Контейнерная виртуализация Android

2 ноября на SECR'2012 представлен доклад Е.Баталов, А.Карташов, К.Кринкин Технология контейнерной виртуализации для платформы Android

Виртуализация широко применяется для настольных и серверных решений и начинает завоевывать область мобильных устройств. В статье рассматривается реализация контейнерной виртуализации для устройств на базе Android. Рассмотрены аналоги (Cells, VmWare Horizon Mobile, TrustDroid, EmbeddedXen). Для построения контейнеров применен механизм linux containers (LXC).

Предлагается оригинальный подход к виртуализации телефонии, аудио- и видео- устройств. Предлагаются политики для маршрутизации входящих и исходящих вызовов, а также схема мультиплексирования звука. Вводится понятие активного контейнера Android, который по-умолчанию, принимает весь пользовательский ввод, осуществляет вывод графики на видимую область экрана, принимает входящие и отправляет исходящие сообщения и звонки. Неактивные контейнеры выполняют приложения в фоне, звук может быть микширован. Пользователю предоствляется интерфейс переключения активного контейнера. Описывается стратегия тестирования производительности и проводится анализ потребляемых ресурсов.

Virtualization is widely used for desktop and server systems and in several years it comes into mobile world. The paper suggests approach for container virtualization for Android operating system. Analogues (like Cells, VmWare Horizon Mobile, TrustDroid, EmbeddedXen) are described, and their advantages and disadvantages are considered. Approach suggested by authors is based on linux containers (LXC) mechanism. It allows virtualization of process identifiers, network resources and also can be used for resource management on top of cgroups. For resource management in multi-container environment new supervisor (AndCont) has been developed. Also suggested solution includes modified Android binder driver for inter-process communications; components for multiplexing user input; GPU and frame buffer virtualization scheme; proxy-based solution for in-coming and out-coming messages and phone calls. Active container is introduced to define Android OS instance which is holding user input queue and able to use physical screen for output graphical content. Other (non-active) Android instances do output into virtual buffers which are not visible for users and they are able to run applications in background. Special proxy-based layer has been developed for telephony virtualization. It includes wrapper around native (proprietary) modules for radio-interface features management and set of rules which are defining calls routing scheme. Sound can be mixed from several applications from different containers but during a call audio tract will be monopolized by telephony. On the power management side in paper suggested two solutions: wake_lock() and wake_unlock() virtualization too keep sleeping state for each container; wait_for_fb_sleep event emulation for rendering interface notification to avoid user interface updating. Most of fixes in regular android drivers (e.g. Alarm, Audio etc) have been made by adding unique state context for each container and adding device wide event and data multiplexor. In paper performance testing strategy and scenarios are described; results are discussed. Adding containers has low impact on battery power consumption but approach should be improved for better memory usage. It’s possible to run two Android with traditional applications (games, players) but set of devices is limited for that. Performance tuning will be continued by authors.

Ссылки

- Слайды: http://www.slideshare.net/osll/android-15003767
- http://www.youtube.com/watch?v=0_PsQb4WnT8&feature=youtu.be
- http://16918.selcdn.ru/androidvm/androidvm-demo.mkv
- http://16918.selcdn.ru/androidvm/demo.m4v

android,, linux,, lxc,, secr2012,, mobile virtualization

From:
http://wiki.osll.ru/ - **Open Source & Linux Lab**

Permanent link:
http://wiki.osll.ru/doku.php/etc:blog:%D0%BA%D0%BE%D0%BD%D1%82%D0%B5%D0%B9%D0%BD%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B2%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_android

Last update: **2012/11/03 05:28**