

# Ordonnancement de Tâches avec Précédences sur Plateforme Edge-Cloud Alimentée par des Sources d’Energie Verte

Aris Koutsiamanis<sup>1</sup>      Jean-Marc Menaud<sup>1</sup>      Clément Mommessin<sup>1</sup>

IMT Atlantique, Nantes Université, École Centrale Nantes, CNRS, INRIA, LS2N, UMR 6004,  
F-44000 Nantes, France

{remous-aris.koutsiamanis, jean-marc.menaud, clement.mommessin}@imt-atlantique.fr

**Mots-clés** : *Ordonnancement, Edge, Cloud, Energie Verte*

## 1 Introduction

On trouve dans la littérature de plus en plus de travaux s’efforçant de réduire l’empreinte carbone des systèmes informatiques et de les rendre plus *verts*. Il y a notamment des approches pour intégrer des énergies renouvelables dans le mix énergétique des datacentres, ce qui pose un bon nombre de difficultés dans la manière de concevoir et de gérer ces systèmes [5]. Certains prennent le parti de rendre les plateformes 100% vertes mais se confrontent à l’intermittence de la disponibilité de l’énergie [4]. D’autres, comme Grange et al. [3], préfèrent utiliser au mieux l’énergie renouvelable quand elle est disponible tout en gardant l’accès à l’électricité *conventionnelle* de la grille, et donc non-renouvelable et plus carbonée, si nécessaire. Nous choisissons ce second parti.

Pour ce travail, nous nous intéressons à un problème d’ordonnancement *on-line* de jobs, composés de tâches parallèles et liées par des contraintes de précédences, sur une plateforme distribuée avec certains sites de calcul qui peuvent tirer profit d’une source locale d’énergie *verte* (panneaux solaires, éoliennes, etc.) pour alimenter les serveurs. Contrairement à d’autres approches de la littérature, nous nous intéressons à une fenêtre de temps suffisamment courte (de l’ordre de quelques minutes) pour supposer une production constante venant des sources d’énergie verte. L’objectif recherché est la minimisation d’énergie conventionnelle (non-renouvelable) nécessaire à l’exécution des jobs.

## 2 Formalisation du Problème

La plateforme est composée de plusieurs sites de calcul interconnectés par un réseau, et plusieurs sites de production d’énergie renouvelable. Chaque site de calcul est caractérisé par un nombre de processeur et une quantité de mémoire disponibles, la vitesse de calcul des processeurs, la puissance statique utilisée quand au moins un processeur est actif (qui exécute une tâche), et la puissance dynamique utilisée pour chaque processeur actif.

Chaque site de calcul est connecté au réseau électrique conventionnel et à *au plus un* site de production d’énergie renouvelable. Chaque site de production d’énergie renouvelable est caractérisé par la puissance produite (supposée constante) durant la fenêtre de temps.

Les jobs soumis sur la plateforme sont caractérisés par leur temps de soumission, leur temps limite de completion et le graphe acyclique dirigé de tâches associé. Chaque tâche est caractérisée par le nombre de processeurs, la quantité de mémoire et la quantité de travail requises pour son exécution. Les tâches sont liées par des contraintes de précédences : une tâche doit commencer son exécution immédiatement après la completion de tous ses prédecesseurs. De plus, un temps de communication est nécessaire quand une tâche ne s’exécute pas sur le même

site de calcul qu'une tâche prédécesseur.

En plus de contraintes d'ordonnement 'classiques', nous introduisons dans notre modèle la notion de filtrage par *étiquettes*, inspiré de Kubernetes. Ces étiquettes sont utilisées pour filtrer pour chaque tâche les sites disponibles pour le calcul, avec par exemple des contraintes spéciales de localité ou de ressources spécifiques. Par exemple, avec une plateforme composée d'un site de calcul avec l'étiquette 'Nantes' et un autre site avec l'étiquette 'Paris', un utilisateur Parisien pourrait forcer l'exécution locale de ses tâches en y ajoutant l'étiquette 'Paris'.

Une solution pour ce problème définit les temps de début et de fin d'exécution de chaque tâche de chaque job soumis sur la plateforme. Le placement des tâches doit satisfaire les contraintes de ressources (processeur et mémoire) ainsi que les contraintes de précédences. De plus, une solution fournit également, pour chaque site de calcul et chaque pas de temps, la proportion de la puissance consommée venant d'une source d'énergie renouvelable ou venant de la grille.

L'objectif du problème est de trouver un ordonnancement qui minimise l'énergie totale venant de la grille.

### 3 Pistes de Résolution

Pour résoudre ce problème, nous nous intéresserons à deux méthodes de résolution. La première méthode est basée sur la programmation par contrainte, avec une modélisation du problème dans *MiniZinc* [1] en vue d'une résolution optimale mais coûteuse en termes de temps de calcul.

La seconde méthode se base sur des algorithmes d'ordonnement plus *classiques* inspirés des travaux de E. Cadorel [2] pour construire une solution approchée mais avec une complexité temporelle limitée.

### Acknowledgments

Ces travaux sont financés par le projet BPI OTPaaS.

### Références

- [1] Minizinc constraint modeling language. <https://www.minizinc.org/>.
- [2] Emile Cadorel. *Energy-aware management of scientific workflows in the Cloud : a Cloud provider-centric vision*. PhD thesis, Ecole nationale supérieure Mines-Télécom Atlantique, 2020.
- [3] Léo Grange, Georges Da Costa, and Patricia Stolf. Green it scheduling for data center powered with renewable energy. *Future Generation Computer Systems*, 86 :99–120, 2018.
- [4] Ayham Kassab, Jean-Marc Nicod, Laurent Philippe, and Veronika Rehn-Sonigo. Green power aware approaches for scheduling independent tasks on a multi-core machine. *Sustainable Computing : Informatics and Systems*, 31 :100590, 2021.
- [5] Gustavo Rostirolla, Léo Grange, T Minh-Thuyen, Patricia Stolf, Jean-Marc Pierson, Georges Da Costa, Gwilherm Baudic, Marwa Haddad, Ayham Kassab, Jean-Marc Nicod, et al. A survey of challenges and solutions for the integration of renewable energy in datacenters. *Renewable and Sustainable Energy Reviews*, 155 :111787, 2022.