Landslide: Systematic Testing in 15-410
User Guide

Ben Blum

1 Preface

Please make sure you have read and understand the lecture slides on systematic testing, which are available on the 15-410 course website. If you have any questions about the research background I would be happy to answer them; you can get in touch at `bblum@cs.cmu.edu`.

2 Instructions

We encourage you to use Landslide on the GHC cluster or UNIX pool machines (`ghcXX.ghc.andrew.cmu.edu` or `unix.andrew.cmu.edu`). Since Landslide uses the open-source simulator Bochs rather than the proprietary Simics, you can also try it on any personal machine which runs Linux. (The Landslide repository includes the Bochs source; you do not need to install it separately.)

1. Clone the repository - `git clone https://github.com/bblum/landslide.git` - this will be your workspace.
2. Run `./p2-setup.sh PATH`, where `PATH` is the absolute path to your p2 implementation.
   - This command will import your p2 into `./pebsim/p2-basecode/` and then attempt to build it. Whenever you update your p2 code, please do so in its original `PATH` and then run `p2-setup.sh` again.
   - `PATH` should be the path with `410kern`, `410user`, `user/libthread`, etc as subdirectories, not the `user/libthread` directory itself.
   - This step may take a few minutes, as it needs to compile the Bochs simulator from source.
3. Run `./landslide OPTIONS` to run tests with Landslide, for various values of `OPTIONS`:
   (a) Use ./landslide -h to view command line options. The most important ones are -t , max testing time, and -p , which test program to run.
   (b) Suggested test configurations to run are as follows. You can, of course, change the time limits or test programs as you see fit.
      - `./landslide -t 30m -p thr_exit_join`
      - `./landslide -t 30m -p mutex_test`
      - `./landslide -t 30m -p paradise_lost`
      - `./landslide -t 30m -p broadcast_test`
      - `./landslide -t 30m -p paraguay`
      - `./landslide -t 30m -p rwlock_downgrade_read_test`

      - `./landslide -t 12h -p thr_exit_join`

- ./landslide -t 12h -p mutex_test
- ./landslide -t 12h -p paradise_lost
- ./landslide -t 12h -p broadcast_test
- ./landslide -t 12h -p paraguay
- ./landslide -t 12h -p rwlock_downgrade_read_test

Please check slide 34 of the lecture before attempting to run cyclone, racer, largetest, etc.

For the longer-running configurations, we recommend running them inside of a screen or tmux session. The GHC and UNIX cluster machines reboot nightly, so using a personal machine is recommended for running tests overnight.

(c) Landslide's bug reports will show up as HTML files.

- If you are using a cluster machine in person, or a personal machine, you can view them by pointing a browser to
file:///PATH/TO/landslide/FILENAME.html
- For remote usage, move the html file to /www in your AFS, and view it at
http://www.contrib.andrew.cmu.edu/ YOUR_ANDREWID/FILENAME.html. Note the www in the URL is mandatory; note also that this will allow the world to see it, so be careful to clean up any traces in /www when you're done!

(d) To build your own custom test case for use with Landslide, put it in pebsim/p2-basecode/user/progs and edit the STUDENTTESTS line in pebsim/p2-basecode/config-incomplete.mk (note: NOT config.mk; the setup script auto-generates that file). Then run ./p2-setup.sh again.

4. After fixing any found bugs, re-run Landslide with the same test options to confirm that your fix works.

Note on simultaneous Landslides: Landslide does not support running multiple tests at once; it needs to create some temporary files to annotate your code, and multiple instances of Landslide can step on each other's toes by clobbering those files. If you want to try multiple tests at once, simply git clone the repository again in a different directory, and run one instance of Landslide in each.

3 Troubleshooting

If you find a Landslide bug, such as an assertion/crash, or such as a bug report that you think is actually a bug in Landslide rather than a bug in your P2, or such as it getting stuck and not exiting in the given time limit, please create a tarball of your workspace, just as you would to report a reference kernel bug, but send it to Ben (bblum@cs.cmu.edu) instead of 410 staff. For any other technical issues getting it to work, or if you have difficulty understanding a bug report, send email to Ben.

This is the first semester we have released Landslide with Bochs instead of Simics, so there may be any manner of unexpected bugs (it's fairly well tested on my own laptop, though!). You may need to install some build dependencies including mtools, libxrandr-dev, libncurses5-dev, and libc6-dev. If you are trying Landslide on a personal machine and have problems with the setup step, please try again on one of the official CMU servers before sending me a bug report. If all else fails, you may try the known-good Simics version: https://github.com/bblum/landslide-simics (which requires a GHC or UNIX cluster machine for the software license).

Finally, we cannot provide help on how to understand why your bug happened or how to fix it. The point of the research is that Landslide's automation plus your own brain should be enough! On the other hand, if you have a design question while deciding between potential ways to fix a bug, send it to 410 course staff as usual.