REPoRT 1: Assessing hosting Options

# Overview

Following is the proposal for choosing the hosting the thematic based video producing application. The total time taken to prepare this report is 8 hours.

We have to evaluate two systems: cloud-based and TAMU-based. The TAMU based ownership is of an on-premises virtualization farm with cloud-based infrastructure as a service (IaaS). Due to the investment by TAMU, the racks are available at 5% the average rates of San Francisco or New York. Besides, the network charge and disaster recovery fees are exempted for the first three years. In the Cloud based, we have the choice of getting some annual subscription of cloud servers and meet our requirements. We can see that the infrastructure bought in the early stages in TAMU-based would become a part of the company’s assets but we have looked beyond the initial cost of an in-house solution vs the monthly cost of the cloud. The real picture of what’s cheaper goes far beyond the cost of a new in-house server or an annual subscription to cloud servers or Infrastructure-as-a-Service and has been done. We have come to a conclusion that the Cloud based hosting is economically favorable in three years and in long term.

# Assumptions and Background

From the given background, first we’ll have to determine and estimate the size of the users and their data that the app will be dealing with. For this particular task, the following assumptions have been made. This is in accordance with the specifications provided by the client.

1. There is no dropping of any pictures or videos from the user database (Google photos, iCloud, Facebook etc.) and no computing task is allocated in filtering any inappropriate contents.
2. The app has a revolutionary idea just like Pinterest or Facebook and therefore the user statistics (like number of daily active users, the amount of photos and videos that they share etc.) of the apps mentioned have been used in estimating the costs.

As the app will grow, there would be hundreds of thousands of requests hitting the infrastructure, hence a long term strategy would be the needed. Another thing is the security and reliability of the data. Therefore selecting the right datacenter or cloud will have a huge impact on the business.

# Basic User Statistics

The two universities that are involved initially are Austin and College Station, with the total number of students equal to 68,603 and 50,950 respectively. In the first month itself the total number of users would be 5% of the students which would be 3,430 and 2,548 respectively which would make the total user base to 5,978. Now these are the total number of users of the app. Assuming that this app continues to grow and follows the trend of Pinterest and Facebook, we can safely assume that 90% of that will be active daily users, which would make the typical daily user base to 5,389. Besides, we can assume that during the peak days, like public holidays, the app will be scraping off images and videos from the entire users and therefore the infrastructure should be able to handle this. Assuming that one picture (in compressed form) has 10 kB of memory and one video is of 20 MB, the total size of the data can be estimated as follows:

* 10 % students who are active will have 100 pics and 10 videos daily
* 70 % will have 5 pics and 0.3 videos daily
* Rest 20 % will only consume data

This makes the total estimated data size per day to be 206.5 MB times the number of users which is calculated from the sizes assumptions taken above. Since we have assumed that the growth of the users will follow the trend of Pinterest, we can say that this rate would be proportional to the number of users, which we can estimate using the growth curve of Pinterest.

|  |  |  |  |
| --- | --- | --- | --- |
| Timeline | Number of Users | Number of Active Users | Data shared per day |
| Month 1 | 5,978 | 5,389 | ~ 100 GB |
| Year 1 | 119,740 | 107,780 | ~ 2 TB |
| Year 2 | 9,878,550 | 8,891,850 | ~ 165 TB |
| Year 3 | 89,805,000 | 80,835,000 | ~ 1.5 PB |

Given that initially these users are located in Austin and College Station, we can ignore the latency that will be associated when a user in some other part of the country, like New York. This can happen when a student goes to his home in New York for vacations and uses the app from there. Besides, this app could scale up and then it would require servers that can serve their own local users.

# Analysis

The total cost for TAMU based can be divided broadly into: direct, indirect and hidden. Direct costs can be the hardware required for an in-house solution (TAMU based). This includes the actual servers, server room to store them, electricity, cooling etc. The indirect costs are the salaries to pay to System Administrator and other staff. The hidden costs could be the lost productivity due to downtime. Every service has varying level of the downtime guarantees and we can evaluate it based on the SLA. There will also be a cost of refreshing the on-premises server but since it’s usually required after 4 years and this analysis is for 3 years, plus the 4 years extended to see the long term impact.

On the other hand, the overall costs for the cloud based depends on the number of vCPUs, RAM, and disk space required. The downtime hours and availability also vary for various vendors, which has to be taken into account. The best benefit of cloud based system comes from scalability and low geographical latency when the app moves out of Texas.

