

ML packing in Debian

Wookey

• Package ML stack – Tom Gall/Linaro

- arm compute library (ARM: Georgios Pinitas)
- ArmNN (ARM: Francis Murtaugh)
- tensorflow/tf-lite (google)
- TVM (ARM: Leandros Nunes)
- ONNX, Caffe, more?

Start at bottom of stack and work up

• Distro Packaging

- Debian process new to arm people
- Explain distro context/requirements
- Unteach bad habits/teach good ones
- Improves code quality
- I had to learn/change processes too
- Get help from community
- Internal packages different from external ones

Initial Packaging Process

- Think about package names (libarmcompute, libarm-compute)
- ITP- 'Intent To Package' bug
- Check codebase copyright/licencing: document, remove non-free bits
- Find/package dependencies
- Package it
- Upload
- Wait in NEW queue
- Re-upload source-only (new!)

Timeline

- Started Dec 2019 for mlplatform project
- Turned out to be 18 months not 2
- ACL:
 - Started Dec 2019 (v19.11)
 - Uploaded May 2020 (to NEW)
 - In unstable July 2020
 - In testing October 2020 (source-only upload delay)
 - v20.08 November 2020 (in bullseye)

Arm Compute Library

- Georgios made a working package himself
 - excludes non-arm arches
 - defaults to static build
 - won't build if not git repo (commit lookup)
 - tarball release not matching git release
 - 350MB of docs in tarball (2Mb codebase)
- Other stuff
 - non-free test images
 - tickled a compiler ICE in gcc9 (neon)
 - mathjax in docs pulling files off net and logo from arm
 - No pkgconfig
 - Add runtime checking for neonless hardware

ArmNN

- use system libraries and pkgconfig
- uses mali - needed dynamic loading of .so
- runtime neon checks - dynamic backends
- ml testing images non-free
- fun with pthreads
- nobble some tests (no network, no GPU)
- **Use named errordefines** (vary by architecture)
- Add 'make install'
- remove rpaths
- Bodge missing tensorflow schema file (and **build from source!**)
- python bindings, dynamic acl bindings

Tensorflow

- Big and complicated, has history.
- Uploaded Nov 2018, removed Aug 2019
- Bazel uploaded Oct 2020
- Repackaged (mostly) by debconf (June 2020, Michael R Crusoe)
- Builds, but not TF-lite
- Some dependencies missing

Learnings

- Don't exclude architectures
- Native-build by default (for any arch)
- Build in context (not docker/qemu/linaro toolchain/ubuntu)
- SONAME inflation: should move with ABI
- Offline building: deterministic, reproducible
- Build from release, not 'today's git'
- Marketing don't get to write the description!

Tooling

- Use sbuild for debian – sbuild-debian-developer-setup
- tarballs+quilt vs 12 flavours of git
 - Salsa.debian.org (devel and CI)
 - Dgit.debian.org (archive in git repo form)
- dgit/git-buildpackage/git-debbase/dgit-merge
 - I needed to learn this stuff, but it sucked up a lot of time.
 - Picked git-debbase for armNN with turned out to be a PITA

People and community

- Everyone was great
 - Georgios and Francis learned fast
 - Francis became DM (Debian Maintainer)
 - They put up with my lumpy attention
- Debian people very helpful
 - Mo Zhou (Debian deep learning team)
 - Olek Wojnar (debian science, debian-bazel)
 - Joe Hicks, Yun Peng (google) keen to help
- Community
 - Packaged ONNX and bazel and tensorflow (partly)