Inside the SGE-SSH Integration

Rayson Ho, Ron Chen

Sept 11, 2007

SGE-SSH Loose Integration

Advantages of using ssh over qrsh (copied from the HOWTO):

- secure connection
- no need to have setuid root programs installed
- much larger no. of running sessions per host
- compression
- supports attaching a TTY to remotely executed commands
- x-forwarding

Limitations of the original Integration

SGE does not know the process IDs of the job -- which means:

- lack of complete accounting data
- lack of dynamic re-prioritization
- potential loss of control

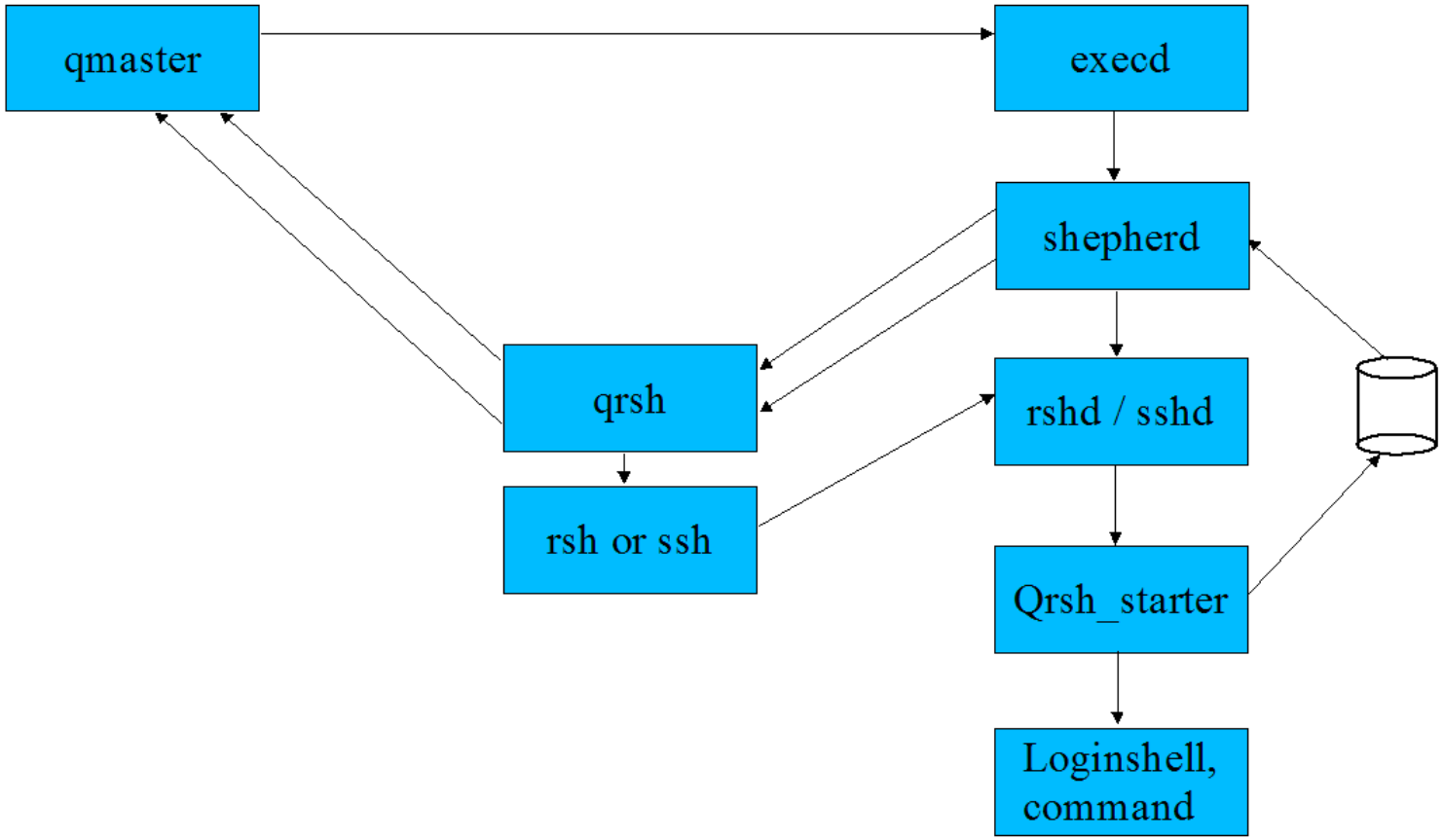
Job execution: rshd vs. sshd

execd starts shepherd which starts either rshd or sshd

- rshd was modified which sets an additional Group ID when switching the user ID of job's owner

