GlideinWMS - Bug #12972

PicklingError: Can't pickle <type 'Boost.Python.enum'>: import of module Boost.Python failed

06/21/2016 03:51 AM - Marco Mascheroni

Status:	Closed	**Start date:**	06/21/2016
Priority:	Normal	**Due date:**	
Assignee:	Marco Mascheroni	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	v3_2_15	**Spent time:**	0.00 hour
First Occurred:		**Stakeholders:**	CMS
Occurs In:			

Description

There has been a report about this error where it looks like `key` is a tuple and the printout fails hiding the real error. Quickfix would be to do str(key) in the `logSupport.log.warning`, but I think the real issue is that something changed with the python bindings. Investigating more.

```
Traceback (most recent call last):
File "/usr/sbin/glideinFrontendElement.py", line 235, in main
rc = self.iterate()
File "/usr/sbin/glideinFrontendElement.py", line 265, in iterate
done_something = self.iterate_one()
File "/usr/sbin/glideinFrontendElement.py", line 355, in iterate_one
pipe_out=forkm_obj.fork_and_collect()
File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 202, in fork_and_collect
results = fetch_fork_result_list(pipe_ids)
File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 108, in fetch_fork_result_list
logSupport.log.warning("Failed to extract info from child '%s'" % key)
TypeError: not all arguments converted during string formatting
```

History

#1 - 06/22/2016 09:34 AM - Marco Mascheroni

- Assignee changed from Marco Mascheroni to Parag Mhashilkar

#2 - 06/22/2016 09:36 AM - Marco Mascheroni

- Target version set to v3_2_15

#3 - 06/22/2016 09:38 AM - Marco Mascheroni

BTW, this allowed me to unveil the real error which is:

```
[2016-06-21 14:28:25,410] ERROR: Forked process '<bound method glideinFrontendElement.get_condor_q of <__main_
_.glideinFrontendElement
instance at 0x1936cf8>>' failed
Traceback (most recent call last):
  File "/usr/lib/python2.6/site-packages/glideinwms/lib/fork.py", line 47, in fork_in_bg
    os.write(w, cPickle.dumps(out))
PicklingError: Can't pickle <type 'Boost.Python.enum'>: import of module Boost.Python failed
```

Right now there are tests in the CMS ITB frontend, waiting for these tests to finish so I can dig more.

#4 - 06/22/2016 05:36 PM - Marco Mascheroni

- Subject changed from Error during string formatting to PicklingError: Can't pickle <type 'Boost.Python.enum'>: import of module Boost.Python failed

I found a way to replicate the issue, here is a small script (maybe we can create a unit test to avoid this in the future?):

```
import htcondor as condor
import cPickle
from glideinwms.lib import condorMonitor
```

```
coll = condor.Collector("vocms0115.cern.ch")
scheddAd = coll.locate(condor.DaemonTypes.Schedd, "crab3test-8@vocms058.cern.ch")
schedd = condor.Schedd(scheddAd)
results = schedd.query('
((JobStatus=?=1)||(JobStatus=?=2)) && (((JobUniverse=?=5) && ((DESIRED_Sites=!=UNDEFINED) || (DESIRED_Gatekeep
ers=!=UNDEFINED)) && (RequestMemory=!=UNDEFINED) && (DESIRED_Sites=!="T2_CH_CERN_HLT")) && ((stringListsInters
ect("US",DESIRED_Overflow_Region) && !stringListsIntersect("T2_US_Vanderbilt,T3_US_CMSLPC,T3_US_TAMU",DESIRED_
Sites)) && (CMS_ALLOW_OVERFLOW=?="True") && (CRAB_UserRole=!="production") && (JobStatus=?=1) && ((CurrentTime
-QDate)>(2*60*60))))', ['DESIRED_Gatekeepers', 'DESIRED_Sites', 'JobUniverse', 'MaxWallTimeMins', '
MyCurrentTime', 'REQUIRED_OS', 'RequestMemory', 'RequestCpus', 'CMS_ALLOW_OVERFLOW', 'CRAB_UserRole', '
DESIRED_Overflow_Region', 'JobStatus', 'x509UserProxyFirstFQAN', 'x509UserProxyFQAN', 'x509userproxy', '
EnteredCurrentStatus', 'ServerTime', 'RemoteHost', 'ClusterId', 'ProcId'])
s = cPickle.dumps(results)
results = condorMonitor.list2dict(results, ['ClusterId', 'ProcId'])
s = cPickle.dumps(results)
```

Interestingly

```
cPickle.dumps
```

works fine before calling

```
list2dict
```

then it fails with the same error as above.

