Automated NUL regressions and performance testing

Michal Sojka

December 15, 2011

Outline I

Motivation

- I didn't want to break things when rewriting disk service.
- It was hard to reproduce other people's configurations (menu.lst@os)
- To get familiar with NUL I wrote several test programs to see how things work.
- Rarely used features tend to break after some time.

What It Can Do

1. `scons test` – run some tests in qemu before commit/push
2. Automated nightly testing
   - http://os.inf.tu-dresden.de/šojka/nul/test-report.html
3. Performance graphs
   - http://os.inf.tu-dresden.de/šojka/nul/performance.html

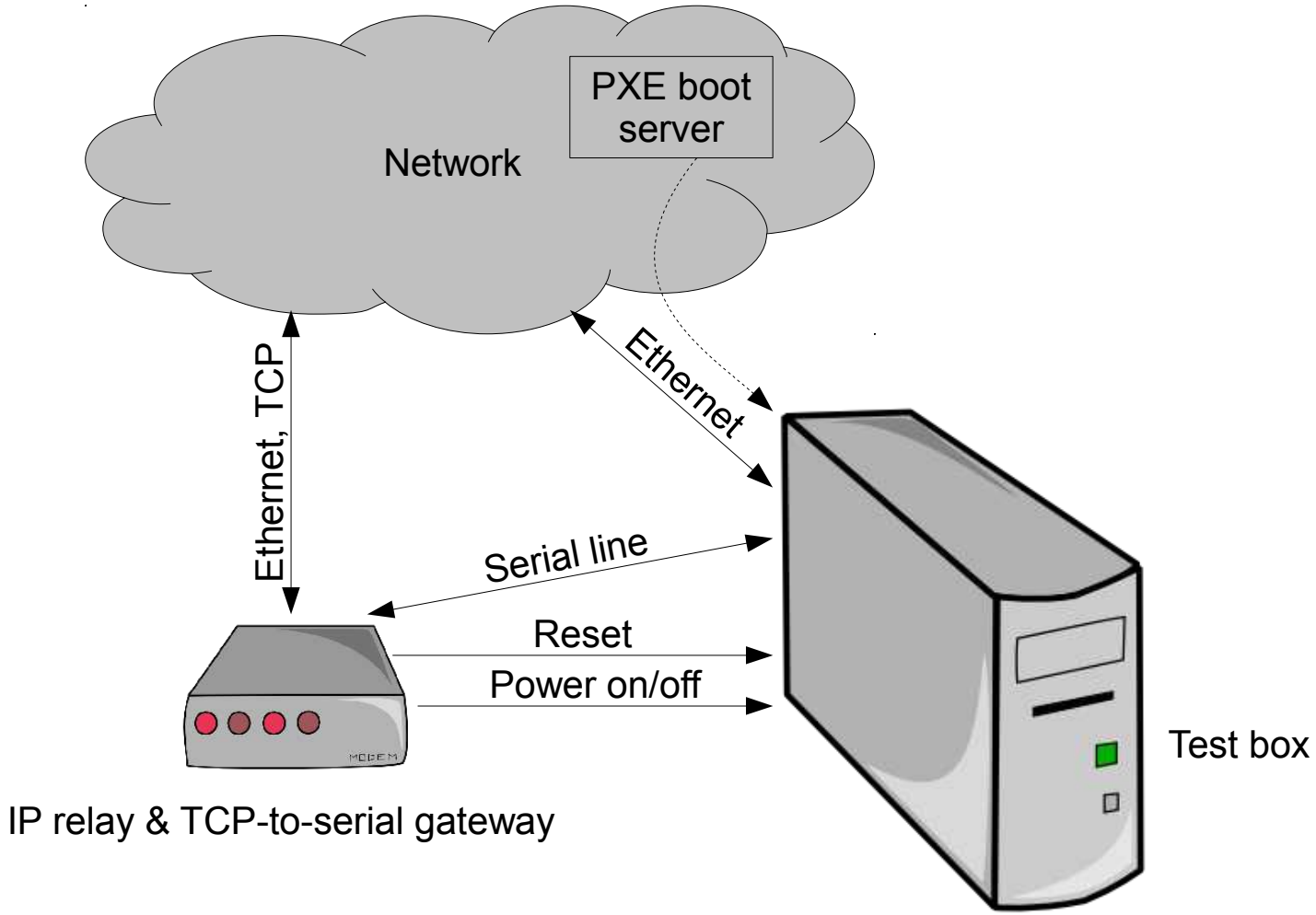
scons test

```
WVTEST_SKIP_TAGS="slow broken_in_qemu needs_net" /home/wsh/devel/nul/michal/wvtest/wvtestrun /home/wsh
l/nul/michal/wvtest/runall --build-dir=.
Testing "all" in /home/wsh/devel/nul/michal/wvtest/runall --build-dir=.:
! /home/wsh/devel/nul/michal/wvtest/sizes.wv  Sizes of binaries: ....... 0.110s ............ ok
! /home/wsh/devel/nul/michal/wvtest/wvtesttest.wv  Tests WvTest framework itself: ........ 2.395s ....
................................................................................................ ok
! /home/wsh/devel/nul/alexb/apps/ipc_test/ipctest.wv  all: .......... 2.378s ............... ok
! /home/wsh/devel/nul/julian/apps/per-cpu-service/per-cpu-service.wv  all: .. 2.392s ....... ok
