



# How to address certification for multi-core based IMA platforms : current status and potential solutions

*Rudolf Fuchsen, SYSGO AG, Klein-Winternheim, Germany [1]*

Présentation de l'article  
Antonin Godard

École Polytechnique de Montréal  
INF6600

# Introduction



# Contexte

---

L'industrie aérienne évolue rapidement...



# Contexte

---

L'industrie aérienne évolue rapidement...

- de plus en plus de sûreté assurée par l'électronique



# Contexte

---

L'industrie aérienne évolue rapidement...

- de plus en plus de sûreté assurée par l'électronique
- contraintes environnementales



# Contexte

---

L'industrie aérienne évolue rapidement...

- de plus en plus de sûreté assurée par l'électronique
  - contraintes environnementales
- électronique plus petite



# Contexte

---

L'industrie aérienne évolue rapidement...

- de plus en plus de sûreté assurée par l'électronique
- contraintes environnementales
- électronique plus petite
- qui consomme moins



# Contexte

---

L'industrie aérienne évolue rapidement...

- de plus en plus de sûreté assurée par l'électronique
- contraintes environnementales
- électronique plus petite
- qui consomme moins
- plus légère





## Contexte

L'industrie aérienne évolue rapidement...

- de plus en plus de sûreté assurée par l'électronique
- contraintes environnementales
- électronique plus petite
- qui consomme moins
- plus légère

### Résultat :

Toutes ces contraintes demandent un partitionnement et des niveaux de sûreté **bien définis**



## Expliquons le titre...

How to address certification for multi-core based IMA platforms : current status and potential solutions

De modules LRU à un AMI :

- *Line Replacable Unit* : module d'un avion effectuant une fonction spécifique, et qui est **remplaçable** rapidement.



## Expliquons le titre...

How to address certification for multi-core based IMA platforms : current status and potential solutions

De modules LRU à un AMI :

- *Line Replacable Unit* : module d'un avion effectuant une fonction spécifique, et qui est **remplaçable** rapidement.



- *Avionique Modulaire Intégré* : système **temps réel** qui permet de rassembler plusieurs modules de calcul permettant de réaliser des fonctions différentes, à plusieurs **niveaux de criticité**



## Expliquons le titre...

How to address certification for multi-core based IMA platforms : current status and potential solutions

De modules LRU à un AMI :

- *Line Replacable Unit* : module d'un avion effectuant une fonction spécifique, et qui est **remplaçable** rapidement.



- *Avionique Modulaire Intégré* : système **temps réel** qui permet de rassembler plusieurs modules de calcul permettant de réaliser des fonctions différentes, à plusieurs **niveaux de criticité**

Ici, on parlera d'AMI **multi-coeurs**



# Expliquons le titre...

How to address certification for multi-core based IMA platforms : current status and potential solutions



LRU



IMA

# Expliquons le titre...

How to address certification for multi-core based IMA platforms : current status and potential solutions

## Certification

Méthodologie formelle de tests et documentation des mécanismes de sécurité, technique ou non-technique, dans un environnement donné et en utilisant des critères établis



## Expliquons le titre...

How to address certification for multi-core based IMA platforms : current status and potential solutions

### Certification

Méthodologie formelle de tests et documentation des mécanismes de sécurité, technique ou non-technique, dans un environnement donné et en utilisant des critères établis

### Système certifié en avionique

Système qui répond à des **standards/normes** de sûreté et de sécurité.



## Expliquons le titre...

How to address certification for multi-core based IMA platforms : current status and potential solutions

La certification est délivrée par des agences.



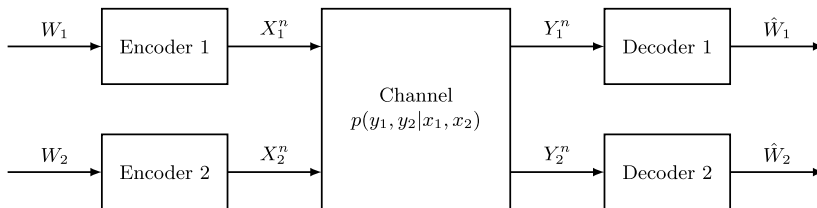
# Federal Aviation Administration





# Canaux d'interférences

Canaux d'interférences *software* et *hardware* entre les partitions dans un AMI multi-coeurs



Source : Wikipedia [3]



# Problématique

---

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AMI mono-coeur sur un AMI multi-coeurs ?

