

TP 4 - Algorithmie avancée - 2022

Branch and bound

JAY Mathilde - Prof. TRYSTRAM Denis

Pour les deux prochaines séances, nous travaillons sur la gestion efficace des ressources de calcul. Nous allons compléter l'analyse du temps d'exécution par une étude de la consommation énergétique.

Sujet : Ordonnement de tâches dans un centre de calcul

Un centre de calcul typique (bien entendu, simplifié pour les besoins d'une étude en temps limité) est composé de m machines multi-coeurs que l'on supposera identiques avec chacune q coeurs et d unité de mémoire, reliées par un réseau d'interconnexion local et rapide. On négligera les temps de communications.

Les n tâches à exécuter sont caractérisées par le nombre de coeurs et la quantité de mémoire dont elles ont besoin. On suppose que l'on connaît toutes les tâches et on ne prend pas en compte la date d'arrivée.

On souhaite optimiser dans un premier temps de temps total d'exécution des tâches. On vous demande d'implémenter une approche gloutonne et une approche branch and bound.

On fera ensuite une analyse multi-critère du temps d'exécution et la consommation énergétique des algorithmes que vous avez implémenté. On complétera avec une étude de la consommation énergétique du centre de calcul.

Analyse de la consommation énergétique

Les deux prochaines sections décrivent les méthodologies à suivre pour le calcul de la consommation énergétique

Consommation de votre algorithme

Il existe de nombreux outils de mesure de consommation énergétique. Je vous propose d'utiliser la librairie Python Code Carbon¹ qui se base sur les interfaces internes RAPL et NVML des processeurs Intel et Nvidia. Votre algorithme doit tourner au moins quelques secondes pour la librairie puisse calculer une énergie. La librairie enregistre les résultats dans un fichier CSV.

Si vous n'utilisez pas Python, vous pouvez lancer votre script avec `os.popen(script)`. Sinon, vous pouvez utiliser la librairie directement dans votre code.

Un exemple d'utilisation se situe dans les annexes.

Consommation du centre de calcul

L'énergie consommée par un centre de donnée ne correspond pas exactement à son utilisation². Une partie correspond à la communication entre les serveurs et à la structure qui les refroidit. Pour estimer l'énergie consommée total, on utilise le ratio PUE (Power Usage Effectiveness)³, tel que :

$$P_{total} = PUE * P_{serveurs}$$

Avec PUE = 1.59 , valeur moyenne mondiale d'après statista⁴

¹ <https://github.com/mlco2/codecarbon>

² https://datacenters.lbl.gov/sites/default/files/Masanet_et_al_Science_2020.full_.pdf

³ <https://www.grid5000.fr/w/Lyon:Hardware#nova>

⁴ <https://www.statista.com/statistics/1229367/data-center-average-annual-pue-worldwide/>

De plus, des études ont montré qu'un serveur consommait de l'énergie même lorsqu'il n'était pas utilisé. Pour ce TP, nous utiliserons les valeurs ci-dessous, obtenues à partir de tests sur des serveurs Nova sur la plateforme Grid5000.

$$P_{inactivite} = 100 \text{ Watt}$$

$$P_{max} = 170 \text{ Watt}$$

$$P_{coeur} = P_{inactivite} * D_{inactivite} + P_{max} * D_{utilisation} \text{ (Wh)}$$

Avec $D_{inactivite}$ et $D_{utilisation}$ les durées en heure durant lesquelles le coeur est inactif ou actif.

$$P_{serveur} = \sum_{coeurs} P_{coeur}$$

$$P_{serveurs} = \sum_{serveurs} P_{serveur}$$

On utilise donc un modèle simplifié puisqu'on suppose que pour n'importe quelle tâche, le coeur tourne à sa puissance maximale, ce qui n'est pas le cas en général.

Dans la pratique, il existe des états qui améliorent la puissance consommée par le serveur en inactivité (avec ces serveurs on peut descendre à 70W). On peut aussi modifier la fréquence des processeurs pour réduire la consommation.

Objectifs

- Déterminer et implémenter l'algorithme de recherche exhaustive
- Ajouter une implémentation Branch and Bound
- Implémenter une modélisation de la consommation énergétique de l'exécution des tâches obtenues
- Mesure de la consommation énergétique de vos algorithmes
- Analyser comparativement le temps d'exécution et la consommation énergétique de vos algorithmes (un graphe)
- Analyser comparativement la consommation énergétique de vos algorithmes et la consommation de l'exécution des tâches obtenues (un graphe - frontière de Pareto)
- Comment optimiser la consommation énergétique du centre de calcul ? Proposer quelques solutions possibles.

Annexe

Exemple d'utilisation de Code Carbon

```
% pip install codecarbon
```

```
from codecarbon import EmissionsTracker
```

```
codecarbon_tracker = EmissionsTracker(  
    project_name=name_algorithm,  
    output_dir=result_directory,  
    save_to_file=result_directory+"consommation.csv"  
)
```

```
codecarbon_tracker.start()  
output = os.popen(script)  
codecarbon_tracker.stop()
```