

# Mesure de variation de la charge des CPU en fonction du temps

## 1 Références

Ceci est un document regroupant les différentes sources utilisées pour effectuer la mesure de variation de la charge des CPU en fonction du temps, le document est divisé en deux parties les références utiles et les références non-utiles.

### 1.1 Références utiles

- Dans [10] l'auteur nous explique comment utiliser Gnuplot qui est un utilitaire graphique portable basé sur la ligne de commande pour Linux. ça m'a été utile pour le traçage des traces récupérées sur l'utilisation des CPU.

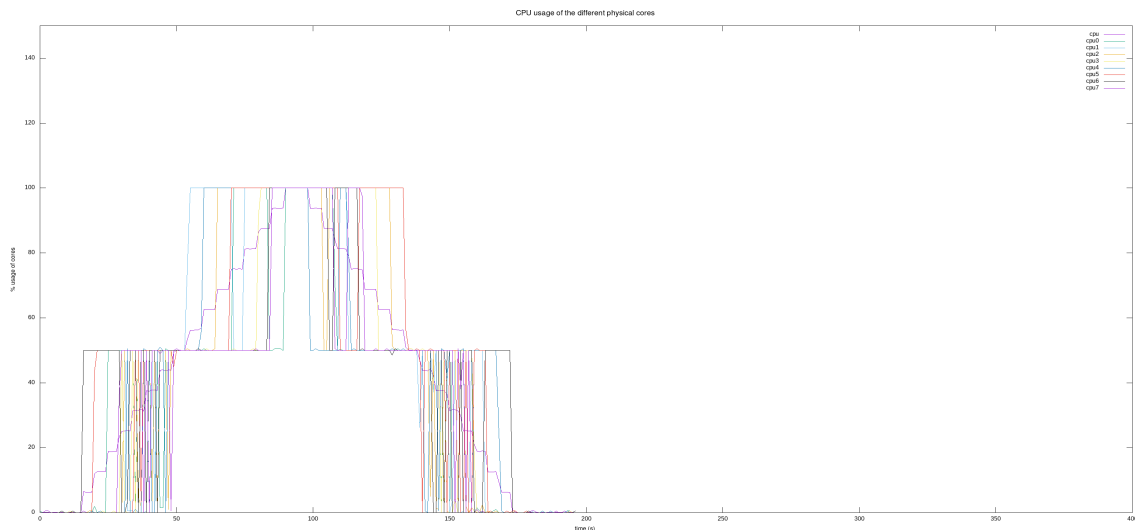


FIGURE 1 – Exemple d'utilisation de gnuplot pour tracer le taux d'utilisation

- Dans [9] l'auteur nous explique le langage bash et sa syntaxe c'est similaire avec la référence [8] mais la différence est la notion des tableaux, on trouve la manipulation des tableaux en bash dans ce livre et pas dans l'autre ce qui m'a aidé pour réaliser le script qui gère les applications en particulier pour arrêter un processus déjà lancé à partir du script.
- Dans [8] l'auteur nous explique les syntaxes pour bien écrire un scripte shell et il nous explique aussi les flux d'entrées sorties, dans les deux chapitres on y trouve les éléments suivants :
  1. Script shell :
    - faire la différence entre shell interactif et non-interactif
    - écrire un script shell (type sh ou bash) en utilisant les structures de contrôle for, case, if, while
    - définir et utiliser les variables
    - utiliser les arguments de la ligne de commande
    - travailler en mode interactif avec l'utilisateur à l'aide de la commande interne read
    - utiliser la commande test (ou `[]`) pour vérifier des conditions, en particulier sur les comparaisons de chaînes ou de nombre et sur les statuts d'un fichier
    - utiliser les fonctions et les appels d'autres scripts au sein d'un script
    - gérer les valeurs de retour de commande et en générer

- définir une variable d'environnement et utiliser les fichiers d'initialisation `.profile`, `.bashrc` pour paramétrer le comportement par défaut de son shell

## 2. Flux et filtres :

- décrire ce qu'est une entrée standard, une sortie standard et une erreur standard
- d'utiliser le caractère pipe `()` pour rediriger la sortie d'une commande en entrée d'une autre commande
- d'utiliser les caractères `>` `<` `>>` `<<` `EOF` `>&2>2` pour rediriger les flux
- utiliser les filtres `sort` et `cut` avec les séparateurs et les commandes pour trier sur une colonne ou couper plusieurs colonnes
- utiliser les filtres `cat`, `wc`, `uniq`, `grep`, `sed` et `tr`

cette référence m'a aidé pour réaliser le script qui génère le fichier log et de plus le script qui gère les applications à lancer pour charger les CPU's.

