webOS, OpenEmbedded Use Case

Prepared for:

Embedded Linux Conference

April 29 – May 1, 2014

Iyad Qumei

LG Electronics

Silicon Valley Lab

Introduction

- What is webOS
- Describe customization work
- Build environment
  - Build speed, scaling issues
  - Build visualization
- Development work-flow challenges
- OE assessment summary
- webOS TV

What is webOS

- Web-centric platform
- At the core is Open webOS, currently hosted on github.com/openwebos
- The kernel is Linux
- Uses OpenEmbedded build system

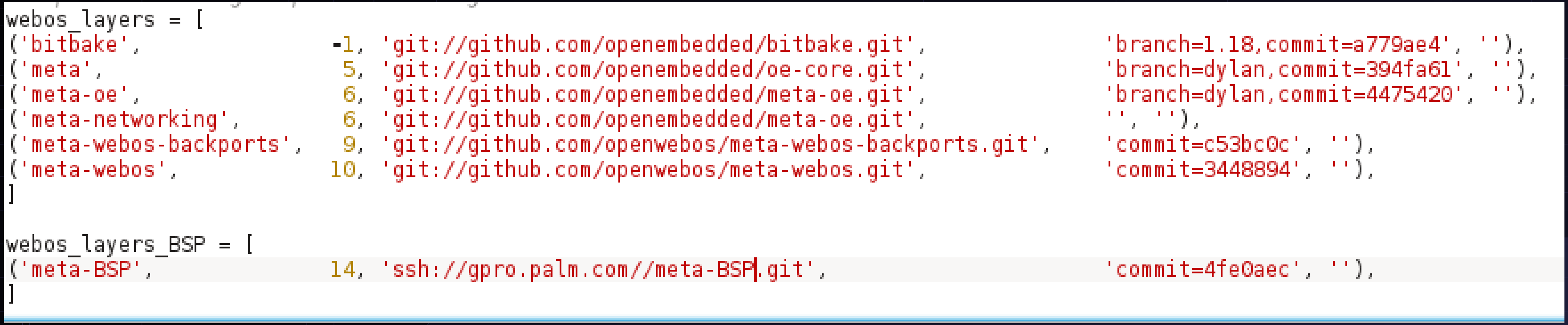
Customization

- WebOS is based on official releases of OE with fixes back-ported between releases on as needed bases.
- Custom Linux kernel and build procedure to accommodate features and packaging
- Custom bootable images for specific target devices
- External toolchain utilization
- Custom build environment setup.

Build Environment

- Relies on OE layers architecture
- Build script to automate assembly of work area
- Layers definition file

  ('layer-name', priority, 'URL', 'submission', 'working-dir'),

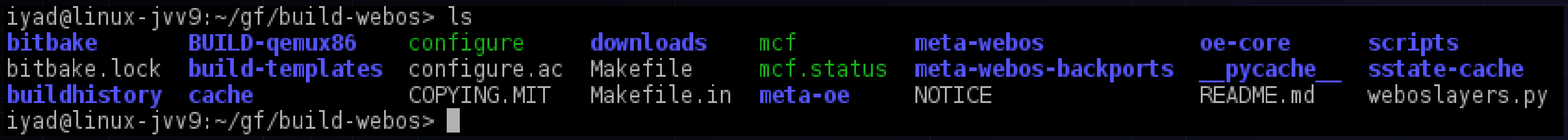


```
webos_layers = [
('bitbake',               -1, 'git://github.com/openembedded/bitbake.git',              'branch=1.18,commit=a779ae4', ''),
('meta',                   5, 'git://github.com/openembedded/oe-core.git',              'branch=dylan,commit=394fa61', ''),
('meta-oe',                6, 'git://github.com/openembedded/meta-oe.git',              'branch=dylan,commit=4475420', ''),
('meta-networking',        6, 'git://github.com/openembedded/meta-oe.git',              '', ''),
('meta-webos-backports',   9, 'git://github.com/openwebos/meta-webos-backports.git',    'commit=c53bc0c', ''),
('meta-webos',            10, 'git://github.com/openwebos/meta-webos.git',              'commit=3448894', ''),
]

webos_layers_BSP = [
('meta-BSP',              14, 'ssh://gpro.palm.com//meta-BSP.git',                      'commit=4fe0aec', ''),
]
```

Build Environment (Cont.)

