

Évaluer la consommation d'un ordinateur

1 Programmes calculant intensivement

QUESTION 1 ► Notez la consommation de votre ordinateur avec aucune autre application s'exécutant que votre terminal :

1.1 Bubble sort / Tri à bulles

Un très mauvais programmeur réalise le programme de tri de données suivant.

```

#define MAX 50000
#include <stdio.h>
#include <stdlib.h>

int * tableau = NULL ;

void bubble_sort(int list[], long n)
{
    long c, d, t;

    for (c = 0 ; c < ( n - 1 ); c++)
    {
        printf ("Etape %ld\n",c);
        for (d = 0 ; d < n - c - 1; d++)
        {
            if (list[d] > list[d+1])
            {
                /* Swapping */
                t = list[d];
                list[d] = list[d+1];
                list[d+1] = t;
            }
        }
    }
}

void creer_tableau()
{
    int i;

    for (i=0;i<MAX;i++)
    {
        tableau = realloc (tableau, i * sizeof(int ) );
        tableau[i]=MAX-i;
        printf("--> Allocation tableau case : %d -> %d\n",i,tableau[i]);
    }
}

int main()
{ int i;

```

```

creer_tableau();
//for (i=0;i<MAX;i++) printf("%d\n",tableau[i]);
printf ("Tableau cree attente de 5 secondes\n");
sleep(5);
printf("Debut du tri\n");

bubble_sort(tableau, MAX);

printf("Fin du tri\n");

for (i=0;i<MAX;i++) printf("%d\n",tableau[i]);

free(tableau);
return 0;
}

```

QUESTION 2 ► Récupérez, compilez et exécutez l'application de tri **bubble sort**. Observez les opérations effectuées par cette application. Si besoin, modifier le nombre MAX d'éléments pour que l'application dure quelques secondes.

QUESTION 3 ► Comme la dernière fois, afin d'observer le profil énergétique de cette application, vous allez devoir vous organiser avec quelqu'un d'autre : pour ne pas perturber la mesure, il faut interroger le PDU depuis une autre machine que celle exécutant l'application.

1.2 Quick sort

Au lieu de trier des bulles, trions en mode *quick* ! Un bon programmeur réalise le programme suivant :

```

#define MAX 300000
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int tableau[MAX];

void quick_sort (int *a, int n) {
    if (n < 2)
        return;
    int p = a[n / 2];
    int *l = a;
    int *r = a + n - 1;
    while (l <= r) {
        if (*l < p) {
            l++;
        }
        else if (*r > p) {
            r--;
        }
        else {
            int t = *l;
            *l = *r;
            *r = t;
            l++;
            r--;
        }
    }
    printf ("Etape %d\n",n);

    quick_sort(a, r - a + 1);
    quick_sort(l, a + n - l);
}

int main ()
{

```

```

int i;
for (i=0;i<MAX;i++)
{
    tableau[i]=rand()*i;
    printf("--> Remplissage  tableau case : %d  valeur : %d\n",i,tableau[i]);
}

sleep(3);
printf("DÃ©but du tri\n");

quick_sort(tableau, MAX);

printf("Fin du tri\n");

for (i=0;i<MAX;i++) printf("%d\n",tableau[i]);

return 0;
}

```

QUESTION 4 ► Récupérez, compilez et exécutez l'application de tri **quick sort** sur un volume suffisant de données pour que l'application dure quelques secondes. Observez le profil énergétique de cette application. Quel est l'impact de la fonction `creer_tableau()` ?

QUESTION 5 ►

A l'aide de deux extractions de logs, calculez la consommation énergétique des deux applications (en joules). Quelle est l'application la plus consommatrice en énergie ?

QUESTION 6 ► Quelle est la consommation en kwh ?

QUESTION 7 ► Quel est l'eq. carbone ?

QUESTION 8 ► Quel est le coût en euros associé, en France, et en supposant un tarif Enercoop récent : 0.1717 euros du kWh ?

QUESTION 9 ► Quel est l'impact des affichages (`printf()`) sur le profil énergétique ? (commentez les lignes pour voir la différence)

2 Modification de la fréquence CPU

QUESTION 10 ► Vous pouvez changer la fréquence du CPU ; moins il calcule vite, moins il consomme. Sous Linux, nous pouvons pour cela utiliser un ensemble d'utilitaires appelé `cpupower`. Lisez l'encart 1 pour des détails sur le fonctionnement du changement de fréquence, puis :

- Indiquez le nombre de processeurs et de cœurs de votre machine :
- Indiquez les fréquences CPU minimale et maximale :
 - Min :
 - Max :

