

Re: program to copy files – problems – unix ksh to java

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2004-02/1055.html>

From: Harald Kirsch (pifpafpuf_at_gmx.de)

Date: 02/10/04

Date: 10 Feb 2004 01:25:18 -0800

kaeli <tiny_one@NOSPAM.comcast.net> wrote in message
news:<MPG.1a9172689199094e989bf1@nntp.lucent.com>...

- > *Cron runs this code every 5 minutes.*
- > *Program looks on one machine, uses ssh to copy a file to another*
- > *machine, changes the filename, the owner and permissions (chmod), and*
- > *then deletes the file from the source machine.*
- > *Sounds simple enough...*
- >
- > *Problems:*
- > *Large files (we're talking gigabytes) take more than 5 minutes to copy.*
- > *Something is causing the file to be deleted before it has finished*
- > *copying. We lose the whole file, as it doesn't show up on either*
- > *machine.*

Hard to believe on a *NIX machine as the system allows to delete files held open by other processes. It may be a possible problem if the file is on NFS. In any case it sounds like the first process, when finished, deletes the file, while the 2nd copy process, started while the first was still running, then gets in trouble and messes things up.

- > *Program gets called again while an instance is running, so it tries to*
- > *copy files that are currently being copied.*

A solution might be to rename the file locally **before** copying. This way the process starting 5 minutes later will not pick up the same file again. If this is not an option, create an empty file with another extension than the big file as a mark that the file is being worked on.

- > *Also, how can I make sure that the copy was finished before deleting the*
- > *source? I expected the script to wait for the copy to finish before*
- > *deleting, but it appears that it is not doing that.*

Assuming that you do **not** start the copy process in the background (&), the script does wait. You have to

comp.lang.java.programmer: Re: program to copy files – problems – unix ksh to java

look for a different reason why the file is deleted too early, maybe as I suggested above.

There is no need to solve this task in Java.

Harald.