Optimisation mémoire dans une architecture NUMA : comparaison des gains entre natif et virtualisé

Gauthier Voron
Gaël Thomas
Pierre Sens
Vivien Quéma

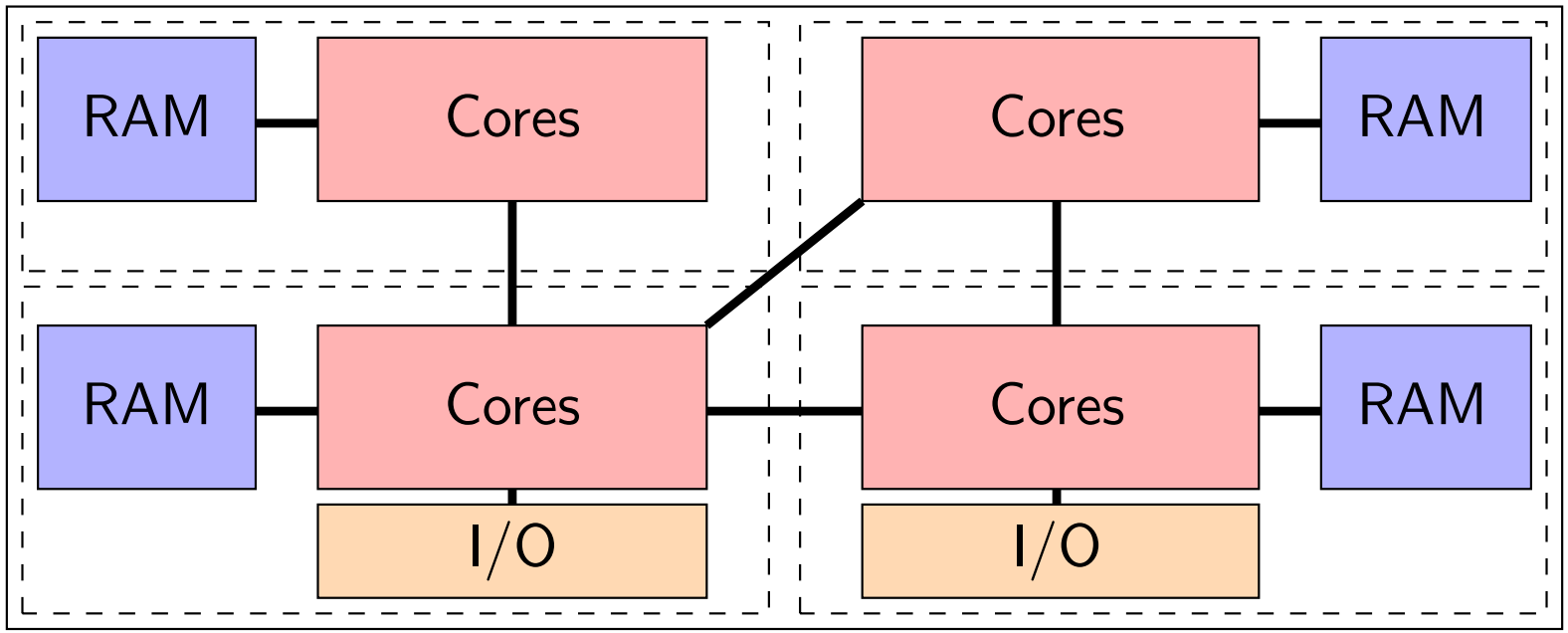
La virtualisation

- Cloud
- Entreprise

- Contrôle des ressources
- Fragmentation, consolidation

- Serveurs, data miniming
- Postes virtuels, services internes

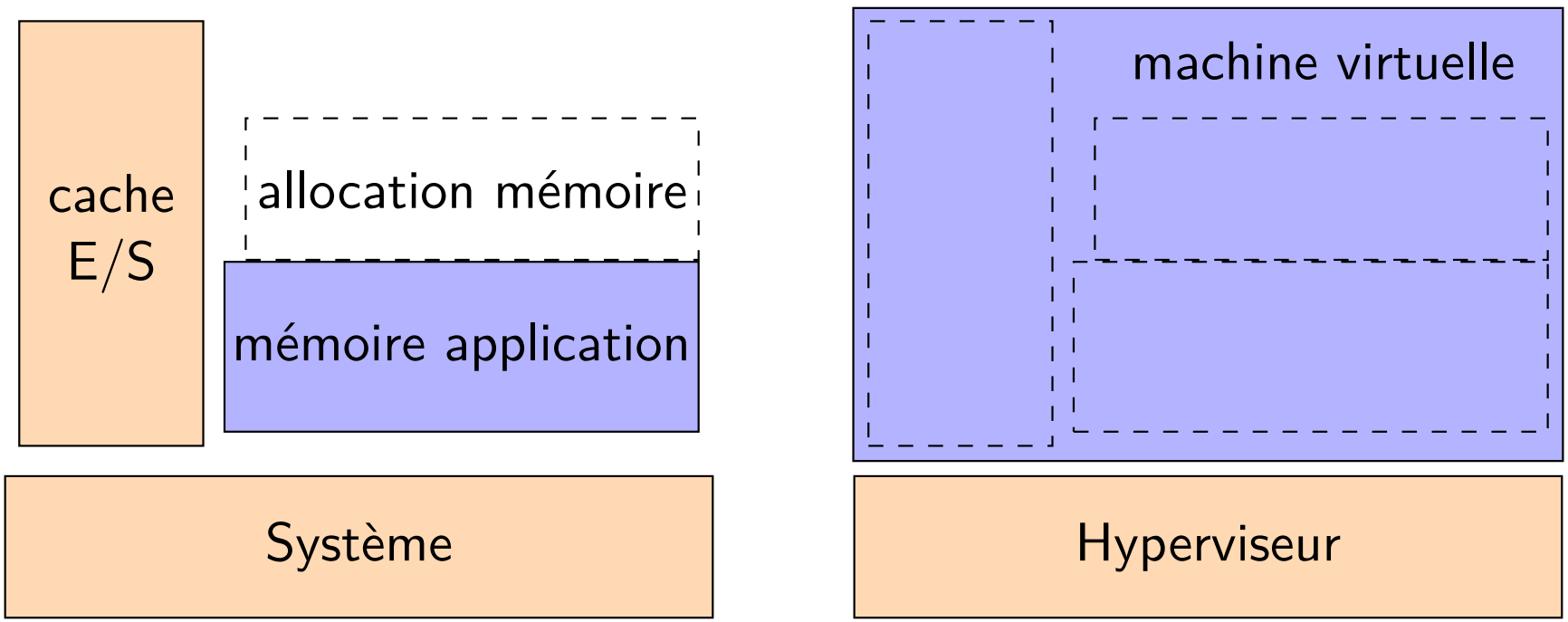
L'architecture NUMA



- Permet de distribuer la charge du bus
- Cohérence mémoire (ccNUMA)

