PATCHWORK CI

Philémon Gardet - Antoine Damhet

July 14, 2017

THE GOAL

PATCH REVIEW IS LONG AND PAINFUL

```
We want to automatically reject malformed, or not working
patches
```

CONSTRAINTS

- Simulate an open-source project with a mailing list
- It's a kernel course
- It must ease the review process
- It must be scalable

PIPELINE OVERVIEW

- Check the patch coding style
- Apply the patch on the Linux master
- Build the resulting kernel on multiple configurations
- Run student and reference tests

PROJECT DESCRIPTION

```yaml
name: Proctree Syscall
builds:
        - arch: x86_64
          config:
                        - config/proctree_syscall
                        - config/virtio_net
                        - config/gcov
                        - config/overlayfs
          lava: True
[...]
        - arch: powerpc64
          config:
                        - config/proctree_syscall

tests:
        path: epita/proctree.yaml
```

PATCHWORK

* The freedesktop.org edition
* Construct series and handle revisions
* Manage patch state (New, under review, merged, etc.)

1. `git-pw`
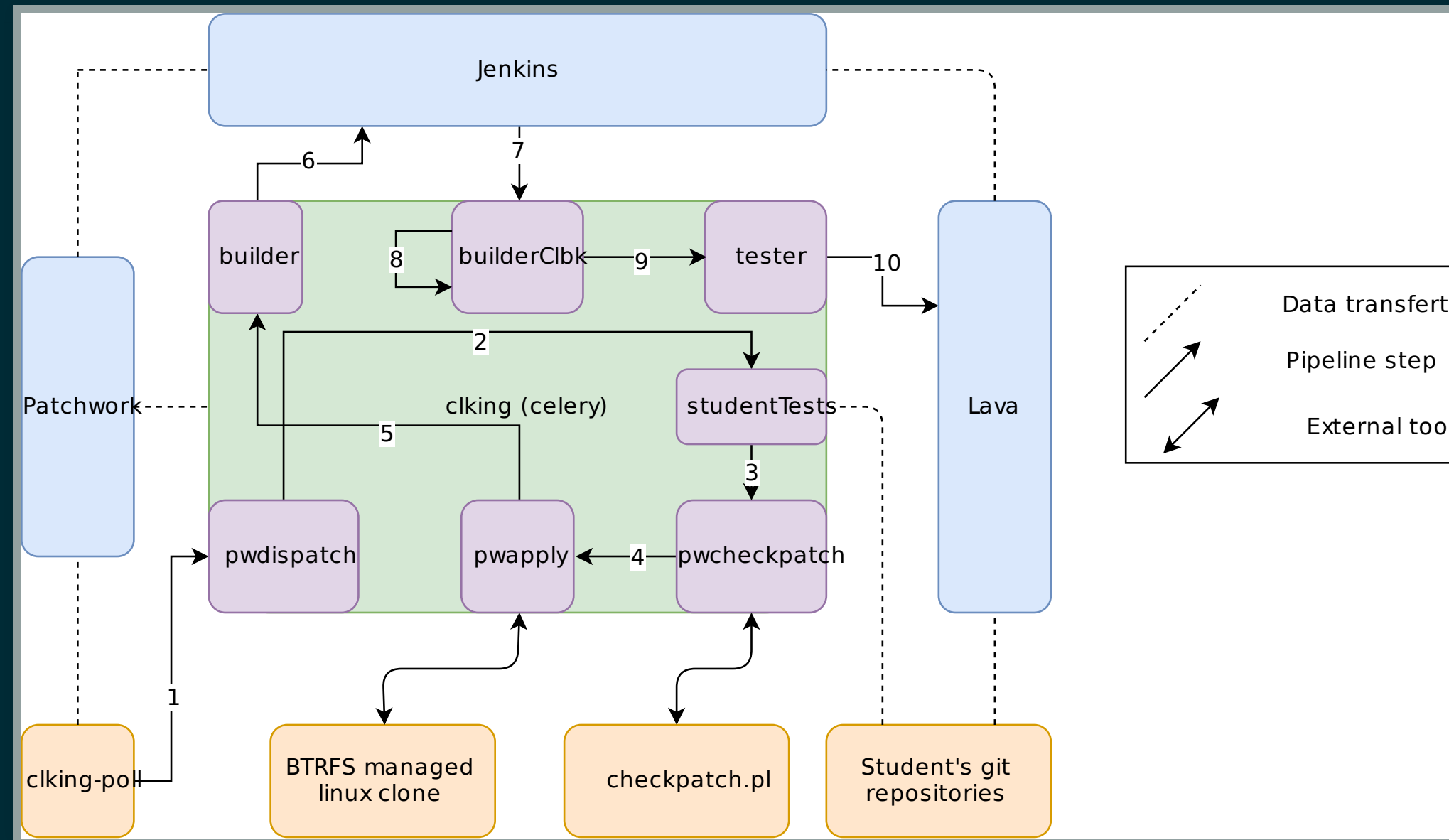2. `pwclient`
3. JSON API

MATCH A MAIL WITH A STUDENT

- Use the `Signed-off-by:`
- Falsehoods Programmers Believe About Names

CELERY

- Redis backend
- Asynchronous
- Flower
- Multiple workers

PIPELINE

A submission is characterized by a couple (serieID, revID)
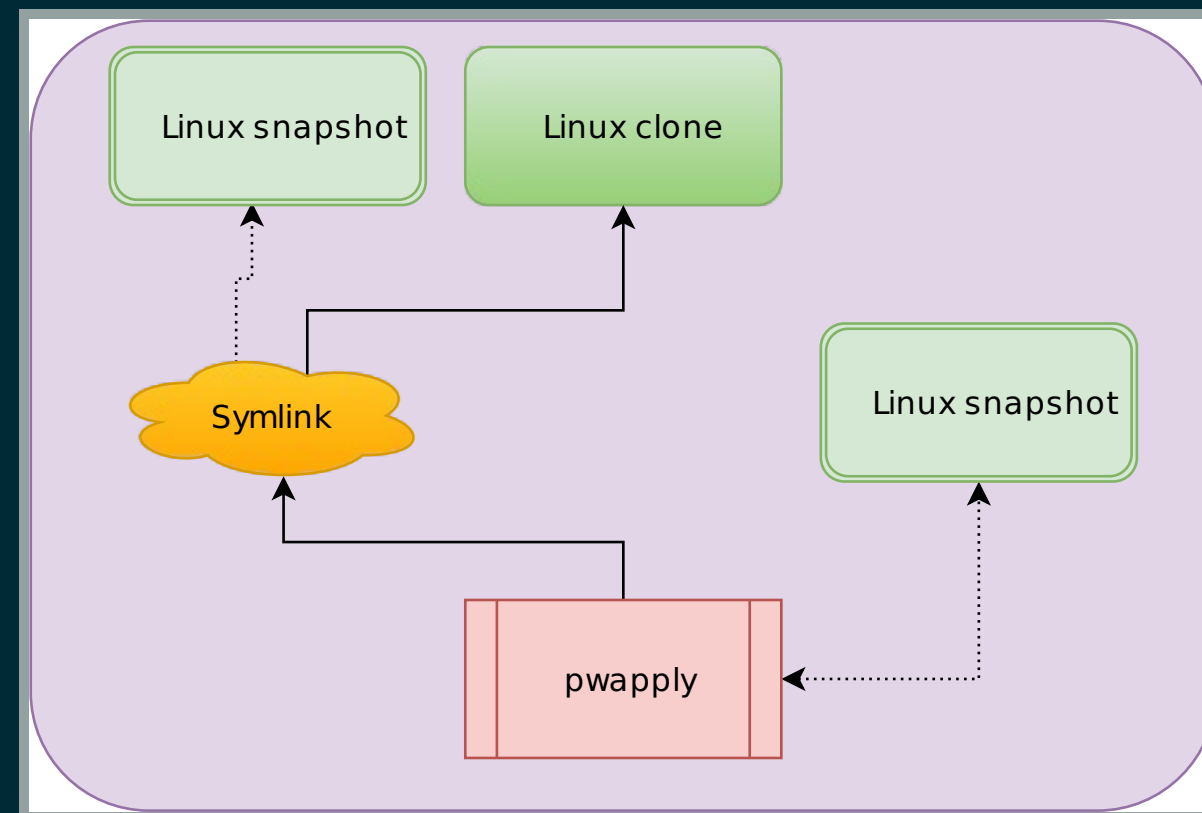
EVENT HANDLING

