Binary Package Feeds for Yocto Project

Jon Mason, Arm

Bruce Ashfield, Xilinx

Yocto Project Summit 2019

What scares Yocto Project people?
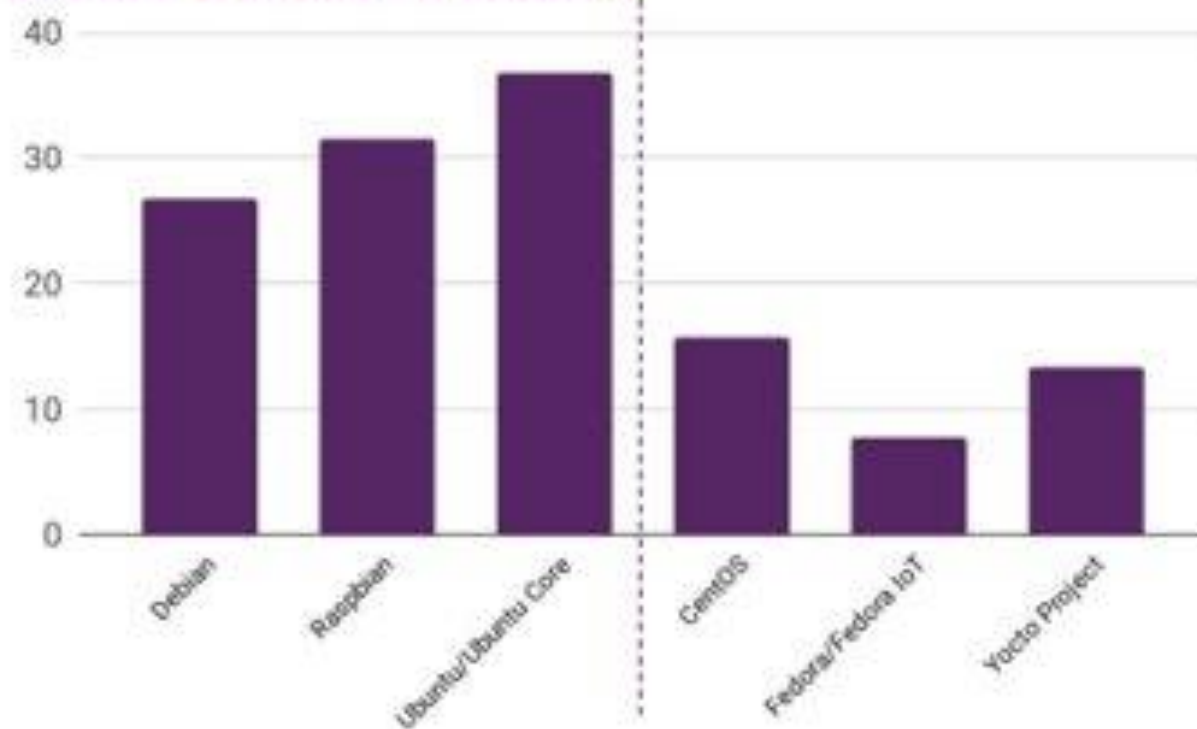
What should we fear?

Linux distributions

It's a Debian World...

Why are people running full distros on embedded devices?

CPUs

x86

- Core 2 Quad (Q6600) - 2006 - 2-3GHz
- Phenom X4 9850 - 2008 - 4 cores @ 2.5GHz
- Phenom II X4 925 - 2009 - 4 cores @ 2.8GHz
- Via Nano QuadCore C4650 - 2015 - 4 Cores @ 2GHz
- Core i3-8300 - 2018 - 4 Cores at 3.7GHz
- Celeron 5205U - 2019 - 2 Cores at 1.9GHz

Arm

- SDM845 - 2018 - 4 + 4 cores (2.8 GHz + 1.8GHz)
- RK3399 - 2016 - 2 + 4 cores (2.0GHz + 1.5)
- Broadcom BCM2711 - 2019 - Quad core Cortex-A72 - 1.5GHz
- Qualcomm Snapdragon 855+ Mobile Platform - 2019 - 485 Octa-core CPU @ up to 2.96 GHz
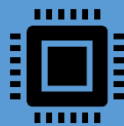
Others

Memory

Storage

PCIE

Distros

Easy to install

Easy to update

Easy to install new software

The cost for using distros is the higher amount of resources (e.g., CPU, RAM, and Storage), but now embedded devices have similar amounts to desktops

Yocto is not just for embedded

- Containers
- Desktop?

What can YP do to address this?

Extensibility - Binary Package Feeds

- The biggest benefit of a traditional distro is their ability to install and upgrade packages

- Yocto already has the ability to build rpms, debs, and ipkgs. So, all we need to do is collect those into a standard location, and have a way of installing them

Issues

- The point of Yocto/OE is the high level of configurability
- A binary package feed would need to be generic to be useful

How would this look?

- x86
  - Build for the generic x86_64

- Arm
  - 32bit, Little Endien, hard float, ARMv7

- Aarch64
  - Build for the generic ARMv8

Alternative funding for software vendors?

- could provide a trusted package location for paying customers that contains packages with the latest CVE fixes for a given release
  - A way to fund LTS?
- Vendor lock-in for this, as the customer would need to keep paying as long as they wanted access

Open Questions

- Initial installation?
- How do we want to handle release upgrades?
- How do we want to handle arches, sub-arches, and other variants

Thanks