Migration On an On-Premise Applications Using DC System To The Cloud

S. Revathi[1], Dr. G. Arutchelven[2]
Department of Computer Science and Application
Adhiparasakthi College of Arts and Science, Kalavai, Tamil Nadu

Abstract: Cloud computing has recently became a wide mentioned topic within the IT business. More and more organizations think about using the Cloud, as a result of it allows a simple and value economical approach of hosting applications, among dynamic scaling and geographical distribution potentialities. Existing application are usually written during an approach that doesn't extremely match a cloud surroundings well. Also, bound quality attributes (e.g. performance, security or portability) are often affected. More studies are required on however existing systems ought to be blocked into the Cloud and what are the implications of the migration. This thesis aims to share expertise associated observations we have a tendency to gain from adopting cloud computing for an on-premise enterprise application during a context of a little code company. Our study produces many valuable results. First, cloud computing opportunities and challenges. Second, cloud platforms were studied and image was developed supported the prevailing system. Finally, this image was used to evaluate the behaviour of comparable system in 2 environments (on-premise and also the Cloud) and beneath totally different conditions within the cloud, addressing such concern as performance and cost.

Keywords: cloud computing, public cloud platform, migration, and enterprise application

.

1. INTRODUCTION

Cloud Computing refers to a utility-based provisioning of virtualized process resources over the net. Albeit computing as a utility isn't a brand new term [1], it became commercially accessible due to recent technological shifts in virtualization, distributed computing and communication technologies. Cloud computing has become a brand new promising trend of the IT business that's getting ready to amendment the approach process resources and software system are designed and purchased. Market-research firm IDC estimates the marketplace for public cloud product and services growing to $42 billion by 2012[3], whereas strategy house AMI-partners predicts that little business payment a cloud computing can hit $100 billion by 2014[4].

The main intention of this work is to analyze what are unit the results of the migration; we have a tendency to perform our study on the instance of existing enterprise industrial application that's delineated

1. A close study of the advantage and also the disadvantages of cloud computing, and also the effects of migration of AN on-premise application into the cloud.

2. AN analysis of existing public cloud platforms so as to create a rational alternative of a particular one appropriate for our purpose.

3. The performance of experiments on the cloud version of our application. Supported our experimental results we have a tendency to draw conclusions on the results of the migration and supply suggestion on the way to extrapolate our expertise to alternative similar software system systems.

2. BACKGROUND

In this section we have a tendency to provide a definition of cloud computing alongside its key characteristics. Additionally, we have a tendency to describe existing cloud classifications counting on the preparation sort and provided capabilities.

2.1Cloud Computing

Cloud computing typically refers to a utility-based provisioning of procedure resources over the web.

Wide used analogies to elucidate cloud computing square measure electricity and water systems just like the Cloud; they supply centralized resources that square measure accessible for everybody.

"Clouds square measure an outsized pool of simply usable and accessible virtualized resources (such as hardware, development platform and/or services). These resources will be dynamically reconfigured to regulate to a variable load (scale), permitting conjointly for an optimum resource utilization. This pool of resources is usually exploited by a pay-per-use model during in which guarantees square measure offered by the infrastructure supplier by suggests that of made-to-order SLAs."

This definition, the same as different descriptions [6] reveals the most cloud characteristics:

•*Virtualization* (abstracted infrastructure): Virtualization allows dynamic infrastructure utilization, resource sharing, isolation and security, in distinction to a customary model once process takes place on specific hardware outlined beforehand, application don't have any static computing place in a much virtualized cloud surroundings.

•*A pay-per-use model*: this is often the key characteristic of cloud computing political economy. The entire resources in the cloud square measure charged supported the amount consumed by them. This model permits coming into the market with no direct investments into own hardware infrastructure.

•*On-demand access*: On-demand access funds that resources like electronic equipment time or storage will be provisioned mechanically once required with none further management effort.

•*Elastic scalability*: Elastic scaling signifies that procedure resources, utilized by the appliance, will be dynamically scaled up or down. In different words, virtualized hardware resources will be resized simply and demand.

•*Network access*: Everything within the Cloud is connected via the network. End-users access services via the web, developers deploy and monitor application within the same approach, communication between different services within the cloud happens through the network. Cloud computing platforms typically give REST-based genus Apis to their services.

•*Usability*: usually cloud computing platforms give straightforward outwardly managed surroundings to cover preparation and operational details from the user. Cloud computing systems give genus Apis to act with the surroundings, which simplifies the event.

