Conception et réalisation d'un système d'exploitation pour processeurs "many-cores"

Ghassan Almaless

Soutenance de thèse, 27 février 2014, UPMC

UPMC SORBONNE UNIVERSITÉS

LIP6 Dpt : Systèmes Embarqués sur Puce

Plan

Introduction : Puissance de calcul et processeurs Many-Cores

1

Plan

Introduction : Puissance de calcul et processeurs Many-Cores

Problématique : Besoin d'un système d'exploitation adapté pour Many-Cores

Plan

Introduction : Puissance de calcul et processeurs Many-Cores

Problématique : Besoin d'un système d'exploitation adapté pour Many-Cores

État de l'art

Plan

Introduction : Puissance de calcul et processeurs Many-Cores

Problématique : Besoin d'un système d'exploitation adapté pour Many-Cores

État de l'art

Contributions Majeures

4

Plan

Introduction : Puissance de calcul et processeurs Many-Cores

Problématique : Besoin d'un système d'exploitation adapté pour Many-Cores

État de l'art

Contributions Majeures

Expérimentation, Résultats et Analyse

Plan

Introduction : Puissance de calcul et processeurs Many-Cores

Problématique : Besoin d'un système d'exploitation adapté pour Many-Cores

État de l'art

Contributions Majeures

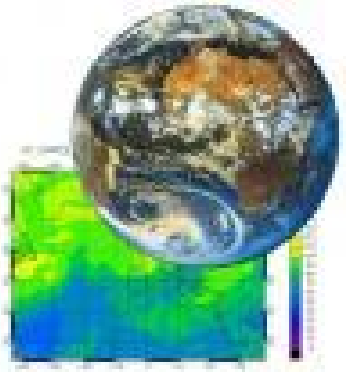
Expérimentation, Résultats et Analyse

Conclusions et Perspectives

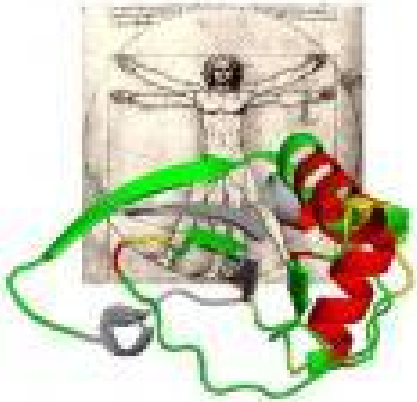
Introduction

Puissance de calcul et l'émergence des processeurs Many-Cores

Forte demande en puissance de calcul

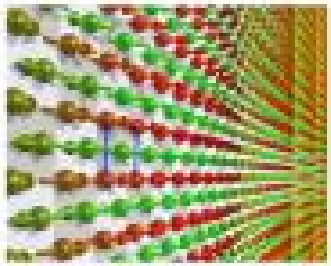


Environnement
Climatologie

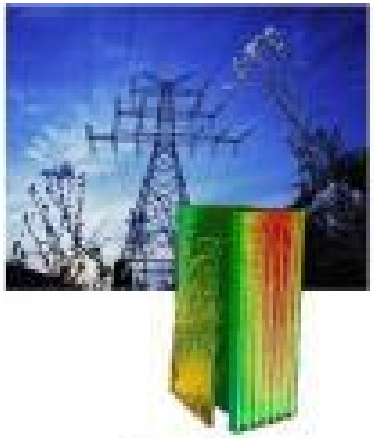


Médecine
Biologie

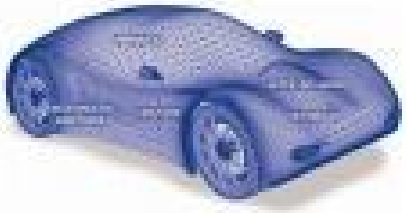
Aéronautique
Aérospatial



Matériaux
Spintronique
Nanosciences



Énergie
Physique des plasmas
Piles à combustible



Automobile

Finance

Centrales électriques
de demain

Source : initiative-hpc-pme.org

La puissance de calcul est à prévoir dans tous les domaines

Source : Intel Datacenter Day, 2013

Problème : les leviers traditionnels de performances ont atteint leurs limites

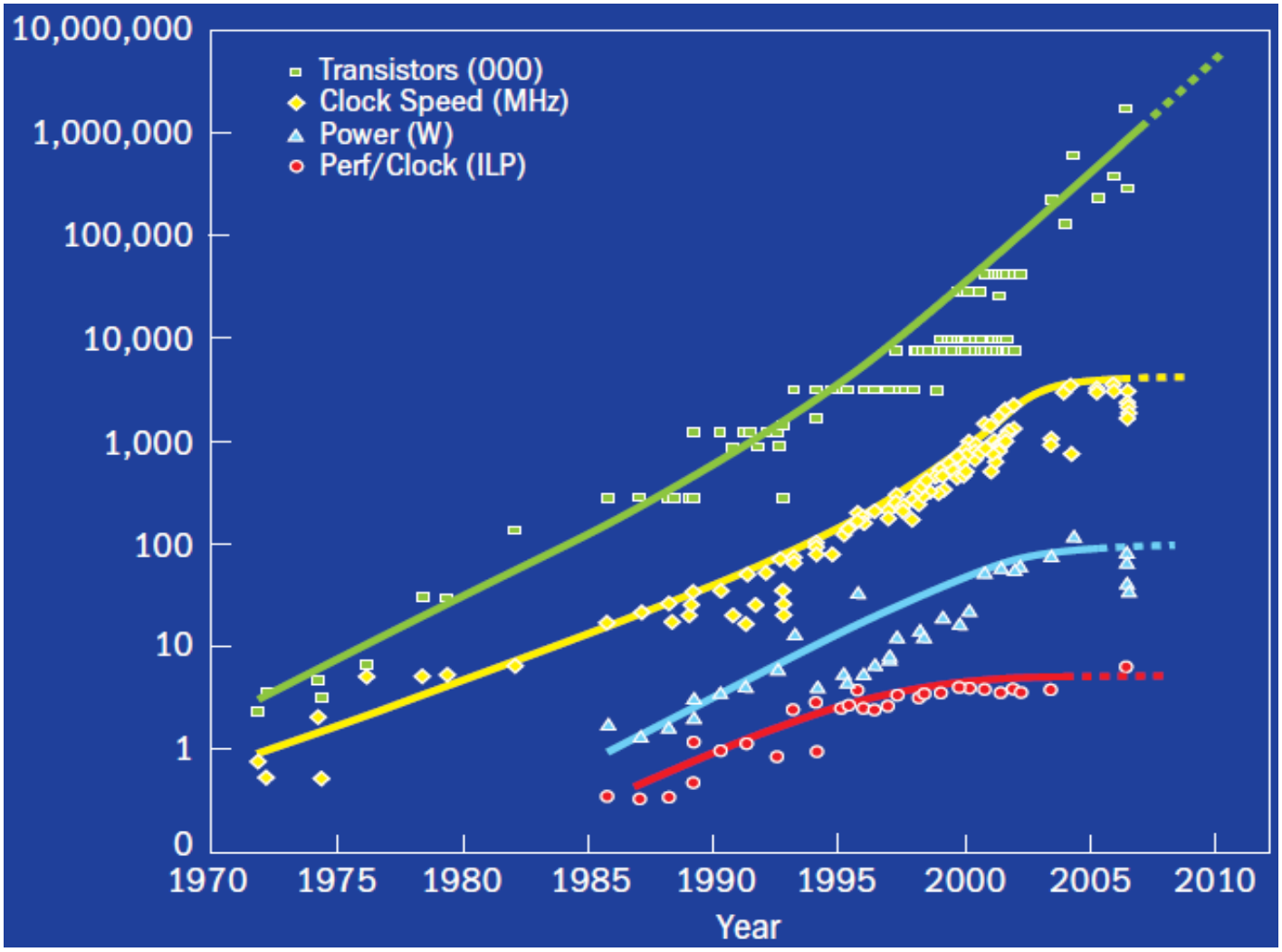
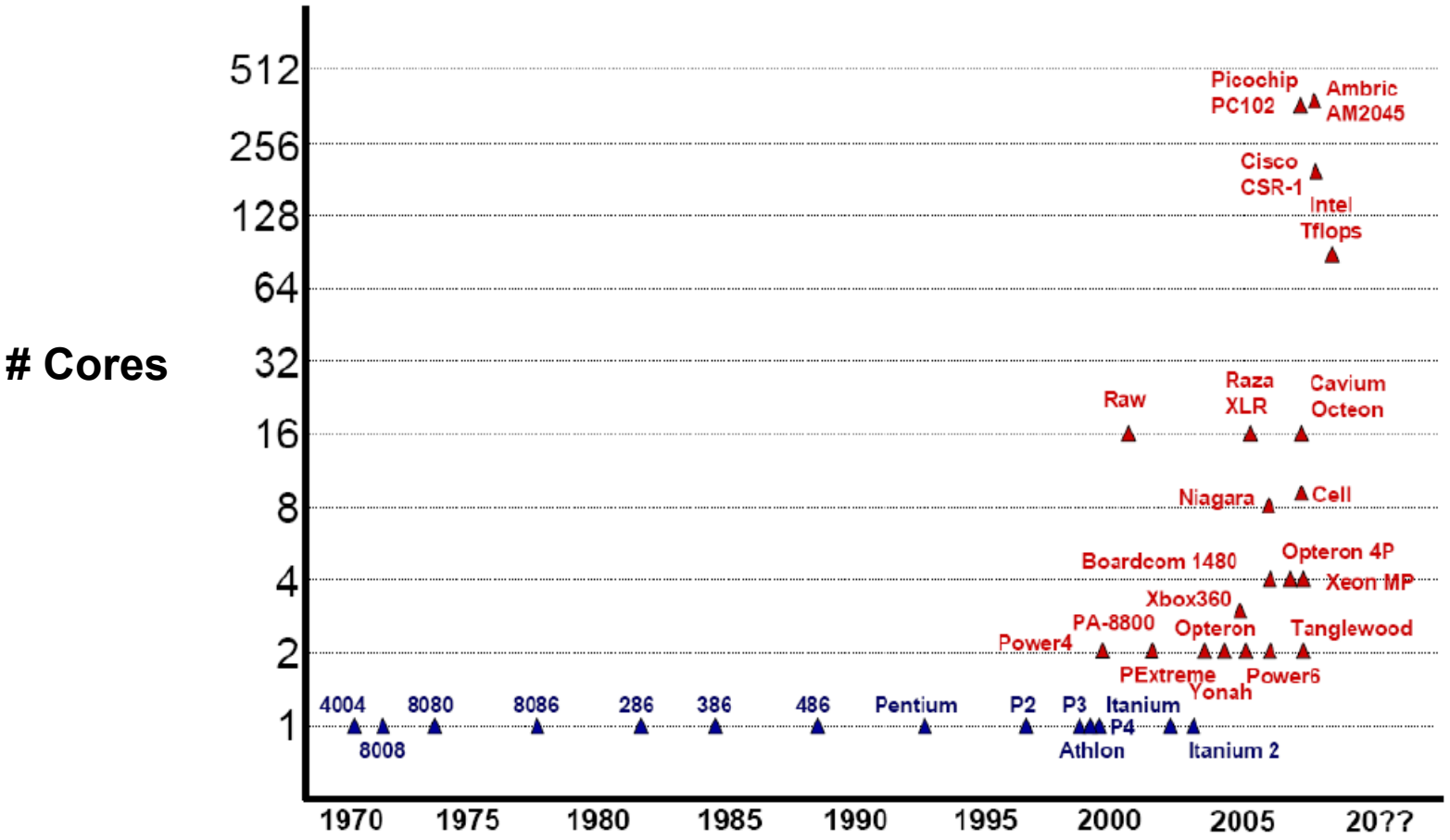


