

20181114 Meeting with Amity

Agenda

Firestore

Firestore Application: The need for a proxy server can be completely eliminated by shifting to a cloud-based architecture for managing files. Instead of SLAC pulling in the generated txt files from MAs around the world, the MAs can themselves push these files to a centralized application server hosted on the cloud; which makes it easier for SLAC to access files as per their need.

Considering the great degree of integration capabilities Google offers with its products, the flexibility arising from using the Google Cloud Platform would be pronounced. Google's Firestore is a mobile and web application development platform that provides developers with a variety of tools and services to help develop high-quality apps, that can scale easily as per changing demands, and which delivers 99.99% uptime. The Firestore SDK allows mobile app developers to quickly add critical and reliable functionality to their applications in a short time. The recommended option here would be to leverage the Cloud Storage for Firestore that allows robust uploads and downloads onto the Google Cloud Storage buckets. Apart from the user authentication module that comes bundled with Firestore, developers can also declare file security parameters so as to allow only certain file types to be uploaded. Having all user uploaded files in one place will then enable SLAC to access and process files as and when needed. Server-side processing can be done on the Google Cloud Platform as well, thereby eliminating any need for SLAC to maintain its own physical infrastructure. This form of implementation can be highly beneficial in any kind of region of the world, also including remote places, like deep inside tropical rainforests, or places that have recently been hit by a natural calamity. As this implementation model is not dependent on any local measuring agent, the android mobile apps can directly deliver data to SLAC over any form of internet connection in minimal time with high reliability.

At our recent PingER meeting we discussed the suggestion made in your PingER paper on using Firestore. We had a few observations, questions, and a suggestion:

- We agree it is another reduction on independence from SLAC, which is good
- It is unclear whether it would work for China since Google services are not available there.
- Is it only free for low utilization?
 - Might work for a cache for the latest data, but doubt it would work for all archived PingER data (hundreds of GB).
- It was developed for Android applications, might be good for Android/PingER
- There may be traffic limits.
- Umar suggests try for one node and see how it works out in practice

Status of Amity MA

The Amity MA is unreliable so using it for a case study does not appear fruitful. Les is working with Amity to try and understand this unreliability (emails 10/6/2018), they say "There is an internet problem we are taking care of it and will get back soon". **The ability to gather data from it is very intermittent, It runs fine for several days, then there is no data for many days.**

Deploying Android PingER

The Android version of the PingER MA, is described with comments at [ePingER on Android Native - Amity project](#) (this is a proposal/description from Aayush Jain)

- It describes a multipurpose, stand-alone device that can be widely distributed, something that we have brainstormed about for a long time.
 - We agreed to request Amity to share the App and instructions with us; we will look at installing on a jailbroken Android phone at the San Francisco end and try it out.
 - They will be attaching the android app for review after testing it again.
 - Topher feels that the Android app (when completed and vetted by his team) could easily be installed as a default service on his rainforest monitors (certainly future ones, not devices already in place). Merging the service data that he already collects with that unique to PingER has the potential to lead to some interesting results.
- The topics are:
 - Amity's documents describe a multipurpose, stand-alone device that can be widely distributed, something that we have brainstormed about for a long time.
 - Bebo mentioned it to Topher (the Principal Investigator (PI) of the RainForest project) and he feels that the Amity PingER/Android app (when completed and vetted by Topher's team) could easily be installed as a default service on his rainforest monitors (certainly future ones, not devices already in place). Merging the service data that he already collects with that unique to PingER has the potential to lead to some interesting results.
 - How should we proceed?
 - We (Topher and the PingER team) agreed to request Amity to share the App and instructions with us; we will look at installing on a jailbroken Android phone at the San Francisco end and try it out.
 - We also need to finish up the proxy acquisition of data from the Android MAs. A first step is outlined in <https://confluence.slac.stanford.edu/display/IEPM/Proxy+support+for+PingER>.
 - A later step is to look at, evaluate and find resources to implement the ideas expounded in the Future Proposals section of the paper you are working on (see below)