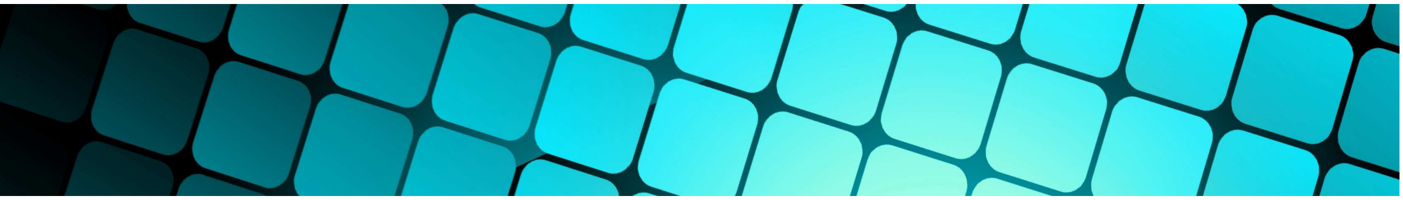
Introduction to Cloud Computing

Embracing a Disruptive Force

by

Jonathan Parri



April 8, 2011

Abstract

CLOUD computing has taken the world by storm. It is a term thrown around, especially in IT circles, from support personnel to the development elite. Here, we look to introduce cloud computing to the uninitiated and focus on current aspects that interest those in the field of hardware/software codesign and speculate on future trends and implementations relevant to those individuals. Cloud computing is a disruptive force affecting users, vendors and corporations as a whole. The idea itself describes the computer as a service and exemplifies offering such services as platforms, software and full infrastructures. Each cloud concept has key software and hardware considerations that affect how and which type of service can be offered while exemplifying critical issues relevant to the field of engineering such as performance, power and utilization. We look to expose and discuss these points and move past the web service curtain.

Contents

Chapter 1

Up up and Away-
The Cloud Computing Trend

1.1 Chapter Overview

Clouds grace our skies and provide an infinite source of wonder and aw. Cloud computing has followed its namesake in catching the undivided attention of IT personnel, developers and IT pundits. The term itself is often confusing; however, there is no denying the disruptive force it is causing within the IT world.

In this chapter, we look to introduce the cloud computing concept and the underlying idea of the computer being offered as service. Furthermore, recent market trends are presented showing the continual growth and the importance of this emerging technology.

1.2 The Cloud

The *cloud* in cloud computing is a new model describing a computation infrastructure that delivers and provides services via ad-hoc resource provisioning. The key idea of the cloud is that the end-user is unaware of the physical location and configuration of the cloud IT environment. The cloud provides a variety of on-demand services in a manner that is easy to understand.

A simple example to illustrate the current cloud idea is the comparison between traditional Microsoft Exchange, a conventional email server product used by large companies, and the Google GMail model shown in Fig. 1.1 and 1.2 respectively.

Microsoft Exchange has become a staple within corporations for email support deployment. Consider, that with such an installation is the requirement of dedicated servers and technical personnel leading to a high-cost and manpower overhead. Such considerations are often beyond the scope of small to medium sized businesses (SME). The Google Apps GMail service alternative provides email support using the existing Google data center at a lower-cost to the end user while provisioning and assigning appropriate resources unbeknown to the service requester or end-user.

The cloud takes shape when a company requires multiple services and is looking to transfer the IT footprint issues to another entity, the cloud. Existing cloud based services cover a multitude of needs from sales-tracking systems and email to project management. Custom cloud applications are becoming common-place with some providers offering easy and abstracted customization tools [1] for clients to add any application imaginable to the available cloud resource pool.

1.3 Offering the Computer as a Service

Offering services from an IT infrastructure is not a new concept. Service-oriented programming came about to introduce a service base unit of computer work. Service-oriented programming led to the concept of service-oriented architecture (SOA) which is widely used set of principles for systems development.

Cloud computing and service-oriented programming share a clear element, offering of a service; however, cloud computing takes this abstraction many levels further. Two guiding principles of service-oriented architectures involve the provisioning and delivery of the service as a static boundary [2]. In cloud computing, the provisioned platform and the delivery mechanism are a black-box and are not required to be known (Fig. 1.3).

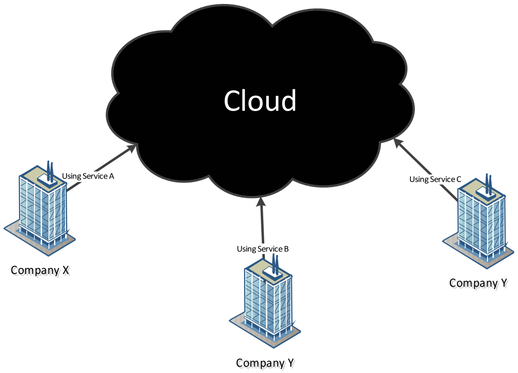


Figure 1.3: Multiple companies using one cloud provider.

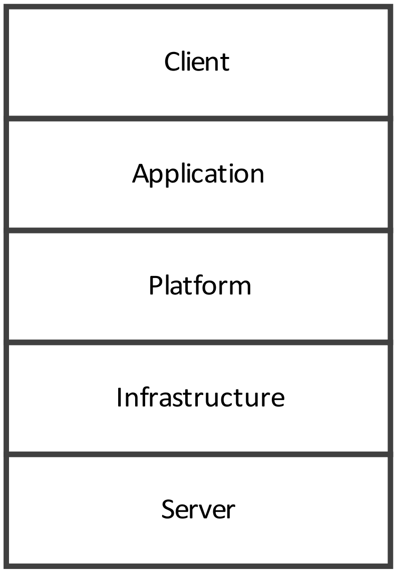


Figure 1.4: Cloud Computing Service-Oriented Layers.

There are multiple layers to the cloud computing paradigm and are illustrated in in Fig. 1.4. Cloud clients consist of hardware and software which rely on cloud computing services for application functionality. Examples of cloud clients include home computers and mobile phones with appropriate software such as a web browser. Cloud servers are high-performance servers on which the cloud services are deployed. Typically, cloud controllers, file servers, application servers and databases will make up this cloud backbone. Remember that the key to cloud computing is to not have the customer concerned with such architecture and implementation ideas. The installed cloud-oriented operating systems deal with such abstractions. We will discuss this further in subsequent sections.

The sandwiched application, platform and infrastructure layers are offered individually depending on the model of the cloud provider with others abstracted. Each service-oriented layer is briefly presented next. Further implications of these service layers are discussed in Chapter 4. An overview of trade-offs is shown in Fig. 1.5.

1.3.1 Software as a Service (SaaS)

The application layer provides software as a service (SaaS). Here, application software is delivered over the Internet without the need for client or customer installation and maintenance. Services at this layer can be considered 3$^{\mathrm{rd}}$ party hosted and maintained applications. SaaS offerings have well-known markets in collaboration suites, content management systems (CMS), change management, service desk management, human resource management (HRM), enter-

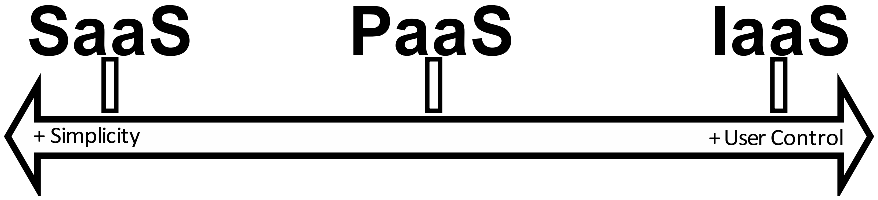


Figure 1.5: Available control and complexity of cloud service offerings.

prise resource planning (ERP) and customer relationship management (CRM) [3].

