RPC and TI-RPC Test Suite
Test Plan Document

Cyril LACABANNE
Bull S.A.S.

Version 1.3
12 July 2007

Revision history

Version	Description
1.0	First release
1.1	Several correction on §1, 5, 8, 14
1.2	Add first page Several correction on §2, 8, 14
1.3	Section 14

Introduction

This project provides a test Suite for RPC and TI-RPC libraries for Linux and other Operating Systems. This “Test Plan” document, follows IEEE 869-1998 rules and is designed to give all information on this project such as, audience, tested functions, and how to install, configure and launch this test Suite.

1. Test plan identifier

Project name : RPC and TI-RPC test Suite.
Audience : Linux Test Project (LTP), Open Source community.
Note : this document follows the IEEE 829-1998 “Test Plan” rules.

2. References

See below the list of all documents that support this test plan :

- Project Plan document (project development purpose)
- rfc 1831, RPC specifications (project documentation purpose)
- rfc 1832, XDR specifications (project documentation purpose)
- rfc 1833, RPCBind* specifications (project documentation purpose)
- Tests Design document (detailed explanation on tests cases)
- Linux TI-RPC how to document (methodology and examples)

3. Introduction

The aim of that test Suite is to provide a suitable test tool for testing RPC (Remote Procedure Call) and TI-RPC (Transport Independent RPC) under Unix based operating system such as Linux on different software version and hardware architecture.

4. Test items

RPC

- Create and destroy API functions
  - svc_destroy
  - svcfd_create
  - svctcp_create
  - svcudp_create
  - svcraw_create
  - svcudp_bufcreate
  - clnt_destroy
  - clnt_create
  - clntraw_create
  - clnttcp_create
  - clntudp_create
  - clntudp_bufcreate
- Register and unregister API functions
  - xprt_register
  - xprt_unregister
  - svc_register
  - svc_unregister
  - registerrpc

* See glossary (section 19)

- Standard API functions
  - clnt_call
  - callrpc
  - svc_getcaller
  - svc_getargs
  - svc_freeargs
  - svc_sendreply
  - clnt_freeres
  - clnt_geterr
  - clnt_control
  - clnt_broadcast
- Address management API functions
  - pmap_set
  - pmap_unset
  - pmap_getport
  - pmap_getmaps
  - pmap_rmtcall
  - get_myaddress
- Error management API functions
  - clnt_pcreateerror
  - clnt_perrno
  - clnt_perror
  - clnt_spcreateerror
  - clnt_sperno
  - clnt_sperror
  - svcerr_noproc
  - svcerr_noprog
  - svcerr_progvers
  - svcerr_systemerr
  - svcerr_auth
  - svcerr_weakauth
- Authentication API functions
  - authnone_create
  - authunix_create
  - authunix_create_default
  - auth_destroy

TI-RPC

- Simplified level API functions
  - rpc_reg
  - rpc_call
  - rpc_broadcast
- Top level API functions
  - clnt_create
  - clnt_create_timed
  - svc_create
  - clnt_call
- Intermediate level API functions
  - clnt_tp_create
  - clnt_tp_create_timed

    - svc_tp_create
    - clnt_call
    - clnt_control
- Expert level API functions
    - clnt_tli_create
    - svc_tli_create
    - rpcb_set
    - rpcb_unset
    - svc_reg
    - svc_unreg
    - clnt_call
- Bottom level API functions
    - clnt_dg_create
    - svc_dg_create
    - clnt_vc_create
    - svc_vc_create
    - clnt_call
- Address management API functions
    - rpcb_getaddr
    - rpcb_getmaps
    - rpcb_rmtcall

Error management API functions

- clnt_pcreateerror
- clnt_perno
- clnt_perror
- svcerr_noproc
- svcerr_noprog
- svcerr_progvers
- svcerr_systemerr
- svcerr_weakauth
- Authentication API functions
    - authnone_create
    - authunix_create
    - authunix_create_default
    - authdes_create
    - authdes_seccreate

5.Software risk issues

The major software risk issue results from a misunderstanding of the requirements. Section 8 describes what is that requirement.

6.Features to be tested

All API functions of Linux RPC (libc) and Linux TI-RPC ported project from Sun ONC+ TI-RPC (tirpc) will be tested.
Tests of **RPC** will be run in two times. First, using standard Linux portmapper* program and then using RPCBind program.
TI-RPC tests will be run using RPCBind daemon only.

For the complete list of API functions to test, see Tests Design Document.

Here is the tests categories list :

- Basic test (simple call)
- Stress and load test (test under bad conditions, low network quality, high process count)
- Limits and overflow test (test of different possibility of arguments given to function)
- Multi-threaded test (tests under MT context)
- Scalability test (test Resources/nodes)
- Data integrity test (interoperability test, between data sent and data received)
- Performance test (Benchmark test)
- Complex and scenario test (a combination of API function and, or combination of test)

7.Features not to be tested

RPC and also TI-RPC are using XDR Data encoding/decoding. Only top-level XDR functions will be tested. Not specific low level XDR functions.

