MultiDroid: A Novel Solution to Consolidate Interactive Physical Android Clients on One Single Computing Platform

Bin Yang **Intel R&D Center**

Shoumeng, Yan **Intel Labs**

Agenda

- Background and Scenarios
- Solution
- Evaluation
- Summary and Next

Background

- Leverage computing power of modern CPU to consolidate workloads
- Requirements come from different scenarios such as Retail, Automotive and Game cloud.

Scenario: Rear Seat Entertainment

- HVAC
- Video
- Music
- Game
- Map
- Conference

Scenario: Cloud Gaming

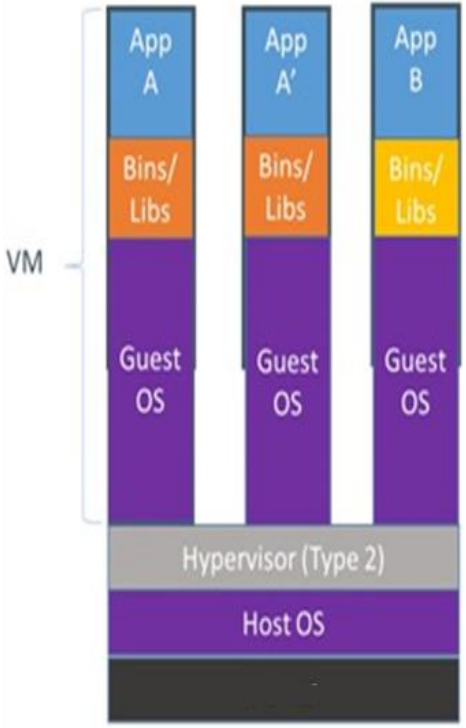
Scenario: Robot

Scenario: Auto Vending Machine

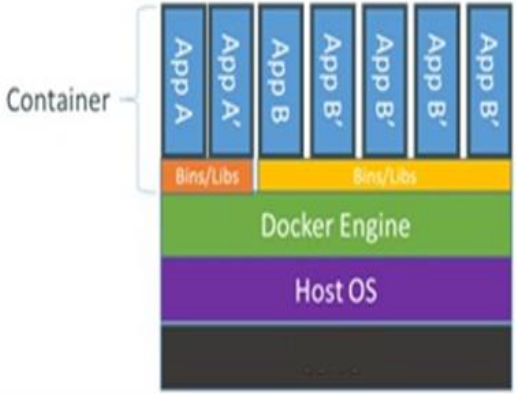
Solution - *MultiDroid*

- Solutions Comparing
- Architecture
- Advantages
- Implementation

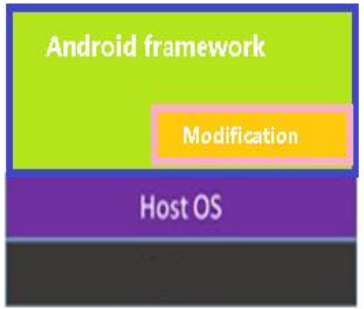
Solutions Comparing



Containers are isolated, but share OS and, where appropriate, bins/libraries

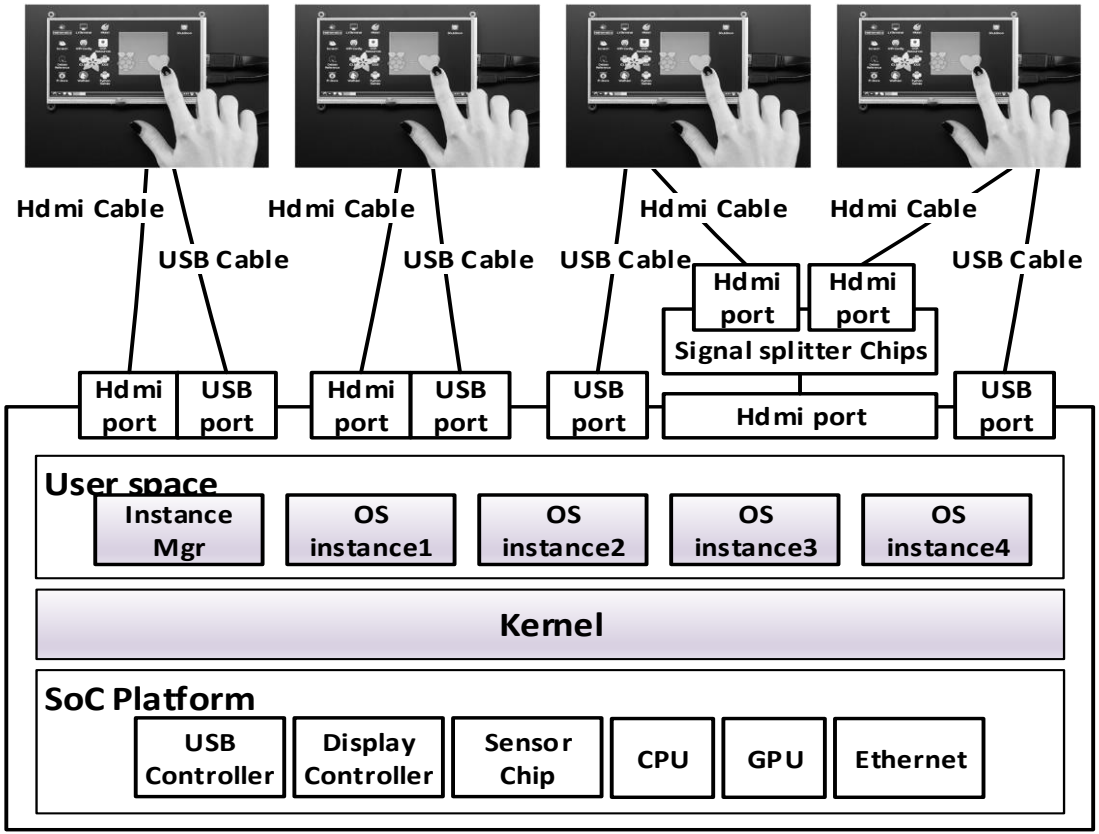


Modify framework



MultiDroid

Architecture



Advantages

- Lightweight
- Supports different Linux-based OS
- Little overhead.
- Simple I/O sharing solution.
- Recovery mechanism
- Security Guarantee
- Easy Deployment & Maintenance