Enjeux :



# Problématique

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AMI mono-coeur sur un AMI multi-coeurs ?

Enjeux :

- Performance CPU



# Problématique

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AMI mono-coeur sur un AMI multi-coeurs ?

Enjeux :

- Performance CPU
- Ressources mémoires



# Problématique

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AML mono-cœur sur un AML multi-cœurs ?

Enjeux :

- Performance CPU
- Ressources mémoires
- Bande passante I/O



# Problématique

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AML mono-coeur sur un AML multi-coeurs ?

Enjeux :

- Performance CPU
- Ressources mémoires
- Bande passante I/O
- Sûreté suffisante



# Problématique

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AML mono-cœur sur un AML multi-cœurs ?

Enjeux :

- Performance CPU
- Ressources mémoires
- Bande passante I/O
- Sûreté suffisante
- Comportement déterministe et prédisible



# Problématique

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AML mono-coeur sur un AML multi-coeurs ?

Enjeux :

- Performance CPU
- Ressources mémoires
- Bande passante I/O
- Sûreté suffisante
- Comportement déterministe et prédisible
- Partitionnement temps CPU et de ressources sûr





# Problématique

## Problématique de l'article

Peut-on atteindre une meilleure performance et le même niveau de déterminisme qu'un AML mono-coeur sur un AML multi-coeurs ?

Enjeux :

- Performance CPU
- Ressources mémoires
- Bande passante I/O
- Sûreté suffisante
- Comportement déterministe et prédisible
- Partitionnement temps CPU et de ressources sûr
- Certifiable



Que prendre en compte pour la certification ?



# Certification de processeur multi-coeurs

---

Jusqu'à aujourd'hui :



# Certification de processeur multi-coeurs

---

Jusqu'à aujourd'hui :

- processeur mono-coeur



# Certification de processeur multi-coeurs

---

Jusqu'à aujourd'hui :

- processeur mono-coeur
- architecture RISC



# Certification de processeur multi-coeurs

---

Jusqu'à aujourd'hui :

- processeur mono-coeur
- architecture RISC
- beaucoup de *In Service Experience*



# Certification de processeur multi-coeurs

---

Jusqu'à aujourd'hui :

- processeur mono-coeur
- architecture RISC
- beaucoup de *In Service Experience*

Avec les nouveaux AMI :



# Certification de processeur multi-cœurs

---

Jusqu'à aujourd'hui :

- processeur mono-cœur
- architecture RISC
- beaucoup de *In Service Experience*

Avec les nouveaux AMI :

- processeurs multi-cœur





## Certification de processeur multi-cœurs

---

Jusqu'à aujourd'hui :

- processeur mono-cœur
- architecture RISC
- beaucoup de *In Service Experience*

Avec les nouveaux AMI :

- processeurs multi-cœur
- mêmes fonctionnalités que mono-cœur



## Certification de processeur multi-cœurs

---

Jusqu'à aujourd'hui :

- processeur mono-cœur
- architecture RISC
- beaucoup de *In Service Experience*

Avec les nouveaux AMI :

- processeurs multi-cœur
- mêmes fonctionnalités que mono-cœur
- + communication inter-processeurs / parallélisation / ajout de caches...



## Certification de processeur multi-cœurs

---

Jusqu'à aujourd'hui :

- processeur mono-cœur
- architecture RISC
- beaucoup de *In Service Experience*

Avec les nouveaux AMI :

- processeurs multi-cœur
- mêmes fonctionnalités que mono-cœur
- + communication inter-processeurs / parallélisation / ajout de caches...
- peu de documentation car technologies récentes



## Certification de processeur multi-cœurs

Jusqu'à aujourd'hui :

- processeur mono-cœur
- architecture RISC
- beaucoup de *In Service Experience*

Avec les nouveaux AMI :

- processeurs multi-cœur
  - mêmes fonctionnalités que mono-cœur
  - + communication inter-processeurs / parallélisation / ajout de caches...
  - peu de documentation car technologies récentes
- difficiles à certifier



# Détection d'erreurs et correction

---

- Les processeurs multi-coeurs sont plus sensibles aux interférences



## Détection d'erreurs et correction

---

- Les processeurs multi-coeurs sont plus sensibles aux interférences
- Besoin d'une protection ECC (Error Correction Code)



## Détection d'erreurs et correction

---

- Les processeurs multi-coeurs sont plus sensibles aux interférences
- Besoin d'une protection ECC (Error Correction Code)
- Redondance afin de minimiser les erreurs



