Host Factory Case Studies

What is Host Factory?

Linux system administration done by managing files on hosts with closed-loop feedback, using a novel filetree mirroring tool inspired by rdist and rsync.

$\sim\sim\sim$

See also introductory slides from October, 2014 GatorLUG at:

https://www.workver.com/wv/Scenarios/introducing.pdf

Development progress

First public presentation made at GatorLUG in October, 2014

http://www.gatorlug.org/node/352

18 public versions released

https://www.workver.com/wv/News/index.html

This presentation made at GatorLUG in March, 2017

http://www.gatorlug.org/meetings/2017/03

Three case studies

- Two Raspberry Pi's, removal of introduced files, live demo
- Linode virtual host behind www.workver.com
- 35 Gigabyte testbed extracted from 1 Terabyte system

Two Raspberry Pi's

- Check shows clean

  ```
  hfcheck ix
  ```

- Introduce file modification

  ```
  ssh pirogue
  echo bad >> /etc/shadow
  echo bad > /etc/badprogram
  ```

- hfcheck, shows dirty

  ```
  hfcheck -l etc ix
  hflog /tmp/oe.XXXXX/hosts/pirogue/hfcheck.hflog
  ```

- Pull file modifications back

  ```
  hfcheck -w -e -l etc priogue
  ```

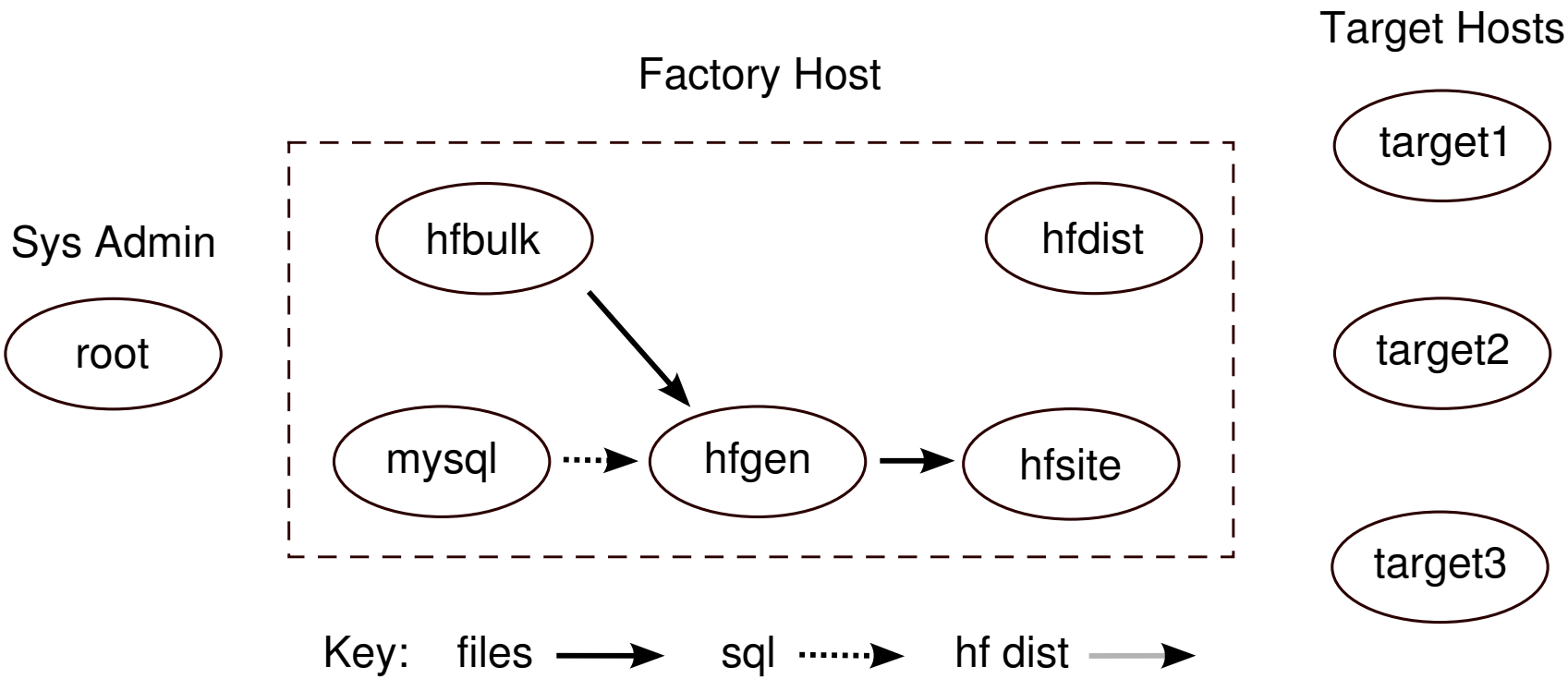
- Examine file modifications

  ```
  statlist -tr /tmp/oe.XXXXX/hosts/pirogue/examine
  ```