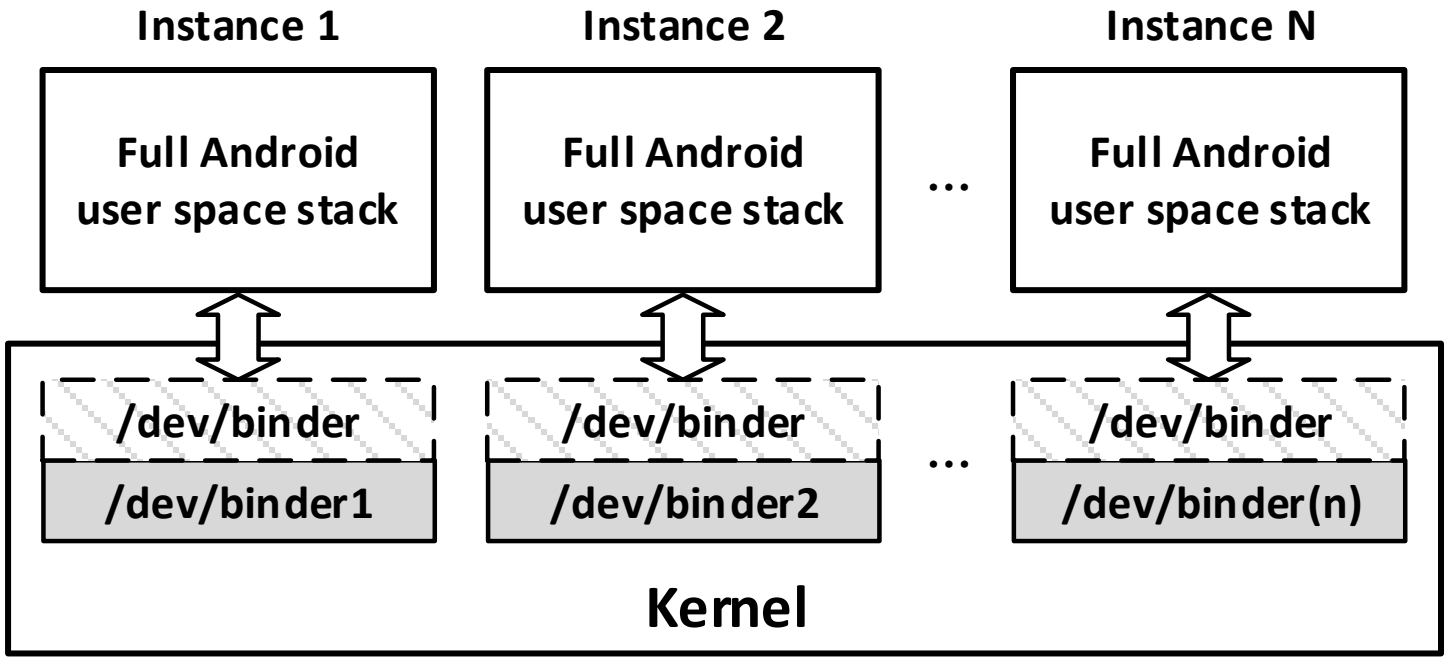
Implementation: MultiDroid

- File System Layout
- Kernel
- I/O Devices Virtulization
- AOSP

File System Layout

- Different data partitions for different Android instances.
- system.img is shared between different Android instances.
- Overlay fs can be used if different system.img is required.

