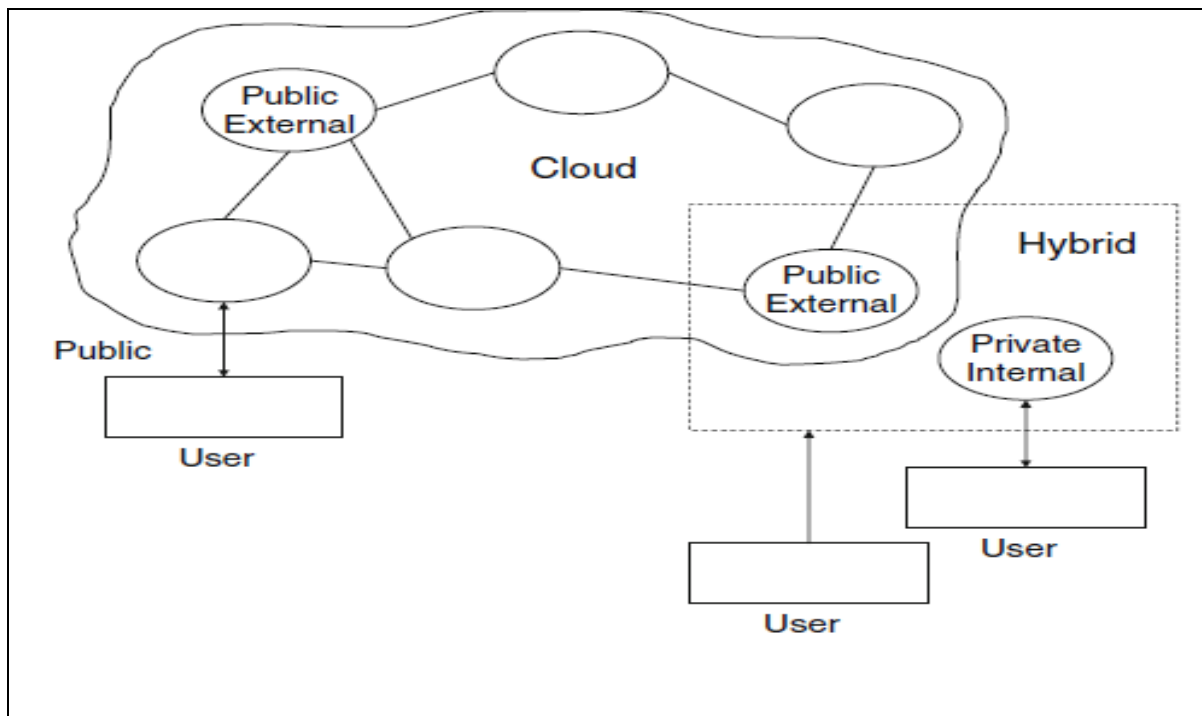


Figure 7: The poster “The three types of cloud computing” (Furht, 2010, p.7)

How much will it cost an enterprise/organization/developer/user to move to the cloud?

Sarna (2010.p.55) gives an example for cost by vendor and for certain usage. Specifically, if we use a normalized example of a virtual 8 GB RAM system with 320 GB of disk running Windows Server Enterprise 64-Bit operating system, operating continuously for a month, the charges from four large vendors are as shown in Table 5.

Table 5: The table “Vendor prices”, Sarna (2010.p.55).

Vendor	Month	Hourly	What is Provided
Rackspace Cloud	\$ 417	\$0.58	8 GB RAM 320 GB Disk included in base
Amazon—AWS	\$ 394	\$0.48	7.5 GB RAM, Incl. 320 GB S3 Storage
Terremark—vCloud	\$ 607	\$0.722	Incl. 8 GB RAM and 4 CPU + 320 GB Disk
SoftLayer—CloudLayer	\$ 435	\$0.50	Incl. Windows Server + 350 GB Disk

However, bandwidth charges can vary significantly (and are excluded), as can surcharges for heavy processor (CPU) usage and for extra disk space. Linux platforms are less expensive, as no licensing fees are due to Microsoft.

Another example is given by (Furht, 2010, p.15) in table 6.

Table 6: The table “Pricing comparison for major cloud computing platforms”, (Furht, 2010, p.15).

