International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.1, No.4, October 2012, pp. 183~199
ISSN: 2089-3337

Evolution of Cloud Computing and Enabling Technologies

Rabi Prasad Padhy*, Manas Ranjan Patra**
*Senior Software Engg, Oracle Corp., Bangalore, Karnataka, India
** PG Departement of Computer Science, Berhampur University, Odisha, India

Article Info

Article history:

Received Augt 01th, 2012
Accepted Sept 09th, 2012

Keyword:

Internet Computing
Grid Computing
Cloud Computing
Computing Paradigm
Virtualization
Web 2

ABSTRACT

We present an overview of the history of forecasting software over the past 25 years, concentrating especially on the interaction between computing and technologies from mainframe computing to cloud computing. The cloud computing is latest one. For delivering the vision of various of computing models, this paper lightly explains the architecture, characteristics, advantages, applications and issues of various computing models like PC computing, internet computing etc and related technologies and also we summarized the current model cloud computing. Finally in this research paper, we described the need for convergence of competing IT paradigms to deliver our vision.

Corresponding Author:

Rabi Prasad Padhy,
Senior Software Engineer,
Oracle Corporation, Bangalore
Karnataka , India.
Email: rabi.padhy@gmail.com

1. INTRODUCTION

The Fifth Generation of Computing (after Mainframe, Personal Computer, Client-Server Computing, and the web) is the Cloud Computing. The Computing is the systematic study of algorithmic processes that describe and transform information, their theory, analysis, design, efficiency, implementation, and application. The term "computing" is synonymous of counting and calculating. Cloud computing is defined as Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like the electricity grid." Firstly, in 1960, when John McCarthy opined that "computation may someday be organized as a public utility" a smell of a big advanced technology spread in world. With digital Internet became widely used in the second half of the 1990s ask any web developer, solution architect or anyone involved in web application development in any capacity. Cloud computing has derived its name from the same line of thinking. With the turn of the 21st century, the term "cloud computing" began to appear more widely used. There, are many new computing paradigms have been developed and adopted, with the emergence of technological advances such as multi-core processors and networked computing environments, to edge closer toward achieving the grand vision of computing. These new computing paradigms are cluster computing, Grid computing, P2P computing, service computing, market oriented computing, and most recently Cloud computing. Cloud computing has become a significant technology trend, driven by big players like Amazon, Microsoft, Google, Salesforce.com, and transforming our current IT industry. Cloud computing delivers large-scale utility computing services to a wide range of consumers. Within cloud computing, users on various types of devices access programs, storage, processing and applications over the Internet, offered by cloud computing providers, resulting in a previously unprecedented elasticity of resources.

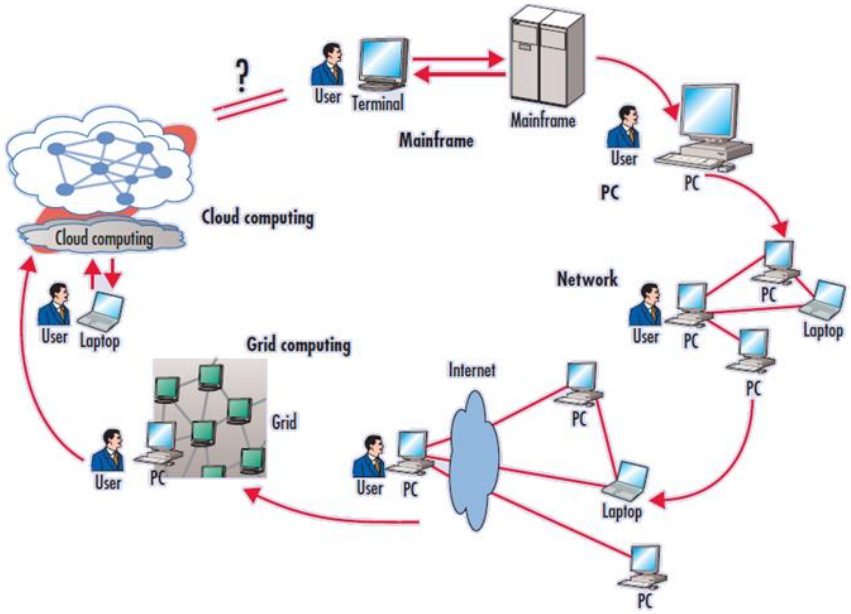

Figure 1. Evolution of Various Computing Models

2. EVOLUTION OF COMPUTING MODELS

2.1 Mainframe Computing

In early 1950s, programmers were having limitation in the form of batch files, in which complete jobs were submitted on the punch cards and operators were supposed to run it on a complex computer hardware consisting of power consuming mechanical relays, transistors, vacuum electronic I/O devices. Multi-programmed, batched systems were introduced to effectively utilize various system resources which then extended into time sharing systems in which CPU executes multiple jobs for several users switching among them. In time sharing system, now group of people, mainly programmers, could interact with computer, so it was significant milestone in interactive use of computer. In Time sharing systems, programmers were interacting with the computer using the terminals attached to the mainframes using a command line interface. Command line interface is the mechanism by which one can interact with computer system to operating system software on computer, by typing a textual command. A predefined set of instructions or commands are defined. One has to type there commands to perform specific tasks. Once the command is entered, its validity is checked and the corresponding task gets executed. The programmer could now collaborate with the computer in a more reactive and spontaneous manner within the limits of command language, increasing the information processing throughput and programmer's productivity. Mainframe computers are large, powerful computers that are usually used in big organizational settings. The application of mainframe computers include the processing of voluminous data as required for resource planning, census statistics, industry and consumer statistics and large financial transactions. The mainframes we use today date back to April 7, 1964, with the announcement of the IBM System/360™. System/360 was a revolutionary step in the development of the computer for many reasons, including the following:

- System/360 could do both numerically intensive scientific computing and input/output intensive commercial computing.
- System/360 was a line of upwardly compatible computers that allowed installations to move to more powerful computers without having to rewrite their programs.
- System/360 utilized dedicated computers that managed the input/output operations, which allowed the central processing unit to focus its resources on the application.
- These systems were short on memory and did not run nearly as fast as modern computers. For example, some models of the System/360 were run with 32K (K, as in 1,024 bytes) of RAM, which had to accommodate both the application and the operating system. Hardware and software had to be optimized to make the best use of limited resources.

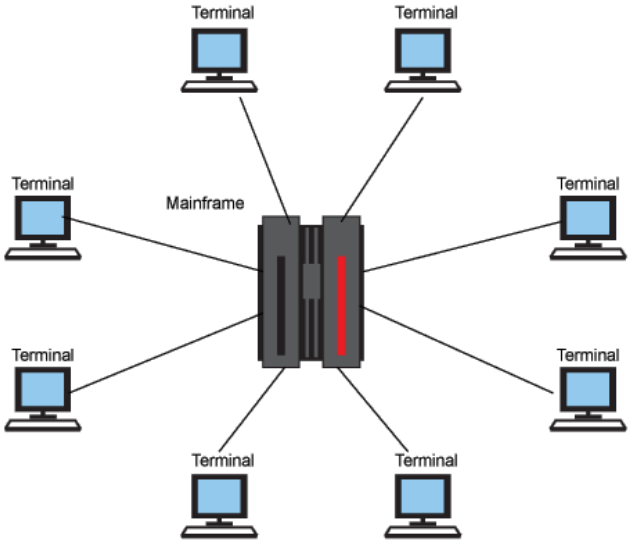

Figure 2. Mainframe Architecture

IBM have always dominated this market. Mainframe computers were under the control of professional programmers and systems operators and were highly centralized. As they developed, mainframe computers become powerful enough to support hundreds of online remote terminals connected to the centralized mainframe. With the advent of the personal computer, many people thought in the 1980s that mainframes would cease to exist. However, their ability to store and process huge amounts of data means that mainframes are still an important component of many IT infrastructures.