Kernel - Binder



I/O Devices Virtulization

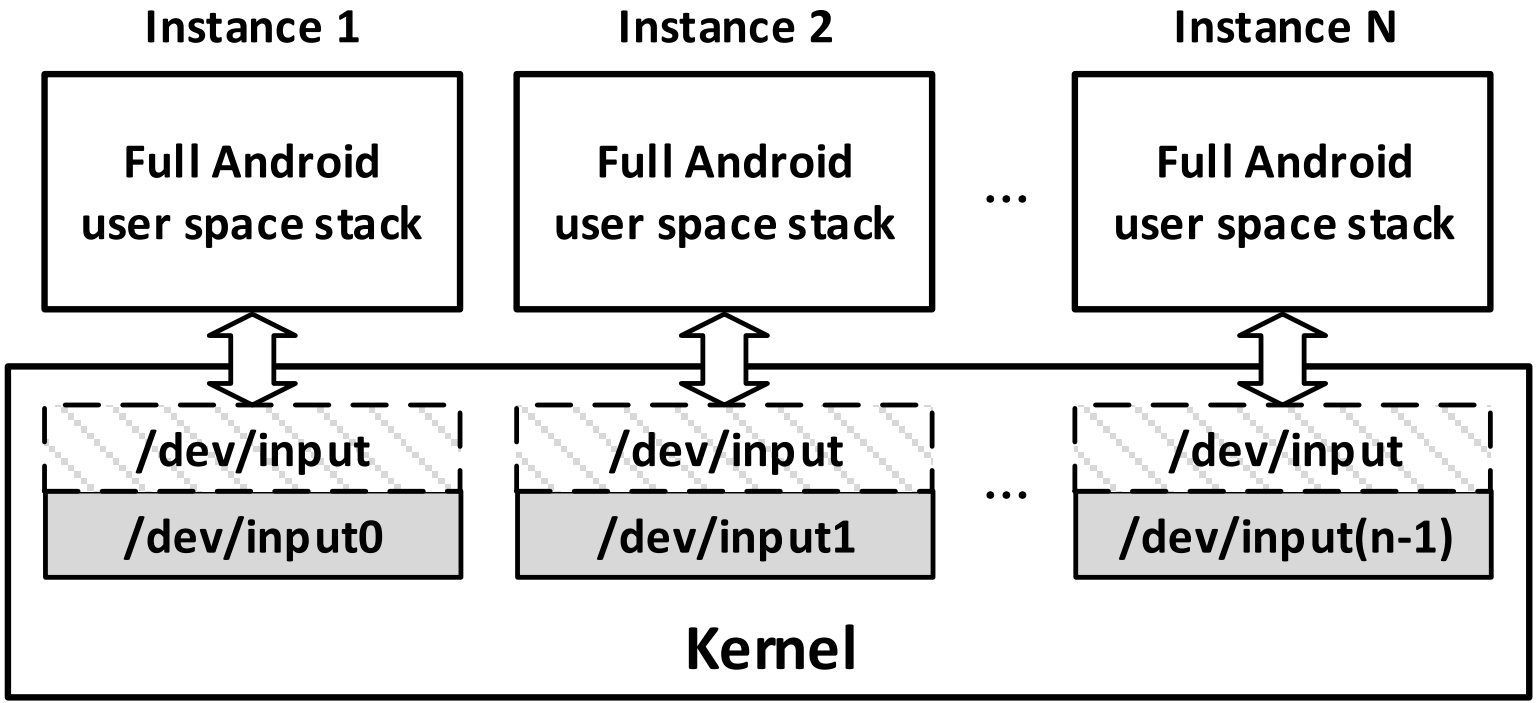
- Dedicated I/O devices

  e.g. Usb Host

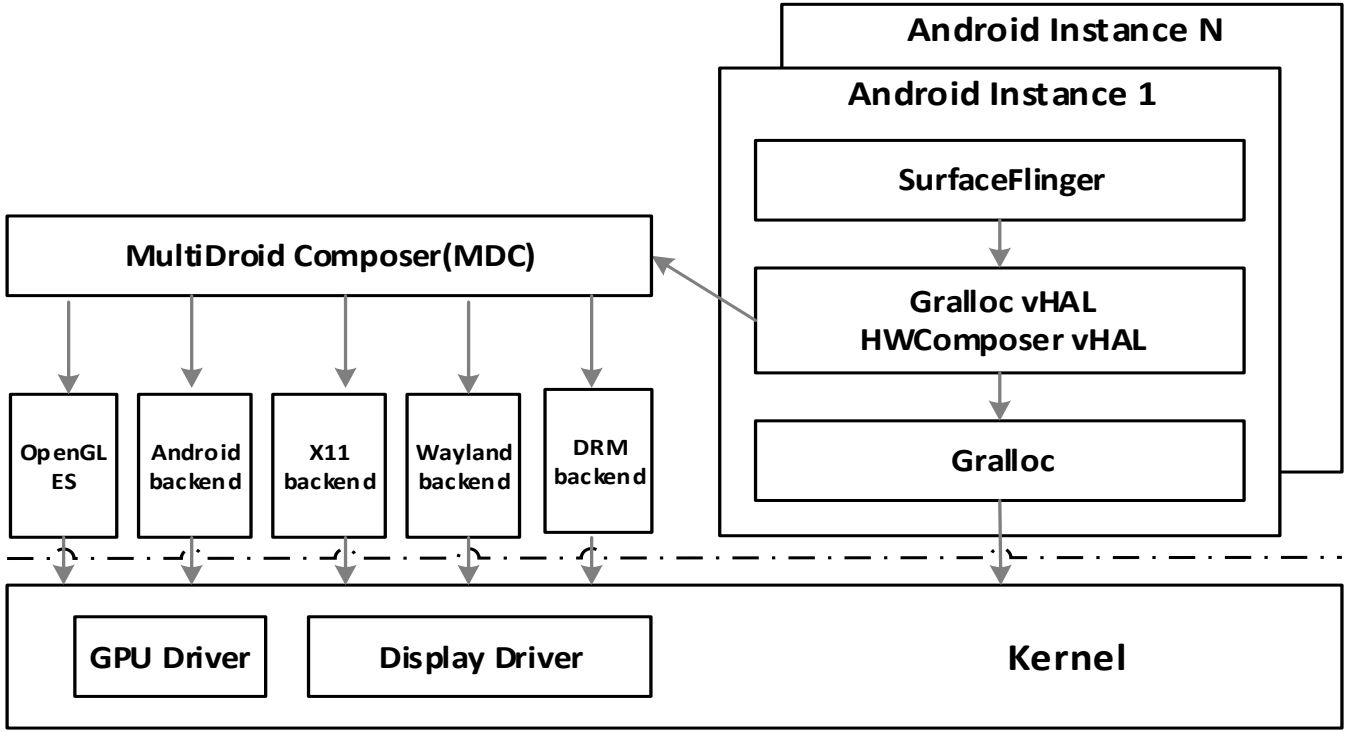
