ANDROID APPLICATION CRASH AND USER BEHAVIOR ANALYSIS

With the development of mobile networks, the mobile Internet connectivity becomes universe, which also motivates a great surge in the availability of various mobile devices, especially Android devices initially developed by Google. Nowadays, every Android device, like smartphone and tablet runs tens or even hundreds of mobile applications, also terms "App" in short. Though some these apps are prebuilt in the mobile operating system or provided by the device vendors, most of the installed apps are 3rd-party apps downloaded from online app market websites.

It is quite common that app developers want to retrieve feedbacks about their apps from users' devices, to improve app design, adjust operation strategies and improve user experience. This feedback phase serves as an important stage in mobile app developments.

Due to the openness nature of the Android ecosystem, there exist a large number of different Android devices with slight difference from the official distribution, also termed as the "Android fragmentation problem". App developers face the challenges to develop an app that can adjust to hundreds of different device models. Whether the app would crash on certain device at certain environment is a vital information for the developers to improve app quality.

This thesis introduces Appetizer, a system that collects app runtime behavior, crash and lag information. Appetizer serves as a lightweight development kit, being integrated into developers' apps. It also has a server side unit that process incoming data and render statistics for developers.

The design features, implementation and comparison with various off-the-shelf commercial solutions are presented in this thesis. The result showed that Appetizer cover wide functions in Android device information collecting, Appetizer SDK size is significantly smaller than commercial solutions, overhead won't decrease the user experience.

Typically, each smartphone running tens to hundreds of applications, shortly named App. Part of which is manufacturing equipment manufacturer or operating system manufacturer customize Android App, but the user spends more time playing with third-party application manufacturer or independent developers to develop and requires an Internet connection of App.

In the Android ecosystem, the developer after the completion of Android App development, such as through Google Play Store, Tencent application market, XiaoMi application market and other Android App Store application market release App, those application market will check developers' applications have submitted security, content and audit procedures on the stability, approved, users then download, install and use the application from App market.

Obtain feedback from the user, based on feedback information to improve App design and experience, adjust the way they operate is a very important aspect in the mobile Internet ecosystem, because the user experience a direct impact on the number of users of App.

Some user can grade and comment applications market reflected as feedback, but such feedback way only including little information, only a simple text description and evaluation scores, can only play a reference role to other users, it is difficult to obtain more in-depth analysis of value data, the smaller the value of the development team. In addition, feedback from the Android application market has been subject to the application store, also exist malicious competition and other issues, what's more, the developing team can't fully control those feedbacks.

A major problem in Android operating system is the fragmentation of the various mobile phone manufacturers worldwide have countless different models of mobile phones running different versions of the Android operating system, these different phone hardware parameters, different system versions. Thanks for the "Open Mobile Alliance (Open Handset Alliance)", the existence of the provisions of the equipment running the Android system will need to meet certain compliance requirements, which makes the Android App can be run on most Android devices comply with norms, certain enhance the this loosely Android camp.

Android fragmentation problems including device fragmentation and system version fragmentation. Device fragmentation refers to the different types of equipment from different manufacturer, different hardware configurations. Such as Android phones use a variety of screens, these parameters include screen size, resolution, screen capacitive and resistive screen, support pressure, multi-touch support ceiling for different screen parameters for App experience is not same, in particular resolution directly related to the content displayed in App front end display how much contents.

In addition to the screen, CPU, GPU, memory, video cameras, and so many other hardware like flowers bloom together, the result is the same App runs on different types of Android devices, the effect is totally different, it's impossible to test an App is suitable for "all" devices.

System version fragmentation refers to different devices running different versions of the Android system, different versions of the system and the number provided by the API library implementation is not the same, larger changes in the underlying mechanisms of the system version number will be different for each Android App specify the range of the system can run major version number.

Android system version branch is very complex, the current situation is dominated by Google's "Android Open Source Project", shortly named AOSP, the branch is open to the open source community, and was sees as the purest Android branch, Android system is the major version number in the usual sense is followed by AOSP branch.

Google's Nexus series of devices running the Google Inc. combines AOSP and not open source Google Framework of ROM, commonly referred to as native Android ROM. But in China, the use of a wider range of users is the major manufacturers to modify AOSP based custom Android operating system, MIUI millet company is represented. App can run on most of the AOSP also able to run on these systems, but these systems in some places on the AOSP source code has been modified, the behavior is slightly different, these differences prone to unexpected problems.

For software developers, the App achieve collect usage from all user devices, you need to consider large-scale concurrent, persistent memory problems, stability, high degree of difficulty and workload. For large team, they may be able to achieve collect feedback function specially, but for personal developers who want to learn the status of users' behaviors, such a special function is too large, even heavier than the App major business function itself.

In addition, developers will have to face hardware fragmentation problem, running various versions of the system user equipment to ensure the development of the App does not crash on some special Android devices, and have a more positive user experience. During the testing phase

development team can’t cover all of Android devices, what’s more, the test is difficult to cover all situations.

There is a case, that an App can run on most devices can run correctly, only a section of users in a few Android devices will collapse, resulting in this part of the user can not use. If this device can send App crash information to the developer, the developer will be able to locate and solve the problem, and updating the App improve the compatibility and stability.

This article describes the tool, code-named "Appetizer", to solve the problems described above. "Appetizer" includes an Android application can be integrated in a lightweight software development kit, referred to as the client SDK, and has good scalability for receiving and processing data in the server program.

Appetizer client SDK can gather user using the operation time App paths and collected when App crash program call stack and device status information to detect Android application not respond, recording black and white screen every time App startup, storage those information, send the information to server when and network conditions allowed.

Appetizer server capable of storing persistent client message sent SDK for App use information, and make processing and statistical analysis, to calculate and show valuable content to the App development team and operations team, including daily active users, crash information classification, App device distribution and other processed data.

Appetizer client SDK is characterized as lightweight and stable, Appetizer client SDK don’t use any third-party libraries, minimize SDK size, reduce the use of the SDK App size increase. For stability, will not cause a App crash problem because of the SDK itself crashes. Appetizer server architecture ensures good scalability, Appetizer server can withstand greater client pressure by simply increasing the number of servers.