移动虚拟化技术与Android安全

- 艾奇伟

目前主要的移动安全威胁

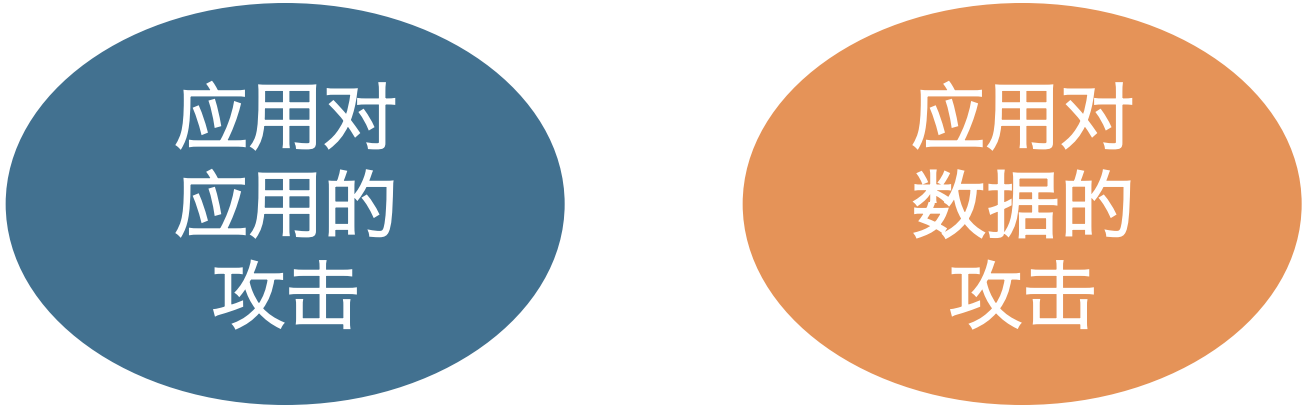
应用安全威胁

- 金融支付威胁
- 企业业务威胁
- 个人应用威胁 - 游戏。。。
- 个人隐私

数据安全威胁

- 个人隐私数据
- 企业业务数据

移动安全的两大隐患



终端侧主要的安全解决方法

虚拟化技术是终端侧实现隔离的基础技术

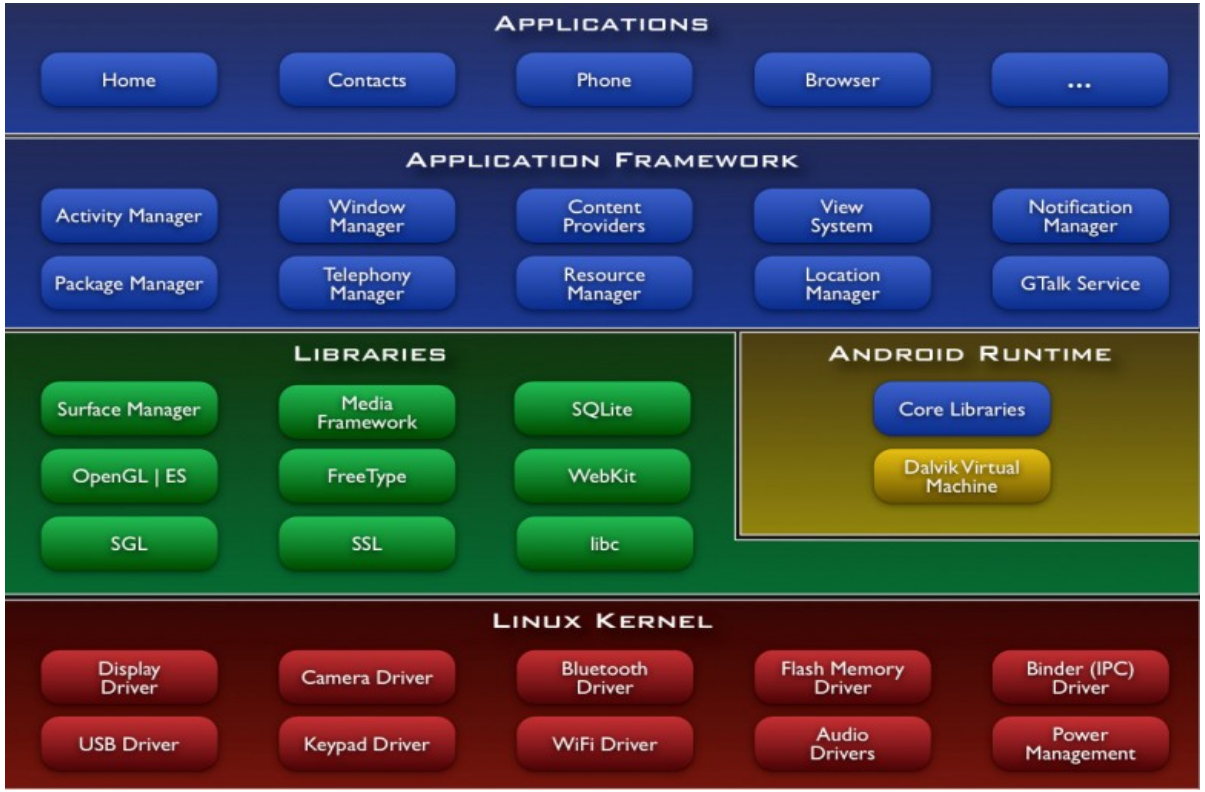
Android 虚拟化类型

这一刀切在哪里

App Wrapper

Framework Virtualization

Hypervisor



Hypervisor

- 技术方案

-基于CPU和内核的准虚拟化(Paravirtualization)
-主要工作集中在Host Kernel, Guest Kernel和HAL

- 优点

-应用、框架、库存Runtime基本不受影响

