# 安卓手机优化，修改build.prop!让手机飞起来把

最近有好多人找小Z优化手机，用了很多办法。只有root和修改build.prop文件屡试不爽。有网友要问了，root?我的国产小厂机，不行啊。那么我给大家推荐一款咱么的神器——eroot。相信你就可以轻易获得权限了。

好了，不废话了，咱们转入正题，在手机系统根目录有个叫build.prop的文件，大家root以后可以选择用ex文件浏览器修改，好处是改完不用修改权限，也可以把它导出来用电脑修改。打开build.prop文件以后开有没有以下值，如果没有，将文本拉到最下面另起一行添加就可以了。添加完记得保存重启(⊙o⊙)哦。

另：玩机有风险，改机要小心

1.更好的录像.照相优化

ro.media.capture.maxres=8m

ro.media.capture.fast.fps=4

ro.media.capture.slow.fps=120

ro.media.panorama.defres=3264x1840

ro.media.panorama.frameres=1280x720

ro.camcorder.videoModes=true

2.wifi速度优化

net.ipv4.tcp\_ecn=0

net.ipv4.route.flush=1

net.ipv4.tcp\_rfc1337=1

net.ipv4.ip\_no\_pmtu\_disc=0

net.ipv4.tcp\_sack=1

net.ipv4.tcp\_fack=1

net.ipv4.tcp\_window\_scaling=1

net.ipv4.tcp\_times\*\*ps=1

net.ipv4.tcp\_rmem=4096 39000 187000

net.ipv4.tcp\_wmem=4096 39000 187000

net.ipv4.tcp\_mem=187000 187000 187000

net.ipv4.tcp\_no\_metrics\_save=1

net.ipv4.tcp\_moderate\_rcvbuf=1

3.提高jpg质量100%

ro.media.enc.jpeg.quality=100

4.渲染GPU的UI

debug.sf.hw=1

5.拨号时间减少

ro.telephony.call\_ring.delay=0

6.提高滚动反映

windowsmgr.max\_events\_per\_sec=150

5.电池优化

wifi.supplicant\_scan\_interval=180

pm.sleep\_mode=1

ro.ril.disable.power.collapse=0

6.提高整体触摸反应

debug.performance.tuning=1

video.accelerate.hw=1

7.网络调整

ro.ril.hsxpa=2

ro.ril.gprsclass=12

ro.ril.hep=1

ro.ril.hsdpa.category=10

ro.ril.enable.3g.prefix=1

ro.ril.htcmaskw1.bitmask=4294967295

ro.ril.htcmaskw1=14449

ro.ril.hsupa.category=6

ro.ril.def.agps.mode=2

ro.ril.def.agps.feature=1

ro.ril.enable.sdr=1

ro.ril.enable.gea3=1

ro.ril.enable.fd.plmn.prefix=23402,23410,23411

ro.ril.enable.a52=1

ro.ril.enable.a53=1

ro.ril.enable.dtm=1

8.网络速度调整

net.tcp.buffersize.default=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.wifi=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.umts=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.gprs=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.edge=4096,87380,256960,4096,16384,256960

9.修复通话结束后黑屏问题

ro.lge.proximity.delay=25

mot.proximity.delay=25

10.修复一些应用问题

ro.kernel.android.checkjni=0

11.禁用开机动画

debug.sf.nobootanimation=1

12.使用Stagefright让视频和音乐更快

media.stagefright.enable-player=true

media.stagefright.enable-meta=true

media.stagefright.enable-scan=true

media.stagefright.enable-http=true

上面的代码在system里build。prop最后另外起一行。

   上面wifi优化有两个\*\*是t  a   m     （无空格）

转载自“魔趣 ViPER520”，高手，我525现在用的还是他的包，稳定，流畅，省电。

再次说明一下，本教程中说的参数基本是Android 2.3系统通用的，不论你是CM7还是官方，或是Miui等等，都是通用的。除非教程中有说明只能用于官方ROM或CM7 ROM的。另外，其他机型也可以使用，不仅限于Moto，除非是教程中提到为Moto专用参数。

如果你发现你的Build.prop中没有教程里讲的参数，那么就手动添加到最后面好了，没必要纠结于“我的Build.prop和教程里的完全不一样”。

发现一些机油兄弟对本教程所讲内容不太明白或不理解该如何使用，故稍微介绍一下。本文所讲均是手机/system/build.prop文件中的内容，可以按照教程在电脑上使用文本编辑软件编辑build.prop文件，向其中添加、修改和删除本文所讲的字段条目。另外，官方或者CM ROM中的build.prop文件内可能不包含本文提到的一些字段条目，若各位看官需要，可以自行将其添加到build.prop文件最末行，即可生效。本文绝对原创，如转载请注明“魔趣 ViPER520”，谢谢！

在Android系统中有一个类似Windows系统注册表的文件build.prop。这个文件内定义了系统初始(或永久)的一些参数属性、功能的开放等。通过调整/增加参数可以达到较调系统性能偏重点和附加功能开启的作用。在Android 2.2、2.3、4.0中虽然每一版都有自己独有的参数，但绝大部分都是通用的，且可以起到关键性作用的。本文将以摩托手机Android 2.3系统为例，对Build.prop中常用的参数进行详解，供广大机友对自己的系统做较调。

