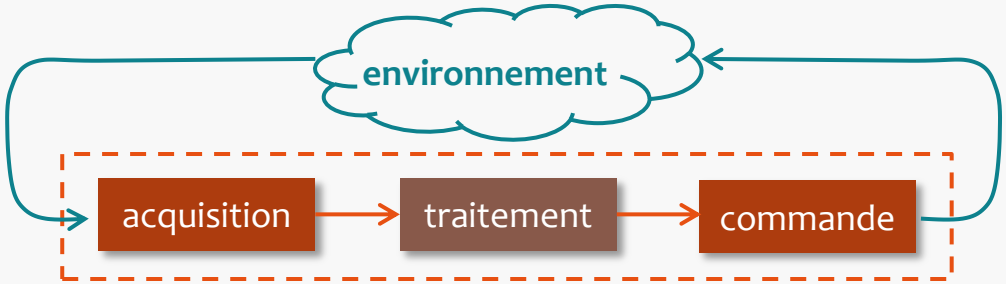
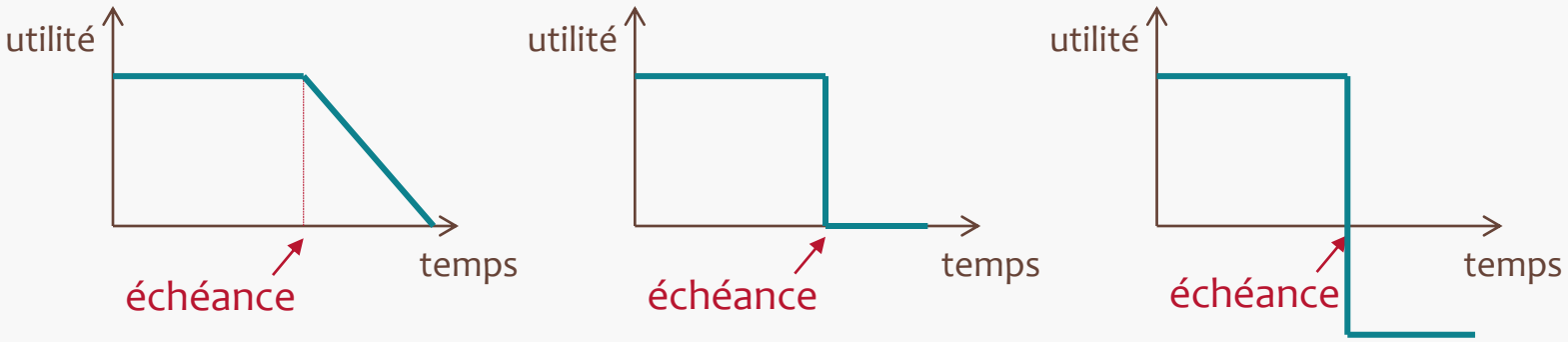
Architectures parallèles et temps d'execution pire cas

Christine Rochange

Ecole thématique ARCHI'11 – 17 juin 2011

Temps-réel strict



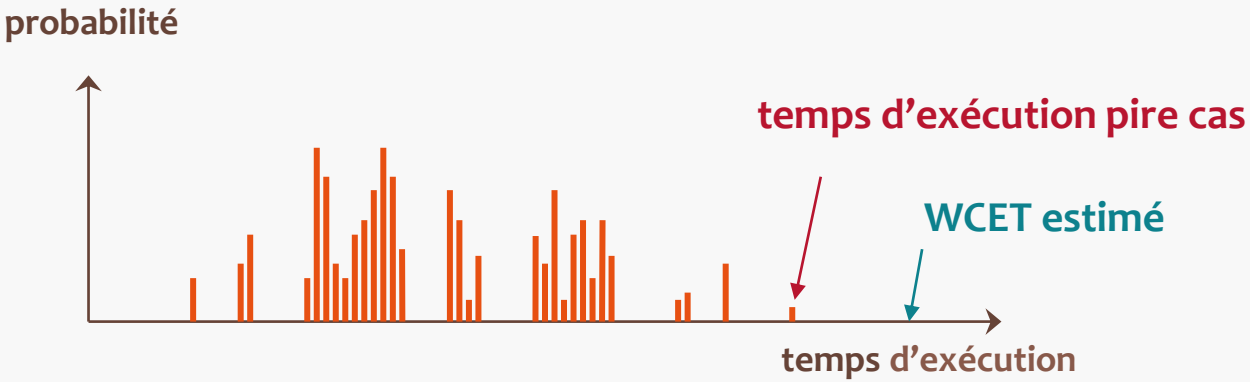


Temps d'exécution pire cas

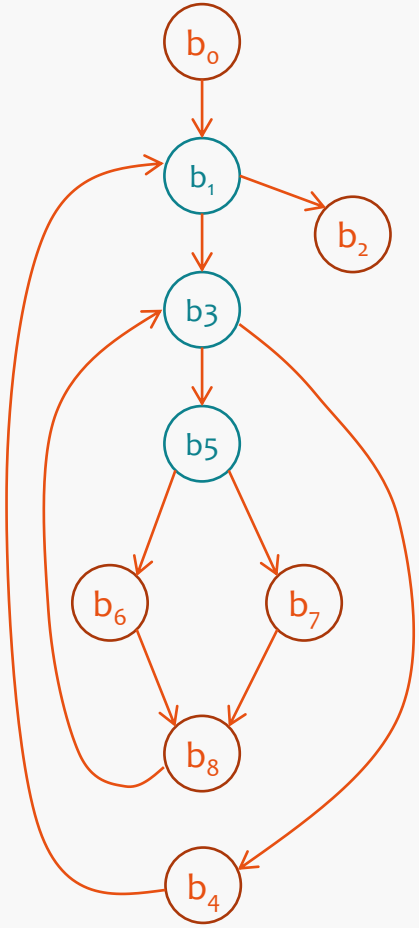
WCET = Worst-Case Execution Time

Le WCET : qu'est-ce que c'est ?

Université de Toulouse



WCET = Worst-Case Execution Time



Le WCET : à quoi ça sert ?

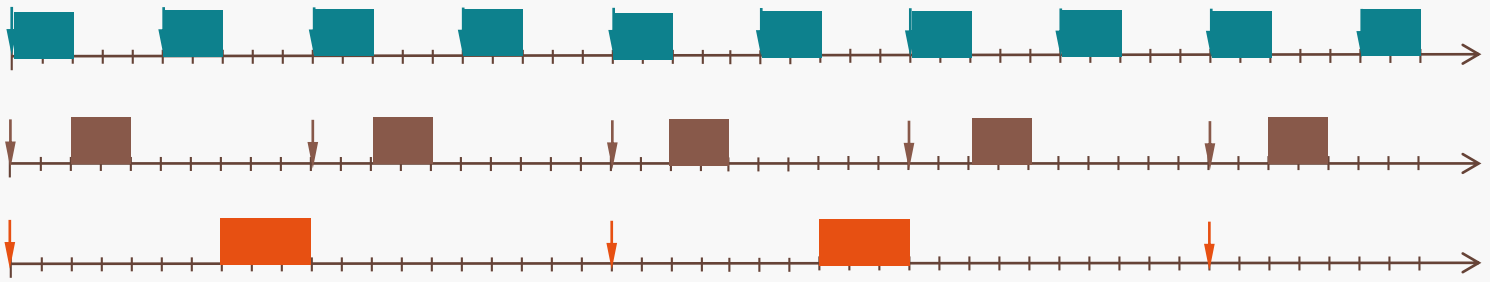
Université de Toulouse

- A répondre aux questions suivantes :
  - **Est-ce qu'une tâche isolée peut respecter une contrainte de temps ?**
    - est-ce que le matériel fournit une puissance de calcul suffisante ?
    - est-ce que le matériel est sur-dimensionné ?
  - **Est-ce qu'on peut générer un ordonnancement d'un ensemble de tâches qui respecte leurs contraintes ?**
    - préemptif ou non préemptif
    - en ligne ou hors ligne

Test d'ordonnançabilité

Université de Toulouse

	C_i =WCET	$P_i=D_i$
τ_1	2	5
τ_2	2	10
τ_3	3	20



Test d'ordonnançabilité pour l'algorithme EDF non préemptif :

- $\sum_{i=1}^{n} \frac{C_i}{P_i} \leq 1$
- $C_i + \sum_{j=1}^{i-1} \left\lfloor \frac{L-1}{P_j} \right\rfloor \times C_j \leq L$ avec $1 < i \leq n$ et $P_1 < L < P_i$

$$\frac{2}{5} + \frac{3}{20} + \frac{1}{10} = 0{,}65 \leq 1$$

τ_1 ordonnançable, τ_2 ?

$$L = 6: \quad 2 + \left\lfloor \frac{6-1}{5} \right\rfloor \times 2 = 4 \leq 6$$

$$L = 9: \quad 2 + \left\lfloor \frac{9-1}{5} \right\rfloor \times 2 = 4 \leq 9$$

τ_2 ordonnançable, τ_3 ?

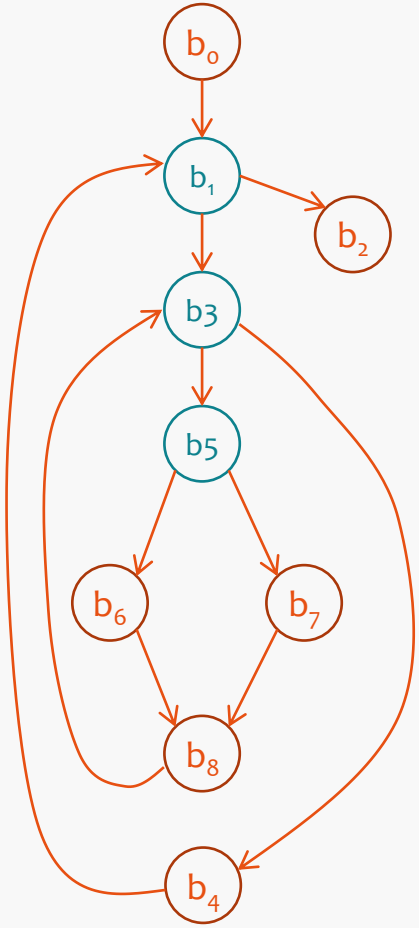
$$L = 6: \quad 3 + \left\lfloor \frac{6-1}{5} \right\rfloor \times 2 + \left\lfloor \frac{6-1}{10} \right\rfloor \times 2 = 5 \leq 6$$

$$L = 19: \quad 3 + \left\lfloor \frac{19-1}{5} \right\rfloor \times 2 + \left\lfloor \frac{19-1}{10} \right\rfloor \times 2 = 11 \leq 19$$

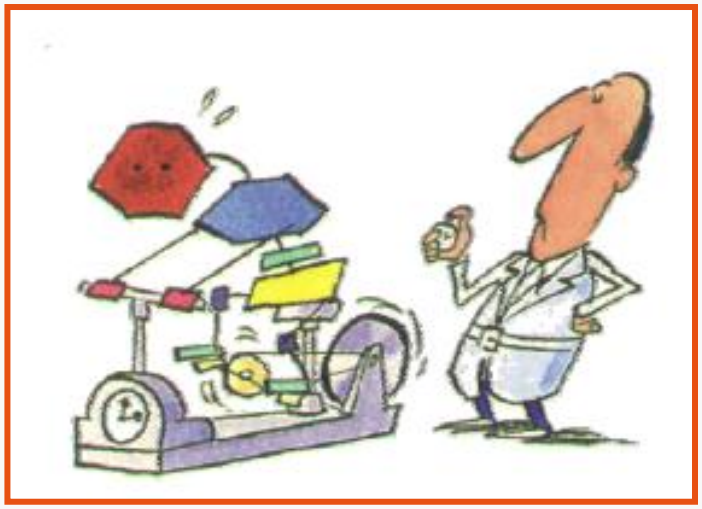
Calcul de WCET : que doit-on déterminer ?