- Submission field contains repository checkout information for each layer
- Additional BSP layers definition

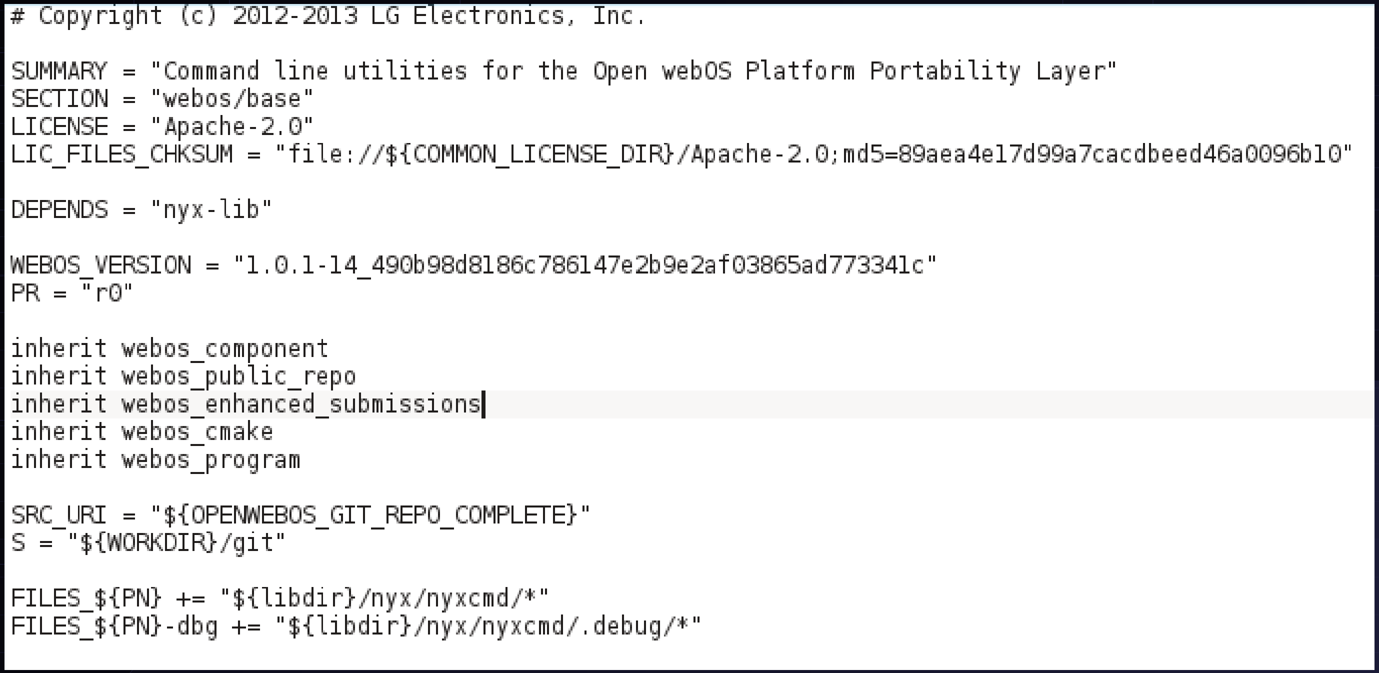


```
iyad@linux-jvv9:~/gf/build-webos> ls
bitbake       BUILD-qemux86    configure     downloads    mcf         meta-webos            oe-core      scripts
bitbake.lock  build-templates  configure.ac  Makefile     mcf.status  meta-webos-backports  __pycache__  sstate-cache
buildhistory  cache            COPYING.MIT   Makefile.in  meta-oe     NOTICE                README.md    weboslayers.py
iyad@linux-jvv9:~/gf/build-webos> 
```

Bill of Material (BOM)

- Central vs distributed definition
- Composite version format for individual components

<component-version>-<submission>[_<40-character-revision-hash>])



```
# Copyright (c) 2012-2013 LG Electronics, Inc.

SUMMARY = "Command line utilities for the Open webOS Platform Portability Layer"
SECTION = "webos/base"
LICENSE = "Apache-2.0"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/Apache-2.0;md5=89aea4e17d99a7cacdbeed46a0096b10"

DEPENDS = "nyx-lib"

WEBOS_VERSION = "1.0.1-14_490b98d8186c786147e2b9e2af03865ad773341c"
PR = "r0"

inherit webos_component
inherit webos_public_repo
inherit webos_enhanced_submissions
inherit webos_cmake
inherit webos_program

SRC_URI = "${OPENWEBOS_GIT_REPO_COMPLETE}"
S = "${WORKDIR}/git"

FILES_${PN} += "${libdir}/nyx/nyxcmd/*"
FILES_${PN}-dbg += "${libdir}/nyx/nyxcmd/.debug/*"
```