8.Approach (strategy)

Special preparation to use that test Suite

If you plan to test TI-RPC under IPv6 transport, you must have an **IPv6 compliant OS**. Currently, and under Linux OS a patch will be found at http://oss.oracle.com/~cel/linux-2.6/2.6.19/release-notes.html for 2.6.19 Kernel. This patch provides IPv6 support for TI-RPC under Linux 2.6.19 OS.

Test Suite scripts use "**ssh**" to run tests on remote machine (client, server). So, ssh must be installed and correctly configured, but also remote machine should share **public key authentication** to avoid entering password (use ssh-keygen tool for that).

* See glossary (section 19)

Metrics collected

After the end of each test, a result will be :

- "execution: PASS"
- "execution: INTERRUPTED"
- "execution: FAILED"
- "execution: HUNG"

All results will be saved into a log file. Finally, a document called "Tests report" will be used to sum up a test Suite execution such as hardware specification, software version and the collection of all results.

Hardware specification

If you plan to use this test Suite to test the maximum cases possibilities, you will need to run it using several hardware configurations. The list bellow shows what is the impact on tests results when running this test Suite on different hardware configurations :

- number of CPU : action on stressed environment, multi-threaded environment and scalability
- CPU architecture : will test data integrity and interoperability
- LAN speed : will affect performance testing
- Amount of RAM : will affect performance and scalability testing

Software and environment

Requirement

Software required to run this test Suite

- Shell script interpreter
- OpenSSH 4.X

Software required to test RPC library

- libc version 2.5 or above
- portmap daemon

Software required to test TI-RPC library

- tirpc library version 0.1.7 or above
- rpcbind daemon version 0.1.4-1 or above

Software required to compile this test Suite

- gcc 4.X or above (test Suite developed with gcc 4.1.1 2007-01-05)
- make 3.80 or above (test Suite developed with GNU make 3.81)
- tirpc sources files and headers

Optional software

- tsLogParser from Bull tsLogParser Open Source project to manage results under user friendly HTML web pages
  found at http://tslogparser.sourceforge.net/

Environment

Test Suite will be run under TCP or UDP transport, depending on API function tested. All RPC tests are only supported on IPv4. TI-RPC tests can be run either on IPv4 or IPv6, the choice is let to administrator who plan to test RPC and TI-RPC libraries, see section 14 for more information on how to choose IPv4 or IPv6 to run all TI-RPC tests.
Note that fully compliant IPv6 operating system is required to run TI-RPC under IPv6 protocol.

Linux case

Under Linux based operating system, if you plan to test TI-RPC library you must patch your system. When this documentation was written, a patch for kernel 2.6.19 exists and brings TI-RPC IPv6 support to the system.

9.Item pass/fail criteria

Criteria depends on tests. See Tests Design document for a full explanation of each test.

10.Start and stop test Suite criteria

Test Suite will be delivered with an administrator tool, which enables you to choose what test you want to run. With that tool you can choose to run :

- The full test Suite (RPC, TI-RPC tests)
- Only RPC tests
- Only TI-RPC tests
- Some specified tests

Test Suite will be automatically stopped after test run. See **section 14** to know **how to** install, configure, launch, and manage results of test Suite.

11.Suspension criteria & resumption requirements

Before launching the test Suite you can choose what test should be run (described section 10). While tests are running you can abort then with all Unix possibilities, such as kill tests jobs, CTRL+C inside terminal...

12.Test deliverables

Documents

- Tests Design document (PDF)
- Tests Plan document (that document, also PDF)
- Linux TI-RPC “how to” document (PDF)

Reports

- Tests reports after testing on some configurations.

Test Suite

All tests sources, and administration tools will be provided to the Linux community (LTP).