本教程将分为5期，第一期是Dalvik虚拟机相关参数，第二期是系统版本、定义相关参数，第三期是基本性能相关参数，第四期是基本耗电相关参数，第五期是扩展性能较调及附加功能开启。

第一期：Dalvik虚拟机相关的参数属性。Dalvik虚拟机是Android操作系统的核心，是一切应用程序的基础。所有程序在运行时均有Dalvik虚拟机对其进行解析和执行。

dalvik.vm.startheapsize，本参数控制Dalvik虚拟机在启动一个应用程序之后为其分配的初始堆栈大小，可填写的值为2m~48m。例如：dalvik.vm.startheapsize=8m，就表示应用程序启动后为其分配的初始堆栈大小为8兆字节。这里分配的内存容量会影响到整个系统对RAM的使用程度，和第一次使用应用程序时的流畅程序。这个值越大，系统消耗RAM则越快，但是应用程序打开后的反应也越快。值越小，系统的RAM剩余则越多，但是程序在启动后会很卡。建议值是8m，既可以保持140M左右的RAM，程序的反应速度也会大幅度提高。

dalvik.vm.heapsize，本参数控制Dalvik虚拟机给一个应用程序分配的最大堆栈量，可填写的值为12m~48m。例如：dalvik.vm.heapsize=48m，就表示应用程序在任意时刻内可以使用的最大堆栈大小为48兆字节。这里分配的内存容量会影响到整个系统对RAM的使用程序，和程序在运行一段时间后的反应速度。这个值越大，系统消耗RAM则越快，但是程序会运行的非常稳定，尤其是游戏和视频程序的内容加载速度可以大幅度提升。值越小，系统的RAM剩余则越多，但是程序会很卡，尤其是游戏在切换场景Loading的时候会花费很多的时间。若应用程序需要使用超过这个值的内存时，将会触发系统的垃圾收集器，系统和程序就会卡顿。建议值是40~40m。

dalvik.vm.lockprof.threshold，本参数控制Dalvik虚拟机调试记录程序内部锁资源争夺的阈值，默认值是500。多用于程序的数据统计，对性能较调意义不大。

dalvik.vm.stack-trace-file，本参数控制Dalvik虚拟机的堆栈记录调试文件。用于系统调试，一般用户对其调整无意义。

dalvik.vm.execution-mode，本参数控制Dalvik虚拟机的程序执行机制。可填写的值有"int:portable"、"int:fast"和"int:jit"。int:portable表示以兼容模式运行(脚本翻译模式)，此模式下程序的兼容性最高，但其执行效率最低(程序优化度依赖于dalvik虚拟机版本)。官方默认此模式。int:fast表示以快速自优化模式运行(脚本翻译和预优化混合)，此模式下程序的兼容性很高，执行效率也比较高。因为此时dalvik虚拟机允许程序使用自己的预定义优化模式和代码(包括C/C  /汇编代码)。推荐使用。int:jit表示以Just-In-Time模式运行(JIT模式)，此模式下程序的兼容性最差，但程序一旦加载后其运行效率最高(与C/C  直接编写的程序效率无异)，因为在此模式下dalvik虚拟机会预先将Java程序翻译成针对机器平台的本地语言(Native)，同时完全允许代码中的所有预优化和代码，允许所有不安全的非托管代码，同时不严谨的程序如果运行在JIT模式可能会造成内存泄露。但要注意，很多Dalvik虚拟机并不支持此模式(如官方2.2)。

dalvik.vm.dexopt-flags，本参数控制Dalvik虚拟机的程序代码校验和优化。可填写的值有m、v和o。m为标准选项，可以是m=y或m=n。若m=y则启用不安全代码的校验和托管代码的优化。兼容性和安全性最高，推荐使用。v为校验选项，可与o并存。可以是v=a或v=n。若v=a则表示校验所有代码，v=n则关闭代码的校验。o为优化选项，可与v并存。可以是o=v或o=a。若o=v则表示优化以校验过的代码，o=a则表示优化所有代码。例如：dalvik.vm.dexopt-flags=m=ydalvik.vm.dexopt-flags=v=n,o=v注意，这个参数只会影响到安装APK之后或初次使用APK时生成dex文件时有效。若整个系统(包括应用程序)为odex化，则无意义。

dalvik.vm.verify-bytecode，本参数控制Dalvik虚拟机是否验证应用程序的可执行代码。可以与上一个参数配合使用。可填写的值为true和false。其具体意义与dalvik.vm.dexopt-flags的v=n一模一样。但可以与dalvik.vm.dexopt-flags配合使用以取得更好的效果。例如：dalvik.vm.dexopt-flags=v=n,o=vdalvik.vm.verify-bytecode=false这样可以令后来安装的apk文件可以被优化而不被检验。dalvik.vm.checkjni，本参数控制Dalvik虚拟机在调用外部jni链接库的时候是否对其做安全性检验。可填写的值为true和false。此参数会覆盖ro.kernel.android.checkjni。若值为true，会增加程序的兼容性和稳定性，但也会增加其加载和执行的时间。推荐为false。

dalvik.vm.deadlock-predict，本参数控制Dalvik虚拟机对程序死锁预测处理。可填写的值有off、warn和err。off表示关闭死锁预测功能(默认设置)。warn表示在继续程序运行的同时只记录该死锁预测(如果为真死锁就会出现程序假死现象，然后等N久出现关闭)。err表示预测到死锁时马上弹出FC。注意：有些Dalvik虚拟机版本并不支持此参数。总结：对于本期此处给出