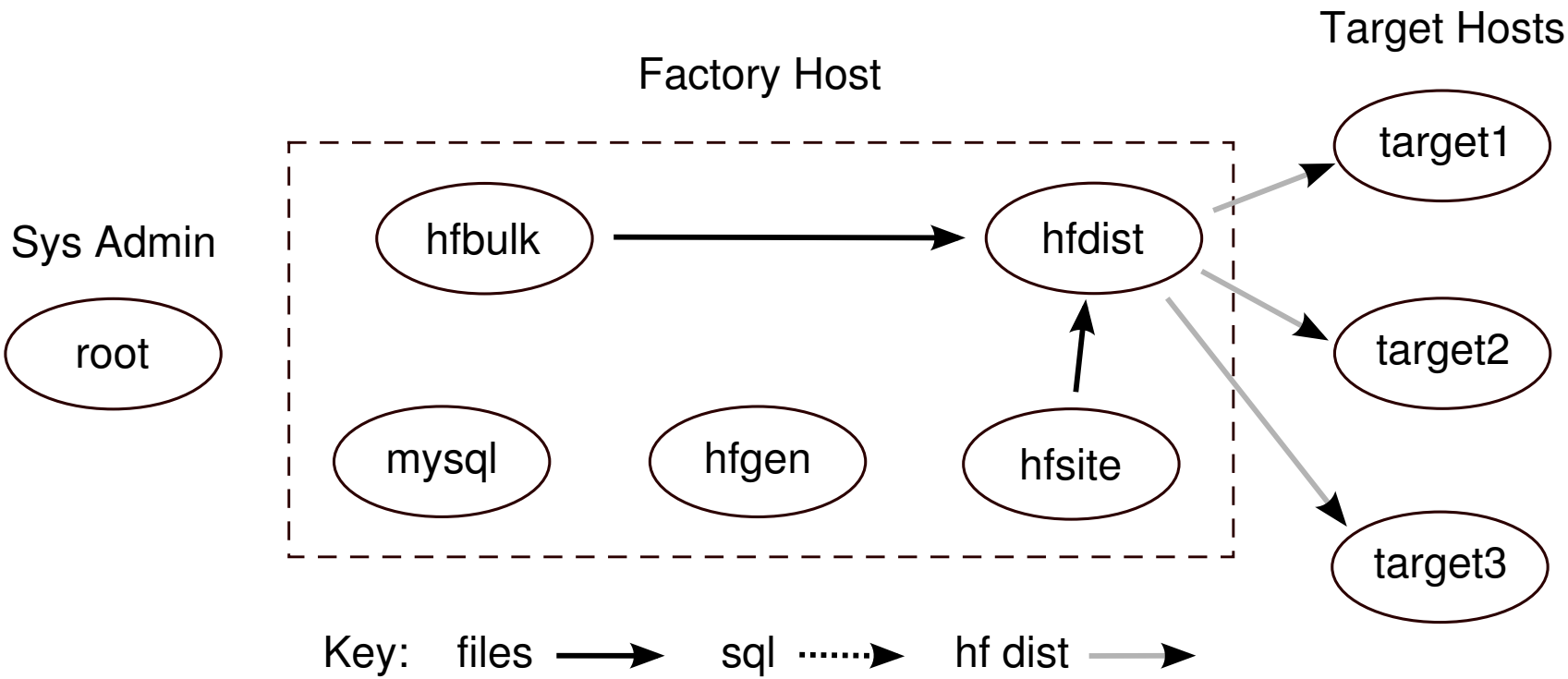
- Overwrite modifications

  ```
  hfcheck -wd -l etc pirogue
  ```

hfgen dataflow



hfcheck dataflow



Performance

3.5 minutes to `hfcheck` 3.1 Gigabytes of 138K total files and dirs each on two Raspberry Pi's, checked across local ethernet.

