

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.

## 2 Requisiti e dipendenze

- installazione di InfluxDB, Kapacitor, Chronograf
- che siano attivi i servizi InfluxDB, Kapacitor e Chronograf
- librerie Python Easysnmp, Matplotlib, Influxdb-client
- installazione servizio SNMP

### 3 Esecuzione

- avviare i servizi Influx, Kapacitor e Chronograf
- eseguire il comando `python3 sysMonitorARM.py`
- su Telegram avviare il bot @InfluxdbNotifybot
- aprire il browser sulla pagina `http://localhost:8088`
- nella barra laterale andare sulla dashboard per il monitoraggio in tempo reale
- per terminare l'esecuzione digitare `ctrl+c`

N.B. Gli alert su Telegram arrivano solamente a un dispositivo per volta che va configurato nell' Handler dell'alert tramite il ChatID.

### 4 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.

### 5 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.

## 6 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. Le notifiche Telegram funzionano anche se non si apre il pannello di controllo dal browser.

## 7 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/>

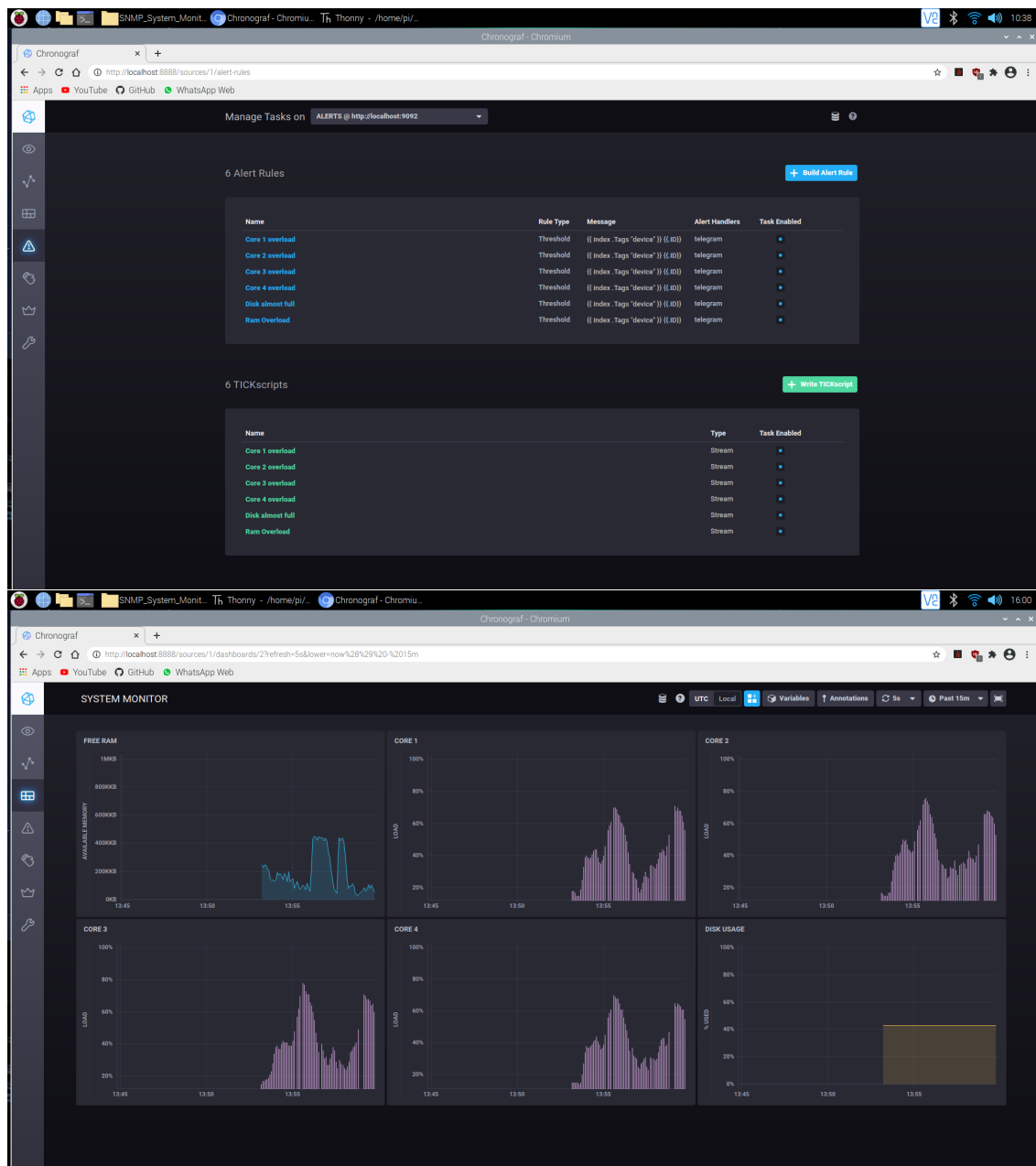


Figure 1: Interfaccia Web di Chronograf