

## 1 Summary

The cloud computing / distributed computing systems have made it possible to extend both functionality and battery life of mobile devices. The storage element no doubt plays an important role in these computing modals. A tool that is able to access and manage the files on a remote system will be of great help on the Android platform. Our plan is to implement a synchronized user space file system so that we could use the remote computing platform to make the android devices to be a more powerful palmar workstation.

To serve the need of remote computing task, the application should be able to provide a file system to be mounted into Android, capable of operating on files on a remote computer. Further more, the user should be able to mirror selected files and folders into local file system as cache, i.e internal storage or external storage card. These cached files will be synchronized with remote system. Then, the final goal is to allow files created or modified on Android be able to sync to remote storage system and processed by services on remote server, then the result stored in remote storage system sync back to Android.

For market reasons, there are several cloud storage providers using different interface / protocols. At the initial launch of our application, we avoid the mass of supporting these incompatible systems. We choose to implement our system for the widely used SSH protocol. Through sshfs, we will be able to mount the remote unix system and through ssh we will be able to launch the remote services. Still, when we construct our application framework, we intend to use the module / plugin structure. This mechanism will allow us to introduce the support for other cloud computing platform later.

## 2 Milestones

### M1. Integrate FUSE and sshfs

The first step will be to port FUSE and sshfs to android. As FUSE and sshfs are native linux programs, the difficult part will be to compile and install FUSE as a kernel module. Also, we need to implement a GUI capable of setting/saving configurations and invoking the sshfs process.

### M2. Mirror, Sync and Interface

After integrate the FUSE and sshfs into android system, the next step will be implement the functionality of mirror part of the remote file system into local cache. At this milestone we only plan to implement a system service to keep cache in a separate place other than the mounted file system. The system service also carry out the task of keeping cache synchronized with remote storage system.

### M3. Remote Execution

To achieve our goal, we need to figure out a way to communicate with the remote server. At this point, we use SSH protocol to execute shell command on remote servers. Before the execution local file system will “commit” to keep remote system up-to-date, and after execution, local file system will “update” to get the result.

### ME. Future features

Consider this is a one-month project, we have to give up some great features in the 1st release, such as

1. Simple Editor
2. Cache file within mounted file system
3. Multiple platform support