International Journal of Engineering Research in Computer Science and Engineering(IJERCSE) Vol 3, Issue 3, March 2016

Advanced Chatting Powered by Cloud – An Analysis

[1] Y Akshobhya, [2]Navin K,
[1]M.Tech Cloud Computing,[2]Assiatant professor(OG)
[1]akshobhya.y.55@gmail.com, [2]navin.k@ktr.srmuniv.ac.in

Abstract*— In this paper we are going to explore implementation of Advanced chatting using the herculean powerful technology of the contemporary computing industry which is Cloud Computing. This compares to the traditional method which is used for chatting or instant messaging between mobile devices, whereby an application is installed on the users' device which communicates with a server. In today's world the number of people using these chatting or instant messaging applications follows a trend of exponential growth. To provide a facility capable providing services to the increasing number of users a significant server should be equipped and maintained which is becoming cost-inconsequential in the current Information Industry. With Technology industry changing its colors day-to-day and providing noteworthy services we can build a robust chat application while lowering the infrastructure cost to implement by moving our server side application to cloud by purchasing cloud services for the server. Using Platform as a Service (PaaS) which is one of the offerings of Cloud Computing we can build a server which will function in the same way as traditional server does. Then we can implement a client application with specialized features, such as marking a read message as unread, forwarding a message or messages to multiple users in the contact list and storing their images into the cloud storage. One important feature of significant importance to this application is implementing a 'balanced Cloud'. If we can balance particular data such as duplicated images or videos without any replications then it could reduce the storage space consumed by the application. Also managing the server using the cloud account from the PaaS service provider will make developers' job much easier. The cloud will not only serve as an application server but can also be utilized to back up the user data which is referred to as Storage Service, an optional facility subscribed along with PaaS. Whenever the user will change their device their data will follow them to the new device, such as Mobile Phone, Tablet, Computer, etc.

Index Terms—*Cloud Computing, Paas, Cloud Storage, Balancing Cloud*

I . INTRODUCTION

The Internet has revolutionized the way we communicate. E-mail has been the most rapidly adopted form of communication ever known. But getting a reply to a sent message takes a little time in the e-mail system and also users need to go through discrete steps for composing a new message. So the evolving technology brought a simpler way of communication in instant messaging (chatting) which has been successfully implemented both on a commercialas well as casual basis.

In instant messaging there is no need to type the email addresses of the users to whom we are intending to send a message; there will be a provision of a contact list where we just need to add the persons to which we wish to send a message or multimedia content. Then the person at the other end can instantly view the message by a connection to the internet and also send a reply by typing straight into the reply area. This mode of superfast communication has become wildly popular with the current generation of mobile device users.

A conventional way to implement these chatting applications is to setup a server to which the user's applications will connect and send the messages to all the other users in the contact list who are connected to the server. If a particular user is not connected to the server we will not be able to send messages to them but a few applications are allowing forwarding/sending messages where they will be stored in the server for few hours to days until the user will connect to the server. Once they are connected to the server the messages that were sent while the user was offline will be delivered to the user as notification that they have received a new message, while the stored messages are discarded from the server. Today's chatting applications are implemented inefficiently, withfewer features than have been implemented in the e-mailing system. The cost of the instant messaging is high due to setting up a server which should be able to provide services like transferring the messages and multimedia across the globe. Also once we setup a server expecting the traffic of data that may fluctuate in a real case scenario. Say in case we expect 10 million live users a day it may vary abruptly up or down. Therefore our server setup must be over-engineered to cope with those situations.

The cost of the server can be brought down by usingrobust cloud computing technology while integrating divergent features within the application. Cloud Computing provides Platform as a Service through which the implementation of server will become very straightforward. In PaaS we will deploy our server application to the Cloud Service Provider from where we get the services and where the application will be executed. The Service provider will provide us services such as compute, storage, memory management and variousApplication Programming Interface (API's). By aggregating them all together we can build a robust application at lowest cost. The features that are considered in this application are:

- Tagging – putting all the important messages together
- Forwarding – Sending messages or multimedia content to several other users at same time, or as they come online

- Cloud Storage – Providing storage space to the users and also taking backup of their chat data and multimedia content such as Audio, Videos, Images.