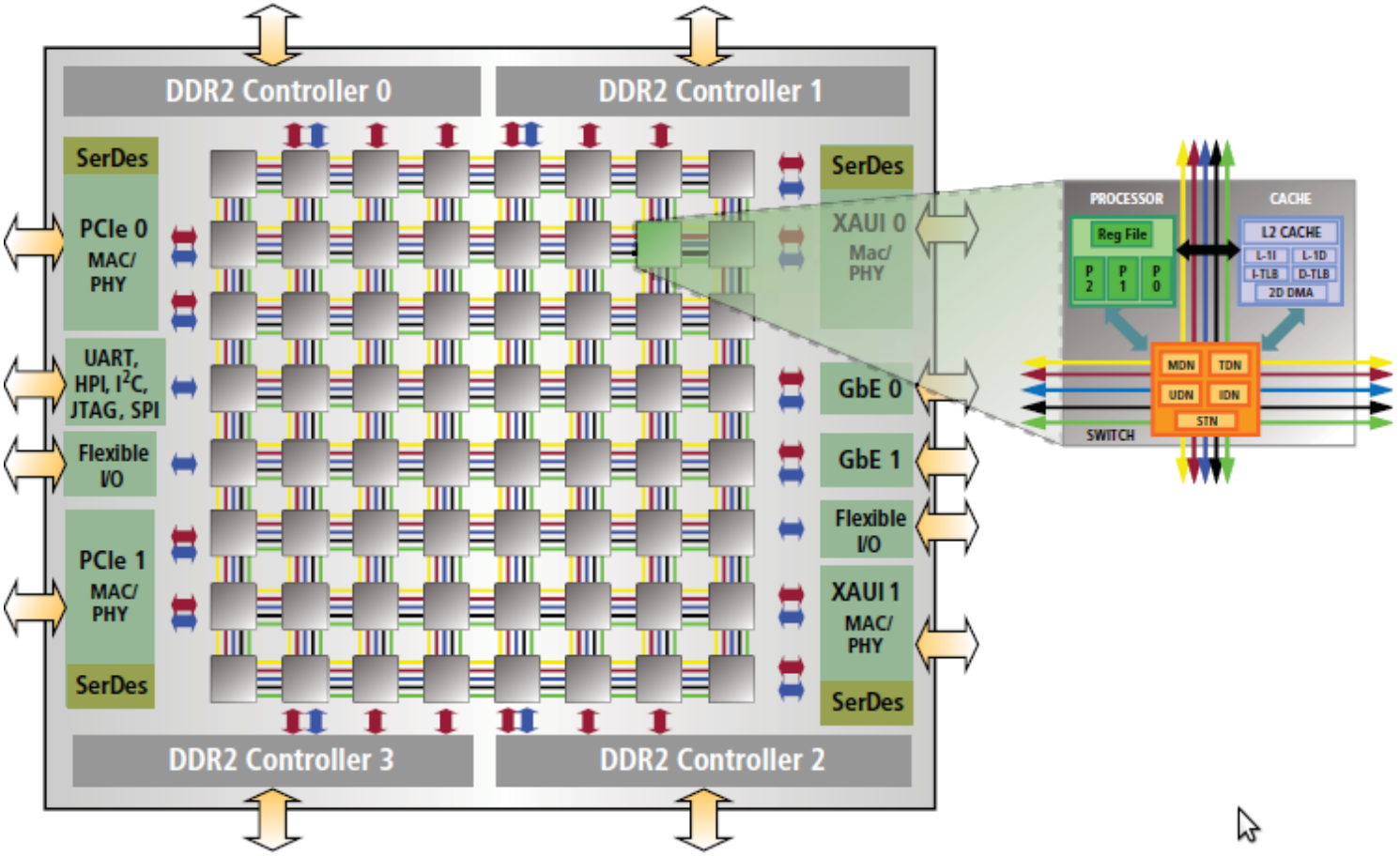
Illustration: A. TOVEY, Source: D. Paterson, UC-Berkeley

