Cloud Computing: Implications for Enterprise Software Vendors (ESV)

by
Leonard Francis
M.S., University of Bridgeport
B.E., Shri G.S. Institute of Technology and Science

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

January 2009

Signature of Author ______________________________
Leonard Francis
System Design and Management Program

Certified by ______________________________
Michael A. M. Davies, Thesis Supervisor
Senior Lecturer, MIT Sloan School of Management

Accepted by ______________________________
Patrick Hale
Director, System Design and Management Program

ABSTRACT

'Cloud computing', is a broad concept and in general is a term used for internet-based computing resources that are in an unspecified remote location or locations and that are flexible and fungible. Clouds provide a wide range of computing capability available as a service where users are separated from the underlying technology by a set of APIs. These computing capabilities are made available by abstracting at different levels; at the hardware level, development platform or the applications level. Cloud computing is particularly helpful to application developers and IT operations because it allows them to focus on the service/application provided rather than worrying about scaling, failure, maintenance or reliability of these computing resources. By consolidating and sharing computing resources among multiple tenants thus improving utilization, cloud computing brings cost savings to end users. The higher the abstraction level, greater are the benefits resulting from better resource utilization and thus more cost savings, both for providers and end users.

As computing resources become cheaper, network connectivity and bandwidth improve both in terms of availability and pricing and human resources becomes expensive, cloud computing is increasingly seen as viable replacement of enterprise owned local IT infrastructure. With the adoption of cloud computing comes a major shift in the underlying architecture of how we develop, deploy, deliver and run applications compared to existing behavior where we run applications on local computing resources and thus increasing pressure on enterprise software vendors to adopt these new business model for software development and new alternate software delivery models that are supported by and derive the benefits of cloud computing.

While legacy enterprise software can simply be installed and run on instances on the cloud using cloud based infrastructure services, maximum benefits are realized by end users when these applications itself are provided as a service in the form of a platform or software. To do so, in most cases, legacy enterprise software would have to go through an architecture overhaul to be able to deliver existing functionalities as a platform or software as a service. Enterprise software vendors would also have to change their current business models where large license revenues, high maintenance cost of antiquated versions and heavily invested customers are the standard and move to pay-per-use cloud computing model.

This thesis aims to study the implications of 'Cloud Computing' trends on the development, distribution, business models and the business of enterprise software vendors. This thesis tries to chart and predict the progress of trends in computing towards 'cloud computing', connect those trends to enterprise software usage changes and determine the impact on enterprise software vendors. This will help enterprise software vendors to determine what if any strategic options available will help adopt this technological innovation and conform to future enterprise software requirements based on this trend.

Thesis Supervisor: Michael A. M. Davies
Title: Senior Lecturer, MIT Sloan School of Management

Table of Contents

1. Introduction **1**
1.1 Technology Evolution 8
1.2 Business Model 13

2. Approach **23**
2.1 Motivation 23
2.2 Methodology 24

3. Analysis: Impact of cloud computing on ESVs **26**
3.1 Ecosystem Changes introduced by Cloud Computing 26
3.1.1 Scalability 26
3.1.2 Cost: Cloud vs. Owning 26
3.1.3 Adoption 36
3.2 Application of Frameworks 38
3.2.1 Impact on Revenues 38
3.2.2 Technical and Product Impact for ESVs 38
3.2.3 Impact on Costs 42
3.2.4 Business Impact 43
3.2.5 Theory of Disruptive Innovation 43
3.2.6 Clock Speed and the Double Helix Industry Movement 46

4. Recommendations **54**
4.1 Product Recommendations 54
4.2 Business Recommendations 59

5. Risks & Risk Management **62**
5.1 Risk 62
5.2 Risk Management 63

Appendix: Terminology **65**

Bibliography **67**

Table of Figures

Figure 1 Cloud computing classification....5
Figure 2 Plot of Benefits v/s User Capability Restrictions for Cloud Computing....7
Figure 3 Virtualization at different levels....10
Figure 4 Technology Evolution....11
Figure 5 Stakeholders in the Cloud ecosystem....16
Figure 6 Cloud Business Model for Enterprise Software....18
Figure 7 Cloud Provider Stack....19
Figure 8 Graph comparing cost for cloud vs. owned computing resources....29
Figure 9 Total expenditure comparisons for cloud vs. owned computing resources....32
Figure 10 Compute resource usage simulation for a day....33
Figure 11 Cost comparison for real world usage simulation....34
Figure 12 Cost Comparison: Amazon Cloud vs. Local IT....35
Figure 13 Cloud computing, a new market disruption....44
Figure 14 The Double Helix – Industry Movement....47
Figure 15 Clock Speed for Industries in the Software Value Chain....49
Figure 16 Enterprise Software Value Chain – Legacy....50
Figure 17 Enterprise Software Value Chain – With Cloud Computing....51
Figure 18 Network Centric innovation model....59

Acknowledgement

When I started this thesis, I thought I would mostly be using what I had learnt over the past two years at MIT. But going through the process I not only used what I learnt but I also learnt a lot. This process was made enjoyable by several helping hands and kind souls.

I would like to thank Michael Davies, my thesis advisor, who I have grown to understand and respect in this short period of time. With his quick invaluable insight, strong assertions and eagerness to bring out tangible benefits out of this thesis he has been a beacon of guidance in shaping this thesis.

I would like to thank Pat Hale, Bill Foley, Chris Bates and others in the SDM office for allowing me to choose the path along my journey and helping me enjoy what has been a wonderful journey.

I would like to acknowledge the sacrifices by my wife Vinita and son Rohan, without their love, enthusiasm and encouragement I would not have been able to embark on this journey in the first place.

1. Introduction

The cloud is a metaphor used for internet based computing resource derived from the common depiction of a cloud in network diagrams. The concept of cloud computing originated in the 1960's as a form of computation organized as a public utility. Cloud computing is a way of delivering IT-enabled services in the form of software, platforms or Infrastructure [29]. Cloud computing is:

- Computing resource that is 'out there', connected to via IP, typically over the Internet
- Is a Flexible and Fungible computing resource, i.e. it can be scaled up or down very easily and can be replaced without worrying about underlying hardware architecture.

That said, in general cloud computing has the following additional characteristics, which in no way defines what clouds are supposed to be but are ways in which clouds are typically implemented:

- Elasticity of the computing model: The cloud computing model allows users to scale up or down different resources allocated to them. Cloud providers charge users for computing resources just as a utility, i.e. on the basis of the amount of resources and the time period for which those resources are used. Elasticity allows users to use just as much resource they want and not get charged when not using any resource.

- Fault-tolerant or self-healing: In case of a failure the computing resource will be able to continue running the application without disruption using alternate resources. This allows achieving redundancies and guaranteeing uptime.

- Multi-tenancy: Multi-tenancy allows several clients to share the same instance of a software application where the application separates data and configuration allowing customers to experience a customized virtual application instance. Multi-tenancy allows underlying resources to be shared consolidating computing resources and achieving economies of scale.

- Utility computing driven by SLA: Provides computation and/or storage services on a metered basis managed by a service level agreement (SLA). The SLA determines the guaranteed uptime of the cloud, computing response time, failover recovery time etc.

- Autonomic features: where systems have self management features and require little or no administration of hardware resources by end users and thus reduce system administration and management efforts.

- Virtualization: Applications are decoupled from the underlying hardware resource. This allows running multiple applications on a single piece of hardware or a single application to run on multiple computer systems. Virtualization allows abstracting the hardware level so that any operating system can be run on available resources rather than worrying about a coupled hardware and operating system to run on it. Virtualization allows better utilization of computing resources.

Clouds are seen to have been implemented in different ways. One of the most common trends to building a commercial cloud is to build huge data warehouses with clusters that include thousands of servers in a geographical location where it is cheap to house and power these servers, run virtualization software that pool these resources together or help slice and distribute the chunks of the task to be performed. In some cases clouds may be built using grids with additional interface and service software running. Grids are basically an application of several computers, supported by interoperability technology, to divide, distribute and compute a single task.

With their different implementations cloud computing can be characterized by the level at which the underlying resources are abstracted. Along those lines the three levels of cloud offerings are:

- Software as a Service (SaaS): This levels includes software applications that are hosted and run on the internet, the user is not concerned where the applications are run or where the data is stored. The user connects to the service using a generic interface like a web browser. Some of the examples of SaaS are online webmail (Gmail, Yahoo mail etc), online subscription based sales and Customer Relation Management software (Salesforce.com), online document management services (Zoho, Google Docs) etc.

- Platform as a Service (PaaS): An application platform is offered at this level where developers can use available tools from the platform provider to write and deploy their applications on the infrastructure available from the provider. Examples are Google App, Force.com, and Microsoft Azure.

- Infrastructure as a Service (IaaS): Computing hardware resources is provided as a service via the internet and users/developers would typically obtain an instance of their compute and storage resource, connect to it remotely and use it just as they would use a server available to them. Examples are Amazon EC2, Flexiscale etc.

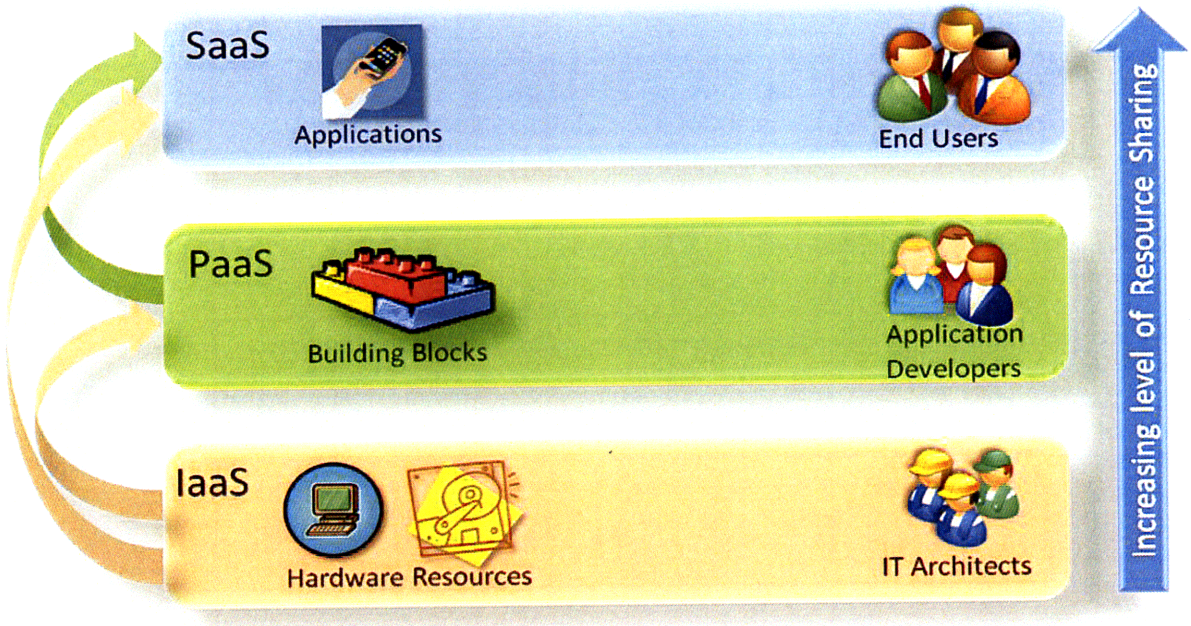


Figure 1 Cloud computing classification

With the above classification of cloud computing based on the type of resources provided as a service we find that SaaS implementations often make use of underlying PaaS and IaaS services. Similarly, PaaS implementations could be using an underlying IaaS service. As we go higher the offering levels, i.e. from IaaS to PaaS to SaaS the level of resource sharing and thus resource utilization increases. At the same time, as we move up the stack from IaaS to SaaS customers tradeoff the capability of what they can use the cloud resources for. At the IaaS level, users can create and run an instance of any operating system supported, then install and run any software they wish to or use that instance for any other purpose like serving a web application etc. At the PaaS level developers use the service to develop application using the provided software development kit and are restricted by the interface, language and the features that the PaaS service provider offers. At the SaaS level, end users are limited to using specific applications offered by providers and might have only limited capability to customize the interface to those applications.

Looking at the benefits at each level we see that the maximum benefits are derived at the SaaS level. The SaaS level involves maximum possible sharing of underlying computing resources, right from the hardware up to the software shared by multiple users. Users do not need to buy licenses at a fixed cost and can pay on usage basis only for the time that they use the software. There are no upgrades or patches to the software that the user needs to worry about and no maintenance related to backup security etc. From the application developers perspective putting out patches and upgrades is easy since only a centralized codebase needs to be updated. This allows putting out patches and upgrades more frequently and quickly.

At the PaaS level, application developers avoid building from scratch by using available prebuilt building blocks that provide required functionality. Developers can focus on the application without worrying about scalability or required infrastructure to develop these applications.

At the IaaS level users get the benefit of being able to scale hardware resources, up or down, based on their consumption. At this level, it is left to the user to install the required software on barebones computing instances and to configure and use these instances as the user wishes to.