Frame 1: Changement de fréquence du CPU sous Linux

D'abord, pour connaître le nombre de cœurs disponibles sur votre machine : `cat /proc/cpuinfo`
Ensuite, il existe sous Linux un composant logiciel appelé *governor* en charge de régler la fréquence du CPU. Différents types de *governors* existent selon le comportement que vous voulez. Par exemple, pour qu'un utilisateur ait le droit de changer la fréquence CPU, il faut utiliser un certain *governor*. Les différents types disponibles sur votre machine sont listables grâce à la commande `cpupower frequency-info`. Lorsque le noyau est configuré comme il faut (cf. annexes), vous pouvez utiliser les *governors* suivants

userspace : Permet un programme utilisateur de positionner la fréquence CPU ;

powersave : Positionne la fréquence CPU au minimum;

performance : Positionne la fréquence CPU au maximum ;

ondemand : Utilise une heuristique pour adapter la fréquence CPU à la charge ;

conservative : Utilise une heuristique pour adapter la fréquence CPU à la charge, mais l'augmente et la diminue doucement plutôt que de changer d'un seul coup ;

schedutil : Utilise encore une autre heuristique pour adapter la fréquence CPU à la charge. Cette heuristique utilise certaines connaissances de l'ordonnanceur ;

On ne va pas tous les utiliser dans ce TP mais si vous voulez plus d'infos : <https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt>

Pour changer la fréquence CPU :

- sélectionner le *governor* : `cpupower frequency-set --governor userspace`
- sélectionner la fréquence: `cpupower --cpu all frequency-set --freq 800MHz`

Pour avoir des infos sur ce qui est fait en ce moment, lisez `/sys/devices/system/cpu/cpu*/cpufreq/` :

`scaling_driver` est le driver utilisé.

`scaling_available_governors` liste les *governors* disponibles pour le driver utilisé ;

`scaling_available_frequencies` liste les fréquences disponibles avec votre CPU ;

`cpuinfo_min_freq` et `cpuinfo_max_freq` donne les fréquence min et max disponibles.

QUESTION 11 ► Que fait le script Shell ci-dessous ? Copiez ces ligne dans un fichier `.sh` au nom adapté. Adaptez-le pour qu'il ait du sens sur votre machine. Notamment, à la place de **2.1GHz**, indiquez la fréquence max de vos CPU ; et aussi le nombre de cœurs (8 dans le cas de ce script) ne correspond peut-être pas.

QUESTION 12 ► Exécutez ce script, vérifiez dans les fichiers de `/sys/devices/...` qu'il a effectivement un effet, puis mesurez la consommation de votre machine.

```
#!/bin/bash

for i in 0 1 2 3 4 5 6 7
do
    sudo cpufreq-set -c $i -g performance
    sudo cpufreq-set -c $i -u 2.1GHz
done
```

QUESTION 13 ► Écrivez un autre script Shell positionnant la fréquence de tous les cœurs à la fréquence minimum. Quelle consommation électrique observez-vous ?

3 Mémoire et entrées-sorties

QUESTION 14 ► Écrivez un petit programme qui, selon vous, utilise au maximum la mémoire et mesurez la consommation de votre machine pendant l'exécution de ce programme. Notez-la ci-après :

QUESTION 15 ► Utilisez la commande `stress` pour utiliser au maximum la mémoire (au besoin, utiliser `man stress` pour avoir des infos). Arrivez-vous au même résultat qu'avec votre programme ?

QUESTION 16 ► Écrivez un petit programme qui, selon vous, utilise au maximum le disque dur, et mesurez la consommation de votre machine pendant l'exécution de ce programme.

QUESTION 17 ► Utilisez la commande `stress` pour utiliser au maximum les entrées-sorties. Arrivez-vous au même résultat qu'avec votre programme ?

QUESTION 18 ► Utilisez la commande `stress` pour utiliser au maximum le CPU, la mémoire, et les entrées-sorties. À quelle puissance maximum arrivez-vous ?

4 Applications

QUESTION 19 ► Notez la consommation de votre machine lorsque tous les cœurs sont surchargés (grâce à la commande `stress`) :

QUESTION 20 ► Mesurez la consommation lors de l'utilisation de diverses applications locales

- Lecture vidéo :
- Lecture musique :
- Autres... :

QUESTION 21 ► Petite compétition gentille pour tout le groupe : quelle visite de site web fait grimper la puissance consommée par votre machine ?

1 Annexe A : préparation des machines

- Installer le paquet `cpufrequtils`
Attention, selon les distributions Linux, les commandes fournies peuvent être `cpufreq` plutôt que `cpupower` !
- Sur les machines Intel, il y a de fortes chances que le driver utilisé soit `intel_pstate`, qui n'autorise que l'utilisation des gouverneurs `powersave` et `performance`. Le gouverneur `userspace` est disponible seulement avec le driver plus vieux `acpi-cpufreq`. Donc il faut dans ce cas :
 - désactiver le driver intel : ajouter `intel_pstate=disable` à la commande de lancement du noyau. C'est à dire qu'il doit y avoir une ligne `GRUB_CMDLINE_LINUX="intel_pstate=disable acpi=force"` dans le fichier `/etc/default/grub`
 - Démarrer et charger le module `userspace` : `modprobe cpufreq_userspace`
 - Le fichier `/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors` doit contenir la ligne suivante (les gouverneurs disponibles) :
`conservative ondemand userspace powersave performance schedutil`