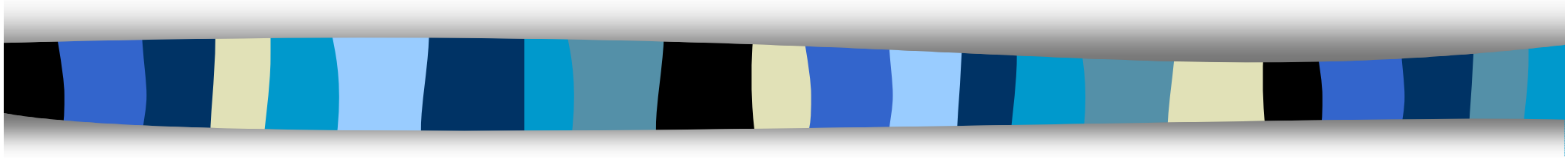
TSTG General Meeting: Current Contents of the MainScript



Gilbert Grosdidier

LAL-Orsay/IN2P3/CNRS
& LCG

The different kinds of Tests [v0.10]

- **Watch out**: several test names are currently far from meaningful

- **CEGate2**: Globus Gatekeeper Unit Testing (CEs, SEs)
  - 6 tests achieved on each node
- **FTP**: GridFTP functionality tests (RB, CEs, SEs)
- **DNS**: reverse DNS Tests (RB, CEs, SEs, MDS, PX)
- **RB**: Functionality tests for RB (Unit testing)
  - it's a suite of 5 small jobs submitted through JDL files
- **SESetup**: Checkup of SE setup (SEs)
- **RMSetup**: Checkup of RM setup (SEs, RC), for EDG-1.4.3
- **JStorm**: Job Storm (All components, Global & Stress testing)
  - Simple jobs with InpOut sandbox transfers, and check of output contents
- **CStorm**: Copy Storm (All components, Global & Stress testing)
  - Achieves above GridFTP tests from jobs submitted to the RB
- **RStorm**: Replica Storm (All components, Global & Stress testing)
  - Achieves above Replica Manager tests from jobs submitted to the RB

The different kinds of Tests (2)

- **MDS**: Consistency tests for MDS + BDII (2 tests in serie)
- **ProXyf**: Security Test for RB (stealing proxies - better if failing)
- **SEs**: GridFTPUmask checks for SEs (should be merged with ?)

- Next 4 tests are intrusive:
  - **CEfch2**: Install&Config + Security tests for CE
  - **RBfch2**: Install&Config + Security tests for RB
  - **UIfch2**: Install&Config + Security tests for UI
  - **SECfch**: Global Security tests (RB, CE, SEs, UI)
    - overlaps somewhat with last 3 tests

The different kinds of Tests: future

- Ready, but not tested:
  - **Globusrun**: checks Globusrun authentication (CEs, SEs)
  - **RLS**: a suite of 5 tests provided by Cal for EDG-2.0 RLS/RM

- New additions:
  - **RMSuite**: merging of GM/MB RM suite
    - migration towards EDG-2.0 ?
  - **Andrea's** Storm
  - Migration of **RMSetup** above towards EDG-2.0
    - and also RStorm
  - Several Cal's tests if they prove useful (CheckVOVars, GridFTPClient, …)

- Refurbishing of several Cal's tests for full implementation of XML/HTML abilities
- Migration of all tests in MainScript to comply to options presented below

The different Options provided

- . MainScript **--TList**="test" (default: DNS)
  - test = CEGate2 FTP DNS RB SESetup RMSetup JStorm CStorm Rstorm
    - also: MDS ProXyf SEs SECfch CEfch2 RBfch2 UIfch2
- . "MainScript **--help**" :
    - Prints this README file, plus the full option List.
- . "MainScript **--MDebug**" :
    - Prints some Variable values from inside the MainScript.
- . "MainScript --TList="test" **--fullHelp**" :
    - To Force printing of a detailed Help about the selected Tests.
- . "MainScript --TList="test" **--showME**" :
    - To Force printing of option values and machine names for the selected Tests.

The different Options provided (2)

- . MainScript --TList="test" **--addOptionList**="--opt1=val1 --opt2=val2 ..."
  - To Provide a list of additionnal Options to the Tests to be achieved.
- . MainScript --TList="test" **--forceMachineList**="node1 node2"
  - To Provide a list of Machines to be used in the tests, overriding the default
- . MainScript --TList="test" **--addMachineList**="node1 node2 ..."
  - To Provide a list of Machines to be used in the tests, adding them to the default ones
- . MainScript --TList="test" **--forcingVO**="otherVO"
  - To Force a VO other than "tstg". Useful in a few tests only.

- Combining showME option with the above ones could be smart to figure out their actual action
- All above options can be used together
- Their usefulness is NOT guaranteed for all tests

Detailed Presentation of the tests

- **DNS** (ReverseDNS)
  - Test: Check that a reverse DNS lookups of the host names work.
  - Arguments: List of fully-specified host names.
  - Machines: CE, P, RB, SE

  - The GSI libraries check that the host name in a given certificate
  - corresponds to the actual host name provided by a reverse DNS lookup.
  - The security check will fail if the reverse DNS lookup fails.