- with a vanilla sshd, the additional Group ID is not set when switching from root to the job owner's user ID

Job execution: process flow



SGE just launches the commands defined by rsh_daemon, whether it be rshd, vanilla sshd, or sshd w/ tight-integration

Tight SGE-SSH Integration (1)

Work began in 2003, and was available as a patch in 2004

- First version works with 3.7.1p2
- Need to patch openssh, configure and build
- Then link against a number of SGE object files

Use the sshd binary instead of the standard one for:
rsh_daemon, rlogin_daemon and qlogin_daemon

Was not included in the official source tree till SGE 6.1

- it was hard to build with earlier versions
- further changes in 6.1 made the build process much easier!!

Tight SGE-SSH Integration (2)

1) download SGE 6.1 or later
2) aimk to build all the SGE libraries and binaries
3) cd 3rdparty/
4) download openssh source tarball
5) gunzip -c openssh-<version>.tar.gz | tar vxf –
6) ln -s openssh-<version> openssh
7) patch sshd.c
8) Build the tight integration with: aimk -tight-ssh

Patching sshd.c

- in main():

```
init_rng();
#ifdef SGESSH_INTEGRATION
  sgessh_readconfig();
#endif
```

- in privsep_postauth():

```
/* Drop privileges */
#ifdef SGESSH_INTEGRATION
  sgessh_do_setusercontext(authctxt->pw);
#else
 do_setusercontext(authctxt->pw);
#endif
```

Tight Integration in Production

- Since the first release of 6.1, the tight integration has been tested and put into production use by a no. of sites
- Including the TSUBAME supercomputer:

 - the most powerful supercomputer in Asia!!

 - documented in “Sun N1 Grid Engine Software and the Tokyo Institute of Technology Super Computer Grid”

Inside the Tight Integration

Internals (1)

In sshd's main(), before sshd changes the working directory, sgessh_read_config() reads the job config file using SGE's internal routine, and saves the supplemental group ID and the start directory.

Internals (2)

- In privsep_postauth(), sshd calls do_setusercontext() to drop privileges
- For vanilla sshd, do_setusercontext() drops everything and only adds the group ID from the “passwd” struct
- In the integration, we added a routine sgessh_do_setusercontext() to switch to the job owner’s ID and add supplemental group ID

sgessh_do_setusercontext()

- This routine is the kernel of the integration
- Most of the code was extracted from SGE's rshd
- For platforms with OS level job support, sets the OS Job ID
- Uses the SGE way to switch user context, and thus keeps the additional group ID assigned by SGE

Internals (3)

- A unique additional group ID is assigned by SGE to each job
- In execd, the PDC scans all the processes running on the system.
- Each new process created by the job has the additional group ID inherited
- Thus execd knows which process belongs to which job

Testing the Integration

- sshd should always be started by shepherd
- The additional group ID should be attached:

```
% qrsh id
uid=3001(sgeuser)gid=10(staff)group
  s=10(staff),20090
```

Future directions

- Sun developers are working on a new remote client that is similar to qrsh but supports TTY
- We (opensource developers) are enhancing execd to use parent–child relationship to keep track of process ownership
- Still need someone (anyone? ☺) to get the sshd.c changes to the official openssh tree

Further readings

1) “Sun N1 Grid Engine Software and the Tokyo Institute of Technology Super Computer Grid”

http://www.sun.com/blueprints/0607/820-1695.html

2) 3rdparty/remote/sgessh.c

http://gridengine.sunsource.net/source/browse/*checkout*/gridengine/source/3rdparty/remote/sgessh.c

3) SSH, The Secure Shell: The Definitive Guide (published by O'Reilly)