2.2 Classifications of the Cloud

There square measure 2 wide used cloud computing classifications. The primary one describes four cloud sorts counting on the preparation location.

1. *Public clouds*. Public or external clouds square measure ancient clouds wherever resources square measure dynamically provisioned via the web by the off-site third-party suppliers. These resources square measure publicly on the market to everybody. Cloud customers square measure charged counting on the amount used. Example square measure Microsoft Azure [7], Google App Engine [8] and Amazon internet Services [9].

2. *Private clouds*. Private clouds typically talk to the emulation of a cloud computing surroundings on non-public infrastructure. Since users still need to purchase hardware and operational instrumentality, non-public clouds square measure typically criticized [10] [11]. Several firms do that form of cloud to verify their code domestically before deploying it to public cloud.

3. *Community clouds*. A community cloud suggests that cloud surroundings established across many organizations. Such clouds will be managed by the organizations or third-parties and put in either on-or off-premise.

4. *Hybrid clouds*. This term refers to a composition of 2 or additional clouds, as well as non-public clouds and public clouds. This model will be used for various functions. An example, archiving or replicating native information within the public cloud.

Another Wide used cloud metaphysics describes 3 cloud models counting on provided capabilities [12]: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).

1. IaaS. The cloud infrastructure layer represents basic resources that compose the bottom for higher layers. IaaS is made directly on the hardware, providing virtualized resources (e.g storing and process capacities) as a service. These resources will be split, dynamically resized and allotted to customers counting on their demand. IaaS is typically subcategorized into procedure resources, information storage and communication [12] the instance of public Iaas suppliers square measure Amazon internet Services and GoGird [13].

2. PaaS. The platform layer suppliers a better level code platform with extended services wherever different systems will run. This layer is sometimes designed on high of Iaas. It delivers programming-language-level surroundings with a group of language-integrated genus Apis for implementing and deploying SaaS applications, that makes PaaS terribly comfy and clear for developers. Microsoft Azure and Google App Engine square measure the instance of PaaS.

3. SaaS. The services exposed during this layer represent alternatives to domestically running applications. They are usually fascinating for a large market, compared to IaaS or PaaS. The appliance layer suppliers interface for skinny or thick purchasers like internet browsers or sensible phones. Usually SaaS applications square measure accessed through web-portals for a few fee. Microsoft Office365 or Gmail square measure e.g cloud services.

3. ADVANTAGES AND CHALLENGES

Despite various blessings, cloud computing brings problems that weigh down its adoption. We tend to believe call that call manufacturers a transparent understanding of the benefits and also the challenges so as to create a rational decision whether or not or to not migrate Associate in nursing existing application. during this section we tend to summarize the knowledge collected from completely different sources [2,3,5,14,15,16,17,20,21,22,23,24].

3.1 Advantages of cloud computing

1. *No direct investments*: firms will merely host their applications on cloud providers' information centers, paying just for consumed resources. This model is very engaging for startups or little organizations as a result of it will considerably cut back time to plug.

2. *On-demand capacity*: On-premise installations typically struggle with completely different load patterns. Some patterns may be foretold whereas others cannot. In step with statistics, company server's are unit sometimes over 80% underutilized [5]. Figure 1(a) shows that IT-capacity for on-premise systems is allotted beforehand to satisfy the anticipated load. (See figure 1(b)). Consequently, the users' are a unit ready to consume minimum needed capability at any time, this cloud utility model even additional financially engaging.

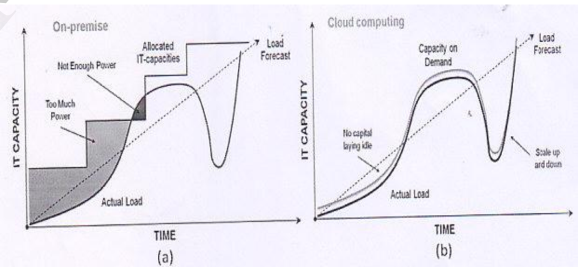

Figure 1. IT-capacity allocation (a) on-premise, (b) within the Cloud

3. *Easier client maintenance*: the upkeep overhead grows with the quantity of users. In distinction to it, cloud platforms enable centralized access to applications. Service suppliers will simply deploy their applications, delivering updates for all customers at the same time with none down time.

