`rshall`: A Tool for Managing Hosts in Parallel

Leon Towns-von Stauber
LinuxFest Northwest, April 2012
http://www.occam.com/

Contents

Introduction	3
Setup	7
Usage	18
Parallel Copies: `cpall`	42
Parallel SQL: `sqall`	46
Conclusion	50

Introduction

- →`rshall` runs commands on multiple hosts in parallel
  - →A host can be anything that accepts SSH or `rsh`
    - Most commonly UNIX-based devices, but has been used on networking gear and other appliances
- →It's fast: Parallel execution means you can get responses from many hosts quickly
  - →`rshall` routinely runs against hundreds of hosts simultaneously, from a single machine

Introduction

- →Development history
  - →Started writing a bunch of shell scripts for mass host management in 1996
  - →In 1997, factored common elements into a template script (`doit.sh`) that made writing them easier
  - →Generalized into an all-purpose Perl script in 1998
  - →Released as open source in 2003 (version 7.0)
  - →This talk covers version 14.1
  - →The name *rshall* is a throwback to the days when rsh was the dominant remote command execution tool
    - It should go without saying that SSH is *highly* recommended now

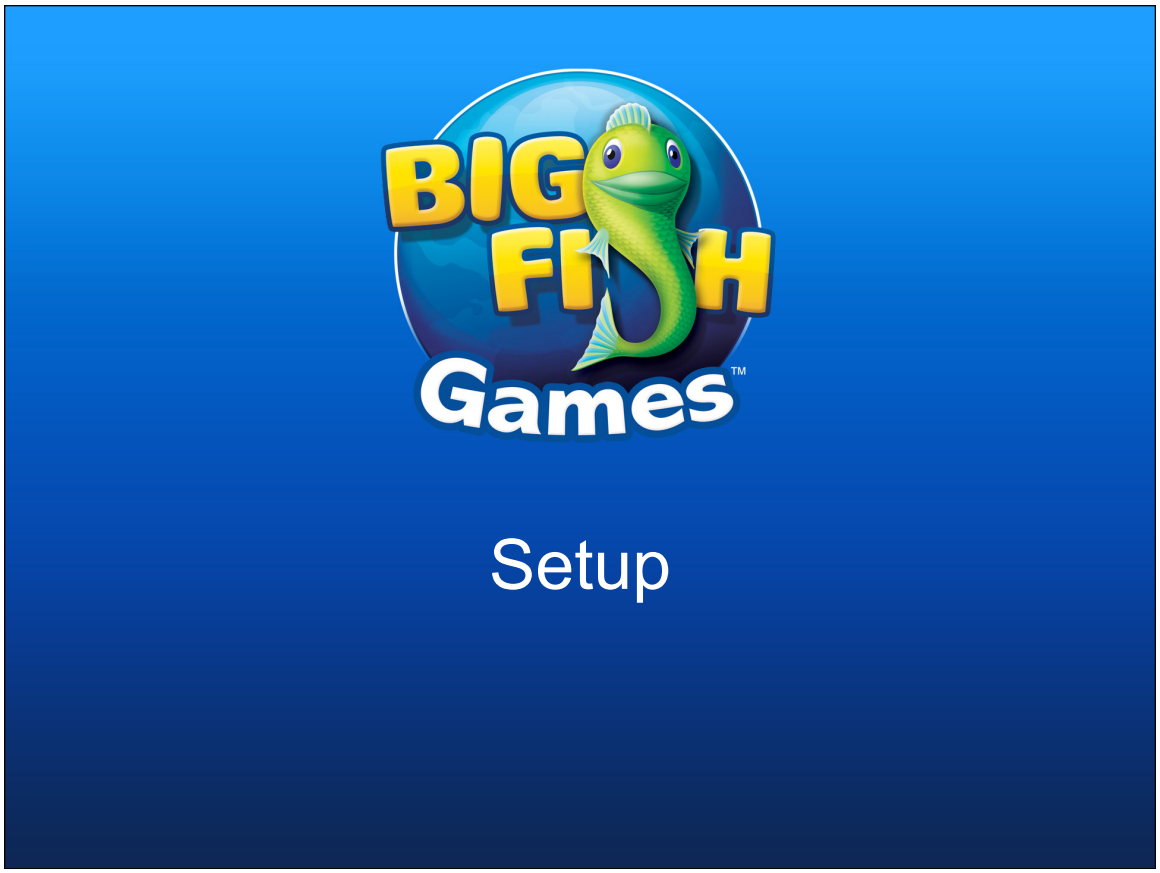
Introduction

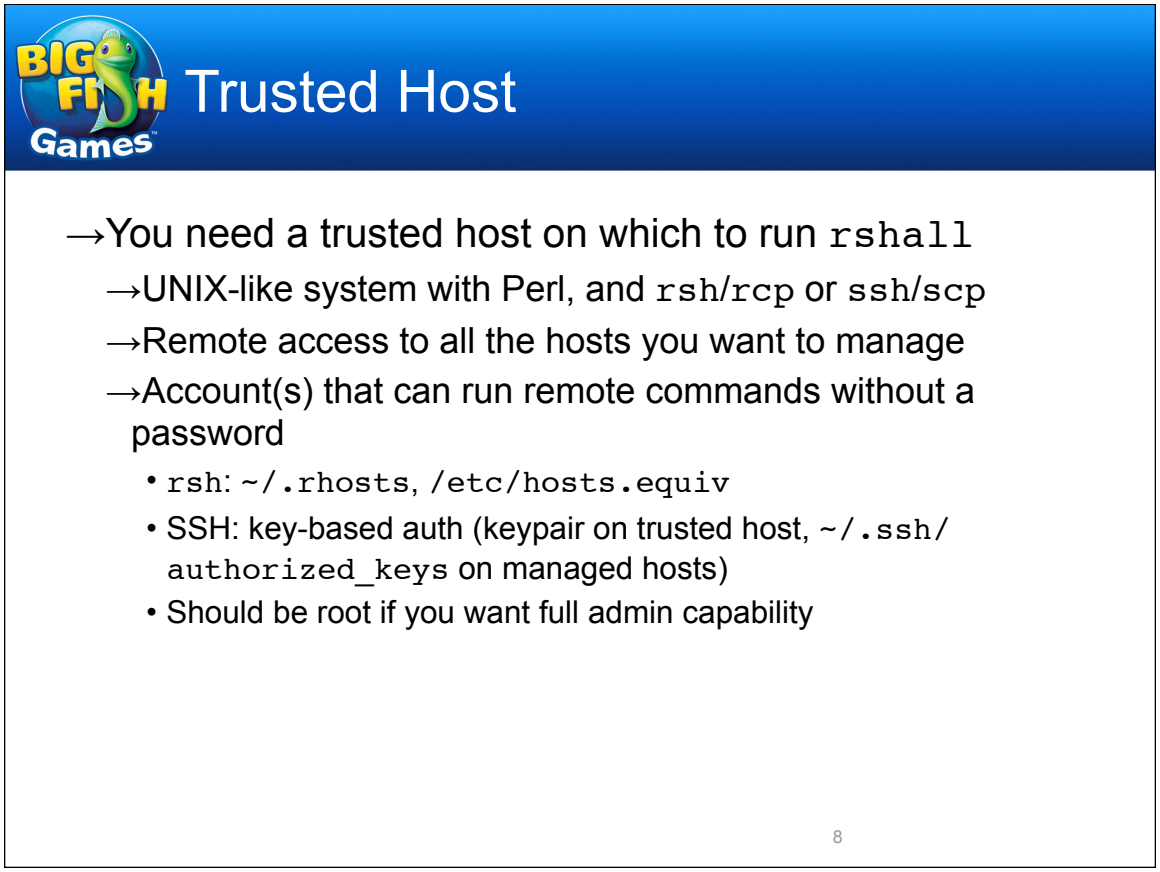
→A number of similar tools have been developed by others

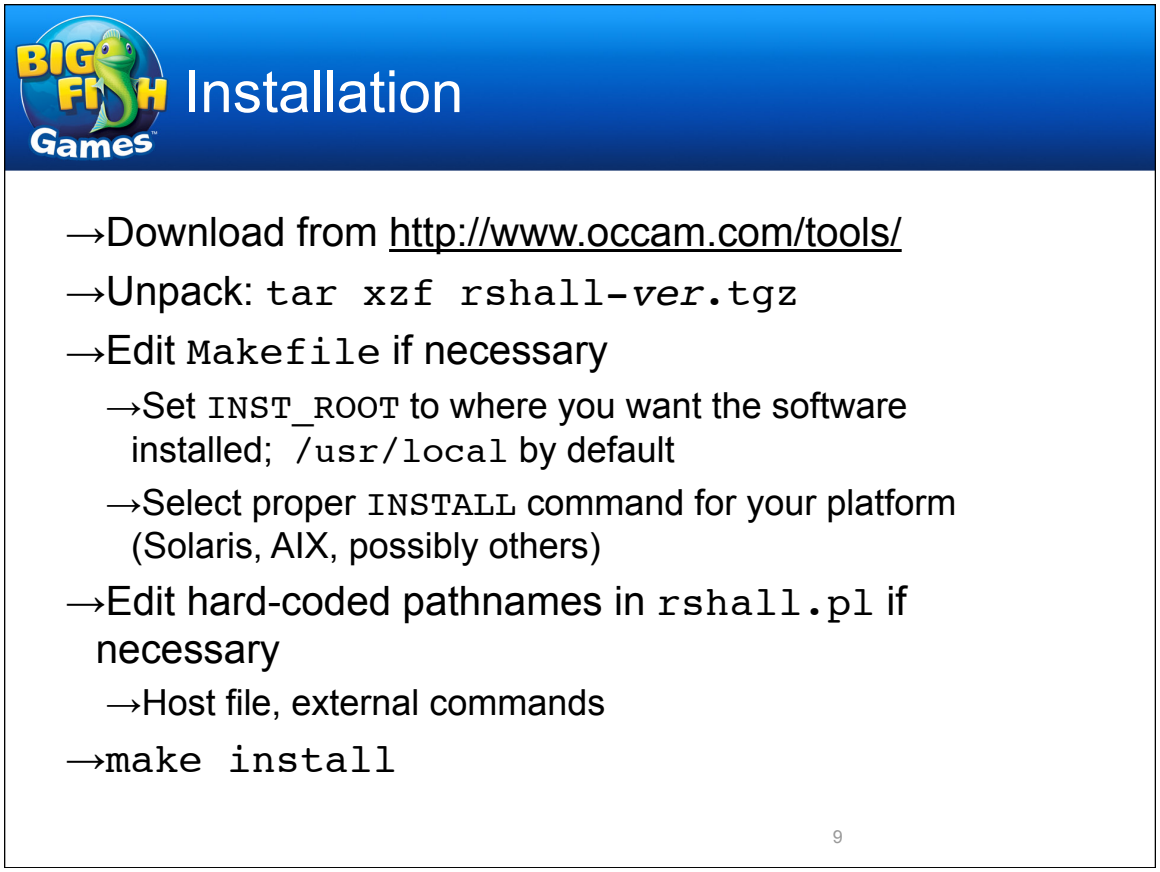
→http://web.taranis.org/shmux/#related

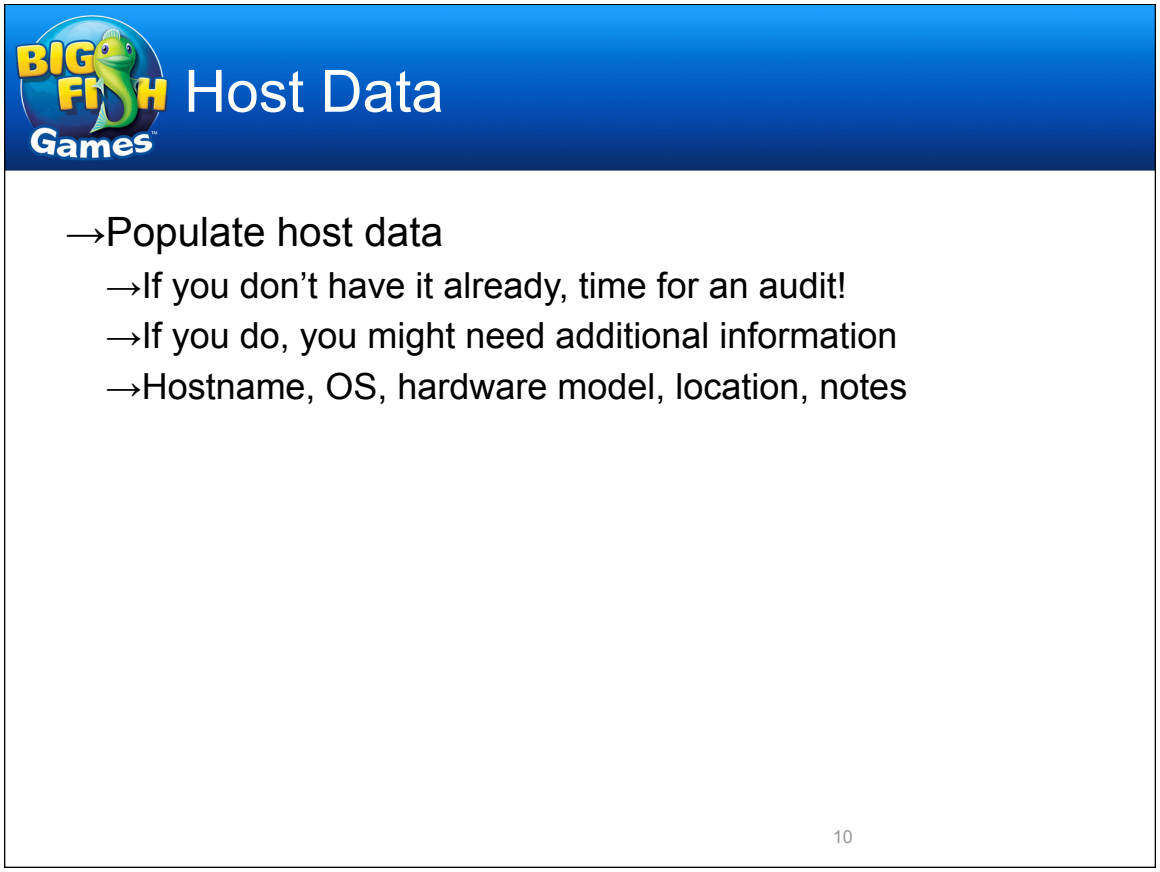
→http://replay.web.archive.org/20090226235234/http://tentakel.biskalar.de/similar/

