TP ANF ENVOL 2023
Mesurer et comprendre la consommation électrique d'un nœud de calcul - Eco-concevoir un code

Vladimir Ostapenco, Laurent Lefèvre (avec qq sections du TP de Julien Lefèvre)
Inria Avalon - Laboratoire LIP - ENS de Lyon - 46 allée d'Italie 69364 LYON Cedex 07
vladimir.ostapenco@ens-lyon.fr, laurent.lefevre@ens-lyon.fr

Table des matières

1 Introduction

Dans cette séance de TP nous allons travailler sur la plate-forme Grid'5000 pour mesurer la consommation électrique d'une machine de calcul lors de différentes phases d'usage. Dans la première partie, vous allez apprendre à utiliser Grid'5000. Dans la deuxième partie, vous allez mesurer et comprendre la consommation énergétique d'un nœud de calcul et de ses différents composants.

Les questions ont un code couleur qui précise si vous travaillez sur la frontale de Grid'5000, si vous travaillez sur votre nœud réservé ou si vous travaillez sur votre machine locale.

2 Créer un compte

Vous avez dû recevoir un mail de Grid'5000 vous demandant de créer un compte pour accéder à la plate-forme.

Question 1
Générez une clef ssh. Pour cela vous utiliserez la commande suivante
`ssh-keygen -t rsa`
qui crée deux fichiers dans le dossier caché `./ssh`. `id_rsa` est la clef privée (à garder secrète) et `id_rsa.pub` la clef publique que vous transmettrez à Grid'5000.

Question 2
Ne lisez pas toutes les instructions du mail mais suivez le lien proposé pour créer votre compte. Vous aurez besoin de transmettre votre clef `id_rsa.pub` au cours du processus.

3 Prise en main

3.1 Description de l'environnement

Grid'5000 est un banc de test ("testbed" en anglais). Il s'agit d'une plate-forme expérimentale informatique qui permet de réaliser des tests rigoureux sur des outils de calcul numérique haute performance, notamment en calcul parallèle et distribué.

Grid'5000 s'organise de la manière suivante :
— 8 sites en France
— 800 noeuds localisés à Grenoble (92), Lille (39), Luxembourg (16), Lyon (132), Nancy (189), Nantes (70), Rennes (173), Sophia (44)
— environ 15000 coeurs au total

Vous pourrez accéder à toutes les caractéristiques techniques du matériel ici.

3.2 Quelques règles simples

Les règles d'utilisation sont détaillées ici. Pour résumer très rapidement, vous ne devez utiliser la plate-forme que pour des usages relatifs au TP proposé [1].

3.3 Première connexion

Le schéma représenté sur cette page vous donne un aperçu de l'architecture de Grid'5000.

1. L'usage de la plate-forme est contrôlé par différents outils. Par exemple le minage de crypto-monnaies sera détecté.

Question 3
Dans un terminal, connectez vous à une machine d'accès via la commande
`ssh <login>@access.grid5000.fr`
puis connectez vous à la frontale (ou *frontend*) de Lyon via la commande
`ssh lyon`

Question 4
Assurez vous que votre répertoire courant ne contient qu'un dossier `public` lui-même contenant un `README.txt` dont vous prendrez connaissance du contenu.

Question 5
A l'aide de la commande `scp` **utilisée depuis votre machine**, copiez le fichier de votre choix vers le dossier `public` de la machine frontale de Grid'5000 de votre choix. Le répertoire de destination sera de la forme :
`<login>@access.grid5000.fr:lyon/public/`
Vérifiez que l'opération s'est passée sans encombre.

3.4 Réservation et utilisation des ressources

L'utilisation des ressources de Grid'5000 est sujette à certaines contraintes et notamment le fait qu'il faut partager ces ressources avec d'autres utilisateurs. Des outils de visualisation (vue générale) permettent de connaître l'utilisation des ressources à un instant donné (avec Monika, voir l'exemple de Lyon) ou sur une certaine durée (avec Gantt, voir ici).

Dans le cadre de ce TP, il est important d'avoir réservé au préalable un certain nombre de clusters pour que le TP puisse bien se passer. A ce stade, vos enseignants se seront chargés de cette tâche. Vous allez donc travailler au sein de ressources réservées appelées *container job* et repéré par un `JOB_ID` qui vous sera donné au début du TP [2].

La réservation de ressources passe par OAR qui est un utilitaire développé par Inria et très utilisé dans le calcul haute performance.

Question 6
Repérez les différents jobs associés à l'utilisateur `llefevre` avec la commande
`oarstat`
En utilisant la visualisation Monika ou Gantt, trouvez le nombre de noeuds utilisés et la durée.