Build Speed

- Build time is critical for both individual and automated work
- Several factors were analyzed to improve build time; parallel threads, ICECC, RAM
- Components configuration and compilation dominated build time, 34.11%, 20.27% respectively
- Distributed computing proved to be most effective factor to reduce build time

Build Speed, Parallel Build

- Parallel build, results depend on build machine configuration
- Optimal configuration is twice number of cores

BB/J	4	8	12	16	32	64
4		2:32:14				
8	2:27:31	2:11:32	2:13:17	2:14:48	2:16:50	2:17:20
12		2:13:43				
16		2:15:33				
32		2:17:10			2:19:15	
64		2:17:13				

Build Speed, ICECC, RAM

- Distributed computing with ICECC

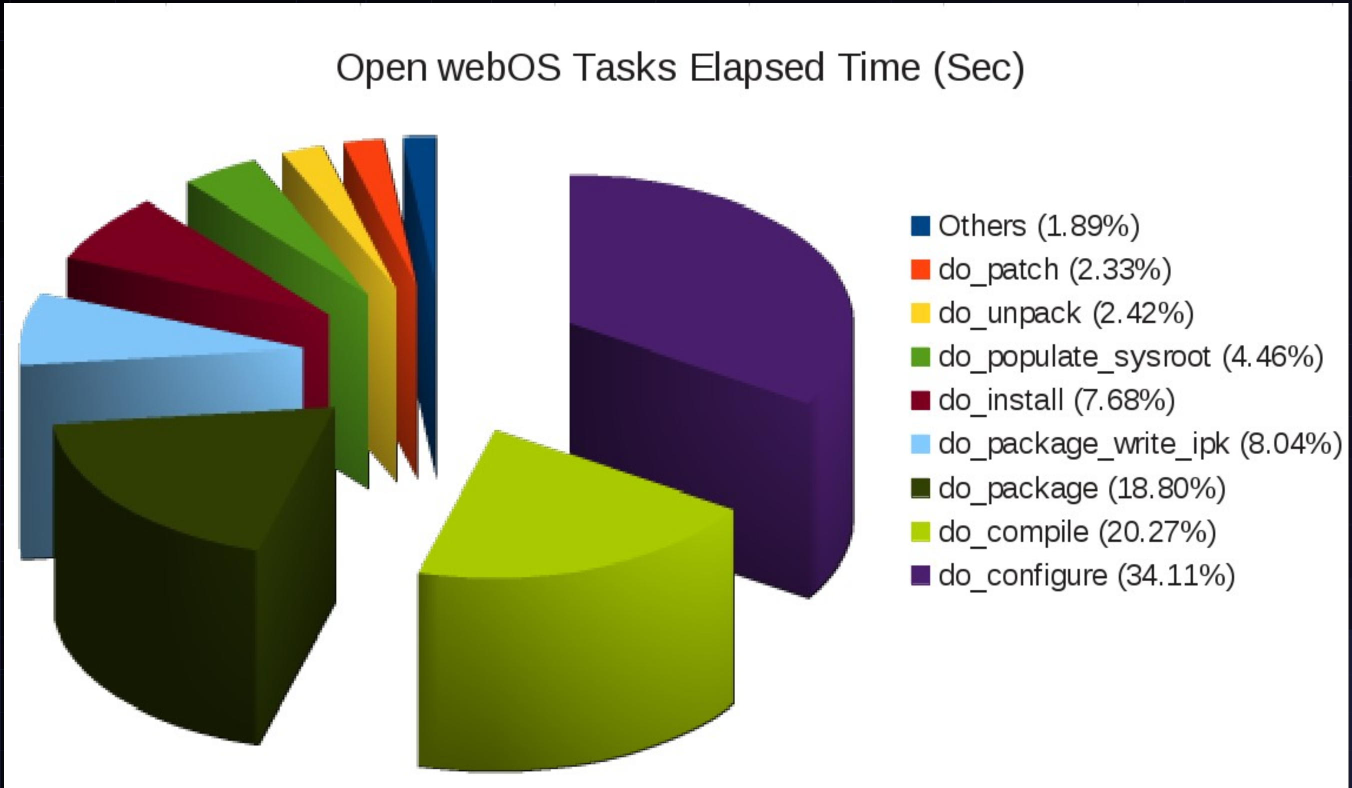
	webOS TV	Open webOS
Without ICECC	Real 150m11.800s User 850m6.800s Sys 95m38.387s	Real 99m00.000s User 371m46.400s Sys 62m21.660s
With ICECC	Real 108m38.352s User 331m6.270s Sys 84m51.826s	Real 79m12.000s User 163m41.670s Sys 48m40.680s

