**Gratia Probes Test Automation**

**General**

This document describes Gratia Probes related Test Automation, which builds upon the [existing Test Framework.](#HowToWriteTests)

**Files impacted**

The following source files were added/modified. All the files are in SVN repository in the directory “osgtest/tests” (example /root/trunk/osgtest/tests)

test\_28\_gratia.py – gratia-service related tests

test\_54\_gratia.py – gratia-service validation tests and all the probes related tests

test\_78\_voms.py – setting up a state variable indicating if the voms certificate was removed

test\_80\_gratia.py – clean up related tests for gratia-service and the probes

**Test Execution**

This section describes how to execute Gratia Probes related Test Automation suite using the SVN trunk.

* Setup a Virtual Machine following instructions at “[FermiCloud\_Open\_Nebula\_Manual](#FermiCloud_Open_Nebula_Manual)”
* Login to the Virtual Machine as root (e.g. ssh –l root fermicloud380.fnal.gov) and execute:
  + *cd ~*
  + *wget --quiet* [*http://vdt.cs.wisc.edu/native/bootstrap-osg-test*](http://vdt.cs.wisc.edu/native/bootstrap-osg-test)
  + *chmod 0755 bootstrap-osg-test*
  + *./bootstrap-osg-test development*
  + *rm -rf /var/lib/mysql*
  + *yum install -y subversion*
  + *yum install -y mysql*
  + *yum install -y mysql-server*
  + *svn co https://vdt.cs.wisc.edu/svn/software/osg-test/trunk*
* Validate that the test automation log files (for the tests based on static data) are present in: /usr/share/osg-test/gratia. [Table\_1](#Table_1) summarizes the various queries used and lists the ones based on static input.
* Execute the following commands:
  + cd ~/trunk
  + *./osg-test -r osg-development -vadi osg-tested-internal -i gratia-probe-sge > ~/o1.txt 2>&1*

**Probes Tested**

The following table summarizes the various probes tested, current query, whether or not static input is used and if the criteria is simply that at least one record result is returned.

|  |  |  |  |
| --- | --- | --- | --- |
| Probe | Current Query | Static Input ? | “At least  one record  check” ? |
| gridftp | select sum(Njobs) from MasterTransferSummary | NO | YES |
| gridftp | select sum(TransferSize) from MasterTransferSummary | NO | YES |
| glexec | select Njobs from MasterSummaryData where ProbeName like 'glexec%' | YES | NO |
| glexec | select WallDuration from MasterSummaryData where ProbeName like 'glexec%' | YES | NO |
| dcache | select TotalSpace,FreeSpace, UsedSpace from StorageElementRecord where ProbeName like 'dCache-storage%' | NO | NO |
| condor | select sum(Njobs) from MasterSummaryData where ProbeName like 'condor%' | NO | YES |
| condor | select sum(WallDuration) from MasterSummaryData where ProbeName like 'condor%' | NO | YES |
| psacct | select \* from MasterSummaryData where ProbeName like 'psac%' and ResourceType='RawCPU' | NO | YES |
| bdii | select count(\*) from ComputeElement | NO | YES |
| pbs | select sum(nJobs) from MasterSummaryData where ProbeName=pbs-lsf:<hostname> | YES | NO |
| sge | select sum(nJobs) from MasterSummaryData where ProbeName='sge:<hostname>’ | YES | YES |

**General Test Sequence**

After a probe is executed:

1. The corresponding outbox directory (example: */var/lib/gratia/tmp/gratiafiles/subdir.psacct\_fermicloud322.fnal.gov\_fermicloud322.fnal.gov\_8880/outbox/*) is checked to validate that it’s empty. (*Only for “gratia-probe-bdii-status”, this check has been temporarily commented out, due to a bug, which was discovered during the automated testing. Tanya is working on this aforementioned bug.)*
2. *“isProbeInfoProcessed” method is invoked to parse the gratia log for patterns signifying that Gratia has processed the probe information.*
3. Finally, as summarized in [Table\_1](#Table_1), an appropriate query is executed to validate results.

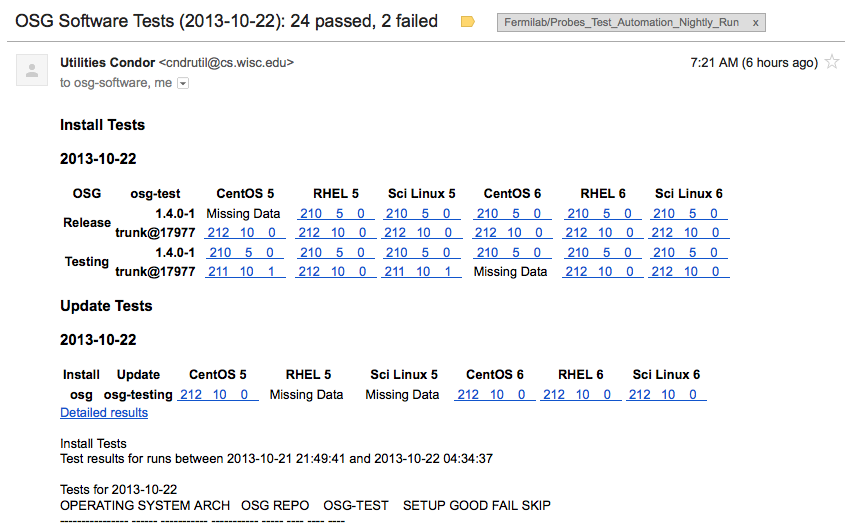
**Notes**

1. The general format used for an sql query in the test scripts is:

*echo \"use gratia; "<some\_sql\_command>; \" | mysql --defaults-extra-file=\"" + filename + "\" --skip-column-names -B --unbuffered --user=reader --port=3306"*

*where the “filename” points to a generated file, which has the appropriate mysql credentials.*

1. Gratia tests are coded such that they could be executed repeatedly on the same machine
2. Gratia tests preserve certain files (such as “ProbeConfig” for each probe) and certain directories (such as */var/lib/gratia, /var/lib/gratia-service and /etc/gratia/services).* The intent is to leave the machine in the state found, as much as possible.
3. The following shows a snapshot of nightly test automation results email. To get added to this distribution list, please send an email to Tim Cartwright ([cat@cs.wisc.edu](mailto:cat@cs.wisc.edu))



**Contact:**

Srinivasan Ramachandran ([srini@fnal.gov](mailto:srini@fnal.gov)) - (630) 840 2598

**References:**

* <http://fclweb.fnal.gov/fermicloud-geeks.html>

* <https://www.opensciencegrid.org/bin/view/SoftwareTeam/HowToWriteTests>