4. *Platform-provided features*: Cloud suppliers apply their information and skill to make their platforms. They use completely different techniques to deal with security, convenience and performance problems. In addition, cloud suppliers provide geographical information distribution and content Delivery Network (CDN)

services that decreases latencies and ends up in a far better end-user expertise.

3.2 Adoption challenges

1.*Security and privacy*: Security and privacy area unit improved through information centralization and security – homeward-bound elements [25], the chance of a knowledge discharge on the thanks to the cloud brings new challenges relating to secure transportation. VPN or encrypted information tunnelling between the native machine and a cloud atmosphere area unit potential solutions.

2. *Availability*: Another cloud adoption issue is convenience. Despite the fact that cloud suppliers provide a high level of convenience through SLAs, outages do occur in cloud platforms. The largest cloud suppliers have seasoned many serious outages for the past many years [18]. There are units some precautions that cloud customers will fancy mitigate the chance.

3. *Performance*: Virtualization and resource sharing cause performance unpredictability, particularly for I/O resource. Performance will become a significant downside, particularly once the {amount, the quantity} of request and also the amount of information increase. Cloud platforms typically give special caching mechanisms and CDN services which will partially compensate these problems.

4. *Multi-tenancy*: ancient on-premise computer code packages will sometimes be made-to-order in numerous ways that as a result of they are put in for every client severally. In distinction to it, SaaS applications area unit multi-tenant that means that one copy of computer code is shared by all users.

5. *Licensing*: Daily computer code licensing model for business computer code doesn't match cloud computing, as a result of licenses ordinarily limit the computers on that the computer code will run. Confusing licensing terms and conditions is that the biggest obstacle, particularly for giant organizations considering the cloud [22].

4. PUBLLIC CLOUD PLATFORMS

In this section we tend to describe 3 major public cloud platforms explicitly, Amazon AWS, Microsoft Azure and Google AppEngine. We tend to emphasize properties that area unit possible to have an effect on the choice.

4.1 Amazon Web Services

Amazon Web Service (AWS) represents a collection of on-line services that type along a cloud computing platform. Amazon has engineered large-scale, reliable and economical IT infrastructure wherever customers will host their applications. Besides REST based mostly genus Apis, AWS has recently discharged a collection of direct language-integrated genus Apis to access the cloud services.

Compute services: Amazon Elastic cipher Cloud (EC2) service permits dealing virtual machines to run custom applications on Amazon's information centres'. EC2 is extremely versatile and supports several operational systems, lots of middleware, and any development platform.

Storage Services: AWS offers varied study and scalable storages for various purposes. Straightforward Storage Service (S3) provides primary information storage for any sort and quantity of information.

4.2 Google AppEngine

Google AppEngine may be a Paas giving for developing and hosting net applications on Google-managed infrastructure. One in all the most important benefits of AppEngine is Google's technologies and services out there for custom applications. Developers will use commonplace language-integrated genus Apis to access most of those services. A collection of SDKs Associate in Nursing an Eclipse plug-in change full native development support. SDKs will simulate AppEgine surroundings on native machine.

Compute services: AppEngine provides secure surroundings wherever applications may be deployed. It presently supports java, Python and Go Runtime environments. Every surroundings provides commonplace protocols and customary technologies for an internet application development.

Storage Services: AppEngine offers many choices to control information. The Datastore is employed for non-relational information with high scan and question performance, machine scaling and dealing support. Not like computer database, it supports “schemaless” entities. The Blobstore is another storage service. Developers ought to use the Blobstore for giant information objects. These objects keep in Blobstore area unit known as “blobs”.

4.3 Micro Azure

Microsoft Azure is comparatively new PaaS giving within the cloud market. Azure platform consists of figure, Networking, Storage and Identity services. Microsoft Azure provides SDKs and tools for VS2010 to boost native development. SDKs will emulate a cloud setting on an area machine wherever developers will run, check and correct applications before moving to the general public cloud. Azure extensively uses existing Microsoft tools and technologies like ASP.NET, .NET MVC, ADO.NET, Visual studio IDE, and Microsoft SQL. Therefore developers will use existing expertise to migrate or develop cloud applications for Microsoft Azure.

