

JobSub - Feature #22347

When we create tarfiles, compress them in place rather than making a tempfile and compressing it

04/10/2019 09:57 AM - Shreyas Bhat

Status:	New	Start date:	04/10/2019
Priority:	Normal	Due date:	
Assignee:	Shreyas Bhat	% Done:	0%
Category:	JobSub Client	Estimated time:	0.00 hour
Target version:	v1.3.1	Spent time:	0.00 hour
Stakeholders:			

Description

Liang Li from gm2 noted that in the gpvm \$TMPDIRs (usually /tmp), they're limited to 2 GB of space. Thus, if they try to create larger tarballs using the jobsub_client tardir:// feature, they can fill up tmp before the tarball is actually created.

This is because in client/jobsubClient.py, we create a temp file using tempfile.mktemp(), put the contents of the intended tarball in there, compress the tarball using `gzip -n`, and then move that file into place. Can we skip making the tempfile, and instead write directly to the final tarfile compressed using something like:

```
tar = tarfile.open(filename, 'w:gz')
```

Here is the original message from Liang:

```
I do have one comment, tardir:// option seems to first tar the designated directory and then compress it, all of which is done at $TMPDIR (normally "/tmp"), this is actually a problem for gm2 VM --- for some reason, /tmp space is limited to merely 2GB for all VMs (Adam and I are starting another discussion about that). This has caused problems when /tmp is filled up. Apparently /tmp can be easily filled up when tardir:// option is used (as I explained above, a tar ball is *first* created and then *compressed*). Of course, one can simply relocate $TMPDIR to circumvent that. But I just thought that it might be more convenient (and probably more efficient) for tardir:// option to act like a "tar cfz" command (which creates tar ball and compresses it at the same time).
```

History

#1 - 04/10/2019 09:57 AM - Shreyas Bhat

I wonder if we do this because we need to use `gzip -n` to ensure we ignore timestamps...

#2 - 04/15/2019 02:58 PM - Shreyas Bhat

Dennis and I discussed how this could be done. We decided to create a new flag called "--tar_output_dir" that users could use in conjunction with the tardir:// URI (to either the -f or the --tar_file_name flags) that would allow users to specify where they wanted the tarballs to be created and then compressed. This option would be passed to the os.temp_file call in the jobsubClient.py code where we actually create the tarball (create_tar or something like that?)

Poll uboone and gm2 to see if this name/behavior works for them.

#3 - 04/15/2019 02:59 PM - Shreyas Bhat

- Assignee changed from Parag Mhashilkar to Shreyas Bhat

#4 - 05/17/2019 11:40 AM - Dennis Box

- Target version set to v1.3.1