Université de Toulouse

- **Quels sont les chemins possibles ?**
  - bornes de boucles
  - informations sur les conditionnelles
  - …
- **Quels sont les temps d'exécution des chemins possibles ?**
  - liés à l'architecture cible
- **Quel est le chemin le plus long et quel est sont temps d'exécution ?**



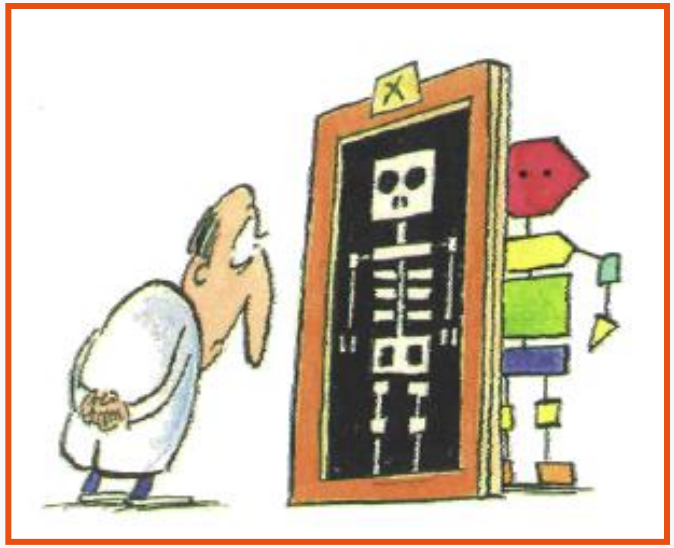
Deux approches



Mesures :

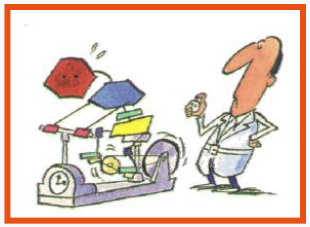
- matériel réel
- simulateur

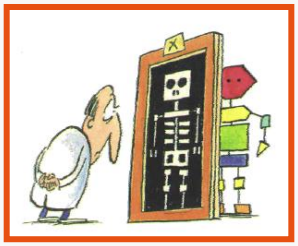
Analyse statique



Identifier tous les chemins possibles

Université de Toulouse

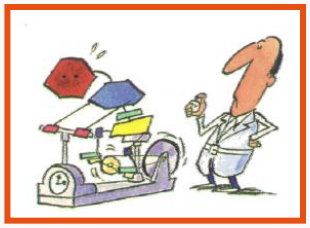


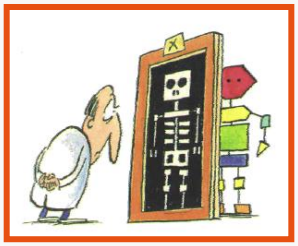


- **Jeux d'entrées**
  - exhaustifs
  - couvrant tous les chemins (PathCrawler - CEA)
- **Exécution symbolique [LuSt99]**
  - propagation des valeurs inconnues des entrées
  - exploration des chemins possibles
- **Chemins observés**
  - pour raffiner les résultats de l'analyse statique

- **Analyse de flot**
  - extraction du CFG
  - calcul d'informations de flot
    - à partir du source ou du binaire
    - bornes de boucles
    - switches
    - chemins infaisables
    - nombreuses références

Obtenir les temps d'exécution

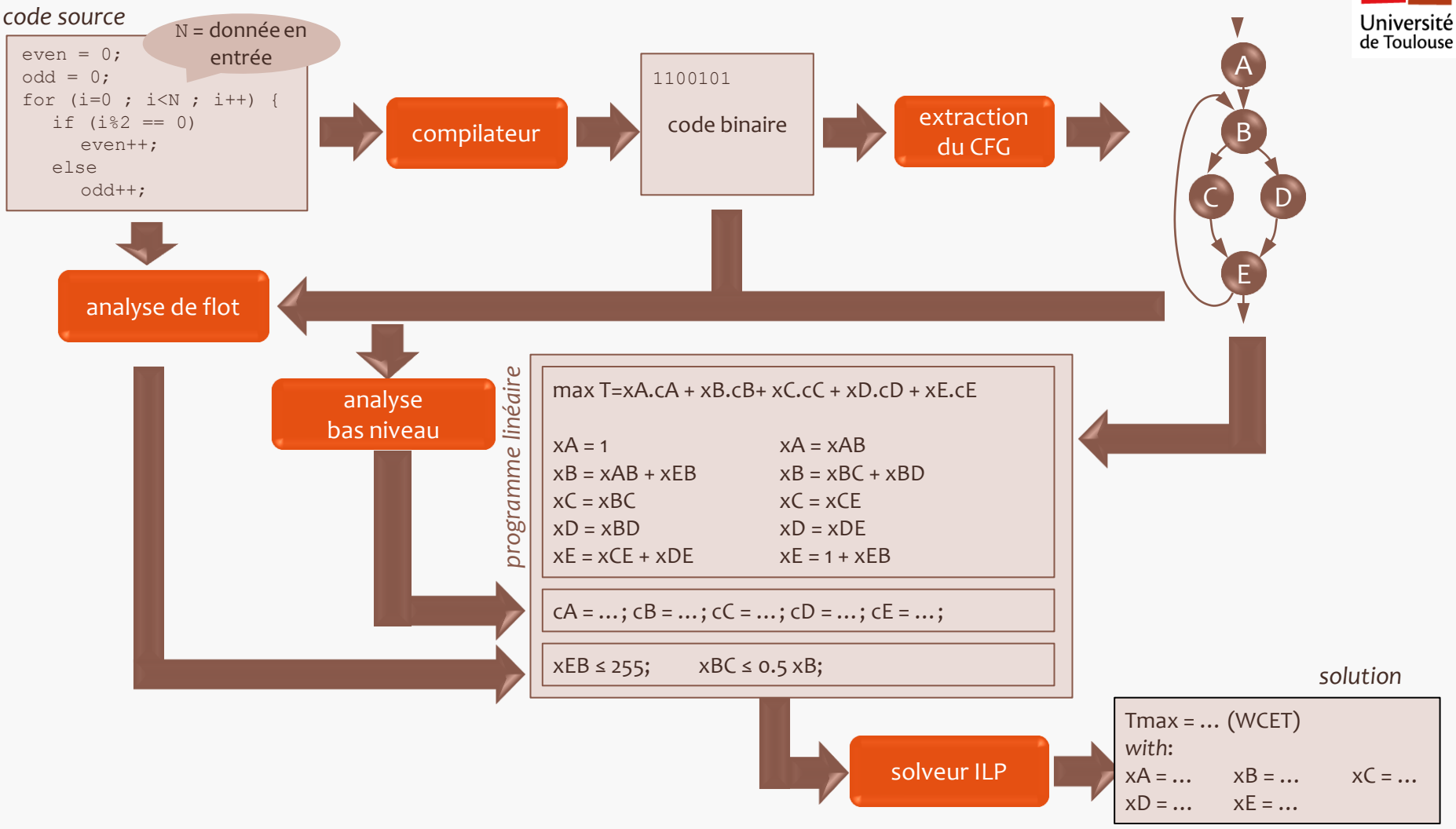




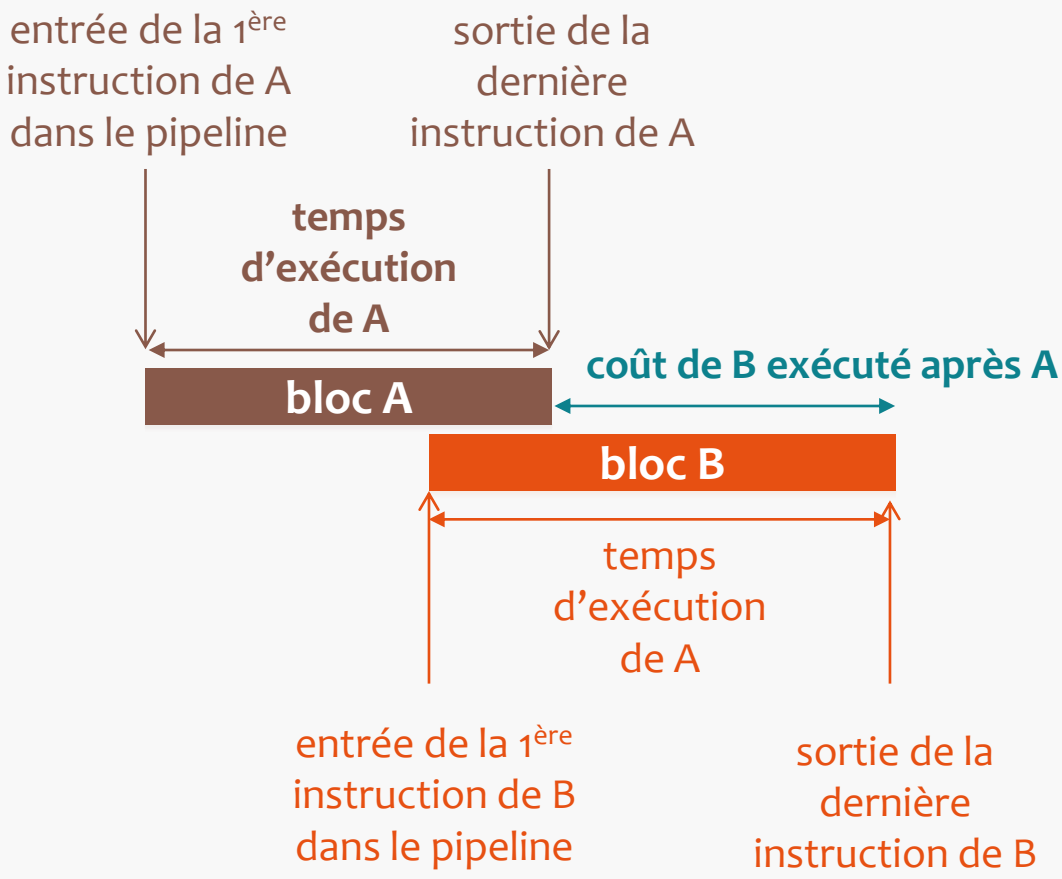
- **Matériel réel ou simulateur ?**
  - matériel :
    - mise en œuvre ? observabilité ?
    - disponibilité ?
  - simulateur :
    - fiabilité ?
- **Quels chemins ?**
  - « marges de sécurité » ???
  - calcul hybride ?
  - chemin unique ? [Pusc03]

- **Modèle de l'architecture**
  - fiabilité ? (cf. docs)
    - traduction de modèle VHDL
  - prise en compte des états possibles du matériel
    - pipeline
    - caches
    - prédicteur de branchement

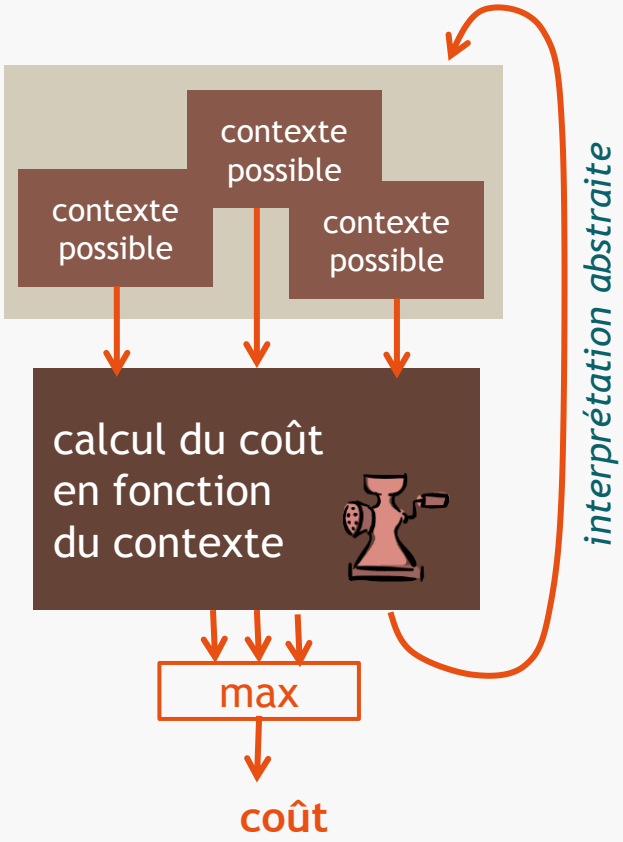
Calcul de WCET : méthode IPET



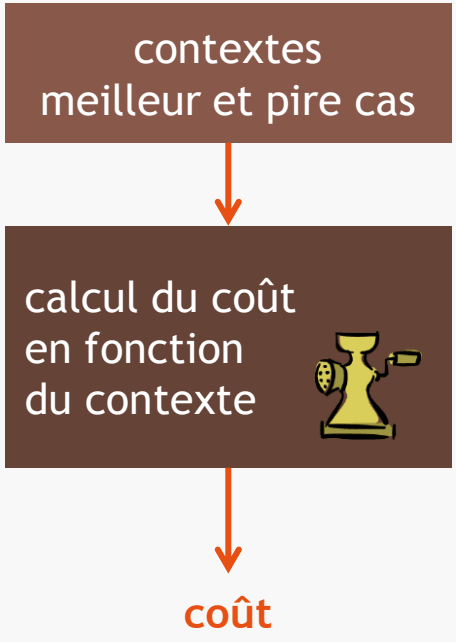
Coût d'exécution d'un bloc de base



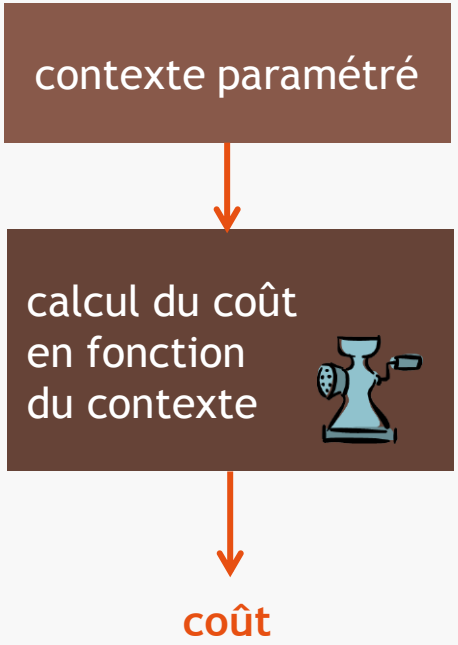
Analyse du pipeline



[Sarrebrücken, AbsInt]