Compute Services: Azure figure provides a special execution setting for hosted services. Services are often designed from completely different roles. Every role represents a part with distinctive practicality running within a virtual server. The net Role is meant to run frontend net applications. The employee Role severs for additional general purpose. The virtual Machine Role is intended to run user-customized OS pictures, giving additional management over the runtime setting. However, Azure supports solely Windows Server 2008 R2 software system. Table 1 show Azure compute instances.

compute instance size	CPU	Memory	Instance storage	I/O performance
Extra small	1GHz	768 MB	20 GB	Low
Small	1.6 GHz	1.75 GB	225 GB	Moderate
Medium	2x1.6 GHz	3.5 GB	490 GB	High
Large	4x1.6 GHz	7 GB	1,000 GB	High
Extra large	8x1.6 GHz	14GB	2,040 GB	High

Table 1 Microsoft Azure Compute instances

Storage Services: There are four storage abstractions supported by Azure Storage: Blob Storage, Table Storage, Queue Storage, and Windows Azure Drive. The Blob Storage is unstructured storage that resembles an everyday filing system. The Table Storage is employed for structured or semi-structured knowledge. The Queue Storage could be a special storage that's wont to organize Associate in nursing asynchronous, loose coupled work flow among roles.

5. MIGRATION DC SYSTEM TO THE CLOUD

In this section we describe the migration of an enterprise system to the cloud. We follow the migration process provided in [19], presenting the most relevant information.

5.1 DC Implementation

InformaIT Company

InformaIT is a small ISV that focuses on document management systems. Most of the systems are based on Microsoft technologies, so developers have rich experience in .NET framework, Visual Studio development environment, SQL Server database system, and windows Server OS.

The Document Comparison system (DC) was selected as a candidate for migration. DC is a small web-based enterprise solution that enhances document management processes. The main purpose is to provide a fast and easy way to compare textual and graphical content of different digital documents.

DC architecture

The system is implemented using Microsoft .NET 2.0 framework and various programming languages including server-side C# and C++, and client-side JavaScript. DC contains five main components: frontend web application, backend engine, distributed cache, database, and shared file store. (see Figure 2).

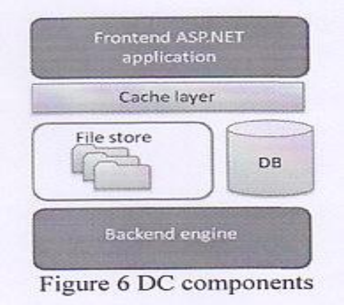

Figure 2. DC components

The *frontend* is a simple ASP.NET web application running under IIS on window OS. It provided web interfaces for end-users. The client-side JavaScript reduces the load on the server and improves usability.

The *backend* is implemented as a Windows Service (also .NET based). It performs long running computational tasks. The library accesses the files via regular file system API. It also requires the registration of a COM component.

The *file store servers* as a shared storage for system components. It keeps persistent data and organizes asynchronous communication between the frontend and the backend. The frontend stores uploaded documents and creates XML task files. The backend is checking for new XML task files, and stores rendered images in the corresponding location.

The *cache layer* keeps frequently used data, which increases the performance of the system. For example, the frontend store user session state there.

The database contains user personal information. Information is needed for authorization and authentication. The frontend uses the database mostly during login and logout procedures.

DEPLOYMENT MODEL

An on-premise distributed deployment model of DC id shown in figure 3. ASP.NET applications are composed into a Web Server Farm. They store frequently used data in a distributed cache that is usually located on a separate server. A customer can choose the number of frontend and backend servers to achieve the required performance. A shared network folder plays the role of persistent file storage. Microsoft SQL server is used as a database. Since it does not require much space and heavy querying, the database can be installed on a shared server.

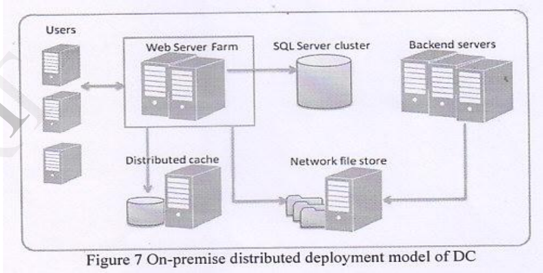

Figure 3. on-premise distributed deployment model of DC

6. EXPRIMENTS

In this section we investigate the main concerns of running DC in the cloud environment. These concerns are cost and performance. Based on our estimates and experiments, we compare DC behaviour in two environments (on-premise and the cloud) and under different condition in the Cloud. Other potential concerns, such as security and privacy, fall outside the scope of this thesis and remain as future work.