2.2 Personal Computing (PC)

Personal Computing systems considerably smaller and less expense then Mainframe Systems and more suitable for typical office environments and necessary for every personal executive. Traditionally executively were used to setting on a desk performing several tasks by using a set of tools spread on the top of the desk. Soon a personal computer starting occupying an important place on an executive's desk and all other tools on the desk including the time-clock become the part of the desktop interface of the PC. In Personal Computing, interaction focused on addressing the single user engaged in a dialog with the computer in order to carry out a series of tasks. Humans are by nature multitasking as they can think about more than one think at a time. They can effortlessly switch between tasks and have ability to carry out the task irrespective of series of external interrupts. The rise of the personal home computer has driven the need for employers to keep up and provide personal computers in the work place too. Microsoft Windows has dominated the personal computer but open source software such as Linux, which is not only free but also good, is starting to challenge this domination.

Figure 3. PC Computing from Desktop to Tablet PC

2.3 Network Computing

In the world of computers, networking is the practice of linking two or more computing devices together for the purpose of sharing data. Networks are built with a mix of computer hardware and computer software. Networks can be categorized in several different ways. One approach defines the type of network according to the geographic area it spans. Local area networks (LANs), for example, typically reach across a single home, whereas wide area networks (WANs), reach across cities, states, or even across the world. The Internet is the world's largest public WAN. Computer networks also differ in their design. The two types of high-level network design are called client-server and peer-to-peer. Client-server networks feature centralized server computers that store email, Web pages, files and or applications. On a peer-to-peer network, conversely, all computers tend to support the same functions. Client-server networks are much more common in business and peer-to-peer networks much more common in homes. Another way to classify computer networks is by the set of protocols they support. Network Protocols are the communication language used by computer devices. Networks often implement multiple protocols to support specific applications. Popular protocols include TCP/IP, the most common protocol found on the Internet and in home networks.

Client-Server: As personal computers and laptops become cheaper, organizations started to replace their mainframe terminals with PCs linked together in a network. At the heart of the network of PCs (clients) is a server (which might be a mainframe or a powerful PC) which stores some of the data, applications software and other instructions that the network users need in order to communicate and process transactions on the network. There are different types of servers. A web server provides web pages to users, an application server assigns specific tasks to other servers to enable a faster more efficient response to client requests than a single mainframe trying to do everything. Large organizations use a multi-tiered client/server architecture that has several different levels of servers. Client/server computing systems are comprised of two logical parts: a server that provides services and a client that requests services of the server. Together, the two form a complete computing system with a distinct division of responsibility. Clients serve as the consumers in a client/server system. That is, they make requests to servers for services or information and then use the response to carry out their own purpose. The server plays the role of the producer, filling data or service requests made by clients. Client/server computing has gained popularity in the recent years due to the proliferation of low-cost hardware and the increasingly apparent truth of the theory that a model relying on monolithic applications fails when the number of users accessing a system grows too high or when too many features are integrated into a single system.

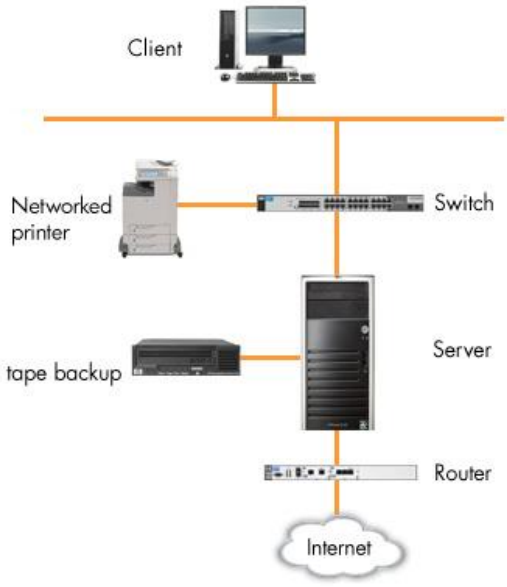

Figure 4. High level view of client server Model

A good example of a client/server system is a simple automated teller machine (ATM) network. Users typically use ATMs as clients to interface with a small sub-server, which collaborates with a larger server that manages all of the smaller servers. In this model, the sub-servers are servers to the ATMs and clients to the master server. ATMs provide the user interface and can be customized as required (e.g. for

multilingual support), while the intermediate servers provide the application logic, such as checking account balances and transferring money between accounts. The sub-servers allow the system to be scaled since adding servers allows an increased number of ATMs to be supported. However, the application logic is provided only with the help of the centralized server. The results of the services are communicated to the user through the ATMs. The centralized server provides additional application logic, such as ensuring that concurrent transactions are handled correctly. It also serves as a central brokerage for all account information so that users can access their accounts from any ATM worldwide.

Benefits of Client/Server Systems: As client/server systems have grown more robust, the computing community has acknowledged their many distinct advantages. Perhaps the most important advantage is the natural mapping of applications into a client/server framework. A typical example of this is an electronic phonebook system. Since the data is relatively static and the data repository needs to be able to respond to queries, it makes sense to construct this portion of the application as a server. A thin client is a logical match since it is difficult to update every user's database of phone numbers, the optimal search algorithm can change at any time, and the space required for the amount of data manipulated is prohibitive for many users' workstations. Application development is simplified since a client and server each fill a specific need, and each properly designed server supports an interface directly related to the realization of one common goal. Client/server models leverage the advantages of commodity-like hardware prices since resource-intensive applications can be designed to run on multiple low-cost systems. Systems can grow since client/server systems scale both horizontally and vertically, meaning that clients can be added with little performance penalty and that extra performance can be extracted from a client/ server system by adding faster server hardware. Despite maintainability issues arising from the distribution of data and code throughout a network and the difficulties vendors have "keeping up" with competing standards, the client/server model is extremely well suited for many applications.

Peer-to-Peer: P2P computing provides an alternative to the traditional client/server architecture. While employing the existing network, servers, and clients infrastructure, P2P offers a computing model that is orthogonal to the client/server model. The two models coexist, intersect, and complement each other.

