



HAL
open science

Gestion orienté qualité d'expérience de systèmes embarqués reconfigurables par réutilisation de modules

Alexis Duhamel, Sébastien Pillement, Wiem Kouki

► **To cite this version:**

Alexis Duhamel, Sébastien Pillement, Wiem Kouki. Gestion orienté qualité d'expérience de systèmes embarqués reconfigurables par réutilisation de modules. XXVIIIème Colloque Francophone de Traitement du Signal et des Images, Sep 2022, Nancy, France. , pp.#ID871, 2022, GRETSI'22. hal-03689663

HAL Id: hal-03689663

<https://hal.science/hal-03689663>

Submitted on 12 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1. Contexte et objectifs

❖ La Reconfiguration Dynamique Partielle (DPR) des FPGAs permet le chargement à chaud de Modules Reconfigurables (RM)

- Des architectures auto-reconfigurables permettent d'exploiter la DPR pour améliorer la résilience des systèmes face aux contraintes sur l'application ou l'architecture [1].
- La DPR peut être exploitée pour changer dynamiquement la nature des applications à exécuter au temps de l'exécution [2].

❖ Partitionner et ordonnancer une application pour l'exécuter sur une architecture auto-reconfigurable est un problème NP-Difficile

- La rapidité d'obtention d'une solution est cruciale pour garantir l'exécution d'une application sur cible sous contrainte de temps réel.

❖ Objectifs

- Gérer l'architecture auto-reconfigurable avec une approche qualité d'expérience et garantie d'exécution de service.
- Réduire le nombre de DPR entre deux itérations de l'application par la réutilisation de RM.