Question 7
Réservez chacun.e **un seul** noeud pour toute la durée du TP via la commande :
`oarsub -t deploy -t inner=<JOB_ID> -l host=1,walltime=03:00:00 "sleep infinity"`

Vous devrez voir s'afficher un texte proche de celui-là

```
# Filtering out exotic resources (gemini, sirius, pyxis, neowise).
OAR_JOB_ID=1379098
```

indiquant que votre réservation est traitée pour être ajoutée dans la file.

2. Si vous voulez en savoir plus, vous pourrez consulter la documentation sur les container jobs.

Question 8
Trouvez le nœud sur lequel votre réservation a été affectée à la commande `oarstat -j <OAR_JOB_ID> -f`.
Le nom du nœud doit être dans la variable `assigned_hostnames`.
Si cette variable n'a pas de valeur, vous devrez attendre que le statut de votre réservation devienne `Running`.
Sur quel nœud votre réservation a-t-elle été placée ?

3.5 Déploiement de l'image du système d'exploitation

Pour ce TP, nous avons préparé une image minimale du système d'exploitation contenant tous les outils et bibliothèques nécessaires.

Question 9
Déployez l'image de l'OS préparée par Vladimir Ostpenco avec l'outil `kadeploy3` disponible sur Grid'5000.
`kadeploy3 -m <NOM_DU_NOEUD> -e ubuntu2004-tp2-cr17 -u vostapenco -k`
Le déploiement de l'image ne doit pas prendre plus de quelques minutes.
Lorsque le déploiement est terminé, vous pouvez le tester en vous connectant au nœud avec la commande `ssh root@<NOM_DU_NOEUD>`.

Question 10
Une fois connecté au nœud réservé, récupérez l'heure actuelle et mémorisez-la, nous en aurons besoin à la fin du TP.
Vous pouvez récupérer l'heure actuelle avec la commande `date +%Y-%m-%dT%H:%M:%S`.

3.6 Premier test de stress

Dans cette section, vous effectuerez votre premier test qui mettra une charge importante sur le processeur (CPU) de nœud.

Question 11
Utilisez la commande `stress` et visualisez la charge du processeur avec la commande `htop`.
Obtenez et enregistrez l'heure de début et l'heure de fin de l'expérience.
Pour obtenir l'heure actuelle, utilisez la commande suivante `date +%Y-%m-%dT%H:%M:%S`.
Pour charger le processeur, utilisez la commande `stress -c 32 -t 10`.

3.7 Visualisation de la consommation

Dans cette section, vous allez visualiser les données de consommation énergétique fournies par les wattmètres physiques installés sur les nœuds.
Les données du wattmètre sont collectées et exposées par `Kwollect`. Kwollect est le service de monitoring disponible dans Grid'5000. Il collecte et met à disposition les métriques environnementales et de performance des nœuds. Vous pouvez trouver plus de détails sur le monitoring de Grid'5000 avec Kollect ici.
Vous pouvez visualiser les métriques collectées par Kwollect de deux manières principales : via les tableaux de bord Grafana et en consultant directement l'API Kwollect. Dans ce TP, vous allez essayer les deux.

Grafana

Grid'5000 met a disposition les tableaux de bord Grafana afin de visualiser les métriques disponibles dans Kwollect pour l'ensemble des noeuds. Vous pouvez consulter les tableau de bord du site Lyon ici. Pour visualiser les données des wattmètres physiques vous devez sélectionner le noeud dans l'onglet device et la métrique `wattmetre_power_watt`.

3.8 Kwollect API

Kwollect met a disposition l'ensemble des métriques via un API web. Vous pouvez récupérer les données de consommation d'un noeud directement depuis l'API.

Question 12
Ouvrez ce lien avec votre navigateur web. En suivant ce lien, vous récupérez les données renvoyées par le wattmètre physique installé sur le nœud nova-1 à Lyon pour la période de 12h00 à 12h10 le 21 septembre 2022.
À quelle fréquence Kwollect renvoie-t-il la métrique `wattmetre_power_watt` ?

Afin de faciliter la récupération des données de consommation depuis l'API Kwollect et leur visualisation, nous avons préparé quelques scripts pour ce TP.