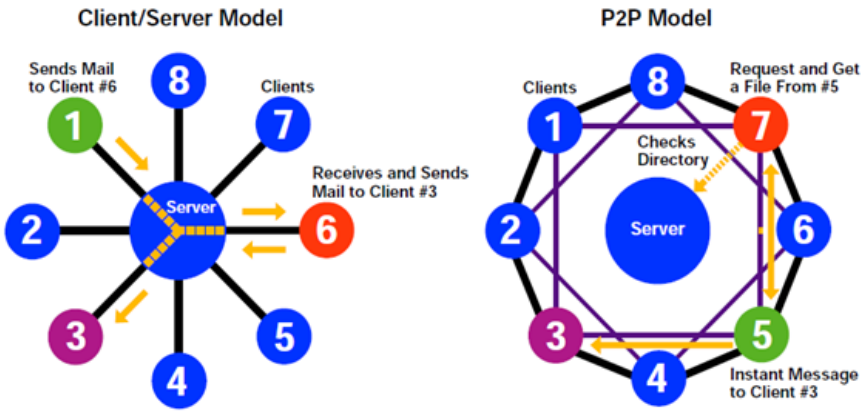

Figure 5. Client Server and P2P Models

In a client/server model, the client makes requests of the server with which it is networked. The server, typically an unattended system, responds to the requests and acts on them. With P2P computing, each participating computer, referred to as peer, functions as a client with a layer of server functionality. This allows the peer to act both as a client and as a server within the context of a given application. P2P applications build on such functions as storage, computations, messaging, security, and file distribution, when handled through direct exchanges between peers. A peer can initiate requests, and it can respond to requests from other peers in the network. The ability to make direct exchanges with other users liberates P2P users from the traditional dependence on central servers. Users have a higher degree of autonomy and control over the services they utilize. One of the greatest benefits of P2P computing is community. P2P makes it possible for users to organize themselves into ad hoc groups that can efficiently and securely fulfill requests, share resources, collaborate, and communicate. P2P allows the elimination of the single-source bottleneck. P2P can be used to distribute data and control and load balance requests across the Net. In addition to helping optimize performance, the P2P mechanism also may be used to eliminate the risk of a single point of failure.

When P2P is used within the enterprise, it may be able to replace some costly data center functions with distributed services between clients. Storage, for data retrieval and backup, can be placed on clients. In addition, the P2P infrastructure allows direct access and shared space, and this can enable remote maintenance capability. Much of the wide appeal of P2P is due to social and psychological factors. For example, users can easily form their own autonomous online communities at the edge of the Net, and run them as they collectively choose. Many of these P2P communities will be ever changing and dynamic in that users can come and go, or be active or not. Other users will enjoy the ability to bypass centralized control. P2P makes users autonomous.

Benefits of P2P Computing

- *e-Business:* P2P can add new capabilities, including connecting and enabling the links of a supply chain, distributing information, content, or software more effectively, and keeping information items on their original node with a central directory or a search capability.
- *Gaming:* A P2P infrastructure provides a natural foundation for the development of online community games that are not centrally controlled. Developers can focus on game features instead of the interface to the communications protocol.
- *Search engines:* Fresh, up-to-date information can be found by searching directly across the space where the desired item is likely to reside.
- *Virus protection:* Relationships among the nodes on the P2P community allow for collaboration in virus detection and warning, as well as automatic quarantining of the community against further attacks.
- *Edge services:* There are instances when it is desirable to place the data, prior to its use, closer to the client requesting it. Online training modules that contain video segments, for example, provide the desired effect when the large data files are located close to the online trainee. Multiple clients offering storage space can provide more flexible and reliable service compared to a server.
- *Collaborative development:* The scope can range from developing software products to composing a document to applications like rendering graphics.

P2P Applications

- File sharing (Napster, Gnutella, Kazza)
- Multiplayer games (Unreal Tournament, DOOM)
- Collaborative applications (ICQ, share whiteboard)
- Distributed computation (Seti@home)
- Ad-hoc networks

Network Computing Challenges

- *Network Availability:* Commercial network availability will affect those with remote network computers and mobile network computers who wish to interoperate over the network at an acceptable speed. Reliability of the network is equally important. Both are essential to effective network computing which will only be viable if connection can be guaranteed and software and data can be transferred at acceptable speed. Fortunately, the telecommunications industry has made significant strides in answering this need and the day of unlimited bandwidth and highly reliable networks is within sight.

- *User Interface:* Web browsers probably have all the functionality required for network computing today. The ideal in the future is a combination of the best features from the browser software that leads the market. In terms of the user interface, key advances in browser technology have included scripting and remote execution, specifically support for ECMA script and Java. These are still evolving in significant ways to improve usability. However, the functions available today are sufficient for business applications.

- *Server Capability:* This covers the range of tasks to be performed by the server in support of the network computer. These tasks now include building and maintaining information about the user environment. If the applications are to be customized for each user or group of users, the server will

have to respond to user requirements by delivering the right subsets of software and data according to need. Although a configuration issue, there may be implications for developers and systems management. This implies a re-engineering of systems in parallel with handling the need to support Java and browser-based applications with the object of identifying and delivering the right functionally related subsets.

- *Software Applications:* In the short term especially, one of the most critical challenges will be the availability of appropriate business software including Java-based applications. There is a good deal of evidence that suppliers are working to fill the gap. This is reported in a Zona Research Paper covering the deployment of Java-based applications, and reflected in the recent announcement by many organizations to develop Java versions of their software. Organizations requiring custom-built software will have to acquire the skills to integrate standard object-based applications to meet their business needs.

2.4 Internet Computing

The Internet is the massive network of networks connects millions of computers together worldwide, forming a network in which any computer can communicate with any other computer provided that they are both connected to the Internet. The World Wide Web (WWW), or simply Web, is a way of accessing information over the medium of the Internet. WWW consists of billions of web pages, spread across thousands and thousands of servers all over the world. It is an information-sharing model that is built on top of the Internet. The most well-known example of a distributed system is the collection of web servers. Hypertext is a document containing words that bond to other documents in the Web. These words are known as links and are selectable by the user. A single hypertext document can hold links to many documents.