Legal Notices


BIG FISH
Games
Setup


Trusted Host
→You need a trusted host on which to run rshall
→UNIX-like system with Perl, and rsh/rcp or ssh/scp
→Remote access to all the hosts you want to manage
→Account(s) that can run remote commands without a password
• rsh: ~/.rhosts, /etc/hosts.equiv
• SSH: key-based auth (keypair on trusted host, ~/.ssh/authorized_keys on managed hosts)
• Should be root if you want full admin capability
8


Installation
→Download from http://www.occam.com/tools/
→Unpack: tar xzf rshall-ver.tgz
→Edit Makefile if necessary
→Set INST_ROOT to where you want the software installed; /usr/local by default
→Select proper INSTALL command for your platform (Solaris, AIX, possibly others)
→Edit hard-coded pathnames in rshall.pl if necessary
→Host file, external commands
→make install
9


Host Data
→Populate host data
→If you don't have it already, time for an audit!
→If you do, you might need additional information
→Hostname, OS, hardware model, location, notes
10

Host Data

- →Data source
  - →Text file
    - By default, `/usr/local/etc/systems`
    - Sample `systems` file included in distribution
  - →Text file formatted for use by `readinfo`
    - `readinfo` also at http://www.occam.com/tools/
    - More flexible than regular text file, more compact, but there's a bit of a learning curve
    - Sample `readinfo`-formatted file included
    - `readinfo` automatically used to process `systems` file if it exists in the same directory as `rshall`
    - Further details beyond the scope of this talk

Host Data

```
#Hostname OS                 Hardware              Loc      Comments      SSH?
#-------- --                 --------              ---      --------      ----
linux1    Red Hat Linux 7.1  Dell PowerEdge 1550    SEA      server,dns    y
linux3    SUSE Linux 9       IBM p5 570             NYC      server        y
mac1      Mac OS X 10.2.5    Apple PowerMac G4      2NW      client        y
mac2      Mac OS X 10.4.3    Apple PowerBook G4     mobile   client        y
solaris1  Solaris 7          Sun Ultra 5            DC1      NULL          NULL
solaris2  Solaris 8          Sun Enterprise 6500    DC2      server        y
solaris3  Solaris 9          Dell Precision WS      NULL     NULL          y
nas1      Data OnTAP 6.1.2R3 NetApp F740            DC1      server        n
```

Sample `systems` file

Host Data

```
#FIELDS   GLOBAL    no=. null=X    host os   prefix=Red Hat Linux    hw   loc
     comment   ssh
linux1    7.1                Dell PowerEdge 1550     SEA       server,dns     y
#linux2   6.2                HP NetServer LPr        X         server
linux3    .SUSE Linux 9      IBM p5 570              NYC       server         y

#FIELDS   GLOBAL    no=. null=X    host os   prefix=Mac OS X    hw   prefix=Apple
     loc  comment   ssh
mac1      10.2.5             PowerMac G4             2NW       client         y
mac2      10.4.3             PowerBook G4            mobile    client         y

#FIELDS   GLOBAL    no=. null=X    host os   prefix=Solaris     hw   prefix=Sun
     loc  comment   ssh
solaris1  7                  Ultra 5                 DC1       X
solaris2  8                  Enterprise 6500         DC2       server         y
solaris3  9                  .Dell Precision WS      X         X              y

#FIELDS   GLOBAL    no=. null=X    host os   prefix=Data OnTAP  hw   prefix=NetApp
     loc  comment   ssh
nas1      6.1.2R3            F740                    DC1       server         n
```

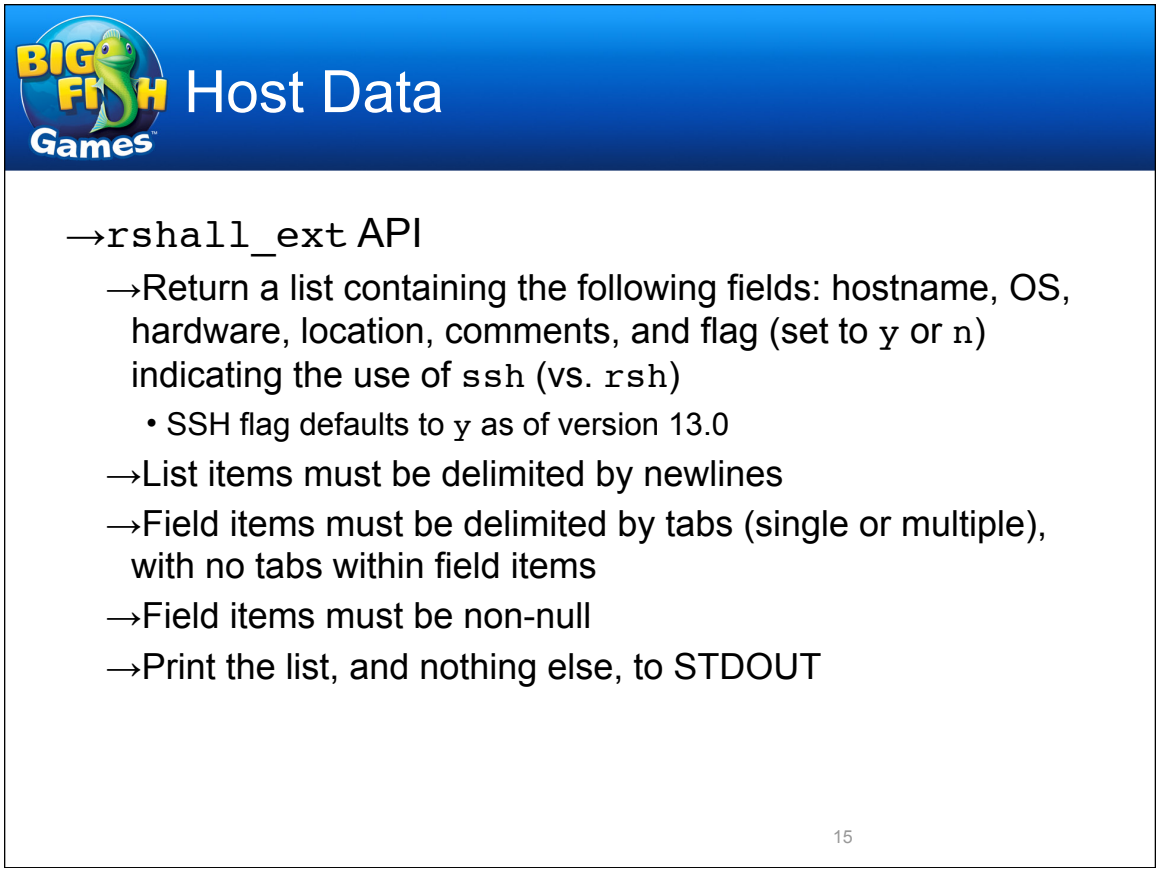
Sample `systems` file for `readinfo`

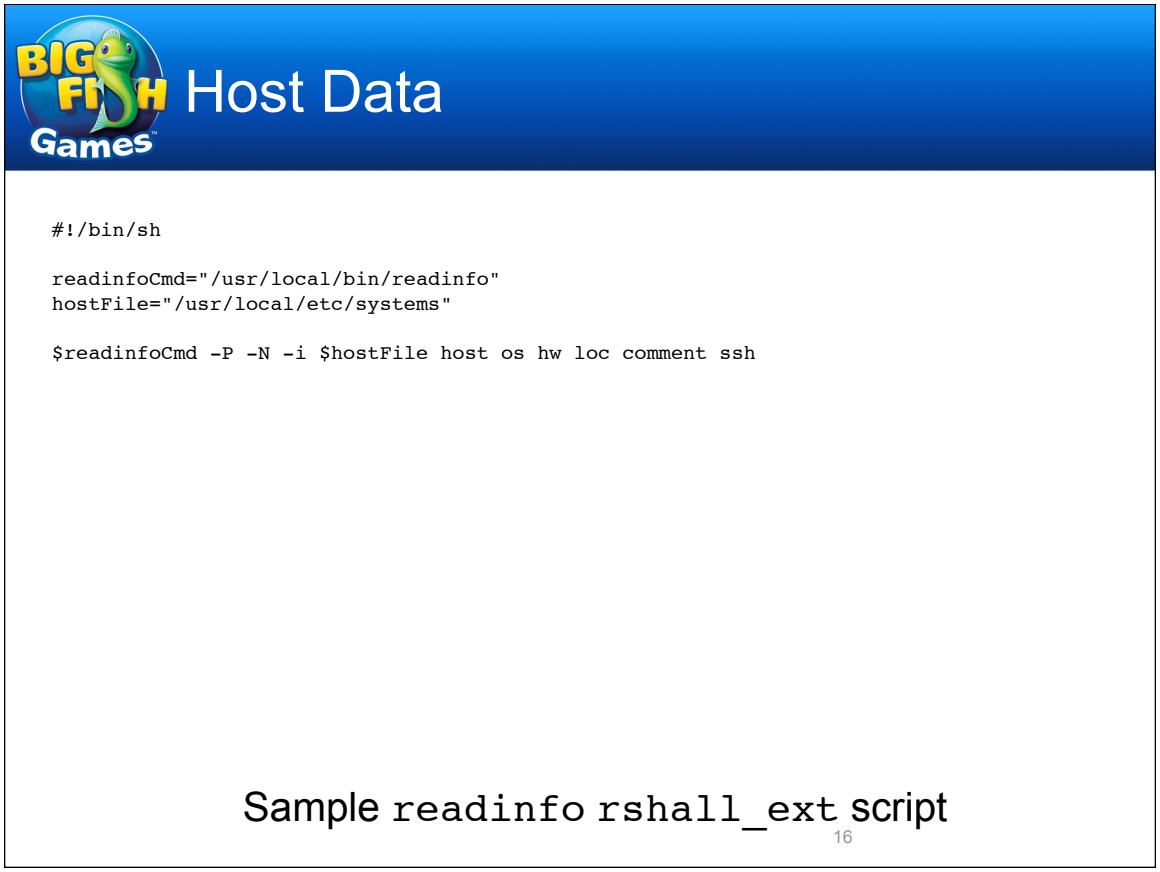
Host Data

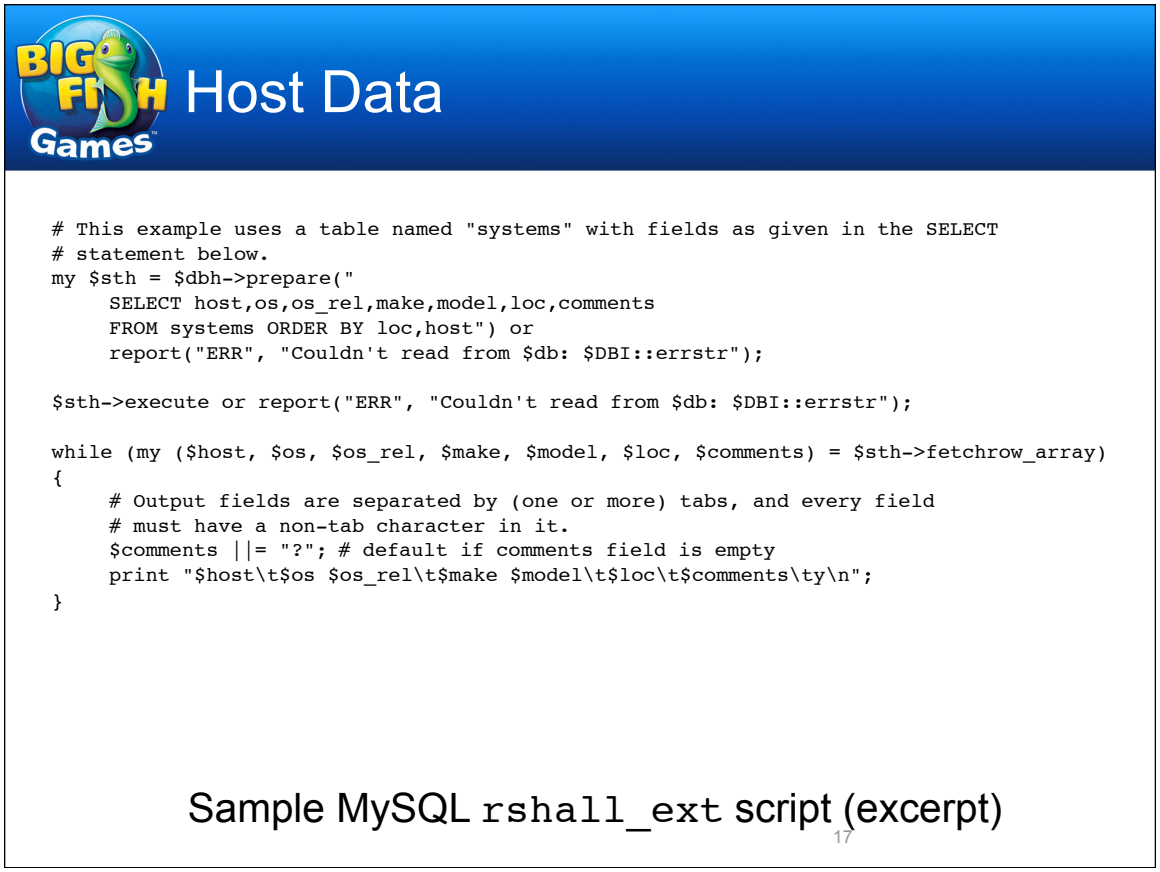
→Data source (cont'd.)