Examples of current SaaS offerings include:

- Google Apps - *GMail, Google Docs etc...*
- Dropbox - *Web-based file hosting*
- Salesforce - *Enterprise cloud computing targeting Customer Relationship Management*
- Postini - *Email and web security service*

1.3.2 Platform as a Service (PaaS)

Platform as a service (PaaS) in an extension of SaaS as a form of customization and simplified application development. A pre-made solution stack is typically included in a PaaS service offering.

Examples of current PaaS offerings include:

- Azure Services Platform - *Microsoft cloud platform offering Windows Azure operating system, SQL Azure and Azure AppFabric*
- Google App Engine - *Platform for developing and hosting web-applications*

1.3.3 Infrastructure as a Service (IaaS)

Infrastructure as a service (IaaS) brings together a whole customizable IT infrastructure as a complete offered service. IaaS combines computing power, network resources, storage and software elements as a customizable package.

Examples of current IaaS offerings include:

- Amazon Web Services - *Amazon EC2 and Amazon S3 platform for running and storing any software application via rented time*
- Eucalyptus - *Open-source EC2 and S3 compatible platform for non-Amazon hosting*

Chapter 2

Knowing the Difference- HPC, Grid, Cluster or Cloud

2.1 Chapter Overview

There is often confusion when discussing cloud computing, grid computing, clustered computing and high-performance computing. Cloud computing is none of these things but borrows advantageous features from each. In this chapter, we review existing technologies and see how they fit with regards to cloud computing.

2.2 High Performance Computing

High performance computing (HPC) is a well established and research intensive field that uses supercomputers or compute clusters to solve difficult computational problems. Supercomputers are computer systems that are at the forefront of processing performance technology. Examples of well-known super computers include the IBM Blue Gene, IBM Roadrunner and Cray Jaguar. Many supercomputer architectures have evolved from individual systems and into entire groups of interconnected systems called clusters. Compute clusters are collections of computers with a high-speed interconnection network. The networked systems work together within a common problem-space.

HPC is used by a variety of everyday industry markets. Market shares distributed by industry are shown in Fig. 2.1. To understand the scale of these HPC clusters, the number of processing units per HPC are presented in 2.2.

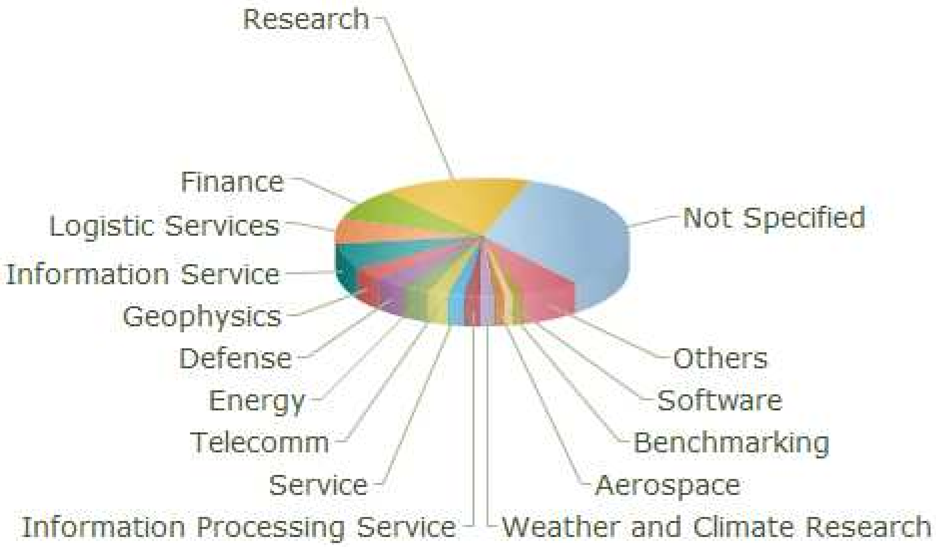


Figure 2.1: HPC application areas (2010)[1].

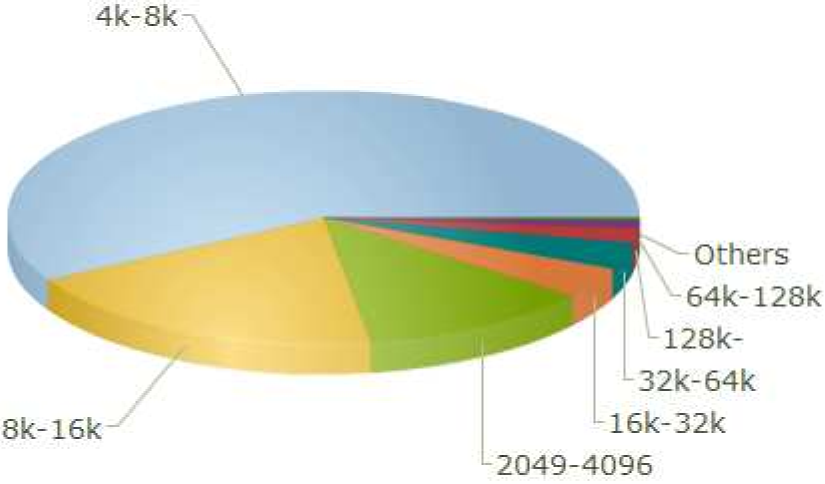


Figure 2.2: Number of processors per HPC system (2010)[1].

[1] Generated from data in [4].

2.2.1 Grid Computing

Grid computing is a further evolution on the idea of compute clusters by creating a distributed virtual supercomputer. Conventional HPC systems are tightly coupled to their other cluster modules through either a high-speed network backbone or switch fabric. Grid computing brings together HPC and distributed computing. In a grid computing environment, a controller system packages portions of the problem-space workload and distributes the pieces to other systems. It is the job of the controller to receive, interpret and package the receipt of individual solution pieces. The idea of grid computing is to utilize a shallow footprint on nodes across a vast computing environment. A widely known example of the grid computing paradigm is the Folding@home[5] project used to perform protein folding simulation, an embarrassingly parallel yet computationally intense computation. Distributed grid nodes run a background application that utilizes unused CPU and GPU power and accept problem segments in the form of work units from a controller system at Stanford University. Work units are further break-downs of different simulation problems such as folding simulations of Alzheimer's disease or sickle-cell anemia.

Three criteria have been established to formulate the grid computing model [6]:

1. **Loosely-Coupled**

   Loosely-coupled systems are made of separate, distinct and autonomous subsystems that have their own resources. Tightly-coupled systems often share system resources such as memory and are connected with a short-distance high-speed network topology or bus.

2. **Geographically Dispersed**

   Grid computing nodes communicate through standard networking protocols and are able to take advantage of asynchronism. This means that nodes do not need to be located in close proximity and do not require a constant synchronized communication mechanism.

3. **Heterogeneous**

   Heterogeneity is easily exploited in grid computer infrastructures. Most application designs allow for a variety of operating system and node hardware architecture possibilities within the entire grid.

