Supporting Multiple Devices in Android

Rob Herring
Goals - HAL Consolidation

● Develop kernel support once across distros (Android, ChromeOS, traditional Linux)
● Make adding devices and updating to new Android versions easier
● Devices with mainline kernels just work
● Reduce/eliminate need for custom HALs
● Create an upstream community for Android devices
How to add a device today

● “Creating a new device is trivial”
● Find some existing device as starting point
● `cp -ar device/<vendor>/<old dev> device/<vendor>/<new dev>`
● Rename files, variables, etc. with new device/product names (goodbye diff)
● Scour thru other devices for examples of any features to add
● Build and run to debug what settings were missed or mistyped
Problems

● How to tell the configuration difference between 2 builds/devices?
  ○ What variables change the compilation?
  ○ Arbitrary structure for inheritance
  ○ Many variables set to the default values
  ○ Stale variables
  ○ No dependency or type checking of variables

● How to upgrade to new Android versions and do it quickly and frequently?

● How to support 2, 10, or 96 different boards?

● How to upgrade to new a Android version (or security update) on 2, 10, or 96 different boards?
One build, many devices

- Goal is one Android build/filesystem per cpu architecture while maintaining configurability for device specific builds: [http://tinyurl.com/zscbbrx](http://tinyurl.com/zscbbrx)
- A directory per feature for features more than just a config variable
- KConfig based configuration for features
- Supporting DB410c, HiKey, Nexus 7, QEMU, RaspberryPi 3
- Tablet/phone or TV targets
- Image and device (HAL) config
KConfig: What’s Next

- Upstream in AOSP?
- Anything the next device needs
- Custom compiler and compiler flags
- Kernel build integration
- malloc selection
- f2fs or other filesystems
How do I add my device?

- Add CONFIG_MY_AWESOME_DEVICE? **No!**
- Have a mainline kernel?
- What is the difference between your device and some existing device?
- New GPU? Probably needs some work.
  - Mesa based? Good
  - Mali? Does it work with other vendors’ Mali?
Issues with Kconfig

- Not integrated into AOSP build (maybe a good thing)
  - Run make in device repo
  - Run lunch
  - Run AOSP make
- May leave stale files in target images
- Breaks the (weak) separation of build config and product config
- defconfigs can get stale and out of sync with Kconfig