The following table is the cost estimates for the TAMU based servers. The cost values have been taken from the internet. As can be seen in the comments, TAMU provides the server infrastructure for 5 % of the rate in California or New York. There have been four sub calculations done for four different scenarios:

1. There is a cluster of four powerful machines and it takes half the time to process
2. There is a cluster of four powerful machines and it takes 75% time to process
3. There is cluster of sixteen smaller machines and it takes half the time to process
4. There is cluster of sixteen smaller machines and it takes 75% the time to process

Depending on it the least cost machines (four powerful with half the time to process) has been taken for comparison.

|  |  |  |  |
| --- | --- | --- | --- |
| **Basic Global Variables** | | | |
| Annual salary for system administrator: | 85,000.00 $ |  | |
| System administrator hourly cost: | 62.77 $ | Includes tools, equipment, vacations, benefits | |
| Number of months for amortization: | 48 |  | |
| Server install (HW + software): | 5 | Hours to physically install a server and install base OS | |
| Server specialization: | 8 | Hours to install and configure the application layer | |
| Server maintenance and support: | 12 | Hours to support and maintain the physical server the first year | |
| Typical cost per TB for High performance SAN: | 6,500.00 $ | Based on pricing for SAN | |
| Typical backup costs per TB: | 0 $ | TAMU providing for free | |
| Yearly maintenance MSFT: | 15.00 | Hours of software maintenance per server per year | |
| Yearly maintenance HW: | 5.00 | Hours of hardware maintenance per server per year | |
|  |  |  | |
| **First Year Costs** | | | |
| Server licensing: | 40,000.00 $ | TAMU Discounted Datacenter licences for the virtualization hosts | |
| Sysadmin time costs: | 25,107.69 $ |  | |
| Backup costs: | 0 $ | TAMU providing recovery for free | |
| Disk space costs: | 13,000.00 $ |  | |
| Electricity: | 3,363.84 $ |  | |
| Server(s): | 184,000.00 $ |  | |
| Infrastructure: | 17,581.92 $ | Networks (data and communication), square footage, Internet, HVAC | |
| Total: | 283,053.45 $ |  | |
|  |  |  | |
| **Recurring Costs** | | | |
| License upkeep: | 7,950.00 $ | 15% of license costs for yearly maintenance (Microsoft, SAN and backup) | |
| Server service contract: | 22,080.00 $ |  | |
| Electricity: | 3,363.84 $ |  | |
| Sysadmin costs: | 20,086.15 $ |  | |
| Infrastructure upkeep: | 5,341.92 $ | Networking support contracts, HVAC, square footage, Internet | |
| Total: | 58,821.91 $ |  | |
|  |  |  | |
| **Sub Calculations 1 for Cluster 1** | | | |
| kw/h | 0.06 $ | | Electricity commercial rate |
| Number of hours in a year: | 8,760 | |  |
| Mean kw/h consumption per server: | 0.40 | | From public Hewlett Packard documentation for mid-range enterprise servers |
| Total disk space required: | 2.00 | | Terabytes from TCO Comparison tab |
| Hardware: | 184,000.00 $ | | Dual six cores, 64GB RAM |
| Servers needed: | 4 | |  |
| Total: | 184,000.00 $ | |  |
|  |  | |  |
| **Sub Calculations 2 for Cluster 2** | | | |
| kw/h | 0.06 $ | | Electricity commercial rate |
| Number of hours in a year: | 8,760 | |  |
| Mean kw/h consumption per server: | 0.40 | | From public Hewlett Packard documentation for mid-range enterprise servers |
| Total disk space required: | 2.00 | | Terabytes from TCO Comparison tab |
| Hardware: | 245,000.00 $ | | Dual core, 16GB RAM |
| Servers needed: | 16 | |  |
| Total: | 245,000.00 $ | |  |
|  |  | |  |
| **Sub Calculations 3 for Cluster 3** | | | |
| kw/h | 0.06 $ | | Electricity commercial rate |
| Number of hours in a year: | 8,760 | |  |
| Mean kw/h consumption per server: | 0.40 | | From public Hewlett Packard documentation for mid-range enterprise servers |
| Total disk space required: | 2.00 | | Terabytes from TCO Comparison tab |
| Hardware: | 212,000.00 $ | | Dual six cores, 64GB RAM |
| Servers needed: | 4 | |  |
| Total: | 212,000.00 $ | |  |
|  |  | |  |
| **Sub Calculations 4 for Cluster 4** | | | |
| kw/h | 0.06 $ | | Electricity commercial rate |
| Number of hours in a year: | 8,760 | |  |
| Mean kw/h consumption per server: | 0.40 | | From public Hewlett Packard documentation for mid-range enterprise servers |
| Total disk space required: | 2.00 | | Terabytes from TCO Comparison tab |
| Hardware: | 196,000.00 $ | | Dual core, 16GB RAM |
| Servers needed: | 16 | |  |
| Total: | 196,000.00 $ | |  |

|  |  |  |
| --- | --- | --- |
| **Cloud Based Calculations** | | |
| vCores: | 24 $ | Latest Generation CPU from INTEL |
| Dedicated RAM: | 9.928 $ | Dedicated ECC High Speed RAM |
| Tier-1 All-SSD SAN Storage: | 0.365 $ | Enterprise SAN |
| Number of vCPUs: | 100 | Virtual CPUs |
| Total pool of ram (GB): | 180 | Computed from the user data table |
| Total pool of disks (GB): | 2000 | Computed from the user data table |
| On-premises minimal guaranteed uptime: | 98.0% |  |
| Amortization/hardware refresh cycle: | 48 |  |
| Average monthly cost on-premises : | 8,572.09 $ |  |
| Cloud server monthly costs : | 4,896.84 $ |  |