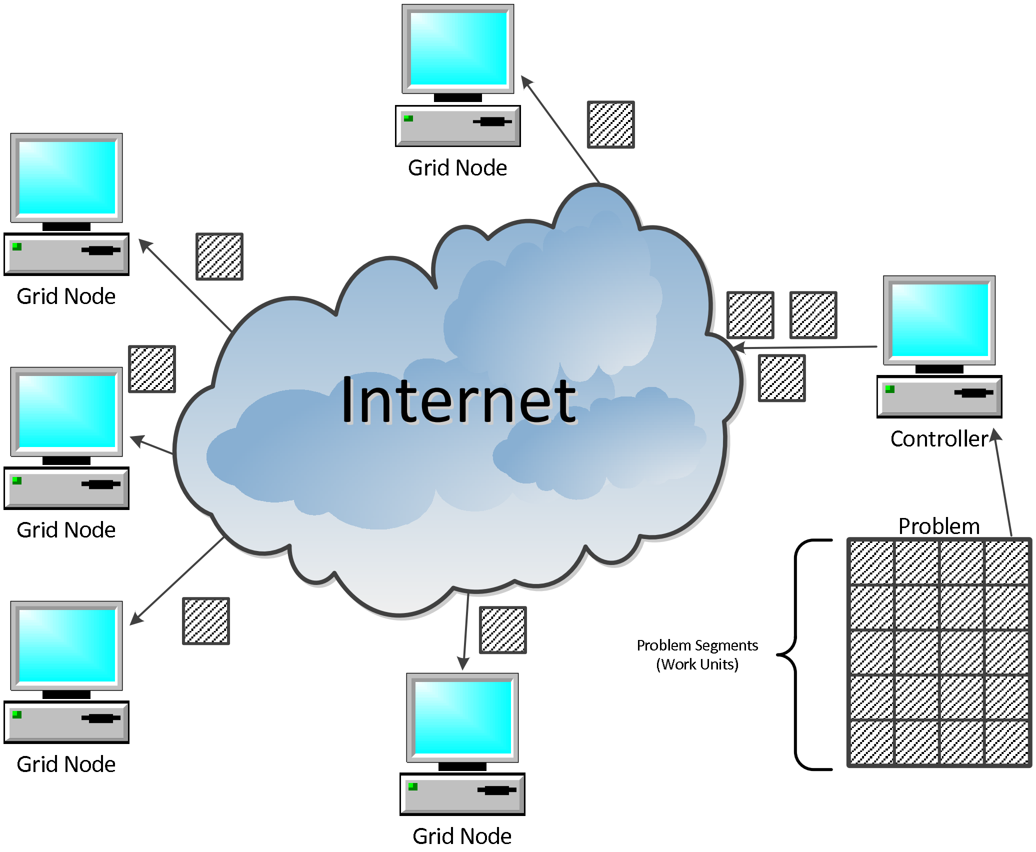


Figure 2.3: Grid computing model.

Fig. 2.3 shows the distributed grid computing model. It is clear that the software aspects of grid computing can be quite difficult. Computational tasks must be broken down into smaller problems with a final solution built up from multiple solution segments. Defining and developing an appropriate software architecture for grid computing is a research intensive area. Task distribution on embarrassingly parallel tasks, such as 3-D transformation are easily dealt with but more complex problems require extensive study. Current grid computing applications have focused on protein folding, weather modeling, oil exploration and vector mechanics. The software controlling the problem execution is a form of middleware. This middleware facilitates the sharing of heterogeneous resources and forms a virtual resource organization.

Due to the size and nature of grid computing systems, scheduling becomes a key consideration. A schedulers job is to tell programs how and when to run while best utilizing available resources. The three phases of a controller's scheduling requirements are defined in [6] as:

1. **Resource Discovery**
   Computing platforms are authenticated along with job requester. Required resources are submitted along with the job designated as the *application definition*. Based on the provided job and requirements, the scheduler attempts to find the first empty execution slot that meets the *application definition*.

2. **System Selection**
   During this phase, information is gathered based on the priority of other jobs and business needs versus resource allocation. Considering this information, the required target nodes for the specified job are selected using a variety of matching techniques (*i.e.*, condor matchmaking or computational economies).

3. **Job Execution**
   The job execution phase entails advanced resource reservation, job submission and cleanup of resources once the job has been completed.

Grid computing offered by a provider is often seen as a SaaS solution with a middleware user-accessible front-end. Such a target market focuses on utility computing. A well known grid computing middleware stack is the Apache licensed *gLite*. gLite was initially designed as part of CERN's Large Hadron Collider project and is now used by ore than 250 computing facilities and 15000 researchers[7] with continual development under the "Enabling Grids for E-SciencE (EGEE)" project.

2.3 Grid Computing Versus Cloud Computing

The current cloud computing model was an evolution of grid computing. Both cloud computing and grid computing are scalable computation platforms. Scalability is exploited through load balancing allowing processing power and networking bandwidth to be provisioned and de-allocated at will. Multienancy, a single application instance serving multiple jobs or users, has become a solid foundation of both computing models.

Grid computing focuses solely on computationally intensive problems whereas cloud computing offers a broader array of uses that can in-fact include computational intensive problems [8] as an application subset. Consider the Amazon EC2 cloud computing service which offers a variety of cloud resources, instances as IaaS. Instances are virtualized servers with matching virtual hardware resources purchased by the client. The available hardware resources span from one processor and 633 MB RAM to larger instances with 34 processors and 18 GB of RAM [9]. A variety of applications from e-commerce to scientific computing have direct niches in the Amazon offerings. Details of individual cloud implementations are discussed later in Chapter 4.

Grid computing leads the pack with respect to scheduling. Cloud computing still lacks the broker/agent aspect of a scheduler [6]. Cloud computing environments currently cannot effectively decide how many resources will be required for a particular job. Continual work is underway to bring this to cloud computing infrastructures.

Chapter 3

The Cloud isn't Real, it's Virtual

3.1 Chapter Overview

Virtualization has created the foundation for the cloud computing movement. In this chapter, we introduce traditional virtualization and move towards its role in cloud computing infrastructures.

3.2 Introducing Virtualization

Virtualization hides and abstracts computing platform resources. The concept of virtualization can be seen in the implementation and development of virtual machines (VM). A virtual machine is an isolated computing platform implemented in software. A virtual machine executes programs the same way as a physical machine but by using virtual resources. There are two types of virtual machines:

1. **System Virtual Machine**
   Full virtualized system support for operating system environment. (*e.g.*, VMWare ESX, VirtualBox, Xen, Microsoft VirtualPC)

2. **Process Virtual Machine**
   Virtualized system support for single application or program. (*e.g.*, Java Virtual Machine, Common Language Runtime, Actionscript Virtual Machine)

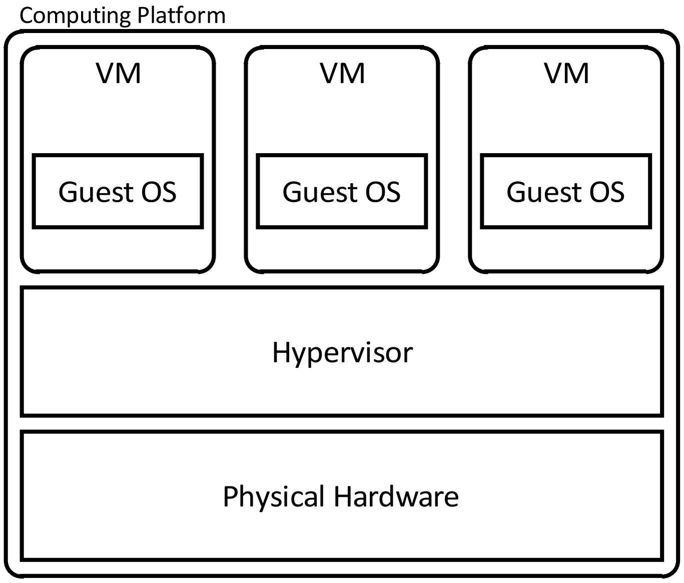


Figure 3.1: Type 1 hypervisor usage on computing platform.