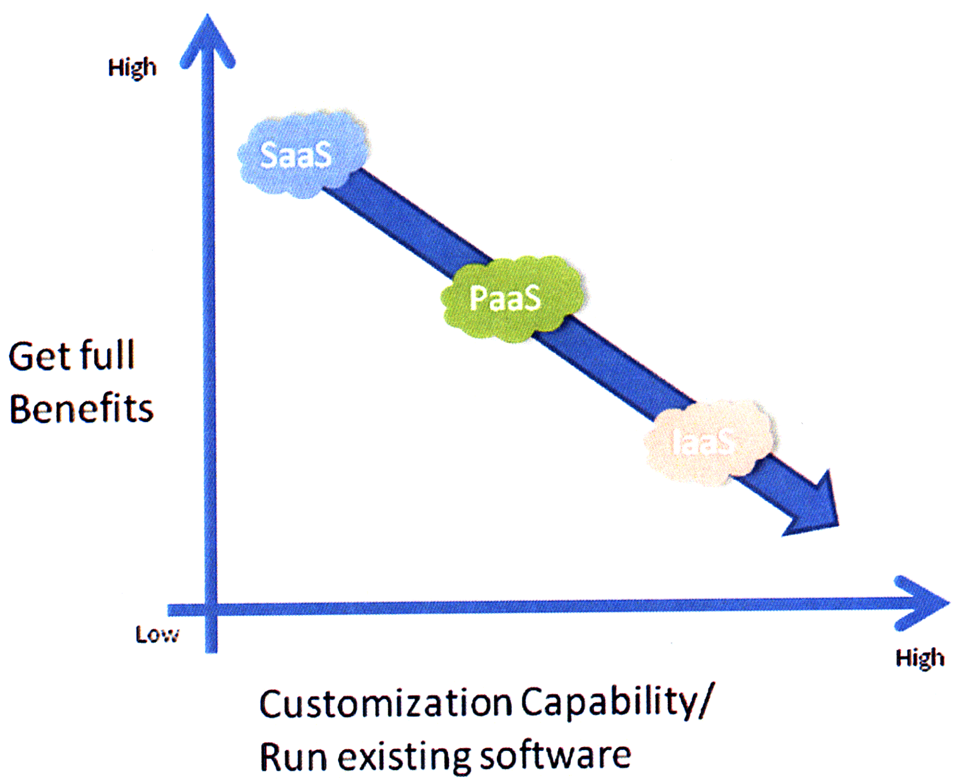


Figure 2 Plot of Benefits v/s User Capability Restrictions for Cloud Computing

1.1 Technology Evolution

Looking at the past technology trend that lead to cloud computing and associated virtualized platforms will help shed some light on where the technology is headed and how it will help shape future computation usage.

The technology evolution on the hardware front has been from mainframes to distributed computing as computing power got smaller and cheaper. Distributed computing involved individual computers connected to the internet for communication, but all of the applications would run on these individual computers and data stored locally. This was followed by evolution into use of clusters. Clusters are a group of linked computers working together closely behaving like a single high performance computer. Clusters are generally used for high computational task and are typically more cost efficient than a single computer of comparable performance. Clusters included

- High Availability Clusters: Also known as failover clusters are clusters with redundant node to improve availability of computing services.
- Load Balancing Clusters: Such clusters distribute workload over multiple nodes.

Grids followed as the next evolutionary step to clusters, which are very much similar to clusters except that grids are more focused on throughput like a computing utility. Grids run workloads that can be divided into many independent jobs that do not need to share data while they are run and execute them in parallel.

In the software area the concept of virtualization came into being with the Application Virtualization which is being able to run applications on alien hardware or operating system. Some of the examples of application virtualization are emulators and cross platform applications. The next stage in virtualization was Resource Virtualization, i.e. virtualizing computing resources like storage, memory or other network resources. Network Attached Storage (NAS) and Redundant Array of Independent Disks (RAID) are examples of storage virtualization. Resource virtualization was followed with Platform Virtualization which allowed to fully virtualize a platform i.e. separates the operating system from the underlying computer hardware by running what is termed as a Hypervisor[1] within which multiple operating systems can run sharing the hardware resources available to the hypervisor.

---

[1] A virtualization platform that allows multiple operating systems to run on a host computer at the same time.

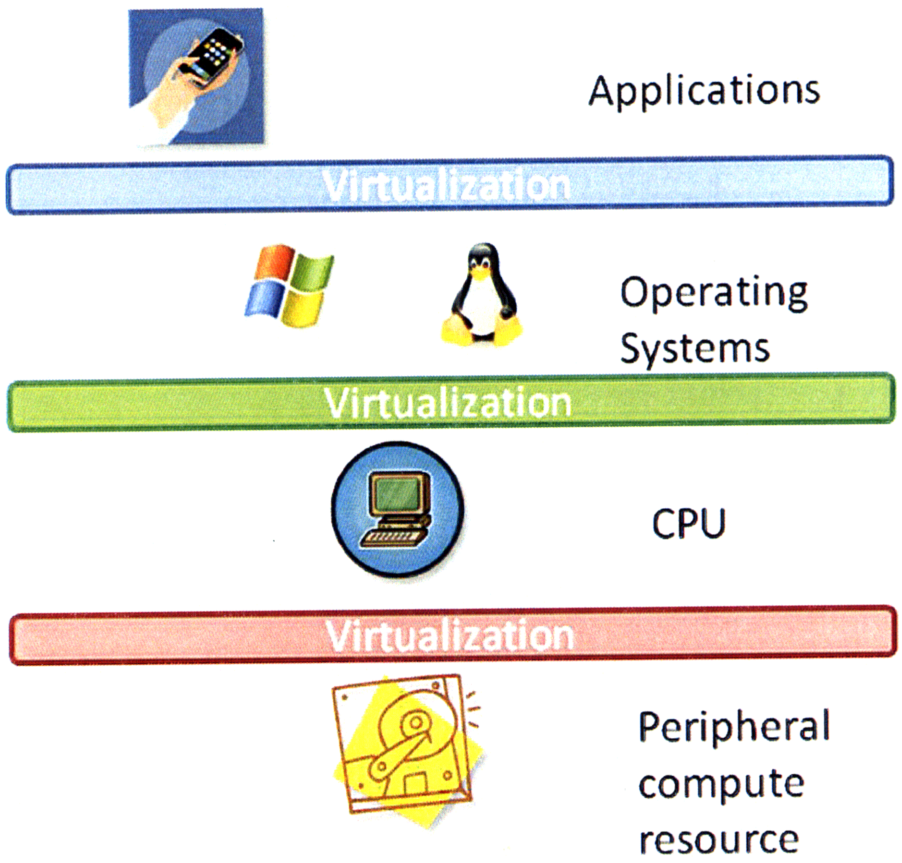


Figure 3 Virtualization at different levels

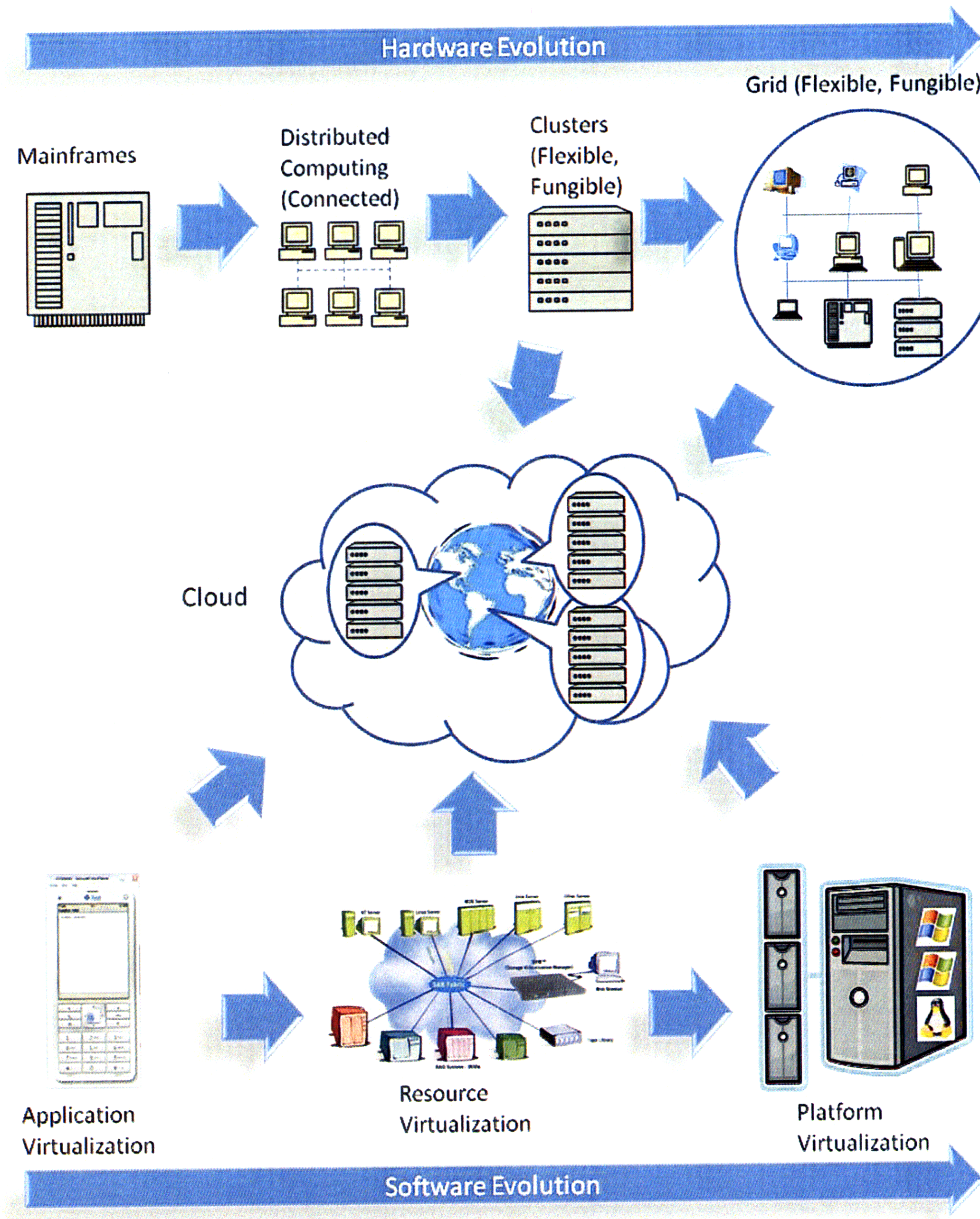


Figure 4 Technology Evolution

The evolution, both in areas of hardware and software has helped launch the next level of computational 'Cloud Services', where firms put together a large, powerful centralized computational resource made up of a high end clusters and grids, run virtualization software on these clusters/grids and help share these computational in as simple a way as plugging into an electrical terminal to connect to the electrical grid.

As opposed to the general meaning of the term 'cloud computing', clouds are not always a remote internet based rented computation service running out of a huge data warehouse from vendors who manage these large data warehouse, referred to as Public Clouds in this paper. Clouds can also be high end in-house/local computation resource pools that are shared using some form of virtualization software that enables to partition the hardware resource and lets multiple users share the underlying hardware each running their own instances or applications. These clouds are referred to as Private Clouds. In general both public and private clouds based on the virtualization software that they run are capable of running

- Specific platform based applications like web application
- Complete instances of operating systems allowing users to run a virtual instance of a server workstation or desktop.

1.2 Business Model

The stakeholders and their relations differ slightly when considering private and public clouds. The stakeholders involved in a cloud based business model are

- The Cloud service provider (only in public clouds): The cloud service provider is the one who hosts the hardware and required cloud infrastructure software and offer their services on a pay-per-use basis. The type of service can vary at different levels, it could be a complete service with all utility computation or a particular component like storage, infrastructure virtualization software etc.

- Application software developer (Enterprise and Open source groups): These are software vendors developing applications that can be targeted for the cloud platforms. Applications are generally scalable and enterprise grade. The providers include both enterprise software vendors and open source software vendors.

- Cloud Software/Platform users

- Local IT departments: In case of a private cloud local IT departments are the ones who host these cloud environments. When using a public cloud IT departments will usually help manage cloud instances on the public cloud.

- Hardware providers

- Cloud infrastructure and management software providers

The business model for private clouds is fairly simple where IT departments with a goal to reduce IT related costs, improve IT services and to incorporate flexibility, scalability and reliability choose to build a private cloud. The IT departments acquire hardware from traditional hardware providers and cloud infrastructure software, in most cases virtualization software, from cloud software firms and setup their very own cloud. Such clouds have a smaller range of scalability and reliability. Two main reasons why enterprises own private clouds are

- For the use as test beds for application development targeted to deployment on the cloud

- To host proprietary applications/content that firms do not want to give control to third party cloud providers. Such clouds are commonly used in the banking and financial industry where firms would like to keep data within their premises.

The private cloud helps improve hardware and software utilization by sharing the same underlying hardware or software resources among multiple users thus reducing the amount of hardware or the required number of software licenses. With increasing adoption of cloud architecture this translate to lower sales for hardware and application software vendors. Cloud Infrastructure software developers stand to gain with more usage of their software to run these clouds.

Compared to private clouds, public clouds are generally of a much larger scale and hence derive more gains by economies of scale, i.e. they are able to time share the same pool of resources among various customers whose usage patterns are different. In a public cloud, depending on the services provided (SaaS, PaaS or IaaS) the cloud provider deals with the hardware, cloud infrastructure software provider and with application software providers. With adoption of cloud computing infrastructure services from IaaS providers cloud computing is brings higher utilization of hardware resources. This will hit hardware providers on two fronts, on one hand it will reduce overall hardware sales and on the other hand it will consolidate hardware customers making IaaS providers large customers who will be making a growing share of the hardware acquisition giving IaaS providers an upper hand over hardware provider.