Resource	UNIT	Amazon	Google	Microsoft
Stored data	GB per month	\$0.10	\$0.15	\$0.15
Storage transaction	Per 10 K requests	\$0.10		\$0.10
Outgoing bandwidth	GB	\$0.10 – \$0.17	\$0.12	\$0.15
Incoming bandwidth	GB	\$0.10	\$0.10	\$0.10
Compute time	Instance Hours	\$0.10 – \$1.20	\$0.10	\$0.12



**Conclusion:**

On the bottom end as Sakr et al. (2011, p. 314) point out, cloud computing is addressing mostly to enterprises for the advantages mentioned below:

1. Eliminates the size of an enterprise which is a critical factor in its economic success
2. Eliminates the need for small companies to make a large capital expenditure in building an infrastructure to create a global customer base (no investments are needed upfront and the cost grows linearly and predictably with the increase of usage (pay-as-you-go).)
3. Allows an enterprise of any size to manage growth with the popularity of its product or service while at the same time also allows the enterprise to cut its losses if the launched product or service does not succeed
4. **Leverages the virtualization technology to maximize the utilization of computing resources.**
5. Allows companies to focus on their business innovation rather than focusing on building larger data centers to achieve scalability goals (For example, if a company is launching a new product that requires a large database, it does not need to go out and buy one. However, it can use one hosted by a cloud provider such as Amazon or Google).
6. Finally, as Jennings (2009, p.7) claimed, enterprises get new products or services to market faster by minimizing time to deploy fixed IT assets, such as servers, switches, and routers, and by eliminating related incremental capital investment in these assets.

On the other hand the main disadvantages which discourages companies to endorse cloud computing is the fact that while NoSQL systems (techniques others than SQL to retrieve/update data in cloud) have got a lot of attractions, enterprises are still very cautious to rely on these systems because there are many limitations still need to be addressed such as:

1. Programming Model: NoSQL systems offer limited support for ad-hoc querying and analysis operations. Therefore, significant programming expertise is usually required even for a simple query. In addition, missing the support of declaratively expressing the important join operation has been always considered one of the main limitations of these systems.
2. Transaction Support: Transaction management is one of the powerful features of traditional relational database systems. The current limited support (if any) of the transaction notion from NoSQL database systems is considered as a big obstacle towards their acceptance in implementing mission critical systems.
3. Maturity: Traditional relational database systems are well-know with their high stability and rich functionalities. In contrast, most NoSQL systems are open source projects or in pre-production stages where many key features are either not stable enough or still under development. Therefore, enterprises are still very cautious to deal with this new wave of database systems.
4. Support: Enterprises look for the assurance that if the system fails, they will be able to get timely and competent support. All RDBMS vendors have great experience in providing high level of enterprise support. In contrast, most NoSQL systems are open source projects. Although there are few firms offering support for NoSQL database systems, these companies are still small start-ups without the global reach, support resources or credibility of an Oracle, Microsoft or IBM.
5. Expertise: There are millions of developers around the world who are familiar with traditional concepts of relational database systems and programming models in every business domain. On the contrary, almost every NoSQL developer is still in a learning mode. It is natural that this limitation will be addressed over time. However, currently,

it is far easier to find experienced RDBMS programmers or administrators than a NoSQL expert.

On the overall, as Moretti et al. (2008, p.1) mentioned in their work, although modern parallel and distributed computing systems provide easy access to large amounts of computing power, it is not always easy for non-expert users to harness these large systems effectively. A large workload composed in what seems to be the obvious way by a naive user may accidentally abuse shared resources and achieve very poor performance.

Future proposals: As presented in the research of Ke et al. (2009, p.1) the architecture of current Cloud computing System is central structured one and all the data nodes must be indexed by a master server which may become bottle neck of the system. In their research, they propose new cloud storage architecture based on P2P and design a prototype system. The system based on the new architecture has better scalability and fault tolerance.

Another research from Nkosi & Mekuria.(2010,p.1) gives cloud computing a humanistic approach. Specifically, it argues that mobile devices are being considered as service platforms for mobile health information delivery, access and communication. However mobiles face challenges with regard to delivering secure multimedia based health services due to limitations in computation and power supply. Since mobile devices have limited computational capacity and run on small batteries they are unable to run heavy multimedia and security algorithms. In their article, a cloud computing framework to relieve mobile devices from executing heavier multimedia and security algorithms in delivering mobile health services is described. The proposed framework uses a Cloud Computing protocol management model which intends to provide multimedia sensor signal processing & security as a service to mobile devices. Their approach suggests that multimedia and security operations can be performed in the cloud, allowing mobile health service providers to subscribe and extend the capabilities of their mobile health applications beyond the existing mobile device limitations.

A very futuristic approach is presented in the paper of Agostinho et al.(2011, p.1110) witch paper presents the design, implementation, and evaluation of a cloud computing environment for networked robotics. The environment aims to support distributed robotics applications that require large shares of computing resources and very low communication delays. Networked robotics applications are built above these services and run on virtualized environments deployed on processing nodes directly connected to the robotic resources manipulated by the applications. The cloud computing environment offers the REALabs platform according to the Platform as a Service (PaaS) model of cloud computing. It also, shows an approach to evaluate robotic experiments by submitting tasks in a work flow cloud system.

