

Test specification for bug CR 6293411

NFS write error on host <NFS server>: Permission denied.

Bug description

When spooling on a NFS mount mounted with root=rw option, an admin user system and job user != admin user, not all lines were written to the trace file and a error message was written to /var/adm/messages.

The error message was:

Aug 5 10:22:31 eomer unix: NFS write error on host es-ergb01-01: Permission denied.

Aug 5 10:22:31 eomer unix: (file handle: 8002d8 2 a0000 174a76 708e925b a0000 2 2a1f5264)

The trace file was created by root during the start of sge_shepherd, was then chown()ed to the job user, because both the parent shepherd (running under root account) and the child shepherd (running under job users account) have to write something to the trace file. As root has the permissions to write to all files (root=rw), we handed the file over to the job user who hence was able to write to the file, too.

In the parent shepherd, running under the root account, all file operations were done with euid=admin user id. This was done because in the case that root has no special permissions (root=ro), it would not have been possible to write any file.

This means: In this special case of spooling (which was thought to be exactly the same as local spooling), files were opened by root and the admin user wrote to the files. But NFS has a problem here, the file permissions are checked again before writing, the admin user doesn't have the necessary permissions, so the line was not written and an error message was added to "/var/adm/messages".

Bug fix

Do not only open the files as root, also write to the files and close them as root.

Testsuite test

Test cases

- Spooling: local, NFS root=rw, NFS root=ro
- All architectures
- Jobs started by: qsub, qrsh, qrsh <param>, qlogin, qsh, tight integrated jobs
- root system, admin user system, normal user system

Pseudo-Code

```
For System in [root, admin_user, normal_user] do {
```

```

Setup_System as System
Set execd_param keep_active=true
For Spooldir in [local, NFS-rw, NFS-ro] do {
  Set execd_spooldir to Spooldir
  Restart execd
  For Architecture in [<all architectures>] do {
    For Test in [qsub, qrsh, qrsh<param>, qlogin, qsh, tight-integrated_job] do {
      Execute Test on Architecture
      Analyze Test_result
      If Test_result = Bad {
        Stop test!!!
      }
    }
  }
}
}
}

```

Test criterias

- Compare the files *trace*, *exit_status* and *error*
- After the test, the files must be owned by the job user (check uid and gid!)
- After a successful test, the *error* file must be empty, the *exit_status* file must contain "0".
- After an intended unsuccessful test, the *error* file must contain the error string and the *exit_status* file must contain the exit status.

- After a successful test, the *trace* file must contain at least these lines:

```

07/26/2005 15:18:20 [151085:19288]: shepherd called with uid = 0, euid = 151085
07/26/2005 15:18:20 [151085:19289]: setting limits
07/26/2005 15:18:20 [151085:19289]: now doing chown(hp150085) of trace and error files
07/26/2005 15:18:20 [151085:19289]: switching to intermediate/target user
07/26/2005 15:18:20 [151085:19289]: now running with uid=151085, euid=151085
07/26/2005 15:18:20 [151085:19289]: execvp(/bin/sh, "-sh"
"/tmp/hp150085/60u5/eomer/job_scripts/14" "5")
07/26/2005 15:18:25 [151085:19288]: wait3 returned 19289 (status: 0; WIFSIGNALED: 0,
WIFEXITED: 1, WEXITSTATUS: 0)
07/26/2005 15:18:25 [151085:19288]: job exited with exit status 0

```