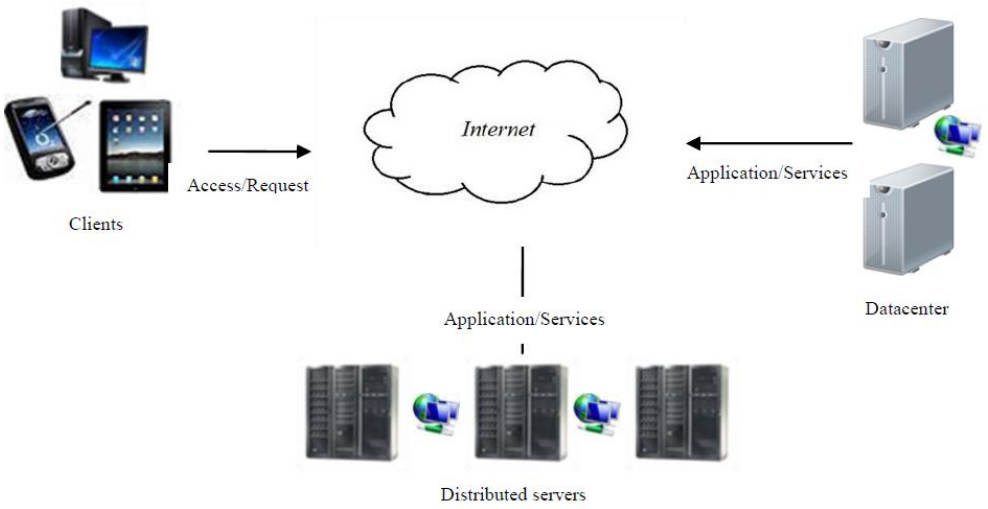

Figure 6. High Level Architecture of Internet Computing

The backbone of WWW are its files, called pages or Web pages, containing information and links to resources - both text and multimedia - throughout the Internet. Internet protocols are sets of rules that allow for inter-machine communication on the Internet. HTTP (Hyper Text Transfer Protocol) transmits hypertext over networks. This is the protocol of the Web. Simple Mail Transport Protocol or SMTP distributes e-mail messages and attached files to one or more electronic mailboxes. VoIP (Voice over Internet Protocol) allows delivery of voice communications over IP networks, for example, phone calls. A web server accepts HTTP requests from clients, and serving them HTTP responses along with optional data contents such as web pages. The operation of the web relies primarily on hypertext as its means of information retrieval. Web pages can be created by user activity. Creating hypertext for the Web is accomplished by creating documents with a language called hypertext markup language, or HTML. With HTML, tags are placed within the text to achieve document formatting, visual features such as font size, italics and bold, and the creation of hypertext links. Servers implementing the HTTP protocol jointly provide the distributed database of hypertext and multimedia documents. The clients access the web through the browser software installed on their system.

The URL (uniform resource locator) indicates the internet address of a file stored on a host computer, or server, connected to the internet.

URLs are translated into numeric addresses using the domain name system (DNS). The DNS is a worldwide system of servers that stores location pointers to web sites. The numeric address, called the IP (Internet Protocol) address, is actually the "real" URL. Once the translation is made by the DNS, the browser can contact the web server and ask for a specific file located on its site. Web browsers use the URL to retrieve the file from the server. Then the file is downloaded to the user's computer, or client, and displayed on the monitor connected to the machine. Due to this correlation between clients and servers, the web is a client-server network. The web is used by millions of people every day for different purposes including email, reading news, downloading music, online shopping or simply accessing information about anything. In fact, the web symbolizes a massive distributed system that materializes as a single resource to the user accessible at the click of a button. In order for the web to be accessible to anyone, some agreed-upon standards must be pursued in the formation and delivery of its content. An organization leading the efforts to standardize the web is the World Wide Web (W3C) Consortium. Due to the ease of connecting, the Internet has expanded in every possible way. The user base has grown from a handful to over a billion, which has increased the business opportunities on it. This has in turn led to the situation where services offered on the Internet have grown rapidly, both in quality and quantity. As an example, the popular Internet search engines serve millions of search queries every day and popular Internet shops have tens of thousands of visitors browsing through their web sites daily. present day, the number of people surfing on the Web has grown from a handful to hundreds of millions. The main reasons for this massive growth are the online services and the ease of connecting to the web.

2.4 Grid Computing

The aim of Grid computing is to enable coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. An infinite number of computing devices ranging from high performance systems such as supercomputers and clusters, to specialized systems such as visualization devices, storage systems, and scientific instruments, are logically coupled together in a Grid and presented as a single unified resource to the user. Figure 3 shows that a Grid user can easily use these globally distributed Grid resources by interacting with a Grid resource broker. Basically, a Grid user perceives the Grid as a single huge virtual computer that provides immense computing capabilities, identical to an Internet user who views the World Wide Web as a unified source of content.

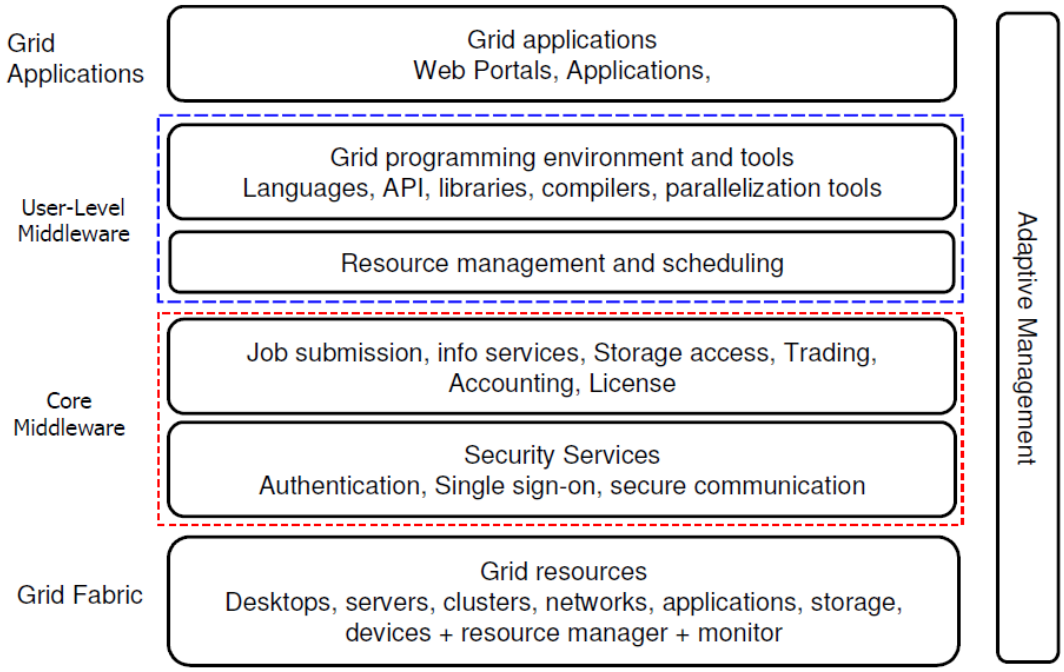

Figure 7. Grid Computing Architecture

A diverse range of applications are currently or soon to be employed on Grids, some of which include: aircraft engine diagnostics, earthquake engineering, virtual observatory, bioinformatics, drug discovery, digital image analysis, high energy physics, astrophysics, and multi-player gaming . Grids can be primarily classified into the following types, depending on the nature of their emphasis.

- Computational Grid: Aggregates the computational power of globally distributed computers (e.g. TeraGrid, ChinaGrid, and APACGrid).
- Data Grid: Emphasizes on a global-scale management of data to provide data access, integration, and processing through distributed data repositories (e.g. LHCGrid and GriPhyN).
- Application Service Provisioning (ASP) Grid: Focuses on providing access to remote applications, modules, and libraries hosted on data centers or Computational Grids (e.g. NetSolve/GridSolve).
- Interaction Grid: Focuses on interaction and collaborative visualization between participants (e.g. Access Grid).
- Knowledge Grid: Aims towards knowledge acquisition, processing, management, and provide business analytics services driven by integrated data mining services (e.g., Italian Knowledge Grid and EU Data Mining Grid).