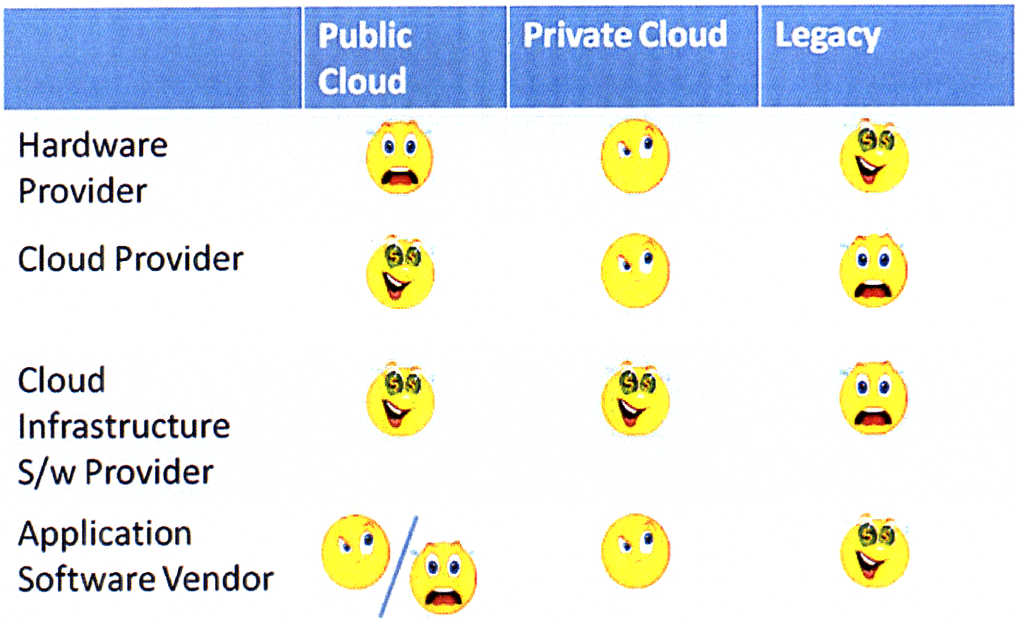


Figure 5 Stakeholders in the Cloud ecosystem

In the software applications space, PaaS and SaaS models bring compelling benefits to development and distribution. Given the choice between using applications the old way i.e. buying a license at a fixed cost, installing it on local hardware and using it as opposed to using application from a SaaS provider to accomplish the same task, users prefer to use the SaaS provided application. The SaaS model is beneficial to users since they pay on a usage basis, do not need any special hardware to run the application, need not worry about upgrades, patch installations, downtime, data backup, security or other maintenance. These benefits coupled with higher utilization of underlying computing resources in the SaaS model reduce cost of application usage for the end user. Similarly, the PaaS model helps developers to develop applications quickly using provided building blocks without the need of capital investment in hardware, software development kits or worrying about scalability and basic infrastructure related issues. SaaS and PaaS models have their own downsides too, with SaaS users have to depend on the provider for application availability, their data resides in the hand of the provider and they lose control over application downtimes, upgrades etc.

With the trend in moving to SaaS platforms end users are looking for SaaS solutions rather than legacy applications. Even though users can acquire infrastructure services from IaaS providers, install these legacy applications on cloud instances and run them on the cloud, at the end of the day the user is still responsible to maintain that piece of software on the cloud instance. IaaS providers, and in some case third parties, generally make available instance image stacks with required software that users can directly use on a pay-per-use basis without worrying about purchasing a license for the software application or installing it. But such a model too will face issues if customers want to take this image and add some customization to it, in which case updating the image with software application updates becomes a task for the user to follow up. That is probably why so far we have not seen images for custom enterprise applications widely offered by third parties or cloud providers. So, without porting a legacy application to the SaaS platform legacy enterprise application is left to partner with IaaS providers to make available instance images including their applications.

Thus, in this new business model the cloud provider is the application distribution channel who controls value in the supply chain and is bound to squeeze or replace other players.

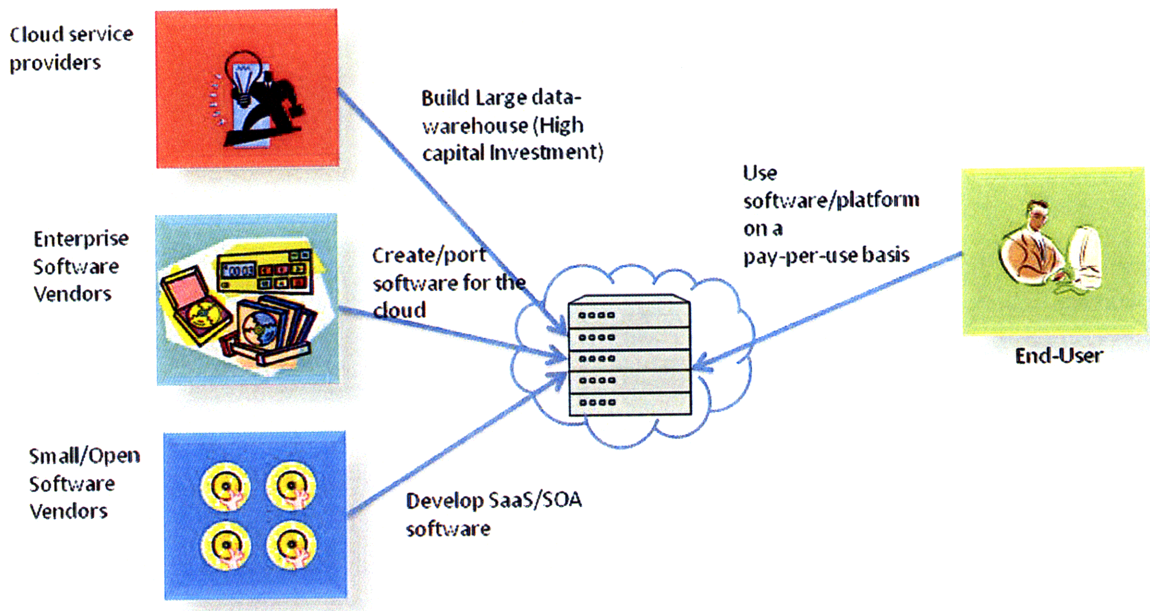


Figure 6 Cloud Business Model for Enterprise Software

Further looking at the cloud service providers in detail they can be broadly put under four categories that stack together to form the overall cloud offering:

- Infrastructure Provider: Provide the hardware infrastructure required to serve computing resource.

- Cloud Storage Providers: Storage services on the cloud where applications and databases can reside and from where they can be accessed and run.

- Infrastructure Software Provider: The core cloud infrastructure software that manages and shares computing resources.

- Cloud Administrative Software Providers: Software that helps users and cloud administrators to manage cloud services and enable the cloud business model.

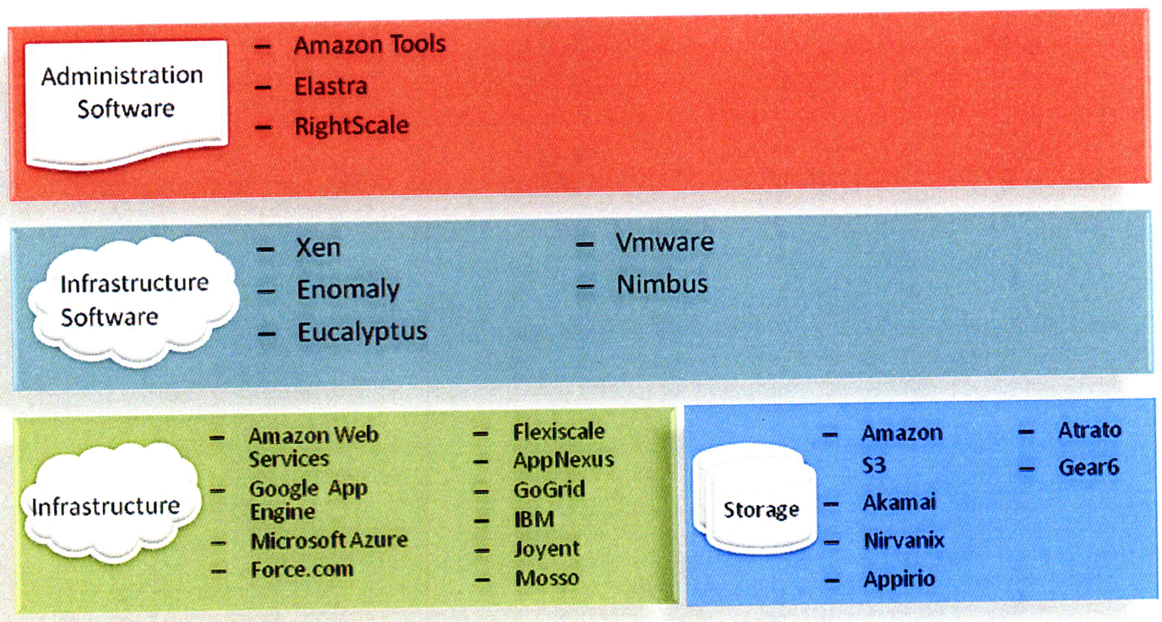


Figure 7 Cloud Provider Stack

Most of the cloud providers usually play in more than one layer of the cloud stack. Among all, Amazon is by far the leader in cloud services with its Amazon Web Services (AWS) that includes Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3) and Amazon SimpleDB services. The other big enterprise cloud players/offerings are Google with their Google App Engine, Microsoft Azure, Force.com from SalesForce, AppNexus, GoGrid, FlexiScale, VMWare etc.

The table below briefly describes some of the players in the cloud computing space and in what capacity are they involved.

Name	***Role Description***
Amazon EC2/S3/SimpleDB	Amazon offers the widest range of cloud services ranging from running web services to being able to use custom Linux, Solaris and Windows instances. Their cloud management tools need to be improved and the command line interface to manage virtual instances can be cumbersome. Amazon uses the open source Xen hypervisor for virtualization and disk storage on the S3 is volatile.
Google App Engine	The Google App Engine is a cloud platform that allows any Python based web application to run. The API is simple but limited and the pricing is based on complex metrics thus making cost difficult to anticipate.

Microsoft Azure	The Microsoft Azure builds upon the development tools and environment prevalent with existing Windows platforms like .Net, ASP and Visual Studio. Though applications developed on Windows under these same tools cannot be directly run on Windows Azure and need to be ported. Pricing for Windows Azure is not yet available
Force.com	This is a web application similar to Google's cloud service and had started as a provider of the slaeforce.com web application. Force.com uses the Apex programming language which is a new tool to develop on the force.com multitenant platform.
Flexiscale	Flexiscale is a UK based cloud service provider that provisions virtual dedicated Windows and Linux servers using ControlPanel or APIs.
AppNexus	AppNexus is primarily a scalable, enterprise self-service hosting service. It provides load balancing and storage facility
GoGrid	GoGrid offers a wide range of machine images that include CentOS and RedHat Linux with Apache, PHP, MySQL or Facebook and Windows server 2003 with IIS, ASP.Net and SQL Server. GoGrid does not offer database storage feature.
Joyent	Joyent provides virtualized servers, known as Accelerator, deployed in a high grade networking and routing environment that supports hardware load balancing and persistent and fast storage.
Mosso	Provides cloud services ranging from simple web application hosting to cloud servers in various flavors of Linux distribution.

	Their pricing includes fixed rate plans, i.e. monthly rate plans for specific virtual instances. Mosso uses VMware to create an EC2 like cloud but with persistent storage.
Xen	Xen is an open source hypervisor which is the fastest and most secure infrastructure virtualization solution available supporting a wide range of guest operating systems including Linux, Windows and Solaris.
Enomaly	Enomaly offers Elastic Computing Platform (ECP) which is a programmable virtual cloud infrastructure tool to design, deploy and manage a local or a public cloud. It supports multiple cloud platforms like Amazon EC2.
VMware	VMware provides enterprise grade infrastructure virtualization software that helps create and manage local or public clouds. VMware is the leader in x86/Windows virtualization hypervisor.
RightScale	Provides a dashboard and frond end for Amazon EC2 and S3 cloud services.

2. Approach

The goal of this thesis is to plot the trends in computing towards cloud computing and how cloud computing will evolve, and then determine its impact on software development and on entities in the enterprise software business ecosystem.

2.1 Motivation

Cloud computing is bringing about a major shift in the way software applications are deployed, distributed and paid for. This in turn stirs up the software business ecosystem impacting existing business models and creating opportunities for new business models. After successfully running free end user applications for non-essential purposes (like free web based email, social networking sites etc.) on the cloud for some years now, cloud computing has improved over recent years in terms of security, reliability, failover etc. making it now feasible to run large enterprise applications.

The required infrastructure to connect to these remote computing resources, broadband connectivity for both enterprises and home based connections has also improved over the years. Enterprises can now afford high bandwidth dedicated connectivity at affordable prices which can handle the data transfer volumes to serve cloud based applications and have a large customer base that connects to these computing resources remotely. With current broadband penetration of 57% in US households[2] making up about 87.49% of active internet users leaving a remaining 12.51% on narrowband connection using 56 Kbps or less.

It is now feasible and economical for end users to connect and use software on the clouds. Thus, within the cloud ecosystem I see enterprise software as one of the area that will see some of the highest impact.