! /home/wsh/devel/nul/michal/apps/echo/echo2sstest.wv  Echo service based on SSession: ....... 16.011s
................................................................................................ ok
! /home/wsh/devel/nul/michal/apps/echo/echo2test.wv  Echo service with sessions: ....... 2.519s ......
................................................................................................ ok
! /home/wsh/devel/nul/michal/apps/echo/echotest.wv  Simple echo service: ... 2.427s ........ ok
! /home/wsh/devel/nul/michal/apps/logdisk/part.wv  Partition table parsing: .... 2.540s .... ok
! /home/wsh/devel/nul/michal/boot/diskbench-ramdisk.wv  all: ..... 4.413s ................... ok
! /home/wsh/devel/nul/michal/boot/diskbench-ramdisk-old.wv  all: ..... 4.410s ............... ok
! /home/wsh/devel/nul/michal/boot/vancouver-linux-basic.wv  all: .. 17.466s ................. ok
! /home/wsh/devel/nul/michal/boot/vancouver-linux-boot-time.wv  all: .... 16.381s .......... ok
! /home/wsh/devel/nul/michal/boot/diskbench-vm.wv  all: ....... 12.014s ..................... ok
! /home/wsh/devel/nul/michal/boot/vancouver-boot-from-disk.wv  all: .. 24.300s ............. ok

WvTest: 73 tests, 0 failures, total time 110.416s.

WvTest result code: 0
scons: done building targets.
wsh@steelpick:devel/nul/build[master]%                                    git:/home/wsh/devel
```