Utility Grid: Focuses on providing all the Grid services including compute power, data, and services to end-users as IT utilities on a subscription basis and the infrastructure necessary for negotiation of required Quality of Service (QoS), establishment and management of contracts, and allocation of resources to meet competing demands from multiple users and applications (e.g. Gridbus and Utility Data Center). The components that are necessary to form a Grid are shown in Figure 5. The layered Grid architecture organizes various grid capabilities and components such that high level services are built using lower-level services. Grid economy is essential for achieving adaptive management and utility-based resource allocation and thus influences various layers of the architecture which are given below.

- Grid fabric software layer: Provides resource management and execution environment at local Grid resources. These local Grid resources can be computers (e.g. desktops, servers, or clusters) running a variety of operating systems (e.g. UNIX or Windows), storage devices, and special devices such as a radio telescope or heat sensor. As these resources are administered by different local resource managers and monitoring mechanisms, there needs to be Grid middleware that can interact with them.

- Core Grid middleware layer: Provides Grid infrastructure and essential services which consists of information services, storage access, trading, accounting, payment, and security. As a Grid environment is highly dynamic where the location and availability of services are constantly changing, information services provide the means for registering and obtaining information about Grid resources, services, and status. Resource trading based on the computational economy approach is suitable given the complex and decentralized manner of Grids. This approach provides incentives for both resource providers and users to be part of the Grid community, and allows them to develop strategies to maximize their objectives. Security services are also critical to address the confidentiality, integrity, authentication, and accountability issues for accessing resources across diverse systems that are autonomously administered.

- User-level middleware layer: Provides programming frameworks and policies for various types of applications, and resource brokers to select appropriate and specific resources for different applications. The Grid programming environment and tools should support common programming languages (e.g. C, C++, Fortran, and Java), a variety of programming paradigms (e.g. message passing and Distributed Shared Memory (DSM), and a suite of numerical and commonly used libraries. Resource management and scheduling should be transparent to the users such that processor time, memory, network, storage, and other resources in Grids can be utilized and managed effectively and efficiently using middleware such as resource brokers.

- Grid applications Layer: Enables end-users to utilize Grid services. Grid applications thus need to focus on usability issues so that end-users can find them intuitive and easy to use. They should also be able to function on a variety of platforms and operating systems so that users can easily access them. Therefore, an increasingly number of web portals are being built since they allow users to ubiquitously access any resource from anywhere over any platform at any time.

The Grid resource broker comprises the following components:

- Job control agent: Ensures persistency of jobs by coordinating with schedule advisor for schedule generation, handling actual creation of jobs, maintaining job status, and interacting with users, schedule advisor, and deployment agent.
- Grid explorer: Interacts with Grid information service to discover and identify resources and their current status.
- Schedule advisor: Discovers Grid resources using the Grid explorer, and select suitable Grid resources and assign jobs to them (schedule generation) to meet users' requirements.
- Trade manager: Accesses market directory services for service negotiation and trading with GSPs based on resource selection algorithm of schedule advisor.
- Deployment agent: Activates task execution on the selected resource according to schedule advisor's instruction and periodically updates the status of task execution to job control agent.

Traditional core Grid middleware focuses on providing infrastructure services for secure and uniform access to distributed resources. Supported features include security, single sign-on, remote process management, storage access, data management, and information services. An example of such middleware is the Globus toolkit which is a widely adopted Grid technology in the Grid community. Utility Grids require additional service-driven Grid middleware infrastructure that includes:

- Grid market directory: Allows GSPs to publish their services so as to inform and attract users.
- Trade server: Negotiates with Grid resource broker based on pricing algorithms set by the GSP and sells access to resources by recording resource usage details and billing the users based on the agreed pricing policy.
- Pricing algorithms: Specifies prices to be charged to users based on the GSP's objectives, such as maximizing profit or resource utilization at varying time and for different users.
- Accounting and charging: Records resource usage and bills the users based on the agreed terms negotiated between Grid resource broker and trade server.

Advantages of Grid Computing

- Transparent and instantaneous access to geographically distributed and heterogeneous resources.
- Improved productivity with reduced processing time.
- Provisioning of extra resources to solve problems that were previously unsolvable due to the lack of resources.
- A more resilient infrastructure with on-demand aggregation of resources at multiple sites to meet unforeseen resource demand.
- Seamless computing power achieved by exploiting under-utilized or unused resources that are otherwise wasted.
- Maximum utilization of computing facilities to justify IT capital investments.
- Coordinated resource sharing and problem solving through virtual organizations that facilitates
- Collaboration across physically dispersed departments and organizations.
- Service Level Agreement (SLA) based resource allocation to meet QoS requirements.
- Reduced administration effort with integration of resources as compared to managing multiple standalone systems.

3. CLOUD COMPUTING

There are a number of service offerings and implementation models under the cloud computing umbrella. The NIST definition of cloud computing defines three delivery models

- Software-as-a-Service (SaaS)
- Platform-as-a-Service (PaaS)
- Infrastructure-as-a-Service (IaaS)

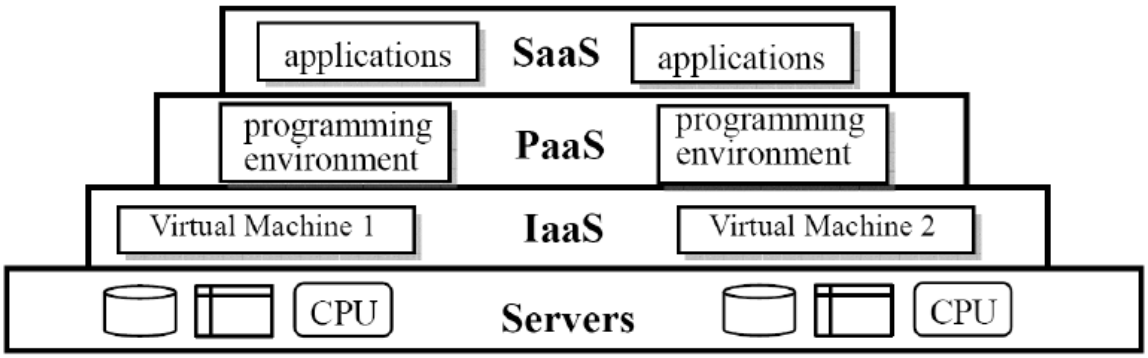

Figure 8. Different Layers of Cloud Computing

3.1 Delivery Models for Cloud Computing

Cloud computing providers can offer services at different layers of the resource stack, simulating the functions performed by applications, operating systems, or physical hardware.

- Software as a Service (SaaS): It offers finished applications that end users can access through a thin client (typically, but not necessarily, a web browser). Prominent examples of SaaS include Gmail, Google Docs, and Salesforce.com. The end user does not exercise any control over the design of the application (aside from some minor customization and configuration options), servers, networking, and storage infrastructure.
- Platform as a Service (PaaS): It offers an operating system as well as suites of programming languages and software development tools that customers can use to develop their own applications. Prominent examples include Microsoft Windows Azure and Google App Engine. PaaS gives end users control over application design, but does not give them control over the physical infrastructure.
- Infrastructure as a Service (IaaS): It offers end users direct access to processing, storage and other computing resources and allows them to configure those resources and run operating systems and software on them as they see fit. Examples of IaaS include Amazon Elastic Compute Cloud (EC2), Rackspace, and IBM Computing on Demand.