Question 13
Connectez-vous à la frontale de Lyon. Veuillez récupérer et extraire l'archive avec les scripts avec les commandes suivantes :

```
wget http://public.lyon.grid5000.fr/~vostapenco/tp-consumption-cr17/scripts_front.tar.gz
tar -xvf scripts_front.tar.gz
```

Question 14
Utilisez le script `get_consumption_kwollect.py` pour récupérer les données de consommation rapportées par Kwollect lors de l'exécution de votre premier stress test.

```
python3 get_consumption_kwollect.py --site lyon --host NOM-DU-NOEUD --start
[DATE-DU-DEBUT] --end [DATE-DE-FIN] --csv kwollect_stress.csv
```

Les dates doivent être fournies au format `2022-09-20T17:52:30`.
Ce script récupérera les données de consommation du nœud entre deux dates et les enregistrera dans un fichier CSV (kwollect_stress.csv).
Analysez le contenu du fichier kwollect_stress.csv.
Combien de lignes contient-il ?

Question 15
Utilisez le script plot_consumption_kwollect.py pour créer un graphique de consommation à partir du fichier CSV.

```
python3 plot_consumption_kwollect.py --csv kwollect_stress.csv --plot
kwollect_plot.jpg
```

Copiez et analysez le graphique obtenu.
Vous pouvez utiliser la commande `scp` pour copier les fichiers.

```
scp <login>@access.grid5000.fr:lyon/kwollect_plot.jpg .
```

4 Comprendre la consommation

Dans cette section, vous allez effectuer une séquence de tests de charge et de configurations pour comprendre la consommation d'un nœud de calcul.

4.1 Consommation en mode inutilisé

Question 16
Récupérez et visualisez la consommation d'un nœud de calcul précédemment réservé en mode inutilisé.
Utilisez les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`.
Pour vous assurer que le nœud est en mode inutilisé, vous pouvez utiliser la commande `htop`.
Quelle est la consommation moyenne du noeud en mode inutilisé ?

4.2 Variation du nombre des coeurs utilisés

Question 17
Combien de cœurs sont disponibles sur le nœud ?
Pour répondre à cette question, utilisez la commande `lscpu` sur le nœud réservé.
Utilisez ensuite la description des nœuds Grid'5000 disponible ici. Trouvez le nombre de cœurs disponibles à partir du modèle de CPU.
NB : Les noeuds du cluster ***nova*** *ont 2 CPU physiques.*
Y a-t-il une différence entre le nombre de cœurs de données par commande `lscpu` et le nombre de cœurs réellement disponibles sur les CPU ?

Lors du premier test de stress, vous avez stressé le CPU avec 32 workers. Un worker d'outil `stress` place une charge importante sur un cœur de processeur.
Dans cette section, vous exécuterez l'outil `stress` avec le nombre différent de workers allant de 1 à 32 avec le pas de puissances de deux. Vous ferez une pause d'environ dix secondes entre les exécutions.

`Pour réaliser cette expérience, vous pouvez automatiser les exécutions à l'aide d'un script bash avec le contenu suivant :`

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Sleeping for 10 seconds before test."
sleep 10
for val in 1 2 4 8 16 32; do
    echo "Launching stress with $val workers for 10 seconds."
    stress -c $val -t 10
    echo "Sleeping for 10 seconds."
    sleep 10
done
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 18
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par example stress_cpu_num.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Pour rendre le fichier exécutable, utilisez la commande ***`chmod +x stress_cpu_num.sh`***.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 19
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`. Utilisez la commande `scp` vue précédemment pour copier le graphe de consommation de la frontale de Lyon.
Avez-vous observé quelque chose d'étrange ?
Faites attention aux dernières exécutions de stress.
Quelle est la consommation moyenne avec chaque nombre de workers ?
Donnez une valeur approximative. Quelle est la puissance par worker ? Que ne voit on pas sur ce graphe ?

4.3 Variation de la fréquence

Dans cette section, vous allez modifier la fréquence du processeur et voir comment cela affecte la consommation énergétique du nœud de calcul. Pour cela, vous utiliserez les scripts que nous avons préparés.

Question 20
Veuillez récupérer et extraire l'archive avec les scripts avec les commandes suivantes sur le nœud réservé :
`wget http://public.lyon.grid5000.fr/~vostapenco/tp-consumption-cr17/scripts_node.tar.gz`
`tar -xvf scripts_node.tar.gz`

Question 21
Visualisez les fréquences disponibles avec la commande `cpupower frequency-info`.
Quelle est la fréquence minimale supportée par le processeur ?
Quelle est la fréquence maximale supportée par le processeur ?

Pour changer la fréquence du CPU vous allez utiliser le script `change_cpu_freq.sh <CPU_FREQ_VALUE>` où `CPU_FREQ_VALUE` est la fréquence de CPU souhaitée (par exemple 1.6Ghz).

Question 22
`Avant de réaliser l'expérience`, si vous avez déjà utilisé le script de changement de fréquence, réinitialisez la configuration du gestionnaire de fréquences CPU avec le script `restore_cpu_governor.sh`.

