

P51(bis): High Performance Networked-Systems

Prof. Andrew W. Moore

Lecture 3

A **huge** thank you to Eben Upton, Raspberry Pi Foundation, and PiHut people for enabling this incarnation of the module at incredibly short notice.

Reproducible Experiments

Reproducibility vs Repeatability

- **Repeatability** measures the variation in measurements taken by a single instrument or person under the same conditions
- **Reproducibility** measures whether an entire study or experiment can be reproduced in its entirety.

Why?

- Establish variance – **Repeatability**
- Establish reliability – **Repeatability**

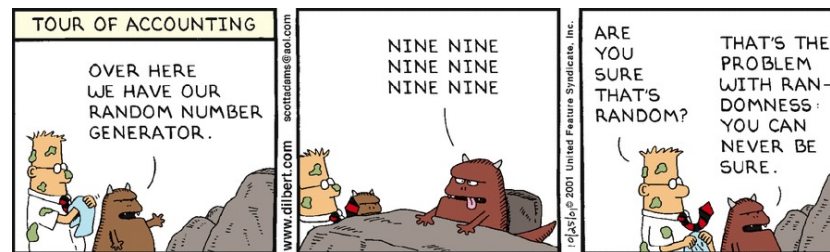
- Evaluate a new method – **Reproducibility**
- Eval. a new environment – **Reproducibility**
- Evaluate a new approach – **Reproducibility**

Variables and Constants

- *Why?* will tell us what we want to *vary*
and
- *Why?* what we need to hold constant

Random-ish

- Rarely do we want true-random
- Typically we want pseudo-random
- Often we want to specify the seed(s)



Method and Environment

- Simulation?
- Emulation?
- Implementation driven evaluation?
- Deployment?
- Partial emulation?
- Partial implementation driven evaluation?

Software tools: Scripts, Make, etc

- We have some quite useful repeatability tools:
 - Make (links dependencies)
- Scripts – sort of documents what you actually need to do to get from (A) to (B)
- So please use them.

Machines (/Hardware)

- Memory? CPU? Disk type and config?
- Hyperthreading and temperature controls?
- Which slots were stuff in?
- Switch config? Switch hardware? Which *actual* Switch?
- Which transceivers? NICs? cables?

- Tell me again which disk did you dump data to?
- (Oh did you mention the periodic process that moved the data from your machine to another machine so the local disk didn't overrun....)

Workloads

- Why is this workload the right one?
 - Stress testing?
- Did you use the workload-generator correctly?
- Record everything from command line options to software and library versions.

Benchmarks

- Often well equipped to run with good reproducibility
- Often not representative of what you want
- Benchmarks might exercise, but just like in fitness training: exercising is not *competing*

Logged data

- Lets talk about time....
 - No god clock
 - Many representations
 - TimeZone is fun
 - UNIX time is fine, sometimes...
- Text records are nice (for humans)
- Binary records are nice (for programs)

So what is meta-data?

The other stuff needed to repeat precisely the same experiment

- Make and Model (and firmware and config)
- DNS (at least the entries for your systems)
- Bootp/dhcp/activedirectory - all state

Documentation

- What is the goal of the experiment?
- How to set it up?
- What are all the dependencies? And versions?
- Are special licenses required?
- What is the command line to run?
- What was the script used in the experiment?
- Can you script the process?

Other Useful Practices

- Snapshot of the code base of the executable we used
 - If the code was change during the experiment – match code to results!
- Photo of the setup
- File headers, comments, README files, ...

Try stuff!
(don't be hipster Flanders)



Other peoples work

To reproduce other peoples work

You must get inside other peoples heads
(so consider their motivations)

Very few *high-bars* in reproducibility, here is one I co-authored earlier...

<http://www.cl.cam.ac.uk/research/srg/netos/qjump/repro.html>