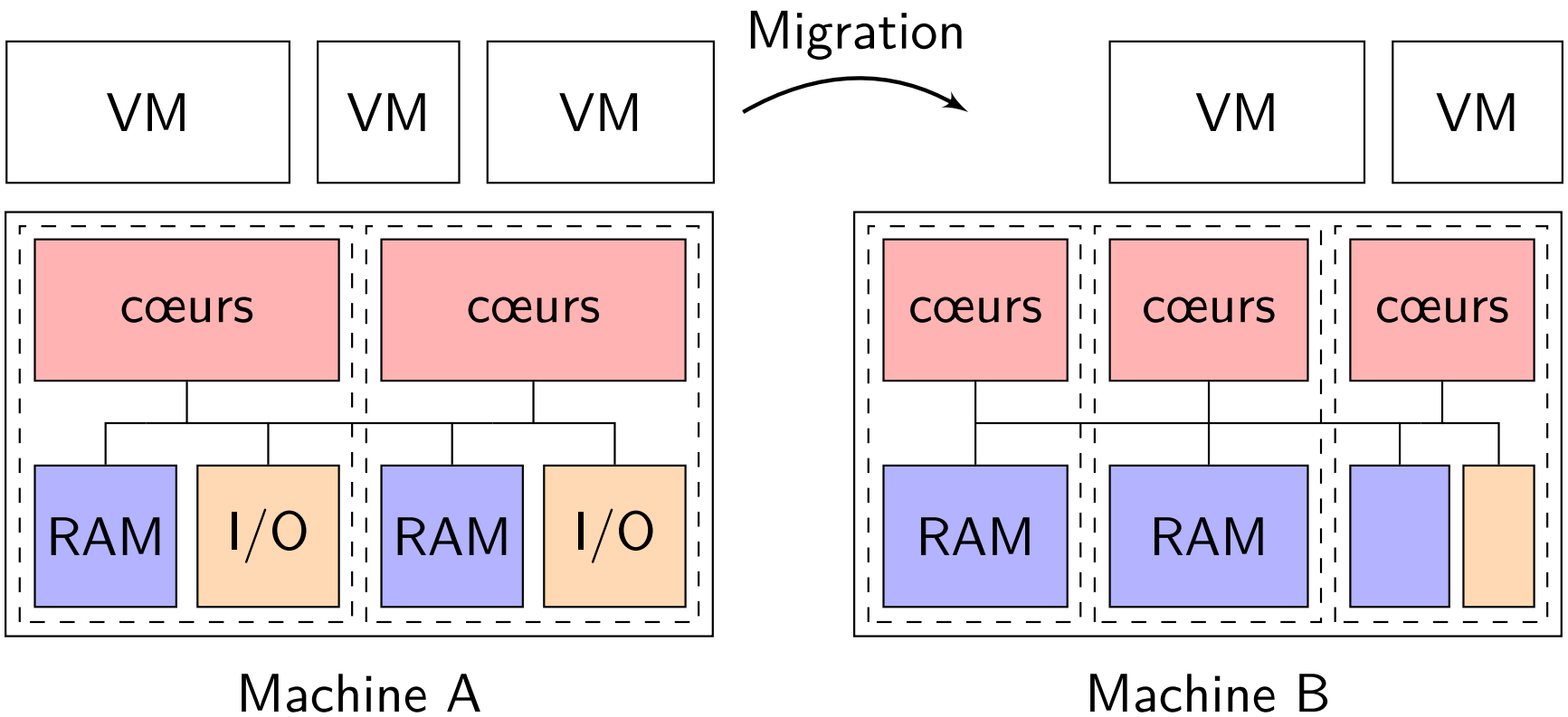
Les hyperviseurs sont inefficaces sur NUMA

- Les invités masquent leur usage de la mémoire à l'hyperviseur
  - Pas d'allocation "first-touch" de l'hyperviseur
  - Pas de distinction : mémoire système / mémoire de l'application



Les systèmes invités sont inefficaces sur NUMA

- Les hyperviseurs masquent la topologie NUMA aux invités
- Abstraction du matériel : migration de machines virtuelles à chaud



Adapter l'état de l'art aux hyperviseurs

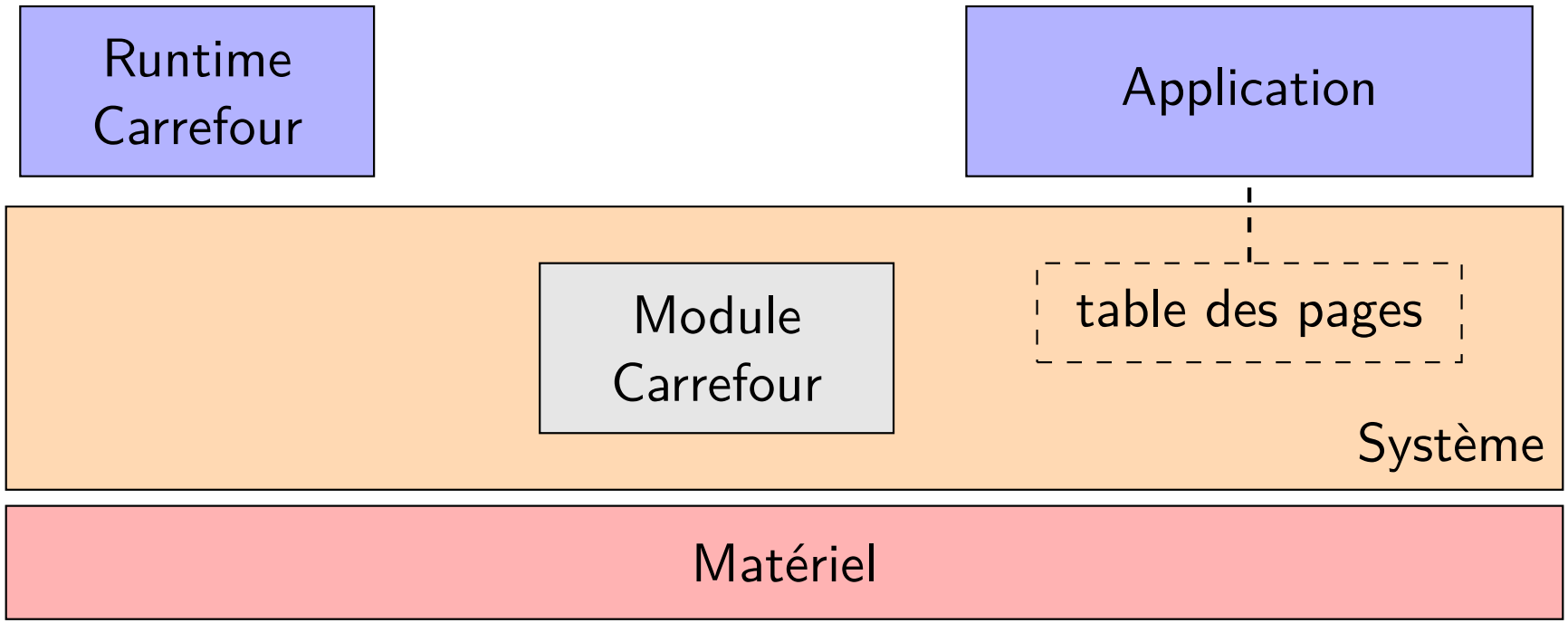
- Etat de l'art : Carrefour (Asplos 2013)
- Déplacement des données après leur allocation