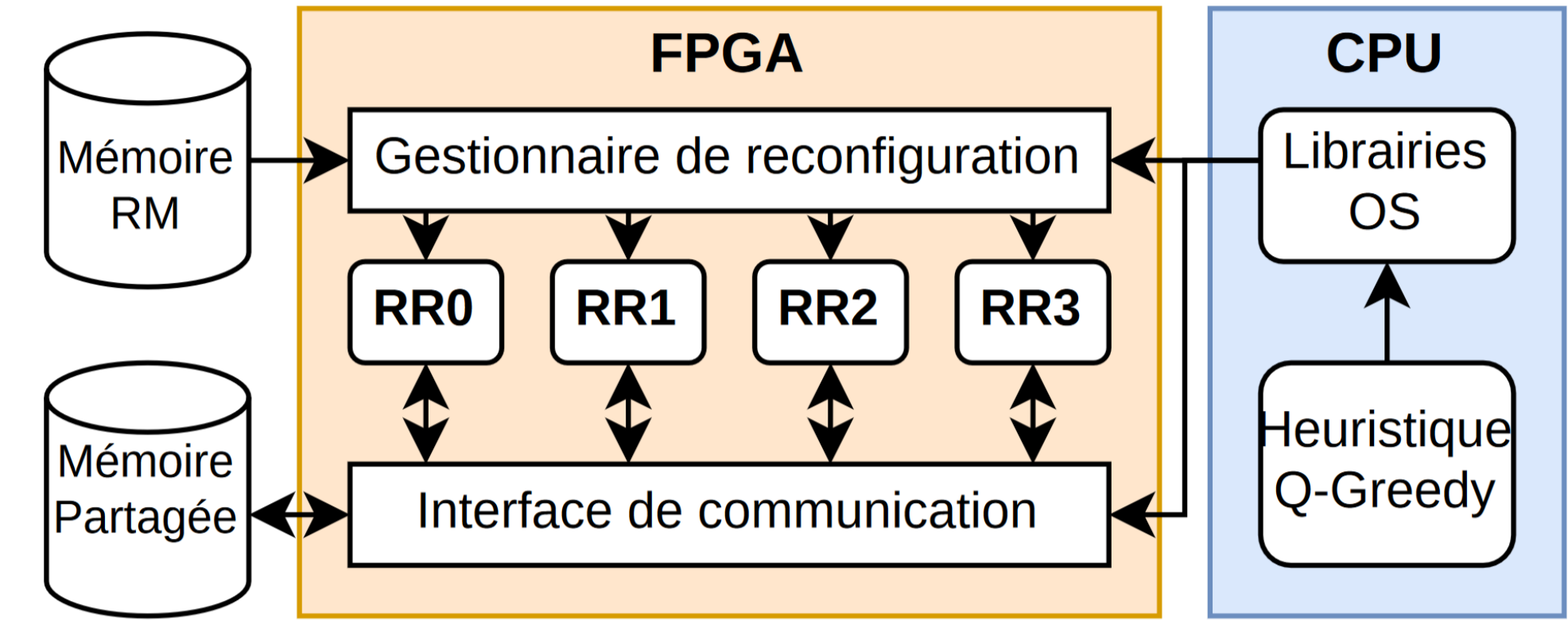


Fig. 1: Vue fonctionnelle de l'architecture auto-reconfigurable

2. Modèles de Qualité

❖ La Qualité d'Expérience (QoE) indique la performance fonctionnelle de l'application

- Peut être modélisée par une fonction mathématique ou empiriquement par *quality assessment* [3].
- Les paramètres fonctionnels définissent les modes d'exécutions, auxquels correspondent un score QoE.

| Mode d'exécution | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------------------|------|------|------|------|------|------|------|------|------|------|
| Nombre de tâches | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 7 | 7 |
| Résolution d'image | 360p | 360p | 480p | 480p | 480p | 480p | 480p | 480p | 480p | 480p |
| Images par seconde | 30 | 60 | 30 | 60 | 30 | 30 | 60 | 60 | 30 | 60 |
| Filtre Sobel | Non | Non | Non | Non | Oui | Non | Oui | Non | Oui | Oui |
| Cryptage AES | Non | Non | Non | Non | Non | Oui | Non | Oui | Oui | Oui |
| Valeur QoE | 0.18 | 0.25 | 0.35 | 0.53 | 0.59 | 0.59 | 0.76 | 0.76 | 0.82 | 1.00 |

Tab. 1 : Modèle QoE d'unencodeur H264

❖ La Qualité de Service (QoS) reflète l'efficacité de l'implémentation de l'application

- Peut dépendre du temps avant échéance temps réel (*slack*), du taux d'occupations des ressources, de la consommation d'énergie, ou une combinaison de plusieurs contraintes.
- Dans notre cas d'étude, nous considérons le *slack* comme métrique de QoS pour évaluer les solutions au problème de partitionnement et d'ordonnancement.

$$Slack = \frac{T_D - T_{App}}{T_D}, \quad \text{avec} \quad \begin{matrix} T_D & \text{temps avant échéance,} \\ T_{App} & \text{durée d'exécution de l'app.} \end{matrix}$$

3. Méthodologie

❖ Méthodologie hybride statique/dynamique

- Génération et évaluation des solutions hors ligne sur machine hôte permettant l'utilisation d'heuristique d'ordonnancement plus gourmand en ressources.
- L'espace des solutions obtenu et évalué par les modèles de qualités est ensuite réduit pour n'obtenir qu'un sous-ensemble de Pareto.

