



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Ping Poller

ADMINISTRACION DE SERVICIOS EN RED

Entregable 1

Autor

Alan Fernando Rincón Vieyra

Profesor

Eduardo Gutiérrez Aldana



México, CDMX, 13 de junio de 2018

1. Introducción	2
1.1. Objetivo	2
2. Análisis y Diseño	3
2.1. Arquitectura propuesta	3
2.2. Especificación de plataforma	3
3. Desarrollo e implementación	4
3.1. Desarrollo	4
3.1.1. Configuración del entorno	4
3.1.2. Realización del programa Ping Poller	5

CAPÍTULO 1

Introducción

En una topología de red es posible que un dispositivo no responda a tiempo a un ping. Para determinar la existencia de una falla en la comunicación con un dispositivo se hace uso de un **ping poller**.

1.1. Objetivo

Desarrollar un ping poller que esté monitoreando a los dispositivos de la red, enviándoles un ping a cada uno y verificando que el tiempo de respuesta sea apropiado.

Este capítulo contiene el plan de trabajo de la presente práctica, detallando el alcance y la especificación del sistema.

2.1. Arquitectura propuesta

En la figura **TODO**: se muestra el diagrama que describe la arquitectura propuesta para la topología de red. La maquina virtual "Debian-1.^{es}" es el dispositivo que tiene implementado el ping poller, que será el encargado de monitorear la conexión de todos los demás dispositivos (routers, pc's).

2.2. Especificación de plataforma

Para el desarrollar de éste componente se hará uso de la herramienta **rrdtool** (versión x) usando el lenguaje de programación **python** (versión 2) en una maquina virtual con el sistema operativo **Debian** (versión 7).

3.1. Desarrollo

En este capítulo se muestran los pasos necesarios para el desarrollo del ping poller.

3.1.1. Configuración del entorno

Objetivo

Instalar las herramientas necesarias para el desarrollo del ping poller.

Descripción

- Instalar python 2.7.9

```
1 $ apt-get update
2 $ apt-get install gcc python=2.7.9-1 python-pip librrd-dev libpython-dev
```

- Instalar rrdtool.

```
1 $ pip install rrdtool
```

Prueba P01-P1: rrdtool instalado

Sistema: Topología de red, Módulo: Ping Poller			
P01-P1 rrdtool instalado			
Pregunta	Si	No	Observaciones
1. Verificar paquetes instalados			
1.1. ¿Se instaló python 2.7.9?	X		Ninguna.
1.2. ¿Se instaló pip?	X		Ninguna.
1.3. ¿Se instaló rrdtool?	X		Me marcaba errores de haders y error en gcc. La solución fue reinstalar gcc.
2. Fin de la Prueba.			

3.1.2. Realización del programa Ping Poller

Objetivo

Realizar un programa en python que mande ping's a los dispositivos conectados en la red, y en caso de que la respuesta del dispositivo sobrepase el tiempo establecido, notificarlo via email.

Descripción

- **ping.py** Programa encargado de realizar un ping a una ip específica y devuelve como resultado el tiempo de respuesta en milisegundos.

```

1  import subprocess, re
2
3  def ping(hostname):
4      ping_response = subprocess.Popen(
5          ['ping', '-c1', '-w1', hostname],
6          stdout=subprocess.PIPE
7      ).stdout.read()
8
9      # Verify packet recieved.
10     packets_recived = int(re.search(
11         r'([\w\sl.\(\|\)\|\-|:|,\\|=|\\n]*) ([0|1]) (\_received)',
12         ping_response).group(2))
13
14     if (packets_recived == 1):
15         time = int(re.search(
16             r'([\w\sl.\(\|\)\|\-|:|,\\|=|\\n]*) (time=) ([0-9]*) ([.|0-9]*) (\_ms)',
17             ping_response).group(3))
18         return time
19     else:
20         return -1

