Redis Tool User Guide	Public
1.0	Total 5 pages

Redis Tool User Guide

Shyju PV (shyju.pv@gmail.com)

Table of Contents

1 Introduction

This document gives details of redis tool configuration and execution with caliper framework.

2 Lab setup for Redis tool

To execute Nginx tool with caliper framework, below setup is required:
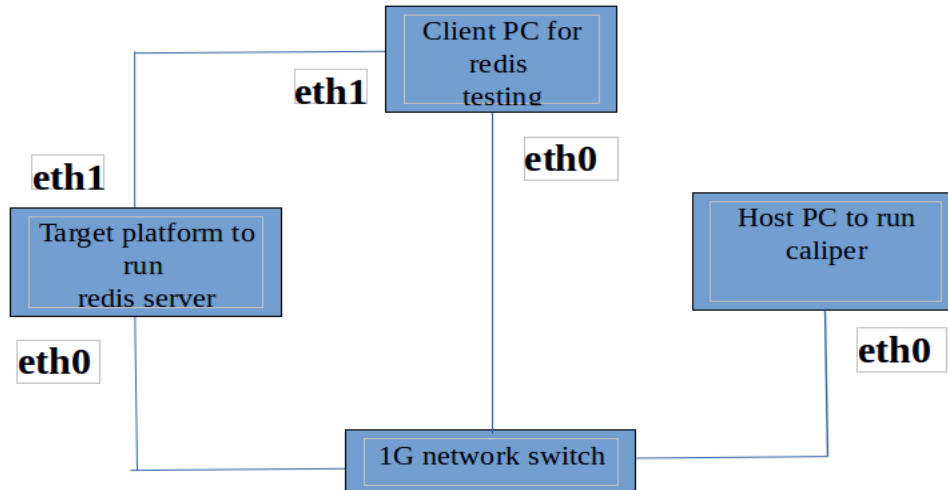
3 Pre-requisite for redis client PC

Run following commands from host PC:
- cd caliper/utils/automation_scripts/Scripts/
- scp run_redis_benchmark.sh $TestNode_user@$TestNode_ip:~/caliper_redis/
- scp generate_inputdata.py $TestNode_user@$TestNode_ip:~/caliper_redis/

TestNode_user and **TestNode_ip** are login credentials of "**TestNode**". Refer below section 4 for **TestNode.**

Note: python should be installed in redis client PC.

4 client_config.cfg file description

Target platform information:

[TARGET]
Platform_name: <give the proper platform name. Do not use underscore in the name>
ip: <eth0 ip of target platform>
port: <give 22 port number for ssh connection>
user: <target platform user name which has root access>
password: <password of the above user name specified>
target_ip_10g: <eth0 ip of target platform>

TestNode platform information:

[TestNode]
ip: <eth0 ip address of >
port: <give unique port number of testnode platform which is available>
user: <user name of the TestNode
password: <password of TestNode>

5 redis_run.cfg file test case description

This file contains the configuration of test case to be run on target platform. The sample test case is given as below:

[Instance_1]
category = Performance application redis_Instance_1
scores_way = exp_score_compute 5 -0.5
scores_way1 = compute_speed_score 1
command = "if [! -d /usr/local/redis-config/]; then mkdir /usr/local/redis-config; cp -r redis-config/* /usr/local/redis-config/ ; fi; ~/irq_11_18.sh; redis-scripts/run_redis_server.sh 0 0;"
parser = redis_parser

- For redis tool, two different score way method has been defined.
- "scores_way" field indicated the scoring algorithm for lateny and "scores_way1" indicates the scoring algorithm for bandwidth.
- irq_11_18.sh file will set the CPU affinity for ethrnet interface which has been used for redis testing. User can create this file according to the target platform and execute that file from the "command" field.

