

GlideinWMS - Support #22509

Singularity processes left orphaned on PBS

05/03/2019 04:41 PM - Marco Mambelli

Status:	Closed	Start date:	05/03/2019
Priority:	Normal	Due date:	
Assignee:	Marco Mambelli	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	v3_6_2	Spent time:	0.00 hour
Stakeholders:			

Description

Singularity action-suid processes seems to be left orphaned

Hi Marco and Chris,

Thanks for the mail. I reconfigured pbs_mom on a few nodes with '\$exec_with_exec = true' to test. Jobs run just fine, although pilots from cern and pilots from ucsc have different process trees (see attachment). Not sure what that means, if anything. I wouldn't expect to see the extra bash process hanging in the ucsc case, but the PBS wrapper script that is typically seen (/var/spool/torque/mom_priv/jobs/<job id>.hammer-adm.rcac.purdue.edu.SC) isn't visible either.

I still see the issue where there is no delay between signals when I qdel a job. But, in my testing of both the ucsc and cern pilots, the glidein scripts and condor_master processes exit correctly and the /tmp/glide_XXXXX folder is cleaned up. That is some progress, but now I see the Singularity action-suid processes getting orphaned. Now that the condor logs are cleaned up, I can't get much information from the WN.

```
systemd├─action-suid─shim-init─condor_exec.exe─python2├─bash─cmsRun──3*[{cmsRun}]
│├─2*[action-suid─shim-init─condor_exec.exe─sh─python─bash─cmsRun──2*[{cmsRun}]
│]
│├─action-suid─shim-init─condor_exec.exe─python2├─bash─cmsRun──15*[{cmsRun}]
│├─action-suid─shim-init─condor_exec.exe─python2├─bash─cmsRun──11*[{cmsRun}]
│├─action-suid─shim-init─condor_exec.exe─python2├─bash─cmsRun──5*[{cmsRun}]
│└─python2
```

Any thoughts?

Thanks,
-Erik

This came out when investigating the shutdown of jobs running in PBS that should be terminated w/ a delay between sigterm and sigkill (instead of following one after the other):

Thanks Chris,
I'm adding in CC Erik, he's the site admin in Purdue that helped OSG identifying and troubleshooting the problem.

According to the documentation this seems to be the way to run scripts that listen to signals: pbs_mom uses the exec command to start the job script rather than the TORQUE default method, which is to pass the script's contents as the input to the shell. This means that if you trap signals in the job script, they will be trapped for the job. Using the default method, you would need to configure the shell to also trap the signals. Default is FALSE.

I don't know if there is any inconvenience/drawback in not having the shell

Erik,
can you check if the parameter in mom_priv/config suggested by Chris is solving the problem?

Thanks,
Marco

On Mar 21, 2019, at 2:25 PM, Christopher Larrieu <larrieu@jlab.org> wrote:

Hi Marco,

I had encountered a similar issue with over-resource termination with our PBS installation, and found that we needed to configure the MOM with the following parameter in mom_priv/config:

```
$exec_with_exec true
```

This needs to be done by your PBS administrators on all nodes. I imagine this would also fix your issue.

Here's a short description of what happens when you don't use this setting:

I've been looking into the way that PBS kills over-limit jobs, and have found that it doesn't allow the time for graceful shutdown that one might expect. The "normal" behavior as outlined in the source code is that the MOM will send a SIGTERM to the job processes, wait 45 seconds, send another SIGTERM, wait five seconds, then send a SIGKILL.

But there's a complicating factor in actuality that I have to assume is a bug. The default disposition of the forked pbs_mom process that subsequently forks the shell to run the user command is that it terminates on SIGTERM. Now there's some logic in the mother superior that looks for orphaned processes of children that have exited. Because the user job descends from the forked pbs_mom that exited, it will be sent a SIGKILL, meaning it will die before the loop iterates after what was supposed to have been a 45 second grace period.

This behavior makes it quite difficult to clean up over-limit jobs in a way that preserves vital information about what happened.

I hope this is helpful, and apologize that I am not really a PBS expert, so don't have much more to offer.

Chris

From: Marco Mambelli <marcom@fnal.gov>
Sent: Thursday, March 21, 2019 3:07 PM
To: Graham Heyes
Cc: Christopher Larrieu
Subject: Re: Question that you had after my presentation yesterday,

Thanks Graham,
it was me.

Chris,

I have a problem in PBS and was wondering if you experienced something similar.

A colleague in OSG has PBS and seems that torque is ignoring the settings to add a delay between signals when killing a job.

When killing a batch job with qdel sigterm is followed shortly after by a sigkill.

To allow jobs to shutdown cleanly we'd like to add a delay. This should be possible (according to manual) with -W option in qdel or the with the kill_delay configuration parameter.

None of the 2 seems to work with batch jobs.

regards,
Marco

Related issues:

Related to GlideinWMS - Bug #20202: Kill not handled properly by glidein_star...

Rejected

06/20/2018

History

#1 - 05/03/2019 05:04 PM - Marco Mambelli

I sent an email to Mascheroni and James to verify that the CMS singularity wrapper is doing an exec and not running singularity, otherwise, condor

could kill the script that is not propagating the signals.
Item will be discussed at the condor-Fermilab meeting

#2 - 05/10/2019 10:19 AM - Marco Mambelli

I checked the CMS wrapper (https://gitlab.cern.ch/CMSSI/CMSSglideinWMSValidation/blob/master/singularity_wrapper.sh) and discussed the issue with HTCondor developers.

Likely these jobs are using privileged singularity.
Setuid binaries even if they then drop privileges do not propagate received signals.
So condor is sending the signal and that is ignored.

The possible solutions are:

1. to run unprivileged singularity (requires EL7.6 and unprivileged user spaces enabled)
2. not to use a user_job_wrapper in condor (this way condor tracks better the children) but this is not an option at the moment because both CMS and OSG need the wrapper to invoke singularity

Asked Erik to run unprivileged singularity

#3 - 08/28/2019 09:59 AM - Marco Mambelli

- *Target version changed from v3_5_1 to v3_6_1*

#4 - 09/17/2019 11:15 PM - Marco Mambelli

- *Target version changed from v3_6_1 to v3_6_2*

#5 - 10/04/2019 05:10 PM - Marco Mambelli

- *Related to Bug #20202: Kill not handled properly by glidein_startup.sh added*

#6 - 03/03/2020 11:03 AM - Marco Mambelli

- *Status changed from New to Closed*

This should be solved by CMS migrating to the GlideinWMS wrapper. Testing is in progress, will happen soon.