- series-new-revision
- patch-state-change

CHECKPATCH

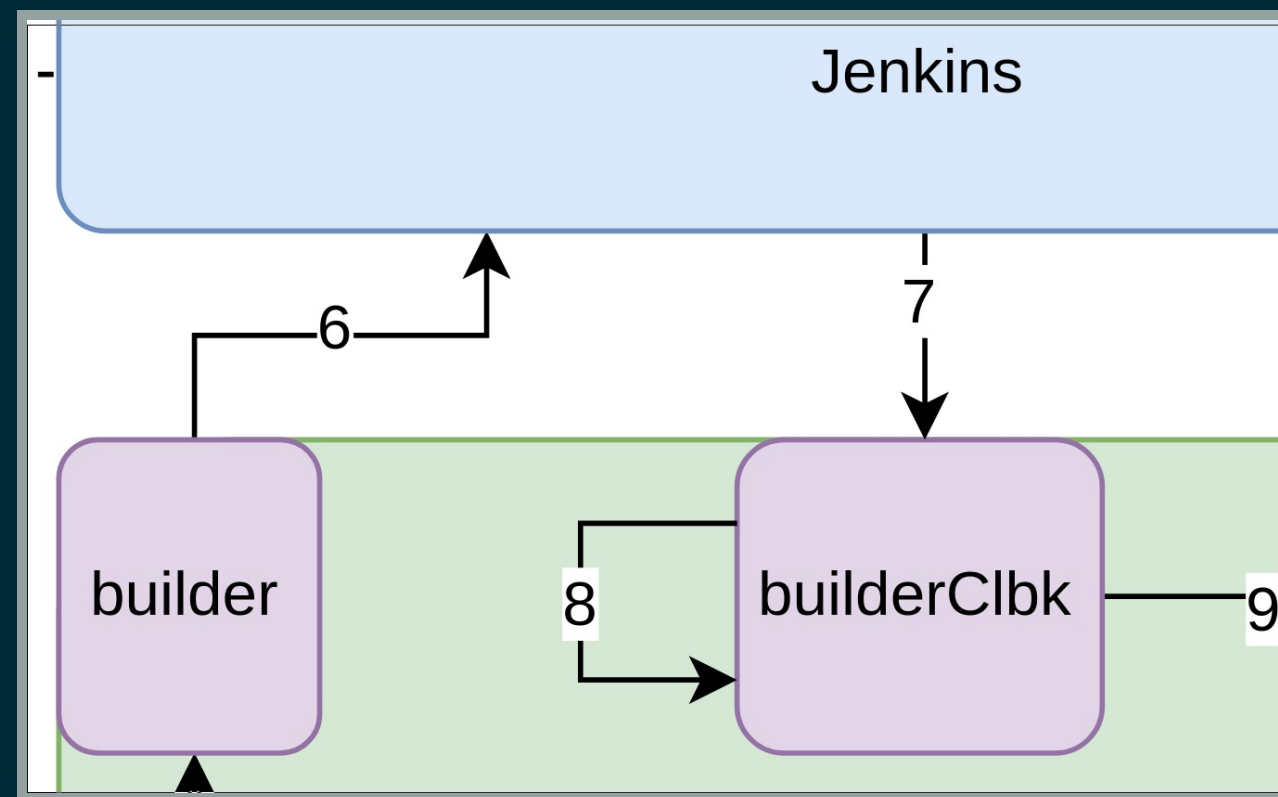Patchwork gives us a patch with DOS newlines...

APPLY ON LINUX MASTER

- btrfs submodules
- routinely update the git
- git-pw

BUILD THE CODE

- Launch every builds at once
- Maintain a refcount to know when it's over
- Maintain an artefact list for lava

JENKINS - KERNEL BUILDING

OBJECTIVES

- Test kernel students compilation
- Build one different architectures
  (i386, x86_64, arm, arm64, powerpc64...)
- Custom students feature enabled by Kconfig flags
- Feeds tests with compiled kernel and modules

INFRASTRUCTURE

- Jenkins
- Jenkins Docker slaves (Debian)
- makecross script (from lkp-tests project)
- toolschains from kernel.org

PROCESS

- Update a local kernel repository
- Get student branch
- Build kernel without / with features