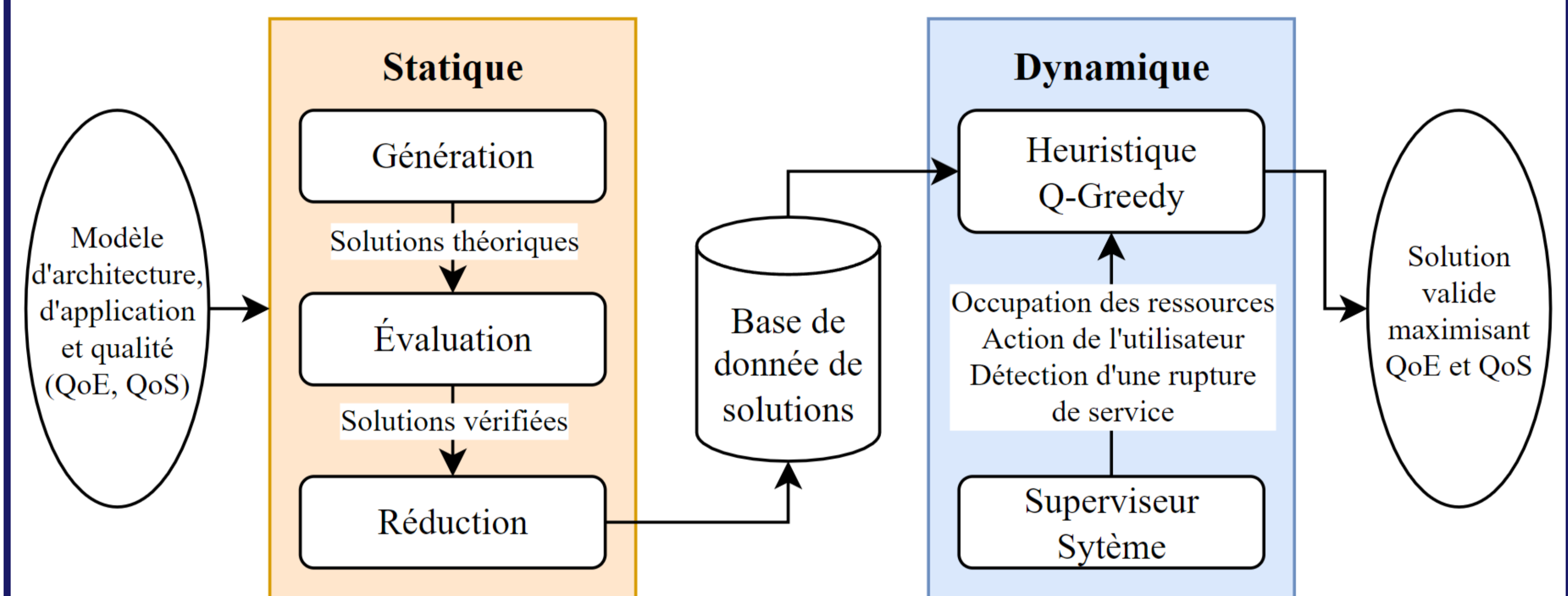


Fig. 2 : Vue d'ensemble de la méthodologie proposée

❖ Optimisation incrémentale de la qualité

- Q-Greedy maximise la QoS d'abord en conservant le mode d'exécution
- Si pas de solution avec une meilleure QoS, il change de mode d'exécution pour un mode avec une meilleure QoE.
- Inversement, il dégrade la QoE s'il ne peut satisfaire les contraintes sur le mode courant (*Slack* < 0), et ce jusqu'à trouver une solution valide.

