

Università di Pisa
Corso di Gestione di Reti A.A.2020/21
Docente: Deri Luca
Monitoraggio risorse su Raspberry Pi3

Candidato: Firera Salvo - 578018

1 Panoramica del sistema

Il programma, implementato in Python 3.7, permette il monitoraggio di alcune risorse hardware della macchina, nello specifico, questa versione funziona su Raspberry Pi 3 e rileva:

- Memoria RAM inutilizzata
- Percentuale disco occupata
- Carico dei singoli core della CPU

Per l'implementazione del sistema è stato utilizzato il protocollo SNMP e per la gestione dei dati InfluxDB, con i suoi tool accessori Kapacitor e Chronograf che forniscono un'interfaccia grafica di monitoraggio in tempo reale oltre alla possibilità di configurare degli alert personalizzati in base a dei criteri specifici. Al termine dell'esecuzione vengono mostrati dei grafici salvati poi in .pdf nella directory ./Analytics. Il sistema è stato testato su Raspbian a 32 bit.

2 Requisiti e dipendenze

- installazione di InfluxDB, Kapacitor, Chronograf
- librerie Python Easysnmp, Matplotlib, Influxdb-client
- installazione servizio SNMP

3 Installazione SNMP

Aprire il terminale ed eseguire i seguenti comandi:

- *sudo apt-get install snmp snmpd snmp-mibs-downloader*
- aprire il file *etc/snmp/snmpd.conf* e nella sezione "ACCESS CONTROL" abilitare i percorsi per interrogare gli OIDs
- aprire il file *etc/snmp/snmp.conf* e accanto accanto alla parola "mibs" aggiungere "+ALL"
- ricaricare il servizio con *sudo service snmpd restart*
- controllare se il servizio è attivo con *sudo service snmpd status* se attivo fare una walk di prova

4 Installazione API Python

Aprire il terminale ed eseguire i seguenti comandi:

- *sudo apt-get install libsnmp-dev*
- *pip3 install easysnmp*
- *pip3 install influxdb-client*
- *pip3 install matplotlib*

5 Creazione bot Telegram per alert

- aprire Telegram e iniziare una conversazione con il profilo *BotFather*
- seguire la procedura guidata scegliendo il nome del bot e altri parametri
- salvare e custodire con cura il token che viene rilasciato, questo servirà per controllare il bot dalle api
- aprire da browser il link *https://api.telegram.org/botNOSTRO-TOKEN/getUpdates*
- aprire una chat col nuovo bot e scrivere un qualsiasi testo.
- ricaricare la pagina web prelevando dal json visualizzato il ChatID che ci servirà in seguito.

6 Configurazione UI per monitoraggio

- installare InfluxDB 1.x.x con `wget https://dl.influxdata.com/influxdb/releases/influxdb-1.8.6-amd64.deb` e `sudo dpkg -i influxdb-1.8.6-amd64.deb`
- installare Cronograf con `wget https://dl.influxdata.com/chronograf/releases/chronograf-1.8.10-amd64.deb` e `sudo dpkg -i chronograf-1.8.10-amd64.deb`
- installare Kapacitor con `wget https://dl.influxdata.com/kapacitor/releases/kapacitor-1.5.9-1-amd64.deb` e `sudo dpkg -i kapacitor_1.5.9-1_amd64.deb` avviare Kapacitor con `sudo service kapacitor start`
- aprire il browser all'indirizzo `http://localhost:8888`
- cliccare su "Get started" e collegare il database locale inserendo i dati richiesti (ID e password sono visibili nel codice python)
- collegare il database a Kapacitor lasciando vuoti i campi user e password.

7 Configurazione alert

Andare nella sezione "Alerting" della barra laterale che sarà divisa in due sezioni. Nella sezione in alto "Alert rules" si definiscono le regole secondo cui mandare gli alert. In quella sotto si definiscono degli handler per gestire gli alert, nel nostro caso andremo a selezionare Telegram e compariranno due campi in cui inserire il token e il ChatID ricavati al punto 5. Con questa procedura creiamo gli alert di nostro interesse.

N.B. Gli alert restano in funzione anche chiudendo il browser finché lo script rimarrà in esecuzione.

8 Utilizzo del sistema

Dopo aver seguito le procedure di cui sopra, possiamo eseguire il programma:

- da terminale lanciamo `python3 sysMonitor_ARM.py` si aprirà automaticamente la pagina web `http://localhost:8888`
- nella barra a sinistra andiamo "Boards" e da qui possiamo seguire la procedura grafica per creare qualsiasi grafico prelevando le informazioni che lo script manda al bucket attraverso la API
- nel frattempo gli alert task si occuperanno di controllare che i record inseriti siano al di sotto (o al di sopra) delle soglie selezionate, eventualmente mandando degli alert sul bot Telegram.
- Ovviamente dobbiamo avviare il bot sul nostro telefono avviando una chat e digitando `/start`
- per terminare il programma, sul terminale digitiamo `ctrl+c`

Alla chiusura il database Influx viene eliminato.

9 Rilevamento dei dati

Il rilevamento dei parametri delle risorse viene fatto utilizzando il protocollo SNMP visto durante il corso, consiste nell'interrogare ad intervalli regolari degli agenti (MIBS) tramite la funzione *snmpget*. I MIBS in questione sono i seguenti:

- iso.3.6.1.4.1.2021.4.11.0: RAM inutilizzata.
- iso.3.6.1.4.1.2021.9.1.9.1: percentuale disco occupata
- iso.3.6.1.2.1.25.3.3.1.2.core: percentuale carico dei core della CPU

Questi OIDs sono stati presi dal sito *oidref.com*. Il protocollo SNMP può essere utilizzato all'interno di applicativi Python tramite la libreria EasySNMP.

