Hacking and contributing to LAVA

Rémi Duraffort
LAVA Teach Lead
LAVA architecture
LAVA simple architecture

- Users
- LAVA logo
- dispatcher 1
- dispatcher 2
- dispatcher N
- server
- tftp&nfs server
- Power control
- Serial relay
LAVA roles

Server:

- Web UI and API
  - Submit jobs
  - Results, logs, ...
  - Access control
    - Users, groups
- Scheduling jobs
  - Priority
  - Private boards
  - Multinode jobs
- Store job logs
- Send notifications

Dispatcher:

- Deploy resources
- Power on/off DUTs
- Check boards health
- Parse logs
  - Kernel panic
  - Bootloader error
- Classify errors
  - Infrastructure, Job, Test, Bug, ...
LAVA services (systemd names)

**Server:**
- apache2
- lava-server-gunicorn
  - unicorn3 lava_server.wsgi
- postgresql
- lava-master
- lava-logs
- lava-publisher
- lava-coordinator

**Dispatcher:**
- lava-slave
  - lava-run(s)
- nfs-kernel-server (nfsd)
- tftpd
- ser2net
- lxc
- docker
- udev
  - udev rules for usb devices
lava-master & lava-slave

**lava-master:**
- Lab master
  - Identify slaves
    - ZMQ certificates
    - Check connectivity
- Schedule jobs
- START/CANCEL jobs
- Render device dictionaries
  - When starting a job
- Send notifications
  - For Running states

- lava-server manage lava-master
- lava_server/management/commands/lava-master.py

**lava-slave:**
- Connect/PING lava-master
  - Initiate the connection
- Waits for commands
  - START/CANCEL/STATUS
- Spawn lava-run
  - Check return value
- Report jobs status
  - Started, finished
  - Final job status

- lava-slave
- lava/dispatcher/lava-slave
lava-run

- Parse job definition
  - Build the pipeline of actions
    - deploy/boot/test
- Parse the device configuration
  - Connection command, power control, ...
- Setup the resources
  - tftp, libguestfs, ...
- Interact and control the DUT
  - Power control
  - Parse the serial output
  - Send commands (u-boot, fastboot, ...)
- Generate the job logs

- lava/dispatcher/lava-run
- lava_dispatcher python module
lava-logs

- Receive logs from **lava-run**
- Parse the logs
  - Create the results on the fly
- Store logs
  - `/var/lib/lava-server/default/media/job-output/<year>/<month>/<day>/<id>/output.yaml`
- Send notifications
  - For finished state

- `lava-server manage lava-logs`
- `lava_server/management/commands/lava-logs.py`
Hacking LAVA
Developer setup v1

- **Install debian packages**
  - apt-get install lava-server lava-dispatcher

- **Clone the source code**
  - git clone https://git.lavasoftware.org/lava/lava/

- **Replace with symlinks**
  - rm -rf /usr/lib/python3/dist-packages/lava_common
  - ln -s $(SRC)/lava_common /usr/lib/python3/dist-packages/lava_common
  - **Every python package**
    - lava_common, lava_dispatcher
    - lava_rest_app, lava_results_app, lava_scheduler_app, lava_server, linaro_django_xmlrpc
  - **The binaries**
    - /usr/bin/lava-server => $(SRC)/lava_server/manage.py
    - /usr/bin/lava-run   => $(SRC)/lava/dispatcher/lava-run
    - /usr/bin/lava-slave => $(SRC)/lava/dispatcher/lava-slave

- **Restart everything**
  - lava-logs, lava-master, lava-publisher, lava-server-gunicorn, lava-slave
Developer setup v1

- Run tests
  - ./ci-run

- Go to [http://localhost/](http://localhost/)

- When changing source code
  - Restart the corresponding services
    - lava-server source code:
      - Web UI code:
        - lava-server-gunicorn
    - Or
      - lava-logs, lava-master, lava-publisher, lava-slave
    - lava-dispatcher:
      - lava-slave only if changing lava-slave itself
      - Resubmitting will use the new source code
Developer setup v2

- Install basic tools
  - `git python3 python3-yaml`
  - `postgresql postgresql-client python3-django-extensions python3-werkzeug`
  - `black nfs-kernel-server`

- Clone sources
  - `git clone https://git.lavasoftware.org/lava/lava`

- Install lava-server and lava-dispatcher dependencies
  - `$(python3 share/requires.py -p lava-server -d debian -s buster -n)`
  - `$(python3 share/requires.py -p lava-server -d debian -s buster -n -u)`
  - `$(python3 share/requires.py -p lava-dispatcher -d debian -s buster -n)`
  - `$(python3 share/requires.py -p lava-dispatcher -d debian -s buster -n -u)`
Developer setup v2

● Create test database
  ○ `sudo -u postgres psql -c "CREATE ROLE devel NOSUPERUSER CREATEDB NOCREATEROLE INHERIT LOGIN ENCRYPTED PASSWORD 'devel'"`
  ○ `sudo -u postgres psql -c "CREATE DATABASE devel OWNER devel"`
  ○ `python3 lava_server/manage.py migrate`

● Run tests
  ○ `./ci-run`

● Start services using foreman
  ○ `apt-get install ruby-foreman`
  ○ `foreman start`

Integrating a new device-type

- Using **existing deploy and boot** methods
  - fastboot, nbd, tftp, ...
  - barebox, dfu, fastboot, pyocd, u-boot, ...

- Create a new device-type
  - etc/dispatcher-config/device-types/
  - Look at existing ones
  - Should extend **base.jinja2**

- Create a device dictionary
  - Test the rendering
  - lavacli utils templates render --path device-types/ devices/<device>.jinja2

- Run some tests from the web interface
Integrating a new device-type

- Using **new deploy and/or boot** methods
  - For advanced users
  - Not difficult ... but also not easy
- **Talk to us first!**
  - Maybe just a small change
  - Maybe already a WIP

- Create a new action class
  - `lava_dispatcher/actions/(deploy|boot)/<action>.py`
- Import the action in the strategies
  - `lava_dispatcher/actions/(deploy|boot)/strategies.py`
- Update the schema
  - `lava_common/schemas/(deploy|boot)/<action>.py`
- **Test!**
Contributing
Contributing

- GitLab instance at [https://git.lavasoftware.org/](https://git.lavasoftware.org/)
- Contributing
  - Create an account
  - Fork lava/lava
  - Push to a branch
  - Ask for DIND permissions
    - sast and docker tests will timeout
  - Create a MR
  - Fix CI issues

- Classical CI issues
  - Use black for code format
    - `apt-get install black`
    - `black <your code>`
  - SAST or docker tests timing out
    - Ask for DIND worker permissions
Contributing: testing LAVA

- Add unit tests whenever you can
- If you found a bug
  - Add a unit test!
  - Will fail until the issue is fixed

- Add boards/labs to lavafed
  - We can only tests on boards we can access
Questions?
Thank you

Join Linaro to accelerate deployment of your Arm-based solutions through collaboration

contactus@linaro.org