2.2 Methodology

The method used to determine the impact of shift towards cloud computing on enterprise software vendors is based on determining answers to the following questions:

- How is computing shifting towards cloud computing?
- How is cloud computing evolving and how is it being accepted as a reliable environment to run critical application?

---

[2] From http://www.nielson-online.com/

- What is the business model with cloud computing?

- How would the changes in computing moving towards cloud computing impact the related business ecosystem which includes enterprise software vendors as one of the giants playing in this area?

- How would cloud computing business models force changes in legacy enterprise software business model?

To determine the evolution of the cloud technology existing literature on cloud computing was surveyed and a close tab kept on current information related to different players in the cloud computing space. Cloud computing resources from some of the providers was used to understand usage of these services and how these services would be used to run applications.

Various theories were applied ranging from Christensen and Raynor's theory of Disruptive Innovation to Charlie Fine's theory of Clock Speed to determine the evolution of the cloud computing technology and the evolution of the software business ecosystem. This is followed by a prescriptive recommendation on what challenges will enterprise software vendors face in light of these predictions and what steps they can take to mitigate threats and utilize the cloud technology development their advantage.

3. Analysis: Impact of cloud computing on ESVs

To determine the impact of cloud computing on ESVs the research looked at the pros and cons of cloud computing, how it impacts the overall computing ecosystem, where in its maturity stage is this technology, what is the market acceptance and what are the forecasts for cloud computing technology. Following this, the results were then translated into these ecosystem changes to impacts on ESVs.

3.1 Ecosystem Changes introduced by Cloud Computing

3.1.1 Scalability

Cloud computing makes it easy to scale computing resources up or down and makes it much easier to maintain these resources. Scaling on a cloud is as simple as running a few APIs to request more computing resources be it more processing power, more instances, more storage or capacity to handle more request by an application. On a local IT infrastructure scaling in most cases involves ordering and waiting weeks for additional hardware resource if you have crossed your limits on hardware resources.

3.1.2 Cost: Cloud vs. Owning

To determine cost impact of cloud computing IaaS and SaaS were analyzed. This analysis focused on Amazon EC2, one of the dominant IaaS provider, calculating the cost of running a fairly small IT system of about 200 compute units on Amazon EC2 and compared it to calculated cost when the IT system is built in house. Amazon EC2 provides various standard server instances with different compute, storage and I/O capacity. Each instance is charged at a standard cost per unit hour of usage, i.e. charges for instances are accrued only when they are used, once the instance is shutdown there is no charge for that compute instance. This analysis considered what is called the Extra Large instance available on Amazon EC2 charged at $0.80 per hour running a Linux/Unix operating system. It is assumed that one system administrator is required to maintain and manage these 200 compute instances in case of a local IT infrastructure but can manage about 2000 instances when managing Amazon instances since there is very little required in terms of maintenance etc. Assume that the System administrator is paid an annual salary of $120,000.

- Amazon EC2 Cost

Extra Large CPU instance cost per hour: \$0.80 per hour[3]

Maintenance cost per hour (system administrator) = \$120,000/year/12 months/30 days/24 hours/2000 machines = \$0.007

Total Cost per compute unit per hour = \$0.80 + \$0.007 = \$0.807

- Local owned IT infrastructure

Cost of a Single equivalent machine with 3 year service warranty = \$2500

Machine cost per hour amortized over 3 years = \$2500/12/3/30/24 = \$0.096

Cost of cabinet rent that houses 40 machines and power for a month = \$2500

Cabinet and Power cost per hour = \$2500/30/24/40 = \$0.087

Maintenance cost per hour (system administrator) = \$120,000/year/12 months/30 days/24 hours/200 machines = \$0.067

---

[3] From http://aws.amazon.com/ec2/

Total Cost per compute unit per hour = \$0.096 + \$0.087 + \$0.067 = \$0.253

This analysis shows that the cost per compute unit per hour is much cheaper in a locally owned IT infrastructure scenario, i.e. compute resource on the cloud is nearly 3 to 4 times the cost compared to owning those resources. Plotting per unit per hour cost both for cloud instance and locally owned IT for various figures of instances we see that the cost per hour on the cloud is almost 3 times the cost irrespective of the number of compute instances.

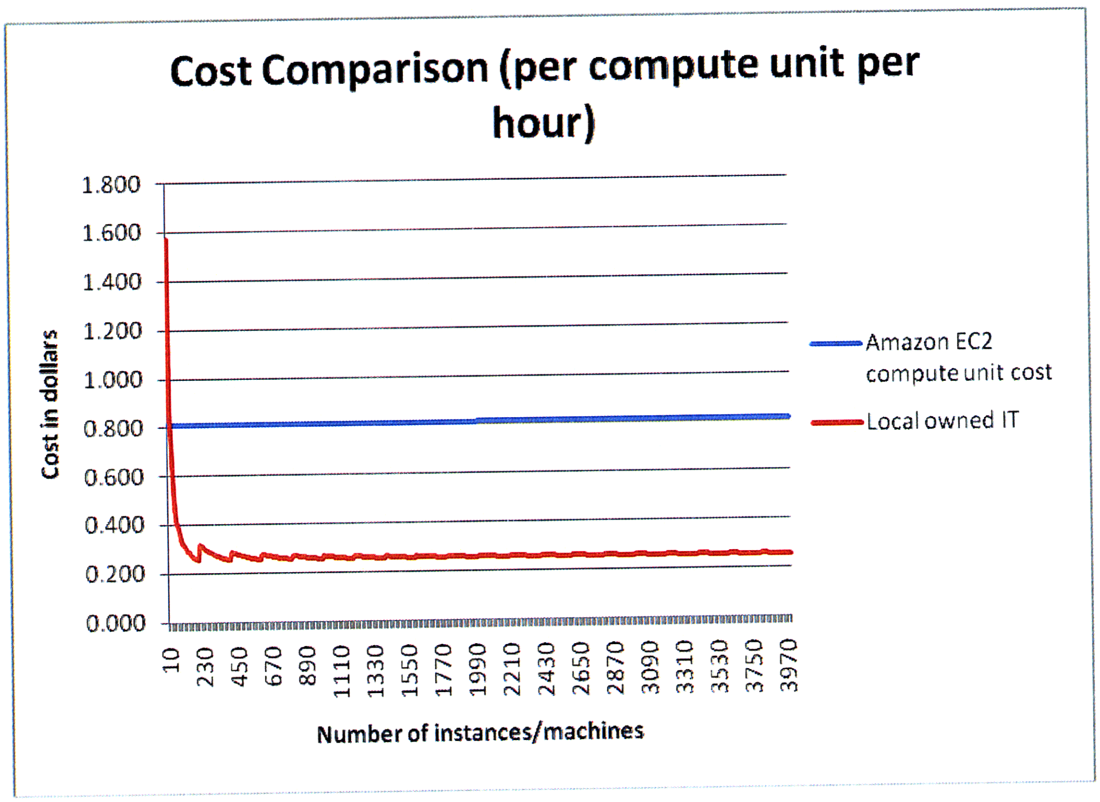
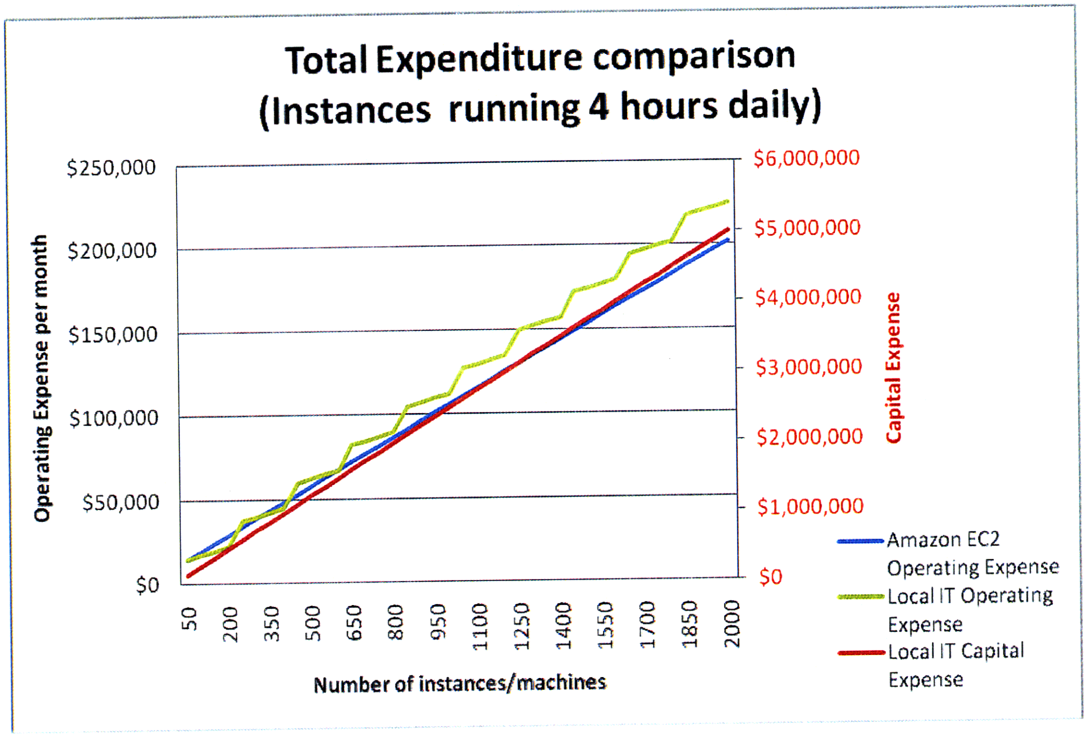


Figure 8 Graph comparing cost for cloud vs. owned computing resources

But the analysis makes one major assumption which skews the results. It computes the cost of compute resources per unit per hour and assumes that instances/machines are run 24x7. If these instances are run only a few hours per day the total cost incurred shows an entirely different picture. Also if we look closely into the analysis and look at the real cost that needs to shelled out we see that locally owned IT infrastructure requires a large capital expenditure where as renting compute resources in the cloud does not require any capital expenditure at all. Plotting the total expenditure comparison when instances operate only a few hours a day for Amazon EC2 and local IT we find that total expenses for Amazon cloud compute resources is very less compared to owning IT and paying capital as well as operating expense because of the fact that the operating expense on the cloud bills only for the instance in use. If the instance is shutdown after the 4 hour use daily Amazon will charge only for the 4 hours that the instance was running. Such scenarios represent applications with spikes in resource consumption, i.e. high resources consumption is required only for a short period of time.


Total Expenditure comparison
(Instances running 4 hours daily)
Operating Expense per month
$250,000
$200,000
$150,000
$100,000
$50,000
$0
Capital Expense
$6,000,000
$5,000,000
$4,000,000
$3,000,000
$2,000,000
$1,000,000
$0
50
200
350
500
650
800
950
1100
1250
1400
1550
1700
1850
2000
Number of instances/machines
Amazon EC2 Operating Expense
Local IT Operating Expense
Local IT Capital Expense

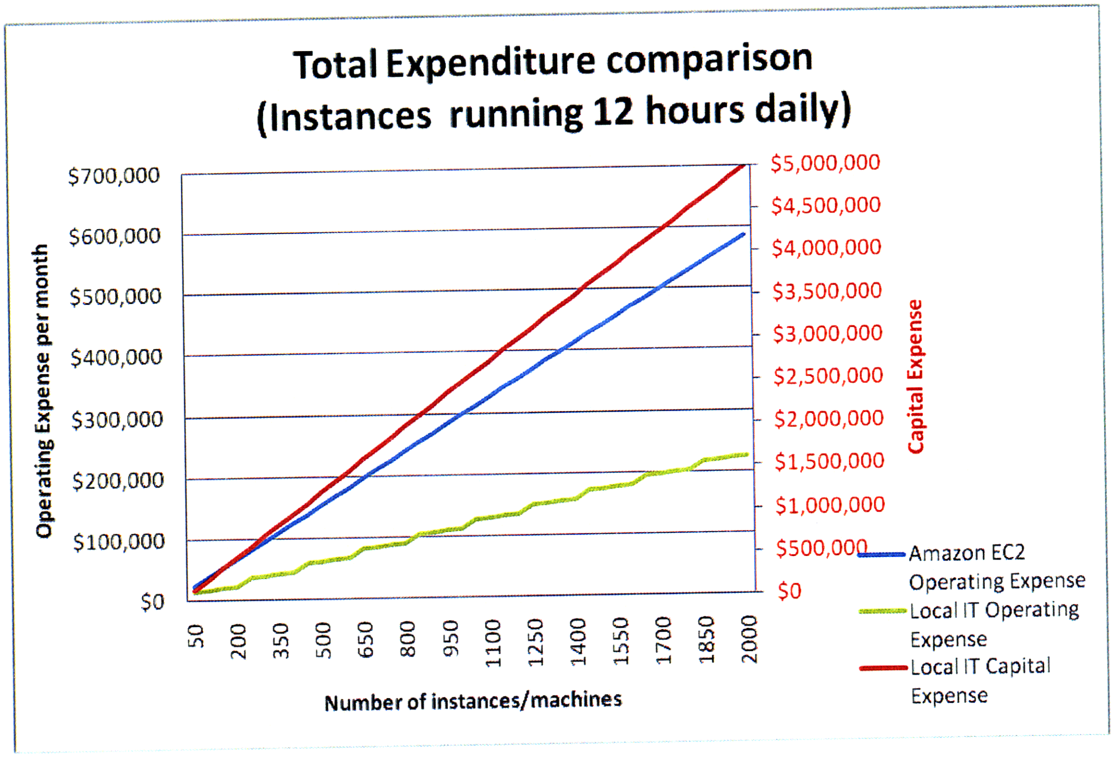


Figure 9 Total expenditure comparisons for cloud vs. owned computing resources

But the picture changes quickly as the usage of the cloud instances goes up. When usage was changed to 6, 8 or 12 hours per day, cloud computing operating expenses quickly surpassed local owned IT operating expenses and over a period of time caught up and even exceeding the capital expenditure for locally owned IT infrastructure.