OUTPUTS

- kernel binary
- Modules and Linux headers in a ext2 partition image

PROJECT DESCRIPTION

```yaml
name: Proctree Syscall
builds:
        - arch: x86_64
          config:
                        - config/proctree_syscall
                        - config/virtio_net
                        - config/gcov
                        - config/overlayfs
          lava: True
[...]
        - arch: powerpc64
          config:
                        - config/proctree_syscall

tests:
        path: epita/proctree.yaml
```
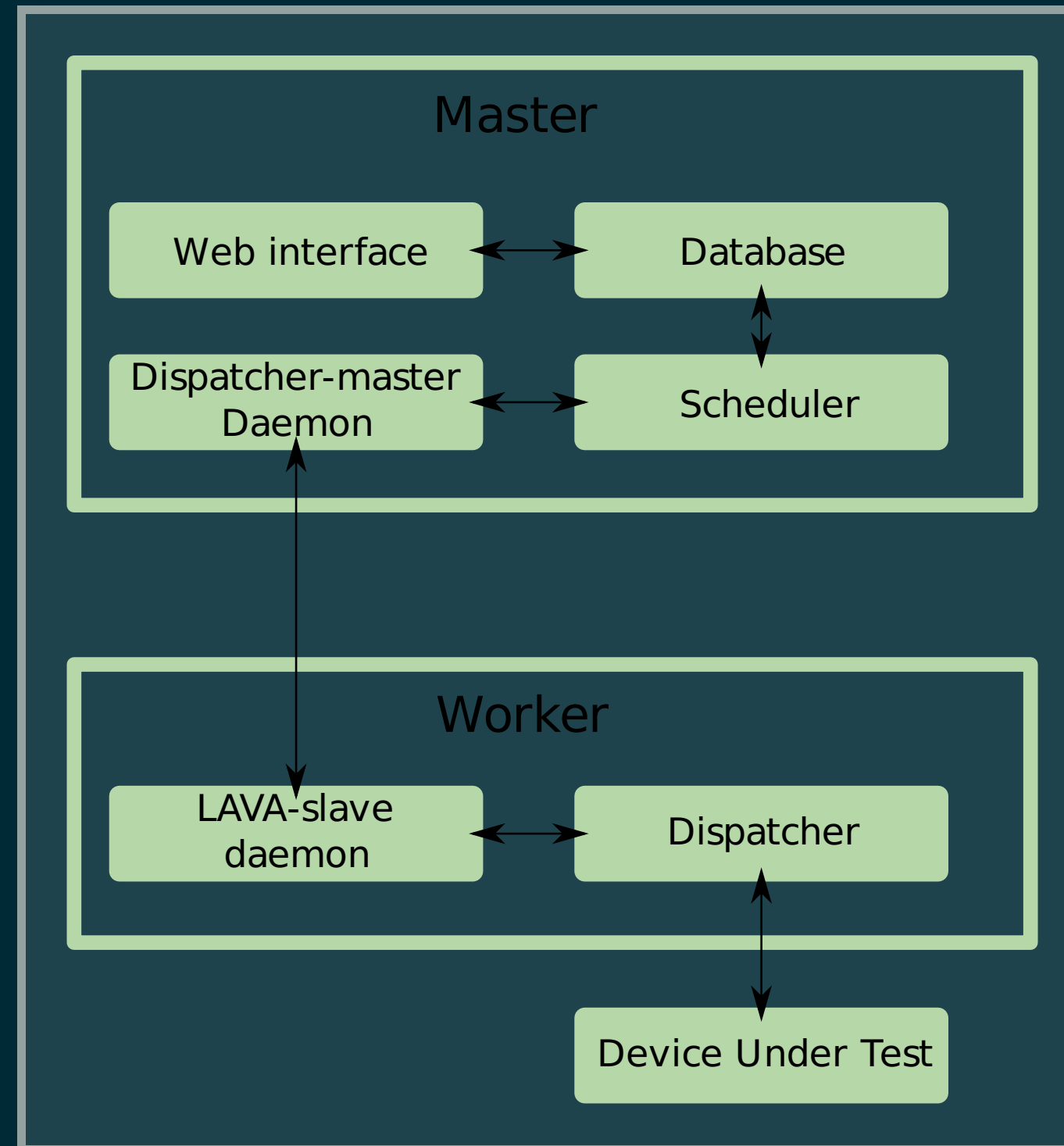
BACK TO CLKING

- Build ID
- opaque data
  - serie ID
  - revision ID
  - target ID

LAVA - TEST KERNELS

LAVA PROJECT

- Project from Linaro team
- Test kernel / images on embeded systems

LAVA STRUCTURE

Master

Web interface ↔ Database

Dispatcher-master Daemon ↔ Scheduler

Worker

LAVA-slave daemon ↔ Dispatcher

Device Under Test

OUR INFRASTRUCTURE

- A lava master
- One lava worker by cluster stations
- Worker spawns QEMU virtual machines (virtualization nested)

TEST IMAGE

- A Debian base image
  (with all needed tools like gcc, python...)
- Kernel student used with QEMU
- Ext2 partion with student modules / headers
  mounted as overlay

TEST DESCRIPTION - YAML

```yaml
metadata:
    format: Lava-Test Test Definition 1.0
    name: smoke-tests-dummy
    description: "Smoke tests everyday"

install:
        deps:
          - gcc
          - gcc-multilib
run:
    steps:
      - ./epita/script_dummy.sh
```

TEST SCRIPTS BASH

```
lava-test-case [case-name] --result {pass,fail}
```

OUTPUTS

- Logs
- Test results in csv files

DIFFICULTIES

- Lava doc not very usefull
- The real doc is the source code
- Find bugs in lava with students work
- Where fix: lava master / worker ?

RESPONSE TO THE STUDENT

LSE BOT - TEMPLATES

- Sign off missing
- Invalid or missing test description
- Wrong coding style
- Apply failed
- Build failed
- Test results

EXAMPLE

```
Hi John Smith,

Project: HTS Syscall
Patch: [v9] syscall: add dummy system call
Tests suite: dummy

You have passed 7 tests on 7.

Perfect! Your work will be examined.

If you think this is a false positive,
please reply-all to this message.

Sincerely,
LSE Team
```

WHAT'S NEXT

- Fix or replace `patchwork`
- Make a working version (or use `KernelCI`)
- Scale

- Questions ?

LSE