

glideinWMS - Bug #19360

Critical bug in 3.2.21 leads to leaking glideFactoryEntryGroup.py processes

03/13/2018 08:09 AM - Marco Mascheroni

Status:	Closed	Start date:	03/13/2018
Priority:	Normal	Due date:	
Assignee:	Marco Mambelli	% Done:	0%
Category:	Factory	Estimated time:	0.00 hour
Target version:	v3_2_22	Spent time:	0.00 hour
First Occurred:	v3_2_21	Stakeholders:	
Occurs In:			

Description

Hi,

Jeff and I installed 3.2.21 on the production SDSC factory. Since then the factory started running out of memory if left untouched for about one day. I tracked down the reason to this exception [1], where it seems to me that one of the new ('GlideinsIdleCores', 'CoresIdle'), ('GlideinsRunningCores', 'CoresRunning'), ('GlideinsTotalCores', 'CoresTotal')) keys contains somehow strings instead of integers.

I am trying to fix this by replacing the line

```
el[ek] += (client_monitor[ck] * fraction)
```

with

```
el[ek] += (int(client_monitor[ck]) * fraction)
```

This should convert things like "4" to 4 allowing them to be multiplied by fraction which is a float.

However I am not 100% sure this is the correct solution, the string might still cause problems in other places, but I haven't been able to find out where it is set exactly. Maybe we should discuss this at the meeting.

[1] (found in /data/gwms-factory/log/server/factory/group_0.err.log)

```
[2018-03-13 04:35:20,105] WARNING: glideFactoryInterface:918: glidefactoryclient classad file /tmp/gfi_adm_gfc_470940784_517051 does not exist. Check if frontends are allowed to submit to entry
```

```
[2018-03-13 04:36:58,090] WARNING: cleanupSupport:23: Earlier cleanup PIDs [1761288, 1761290] still exist; skipping this cycle
```

```
[2018-03-13 04:37:18,096] DEBUG: glideFactoryEntryGroup:293: Setting parallel_workers limit of 24
```

```
[2018-03-13 04:49:59,978] WARNING: fork:55: Forked process '<function forked_check_and_perform_work at 0x26eacf8>' failed
```

```
[2018-03-13 04:49:59,978] ERROR: fork:56: Forked process '<function forked_check_and_perform_work at 0x26eacf8>' failed
```

Traceback (most recent call last):

```
File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 52, in fork_in_bg
```

```
out = function_torun(*args)
```

```
File "/usr/sbin/glideFactoryEntryGroup.py", line 216, in forked_check_and_perform_work
```

```
factory_in_downtime, entry, work[entry.name])
```

```
File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryEntry.py", line 1073, in check_and_perform_work
```

```
params, in_downtime, condorQ)
```

```
File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryEntry.py", line 1438, in unit_work_v3
```

```
max_glideins, work, log=entry.log, factoryConfig=entry.gflFactoryConfig
```

```
File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryLib.py", line 902, in logWorkRequest
```

```
work_el["internals"], fraction)
```

```
File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryMonitoring.py", line 364, in logClientMonitor
```

```
el[ek] += (client_monitor[ck] * fraction)
```

```
TypeError: can't multiply sequence by non-int of type 'float'
```

```
[2018-03-13 04:50:16,452] WARNING: glideFactoryEntryGroup:373: Error occurred while trying to find and do work.
```

```
[2018-03-13 04:50:16,453] ERROR: glideFactoryEntryGroup:374: Exception:
```

Traceback (most recent call last):

```
File "/usr/sbin/glideFactoryEntryGroup.py", line 371, in iterate_one
```

```
group_name, my_entries)
```

```
File "/usr/sbin/glideFactoryEntryGroup.py", line 306, in find_and_perform_work
post_work_info = forkm_obj.bounded_fork_and_collect(parallel_workers)
File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 310, in bounded_fork_and_collect
post_work_info_subset = fetch_ready_fork_result_list(pipe_ids)
File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 208, in fetch_ready_fork_result_list
out = fetch_fork_result(fd, pid)
File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 97, in fetch_fork_result
out = cPickle.loads(rin)
EOFError
[2018-03-13 04:50:16,453] DEBUG: glideFactoryEntryGroup:376: Group Work done: {}
```