三种常用的配置(以Defy为机型)。

超级急速流畅型：dalvik.vm.startheapsize=16m

dalvik.vm.heapsize=48m

dalvik.vm.execution-mode=int:jit

dalvik.vm.dexopt-flags=v=n,o=v

dalvik.vm.verify-bytecode=false

dalvik.vm.checkjni=false

常用稳定加流畅型：dalvik.vm.startheapsize=8m

dalvik.vm.heapsize=40m

dalvik.vm.execution-mode=int:fast

dalvik.vm.dexopt-flags=m=y

dalvik.vm.verify-bytecode=false

dalvik.vm.checkjni=false

超级稳定大内存型：dalvik.vm.startheapsize=4m

dalvik.vm.heapsize=30m

dalvik.vm.execution-mode=int:portable

dalvik.vm.dexopt-flags=v=a,o=v

dalvik.vm.verify-bytecode=true

dalvik.vm.checkjni=true

第二期：系统版本、定义等参数。本期将介绍系统版本、定义等相关参数。主要用于定义系统版本特征字串，OTA字串等。由于较少用到，因此只粗略介绍。

ro.build.id，本参数定义了系统的版本ID。为系统内部使用，OTA时作为粗略版本比较。更改后可避免OTA提示，但可能会引起预装程序(如Blur)的稳定性。

ro.build.display.id，本参数定义了设置中显示的系统版本号。主要用于设置中显式出现可读版本，一般用于个性化定制和第三方应用程序对系统版本的判断(如魔趣设置)。更改后可自定义版本显示，但某些第三方应用程序会出现错误(如魔趣设置无法实现机器保修查询)。

ro.build.version.incremental，本参数定义了系统的升级字。主要用于系统OTA精确版本比对，同时与ro.build.description和ro.build.fingerprint相匹配。更改后可以免OTA提示(如避免Miui的升级提示和Blur的升级提示)。

ro.product.model，本参数定义了机器的型号字符串。主要用于机器型号显式定义(如系统设置中的手机型号和Blur、Google设置向导中的机型等)。更改后可自定义手机型号名称。

ro.product.locale.language，本参数定义了系统的初始(默认)语言。此处注意是语言，如中文是zh，英文是en。更改后改变系统初次启动时的语言设置。

ro.product.locale.region，本参数定义了系统的初始(默认)区域。此处注意是区域，如中国大陆为CN，台湾为TW，美国为US。更改后改变系统初次启动时的区域设置。

ro.build.description和ro.build.fingerprint均为ROM的编译综合说明。其中包含了平台硬件、Android版本、源代码分支和标签、OTA详细版本等。其中的OTA部分，例如：umts\_jordan\_china-user 2.3.6 4.5.3-109\_DPP-141323416413release-keys将此数字与ro.build.version.incremental一同更改可避免OTA升级提醒(如Miui和Blur等)。

第三期：基本性能相关参数。本期将介绍与系统性能(流畅操作体验、功能速度、内存管理等)相关的参数属性和其调整方法。虽然Defy的CPU只有800MHz，虽然Defy的RAM只有512MB，虽然摩托官方的系统优化很差，但通过本期的参数调整，依然可以获得不俗的性能。

windowsmgr.max\_events\_per\_sec，本参数定义了Android系统的窗体事件管理器在单位时间内可以处理的最大事件数量。通过更改本参数可以获得非常明显的丝滑流畅体验。可填写的值范围为“大于0的正整数”，官方默认为60。建议150、200、260、300这几个值。当此值变大时，系统触控平滑度明显提高，但对应的CPU使用率也会升高，最终的结果就是电池续航能力下降。以我个人的经验来说，此值取到240左右时在系统设置中滑动可以得到接近WP7的流畅和平滑度。

ro.min\_pointer\_dur，本参数定义了两次触摸之间的最短时间间隔，单位是毫秒。默认值为25，推荐值是10。通过调整此参数可以提高系统触控的灵敏度或稳定度。当此值越大时，触控越稳定。此值越小，触控越灵敏。

mot.proximity.delay，本参数定义了手机光纤感应器的抖动消除时间，单位是毫秒。默认值是500，推荐值是250。通过调整此参数可以提高在通话结束后屏幕点亮的速度。当此值越大时，通话结束后屏幕点亮所需要的时间越长，但在通话过程中如果手机意外瞬间离开脸部也不会点亮屏幕，可防止通话过程中的误操作(比方说通话时不小心手机移动了一下，屏幕就会点亮，此时如果脸部触碰到了屏幕就会对通话造成影响)。此值越小，则当手机离开脸部或装入口袋后会立即点亮或关闭屏幕。

mot.proximity.distance，本参数定义了手机屏幕上的两个触摸点之间的最短距离，若距离小于此值则认为是一个触摸点，单位是像素。默认值是60，推荐值是100。为什么推荐100呢？因为Defy的屏幕分辨率为480x854，也就是说横向有480个像素点，对应上去也就相当于是横向并排允许4个触摸点，平均一个手指一个点，这样在类似于杀西瓜等游戏中可以提升游戏操作。