3.2 Service Deployment Models

The National Institute of standards and technology (NIST) definition defines four deployment models: Private cloud, Community cloud, Public cloud and Hybrid cloud.

- Private Cloud: this cloud infrastructure exclusively used by a single organisation with multiple business units. It may be managed, operated and owned by the organisation or a third party or some combination of them.
- Community Cloud: This cloud infrastructure is shared by several organizations. It supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them.
- Public Cloud: This cloud infrastructure is available to the general public or a large industry group and it may be owned, managed, and operated by a business, academic, or government organization, or some combination of them.
- Hybrid cloud: it is the combination of two or more clouds (public, private or community). It is remain a unique entity but is bound together.

3.3 Services offered by cloud

There are many services that can be delivered using cloud computing, through advantage of the distributed cloud model.

- Hosted Desktops: A hosted desktop behave and looks like a regular desktop PC, but the software and data customers use are housed in remote, highly secure data centers, rather than on their own machines.

- Hosted Email: Emails is stored centrally on managed servers, providing redundancy and fast connectivity from any location. This allows users to access their email, calendar, contacts and shared files by a variety of means.
- Voice over IP (Hosted Telephony): Is a means of carrying phone calls and services across digital internet networks.
- Cloud Storage: It is the delivery of data storage as a service, from a third party provider, with access via the internet and billing calculated on capacity used in a certain period
- Dynamic Servers: A provider like Think Grid gives its customer access to resources that look and feel exactly like a dedicated server, but that are fully scalable.

3.4 Present market status

Cloud computing continues to gain more mainstream adoption as more companies move into the cloud. Market growth in different type of cloud service models are given below. SaaS Market: There CRM, Financial Planning, Human Resources, Word processing type of applications are offered by the providers. E-mail services also come in SaaS category. Some SaaS service Providers are given below.

- Salesforce.com: Provide call center, incident management, complaint tracking, and service portal services.
- Email cloud: Email cloud is a premium email messaging service provider. This services save user time, bandwidth and the hassle of dealing with email messaging issues such as spam, viruses, downtime and relaying services.
- Google Apps: is a service from Google providing independently customizable versions of several Google products under a custom domain name. It features several Web applications with similar functionality to traditional office suites, including: Gmail, Google Groups, Google Calendar, Talk, Docs and Sites. up to march, 2010 25 million people had "switched to Google Apps" According to Google blogs.

PaaS market: There application design, application development, testing, deployment and hosting as well as application services such as team collaboration, web service integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation and developer community facilitation type of services provided. Some PaaS service Providers are given below.

- Google AppEngine: Google AppEngine is targeted exclusively at traditional web applications, enforcing an application structure of clean separation between a stateless computation tier and a stateful storage tier. AppEngine’s impressive automatic scaling and high availability mechanisms and the proprietary MegaStore data storage available to AppEngine applications, all rely on these constraints.
- Microsoft’s Azure are: It is written using the .NET libraries, and compiled to the Common Language Runtime, a language-independent managed environment. Thus, Azure is intermediate between application frameworks like AppEngine and hardware virtual machines like EC2 is the provider of cloud computing as PaaS.
- Rackspace: The Rackspace Cloud was originally launched as Mosso LLC, a wholly owned subsidiary startup billed as a utility computing offering. The Rackspace Cloud is a web application hosting/cloud platform provider ("Cloud Sites") that bills on a utility computing basis. It has since branched out into cloud storage ("Cloud Files"). It was one of the first commercial cloud computing services.
- AWS: S3: vCloud is a cloud computing initiative from VMware which will allow customers to migrate work on demand from their "internal cloud" of cooperating VMware hypervisors to a remote cloud of VMware hypervisors. The goal of the initiative is to provide the power of cloud computing with the flexibility allowed by virtualization.

IaaS market: The service offers multiple options for computing, memory, network configuration, operating system and Disaster Recovery (DR) to fit client’s specific needs. Some IaaS service Providers are given below.

- Flexiscale: FlexiScale is a complete rebuild of Europe’s first cloud computing platform using Flexiant’s revolutionary Extility technology.

- AWS: EC2 Amazon Elastic Compute Cloud delivers scalable, pay-as-you-go compute capacity in the cloud. Amazon EC2 is at one end of the spectrum. An EC2 instance looks much like physical hardware, and users can control nearly the entire software stack, from the kernel upwards.
- NetMagic Solutions : These provider team monitors all critical parameters related to the performance of the network and the servers being hosted.
- InstaCompute (from Tata Communications): It provides pre- or-post-paid payment options. Multiple processing power options. Private or Public Connectivity. InstaCompute changes with our changing requirements.

4. CLOUD COMPUTING ENABLING TECHNOLOGIES

- Virtualization
- Web 2.0
- World-wide distributed storage system
- Distributed Computing
- Grid Computing
- Utility Computing
- Network Bandwidth & Latency
- Fault-Tolerant Systems
- Programming Models

A number of enabling technologies contribute to Cloud computing which are described below in details here.

Virtualization technology: Virtualization technologies partition hardware and thus provide flexible and scalable computing platforms. Virtual machine techniques, such as VMware and Xen offer virtualized IT-infrastructures on demand. Virtual network advances, such as VPN support users with a customized network environment to access Cloud resources. Virtualization techniques are the bases of the Cloud computing, since they render flexible and scalable hardware services. Hardware virtualization is a technology that organizations are widely adopting to enable better utilization for available computing resources. Virtualization is accomplished by offices/departments. Implementers will need to learn how to use resources and services of selected provider(s). Developers should learn provider's APIs to allow their applications to dynamically scale up or down their usage in accordance with actual needs. Administrators should know how to manage and monitor used services. In practice, first implantation will come out with a list of lessons learned that can be usefully applied to future projects. This strategy will help organizations get hands-on experience as well as minimize risks associated with the decision to adopt a new technology

Web 2.0: Web 2.0 is an emerging technology describing the innovative trends of using World Wide Web technology and Web design that aims to enhance creativity, information sharing, collaboration and functionality. The essential idea behind Web 2.0 is to improve the interconnectivity and interactivity of Web applications. The new paradigm to develop and access Web applications enables users access the Web more easily and efficiently. Cloud computing services in nature are Web applications which render desirable computing services on demand. It is thus a natural technical evolution that the Cloud computing adopts the Web 2.0 technique. The last 10 years has seen many advances in web technologies. Innovations included different data formats and forms of accessibility to information available on the Internet such as RSS, Blogs, Portals, Wikis, XML, Web Services etc,. These techniques helped organizations to offer their information as sets of services that allow others to easily access them to mix and match underlying functionalities in their own websites/applications.

Web service and SOA: Computing Cloud services are normally exposed as Web services, which follow the industry standards such as WSDL,33, SOAP28) and UDDI.26). The services organization and orchestration inside Clouds could be managed in a Service Oriented Architecture (SOA). A set of Cloud services furthermore could be used in a SOA application environment, thus making them available on various distributed platforms and could be further accessed across the Internet.