-性能和体验良好

-VM之间高度隔离

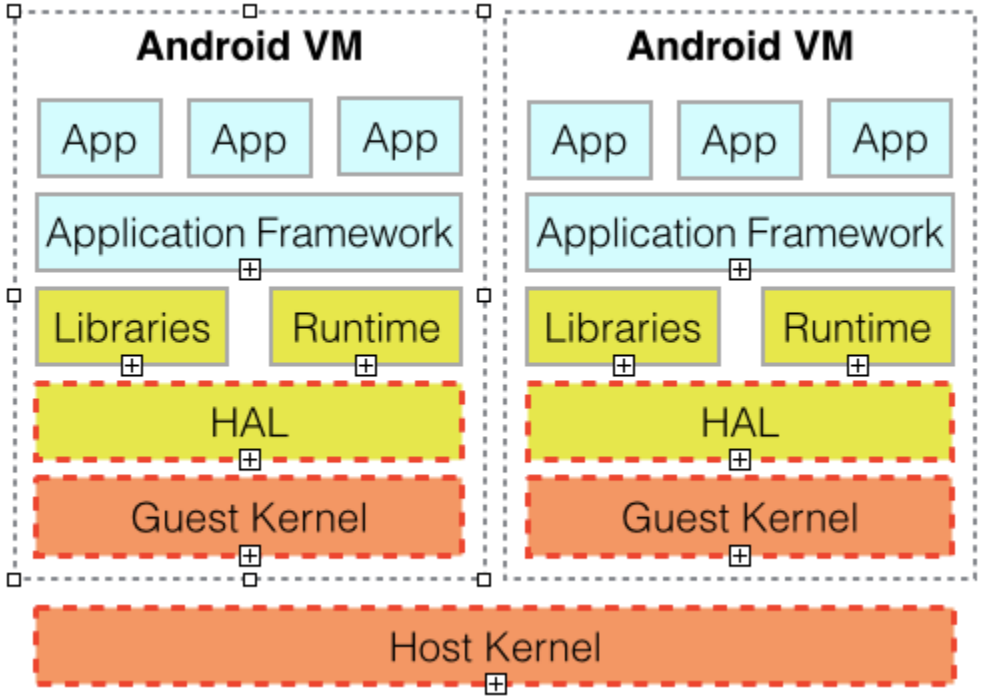
- 缺点

-开发量大

-与硬件高度相关，需针对硬件移植

-VM之间资源共享困难

-对硬件资源要求高



App Wrapper

- 技术方案

-针对Android App的安全封装

-为App提供或替换API，提供互相隔离的服务

- 优点

-对操作系统要求低

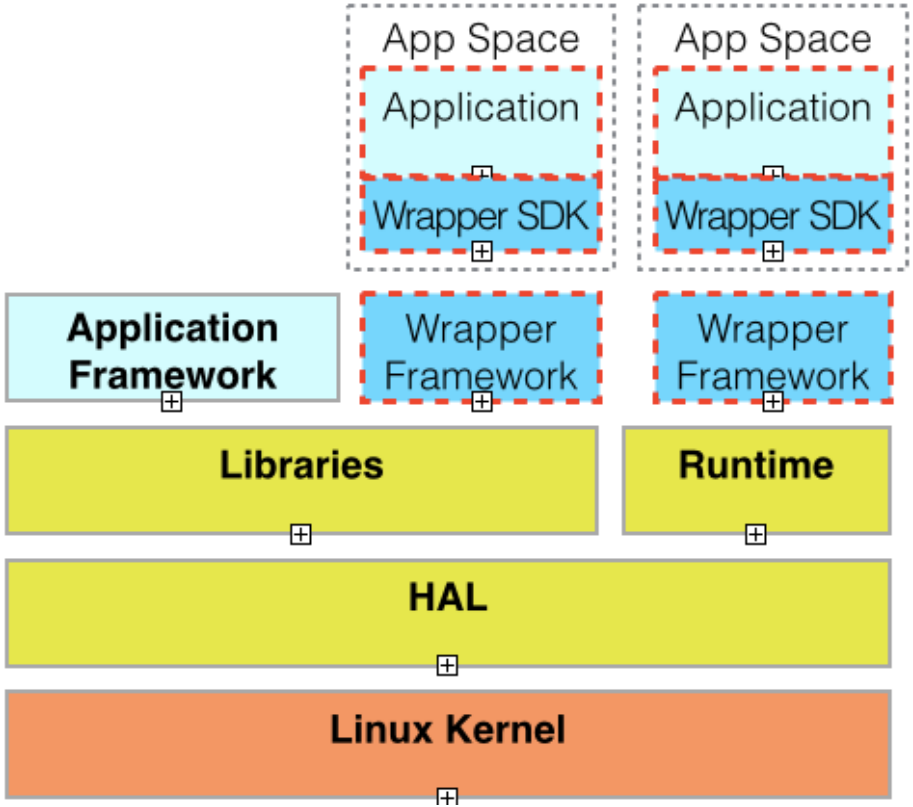
-无特殊硬件要求

-轻量

- 缺点

-安全性差，存在大量共用的服务和资源

-App需改造或重新打包，对App兼容性差，对资源有很多限制



Framework Virtualization

- 技术方案

-利用LXC，namespace等内核技术创造多个互相隔离的进程环境(namespace)
-在隔离的环境分别运行服务和APP
-改造HAL和服务以公用硬件资源和实现必要的namespace间通讯