## Partitionnement du temps et des ressources

---

- Pas de différence majeure pour le partitionnement des ressources entre mono et multi-coeurs





## Partionnement du temps et des ressources

---

- Pas de différence majeure pour le partitionnement des ressources entre mono et multi-coeurs
- Pour le partitionnement du temps c'est plus compliqué :



## Partitionnement du temps et des ressources

- Pas de différence majeure pour le partitionnement des ressources entre mono et multi-coeurs
- Pour le partitionnement du temps c'est plus compliqué :
  - Sur un mono-coeur : un seul thread à la fois, qui peut être interrompu



## Partitionnement du temps et des ressources

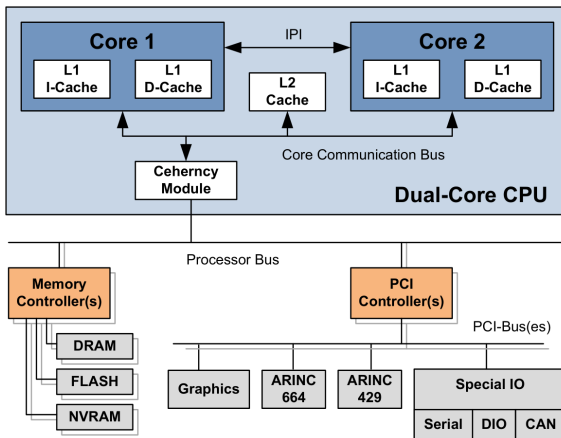
- Pas de différence majeure pour le partitionnement des ressources entre mono et multi-coeurs
- Pour le partitionnement du temps c'est plus compliqué :
  - Sur un mono-coeur : un seul thread à la fois, qui peut être interrompu
  - Sur un multi-coeur, exécution concurrente donc interférences entre processus



## Canaux d'interférences physiques

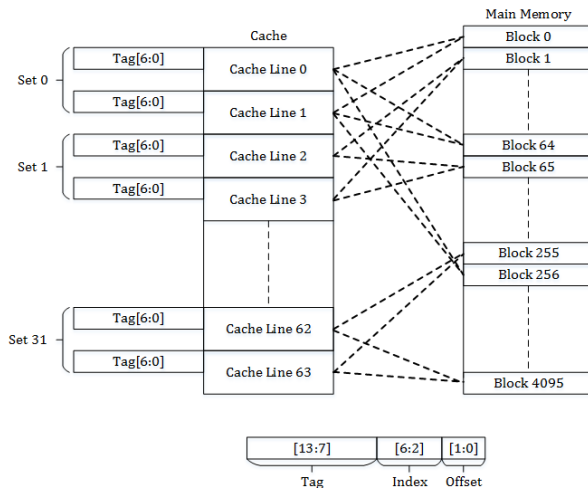


# Architecture d'un processeur AML multi-coeur



Architecture d'un CPU deux coeurs [1]

# Caches



Memory Size = 16Kbytes  
 Memory Block Size = 4 bytes  
 Cache Size = 256 bytes  
 Block Size = 4 bytes  
 Associativity = 2  
 Number of Sets = 32

## Exemple de cache set-associatif [2]

# Partage de cache

---

Comparaison :



# Partage de cache

---

Comparaison :

- Processeur Intel :
  - Cache L1 64Kbyte par processeur
  - Cache L2 2Mbyte **partagée**





# Partage de cache

---

## Comparaison :

- Processeur Intel :
  - Cache L1 64Kbyte par processeur
  - Cache L2 2Mbyte **partagée**
- Processeur AMD :
  - Cache L1 128Kbyte par processeur
  - Cache L2 512Kbyte **par processeur**



## Partage de cache

---

### Comparaison :

- Processeur Intel :
    - Cache L1 64Kbyte par processeur
    - Cache L2 2Mbyte **partagée**
  - Processeur AMD :
    - Cache L1 128Kbyte par processeur
    - Cache L2 512Kbyte **par processeur**
- Input : données de taille variables que les processeurs lisent ou modifient
- On mesure le throughput



## Partage de cache

---

### Comparaison :

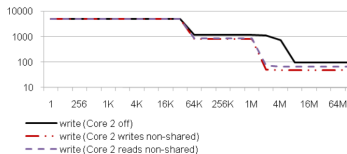
- Processeur Intel :
    - Cache L1 64Kbyte par processeur
    - Cache L2 2Mbyte **partagée**
  - Processeur AMD :
    - Cache L1 128Kbyte par processeur
    - Cache L2 512Kbyte **par processeur**
- Input : données de taille variables que les processeurs lisent ou modifient
- On mesure le throughput



