Lost Child

Krishna Aditya Gabbita[1], Sahil Shah[2]
1[Computer Science Department, CMU], 2[The Robotics Institute, CMU]

INTRODUCTION

Finding a missing child in the crowd is a crucial and non trivial task. This project aims to reduce the burden on anxious parents and alleviate their unpleasant experience.

- **How do we help?**

The Parent uses the Lost Child app on her/his phone, to upload 10 recent pictures of the missing child and request for help. An user of the app in the crowd will stream video via a Google Glass/ phone, from which the child is located and the parent is notified with the image and location.

- **What do we aim to learn?**
  - Validate if ***cloudlets*** as backend cognition engine can help in reducing latency, while performing computationally intensive tasks
  - Evaluate performance for cloudlets vs public cloud
  - Limitations of wearable computing in continuous usage (video streaming, battery life)

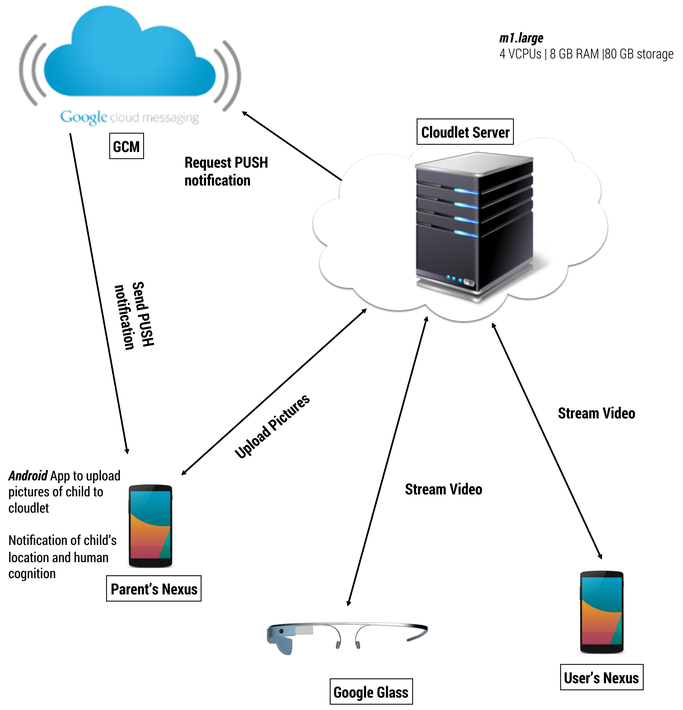
- **Tasks**
  - Build a face recognition model (from the child's images)
  - Stream video from Google Glass to the cloudlet server
  - Find child in video frames from Glass/phone
  - Send notification (with location of missing child) to parent's phone

- **Devices**
  - Google Glass
  - Android Phone
  - Cloudlet Server

USE CASE

- Parent launches the Lost Child app on her/his Android phone and ***uploads*** 10 recent pictures of the missing child to the cloudlet
- The server builds a ***face recognition model*** of the child, and notifies the users of the app in the area to help find the missing child
- Users volunteer to ***stream video*** of the crowd using their Google Glass or Android phone to the cloudlet
- The cloudlet server finds the child in the video frame and sends the location with the frame to the parent
- Parent confirms the found child, and the server notifies the user to stop streaming

SYSTEM ARCHITECTURE



SOFTWARE STACK

- **Cloudlet Server**
  - Resource Management – ***OpenStack***
  - Video streaming backend – ***Gabriel***[1] server
  - Cognitive engine – ***OpenFace***
  - HTTP server – Flask

- **Push Notifications**
  - Google Cloud Messaging

- **Phone and Glass**
  - OS – Android
  - Streaming front end – Modified Gabriel client (on Android API 19/GDK preview)

PERFORMANCE EVALUATION

Ping times (in milliseconds)

	Min	Max	Mean	Std Dev
AWS	73.159	148.477	77.732	8.999
Cloudlet	1.554	95.974	5.844	10.113

Latency (in milliseconds)
AWS c3.2xlarge – US West (Oregon)

	Min	Max	Mean	Std Dev
Face	1205	1754	1511.667	210.266
No Face	176	2980	1347.505	790.224
Face + Push	1524	1864	1640	194.031

Cloudlet

	Min	Max	Mean	Std Dev
Face	356	461	405.75	29.988
No Face	87	442	170.235	62.536
Face + Push	402	556	496.833	59.65

CONCLUSION

- Mobile and wearable devices are resource poor, and cannot perform computationally intensive tasks (face recognition) locally. Although hyper scale public clouds overcome computation constraints, they aren't a suitable choice for latency sensitive applications.
- Cloudlets are a viable alternative to perform computationally intensive tasks with a lower latency than hyper scale public clouds, due to their presence at the edge of the networks.

ACKNOWLEDGEMENTS

We are thankful to Prof. Satya and Prof. Dan for this insightful class on Mobile and Pervasive computing. And, we're thankful to Dr. Khalid Elgazzar for mentoring us throughout the project. Also, we're grateful to Prof. Satya's lab and Kiryong Ha for his Gabriel implementation, and Brandom Amos for his OpenFace implementation.

REFERENCES

[1] Ha,K., Chen, Z., Hu, W., Richter, W., Pillai, P., Satyanarayanan, M. Towards Wearable Cognitive Assistance. *Proceedings of the Twelfth International Conference on Mobile Systems, Applications, and Services.* Bretton Woods, NH, June, 2014.