- 优点

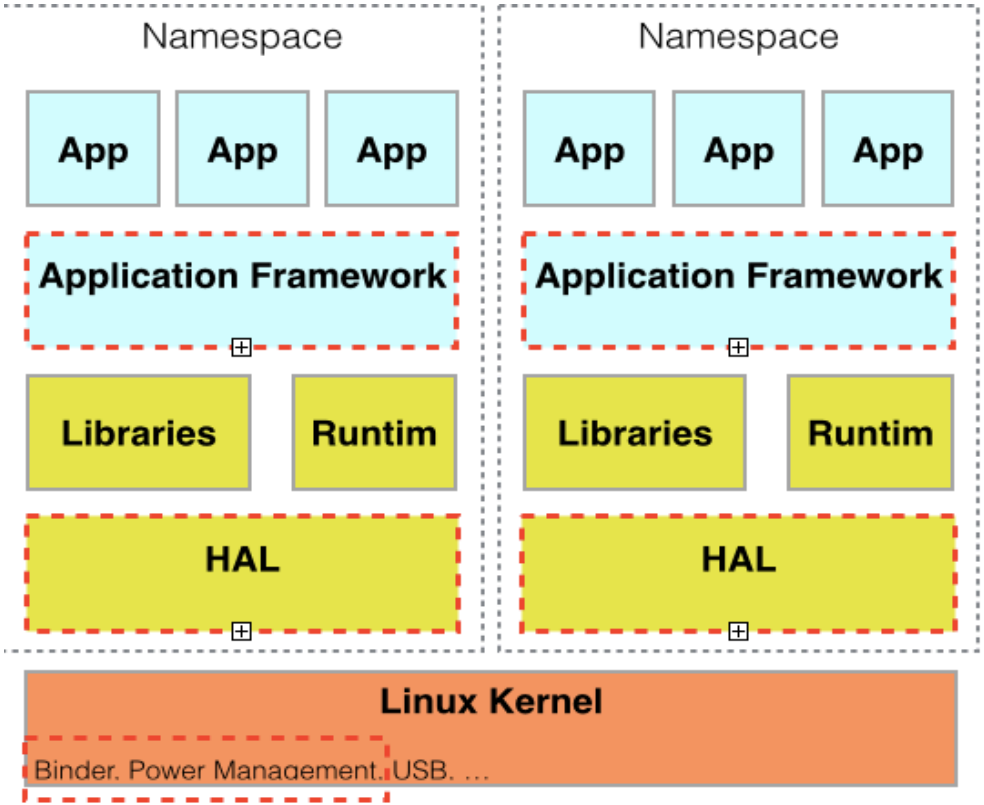
-对APP兼容性高，不需要移植

-应用隔离和资源共用可以调整

-对硬件无特殊要求

- 缺点

-针对不同硬件和Android发行版仍有一些移植的工作量
-HAL的资源共享和调度还有一定的工作需要做



Hypervisor vs Framework Virtualization

- Xen(有硬件辅助),Hypervisor
- kvm(有硬件辅助),Hypervisor
- Lxc(原生linux), Framework Virtualization