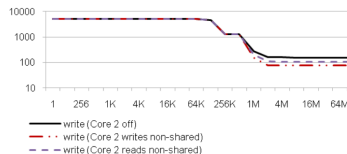
# Partage de cache

	L1	L1 + L2	Cache L2 partagée
Intel	64Kbytes	2Mbytes	Oui
AMD	128Kbytes	750Kbytes	Non

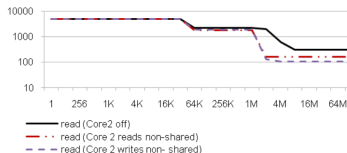
Intel (Copy Back Mode) / Write Non-Shared



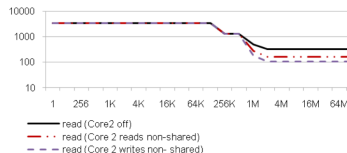
AMD (Copy Back Mode) / Write Non-Shared



Intel (Copy Back Mode) / Read Non-Shared



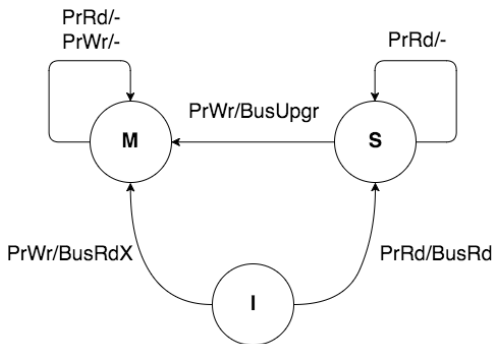
AMD (Copy Back Mode) / Read Non-Shared



Résultats [1]

## Cohérence de cache

- Protocole MSI (Modified, Shared, Invalid)



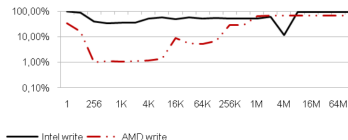
Transitions entre états d'une ligne de cache [4]



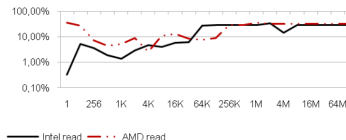
# Cohérence de cache

	L1	L1 + L2	Cache L2 partagée
Intel	64Kbytes	2Mbytes	Oui
AMD	128Kbytes	750Kbytes	Non

Intel/AMD Performance Impact on Write



Intel/AMD Performance Impact on Read



Résultats [1]

# Bus, I/O Devices, et Interruptions

---

- Bande passante de bus partagée, performance réduite si les données sont trop grandes ( $>$  taille de cache)



## Bus, I/O Devices, et Interruptions

---

- Bande passante de bus partagée, performance réduite si les données sont trop grandes ( $>$  taille de cache)
- I/O Devices : une requête à la fois donc bien choisir les bus de communication (PCI, etc)





## Bus, I/O Devices, et Interruptions

- Bande passante de bus partagée, performance réduite si les données sont trop grandes ( $>$  taille de cache)
- I/O Devices : une requête à la fois donc bien choisir les bus de communication (PCI, etc)
- Interruptions : un processeur doit avertir l'autre s'il existe un seul même ligne d'interruption



## Canaux d'interférence logiciels



# Concepts

---

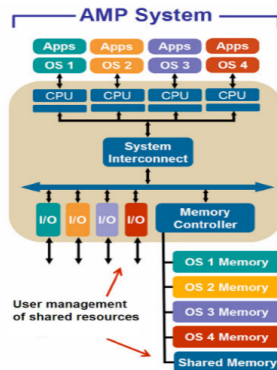
- la configuration software (partitions, etc.) doit s'adapter au processeur
- les modules AMI doivent distribuer un bande passante **élevée** et **constante** aux différentes applications en fonction du partitionnement de temps



# Concepts

Approche AMP (asymmetric multiprocessing) :