The idea of SOA is to turn functionalities of both existing and new applications into a set of granular components. SOA has encouraged software vendors to offer their products as services that clients can use/reuse and compose together to fulfill business requirements in an agile manner. This agility applies to cloud computing as well making it easier to access available hardware and software resources. *World-wide distributed storage system*: A network storage system, which is backed by distributed storage providers (e.g., data centers), offers storage capacity for users to lease. The data storage could be migrated, merged, and managed transparently to end users for whatever data formats. Examples are Google File System11) and Amazon S3.16) A Mashup13) is a Web application that combines data from more than one source into a

single integrated storage tool. The SmugMug29) is an example of Mashup, which is a digital photo sharing Web site, allowing the upload of an unlimited number of photos for all account types, providing a published API which allows programmers to create new functionality, and supporting XML-based RSS and Atom feeds.

Distributed Computing: A distributed data system which provides data sources accessed in a semantic way. Users could locate data sources in a large distributed environment by the logical name instead of physical locations. Virtual Data System (VDS) is good reference.

Grid Computing: A distributed computing model that tends to gather underutilized computing resources available in organizations to process computing-intensive tasks faster. This model has given organizations like Amazon the idea to lease unused resources (both processing units and storage) to clients in need of them.

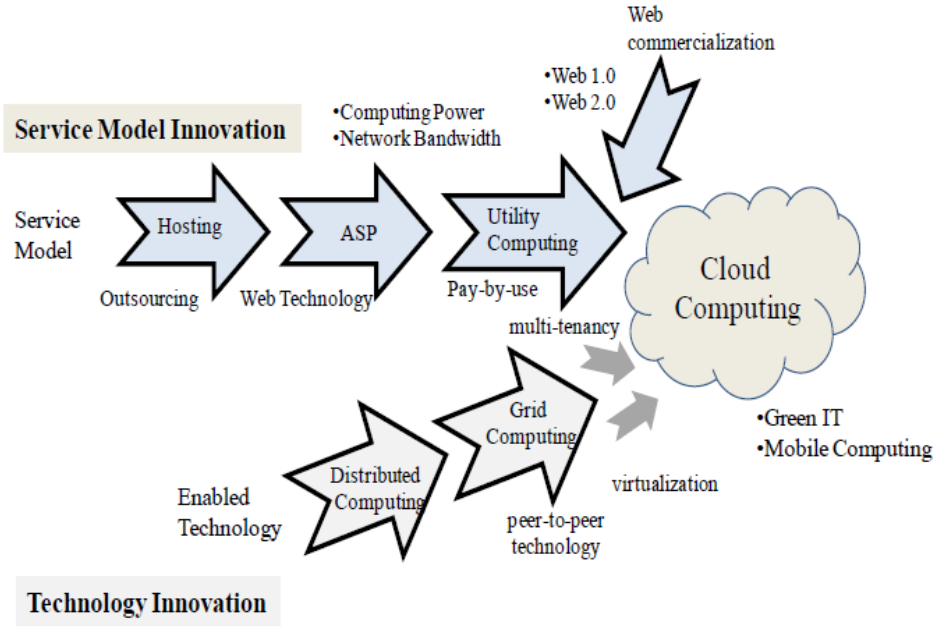

Figure 10. Evolution of Cloud Computing

Utility Computing: Utility computing represents the desire to have IT acquired, delivered, used, paid for, and managed in a manner similar to how we use other commoditized utilities such as electricity, telephone service, cable television, etc. The principal appeal of utility computing lies in the systematized framework it could create for the interaction between providers and consumers of IT resources. One of the foundational features of a utility is accounting the ability of both the provider and the consumer to accurately measure the usage of the commodity being exchanged. Accurate resource accounting is desirable for a variety of important reasons, ranging from billing and auditing to effective resource allocation, anomaly detection and resolution.

Network Bandwidth & Latency: Bandwidth, sometimes known as throughput, is often defined as the rate at which data is transmitted over a network. It is described in bits per second and represents the capacity of the connection. The higher the capacity, the better the performance subject to other factors such as latency. Cloud computing service providers calculate customer bandwidth requirements by taking into account the amount of available bandwidth and the average bandwidth utilization required by different kinds of applications. They also take into consideration latencies in transmission while computing the time it will take to upload the initial backup and all subsequent backups. Online backup and cloud service providers try to optimize Internet bandwidth in a number of ways. They try to reduce the amount of data that flows through the pipes. They may use link load balancing technologies or unique binary patching and incremental backup technology to extract and transmit only changes to files in order to balance / reduce the amount of data that is transmitted. De-duplication and compression techniques may be used to further reduce the size of files that are being transmitted across the network. However, bandwidth and latency are interconnected. The speed of the network is a function of bandwidth and latency. While bandwidth can be increased, latency cannot be reduced drastically.

Fault-Tolerant Systems: A fault-tolerant system may be able to tolerate one or more fault-types including transient, intermittent or permanent hardware faults, software and hardware design errors, operator errors, or externally induced upsets or physical damage. An extensive methodology has been developed in this field over the past thirty years, and a number of fault-tolerant machines have been developed and most dealing with random hardware faults, while a smaller number deal with software, design and operator faults to varying degrees. Fault tolerance is a major concern to guarantee availability and reliability of critical services as well as application execution. In order to minimize failure impact on the system and application execution, failures should be anticipated and proactively handled. Fault tolerance techniques are used to predict these failures and take an appropriate action before failures actually occur.

Programming models: Users drive into the computing Cloud with data and applications. Some Cloud programming models should be proposed for users to adapt to the Cloud infrastructure. For the simplicity and easy access of Cloud services, the Cloud programming model, however, should not be too complex or too innovative for end users. The MapReduce is a programming model and an associated implementation for processing and generating large data sets across the Google worldwide infrastructures. The MapReduce model firstly involves applying a "map" operation to some data records – a set of key/value pairs, and then processes a "reduce" operation to all the values that shared the same key. The Map-Reduce-Merge35) method evolves the MapReduce paradigm by adding a "merge" operation. Hadoop25) is a framework for running applications on large clusters built of commodity hardware. It implements the MapReduce paradigm and provides a distributed file system – the Hadoop Distributed File System. The MapReduce and the Hadoop are adopted by recently created international Cloud computing project of Yahoo!, Intel and HP.

CONCLUSION AND FUTURE DIRECTIONS

This paper presented the evolution of computational models, definition, characteristics, architecture and enabling technologies. It also presents essential terms related to cloud computing with the aim to answer questions frequently asked by people who are in the computer field. In this research paper we also described the advantages, disadvantages of these models with respect to to both providers and clients, challenges to adopt. Finally we have more focused on cloud computing and its basics. In feature research, we will pay much attention on future aspects of cloud computing like how we make it secure, fast, real time access and what add-on make it more productive.