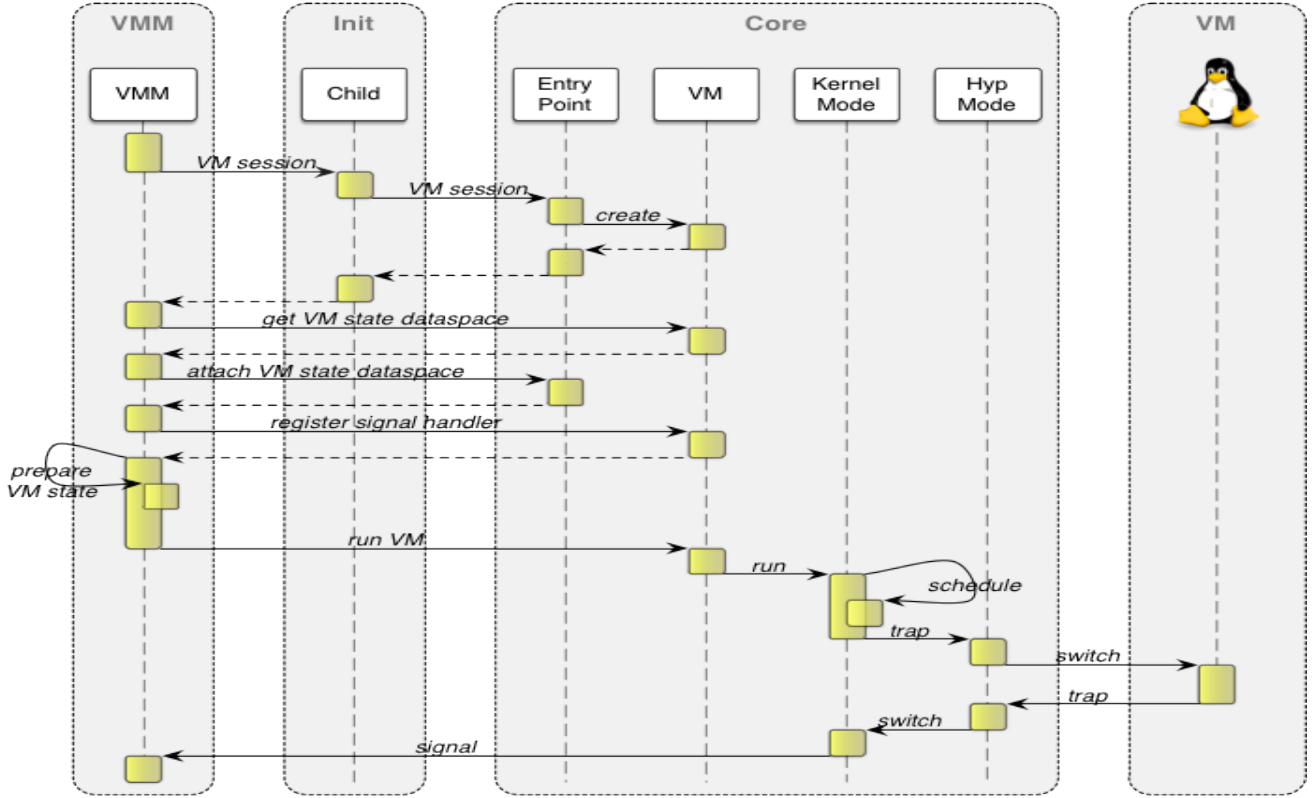
硬件辅助的Hypervisor模式

	ARM	X86
VMM	Y	Y
vIRQ （后期加上）	Y	Y
vMMU（后期加上）	Y	Y
vIOMMU（后期加上）	Y	Y

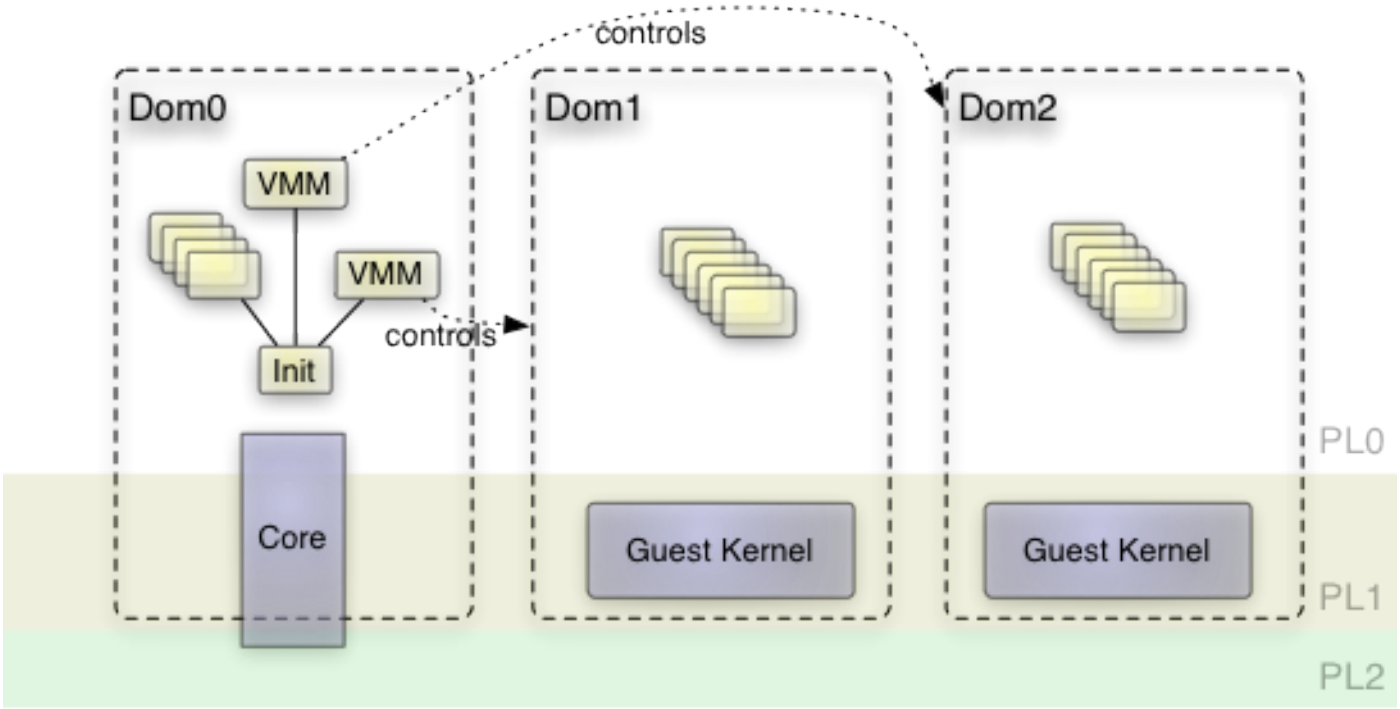
硬件辅助的Hypervisor模式

VMM	VIRTUAL MACHINE MONITOR
VIRO	VIRTUAL INTERRUPT REQUEST
VMMU	VIRTUAL MEMORY MANAGEMENT
VIOMMU	VIRTUAL IO MMU

硬件辅助下Hypervisor的实现原理



硬件辅助下Hypervisor的实现原理



Hypervisor性能测试

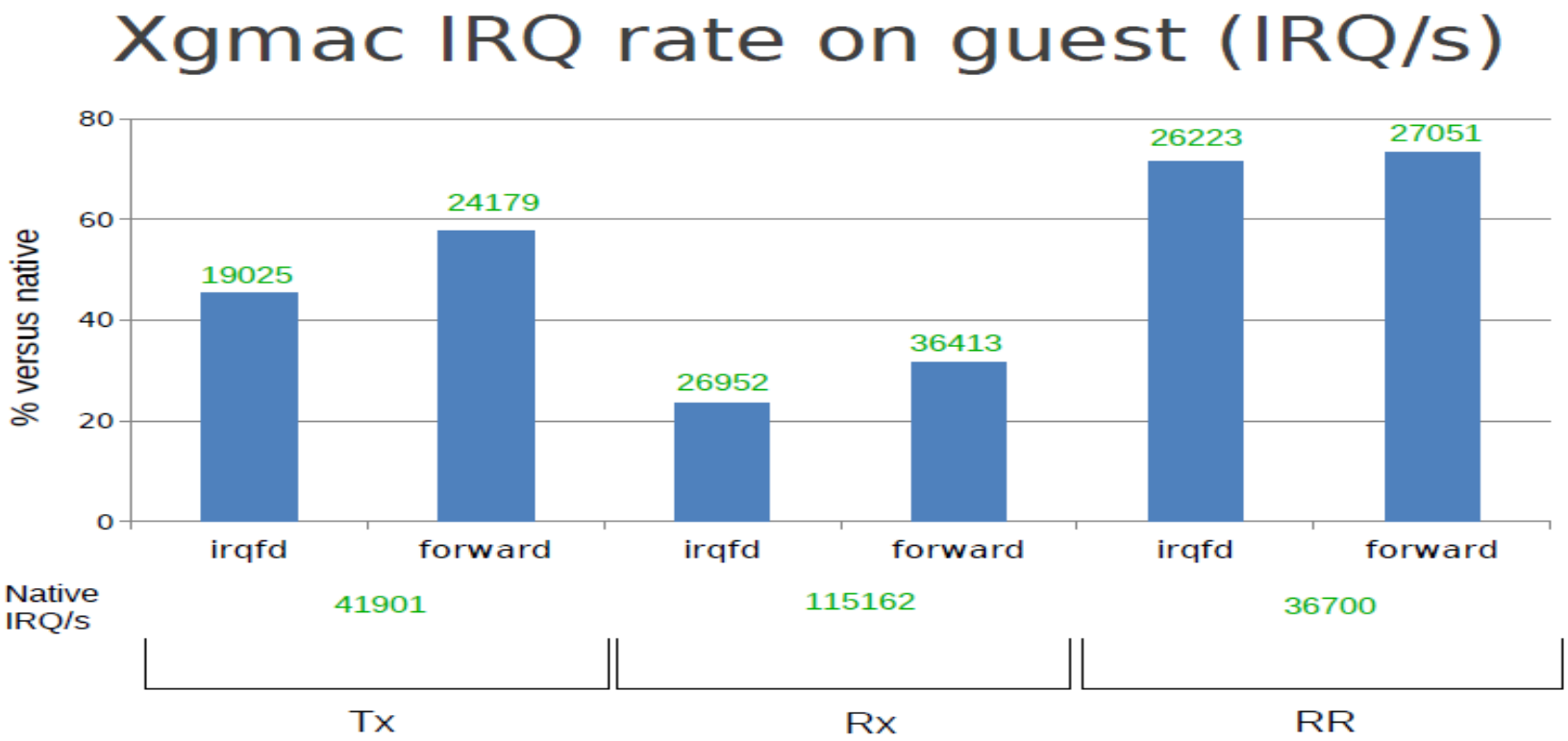