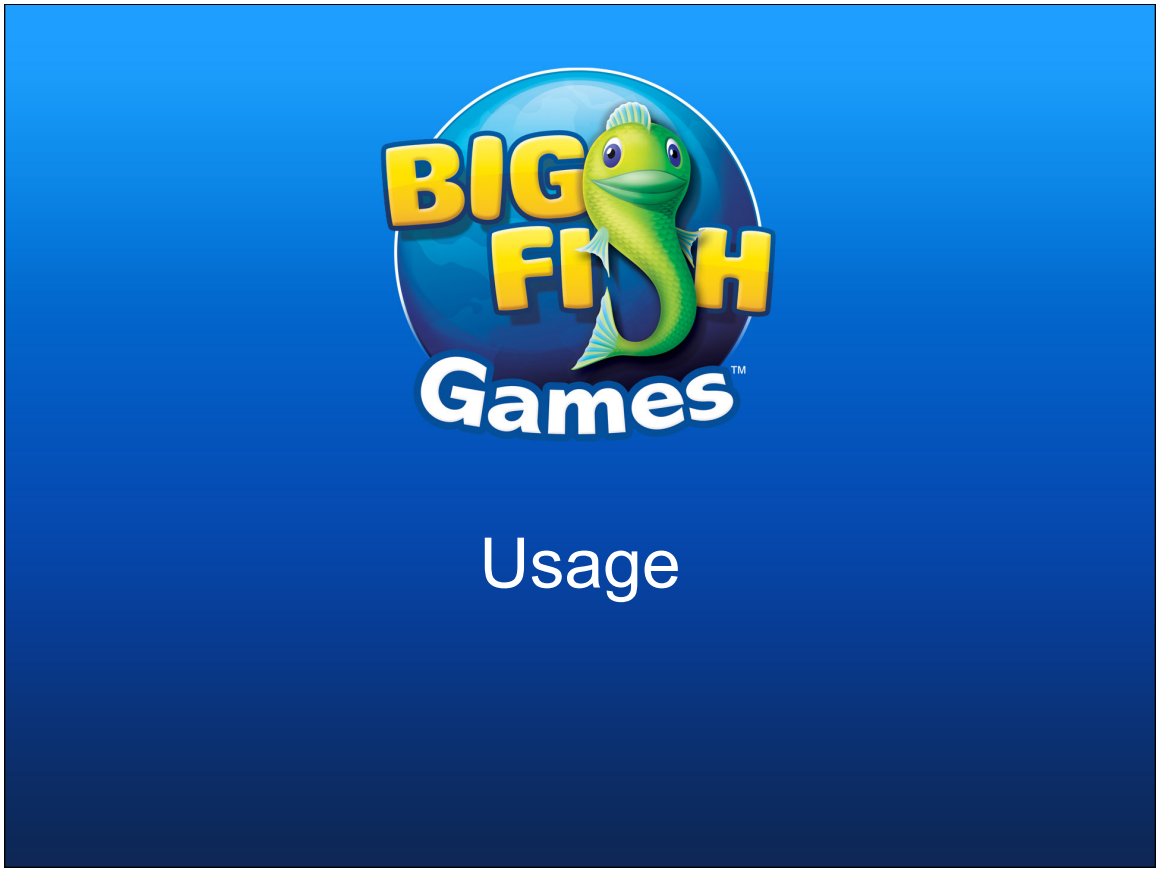
→Custom-formatted flat file, database, LDAP, etc.: `rshall` extension

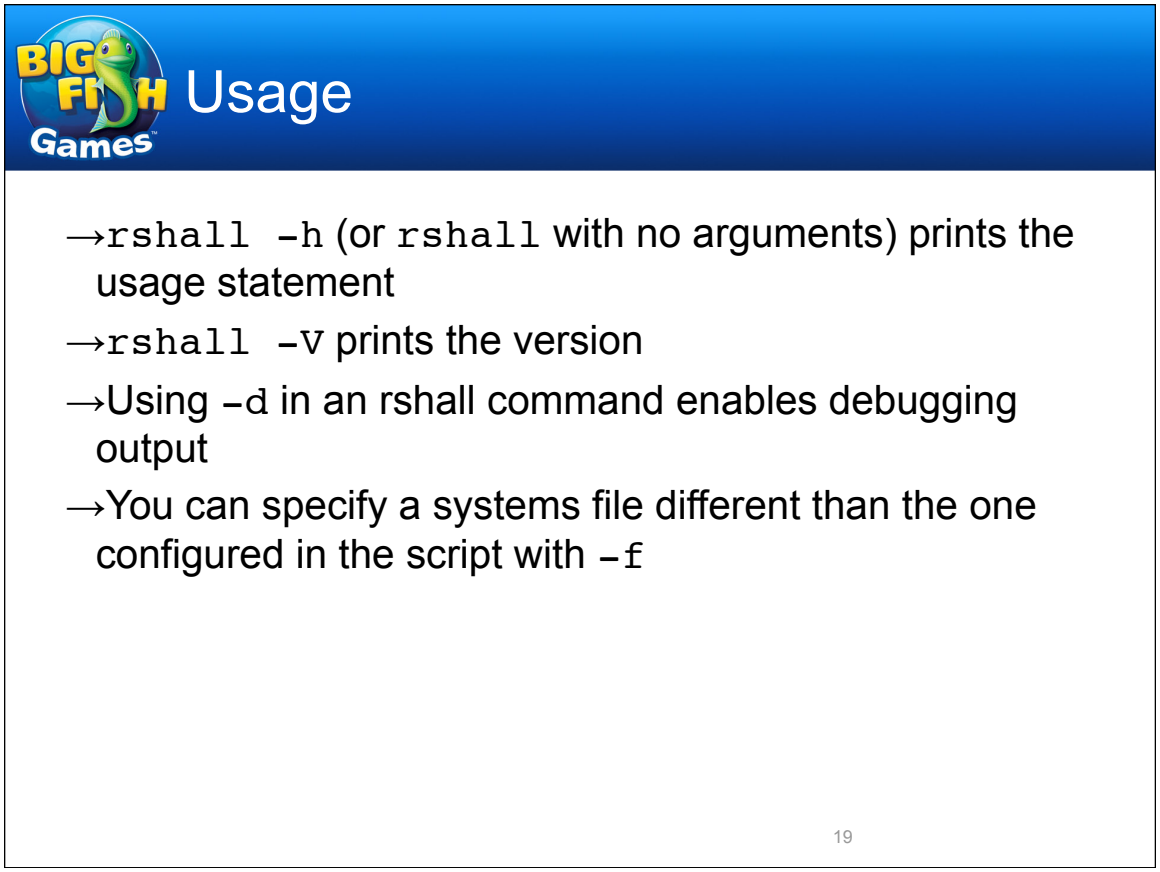
- If an executable script named `rshall_ext` exists in the same directory as `rshall`, it will be used in preference to `readinfo` or to a direct read of the `systems` file
- Your `rshall_ext` script gathers data any way you need, and returns it in a form `rshall` can use
- Sample `rshall_ext` scripts included for `readinfo` (4-line shell script) and for a MySQL database (~45 lines of Perl)

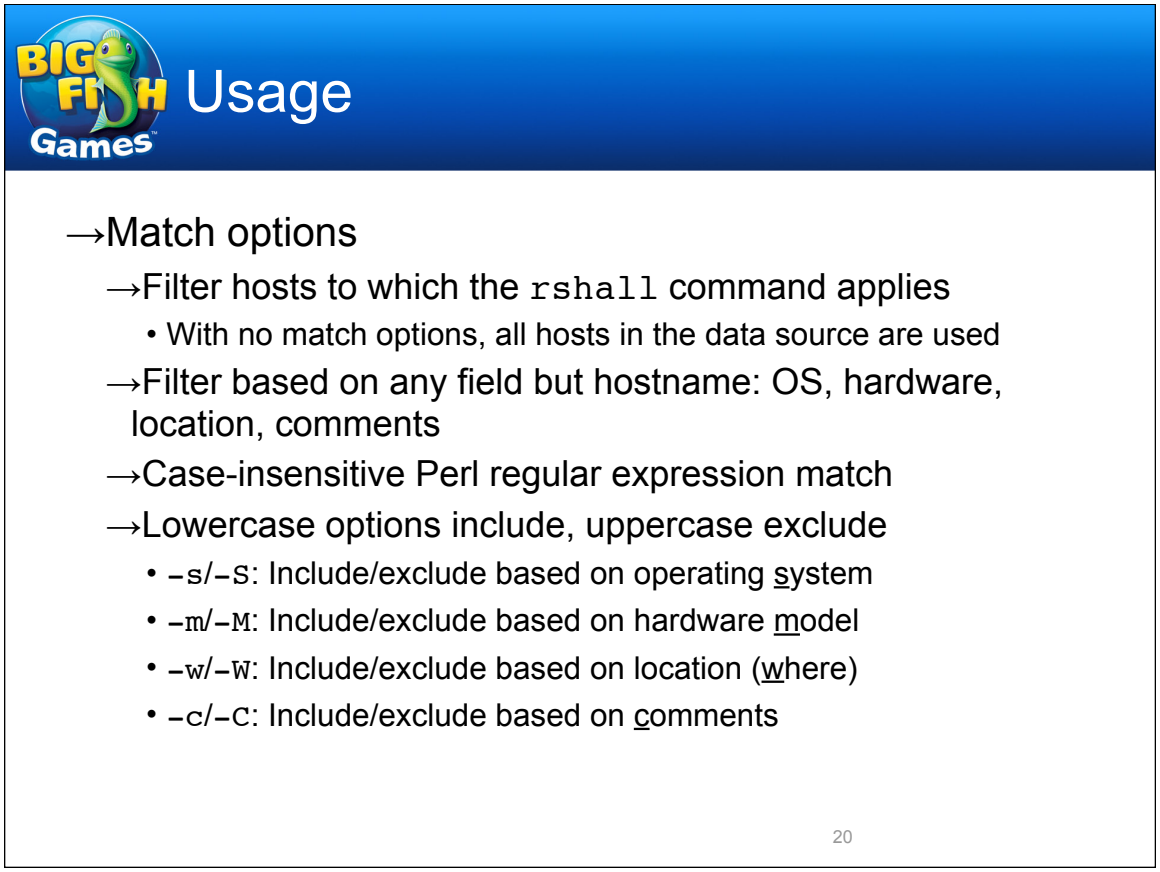

Host Data
→rshall_ext API
→Return a list containing the following fields: hostname, OS, hardware, location, comments, and flag (set to y or n) indicating the use of ssh (vs. rsh)
• SSH flag defaults to y as of version 13.0
→List items must be delimited by newlines
→Field items must be delimited by tabs (single or multiple), with no tabs within field items
→Field items must be non-null
→Print the list, and nothing else, to STDOUT
15


Host Data
#!/bin/sh
readinfoCmd="/usr/local/bin/readinfo"
hostFile="/usr/local/etc/systems"
$readinfoCmd -P -N -i $hostFile host os hw loc comment ssh
Sample readinfo rshall_ext script
16


Host Data
This example uses a table named "systems" with fields as given in the SELECT
statement below.
my $sth = $dbh->prepare("
SELECT host,os,os_rel,make,model,loc,comments
FROM systems ORDER BY loc,host") or
report("ERR", "Couldn't read from $db: $DBI::errstr");
$sth->execute or report("ERR", "Couldn't read from $db: $DBI::errstr");
while (my ($host, $os, $os_rel, $make, $model, $loc, $comments) = $sth->fetchrow_array)
{
# Output fields are separated by (one or more) tabs, and every field
# must have a non-tab character in it.
$comments ||= "?"; # default if comments field is empty
print "$host\t$os $os_rel\t$make $model\t$loc\t$comments\ty\n";
}
Sample MySQL rshall_ext script (excerpt)
17