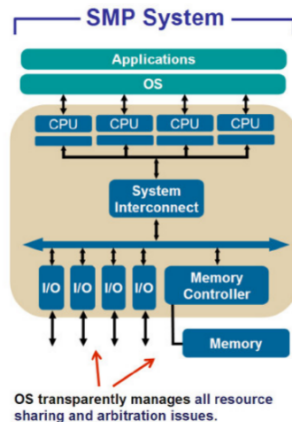
- + le software n'a pas besoin de savoir que le processeur est multi-cœur
- + software optimisé pour la tâche associée
- instances certifiées au niveau maximum car elles ont un processeur exclusif
- partitionnement ressources simplifié
- synchronisation entre tâche plus complexe



# Concepts

Approche SMP (symmetric multiprocessing) :

- + un seul OS / une seule configuration
- + isolation totale des tâches critiques
- meilleure balance de charge (load-balancing)
- software plus complexe
- faux partages en cache
- plus d'interférences



# Concepts

---

- Parallélisation par threads :
  - + plus de contrôle sur les tâches
  - données partagées donc faux partage possible
- Parallélisation par instruction :
  - + OpenMP permet de répartir un calcul sur plusieurs coeurs
  - données partagées donc faux partage possible

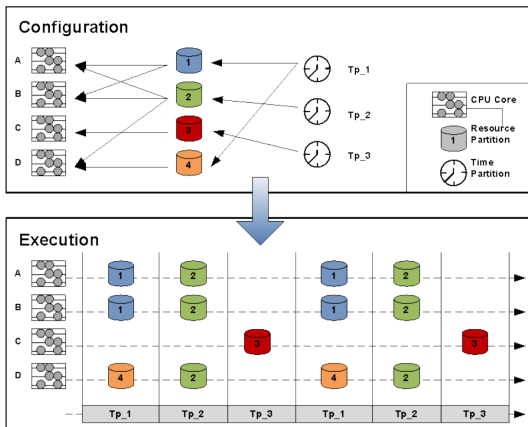


## Approche SMP pour les AMI

- De plus en plus d'application qui demandent une haute performance avec un bas niveau de criticité
- Mais comment faire rouler une application de haute criticité (sûreté) à côté d'une application de basse criticité sans avoir d'interférences ?
- Une solution : PikeOS



# Approche SMP pour les AMI



Exemple de configuration et d'exécution avec PikeOS [1]



# Conclusion



# Conclusion

---

- Les CPUs multi-coeurs vont devenir une nécessité dans l'avionique



# Conclusion

---

- Les CPUs multi-coeurs vont devenir une nécessité dans l'avionique
- Besoin de *In Service Experience* sur des systèmes **moins critiques**



# Conclusion

---

- Les CPUs multi-coeurs vont devenir une nécessité dans l'avionique
- Besoin de *In Service Experience* sur des systèmes **moins critiques**
- Besoin d'un design réfléchi et examiné au plus près



# Conclusion

---

- Les CPUs multi-coeurs vont devenir une nécessité dans l'avionique
- Besoin de *In Service Experience* sur des systèmes **moins critiques**
- Besoin d'un design réfléchi et examiné au plus près
- Choix de processeur



# Conclusion

---

- Les CPUs multi-coeurs vont devenir une nécessité dans l'avionique
  - Besoin de *In Service Experience* sur des systèmes **moins critiques**
  - Besoin d'un design réfléchi et examiné au plus près
  - Choix de processeur
- Systèmes plus sûrs



# Conclusion

---

- Les CPUs multi-coeurs vont devenir une nécessité dans l'avionique
  - Besoin de *In Service Experience* sur des systèmes **moins critiques**
  - Besoin d'un design réfléchi et examiné au plus près
  - Choix de processeur
- Systèmes plus sûrs

Merci !



## References |

---

[1] R. Fuchsen.

How to address certification for multi-core based ima platforms : Current status and potential solutions.

*In 29th Digital Avionics Systems Conference, pages 5.E.3–1–5.E.3–11, Oct 2010.*

[2] Wikipedia.

Cache placement policies — Wikipedia, the free encyclopedia.

<http://en.wikipedia.org/w/index.php?title=Cache%20placement%20policies&oldid=919393503>, 2019.

[Online; accessed 16-November-2019].





## References II

---

[3] Wikipedia.

Interference channel — Wikipedia, the free encyclopedia.

<http://en.wikipedia.org/w/index.php?title=Interference%20channel&oldid=868954220>, 2019.  
[Online; accessed 12-November-2019].

[4] Wikipedia.

MSI protocol — Wikipedia, the free encyclopedia.

<http://en.wikipedia.org/w/index.php?title=MSI%20protocol&oldid=905172167>, 2019.  
[Online; accessed 16-November-2019].