4. Réutilisation de Modules

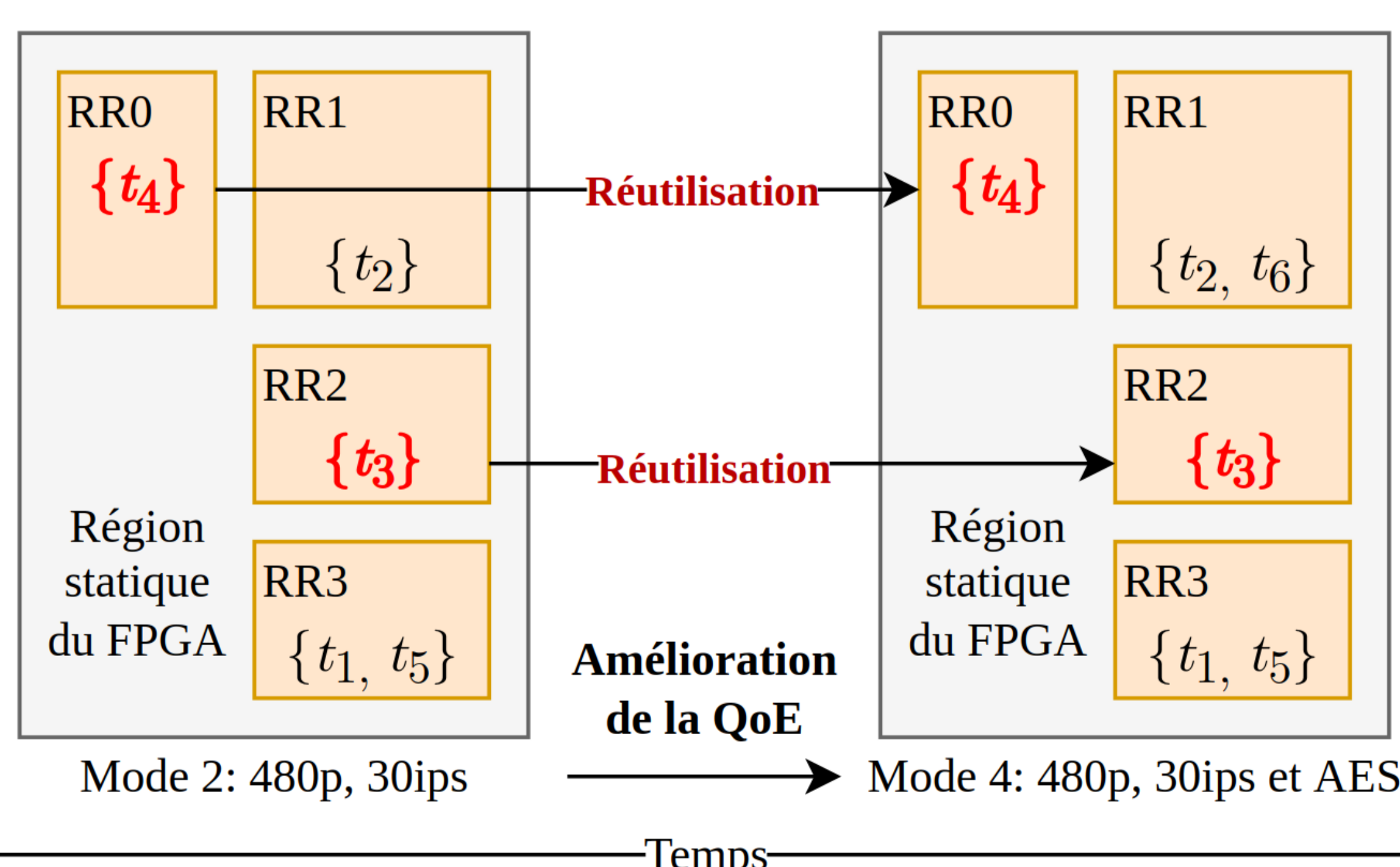


Fig. 3 : Réutilisation des tâches t_3 et t_4 sur les régions reconfigurables (RR) lors d'un changement de mode d'exécution

5. Résultats et perspectives

❖ Étude de cas sur l'encodeur H264

- L'approche Q-Greedy atteint jusqu'à **96% de l'optimal** par recherche exhaustive sur le score QoE.
- La réutilisation de RM est effective dans 12% des scénarios et **sans surcoût de latence** sur la décision.

