Face Recognition Using Mobile Cloud Computing

Engr 692 Section 62 Fall 2015 – Cloud and Distributed Computing

Final Project – Harsh Patel

***Abstract*— Face recognition applications can benefit from the collaborative coupling of mobile and cloud computing as they become widely available today. This paper discusses my work with the design and implementation of face recognition applications using Mobile Cloud Computing. Despite increasing usage of mobile computing, exploiting its full potential is difficult due to its inherent problems such as resource scarcity, frequent disconnections, and mobility. Mobile cloud computing can address these problems by executing mobile applications on resource providers external to the mobile device. It always been a question that whether to offload mobile application to cloud or not considering the communication cost between mobile device and cloud. The results show that as communication cost in Face Recognition application is really low compared to processing cost, its worth to offload face recognition application on cloud.**

***Index Terms*— MCC-Mobile Cloud Computing, Face Detection, Face Recognition.**

I. Introduction

In recent years, applications targeted at mobile devices have started becoming abundant with applications in various categories such as entertainment, health, games, business, social networking, travel and news. The reason for this is that mobile computing is able to provide a tool to the user when and where it is needed irrespective of user movement, hence supporting location independence. Indeed, 'mobility' is one of the characteristics of a pervasive computing environment where the user is able to continue his/her work seamlessly regardless of his/her movement. However, with mobility comes its inherent problems such as resource scarceness, finite energy and low connectivity. Real time applications, such as Face Recognition are just one type of mobile applications that demand high levels of responsiveness that in turn, demand intensive computing and storage resources.

In recent years, this problem has been addressed by researchers though cloud computing. Cloud computing with resource constraint mobile devices, ubiquitous wireless infrastructure, mobile web, and location-based services provides a ground for a new computing paradigm called MCC (Mobile Cloud Computing). The aforementioned limitations of mobile devices are always obstacles for computationally intensive and storage demanding applications on a mobile. To augment the capability, capacity and battery time of the mobile devices, computationally intensive and storage demanding jobs should be moved to cloud. On the basis of the above discussion, MCC can be defined as:

"A service that allows resource constrained mobile users to adaptively adjust processing and storage capabilities by transparently partitioning and offloading the computationally intensive and storage demanding jobs on traditional cloud resources by providing ubiquitous wireless access".

Careful planning is required before offloading the jobs on a cloud server by considering the network conditions and communication overhead to make offloading beneficial for mobile users. The architecture of the MCC is depicted in Fig. 1.

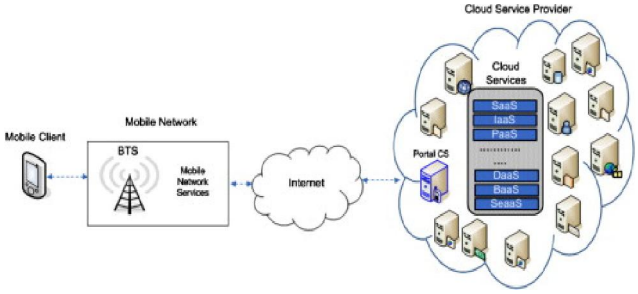

Fig. 1. Mobile cloud computing architecture

Face recognition from image or video is a popular topic in biometrics research. Many public places usually have surveillance cameras for video capture and these cameras have their significant value for security purpose. It is widely acknowledged that the face recognition have played an important role in surveillance system as it doesn't need the object's cooperation. The actual advantages of face based identification over other biometrics are uniqueness and acceptance. As human face is a dynamic object having high degree of variability in its appearance, that makes face detection a difficult problem in computer vision. In this field, accuracy and speed of identification is a main issue.

The following paper discuss about face recognition application in section II, section III discuss advantages of offloading face recognition application on cloud. Section IV provide details about application architecture used in the project. Section V shows performance evaluation results. Section VI is a conclusion phase.

II. Face Recognition Application

Given a picture taken from a Smartphone camera, we'd like to know if there is any person inside, where his/her face locates at, and who he/she is. Towards this goal, face recognition procedure is divided into three steps: Face Detection, Feature Extraction, and Face Recognition.