`Pour réaliser l'expérience, vous procéderez comme suit :`
Notez le temps du début, attendez 10 secondes, changez la fréquence du CPU à la fréquence minimale, attendez 10 secondes, exécutez un stress pendant 10 secondes avec 16 workers, attendez 10 secondes, notez le temps de fin, réinitialisez la configuration du gestionnaire de fréquences CPU avec le script `restore_cpu_governor.sh`
`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant` :

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Sleeping for 10 seconds before test."
sleep 10
./change_cpu_freq.sh <CPU_FREQ_VALUE>
sleep 10
stress -c 16 -t 10
sleep 10
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Restore CPU Governor after experience."
./restore_cpu_governor.sh
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Dans ce script vous devez mettre la fréquence du CPU minimale au lieu de `<CPU_FREQ_VALUE>`.

Question 23
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par exemple cpu_freq_exp.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.
Regardez ce que vous retourne la commande `cpupower frequency-info`

Question 24
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`. Utilisez la commande `scp` vue précédemment pour copier le graphe de consommation de la frontale de Lyon.
Comment évolue la consommation lors du changement de fréquence du CPU au repos et lors d'un stress ?
NB : La modification manuelle de la fréquence nécessite la désactivation des optimisations du processeur qui sont appliquées lorsqu'il est en mode inutilisé.

Question 25
`Rejouez la même expérience mais cette fois à la fréquence maximale du CPU.`

Question 26
Récupérez et visualisez la consommation du nœud.
Qu'observez vous ?

5 Mesurer avec des Wattmètres logiciels

Dans les sections précédentes, vous avez utilisé les données de consommation d'énergie des nœuds de calcul fournies par un wattmètre physique. Ces wattmètres physiques sont installés entre la prise électrique et l'alimentation de chaque nœud de calcul.

C'est le moyen le plus précis pour mesurer la consommation énergétique d'un serveur. Cependant, ils sont difficiles à déployer à grande échelle et peuvent être coûteux. De plus, un wattmètre physique est capable de mesurer uniquement la consommation globale du serveur, sans donner le détail de la consommation des différents composants ou services lancés sur ce serveur.

En parallèle, il existe des interfaces internes comme Intel RAPL qui fournissent des informations sur la consommation énergétique des différents composants d'une machine (`CPU, RAM et GPU intégrés`) et sont disponibles sur la plupart des nœuds modernes. La récupération des données de consommation fournies par ces interfaces peut être difficile, nous avons donc besoin d'outils pour faciliter cette tâche.

Dans cette section, nous allons utiliser l'outil `Scaphandre`. `Scaphandre` est un agent de suivi dédié aux métriques de consommation énergétique. Il s'agit d'un outil open-source développé par Benoit Petit de l'organisation Hubblo. `Scaphandre` est un agent capable de rapporter la consommation d'énergie avec une granularité au niveau du processus. Afin de collecter des mesures de consommation d'énergie, il utilise le module du noyau Linux *powercap*, qui récupère les données de l'interface Intel RAPL.

`Scaphandre` est déjà installé et configuré dans l'image déployée sur le nœud au début du TP.

Dans cette section, vous allez exécuter un test de stress CPU en parallèle avec Scaphandre. Ensuite, vous allez récupérer et comparer les données de consommation rapportées par Kwollect et Scaphandre.

`Pour réaliser cette expérience, vous procéderez comme suit :`
Notez le temps du début, attendez 10 secondes, lancez en arrière plan Scaphandre pendant 30 secondes avec la fréquence de reporting de 1 seconde et le nombre de processus à surveiller égal à zéro, attendez 10 secondes, exécutez le stress avec 16 workers CPU pendant 10 secondes, attendez 10 secondes, notez le temps de fin.

`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant :`

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Sleeping for 10 seconds before test."
sleep 10
scaphandre json -t 30 -s 1 -m 0 -f scaphandre_stress.json &
sleep 10
stress -c 16 -t 10
```

```
sleep 10
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 27
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par example stress_cpu_scaphandre.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 28
Depuis la frontale, copiez avec la commande `scp` le fichier `scaphandre_stress.json` sur la frontale.
scp root@<NOM_DU_NOEUD> :./scaphandre_stress.json .
Récupérez la consommation du nœud rapportée par Kwollect avec le script `get_consumption_kwollect.py`.
Affichez sur le même graphique la consommation donnée par Scaphandre et Kwollect.
Vous pouvez le faire avec le script :
`python3 plot_consumption_kwollect_with_scaphandre.py --csv kwollect_stress.csv --json scaphandre_stress.json --plot kwollect_scaphandre_stress.jpg`
Utilisez la commande `scp` vue précédemment pour copier le graphe de consommation de la frontale de Lyon.
Analysez les valeurs de consommation données par Scaphandre et Kwollect. Expliquez les résultats.