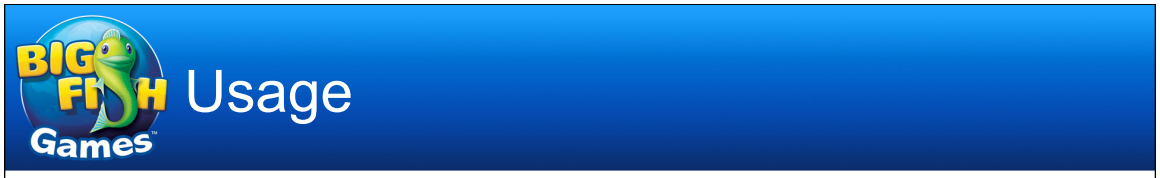

Usage


Usage
→rshall -h (or rshall with no arguments) prints the usage statement
→rshall -V prints the version
→Using -d in an rshall command enables debugging output
→You can specify a systems file different than the one configured in the script with -f
19


Usage
→Match options
→Filter hosts to which the rshall command applies
• With no match options, all hosts in the data source are used
→Filter based on any field but hostname: OS, hardware, location, comments
→Case-insensitive Perl regular expression match
→Lowercase options include, uppercase exclude
• -s/-S: Include/exclude based on operating system
• -m/-M: Include/exclude based on hardware model
• -w/-W: Include/exclude based on location (where)
• -c/-C: Include/exclude based on comments
20


```
rshall 14.1

usage:    rshall { -h | -V }
     rshall [-d] [-f filename] [match_option match_arg]... -l [-v]
     rshall [-d] [-f filename] [match_option match_arg]...
          [-t timeout] [-n max_conns] [-r]
          [-1 | -F base_path] [[-D | -L] max_lines] command

     -h: Prints this usage statement and exits.
     -V: Prints version number and exits.
     -d: Enables debugging output.
     -f: Selects file with host info. Defaults to /usr/local/etc/systems.
     -l: Lists matching hosts, without executing remote commands.
     -v: When listing hosts, prints associated info.
     -t: Connection timeout, in seconds. Defaults to 10.
     -n: Maximum simultaneous connections. Defaults to no limit (0).
          Setting this to 1 forces serialized connections.
     -r: Makes connections as root (using sudo), instead of as calling user.

...
```

rshall -h




```
...

     Match options are used to restrict which hosts are listed or contacted.
     The arguments to these options are used in case-insensitive substring
     matches. Match options include:

     -s: Includes hosts that match operating system name and/or version.
     -S: Excludes hosts that match operating system name and/or version.
     -m: Includes hosts that match hardware model.
     -M: Excludes hosts that match hardware model.
     -w: Includes hosts that match location.
     -W: Excludes hosts that match location.
     -c: Includes hosts that match comments.
     -C: Excludes hosts that match comments.

     -1: Produces output in a compact format, suitable for commands that
          generate single-line output.
     -F: Instead of printing to standard output, sends output to individual
          files named by host, as "<base_path><hostname>".
     -D: Show differences; discard lines appearing more than max_lines times.
     -L: Display by line; discard lines appearing more than max_lines times.
     command: Command to execute on hosts. No need for quotes, unless you
          include shell metacharacters.
```

rshall -h

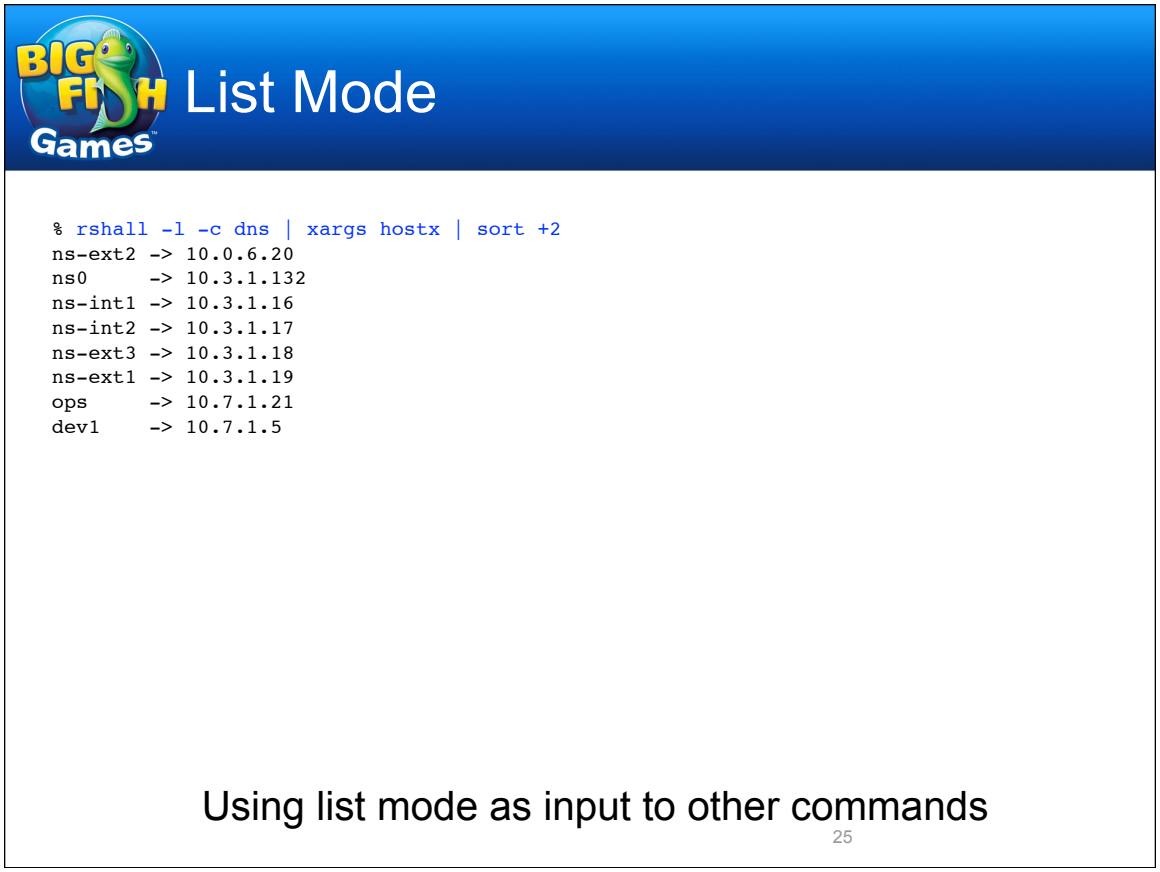
List Mode

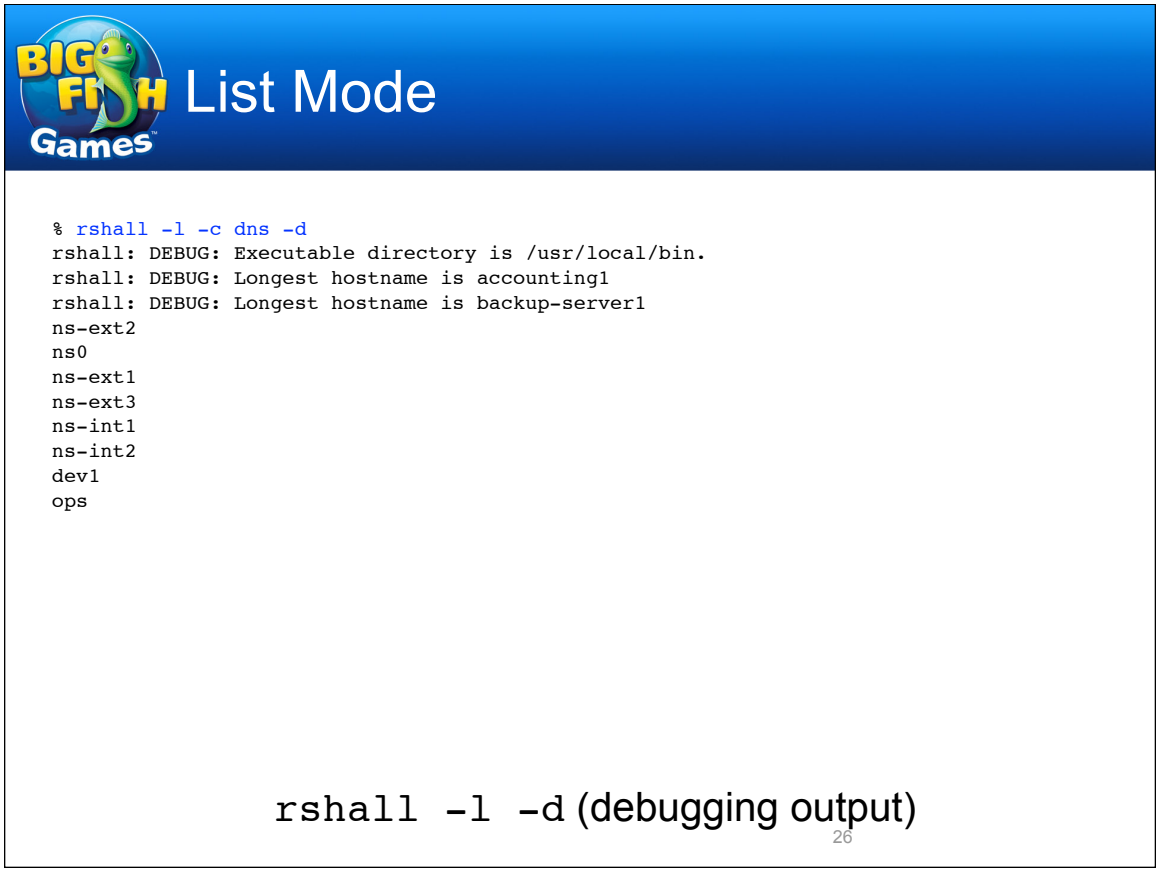
- →Using `rshall -l` lists hosts that match according to the options you provide
  - →No remote commands are run
- →Use `rshall -lv` to list hosts with data
- →Order of results determined by the data source
- →Useful for:
  - →Quickly accessing basic host data
  - →Generating lists of hosts for use by other commands or scripts
  - →Testing your match options before running a remote command

List Mode

```
% rshall -l -c dns
ns-ext2
ns0
ns-ext1
ns-ext3
ns-int1
ns-int2
dev1
ops

% rshall -lv -c dns
ns-ext2   RHELinux 4.6   VMware VM DC1.1-6   ext dns,vmhost1
ns0       RHELinux 4.6   HP DL360 G5    DC2.1-1   ops,int dns
ns-ext1   RHELinux 5.3   HP DL360 G5    DC2.3-1   ext dns
ns-ext3   RHELinux 4.6   HP DL140 G3    DC2.3-1   ext dns
ns-int1   RHELinux 4.4   HP DL360 G5    DC2.1-5   int dns,ntp,dhcp
ns-int2   RHELinux 4.6   HP DL360 G5    DC2.1-4   int dns,ntp
dev1      RHELinux 4.6   HP DL360 G5    DC3.2-1   dev,int dns,ntp,ldap
ops       RHELinux 5.4   HP DL360 G6    DC3.2-1   ops,int dns,ntp,dhcp,ks,ldap,db

% rshall -lv -c dns -w DC2 -C "dhcp|ops" -M dl140
ns-ext1   RHELinux 5.3   HP DL360 G5    DC2.3-1   ext dns
ns-int2   RHELinux 4.6   HP DL360 G5    DC2.1-4   int dns,ntp
```

`rshall -l`


List Mode
% rshall -l -c dns | xargs hostx | sort +2
ns-ext2 -> 10.0.6.20
ns0 -> 10.3.1.132
ns-int1 -> 10.3.1.16
ns-int2 -> 10.3.1.17
ns-ext3 -> 10.3.1.18
ns-ext1 -> 10.3.1.19
ops -> 10.7.1.21
dev1 -> 10.7.1.5
Using list mode as input to other commands
25


List Mode
% rshall -l -c dns -d
rshall: DEBUG: Executable directory is /usr/local/bin.
rshall: DEBUG: Longest hostname is accounting1
rshall: DEBUG: Longest hostname is backup-server1
ns-ext2
ns0
ns-ext1
ns-ext3
ns-int1
ns-int2
dev1
ops
rshall -l -d (debugging output)
26

Command Mode

- →If you don't specify `-l`, then `rshall` attempts to run a provided command on all matching hosts
  - →Command output is printed to standard output as it's received, separated by host
- →Additional options in command mode
  - →`-t`: Connection timeout. Defaults to 10 seconds.
  - →`-n`: Maximum simultaneous connections. By default this is unlimited, but you can have it run the command on only 5, 10, 50, etc. hosts at a time. As hosts respond, new connections are made to keep the total at or under this number. If you specify 1, then `rshall` runs serially instead of in parallel, which may be useful if you want to run commands on hosts in order.