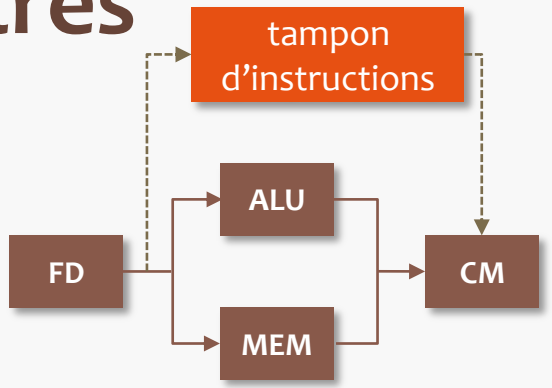
[Singapour]

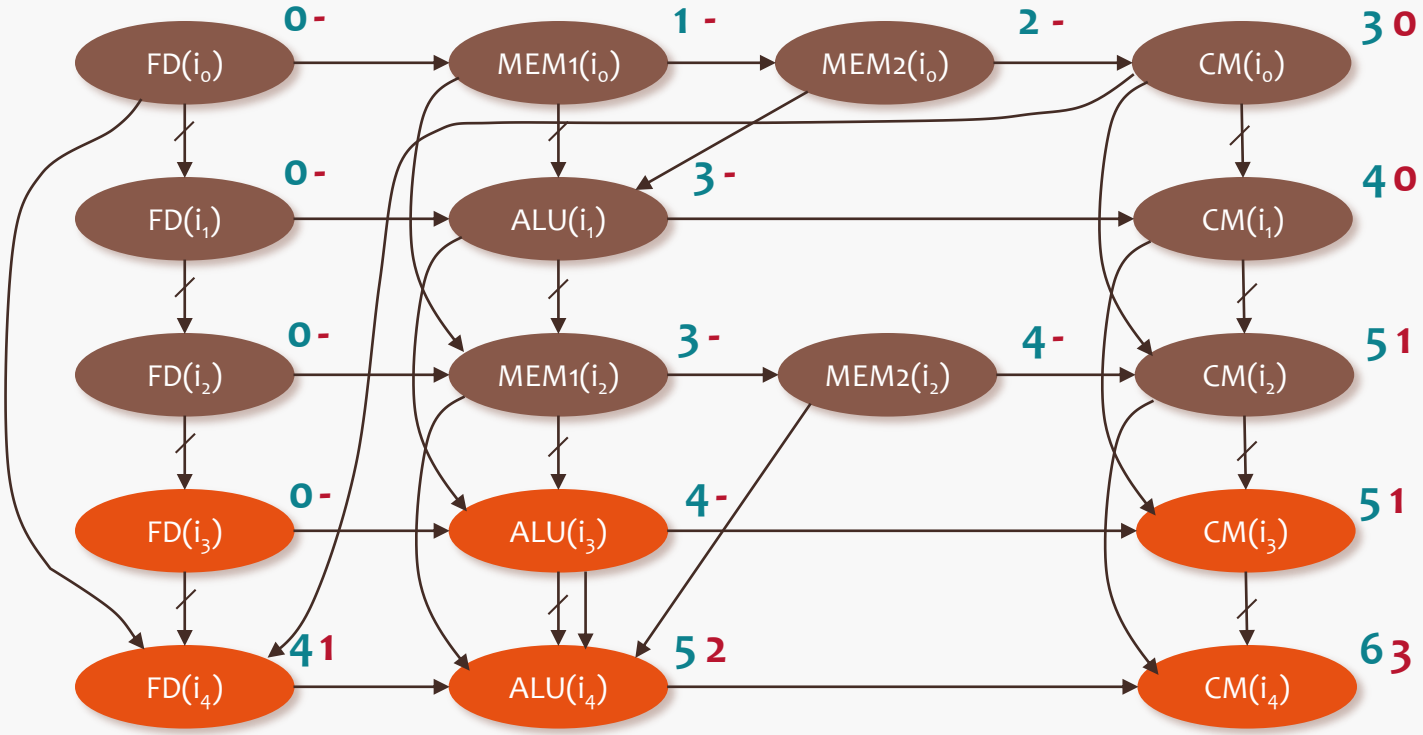


[Toulouse]

Graphes d'exécution paramétrés


```
i0: r0 ← MEM[@x]
i1: r1 ← r0 + 8
i2: r2 ← MEM[@y]
i3: r3 ← r10 + 12
i4: r4 ← r2 + r3
```





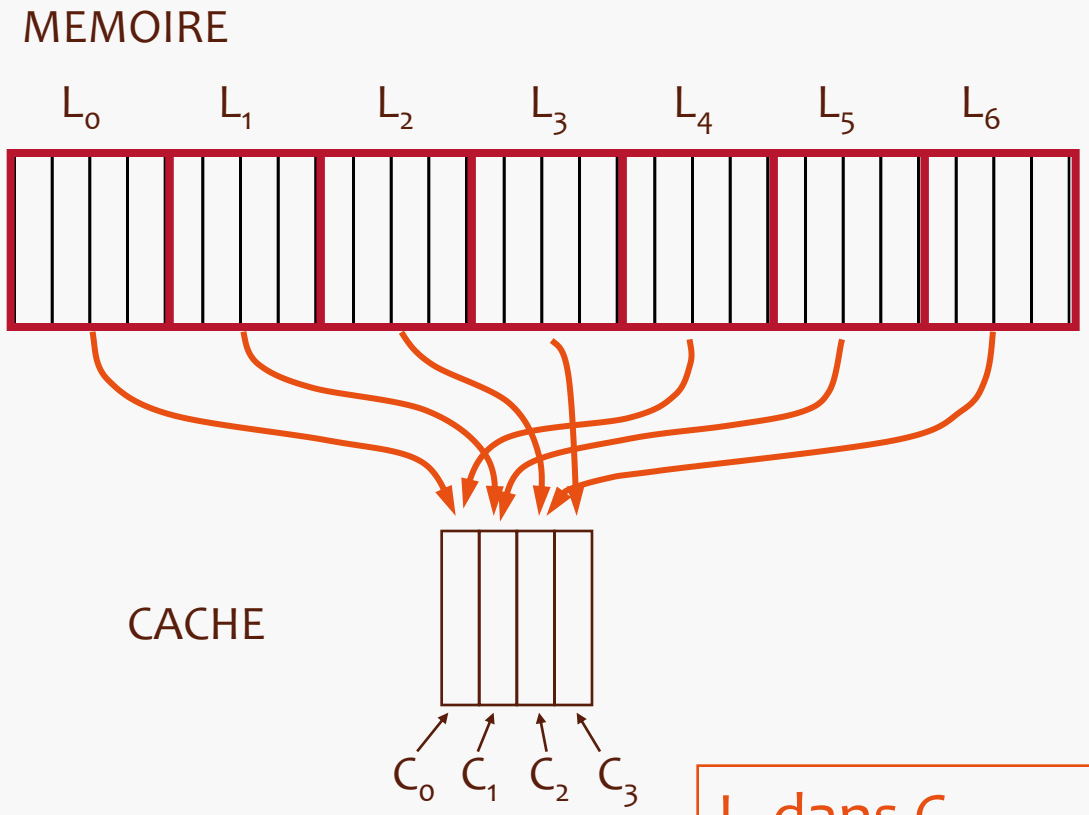
x = délai par rapport à l'entrée dans le pipeline

x = délai par rapport à la disponibilité d'un premier slot de l'étage CM

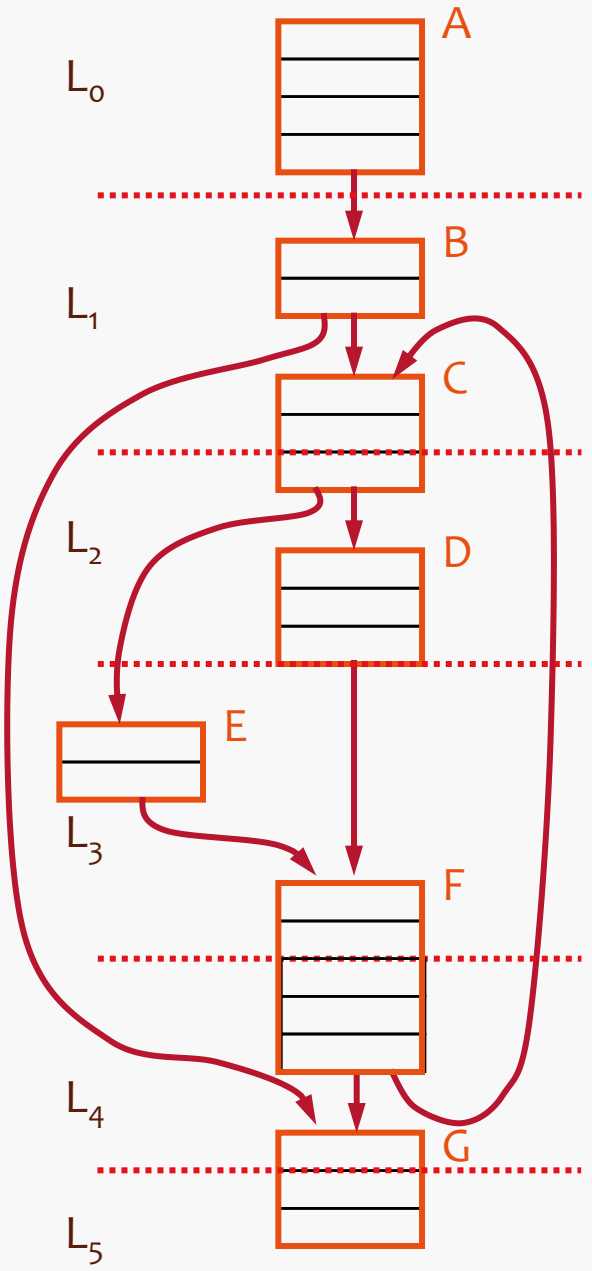
WCC = max(1,2, …)

Analyse d'un cache (1)

- **Exemple**



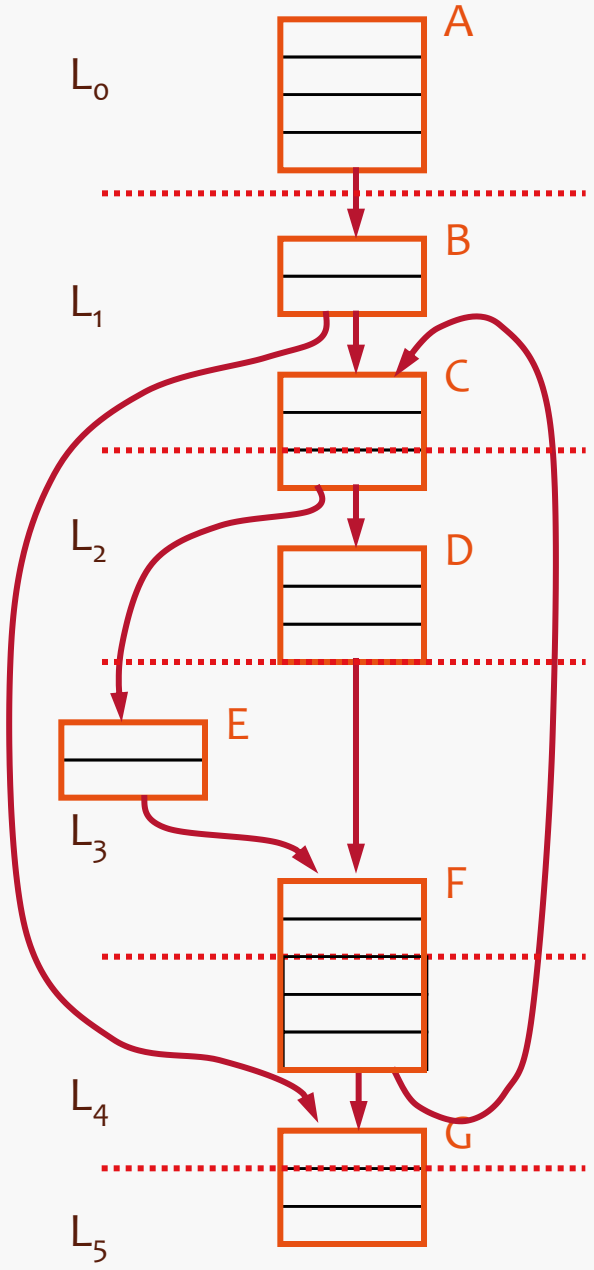
L_i dans $C_{i \bmod 4}$



Analyse d'un cache (2)