- Désaturation des contrôleurs mémoire locaux
  - Si page P accédée par plus d'un noeud $N_0 \ldots N_n$
  - Alors migrer P sur un des noeuds $N_0 \ldots N_n$ au hasard

- Désaturation de l'interconnect
  - Si page P accédée exclusivement par un noeud N
  - Alors migrer P sur N

Adapter l'état de l'art aux hyperviseurs

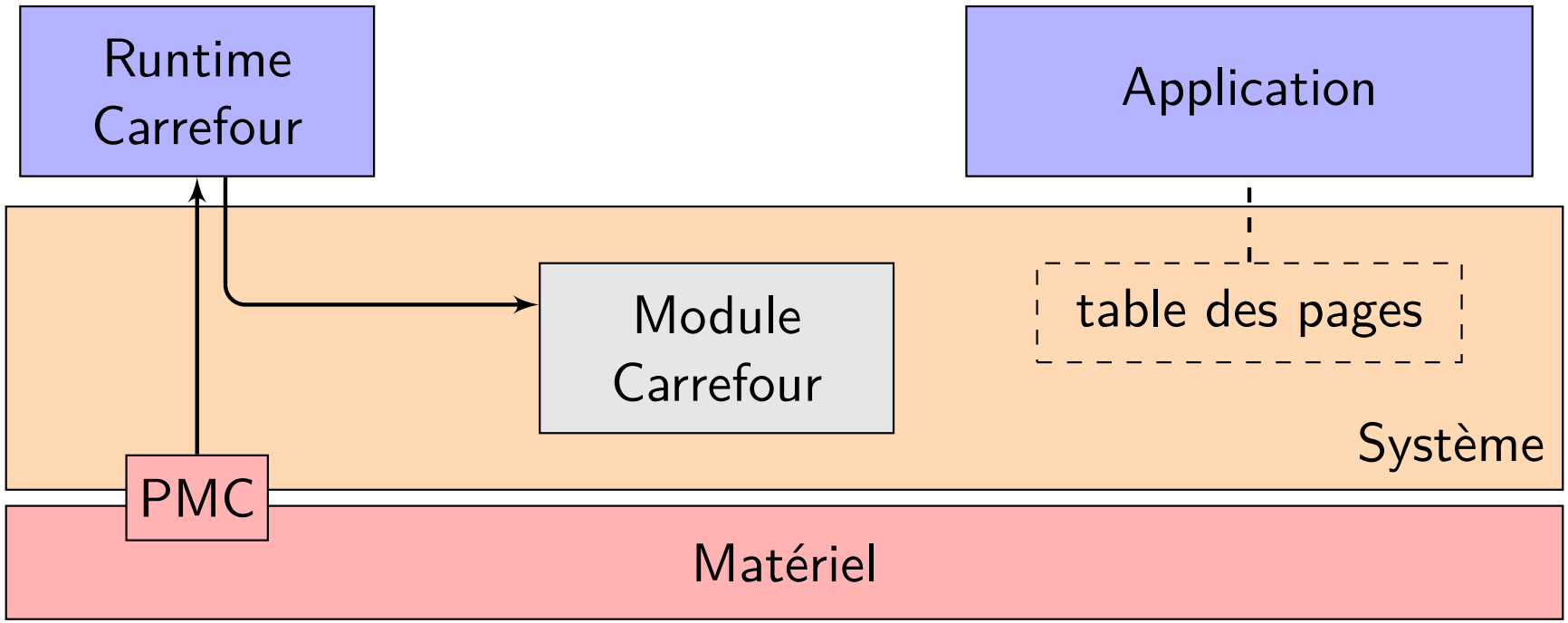
- Structure de Carrefour



- Le runtime collecte des données globales et décide d'activer le module
- Le module collecte des données locales et décide où placer les pages