An objective is to balance the cloud storage to optimize application efficiency.In particular, if we are providing users with free storage space for their multimedia contentit is ideal to utilize the balancing feature.Let us consider the following example for better understanding the balance storage feature:Consider three users of our chatting application A, B and C. Imagine A sends an image to B and C, which will be stored into their cloud storage respectively. Now if B forwards the content,as user B is unaware that A has sent same image to C then there will replica of same image in the C's cloud storage. This will tend toconsume the storage capacity which will be offeredfor free, as a limited offering, to the user. So if we could balance that part by utilizing a pointer to the image sent between users, the same image sent from the A and B at different times then the cloud storage space given to the user can be better optimized. This is also referred to as datadeduplication.

As far as developers are concerned they can remain focused on developing and maintaining aspects where as they can track the application performance from the service provider. Tracking the performance of the application involves monitoringmetricssuch as numbers of live users in chat, volumes of multimedia being shared and storage space occupied, CPU being utilized by the app and other assorted metrics. The developer can sign in to the administration account provided by the Platform as a Service provider they can track and get to know the status of the application. The application creatorcan also subscribe for the services they require and utilize; specifically they pay for what they use. Based on criteria of the utilization and services Platform as a Service provider and developer of the app will have Service Level Agreement between them through which the services are provisioned.

Hence the developing chatting application is noteworthy inasmuch as it is being developed at lowest cost and withminimal support requirements. It brings the commercialization with its vibrant features and also reflects the power of utilizing a modern day technology: cloud computing by demonstrating that it's simple to develop, deploy and support the application.

II Survey

Let us get to know some concepts related to existing chat applications which are the precursors to our current Advanced Chatting app so that the sequel will bring more feature enhancements. The user interface and various mechanisms can utilize the U.I. of existing chat applications. Some applications utilize a 'store and forward' mechanism for exchanging messages between two users. When a user sends a message, it firstly travels to the server where the message is stored. Then server repeatedly attempts delivery and acknowledgment of receipt of the message. As soon as message is acknowledged, server drops the message which means it is no longer available in database of server. The server keeps the message only for, say typically, 30 days in its database when it is not delivered to receiver, then discarded. Also the encryption of the messages using their phone number before storing them into the database can be a mechanism to capitalize because of being successfully utilized in security aspects. Multimedia messages are sent by uploading the image, audio or video to be sent to an HTTP Server and then sending a link to end recipient where the message will be automatically downloaded. There are also emoticons in virtually every chatting and email application. These can be incorporated into our own application hence the users can feel very familiar. They shouldn't get into a feel of using and learning a whole set of new application which may not be able to connect with everyone who is not technically inclined, including those who have already got used to certain chatting applications. We just need to enhance the chat application by putting up a whole new different approach to develop, maintain and different set of new features. But according to my survey though some applications are using as cloud backup they are not making their storage with the balancing/deduplication option and also their server development and management is in a traditional way. Hence this Advanced Chatting powered by cloud can stand as a potential market leader when compared to the other instant messaging applications.

III. CLOUD ARCHITECTURE

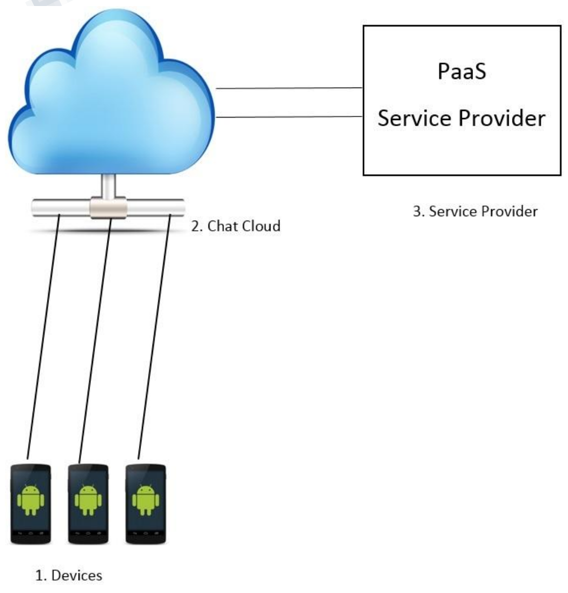

Figure 1:Advanced Chatting Cloud Architecture

The architecture for the system that will be implemented is described as follows:

Devices need to install the application package of the client that can be downloaded, for example from a 'store'. When the devices execute the application they need to have the internet to connect to the Chat Cloud. At the first time the application needs users to register themselves, by providing their name, phone number and e-mail. These details will be stored into the respective data base by creating an instance for the user in the chat cloud where service is acquired from the Platform as Service provider. Then a filtered list of users will turn up as we move on from registering to next page. Now the user can select a particular member from the list and subsequently send them text messages, images, audio, video. These message and multimedia data will be first received by the chat cloud, then it would be sent to the corresponding member. If the member is connected to the internet and not using the application then they would be displayed in the device notification area.

Chat cloud is actually the server side code that has the business logic to make users chat through their devices and also it will be persisting all the corresponding chat data. Chat cloud is also responsible to implement the balance/deduplication for the cloud storage of images. The chat cloud will be deployed into a particular Platform as a Service Provider (PaaS) where the developers will have a subscription account and Service Level Agreement between parties. In this way the PaaS services can be utilized by the mobile devices through the chat cloud. The users will get the services like cloud storage, backup and additional API's may be provided which depends on the offerings of thePaaS service provider we choose.

The whole system can be divided in to 3 major components/layers for development:

i. Chatting and Multimedia Sharing
ii. Cloud deployment
iii. Balancing Cloud and Analysis

Chatting and Multimedia Sharing:

In this component developers need to build the necessary chatting and multimedia sharing component. For this developerhave two major projects to focus: the client side and server side applications. Client application will have to pass contents like Name, phone number e-mail address after the application installation and this information is passed to the registration module. Then they will be navigated to their second activity in which they will be displayed a filtered list of users that they can chat with. This list is obtained by filtering fields of their contacts database details and finding the other related users registered with the application. Next they will be provided with a vibrant user interface with options like forwarding to multiple users, tagging their sent messages etc. Now the server side application will take the details from the user's registration activity and pass them on to the database access module where it will also create an instance for the user pointing to the database and backup storage. It will collect a little information like device registration id so that users need not login every time they invoke the client application in their devices. Hence from the second time they start using the application they will be directly directed to the user list activity. The server a will take the backup of their multimedia content at regular intervals and store it in their cloud storage in an area reserved for them. Server code will also be responsible for interfacing between the users and the Paas API services which the application provideracquires from the cloud service provider. Once we successfullydevelop this component with those two applications then we can move on to the second layer.

Cloud Deployment:

Here we are going to select a Cloud Platform as a Service provider. Our server side code will be deployed into the service provider's services. Every Developer can have an evaluation/development account with these service providers and choose the best offerings accordingly. At the service provider's end the developer will need to create an account and register.For the payments, services, billing related information they will be having a Service Level Agreement (contract) between them. The developer will be asked to enter the credit card details for payment which will be charged according to use.The PaaS service provider will be provisioning/charging resources such as CPU usage, memory usage, Storage occupied, Database details and accesses, number of virtual machines being used by the application, Internet traffic and possibly other resource metrics.The developer will scale as per the expected number of live users using the application at one time. This kind of environment will allow the developer to know the most about the application such as features required to improve the user experience, and also they can have an opportunity to utilizethe API's and incorporate them into the chatting application. Example of the API's being offered are Language Translations, Big Query which will be useful for the analysis of the application and also we can have security related API's such as user authentication services. Hence this component will allow us to deploy our server side application and also permit it to use cutting edge cloud computing technologies.

