Dandified way to package management in Yocto Project

Jun 21th, 2018
Zheng Ruoqin, Fujitsu
zhengrq.fnst@cn.fujitsu.com
whoami

- Zheng Ruoqin, Fujitsu Ltd.
- Embedded Linux Developer
- In-House Embedded Linux Distributor of Fujitsu

- Our Distribution includes LTSI Kernel and is built with Yocto Project
- Our Distribution is used for
  - IVI, Server System Controller, Storage System, Network Equipment, Printer, etc.
Fujitsu’s contributions to Yocto community

- Data comes from yocto (2017-06-01 ~ 2018-06-01)

Contributions in yocto

- commits

<table>
<thead>
<tr>
<th>Layers</th>
<th>Changesets</th>
</tr>
</thead>
<tbody>
<tr>
<td>poky</td>
<td>112</td>
</tr>
<tr>
<td>oe-core</td>
<td>112</td>
</tr>
<tr>
<td>meta-oe</td>
<td>121</td>
</tr>
<tr>
<td>bitbake</td>
<td>0</td>
</tr>
</tbody>
</table>
Fujitsu’s contributions to Yocto community

- Data comes from yocto (2017-06-01 ~ 2018-06-01)

### Top changeset contributors by employer

<table>
<thead>
<tr>
<th>No.</th>
<th>employer</th>
<th>Changesets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intel</td>
<td>3285</td>
</tr>
<tr>
<td>2</td>
<td>Wind River</td>
<td>1399</td>
</tr>
<tr>
<td>3</td>
<td>Fujitsu</td>
<td>345</td>
</tr>
<tr>
<td>4</td>
<td>Axis Communications</td>
<td>118</td>
</tr>
<tr>
<td>5</td>
<td>simens</td>
<td>79</td>
</tr>
</tbody>
</table>

### Developers with the most changesets

#### poky

<table>
<thead>
<tr>
<th>No.</th>
<th>Our Developer</th>
<th>Changesets</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Huang Qiyu</td>
<td>83 (1.8%)</td>
</tr>
<tr>
<td>40</td>
<td>fan.xin</td>
<td>24 (0.5%)</td>
</tr>
</tbody>
</table>

#### oe-core

<table>
<thead>
<tr>
<th>No.</th>
<th>Our Developer</th>
<th>Changesets</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Huang Qiyu</td>
<td>83 (2.4%)</td>
</tr>
<tr>
<td>36</td>
<td>fan.xin</td>
<td>24 (0.7%)</td>
</tr>
</tbody>
</table>

#### Meta-oe

<table>
<thead>
<tr>
<th>No.</th>
<th>Our Developer</th>
<th>Changesets</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Huang Qiyu</td>
<td>78 (3.7%)</td>
</tr>
<tr>
<td>16</td>
<td>Zheng Ruoqin</td>
<td>20 (0.9%)</td>
</tr>
</tbody>
</table>
Agenda

Review of Yocto
- What is Yocto
- DNF in Yocto - overview
- DNF in Yocto - Cons

dnf-host - A way out of Yocto build system
- dnf-host overview
- dnf-host features

How to use dnf-host
- dnf-host Getting Started
- Demo

Future
- Support more package types
- Support different filesystem of rootfs
- Make TUI more user friendly
- Contribution plan
Review of Yocto

- What is Yocto
- DNF in Yocto - overview
- DNF in Yocto - Cons
What is Yocto

Yocto Project is an open source collaboration project that help you create custom Linux-based systems for embedded products.

https://www.yoctoproject.org/
What is Yocto

Build System

Output Image Data

Packages

Source Fetching

Patch Application

Config/Compile/Autoconf as needed

Output Analysis for Package Splitting plus Package relationships

.deb generation

.rpm generation

.ipk generation

QA Tests

Output Packages (deb, rpm, ipk)

Image Generation

SDK Generation

Image

Application Development SDK
In Yocto, DNF is the default Package management to generate rootfs and SDK.

Here is where DNF works

After packages were generated, The Yocto Build System will use DNF to generate the rootfs and SDK by installing Packages.
What is DNF

• **DNF** is a software package manager that installs, updates, and removes **packages** on **RPM**-based Linux distributions. It automatically computes dependencies and determines the actions required to install packages. DNF also makes it easier to maintain groups of machines, eliminating the need to manually update each one using **rpm**. Introduced in Fedora 18, it has been the default package manager since Fedora 22.

More information here:

https://github.com/rpm-software-management/dnf
DNF in Yocto - Cons

Issues we met

Typical case you meet when you try to modify your rootfs.

- Not the proper rootfs you want?
- Reconfigure the image bb file
- Rebuild the image again

User Configuration

Yocto Build System

Rootfs
Application Development SDK
New Rootfs
Issues we met

- You can only customize rootfs in Yocto build system which is not flexible enough.
- High waste of machine, time and developers for rootfs customization.
- Indeterminacy caused by rebuild.

Do we have a solution to avoid these cases?

Use dnf-host instead of the DNF in yocto build system.
dnf-host - A way out of Yocto build system

- dnf-host overview
- dnf-host features
Introduction

We redeveloped the DNF and Yocto to solve the issues mentioned above. We call it **dnf-host** which means it can work on the Cross-Development environment in your host for package management.

**Git-Repo:** [https://github.com/ubinux/dnf/tree/dnf-yocto2.5/](https://github.com/ubinux/dnf/tree/dnf-yocto2.5/)

**Status:** Support Yocto 2.4 and 2.5
dnf-host features

- Work out of Yocto build system.

- Make customization more convenient with TUI.

- BoM Management.
  - SPDX Management

- Save and reuse package list.

