

TP série n°4 : optimisation du temps d'exécution d'une application «critique»

Une équipe d'ingénierie logicielle est chargée de développer une application qui automatise la commande d'une batterie antimissiles. S'agissant d'une application temps réel critique de fonctionnement, l'équipe propose de tester le **déploiement** sur une architecture multi-CPU des **threads** qui la composent. Un thread est une **tâche** ou processus ne possédant pas de ressources propres (il s'exécute sur des ressources virtuelles). Déployer un thread consiste à l'affecter à la CPU chargée de l'exécuter complètement.

L'analyse du **cycle d'exécution de l'application** met en évidence **12** threads indépendants codés de **X₁** à **X₁₄**. Ces threads sont exécutés dans le respect des règles de synchronisation exposées dans le tableau ci-dessous. La durée (en nombre de quantum) indiquée pour chaque thread est une **estimation maximisée** de son temps d'exécution.

Code thread	Durée maximale	thread(s) précédent(s)
X ₁	4	-
X ₂	8	-
X ₃	1	-
X ₄	1	X ₃
X ₅	6	X ₁
X ₆	3	X ₁
X ₇	5	X ₂
X ₈	3	X ₅ , X ₆ , X ₇
X ₉	1	X ₄
X ₁₀	2	X ₉
X ₁₁	2	X ₈
X ₁₂	5	X ₁₀ , X ₁₁

Pour l'ingénieur chargé de déployer le logiciel d'application sur une plateforme multi-CPU, l'optimisation du temps d'exécution d'un **cycle** est primordiale et passe par la détermination des 3 facteurs suivants:

- la date au **plus tôt** d'activation de chaque thread,
- la date au **plus tard** d'activation de chaque thread,
- la **durée minimale** du cycle d'exécution,

1-Proposer un modèle de graphe pour représenter le cycle d'exécution de l'application.

On peut utiliser les conventions suivantes:

-les **sommets** du graphe représentent les threads d'application; à chaque sommet, on associe 3 données :

- le **code** du thread,
- sa **date d'activation au plus tôt**, (qui sera calculée en 4)
- et sa **date d'activation au plus tard**. (qui sera calculée en 5)

-les **arcs** représentent les **relations de précédence** entre les threads : l'arc (x,y) signifie que le thread y est précédé immédiatement du thread x .

-le coût associé à un arc (x,y) correspond à la durée maximum d'exécution du thread y .

-deux sommets particuliers sont à distinguer:

- l'un, appelé **Début**, représente un thread virtuel qui **initialise** le cycle: sa durée est nulle; il n'est précédé d'aucun thread. Sa date au plus tôt et sa date au plus tard sont égales à 0.

- l'autre, appelé **Fin**, représente un thread virtuel qui termine le cycle : sa durée est nulle; il ne précède aucun thread,

2- Montrer, sur la base du modèle précédent, que le problème de calcul de la date d'activation **au plus tôt** d'un thread **T** se ramène au problème de recherche d'un **chemin optimal** partant du sommet **Début** et atteignant le sommet représentant le thread **T**.

3-Compte tenu de ce qui précède, proposer une solution pour calculer les dates au **plus tôt** d'activation des threads et la durée optimale du cycle d'exécution

4. Les ingénieurs souhaitent savoir, dans le cadre de la solution proposée, quels sont :

- les threads « critiques » : les threads dont la date d'activation au plus tôt **ne peut pas être retardée**,
- les threads pour lesquels il est possible de retarder la date d'activation au plus tôt sans remettre en cause la solution d'optimisation du cycle d'exécution.

D'où la nécessité de calculer, pour chaque thread, la date d'activation **au plus tard**.

5- Montrer que le problème de calcul des dates **au plus tard** passe par le problème de recherche d'un chemin **optimal** entre **chaque sommet** et le sommet **Fin**.

6-Proposer une solution pour calculer les dates au plus tard.