History

#1 - 03/13/2018 08:15 AM - Marco Mascheroni

Here is the formatted exception, it is unreadable otherwise:

```
[2018-03-13 05:21:02,800] WARNING: fork:55: Forked process '<function forked_check_and_perform_work at 0x26eac
f8>' failed
[2018-03-13 05:21:02,800] ERROR: fork:56: Forked process '<function forked_check_and_perform_work at 0x26eac
f8>' failed
Traceback (most recent call last):
  File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 52, in fork_in_bg
    out = function_torun(*args)
  File "/usr/sbin/glideFactoryEntryGroup.py", line 216, in forked_check_and_perform_work
    factory_in_downtime, entry, work[entry.name])
  File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryEntry.py", line 1073, in check_and_per
form_work
    params, in_downtime, condorQ)
  File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryEntry.py", line 1438, in unit_work_v3
    max_glideins, work, log=entry.log, factoryConfig=entry.gflFactoryConfig
  File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryLib.py", line 902, in logWorkRequest
    work_el['internals'], fraction)
  File "/usr/lib/python2.6/site-packages/glideinwms/factory/glideFactoryMonitoring.py", line 364, in logClient
Monitor
    el[ek] += (client_monitor[ck] * fraction)
TypeError: can't multiply sequence by non-int of type 'float'
[2018-03-13 05:21:32,468] WARNING: glideFactoryEntryGroup:373: Error occurred while trying to find and do work
.
[2018-03-13 05:21:32,469] ERROR: glideFactoryEntryGroup:374: Exception:
Traceback (most recent call last):
  File "/usr/sbin/glideFactoryEntryGroup.py", line 371, in iterate_one
    group_name, my_entries)
  File "/usr/sbin/glideFactoryEntryGroup.py", line 306, in find_and_perform_work
    post_work_info = forkm_obj.bounded_fork_and_collect(parallel_workers)
  File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 310, in bounded_fork_and_collect
    post_work_info_subset = fetch_ready_fork_result_list(pipe_ids)
  File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 208, in fetch_ready_fork_result_list
    out = fetch_fork_result(fd, pid)
  File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 97, in fetch_fork_result
    out = cPickle.loads(rin)
EOFError
[2018-03-13 05:21:32,469] DEBUG: glideFactoryEntryGroup:376: Group Work done: {}
```

#2 - 03/13/2018 08:21 AM - Marco Mascheroni

- Subject changed from *Critical bug in 3.2.21 leads to leaking glideFactoryEntryGroup.py* to *Critical bug in 3.2.21 leads to leaking glideFactoryEntryGroup.py processes*

#3 - 03/13/2018 10:10 AM - Parag Mhashilkar

Mambelli is looking into. I did some basic investigation and here is what I found. Question for both Marcos, why is the value string?

```
[mac-128660:glideinwms]$ condor_status -pool gfactory-1.t2.ucsd.edu -constraint '(Name=="43079_CMSHTPC_T2_US_U
CSD_gw6@gfactory_instance@SDSC@uclhc-fe-1-t2-ucsd-edu_OSG_gWMSFrontend.osg") && (glideinmytype=="glideclient) '
-any -l | grep RunningCores
GlideinMonitorGlideinsRunningCores = "11"
```

Irrespectively, would be good idea to do the type checking and conversions when you read the value from the glideclient classads before performing any numerical operations.

#4 - 03/13/2018 10:30 PM - Brian Bockelman

Is there something in the exception cleanup path that is broken?

Regardless of the issue at hand, this shouldn't be leaking memory / processes when a failure is encountered...