ro.kernel.android.checkjni，本参数定义了Dalvik虚拟机在执行程序的时候是否要做Jni链接库的检查工作。详细见Dalvik参数属性期。若考虑稳定性可使用true，若需要性能可使用false。注意：此参数会被Dalvik参数覆盖。

ro.media.enc.jpeg.quality，本参数定义了JPEG图像编码器所使用的质量因子，可填写的值为1~100，默认为80，推荐为100。想照出更好的照片吗？想让照片的大小轻松上M吗？那就使用100吧。

debug.sf.hw，本参数定义了系统是否启用GPU来渲染程序的UI，默认为0，推荐为1。但要注意，如果此值为1，在某些应用程序中可能会出现显示错乱的现象(极少见)。

persist.sys.use\_dithering，本参数定义了系统渲染器对图像的缩放是否启用抖动技术。可填写的值为0或1。当开启抖动后，图像的显示(指背景、解锁等的图像，并非图库、相机那些的)会很柔和，但会增加CPU负载，最终导致ROM卡顿。

persist.sys.purgeable\_assets，本参数定义了系统是否可以清除暂时不用的数据以释放更多的RAM。可填写的值为0或1。当值为1时，系统会定期清理不用的数据以释放更多的RAM，同时作为代价就是下次启动程序或游戏加载数据会变慢。

video.accelerate.hw，本参数定义了系统是否对视频启用硬件加速功能。这里的视频指代屏幕上显示的东西，不仅仅是“电影视频”。可填写的值为0或1。需要注意的是：摩托官方的2.2和2.3系统对此功能支持的不是很好，开启后有时反而会降低系统流畅度。但CM系统绝对建议开启。

debug.performance.tuning，本参数定义了系统是否针对性能做较调。可填写的值为0或1。需要注意的是：摩托官方的2.2和2.3系统对此功能支持的不是很好，开启后有时反而会降低系统流畅度。但CM系统绝对建议开启。\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ro.HOME\_APP\_ADJro.FOREGROUND\_APP\_ADJro.VISIBLE\_APP\_ADJro.PERCEPTIBLE\_APP\_ADJro.HEAVY\_WEIGHT\_APP\_ADJro.SECONDARY\_SERVER\_ADJro.BACKUP\_APP\_ADJro.HIDDEN\_APP\_MIN\_ADJro.EMPTY\_APP\_ADJ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以上参数定义了各种应用程序的管理机制，这些并非一两句话可以说清楚的，想深究的同学可以Google一下OOM Killer。可填写的值为整数。这里只给出值的规律，0代表降低进程的优先级且驻留内存，1代表驻留内存，4代表缓存较多的内存，15代表尽量缓存内存。也就是说内存缓存器是按照ADJ从大到小来进行缓存的。大家可根据自系统中自己对各种应用程序的要求进行更改。以下给出一个经典用例：ro.FOREGROUND\_APP\_ADJ=0       前台程序驻留内存(不缓存)ro.VISIBLE\_APP\_ADJ=1        可见的程序驻留内存(不缓存)ro.PERCEPTIBLE\_APP\_ADJ=2        缓存的RAM多一些ro.HOME\_APP\_ADJ=3        桌面程序，缓存的RAM稍多一些ro.HEAVY\_WEIGHT\_APP\_ADJ=4        缓存的RAM再多一些ro.SECONDARY\_SERVER\_ADJ=5        缓存的RAM再再多一些ro.BACKUP\_APP\_ADJ=6       缓存的RAM再再再多一些ro.HIDDEN\_APP\_MIN\_ADJ=7        隐藏的程序，根据程序的类型进行内存管理，最低为缓存的RAM再再再再多一些，最高就是直接缓存内存。ro.EMPTY\_APP\_ADJ=15        已经退出的程序，直接缓存内存\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ro.FOREGROUND\_APP\_MEMro.VISIBLE\_APP\_MEMro.PERCEPTIBLE\_APP\_MEMro.HEAVY\_WEIGHT\_APP\_MEMro.SECONDARY\_SERVER\_MEMro.BACKUP\_APP\_MEMro.HOME\_APP\_MEMro.HIDDEN\_APP\_MEMro.CONTENT\_PROVIDER\_MEMro.EMPTY\_APP\_MEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以上参数定义了各种类型的应用程序在内存缓冲的大小，单位是页，应用上面ADJ参数相对应。下面给出一个经典用例：ro.FOREGROUND\_APP\_MEM=1280ro.VISIBLE\_APP\_MEM=2560ro.PERCEPTIBLE\_APP\_MEM=3840ro.HEAVY\_WEIGHT\_APP\_MEM=6400ro.SECONDARY\_SERVER\_MEM=7680ro.BACKUP\_APP\_MEM=8960ro.HOME\_APP\_MEM=5120ro.HIDDEN\_APP\_MEM=12800ro.CONTENT\_PROVIDER\_MEM=15360ro.EMPTY\_APP\_MEM=20480