Testing environment

Experiments and measurements are done for North Europe deployment location of Microsoft Azure. All Azure compute instances have a small size, which provides. 1.75 GB memory, 225 GB local disk space, moderate I/O performance, and CPU performance equivalent to one 1.6GHz core.

We use small instances as a part of Azure free trial subscription, which gives necessary resources to perform our experiments for no fees. Testing on the client side is executed in a non-virtualized environment, external to the Cloud, with a direct connection to the Internet via a high-speed wired Ethernet.

6.1 Performance

As we identified earlier, a cloud environment entails increased latencies and unknown hardware underneath. Therefore, DC can have the following performance bottlenecks in the cloud: the execution of heavy computational tasks (like digital document rendering) that require efficient hardware; and session handling that is latency sensitive. These operations represent the highest risk when moving DC to the cloud environment, because they might lead to significant system performance degradation.

6.1.1 Page rendering time

We use "second per page" metric for measuring page rendering speed. This is a natural metric that represents the time required to create an image from a digital document page. To suggest a page standard, we have analyzed a production set of documents and classified two main types:

1. Textual documents with little graphic. they usually contain many (up to 40) A4 format
   Pages are rendered fast. We refer to this type as A4 page.
2. Graphic documents with little text. They usually contain one or two A3 –A2 format pages that are rendered considerably slower. We refer to this type as A3 page.

We pick up one A3 page and one A4 page for our experiments. The size of the A3 page is 25kb, with a 495Kb corresponding rendered image. The size of A4 page is 2Mb, with a 1.4Mb Corresponding image.

The rendering library gets regular file system paths as input parameters. Since the original on-premise system keeps all documents in a regular file system, it calls this library directly. Cloud DC uses Azure Storage. This requires some pre-processing and past-processing steps to render a page.

1. Download a required document from Blob storage to local file storage.
2. Call the rendering library.
3. Upload rendered images to Blob Storage.
4. Cleanup local Storage.

So the local rendering time in the cloud can be decomposed into these four steps. Figure 4 illustrates the observed time for A3 and A4 pages in the cloud environment.

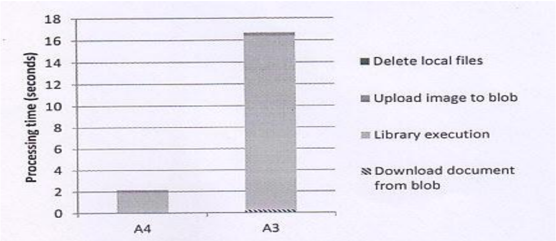

Figure 4: page rendering time

As shown in figure 4, pre-processing and post-processing steps for A3 page take about 0.5 seconds, which is only 3% of the total time (16.7 seconds). For A4 page these steps take 0.28 seconds, which is 13% of the total time (2.15 seconds)

As the next step, we compare page rendering time for cloud and on-premise DC versions. For a cloud version we use a small Azure compute instance, while on-premise installations have Core2Duo P7350 2.0 GHz Mx86 (laptop), Core2Duo E7500 2.9GHz x86 (workstation), Core i3 540 3.07GHz x64 (dedicated local server). Figure 5 illustrates the results of our experiments.

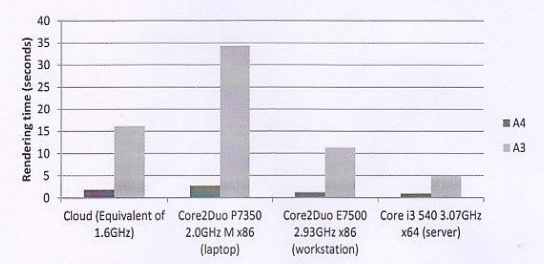

Figure 5. Page rendering time comparison for cloud and non-virtualized environment

6.1.2 Session storing/retrieving time

In this section we compare on-premise and Cloud DC session handling performance and also test two alternatives in Azure platform. For on-premise installation we evaluate standard ASP.NET in-process and state server modes. For cloud installation we evaluate AppFabric Cache, and a custom session handler that uses Azure Table.

Session handling is very important for the frontend ASP.NET application, because it retrieves and stores session data on every page load as a part of the ASP.NET application lifecycle [26]. We perform this experiment against different storages and different object size: 1 Kb, 1Mb, and 10Mb (assuming that session should not exceed 10Mb).