Balancing Cloud and Analysis:

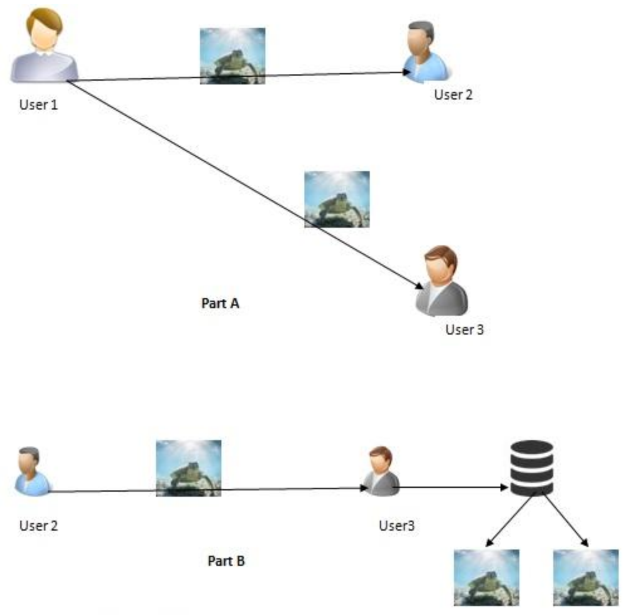

Figure: Depicting problem statement for balanced cloud

The above figure depicts the need of the balanced cloud for storage, without which we will have inefficiency and potentially runaway storage issues. Now let's elaborate the problem and know why we require the balanced cloud in the storage. In the above figure user1 will be sending an image using the chatting application to the user2 and user3 (Part A). If user2 forwards the same image to user3, user2 will be unaware that user1 has sent the image to user3 where there will be a replica of the same image in the storage of user3 (Part B). Let it be phone storage or the free cloud storage being provided to the user it gets consumed as the multimedia data in the chatting will be exponentially trending upwards. We either have to manually observe and delete the replicated multimedia content or have to use an external application to avoid these replications which is not ideal. For the users to delete replicas manually will be very time consuming also sometimes they want to keep the two images as they are being sent from two different users at different time stamps; also they belong to different conversations. Hence the one solution for this problem will be usage of pointers in the database linking single instances of object storage. Now let's explore the solution concept in depth, we will be encrypting the multimedia content; say images, video,etc with the receiving users phone number, store the encrypted data into the database and the received multimedia content into the storage. Hence if we receive new content and once it's downloaded, it will again get encrypted with the receiving user's phone number where this time we compare the content of encrypted data with the encrypted data stored in the database. If we do not find a match then we will store it as a new encrypted data into the database, received content in the storage. Else if we found a match then we will be retaining only one instance of content and instead put in a link to this object in the database. This situation will be occurring whenever the user will be browsing or storingthe images. The information consists of the names of the users sent that content and different date and time stamps they have sent the content. It's true to say this information is also consuming space in the database but we note that it is textual/pointer information will be taking much lesser space when compared to replicas of multimedia content.This way we can also store only one image into the cloud storage space hence it will not get consumed rapidly and also save money to the developer who will be paying to the PaaS service provider for storage.If the user wants to purge the content from the storage or in the chat then a set of options will be provided to them. These options will be presented upon selecting the delete option on the content and the user will be given a list of all the instances of the content stored in the database. For example considering our earlier scenario if we stored only a single image in the user3 storage and put a link in the database indicating that the image has been sent by the user1 and user2,when deleting we will display which instance of the image user3 wants to delete either the instance the image sent by user1 or by user2. They can also have a third option delete every instance and erase complete image from storage. This gives users full control over content which they have custodianship of.Now that we have found a feasible solution regarding the problem and known about balancing cloud we can transit into the next unique feature of the application.