第四期：基本耗电相关参数本期将介绍与ROM耗电相关的参数和属性。通过调节各种参数可以达到节电等的目的。

wifi.supplicant\_scan\_interval，本参数定义了Wifi扫描已保存节电的时间间隔。当点亮屏幕或打开Wifi时，系统会不停的扫描环境中是否存在已经保存的Wifi节点，当发现后则进行连接，而这个参数控制了每次扫描的时间间隔。单位是秒。取值范围是正整数。官方默认为45，推荐180。

ro.mot.battmanager.wifictrl，本参数定义了电源管理模块对Wifi的控制。默认为0。当此值为1时可以明显节电，但有时Wifi会出现不稳定的情况(不是所有ROM都如此)。

ro.mot.deep.sleep.supported，本参数定义了是否开启摩托的“休眠”模式。取值为true或false。当值为true时，在电源菜单中会出现“休眠”模式。此模式类似于电脑的睡眠，即将CPU等部件的电源全部关闭，只为RAM供电以保存休眠前的系统状态。耗电量比完全关机多一些，但可以做到瞬间开机。仅在官方ROM有效。

pm.sleep\_mode，本参数定义了系统待机时的睡眠深度，在所有Android系统上有效。取值范围是0~4，对应解释如下。0：强制关闭除RAM之外的所有部件，此状态下最省电。Defy几乎可以纯待机3~4个礼拜。但是此模式与“休眠”类似，一旦进入之后射频也会关闭，手机的2G/3G信号也就断了(语音和数据)。

1：让ARM进入中断触发的待机(超低功耗)模式。与模式0相比，本模式下射频不会关闭，而ARM可以通过软件(闹铃)和硬件(来电)中断来唤醒，因此耗电方面远大于模式0，Defy可以纯待机7天(不安装任何软件)。非常建议使用。

2：将所有应用程序挂起到后台。与模式1相比，本模式下硬件几乎不参与多少节电，耗电自然比模式1多很多。当应用程序被挂起后，CPU的负载会大幅度降低，从而节电。此模式下Defy纯待机5天。

3：将CPU的频率和电压降至最低，低到主频只有几十MHz的水平，而此时CPU接受外部中断(通过中断来恢复频率和电压)。与模式2相比，本模式下CPU通过降频和降压参与了节电，因此本模式的耗电比模式2多了一点。Defy纯待机约4~5天。本模式也是官方ROM和官方CM系统的默认值。

4：CPU接受外部中断。与上述4个模式相比，此模式下几乎不做任何节电，只是关闭了屏幕和按键背光而已。Defy纯待机约2天。将上述5个模式的节电按照星级来分就是，模式0和1为5颗星，模式2和3为3颗星，模式4为1颗星。综上所属就是，模式0和模式1基本一样，是靠完全关闭几乎所有硬件部件来进行节电，省电效果最佳。模式2和模式3是靠调节CPU频率来进行节电。个人强烈推荐采用pm.sleep\_mode=1，即省电又稳定。如果想用模式0但又担心基带射频的同学可以继续往下看，解决办法在下面。

ro.ril.disable.power.collapse，本参数定义了是否禁止射频参与电源休眠。取值是0或1。这个参数的使用需要与上一个参数相匹配(我看到很多ROM中的这两个参数都是不匹配的，最终造成的效果就是点亮屏幕后信号存在问题)。当本参数为1的时候即射频永远打开，为0的时候根据上一个参数pm.sleep\_mode来判断是否关闭射频。永远打开射频必然费电，但是如果射频关闭，那手机就没信号了。那么当pm.sleep\_mode=0的时候，上面说过，此时待机会关闭几乎所有硬件部件，包括射频。而此时如果ro.ril.disable.power.collapse=1，就会保持射频的开启(即使进入休眠模式也一样)。这样即使待机，手机也有信号。但是又存在这样一个现象，在有些ROM中pm.sleep\_mode=0会带来更多的问题， 如睡死、亮屏后Wifi打不开、蓝牙打不开等。因此建议同学们可以先尝试一下pm.sleep\_mode=0和ro.ril.disable.power.collapse=1组合使用，看看是否有bug，如果没有那自然使用此种模式，毕竟最省电了(极端省电)。对于稳定与省电兼得，可使用如下组合：pm.sleep\_mode=1ro.ril.disable.power.collapse=0这样射频在pm.sleep\_mode=1下不会被关闭，而进入休眠模式后射频会关闭。

电量方面就先写这些。通过调节本期参数可以做到待机只有1天~待机几乎5天(安装常用软件)。其他的例如射频锁休眠时间，资源分配优先级等一些参数此处不再给出。主要原因是那些东西太过底层，说实话专门做射频的工程师都不一定调的好。大家用起来容易被表象所迷惑，而且那些参数对节电用处不大，信号强弱也可以切换基带实现。望诸位谅解。

Build.prop (编辑 /system/build.prop 文件(需要root, 可以用文件管理器或者其他root explorer)

如果build.prop里面有这些参数,只需要把数值更改

1. 强制把Home程序驻入内存.