Here, we are mainly concerned with system virtual machines. System virtual machines share the underlying hardware of the host machine and can be used for multiple OS implementations, server virtualization, protected environments or to simplify configuration management needs. Virtual machines are commonly used in server infrastructures to run multiple operating systems with matching applications on the same physical server box. Such a technique improves resource utilization, drastically increases system security through sandboxing and facilitates configuration management (*i.e.*, snap-shots).

A hypervisor provides virtualization control support at the software level. Note that within industry, the terms virtual machine monitor and hypervisor are used interchangeably. It is the job of the hypervisor to present and maintain available virtual computer resource to guest operating systems (Fig. 3.1). The hypervisor must map presented virtual resources to available physical computing resources. The first hypervisor technologies appeared in the 1970's for IBM mainframe systems[10] to share processing, memory and storage IO resources. The base concept for the early IBM hypervisor is still found in today's mainframe hardware. Hypervisor and virtual machine technology began in the server/mainframe industry and worked its way into both desktop and embedded platform markets while tackling different problems.

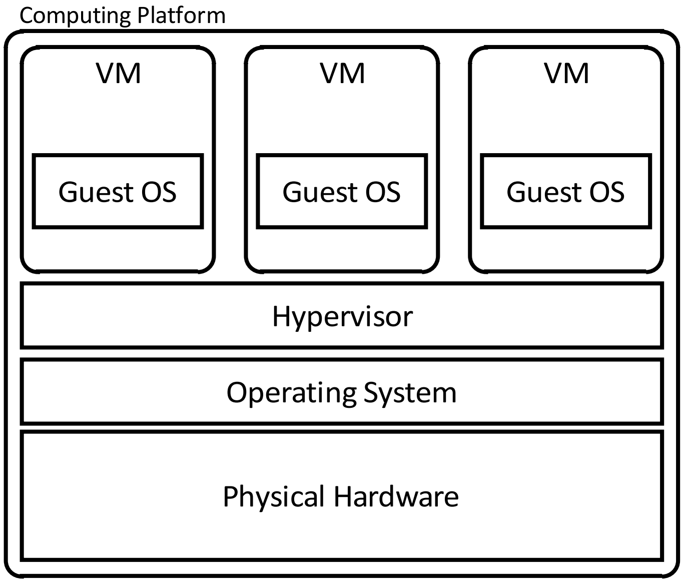


Figure 3.2: Type 2 hypervisor usage on computing platform.

Two categories of been devised to categorize hypervisor technologies [11]:

1. **Type 1 Hypervisor**
   A hypervisor that runs directly above the physical hardware layer as shown in Fig. 3.1. Examples of such implementations include the open-source Xen, VMWare ESX and Microsoft's Hyper-V.

2. **Type 2 Hypervisor**
   A hypervisor which runs within a conventional operating system environment. Oracle's VirtualBox is an example of a popular type 2 hypervisor product used to install guest operating systems on top of a user's main and everyday operating system. Such a model is shown in Fig. 3.2.

The work of [11] was further extended to define key required aspects of virtual machines and their hypervisor implementations. Hypervisor requirements were defined in [12] as:

1. **Equivalence or Fidelity**
   A program running under a hypervisor should exhibit the same behavior when also running directly on the machine

2. **Resource Control of Safety**
   The hypervisor must maintain control of virtualized hardware resources.

3. **Efficiency or Performance**
   The majority of executed machine-level instructions should be able to execute without hypervisor translation or modification.

Most hypervisors adequately address the second requirement of resource control while trying to best cover the first and third. Ensuring performance can be difficult depending on software emulation requirements or native code concerns. Hypervisors that do in-fact address all three concerns have been colloquially called efficient hypervisors.

3.2.1 Supporting Virtualization

Virtual machines and their hypervisors are supported either through hardware virtualization, software emulation techniques or a combination of both.

Initial hypervisors lacked any physical hardware support and relied solely on software emulation techniques. Host systems are required to prevent the guest operating system from directly accessing the processor and retain control themselves. It becomes difficult at this point to trap on privileged instructions that are supposed to occur in kernel mode within the guest operating system. Binary translation is a common technique to overcome such an issue. Binary translation rewrites selected instructions, such as those that would automatically fail if run in user mode instead of kernel mode. This technique is called *trap and virtualize*. Much of the binary translation and emulation work comes from the open-source QEMU[13] processor emulator project. The memory management unit (MMU) proved to be another road block that needed to be overcome. The MMU is a hardware device which handles memory access requested by the processor in a translation effort to convert virtual addresses to physical locations. Since the guest operating system could not be given direct access to the system's physical MMU, otherwise the host operating system or hypervisor kernel would have non-coherent memory accesses, MMU functionality had to be duplicated in software with control given to the hypervisor. A shadow MMU is emulated in software with the hypervisor at the host level managing the physical MMU and the shadow copy. There is clearly overhead with the shadow duplication, translation requirements and I/O device emulation. Some vendors and developers did not follow these design decisions and ported operating systems to

Table 3.1: AMD-V Instruction Set Extensions.

CLGI	Clear global interrupt flag.
INVLPGA	Invalidate a specific TLB entry value.
MOVCRN	Move control register.
SKINIT	Secure initialization with jump.
STGI	Set global interrupt flag.
VMEXIT	Stop guest VM execution and start host execution.
VMLOAD	Load VM state.
VMMCALL	Call hypervisor.
VMRUN	Run VM instance.
VMSAVE	Save current VM state.

remove instruction calls that cause virtualization problems using an application binary interface. The most famous example of this is the Xen hypervisor. Running a modified or ported guest operating system is called paravirtualization.

Many modern x86-64 processors include a x86 hardware virtualization feature allowing multiple operating systems to efficiently and securely share host system resources. Both AMD and Intel have added instruction set extensions to the x86 instruction-set architecture to support virtualization.

AMD-V

AMD's virtualization technology, AMD-V[14] not only introduces virtualization specific instructions but a variety of virtualization specific features. Additions to the instruction-set are shown in Table 3.1. Besides additional instructions, a tagged TLB (translation lookaside buffer) was added. The tagged TLB allows for efficient virtual machine switching. New hardware for better memory management was also added denoted as "Rapid Virtualization Indexing". I/O virtualization additions allow direct device access by a virtual machine, bypassing the hypervisor.

VT-x

Intel's response to hardware virtualization support came in the form of its own virtualization technology called Intel VT[15]. Added virtualization instructions are given in Table 3.2. Not only new virtualization instructions were added to facilitate virtualization. A new priority system was introduced to ensure that higher priority tasks are given appropriate attention. I/O device hypervisor bypasses are also available.

