

RPI monitoring. Part 1

Dainius Jocas

June 7, 2013

1 Introduction

This document describes how to deploy Raspberry Pi (RPi) cluster monitoring software.

1.1 Scripts Used

- `server_jocas.py` – this script should be put on a monitor (server).
- `client_1_jocas.py` – to be deployed on the agent (client).
- `client_2_jocas.py` – to be deployed on the agent (client).
- `DJ-MIB.py` – SNMP helper script. To be deployed on the agent (client).
- `DJ-MIB` – custom MIB descriptor. To be deployed on the agent.

2 Deployment of scripts

In this section guidelines (and not step by step instructions) will be provided. Note that no description how to get/generate actual code is given.

2.1 Assumptions

Agent knows address and the port of the monitor.

2.2 On the monitor side

So far monitor is super trivial – all the information it receives script just forwards to the standard output. To run the server start `server_jocas.py` script.

2.3 On the agent side

Steps to get project running:

1. On the agent machine SNMP should be properly configured (lots of tricky work).
2. DJ-MIB should be deployed in one of the default directories for MIB. Check which are the directories can be done using command

```
# net-snmp-config --default-mibdirs
```

3. Then remove snmpd program from startup programs.
4. Having root permissions startup client_1_jocas.py script.
5. Start client_2_jocas.py script with proper parameters, e.g.

```
# python client_2_jocas.py [hostname] [port] [timeout in seconds]
```

Thats it. If everything is OK monitor should receive reports on the state of the agent.

Glossary

agent RPi which is being monitored. Also, this machine can be called "client". 1

MIB Management Information Base¹. 1

monitor RPi which monitors agents. Also, this machine can be called "server". 1

SNMP Simple Network Management Protocol ². 1

3 Part 2

Here second part of the project will be described.

¹http://en.wikipedia.org/wiki/Management_information_base

²http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

3.1 Database

To store data about RPis we decided to use SQLite database. It is because SQLite is very lightweight but at the same time it provides basic SQL functionality.

In the monitor install SQLite DBMS and python connector:

```
# apt-get install sqlite
# apt-get install python-sqlite
```

Create database file:

```
# sqlite3 monitor.db
sqlite> .tables
```

It will be created in the same directory.

Create a table rpi in the database run script create_table.py:

```
# python create_table.py
```

Columns of table rpi are: { piID piHostname piActivity piCPULoad piGateway piStorageState piIP piNetmask piProcesses piCluster piNetworkLoad piDNS}. Names of the columns are taken from DJ-MIB.

3.2 Sequence to be up and running

1. run SNMP agent
2. prepare file with list of monitors (put accurate IPs there)
3. prepare databases in monitors
4. make sure that the server is up and running
5. start client_strategy_jocas script
6. enjoy the show.

3.3 Some open issues

1. what about those bad fields like "cluster" because it is impossible to get them?
2. How to define what is going on in the RPI?
3. What would be optimal DB structure?
4. where should DB be located?
5. monitoring part? how to get data from all monitors?