- **Première passe**

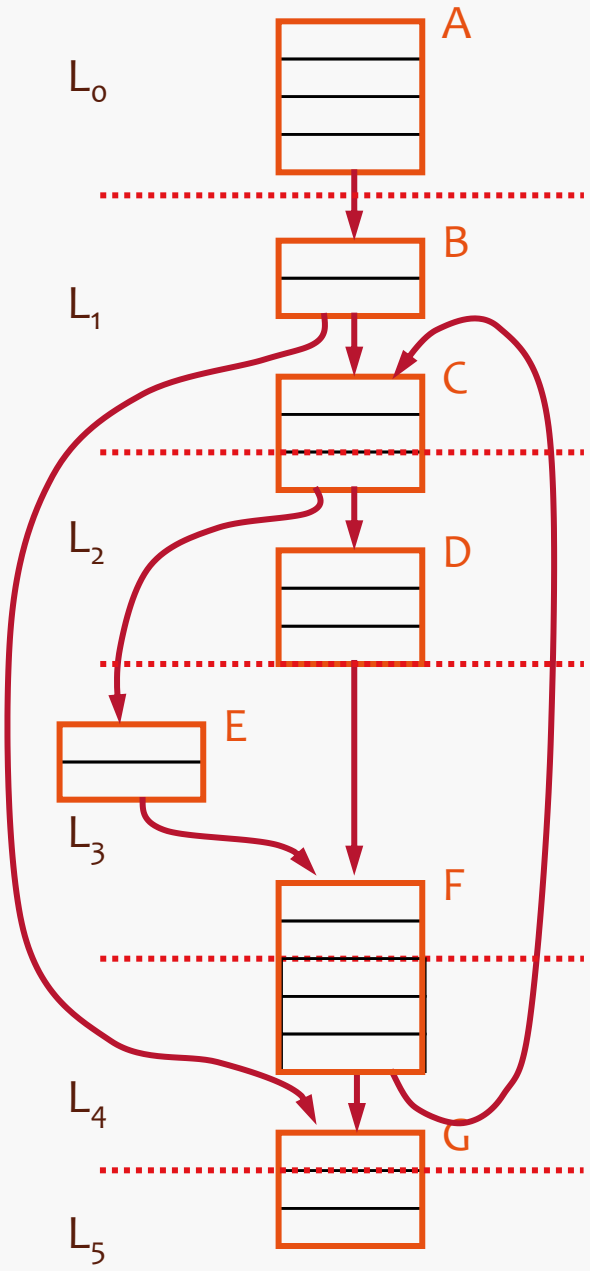
	entrée				sortie			
	C_0	C_1	C_2	C_3	C_0	C_1	C_2	C_3
A	∅	∅	∅	∅	L_0	∅	∅	∅
B	L_0	∅	∅	∅	L_0	L_1	∅	∅
C	L_0	L_1	∅	∅	L_0	L_1	L_2	∅
D	L_0	L_1	L_2	∅	L_0	L_1	L_2	∅
E	L_0	L_1	L_2	∅	L_0	L_1	L_2	L_3
F	L_0	L_1	L_2	∅/L_3	L_4	L_1	L_2	L_3
G	$L_{0/4}$	L_1	∅/L_2	∅/L_3	L_4	L_5	∅/L_2	∅/L_3



Analyse d'un cache (3)

- **Seconde passe**

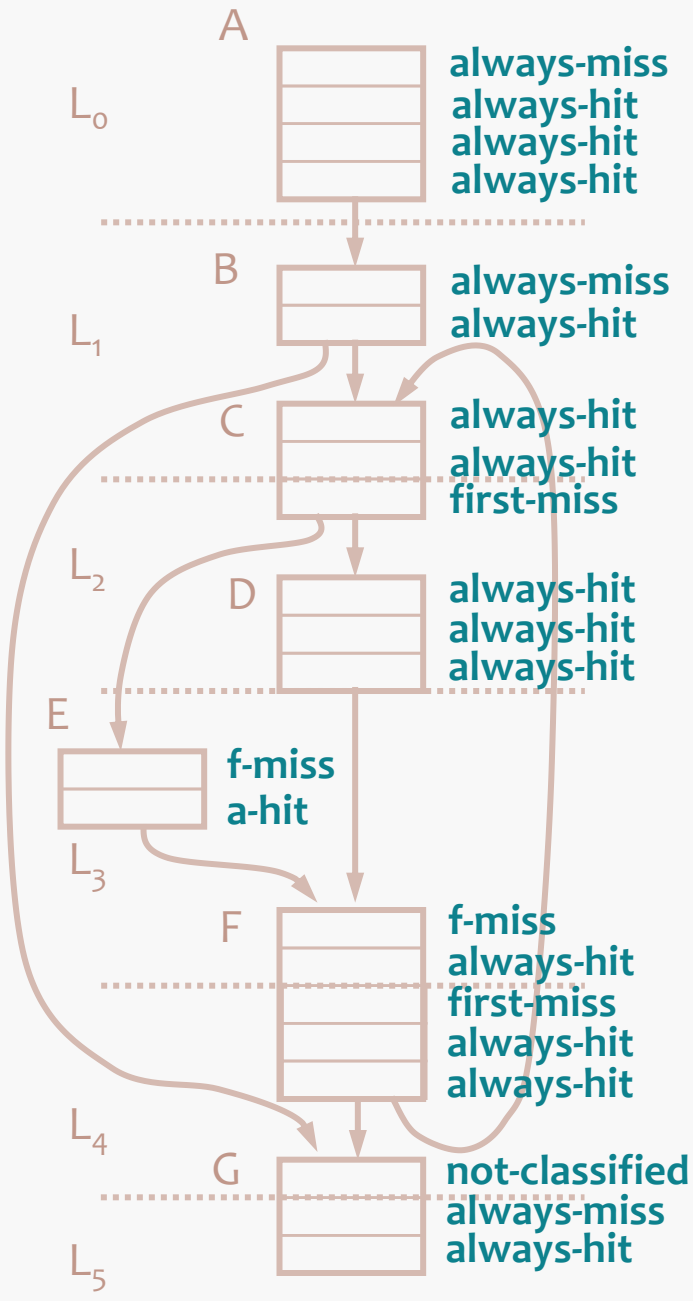
	entrée				sortie			
	C_0	C_1	C_2	C_3	C_0	C_1	C_2	C_3
A	∅	∅	∅	∅	L_0	∅	∅	∅
B	L_0	∅	∅	∅	L_0	L_1	∅	∅
C	$L_{0/4}$	L_1	∅/L_2	∅/L_3	$L_{0/4}$	L_1	L_2	∅/L_3
D	$L_{0/4}$	L_1	L_2	∅/L_3	$L_{0/4}$	L_1	L_2	∅/L_3
E	$L_{0/4}$	L_1	L_2	∅/L_3	$L_{0/4}$	L_1	L_2	L_3
F	$L_{0/4}$	L_1	L_2	∅/L_3	L_4	L_1	L_2	L_3
G	$L_{0/4}$	L_1	∅/L_2	∅/L_3	L_4	L_5	∅/L_2	∅/L_3



Analyse d'un cache (4)

Catégorisation des instructions

	entrée				sortie			
	C_0	C_1	C_2	C_3	C_0	C_1	C_2	C_3
A	∅	∅	∅	∅	L_0	∅	∅	∅
B	L_0	∅	∅	∅	L_0	L_1	∅	∅
C	$L_{0/4}$	L_1	∅/L_2	∅/L_3	$L_{0/4}$	L_1	L_2	∅/L_3
D	$L_{0/4}$	L_1	L_2	∅/L_3	$L_{0/4}$	L_1	L_2	∅/L_3
E	$L_{0/4}$	L_1	L_2	∅/L_3	$L_{0/4}$	L_1	L_2	L_3
F	$L_{0/4}$	L_1	L_2	∅/L_3	L_4	L_1	L_2	L_3
G	$L_{0/4}$	L_1	∅/L_2	∅/L_3	L_4	L_5	∅/L_2	∅/L_3

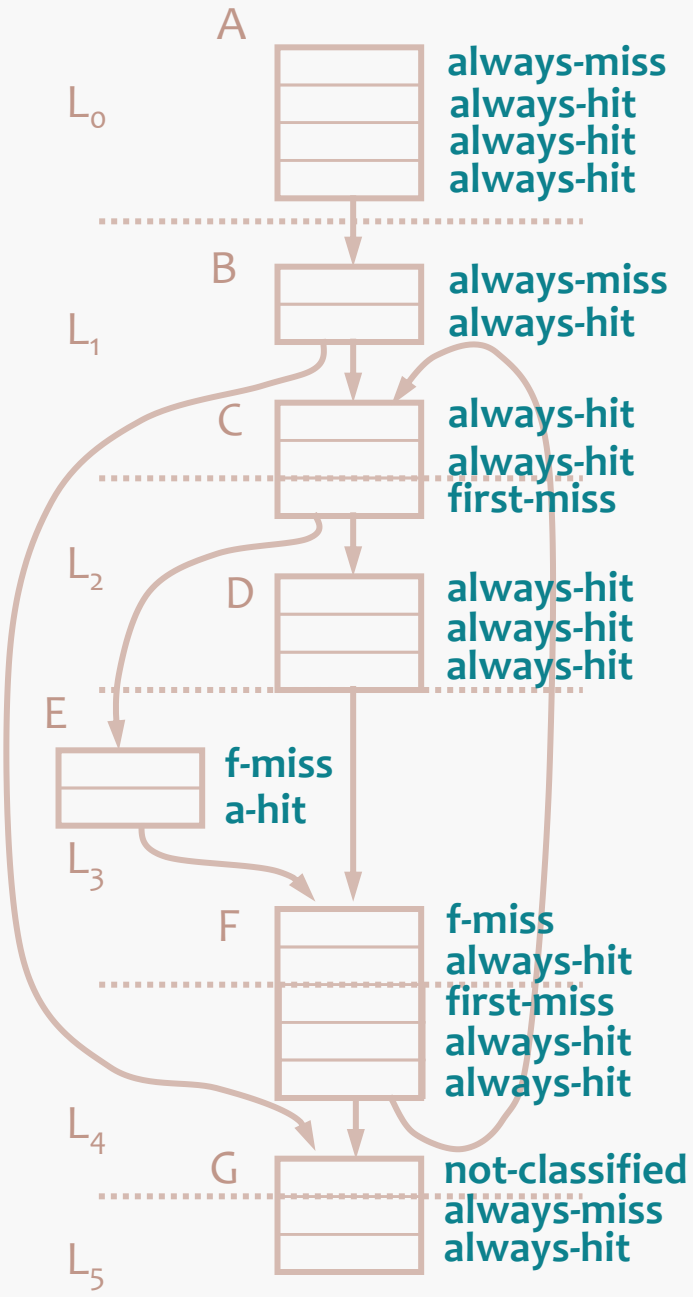


Analyse d'un cache (5)

- **Intégration dans le programme linéaire (IPET)**

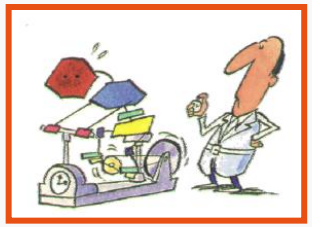
$$\max T = x_A.C_A + x_B.C_B + x_C.C_C + x_D.C_D + x_E.C_E + x_F.C_F + x_G.C_G + (x_A + x_B + 1 + 1 + 1 + 1 + x_G).P$$