Command Mode

- →Additional options in command mode (cont'd.)
  - →`-r`: If you run `rshall` as a non-root user, attempts to make the remote connections as root by calling `sudo`. This option was implemented for an environment in which access to host data (from LDAP) was restricted to accounts for real users.
  - →`-1`: If you expect single-line command output, this compacts `rshall` output to one line per host, making it suitable for piping into other commands (`grep`, `sort`, `cut`, etc.)
  - →`-F`: Instead of printing to standard output, dumps output to files named by host.

Command Mode

→Additional options in command mode (cont'd.)

→`-D`: Provide a numerical argument, and `rshall` discards lines of output that appear more than that many times, allowing you to focus on outliers.

→`-L`: Like `-D`, but output is organized by line instead of by host. Use to get lists of hosts that return the same output.

→These two options will take longer to return, as they need to gather output from all hosts before it can be trimmed and displayed

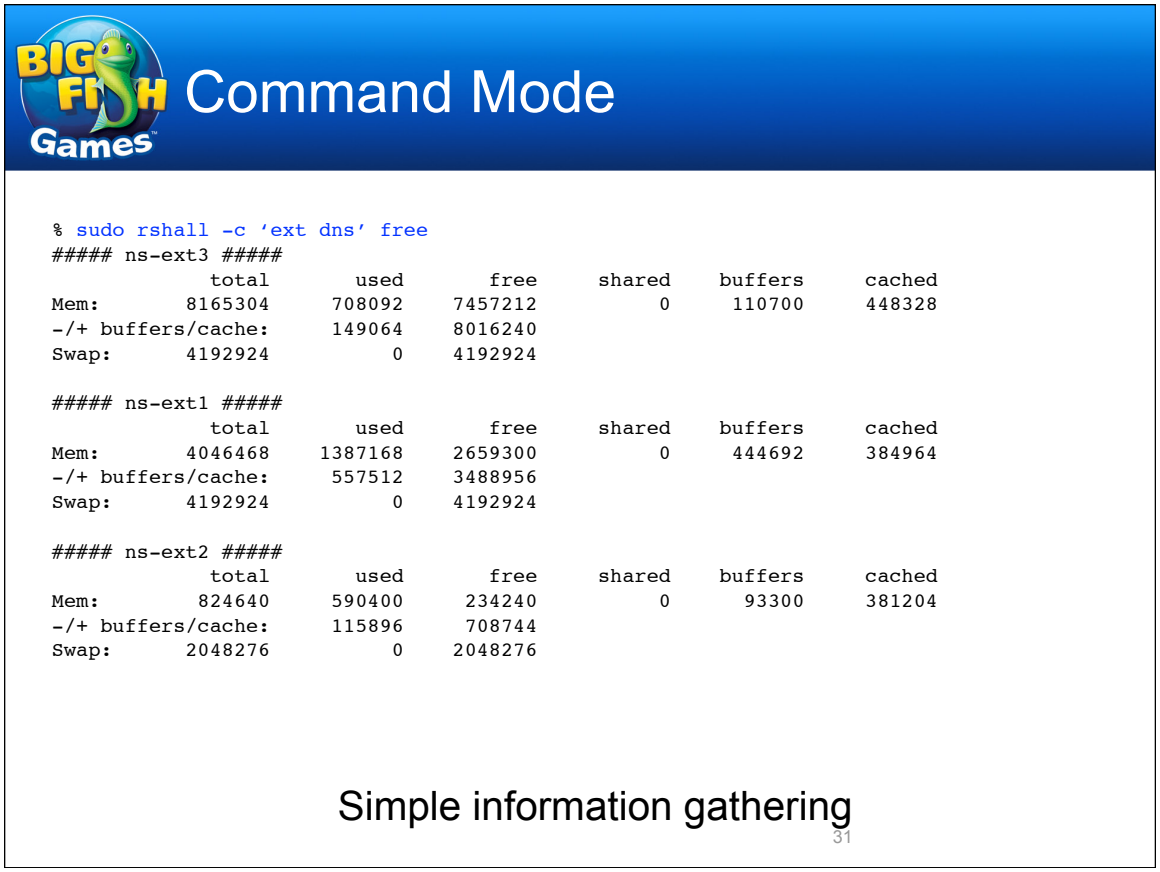
Command Mode

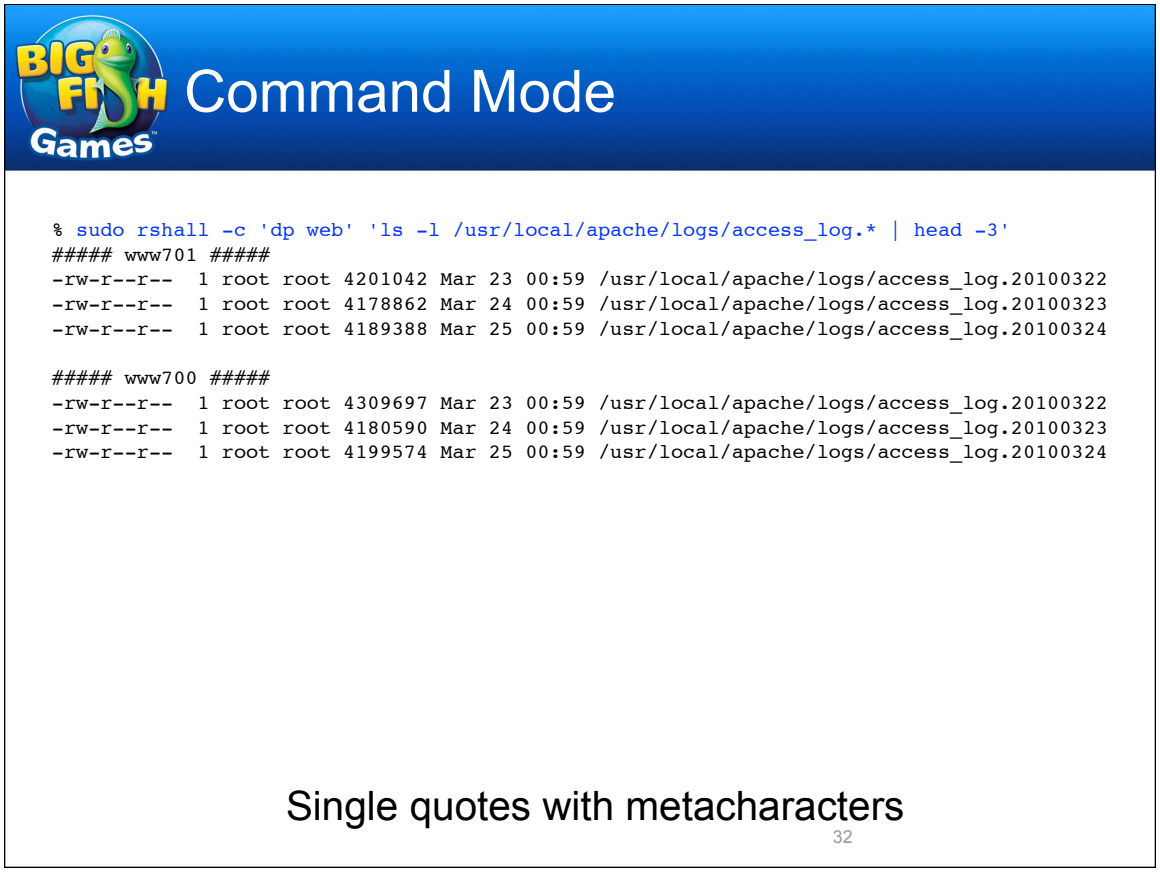
→The command argument doesn't need to be quoted unless you use shell metacharacters (semicolon, pipe, backtick, angle brackets, glob characters, etc.)

→In that case, simply single-quoting the entire command string will probably work

→If not, use backslashes to escape special characters

- I think I've got all the common metacharacters auto-escaped in the code, so if you find you have to do it yourself, please let me know at <tools@occam.com>


Command Mode
% sudo rshall -c 'ext dns' free
ns-ext3
total used free shared buffers cached
Mem: 8165304 708092 7457212 0 110700 448328
-/+ buffers/cache: 149064 8016240
Swap: 4192924 0 4192924
ns-ext1
total used free shared buffers cached
Mem: 4046468 1387168 2659300 0 444692 384964
-/+ buffers/cache: 557512 3488956
Swap: 4192924 0 4192924
ns-ext2
total used free shared buffers cached
Mem: 824640 590400 234240 0 93300 381204
-/+ buffers/cache: 115896 708744
Swap: 2048276 0 2048276
Simple information gathering
31


Command Mode
% sudo rshall -c 'dp web' 'ls -l /usr/local/apache/logs/access_log.* | head -3'
www701
-rw-r--r-- 1 root root 4201042 Mar 23 00:59 /usr/local/apache/logs/access_log.20100322
-rw-r--r-- 1 root root 4178862 Mar 24 00:59 /usr/local/apache/logs/access_log.20100323
-rw-r--r-- 1 root root 4189388 Mar 25 00:59 /usr/local/apache/logs/access_log.20100324
www700
-rw-r--r-- 1 root root 4309697 Mar 23 00:59 /usr/local/apache/logs/access_log.20100322
-rw-r--r-- 1 root root 4180590 Mar 24 00:59 /usr/local/apache/logs/access_log.20100323
-rw-r--r-- 1 root root 4199574 Mar 25 00:59 /usr/local/apache/logs/access_log.20100324
Single quotes with metacharacters
32

Command Mode

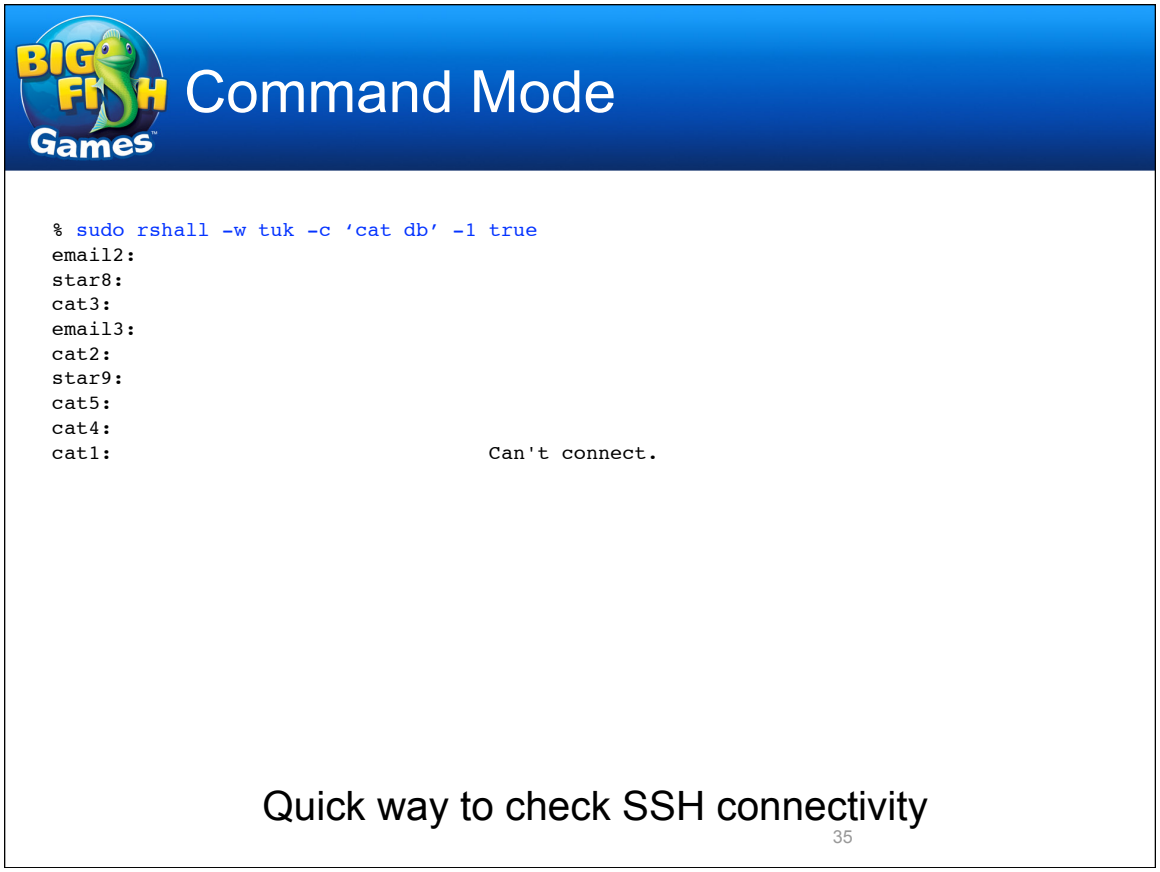
```
% sudo rshall -c 'dp web' 'cat /usr/local/apache/logs/access_log.20100422 | cut -d" " -
f2 | grep -v ^10. | sort | uniq -c | sort -rn | head -5'
##### www700 #####
    223 210.34.0.170
     40 97.73.64.150
     35 24.236.92.33
     33 98.207.86.165
     30 71.224.86.219

##### www701 #####
    256 210.34.0.170
     43 98.253.6.18
     30 99.188.204.122
     28 71.107.192.30
     28 68.5.88.127
```