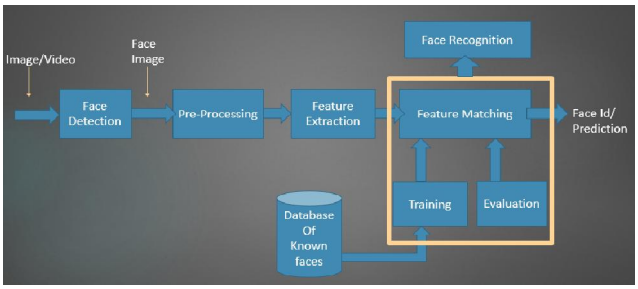

Fig. 2. Face Recognition Phases

A. Face Detection:

The main function of this step is to determine (1) whether human faces appear in a given image, and (2) where these faces are located at. The expected outputs of this step are patches containing each face in the input image. In order to make further face recognition system more robust and easy to design, face alignment are performed to justify the scales and orientations of these patches. Besides serving as the pre-processing for face recognition, face detection could be used for region-of-interest detection, retargeting, video and image classification, etc.

B. Feature Extraction:

After the face detection step, human-face patches are extracted from images. Directly using these patches for face recognition have some disadvantages, first, each patch usually contains over 1000 pixels, which are too large to build a robust recognition system. Second, face patches may be taken from different camera alignments, with different face expressions, illuminations, and may suffer from occlusion and clutter. To overcome these drawbacks, feature extractions are performed to do information packing, dimension reduction, salience extraction, and noise cleaning. After this step, a face patch is usually transformed into a vector with fixed dimension or a set of fiducial points and their corresponding locations.

C. Face Recognition:

The last step is to recognize the identities of the faces. In order to achieve automatic recognition, a face database is required to build. For each person, several images are taken and their features are extracted and stored in the database. Then when an input face image comes in, we perform face detection and feature extraction, and compare its feature to each face stored in the database. There have been many researches and algorithms proposed to deal with this classification problem. There are two general applications of face recognition, one is called identification and another one is called verification. Face identification means given a face image, we want the system to tell who he / she is or the most probable identification; while in face verification, given a face image and a guess of the identification, we want the system to tell true or false about the guess.

III. ADVANTAGES OF FACE RECOGNITION USING MOBILE CLOUD COMPUTING

Face Recognition is a typical application which is well suited to use in a cloud computing environment. There are number of reasons for which we can say I's better to offload face recognition application on cloud rather than just using local mobile platform.

- *Processor and Data Intensive Application:*

Comparing test image with all the images in the face database takes too much processing power. Limited processing power and storage of a mobile device makes the face recognition task very difficult especially when you have large database of known faces. Using large number of available resources on cloud, we can easily provide face recognition on mobile devices.

- *Easy to maintain central database of faces on Cloud:*

With the central face database, it is easy to provide access to number of devices and maintain the database. We can easily insert new face images in the database and keep our application updated.

- *Scalability:*

In some cases, the known face database can grow largely. With limited storage of mobile platform, it is not possible to keep up with the growing database. We can easily manage this difficulty using cloud resources.

- *Low-Cost Solution:*

Taking advantage of cloud providers' on-demand service, we can easily manage varying load of application usage. Also, we don't have to pay for the resources if we are not using it.

IV. APPLICATION ARCHITECTURE

In current architecture of the application mobile clients are represented as a thin client. On the client side image on which we need to perform face recognition is sent using HTTP request. As a response from the cloud side image is received back with the identity of the person in the image if the person is recognized.