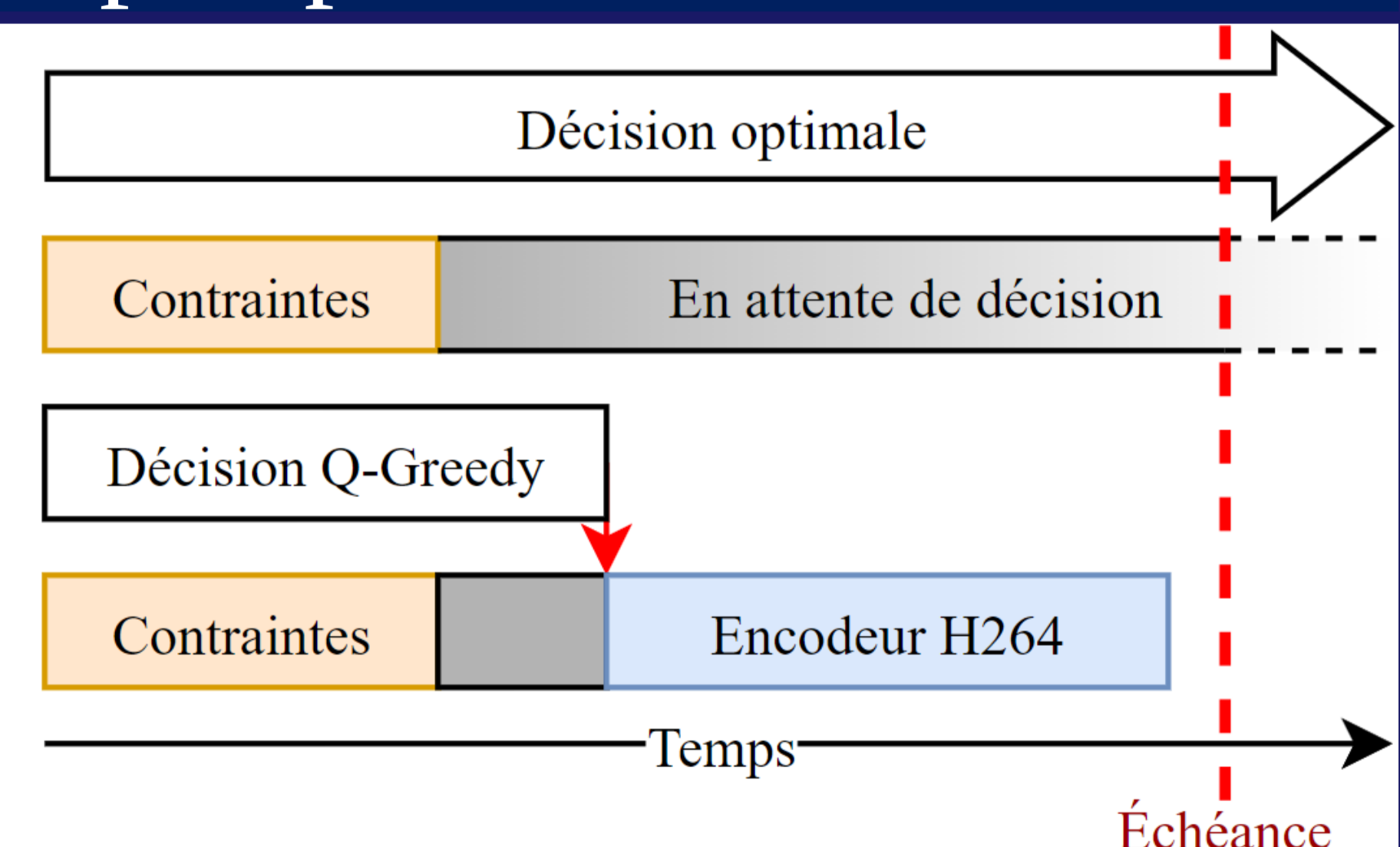


Fig. 4 : Impact du temps de décision sur la latence de l'application

❖ Travaux futurs

- Améliorer la flexibilité de l'approche avec un heuristique d'ordonnancement efficace au temps de l'exécution.
- Définir une méthodologie permettant d'obtenir le meilleur partitionnement d'une application flux de donnée par rapport aux régions reconfigurables.

[1] Sahoo et al., "Multi-objective design space exploration for system partitioning of FPGA-based Dynamic Partially Reconfigurable Systems", Integration, 2019.

[2] Hireche et al., "Context/Resource-Aware Mission Planning Based on BNs and Concurrent MDPs for Autonomous UAVs", MDPI, 2018.

[3] Debattista et al., "Frame Rate vs Resolution: A Subjective Evaluation of Spatiotemporal Perceived Quality Under Varying Computational Budgets", Computer Graphics Forum, 2017.