Detailed Presentation of the tests

- FTP (GridFTP):

  - Test: Checks that basic transfers and third-party transfers work.
  - Arguments: List of GridFTP servers to check.
  - Machines: CE, RB, SE

  - This test checks that transfers between the UI machine and all of the
  - given hosts work. It checks both absolute URLs and URLs relative to
  - the user's home area. In addition, third party transfers between all
  - pairs of hosts are checked.

  - # The EDG commands required to run this test.
    - $self->{commands} = ["globus-url-copy",
    - "grid-proxy-info",
    - "edg-gridftp-rm"];

Detailed Presentation of the tests

- **SESetup** (SESetupTest):
  - Test: Checks that the basic SE configuration is correct.
  - Arguments: List of SEs to be checked.
  - Machines: SE
  - Example: ./RunTest BaseTest::SESetupTest lxshare0393.cern.ch

  - This checks that the basic SE configuration is correct. It specifically, checks:
  - * That the SE GRIS is running.
  - * Supported VO(s) are defined in the GRIS.
  - * The GridFTP server is running.
  - * The storage path exists.
  - * The "magic" softlink exists for NFS access (via BrokerInfo).
  - * The edg_query_vo_storage cron job is running.
  - This does not check the data management software or configuration.

```
$self->{commands} = ["grid-proxy-info",
                     "edg-gridftp-exists",
                     "edg-gridftp-ls"];
```

Detailed Presentation of the tests

- **CEGate2** (GlobusGatekeeper):

  - Test: Checks that gatekeeper functions correctly.
  - Arguments: List of nodes running gatekeepers to check.
  - Machines: CE, SE

  - This test checks that a globus-job-run, a globus-job-submit, and a
  - globusrun with two-phase commit all work. The two-phase commit test
  - is extremely important as dg-job-submit's will fail unless this works
  - correctly. (Note this involves an (as yet) unidentified port on the Gatekeeper.)

```
$self->{commands} = ["globus-job-run",
                     "globus-job-submit",
                     "globus-job-status",
                     "globus-job-get-output",
                     "globus-job-clean",
                     "grid-proxy-info"];
```

Detailed Presentation of the tests

- **RB** : Functionality tests for RB (Unit testing)
  - A sequence of 5 Cal's tests executed in turn to test the RB:
    - BaseTest::EDGLifecycle::HelloWorld
    - BaseTest::EDGLifecycle::HelloScript
    - BaseTest::EDGLifecycle::Sleep
    - BaseTest::EDGLifecycle::Checksum
    - BaseTest::EDGLifecycle::BrokerInfo
  - Explainations on next slides

Detailed Presentation of the tests

- **HelloWorld:**
  - Test: Checks that basic EDG job submission works.
  - Arguments: none (accepts site option)
  - Just runs a HelloWorld job (no input sandbox, stderr and stdout in output sandbox). Succeeds if the output contains the correct string.
  - To test a specific site, it is recommended to use the "site" option with a string which occurs in your CEId's (typically the site name).
  - # The EDG commands required to run this test.
    - $self->{commands} = ["dg-job-submit",
    - "dg-job-get-output",
    - "dg-job-status",
    - "grid-proxy-info"];

Detailed Presentation of the tests

- **HelloScript**:

  - Test: Checks that staging a script and running a script works.
  - Arguments: none (accepts site option)

  - This just runs the equivalent of HelloWorld, but by staging a script.
  - This tests if the staging through the input sandbox works for small files.
  - To test a specific site, it is recommended to use the « site » option with a string which occurs in your CEId's (typically the site name).

  - # The EDG commands required to run this test.
    - Same as above ...

Detailed Presentation of the tests

- **Checksum**:

  - Test: Stage a file and check that checksum is correct.
  - Arguments: none (accepts sizemb and site options)

  - Transfers a file via the input sandbox and checks that the checksum of
  - the file is correct. The size can be specified by the sizemb option
  - and the site option can be used to direct the job to a specific site.

  - # The EDG commands required to run this test.
    - Same as above ...

Detailed Presentation of the tests

- **BrokerInfo:**

  - Test: Ensure that .BrokerInfo file exists.
  - Arguments: none (accepts site option; do not use resource option)
  - Example: ./RunTest BaseTest::EDGLifecycle::BrokerInfo

  - Check that the .BrokerInfo file is correctly transferred to the worker node and that all of the environmental variables point correctly to it.
  - Do not use the resource option with this test as this bypasses all brokering and therefore does not produce a .BrokerInfo file. Use the site option instead.

  - # The EDG commands required to run this test.
    - Same as above ...

Detailed Presentation of the tests

- **Sleep**:

  - Test: Submit job and sleep for the given time.
  - Arguments: Time to sleep (e.g. 30s, 5m, 2h, 1d) (accepts site)
  - Example: ./RunTest BaseTest::EDGLifecycle::Sleep --time=30s

  - Check that a job of the given durations works. There have been problems at a few minutes, 30 minutes, and 1 day, so spanning these times should be done to ensure that everything is working.