6 Consommation énergétique du code

Dans cette section, vous allez exécuter et évaluer les différentes implémentions des algorithmes de trie.

6.1 Bubble sort

Le bubble sort ou tri à bulles est un algorithme de tri basique. L'algorithme compare plusieurs fois les éléments consécutifs d'un tableau et les échange lorsqu'ils sont mal triés. Cet algorithme n'est pas très efficace et a une complexité moyenne de $O(n^2)$.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define MAX 300

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(long));
        tableau[i] = rand()*i;
        printf("--> Allocation tableau case: %d \n" , i) ;
    }
}

int main ()
{
    int i;

    creer_tableau();

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

    sleep(3);

    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 29
Enregistrez le code source de l'application de tri à bulles dans un fichier avec l'extension `.c` (par exemple `bubble_sort.c`).
Compilez et exécutez l'application. Pour compiler, utilisez la commande `gcc bubble_sort.c -o bubble_sort`.
Observez les opérations effectuées par cette application.

Question 30
Modifiez le nombre MAX d'éléments pour que l'application dure une trentaine de secondes. Recompilez et relancez l'application.

Pour analyser la consommation lors de l'exécution de l'algorithme de tri à bulles, vous pouvez utiliser le script *bash* avec le contenu suivant :

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
sleep 3
./bubble_sort
sleep 3
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 31
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par exemple bubble_sort_exp.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 32
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`.
Observez le profil énergétique de cette application.

Question 33
Calculez l'énergie consommée par le nœud pendant l'exécution de l'application :
`python3 calculate_energy_kwollect.py --csv kwollect_consumption.csv`
Quelle quantité d'énergie le nœud a-t-il consommé pendant l'exécution de l'application ?

6.2 Quick sort

Le quick sort ou le tri rapide est un algorithme de tri basé sur la méthode consistant à placer un élément du tableau à une place fixe et échanger tous les éléments de sorte que tous ceux en dessous de l'élément fixe soient à sa gauche et que tous ceux au-dessus de l'élément soient sur sa droite.
Cet algorithme est plus efficace que le tri à bulles et a une complexité moyenne de $O(n \log n)$.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define MAX 300

int * tableau = NULL;

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);
}

void creer_tableau()
{
    int i;

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

int main ()
{
    int i;

    creer_tableau();

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

    sleep(3);

    printf("Debut du tri \n");
    quick_sort(tableau , MAX) ;
    printf("Fin du tri \n");

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

    free(tableau);
    return 0;
}
```

Question 34
Enregistrez le code source de l'application de tri à bulles dans un fichier avec l'extension `.c` (par exemple `quick_sort.c`).
Compilez et exécutez l'application. Pour compiler , utilisez la commande `gcc quick_sort.c -o quick_sort`.
Observez les opérations effectuées par cette application.

Question 35
Mettez le même nombre MAX d'éléments que pour le tri à bulles.
Recompilez et relancez l'application.
Comparez le temps d'exécution avec celui du bubble sort.

Question 36
Modifiez le nombre MAX d'éléments pour que l'application dure une trentaine de secondes.
Recompilez et relancez l'application.

Pour analyser la consommation lors de l'exécution de l'algorithme quick sort, vous pouvez utiliser le script *bash* avec le contenu suivant :

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
sleep 3
./quick_sort
sleep 3
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 37
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par exemple quick_sort_exp.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 38
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`.
Observez le profil énergétique de cette application.

Question 39
Calculez l'énergie consommée par le nœud pendant l'exécution de l'application :
`python3 calculate_energy_kwollect.py --csv kwollect_consumption.csv`
Quelle quantité d'énergie le nœud a-t-il consommé pendant l'exécution de l'application ?

6.3 Green sort

Dans cette section, nous allons essayer de rendre plus efficaces les algorithmes présentés précédemment en supprimant les affichages et en optimisant la fonction `creer_tableau()`.

Question 40
Supprimez tous les instruction d'affichage (`printf`) des implémentations précédentes, recompilez et réexécutez les expériences.
Quel est l'impact des affichages sur le profil et la consommation énergétique du bubble sort ?
Quel est l'impact des affichages sur le profil et la consommation énergétique du quick sort ?

Question 41
Ajustez le nombre MAX d'éléments pour que le quick sort dure une trentaine de secondes.
Recompilez et relancez l'expérience.
Observez le profil énergétique de cette version du quick sort.
Quelle quantité d'énergie le nœud a-t-il consommé lors de l'exécution de cette version de l'application ?

`Exemple d'implémentation plus efficace de la fonction creer_tableau():`

