Improving performance of key “External Projects” in Android

Khasim Syed Mohammed, Linaro

Objective

- What these external projects are ?
- Why we should be improving these ?
- How to approach
- What Linaro has done in this space so far ?
- What's on our road map ?
- Community participation request ..

Android's external folder

aosp@android: $ ls

abi	build	development	frameworks	Makefile
prebuilts	art	cts	device	hardware
ndk	sdk	bionic	dalvik	docs
libcore	packages	**external**	bootable	developers
system	libnativehelper	pdk	tools	

We are here

Android’s external folder

•Android is, in many ways, just another Linux distribution

•As such, it includes code from many FOSS projects in the “external” folder ...

•… and quite frequently, isn't in sync with what upstreams are doing nor improved for performance.

Current situation

Android imports an external FOSS project into its git repository

- (sometimes a released version, sometimes a git or svn snapshot)

- Patches to make it work with Android (and sometimes to add, remove or modify some functionality) are added inside Android's git repository

- There is little or no effort made to upstream those changes, some changes are a little bogus (checking in a config.h generated by autoconf to avoid the need to call configure, ...)

- A newer upstream release may or may not be merged into Android – if at all, merges typically happen months after the upstream release

- Android has no concept of updating an individual component (e.g. openssl) – often leading to important upstream updates being ignored by device makers

Analyzing external folder

Switching to another document that gives more details

Overview of external folder in Linaro repositories

Out of 175 components in the external folder,

Team	Projects	Details
Google	11	Updated to latest versions, might be tracking and maintaining for performance.
Linaro	11	Improving performance and upstreaming
Linaro	5	Might pick additional few projects
Not applicable	61	These projects are not applicable for performance, falls under the category of fonts, no source availability, to run on host machine, etc.
	30	Related to testing, used in CTS, can be improved
	57	No takers yet ..
	175	Total components in external folder

Improving the performance - Approach

- Approach I
  - Many components can be improved for performance and functionality by just migrating to latest versions
  - Identify and file AOSP bugs for such components.

- Approach II
  - Profile and benchmark the software component, identify the bottleneck
  - Find possible improvement by using external hardware accelerator, new libararies, compiler options, CPU v/s GPU etc.

AOSP v/s external projects

- Patches needed for Android go upstream ASAP (#ifdef-ed if necessary)

- Upstream releases can be merged into Android quickly and painlessly because all relevant patches are already there

- Upstream releases actually are merged in a timely manner

- Android adopts some sort of upstream packaging so e.g. an OpenSSL security update can be pushed even to people who are stuck with an ancient Android version

Importance of migration to latest versions

•They introduce improvements and add new features that continually increase efficiency.

•Latest version ensures they have fixes for prior bugs.

•Obsoleted versions may inadvertently result in 'forgotten' and eventually restraining with a version that is no longer supported.

•Unfortunately, accommodating a recent release may require more effort, and delay.

Linaro is working on …

Approach – I :
Code modifications and upstream to AOSP or respective project

clang		***mesa3d***
compiler-rt		***qemu***
e2fsprogs		***sqlite***
fdlibm		***webp***
jpeg		***webrtc***
libpng		

Approach - II :
Bug Request to migrate to latest

flac		***libxml2***
libogg		***libxslt***
libvorbis		***tinyxml***
gcc-demangle		***tinyxml2***
sfntly		

Few optimizations explained in detail

BIONIC – Cortex C strings optimizations

Approach – II :
https://wiki.linaro.org/Platform/Android/CortexStringsInBionic

The following functions from cortex-strings still perform better than AOSP master and should be merged:

- strlen
- memchr
- memcpy

The following functions from AOSP master perform better than cortex-strings and should be merged the other way so glibc can benefit:

- memset

There is no relevant difference in performance between both implementations of:

- strcpy
- cortex-strings currently doesn't implement strchr

OpenSSL - Optimizations

Approach – I (just migrating to latest version improves a lot)

Wiki – TBD.

- Open SSL on AOSP is 1.0.1f and there is NEON optimizations available in 1.0.2 beta
- Google might be migrating only after a stable release is publicly available.
- Linaro provides the patch that can get 1.0.2 beta working on AOSP master.