  - # The EDG commands required to run this test.
    - Same as above ...

Detailed Presentation of the tests

- **RMSetup** (RMSetupTest):
  - Checks that the RM setup is correct, and involves
  - Arguments: --mds= --mdsport= --bdii= --bdiiport= --VO=
  - Machines: MDS, BDII, SEs and RC, if any.
  - Was developped using the concatenation of several use cases shown in Chap. 8 of the EDG-1.4.3 User's Guide: it tests the basic functionalities of the Replica Manager from the UI.
  - **Basically**:
    - queries for all env variables involving a given VO thru edg-vo-env
    - pulls CE & SE resources from MDS and BDII thru ldapsearch
    - checks they are identical in both searches
    - requires at least 2 SEs are available
    - extracts from SE II the mount points for this VO thru ldapsearch
    - checks these mount points match with the GDMP config thru gdmp_job_status
    - checks that for each SE pair the SE can communicate thru gdmp_host_subscribe
    - creates on each SE the relevant test directory with edg-gridftp-mkdir
    - erases all leftover files on these subdirs thru edg-gridftp-rm
    - checks that everything is fine on these subdirs thru edg-gridftp-ls
    - (end of mutual stuff)

RMSetup (2)

- # The EDG commands required to run this test.
  - $self->{commands} = ["grid-proxy-info",
  - "edg-vo-env",
  - "edg-gridftp-rm",
  - "edg-gridftp-ls",
  - "edg-gridftp-mkdir",
  - "gdmp_host_subscribe",
  - "gdmp_job_status",
  - "globus-replica-catalog",
  - "ldapsearch",
  - "edg-replica-manager-replicateFile",
  - "edg-replica-manager-copyAndRegisterFile",
  - "edg-replica-manager-deleteFile",
  - "edg-replica-manager-listReplicas"];

Detailed Presentation of the tests

- **JStorm**:

  - Could easily be one job only (change in MainScript).
  - Based on BaseTest::EDGLifecycle::Stack::JobStorm Gilbert's test.
  - Exercises Input and Output sandbox transfers, job submission and retrieval using the RB, and checks the contents of the output files.

  - ```
    # The EDG commands required to run these tests.
    $self->{commands} = ["dg-job-submit",
                         "dg-job-get-output",
                         "dg-job-status",
                         "dg-job-cancel",
                         "dg-job-get-logging-info",
                         "grid-proxy-info"];
    ```
    ```

Detailed Presentation of the tests

- **CStorm:**
  - Based on BaseTest::EDGLifecycle::Stack::CopyStorm Gilbert's test.
  - Reuses tools developped in the FTP test above: exercises, from a job sent to run on a WN thru the RB and the CE, the GridFTP functionalities, and checks that the contents are correct after transfer.
  - # The EDG commands required to run these tests.
    - $self->{commands} = ["dg-job-submit",
    - "dg-job-get-output",
    - "dg-job-status",
    - "dg-job-cancel",
    - "dg-job-get-logging-info",
    - "grid-proxy-info"];

Detailed Presentation of the tests

- **RStorm:**
  - Based on BaseTest::EDGLifecycle::Stack::RepStorm Gilbert's test.
  - Reuses tools developped in the RMSetup test above: exercises, from a job sent to run on a WN thru the RB and the CE, the Replica Manager functionalities, ands checks that the catalog contents and file contents are correct after transfer.
  - # The EDG commands required to run these tests.
    - $self->{commands} = ["dg-job-submit",
    - "dg-job-get-output",
    - "dg-job-status",
    - "dg-job-cancel",
    - "dg-job-get-logging-info",
    - "grid-proxy-info"];

Detailed Presentation of the tests

- **SEs** (GridFTPUmask):

  - Test: Checks that the umask is set correctly to allow group write privilege.
  - Arguments: List of GridFTP servers to check.
  - Machines: CE, RB, SE
  - Example: ./RunTest BaseTest::GridFTPUmask lxshare0381.cern.ch

  - This checks that the umask is set correctly to allow group write privilege by default on files transferred to the server.
  - This is necessary with pooled accounts otherwise users may lose control of files they have written when accounts are recycled.

Detailed Presentation of the tests

- **MDS**:
  - This one is made of 2 tests indeed:
  - BaseTest::CheckInfo::CheckConsistency
    - This test looks up all of the CEs and SEs in the top-level MDS and in the BDII. It compares the lists and reports any discrepancies.
  - BaseTest::CheckInfo::CheckGRIS
    - This test looks up all of the available CEs and SEs in the information system. It then queries each one to ensure that each GRIS is alive.
  - Arguments: --mds= --mdsport= --bdii= --bdiiport= (--VO=)
  - Machines tested for both: mainly MDS and BDII, but also CEs and SEs.

  - # Commands necessary to run this test.
    - $self->{commands} = ["ldapsearch"];

Detailed Presentation of the tests

Detailed Presentation of the tests

Detailed Presentation of the tests

Detailed Presentation of the tests

Detailed Presentation of the tests