参数:

ro.HOME\_APP\_ADJ=1

2.提高 JPG 质量为 100%

参数:

ro.media.enc.jpeg.quality=100

3. VM 虚拟堆大小; 提高 RAM

参数:

dalvik.vm.heapsize=48m

4. 使用 GPU 渲染UI

参数:

debug.sf.hw=1

5. 减少拨号后出现的延时

参数:

ro.telephony.call\_ring.delay=0

6.提高滑动响应

参数:

windowsmgr.max\_events\_per\_sec=150

7.电池优化

参数:

wifi.supplicant\_scan\_interval=180

pm.sleep\_mode=1

ro.ril.disable.power.collapse=0

8. 禁止调试通知图标出现在状态栏处

参数:

persist.adb.notify=0

9. 提高全局触摸屏响应

参数:

debug.performance.tuning=1

video.accelerate.hw=1

10. 提高图片和摄像质量

参数:

ro.media.dec.jpeg.memcap=8000000

ro.media.enc.hprof.vid.bps=8000000

11. (3G) 信号优化

参数:

ro.ril.hsxpa=2 ro.ril.gprsclass=10

ro.ril.hep=1 ro.ril.enable.dtm=1

ro.ril.hsdpa.category=10

ro.ril.enable.a53=1

ro.ril.enable.3g.prefix=1

ro.ril.htcmaskw1.bitmask=4294967295

ro.ril.htcmaskw1=14449

ro.ril.hsupa.category=5

12. 网络速度优化

参数:

net.tcp.buffersize.default=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.wifi=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.umts=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.gprs=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.edge=4096,87380,256960,4096,16384,256960

13. 禁止拨号后出现黑屏.

参数:

ro.lge.proximity.delay=25

mot.proximity.delay=25

14.修复应用程序出现问题.

参数:

ro.kernel.android.checkjni=0

15.不通过按加/减音键唤醒手机

参数:

ro.config.hwfeature\_wakeupkey=0

16.屏幕点亮时强制开启功能键背光

参数:

ro.mot.buttonlight.timeout=0

17.不显示开机动画(system/media/bootanimation.zip动画将不显示,加速开机速度)

参数:

debug.sf.nobootanimation=1

18.其他优化

参数:

ro.config.hw\_menu\_unlockscreen=false

persist.sys.use\_dithering=0

persist.sys.purgeable\_assets=1

dalvik.vm.dexopt-flags=m=y

ro.mot.eri.losalert.delay=1000

19. 以下为适合 LG 机型的优化 ( 并不知道是什么优化, 木有棒机 )

参数:

persist.service.pcsync.enable=0

persist.service.lgospd.enable=0

user.feature.flex=true

user.feature.lgdrm=false

user.feature.lgresource=false

user.feature.lgpoweroff=false

user.feature.ls\_event=false

user.feature.ls\_normal=false

user.feature.sui=false

Init.d (system/etc/Init.d 一般民间rom都有该目录, 木有的话自己创建)

(需要 ROM 有/system/etc/init.d 目录, 并且需要busybox能正常使用. 创建一个空白文件,每个文件第一行都需要有 —> #!/system/bin/sh , 拖拽进 /system/etc/init.d 然后命名类似77tweaks \*( 数字,然后名称)

1. strict minfree handler 优化

参数:

echo “2048,3072,6144,15360,17920,20480″ > /sys/module/lowmemorykiller/parameters/minfree

2. 网络速度优化

参数:

echo “0″ > /proc/sys/net/ipv4/tcp\_times\*\*ps;

echo “1″ > /proc/sys/net/ipv4/tcp\_tw\_reuse;

echo “1″ > /proc/sys/net/ipv4/tcp\_sack;

echo “1″ > /proc/sys/net/ipv4/tcp\_tw\_recycle;

echo “1″ > /proc/sys/net/ipv4/tcp\_window\_scaling;

echo “5″ > /proc/sys/net/ipv4/tcp\_keepalive\_probes;

echo “30″ > /proc/sys/net/ipv4/tcp\_keepalive\_intvl;

echo “30″ > /proc/sys/net/ipv4/tcp\_fin\_timeout;

echo “404480″ > /proc/sys/net/core/wmem\_max;

echo “404480″ > /proc/sys/net/core/rmem\_max;

echo “256960″ > /proc/sys/net/core/rmem\_default;

echo “256960″ > /proc/sys/net/core/wmem\_default;

echo “4096,16384,404480″ > /proc/sys/net/ipv4/tcp\_wmem;

echo “4096,87380,404480″ > /proc/sys/net/ipv4/tcp\_rmem;

3. vm 虚拟机管理优化

参数:

echo “4096″ > /proc/sys/vm/min\_free\_kbytes

echo “0″ > /proc/sys/vm/oom\_kill\_allocating\_task;

echo “0″ > /proc/sys/vm/panic\_on\_oom;

echo “0″ > /proc/sys/vm/laptop\_mode;

echo “0″ > /proc/sys/vm/swappiness

echo “50″ > /proc/sys/vm/vfs\_cache\_pressure

echo “90″ > /proc/sys/vm/dirty\_ratio

echo “70″ > /proc/sys/vm/dirty\_background\_ratio

4. 内核优化

参数:

echo “8″ > /proc/sys/vm/page-cluster;

echo “64000″ > /proc/sys/kernel/msgmni;

echo “64000″ > /proc/sys/kernel/msgmax;

echo “10″ > /proc/sys/fs/lease-break-time;

echo “500,512000,64,2048″ > /proc/sys/kernel/sem;