- Shared I/O devices

  e.g. GFX, Trusty

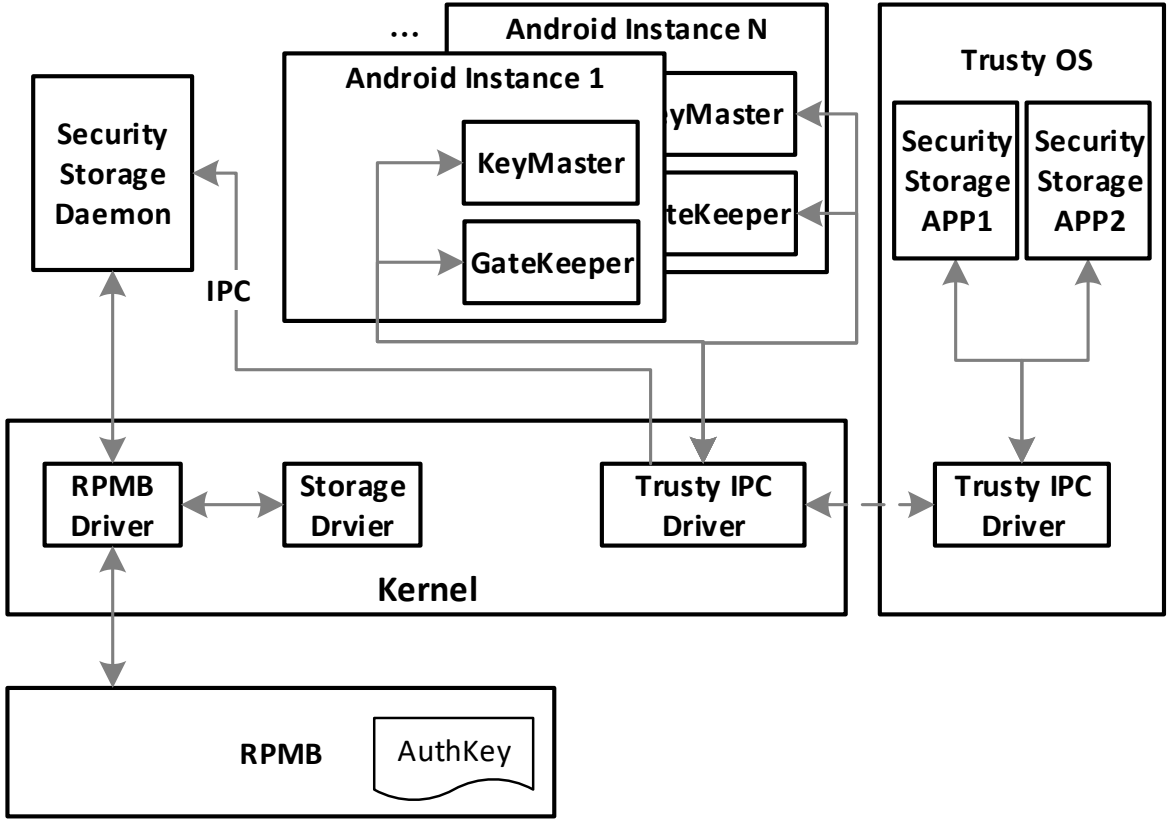
I/O - USB Touch


Instance 1
Instance 2
Instance N
Full Android user space stack
Full Android user space stack
...
Full Android user space stack
/dev/input
/dev/input
/dev/input
/dev/input0
/dev/input1
...
/dev/input(n-1)
Kernel

