

Re: Sort files by filename

Source: <http://linux.derkeiler.com/Mailing-Lists/Fedora/2007-07/msg05267.html>

- *From:* Les <hlhowell@xxxxxxxxxxxx>
 - *Date:* Tue, 31 Jul 2007 14:23:03 -0700
-

On Tue, 2007-07-31 at 15:12 -0400, Mark Haney wrote:

Les Mikesell wrote:

Is there a reason why rsync cannot be used for this?

Unfortunately, yes, due to the method that I receive the files, ie from another application that has it's own mechanism to feed the files to client machines. I really wish this wasn't the case, but I have to live with what I got.

If you request the resends with http you could use wget with the option to only transfer if the server's copy is newer than yours, and just ask for all of them every time.

Or, if you can construct the (sorted)list of all the names you expect to have you can:

```
ls * | comm -13 - /path/to/list
```

and get the list of names in the list but not in the directory.

With apologies to the 2 Les', the situation isn't like that and I apologize if I've not been clear. The application that I'm working with is running on a server that simply relays the data to all our customers, it doesn't store a copy of the files and then feed them. The

Re: Sort files by filename

NWS weather data requires as close to real-time performance and the 'series of tubes' allows. That said, I'm running another server that runs the same application but is designed to pull the data feed and then store the files locally. I /can/ store the files on the primary server, and I have, but this is a production server that feeds 13MB/hr for each of the 60 or so radar sites it handles 24/7 so I don't like asking it to do more than it does.

So, in essence I'm stuck with these files being dumped on a server via a proprietary method. So I need to sort the files and check for missing ones on the filesystem.

Sort will give you the list. I don't know about sorting on a substring with a command other than creating one. In C, you could read the directory, then choose the substring using parsing, and finally look at the last two characters prior to the period to get the sequence and look for missing files. Do you know the first number and last number? If not, then this won't work, because the first file and last file would not have partners on each side to help you figure out if it was missing.

I would then guess that the sourcing application is using a stream, and if so, then you may be able to "T" the stream to get some information from it. However, no matter what method you choose you won't know about the first and last without some indication from the source about what those files' index numbers would be.

This is not a simple matter. I would normally suggest that you approach the original vendor to see if they are checking that the files are opened correctly. There may be a problem where the files are not properly opened, or a queuing issue that makes them appear out of order, and if that is the case, how are they dealing with that? In other words, how do you know if the file really exists? Especially the first and last.

Handling it with a bash script means the files have to exist already, anyway, so that is not the limitation of the rsync method. And if the issue is "realtime", the networking delays are a problem anyway unless the files are being sent by a VPN type architecture where the routing is consistent. Otherwise routing delays could cause you additional problems. In addition, how do you check the data for security? Is it encrypted, compressed or tokenized in some way with checksums and so forth? I know that these questions appear out of order to the question you asked, but they deal with how the data is being handled, and that in turn deals with the issues of delays and file appearance scheduling. Which in turn affects how you might choose to access them with the least delay and overhead.

The early suggestions were great and I'm trying each one and tweaking to

Re: Sort files by filename

see if I can make them work with what I have. But any additional bash tips would be helpful as I am pressed for an answer to this issue.

I primarily code in C and do not use bash or perl much because the overhead of scripts was too great for the applications I was working on. Remote files have a whole different set of issues, from where they are located, to the routing and delays as I discussed above, to how they are verified for completeness, and the sequence of appearance.

Using C, you could open the directory, sort the list, compare for a desired sequence from a starting value to an ending value and pass out a list of missing files, and it would take only milliseconds, primarily limited by disk access speed.

Regards,
Les H

--

fedora-list mailing list

fedora-list@xxxxxxxxxx

To unsubscribe: <https://www.redhat.com/mailman/listinfo/fedora-list>