# MONTHLY COST COMPARISON for Tamu Based and CLoud based

|  |  |
| --- | --- |
| **TAMU based total cost (monthly)** |  |
| Refresh cycle first year | 232,085.95 $ |
| Following years: | 51,176.79 $ |
| Average monthly cost: | 8,572.09 $ |
|  |  |
| **Cloud total cost (monthly)** |  |
| Cloud Servers : | 4,896.84 $ |
| Average monthly savings in $: | 3,675.25 $ |
| Average monthly savings in %: | 43% |

# YEARLy COST COMPARISON for Tamu Based and CLoud based

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **TAMU Based** | **Cloud Based** | **Savings $** | **Saving %** |
| Year 1 | 232,085.95 $ | 58,762.08 $ | 173,323.87 $ | 75% |
| Year 2 | 51,176.79 $ | 58,762.08 $ | (7,585.29) $ | -15% |
| Year 3 | 51,176.79 $ | 58,762.08 $ | (7,585.29) $ | -15% |
| **Total:** | **334,439.53 $** | **176,286.24 $** | **158,153,29 $** | **47.3%** |
| Year 4 | 51,176.79 $ | 58,762.08 $ | (7,585.29) $ | -15% |
| Year 5 | 232,085.95 $ | 58,762.08 $ | 173,323.87 $ | 75% |
| Year 6 | 51,176.79 $ | 58,762.08 $ | (7,585.29) $ | -15% |
| Year 7 | 51,176.79 $ | 58,762.08 $ | (7,585.29) $ | -15% |
| **Total:** | **720,055.85 $** | **411,334.56 $** | **308,721.29 $** | **43%** |

# Conclusion, PROS and COns

We show here that in three years the savings would be 47.3 % when the cloud based hosting is considered. It has also been shown that in the long term (including the refresh), the overall savings is 43% in 7 years. Monthly savings would be also economical for the cloud based system.

Hence the recommended one is going with the cloud based. Other advantages of the cloud based system are reduced latency for geographical scale. As and when required the servers could be kept near the user locally and reduce the latency of operations. It is also easy to migrate in such a setting.

The pros of having a TAMU based approach is that the datacenter architecture is in the control of the client. It is very cheap as compared to a market value data centers.

REPoRT 2: Assessing hEROKU

# OVerview

This report is to assess the ease of use of the Heroku cloud platform. The time taken for this section is 2 hours.

# STEPS and Problems Faced

First I created a Heroku account. Now to connect to this, I did an ssh. The three basic commands were:

$ ssh-keygen -t rsa  
$ heroku login  
$ heroku keys:add

Now a new container could be created using

$ heroku create

Heroku will assign your app a whimsical name such as luminous-coconut-237; once our app is deployed, we can access it at http://luminous-coconut-237.herokuapp.com. We can login to the Heroku website if you want to change the name of your app.

Finally, we deploy our app to Heroku:

$ git push heroku master

There was one warning that we saw:

The authenticity of host 'heroku.com (50.19.85.132)' can't be established.  
RSA key fingerprint is 8b:48:5e:67:0e:c9:16:47:32:f2:87:0c:1f:c8:60:ad.  
Are you sure you want to continue connecting (yes/no)?   
Please type 'yes' or 'no':

After pressing ‘yes’, we never saw that error message again.

We can’t see out application on the URL unless we do a

$ heroku run rake db:migrate

We can navigate to the heroku URL that is printed at the end of the results from git push heroku master. Initially we'll get a "We're sorry, but something went wrong." error in the browser if we haven’t run the above command.

We can get a hint to solve some errors by running the following command:

$ heroku logs

# Heroku Usefulness and initial Impressions

Heroku seems to be a useful tool to deploy the website immediately and bypass the traditional problems faced by web developers. Besides Heroku is free. The benefits are:

* PAAS (Platform as a Service)
* Documentation is very good.
* Has built-in tools and architecture.
* Limited control over architecture while designing app.
* Heroku is best at what they provide.
* Deployment is taken care of (through git commands only).
* Good support.
* Not time consuming.

However, we don’t see many of the features that AWS provides which are:

* IAAS (Infrastructure as a Service)
* AWS is versatile. They have many kinds of products. EC2, LAMBDA, EMR etc.,.
* Can go for Dedicated instance; more control over the architecture, like choosing OS, Version of the softwares, etc. There's more than one backend layers.
* Can use the automated deployment, or roll your own.
* Great support.

Hence, it depends on options/efforts that the developer wants to have/take. If we want to start an app right away, without much customization of the architecture, then we should choose Heroku. If we want to build an app, and would like to focus on various things like architecture, using a different web server, and also want to use the other services that AWS provides, then we may have to choose AWS. AWS also comes with many services/products to plug/play.