Réponse : intégrer plus de coeurs de calcul sur la même puce (many-cores)



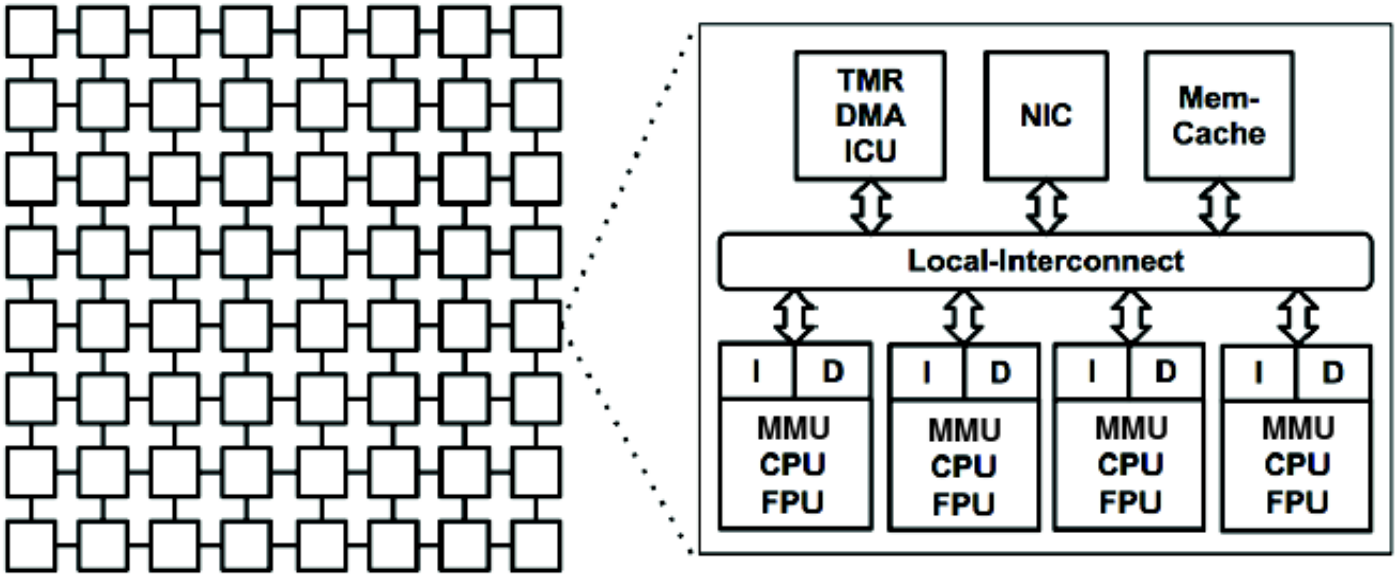
Source: Saman Amarasinghe, MIT (6.189 2007, Lec1)

