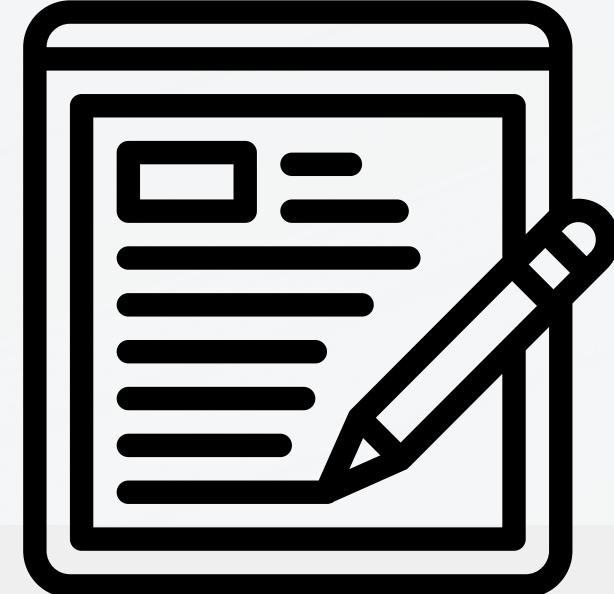


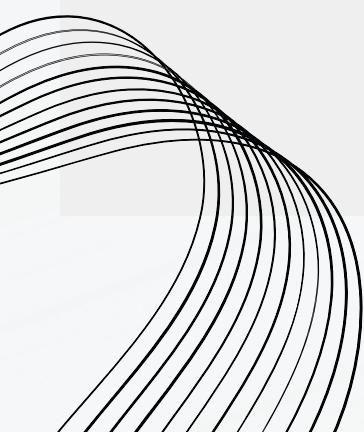
Techniques d'Ordonnancement dans les Systèmes Distribués et Particularités de UML-MARTE

Elaboré par:
Aya Grami
Fatma Bouhari



INTRODUCTION

Dans le contexte des systèmes distribués, l'ordonnancement adaptatif et la modélisation précise des processus et ressources sont cruciaux pour la performance et la fiabilité. L'article analysé met en lumière l'importance d'une architecture multiagent pour l'ordonnancement adaptatif, qui permet une gestion flexible et réactive face aux changements et aux interruptions. Cette présentation se penche sur les stratégies spécifiques d'ordonnancement mentionnées dans l'article et explore en détail les avantages de l'utilisation de UML-MARTE par rapport à UML standard pour la modélisation de ces systèmes complexes. En intégrant ces techniques et outils, nous visons à illustrer comment ils peuvent significativement améliorer l'ordonnancement dans les systèmes distribués.





TECHNIQUES D'ORDONNANCEMENT DES Systèmes Distribués

TECHNIQUES D'ORDONNANCEMENT DES SYSTÈMES DISTRIBUÉS

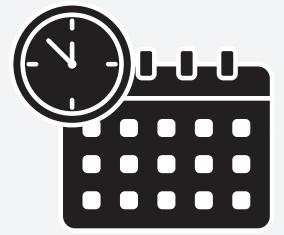
Heuristique de
Décomposition et
Migration (DMH)

Heuristique de
Dégradation (DH)

Architecture Multiagent
pour l'Ordonnancement
Adaptatif

Heuristique de
Suppression (RH)





ARCHITECTURE MULTIAGENT POUR L'ORDONNANCEMENT ADAPTATIF

L'architecture est conçue pour des systèmes distribués nécessitant une adaptation face à des changements dynamiques tels que l'activation de nouvelles tâches et les pannes matérielles ou logicielles.



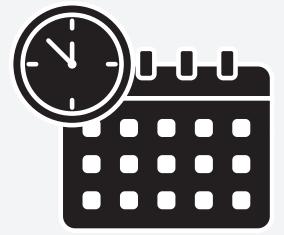
Agents locaux :

- Superviseur : Déetecte les événements imprévus et initie des scénarios de reconfiguration.
- Ordonnanceur : Analyse la faisabilité des tâches en fonction des contraintes temps-réel et énergétiques.
- Gestionnaire de batterie : Prévoit la disponibilité énergétique à court terme.
- Gestionnaire de reconfiguration : Implémente les stratégies de décomposition, dégradation ou suppression.

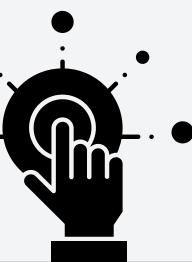


Agent Coordonnateur :

Un agent coordonnateur global est défini pour coordonner entre les sous-systèmes réseautés et gérer toutes les exigences de reconfiguration concurrentes.



ARCHITECTURE MULTIAGENT POUR L'ORDONNANCEMENT ADAPTATIF



Interactions et Contrôle :

- Les agents au sein d'un sous-système interagissent pour contrôler les scénarios de reconfiguration et vérifier la faisabilité du système.
- Les interactions entre les sous-systèmes via l'agent coordonnateur permettent d'autoriser et d'appliquer des scénarios de reconfiguration globale.



Optimisation de Performance :

- Le système vise à optimiser la performance globale et garantir l'exactitude à travers le contrôle distribué, en utilisant l'architecture multiagent pour simuler le comportement réel du système physique.

