IF Data Handling Client Tools (ifdhc) - Feature #24556

ifdh cp should behave like posix cp

06/22/2020 02:29 PM - Alexander Himmel

Status:	New	**Start date:**	06/22/2020
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

This has been a long-time frustration, but was inspired by a recent incident of a user needing to request help from experts in the DUNE slack in order to copy files between dCache volumes.

ifdh cp presents significant challenges for new users because of it presents itself as behaving like cp, and does sometimes, but not all the time. There are two specific problems which I think come up frequently:

ifdh cp <file> <directory> doesn't work

ifdh cp -r <directory> <new location> sometimes doesn't work depending on the details

It's totally understandable to globus-url-copy or whatever else is working on the backend doesn't have posix behavior, but they also don't present themselves as having those features in the way ifdh cp does. If the underlying tools, for good reasons, can't support the above, it seems like some relatively simple scripting inside ifdh before passing along to external tools could eliminate a lot of useless user frustration and calls for help to experiment experts.

History

#1 - 06/22/2020 04:32 PM - Marc Mengel

This was tried a long time ago; the problem is doing a directory listing with every copy
to see if the destination is a directory or not **kills** the DCache namespace service.
Even having individual experiments do a directory listing per copy has severely impaired
DCache namespace performance.

The other possible implementation is to first assume it's a directory, and retry assuming not if that initial copy fails. This is also going to create a fair amount of extra load
for those failed copy attempts; although probably less than the directory listing approach.

If we assume the destination-is-directory choice is false about half of the time, it seems a bad design choice to impose a 20% or more load increase on the fileservers to avoid a user education issue -- pass the -D flag when the destination is a directory.

#2 - 06/23/2020 09:36 AM - Alexander Himmel

Are these checks cheaper if the disk is NFS-mounted? Could the checks be specific to that circumstance?

#3 - 06/23/2020 09:37 AM - Alexander Himmel

FWIW, returning an informative error message would help, too.

The recursive copies seems like a different issue, though, right?

#4 - 06/23/2020 10:09 AM - Robert Illingworth

The basic problem is that dCache doesn't really behave like a conventional filesystem. A lot of issues with the NFS interface stem from that. So using the NFS mount doesn't really fix anything in that respect.

I'm afraid that POSIX is likely to disappear in the long run as an interface to bulk data - it's just not efficient for a lot of operations. Trying to fake it too closely is doing people a disservice - it will just cause confusion.

#5 - 06/25/2020 02:40 PM - Marc Mengel

Okay, so I have a draft of an improved error message in the code; basically if you do a copy which gives an error message involving "directory" or "exists", we give an error message about possibly missing a -D option; although on reflection this also may want the unlink_on_error stuff mentioned as well...

As to the ifdh cp -r vs cp -r, do you have any specific examples of differences you would like addressed?