10 Salvataggio dei dati

I valori rilevati tramite SNMP vengono immagazzinati in un database no-SQL. In questo caso è stato utilizzato InfluxDB, un database a serie temporali che oltre allo storage di dati offre una serie di tool come Kapacitor e Chronograph che permettono un accesso intuitivo tramite UI dal browser web. Il programma crea un database e a intervalli regolari scrive i valori letti tramite SNMP. Il database è raggiungibile sulla porta 8086 del localhost, mentre l'interfaccia grafica (offerta da Chronograf) è utilizzabile collegandosi alla porta 8088. Kapacitor fa da intermediario tra il database (locale) e l'interfaccia web di monitoraggio. Quando si termina l'applicazione viene eliminato il database. Per architetture ARM è disponibile solo la versione 1.X di influxDB che necessita di Chronograf e Kapacitor, nella versione 2 questi servizi sono integrati in InfluxDB.

11 Alerts personalizzabili

Per mandare gli alert è stata fatta un collegamento con un bot Telegram, automatizzando l'invio di messaggi quando i parametri superano (nel caso della ram sono al di sotto) di un certo valore. Sul Raspberry, avendo risorse limitate, gli alert possono risultare particolarmente frequenti, soprattutto la RAM. Gli alert possono essere inviati ad un dispositivo soltanto, questo viene impostato inserendo il ChatID della finestra di configurazione dell'handler.

12 Link utili

EasySNMP: <https://pypi.org/project/easysnmp/>

Matplotlib: <https://matplotlib.org/>

InfluxDB-client: <https://docs.influxdata.com/influxdb/cloud/tools/client-libraries/python/>

InfluxDB Tools: <https://portal.influxdata.com/downloads/>

OID Ref: <https://oidref.com/>

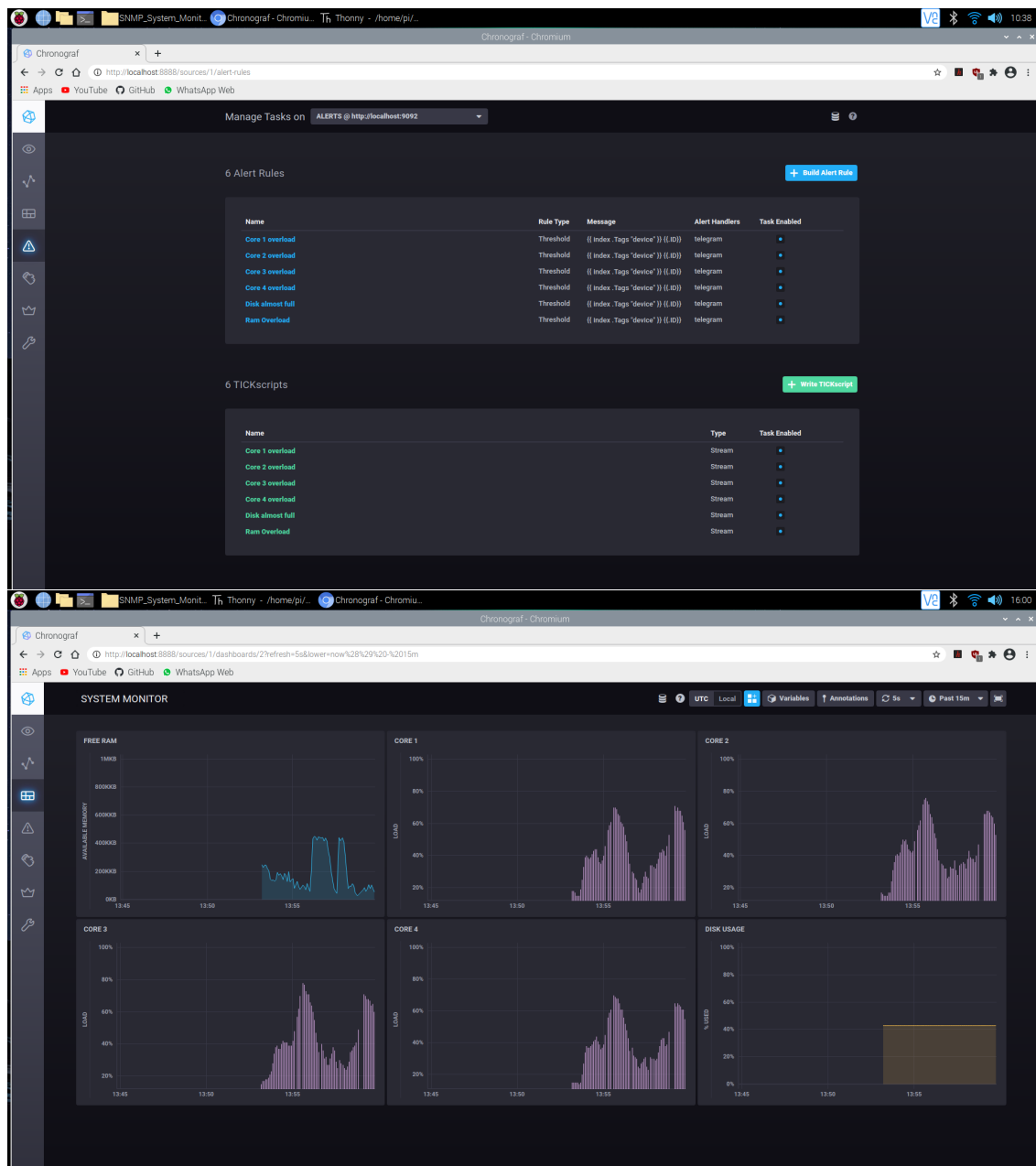


Figure 1: Interfaccia Web di Chronograf