DRAC: Un système de contrôle d'exécution pour multiprocesseur à mémoire partagée.

Maurício Aronne Pillon

Directeur de thèse : M^me^ Brigitte Plateau
Co-encadrant : M. Olivier Richard

Laboratoire ID-IMAG (UMR 5132), Grenoble, France.

Conselho Nacional de Desenvolvimento Científico e Tecnologico

Laboratoire Informatique et Distribution

Institut National Polytechnique de Grenoble

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Introduction

Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.

Problème

- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.

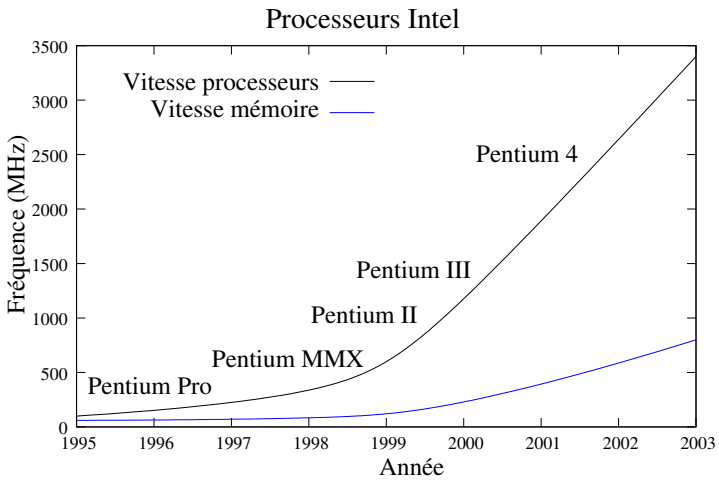
Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.

Problème

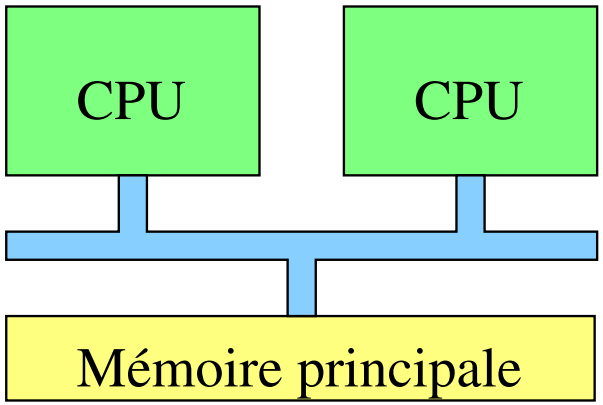
- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.



Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.



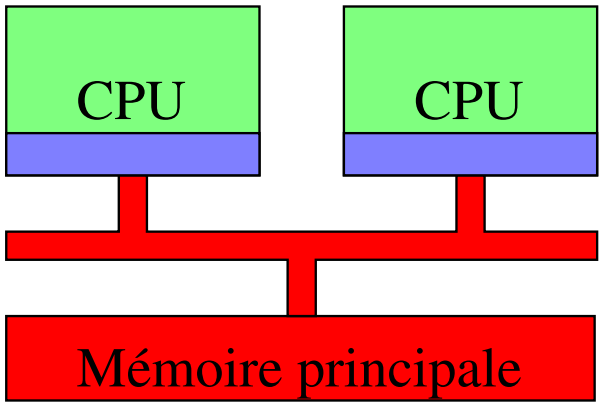
Problème

- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.

Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.



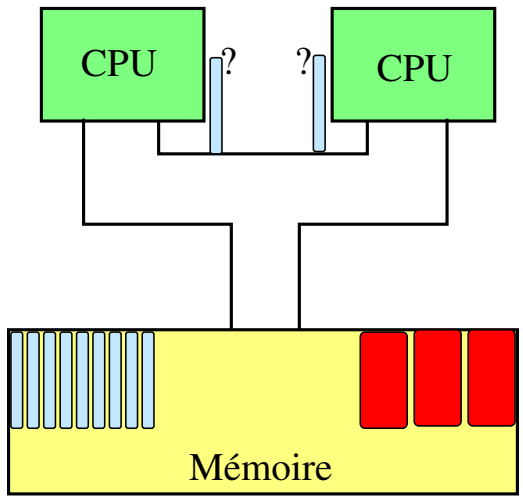
Problème

- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.

Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.



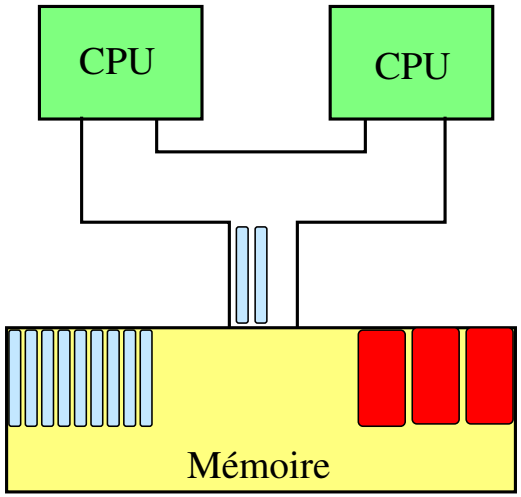
Problème

- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.

Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.



Problème

- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.

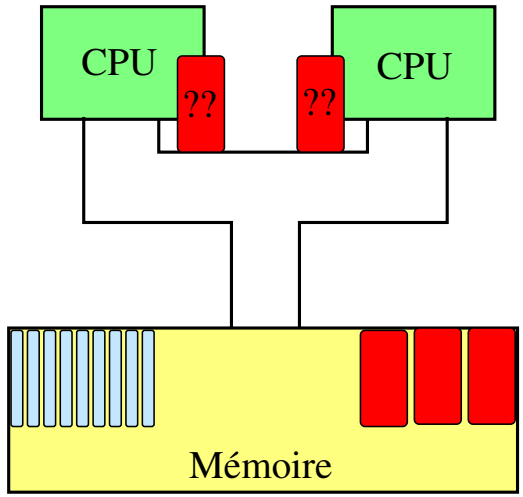
Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.

Problème

- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.



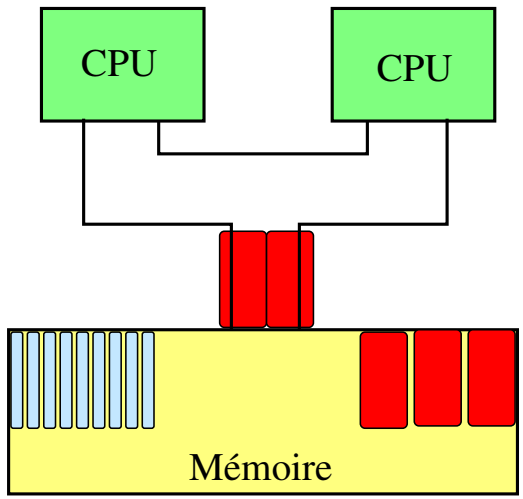
Introduction

Motivations

- Besoins continus en puissance de calcul.
- Évolution des technologies de fabrication : hiérarchie mémoire vs. vitesse des processeurs.
- Multiprocesseurs à mémoire partagée : goulot d'étranglement mémoire.

Problème

- processus léger vs. processus léger (en débit mémoire) : sous-utilisation de la hiérarchie mémoire.
- processus lourd vs. processus lourd (en débit mémoire) : saturation de la hiérarchie mémoire.



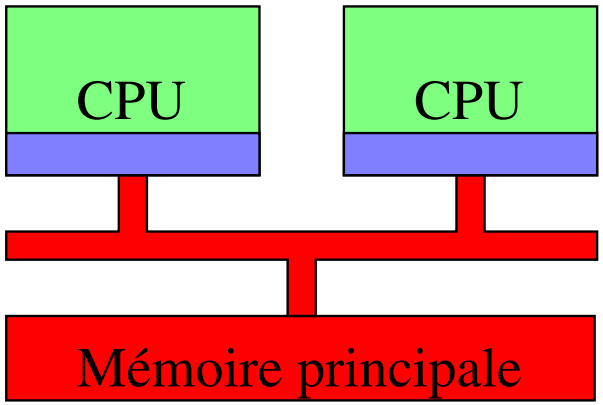
Introduction (cont.)

Objectif

- Augmenter le rendement de la machine.

Solutions possibles

1. Augmenter les voies de communication entre la mémoire et les processeurs.
2. Restructurer les codes applicatifs.
3. Équilibrer l'utilisation des ressources.



Proposition

- Système de contrôle modifiant l'ordre d'exécution des processus lors de l'identification de la présence d'une contention sur la hiérarchie mémoire.

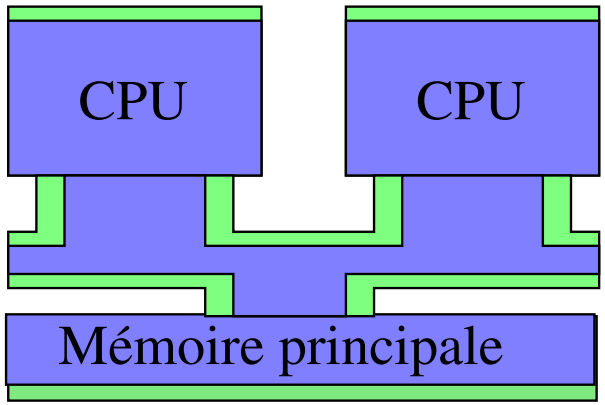
Introduction (cont.)

Objectif

- Augmenter le rendement de la machine.

Solutions possibles

1. Augmenter les voies de communication entre la mémoire et les processeurs.
2. Restructurer les codes applicatifs.
3. Équilibrer l'utilisation des ressources.



Proposition

- Système de contrôle modifiant l'ordre d'exécution des processus lors de l'identification de la présence d'une contention sur la hiérarchie mémoire.

Introduction (cont.)

Objectif

- Augmenter le rendement de la machine.

Solutions possibles

1. Augmenter les voies de communication entre la mémoire et les processeurs.
2. Restructurer les codes applicatifs.
3. Équilibrer l'utilisation des ressources.

Proposition

- Système de contrôle modifiant l'ordre d'exécution des processus lors de l'identification de la présence d'une contention sur la hiérarchie mémoire.

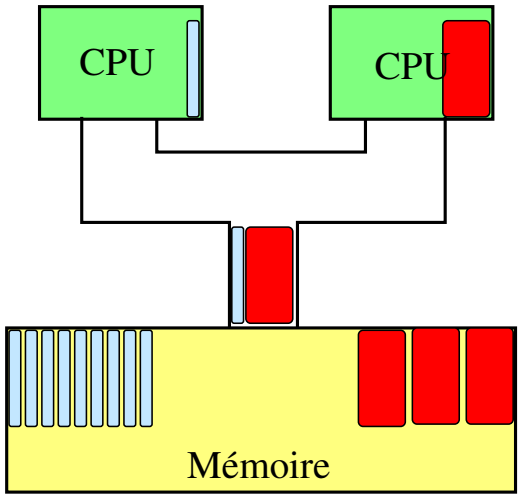
Introduction (cont.)

Objectif

- Augmenter le rendement de la machine.

Solutions possibles

1. Augmenter les voies de communication entre la mémoire et les processeurs.
2. Restructurer les codes applicatifs.
3. Équilibrer l'utilisation des ressources.



Proposition

- Système de contrôle modifiant l'ordre d'exécution des processus lors de l'identification de la présence d'une contention sur la hiérarchie mémoire.

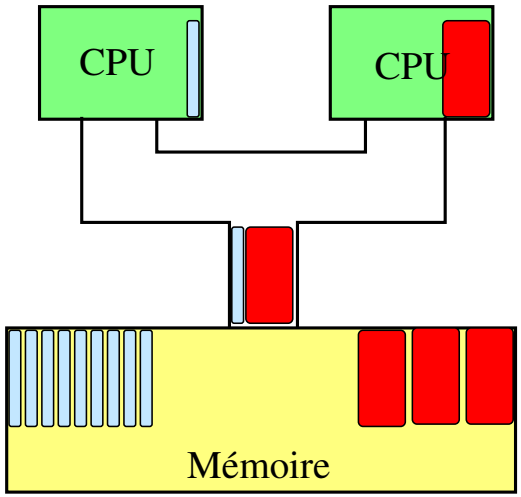
Introduction (cont.)

Objectif

- Augmenter le rendement de la machine.

Solutions possibles

1. Augmenter les voies de communication entre la mémoire et les processeurs.
2. Restructurer les codes applicatifs.
3. Équilibrer l'utilisation des ressources.



Proposition

- Système de contrôle modifiant l'ordre d'exécution des processus lors de l'identification de la présence d'une contention sur la hiérarchie mémoire.

Introduction (cont.)

Conditions nécessaires à la réalisation du système

1. détecter dynamiquement le rendement de la machine.
   - compteurs matériels de performances.
2. prendre des décisions et les appliquer sur le système lors de l'exécution.
   - systèmes adaptables.
3. avoir des applications ayant un comportement stable.
4. avoir des applications ayant des besoins en débit mémoire différents.

Introduction (cont.)

Conditions nécessaires à la réalisation du système

1. détecter dynamiquement le rendement de la machine.
   - compteurs matériels de performances.
2. prendre des décisions et les appliquer sur le système lors de l'exécution.
   - systèmes adaptables.
3. avoir des applications ayant un comportement stable.
4. avoir des applications ayant des besoins en débit mémoire différents.

Introduction (cont.)

Conditions nécessaires à la réalisation du système

1. détecter dynamiquement le rendement de la machine.
   - compteurs matériels de performances.
2. prendre des décisions et les appliquer sur le système lors de l'exécution.
   - systèmes adaptables.
3. avoir des applications ayant un comportement stable.
4. avoir des applications ayant des besoins en débit mémoire différents.

Introduction (cont.)

Conditions nécessaires à la réalisation du système

1. détecter dynamiquement le rendement de la machine.
   - compteurs matériels de performances.
2. prendre des décisions et les appliquer sur le système lors de l'exécution.
   - systèmes adaptables.
3. avoir des applications ayant un comportement stable.
4. avoir des applications ayant des besoins en débit mémoire différents.

Étude préalable à réalisation du système

Compteurs matériels des performances

Définition et objectif

- Un compteur matériel est un registre placé dans les processeurs.
- Il permet l'observation précise des événements (Ex. *Cache miss*, *Floating-point operations*) au niveau matériel.