Taking the model a step further and simulating real world usage with variable user load or resource requirement spread across a typical day as shown below.

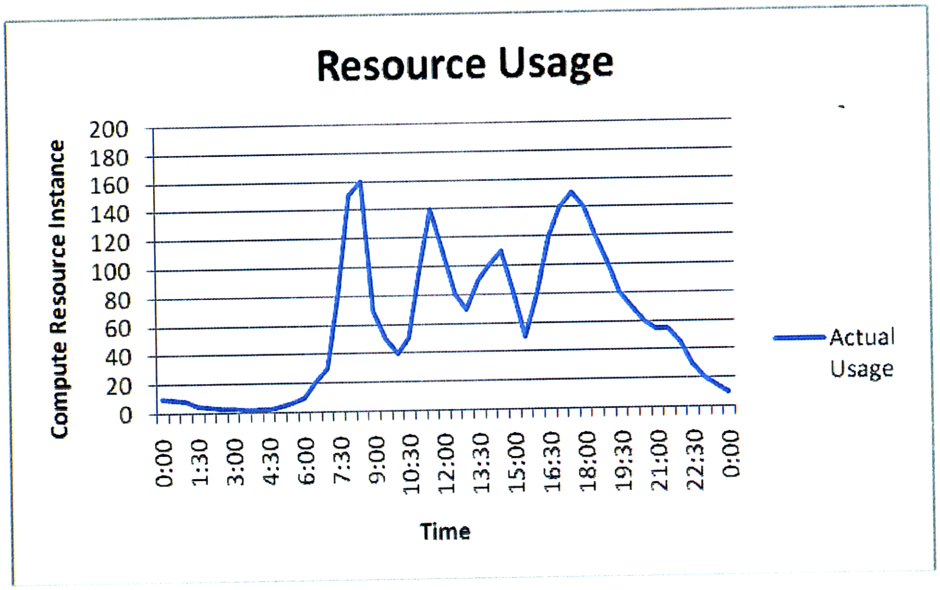


Figure 10 Compute resource usage simulation for a day

The compute resource usage chart shows number of computing instances used over the period of day. Resource usage is highest at the start of a business day, dropping sharply after that then resuming picking up around 12:00 in the noon, with maybe another peak around 15:00 in the afternoon and finally peaking at around 17:00-18:00 and falling slowly during the evening. This is a typical representation of compute resource utilized for a business application like a Customer Relation Management or sales related application. With local owned IT the infrastructure needs to be built for a maximum of 80% peak i.e. if the peak usage is 160 compute instances, the local IT infrastructure needs to have at least 200 compute instances at its disposal. Keeping that in mind, the simulation creates a local IT infrastructure with 200 machine instances. While the cloud model using Amazon instances can bring up and shutdown instances on demand it will accumulate charges only for the resources used. The graph for the cost based on simulated resource usage is below.

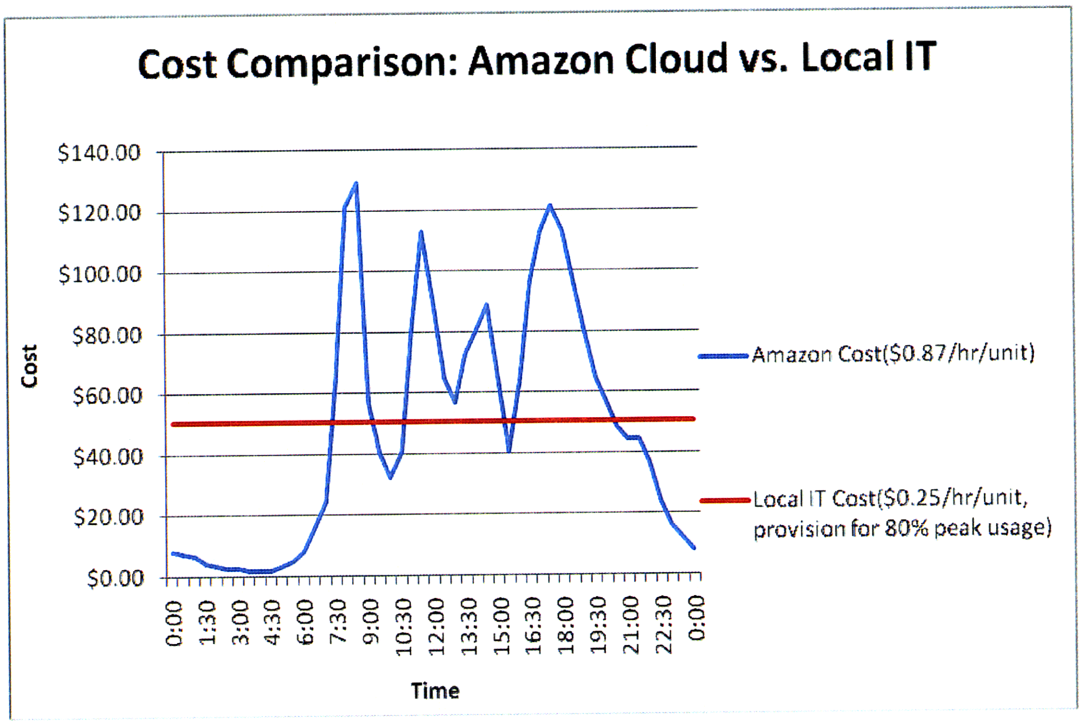


Figure 11 Cost comparison for real world usage simulation

While the cost for local owned IT infrastructure is fixed, since we have to build for a fixed 200 machine instances, the cost for Amazon cloud instances fluctuates with the usage pattern. The total cost accumulated over the period of a day show Amazon cloud resource costing less than locally owned IT infrastructure.

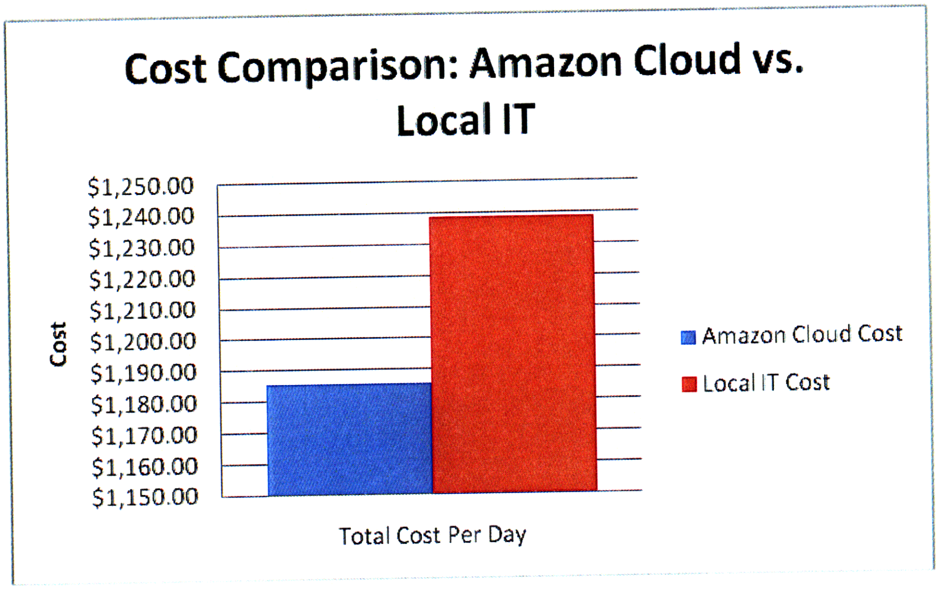


Figure 12 Cost Comparison: Amazon Cloud vs. Local IT

As shown in the simulation cloud resources can be cost efficient under conditions which see variations in resource usage. The elasticity offered under cloud services allows scaling up or down and thus allows minimizing cost incurred on the basis of actual consumption of computing resources.

A similar scenario was seen when comparing cost of SaaS based software to software installed and running on local IT infrastructure. SaaS based software afforded zero capital expenditure in terms of license cost or allocation of required hardware to run those applications. The results from the analysis point to cloud computing resources costing less in low consumption phases and can overshoot local IT cost if consumption of resources is high and very stable, does not have much variation.

This analysis considered only quantifiable metrics; other important factors such as ability to scale, other maintenance requirements like backups, crash recovery, downtime etc were not included in estimating cost for local IT infrastructure or in evaluating the alternatives. That said, the cost of cloud computing resources is predicted to fall further as the technology matures and there are more providers increasing competition and thus driving down cost.

3.1.3 Adoption

Cloud computing is in its early stages of adoption and has considerable time before it can gain the status of being the dominant mode of how computing resource are acquired and used. As the adoption of cloud computing grows there are major issues to be tackled before which enterprises will truly consider moving mission critical applications out to the clouds. Among them crucial are issues related to security and interoperability at the IaaS level. Adoption at the PaaS and SaaS level are higher as it is easier to plug into an application without worrying about how to manage your instance image or learning new scripts to bring up your instances.

SaaS is forecast to have a 23.8% compound annual growth rate through 2012 for the aggregate enterprise application markets, far exceeding the total market CAGR of 11.4%.

Predictions for SaaS growth project that by 2012 more than 33% of independent software vendors will be offering some of their applications as SaaS. By 2010, 15% of large companies will have started projects to replace their ERP backbone (including financial, human capital management and procurement) with new SaaS based solutions. A sampling of SaaS adoption number below shows roughly 17-23% adoption rate among data integration tools, data quality tools and data warehouse and business intelligence software. On premise hosting still shares the major chunk of software deployment.

Methods of Deploying Packaged Data Integration Tools	
Deployment Method	Percentage
Hosted on premises	74.1
Hosted externally	26.9
Internally managed	73.6
Outsourced	27.5
Software as a service (utility)	27.7
Other	0.2
Don't know	0.2
Number of respondents: 564	

Source: Gartner (September 2008)

3.2 Application of Frameworks

Different frameworks related to technology evolution and business models have been applied to determine the impact of cloud computing on enterprise software and enterprise software vendors.

3.2.1 Impact on Revenues

As the trend of using efficient cloud computing resources picks up the software usage model where software licenses are purchased and software installed on on-premise IT infrastructure will give way to off-premise software deployment with pay-per-use based pricing driven by Service Level Agreements (SLA). While a SLA driven model reduces overall cost for end users as the one time software license is broken down into a recurring operational cost based on usage the software vendors see just a small portion of revenue from each customer. Up until recently enterprise application have usually been a high margin sell to a select few customers and sales efforts were targeted towards these few customers who pre-selected on the basis of the license cost that they could afford. The revenue model, in essence, has shifted from high margin-low volume to low margin-high volume.

3.2.2 Technical and Product Impact for ESVs

With the trend of applications users shifting to cloud based computing resources applications itself will have to undergo major architecture shifts to be able to utilize the benefits derived from moving to the clouds:

- Product Metering: With conventional on-premise applications CPU usage, network data transfer, storage usage and other resources are not watched very closely unless an application is noticeably exceeding threshold usage. A few more CPU cycles, additional packets send back and forth, etc. do not increase the total cost of operations since most of these resources are in-house and more or less have a flat cost for usage. But once applications move to the cloud the customer ends up paying for each and every packet send back and forth to the cloud or for any addition CPU consumption that is required by the application. Storage for the application is paid on a metered basis too. This environment change will force application developers to look at their application and trim resource usage so that total operational cost to use their application is lower when these applications are moved to the cloud. This will require metering of resource built into products.

- Licensing & Interface to Billing Application: As the licensing model changes from one-time license purchase to pay-per-use, applications or their environment will need to meter application usage and if required interface with a billing system to bill the end user.

- Lock-in with particular platform or cloud provider: Cloud providers offer different level of proprietary API set. There is no standardization yet for cloud services as the technology is still in the early stages of its life cycle. Using cloud services requires either using a custom application from a SaaS provider with few available customization options, or using a platform with proprietary APIs offered by PaaS providers, or learning new custom scripts and tools to manage instances on a cloud where these instances cannot be easily migrated to other clouds or local clouds. Even though most of the cloud providers use open source software to build and run their cloud services that alone does not guarantee their systems being interoperable. Thus when software vendors choose a particular cloud option, be it SaaS or IaaS, at this stage, it means locking into some form of custom/proprietary interface that is not easily portable across different cloud providers. This is especially tricky for enterprise software vendors, who already have a large customer base and have to carefully choose their SaaS offerings so as to not lose their loyal customer base. To avoid getting stuck with the wrong choice ESVs should hence look to offer a portfolio of SaaS offerings instead of laying all bets on a single platform. For start-ups there might not be a question for lock-in if they desire to use the distribution channels already created by these cloud service providers and so it might be justifiable for them to lock-in to a particular platform. This issue has given rise to a bunch of tools that enable interoperability across various cloud offerings.