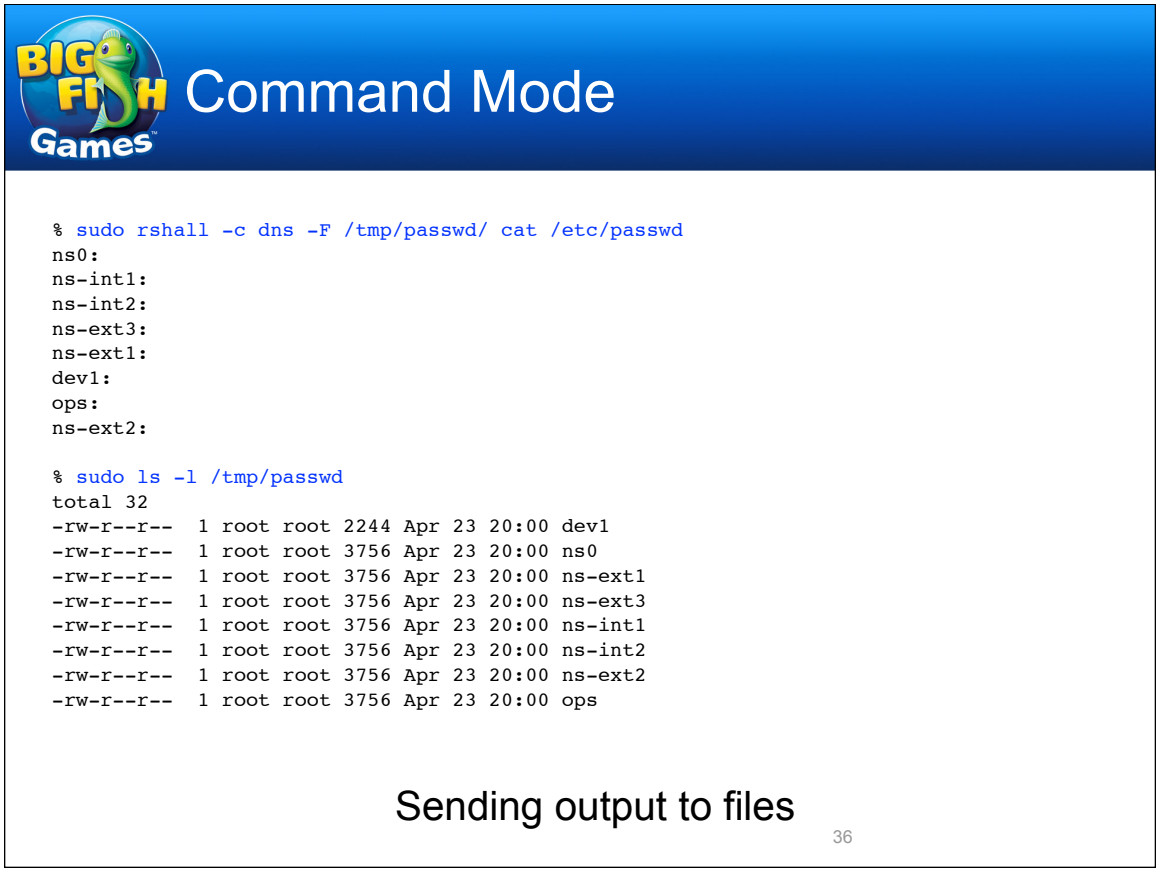
Complex pipeline

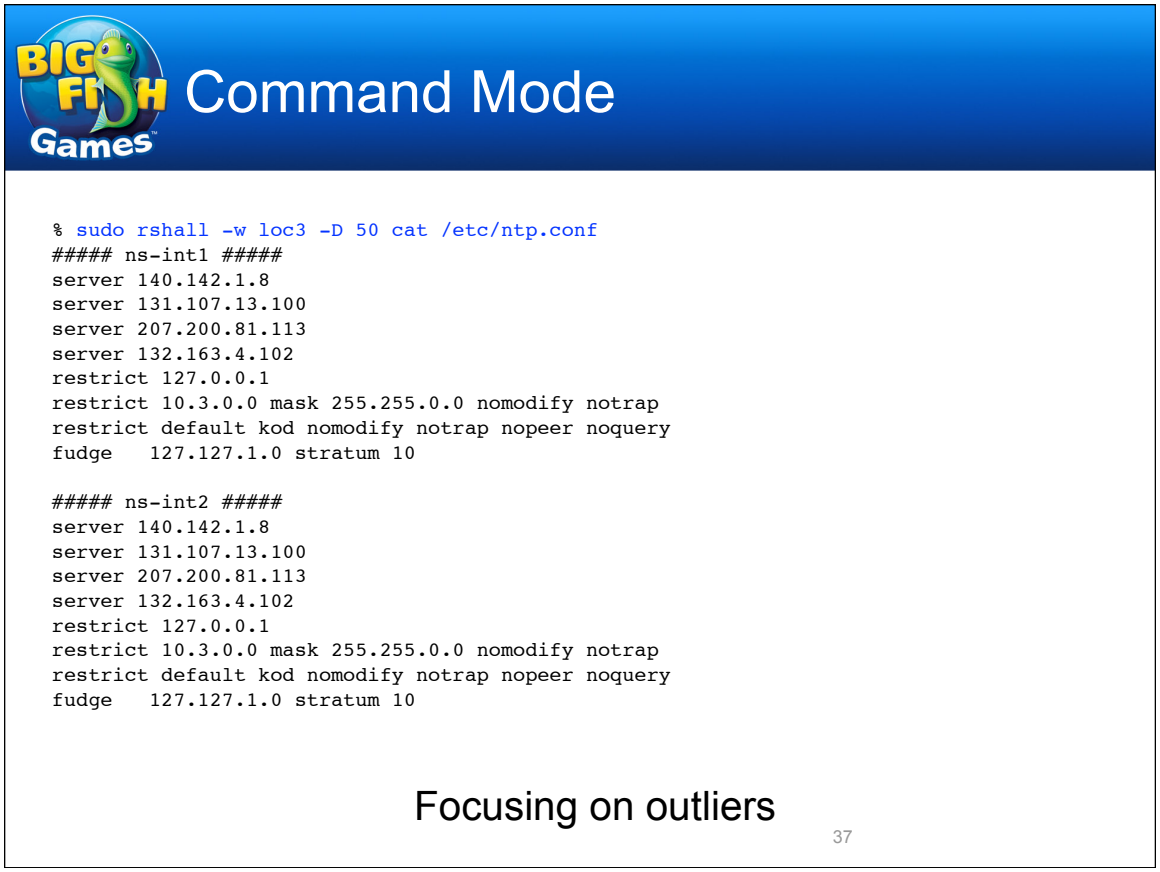
Command Mode

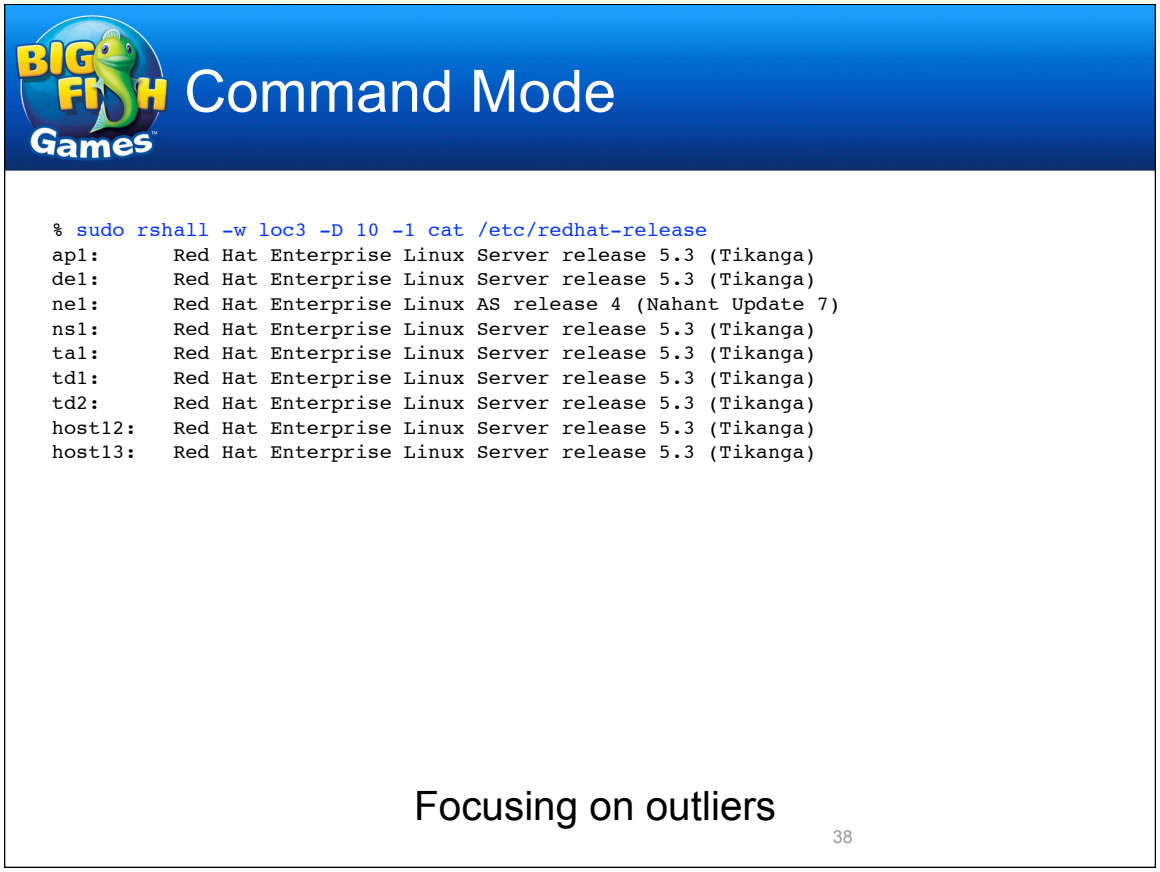
```
% sudo rshall -c dns -1 uptime
ns0:            16:22:26 up 230 days, 17:40, 20 users,  load average: 0.34, 0.34, 0.28
ns-ext3:        16:22:26 up 229 days,  4:35,  0 users,  load average: 0.07, 0.02, 0.00
ns-int1:        16:22:26 up 209 days,  3:18,  0 users,  load average: 0.00, 0.00, 0.00
ns-int2:        16:22:26 up 229 days,  4:29,  0 users,  load average: 0.00, 0.00, 0.00
ns-ext1:        16:22:26 up 230 days, 17:30,  0 users,  load average: 0.00, 0.00, 0.00
ops:            16:22:26 up 104 days, 18:39,  0 users,  load average: 0.02, 0.04, 0.02
dev1:           16:22:26 up 2 days, 16:43, 72 users,  load average: 1.22, 0.90, 0.80
ns-ext2:        16:22:26 up 18 days,  1:16,  0 users,  load average: 0.00, 0.00, 0.00

% sudo rshall -c dns -1 'df -h / | tail -1'
ns0:              /dev/cciss/c0d0p1     6.9G  3.1G  3.6G  46% /
ns-ext3:          /dev/sda1             6.9G  2.5G  4.2G  38% /
ns-int1:          /dev/cciss/c0d0p1     6.9G  3.9G  2.7G  60% /
ns-int2:          /dev/cciss/c0d0p1     6.9G  319M  6.3G   5% /
ns-ext1:          /dev/cciss/c0d0p1     6.8G  412M  6.1G   7% /
ops:              /dev/cciss/c0d0p1     6.8G  888M  5.6G  14% /
dev1:             /dev/cciss/c0d0p1     6.9G  4.5G  2.1G  69% /
ns-ext2:          /dev/sda1             4.0G  1.9G  1.9G  51% /

% sudo rshall -c 'dp web' -1 'ps aux | grep httpd | grep -v grep | wc -l'
www701:            38
www700:            41
```