```

- **rrdtool-db.py** Programa encargado de crear la base de datos round robin donde se almacenará el tiempo de retardo de los ping's realizados a una ip determinada.



```
1  #!/bin/python
2  import sys, rrdtool, time
3
4  tMinute = 60
5  tHour = tMinute * 60
6  tDay = tHour * 24
7  #####
8  fname = 'ping.rrd'
9  step = 1 # Time interval between every muestra. (1 sec)
10 time_average = 1 # Time interval to average. (1 sec)
11 hostname = sys.argv[1].replace('.', '_')
12 #####
13 stime = int(time.time())
14 duration = tHour
15
16 # Un archivo round robin, que promedia las muestras de 5s durante una hora.
17 ret = rrdtool.create(
18     fname,
19     '--start', str(stime),
20     '--step', str(step),
21     'DS:ping_%s:GAUGE:3:U:U' % hostname,
22     'RRA:AVERAGE:0.5:%d:%d' % (time_average, duration / time_average),
23 )
24
25 if ret:
26     print rrdtool.error()
```

Uso:

```
1  $ ./rrdtool-db.py <IP>
```

- **rrdtool-ping.py** Programa encargado de actualizar la base de datos round robin con el tiempo de retardo de un ping realizado a una ip determinada.

```
1  #!/usr/bin/python
2  import rrdtool, time, sys
3  from ping import ping
4
5  fname = 'ping.rrd'
6  hostname = sys.argv[1]
7
8  #####
9
10 ping_time = ping(hostname)
11
12 # Se actualiza la bd con valores de [0, 1000] ms cada segundo.
13 rrdtool.update(
14     fname,
15     '-t', 'ping_%s' % hostname.replace('.', '_'),
16     '%d:%d' % (int(time.time()), ping_time)
```

17)

Uso:

```
1 $ ./rrdtool-ping.py <IP>
```

- **rrdtool-draw.py** Programa encargado de crear una grafica del tiempo de retardo de los ping's realizados a una ip determinada durante los ultimos 10 minutos.

```
1  #!/usr/bin/python
2  import rrdtool , time , random, sys
3
4  tMinute = 60
5  tHour = tMinute * 60
6  tDay = tHour * 24
7  #####
8  hostname = sys.argv[1].replace('.', '_')
9  fname = 'ping.rrd'
10 gfname = 'ping_' + hostname + '.png'
11 duration = 10 * tMinute
12 step = 1 # One second.
13 time_average = 10 # Time interval to average. (10 sec)
14 #####
15 dpoints = duration / step
16 etime = int(time.time())
17 stime = etime - dpoints
18
19 rrdtool.graph(
20     gfname,
21     '--imgformat', 'PNG',
22     '-w', '1100',
23     '-h', '300',
24     '--title', 'Pings',
25     '--vertical-label', 'Tiempo(s)',
26     '--lower-limit', '0',
27     '--start', str(stime - 1),
28     '--end', str(etime + 1),
29     'DEF:rate=%s:ping_%s:AVERAGE' % (fname, hostname),
30     'CDEF:ms=rate',
31     'VDEF:msmax=ms,MAXIMUM',
32     'VDEF:msavg=ms,AVERAGE',
33     'VDEF:msmin=ms,MINIMUM',
34     'LINE1:ms#FF0000:Retardo',
35     r'GPRINT:msmax:Max\:%6.1lfms',
36     r'GPRINT:msavg:Prom\:%6.1lfms',
37     r'GPRINT:msmin:Min\:%6.1lfms\l',
38 )
```

Uso:


```
1 $ ./rrdtool-draw.py <IP>
```

Prueba P01-P2: Ping Poller

Sistema: Topología de red, Módulo: Ping Poller			
P01-P2 Ping Poller			
Pregunta	Si	No	Observaciones
1. Verificar que se grafque correctamente.			
1.1. ¿Al ejecutar el programa rrdtool-db.py se creó el archivo ping.rrd?	X		Ninguna.
1.2. ¿Se ejecuta correctamente el programa rrdtool-ping.py?	X		Ninguna.
1.3. ¿Al ejecutar el programa rrdtool-draw.py se creó el archivo ping_< IP >.png?	X		Ninguna.
2. Fin de la Prueba.			