I/O – GFX and Display



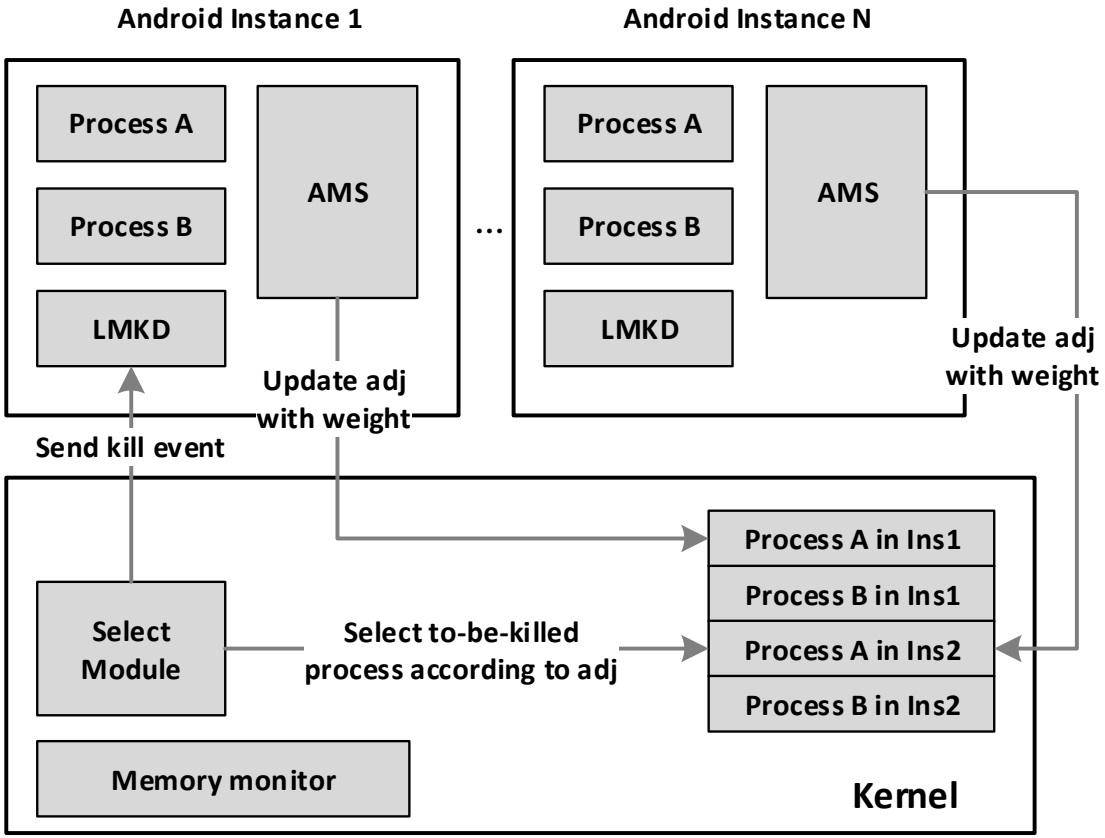
I/O - Trusty


...
Android Instance N
Android Instance 1
yMaster
KeyMaster
teKeeper
GateKeeper
Security Storage Daemon
IPC
Trusty OS
Security Storage APP1
Security Storage APP2
RPMB Driver
Storage Drvier
Trusty IPC Driver
Trusty IPC Driver
Kernel
RPMB
AuthKey