Adapter l'état de l'art aux hyperviseurs

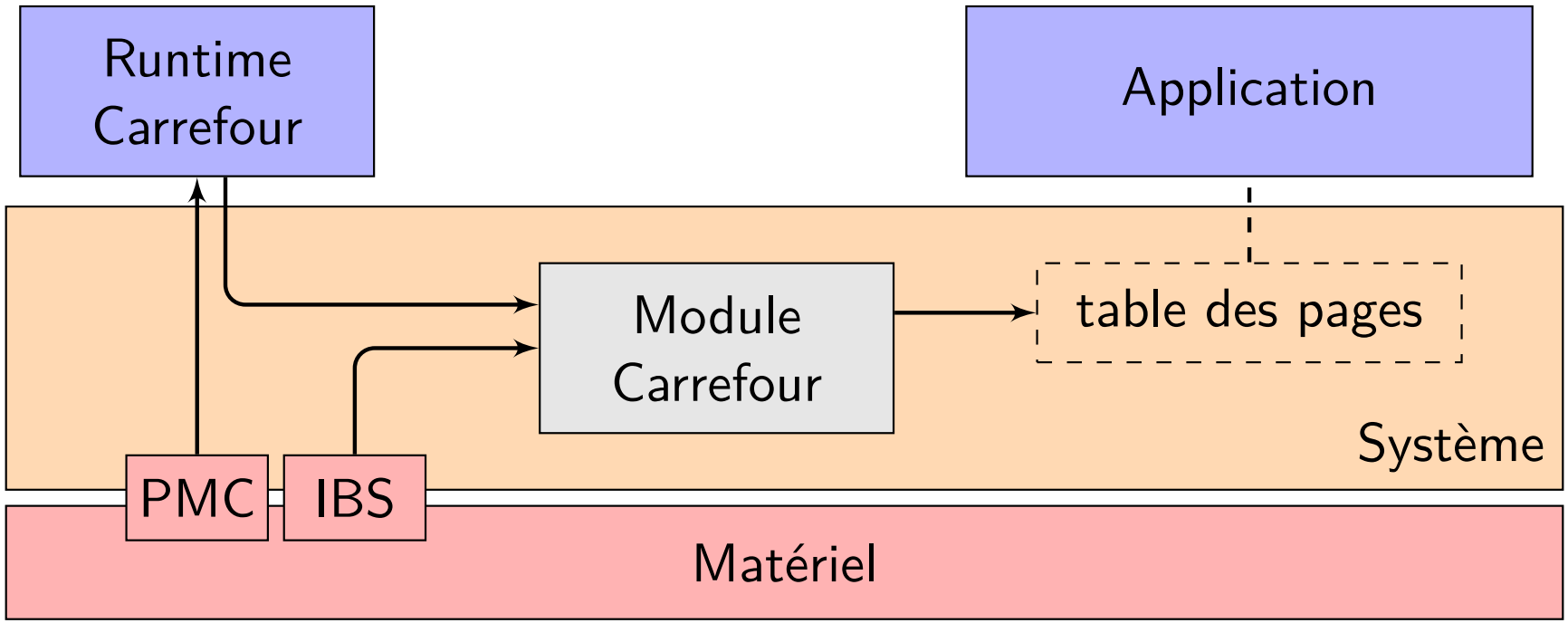
- Structure de Carrefour



- Le runtime collecte des données globales et décide d'activer le module
- Le module collecte des données locales et décide où placer les pages

Adapter l'état de l'art aux hyperviseurs

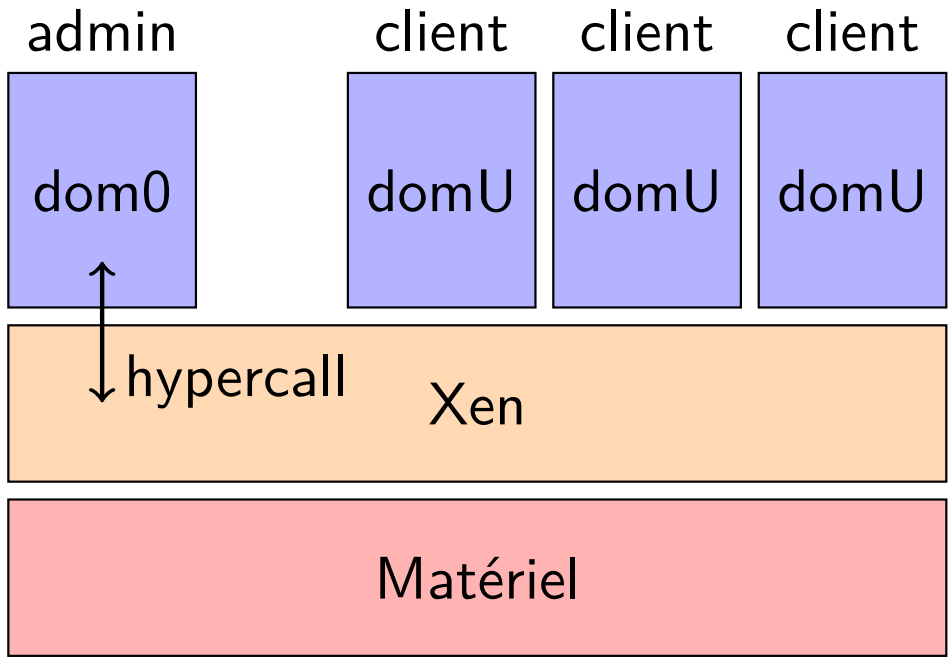
- Structure de Carrefour



- Le runtime collecte des données globales et décide d'activer le module
- Le module collecte des données locales et décide où placer les pages

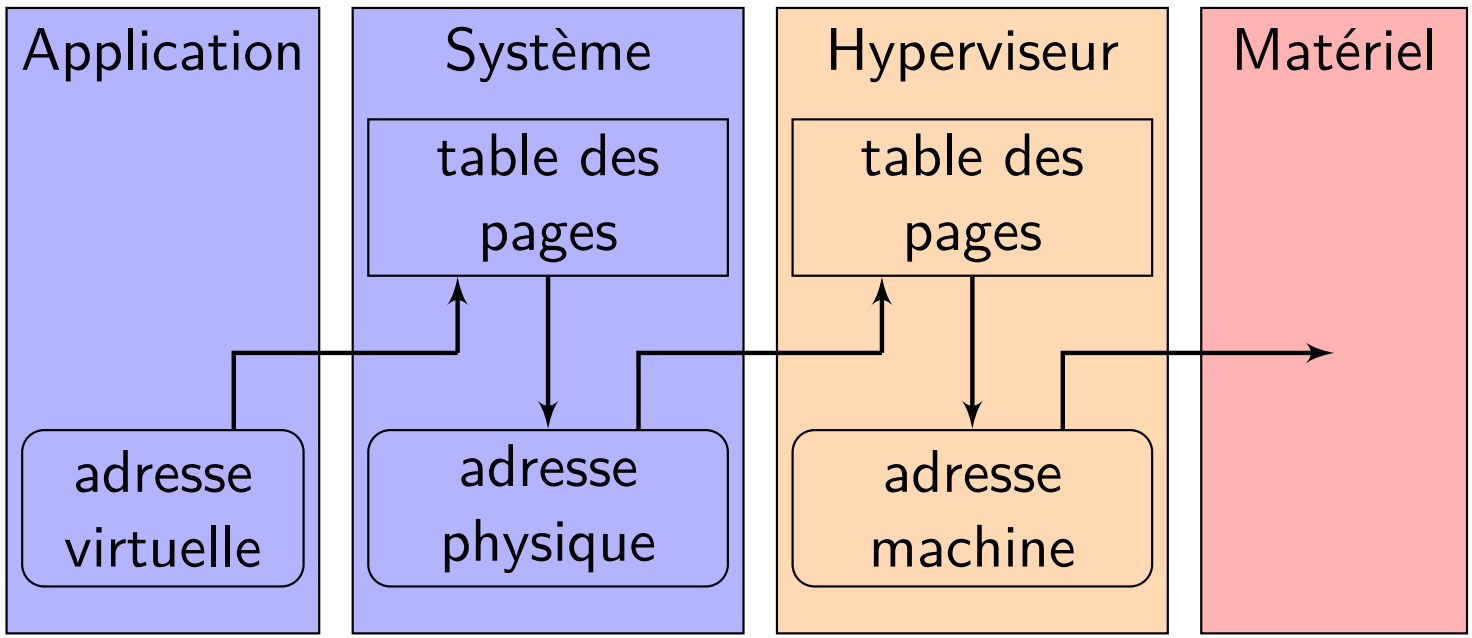
Xen et la virtualisation matérielle

- Xen exécute un domaine dom0 et des domaines domU
- Le Hardware Assisted Paging ajoute un niveau de traduction
- Xen alloue sa mémoire en tourniquet par bloc de 1 Gio