	Bare Metal A	Bare Metal B	Bare Metal Avg	KVM A	KVM B	KVM Avg	KVM to Bare Metal相对裸机	Xen A	Xen B	Xen Avg	Xen to Bare Metal
C-Ray	35.32	35.38	35.35	35.64	35.68	35.66	0.87%	36.13	36.13	36.13	2.16%
POV-Ray	230.05	229.99	230.02	232.74	232.14	232.44	1.04%	236.33	235.45	235.89	2.49%
Smallpt	160	160	160	162	162	162	1.23%	168	167	167.5	4.48%
Blowfish	3028	3024	3026	2993	2990	2991.5	-1.15%	2839	2873	2856	-5.95%
DES	7374000	7375667	7374833.5	7270667	7273000	7271833.5	-1.42%	6858667	6963667	6911167	-6.71%
MD5	49568	49528	49548	48882	48917	48899.5	-1.33%	46428	46879	46653.5	-6.20%
OpenSSL	397.73	397.63	397.68	394.6	393.3	393.95	-0.95%	387.5	389	388.25	-2.43%
7-Zip	12483	12452	12467.5	12196	12063	12129.5	-2.79%	11854	11904	11879	-4.95%
Timed MAFFT Alignment	7.76	7.8	7.78	7.78	7.81	7.795	0.19%	8.5	8.34	8.42	7.60%
CLOMP	3.3	3.3	3.3	3.28	3.29	3.285	-0.46%	3.09	3.16	3.125	-5.60%
PostMark	3658	3676	3667	3791	3857	3824	4.11%	3205	3205	3205	-14.41%
性能降低							1.85%				6.31%

性能表现很好，是吧?