- Scalability: Running under a different model, in most cases running as a service the software will need to be architected so that it scales within an environment that allows it to scale making use of a much larger set of computing resources.

- Security: Applications running on the cloud will face any of the security risk faced by any externally run service and is open to security threats in the form of virus attacks and break in by hackers. Also since the cloud will usually have access to private data stored by customers hackers getting into the system will immediately have access to a large treasure chest of private data. In some cases application users might have issues in the first place with private data being managed by a third party, i.e. the cloud provider. Applications will thus have to be aware of such issue and application would have to be designed so that the architecture allows precautionary steps to protect the data as best as it can.

- Failover: With the new cloud business models applications will need to be aware of the different failover features available on the cloud that it is running on and be able to autonomously handle failover without little or no human intervention. If running under a SaaS model it then becomes application provider's responsibility and thus the application's duty to manage failover.

3.2.3 Impact on Costs

The cost of developing or porting application software for cloud computing has the benefits of not requiring large hardware investment or any software license purchases. The barrier to develop and deploy software is very low and so software vendors can put out test packages out more easily and more frequently. Moving to developing on the cloud allows developers to collaborate in new ways and improves code reuse. Though at this stage, there are is no single standard means of developing software for the cloud consensus develop in the coming years.

Other than developing the major cost for software developers comes from distribution, support and maintenance. The SaaS and IaaS models helps reduce support and maintenance cost by making applications multitenant. Updates to the software can almost be delivered immediately for all customers with the single update of the cloud serving the SaaS application.

3.2.4 Business Impact

ESV will also need to look at other business impacts resulting from migrating to a cloud computing model. As seen with the product changes resulting from this migration to cloud platforms there can also be legal issues with the new SLA replacing old licensing. ESVs will own the cloud applications and as such be responsible to safeguard and secure user data, abide by privacy laws related to user data and be held responsible in case of loss of this data. If these applications are used by industries that have their information regulated, like the financial industry, then the ESVs working together with the cloud provider would be responsible to ensure compliance with industry regulations.

3.2.5 Theory of Disruptive Innovation (Christensen & Raynor, 2003)

The theory of disruptive innovation states that the generally when a disruptive innovation is introduced due to circumstances and the drive to maximize profits existing well run incumbents ignore this disruption. This leads to the disruptor slowly eroding the market share and market of the incumbent and the incumbent often fails because by the time the incumbent realizes the disruptor has already overtaken the incumbent.

With the initial computing resource market focused on processing speed as the key to acquiring market share PC/desktop, mainframes and the grid/cluster manufacturers constantly worked along the lines of improving processing speed and increasing input/output. Cloud computing started came in as a new market disruptor entering with the low end web applications that required very little processing and did not matter if it failed like web based email service, online media sharing applications etc. As broadband connectivity

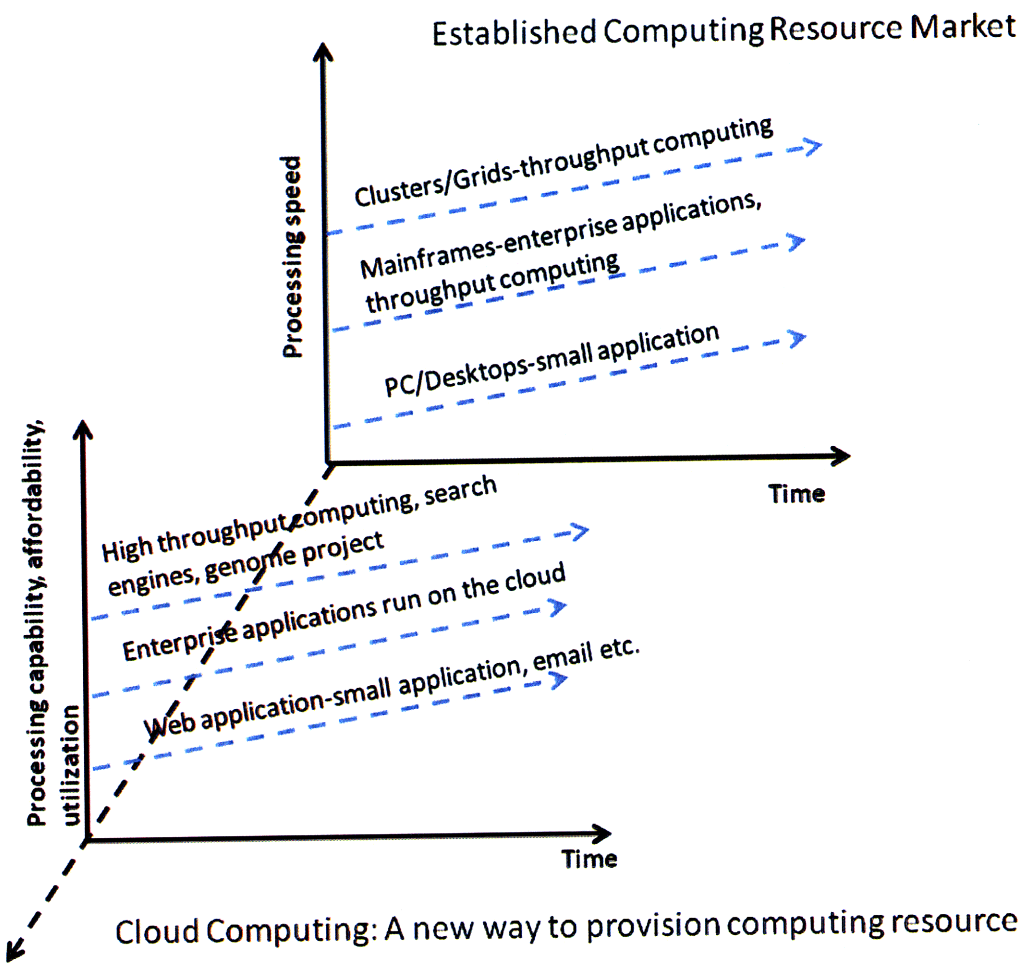


Figure 13 Cloud computing, a new market disruption

improved and technology to scale web applications developed more serious enterprise grade applications are being served by cloud computing. At the high end process that require huge computation resources and throughput intensive jobs like running an internet search engine, storing and managing the human genome data etc are being targeted to run on the cloud. Thus we see that cloud computing has disrupted the established computing resource market in terms of both hardware as well as software distribution chains.

Looking back at history we see patterns repeated with the introduction of a new disruptive technology or disruptive business model. Take for example the disruption of railway transportation around the 1850s. It impacted everyone in the canal boat industry and the ecosystem surrounding transportation by canal boats. Previously, as the canals were the arteries and acted as the life-line to towns and cities that sprung up around these canals saw rapid decline once the railways were adopted as mass transportation for goods and people. The canals and canal boats were functioning perfectly well and the industry had healthy profits. This leads to few insights:

- It is not the failure of the incumbent technology/business model that leads to the incumbent losing market share; the consumer market just adopts the new technology or business model that is cheaper, more efficient and more easy to use.

- When faced by a disruptive technology or disruptive business model the incumbent often is left off-guard and once the markets move quickly falls into a death spiral. Only the very dynamic and responsive players will be able to react in time to either challenge the disruption or adopt it.

With these lessons in mind looking at the cloud computing, as a disruptive technology and business model it promises to impact the software distribution and surrounding ecosystems. Any business that makes use of computing resources and software applications is bound to face the decision of continuing its legacy IT systems or migrate to more cost effective

Of late some of the enterprise software vendors realizing the impact that clouds can have on their businesses have jumped into the bandwagon and have started their own cloud services and ported their applications to run on their cloud. Microsoft Azure is one such example which has been a late entrant and finally started its own cloud services that run only Microsoft software on its cloud.

3.2.6 Clock Speed and the Double Helix Industry Movement (Fine, 1998)

According to Charlie Fine industries tend to evolve cyclically from being integrated to modular and back to being integrated. When an industry is modular, technological advances, supplier market power and higher profitability from proprietary systems pressure the industry to integrate and firms to serve across all layers of the value chain thus becoming vertical. Once the industry is dominated by integrated players, rising complexities due to integration, organizational rigidities and competition from niche players pressures the industry back to modular. This cycle repeats itself at a pace unique to each industry based on the speed at which technological and business model innovation dissipate in that industry.

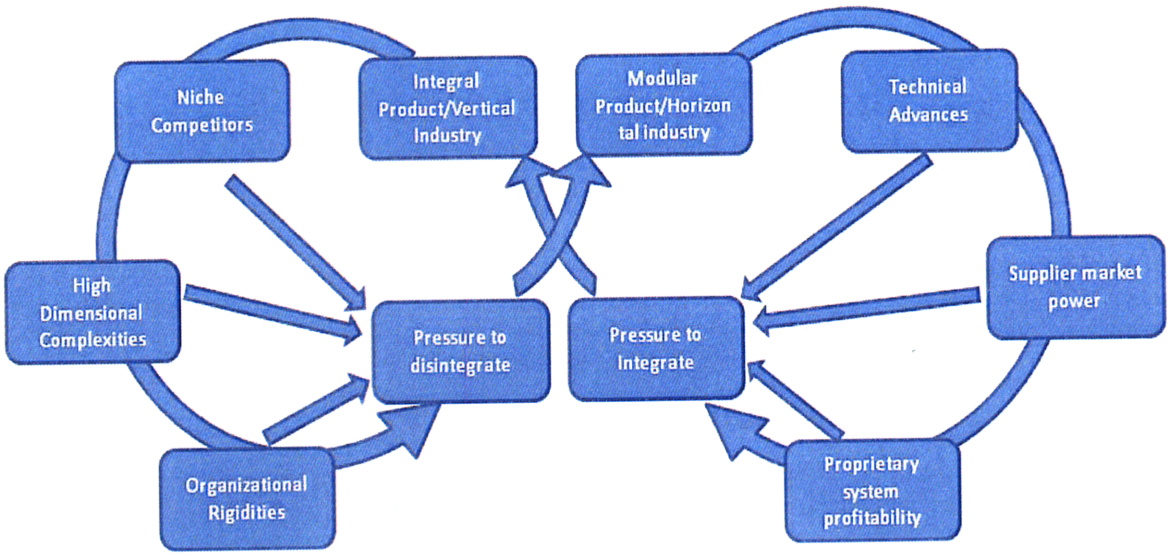


Figure 14 The Double Helix – Industry Movement

Applying the double helix movement to the enterprise software ecosystem we find that the current state of legacy enterprise software is in a highly integrated state. This has resulted in highly segmented enterprise software where applications from one software vendor do not talk to applications from other vendors or at least it is quite complex and difficult to enable such interaction with each other.

This rate at which industries evolve is termed as the clock-speed of that industry and is in some way dependent on three sub metrics i.e. product clock speed, process clock speed and organizational clock speed. The higher the clock speed the higher will be the rate at which the industry evolves through the double helix.

Plotting the clock speed for the enterprise software ecosystem we see that the typical clock speed for equipment maker that is used to manufacture chips is three to six years. That is the typical life cycle period of such equipment/technology is three to six years. Similarly, the clock speed for a chip manufacturer is around two to four years. A PC manufacturer perhaps introduces new products every four to six months. At the extreme downstream is a Web site that possibly has a very short life cycle of maybe weeks or days.

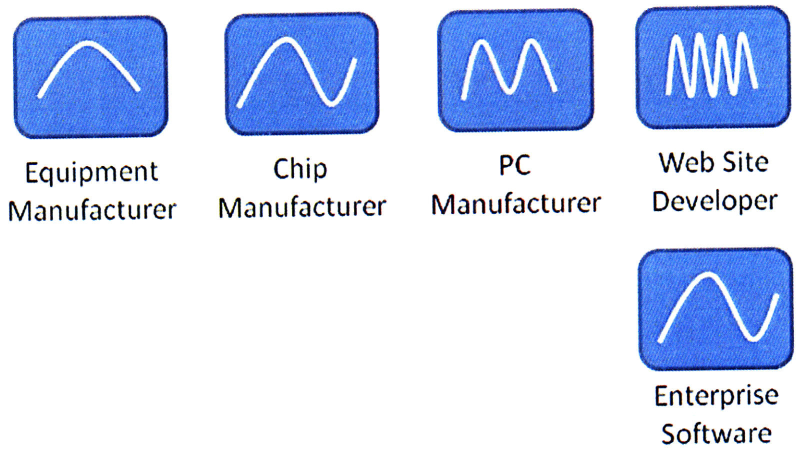


Figure 15 Clock Speed for Industries in the Software Value Chain

While clock speed increases as we move downstream closer to the end customer we find that the clock speed for enterprise software is comparatively higher i.e. in the range of two to four years.

Next we draw the value chain for enterprise software with and without cloud computing in play and try to determine the impact on enterprise software clock speed that cloud computing will bring about.