- Dans [6] l'auteur nous comment utiliser le protocole `sftp` pour exploiter les fichier d'un serveur à distance.

```

1      $ sftp user_name@server_name
2      Connected to server_name
3      sftp> dir
4      file1 file2 file3
5      sftp> pwd
6      Remot worjng directory: /home/etis/tests
7      sftp> get file2
8      Fetching /home/etis/tests/file2 to file2
9      /home/etis/tests/file2
10     sftp> bye
11

```

Listing 1 – Récupérer un fichier depuis le serveur en utilisant `sftp`

- Dans [4] l'auteur nous donne une solution pour déterminer les coeurs logiques qui forment un coeur physique et aussi de savoir combien de coeurs on en a. Ceci en utilisant le fichier `/proc/cpuinfo`.
- Dans [3] l'auteur nous explique comment calculer la variation de la charge des CPU en fonction du temps en utilisant le fichier `/proc/stat`, aussi il nous explique la différence entre cette méthode et la commande `top`.
- Dans [2] l'auteur nous donne des explications et des exemples d'utilisation de la commande `sleep` en bash, celle-ci est utilisé pour le calcule le la variation de la charge des CPU en fonction du temps et permet de faire une pause pour un délais définit entre les différentes captures.
- Dans [1] l'auteur nous fournis plusieurs commandes/scripts qui vont charger le serveur pour faire les tests de consommation. Donc on en déduit que l'utilisation effective peut être calculée comme la somme de `%us`, `%sy`, `%ni`, `%hi` et `%si`.

## 1.2 Références non-utiles

- Dans [7] on trouve un guide plus détaillé sur la programmation en bash. Mais ceci était déjà abordé sur les références [8] et [9].
- Dans [5] on trouve divers commandes qui font le calcul du taux d'utilisation des CPU, mais ceci n'était pas très utile car en passant par un script et en utilisant les flux et la commande `sleep` le temps sera légèrement retardé. Donc il fallait exploiter directement le fichier `/proc/stat`.

**AYAD Ishak**

## Références

- [1] [ linux ] créer une charge cpu / load. <https://www.djerfy.com/linux-creer-une-charge-cpu-load/>.
- [2] Bash comment mettre en veille ou retarder un laps de temps spécifié. <https://www.cyberciti.biz/faq/linux-unix-sleep-bash-scripting/>.
- [3] Calculer le taux d'utilisation de la cpu, comparaison entre la commande top et le fichier /proc/stat. <https://medium.com/@yogita088/how-to-calculate-cpu-usage-proc-stat-vs-top-e74f99f02d08>.
- [4] Chek cpu info. <https://jin-yang.github.io/reference/linux/monitor/check-cpu-info-linux.pdf>.
- [5] Command line tools to check cpu usage in linux. <https://linoxide.com/monitoring-2/10-tools-monitor-cpu-performance-usage-linux-command-line/>.
- [6] Logging in to a remote system to copy a file (sftp). [https://docs.oracle.com/cd/E26502\\_01/html/E29001/remoteho14.html](https://docs.oracle.com/cd/E26502_01/html/E29001/remoteho14.html).
- [7] Mendel Cooper. Advanced bash scripting guide. <http://tldp.org/LDP/abs/html/>.
- [8] Tuyêt Trâm DANG NGOC. *Pogrammmation système*, chapter "Scripts shell, Flux et filtros". Université de Cergy Pontoise, Cergy, FR, 2018.
- [9] Eric Sanchis. *Introduction à la programmation en Bash*. Université de Toulouse 1 Capitole, Toulouse, FR, 2015.
- [10] Colin Kelley Thomas Williams. *An Interactive Plotting Program*. 2017.