#5 - 03/14/2018 05:38 AM - Marco Mascheroni

- Priority changed from *Immediate* to *Normal*

Hi all,

I am lowering the priority of this, Jeff and I have been watching the factory in the past day and it seems the simple fix I proposed is doing the job. Also, we can deploy the factory on goc-itb and cern, and then apply the fix without the need of new rpms. I still think it is worth getting the value to integer upstream, but that's not a blocker anymore.

About Brian's comment, I also think the exception is not handled correctly and it's strange that the processes were "leaking" (BTW, when I tried to trace one of the stale processes it immediately exit, weird). On the other hand this behavior helped us finding the issue, so I am not too worried since this was something it is not supposed to happen and "normal" exceptions are handled properly.

Marco

#6 - 03/14/2018 08:09 AM - Brian Bockelman

Hi,

Is the proposal then that we don't fix the process leaking bug because it is a good thing that the factory crashes when it is triggered by other bugs?

Brian

#7 - 03/14/2018 10:19 AM - Parag Mhashilkar

Marco Mascheroni wrote:

Hi all,

I am lowering the priority of this, Jeff and I have been watching the factory in the past day and it seems the simple fix I proposed is doing the job. Also, we can deploy the factory on goc-ib and cern, and then apply the fix without the need of new rpms. I still think it is worth getting the value to integer upstream, but that's not a blocker anymore.

About Brian's comment, I also think the exception is not handled correctly and it's strange that the processes were "leaking" (BTW, when I tried to strace one of the stale processes it immediately exit, weird). On the other hand this behavior helped us finding the issue, so I am not too worried since this was something it is not supposed to happen and "normal" exceptions are handled properly.

Marco

Yes your fix should work for now. Based on my quick look at things, there are couple of issues here

- Frontend is expected to put the numbers in classad as int. Clearly one of the frontend is not. Needs to be investigated by whoever is going to work on it.
- All the classad values used in calculations should be converted to int as soon as read. Currently we don't as the expectation was the values are of correct type. This example clearly violated this exposing the issue. We surely want to avoid this in future.

My investigation has been deep enough to comment on the leak observed.

#8 - 04/02/2018 05:18 PM - Marco Mambelli

- *Status changed from New to Feedback*

- *Assignee changed from Marco Mambelli to Dennis Box*

Added the int casting suggested by Marco Mascheroni and Jeff and two more to protect against other parameters.

Following up w/ Brian's comment I worked also to understand why a memory leak was happening and fix it.

The forked process is actually in a try-except already and works correctly. The exception is printed, pipe closed and the process is exiting.

The memory leak I think was happening because the poll/epoll calls were not listening to all possible events in the pipe to read the outputs, e.g. the closing due to exception, and were leaving the descriptors open. I fixed that. The program will read the empty output, close the pipe and recognize the error.

This should avoid memory leaks also in case of new unexpected failures.

Changes are in v3/19360 and 3.2.22.rc1

Sent email to investigate why the problem happened in the first place (we were unable to reproduce it with the current version)

#9 - 04/03/2018 03:49 PM - Marco Mambelli

- Status changed from *Feedback* to *Resolved*

- Assignee changed from *Dennis Box* to *Marco Mambelli*

#10 - 04/05/2018 12:09 PM - Marco Mambelli

From Jeff's email:

As an update, in the redmine ticket Parag cited a wrongly quoted string from the uclhc frontend.

At the time this was running:

```
/var/log/yum.log-20180101:Jan 20 16:41:16 Installed: glideinwms-vofrontend-standalone-3.2.15-1.osg33.el6.noarch
```

Edgar has recently upgraded this fe to 3.2.21:

```
Mar 22 08:42:52 Updated: glideinwms-vofrontend-standalone-3.2.21-2.osg34.el6.noarch
```

Perhaps this is why we no longer see the issue?

Command

```
condor_status -pool gfactory-1.t2.ucsd.edu -constraint '(glideinmytype=="glideclient")' -any -l | grep Running  
Cores | grep \"
```

shows no quoted strings at the moment

#11 - 04/17/2018 05:15 PM - Marco Mambelli

- Status changed from *Resolved* to *Closed*