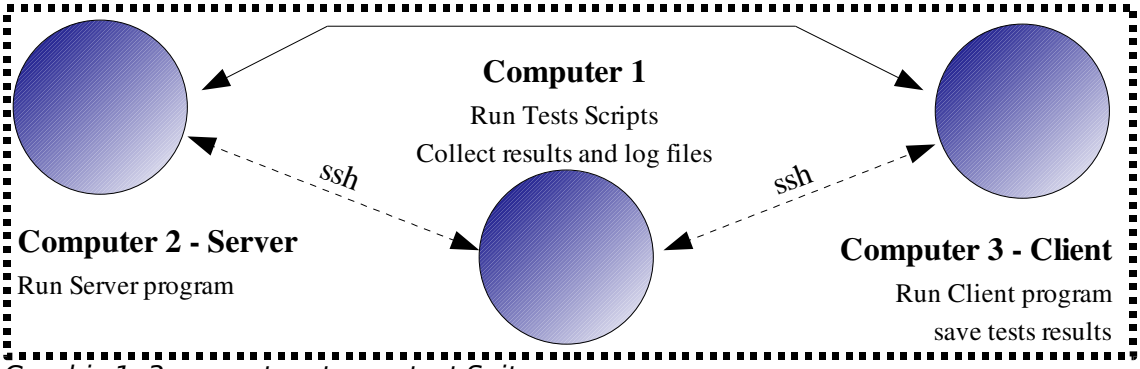
13.Remaining test tasks

N/A

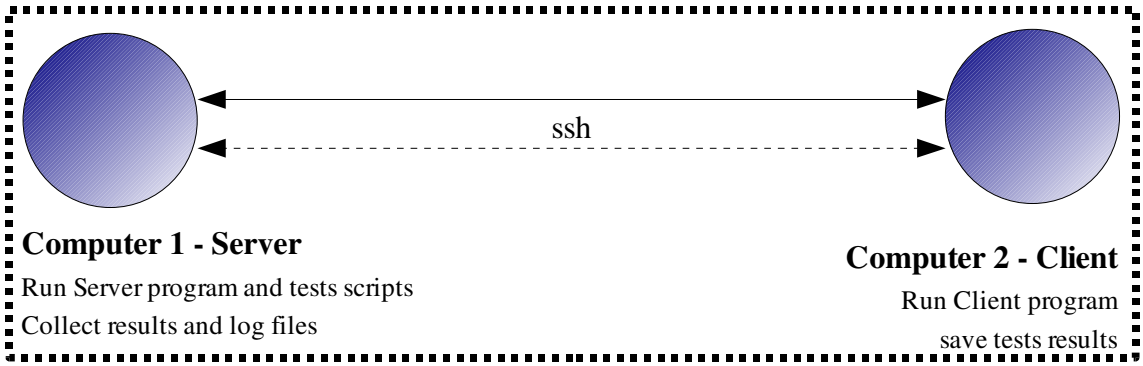
14.Environmental needs (how to use this test Suite)

Preparation

This test Suite written both in Shell Script and C must be executed on two or more computers. At least one “server computer” and one “client computer”. Script Shell programs control tests execution such as start test (server and client), check returned value of each test, save it into a log file and collect all results into local machine. Note that in some case it is possible to configure tests scripts to run on a third machine.
See below a graphic to sum up that points :



Graphic 1: 3 computers to run test Suite



Graphic 2: 2 computers to run test Suite

Note that in case of 2 computers used to run this test Suite, either the server computer or the client computer can be configured to run the tests script. A last case is possible : all tests and scripts are run on the same computer.

Install your test Suite release

After downloading your test Suite (tar.gz file), you have to install it. First **untar** your test Suite in a directory (i.e. /root/rpc_ts_admin). Open a terminal, in the test Suite directory (cd /root/rpc_ts_admin) launch **./configure** script. Script will ask you some details for installation such as server computer IP, login, client computer IP, login, sources of this test Suite (the directory tests_pack in your directory rpc_ts_admin). After this step, you have to compile all tests binaries. Use Makefile to do it. First run **make deploy**, this will compile all binaries on server and client computers. Then run **make all** to compile your test Suite locally.
Now your test Suite is ready to use.

Run your test Suite

A script utility can be used to launch the test Suite. Just run **./rpc_ts_wizard.sh** to start it. This interactive script will ask you the way you want to launch test Suite. You can choose to start all tests including RPC and TI-RPC tests, or select a part of the test Suite.
If you want to run all tests of that test Suite (no interactive mode), run **./rpc_ts_wizard.sh -all.**
You also can start your test Suite using Unix command line. Follow this syntax :

```
./rpc_ts_run.sh -l script_lib.sh [-v -m manycouple -n X]
```

-v : verbose mode
-m manycouple : this option launches several couples of client/server per test, instead of starting one server for several client.
-n X where X is a number : indicates the number of instances to launch per test.
-l script_lib.sh : script_lib.sh is a script containing a tests pack. In example, rpc_err_basic.sh is the script containing all tests for RPC domain, Error sub-domain and basic category.
Note : you can launch several script_lib.sh using several -l flags.

Get and manage results

After the test Suite run, you will find all results in RUN_DATE-TIME sub directory of logs directory of your test Suite (i.e. : /root/rpc_ts_admin/logs/RUN_DATE-TIME).
Results files are rpc_ts.log, containing all tests results and test_run_details.log containing a summary of test Suite run.
You can use tsLogParser (http://tslogparser.sourceforge.net/index.php) to list and analyze your tests results.
Don't forget to add the module rpc_ts.mod.php (http://nfsv4.bullopensource.org/doc/rpc_testsuite.php) in the tsLogParser admin/modules directory.
You will find the test Suite prepared for tsLogParser in your test Suite directory (i.e. : /root/rpc_ts_admin/rpc_ts.tar.gz). Add this rpc_ts.tar.gz file from tsLogParser administration page (add new test suite version button) and then add you rpc_ts.log file (Add new log file button).
Note : you must have configured apache to accept more than 20M post and upload file in php.ini configuration file.

15.Staffing needs

N/A

16.Responsibilities

N/A

17.Planning risks and contingencies

18.Approvals

19.Glossary

Portmapper

Daemon which is able to register a procedure with a service program. When a client call for an RPC, portmapper checks out in its directory the procedure number and gives it to server.

RPCBind

The same as portmapper, but it also provides TI-RPC (IPv4, IPv6) support