- Vendors can customize installation samples.
**dnf-host features**

**Yocto Build server**

Build Once, Deploy Anywhere

Build and Deploy for Each Targets

**Targets:** 50

Before:
- Rebuild time: 0.1 hour
- Deploy time: 1 hour
- Cost = (0.1 + 1) * 50 = 55 hours

After:
- Upgrade time: 0.1 hour
- Cost = 0.1 * 50 = 5 hours

Build Once, Deploy Anywhere

Simulation
dnf-host features

**dnf-host TUI**

- You can use the Text-based UI we developed for dnf-host

![Package Installer Interface](image)

- F1: select/unselect all
- F2: search
- F3: next
- F4: back
- F5: info
- F9: exit
dnf-host features

BoM Management

- Packages
- SRPM packages
- SPDX files
- Recipe files
- dnf-host
  - Deploy
  - Upgrade
  - Create archives

Build

Copyright 2018 FUJITSU LIMITED
What is SPDX

• The full name of SPDX is **Software Package Data Exchange**, which is a standard format for communicating the components, licenses and copyrights associated with a software package.

Kernel v4.14 added one-liners come from SPDX

```
# SPDX-License-Identifier: GPL-2.0
```

```makefile
# Makefile for the linux kernel.

CPPFLAGS_vmlinux.lds := -DTEXT_OFFSET=$(TEXT_OFFSET)
AFLAGS_head.o := -DTEXT_OFFSET=$(TEXT_OFFSET)
CFLAGS_armv8_deprecated.o := -I$(src)
CFLAGS_REMOVE_ftrace.o = -pg
CFLAGSREMOVE_insn.o = -pg
CFLAGS_REMOVE_return_address.o = -pg
```
**BoM Management - SPDX**

- Git Repository: https://github.com/dl9pf/meta-spdxscanner

- FOSS

- Pacthes come from 3rd party

Yocto

- meta-spdxscanner (dosocs2)

- meta

- meta-oe

- meta-……

SPDX files
SPDX for FOSS Compliance Management

What’s OpenChain Project?
- Activities to promote FOSS compliance in the software supply chain.
  https://www.openchainproject.org/

Make SPDX files work in OpenChain

Example of Compliance Management End-to-End Process

https://wiki.linuxfoundation.org/_media/openchain/openchain-curriculum-for-1-1.pdf
**dnf-host features**

Save and reuse package list

- Like kernel config, selected package list can be saved and reused.

First time you install

- acl
- attr
- base-files
- base-passwd
- base-passwd-update
- bash
- bash-bashbug
- bash-completion
- bash-completion-extra
- bc
- busybox

Next time you install

```
$ cat .config
base-passwd
bc
base-passwd-update
```

Load package list (Optional)

- Config file is customizable.
Vendors can customize installation samples.

- To satisfy customization requirements for users, vendors can add some predefined samples for installation.

  Select install type

  New --->
  Load package file --->
  Reference1 (busybox based root file system) --->
  Reference2 (systemd based root file system) --->

- The default samples directory is here.

  $ cat /opt/poky/2.5/sysroots/x86_64-pokysdk-linux/usr/share/dnf/busybox
  base-pwsswd
  base-files
  busybox
  busybox-mdev
  busybox-syslog
  busybox-udhcpc
  busybox-udhcpd
How to use dnf-host

- dnf-host  Getting Started
- Demo
Add dnf-host into toolchain

First you need to add dnf-host into your Yocto Project.

Get the Source Code

$ git clone https://github.com/ubinux/dnf.git -b dnf-yocto2.5
$ ls dnf/patches-yocto/
0001-Added-some-nativesdk-oss-for-dnf-host.patch
0002-Modified-the-URL-of-dnf-intead-of-upstream.patch

Apply these patches

$ cd poky
$ patch -p1 < XXX.patch

Bitbake the SDK

Build meta-toolchain to get sdk that includes dnf-host
$ bitbake meta-toolchain
How to use `dnf-host`

Then you can use `dnf-host` on the Cross-Development environment.

Install the toolchain

```
$ sh poky-glibc-x86_64-meta-toolchain-i586-toolchain-2.5.sh
```

Source the toolchain

```
$ . /opt/poky/2.5/environment-setup-i586-poky-linux
```

Init the environment

`prepare your rpm repo, then`

```
$ dnf-host init
```

Use TUI

```
$ dnf-host tui
```
There is a Demo!

You can reference to:
https://github.com/ubinux/dnf/tree/dnf-yocto2.5/README.md
Future

- Support more package types
- Support different filesystem of rootfs
- Make TUI more user friendly
- Contribution plan
Support more package types

- Support more pkg types, e.g. package_deb, package_ipk…

```
<table>
<thead>
<tr>
<th>RPM</th>
<th>DEB</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPK</td>
<td></td>
</tr>
</tbody>
</table>
```

dnf-host
Support different images of rootfs

- Support more image format, e.g. EXT4, Squashfs, UBIFS and so on.
Make TUI more user friendly

- Display packages in a hierarchical structure based on group information.

[ ] Bash
[*] Base-files
[ ] Base-passwd
……
[*] Coreutils
[*] Util-linux
……
[ ] Httpd
……

[*] Base → [ ] Bash
    [*] Base-files
    [ ] Base-passwd
    ……

[*] Utils → [ ] Coreutils
    [*] Util-linux
    ……

[ ] Web Server → [ ] Httpd
    ……
Contribution plan

- Send patches to oe-core.
- Continuous development.
- Release dnf-host follow the step of Yocto.
Any Questions?

zhengrq.fnst@cn.fujitsu.com
Fujitsu

shaping tomorrow with you