5. 电池优化

参数:

echo “500″ > /proc/sys/vm/dirty\_expire\_centisecs

echo “1000″ > /proc/sys/vm/dirty\_writeback\_centisecs

6. EXT4 优化 (增加 I/O 调度)(需要 /system, /cache, /data 格式化分区为 EXT4))

a)移除 journalism

参数:

tune2fs -o journal\_data\_writeback /block/path/to/system

tune2fs -O ^has\_journal /block/path/to/system

tune2fs -o journal\_data\_writeback /block/path/to/cache

tune2fs -O ^has\_journal /block/path/to/cache

tune2fs -o journal\_data\_writeback /block/path/to/data

tune2fs -O ^has\_journal /block/path/to/data

b) 完美挂载选项

参数:

busybox mount -o remount,noatime,noauto\_da\_alloc,nosuid,nodev,nodiratime,barrier=0,nobh /system

busybox mount -o remount,noatime,noauto\_da\_alloc,nosuid,nodev,nodiratime,barrier=0,nobh /data

busybox mount -o remount,noatime,noauto\_da\_alloc,nosuid,nodev,nodiratime,barrier=0,nobh /cache

7.标识区为不可自旋转并增加缓存大小

参数:

LOOP=`ls -d /sys/block/loop\*`;

RAM=`ls -d /sys/block/ram\*`;

MMC=`ls -d /sys/block/mmc\*`;

for j in $LOOP $RAM

do

echo “0″ > $j/queue/rotational;

echo “2048″ > $j/queue/read\_ahead\_kb;

done

8. SD 卡速度优化

参数:

echo “2048″ > /sys/devices/virtual/bdi/179:0/read\_ahead\_kb;

9. Defrags database files

参数:

for i in \

`find /data -iname “\*.db”`

do \

sqlite3 $i ‘VACUUM;’;

done

9. 移除 logger

参数:

rm /dev/log/main

10. Ondemand governor (超频优化)

参数:

SAMPLING\_RATE=$(busybox expr `cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo\_transition\_latency`

\\* 750 / 1000)

echo 95 > /sys/devices/system/cpu/cpufreq/ondemand/up\_threshold

echo $SAMPLING\_RATE > /sys/devices/system/cpu/cpu0/cpufreq/ondemand/sampling\_rate

自动更改governor 与 I/O 调度

a) I/O 调度 (最佳: MTD 设备 – VR; EMMC 设备 – SIO) – 需要适应的内核

参数:

echo “vr” > /sys/block/mmcblk0/queue/scheduler

or

echo “sio” > /sys/block/mmcblk0/queue/scheduler

b) Governor (Best: Minmax > SavagedZen > Smoothass > Smartass > Interactive) – 需要内核支持.

参数:

echo “governor-name-here” > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor

12. 启动时自动-zipalign apk

(需要zipalign二进制, 位于 system/bin/zipalign, 可参考以下网址自行添加)

参考网址: http://forum.xda-developers.com/showthread.php?t=860586

13. 进程常驻优化

参考网址:

http://forum.xda-developers.com/showthread.php?t=1137554

http://forum.xda-developers.com/showthread.php?t=1205744

移动 dalvik-cache 到缓存分区 (如果分区够大的话) 扩大data分区

参数:

CACHESIZE=$(df -k /cache | tail -n1 | tr -s ’ ’ | cut -d ’ ’ -f2)

if [ $CACHESIZE -gt 80000 ]

then

echo ”Large cache detected, moving dalvik-cache to /cache”

if [ ! -d /cache/dalvik-cache ]

then

busybox rm -rf /cache/dalvik-cache /data/dalvik-cache

mkdir /cache/dalvik-cache /data/dalvik-cache

fi

busybox chown 1000:1000 /cache/dalvik-cache

busybox chmod 0771 /cache/dalvik-cache

# bind mount dalvik-cache so we can still boot without the sdcard

busybox mount -o bind /cache/dalvik-cache /data/dalvik-cache

busybox chown 1000:1000 /data/dalvik-cache

busybox chmod 0771 /data/dalvik-cache

else

echo ”Small cache detected, dalvik-cache will remain on /data”

fi

15. 禁止手机正常化休眠

参数:

mount -t debugfs none /sys/kernel/debug echo NO\_NORMALIZED\_SLEEPER > /sys/kernel/debug/sched\_features

16. OOM 分组和优先调整 – SuperCharger

参考网址: http://forum.xda-developers.com/showthread.php?t=99127

6 GPS.conf(编辑 /system/etc/gps.conf 需要root, 可以用文件管理器或者其他root explorer,如果没有该文件请自行创建)

加速 GPS 定位时间和信号.

a) 亚洲 NTP 服务器 (替换europe为asia )

参数:

NTP\_SERVER=europe.pool.ntp.org

XTRA\_SERVER\_1=http://xtra1.gpsonextra.ne t/xtra.bin

XTRA\_SERVER\_2=http://xtra2.gpsonextra.ne t/xtra.bin

XTRA\_SERVER\_3=http://xtra3.gpsonextra.ne t/xtra.bin

b) A-GPS SE supl 优化 (比 Nokia’s 或 Google’s 的都好)

参数:

SUPL\_HOST=supl.sonyericsson.com SUPL\_PORT=7275

其他优化/小提示

1. 修改hosts文件(屏蔽广告或爬楼梯)

(使用屏蔽广告的hosts前先静心想想,尼玛啊你这样做让做免费app的作者情何以堪啊!!)