Testbed Configuration



Writing a Test

1. Write a test application that generates the right output
2. Specify how to run (boot) the application

Step 1: WvTest

- The dumbest test framework that can even work
- Text based protocol:
  1. `Testing "<something>" in <somewhere>:`
  2. `! <what was tested> <result>`
     Result is either **ok** (success) or an identificationf of failure (typically just **FAILED**).

Step 1: WvTest

- The dumbest test framework that can even work
- Text based protocol:
  1. `Testing "<something>" in <somewhere>:`
  2. `! <what was tested> <result>`

     Result is either **ok** (success) or an identificationf of failure (typically just **FAILED**).
- Other alternatives: Test anything protocol (TAP), ...

Step 1: WvTest

- The dumbest test framework that can even work
- Text based protocol:
  1. `Testing "<something>" in <somewhere>:`
  2. `! <what was tested> <result>`
     Result is either **ok** (success) or an identificationf of failure (typically just **FAILED**).
- Other alternatives: Test anything protocol (TAP), ...

Example (WvTest Output)

```
Testing "Some WvTest Examples" in presentation.pdf:
! michal/wvtest/wvtesttest.cc:29 1 == 1 ok
Arbitrary garbage...
! who_am_i.cc:30 my_name == "you"  FAILED
! tests/timer.cc:38 service->timer(*utcb, msg) EPERM
```

Step 1: WvTest API

- `printf("! this works ok\n")`

Step 1: WvTest API

- `printf("! this works ok\n")`

Macros that make it easier to generate useful messages

- `WVPASS(1==1);`

```
! michal/wvtest/wvtesttest.cc:29 1==1 ok
```

Step 1: WvTest API

- `printf("! this works ok\n")`

Macros that make it easier to generate useful messages

- `WVPASS(1==1);`

```
! michal/wvtest/wvtesttest.cc:29 1==1 ok
```

- `WVPASSEQ(strlen(desc), 42)`

```
wvtest failed comparison: 43 == 42
! test.cc:123 strlen(desc) == 42 FAILED
```

Step 1: WvTest API

- `printf("! this works ok\n")`

Macros that make it easier to generate useful messages

- `WVPASS(1==1);`

```
! michal/wvtest/wvtesttest.cc:29 1==1 ok
```

- `WVPASSEQ(strlen(desc), 42)`

```
wvtest failed comparison: 43 == 42
! test.cc:123 strlen(desc) == 42 FAILED
```

- `WVNOVA(nova_create_sm(cap_base))`

```
! test.cc:25 nova_create_sm(cap_base) ECAP
```

Step 2: Novaboot

- A program that interprets *novaboot scripts*.
- Novaboot script is a text file that describes all configuration needed for booting the NOVA system.
- novaboot can:
  - Boot NOVA in qemu
  - Create bootloader configuration file (e.g. GRUB's menu.lst)
  - Generate configuration files on the fly (shell heredoc syntax)
  - Copy all needed files to TFTP server (e.g. to os:boot/)
  - Create bootable ISO images
  - Communicate with IP relay to reboot the test machine
  - Run `dhcpd` and `tftpd` on your box to PXE-boot the machine connected to it via Ethernet cable
- If possible, the serial output of the booted system is passed to stdout.

Step 2: Novaboot Examples

Example (Hello world novaboot script)

```
#!/usr/bin/env novaboot
bin/apps/sigma0.nul S0_DEFAULT hostserial hostvga script_start:1
bin/apps/hello.nul
hello.nulconfig <<EOF
  sigma0::mem:16 name::/s0/fs/rom name::/s0/admission ||
  rom://bin/apps/hello.nul
EOF
```

Step 2: Novaboot Examples

Example (Hello world novaboot script)

```
#!/usr/bin/env novaboot
bin/apps/sigma0.nul S0_DEFAULT hostserial hostvga script_start:1
bin/apps/hello.nul
hello.nulconfig <<EOF
  sigma0::mem:16 name::/s0/fs/rom name::/s0/admission ||
  rom://bin/apps/hello.nul
EOF
```

- Running the novaboot script

  `./hello` run the configuration in QEMU
  `./hello -server` copy all needed files to TFTP server
  `./hello -server -iprelay` after copying files, reset the test box and receive its serial output

WvTest Protocol Extensions for NOVA

Prefixes

- Sigma0 puts console number in front of the application output
- Vancouver adds “# “

```
(5) #    ! init.cc:20 Linux booted  ok
```

WvTest Protocol Extensions for NOVA

Prefixes

- Sigma0 puts console number in front of the application output
- Vancouver adds "# "

```
(5) #    ! init.cc:20 Linux booted  ok
```

Additional information for performance graphs

- ! init.cc:35 PERF: uptime 0.41 s ok
- Testing "⟨date⟩ ⟨time⟩, commit: ⟨git-describe⟩" in nul-nightly.sh:

wvnulrun script

- Usage: `wvnulrun` ⟨command⟩
- Forks/execs the command and processes its output
- Detects timeouts
  - Tests can specify custom timeouts by saying: `wvtest: timeout <seconds>`
- End of test detection
  - Pattern matching: “wvtest: done”, “resetting machine via method”, . . .
- Exit status reflects the WvTest reported failures
  - Use in git bisect etc.
  - `while wvnulrun ./race_condition -I; do :; done`

wvtestrun script

- ASCII beautification of WvTest output
- Collapses the output of a passed test to a single line
- Failed tests show all output before failure
- One dot per test
- Produces valid WvTest output

