

Chapitre 2 de nos chroniques

Objectifs : cloner le système sur les cartes microSD, et tester les performances de notre Raspberry Pi (RPi)

Acte 1 - Clonage du système

Initialement nous comptions utiliser un logiciel tel que FileZilla afin de cloner intégralement le système de fichiers mis en place sur le premier RPi, mais afin d'optimiser cette partie William et moi avons opté pour l'écriture d'un script; un script est un programme qui permet d'automatiser une suite de commandes (cf. chapitre 1).

Étape 1 : Historique des commandes

Faire l'historique de toutes les démarches effectuées lors de la première installation.

Étape 2 : Écriture d'un premier script

Nous écrivons toutes les commandes dans un fichier d'extension .sh afin qu'il puisse être lancé depuis un terminal.

Étape 3 : Test du script

Voici le moment du grand saut! Et bien évidemment.. le terminal nous insulte avec véhémence. Les premiers problèmes détectés sont des lignes non commentées et quelques erreurs lors de la rédaction.

Étape 4 : Correction des bogues

Après quelques correctifs nous avons relancé un test. À suivre..

Acte 2 - Test de performance

Dans le but de déterminer le plus précisément possible les performances de calcul des RPi, pour ensuite les comparer avec celles de notre futur data center, nous avons décidé d'une unité de référence : le Pi Performance Factor (PPF).

Étape 1 : Téléchargement des premiers modules Python nécessaires

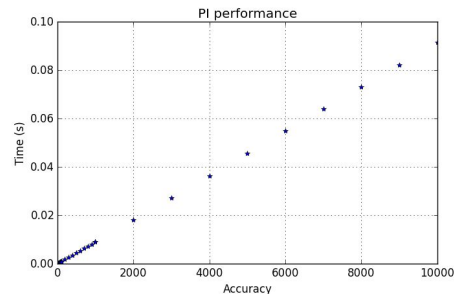
Les modules qui seront nécessaires sont *matplotlib* ainsi que *numpy*.

Étape 2 : Choix du test

Le test de performance choisi est un programme python qui calcule une valeur approchée de Pi. Pour ce faire nous utilisons la méthode des trapèzes entre 0 et 1 sur la dérivé de la fonction $4 \cdot \arctan(x)$ qui est : $4/(1+x^2)$. Car $4 \cdot \arctan(1) = \pi$.

Étape 3 : Affichage des résultats

En faisant varier la précision (Accuracy) nous observons le temps de calcul (Time). La précision n correspond au nombre de division de l'intervalle dans la méthode des trapèzes, avec un pas de $1/n$.



Étape 4 : Détermination du PPF

Nous avons observé que les différentes mesures suivent un modèle linéaire, ainsi nous avons fait une régression linéaire. Pour ce faire nous avons installé un module Python : *sklearn*. Nous avons déterminé le coefficient de corrélation ainsi que le coefficient directeur de la droite. Et en pensant au fait que plus un ordinateur est puissant en terme de calcul plus le coefficient sera faible alors notre unité le PPF est l'inverse de ce coefficient directeur (divisé par 10^3 afin d'avoir une unité assez grande).

Premiers résultats obtenus :

- RPi2 - 109ppf
- L'antiquité de Bastien - 754ppf
- L'avion de chasse de William - 2200ppf

PS : L'algorithme python de test du PPF (pi_performance.py) est disponible sur <https://github.com/ZeromusSoftware/RPi3500?files=1> pour les curieux et ceux qui souhaitent tester leur machine.

Prochainement sur Propulse :

- Le résultat du script de configuration si on arrive à choper une carte SD (une petite pensée pour ce fourbe de Quentin qui les retient en otage)
- Les premiers pas dans Hadoop inch'allah