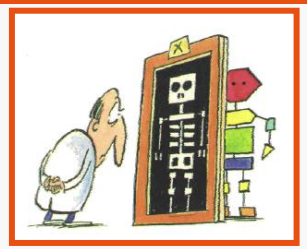
P = pénalité de défaut



Calcul de WCET : outils

Université de Toulouse





RapiTime C++

www.rapitasystems.com

Tid rum

Bound-T

www.bound-t.com

aiT

www.absint.com

SWEET

www.mrtc.mdh.se/projects/wcet/sweet.html

TU-Bound

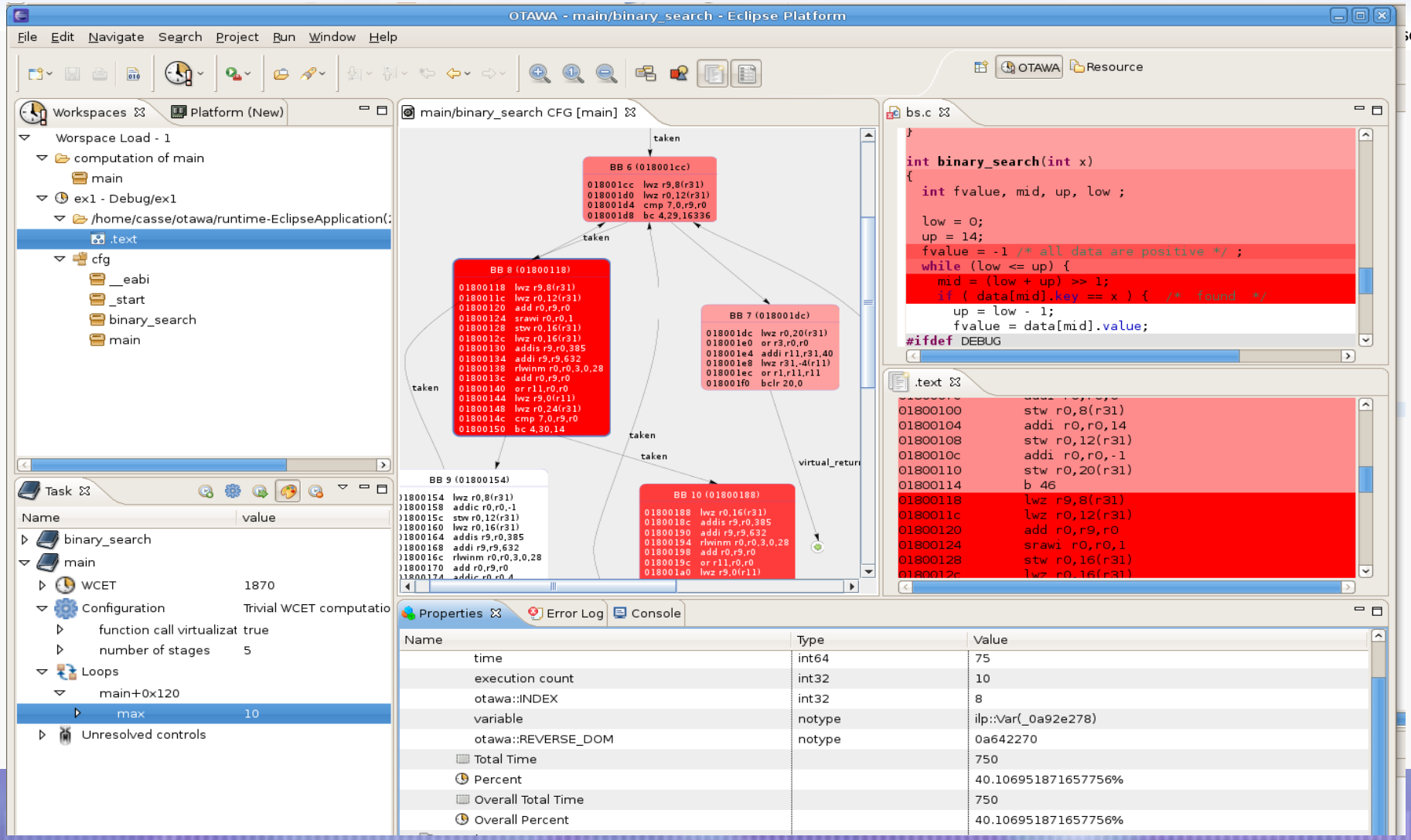
costa.tuwien.ac.at

OTAWA

www.otawa.fr

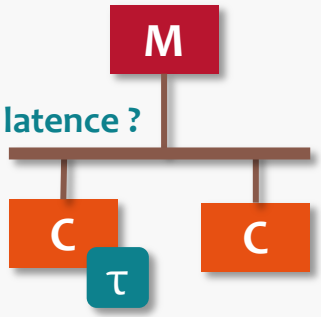
OTAWA

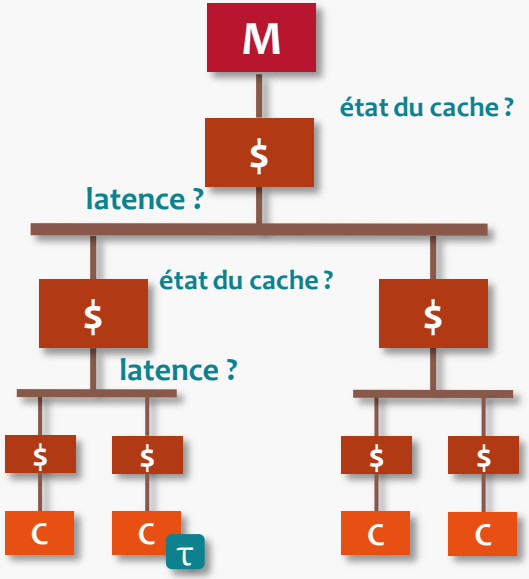
www.otawa.fr

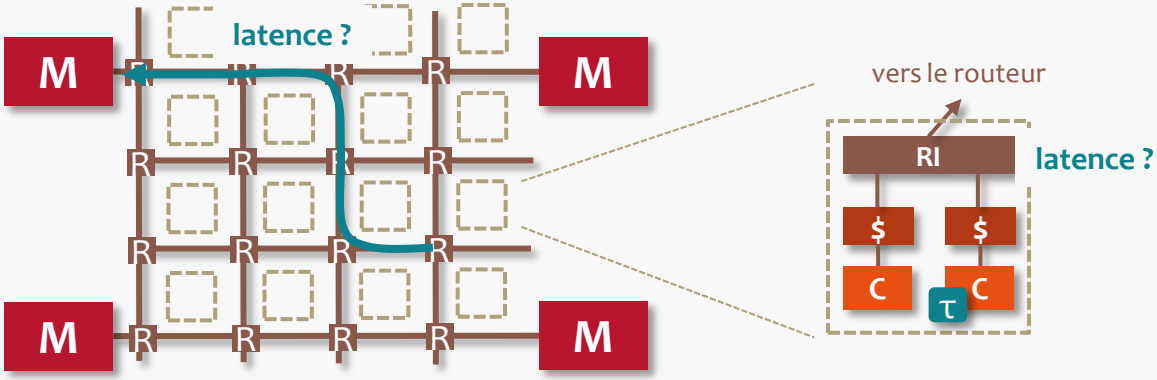


Architectures parallèles et temps d'exécution pire cas

Problèmes posés par le partage de ressources







bloc de base de la tâche τ

états abstraits du pipeline et du cache

état du cache ?

```
cmp r0,#8
beq L14
```

```
ldr r2,[r8,r9]
add r2,r2,#16
ldr r3,[r8,r10]
mul r3,r2,r3
str r3,[r8,r10]
```

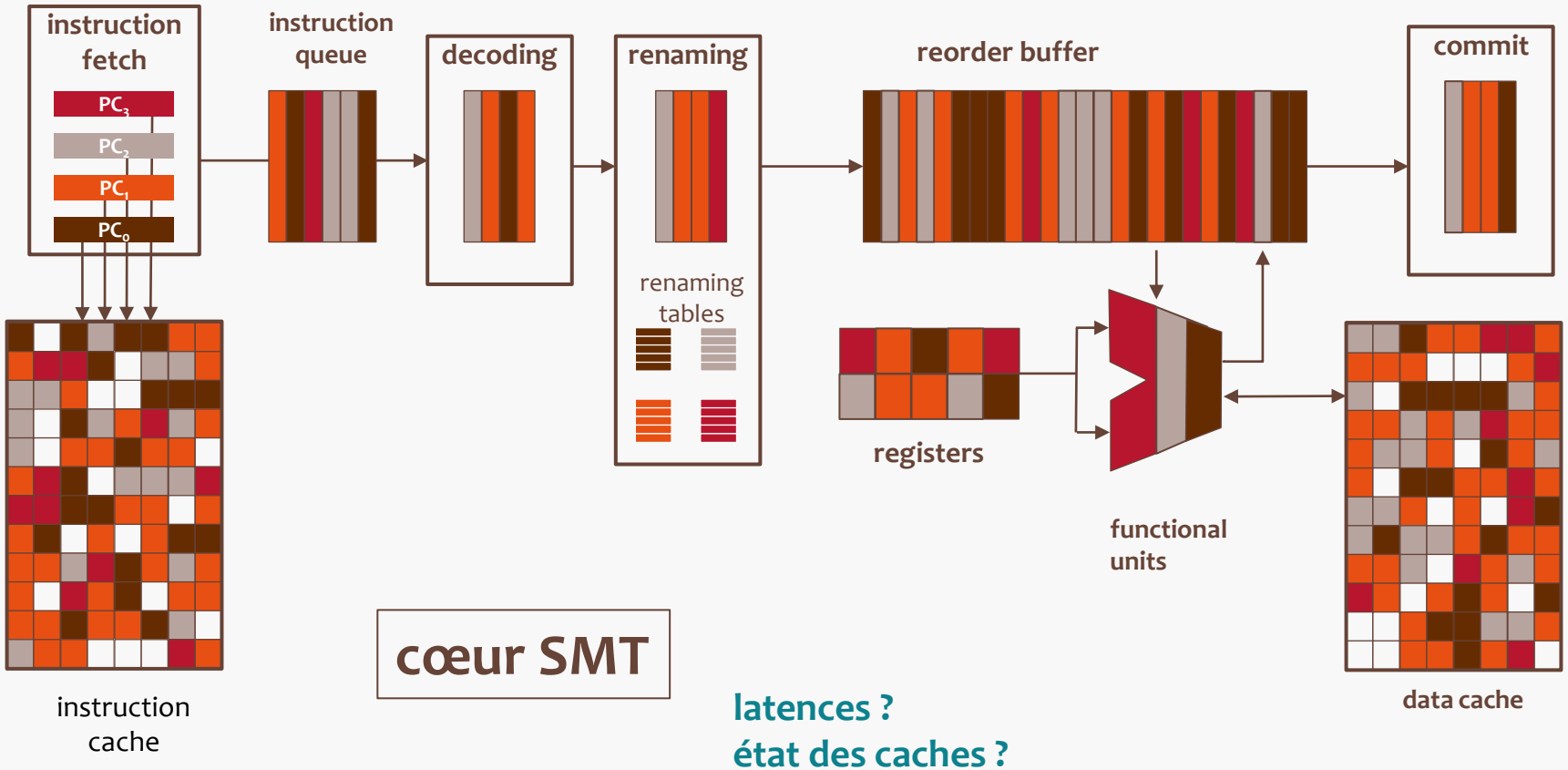
latence ?

latence ?

latence ?

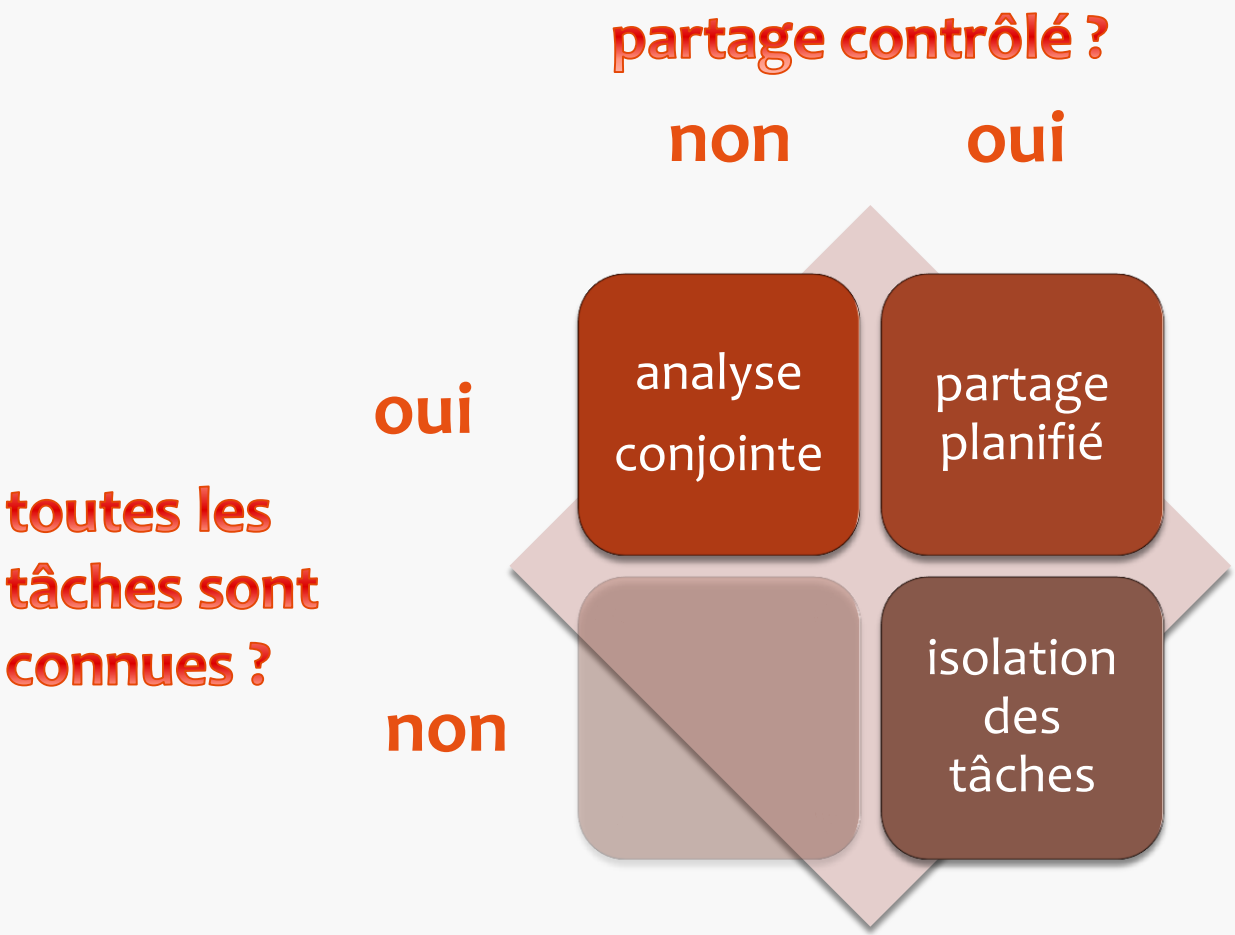
Problèmes posés par le partage de rssources (suite)