Analysis is an intriguing aspect forthe developers in this advanced chatting application. As the PaaS provider will provide the metricssuch as CPU usage, memory usage, storage being used, number of virtual machines the application is running on, and number of live users we can make use of the data. This data can be represented pictorially such as in the form of graphs, pie charts,etc whereas many PaaS service provider will be offering thesesort of graphical representations of resource usage by default. Using this representation of the data the developer can easily understand the resource trends of the application quickly. They will also clearly able to know the amount they are spending on each particular part of the application which makes maintenances of and budgeting for the application a lot easier. If developers set alarms or notifications to trigger whenever the usage scales up or downbeyond certain preset thresholds they will be getting more insight regarding the application like at which particular timeslots the applications behavior is differing. Hence developer can increase the services being provided by PaaS beforehand if they expect heavy traffic incoming as they are going to keep track of the application usage information. For example on any important days in the calendar like New Year's eve or friendship day we can expect a huge spike in traffic for chatting and multimedia sharing in the application than on regular days. So if we require more resources all we just need to provision extra resources with thePaaS service provider.Some providers also offer a facility whereby resources automatically scale out to meet spikes in demand. If we maintain a history of utilization trending ingraphical representation then when that date and time stamp returns again, we can just order the required amount of resources and be aware of the situation being encountered as we have faced it earlier. Allthis analysis data will be in the developers hand within just a few clicks into the account administration created in the PaaS service provider's site or alternately the developer can persist the information into his/her owned storage. So there will be no need of setting up a huge infrastructure for the chatting application, and it hasbecome a lot simpler to maintain. The cost will be less when we compare this sort of implementation to the traditional way by setting up servers. Not only that we should also keep an eye and give credit to the amount of the strength that a developer is getting upon utilizing and implementing the application using Cloud Computing Technology.

IV. CONCLUSION

The Advanced Chatting application described above will provide significant improvements to traditional chatting applications. The Analysis/software design phase will provide great opportunities to the Developer and utilizing current Cloud technologies will make their work much easier and more productive. In particular current cloud computing technologiessuch as the API'sbeing provided by PaaS vendors will reduce the amount of work required to get the solution up and running.As we can clearly observe this implementation model will allow sandboxing the solution (proofof concept) with minimal outlay.Similarly the proposedchatting application will bring ongoing commercialprofitability by optimizing storage use with balancing/deduplicationof data to minimise storage costs. Existing chatting applications could similarly inherit the features being discussed in this paper into their design and also migrate into Cloud technologies to gain the advantages offered.

V. FUTURE WORK

There is ample scope for Chatting Applications to adopt emerging Cloud Technology features, and thusly improve the user experience, add new functions and allow application vendors to improve productivity and cost effectiveness. For example we can incorporate new features being offered in provisioned networking. This will allow us to move forward and utilize cloud computing services like Network as a Service (NaaS), and similarly we canincorporate features ofcloud Database as a Service (DaaS).If we investigate the analysis sidethere is potential to offer cutting edgefeatures where we can derive major outcomes regarding the data and features in which users are interested. An example feature we could implement would be to modernize chats by text analysis of the chat data which permits features like changing text colours for happy message or sad messages, dates and appointments can have different colours (or even be actioned automatically), etc. Provided user agreements permitted we could also aggregate statistical text analysis of the chat content. For example we would analyse how many users were chatting about politics, shopping, etc. Security aspect can also take a huge leap and can be improved a lot, there will be some basic security being provided by the PaaSservice provider and also we can extend security by providing a security layer to our server side code which will adds to the strength to the applications security. An example would be tying in authentication to another application provider (example Microsoft Live, Facebook, etc) to provide a Single Signon experience for users. Additionally, for entertainment, small games/applets like checkers or chess could be implemented between the users. One can imagine many future opportunities to extend the implementation using Cloud Technologies.

REFERENCES

[1] TarunGoyal,Ajit Singh, Aakanksha Agarwal, Cloud Traun 2012

[2] Minqiang CAI The design of Network Chat System Based on Socket and Cloud Computing 2012.

[3] Google Cloud Messaging https://developers.google.com/cloud-messaging/

[4] Mayanka Katyal,Atul Mishra,A Comparitive Study of Load Balancing Algorithms in Cloud Computing Algorithms http://www.publishingindia.com/