Session Size	On-premise DC installation		Cloud DC installation	
	In-process	State server	AppEngine	Table Storage
1Kb	0.0/0.0	0.0/0.0	0.004/0.008	0.094/ 0.113
1Mb	0.0/0.0	0.008/ 0.009	0.098/0.143	0.292/ 0.548
10Mb	0.0/0.0	0.161/ 0.173	0.435/0.583	1.167/ 1.861

Table 2 storing/retrieving time for session data

As we can see in table 2, on-premise DC requires significantly less time for session handling compared to the Cloud installation. In-process mode is obviously the fastest. Since all data is kept in memory, the application needs to store and retrieve only a pointer to the memory location.

6.1.3 Response time

In this section we test response time of the frontend web application against different deployment locations and different scale. response time includes the latency to send a request from a client, the time to redirect the request by a load balancer, the time to process it by the application, and the latency to get a response back from the server (see figure 6). In order to measure response time purely for web Role, we use a stateless .aspx page that does not include any external factors like session handling or document page rendering.

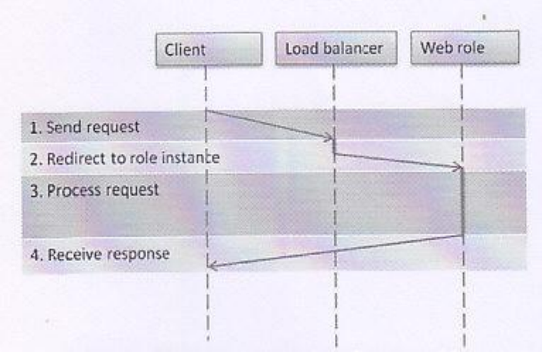

Figure 6: Page response time decomposition

The first experiment evaluates page response time for a different number of role instances. The page makes some calculations and then generates dynamic output content. This dynamic data is needed to ensure that the page is not cached by any CDN service or in the client environment. In order to measure response time, we use Visual Studio 2010 Load Test [27] based on a Web Test simple requests the page..

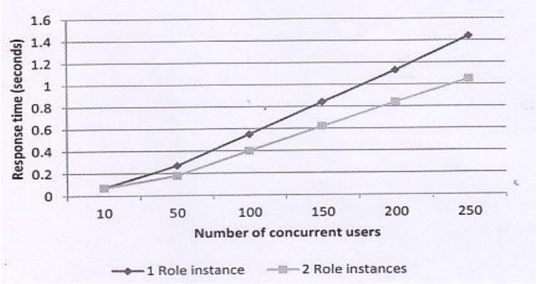

Figure 7: Cloud DC page response time

Figure 7 shows observed response time for both single instance and dual instance setups with an increasing number of concurrent users. Page time starts at about 80 ms for both cases and then glows linearly with angular coefficients. Results show that an additional role instance decreases response time, especially for a heavy load. For 250 concurrent users a dual instance setup performs 400ms faster than a single instance setup.

6.2.Cost

In this section we estimate the cost of DC in the cloud. The cost for every scenario is estimate based on the Microsoft Azure pricing model.

6.2.1 Scenario 1: demo installation

In scenario 1 cloud-enabled DC is used as a demo installation. We have estimate the load and required capacities based on the statistics form current virtual private servers with demo installations. According to our estimations, we need three small compute instances one for a web Role and two Worker Roles. The overall cost is presented in table 3. Figure 8 shows the cost distribution among different services. For more information see the official Microsoft Azure page.

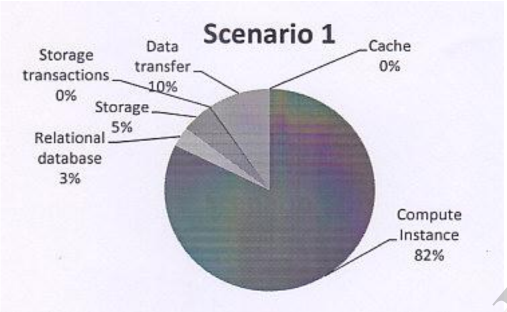

Figure 8: Cost distribution for Scenario 1

Service	Used Capacity	Costper month ($)
Compute Instance	3 small instances (2160 hours)	259
Relational database	1 GB	9.99
Storage	100 GB	15
Storage transactions	1000k transactions	1
Data transfer	20 GB	30
	Total:	314.99 (214.99)

Table 3 DC estimated cost for Scenario 1

6.2.2 Scenario 2: production installation without scaling

In scenario 2 DC is used as a production installation with throughput equivalent to one dedicated server (without elastic scale). For this scenario we require DC to show the same through as the on-premise installation that is running on a server with core i3 540 3.07GHz x64 processor, 500 GB available local storage and 4GB of memory. Table 4 presents the total cost for this scenario, and figure 9 illustrates the cost distribution.