Université de Toulouse

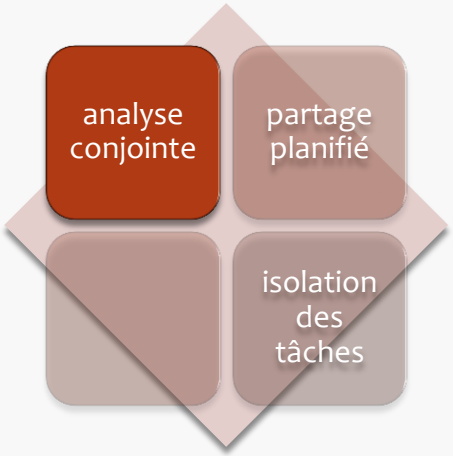


Approches pour analyser et rendre analysable le WCET

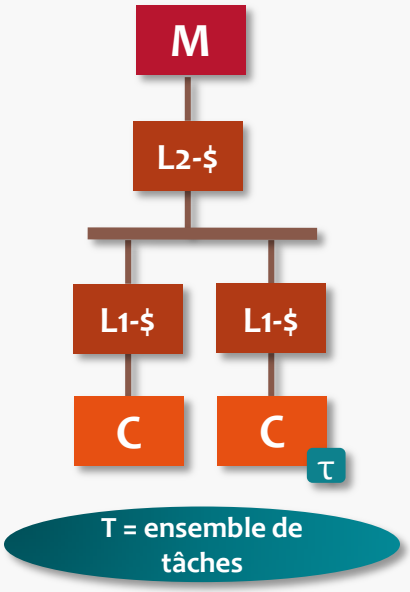
Université de Toulouse

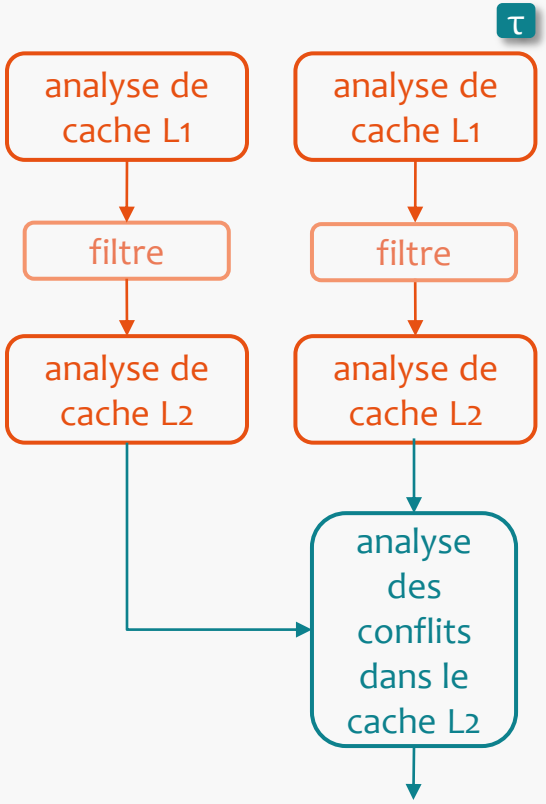


Analyse conjointe



- Cache L2 partagé

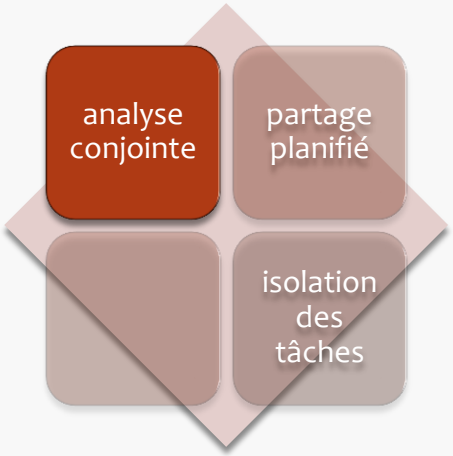




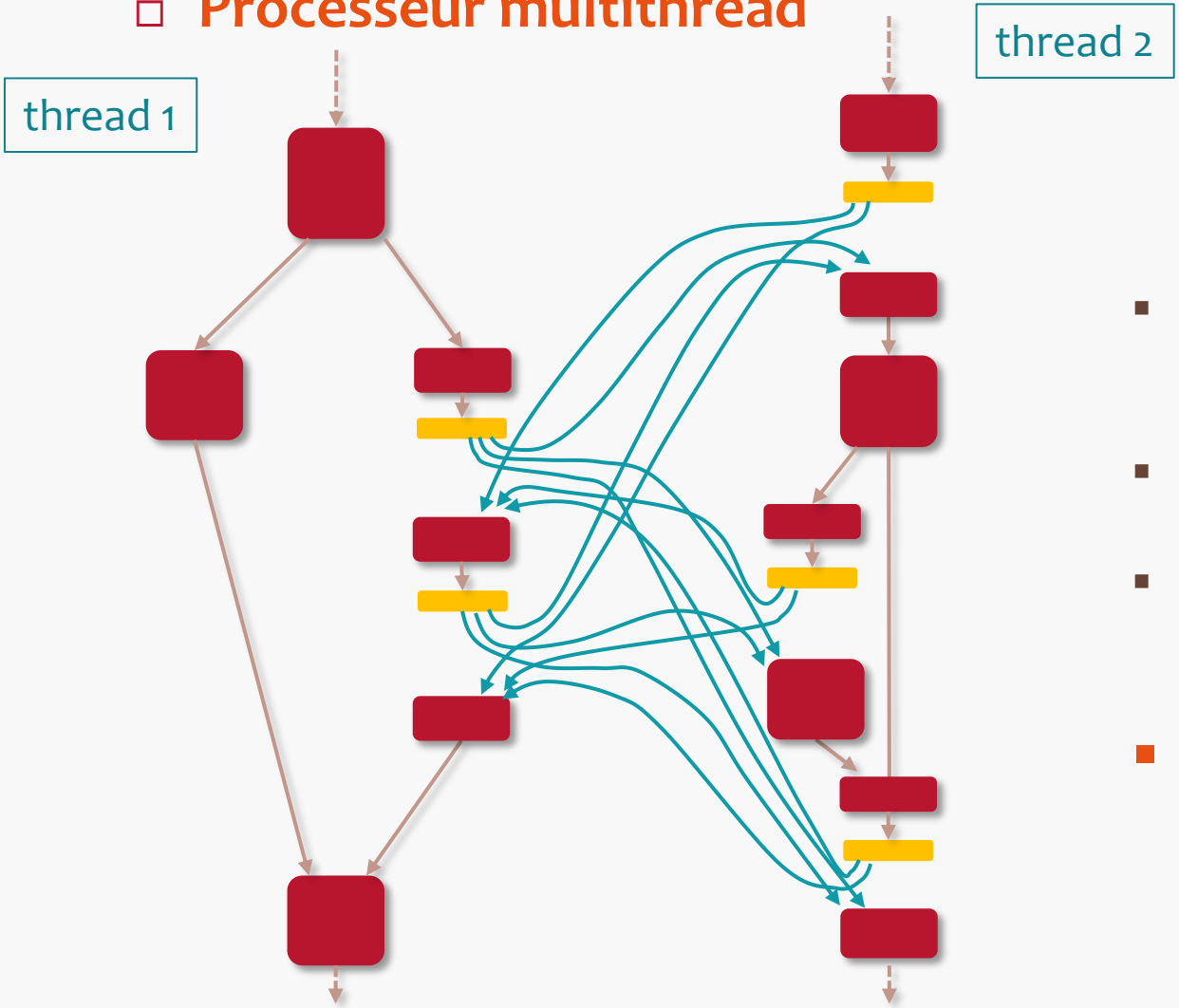
n'importe quel accès à l'ensemble S par une tâche de T est susceptible de corrompre S dans n'importe quel état abstrait du cache calculé pour τ

Yan *et al.*, RTAS2008 Hardy *et al.*, RTSS2009 Li *et al.*, RTSS2009

Analyse conjointe



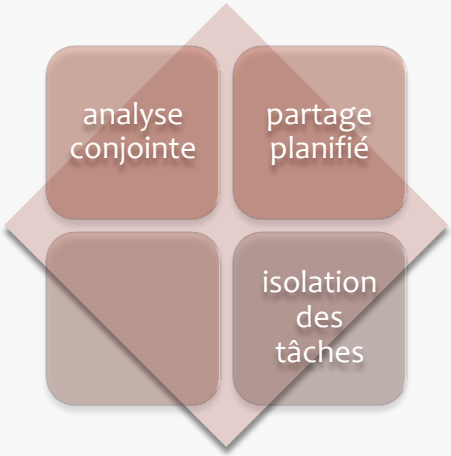
- **Processeur multithread**



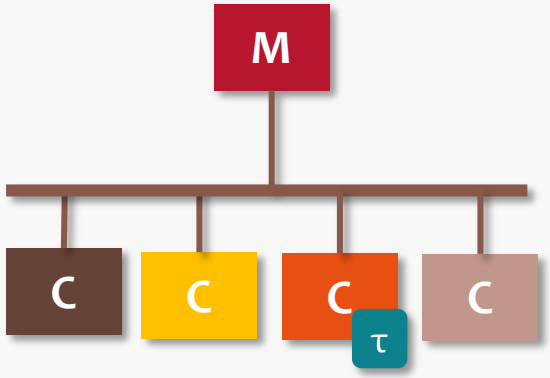
- changement de contexte sur opération à latence longue
- éclater les nœuds du CFG nodes ➤ nœuds **yield**
- arcs additionnels pour le transfert de contrôle
- **complexité !**

Crowley and Baer, WNP2003

Partage planifié



- **Arbitre de bus de type TDMA**



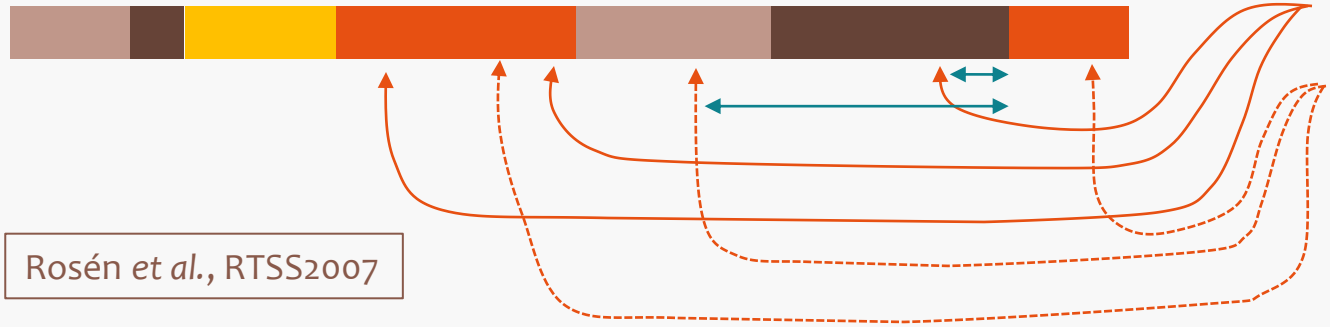
bloc de base de la tâche τ

états abstraits du pipeline et du cache

```
cmp r0,#8
beq L14
```

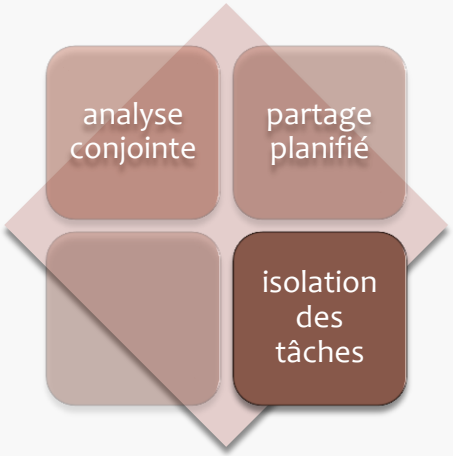
```
ldr r2,[r8,r9]
add r2,r2,#16
ldr r3,[r8,r10]
mul r3,r2,r3
str r3,[r8,r10]
```