The sample file structure is as follows:

```
 1 #!/bin/bash
 2 # eth1 assigned to 0-7
 3 echo 00000000,00000001 > /proc/irq/84/smp_affinity
 4 echo 00000000,00000001 > /proc/irq/85/smp_affinity
 5 echo 00000000,00000002 > /proc/irq/87/smp_affinity
 6 echo 00000000,00000002 > /proc/irq/88/smp_affinity
 7 echo 00000000,00000004 > /proc/irq/90/smp_affinity
 8 echo 00000000,00000004 > /proc/irq/91/smp_affinity
 9 echo 00000000,00000008 > /proc/irq/93/smp_affinity
10 echo 00000000,00000008 > /proc/irq/94/smp_affinity
11 echo 00000000,00000010 > /proc/irq/96/smp_affinity
12 echo 00000000,00000010 > /proc/irq/97/smp_affinity
13 echo 00000000,00000020 > /proc/irq/99/smp_affinity
14 echo 00000000,00000020 > /proc/irq/100/smp_affinity
15 echo 00000000,00000040 > /proc/irq/102/smp_affinity
16 echo 00000000,00000040 > /proc/irq/103/smp_affinity
17 echo 00000000,00000080 > /proc/irq/105/smp_affinity
18 echo 00000000,00000080 > /proc/irq/106/smp_affinity
19 echo 00000000,00000001 > /proc/irq/108/smp_affinity
20 echo 00000000,00000001 > /proc/irq/109/smp_affinity
21 echo 00000000,00000002 > /proc/irq/111/smp_affinity
22 echo 00000000,00000002 > /proc/irq/112/smp_affinity
23 echo 00000000,00000004 > /proc/irq/114/smp_affinity
24 echo 00000000,00000004 > /proc/irq/115/smp_affinity
25 echo 00000000,00000008 > /proc/irq/117/smp_affinity
26 echo 00000000,00000008 > /proc/irq/118/smp_affinity
27 echo 00000000,00000010 > /proc/irq/120/smp_affinity
28 echo 00000000,00000010 > /proc/irq/121/smp_affinity
29 echo 00000000,00000020 > /proc/irq/123/smp_affinity
30 echo 00000000,00000020 > /proc/irq/124/smp_affinity
31 echo 00000000,00000040 > /proc/irq/126/smp_affinity
32 echo 00000000,00000040 > /proc/irq/127/smp_affinity
33 echo 00000000,00000080 > /proc/irq/129/smp_affinity
34 echo 00000000,00000080 > /proc/irq/130/smp_affinity
```

The description of line number 3 is as follows:
- echo 00000000,00000001 indicates CPU number 0.
- 84 indicates that IRQ number 84 is assigned to eth1 Tx/Rx channel. The IRQ number is depends on the platform.
- So eth1 Tx/Rx interrupt will be handled by CPU0
- Same way user can assign the IRQ number of other eth1 Tx/Rx channels to different CPUs.

- User has to assign IRQs equally all the CPUs (CPU range 0 to 7 has been used in above example)

Note:
1) The intention of assinging IRQs to CPUs is to get maximum throughput of redis server.
2) User can use "*cat /proc/interrupt | grep eth1*" command to find the IRQ number assigned to eth1 interface.

In "command" field, run_redis_server.sh script contains two parameters:
- start_cpu_no and end_cpu_no.

Example: "run_redis_server.sh 0 0" indicates redis server will run on CPU 0.

6 redis_application_run.cfg file test case description

[Instance_1]
command = "cd caliper_redis; ./run_redis_benchmark.sh SHORT $target_ip_10g 0 1 0 1 10000000 10 10000"

run_redis_benchmark.sh script takes below parameters:
- types of test cases - SHORT / BASIC / PIPELINE
- $target_ip_10g – target platform 10G port ip address, the ip address will be taken from client_confi.cfg file.
- start_cpu_num
- number of instances of redis
- keep alive enable / disable
- number of threads
- number of requests
- data size in bytes
- key space length