

Documentazione per monitor_connection.py

Enrico Cornacchia
Matricola: 0001070824

14 maggio 2024

1 Introduzione

`monitor_connection.py` è uno script Python che monitora la connettività di una lista di indirizzi IP forniti dall'utente. Lo script esegue un ping a ciascun indirizzo IP e stampa se l'host è online.

2 Requisiti

- Librerie Python: `subprocess`, `platform`, `concurrent.futures`

3 Funzionamento dello script

Lo script utilizza il modulo `subprocess` per eseguire il comando `ping` su ciascun indirizzo IP. Il modulo `concurrent.futures` viene utilizzato per eseguire i ping in parallelo utilizzando un pool di thread.

Le funzioni principali dello script sono:

- `is_valid_ip(ip)`: Verifica se un indirizzo IP è valido.
- `ping_host(ip)`: Esegue un ping a un host e ritorna `True` se l'host risponde, `False` altrimenti.

3.1 Funzione `ping_host`

La funzione `ping_host` esegue un ping a un host specificato da un indirizzo IP. Funziona così:

1. Controlla se l'indirizzo IP fornito è valido usando `is_valid_ip`. Se l'indirizzo IP non è valido, stampa un messaggio di errore e ritorna `False`.
2. Imposta il parametro per il comando ping in base al sistema operativo. Su Windows, il parametro è `-n`, mentre su altri sistemi operativi (come Linux o MacOS) il parametro è `-c`.

3. Crea il comando ping come una lista di stringhe, composta dal comando `ping`, il parametro impostato al passo precedente, il numero di ping da eseguire (1 in questo caso visto che vogliamo solo controllare se è online o no l'host) e l'indirizzo IP dell'host.
4. Tenta di eseguire il comando ping utilizzando il modulo `subprocess`. Cattura l'output del comando e imposta un timeout di 5 secondi.
5. Se il comando ping riesce, ritorna `True`. Se il comando ping fallisce o se il timeout scade, stampa un messaggio di errore e ritorna `False`.

La funzione `ping_host` è progettata per essere eseguita in un thread separato per ogni host, permettendo di monitorare più host contemporaneamente.

3.2 Esecuzione del ping agli host

Dalla linea 27 alla linea 39 dello script, viene eseguito il ping agli host specificati dall'utente e viene stampato se ogni host è online o offline. Ecco una descrizione dettagliata del suo funzionamento:

1. Alla linea 27, lo script chiede all'utente di inserire gli indirizzi IP degli host da monitorare, separati da virgole. Gli indirizzi IP vengono poi separati e memorizzati in una lista.
2. Alla linea 29, viene creato un pool di thread utilizzando il modulo `concurrent.futures`. Questo permette di eseguire il ping a più host contemporaneamente.
3. Alla linea 30, per ogni host nella lista, viene sottomesso un task al pool di thread per eseguire la funzione `ping_host`. I Future restituiti dalla funzione `submit` vengono memorizzati in un dizionario, con l'indirizzo IP come valore associato.
4. Dalla linea 31 alla linea 39, per ogni Future nel dizionario (in ordine di completamento), viene controllato se l'host è online o offline. Se il Future restituisce `True`, l'host è online e viene stampato un messaggio appropriato. Se il Future restituisce `False` o se si verifica un'eccezione, l'host è offline o non raggiungibile e viene stampato un messaggio di errore.

4 Utilizzo dello script

Per utilizzare lo script, eseguire `python monitor_connection.py` da una shell o un terminale.

4.1 Esempio di input:

Quando viene richiesto, inserire gli indirizzi IP degli host da monitorare, separati da virgole.

```
Inserisci gli indirizzi IP degli host da monitorare, separati da virgole: 192.0.2.0, 8.8.8.8, 1.1.1.1
```

4.2 Esempio di output:

Ritorna lo status degli host inseriti in input

```
8.8.8.8 online  
1.1.1.1 online  
192.0.2.0 offline
```

5 Considerazioni aggiuntive

- **Gestione degli errori:** Lo script gestisce correttamente vari tipi di errori, come indirizzi IP non validi, timeout scaduti ed eccezioni generiche. Questo rende lo script robusto e affidabile.
- **Compatibilità multi-piattaforma:** Lo script è progettato per funzionare sia su Windows che su sistemi Unix-like (come Linux o MacOS). Questo è ottenuto utilizzando il modulo `platform` per determinare il sistema operativo e impostare il parametro appropriato per il comando `ping`.
- **Esecuzione parallela:** Lo script utilizza il modulo `concurrent.futures` per eseguire il ping a più host contemporaneamente. Questo può migliorare significativamente le prestazioni quando si monitorano molti host.
- **Miglioramenti futuri:** Attualmente, lo script supporta solo indirizzi IP IPv4. Potrebbe essere esteso per supportare anche indirizzi IP IPv6. Inoltre, potrebbe essere utile aggiungere la possibilità di leggere gli indirizzi IP da un file o da un database, invece di inserirli manualmente.