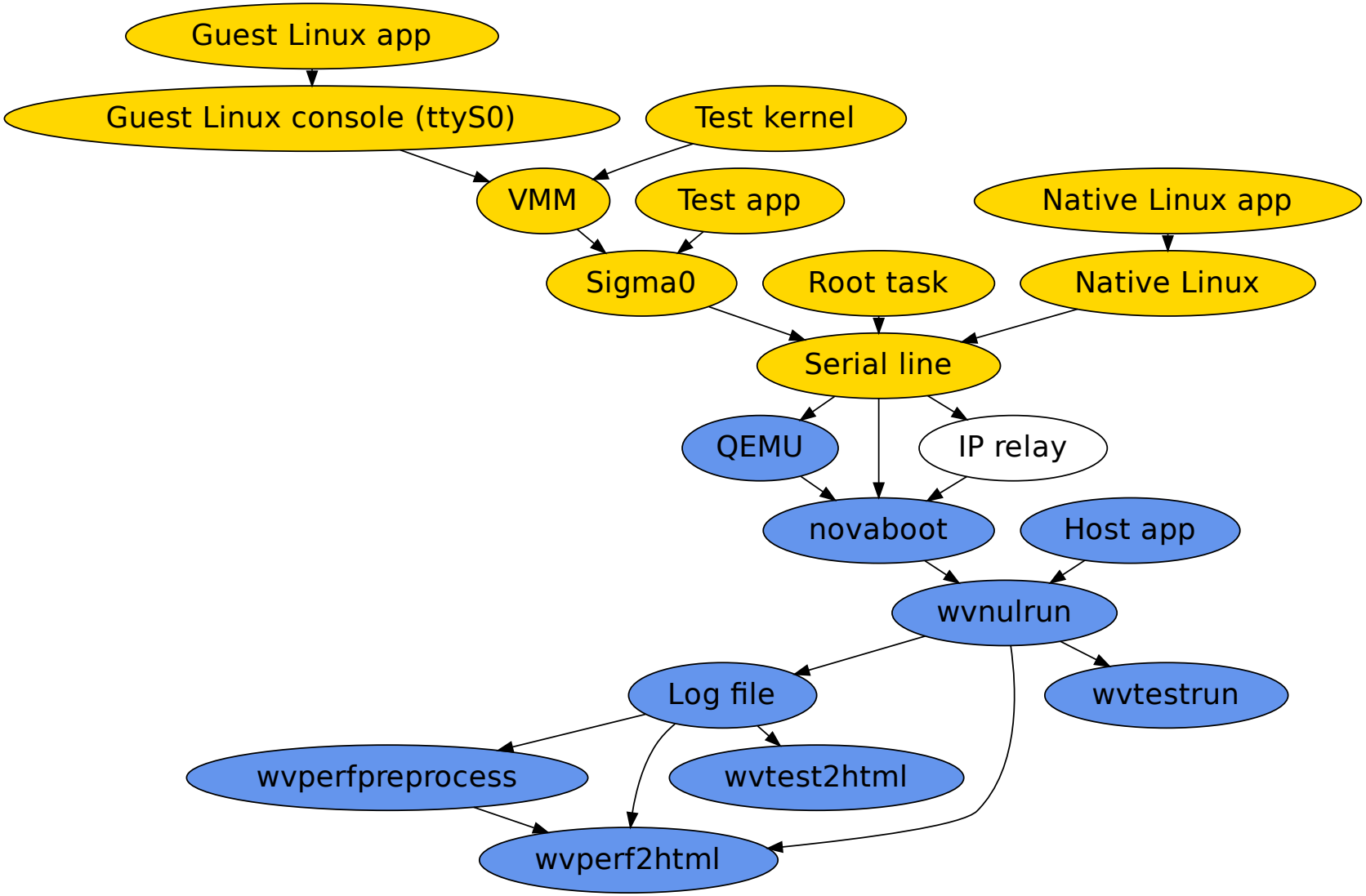
```
WVTEST_SKIP_TAGS="slow broken_in_qemu needs_net" /home/wsh/devel/nul/michal/wvtest/wvtestrun /home/wsh/deve
l/nul/michal/wvtest/runall --build-dir=.
Testing "all" in /home/wsh/devel/nul/michal/wvtest/runall --build-dir=.:
! /home/wsh/devel/nul/michal/wvtest/sizes.wv  Sizes of binaries: ....... 0.110s ............ ok
! /home/wsh/devel/nul/michal/wvtest/wvtesttest.wv  Tests WvTest framework itself: ........ 2.395s .........
.......................................................................... ok
! /home/wsh/devel/nul/alexb/apps/ipc_test/ipctest.wv  all: .......... 2.378s ............... ok
! /home/wsh/devel/nul/julian/apps/per-cpu-service/per-cpu-service.wv  all: .. 2.392s ....... ok
! /home/wsh/devel/nul/michal/apps/echo/echo2sstest.wv  Echo service based on SSession: ....... 16.011s ....
.......................................................................... ok
! /home/wsh/devel/nul/michal/apps/echo/echo2test.wv  Echo service with sessions: ....... 2.519s ..........
.......................................................................... ok
! /home/wsh/devel/nul/michal/apps/echo/echotest.wv  Simple echo service: ... 2.427s ........ ok
! /home/wsh/devel/nul/michal/apps/logdisk/part.wv  Partition table parsing: .... 2.540s .... ok
! /home/wsh/devel/nul/michal/boot/diskbench-ramdisk.wv  all: ..... 4.413s ................. ok
! /home/wsh/devel/nul/michal/boot/diskbench-ramdisk-old.wv  all: ..... 4.410s ............. ok
! /home/wsh/devel/nul/michal/boot/vancouver-linux-basic.wv  all: .. 17.466s ............... ok
! /home/wsh/devel/nul/michal/boot/vancouver-linux-boot-time.wv  all: .... 16.381s .......... ok
! /home/wsh/devel/nul/michal/boot/diskbench-vm.wv  all: ....... 12.014s ................... ok
! /home/wsh/devel/nul/michal/boot/vancouver-boot-from-disk.wv  all: .. 24.300s ............. ok

WvTest: 73 tests, 0 failures, total time 110.416s.
```