```
void creer_tableau()
{
    int i;

    srand(time(NULL));

    tableau = (int *) malloc(MAX * sizeof(long));
    for (i = 0; i < MAX; i++){
        tableau[i] = rand()*i;
    }
}
```

Question 42
Remplacez la fonction `creer_tableau()` dans l'implémentation du quick sort.
Quel est l'impact sur le profil et la consommation énergétique ?

6.4 Variation de la fréquence du CPU

Dans le premier TP, vous avez vu comment la fréquence du CPU impacte la consommation énergétique du nœud de calcul. Dans ce TP, vous allez voir comment le changement de fréquence affecte le temps d'exécution et donc la consommation énergétique totale de l'exécution.

Gestion de la fréquence par défaut

Dans cette section, vous verrez comment est gérée la fréquence des cœurs du processeur par le gestionnaire de fréquence par défaut.

Question 43
Surveillez la fréquence de chaque coeur du processeur avec la commande `watch -n 1 cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_cur_freq`.
Connectez-vous avec un autre terminal au nœud réservé et exécutez la dernière version du quick sort (sans l'affichage et qui a duré environ 30 secondes dans la section précédente).
Exécutez le quick sort via le script `bash` *créé précédemment.*
Surveillez la fréquence de chaque cœur de processeur pendant l'exécution.
Que pouvez vous constater ?
Mesurez et notez le temps d'exécution et l'énergie consommée par l'application.

Exécution à la fréquence maximale

Question 44
Veuillez récupérer et extraire l'archive avec les scripts avec les commandes suivantes sur le nœud réservé :
`wget http://public.lyon.grid5000.fr/~vostapenco/tp-consumption-cr17/scripts_node.tar.gz`
`tar -xvf scripts_node.tar.gz`

Pour changer la fréquence du CPU vous allez utiliser le script `change_cpu_freq.sh <CPU_FREQ_VALUE>` où `CPU_FREQ_VALUE` est la fréquence de CPU souhaitée (par exemple 1.6Ghz).

Question 45
Visualisez les fréquences disponibles avec la commande `cpupower frequency-info`.
Quelle est la fréquence minimale supportée par le processeur ?
Quelle est la fréquence maximale supportée par le processeur ?

Question 46
Changez la fréquence à la fréquence maximale et visualisez la fréquence des cœurs des CPU avec la commande `cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_cur_freq`.
Expliquez le résultat.

Question 47
Réinitialisez la configuration du gestionnaire de fréquences avec la commande `restore_cpu_gouvernor.sh`.

Vous aller exécuter la dernière version du quick sort à la fréquence maximale du processeur tout en surveillant le temps d'exécution et la consommation d'énergie du nœud.
`Pour réaliser cette expérience, vous procéderez comme suit :`
Changez la fréquence du CPU à la fréquence maximale, notez le temps du début, exécutez la dernière version du quick sort (sans l'affichage et qui a duré environ 30 secondes dans la section précédente), notez le temps de fin, réinitialisez la configuration du gestionnaire de fréquences CPU avec le script `restore_cpu_governor.sh`

`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant:`

```
#!/bin/bash
./change_cpu_freq.sh <CPU_FREQ_VALUE>
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
sleep 3
./quick_sort
sleep 3
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Restore CPU Governor after experience."
./restore_cpu_gouvernor.sh
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Dans ce script vous devez mettre la fréquence du CPU maximale au lieu de `<CPU_FREQ_VALUE>`.

Question 48
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par exemple quick_sort_freq.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.
Calculez et notez la durée d'exécution de l'expérience.

Question 49
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`.
Observez le profil énergétique de cette application.

Question 50
Calculez l'énergie consommée par le nœud pendant l'exécution de l'application :
`python3 calculate_energy_kwollect.py --csv kwollect_consumption.csv`
Quelle quantité d'énergie le nœud a-t-il consommé pendant l'exécution de l'application ?

Diminution de la fréquence

Dans cette section, vous allez progressivement diminuer la fréquence afin de comprendre comment cela affecte le temps d'exécution et donc la consommation énergétique totale de l'exécution.

Question 51
Rejouez la même expérience que précédemment avec les fréquences 2.1Ghz, 2.0Ghz, 1.8Ghz et 1.2Ghz.

Question 52
Après avoir effectué toutes les expériences, que pouvez-vous conclure ?

6.5 BONUS : Exécutions parallèles

Dans cette section, vous exécuterez plusieurs instances de quick sort en parallèle tout en surveillant la consommation des nœuds et la fréquence des cœurs de CPU.