REFERENCES

[1] K. Lyytinen and G. M. Rose, "The Disruptive Nature of Information Technology Innovations: The Case of Internet Computing in Systems Development Organizations", MIS Quarterly, 2003, pp. 557-595.
[2] N. Melville, K. Kraemer, and V. Gurbaxani, "Review: Information Technology and Organization Performance: an Integrative Model of IT Business Value", MIS Quarterly, 2004, pp. 283-322.
[3] T. Ravichandran and C. Lertwongsatien, "Effect of Information Systems Resources and Capabilities on Firm Performance: A Resource-Based Perspective", JMIS, 2005, pp. 237-276.
[4] M. Benaroch, S. Shah, and M. Jeffery, "On the Valuation of Multistage Information Technology Investments Embedding Nested Real Options", JMIS, 2006, pp. 239-261.
[5] P. E. D. Love and Z. Irani, "An Exploratory Study of Information Technology Evaluation and Benefits Management Practices of SMEs in the Construction Industry", 2004, pp. 227-242.
[6] L. Motiwalla, M. R. Khan, and S. Xu, "An Intra- and Inter-Industry Analysis of E-business Effectiveness", I&M, 2005, pp. 651-667.
[7] J. F. Fairbank, G. Labianica, H. K. Steensma, and R. Metters, "Information Processing Design, Choices, Strategy, and Risk Management Performance", JMIS, 2006, pp. 293-319.
[8] T. A. Byrd, B. R. Lewis, and R. W. Bryan, "The Leveraging Influence of Strategic Alignment on IT Investment: An Empirical Examination", Information and Management, 2006, pp. 308-321.
[9] R. K. Yin, Case Study Research, Design and Methods, 2nd Edition, Sage Publications, CA, 1994.
[10] G. McCulloch, Documentary Research in Education, History and the Social Sciences, Routledge Falmer, London, 2004.
[11] M. Callon, "Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of St. Brieuc Bay", in: Law, J. (ed.), Power, Action and Belief, Routledge and Kegan Paul, London, 1986, pp.197–233
[12] Pueschel, T. and D. Neumann (2009) Management of Cloud Infastructures: Policy-Based Revenue Optimization. ICIS 2009 Proceedings, Paper 178.
[13] Armbrust, M., et al., Above the Clouds: A Berkeley View of Cloud Computing. 2009, EECS Department, University of California, Berkeley.
[14] Lakshmanan, Relevance to Enterprise. Cloud Computing Journal, 2009.
[15] Cattaneo, E.R., Time Sharing Seminar in print. Data Processing Magazine, 1965. Sept(7): p. 18.
[16] Bell, G., Fundamentals of Time Shared Computers. Computer Design, 1968. 1(1).

[17] Ross, V., Factors influencing the adoption of cloud computing by decision making managers. 2010, Capella University: United States Minnesota. p. 97.
[18] Mei, L., Z. Zhang, and W.K. Chan, A tale of clouds: Paradigm comparisons and some thoughts on research issues, in Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference. 2008. p.464-469.
[19] Aymerich, F.M., G. Fenu, and S. Surcis (2008) An approach to a cloud computing network Proceedings of the First International Conference on the Applications of Digital Information and Web Technologies, 113-118.
[20] Wang, S., N.P. Archer, and W. Zheng, An exploratory study of electronic marketplace adoption: A multiple perspective view. Electronic Markets, 2006. 16(6): p. 337-348.
[21] Youseff, L., M. Butrico, and D. Da Silva, Toward a unified ontology of cloud computing, in Proceedings of the Grid Computing Environments Workshop. 2008. p. 1-10.
[22] Leavitt, N., Is cloud computing really ready for prime time Computer, 2009. 42(1): p. 15-20.
[23] NIST (2009) National Institute of Standards and Technology, Cloud Computing. Information Technology Laboratory.
[24] Geelan, J. (2009) Twenty-one experts define cloud computing. Cloud Computing Journal
[25] Vaquero, L.M., et al., A break in the clouds: towards a cloud definition. SIGCOMM Computer Communications, 2008. 39(1): p. 50-55.
[26] Staten, J., et al. (2008) Is cloud computing ready for the enterprise. Forrester Research.
[27] Christensen, C., The Innovator's Dilemma. Harper Business Essentials. 2003, New York: HarperCollins Publishers Inc.
[28] Weiss, A., Computing in the clouds. NetWorker, 2007. 11(4): p. 16-25.
[29] Holden, E., et al., Databases in the cloud: a work in progress, in Conference On Information Technology Education, Proceedings of the 10th ACM conference on SIG-information technology education. 2009. p. 138-143.
[30] Kroeker, K.L., The evolution of virtualization. Commun. ACM, 2009. 52(3): p. 18-20.
[31] Ziegler, J.R., WHAT IS TIME SHARING? Management Review, 1968. 57(4): p. 52.
[32] Buyya, R., C.S. Yeo, and S. Venugopal (2008) Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities Proceedings of the 10th International Conference on High Performance Computing and Communications, 5-13.
[33] Foster, I., et al., Cloud computing and grid computing 360-degree compared, in Proceedings of the 2008 Grid Computing Environments Workshop. 2008. p. 1-10.
[34] Delic, K.A. and M.A. Walker, Emergence of the Academic Computing Clouds. Ubiquity, 2008. 2008(August): p. 1-1.
[35] Hayes, B., Cloud Computing. Communications of the ACM, 2008. 51(7).
[36] Androutsellis-Theotokis, S. and D. Spinellis, A survey of peer-to-peer content distribution technologies. ACM Comput. Surv., 2004. 36(4): p.335-371.
[37] Sprague, R.E., The information Utilities Business Automation, 1965: p.42.
[38] Emerson, M., The Small Computor Versus Time-Shared Systems. Computer and Automation, 1965. XIV(September): p. 19.
[39] Raymond, R.C., USE OF THE TIME-SHARING COMPUTER IN BUSINESS PLANNING AND BUDGETING. Management Science, 1966. 12(8): p. B-363-B-381.
[40] Apache Software Foundation (2010) Apache HTTP Server 2.2 Official Documentation—Volume I. Server Administration. Fultus Barr J (2010) Host your web site in the cloud: Amazon web services made easy. SitePoint Pty, Melbourne Capra E, Wasserman AI (2008) A framework for evaluating managerial styles in open source projects. In: Proc. 4th int'l conference on open source systems, pp 1–11

BIOGRAPHY OF AUTHORS

Rabi Prasad Padhy is currently working as a Senior Software Engineer - Oracle India Private Ltd. Bangalore, india. He has achieved his MCA degree from Berhampur University. He carries 8 years of extensive IT Experience with MNC's like EDS, Dell, IBM and Oracle. His area of interests include IT Infrastructure Optimization, Virtualization, Enterprise Grid Computing, Cloud Computing and Cloud databases. He has published several research papers in national and international journals. He is a certified professional for Oracle, Microsoft SQL Server database, Enterprise Linux Administration and ITIL certified.
.

Dr. Manas Ranjan Patra holds a Ph.D. Degree in Computer Science from the Central University of Hyderabad, India. Currently he is an Associate Professor in the Post Graduate Department of Computer Science, Berhampur University, India. He has about 24 years of experience in teaching and research in different areas of Computer Science. He had visiting assignment to International Institute for Software Technology, Macao as a United Nations Fellow and for sometime worked as assistant professor in the Institute for Development and Research in Banking Technology, Hyderabad. He has about 90 publications to his credit. His research interests include Service Oriented Computing, Software Engineering, Applications of Data mining and E-Governance. He has presented papers, chaired technical sessions and served in the technical committees of many International conferences.