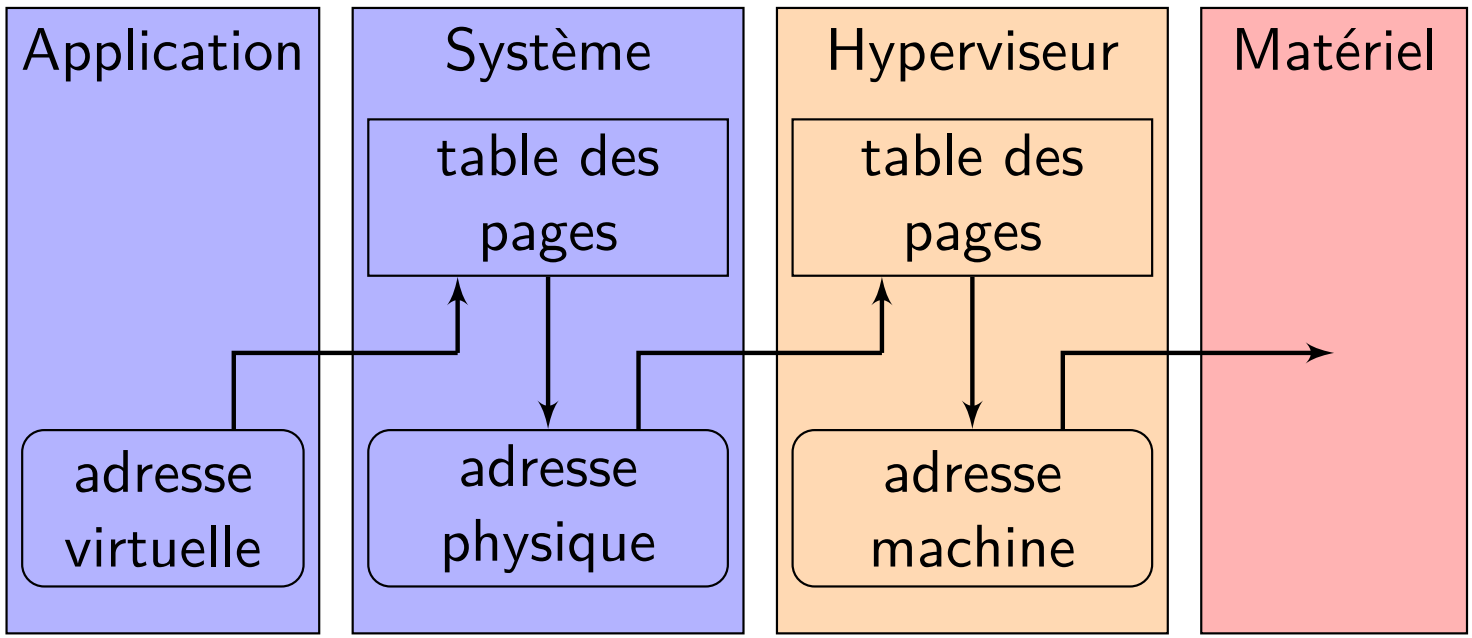
Xen et la virtualisation matérielle

- Xen exécute un domaine dom0 et des domaines domU
- Le Hardware Assisted Paging ajoute un niveau de traduction
- Xen alloue sa mémoire en tourniquet par bloc de 1 Gio



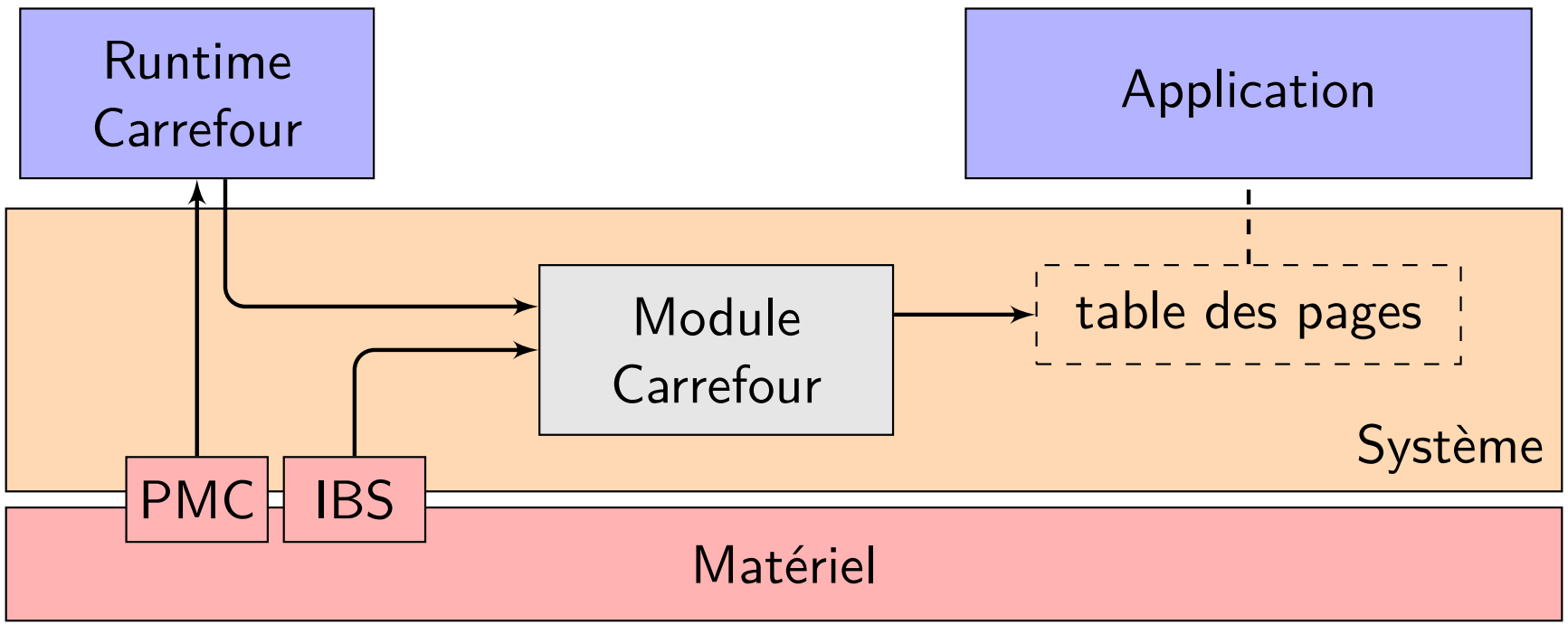
Xen et la virtualisation matérielle

- Xen exécute un domaine dom0 et des domaines domU
- Le Hardware Assisted Paging ajoute un niveau de traduction
- Xen alloue sa mémoire en tourniquet par bloc de 1 Gio