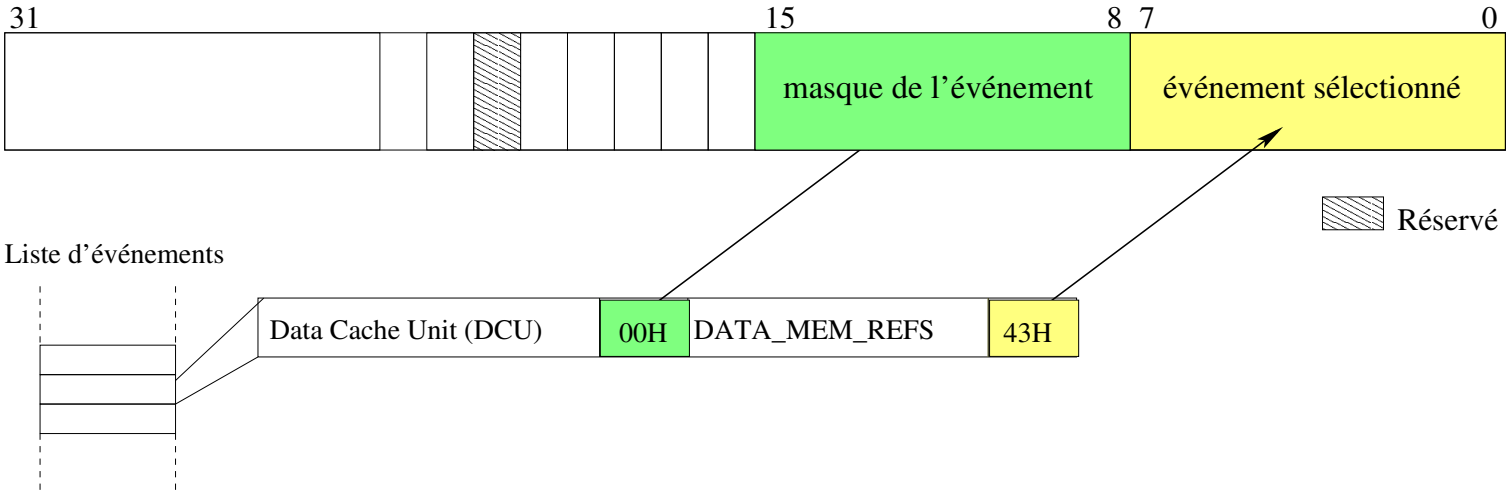
Architectures processeurs et leurs compteurs matériels

Architectures processeurs	Nb de compteurs	Nb d'événements	Nb d'événements mémoire
Famille Pentium 6	2	$\sim$ 75	$\sim$ 30
Famille Pentium 4 et Xeon	18	+200	$\sim$ 100
Famille Itanium	4	+150	$\sim$ 100
Athlon (32*bits*)	4	$\sim$ 20	$\sim$ 10
Athlon (64*bits*) et Opteron	4	+150	$\sim$ 80
MIPS 10000	2	$\sim$ 30	$\sim$ 15
MIPS 12000	4	$\sim$ 30	$\sim$ 15
Sun UltraSparc I/II/III	2	$\sim$ 25	$\sim$ 15
PowerPC 604	2	$\sim$ 25	$\sim$ 10
PowerPC 604e	4	$\sim$ 100	$\sim$ 25
Power3/II	8	+100	$\sim$ 50
DEC Alpha 21264	2	$\sim$ 10	aucun
DEC Alpha 21164	3	$\sim$ 49	$\sim$ 30

Exemple de configuration d'un événement matériel

Processeurs de la famille Intel Pentium 6

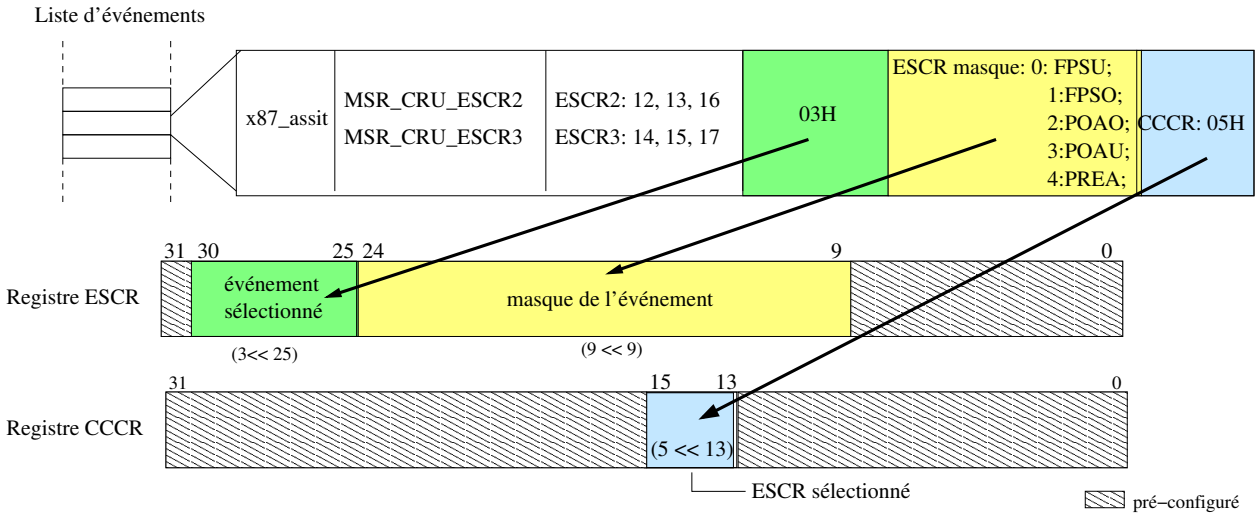
- 2 compteurs matériels et $\sim$ 75 événements.
- Registre de 40*bits* (PerfEvtSel).
- Les événements sont divisés par groupe selon les activités qu'ils observent (Ex. *Data Cache Unit*, *Floating-Point Unit*).



Exemple de configuration d'un événement matériel

Processeurs de la famille Intel Pentium 4

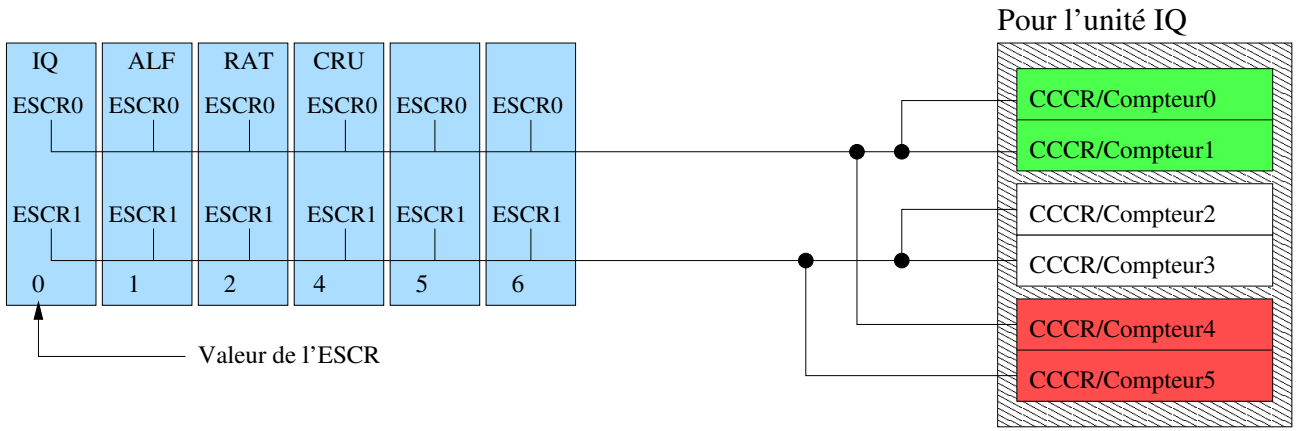
- 18 compteurs matériels et +200 événements.
- Registres ESCR (contrôle de sélection d'événement) et CCCR (contrôle de configuration du compteur).
- Chaque événement est lié à une unité et par conséquent à un groupe de ESCR.



Exemple de configuration d'un événement matériel

Processeurs de la famille Intel Pentium 4

- 18 compteurs matériels et +200 événements.
- Registres ESCR (contrôle de sélection d'événement) et CCCR (contrôle de configuration du compteur).
- Chaque événement est lié à une unité et par conséquent à un groupe de ESCR.



Bibliothèques d'accès aux compteurs de performances

Objectif

- Fournir l'accès aux compteurs matériels de performances sur un ensemble d'architectures.

Exemple

- HPM Toolkit - *Hardware Performance Monitor* (IBM)
- Perfex (processeurs MIPS)
- Vtune - *VTune Performance Analyser* (Intel)
- PAPI - *Portable Programming Interface for Performance*
- PCL - *Performance Counter Library*

Bilan

- HPM, Perfex et VTune supportent les architectures d'un seul fabricant.
- PAPI et PCL proposent des interfaces génériques d'accès aux compteurs.

Bibliothèques d'accès aux compteurs de performances

Objectif

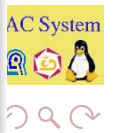
- Fournir l'accès aux compteurs matériels de performances sur un ensemble d'architectures.

Exemple

- HPM Toolkit - *Hardware Performance Monitor* (IBM)
- Perfex (processeurs MIPS)
- Vtune - *VTune Performance Analyser* (Intel)
- PAPI - *Portable Programming Interface for Performance*
- PCL - *Performance Counter Library*

Bilan

- HPM, Perfex et VTune supportent les architectures d'un seul fabricant.
- PAPI et PCL proposent des interfaces génériques d'accès aux compteurs.



Définition d'un système adaptable

Définition (domaine d'automatisme)

- Un système adaptable est un système capable d'ajuster les conditions de son exécution en fonction de ses besoins ou de ceux d'un autre système.

Fonctionnalité

- L'adaptation d'un système informatique peut consister à optimiser l'utilisation de l'ensemble des ressources de la machine (le processeur, la mémoire, le disque, etc).

Objectifs

- Augmenter les performances d'une application.
- Améliorer l'utilisation des ressources d'une machine.

Définition d'un système adaptable

Définition (domaine d'automatisme)

- Un système adaptable est un système capable d'ajuster les conditions de son exécution en fonction de ses besoins ou de ceux d'un autre système.

Fonctionnalité

- L'adaptation d'un système informatique peut consister à optimiser l'utilisation de l'ensemble des ressources de la machine (le processeur, la mémoire, le disque, etc).

Objectifs

- Augmenter les performances d'une application.
- Améliorer l'utilisation des ressources d'une machine.

Définition d'un système adaptable

Définition (domaine d'automatisme)

- Un système adaptable est un système capable d'ajuster les conditions de son exécution en fonction de ses besoins ou de ceux d'un autre système.

Fonctionnalité

- L'adaptation d'un système informatique peut consister à optimiser l'utilisation de l'ensemble des ressources de la machine (le processeur, la mémoire, le disque, etc).

Objectifs

- Augmenter les performances d'une application.
- Améliorer l'utilisation des ressources d'une machine.

Composants d'un système adaptable

Éléments du système

- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel

Système de contrôle

Composants d'un système adaptable

Éléments du système

- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel

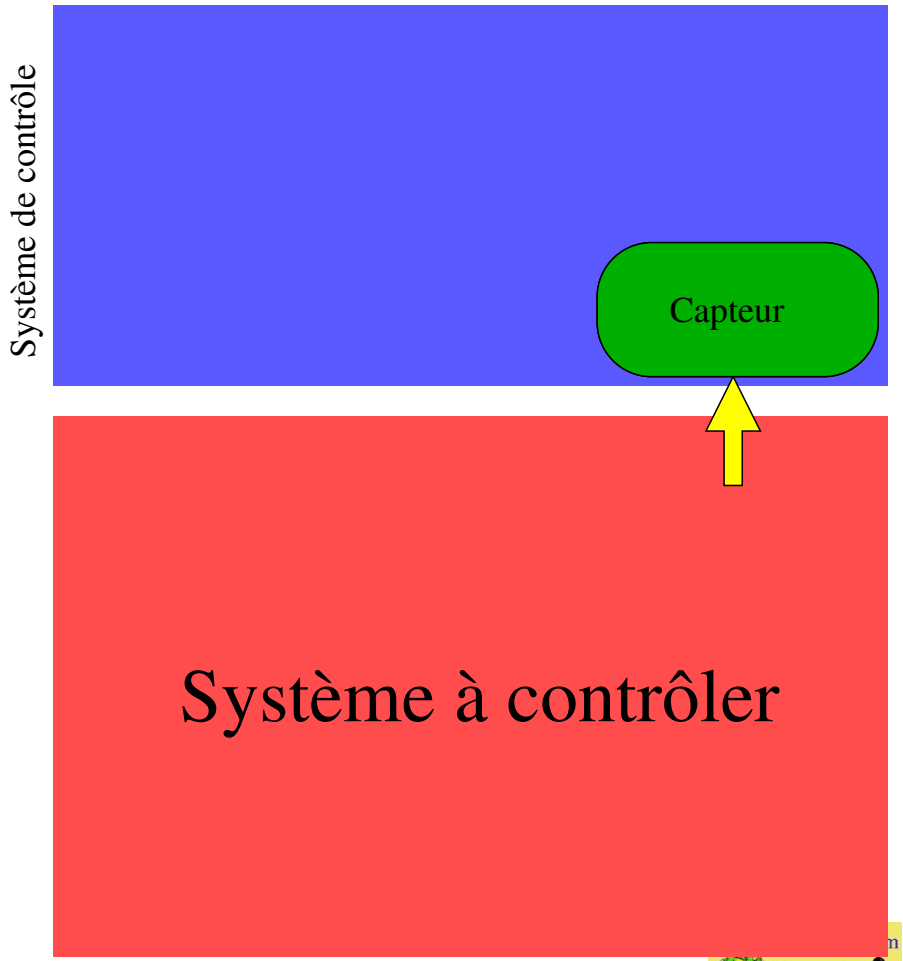
Système de contrôle

Système à contrôler

Composants d'un système adaptable

Éléments du système

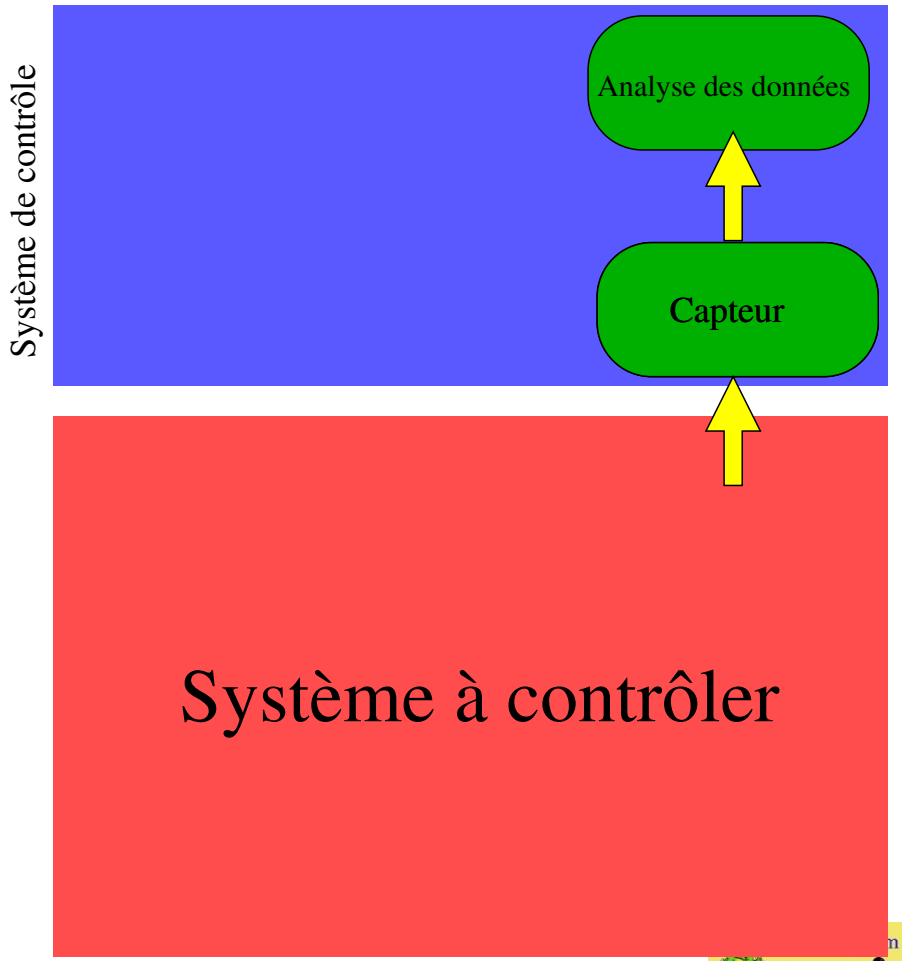
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