Service	Used capacity	Cost per month ($)
Compute Instance	11small instance (7920 hours)	950
Relational database	1 GB	9.99
Storage	500 GB	75
Storage transactions	5000k transactions	5
Data transfer	1000 GB	150
Cache	512 MB AppFabric Cache	75
	Total:	1264.99 (1084.99)

Table 4 DC estimate cost for Scenario 2

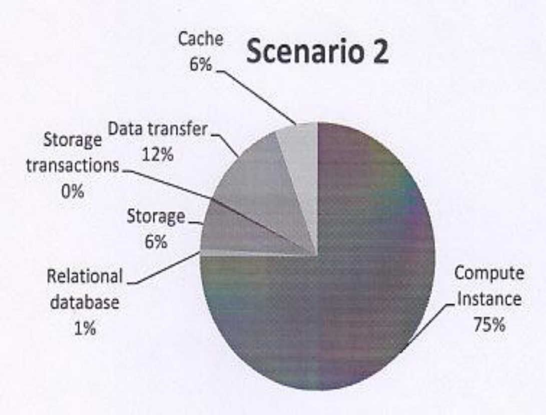

Figure 9: Cost distribution for Scenario 2

6.2.3 Scenario 3: production installation with scaling

In scenario 3 DC is used as a production installation with throughput equivalent to one dedicated server (using elastic scale), in this scenario we use the same capacities as in scenario 2, but leveraging cloud elastic scalability. We assume DC has a typical enterprise system load pattern high load during working hours and almost no load during the rest time. The estimate cost is presented in table 5. Figure 10 shows the cost distribution among different services.

Service	Used capacity	Cost per month ($)
Compute Instance	3-11 small instance (3920 hours)	470
Relational database	1 GB	99.9
Storage	500 GB	75
Storage transactions	5000k transactions	5
Data transfer	1000 GB	150
Cache	0-512 MB	55
	Total:	764.99 (664.99)

Table 5 DC estimate cost for scenario 3

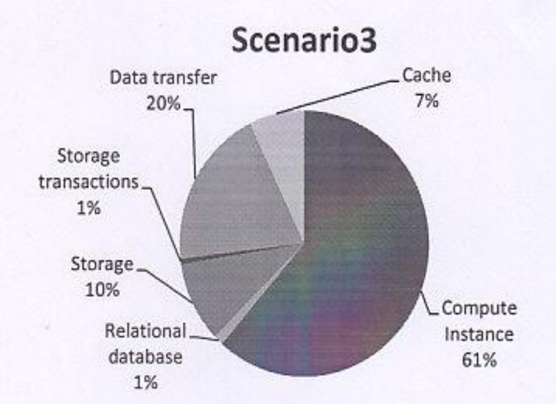

Figure 10: Cost distribution for Scenario 3

Based on our estimate we can conclude that compute service dominate in all scenarios. It makes up 82%, 75%, and 61% of the total cost for scenario 1, 2, and 3 accordingly. On the other hand, storage transactions have the least cost. SQL Azure also has a cost share: 1% for scenario 2 and 3 and 3% for scenario 1. However, this is because DC is not database centric. We found that the cost of DC can drop by 40 percent (764.99$ compared to 1264.99$) when leveraging elastic scalability.

FUTURE WORK

We have discussed many questions regarding the migration of applications to the cloud. Still, there are some concerns of running applications in the cloud environments that we haven't investigated. For example, we did not evaluate security, privacy, and availability of cloud-enabled services. Also, we did not test the scalability of cloud persistent storages.

Furthermore, it would be interesting to examine application behaviour, in other cloud platforms-like Amazon AWS – that also fit our case. Later, the results observed across different cloud environments could be analyzed and compared giving more comprehensive knowledge about the consequences of the migration.

We will also continue examining the DC system in the cloud environment in order to compare the estimated and the real costs. In addition to that, load and performance will be monitored so that we can suggest a better scaling strategy for DC.