Finally, as Ohlman and Eriksson (2009, p.1) illustrate in their research networking of Information (NetInf) is an information centric networking paradigm that can support cloud computing by providing new possibilities for network transport and storage. It offers direct access to information objects through a simple API, independent of their location in the network. Their abstraction can hide much of the complexity of storage and network transport systems that cloud computing today has to deal with. In their paper they analyze how cloud computing and NetInf can be combined to make cloud computing infrastructures easier to manage, and potentially enable deployment in smaller and more dynamic networking environments. NetInf should thus be understood as an enhancement to the infrastructure for cloud computing rather than a change to cloud computing technology as such.

## References

- Abadi J. D. (2009). *Data Management in the Cloud: Limitations and Opportunities*. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 1-10. Retrieved from IEEE.
- Agostinho L., Olivi L., Feliciano G., Paolieri, F., Rodrigues D. & Cardozo E.(2011). *A Cloud Computing Environment for Supporting Networked Robotics Applications*. 9<sup>th</sup> IEEE. International Conference on Dependable, Automatic and Secure Computing . IEEE computer society. 1110-1116. DOI 10.10.1109/DASC.2011.181.
- Roch E. (2010). *Cloud Computing Quality of Service*. Retrieved from <http://it.toolbox.com/blogs/the-soa-blog/cloud-computing-quality-of-service-37416>
- Grobauer B., Walloschek T. & Stöcker E.(2011). *Understanding Cloud Computing Vulnerabilities*. copublished by the IEEE computer and reliability societies. IEEE security & privacy. 50-57. Retrieved from [www.computer.org/security](http://www.computer.org/security).
- Furht, B. & Escalante, A.(2010). *Handbook of Cloud Computing*. DOI 10.1007/978-1-4419-6524-0.
- Jennings R. (2009). *Cloud Computing with the Windows® Azure™ Platform*. Indianapolis, Indiana, USA, Wiley Publishing, Inc
- Johnston S. [image]. (2009). *Diagram showing overview of cloud computing*. Retrieved from [http://en.wikipedia.org/wiki/File:Cloud\\_computing.svg](http://en.wikipedia.org/wiki/File:Cloud_computing.svg).
- Hurwitz, J., Bloor, R., Kaufman, M., & Dr. Halper, F. (2010). *Cloud Computing For Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc.
- Ke X. , Meina S. , Xiaoqi Z. & Junde S.(2009). *A Cloud Computing Platform Based on P2P*. DOI 978-1-4244-3930-0/09.Created by Sam Johnston using OminGroup's OmniGraffle and Inkscape (includes Computer.svg by Sasa Stefanovic)
- Miller M. (2009).*Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online*. Indianapolis, Indiana, USA, Que Publishing.
- Moretti C., Bulosan J., Thain D., & Flynn J. P.(2008). *All-Pairs: An Abstraction for Data-Intensive Cloud Computing*.DOI 978-1-4244-1694-3/08.
- Nkosi T.M.& F. Mekuria F.SM.(2010). *Cloud Computing for Enhanced Mobile Health Applications*. 2<sup>nd</sup> IEEE. International Conference on Cloud Computing Technologies & Science. DOI 10.1109/CloudCom.2010.31.
- Ohlman B.& Eriksson A.(2009). *What Networking of Information Can Do for Cloud Computing*. 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, DOI 10.1109/WETICE.2009.27

Rajan S. & Jairath A.,(2011). *Cloud Computing: The Fifth generation of Computing*. International conference on Communications Systems and Network Technologies . DOI 10.1109/CSNT.2011.143.

Rajkumar S. & Priyesh K. (2011). *An Intelligent Cloud Computing Architecture Supporting e-Governance*. Proceedings of the 17th International Conference on Automation & Computing, University of Huddersfield, UK, 10th September 2011, (pp 1-5).

Sakr, S., Liu, A., Batista, D. M. & Alomari, M.(2011). *A Survey of Large Scale Data Management Approaches in Cloud Environments*. Communications Surveys & Tutorials, Vol. 13, 311-336. Retrieved from IEEE.

Sarna, D., E.Y., (2010). *Implementing and developing cloud computing applications*. Boca Raton, Auerbach Publications, Taylor & Francis Group.

Teeter R. & Barksdale K. (2008). *Google™ Apps For Dummies*. Indianapolis, Indiana, USA, Wiley Publishing, Inc

Wu, J., Ping, L., Ge, X., Wang, Y. & Fu, J. (2010). *Cloud Storage as the Infrastructure of Cloud Computing*. International Conference on Intelligent Computing and Cognitive Informatics, (pp. 1-4). Retrieved from IEEE.

Velte T.A. , Velte J. T. & Elsenpeter R.(2010). *Cloud Computing: A Practical*. USA McGraw-Hill Companies.