The problem seems to be a undefined classad. See below, if I remove it everything works fine. Not really sure why it works before calling list2dict, maybe it is related to the internals of the python binding, IMHO it is worth to bring this up with condor devs.

```
(Pdb) results[(173627L, 0L)]['CRAB_UserRole']
classad.Value.Undefined
(Pdb) cPickle.dumps(results[(173627L, 0L)])
*** PicklingError: Can't pickle <type 'Boost.Python.enum'>: import of module Boost.Python failed
(Pdb) del results[(173627L, 0L)]['CRAB_UserRole']
(Pdb) cPickle.dumps(results[(173627L, 0L)])
'(dp1\nS\'TargetType\'\np2\nS\'Machine\'\np3\nsS\'DESIRED_Sites\'\np4\nS\'T2_US_Nebraska_HOTSTUFF\'\np5\nsS\'R
equestMemory\'\np6\nL2000L\nsS\'ServerTime\'\np7\nL1466633551L\nsS\'x509UserProxyFQAN\'\np8\nS\'/DC=ch/DC=cern
/OU=Organic Units/OU=Users/CN=clundst/CN=514102/CN=Carl Lundstedt,/cms/Role=NULL/Capability=NULL,/cms/uscms/Ro
le=NULL/Capability=NULL\'\np9\nsS\'OVERFLOW_IT\'\np10\ncclassad\nExprTree\np11\n(S\'ifthenelse(regexp("T[1,2]_
IT_",DESIRED_Sites),"True",undefined)\'\ntRp12\nsS\'OVERFLOW_UK\'\np13\ng11\n(S\'ifthenelse(regexp("T2_UK_Lond
on_",DESIRED_Sites),"True",undefined)\'\ntRp14\nsS\'CMS_ALLOW_OVERFLOW\'\np15\nS\'True\'\np16\nsS\'MaxWallTime
Mins\'\np17\nL1250L\nsS\'JobStatus\'\np18\nL1L\nsS\'DESIRED_Overflow_Region\'\np19\nS\'US,none,none\'\np20\nsS
\'OVERFLOW_US\'\np21\nS\'True\'\np22\nsS\'x509UserProxyFirstFQAN\'\np23\nS\'/cms/Role=NULL/Capability=NULL\'\n
p24\nsS\'x509userproxy\'\np25\nS\'/data/srv/glidecondor/condor_local/spool/3620/0/cluster173620.proc0.subproc0
/50e4f33198f855389beb133ecb3ec229ad546b98\'\np26\nsS\'JobUniverse\'\np27\nL5L\nsS\'MyType\'\np28\nS\'Job\'\np2
9\nsS\'EnteredCurrentStatus\'\np30\nL1466544901L\nsS\'RequestCpus\'\np31\nL1L\ns.'
```

A workaround for this that I have tested is to substitute line 897 of condorMonitor.py from:

```
                dict_el[a] = list_el[a]
```

to

```
                dict_el[a] = list_el[a] if list_el[a] != classad.Value.Undefined else None
```

I have already been testing the patch in ITB and it seems to work.

Thoughts? Can I commit this change to the branch associated to this ticket?

#5 - 06/22/2016 05:50 PM - Marco Mascheroni

Also, I do think this is an important patch to include in 3.2.14 before we put it in production. Would it be possible to create a new release candidate 3.2.14_2 ?

#6 - 06/29/2016 10:15 AM - Parag Mhashilkar

- Status changed from New to Feedback

#7 - 06/29/2016 10:17 AM - Parag Mhashilkar

- Stakeholders updated

#8 - 07/08/2016 02:47 PM - Parag Mhashilkar

- Assignee changed from Parag Mhashilkar to Marco Mascheroni

So I looked at the output of condor_q -xml -format "%s" foobar and it will not print anything if foobar is not available in the classad. Addition of undefined is something new with the auto format (-af) or the bindings. So why don't we just ignore the classad attrs that are undefined. This should keep the behavior same. I did not test this, so please check if I haven't missed anything

Essentially change following

```
        for a in list_el:
            if not (a in attr_list):
                dict_el[a] = list_el[a]
                try:
                    if ((USE_HTCONDOR_PYTHON_BINDINGS == True) and
                        (list_el[a].__class__.__name__ == 'ExprTree')):
                        # Try to evaluate the condor expr and use its value
                        # If cannot be evaluated, keep the expr as is
                        a_value = list_el[a].eval()
                        if a_value != classad.Value.Undefined:
                            dict_el[a] = a_value
                except:
                    # Do not fail
                    pass
```

to

```
        for a in list_el:
            if not (a in attr_list):
                try:
                    if (USE_HTCONDOR_PYTHON_BINDINGS == True):
                        if (list_el[a].__class__.__name__ == 'ExprTree'):
                            # Try to evaluate the condor expr and use its value
                            # If cannot be evaluated, keep the expr as is
                            a_value = list_el[a].eval()
                            if a_value != classad.Value.Undefined:
                                dict_el[a] = a_value
                        elif list_el[a] != classad.Value.Undefined:
                            dict_el[a] = list_el[a]
                    else:
                        dict_el[a] = list_el[a]
                except:
                    # Do not fail
                    pass
```

We already talked about this being in v3.2.15. So whenever you are done with all your changes and tested them send it over to me for feedback.

#9 - 07/20/2016 10:47 AM - Marco Mascheroni

- Status changed from Feedback to Resolved

- % Done changed from 0 to 100

#10 - 08/05/2016 01:47 PM - Parag Mhashilkar

- Status changed from Resolved to Assigned

Hi Marco, I found an issue with the changes I proposed in my last comment. Looks like some of the attributes when queering the jobs are skipped. Reverting back the for loop to previous version fixes the problem. I suspect it may have to do with the code in try block. Did you test these changes? Errors found in the group log in the factory' log dir

#11 - 08/07/2016 08:58 PM - Parag Mhashilkar

Marco, I made the changes while I was working on other ticket. I have merged them and tagged rc2. Can you please test them for the test cases for this ticket? Once you can confirm I will announce the rc2.

#12 - 08/12/2016 09:20 AM - Parag Mhashilkar

Recent changes seem to work. Merged

#13 - 08/12/2016 09:20 AM - Parag Mhashilkar

- Status changed from Assigned to Resolved

#14 - 08/17/2016 01:43 PM - Parag Mhashilkar

- Status changed from Resolved to Closed