Linaro CONNECT
ASIA 2013 (LCA13)

LAVA & CI Component Review

LAVA & Infrastructure teams

Current architecture by James, Danilo, Dave

- Source code (git, bzr)
- Builds (ci.linaro.org, android-build, cbuild, PPAs)
- Publishing (snapshots.linaro.org)
- Testing (LAVA)
- Data collection (android-build, LAVA dashboards...)
- Releasing (releases.linaro.org)

Ongoing changes by James

- Web interface scaled back to reporting only
- Job management by command line interface
- Job descriptions are Python scripts
- We provide the API and will adapt as needed.
- Jobs can be run locally as well as on the CI infrastructure
  - Codify your workflow into a job
  - Run the parts of a job you need (e.g. run build + test steps locally pre-checkin without the checkout step).

Code hosting/Gerrit by Stevan

- Launchpad, git.linaro.org and android.git.linaro.org
- public and private repos, SSH keys auth
- gitweb for web interface
- gerrit integration for code review purposes http://review.android.git.linaro.org
- future goals:
  - http://cards.linaro.org/browse/CARD-148
  - better ACL and privacy control
  - LDAP sync
  - Repo management over web (no SSH required)
  - Rhodecode at https://staging.git.linaro.org/

Jenkins & ci.linaro.org by Stevan

- continuous integration with Jenkins
- perform dynamic, parameterized builds on EC2 cloud
- enables you to build and test your code submissions
- gcc, kernel, openembedded, etc..
- submits build results for LAVA testing
- how to request CI job, monitor build status, test results and more can be found at https://wiki.linaro.org/Platform/Infrastructure/LinaroCI

Android Build by Paul

- Builds Android platform, submits for testing to LAVA, provides integrated display of results
- Jenkins-based system with custom frontend for scheduling builds and displaying results
- Frontend is not very scalable and maintainable, but works so far
- Most of issues with the system are due to Jenkins side of things
- Possible improvement is migration to general CI solution

cbuild by Paul

- System for toolchain testing (gcc and other parts like libc's)
- Primary stakeholder: Toolchain WG
- Mostly separate from the rest of CI infra (implementation predates general CI infra setup in Linaro)
- Does cross as well as native builds (native builds are what differentiates it from other CI infra)
- Uses own ARM board pool, or more recently, LAVA
- LAVA is used as "dumb executor", better integration is intended
- UI provides some advanced features (like diffing between test results)
- At the same time, UI is that of internal tool - not easy to learn or use

LAVA Scheduler by Antonio

- **Web frontend:**
  - XMLRPC API: Receives job submissions from users and stores them in the DB in Submitted state
  - also provides views and reports for the lava-server Django app: device status, job status and logs
- **Daemon:**
  - multiple worker nodes
  - polls the DB for jobs
  - decides which device will run the job
  - spawns dispatcher instance
- **Plans**
  - add support for scheduling multi-device jobs

LAVA Dispatcher by Antonio

- Responsible for running jobs
- Designed as a one-shot CLI application:
  - takes a job as input, runs the job in the given device, exits when finished
- Needs to be changed when a new device type has to be supported
  - usually just adding a new class
  - but not always so easy ;-)
- Plans
  - running multi-device jobs

lava-test-shell by Senthil

- Part of the LAVA dispatcher
- Being pushed as the gold standard for writing LAVA test suite (or jobs in general) writers
- black-box style
- tests are just shell commands run on the target
  - API to declare test case and results (pass/fail)
- Plans
  - enhance user experience of test suite writers
  - document best practices
  - probable integration with new test suite helper tool

lava-lab by Dave

- Configuration management repository for the LAVA lab
  - lp:lava-lab
- Uses salt
- New systems provisioned very quickly
- Manages, among others:
  - packages installation (APT and custom sources)
  - available users
  - service configuration (DHCP, DNS, etc)
- We are incrementally building it
  - new systems fully managed with salt
  - older systems partially managed
  - lava install, upgrades and config already managed
- SSH access from launchpad - move to ldap?

LAVA Dashboard by Antonio and Senthil

- Results submitted at the end of each job go into bundle streams
- web interface
  - Reports
  - Filters: one can be notified of every new test result that is uploaded and matches a user-defined criteria
  - Bundle streams: private or public, individual or per-team
- Plans
  - More intuitive test management UI
  - Ability to submit jobs from the Dashboard

Publishing framework by Georgy

- Runs snapshots.linaro.org & releases.linaro.org
- Holds build results for builds happening on jenkins instances
- Provides click-through licensing protection where needed
- Provides OpenID protection for builds restricted to sets of users
- BUILD-INFO.txt files: produced by builds
- Future goals:
  - Provide simple REST API for publishing and downloading
  - Move to "cheap" cloud-based storage like S3

linaro-image-tools by Milo

- Collection of command line tools
- Creates and supports hardware packs
- Supports Android hardware packs (it's a configuration file!)
- Creates and installs bootable systems
  - Just need an hardware pack, and a root filesystem
  - For Android images it combines several components
- Supports multiple boards
- Future Plans:
  - Clean-up documentation
  - Clean-up bug queue
  - Python3 port/fixes

Get in touch by Danilo

- Reach us all on #linaro, #linaro-infrastructure and #linaro-lava IRC channels
- Now you know us: grab us while at Connect in Atrium 2 (afternoons)
- Joint team, others with experience (Fathi, Marcin, ScottB...)

- Meet Tyler Baker

- Questions?

More about Linaro Connect: www.linaro.org/connect/
More about Linaro: www.linaro.org/about/
More about Linaro engineering: www.linaro.org/engineering/