Table 3.2: Intel VT Instruction Set Extensions.

`VMCALL`	Call to hypervisor.
`VMCLEAR`	Clear virtual machine control status.
`VMLAUNCH`	Launch VM.
`VMRESUME`	Resume VM.
`VMPTRLD`	Load pointer to VM.
`VMPTRST`	Save pointer to VM.
`VMREAD`	Read VM pointer.
`VMWRITE`	Write control data to VM pointer..
`VMXOFF`	Exit VM operation.
`VMXON`	Begin VM operation.

3.2.2 Commercial Server Grade Type 1 Hypervisors

VMWare ESX

VMWare ESX is a proprietary industry offering which provides a hypervisor that sits directly on top of the physical hardware. It is used in many conventional IT infrastructures to host multiple virtual servers on the same physical machine. The ESX hypervisor is based off the Linux kernel and provides advanced resource management, memory management, performance and security. VMWare ESX supports a variety of operating system guest operating systems including Microsoft Windows and Linux. Most notably, VMWare ESX, was one of the first to introduce live migration allowing a virtual machine to move between hosts with virtually no downtime.

Hyper-V

Hyper-V marked the entrance of Microsoft into the virtualization forefront. Hyper-V is again a Type 1 hypervisor running as a bare-metal application. Hyper-V requires that its installation partition also contain a Windows Server instantiation. Due to the requirements of the parent partition need, a large hypervisor footprint is present. As of Windows Server 2008 R2, live migration support has been added eliminating the need for a full fail over. Similar to other offerings, a variety of Windows and Linux flavors are supported.

3.2.3 Xen

Xen is an open-source and proprietary enterprise hypervisor which is currently owned by Citrix. Citrix provides proprietary enterprise hypervisor products using Xen technology under the name Citrix XenServer. Xen initially required guest operating system modification, paravirtualization, to run if virtualization hardware is not available. Similar to Hyper-V, Xen requires a control operating system to be installed alongside the hypervisor to control the guest operating systems. Due to the advances of Intel and AMD in providing recent virtualization extensions for full virtualization, Windows is now fully supported on Xen without requiring operating system porting. In all cases though, paravirtualization has shown higher performance compared to full virtualization. Developers are continuing to develop patches for a variety of operating systems such as Linux and Windows to take advantage of the well-known performance boost of paravirtualization.

3.2.4 The Role of Virtualization in the Cloud

Virtualization separates the concept of system resources from the underlying system architecture and adds quite a bit of flexibility into how applications can be deployed and what resources they use. Hypervisors act as the gateway appliance for all cloud computing functionality. All applications deployed to a cloud computing environment are executed on a virtual platform, inside a virtual machine. We've just seen how we can create multiple virtual platforms on a single physical target machine. We now need to consider the cloud mentality of spreading these virtual images across any number of physical target machines in an effort to maximize availability and performance.

The traditional IT infrastructure runs a multitude of corporate applications such as email and an intranet and dedicates a physical server to each task. This approach is clearly costly and may not be the best use of resources as many applications are not performing at a peak rate 100% of the time. The cost of power alone has become a huge issue in large infrastructure setups and data centers. A portion of this utilization issue was addressed with the adoption of virtualization, where multiple physical systems were migrated onto a single physical machine as virtual machine images. It is very unlikely that any large IT infrastructure can subside on a single server box, virtualized or not. Noting this, we can deduce that the utilization problem still arises. Consider a server farm that is running many virtual machine images on each server box. The question cloud computing looks to address is, *"how can we seamlessly move these images from* *machine to machine for best resource utilization, performance and availability given a group of servers?"*

As previously discussed, new virtualization hardware advances in processor technology have allowed for a low-level pause and resume of executing virtual machine instances. This technology facilitates the ability to pause an application running within an instance and pass the application request directly to a newly spawned instance if a load were to become too large to handle. Virtualization facilitates dynamic load balancing considering the potential for automatic spawning and closing of further virtual machine instances based on temporal application load and performance requirements. Vendors are providing trimmed down operating system kernels which avoid full-feature duplication across the same instances lowering the required application overhead footprint.

Cloud computing extends this dynamic virtualization idea by not confining the automated migration and spawning to a locally owned private group of servers but to a group of servers that don't have a specific location, they are in the cloud (Fig. 3.3). The cloud computing concept facilitates the idea that we need not know about the server or be responsible for the server. A cloud service provider will offer different services as previously explored: SaaS, PaaS or IaaS. Depending on the service in the cloud, the customer might have access to an application, operating system or both.

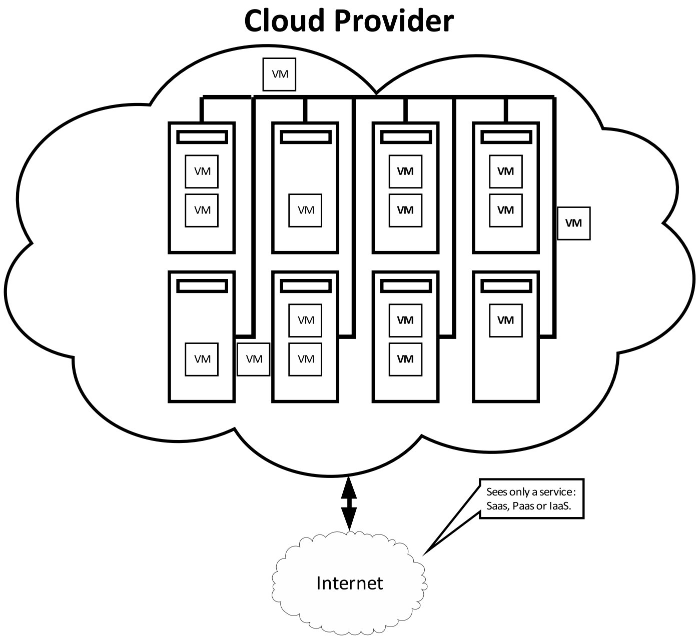


Figure 3.3: High-level view of the cloud.

Chapter 4

Cloud Computing Technologies for Today's Infrastructures

4.1 Chapter Overview

In this chapter, we discuss conventional application development for cloud targets and how the applications are delivered to the end-user. Furthermore, current cloud computing vendor offerings are analyzed and discussed.

4.2 Applications in the Cloud

4.2.1 Abstracting Some More-The Development Model

As cloud computing progresses, the next abstraction stage is that of the operating system itself for entire offerings. We have advanced in programming language abstraction moving away from assembly and database connection abstraction moving away from direct SQL commands. Interpreted programming languages such as Java or Adobe AIR have abstracted even further away from the operating system. Cloud computing will one day offer the ultimate abstraction where a new cloud job standard has been developed to let developers describe their application without any operating system, coding language, storage, target machine specifications or after-thought. Such a new abstracted standard would also allow for interoperability between cloud vendors which is currently not available. Cloud computing has pushed the edge of computing technology and application abstraction and will continue to push well into the future.

Developers utilizing SaaS or PaaS services are generally not so concerned with such future abstraction talk; however, the lack of customization which IaaS overcomes will be at the forefront of cloud advances in forcing new abstraction levels. Market researchers [16] have closely watched IaaS and its recent explosion and are both equally excited by the implications and concerned about the need for standardization due the immaturity of the product.