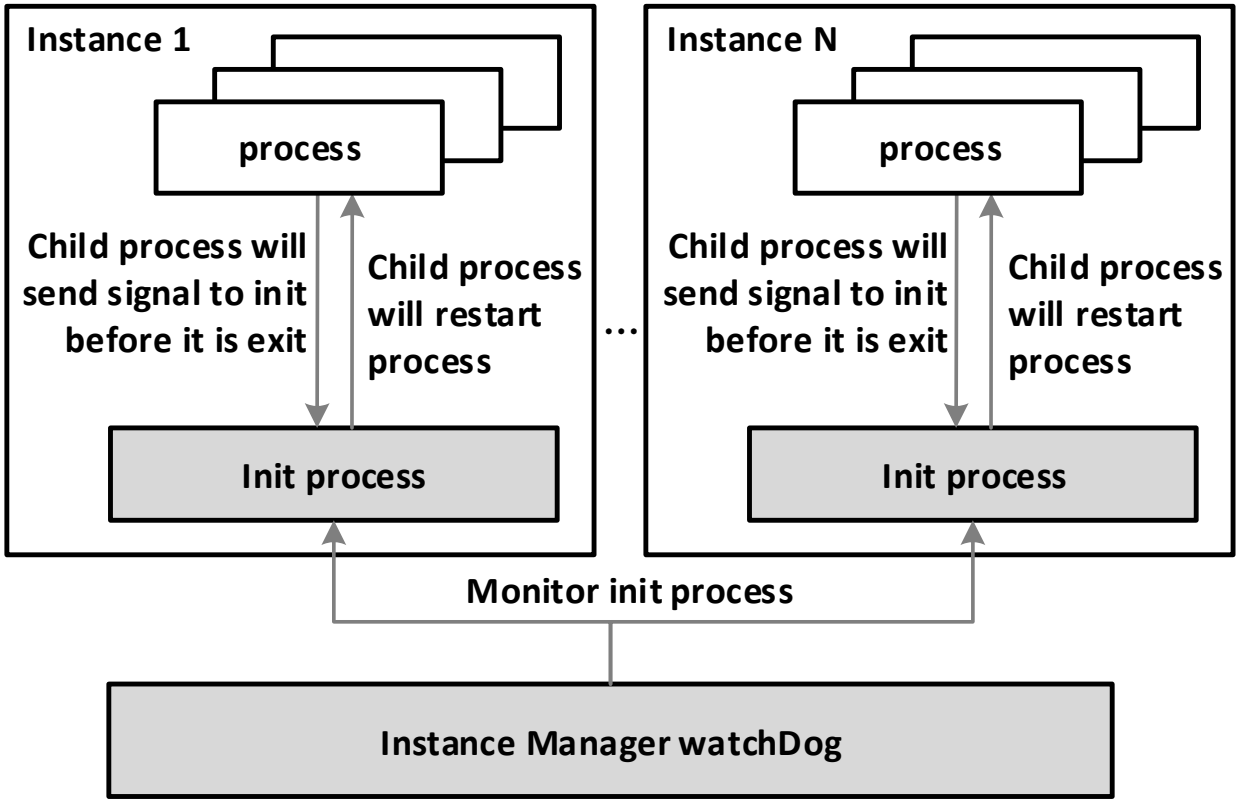
AOSP

- Low memory Killer
- Recovery

Low Memory Killer



Recovery


Instance 1
process
Child process will send signal to init before it is exit
Child process will restart process
Init process
...
Instance N
process
Child process will send signal to init before it is exit
Child process will restart process
Init process
Monitor init process
Instance Manager watchDog