7 seconds to `hfcheck` 1.7 Megabytes of 1.7K total files and dirs in /etc each on two Pi's, checked across local ethernet.

35 seconds to `oh snapshot` after trivial update to 3.1 Gig of 138K total files and dirs.

Run from my Dell Inspiron N7110 laptop containing 8 Gig of memory and Intel© Core™ i7-2670QM CPU @ 2.20GHz.

Linode virtual host

- Linode is in Atlanta
- No malware or unauthorized updates detected
- Found missing personal software development updates
- Found masking of run-time changes is not perfect

Linode: in Atlanta

```
---------- logfilename 'oe.17267.before/hosts/main/hfcheck.hflog'
----- host 1 direction 'push' channel 'ssh' ip '50.116.33.94'
source_root '/opt/hfbulk/main3/m/g' target_root '/'
dowrite 0 dodelete 0 dochecksums 1 donotfixhardlinks 0 opt_x 0 euid 0
fail_no_remote_answer 0 fail_wrong_remote_version 0 checkreadiness 1 fail_connection 0

time_begin 1489448794 Mon Mar 13 19:46:34 2017
time_end   1489451195 Mon Mar 13 20:26:35 2017  total 00:40:01  hours 0.67  mins 40.02  secs   2401

             file       dir   symlink      pipe   chardev    blkdev    socket       all    file_byte_seen
source    340,554    39,686    85,573         0       650        43         1   466,507     9,143,667,915
target    340,565    39,694    85,573         0       650        43         1   466,526     9,087,442,601
source ----------------------------------------------------------------------dddddddd1111111111111111111
target ----------------------------------------------------------------------dddddddd1111111111111111111

   want_byte_sent want_file      file_byte_sent file_sent   fixlink dir_mtime  diff_tot  fail_tot
      253,966,713       110                   0         0         0         5       302         0
source
target

diffs create     modify     reload     delete | fails  connection      audit     layers
          20        127         91         64 | local           0          0          0
cccmmmmmmmmmmmmmmmmmmmrrrrrrrrrrrrrrddddddddddd | remote          0          0          0
```

40 minutes to check 9 Gigabytes in Atlanta using file checksums; 5 minutes with better Internet and no checksums.

`hfgen`, `hfcheck -k main`, `oewatch -w`

Linode: missing software updates

```
---------- logfilename 'oe.17267.before/hosts/main/hfcheck.hflog'
----- host 1 direction 'push' channel 'ssh' ip '50.116.33.94'
what     why       want        have        t  name
-------  --------  ----------  ----------  -  --------------------------------
fix      owner           1000           0  -  depot/misc-1.0/bin/rsync-dir
fix      group           1000           0  -  depot/misc-1.0/bin/rsync-dir
fix      mtime     1436304551  1466887042  -  depot/misc-1.0/bin/rsync-dir
replace  size           2,195       2,219  -  depot/misc-1.0/bin/rsync-dir

fix      mtime     1446258920  1469348547  d  usr/local/depot/shellrc-1.0/lib
fix      mtime     1446258920  1469348522  -  usr/local/depot/shellrc-1.0/lib/bash-settings
replace  checksum  1816b941abdd0c4fb0af24cf9e203eed  856c37fccaac094744371f0e6a83344b  -  usr/local/depot/
```

I recognized both of those on sight. An improvement to rsync-dir and a new version of Host Factory on the PATH.