Other Scripts

- **listtests** – outputs all tests (`*.wv`) found in the repository in a defined oreder
- **runall** – runs all tests returned by listtests
- **wvperf2html** – generates HTML with performance graphs from logs
- **wvperfpreprocess** – combine multiple tests to a single graph, etc.
- **wvtest2html** – convert the log to HTML

WvTest Protocol Dataflow



Testing the Box From Outside

Example (Test for libvirt)

```
#!/bin/bash

. ./wvtest.sh

WVSTART libvirt

WVTEST_EXIT_PATTERN="NOVA management daemon is up." \
  wvnulrun ./passive "$@" | tee log
NOVA_IP=$(sed -ne '/.*got ip=\([^ ]*\).*/ s//\1/p' log)

# Tests for libvirt commands
WVPASS virsh -c nova+tls://$NOVA_IP:9999 nodeinfo
```

Test Scripting

- for-each-commit 5296754..ff8af37 wvnulrun ./some-test.wv -l | tee log | wvperf2html > graph.html

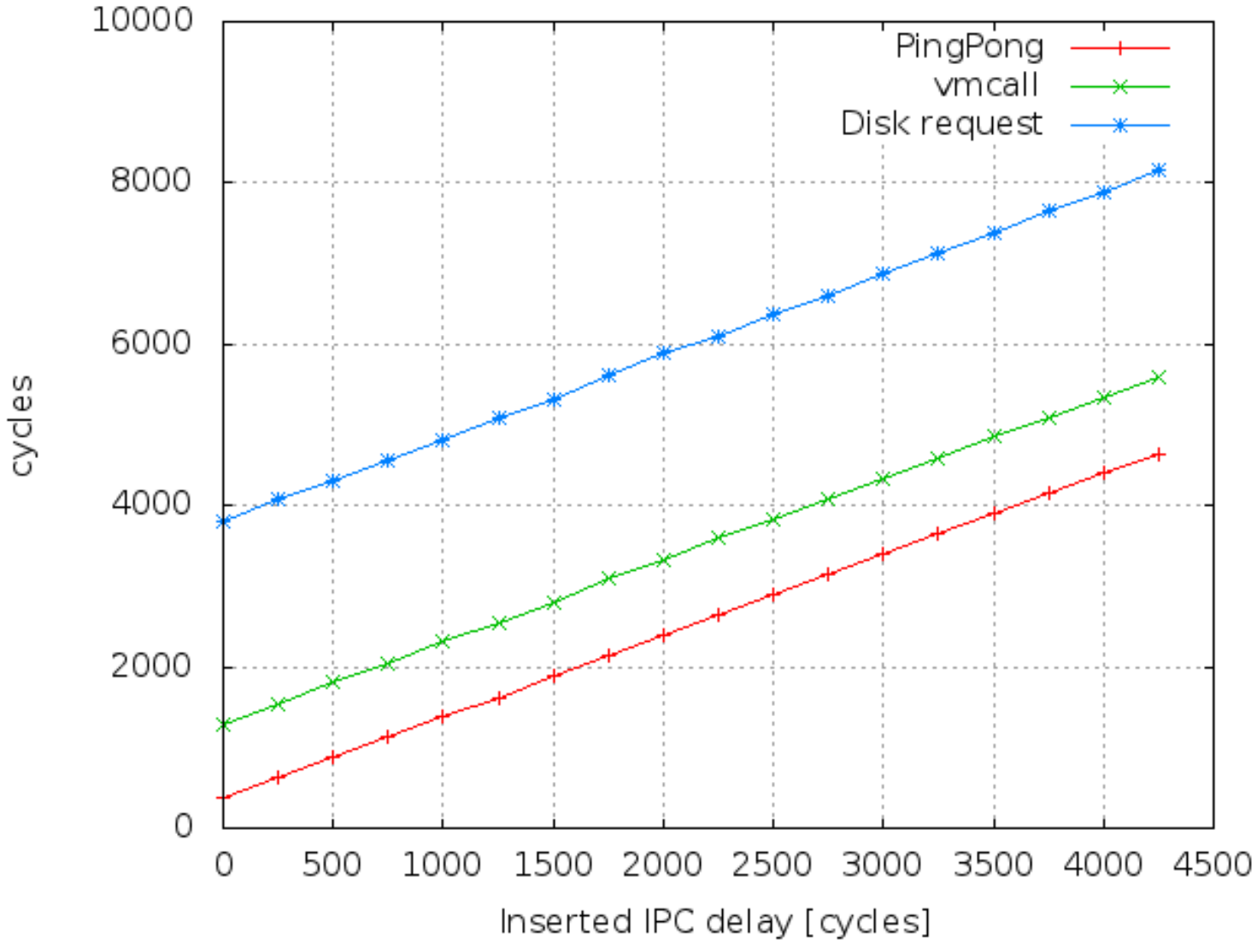
Test Scripting

- for-each-commit 5296754..ff8af37 wvnulrun ./some-test.wv -I | tee log | wvperf2html > graph.html