Evaluation

- Environment setup
- Evaluation result

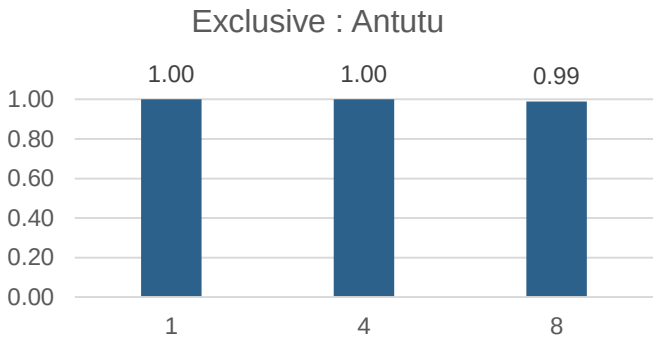
Evaluation Setup

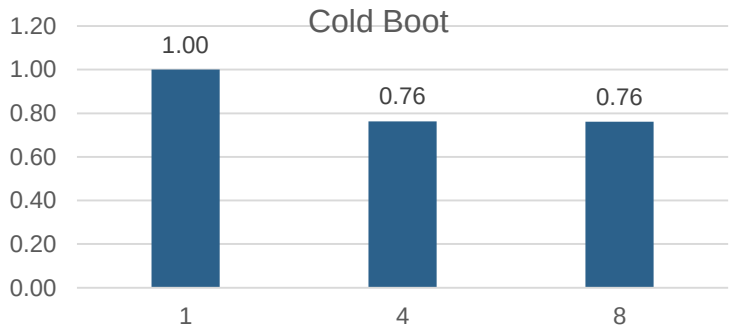
- Exclusive test: single active instance while multiple instances are created but in idle state
- Concurrent test: system performance where multiple instances run with continuous workload simultaneously
- Memory consumption

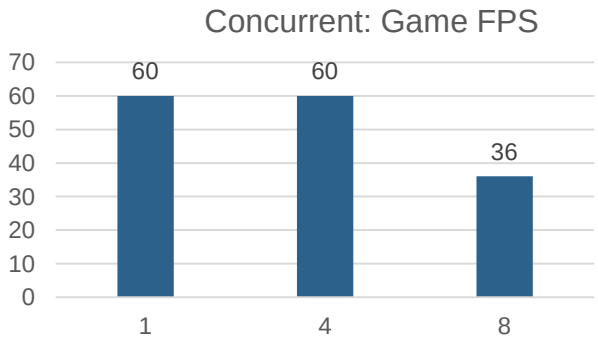
Hardware: Intel mini Kaby Lake PC Intel(R) Core(TM) i5-7260U CPU @ 2.20GHz *2 /HDMI display and touch

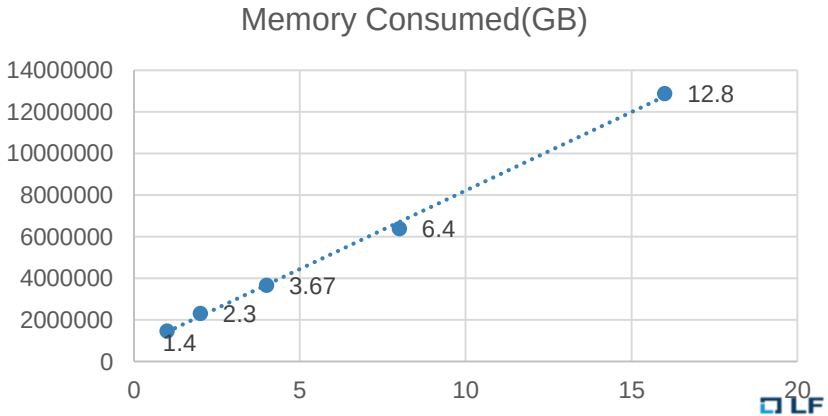
Software: Android 7.1.2

Result


Exclusive : Antutu
1.00
1.00
0.99
1.00
0.80
0.60
0.40
0.20
0.00
1
4
8


Cold Boot
1.00
0.76
0.76
1.20
1.00
0.80
0.60
0.40
0.20
0.00
1
4
8


Concurrent: Game FPS
60
60
36
70
60
50
40
30
20
10
0
1
4
8


Memory Consumed(GB)
12.8
6.4
3.67
2.3
1.4
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
0
5
10
15
20

Summary

- Raised the idea to consolidate workloads using container technology.
- Designed simple I/O sharing solution.
- Implemented POC to consolidate 16 Android Instances in KabyLake Nuc.

Next

- Enhance security.
- Optimize boot time and memory usage.
- Integrate different Linux based OS in one platform.
- Explore the possibility to integrate with VM solution.

LINUXCON
containercon
CLOUDOPEN
CHINA 中国
THINK OPEN
开放性思维
LF ASIA, LLC