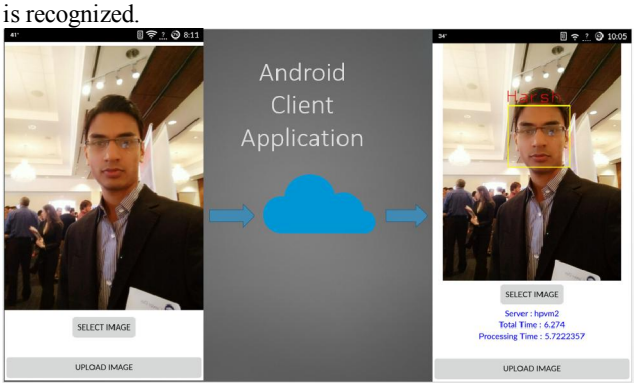

Fig. 3. Mobile Client Application

On cloud side the request is first received by load balancer node. Load balancer directs the request to the server that has

least connections right now. On the server image is processed for the face recognition and result will be send directly to the client from that server.

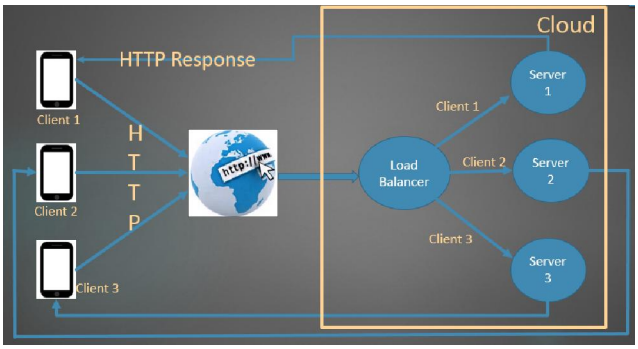

Fig. 4. Application Architecture

V. PERFORMANCE EVALUATION

Client application is developed on Android Lollipop. All the code is written in java. To send the image to the cloud HTTP request is used and result is received using HTTP response. I used Openstack Cloud Computing platform for deploying the application. On the cloud side, all the instances are using Ubuntu-Server-14.04-LTS image. Also, Apache2 HTTPD server is installed on all the instances to handle the http request. OpenCV computer vision library is used to write the code of face recognition. All the face recognition code is written in C++. I used face database provided by AT&T Laboratories Cambridge University to test the face recognition application. The database contains 10 different images for each of 41 different persons. For testing the application I also inserted 10 images of my face. Images were processed and resized to be ready for face recognition purposes.

To decide on whether this application is worth to offload on cloud or not, it was important to check the communication cost. For that I tested how time is taken to send the image to the cloud and receive the response back. I also checked how much time is taken on cloud side to process the image for face recognition. Networking time is time taken for communication between mobile device and cloud. Processing time is time taken to process the image on the cloud side for face recognition. As shown in Fig. 5, the networking time is negligible compared to processing time with even different number of client requests. The results in Fig. 5 proves that the application is worth to offload to cloud.

Fig. 5. Processing Time vs. Networking Time

Load balancing was very important part of this project. To balance the incoming load of client requests I used Openstack's Load Balancer as a Service (LBaaS). I used three different instances to server the client requests. Load balancer is using least connections method for load balancing. I first used all the instances with the same capacity. Table 1 shows the results of load balancing. For 10 client requests Instance 1 has served 3

Number of Requests	Instance1	Instance2	Instance3
1	0	1	0
2	0	1	1
3	1	1	1
4	2	1	1
5	1	2	2
6	2	2	2
7	2	3	2
8	3	2	3
9	2	3	4
10	3	4	3

Table. 1. Load Balancing on Instances with Same Capacity

requests, Instance 2 has served 4 and Instance 3 has served 3 requests. The results show that load balancing is working quite well and distributing same amount of load among the available servers.

In cloud computing environment we might have resources with different capacities. In that case more requests should be handled by the server with higher capacity compare to the other server with lower capacity. For experimental purposes I then tested load balancing using servers with different capacities. As shown in Table 2 hpvm2 has highest capacity among the 3 servers used, while hpvm has the lowest capacity. And as

Instance	Flavor	RAM	VCPUs
hpvm	m1.small	2 GB	1
hpvm2	m1.large	8 GB	4
hpvm3	m1.medium	4 GB	2

Table. 2. Openstack Instances

shown in Table 3. hpvm2 has handled most requests compared to other instances. Even comparing hpvm3 and hpvm, hpvm3 has served more requests as it has a higher capacity than hpvm. Fig. 6 shows the results for using multiple instances to serve the clients request and using single instance to server same amount of requests. The results show that because of lower capacity hpvm takes very high time when handling multiple

client requests at the same time. While hpvm2 takes comparatively less time even compared to three instances handling multiple client requests.