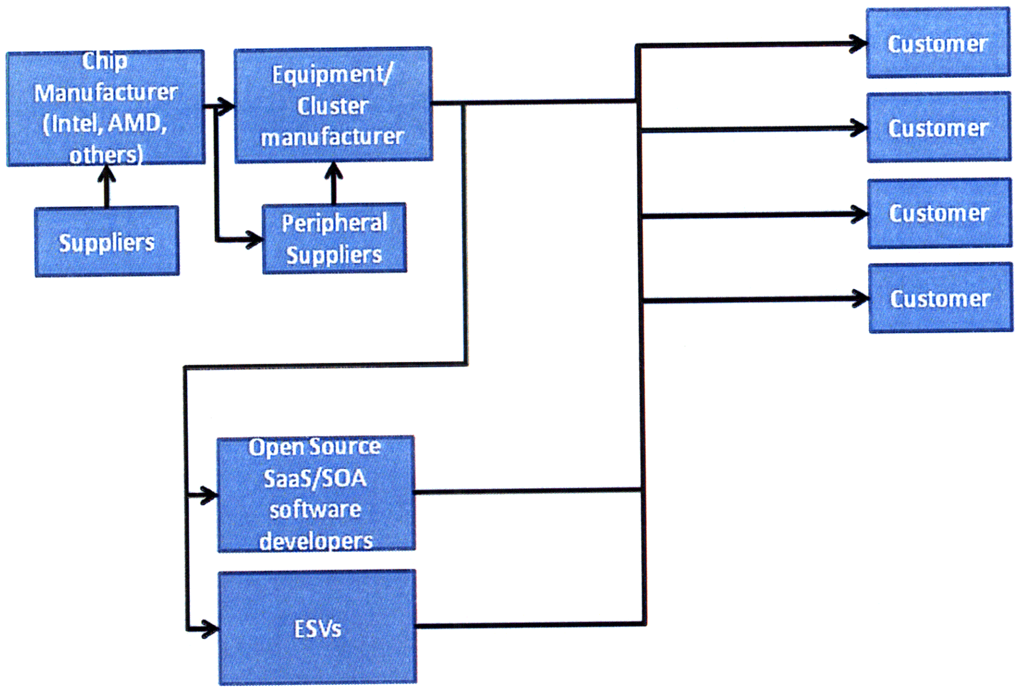


Figure 16 Enterprise Software Value Chain – Legacy

Looking at the enterprise value chain before cloud computing we see that the end customer is dependent on enterprise software vendors, open source developers as well as dependent on equipment manufacturer which tends to slow down the clock speed for end user applications due to multiple dependencies. Open source software typically has a higher clock speed compared to enterprise software due to the development model of open source and due to the amount of resources that the open source community has access to. Enterprise software vendors on the other hand, due to the relatively more limited resources that they have at their disposable have a lower clock speed. Typically, a new version of enterprise grade software product is released in two to four years. Open source communities on the other hand typically throw out patches or new versions much more frequently.

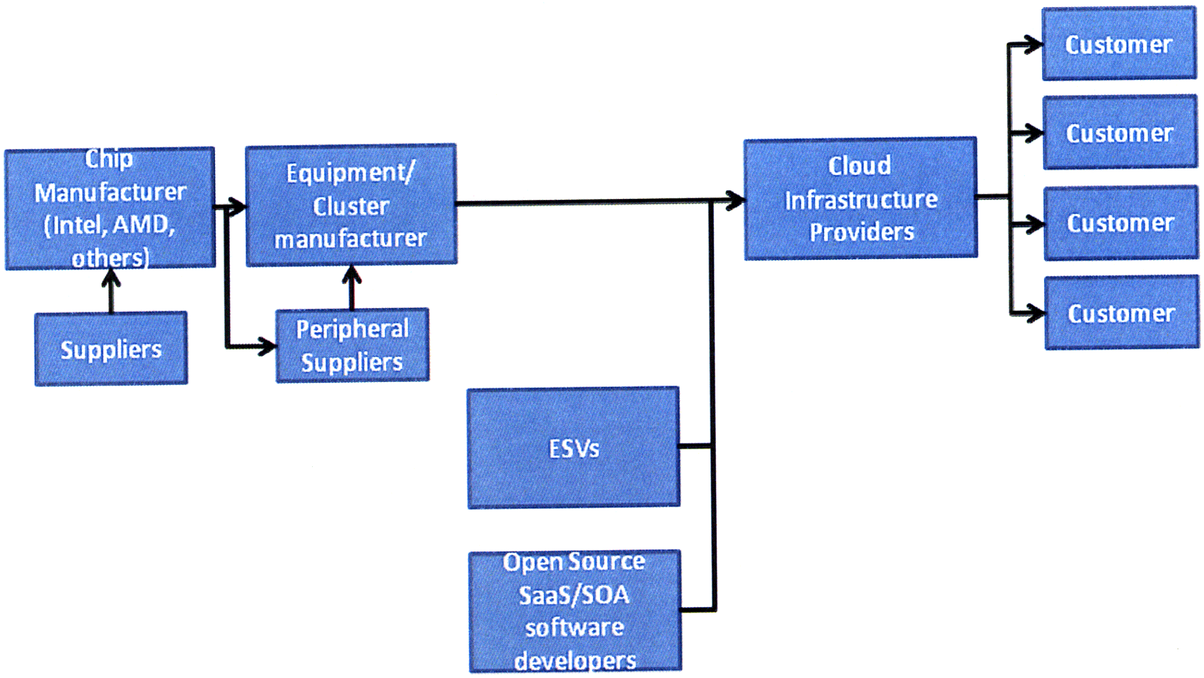


Figure 17 Enterprise Software Value Chain – With Cloud Computing

With cloud computing is in play the end user is not dependent on hardware manufacturers to cycle through their hardware systems. They are only dependent on the software providers and hardly care about upgrading the underlying hardware infrastructure ever so often. The hardware dependency for end users that holds back upgrades no longer exists, increasing pressure on the application developer front to increase clock speed and provide shorter periods between successive product/service introductions. While open source and applications migrated to a SaaS model (web applications) will quickly be able to meet this demand of higher clock speed enterprise software vendors would be challenged to catch-up.

The key insights from applying the double helix/clock speed/value chain framework are

- Cloud computing will become the key software provisioning platform. Driven by more efficient and cheaper means of software provisioning this channel will grow and replace in most cases the old legacy way of buying software licenses and running on local computing resources. As enterprises realize the benefits of cloud computing and migrate to cloud platforms, network effect will catapult cloud platforms as the means of delivering and running software.

- Enterprise software vendor will be pressured to increase their clock speed as a result of the shift towards cloud computing. Enterprises can increase their clock speed by either

    - Adding more resources to the development of software applications, which increases development cost and thus cost of applications to end users. Resource can also be added by farming out development to online communities thereby increasing the resource pool and sharing profits from products with the online community of application writers. This is similar to the model that Apple has used to develop iPhone applications. Developers are provided the development tools and infrastructure to develop and can make their applications available to end users through Apple's iTune store. The developers receive 70% of the revenue from the sales of their applications sharing 30% of the revenue with Apple. A similar environment can be created to help enterprises develop applications and online community developers are rewarded on the basis of usage and quality of the modules they code.

  - Write applications in a more efficient manner by increasing reuse of already coded modules. This in turn requires that application architecture be modified to models that support reuse of code following Service Oriented Architecture (SOA) or Web Oriented Architecture.

- Enterprise software industry moving in the double helix to becoming a more modular industry. This indicates the overall software application industry breaking down from vertical silos to modular components that are interoperable modularization and customization by mixing and matching various components to build the final applications.

4. Recommendations

Following the analysis of cloud computing and its impact on ESV here are recommendations for ESVs to help them navigate through this technological innovation and emerge as players that not just survive this next disruption but use cloud computing to their own advantage. Cloud computing is disruptive to the existing technology of software application provisioning and the existing enterprise software business model and creates new software distribution channels. ESVs need to look at cloud computing seriously and adopt these new distribution and business model as their customers adopt cloud computing. ESVs should exploit these new technologies to their benefit, bring efficiencies in usage of computing resource utilization, reduce cost of operating their applications and thus help retain their customers. Below are recommendations that ESVs can adopt to align themselves along the cloud computing trend and in a position to be able to leverage cloud computing to their benefit.

4.1 Product Recommendations

Enterprise software vendors need to adapt their products to be able to deploy and run on cloud platforms and utilize the SaaS based architecture to improve computing resource utilization. As mentioned previously, while products can be run on a basic IaaS platform, this approach would not derive all the possible benefits from a PaaS or SaaS model. The product architecture also needs to change to be able to deliver applications as a service:

- **Migrate legacy applications to SaaS platforms**: While it is recommended to start developing application for a SaaS delivery model from the grounds up analysis needs to be done on cost related to development, maintenance, support and time-to-market to decide if they favor porting existing applications in the short run. The two approaches to migrate to using SaaS based delivery models are

    - **Port existing applications**: Decompose and repartition application to adopt a service oriented architecture (SOA). These services can then be used to deliver web services applications to the end user. This approach can help to quickly move to SaaS based delivery. In the long run solutions from this approach could prove costly due to maintenance cost and possible underlying architecture restrictions that does not allow multi-tenancy both at the application and database level.

    - **Start fresh**: Start from scratch and develop required functionalities using an available platform to quickly develop a SaaS version of your application. In this case firms use their domain knowledge to develop new SaaS based applications. All other complexities like multi-tenancy, scaling, versioning, security etc. are handled by the platform or can very easily be provided as external services that the application can use. Applications developed from the grounds up will be able to easily incorporate multi-tenancy both at the application and underlying database level (Multi-tenant data architecture) to squeeze the benefits of SaaS.

- **Build migration utilities for applications users to migrate to SaaS**: Build utilities that help application users migrate to the SaaS offering of the application. This would involve developing utilities to migrate information/databases in legacy form that resides within the users IT infrastructure from an isolated to a shared multi-tenant data architecture supported by the SaaS based application and tools to help guide users get started and customize SaaS based application interface.

- **Offer a portfolio of SaaS solutions**: Build SaaS applications with metering/billing built into each logical functionality unit so that one can offer services at a lower granularity. Offer various levels of service along the line of different logical functionalities and different market segments at different pricing levels. This will allow customers to pick and choose and customize their SaaS experience and can cover most of their existing customer base. ESVs would likely not have enough leverage to impose a single SaaS model on their entire existing customer base.

- **Build additional security into application**: Additional security needs to be built into the applications running in a cloud environment since cloud computing supports multi-tenancy and multiple clients share the same instance of the running application and underlying resource. The software application partitions the data and interface so that each client sees a customized virtual instance of the application. Security can be added in the form of stronger authentication, encryption of user data in database so that even if another user gets to someone else's data they cannot use it and data redundancy to use in case of failures.

- **Build metering into applications to optimize for efficient use of computing resources:** In the cloud environment applications would have control over and would need to get and release computing resources efficiently so that resources are not held when they are not needed. Applications should meter and optimize use of
    - CPU
    - Storage
    - Data transfer
- **Adopting open source/network centric development models**: Analyzing the clock speeds for players down the value the chain of enterprise software it is clear that open source with their larger human resource pool and distributed development model can spin out software at a much faster pace. Supported by cloud computing resources that

are available on the cheap and being able to quickly develop and deploy applications without much investment in distribution open source software is going to become a bigger challenge to ESVs. ESVs need to utilize some of the collaborative software development methods supported by cloud computing and utilize the online community resource help develop application for the ESVs. Tapping into the online community of developers will help ESVs to innovate at a much faster rate, decentralize their development efforts and be able reduce their turnover time (higher release rates).

With added benefits of collaboration in cloud computing projects, network centric models can be advantageous, specially to develop large scale enterprise application. The model tracks Network Leadership from centralized to diffused and Innovation Space from defined to emergent on a 2x2 matrix. ESVs can use the Creative Bazaar model so that they have control over what is being developed but at the same time look at emerging technologies and capture these emerging features in their application. If building their own public cloud infrastructure they can use the MOD station model that utilizes defined set of hardware resource sourced from various external hardware vendors and assembled/modified to provide maximum return on capital investment.

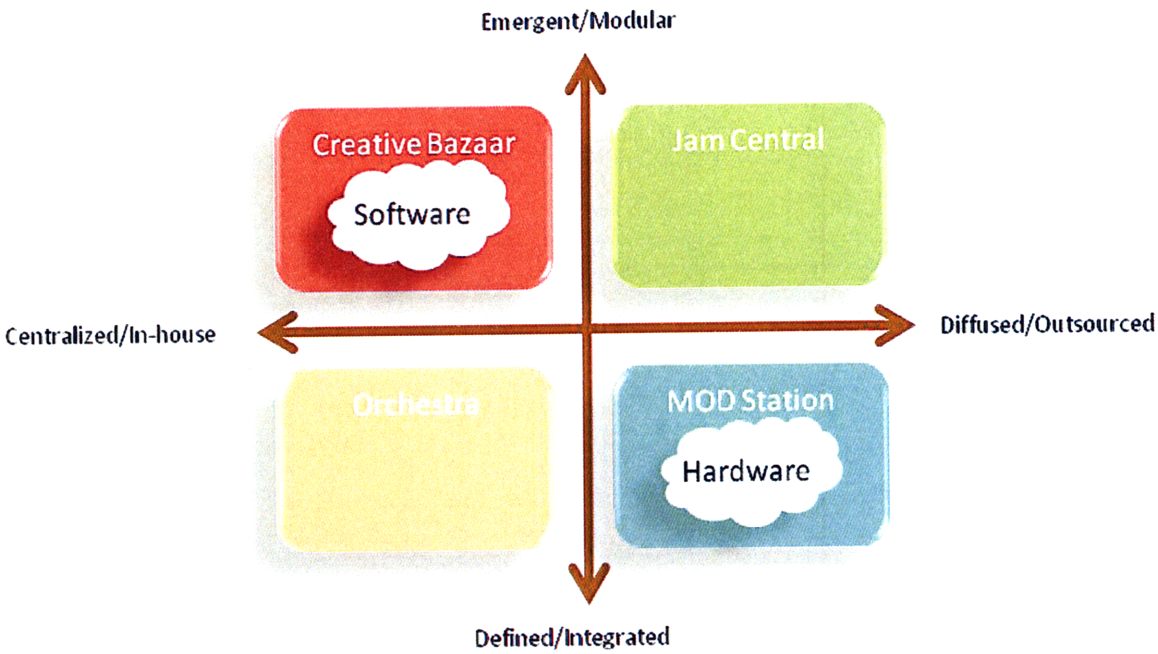


Figure 18 Network Centric innovation model

4.2 Business Recommendations

In addition to product changes ESVs would also have to align their business processes along the new business models supported by cloud computing. These new business models promote use of computing resources, including hardware and software, as utilities, i.e. on a pay-per-use basis. Below are recommendations for ESVs to follow along the product recommendations to fully align their businesses to make the best of the cloud computing trend.