DÉCOMPOSITION ET MIGRATION

- **Principe :** Les tâches qui surchargent un processeur sont décomposées en branches et redistribuées sur d'autres processeurs.
- **Avantages :**
 1. Réduction des surcharges locales.
 2. Amélioration de l'utilisation globale des ressources.
- **Exemple visuel :**

Un processeur (P1) est surchargé. Une tâche est migrée vers un processeur sous-utilisé (P2).

- **Etapes:**



SÉLECTION DES
BRANCHES À
MIGRER

Basée sur la criticité
et la densité
énergétique.



CHOIX DU
PROCESSEU
R CIBLE

Priorité aux processeurs
avec plus d'énergie et
moins de charge.

HEURISTIQUE DE DÉGRADATION

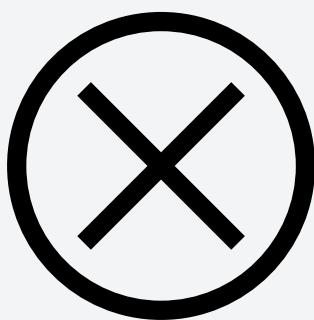
- **Principe** : Les tâches non critiques peuvent être exécutées dans un mode dégradé selon le modèle (m, k) -firm.
- m tâches parmi k doivent respecter leurs délais.
- **Exemple** : Si une tâche a un modèle $(3, 5)$ -firm, alors sur 5 exécutions consécutives, au moins 3 doivent être réussies.

- **Avantages** :
 - Préservation des délais critiques.
 - Réduction de l'énergie consommée pour les tâches non critiques.

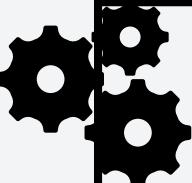


$$U_{m,k} = \sum_{i=1}^n \left(\frac{C_i}{T_i} \cdot \frac{m_i}{k_i} \right)$$

CALCUL DE LA FAISABILITÉ :



HEURISTIQUE DE SUPPRESSION



Mécanisme :

- Supprime les branches ou les tâches du graphe de tâches (DAG) qui présentent les densités les plus élevées.
- La densité est définie comme le ratio entre l'énergie consommée et le temps d'exécution, permettant de cibler les tâches les plus coûteuses en ressources.

Densité énergétique (γ): $\gamma = \frac{E_{ni}}{T_i}$



Critère :

- Priorité à la suppression des tâches non critiques avec une densité énergétique élevée.



Impact :

- Réduction immédiate de la surcharge.
- Meilleure utilisation des ressources pour les tâches critiques.



UML-MARTE

Qu'est-ce qu'UML-MARTE ?

- *Une extension d'UML pour modéliser les systèmes temps réel et embarqués.*
- *Ajoute des capacités pour capturer les contraintes temporelles et énergétiques.*

Applications :

- *Modélisation des architectures matérielles et logicielles.*
- *Simulation des délais et de la consommation énergétique.*
- *Analyse des chemins critiques dans les graphes de tâches.*

APPLICATION PRATIQUE DE UML-MARTE

Cas d'Utilisation : Modélisation d'un Système Temps Réel Distribué

- Utilisation de UML-MARTE pour représenter un système distribué composé de multiples sous-systèmes interconnectés.
- Modélisation des contraintes temporelles et des dépendances entre tâches grâce aux stéréotypes MARTE.
- Analyse de faisabilité pour l'ordonnancement adaptatif basé sur des contraintes énergétiques et des échéances strictes.

Résultats Observés :

- Réduction des conflits dans l'ordonnancement grâce à une meilleure gestion des ressources.
- Démonstration de la capacité de UML-MARTE à faciliter l'identification des goulets d'étranglement dans les processus.
- Validation des scénarios de reconfiguration dans des environnements simulés.

Différences UML vs UML-MARTE

Critères	UML	UML-MARTE
Modélisation temps réel	Basique	Avancée (WCET, délais)
Contraintes énergétiques	Absentes	Intégrées
Reconfiguration dynamique	Non pris en charge	Pris en charge
Simulations	Limitées	Compatible avec des outils avancés

AVANTAGES DE UML-MARTE POUR LES SYSTÈMES DISTRIBUÉS

01

02

03

04

SUPPORT DES
CONTRAINTE
S TEMPORELLES ET DES
RESSOURCES

AMÉLIORATION
DE LA
PRÉCISION DE
MODÉLISATION

FACILITATION
DE L'ANALYSE
ET DE LA
VÉRIFICATION

INTEROPÉRABILITÉ
AVEC DES OUTILS
D'ANALYSE DE
SYSTÈMES

CONCLUSION

En conclusion, les systèmes distribués nécessitent des techniques d'ordonnancement adaptatives et robustes pour gérer efficacement les défis dynamiques tels que les pannes et les contraintes de ressources. L'article étudié met en lumière des approches innovantes, notamment l'architecture multiagent et des heuristiques spécifiques comme la décomposition, la dégradation et la suppression, qui optimisent la gestion des tâches et garantissent la résilience du système. Par ailleurs, UML-MARTE se révèle être un outil puissant pour la modélisation des systèmes temps réel, grâce à ses extensions dédiées permettant une représentation précise des contraintes temporelles et des ressources. Ces contributions offrent des perspectives prometteuses pour améliorer la performance et la fiabilité des systèmes distribués, tout en ouvrant la voie à de futures recherches sur l'optimisation des outils et des techniques pour répondre aux besoins des environnements encore plus complexes comme l'IoT et les systèmes autonomes.

**MERCI POUR
VOTRE
ATTENTION**

