

Conception d'un algorithme d'ordonnancement temps réel pour processeurs à vitesse variable



Financement : Projet ANR [SHRIMP](#)

Laboratoire : [Laboratoire d'Informatique et d'Automatique pour les Systèmes \(LIAS\)](#), ENSMA, Chasseneuil-du-Poitou

Équipe : Temps réel

Encadrement : Antoine Bertout, Thomas Gaspard, Joël Goossens et Pascal Richard

Contact : antoine.bertout@univ-poitiers.fr

Mots-clés : Ordonnancement temps réel, simulation

Début de stage : Pas avant janvier 2025

Contexte

Ce stage s'inscrit dans la thématique de l'ordonnancement temps réel où l'objectif est d'assurer le respect des contraintes fonctionnelles et temporelles d'un système de traitement (tâches) s'exécutant sur un système embarqué et critique.

Le projet SHRIMP a pour objectif de développer un ordonnanceur (algorithme d'ordonnancement) temps réel en ligne, global, praticable et efficace pour des plateformes disposant de clusters de cœurs d'architectures et performances hétérogènes, telles que le [SoC RK3399](#) (clusters ARM Cortex-A53 et -A72). Une thèse est en cours sur le sujet et s'attache à développer un algorithme optimal de cette catégorie sur le plan théorique.

Les algorithmes optimaux ont la capacité de pouvoir trouver une séquence d'ordonnancement valide (respect des échéances, pas de parallélisme sauf si autorisé) des tâches sur les processeurs, si elle existe. En théorie ces algorithmes peuvent traiter des systèmes de tâches chargés à 100% comme illustré en Figure 1. En pratique, ces approches peuvent néanmoins avoir des performances très dégradées (nombreuses préemptions/migrations), voire même être difficilement implémentable sur un système d'exploitation. En conséquence, Il est parfois intéressant de sacrifier les performances théoriques (capacité de traiter des systèmes jusqu'à 66% d'utilisation maximum par exemple) pour obtenir une solution applicable en pratique. Ces algorithmes non-exacts sont donc des heuristiques.

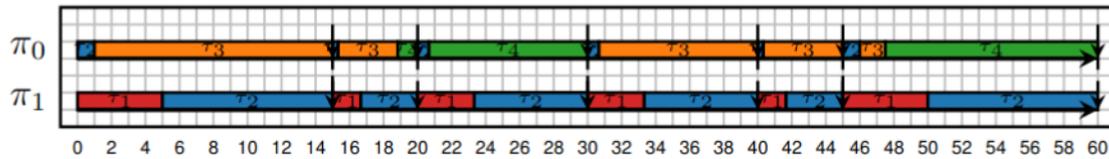


Figure 1: Ordonnancement faisable de quatre tâches (temps d'exécution pire-cas, période/échéance relative) $\tau_1(5, 15)$, $\tau_2(16, 20)$, $\tau_3(35, 30)$, $\tau_4(42, 60)$ sur plateforme uniforme à deux processeurs π_1 et π_2 de vitesse 2 et 1.

Sujet du stage

Les objectifs du stage sont les suivants :

1. Identifier comment un algorithme préemptif existant (non-optimal, une heuristique) pour des processeurs identiques, comme EDZL[1], peut être adapté à des processeurs à vitesses différentes (dits uniformes).
2. Implémenter cet algorithme ainsi qu'un algorithme optimal de la littérature dans un simulateur temps réel déjà existant, programmé en Python.
3. Réaliser des campagnes de simulation pour comparer les performances (nombre d'ensembles ordonnançables, nombre de préemptions, migrations, etc.)

Profil du candidat

Le candidat devra être inscrit en Master 2 ou en dernière année d'école d'ingénieur et posséder des connaissances en programmation. Des notions d'ordonnancement temps réel sont un atout pour ce stage. Un bon niveau en français et en anglais est nécessaire.

Documents à fournir

- Curriculum Vitae ;
- Lettre de motivation ;
- Notes de Master ou équivalent ;
- Tout autre document jugé nécessaire par le candidat pouvant enrichir le dossier de candidature.

Références

[1] Cirinei, M., & Baker, T. P. (2007, July). EDZL scheduling analysis. In *19th Euromicro Conference on Real-Time Systems (ECRTS'07)* (pp. 9-18). IEEE.