- RAM usage

	Elapsed	User Sec	Sys Sec	CPU Sec	Max Res. Mem	F/S In	F/S Out	Page Faults (I/O)	Page Faults Rec.
RAM	2:11:32	49711.06	4936.74	692	2,694,320	2,314,240	17,532,208	13050	1,492,283,491
Disk	2:14:55	48972.36	4976.15	666	2,694,784	3,198,312	149,395,616	18802	1,492,656,412

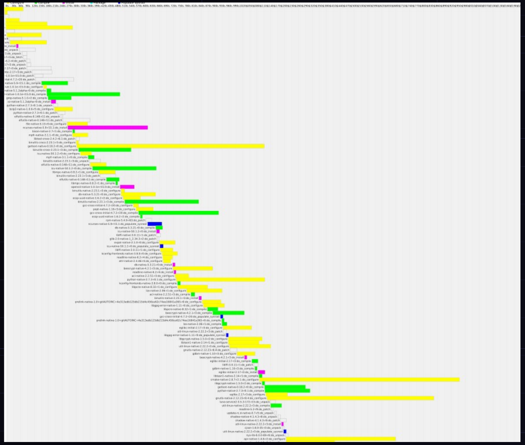
Build Visualization

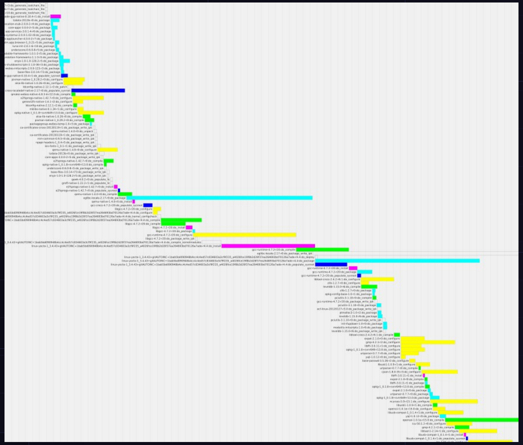
- Elapsed time per task as portion of total time

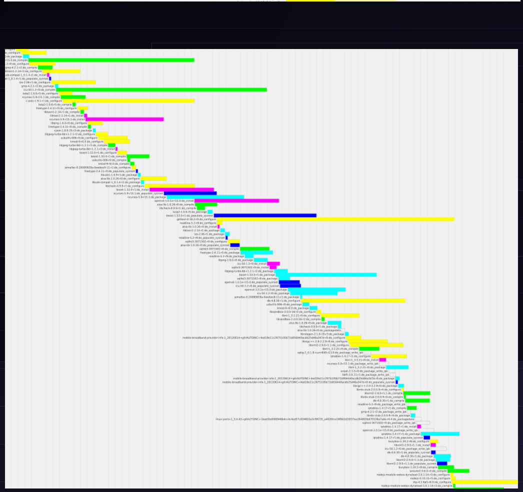


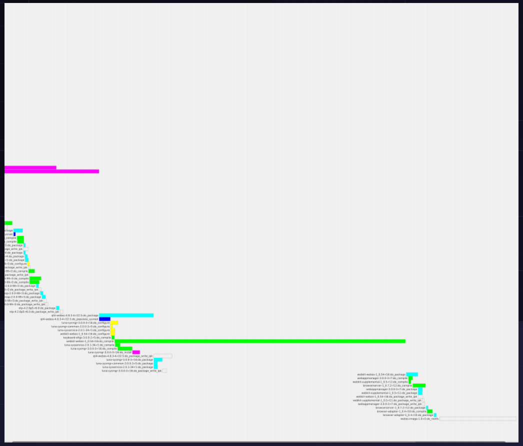
Build Visualization (cont.)