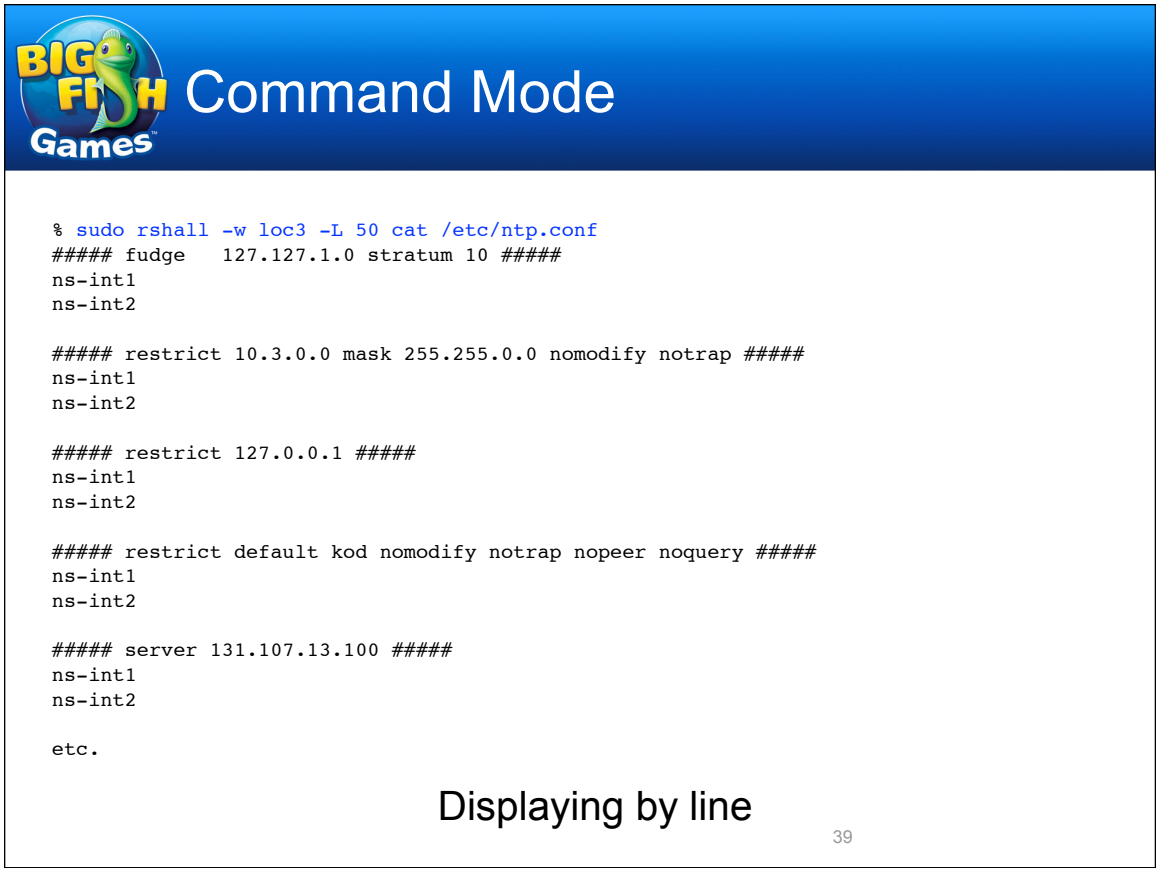
Single-line output format

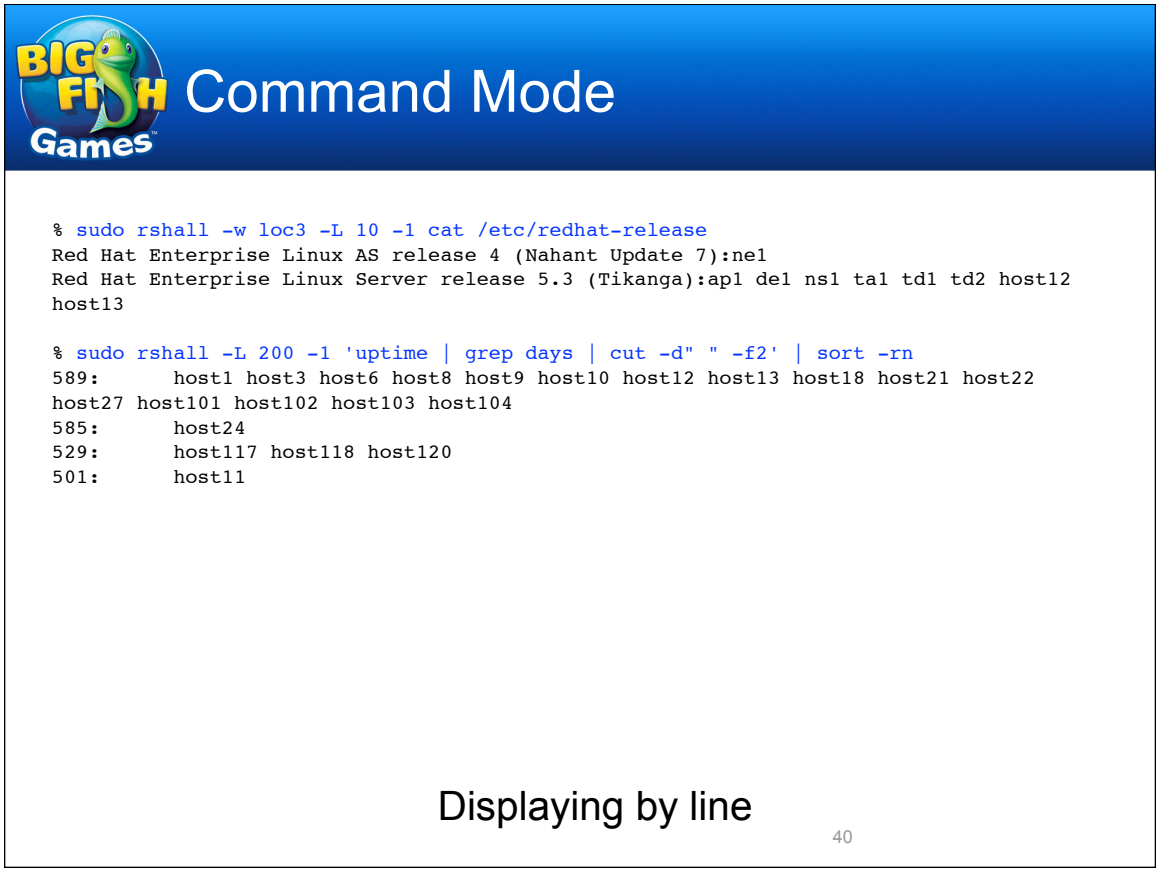

Command Mode
% sudo rshall -w tuk -c 'cat db' -l true
email2:
star8:
cat3:
email3:
cat2:
star9:
cat5:
cat4:
cat1: Can't connect.
Quick way to check SSH connectivity


Command Mode
% sudo rshall -c dns -F /tmp/passwd/ cat /etc/passwd
ns0:
ns-int1:
ns-int2:
ns-ext3:
ns-ext1:
dev1:
ops:
ns-ext2:
% sudo ls -l /tmp/passwd
total 32
-rw-r--r-- 1 root root 2244 Apr 23 20:00 dev1
-rw-r--r-- 1 root root 3756 Apr 23 20:00 ns0
-rw-r--r-- 1 root root 3756 Apr 23 20:00 ns-ext1
-rw-r--r-- 1 root root 3756 Apr 23 20:00 ns-ext3
-rw-r--r-- 1 root root 3756 Apr 23 20:00 ns-int1
-rw-r--r-- 1 root root 3756 Apr 23 20:00 ns-int2
-rw-r--r-- 1 root root 3756 Apr 23 20:00 ns-ext2
-rw-r--r-- 1 root root 3756 Apr 23 20:00 ops
Sending output to files


Command Mode
% sudo rshall -w loc3 -D 50 cat /etc/ntp.conf
ns-int1
server 140.142.1.8
server 131.107.13.100
server 207.200.81.113
server 132.163.4.102
restrict 127.0.0.1
restrict 10.3.0.0 mask 255.255.0.0 nomodify notrap
restrict default kod nomodify notrap nopeer noquery
fudge 127.127.1.0 stratum 10
ns-int2
server 140.142.1.8
server 131.107.13.100
server 207.200.81.113
server 132.163.4.102
restrict 127.0.0.1
restrict 10.3.0.0 mask 255.255.0.0 nomodify notrap
restrict default kod nomodify notrap nopeer noquery
fudge 127.127.1.0 stratum 10
Focusing on outliers
37


Command Mode
% sudo rshall -w loc3 -D 10 -1 cat /etc/redhat-release
ap1: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
de1: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
ne1: Red Hat Enterprise Linux AS release 4 (Nahant Update 7)
ns1: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
ta1: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
td1: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
td2: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
host12: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
host13: Red Hat Enterprise Linux Server release 5.3 (Tikanga)
Focusing on outliers
38


Command Mode
% sudo rshall -w loc3 -L 50 cat /etc/ntp.conf
fudge 127.127.1.0 stratum 10
ns-int1
ns-int2
restrict 10.3.0.0 mask 255.255.0.0 nomodify notrap
ns-int1
ns-int2
restrict 127.0.0.1
ns-int1
ns-int2
restrict default kod nomodify notrap nopeer noquery
ns-int1
ns-int2
server 131.107.13.100
ns-int1
ns-int2
etc.
Displaying by line
39


Command Mode
% sudo rshall -w loc3 -L 10 -1 cat /etc/redhat-release
Red Hat Enterprise Linux AS release 4 (Nahant Update 7):ne1
Red Hat Enterprise Linux Server release 5.3 (Tikanga):ap1 de1 ns1 ta1 td1 td2 host12
host13
% sudo rshall -L 200 -1 'uptime | grep days | cut -d" " -f2' | sort -rn
589: host1 host3 host6 host8 host9 host10 host12 host13 host18 host21 host22
host27 host101 host102 host103 host104
585: host24
529: host117 host118 host120
501: host11
Displaying by line
40

Command Mode

```
% sudo rshall -c 'int dns' -1 chown named:named /etc/named.conf
ns0:
ns-int1:
ns-int2:
dev1:
ops:

% sudo rshall -c 'int dns' -1 'sed s/named/bind/ /etc/passwd > /etc/passwd.tmp; grep
bind /etc/passwd.tmp'
ns0:            bind:x:25:25::/etc/namedb:/sbin/nologin
ns-int2:        bind:x:25:25::/etc/namedb:/sbin/nologin
ns-int1:        bind:x:25:25::/etc/namedb:/sbin/nologin
dev1.:          bind:x:25:25::/etc/namedb:/sbin/nologin
ops:            bind:x:25:25::/etc/namedb:/sbin/nologin
% sudo rshall -c 'int dns' -1 mv /etc/passwd.tmp /etc/passwd
ns0:
ns-int1:
ns-int2:
dev1:
ops:
```

Making changes - Be careful!

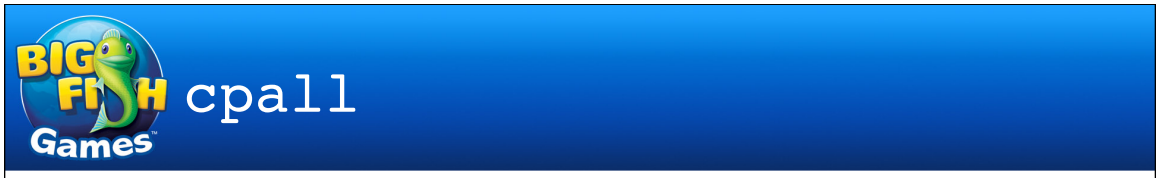
Parallel Copies: `cpall`

→When you install `rshall`, a symlink is created

```
lrwxrwxrwx  1 root root 6 May  5  2008 /usr/local/bin/cpall -> rshall
```

→When invoked as `cpall`, the behavior of `rshall` changes

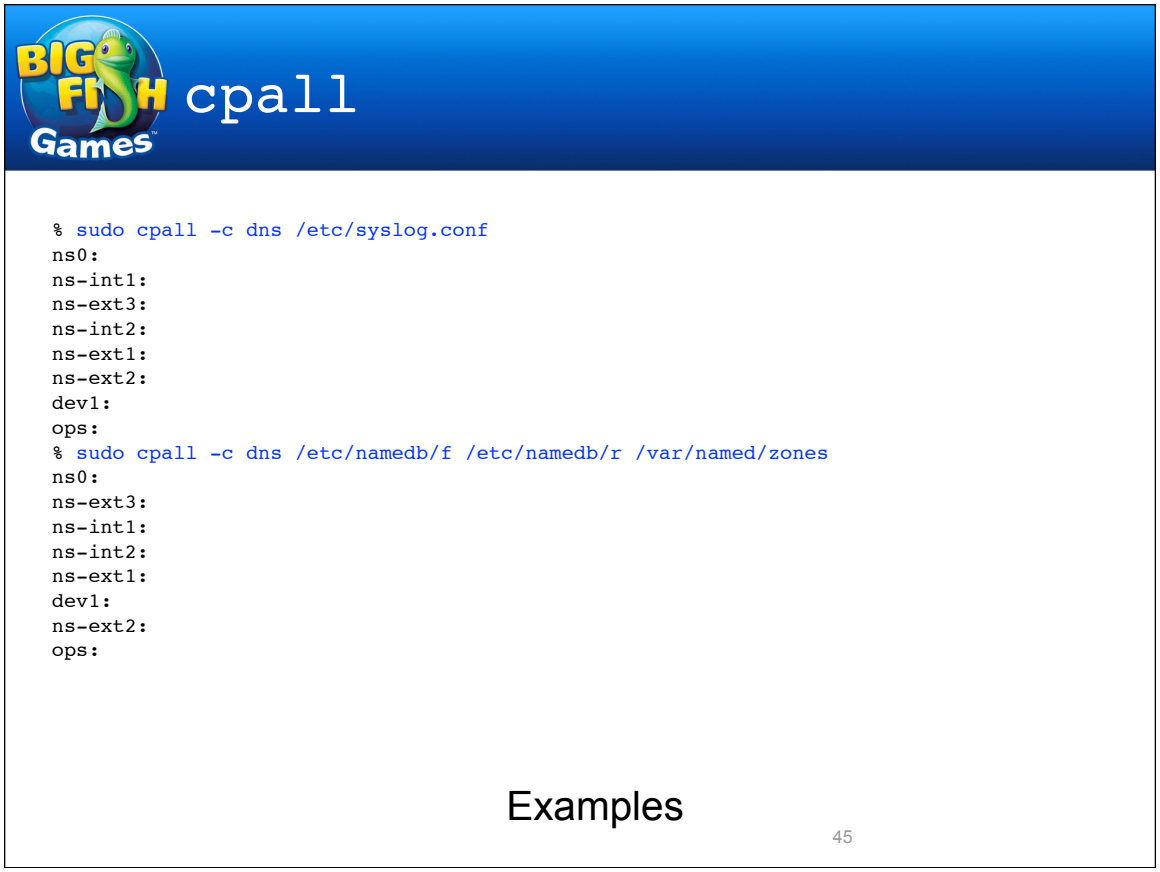
- →Most of the options, including match options, remain the same
- →Instead of a command, `cpall` takes one or more pathnames as arguments
- →`cpall -h` displays the usage statement