Ordonnancement du bus



Rosén *et al.*, RTSS2007

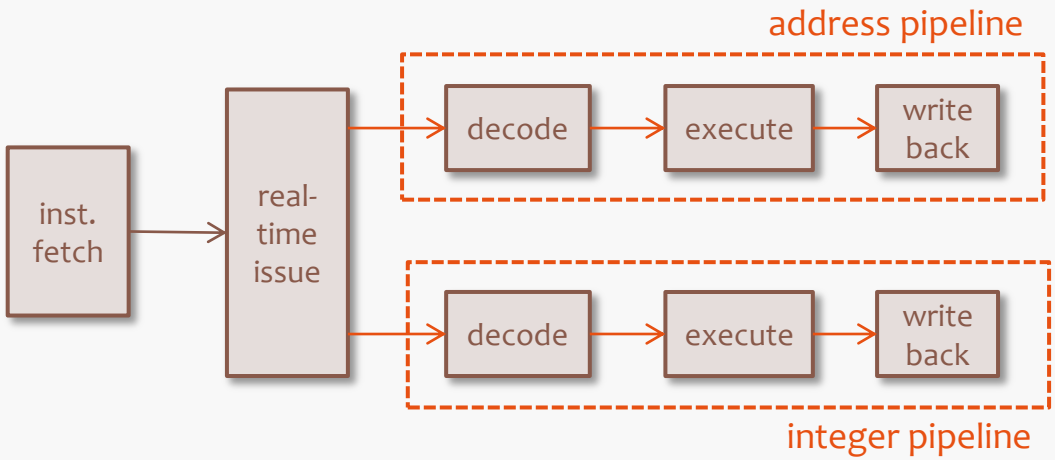
Isolation des tâches (1)



- **Partitionnement de bande passante**

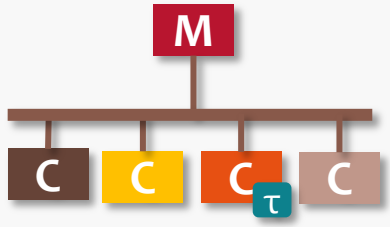
priorité au thread critique

CarCore



Mische *et al.*, ARCS2010

arbitre de bus round-robin

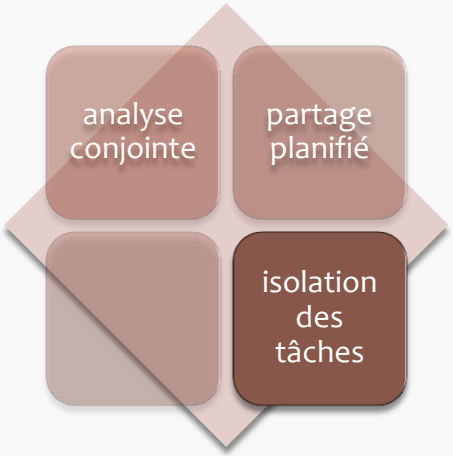


temps pour accéder au bus :

requête par τ

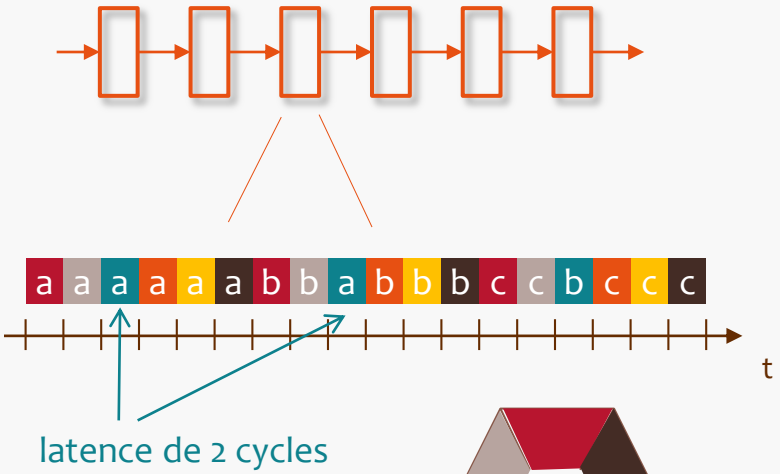
Paolieri *et al.*, ISCA2009

Isolation des tâches (2)



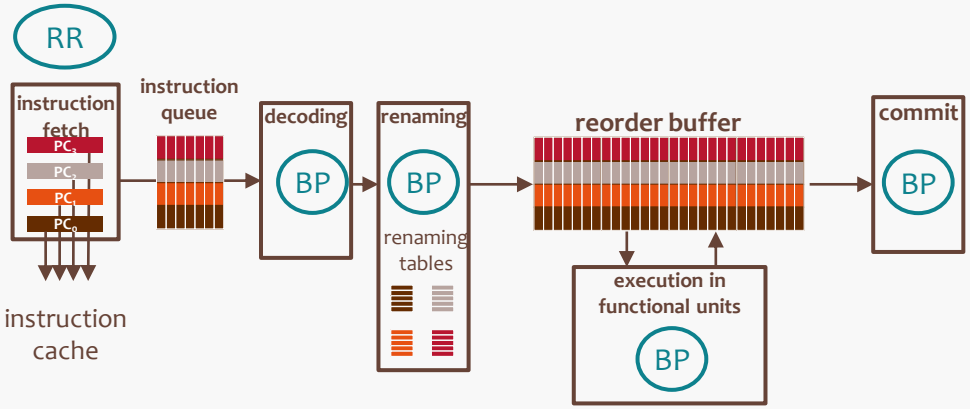
Partitionnement de bande passante

architecture PRET



roue mémoire

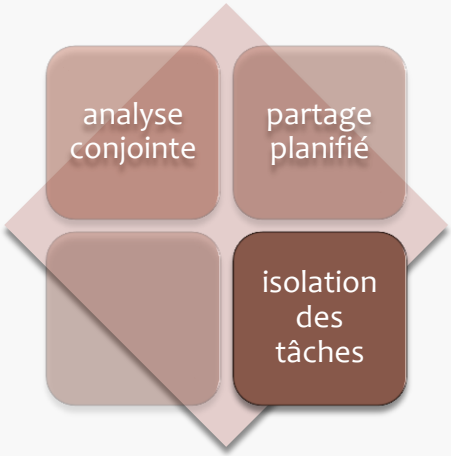
Lickly *et al.*, CASES2008



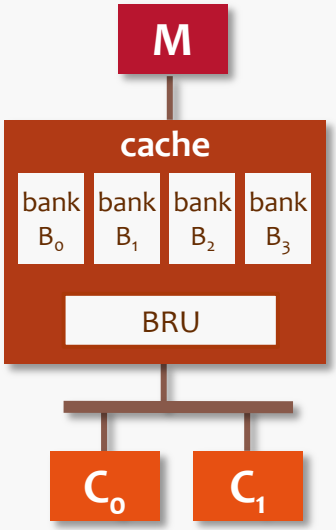
bande passante totale	1 thread critique	2 threads critiques	4 threads critiques
8 inst/cycle	8 inst/cyle	4 inst/cyle	2 inst/cyle
2 inst/cycle	2 inst/cycle	1 inst/cycle	1 inst. tous les 2 cycles

Barre *et al.*, SAMOS2008

Isolation des tâches (3)



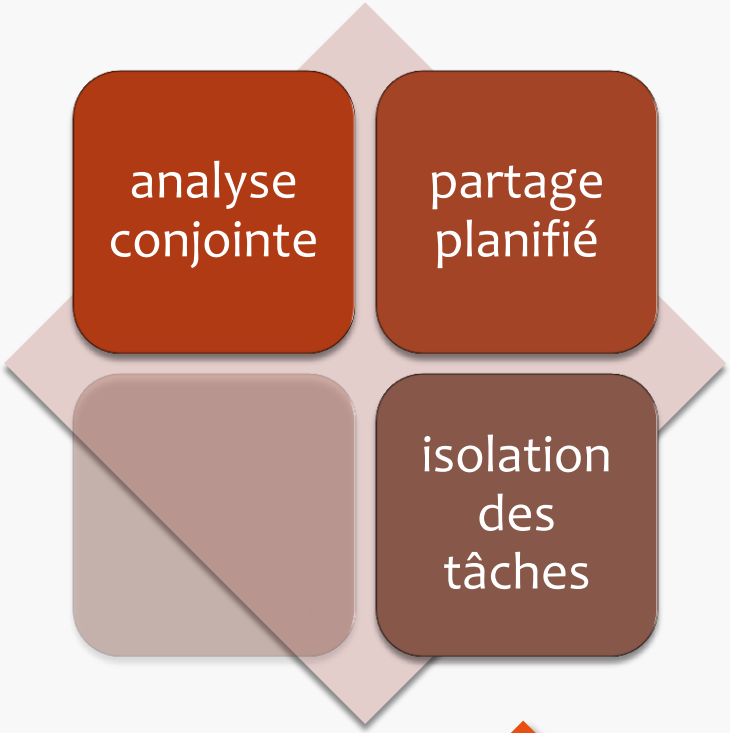
- **Partitionnement et verrouillage des mémoires**



Paolieri *et al.*, ISCA2009

En résumé …

Université de Toulouse



toutes les tâches doivent être connues

- applicable ?
- complexe
- et si une des tâches change ?
- impact de l'ordonnancement ?

matériel spécifique

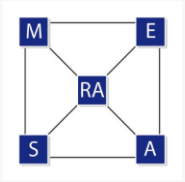
- analyse de WCET comme sur un mono-coeur
- performance (pire cas) réduite

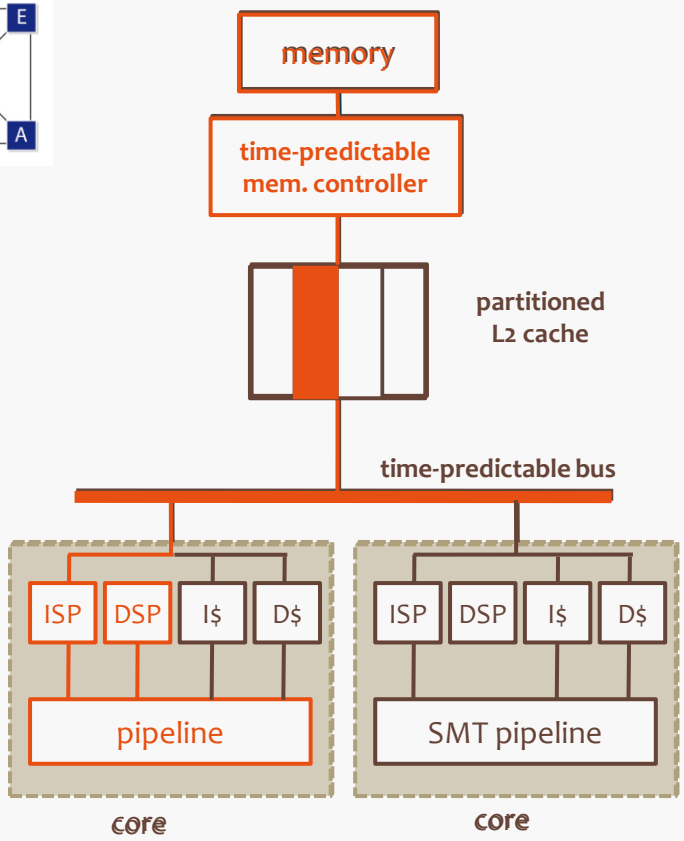
Le projet MERASA

- **Objectifs :**
  - développer un **processeur multicoeur** (de 2 à 16 coeurs) et le **support logiciel** pour des charges de travail de criticalité mixte
  - développer des **outils de calcul de WCET**
    - analyse statique : OTAWA
    - fondé sur des mesures : RapiTime
  - les évaluer sur des **études de cas** soumises par les partenaires industriels

- **Partenaires :**
  - **Participants:** Univ. Augsburg, Barcelona Supercomputing Center, Univ. Toulouse, Rapita Systems Ltd., Honeywell
  - **Industrial Advisory Board:** Airbus, BAUER Maschinen GmbH, European Space Agency, Infineon , NXP

L'architecture MERASA





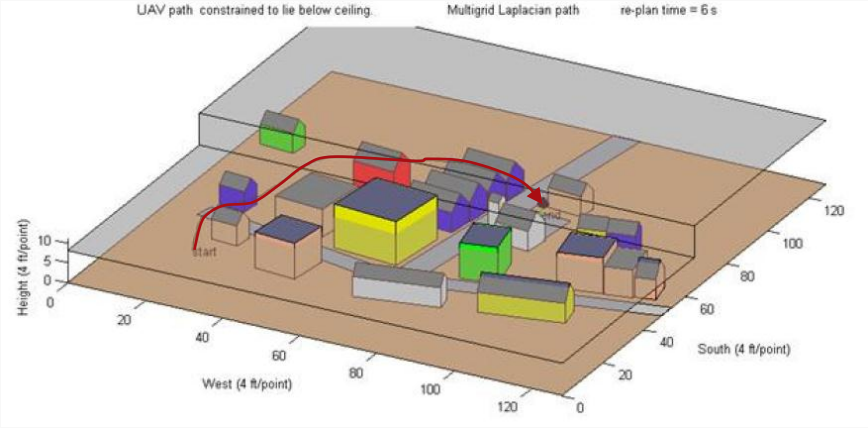
- **Isolation des threads critiques**
  - ordonnancement dans le pipeline
  - bus prévisible
  - cache L2 partitionné
  - contrôleur mémoire prévisible
- **Support logiciel**
  - gestion prévisible de la mémoire
  - synchronisations type POSIX prévisibles
  - ordonnancement des threads à deux niveaux

WCET de programmes parallèles

Solveur 3D multigrille

Université de Toulouse

- **Elément d'une application d'évitement de collision pour des drones (Honeywell)**
  - planification de chemin selon l'équation de Laplace
  - solution numérique :
    - discrétisation du domaine → voxels
    - potentiel de chaque point = moyenne de ses voisins
    - plusieurs itérations
  - approche multigrille :
    - plusieurs phases qui font varier la taille de la grille



