Sleep Conflicts

Adriana Adam

Andrei Fangli

Software Engineering - 248

This essay shows a new class of energy bugs called sleep conflicts which can happen in smartphones device drivers. In a more elaborate way a sleep conflict will happened on our smartphones when a component in a high power state is unable to transition back to the base power state all because the system who is suspended when the device driver code responsible got driving the transition is supposed to execute. There is a root who is causing the sleep conflicts and because of that root there are for types of sleep conflict. In order to have them under control is has been implemented a system that perform sleep conflicts avoidance for Android smartphones.

Mobile phones have emerged as the fastest growing computers and for many people are their primary or only computing device and unlike previous computers the normal mode of operation is disconnected from an external power source and the users expect them to run for at least a day before the batteries are gone. For this thing could happen there were made some changes in the power management model. First, after a brief user inactivity the phone enters the default state of system suspended and in this state the phone drains close to zero battery power and second the developers of apps and device drivers are expected to perform the power state management of individual phones devices (GPS, Wi-Fi…) to ensure that when a change in the power state of a device is needed the SOC is on and the CPU can be woken up even if it is in a low power sleep state. So programmers were forced to think about the power management as a first order concern for the correct behavior of apps.

When a high power state is unable to transition back to the base power state because the CPU is asleep, a sleep conflict has occurred and it will drastically reduce the standby time of the phone, the standby power consumption can be increased by 10 to 25 times. To support background services under the aggressive CPU sleeping policy here eve been provided 3 general mechanism

to keep the system suspending or to wake it up. This means that the mechanism can be used to prevent the system from being suspended and therefore to prevent a sleep conflict. Wakelocks is one of those 3 mechanism and it is used for preventing SOC suspend, if any process acquire a wakelock, the SOC cannot suspend. Another mechanism is suspend notifier and it is a callback function, which maintains a list of all the notifiers and when the power manager initiates an SOC suspend process, the first thing it checks if any wakelocks is not release which would abort the process, or otherwise the list of all suspend notifier called one by one and if none of the suspend notifiers return an error, the SOC is suspended. Hardware wakeup provides the mechanism for waking up the SOC from the suspend state and it is essential to support services that should turn at fixed time intervals.

In order to gain insight to the root cause of sleep conflicts bugs it has been made a classification with all the possible scenarios where a sleep conflict could arise. For example, such a scenario could be waiting for a timeout transition. This implies that after a period of active utilization there are some ways that a device could be transitioned to a suspended power state. A good example is the one with the wakeup timer. If the CPU is asleep when the device is to be suspended, the wakeup timer was a software times or a high-resolution timer, neither of which can wake up the system from suspend, the device driver code cannot run and hence cannot put the device back into the suspend state. When this happens the device can stay in the tail power state indefinitely, leading to a sleep conflict.

To solve the problem with the sleep conflicts a solution has been created and that one was Hypnos. Hypnos is a runtime system that performs sleep conflict bug avoidance in smartphones with Android device drivers. It also provides support for the both debugging device drivers before they are deployed and for catching sleep conflicts in smartphones as they happened to avoid the power loss resulting from a sleep conflict. In order to both detect sleep conflict before deployment and to avoid sleep conflicts in deployed system, Hypnos will have to check to see if the necessary conditions for a sleep conflict can exist. There are two condition that should be checked. The first one is to check if all devices are already in the suspend state and the second one if some are not to determine if their drivers have invoked some mechanism to ensure that the system will wake up at the intended time for the power state transition.

For Hypnos to enable the checking of the first condition, the runtime system will maintain a data structure containing the current power state of all the devices with their power model. Secondly to enable checking of the second condition the followings action are taken. At system boot-up the Hypnos runtime system registers a suspend notifier for itself with the kernel power management. In particular, it makes sure that this notifier is registered after the suspend notifiers of the wake lock implementation and after all of the devices have been registered. This will ensure that Hypnos's suspend notifier will run only after the notifiers of all the devices and the wakelock implementation have already run and given the "green light" and hence the system is really about to suspend.

Our opinion regarding this theme, more precisely about the proposed solution for the sleep conflict bug problem is that Hypnos is able to detect several sleep conflict bugs and prevent them from draining the battery which is an important issue for most of the people, and is important that it has been developed because we need to have our smartphone available for more than a day and for this could be happening as a first step in this technology Hypnos has been created.

As a possible suggestion for this study is that Hypnos is being used only on smartphones who are using Android as an operating system. There are existing many users who are using phones with IOS or Windows Phone, which are facing similar problems, as the ones with Android phone users. So as a next step for Hypnos it should have an upgrade and try some others operating systems.