4.2.2 Bringing the Customer to the Cloud

There are a variety of considerations when looking at how an application deployed in the cloud will actually have its service be used by end-users. The traditional definitions of thick and thin clients apply to this scenario as well:

1. **Thin Client for Cloud Service**
   A web browser is used as the form of interaction requiring no extra standalone application. Web services are traditionally delivered in this fashion (*i.e.*, YouTube or Google Docs.)

2. **Thick Client for Cloud Service**
   A thick client will require additional software application(s) to be installed on a user's system to interact with the service being offered by your application in the cloud. (*i.e.*, Dropbox).

There is a clear argument for both interaction methods. The debate focuses on interoperability and ease versus feature rich interaction and user system integration.

4.2.3 Examples of Cloud Application Types

There are a variety of definitions trying to pinpoint exactly a way to categorize the types of clouds that can offer services. Example cloud application types from [6] are discussed next.

Processing Clouds

Processing clouds are cloud computing services which offer on-demand computation power for a variety of applications from data processing to handling web service overload from a local data center. Amazon's EC2 is an example of a processing cloud which is discussed in further detail in the following section.

Storage Clouds

Companies can offload their storage needs to a cloud service instead of investing in a data center infrastructure. Cloud storage can also be used for periods where corporations need additional dynamic storage. Amazon S3, Dropbox and Microsoft's SkyDrive are services offered as storage clouds.

Groupware Clouds

Groupware clouds are cloud services which support desktop office applications, collaboration and project management. Google Docs has emerged the leader in this area, but Microsoft has recently tested the waters with its Office Live linking to its other other cloud services such as SkyDrive.

Anti-Spam Clouds

Cloud services are being used to meet the needs of spam filtering by being scalable and adaptable to the various volume of scan requirements. Postini is an excellent example of such an application implemented as a cloud service. Note that, as cloud services are offered cheaply on a usage basis they have become a haven from which spammers distribute unsolicited email spam [17]. It is ironic that the cloud is providing tools to prevent spam and also tools to facilitate its dispersal.

4.2.4 Making Money as a Cloud Service Provider

Cloud service providers (CSP) provide the cloud infrastructure to consumers and have a variety of methods of garnering revenue. Common billing practices are listed:

1. Computation time

2. Application use charge

3. Memory usage
4. Bandwidth usage as throughput
5. Storage usage

4.3 Cloud Service Providers

4.3.1 Amazon Web Services (AWS)

Amazon has emerged as a cloud computing leader providing a variety of cloud based services. Amazon Web Services (AWS) is the umbrella term used to group all of Amazon's cloud services together. The most popular Amazon cloud computing services are Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3). Both are discussed further below. A listing of all currently available cloud services are listed in Table 4.1. Amazon has deployed its cloud infrastructure at data centers across the world to offer higher availability and better geographical fault tolerance.

Amazon Elastic Compute Cloud (EC2)

The Amazon Elastic Compute Cloud (EC2) is a web service which provides on-demand computing power. EC2 allows customers to rent virtual computers to execute compatible applications. Application deployment is done through the purchase of EC2 instances which run a customer's Amazon Machine Image. Available Amazon EC2 instances range from a *Micro Instance* to *Clusters with GPU Instances*. Examples of available instances for hourly rental are listed next from [18].

1. **Micro Instance**
   - 613MB RAM
   - Up to 2 EC2 Compute Units
   - EBS storage only
   - 32-bit or 64-bit platform
   - I/O Performance: Low
   - $0.02/hour for Linux — $0.12/hour for Windows

Table 4.1: Amazon Web Services Listing.

Amazon Elastic Compute Cloud (EC2)	On-demand compute power in the cloud.
Amazon Elastic MapReduce	Web service facilitating data processing.
Auto Scaling	Automatically scale EC2 capacity based on user rules.
Amazon CloudFront	Low latency content distribution such as streaming media.
Amazon SimpleDB	Database service for EC2 and S3.
Amazon Relational Database Service (RDS)	Service to setup, configure and maintain a database.
AWS Elastic Beanstalk	Automated handling of capacity provisioning, load balancing and application monitoring with simple application deployment.
AWS CloudFormation	Allows developers to group AWS services for better provisioning.
Amazon Fulfillment Web Service (FWS)	Full e-commerce setup.
Amazon Simple Queue Service (SQS)	Hosted storage queue for storing messages between computers and web services.
Amazon Simple Notification Service (SNS)	Service designed for messaging notifications from the cloud.
Amazon Simple Email Service (SES)	Scalable email web service.
Amazon CloudWatch	Cloud monitoring web service for developers and system administrators.
Amazon Route 53	High availability DNS web service.
Amazon Virtual Private Cloud (VPC)	Allows companies to connect existing infrastructure to cloud resources securely.
Elastic Load Balancing	Automatically distributes incoming application traffic across multiple EC2 instances.
Amazon Flexible Payments Service (FPS)	Electronic payment web service.
Amazon DevPay	Billing and account management web service for AWS applications.
Amazon Simple Storage Service (S3)	High availability data storage infrastructure service.
Amazon Elastic Block Store (EBS)	Block level storage for EC2 instances.
AWS Import/Export	Acceleration of data transfer to and from AWS services.

2. **Standard Large Instance**
    - 7.5GB RAM
    - 4 EC2 Compute Units
    - 850 GB instance storage
    - 64-bit platform
    - I/O Performance: High
    - $0.34/hour for Linux — $0.48/hour for Windows
3. **High-Memory Quadruple Extra Large Instance**
    - 68.4GB RAM
    - 26 EC2 Compute Units
    - 850 GB instance storage
    - 64-bit platform
    - I/O Performance: High
    - $2.00/hour for Linux — $2.48/hour for Windows
4. **Cluster Compute Quadruple Extra Large Instance**
    - 23GB RAM
    - 33.5 EC2 Compute Units
    - 1690 GB instance storage
    - 64-bit platform
    - I/O Performance: Very High
    - $1.60/hour for Linux
5. **Cluster GPU Quadruple Extra Large Instance**
    - 22GB RAM
    - 33.5 EC2 Compute Units
    - 2 x NVIDIA Tesla M2050 GPUs

- 1690 GB instance storage
- 64-bit platform
- I/O Performance: Very High
- $2.10/hour for Linux

Its clear that Amazon offers a wide variety of computational solutions including new GPGPU technologies within clustered environments. The information presented in the previous list is self explanatory except for the definition of an EC2 compute unit as opposed to a "processor". The Elastic Compute Unit (ECU) is a processing resource abstraction. According to [18], one ECU provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

The Amazon EC2 service is provided using Xen virtualization as described in the previous chapter. Amazon Virtual Images (AMI) are used to instantiate paid-for EC2 services. AMI are the base unit of deployment for service delivery via the EC2 cloud infrastructure. AMIs are provided with a read-only file-system and operating system. Additional software packages can be purchased on pre-made AMIs such as those including IBM DB2, IBM WebSpehre or Oracle WebLogic Server. AMIs are stored within the Amazon S3 storage service.

Amazon S3

Amazon offers a high availability online storage web service named Amazon S3. The underlying storage service design and architecture has remained proprietary. Fees are charged based on monthly rates for storage usage and data transfer from and to the S3 store. Buckets are the main storage unit abstraction. Arbitrary objects from 1B to 5TB can be stored within buckets, each accessible through a unique identifier. Buckets can be mounted within the EC2 file system, seeded as torrents or accessed through a web interface. Buckets can also be integrated into custom .NET or Java applications.