Number of Requests	hpvm	hpvm2	hpvm3
1	1	0	0
2	0	1	1
3	1	1	1
4	1	2	1
5	1	2	2
6	1	3	2
7	2	3	2
8	2	4	2
9	2	5	2
10	2	5	3

Table. 3. Load Balancing on Instances with Different Capacity

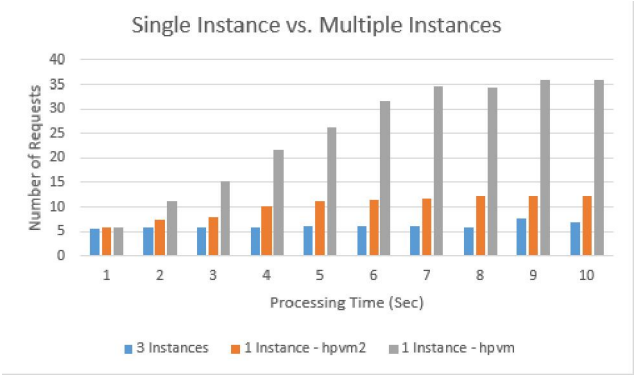

Fig. 6. Single Instance vs. Multiple Instances

VI. CONCLUSION AND FUTURE WORK

Before offloading the application to the cloud we should always check it is worth to offload the application or not. As per the results shown in performance evaluation and also some other reasons shown in section III, we can say that offloading face recognition to cloud is better choice than using it only on local mobile device. Cloud environment helps the application to scale when the face database grows or number of client requests increases. Handling multiple requests from a single client and performing face recognition on video file is included in future work.

ACKNOWLEDGMENT

I would like to thank Dr. Philip Rhodes for his assistance throughout my project work and constructive comments. It was a great opportunity to work on a real cloud computing environment and I learned lots of new things during this project. I really appreciate Dr. Rhodes's effort to get us resources on Chameleon Cloud so that we can deploy our application on real cloud environment.

REFERENCES

[1] https://wiki.openstack.org/wiki/Main_Page

[2] http://opencv.org/

[3] https://help.ubuntu.com/lts/serverguide/httpd.html

[4] Android Developers guide https://developer.android.com

[5] Shishir Kumar, Priyank Singh, Vivek Kumar, "Architecture for Mobile based Face Detection / Recognition" International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010, 889-894

[6] Abdul Nasir Khana,∗ , M.L. Mat Kiaha , Samee U. Khanb , Sajjad A. Madani c, "Towards secure mobile cloud computing: A survey", Future Generation Computer Systems 29 (2013) 1278–1299.

[7] Marco V. Barbera, Sokol Kosta, Alessandro Mei, and Julinda Stefa, "To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing," 2013 Proceedings IEEE INFOCOM.

[8] Pragya Gupta1 , Sudha Gupta2, "Mobile Cloud Computing: The Future of Cloud," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 1, Issue 3, September 2012.

[9] Prasetyawidi Indrawan1 , Slamet Budiyatno2 , Nur Muhammad Ridho3 , and Riri Fitri Sari4, "FACE RECOGNITION FOR SOCIAL MEDIA WITH MOBILE CLOUD COMPUTING," International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol.3, No.1, February 2013.

[10] Sujata G. Bhele1and V. H. Mankar2, "A Review Paper on Face Recognition Techniques," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)Volume 1, Issue 8, October 2012.

[11] Tolga Soyata∗, Rajani Muraleedharan∗, Colin Funai∗, Minseok Kwon†, Wendi Heinzelman, "Cloud-Vision: Real-time Face Recognition Using a Mobile-Cloudlet-Cloud Acceleration Architecture," Conference: Computers and Communications (ISCC), 2012 IEEE Symposium on.