REFERENCE

[1] D. F. Parkhill, "The Challenge of the Computer Utility", *Addison-Wesley Educational Publishers Inc.*, US, 1966

[2] P. Botteri, D. Cowan, B. Deeter, A. Fisher, D. Garg, B. Goodman, J. Levine, G. Messiana, A. Sarin, and S. Tavel,"Bessemer's Top 10 Laws of cloud computing and SaaS", Bessemer Venture Partners, www.bvp.com, 2010

[3] N. Leavitt, "Is Cloud Computing Really Ready for Prime Time?", *Computer*, vol.42, no.1, pp.15-20, Jan. 2009

[4] Andrew R Hichkey, "SMB cloud spending to approach $100 billion by 2014", CRN, at http://www.crn.com/news/cloud/226700149/smb-cloud-spending-to-approach-100-billion-by-2014.htm, August 2010

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of cloud computing", *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[6] Jeremy Geelan, "Twenty one experts define cloud computing", *Virtualization*, Electronic Magazine, available at http://virtualization.sys-con.com/node/612375, Jan. 2010

[7] Microsoft Azure, http://www.microsoft.com/windowsazure/

[8] Google App Engine, http://code.google.com/appengine/

[9] Amazon Web Services, http://aws.amazon.com/

[10] Gordon Haff, "Just don't call them private clouds", CNET News, at http://news.cnet.com/8301-13556_3-10150841-61.html, Jan. 2009

[11] Andrew Conry Murray, "There is no such thing as a private cloud", at http://www.informationweek.com/blog/229207922, InformationWeek, Jan. 2009

[12] L. Youseff, M. Butrico, and D. da Silva, "Toward a unified ontology of cloud computing", *Grid Computing Environments Workshop, 2008. GCE '08* , pp.1-10, 12-16 Nov. 2008.

[13] GoGrid, http://www.gogrid.com/

[14] H.R. Motahari Nezhad, B. Stephenson, and S. Singhal, "Outsourcing Business to cloud

computing Services: Opportunities and Challenges", technical report, http://www.hpl.hp.com/techreports/2009/HPL-2009-23.pdf, 2009

[15] D. Chappell, "Windows Azure and ISVs: A guide for decision makers", whitepaper at http://www.microsoft.com/windowsazure/Whitepapers/AzureAndISV/, Jul. 2009

[16] Wom Kim, Soo Dong Kim, Eunseok Lee, Sungyoung Lee, "Adoption issues for cloud computing", Proceedings of the *11th International Conference on Information Integration and Web-based Applications & Services*, pp. 3-6,ACM, 2009.

[17] M.A. Vouk, "Cloud computing – Issues, research and implementations", *30th International Conference on Information Technology Interfaces*, pp. 31-40, Jun. 2008

[18] B. Rimal, E. Choi, and I. Lumb. "A Taxonomy and Survey of cloud computing Systems", in *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, pages 44–51. IEEE, 2009.

[19] Van Tran, Jacky Keung, Anna Liu, and Alan Fekete, "Application Migration to Cloud: A Taxonomy of Critical Factors", *Proceeding of the 2nd international workshop on Software engineering for cloud computing*, pp. 22-28,ACM, 2011

[20] B. Golden, "The Case Against cloud computing", http://www.cio.com/article/477473/, CIO, Jan. 2009

[21] R. Sean, "Cloud optimization – expanding capabilities, while aligning computing and business needs: A framework for making business decisions about cloud computing", whitepaper at http://www.microsoft.com/windowsazure/Whitepapers/CloudOptimization/, Jun. 2010

[22] B. Glick, "IT leaders are ready for the cloud – but are their suppliers?" http://www.computerweekly.com/blogs/editors-blog/2011/02/it-leaders-are-ready-for-the-c.html, ComputerWeekly.com, 2011

[23] D. Durkee, "Why cloud computing Will Never Be Free", *Queue*, vol. 8, pp. 20–29, ACM, 2010

[24] "Cloud computing survey: Exclusive research from CIO magazine", http://www.cio.com/documents/whitepapers/CIO CloudComputingSurveyJune2009V3.pdf, CIO, Jun. 2009

[25] J. Hughes, "Encrypted Storage and Key Management for the cloud",http://www.cryptoclarity.com/CryptoClarityLLC/Welcome/Entries/2009/7/23_Encrypted_Storage_and_Key_Manage ment_for_the_cloud.html, 2009.

[26] "ASP.NET Application Life Cycle Overview for IIS 7.0", at http://msdn.microsoft.com/enus/ library/bb470252.aspx, 2011

[27] "Running Web and Load Performance Tests", at http://msdn.microsoft.com/en-us/library/ee923688.aspx, 2011