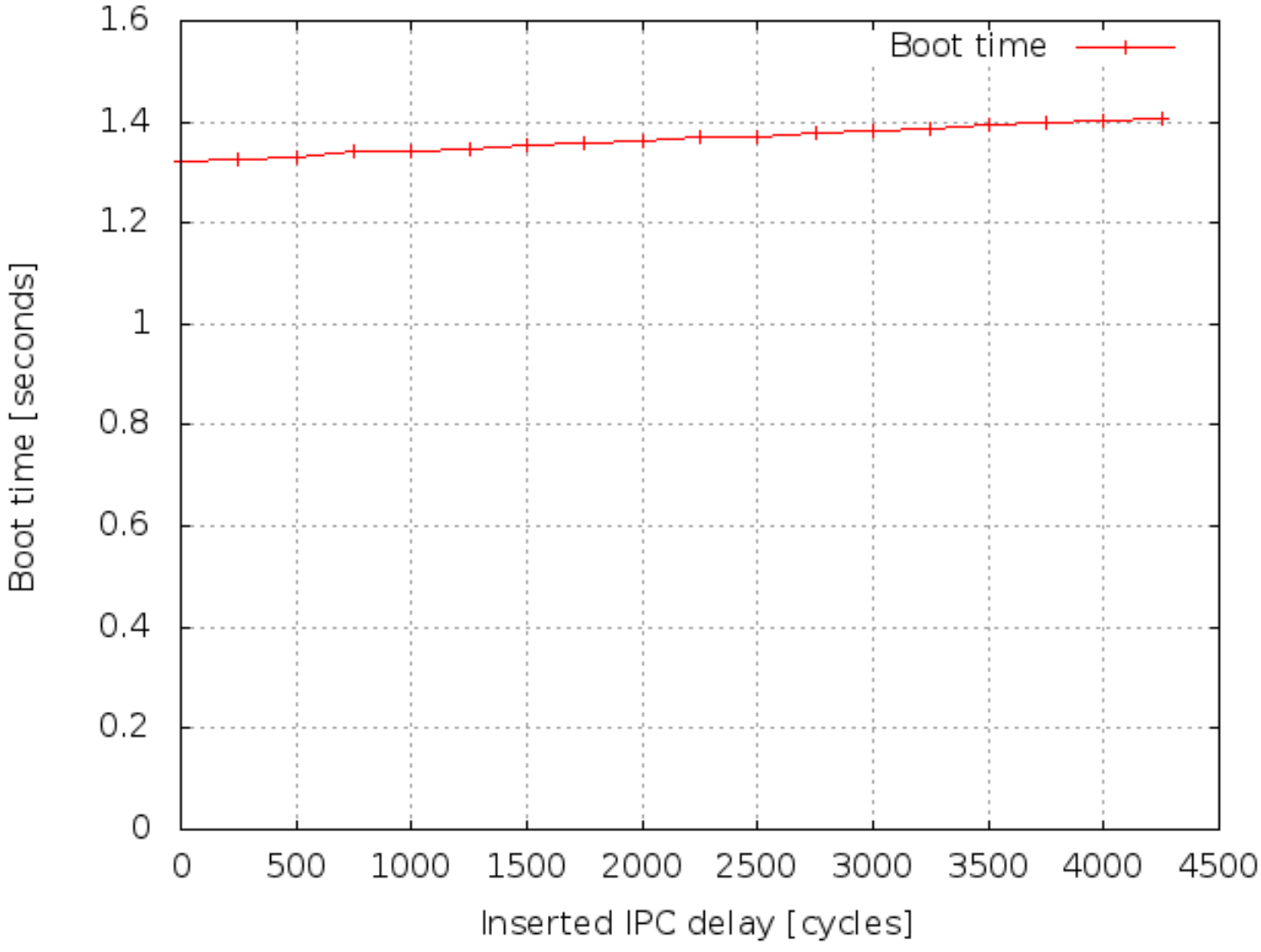
Example (IPC Cost)

```
for i in `seq 0 250 10000`; do
    echo Testing delay of $i cycles
    echo "#define IPC_DELAY $i" > nova/include/delay.h
    scons
    wvnulrun vancouver-linux-boot-time.wv -I
done
```

IPC Benchmarks



IPC Benchmark – Linux VM Boot Time



Some Results from Performance Testing

- Code layout...