Linode: traces from experiments

```
---------- logfilename 'oe.17267.before/hosts/main/hfcheck.hflog'
----- host 1 direction 'push' channel 'ssh' ip '50.116.33.94'
what     why       want        have        t  name
-------  --------  ----------  ----------  -  --------------------------------
fix      mtime     1412043008  1482809860  d  etc/postfix
fix      mtime     1389515320  1482809817  -  etc/postfix/master.cf
```

Only mtime wrong, not file contents, which was examined with checksums. I remember what I was doing there, and then put back, but still left traces.

Linode: hostfactory mailing lists

```
---------- logfilename 'oe.17267.before/hosts/main/hfcheck.hflog'
----- host 1 direction 'push' channel 'ssh' ip '50.116.33.94'
what     why       want        have        t  name
-------  --------  ----------  ----------  -  --------------------------------
fix      mtime     1448070889  1457831755  -  etc/shadow
replace  checksum  78ff920ed837fafc4b3c9b108b350ca3  0a0ccda46f880d216e433e8feb127cf6  -  etc/shadow

fix      mtime     1448070889  1457831755  -  var/backups/shadow.bak
replace  checksum  78ff920ed837fafc4b3c9b108b350ca3  0a0ccda46f880d216e433e8feb127cf6  -  var/backups/shad
```

I was not expecting /etc/shadow to be modified!

Linode: pull file bodies to examine

- `hfcheck -e`
- `-e` switch means, any file contents you were going to update, like /etc/shadow, instead pull that file body back and stick it over there so I can diff it against the contents I thought I wanted
- Turns out I didn't update the generator program for /etc/shadow after I added the hostfactory mailing lists by getting them working live on the host.

Linode: hosting updating kernel

```
---------- logfilename 'oe.17267.before/hosts/main/hfcheck.hflog'
----- host 1 direction 'push' channel 'ssh' ip '50.116.33.94'
what     why       want        have        t  name
-------  --------  ----------  ----------  -  --------------------------------
fix      mtime     1445318226  1488228753  d  lib/modules
delete   dir                               d  lib/modules/4.4.0-x86_64-linode63
delete   file                              -  lib/modules/4.4.0-x86_64-linode63/modules.dep
delete   file                              -  lib/modules/4.4.0-x86_64-linode63/modules.dep
delete   dir                               d  lib/modules/4.5.5-x86_64-linode69
delete   file                              -  lib/modules/4.5.5-x86_64-linode69/modules.dep
delete   file                              -  lib/modules/4.5.5-x86_64-linode69/modules.dep
delete   dir                               d  lib/modules/4.6.3-x86_64-linode70
delete   file                              -  lib/modules/4.6.3-x86_64-linode70/modules.dep
delete   file                              -  lib/modules/4.6.3-x86_64-linode70/modules.dep
delete   dir                               d  lib/modules/4.6.5-x86_64-linode71
delete   file                              -  lib/modules/4.6.5-x86_64-linode71/modules.dep
delete   file                              -  lib/modules/4.6.5-x86_64-linode71/modules.dep
delete   dir                               d  lib/modules/4.7.0-x86_64-linode72
delete   file                              -  lib/modules/4.7.0-x86_64-linode72/modules.dep
delete   file                              -  lib/modules/4.7.0-x86_64-linode72/modules.dep
[...]
```

Linode updating my kernel, something they do in their ordinary maintenance. Doubled file entries is wart from delete in read-only mode not perfectly simulating writing.

Linode: unmasked good changes

```
---------- logfilename 'oe.17267.before/hosts/main/hfcheck.hflog'
----- host 1 direction 'push' channel 'ssh' ip '50.116.33.94'
what     why       want        have        t  name
-------  --------  ----------  ----------  -  --------------------------------
fix      mtime     1389407720  1482810187  d  root/.ssh
delete   file                              -  root/.ssh/known_hosts

fix      mtime     1445696709  1485668364  d  var/lib/squirrelmail/data
fix      mtime     1445696709  1485668364  -  var/lib/squirrelmail/data/bb.pref
replace  size             324         523  -  var/lib/squirrelmail/data/bb.pref

fix      mtime     1446264477           0  -  var/lib/sudo/bb/0
replace  checksum  263ba2d91adacb21ec4e64fc3967abb7  76e0ea078a23adc6c601098c5fdd6ea2  -  var/lib/sudo/bb/
fix      mtime     1446264651           0  -  var/lib/sudo/bb/1
replace  checksum  3e244ec07345ae815fa9baf3918e3c88  fc5da44034a6095e23fd5a026e371dd6  -  var/lib/sudo/bb/
replace  checksum  4cd8dc6bd16ff35712169d34163658da  16ddd0ac7b9a886a1dc5ca25ca7c2fb9  -  var/lib/sudo/bb/
----- diff_count 302
```

More things which should not be reported as incorrect when the mtime or file contents vary.