地址: 屏蔽广告Hosts:

http://www.mvps.org/winhelp2002/hosts.txt

http://pgl.yoyo.org/adservers/se ... =0;hostformat=hosts

2. 使用 CyanogenMOD’s APN 列表文件 – 是目前最全的APN列表. (亲.我们不需要那么全的, 亲..)

参考: system/etc/apns-conf.xml

3. 使用 UOT kitchen 美化主题…

地址: http://uot.dakra.lt/

4. 使用 Google’s dns 服务器

参考:

打开resolv.conf (如果 system/etc 目录木有自己建立一个. 然后添加以下两行:

nameserver 8.8.8.8

nameserver 8.8.4.4

5. 更新Superuser. apk

(授权管理程序 一般都是用CM团队的,查看版本号可以打开那个独眼骷髅骨头头像的那个程序.. 应该就能看见了) 和su 二进制文件到最新版本. 目前最新是3.0 beta)

地址: http://goo-inside\*\*\*superuser/

6. 在sqlite处禁止同步功能

 (国内牛人,现在好像使用I9100 – 他在xda开放了更改后的源码. 不过你们不需要把这个功能禁止) /system/lib/libsqlite.so)

前言:

       build.prop在system的根目录下. 并且需要 root权限才能更改和查看哦.

       你可以在手机上,先把build.prop这个文件复制到你的SD卡上,再用USB链接电脑,然后把他放到电脑里修改,这样更快.

如何修改:?

           你打开之后,直接拉到最下面,然后另起一行(这个意思你懂把?),并且复制这些优化的代码进去就可以了.

更好的录像.照相优化（中文不用复制）

ro.media.capture.maxres=8m

ro.media.capture.fast.fps=4

ro.media.capture.slow.fps=120

ro.media.panorama.defres=3264x1840

ro.media.panorama.frameres=1280x720

ro.camcorder.videoModes=true

wifi速度优化

net.ipv4.tcp\_ecn=0

net.ipv4.route.flush=1

net.ipv4.tcp\_rfc1337=1

net.ipv4.ip\_no\_pmtu\_disc=0

net.ipv4.tcp\_sack=1

net.ipv4.tcp\_fack=1

net.ipv4.tcp\_window\_scaling=1

net.ipv4.tcp\_times\*\*ps=1

(注意，上面那一句出现了\*\*，而正确的应该是 t a m,估计是论坛屏蔽了，以为是“他吗”这句粗口）

net.ipv4.tcp\_rmem=4096 39000 187000

net.ipv4.tcp\_wmem=4096 39000 187000

net.ipv4.tcp\_mem=187000 187000 187000

net.ipv4.tcp\_no\_metrics\_save=1

net.ipv4.tcp\_moderate\_rcvbuf=1

General Performance（普通的基本优化）

debug.sf.hw=1

dalvik.vm.heapsize=48m（假如你是4.0系统，把48改为128）

persist.sys.ui.hw=1

Faster Scrolling（滑动的更快，更流畅的意思）

ro.max.fling\_velocity=12000

ro.min.fling\_velocity=8000

Saves power（省电）

ro.ril.disable.power.collapse=1

pm.sleep\_mode=1

wifi.supplicant\_scan\_interval=180

Raises quality of images（图像质量优化）

ro.media.enc.jpeg.quality=100

media.stagefright.enable-player=true

media.stagefright.enable-meta=true

media.stagefright.enable-scan=true

media.stagefright.enable-http=true

wifi速度优化

net.tcp.buffersize.default=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.wifi=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.umts=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.gprs=4096,87380,256960,4096,16384,256960

net.tcp.buffersize.edge=4096,87380,256960,4096,16384,256960

Makes apps load faster and frees more ram.（使程序加载更快，更多空余的RAM，就是运行内存）

dalvik.vm.dexopt-flags=m=y

Improve 3g data speeds （提高3g网络的数据传输速度）

ro.ril.hsxpa=2

ro.ril.gprsclass=10

ro.ril.hep=1

ro.ril.enable.dtm=1

ro.ril.hsdpa.category=10

ro.ril.enable.a53=1

ro.ril.enable.3g.prefix=1

ro.ril.htcmaskw1.bitmask=4294967295

ro.ril.htcmaskw1=14449

ro.ril.hsupa.category=5

Increase overall touch responsiveness（提高整体触摸响应）

debug.performance.tuning=1

video.accelerate.hw=1

Disable blackscreen issue after a call（通话后禁止屏幕黑屏）

ro.lge.proximity.delay=25

mot.proximity.delay=25

Fix some application issues（修复应用程序报错问题）

ro.kernel.android.checkjni=0

Force button lights on when screen is on（当屏幕亮时，强制点亮按键灯光）

ro.mot.buttonlight.timeout=0

6. Helps scrolling responsiveness 提高滚动响应

Code:

windowsmgr.max\_events\_per\_sec=200

8. Miscellaneous flags多方面优化

Code:

ro.config.hw\_menu\_unlockscreen=falsepersist.sys.use\_dithering=0

persist.sys.purgeable\_assets=1

dalvik.vm.dexopt-flags=m=y

ro.mot.eri.losalert.delay=1000