`Pour réaliser cette expérience, vous procéderez comme suit :`
Notez le temps du début, attendez 10 secondes, exécutez plusieurs instances de quick sort avec le nombre d'instances allant de 1 à 32 avec le pas de puissances de deux, faites une pause de 10 secondes entre les exécutions, notez le temps de fin.

`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant:`

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
sleep 10
for val in 1 2 4 8 16 32; do
    ./launch_parallel.sh ./quick_sort $val
    sleep 10
done
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 53
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par exemple quick_sort_parallel.sh) et rendez le fichier exécutable avec la commande `chmod`.
Connectez-vous avec un autre terminal au nœud réservé et surveillez la fréquence de chaque coeur du processeur avec la commande
`watch -n 1 cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_cur_freq`
Lancez le script `bash`.
Comment change la fréquence des cœurs de CPU lors de l'exécution d'instances de quick sort ?

Question 54
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`.
Observez le profil énergétique de cette expérience.
Comparez le profil énergétique avec celui obtenu lors de la réalisation des stress tests dans le premier TP.

7 De la consommation électrique.. mais pas seulement...

Dans cette section, vous allez calculer l'impact du TP en termes d'énergie consommée et d'équivalent de gCo2eq émis.

7.1 Calculer l'énergie consommée

Pour calculer l'énergie consommée par le nœud de calcul pendant le TP, nous aurons besoin de l'heure de début du TP et de l'heure actuelle.

Nous avons demandé au tout début de TP de récupérer et de mémoriser l'heure. Nous aurons besoin de cette heure dans cette section.

Question 55
Obtenez l'heure actuelle.
Pour ce faire, utilisez la commande suivante `date +%Y-%m-%dT%H:%M:%S`
Récupérez la consommation du nœud depuis le début du TP rapportée par Kwollect avec le script `get_consumption_kwollect.py`.
Calculez l'énergie consommée par le nœud depuis le début du TP avec le script :
`python3 calculate_energy_kwollect.py --csv kwollect_tp_consumption.csv`
Quelle quantité d'énergie le nœud a-t-il consommé pendant le TP ?

7.2 Calculer l'équivalent en CO2

Pour calculer l'équivalent en gCo2eq émis de l'énergie consommée, il faut connaître les émissions de CO2 par kWh d'électricité produite en France. Vous pouvez trouver ces informations sur le site de rte-france.

Question 56
Calculez l'équivalent en Co2 à partir des données RTE.
Quelle quantité de gCo2eq a été émise pour produire l'électricité nécessaire à l'alimentation de votre nœud pendant ce TP ? Extrapolez à l'échelle d'une journée, d'un mois, d'une année...

7.3 Calculer le prix de l'énergie

Pour calculer le prix de l'énergie consommée pendant ce TP, vous devez trouver un prix moyen de l'énergie en France sur internet et effectuer le calcul.

Question 57
Calculez le prix de l'énergie consommée à partir du prix trouvé.
Quelle est le prix de l'énergie consommée pendant ce TP ? Extrapolez à l'échelle d'une journée, d'un mois, d'une année...

8 Bonus.... si on a le temps

8.1 Impact du Turbo Boost

`Turbo Boost` est une technologie qui permet d'augmenter la fréquence du processeur de manière temporaire tout en restant dans des limites de température et de puissance sûres. `Turbo Boost` est activé par défaut sur les systèmes d'exploitation modernes.

Dans cette section, vous allez exécuter le stress avec 16 workers avec et sans Turbo Boost. Puis, vous allez étudier l'impact de cette technologie sur la consommation énergétique du noeud de calcul.

`Pour réaliser cette expérience, vous procéderez comme suit :`
Notez le temps du début, attendez 10 secondes, exécutez le stress avec 16 workers pendant 10 secondes, attendez 10 secondes, desactivez Turbo Boost avec le script `disable_turbo_boost.sh`, attendez 10 secondes, exécutez le stress avec 16 workers pendant 10 secondes, attendez 10 secondes, notez le temps de fin, réactivez le Turbo Boost avec le script `enable_turbo_boost.sh`.

`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant` :

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Sleeping for 10 seconds before test."
sleep 10
stress -c 16 -t 10
sleep 10
./disable_turbo_boost.sh
sleep 10
stress -c 16 -t 10
sleep 10
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Enabling Turbo Boost after experience."
./enable_turbo_boost.sh
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 58
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par example turbo_boost.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 59
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`. Utilisez la commande `scp` vue précédemment pour copier le graphe de consommation de la frontale de Lyon.
Comment l'utilisation de la technologie Turbo Boost impacte la consommation énergétique du noeud ?

8.2 Impact de la mémoire vive

Dans cette section, vous allez charger la mémoire vive (RAM) du nœud de calcul et analyser son impact sur la consommation d'un nœud de calcul.