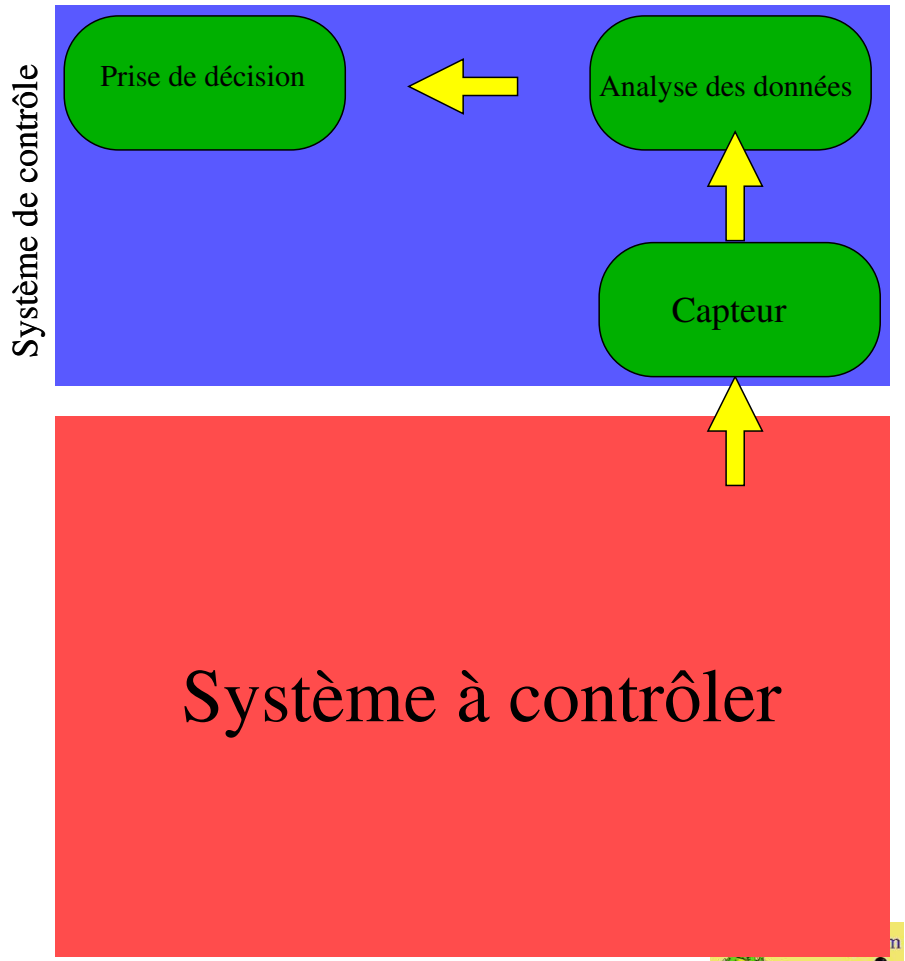
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

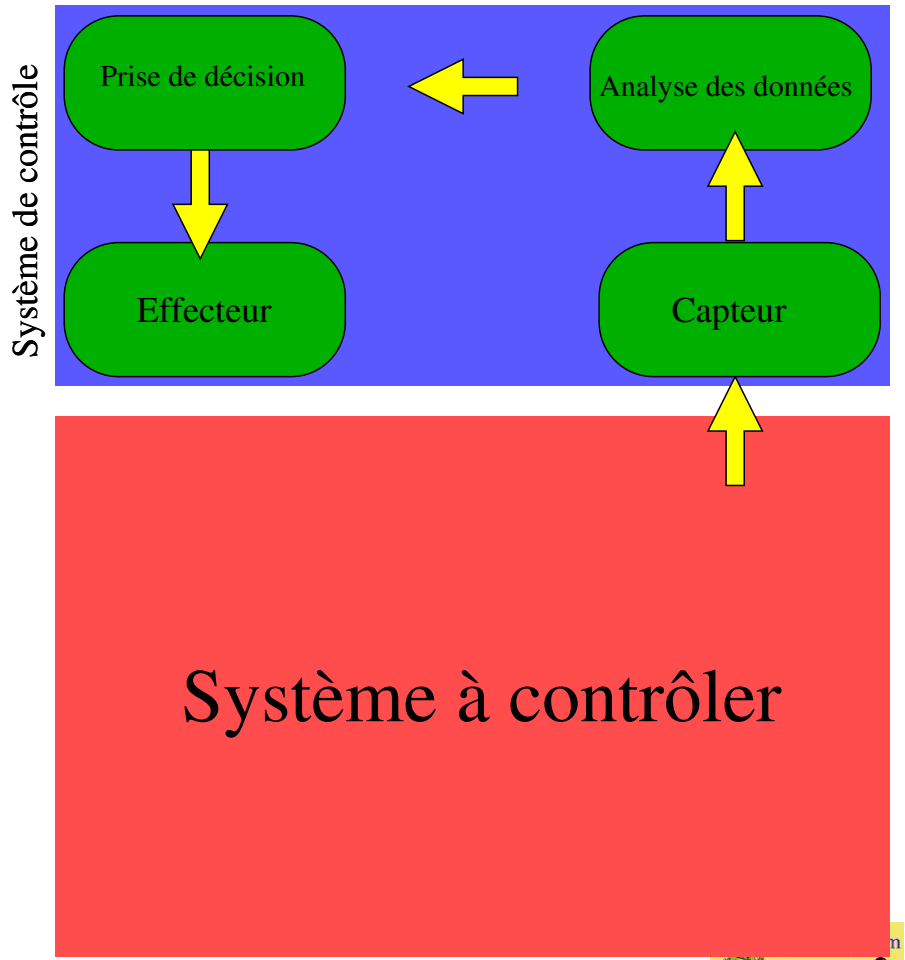
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

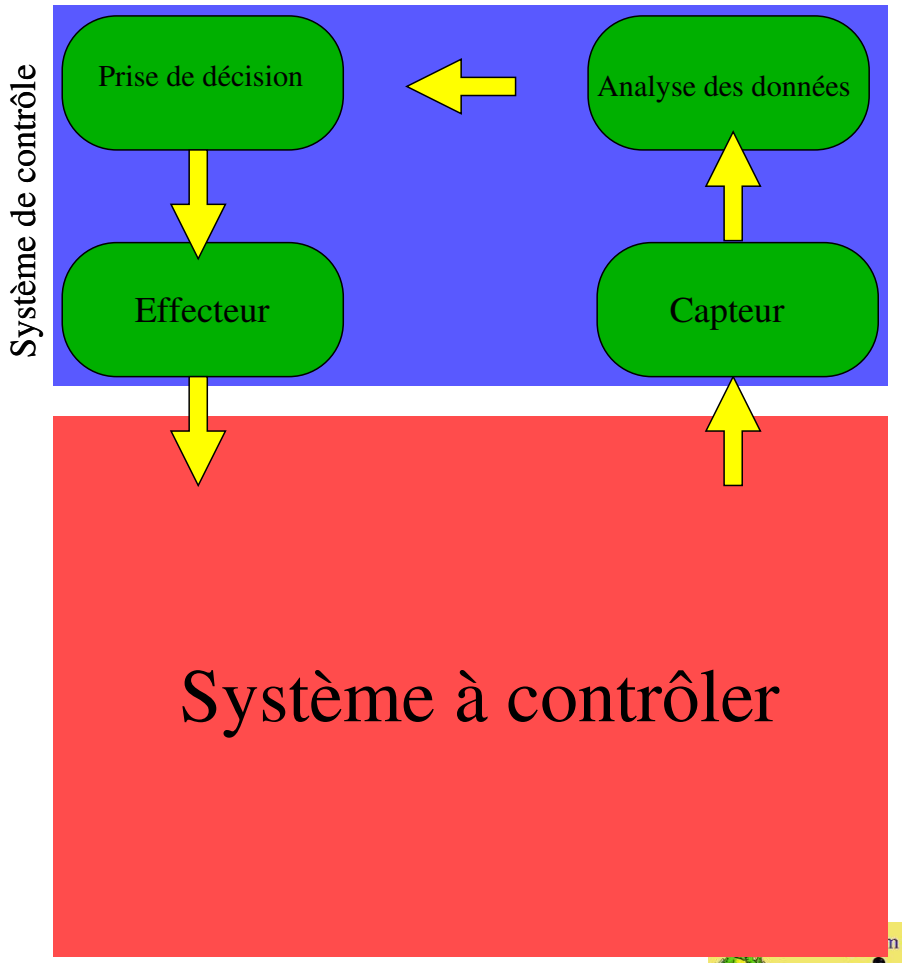
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

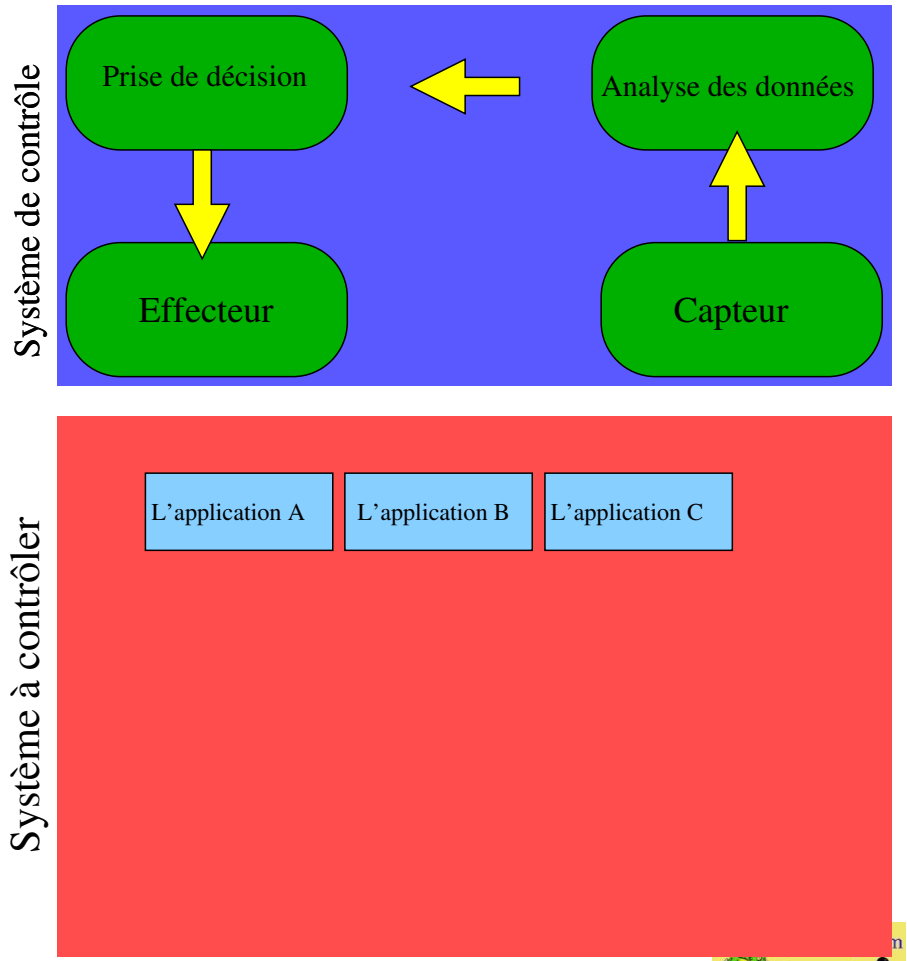
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

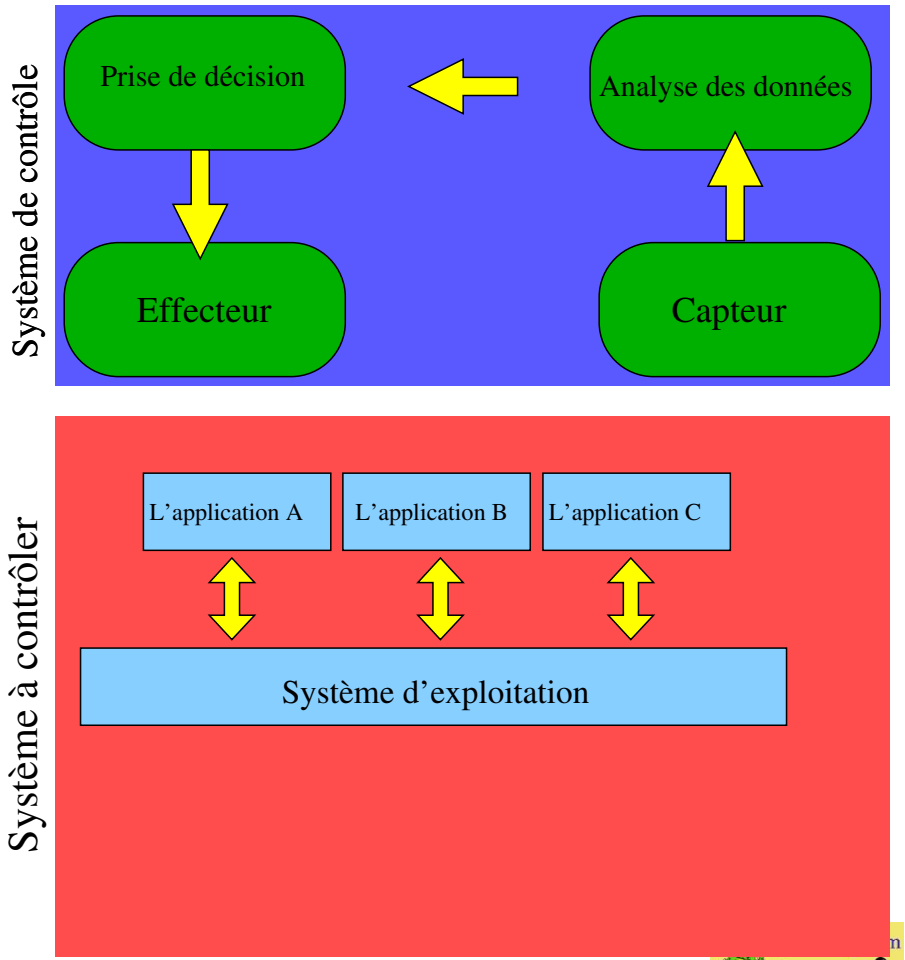
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