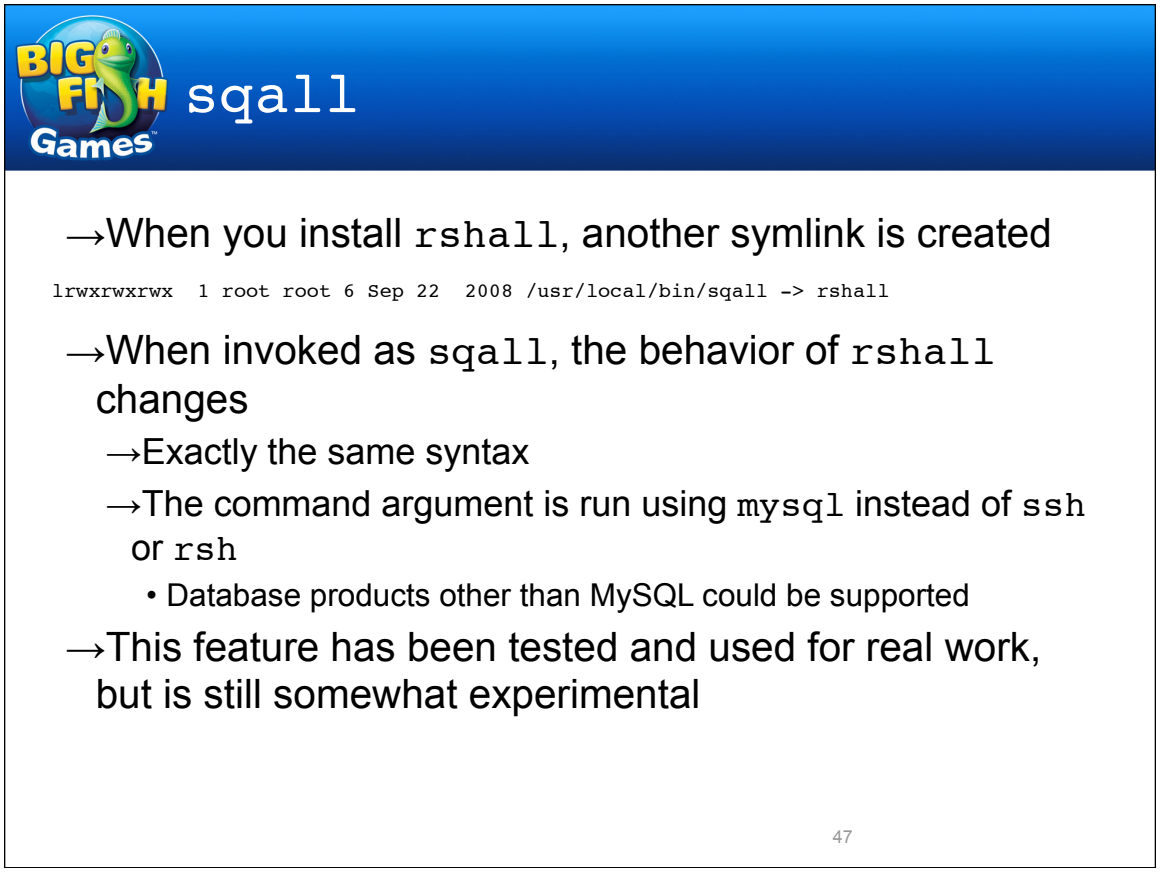
→`cpall` copies files to multiple hosts in parallel

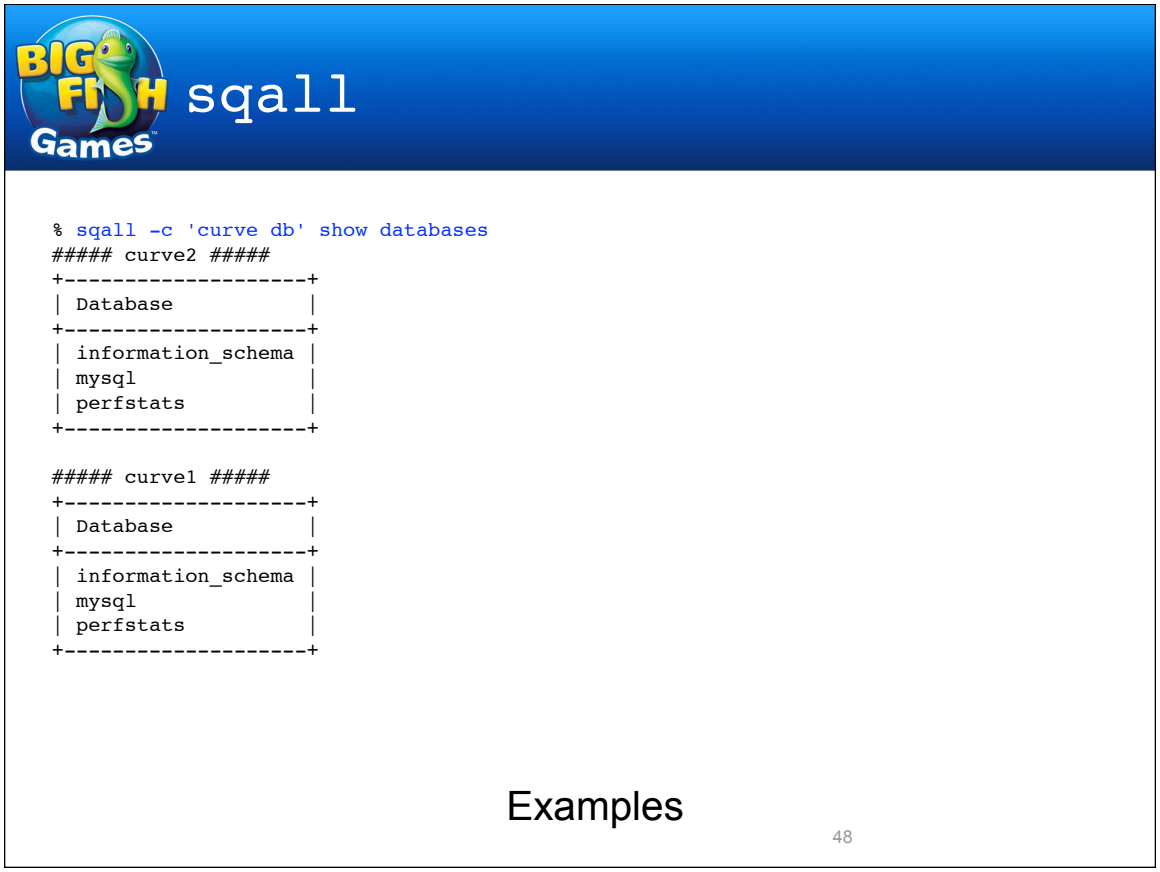
- →If given one pathname as an argument, copies that file or directory from the trusted host to the same location on the remote hosts
- →If given more than one pathname argument, the final argument is the destination on each remote host, and the files or directories specified by all the previous arguments are copied to there

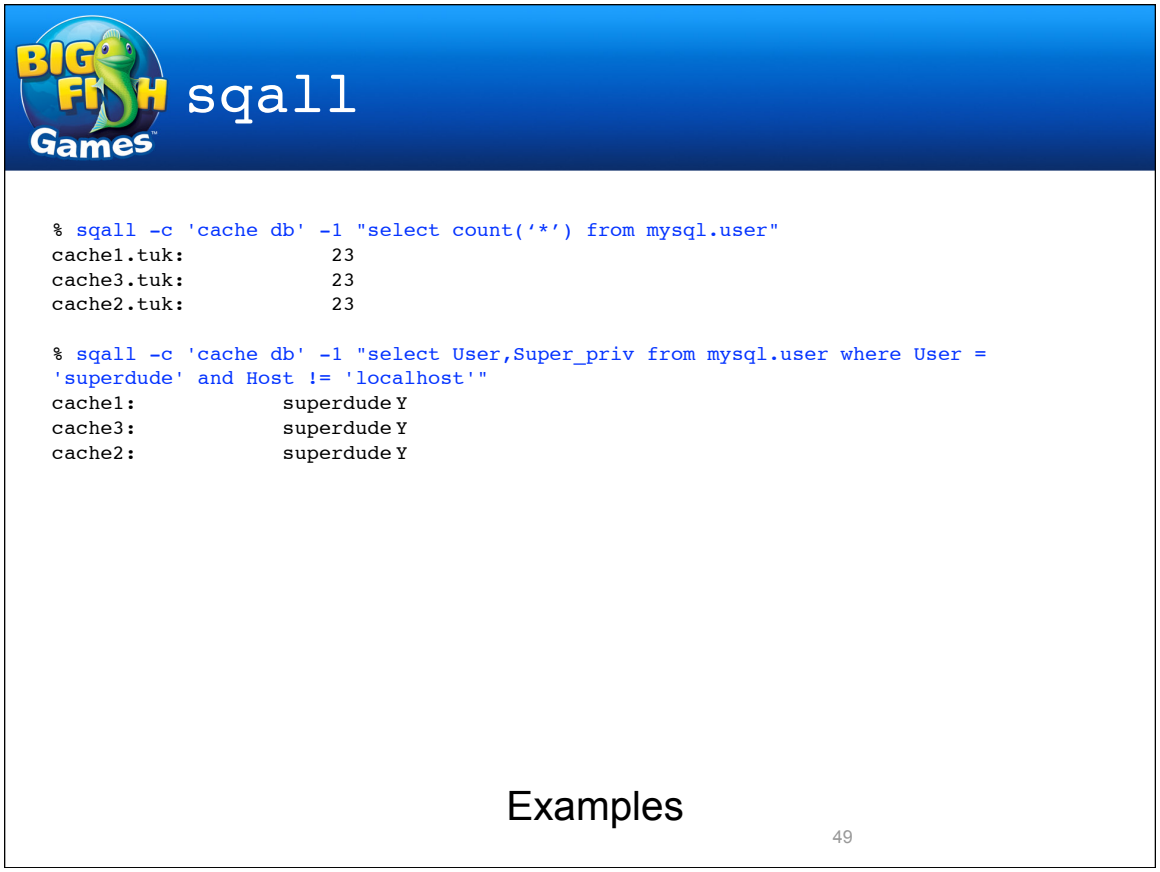
→This is no replacement for a good configuration management system, but it can be useful for one-time file distributions


cpall
% sudo cpall -c dns /etc/syslog.conf
ns0:
ns-int1:
ns-ext3:
ns-int2:
ns-ext1:
ns-ext2:
dev1:
ops:
% sudo cpall -c dns /etc/namedb/f /etc/namedb/r /var/named/zones
ns0:
ns-ext3:
ns-int1:
ns-int2:
ns-ext1:
dev1:
ns-ext2:
ops:
Examples
45


Parallel SQL: sqall


sqall
→When you install rshall, another symlink is created
lrwxrwxrwx 1 root root 6 Sep 22 2008 /usr/local/bin/sqall -> rshall
→When invoked as sqall, the behavior of rshall changes
→Exactly the same syntax
→The command argument is run using mysql instead of ssh or rsh
• Database products other than MySQL could be supported
→This feature has been tested and used for real work, but is still somewhat experimental
47


sqall
% sqall -c 'curve db' show databases
curve2
+--------------------+
| Database |
+--------------------+
| information_schema |
| mysql |
| perfstats |
+--------------------+
curve1
+--------------------+
| Database |
+--------------------+
| information_schema |
| mysql |
| perfstats |
+--------------------+
Examples
48


sqall
% sqall -c 'cache db' -1 "select count(‘*’) from mysql.user"
cache1.tuk: 23
cache3.tuk: 23
cache2.tuk: 23
% sqall -c 'cache db' -1 "select User,Super_priv from mysql.user where User =
'superdude' and Host != 'localhost'"
cache1: superdude Y
cache3: superdude Y
cache2: superdude Y
Examples
49


Conclusion

Conclusion

- →`rshall` is a huge productivity enhancer
  - →Makes managing hundreds of systems almost as easy as managing a handful
  - →Easy to gather information from all your hosts at once
- →Setting up `rshall` motivates collection of information about your hosts that you may not already have, improving your support infrastructure
  - →And provides a simple way to get at that information
- →Available at:
  - →http://www.occam.com/tools/

`rshall`: A Tool for Managing Hosts in Parallel

Leon Towns-von Stauber
LinuxFest Northwest, April 2012
http://www.occam.com/