SQLite - Optimizations

Approach – I and II
https://wiki.linaro.org/Platform/Android/SQLiteOptimization

- Migrating to latest:
  - Android KitKat (4.4) ships with SQLite 3.7.11. An experiment was carried out to move to a newer version (3.8.3.1) and the newer SQLite package showed a 7% improvement over the older version when running the RL Benchmark SQLite Android app.
- Using optimized Cortex C strings :
  - Since all SQL statements are in plain text format, there is inherently a lot of string operations. Thus, one way to improve the performance of SQLite is to optimize the Bionic C string library. Showed an improvement of 15 %
- Switching to F2FS

Immediately on our roadmap

- The Guava project contains several of Google's core libraries that we rely on in our Java-based projects: collections, caching, primitives support, concurrency libraries, common annotations, string processing, I/O, Guava and so forth.

- Zib : Some thing like,https://github.com/0xlab/0xdroid-external_zlib/blob/4f9dfb6ca8dba01798b49c7e82cfe1ecda3415f2/arm/adler32.c

- fdlibm

Open for discussion ..

ant-glob	***dnsmasq***	***jsr305***	***nist-sip***	***v8***
antlr	***ganymed-ssh2***	***libcap-ng***	***oauth***	***valgrind***
apache-harmony	***grub***	***libpcap***	***okhttp***	***webkit***
apache-http	***harfbuzz***	***libphonenumber***	***opencv***	***wpa_supplicant***
apache-xml	***harfbuzz_ng***	***libselinux***	***openssh***	***openssl***
arduino	***icu4c***	***libvpx***	***ppp***	***tinyalsa***
bluetooth	***iproute2***	***libyuv***	***protobuf***	***xmp_toolkit***
chromium	***ipsec-tools***	***marisa-trie***	***regex-re2***	***mp4parser***
dbus	***iptables***	***mdnsresponder***	***smali***	***jhead***
srtp	***jmonkeyengine***	***mksh***	***tcpdump***	***dhcpcd***

At least they can be migrated to latest versions for improvements

Open for discussion – Not confident

Fonts		Host	Updated	No new releases	Other		
cibu-fonts	astl	bison	aac	android-clat	liblzf	dropbear	quake
lohit-fonts	busybox	eclipse-basebuilder	bouncycastle	android_input_bridge	libppp	elfutils	replicaisland
naver-fonts	eyes-free	eclipse-windowbuilder	checkpolicy	bsdiff	libsepol	hyphenation	safe-iop
noto-fonts	ffmpeg	fat32lib	eigen	bzip2	lzma	javasqlite	skia
sil-fonts	htop	genext2fs	giflib	dexmaker	netcat	libffi	stlport
	jack	yaffs2	iputils	doclava	open-vcdiff	libmtp	zxing
	ncurses		lrzsz	expat	speex	libnfc-nci	
	sonivox		openfst	fsck_msdos	tagsoup	libnl	
	srec		tremolo	jline	tinycompress	libusb	
	svox		pixman	jmdns	zlib	libusb-compat	
	x264		freetype	libgsm		markdown	

Open for discussion – Any better method ?

	Unit testing	Benchmark		
android-mock	hamcrest	embunit	blktrace	stressapptest
easymock	proguard	gtest	iozone	stringbench
littlemock	javassist	junit	memtester	xmlwriter
mockwebserver	jdiff		netperf	linux-tools-perf
roboelectric	jsilver		strace	nist-pkits
objenesis	lava-blackbox			
emma	google-diff-match-patch			

- Do we have any better alternative in other Linux distributions that we could use ?
- Do they need to be updated for new parameters or test coverage

Conclusion

- The Android “external” components can be optimized and improved.
- Linaro has been continuously tracking, analyzing important components.
  - Will be improving by necessary code modifications and submitting changes to AOSP.
  - Will file bugs to AOSP to migrate to latest versions where applicable.
- Request community participation in improving external components.
- Any suggestions and improvement areas can be discussed in any public forum.

Let's work together

More about Linaro: www.linaro.org

Register to participate or to get latest updates

linaro-android@lists.linaro.org