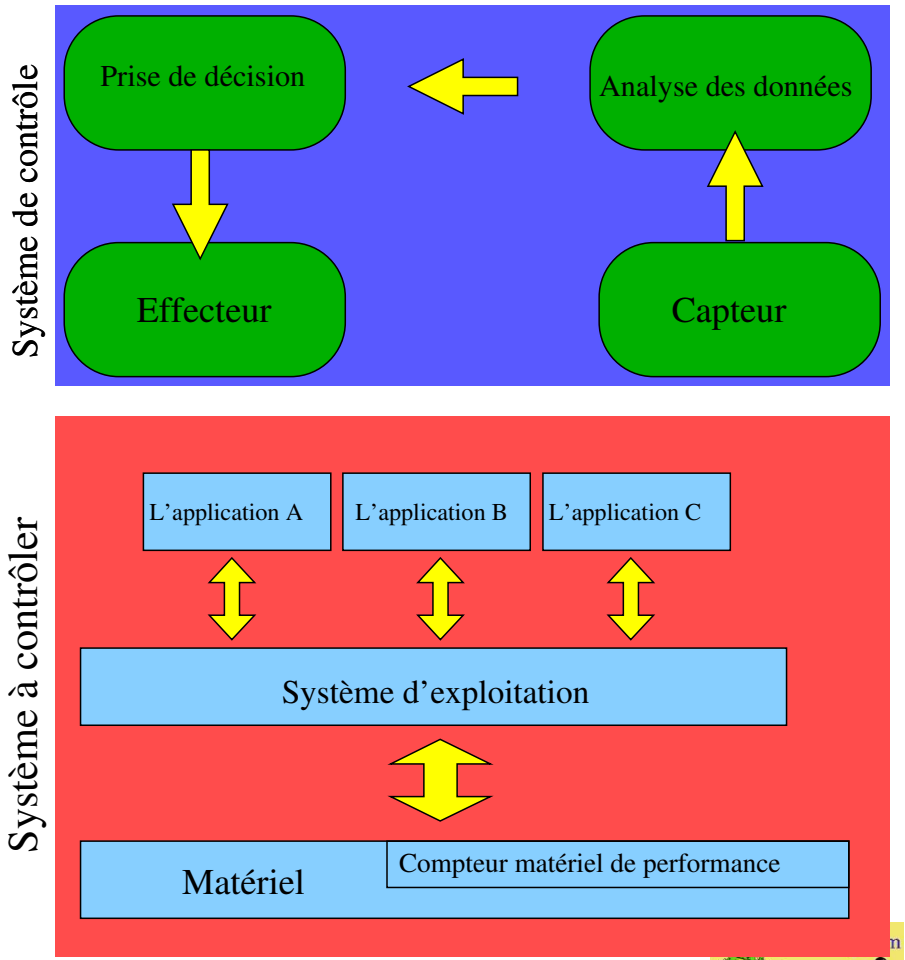
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

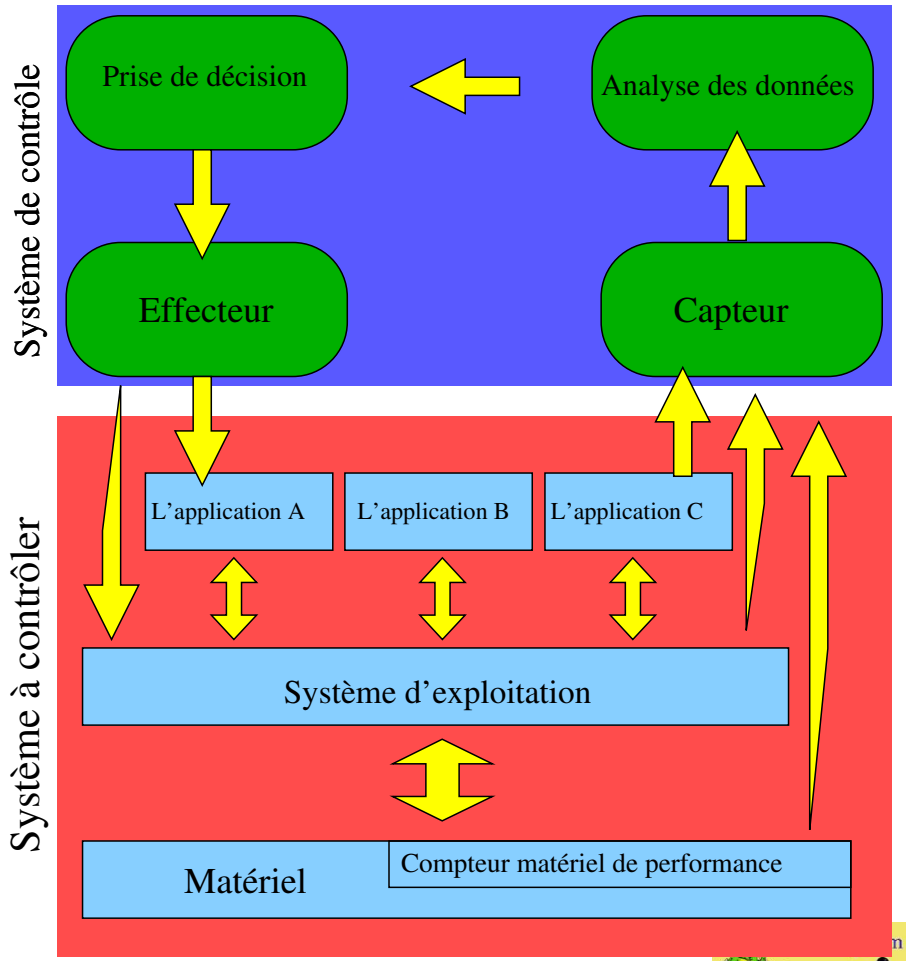
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Composants d'un système adaptable

Éléments du système

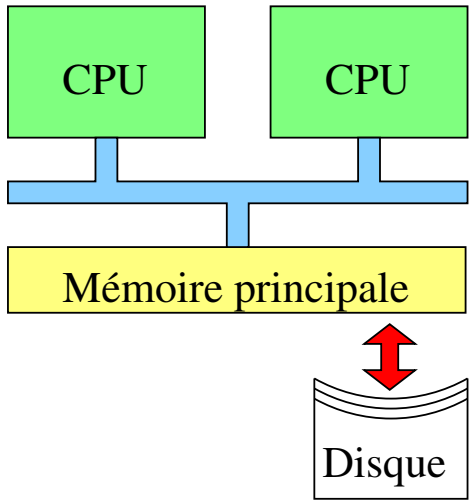
- système de contrôle
  - capteur
  - analyse des données
  - prise de décision
  - effecteur
- système à contrôler
  - applications
  - système d'exploitation
  - matériel



Exemple d'un système adaptable

Exemple d'un système adaptable

- Objectif : traiter le problème de contention mémoire sur les multiprocesseurs.
- Proposition : minimiser les échanges de données entre la mémoire principale et le disque [NP02].



Vérification des conditions nécessaires à la réalisation du système

Méthodologie

- vérification du problème de contention mémoire.
- mesure du rendement de la machine (accélération).
- identification de l'événement matériel capable d'établir le rapport entre l'utilisation mémoire et l'accélération.

Observation des applications

- SPEC2000 - basé en appl. réelles.
- NPB NAS - appl. scientifiques parallèles (OpenMP).
- STREAM - activité mémoire.
- Netperf - activité réseau.
- Bonnie - activité disque.

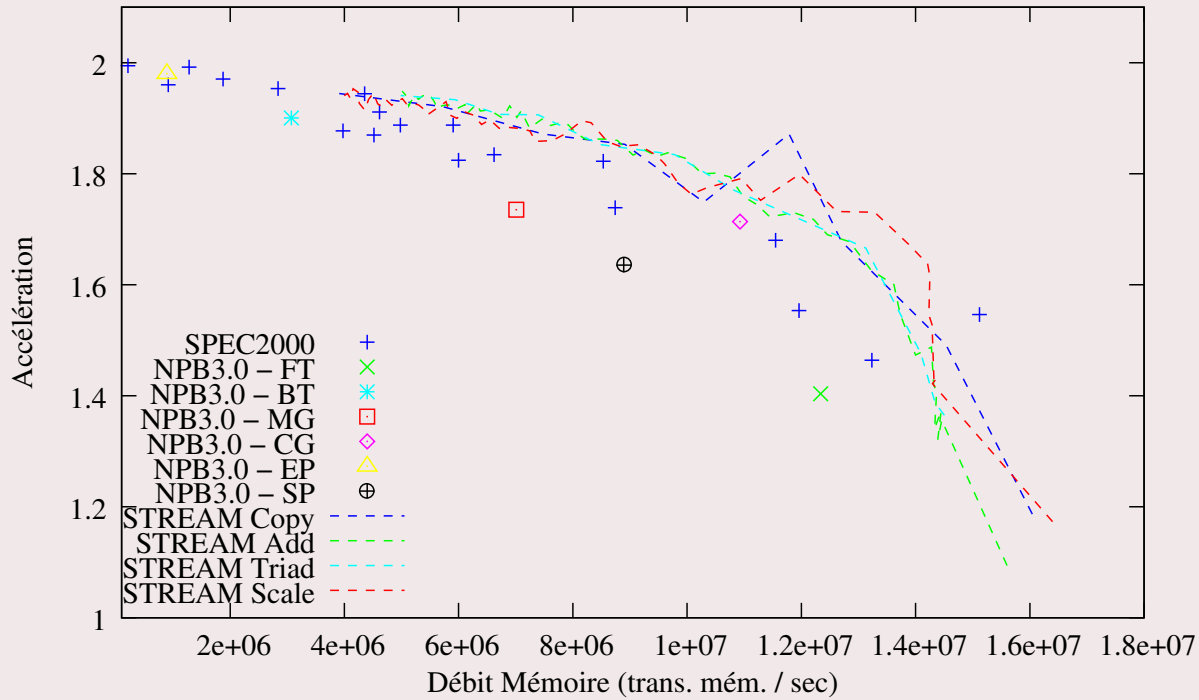
Plate-forme de tests

1. Biproc. Pentium II - 450 MHz - Debian - 2.4.20 - perfctr 2.3
2. Biproc. Pentium 4 - 1700 MHz - Mandrake - 2.4.18 - perfctr 2.3

Estimation de l'accélération en fonction de l'activité mémoire

- STREAM - boucle modifiée (nop).
- SPEC - appl. concurrente.
- NAS - appl. parallèle.
- Granularité élevée.
- Ressource observée : bus mémoire.
- Classement en 3 zones :
  - non-saturée
  - intermédiaire
  - saturée

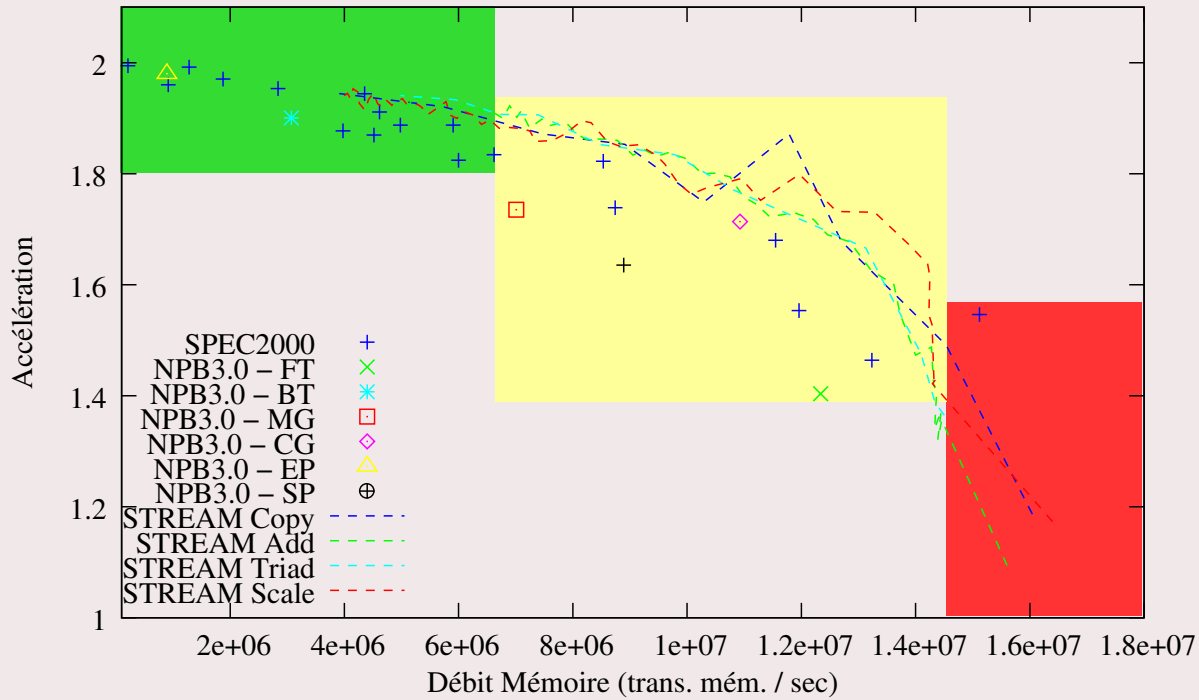
Biprocesseur Pentium II



Estimation de l'accélération en fonction de l'activité mémoire

- STREAM - boucle modifiée (nop).
- SPEC - appl. concurrente.
- NAS - appl. parallèle.
- Granularité élevée.
- Ressource observée : bus mémoire.
- Classement en 3 zones :
  - non-saturée
  - intermédiaire
  - saturée