4.3.2 Eucalyptus

So far, we have seen a rather large offering by Amazon in the form of the Amazon EC2 service. Eucalyptus is an open-source/proprietary software middleware for providing private cloud computing on an available local compute cluster as IaaS. Both the Amazon

EC2 and Amazon S3 interfaces are supported, therefore applications created for Eucalyptus on a private cloud are compatible with Amazon's AWS cloud. Kernel based virtual machines, VMWare and Xen can be used by Eucalyptus for cloud abstraction.

Ubuntu has integrated Eucalyptus into a product, offering the ability for clouds to be deployed easily on private clustered computing environments, Ubuntu Enterprise Cloud (UEC)[19]. The purpose of such a product is to create a private cloud. Private clouds are used as prototyping platforms and for pooling currently available IT resources for a variety of computationally intensive tasks. An overview of the architecture is presented next with the infrastructure shown in Fig. 4.1.

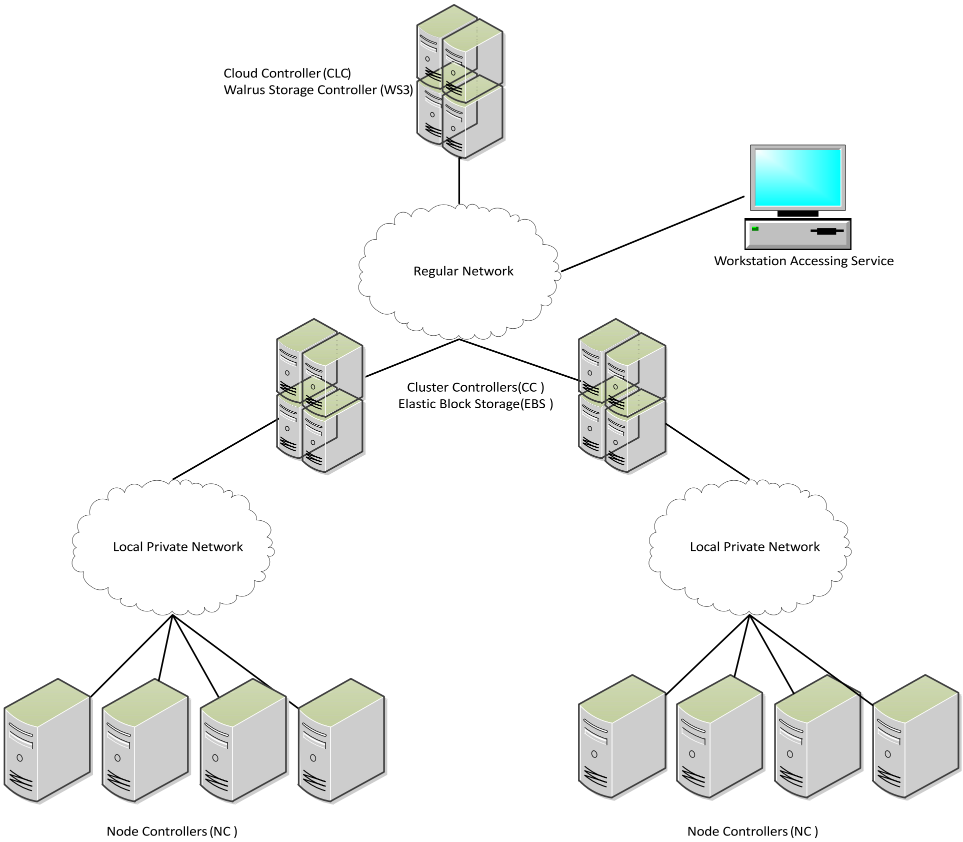


Figure 4.1: UEC cloud infrastructure. Adapted from [19].

Cloud Controller

The cloud controller provides the visible interface to the cloud. It is the job of the cloud controller to ask the cluster controllers to allocate new instances.

Walrus Storage Controller

The Walrus Storage Controller offers a file level storage system with a transactional model compatible with that of Amazon S3.

Elastic Block Storage Controller

The elastic block storage controller is responsible for allowing the creation of persistent block devices by the user. Persistent blocks are mounted by the VM to act as virtual hard disks. File systems can be created on top of these block volumes. It is interesting to note that packets between the elastic block storage controller and node controllers will only be properly transfered if both controllers are on the same Ethernet segment. This Ethernet and storage limitation will be overcome in future versions that support iSCSI.

Cluster Controller

The cluster controller processes allocation requests for virtual machine images from the cloud controller. The cluster controller decides which node controller will run a virtual machine instance based on usage information provided by the node controllers.

Node Controller

The node controller runs on the server boxes on which the virtual machine images will be instantiated upon. It is the job of the node controller to interact with the host operating system and the hypervisor while following the commands of the cluster controller. The node controller must keep track of its available resources and present that to the cluster controller. Upon instantiation request by the cluster controller, authenticity is verified, the virtual machine image is downloaded from the Walrus Storage Controller, a virtual network interface is created and the virtual machine instance is then started.

4.3.3 Microsoft Azure

The Windows Azure Platform[20] signified Microsoft's entrance into the PaaS market segment and has recently brought about certain IaaS features with the introduction of Windows Azure VM. The product is designed to allow clients to deploy applications and data into the Microsoft cloud infrastructure. The Azure platform consists of two products:

1. **Windows Azure**

   Windows Azure is a cloud operating system sitting atop the Microsoft cluster. The operating system acts as a runtime for user selected service applications consisting of three core components: compute, storage and fabric. The compute component focuses on web applications, while the storage addresses scalable storage needs. The fabric component describes the network of computing and storage nodes along with the interconnection network. Resources found within the fabric are managed by a control service, the kernel of Windows Azure. This controller provides scheduling, resource management, device management, fault tolerance and load balancing. Service virtualization at this level is provided by a modified Hyper-V hypervisor implementation called the Windows Azure Hypervisor. AppFabric runs as a middleware on Windows Azure offering a PaaS product. AppFabric provides a development, deployment and management framework abstraction which sits above the Windows Azure cloud operating system. Provided are access control systems and connectivity to business services.

2. **SQL Azure**

   SQL Azure is a service similar to Amazon S3 offering cloud storage solutions along with a relational database connection.

Chapter 5

Issues and Directions for Cloud Computing

5.1 Chapter Overview

In this chapter, we look to point out issues and considerations when discussing cloud computing offerings. Finally, we speculate on future cloud computing directions given the current trends and product offerings.

5.2 Issues with Today's Cloud Computing

Cloud computing has been a disruptive force in IT circles and also a confusing one. Companies and individuals seem to change the definition at the whim of a hat, but finally a solid model is emerging. The most concise definition that most parties seem to be agreeing on surrounds the concept of IaaS, PaaS and SaaS external offerings in an effort to minimize required IT resources and manpower.

As cloud computing remains an evolving technology, key pieces remain missing. We are missing ways to textually and programmatically identify resource needs. Grid computing has paved the way in such a field with job description languages so it is a safe assumption to make that appropriate advances are being made. Standardization is a huge mess right now in the cloud computing arena with each vendor offering different tools, environments and most importantly access APIs. It is safe to say that a clear standard is needed which will facilitate and allow “cloud shopping” giving cloud applications the option to be moved from vendor to vendor. Amazon is currently leading the pack, but only time will tell if a defacto standard will actually emerge.