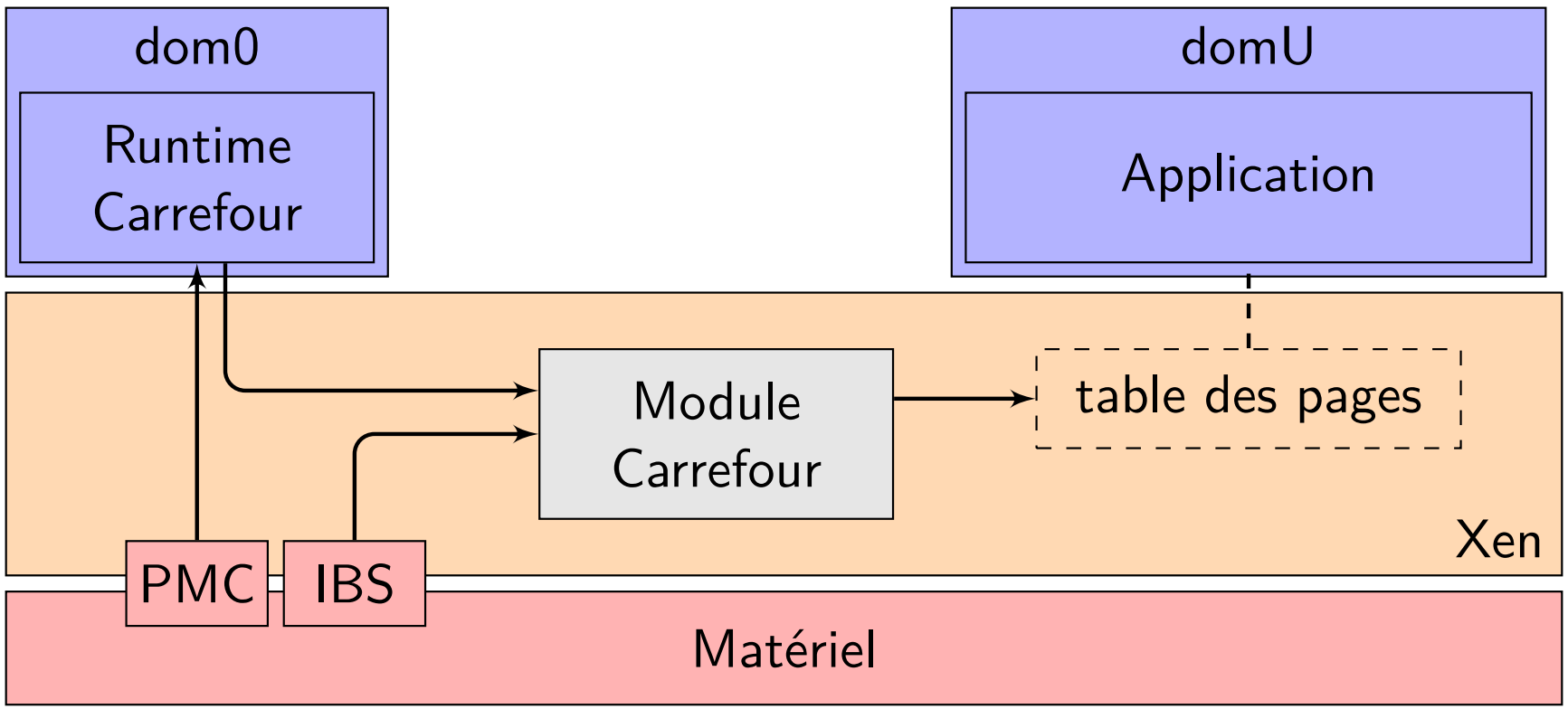
Adapter Carrefour à Xen

- Déplacement du runtime dans le dom0 → hypercall
- Modification de l'association @physique → @machine