L'outil `stress` vous permet de lancer des workers qui simulent une utilisation élevée de la RAM (option `-m`) en effectuant des opérations d'allocation et de libération de mémoire.

`Pour réaliser cette expérience, vous procéderez comme suit :`
Notez le temps du début, attendez 10 secondes, exécutez le stress avec 2 workers RAM pendant 10 secondes, attendez 10 secondes, notez le temps de fin.

`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant :`

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Sleeping for 10 seconds before test."
sleep 10
stress -m 2 -t 10
sleep 10
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 60
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par example stress_memory.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 61
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`. Utilisez la commande `scp` vue précédemment pour copier le graphe de consommation de la frontale de Lyon.
Comment l'utilisation intensive de la mémoire impacte-t-elle la consommation d'un nœud de calcul ?

8.3 Impact du stockage

Dans cette section, vous allez charger le stockage du nœud de calcul et analyser son impact sur la consommation d'un nœud de calcul.

L'outil `stress` vous permet de lancer des workers qui simulent une utilisation élevée du stockage (option `-d`) en effectuant des opérations d'écriture et de suppression d'un fichier volumineux (1 Go par défaut).

`Pour réaliser cette expérience, vous procéderez comme suit :`
Notez le temps du début, attendez 10 secondes, exécutez le stress avec 2 workers disk pendant 10 secondes, attendez 10 secondes, notez le temps de fin.

`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant :`

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Sleeping for 10 seconds before test."
sleep 10
stress -d 2 -t 10
sleep 10
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 62
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par example stress_storage.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 63
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`. Utilisez la commande `scp` vue précédemment pour copier le graphe de consommation de la frontale de Lyon.
Comment l'utilisation intensive du stockage impacte-t-elle la consommation d'un nœud de calcul ?

8.4 Impact du réseau

Dans cette section, vous allez charger l'interface réseau du nœud de calcul et analyser son impact sur la consommation d'un nœud de calcul.

Pour cela vous allez utiliser l'outil `iperf`. L'outil `iperf` permet de surcharger une interface réseau en envoyant un grand nombre de paquets. Cet outil fonctionne en mode client-serveur, donc pour effectuer le test, vous avez besoin d'une instance `iperf` agissant en tant que serveur. Demandez à votre enseignant l'adresse du serveur `iperf`.

Avant de lancer l'expérience, il faut savoir quelle bande passante est supportée par la carte réseau du nœud. La bande passante supportée par la carte réseau nous permet de choisir la quantité de trafic qui sera envoyée par le client `iperf` au serveur. Vous pouvez trouver la bande passante de la carte réseau sur la page de description des nœuds Grid'5000 disponible ici.

Question 64
Quelle est la bande passante de la carte réseau des nœuds nova ?

`Pour réaliser cette expérience, vous procéderez comme suit :`
Notez le temps du début, attendez 10 secondes, lancez le client `iperf` qui envoie des paquets UDP avec le débit égal à la bande passante de la carte réseau du nœud pendant 10 secondes, attendez 10 secondes, notez le temps de fin.

`Vous pouvez automatiser cette expérience à l'aide d'un script bash avec le contenu suivant :`

```
#!/bin/bash
START_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "Sleeping for 10 seconds before test."
sleep 10
iperf -c <NOM_DU_HOST_SERVEUR> -u -t 10 -b <DEBIT_D_ENVOIE>
sleep 10
END_DATE=$(date +%Y-%m-%dT%H:%M:%S)
echo "EXPERIMENT START DATE: $START_DATE"
echo "EXPERIMENT END DATE: $END_DATE"
```

Question 65
Enregistrez ce contenu dans un fichier texte avec l'extension `.sh` (par example stress_network.sh), rendez le fichier exécutable avec la commande `chmod` et lancez le.
Notez les dates de début (`EXPERIMENT START DATE`) et de fin (`EXPERIMENT END DATE`) de l'expérience. Ils seront utilisées lors de l'exécution du script `get_consumption_kwollect.py`.

Question 66
Récupérez et visualisez la consommation du nœud avec les scripts `get_consumption_kwollect.py` et `plot_consumption_kwollect.py`. Utilisez la commande `scp` vue précédemment pour copier le graphe de consommation de la frontale de Lyon.
Comment l'utilisation intensive de l'interface réseau impacte-t-elle la consommation d'un nœud de calcul ?

8.5 Analyse de la consommation

Dans les sections précédentes, vous avez analysé la consommation d'énergie d'un nœud de calcul.

Dans cette section, donnez vos conclusions sur l'impact des différents composants et de leurs configurations sur la consommation globale d'un nœud de calcul.