We have discussed so far how cloud computing can be a huge cost savings. This could in fact not be the case with the recent rise in bandwidth costs. Consider a company which uses a cloud storage service for all its internal applications and data stores. Each access to such data will move over the external metered WAN with variable matching costs. Companies’ internal networks have always incurred only a static installation and maintenance cost. An adequate balance must be found.

Security is one of the largest topics discussed when considering the possibility of leveraging the benefits of utilizing a cloud service. Security comes at multiple levels. It is highly doubtful and expected that mission critical and high-security applications will ever move to a cloud computing provider. Imagine the implications of a financial institution utilizing third party computing and storage services. For typical applications deployed in a cloud, it is up to both the customer and cloud service provider to ensure a secure environment. In today’s age, data is the key asset to protect compared to computing assets many years ago. Data has become a commodity with responsibilities and inherent trust relationships. Hybrid models exist suggesting that mission critical applications remain in-house where other less vital applications are pushed to a cloud.

Availability issues are always in the back of the minds of large companies which require a large online presence. Down-time has a dramatic effect on such entities especially those using cloud resources for sales or billing applications. Problems such as this plagued the initial release of Amazon AWS and have subsided as the technology matured; however, Amazon still only commits to a 99.95%[9] uptime. Service Level Agreements (SLA) with a cloud provider have formed the basis of insurance for these types of needs.

Cloud service providers are popping up on a weekly basis. A transition to cloud computing is an investment and a company must be sure of the promises made by a cloud service provider and also that they will remain in the market long enough to actually offer a benefit especially due to the lack of standardization for seamless transfer between cloud service providers. Management and staff must ask themselves realistically how long this cloud provider will remain in business.

It is been noted that developers are finding it difficult to tune applications especially within the PaaS market as they don’t have intimate of the inner workings under the layers of abstraction. This can pose problems in analyzing application bottlenecks and performance characteristic issues.

5.3 The Future of Cloud Computing

The future of cloud computing is difficult to pinpoint. It has been speculated in [6] that clouds may take a more specialized role in the future. We have recently seen Groupware and collaborative office applications hosted in a cloud environment exploding over the last year and it is a safe guess that their popularity will increase. Such offerings also open the door to many new target consumer devices which can take advantage of application services provided by a cloud. Mobile devices seem to be the next step in pushing collaborative office appliances.

The future of clouds is only limited by the imagination of today's engineers. Advances in middleware layers are very apparent in new cloud service vendor offerings. The underlying cloud hardware has only seen minor improvement with recent advances in GPGPU (general-purpose computation on graphics processing units) offerings on available IaaS instances. There will indeed be a point where cloud vendors must seriously look at the cost of their infrastructures as companies looking to use cloud services are doing right now. Research in the areas of better resource utilization such as [21] offer insight into how such providers can improve their infrastructure and effectively offer a cheaper solution to the consumer. This is imperative in keeping cloud vendor startups profitable and viable. It seems lately that large corporations have just been throwing good cash, potentially after bad, into the cloud computing game with the "me too" attitude screaming in an effort to not be left out.

Following the current trends, we can expect new cloud computing regimes to further utilize abstraction layers in an attempt to further simplify the IT infrastructure and computing platform. Adequate development tools must improve along these lines as well. This is one particular area where companies have really been pushing with well documented SDKs like Amazon or seamless Visual Studio integration for Windows Azure applications from Microsoft. The future will close existing disparities in the development process and entire development community.

5.4 Concluding Remarks

Cloud computing does many things, but its main feature point has been the addressing of business needs. Business concerns come in the form of cost savings as ROI, simplified IT infrastructure management requirements and direct business process integration. Before considering cloud computing as a viable option, the main question any person should ask is, *"can our company even use a cloud environment and will the company benefit from such a transition?"*

Cloud computing is still in its infancy and throwing a company's IT needs directly into the fire has clear implications that must be addressed. Issues like security and standardization are at the forefront of the next feature-add to cloud computing. The current market trends forecast that cloud computing will continue its current growth pattern so it is only a matter of time before we see a large paradigm shift in how companies provide and use technology services. If cloud computing becomes so pervasive that the physical hardware computation platform becomes a thing of the past then cloud computing has succeeded in its ultimate goal of becoming the *computer*.

JP

References

[1] (2011) CRM and Cloud Computing. [Online]. Available: www.salesforce.com

[2] Y. Balzer, "Improve your SOA project plans," *IBM*, 2004.

[3] M. Biddick, "Why You Need A SaaS Strategy ," *InformationWeek*, 2010.

[4] (2011) Number of processors share for 11/2010. [Online]. Available: http://www.top500.org/charts

[5] V. Pande. (2011) Folding@home distributed computing. [Online]. Available: http://folding.stanford.edu/

[6] B. Chee and C. Franklin, *Cloud Computing: Technologies and Strategies of the Ubuquitous Data Center*. CRC Press, 2010.

[7] (2011) gLite Open Collaboration. [Online]. Available: http://glite.cern.ch/open_collaboration

[8] (2009) Cloud Computing Versus Grid Computing. IBM. [Online]. Available: http://www.ibm.com/developerworks/web/library/wa-cloudgrid/

[9] (2011) Amazon Elastic Compute Cloud (Amazon EC2). Amazon. [Online]. Available: http://aws.amazon.com/ec2/

[10] (2008) System z PR/SM. IBM. [Online]. Available: http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp?topic=/eicaz/eicazzlpar.htm

[11] R. Golberg, "Architectural Principles for Virtual Computer Systems," *Unknown Journal*, pp. 22–26, 1973.

[12] G. Popek and R. Goldberg, "Formal Requirements for Virtualizable Third Generation Architectures," 1974.

[13] F. Bellard. (2011) QEMU Open Source Processor Emulator. [Online]. Available: http://wiki.qemu.org/Main_Page

[14] (2011) AMD Virtualization Technology. AMD. [Online]. Available: http://sites.amd.com/us/business/it-solutions/virtualization/Pages/amd-v.aspx

[15] (2011) Virtualization. Intel. [Online]. Available: http://www.intel.com/technology/virtualization/

[16] "2011 trends to watch: Cloud computing technology," *2010 Trends Brief*, 2011.

[17] T. Samson, "Could Amazon's bulk-email service spawn spam and malware?" *InfoWorld Tech Watch*, 2011. [Online]. Available: http://www.infoworld.com/t/web-services/could-amazons-bulk-email-service-spawn-spam-and-m

[18] (2011) Amazon EC2 Instance Types. AMazon Web Services. [Online]. Available: http://aws.amazon.com/ec2/instance-types/

[19] S. Wardley, E. Goyer, and N. Barcet, *Ubuntu Enterprise Cloud Architecture*, Canonical, 2009.

[20] (2011) Windows azure microsoft's cloud services platform. Microsoft. [Online]. Available: http://www.microsoft.com/windowsazure/

[21] J. Parri, D. Shapiro, M. Bolic, and V. Groza, "Returning Control to the Programmer: SIMD Intrinsics for Virtual Machines," *ACM Queue*, vol. 9, pp. 30:30–30:37, February 2011. [Online]. Available: http://doi.acm.org.proxy.bib.uottawa.ca/10.1145/1943176.1945954