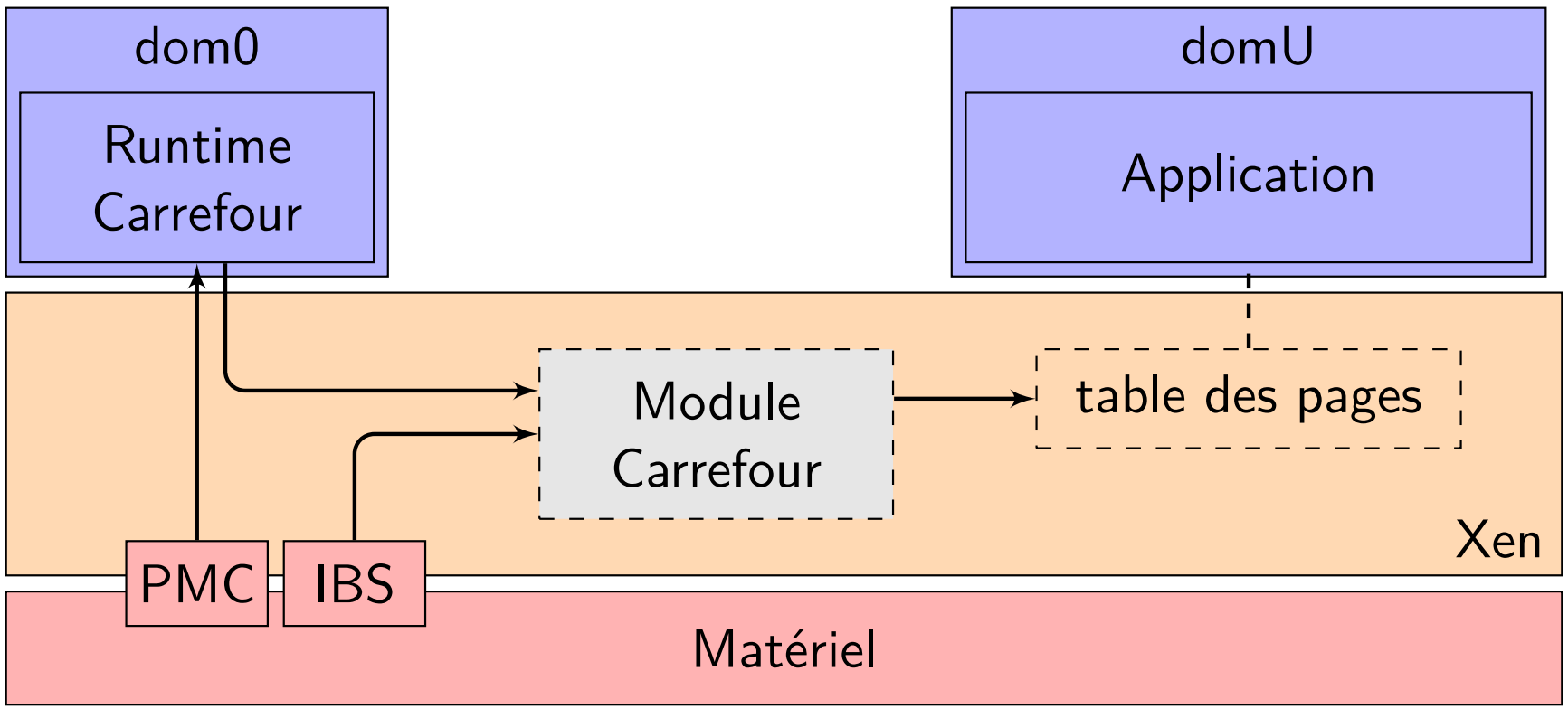
Adapter Carrefour à Xen

- Déplacement du runtime dans le dom0 → hypercall
- Modification de l'association @physique → @machine



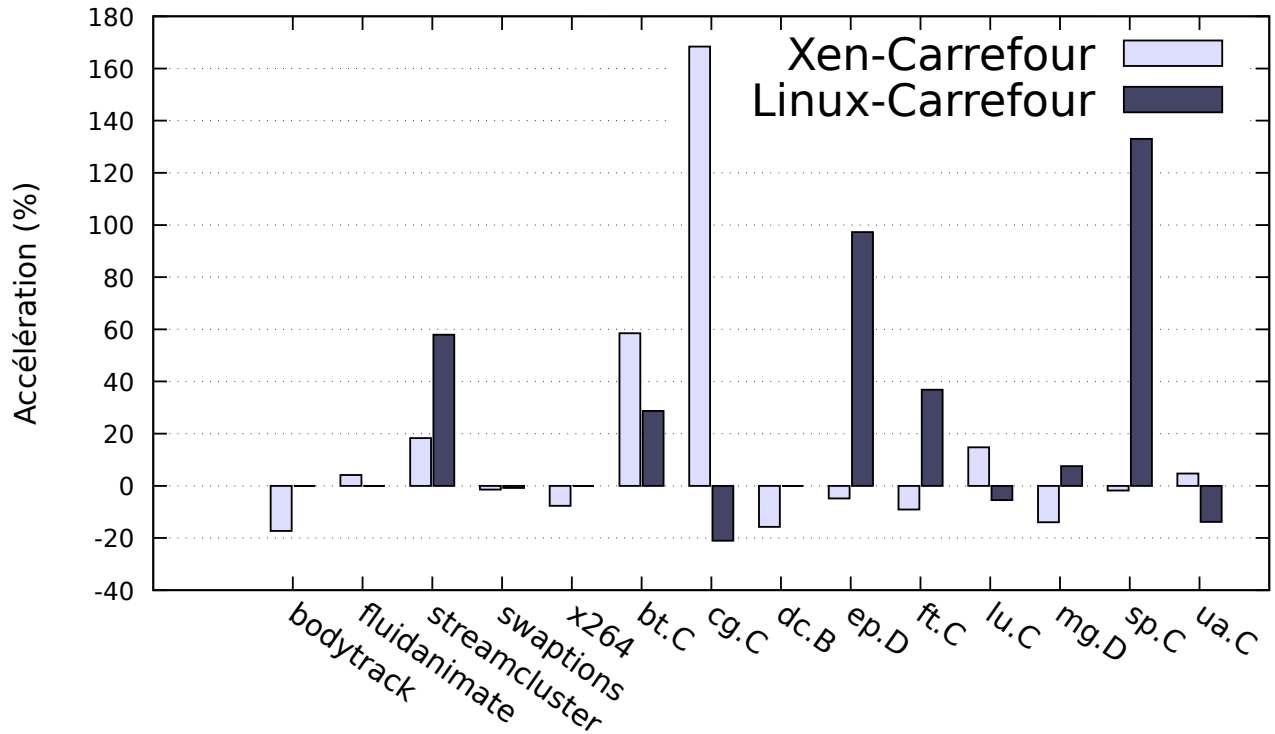
Adapter Carrefour à Xen

- Déplacement du runtime dans le dom0 $\rightarrow$ hypercall
- Modification de l'association @physique $\rightarrow$ @machine



