

# XrootD Testing Framework

## Basic documentation

Version: 0.0.1

Author: Łukasz Trzaska <[ukaszt@gmail.com](mailto:ukaszt@gmail.com)>

Date: 30.03.2012

## 1 General overview

### 1.1 Framework components

#### 1.1.1 Master

**Module file:** *XrdTestMaster.py*

*Master* is the user entry point for the testing framework. The service is configured via config file and exports a web service showing the current statistics of the service as well as the status of the tests that are being run and have been run in the past. It accepts connections from *Slave* and *Hypervisor* daemons and dispatches commands to them.

Quick summary:

- user entry point to the framework
- supervise and synchronize all system activities
- accepts connections from Slaves and Hypervisors and dispatches commands to them
- is run as a system service (daemon), configured via batch of configuration files

#### 1.1.2 Hypervisor

**Module file:** *XrdTestHypervisor.py*

Daemon which receives the machines and network configurations from the *Master* and starts/stops/configures the virtual machines accordingly. It uses Qemu with the KVM kernel module in Linux and *libvirt* virtualization library to as a layer to communicate with Qemu.

Quick summary:

- application to manage the virtual machines clusters and networks on demand of Master
- is run as a system service (daemon), configured via configuration file
- it starts/stops/configures virtual machines
- uses *libvirt* for managing virtual machines

### 1.1.3 Slave

**Module file:** *XrdTestSlave.py*

Daemon installed on virtual or physical machines that runs the actual tests. In the first iteration it is supposed to receive a bunch of shell scripts from the Master and run them. It connects to Master automatically, having by its name (*Hypervisor* redirects it by IP or master's address is known because user provides it while starting).

Quick summary:

- the actual application which runs tests
- may be running on virtual or physical machines
- is run as a system service (daemon), configured via configuration file
- it receives test cases from *Master* and runs them synchronously with other Slaves

## 2. Users guide

### 2.1 Automatic documentation

Code in each module is documented in *pydoc* style. To create HTML version of documentation for all *.py* files in directory, run following shell command, which create *HTML* files for all *.py* files:

```
# cd PROJECT_DIR
# pydoc -w ./
```

and for specific file e.g. *XrdTestMaster.py* run:

```
# cd PROJECT_DIR
# pydoc -w XrdTestMaster.py
```

where *PROJECT\_DIR* is the directory where testing framework source code is stored.

### 2.2 Installation

Framework is available in *RPM* packages for Linux SLC5. All application requires Python 2 in version at least: 2.4. It comprises of the following three main RPMs (the libraries required by them are also listed below):

1. *xrdtest-master-0.0.1-1.noarch*  
**required RPMs:** *python-apscheduler-2.0.2*, *python-cheetah-2.0.1*, *python-cherrypy-2.3.0*, *python-inotify-0.9.1*, *python-ssl-1.15*, *python-uuid-1.30*
2. *xrdtest-hypervisor-0.0.1-1.noarch*  
**required RPMs:** *python-ssl-1.15*, *libvirt-python-0.9.3*, *libvirt-0.9.3*
3. *xrdtest-slave-0.0.1-1.noarch*  
**required RPMs:** *python-ssl-1.15*

To install each module you have to follow typical RPM installation instructions.

## 2.3 Starting applications

### 2.3.1 Configuration files

Default place for configuration files is directory: `/etc/XrdTest/<CONFIG_FILE_NAME>.conf`

#### Important remarks.

### 2.3.2 Running as a system service (daemons)

If applicatin was installed from RPM it is automatically added to system services, thus it will be started automatically during the system start, but can also be started manually from the command line as follows:

```
# service COMPONENT_NAME start
```

where the `COMPONENT_NAME` is accordingly

```
for master:      xrdtestmasterd
for slave:       xrdtestslaved
for hypervisor:  xrdtesthypervisord
```

### 2.3.3 Running in debug mode

To start each of components (Master, Hypervisor or Slave) in debug mode (it shows log messages on the screen instead of writing them to log file) run the shell command below:

```
# export PYTHONPATH=<PROJECT_DIR>/lib
# cd <PROJECT_DIR>
# python XrdTestMaster.py -c XrdTestMaster.conf
```

Where `<PROJECT_DIR>` should be replaced with actual directory where framework source `.py` files are stored. One can start Hypervisor or Slave replacing `XrdTestMaster` with `XrdTestHypervisor` or `XrdTestSlave` accordingly.

### 2.3.4 Running in background mode

To start application in background mode (as a daemon) add option `-b` to shell starting command. It will then store LOG file and PID file in proper directories specified in configuration files. Go to directory where application is stored and run:

```
# python XrdTestMaster.py -d -c XrdTestMaster.conf
```