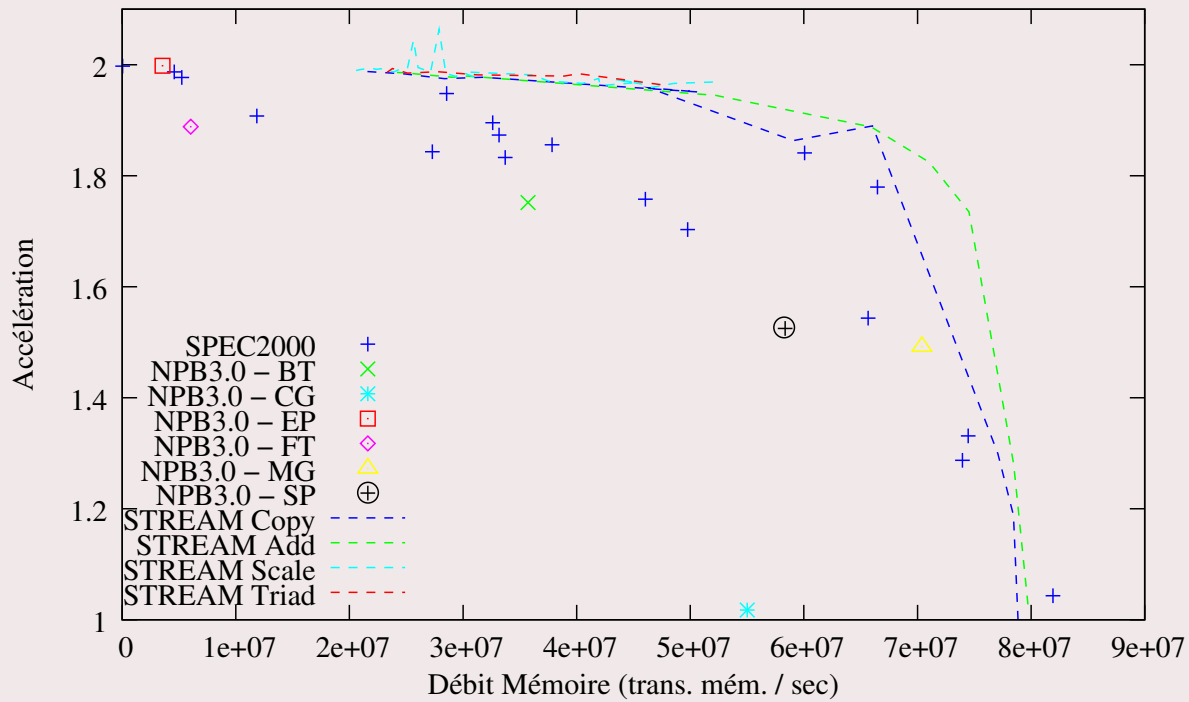
Biprocesseur Pentium II



Estimation de l'accélération en fonction de l'activité mémoire

- Ressource observée : bus mémoire.
- Classement en 2 zones :
  - non-saturée
  - intermédiaire / saturée
- **NAS CG**
  - inhibition des verrous.
  - absence de trafic de cohérence de cache.

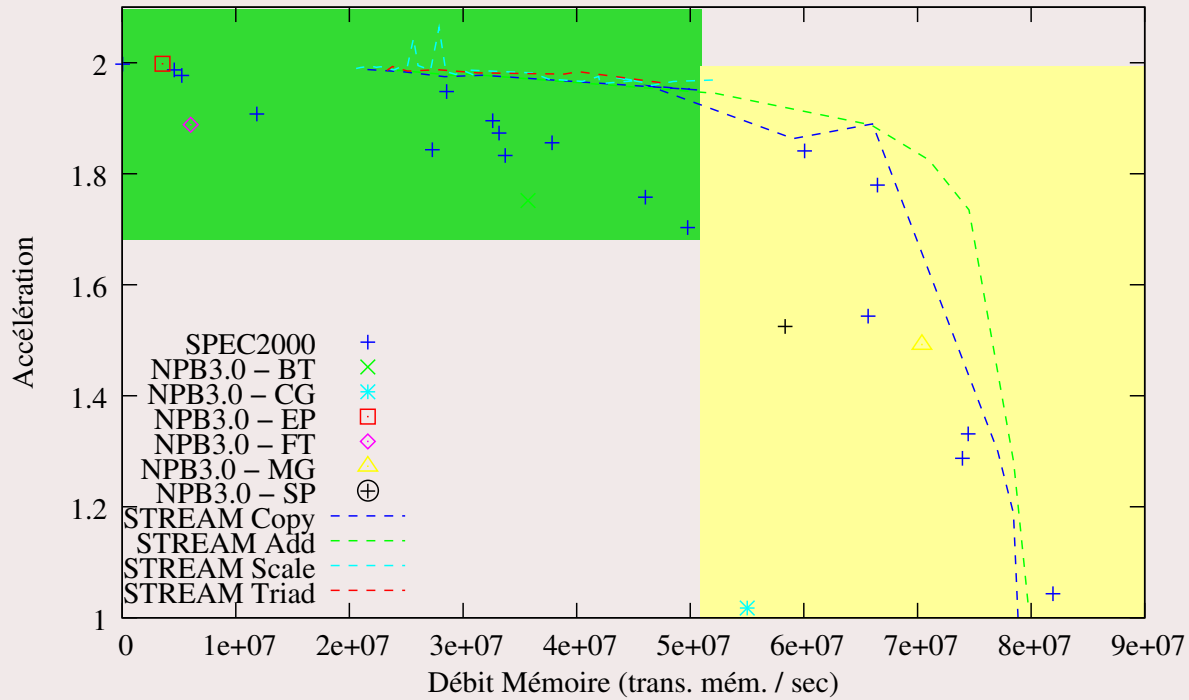
Biprocesseur Pentium 4



Estimation de l'accélération en fonction de l'activité mémoire

- Ressource observée : bus mémoire.
- Classement en 2 zones :
  - non-saturée
  - intermédiaire / saturée
- **NAS CG**
  - inhibition des verrous.
  - absence de trafic de cohérence de cache.

Biprocesseur Pentium 4



Étude préalable à réalisation du système

Bilan

- **Compteurs matériel de performances :**
  - La plupart des architectures processeurs étudiées possèdent entre 2 et 4 compteurs.
  - Le nombre d'événements et le format varient d'une architecture à une autre.
  - La complexité d'utilisation dépend du format des registres et de leurs relations avec les registres de sélection.
- **Bibliothèque d'accès aux compteurs :** PAPI (couche dépendante **perfctr**).
- **Systèmes adaptables :** permettre le contrôle dynamique du système.

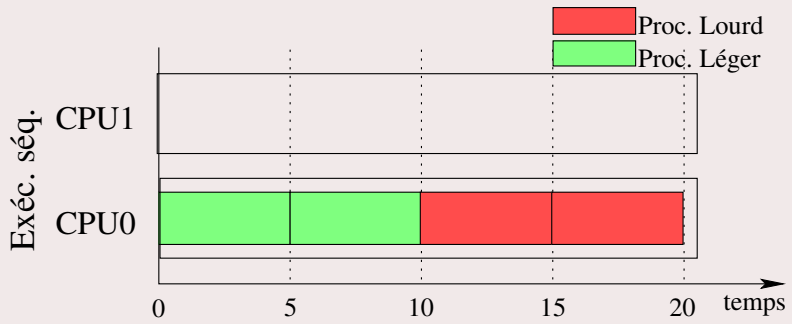
Étude préalable à réalisation du système

Bilan (cont.)

- Si l'ensemble des applications ont des forts besoins en débit mémoire leurs performances sont dégradées.
- Si l'ensemble des applications ont des faibles besoins en débit mémoire leurs performances sont proches du maximum.
- Le mécanisme utilisé pour établir la relation entre l'activité mémoire et l'accélération fonctionne sur quelques architectures.
- La grande majorité des applications observées ont des comportements stables.
- L'impact des activités d'E/S s'est montré faible.
- Les principales conditions pour la réalisation d'un système de contrôle d'exécution sont réunies.

DRAC : Système de contrôle d'exécution

DRAC : Principe du système



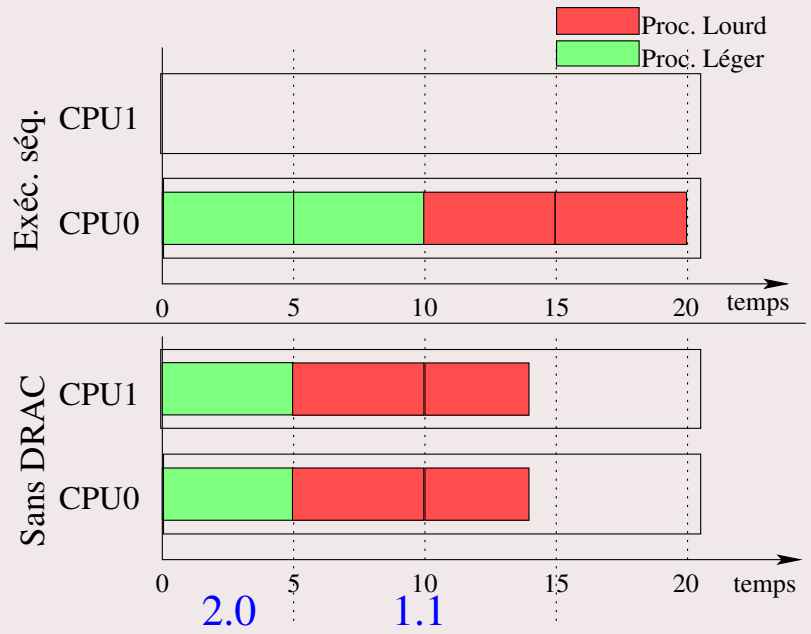
Caractéristiques du système DRAC

- Extérieur aux applications.
- Niveau utilisateur.

Objectif

- Améliorer le rendement des multiprocesseurs en ordonnançant les applications en fonction de leurs besoins en débit mémoire [Pil02] [PRD04].

DRAC : Principe du système



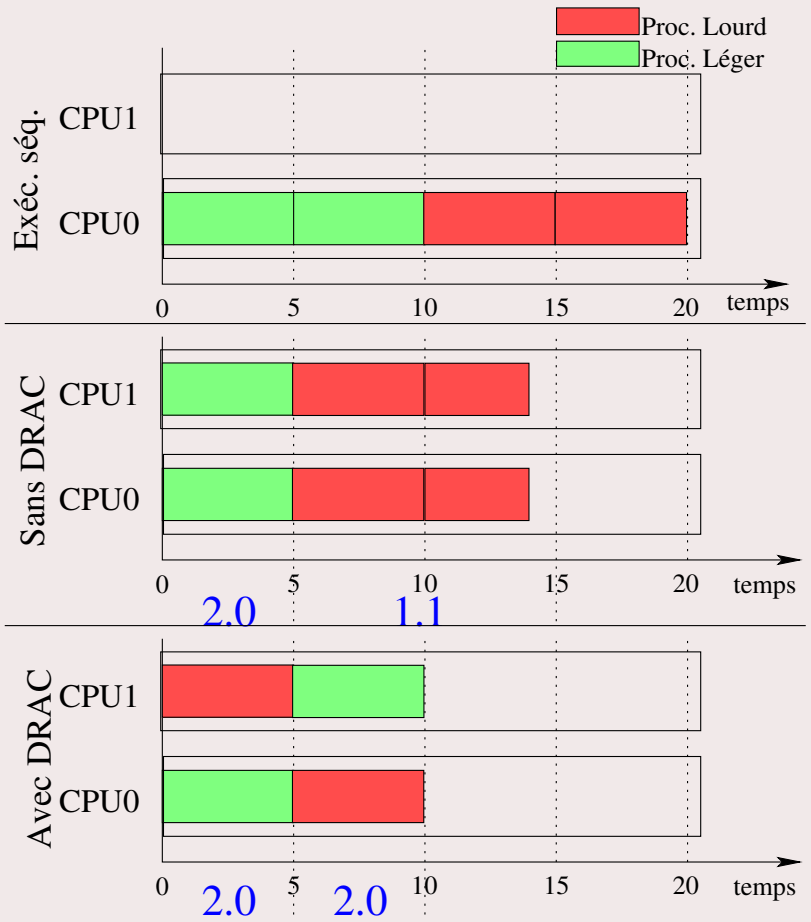
Caractéristiques du système DRAC

- Extérieur aux applications.
- Niveau utilisateur.

Objectif

- Améliorer le rendement des multiprocesseurs en ordonnançant les applications en fonction de leurs besoins en débit mémoire [Pil02] [PRD04].

DRAC : Principe du système



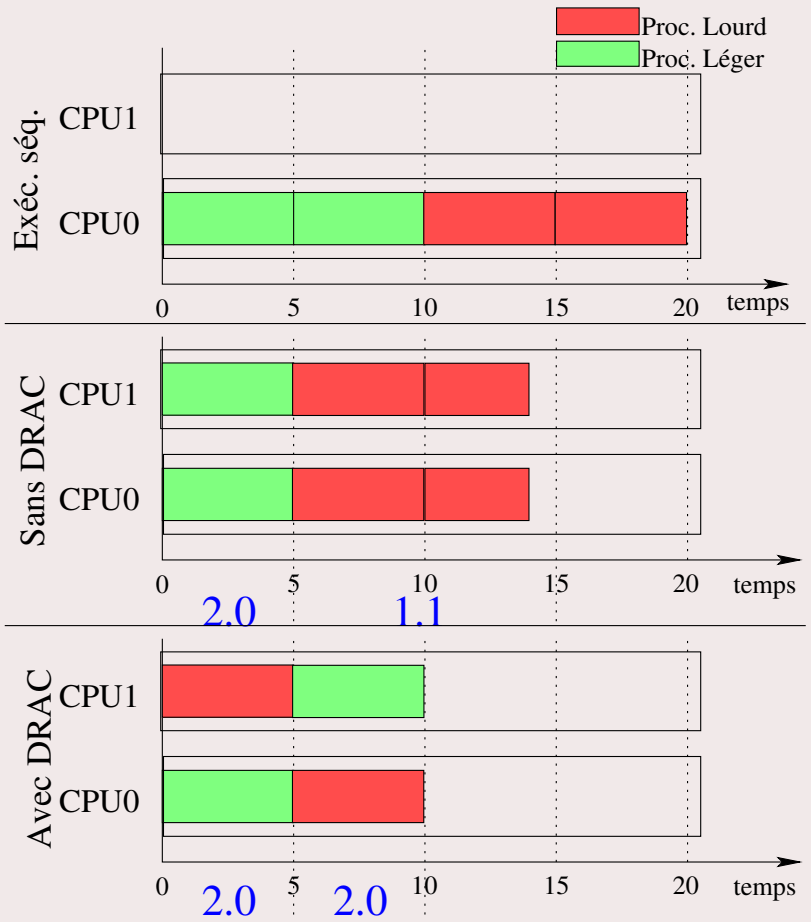
Caractéristiques du système DRAC

- Extérieur aux applications.
- Niveau utilisateur.