- **Partner with cloud providers**: As seen in the value chain cloud providers have inserted themselves in the value chain between ESVs and application users. Public cloud providers are now a means of provisioning/distributing software applications. While maximum value, and thus revenues, can be captured by ESVs if they themselves roll out and offer public clouds to run their application, they might not hold the expertise to build and run a public cloud and also it might not be the best use of limited resource they own. Hence, ESVs need to partner with cloud service providers to be able to provision their applications.

- **Determine optimal SaaS Pricing**: ESVs will need to determine competitive pay-per-use pricing for their offerings. The pay-per-use or subscription model pricing would be determined by various factors cost factor like cost of development, maintenance and support and cost of hosting the application on a cloud. The pricing should also take in consideration the existing license pricing, i.e. the cost for a customer to use the SaaS offering should be comparable, if not less, to the cost incurred using the license model.

- **Frame an acceptable Service Level Agreement (SLA)**: When offering their applications as a service licenses are replaced by a Service Level Agreement (SLA) which determines the service quality, priorities, responsibilities, guarantees and warranties. Special attention should be given to the SLA so that it does not promise anything more than the application can provide and also does guarantee a higher level of service than offered by the underlying cloud provider where the application is running.

- **Restructure sales incentives**: Sales organizations are arranged along strong incentives based on revenues which could be easily computed based on lump sum license and maintenance revenues. In the SaaS model, revenues are spread across a period of time in the form of subscription or pay-per-use. Hence sales incentives too would have to spread across a period of time determined by customer retention.

5. Risks & Risk Management

5.1 Risk

There is considerable risk going forward and betting on a particular cloud computing model. Once ESVs adopt a particular cloud offering they effectively tie the success to their software distribution and customer retention to the success of that platform. The major risks include:

- **Lower cloud adoption rates**: The rate of cloud adoption might be slower than expected, which will make returns on cloud initiatives longer to recover

- **Regulations**: Geo-political, regulation and security issue might prevent or complicate usage of clouds with data warehouses outside geographical limits thus requiring a slightly different approach for each geographical region where the ESVs sell their applications.

- **Hacker/Virus attacks**: Centralized resources makes cloud an easier target for attacks, virus or other forms of attacks on a data warehouse can easily cripple a major chunk of services. ESVs will have usually locked in to a particular cloud service and platform to serve their applications and attacks or downtime would impact their customer base that uses their applications.

- **Uncharted waters**: Though cloud providers guarantee certain uptime under their SLAs their failover procedures under high volume usage are as yet mostly untested in real life situations.

5.2 Risk Management

ESVs can take the following precautionary steps to mitigate some of the new risks brought in by moving to cloud computing.

- **Get customer feedback, offer trial/beta versions**: Carefully analyze the business problem being solved by their applications, how well it fits in the cloud computing model, who is the customer and if the customer accepts a multi-tenant solution. Involve the customer and other partners in the ecosystem to share and reduce risk. ESV's can start offering beta versions of SaaS applications to willing customers to collect feedback from customers.

- **Approach the cloud transition on a step-by-step basis**: ESVs can start by first moving applications to a service oriented or web oriented architecture. This will help them provide components of their application suite as a service without interfering with existing usage. Once they have their complete application ready to be offered as a service they can segment their customer base and move a segment of their customers at a time to the SaaS model.

- **Build interoperable SaaS applications**: Build SaaS applications that are interoperable on target cloud platforms as well as internally provisioned services. With the trend of enterprises using a hybrid cloud solution, i.e. owning a small internal cloud infrastructure and outsourcing the rest of IT requirements to cloud providers enterprises might want to run some components of applications within their local cloud and the rest on public clouds. Making applications interoperable on different public clouds also prevents getting locked-in to a single cloud provider and allows enterprises to choose public cloud providers on the basis of performance and cost.

- **Select a cloud provider very carefully**: Investigate the cloud provider on the basis of availability of the offerings, number of references, relevant growth and financial stability of the provider and the service level agreement (SLA) offered.

Appendix: Terminology

EUCALYPTUS	Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems
Hypervisor	A virtualization platform that allows multiple operating systems to run on a host computer at the same time. Hypervisors are currently classified in two types: A **Type 1** (or native or bare-metal) hypervisor is software that runs directly on a given hardware platform (as an operating system control program). A guest operating system thus runs at the second level above the hardware. The classic type 1 hypervisor was CP/CMS, developed at IBM in the 1960s, ancestor of IBM's current z/VM. More recent examples are Oracle VM,VMware's ESX Server, LynxSecure from LynuxWorks, L4 microkernels, Green Hills Software's INTEGRITY Padded Cell, VirtualLogix's VLX, TRANGO, IBM's POWER Hypervisor (PR/SM), Microsoft's Hyper-V (released in June 2008), Xen, Citrix XenServer, Parallels Server (released in 2008), ScaleMP's vSMP Foundation (released in 2005) and Sun's Logical Domains Hypervisor (released in 2005). A variation of this is embedding the hypervisor in the firmware of the platform, as is done in the case of Hitachi's Virtage hypervisor. KVM, which turns a complete Linux kernel into a hypervisor, is also Type 1. A **Type 2** (or hosted) hypervisor is software that runs within an operating system environment. A "guest" operating system thus runs at the third level above the hardware. Examples include VMware Server (formerly known as GSX), VMware Workstation, VMware Fusion, the open source QEMU, Microsoft's Virtual PC and Microsoft Virtual Server products, Sun's (formerly InnoTek) VirtualBox, as well as SWsoft's Parallels Workstation and Parallels Desktop. Pasted from <http://en.wikipedia.org/wiki/Hypervisor>
PaaS	Platform as a Service
HaaS /IaaS	Hardware as a Service/Infrastructure as a Service

SaaS	Software as a Service
SLA	Service Level Agreement in cloud computing context, Software License Agreement
CDI	Cloud Desktop Infrastructure, desktops in the cloud
VDI	Virtual Desktop Infrastructure
KVM	Kernel based Virtual Machine - is a Linux kernel virtualization infrastructure. Redhat and Ubuntu using KVM hypervisor
VPC	A VPC is a method for partitioning a public computing utility such as EC2 into a quarantined virtual infrastructure. A VPC may encapsulate multiple local and remote resources to appear as a single homogeneous computing environment allowing you to securely utilize remote resources as part of a seamless global compute infrastructure. Pasted from <http://www.enomaly.com/FAQ.402.0.html#q10>
Cloud Bursting	Cloud Bursting allows you to automatically scale to sudden and extreme spikes in demand by enabling a hybrid cloud computing model which combines both private data center resources and remote cloud resources such as Amazon EC2 Pasted from <http://www.enomaly.com/FAQ.402.0.html#q10>
SMP	Symmetric Multiprocessing involves a multiprocessor computer-architecture where two or more identical processors can connect to a single shared main memory. Pasted from <http://en.wikipedia.org/wiki/Symmetric_multiprocessing>
Multi-tenant	A single instance supporting multiple users. This can be in context of an application where a single application instance supports multiple users or in context of a database where a single database supports multiple user data partitioning data so that each user sees only his relevant data.

Bibliography

[1]. **Willis, John M.** Cloud Computing. *IT Management and Cloud Blog.* [Online] December 2008. http://www.johnmwillis.com.

[2]. **Weinman, Joe.** Complex Models. *Complex Models.* [Online] November 2005-2008. http://complexmodels.com.

[3]. **Wayner, Peter.** Cloud versus cloud: A guided tour of Amazon, Google, AppNexus, and GoGrid. *InfoWorld.* July 21, 2008, pp. 1-8.

[4]. **Valdes, Ray.** *Cloud-Based Application Development Platforms Enable New Modes of Collaboration.* s.l. : Gartner, 2008. G00163244.

[5]. **UCSB.** Eucalyptus. *Eucalyptus.* [Online] http://eucalyptus.cs.ucsb.edu.

[6]. **Sawhney, Satish Nambisan & Mohanbir.** *The Global Brain.*

[7]. **Perry, Geva.** How Cloud and Utility Computing Are Different. *GigaOm.* [Online] February 2008. http://gigaom.com/2008/02/28/how-cloud-utility-computing-are-different.

[8]. **Naone, Erica.** Opening the Cloud. *Technology Review.* November 06, 2008, p. http://www.technologyreview.com/web/21642/page1/.

[9]. **Mosso.** Mosso: The Rackspace Cloud. *Mosso.* [Online] http://www.mosso.com.

[10]. **Jay Heiser, Mark Nicolett.** *Assessing the Security Risks of Cloud Computing .* Stamford : Gartner, 2008.

[11]. **Ignacio Martín Llorente, Ruben S. Montero, José Luis Vázquez-Poletti.** DSA Research Group. *DSA Research Group.* [Online] 2008. http://www.dsa-research.org.

[12]. **Hoff, Todd.** High Scalability. *High Scalability.* [Online] January 2009. http://highscalability.com.

[13]. **Hippel, Eric Von.** *Democratizing Innovation.* Cambridge : The MIT Press, 2005.

[14]. **Flixiscale.** Flexiscale. [Online] http://www.flexiscale.com.

[15]. **Fine, Charles H.** *Clock Speed - Winning Industry Control in the Age of Temporary Advantage.* Reading : Perseus Books, 1998.

[16]. **Duxbury, Bryan.** Rent or Own: Amazon EC2 VS. Colocation Comparison for Hadoop Clusters. *Engineering Rapleaf.* [Online] December 2008. http://blog.rapleaf.com.

[17]. **Cohen, Rueven.** ElasticVapor :: Life in the Cloud. *ElasticVapor :: Life in the Cloud.* [Online] December 2008. http://www.elasticvapor.com.

[18]. **Christensen, Clayton M. and Raynor, Michael E.** *The Innovator's Solution.* Cambridge : Harvard Business School Press, 2003.

[19]. **Asay, Matt.** CIOs sick of enterprise software pricing, Forrester finds. *CNET News.* [Online] February 6, 2008. http://news.cnet.com/8301-13505_3-9866337-16.html.

[20]. Wikipedia. *Wikipedia.* [Online] 2008. http://en.wikipedia.org.

[21]. **Nirvanix.** Nirvanix Home Page. *Nirvanix.* [Online] December 2008. http://www.nirvanix.com.

[22]. **Nimbus.** Nimbus - Open Source Cloud Toolkit. *Nimbus.* [Online] December 2008. http://workspace.globus.org.

[23]. **Frederick Chong, Gianpaolo Carraro, and Roger Wolter.** Multi-Tenant Data Architecture. *Microsoft MSDN.* [Online] Microsoft, June 2006. http://msdn.microsoft.com/en-us/library/aa479086.aspx.

[24]. *Gartner Says Cloud Computing Will Be As Influential As E-business.* Stamford : Gartner, 2008.

[25]. **Gartner.** Gartner Identifies the Top 10 Strategic Technologies for 2009. *Gartner.com.* [Online] October 12-16, 2008. http://www.gartner.com/it/page.jsp?id=777212.

[26]. **Salesforce.com.** Force.com AppExchange. *SalesForce.com.* [Online] December 2008. http://developer.force.com.

[27]. Enomaly - Elastic Computing. *Enomaly - Elastic Computing.* [Online] http://www.enomaly.com.

[28]. **Yefim V. Natis, Nicholas Gall, David W. Cearley, Lydia Leong, Robert P. Desisto, Benoit J.Lheureux, David Mitchell Smith, Daryl C. Plummer.** *Cloud, SaaS, Hosting and Other Off-Premises Computing Models.* s.l. : Gartner, 2008. G00159042.

[29]. **Daryl C. Plummer, Thomas J. Bittman, Tom Austin, David W. Cearley, David Mitchell Smith.** *Cloud Computing: Defining and Describing an Emerging Phenomenon.* Stamford : Gartner, 2008.

[30]. **Thomas Otter.** *Cloud Computing and HR.* s.l. : Gartner, 2008. G00157858.

[31]. Bungee Connect - Platform as a Service. *Bungee Connect.* [Online] 2008. http://www.bungeeconnect.com/.

[32]. **Jay Heiser, Mark Nicolett.** *Assessing the Security Risks of Cloud Computing.* s.l. : Gartner, 2008. G00157782.

[33]. **AppNexus.** AppNexus Home Page. *AppNexus.* [Online] December 2008. http://www.appnexus.com.

[34]. **Amazon Web Services LLC.** Amazon Elastic Compute Cloud. *Amazon Web Services.* [Online] 2008. http://aws.amazon.com/ec2/.