请注意，测试所用是功能软件（如加密，解密等。

请看下边

一个虚拟网卡的IRQ性能测试



分析

在所有硬件辅助都已经使用的情况下，
网络中断响应能力已经下降50%
核心是: I/O
若存在大量的设备I/O性能会更加劣化

移动虚拟化技术

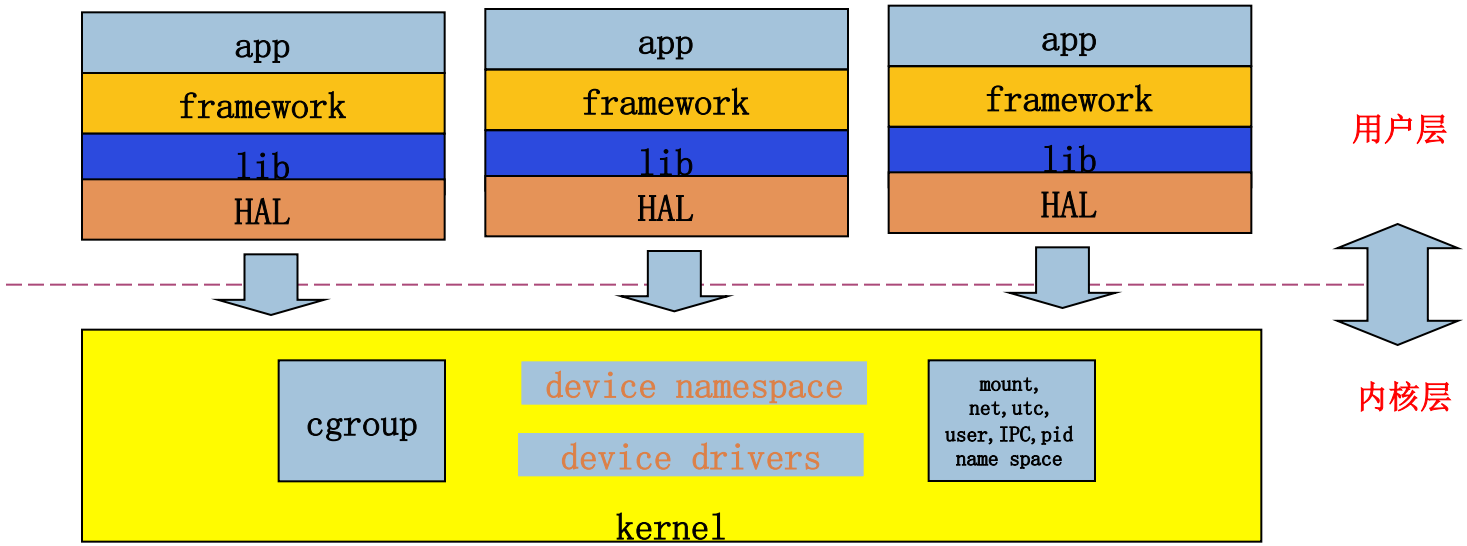
交互是移动设备的核心,

所以, 存在大量的 I/O.