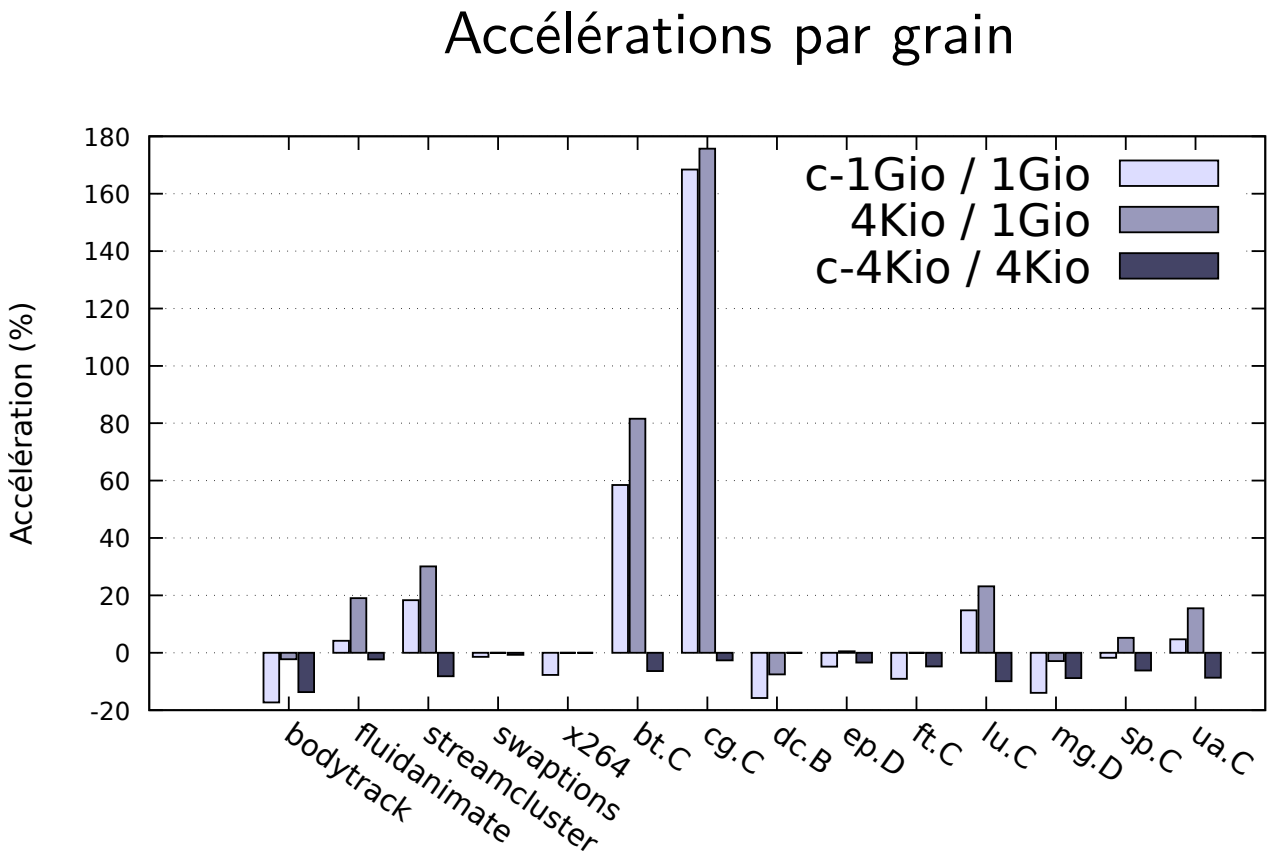
Evaluation des performances

Accélération pour Xen-Carrefour et Linux-Carrefour



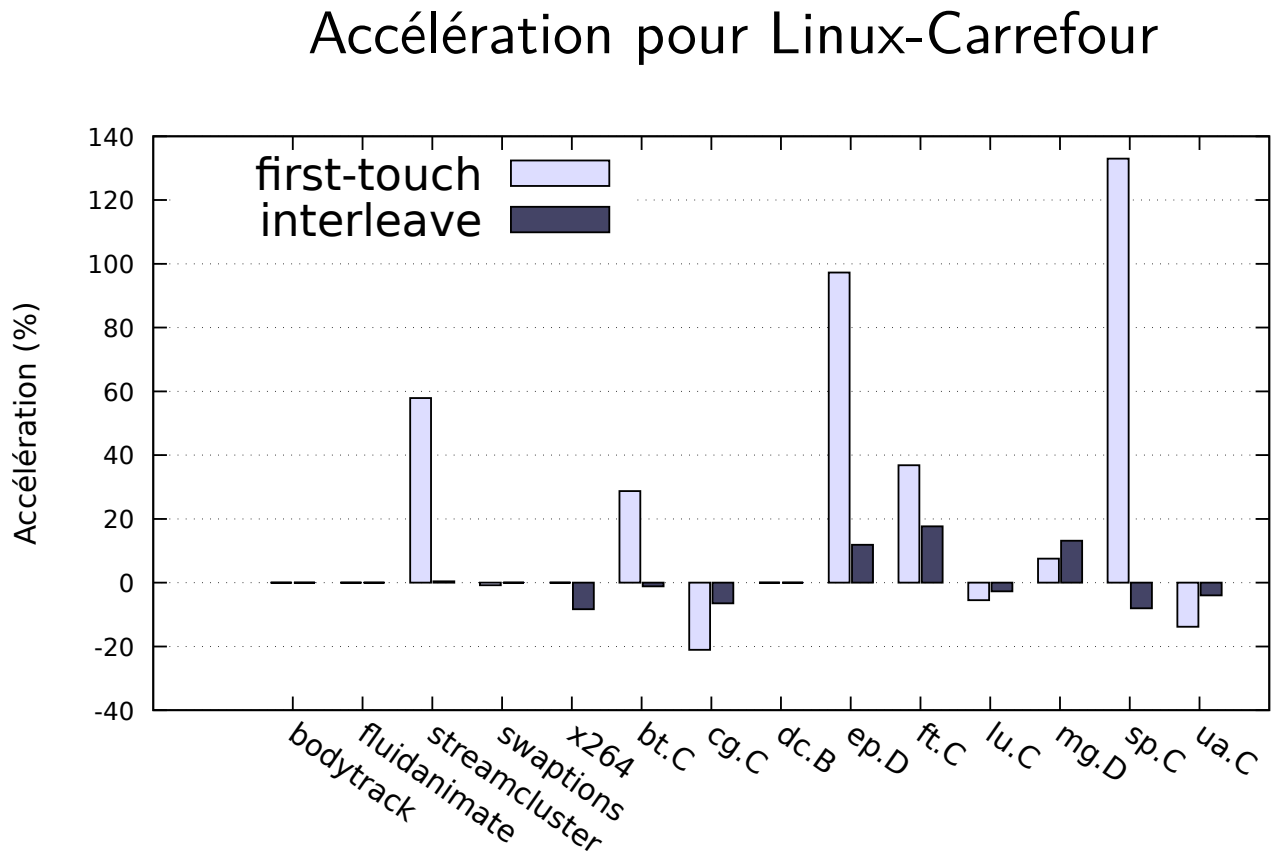
- Accélération pour certaines applications
- Pas d'effet ou faible dégradation pour les autres
- Répartition des applications : Xen $\neq$ Linux

Evaluation des performances : équilibrage



- Accélération quand le grain d'entrelacement est gros
- Dégradation quand le grain d'allocation est fin
- Xen-Carrefour équilibre la charge

Evaluation des performances : localité



- Accélération quand le grain d'allocation est gros
- Accélération (faible) quand le grain d'allocation est fin
- Linux-Carrefour améliore la localité

Conclusion

- Les optimisations existantes sont partiellement applicables
  - Implémentation possible dans les hyperviseurs
  - Equilibrage de charge efficace
- Les solutions actuelles sont insuffisantes
  - Echantillonage mal gérée pendant la virtualisation
  - La localité n'est pas améliorée
- Merci pour votre attention

Conclusion

- Les optimisations existantes sont partiellement applicables
  - Implémentation possible dans les hyperviseurs
  - Equilibrage de charge efficace
- Les solutions actuelles sont insuffisantes
  - Echantillonage mal gérée pendant la virtualisation
  - La localité n'est pas améliorée
- Merci pour votre attention