- **Version considérée dans cette étude :**
  - trois phases -- chaque phase = **interpolation** + **itération**

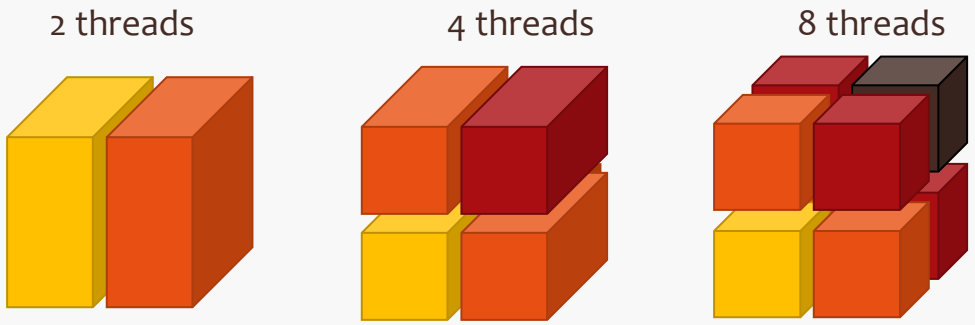
```
for (x=0; x<NX; x++)
   for (y=0; y<NY; y++)
      for (z=0; z<NZ; z++)
         v[x][y][z] = compInterpolate(old_v);
```

```
for (i=0; i<NUM_ITE; i++)
   for (x=0; x<NX; x++)
      for (y=0; y<NY; y++)
         for (z=0; z<NZ; z++)
            v[x][y][z] = compIterate(v);
```

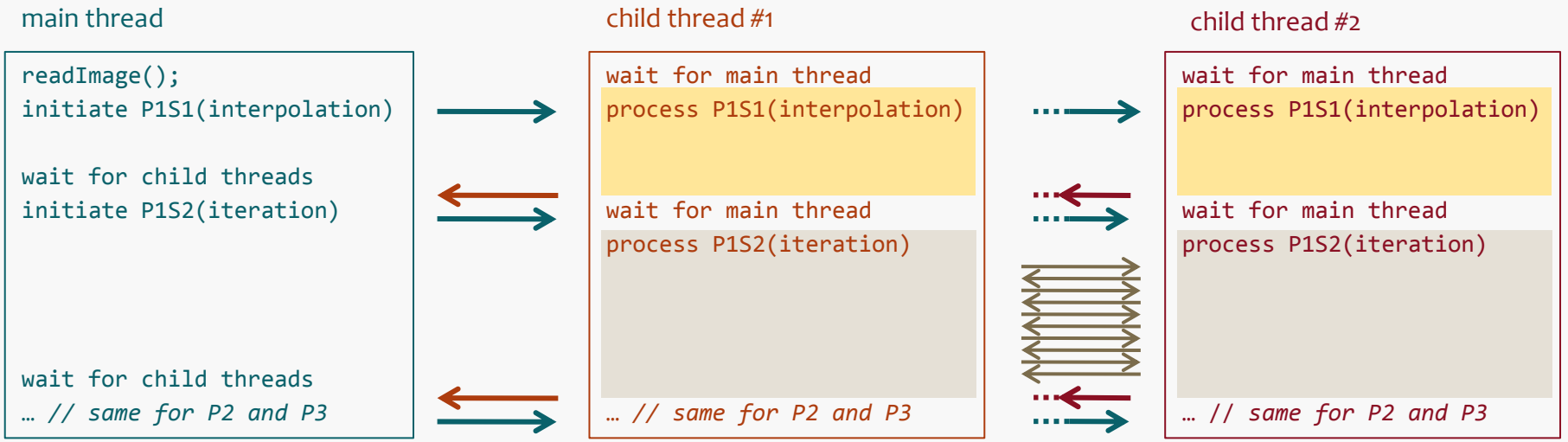
Version parallèle du solveur 3D

Université de Toulouse

- **Partitionnement**



- **Orchestration**

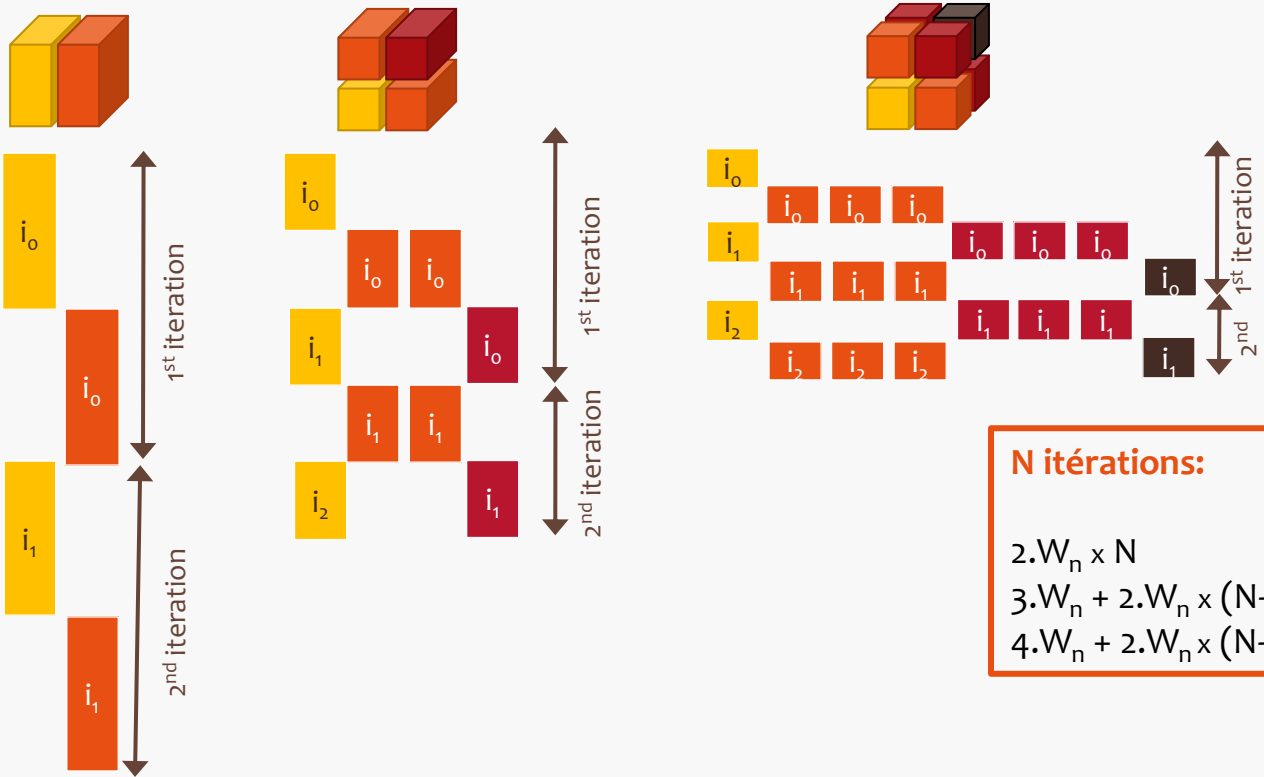


Décomposition du WCET

Université de Toulouse

- **Premier niveau**

- **Second niveau**



N itérations:

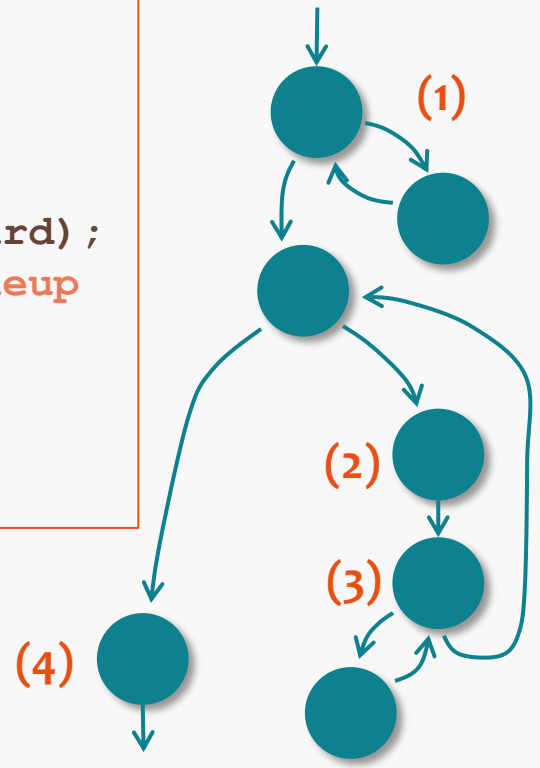
$2.W_n \times N$	(2 threads)
$3.W_n + 2.W_n \times (N-1)$	(4 threads)
$4.W_n + 2.W_n \times (N-1)$	(8 threads)

Analyse des synchronisations

Université de Toulouse

- Exemple: mutex lock

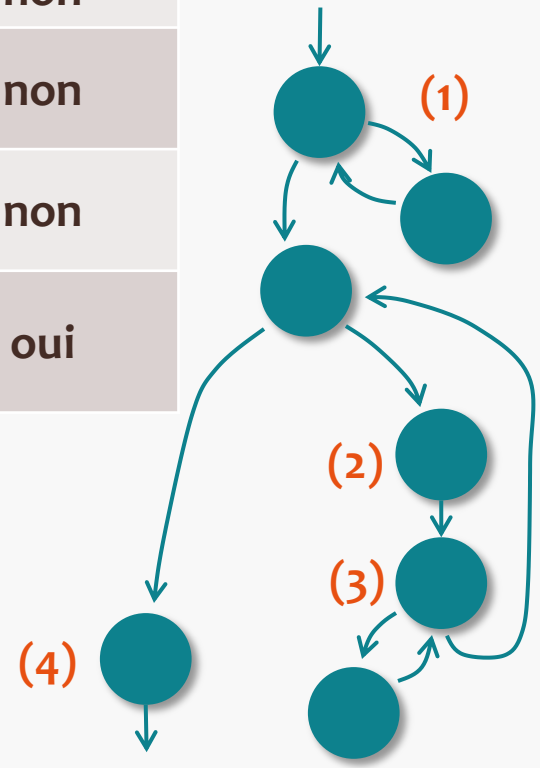
```
    int mutex_lock(mutex_t mutex){
(1)     sl_lock(&mutex->guard);
        …
        while (trylock(&mutex->thelock){
            … // insert thread into queue
(2)         sl_unlock_and_set_suspended(&mutex->guard);
(3)         spinlock_lock(&mutex->guard); // on wakeup
        }
        …
(4)     sl_unlock(&mutex->guard);
    }
```



Analyse des synchonisations (suite)

Université de Toulouse

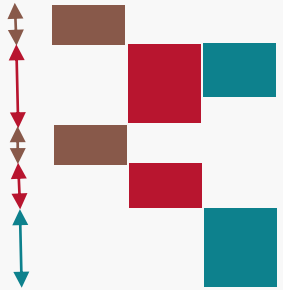
Terme		dépend de …	
		# threads	application
T_e	Temps d'exécution quand la variable est libre	non	non
T_{w1}	Surcoût quand le thread doit attendre la variable	non	non
T_{w2}	Temps d'attente lié aux variables du niveau système	oui	non
T_{w3}	Temps d'attente lié aux variables du niveau application	oui	oui



Bilan et pistes …

Université de Toulouse

- **L'analyse du WCET d'applications parallèles nécessite :**
  - de comprendre la structure de l'application
    - analyse des dépendances
  - d'analyser les synchronisations
    - où sont-elles ?
    - combien de threads concurrents?
    - sections critiques ?



- **Pour une analyse automatique :**
  - extraire les informations requises (structure + synchronisations)
    - analyse du code source/objet
    - annotations de l'utilisateur
  - piloter le calcul du WCET

- **Recommandations pour la parallélisation**
  - règles, patrons, …

En résumé …

Calcul de WCET et prévisibilité

Université de Toulouse

- **Techniques de calcul de WCET**
  - analyse statique (applications critiques)
  - complexe, coûteux
  - attention au choix du processeur cible !

- **Limiter les interactions entre tâches et activités !**
  - cible **monocœur** :
    - entrées/sorties ?
    - interruptions ?
    - préemption ?
  - cible **multicœur** et/ou **multiflot** :
    - partage de ressources ?
    - communications et synchronisations ?

Pour en savoir plus …

Université de Toulouse

- **Conférences et workshops**
  - Workshop on WCET analysis (adossé à la conférence ECRTS)
  - Workshop PPES = Bringing Theory to Practice: Predictability and Performance in Embedded Systems
  - Sessions “Timing Analysis” : ECRTS, RTSS, RTAS, RTCSA, …
- **Revue**
  - Journal on Real-Time Systems (Springer)
- **WCET Tool Challenge**
  - www.mrtc.mdh.se/projects/WCC
- **Action ACTRISS du GdR ASR**
  - site web à venir
  - réunions thématiques