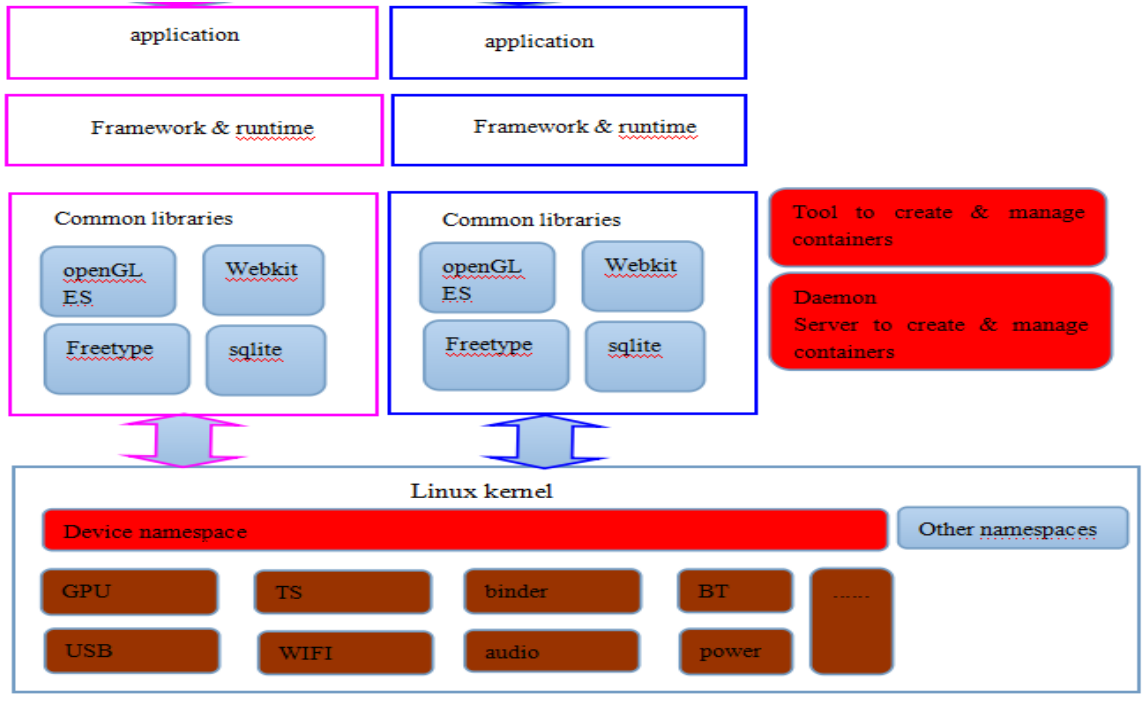
所以我们现在选择 lxc,

不是kvm, 不是xen

结构示意


app
framework
lib
HAL
app
framework
lib
HAL
app
framework
lib
HAL
用户层
内核层
cgroup
device namespace
device drivers
mount,
net, utc,
user, IPC, pid
name space
kernel

- 增加device namespace支持
- 基于device namespace更改设备驱动
- 使用mount namespce
- 使用pid namespace
- 使用所有linux kernel中存在的namespace
- 使用cgroup控制资源
- 而后, 我们有了containers


application
application
Framework & runtime
Framework & runtime
Common libraries
openGL ES
Webkit
Freetype
sqlite
Common libraries
openGL ES
Webkit
Freetype
sqlite
Tool to create & manage containers
Daemon Server to create & manage containers
Linux kernel
Device namespace
Other namespaces
GPU
TS
binder
BT
.....
USB
WIFI
audio
power
To create
To modify
For phone1
For phone2

我们完成了

- containers控制中心,
  create, start, stop, info, destroy, etc.
- containers交互代理.
- 新的device namespace
- 修改所有相关
  linux kernel device drivers

所以，合二为一

方案说明

- 新平台移植只需修改linux kernel平台相关device drive
- 原平台各版本的android原则上不需修改
- 同时启动不同版本的android需额外存储空间约330M
- 同时启动相同版本的android不需额外存储空间
- 每多启动一个android新增约170M内存
- 每多启动一个android性能损失约%0.1(粗测)

测试


9:18
Details - v5.7
17919
AOSP on Grouper
UX
Multitask: 3385
Runtime: 1157
CPU
CPU integer: 1725
CPU float-point: 1771
CPU
Single-thread integer: 1263
Single-thread float-point: 1079
RAM
RAM Operation: 1584
RAM Speed: 667
GPU
2D graphics: 700
3D graphics: [1280x800] 3510
Test Again
Performance
nexus 7 1st gen
1 android running
9:31
Details - v5.7
17905
AOSP on Grouper
UX
Multitask: 3376
Runtime: 1146
CPU
CPU integer: 1720
CPU float-point: 1783
CPU
Single-thread integer: 1267
Single-thread float-point: 1078
RAM
RAM Operation: 1590
RAM Speed: 696
GPU
2D graphics: 677
3D graphics: [1280x800] 3503
IO
Storage I/O: 459
Database I/O: 610
nexus 7 1st gen
2 androids running
Test Again
Performance

我们正在做

向更多的手机移植我们的方案

新的FOTA机制

新的DDMS支持samsung s4, nexsus 5, 定制 手机

我们也许会做

- ARM/KVM, 加入VFIO， 以及所有设备virtIO化，到这个时候

I/O 将不再成为瓶颈

安全吗?

Hypervisor：虚拟机逃逸

Framework Virtualization: 内核漏洞

参考

- http://www.linux-kvm.org/page/KVM_Forum_2014
- http://systems.cs.columbia.edu/files/wpid-asplos2014-kvm.pdf
- https://major.io/2014/06/22/performance-benchmarks-kvm-vs-xen/
- https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQGOKrjfmh1sHTnTMB-H-2Y4Ry77kxfuvMlmk1MKMccAsaKz0hN

谢谢！

Kiwi.e4gle

北京 朝阳

扫一扫上面的二维码图案，加我微信