Linode: work through piecewise

- Worked through exceptions one functional area at a time
- `hfcheck -l depot/misc-1.0/bin`
  Pushed corrections quickly by limiting filetree walk to area worked on like /depot/misc-1.0/bin, /etc, /lib/modules, /root, /usr/local/depot, /var/lib/squirrelmail, /var/lib/sudo. These updates take 30 seconds, not 30 minutes.

Linode: result is a clean check

```
---------- logfilename 'oe.25240.after/hosts/main/hfcheck.hflog'
----- host 1 direction 'push' channel 'ssh' ip '50.116.33.94'
source_root '/opt/hfbulk/main3/m/g' target_root '/'
dowrite 0 dodelete 0 dochecksums 0 donotfixhardlinks 0 opt_x 0 euid 0
fail_no_remote_answer 0 fail_wrong_remote_version 0 checkreadiness 1 fail_connection 0

time_begin 1489465574 Tue Mar 14 00:26:14 2017
time_end   1489467730 Tue Mar 14 01:02:10 2017  total 00:35:56  hours 0.60  mins 35.93  secs   2156

             file       dir   symlink      pipe   chardev    blkdev    socket       all    file_byte_seen
source    340,562    39,694    85,573         0       650        43         1   466,523     9,143,668,080
target    340,562    39,694    85,573         0       650        43         1   466,523     9,147,392,927
source ------------------------------------------------------------------------dddddddd1111111111111111111
target ------------------------------------------------------------------------dddddddd1111111111111111111

   want_byte_sent want_file      file_byte_sent file_sent   fixlink dir_mtime  diff_tot  fail_tot
                0         0                   0         0         0         0         0         0
source
target

diffs create     modify     reload     delete | fails  connection      audit     layers
           0          0          0          0 | local           0          0          0
                                               | remote          0          0          0
```

Ended with a clean check.

Testbed extracted from production

- Application software development customer has a machine with a Terabyte of data in Oracle collected over 20 years
- I want to extract the minimal sized machine image copy into virtualbox so I have a local test machine for application development, which I can roll back to snapshot discarding any buggy inserts into Oracle data
- Machine also has a lot of date-named directories and tarfiles for backup, many of which are large
- Machine is 2,700 miles away, and customer is accessed through VPN on virtualbox'ed Windows. Minimizing data transfer volume is useful.
- When customer updates operating system, I want to identify and propagate those changes down to my virtualbox

Testbed: how to stage this

- `hfdist -P '[-p, 22000]'`
  send the hfdist ssh through a port which enters the Windows VPN
- Copy downloaded filetree into a staging area
- Load a virtualbox machine with same base operating system version as customer's
- After download completes, use hfdist to modify base operating system to match staging area copy
- Also made changes for virtualbox like ethernet interfaces, RAID hardware, etc.

Questions?

https://www.workver.com/wv/Screenshots/index.html

https://www.workver.com/wv/Scenarios/introducing.pdf

Brian Bartholomew ⟨bb@workver.com⟩

About Brian Bartholomew

Brian Bartholomew has been a Unix system administrator for 25 years. His first system ran SCO Xenix on a Compaq desktop PC running a 286 (not a 386). Half his work experience has been in the commercial world in Boston, and half at the University of Florida. In Boston he supported financial trading floors for mutual fund firms and banks; at UF he supported two departments of mathematicians, both theoretical and applied, and the PeopleSoft enterprise accounting system. His career speciality in version control of running Unix systems started in 1990 with rdist. The first version of the dist program appeared in 1994, written in perl 4. A Linux Journal article appeared in 1997, featuring a versioning filesystem implemented as a user space NFS server talking to PostGres.