Objectif

- Améliorer le rendement des multiprocesseurs en ordonnançant les applications en fonction de leurs besoins en débit mémoire [Pil02] [PRD04].

DRAC : Principe du système



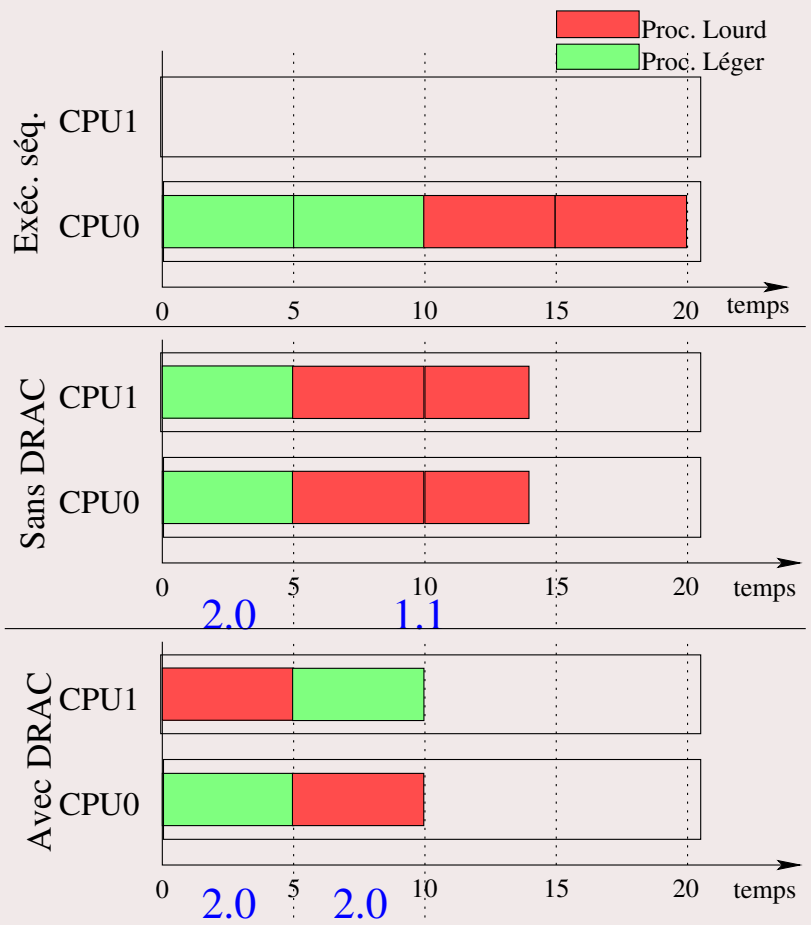
Caractéristiques du système DRAC

- Extérieur aux applications.
- Niveau utilisateur.

Objectif

- Améliorer le rendement des multiprocesseurs en ordonnançant les applications en fonction de leurs besoins en débit mémoire [Pil02] [PRD04].

DRAC : Principe du système



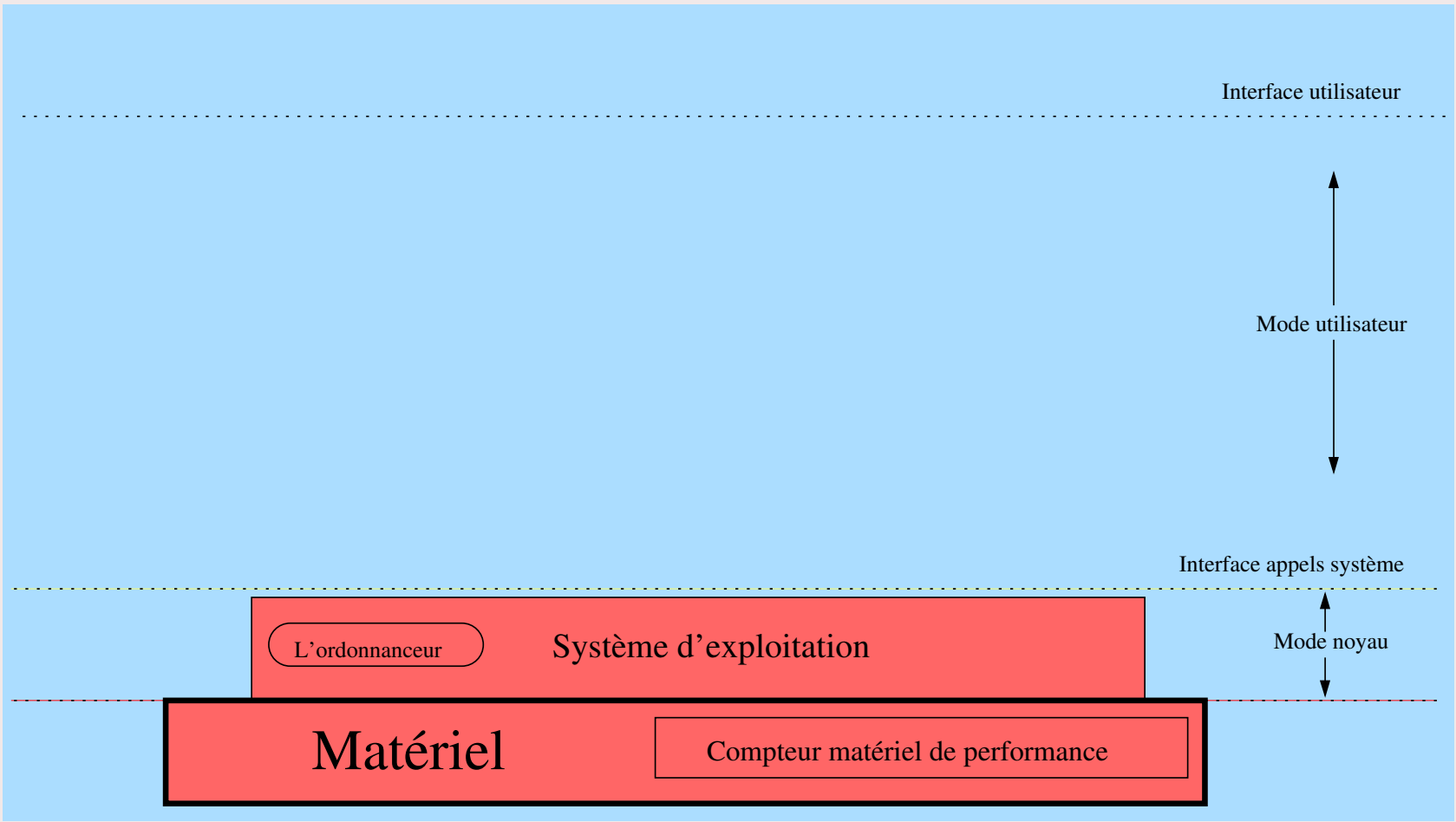
Caractéristiques du système DRAC

- Extérieur aux applications.
- Niveau utilisateur.

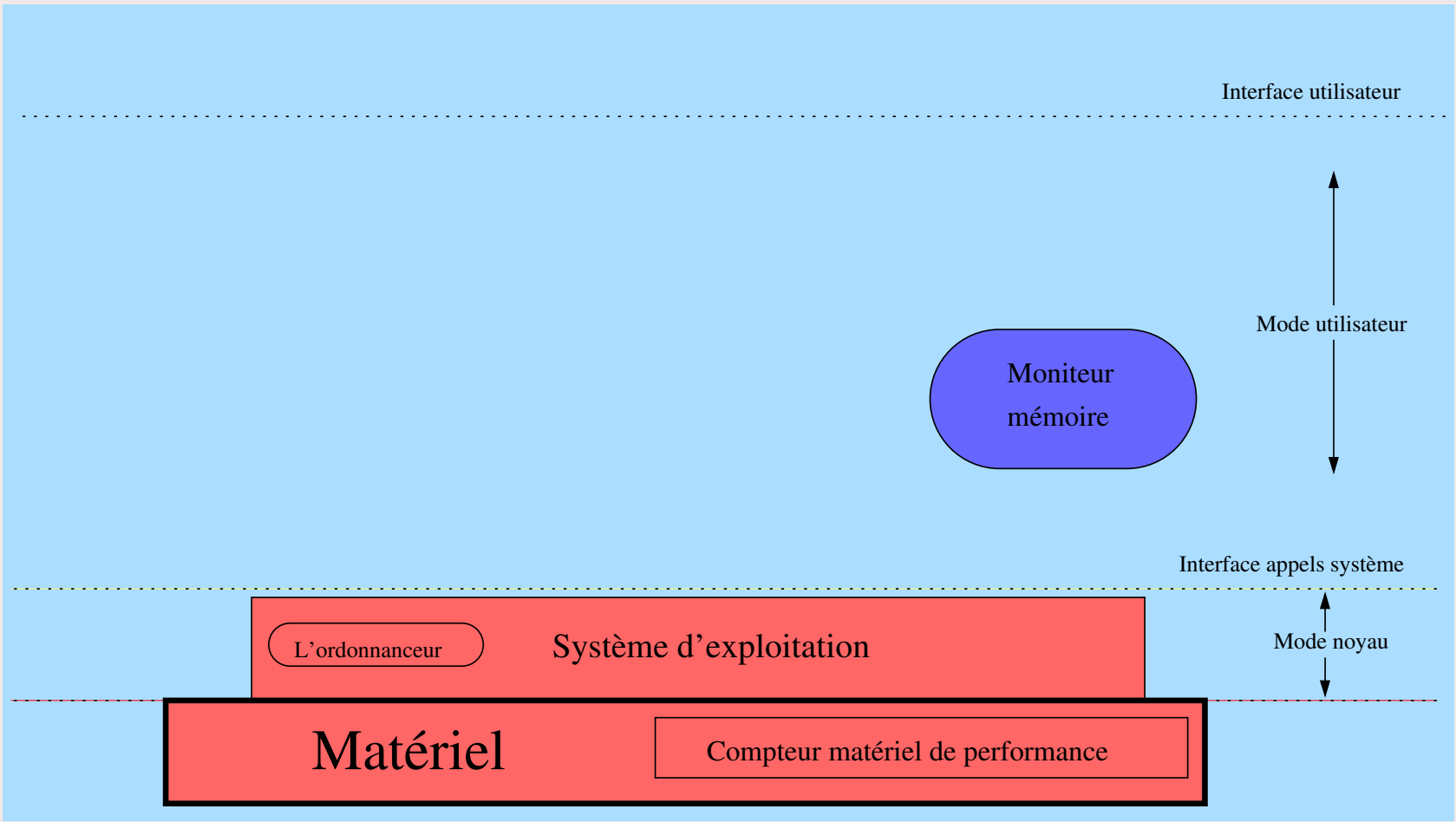
Objectif

- Améliorer le rendement des multiprocesseurs en ordonnançant les applications en fonction de leurs besoins en débit mémoire [Pil02] [PRD04].