- With parallel options, bb & j, set build remains sequential
- This is shared characteristic with baseline OE “core-image-{minimal,basic...}
- Scaling issues with larger projects



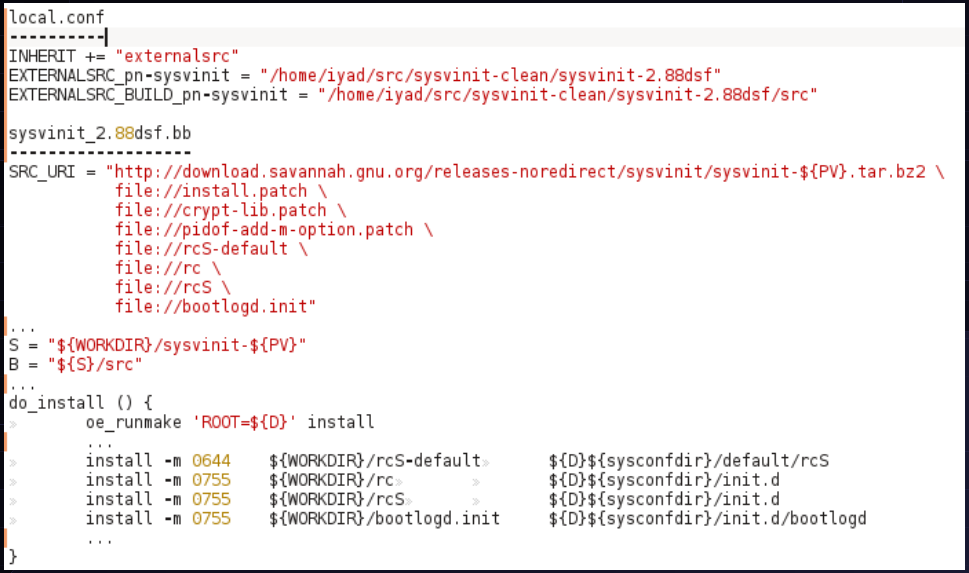






Development Work-Flow Limitations

- Typical work flow involves development for individual component
- Customized procedure introduced in the meta data can not be easily applied to local source



```
local.conf
----------
INHERIT += "externalsrc"
EXTERNALSRC_pn-sysvinit = "/home/iyad/src/sysvinit-clean/sysvinit-2.88dsf"
EXTERNALSRC_BUILD_pn-sysvinit = "/home/iyad/src/sysvinit-clean/sysvinit-2.88dsf/src"

sysvinit_2.88dsf.bb
-------------------
SRC_URI = "http://download.savannah.gnu.org/releases-noredirect/sysvinit/sysvinit-${PV}.tar.bz2 \
           file://install.patch \
           file://crypt-lib.patch \
           file://pidof-add-m-option.patch \
           file://rcS-default \
           file://rc \
           file://rcS \
           file://bootlogd.init"
...
S = "${WORKDIR}/sysvinit-${PV}"
B = "${S}/src"
...
do_install () {
        oe_runmake 'ROOT=${D}' install
        ...
        install -m 0644    ${WORKDIR}/rcS-default      ${D}${sysconfdir}/default/rcS
        install -m 0755    ${WORKDIR}/rc               ${D}${sysconfdir}/init.d
        install -m 0755    ${WORKDIR}/rcS              ${D}${sysconfdir}/init.d
        install -m 0755    ${WORKDIR}/bootlogd.init    ${D}${sysconfdir}/init.d/bootlogd
        ...
}
```

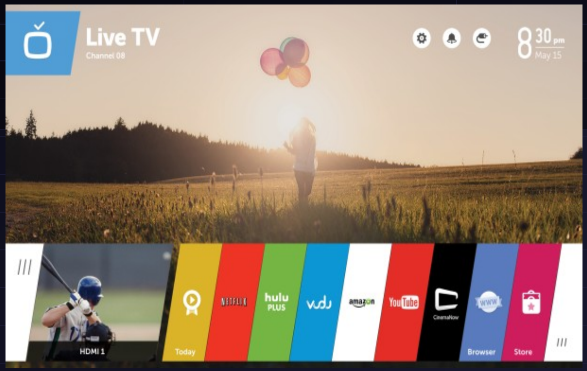
OpenEmbedded Assessment

- OE is a valuable build system. Lots of features, and the ability to customize makes it versatile
- Build procedure needs optimization to better control build time
- Component development from source has limitations, primarily utilization of changes to source introduced through meta data.
- Learning curve is significant
- More documentation on OE internals

WebOS TV

- LG introduced webOS TV during Consumer Electronics Show (CES) 2014
- WebOS TV was well received with positive coverage by the press

“*There's little doubt that LG has come up with a clever and simple way to run your TV, compared to the commonly overwrought interfaces seen on most other smart TVs.*” Theverge.


Live TV
Channel 08
8 30 pm
May 15
HDMI 1
Today
NETFLIX
hulu PLUS
vudu
amazon
YouTube
CinemaNow
Browser
Store