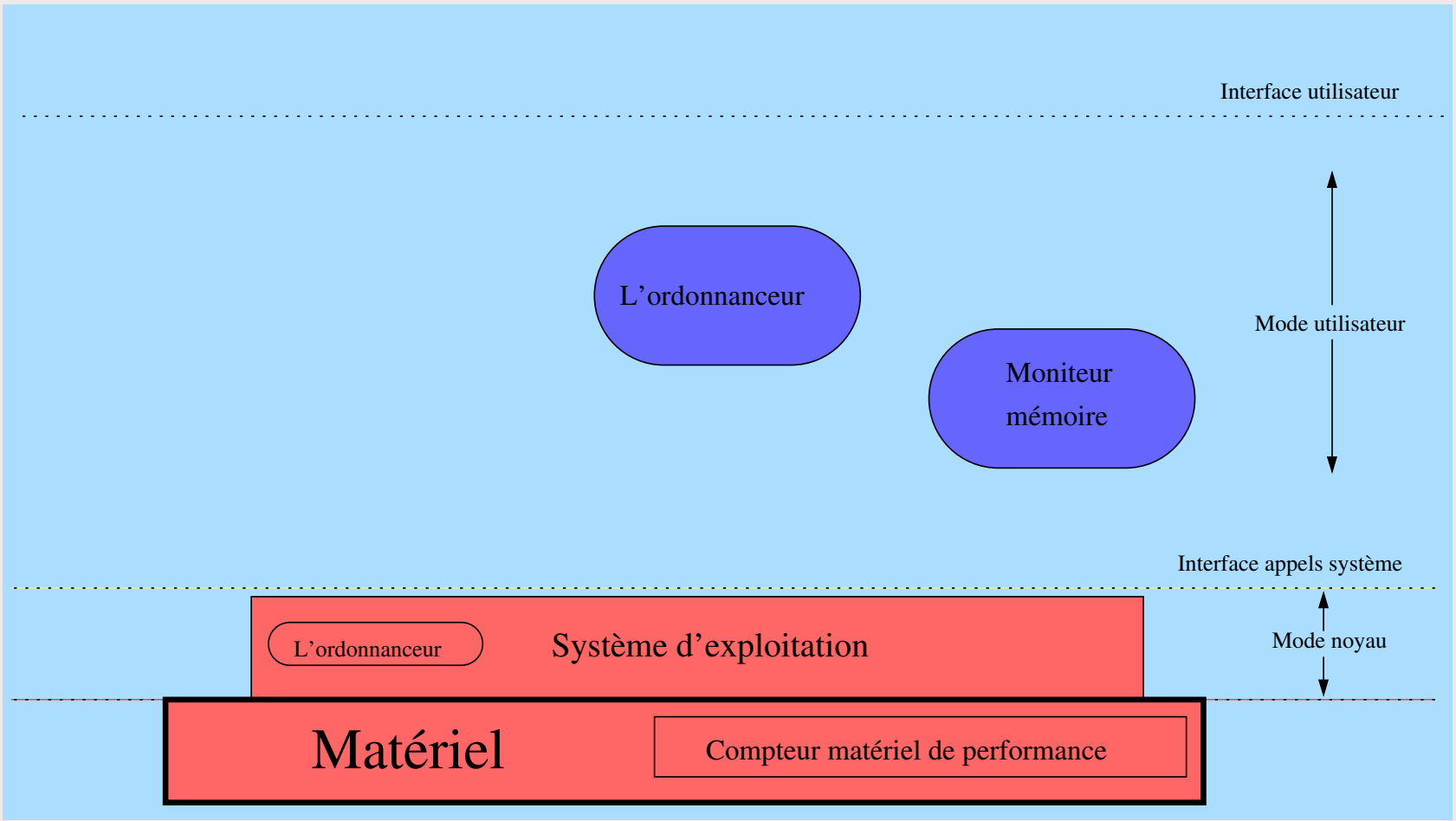
Architecture du système DRAC



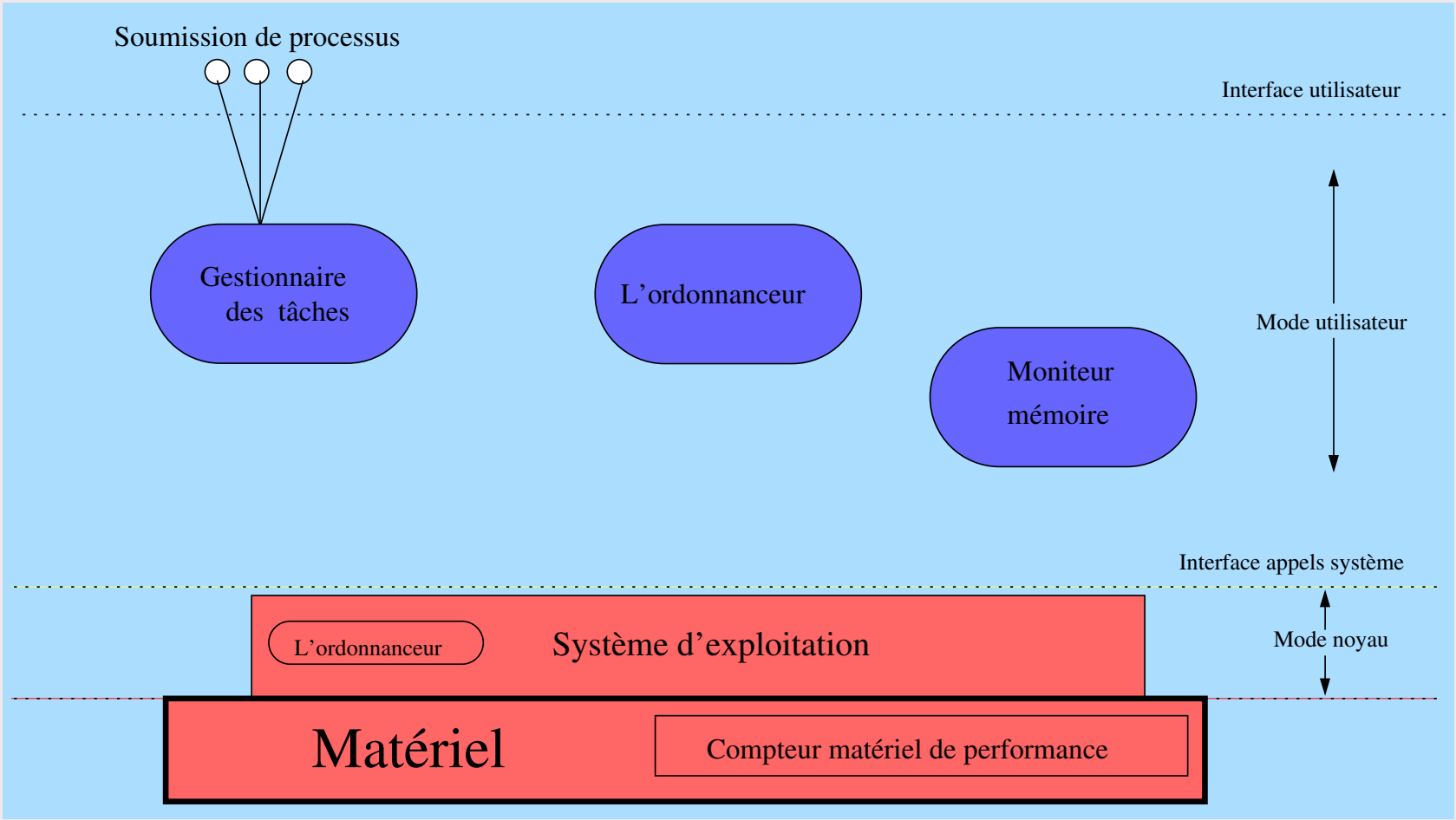
Architecture du système DRAC



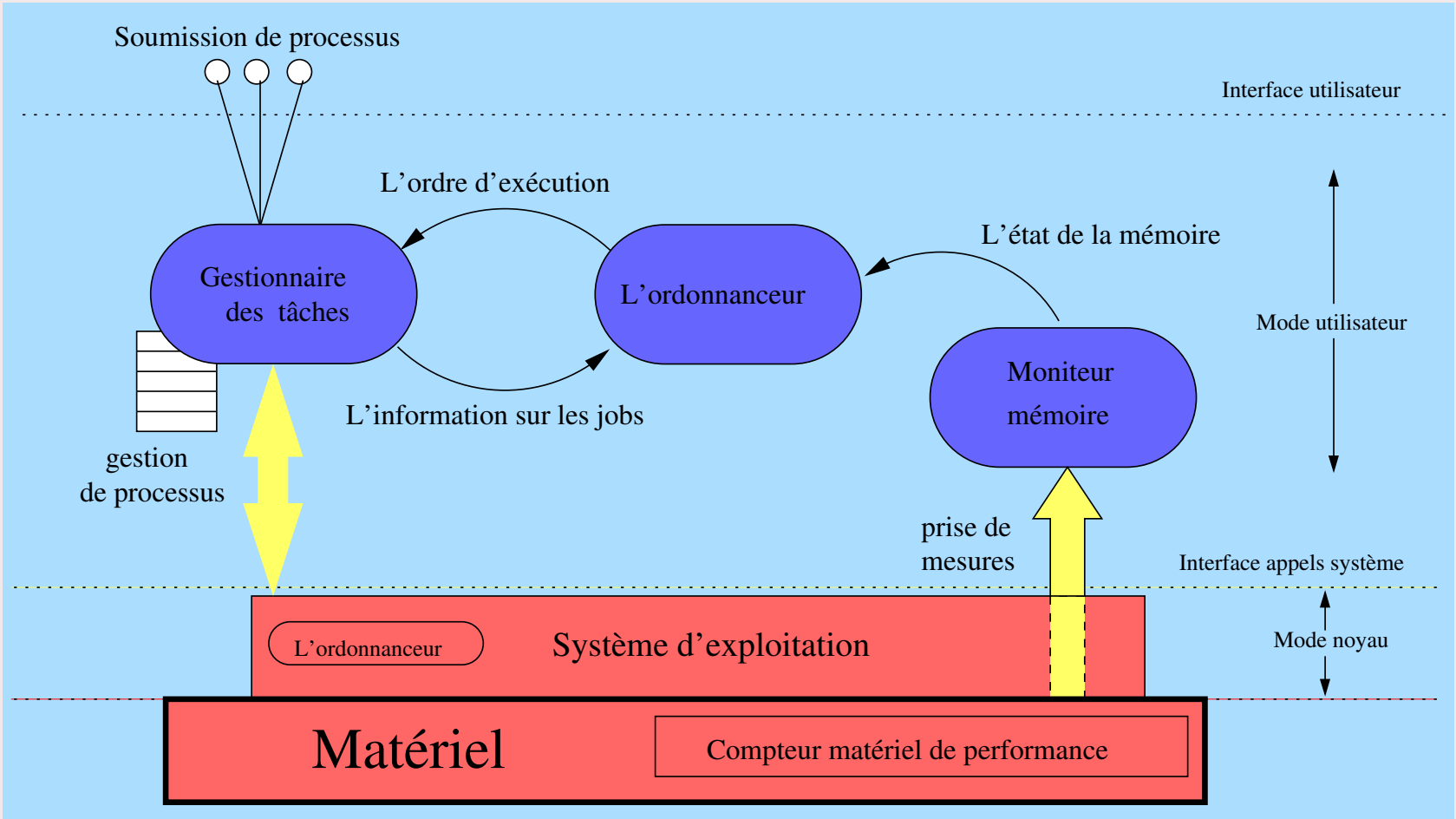
Architecture du système DRAC



Architecture du système DRAC



Architecture du système DRAC



DRAC : Système de contrôle d'exécution

Moniteur mémoire

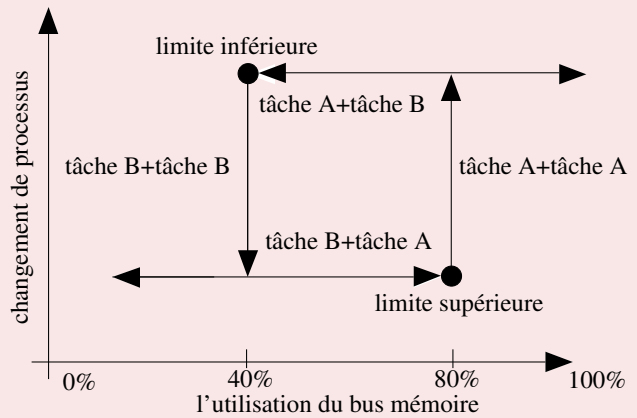
- Observation du niveau d'utilisation de la hiérarchie mémoire.
- Composants :
  - **capteur :** collecte des activités du bus mémoire par échantillonnage.
  - **traitant :** transformation des informations brutes extraites des compteurs en données utilisée par l'analyseur.
  - **analyseur :** filtrage les données et transmission de l'état d'utilisation du bus mémoire.

DRAC : Système de contrôle d'exécution

L'ordonnanceur

- contrôle de l'exécution des processus soumis à DRAC.
- Aucun lien avec l'ordonnanceur du système d'exploitation.
- **politique d'ordonnancement :** heuristique que réévalue régulièrement les tâches à mettre ensemble.
- **algorithme de contrôle :** Robustesse.
  - limite inférieure (sous-utilisation du bus mémoire).
  - limite supérieure (saturation du bus mémoire).

Algorithme de contrôle



DRAC : Système de contrôle d'exécution

Gestionnaire des tâches

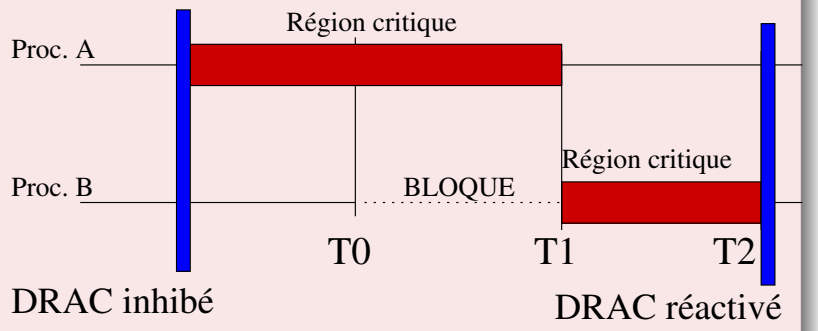
- contrôler la gestion effective des processus du système DRAC.
  - soumission
  - retrait
  - démarrage
  - suspension
  - synchronisation
- **DRACsub : gestionnaire de soumission de processus :**
  - attente de soumission
  - contrôle
  - lancement du processus
  - mise en attente

DRAC : Système de contrôle d'exécution

Gestionnaire des tâches

- **Contrôle de processus :**
  - retrait / démarrage / suspension
  - synchronisation
    - exclusion mutuelle - garantir la non-suspension d'un processus dans une région critique.
    - barrières - Éviter que l'ordonnanceur DRAC endorme un processus devant libérer une barrière.

Scénario : Exclusion mutuelle

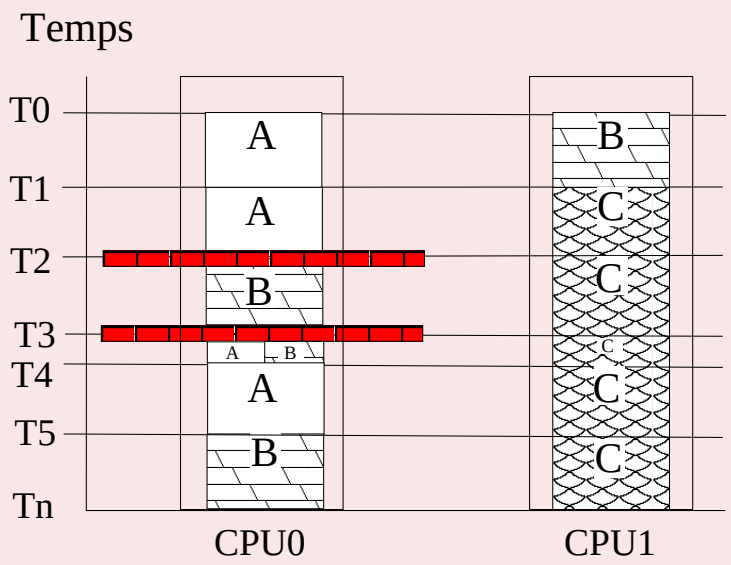


DRAC : Système de contrôle d'exécution

Gestionnaire des tâches

- **Contrôle de processus :**
  - retrait / démarrage / suspension
  - synchronisation
    - exclusion mutuelle - garantir la non-suspension d'un processus dans une région critique.
    - barrières - Éviter que l'ordonnanceur DRAC endorme un processus devant libérer une barrière.

Scénario : barrière



Modélisation et Évaluations

Étude de cas : Biprocesseur

Hypothèses

1. Un bus mémoire est partagé par tous les processeurs.
2. Les processus en cours d'exécution ont une seule ressource commune, l'accès au bus mémoire.
3. Le débit d'accès mémoire des processus est constant par morceaux.

Définitions :

1. n le nombre total de processus à exécuter.
2. $\alpha(i) = [0, 1]$ est la charge d'un processus P_i.
3. $P^{(0)}$ est un processus léger en débit mémoire et $P^{(1)}$ est un processus lourd en débit mémoire.
4. L'accélération de $(P^{(0)}|P^{(0)})$ est $\delta^{(0)} = 2$ et de $(P^{(1)}|P^{(1)})$ est $\delta^{(1)} = \frac{2}{max(1,2\alpha(1))}$.
5. β represente la répartion de processus lourd/léger $= \frac{\text{Nb de } P^{(1)}}{n}$.

Étude de cas : Pire ordonnancement

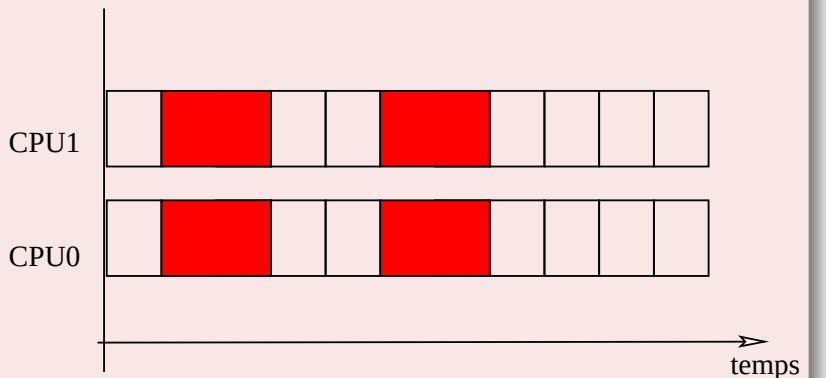
Définition

- Il fait en sorte que la charge sur le bus mémoire soit la plus grande possible pendant l'exécution de l'ensemble des processus.

Formule

$$T_{pire} = \frac{n}{2}(1 + \beta(\frac{2}{\delta(1)} - 1))$$

Exemple

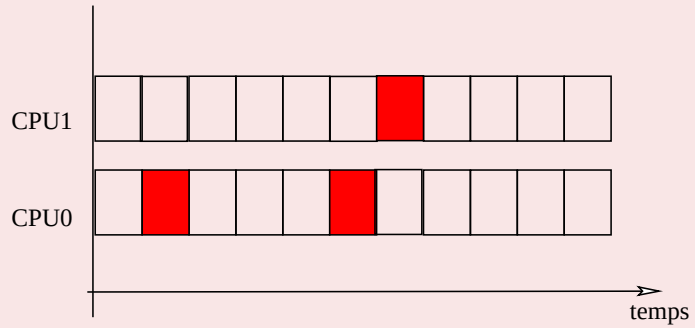


Étude de cas : Meilleur ordonnancement

Définition

- Il fait en sorte que la charge sur le bus mémoire soit la plus petite possible pendant l'exécution de l'ensemble des processus.

Exemple1



Formule

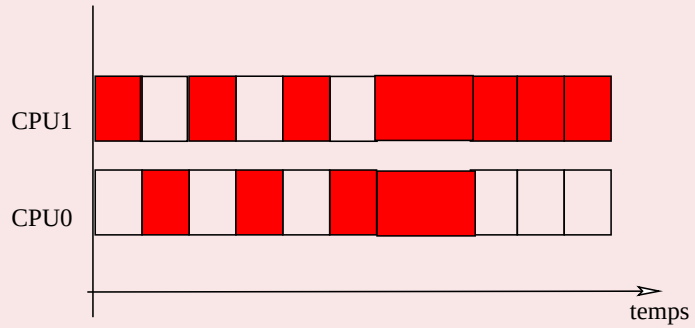
- Si $0\% \leq \beta \leq 50\%$: $T_{meilleur} = \frac{n}{2}$
- Si $50\% \leq \beta \leq 100\%$: $T_{meilleur} = n(1 - \beta + \frac{1}{\delta(1)} \cdot (2 \cdot \beta - 1))$

Étude de cas : Meilleur ordonnancement

Définition

- Il fait en sorte que la charge sur le bus mémoire soit la plus petite possible pendant l'exécution de l'ensemble des processus.

Exemple2



Formule

- Si $0\% \leq \beta \leq 50\% : T_{meilleur} = \frac{n}{2}$
- Si $50\% \leq \beta \leq 100\% : T_{meilleur} = n(1 - \beta + \frac{1}{\delta(1)} \cdot (2 \cdot \beta - 1))$

Étude de cas : Ordonnancement moyen

Définition

- Il est basé sur la probabilité d'occurrence de chaque *groupe d'exécution* possible.

Groupe d'exécution

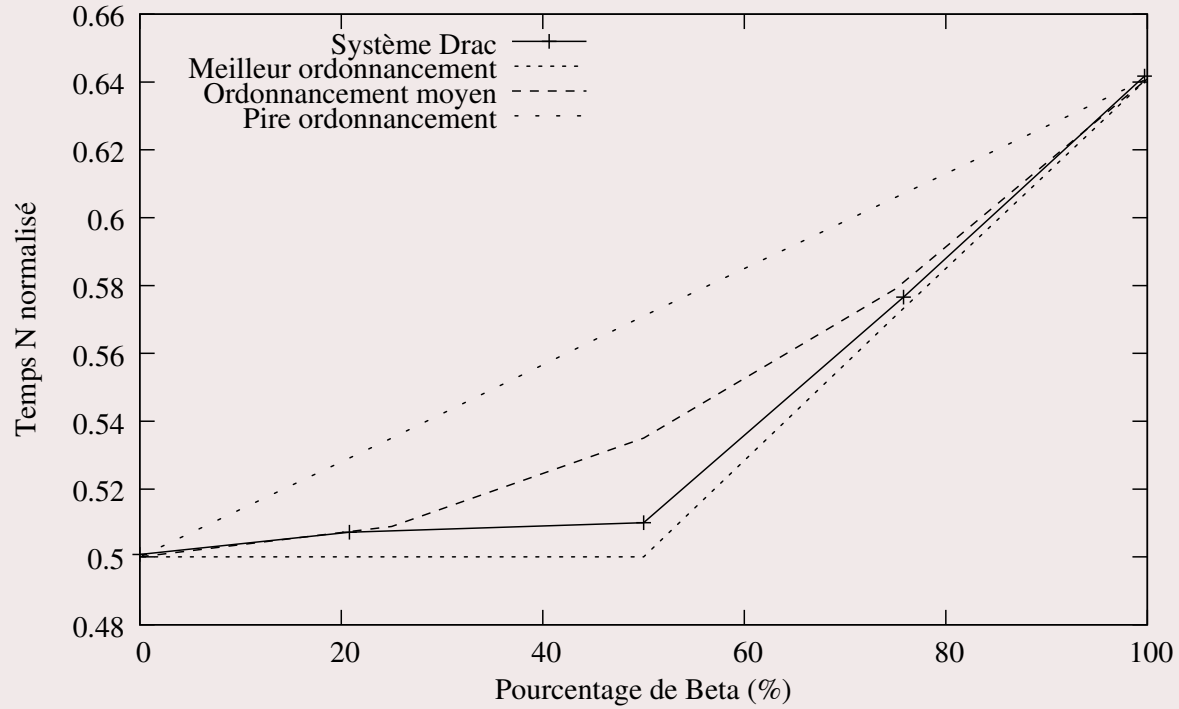
$$(P^{(0)}|P^{(0)})$$
$$(P^{(0)}|P^{(1)})$$
$$(P^{(1)}|P^{(0)})$$
$$(P^{(1)}|P^{(1)})$$

Formule

$$T_{moyen} = \frac{n}{2}(1 + \beta^2(\frac{2}{\delta^{(1)}} - 1))$$

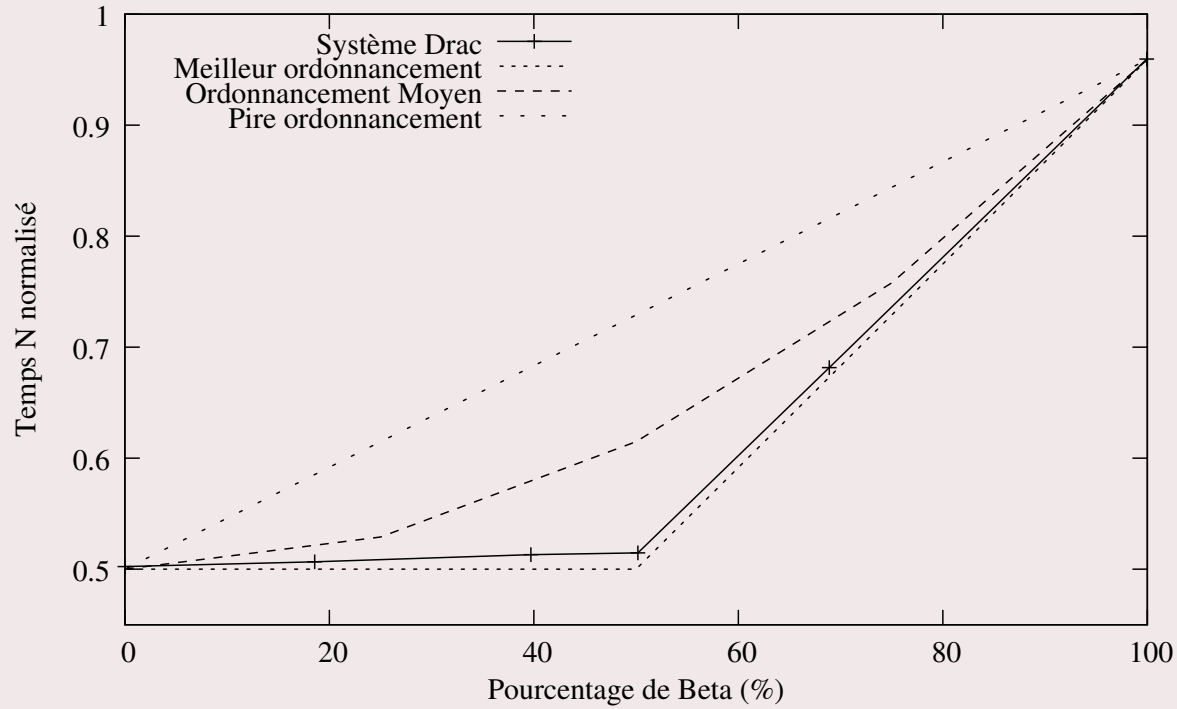
Biprocesseur Pentium II (Exécution en concurrence)

- SPEC ART (1.54)
- SPEC EON (1.99)
- 5% plus rapide que l'ordonnancement moyen
- 12% plus rapide que le pire ordonnancement



Biprocesseur Pentium 4 (Exécution en concurrence)

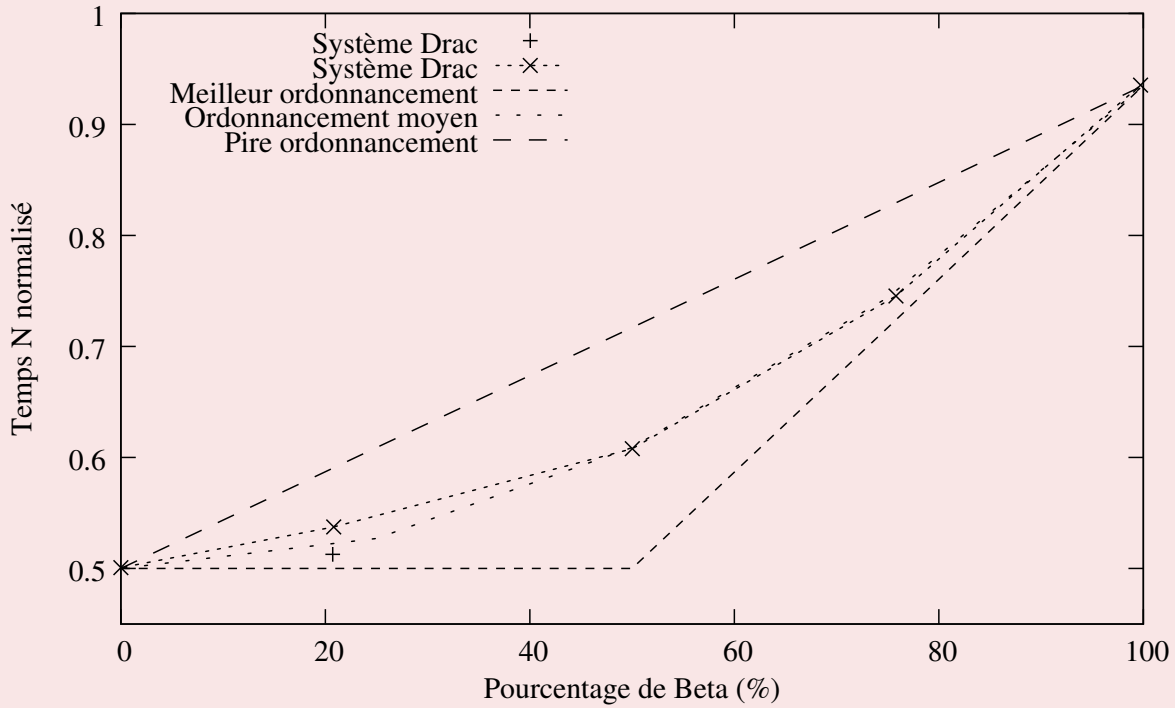
- SPEC ART (1.04)
- SPEC EON (1.99)
- 20% plus rapide que l'ordonnancement moyen
- 41% plus rapide que le pire ordonnancement



Biprocesseur Pentium 4 (Exécution parallèle - OpenMP)

- NAS CG (1.07)
- NAS EP (2.00)
- 7.5% plus lent que l'ordonnancement moyen.
- 15% plus rapide que le pire ordonnancement.
- Granularité trop fine (barrière-$\sim 100ms$).
- 2.6% plus rapide que l'ordonnancement moyen.

Résultats préliminaires



Modélisation et Évaluations

Bilan

- Le modèle présenté permet de calculer le temps d'exécution avec le ralentissement produit par l'utilisation de la hiérarchie mémoire.
- Les résultats sur les machines biprocesseurs montrent des gains de plus de 40% dans certains cas.
- Les performances du système dépendent de la granularité des applications.

Conclusion et travaux futurs

Conclusion

1. Proposition d'une solution pour le problème de la saturation mémoire sur les multiprocesseurs.
2. L'objectif visé est d'améliorer l'utilisation des ressources.
3. **DRAC :** système de contrôle d'exécution
   - détection de saturation ou de sous-utilisations.
   - modification de l'ordonnancement.
4. Les principales difficultés :
   - utilisation des compteurs matériels de performances
   - instabilité des bibliothèques d'accès aux compteurs
   - identification de l'événement adapté (s'il existe)

Travaux futurs

Court terme

1. Optimiser le module qui contrôle la synchronisation entre les processus.
2. Portage du système sur les architectures Itanium 

.
   - Vérifier le goulot d'étranglement mémoire.
   - Identifier l'événement adapté (établir le rapport entre l'accélération et l'utilisation mémoire).
   - Optimiser le code pour une architecture 64*bits*.

Moyen terme

1. Observer l'impact de la hiérarchie mémoire sur les processeurs multi-cœur.
2. Étendre cette étude au cadre de grappes et des grilles (contention sur le réseau - projet de Post-doc à l'UFRGS).

Questions ? ! ? ! / Questões ? ! ? !

Merci bien pour votre attention !

Obrigado pela atenção !

Questions ? ! ? ! / Questões ? ! ? !

Merci bien pour votre attention !

Obrigado pela atenção !

Questions ? ! ? ! / Questões ? ! ? !

Merci bien pour votre attention !

Obrigado pela atenção !

Bibliographie

D. S. Nikolopoulos and C. D. Polychronopoulos.

Adaptive scheduling under memory pressure on multiprogrammed clusters.

In *Proc. of the Second IEEE/ACM Int. Symp. on Cluster Computing and the Grid (CCGrid 2002), Berlin, Germany, (Best Paper Award)*, May 2002.

Mauricio A. Pillon.

Utilisations des compteurs matériels dans le multiprocesseurs estimation de l'accélération et contrôle-commande d'exécution.

In *RENPAR2002 - 14èmes Rencontres Francophones du Parallèlisme*, pages 193–200, 10–13 April 2002.

Hammamet - Tunisie.

Maurício Pillon, Olivier Richard, and Georges DaCosta.

DRAC : Adaptive Control System with Hardware Performance Counters.

In *Euro-Par*, Lecture Notes in Computer Science. Springer, 2004.

Un grand merci à tous ...