Tilera : Un seul processeur à 100 cores



Tile64, Source: Tilera website (www.tilera.com)

LIP6 : Processeur TSAR à +1024 cores



- Un projet européen piloté par Bull
- Conçu par le LIP6 en coopération avec Bull
- Micro-architecture simplifiée pour les cores
- Les clusters sont inter-connectés par un NoC 2D

Plan

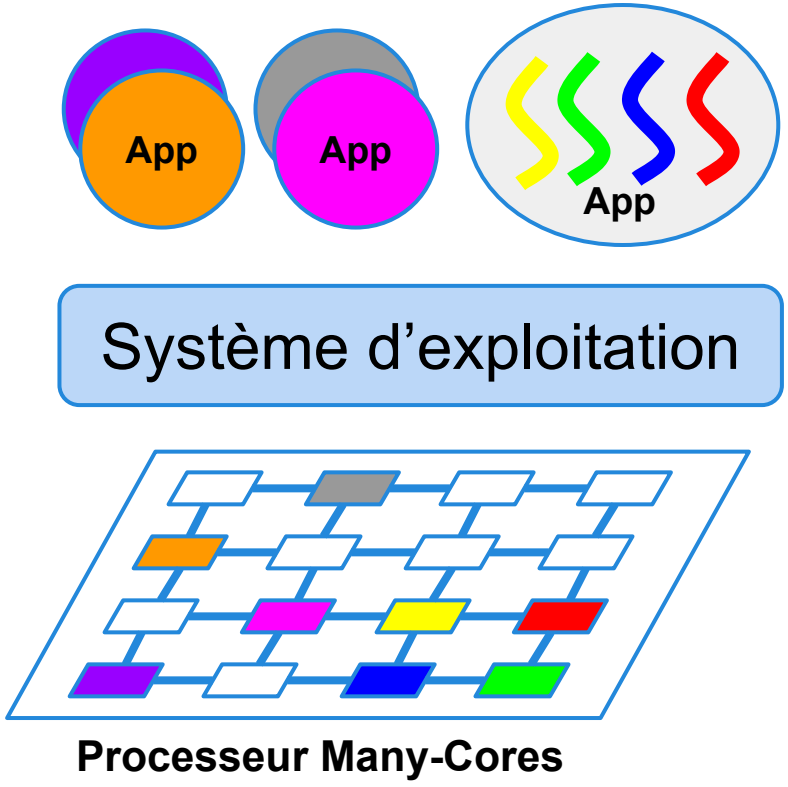
Introduction : Puissance de calcul et les processeurs Many-Cores

Problématique : Besoin d'un système d'exploitation adapté pour Many-Cores

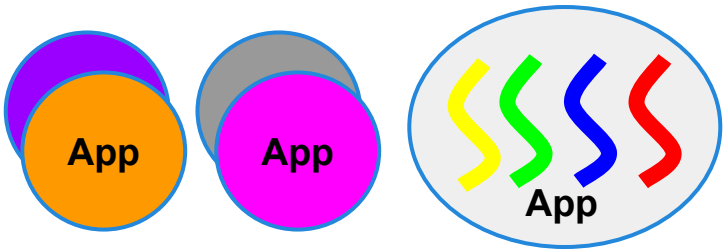
2

État de l'art

Besoin d'un système d'exploitation adapté aux many-cores

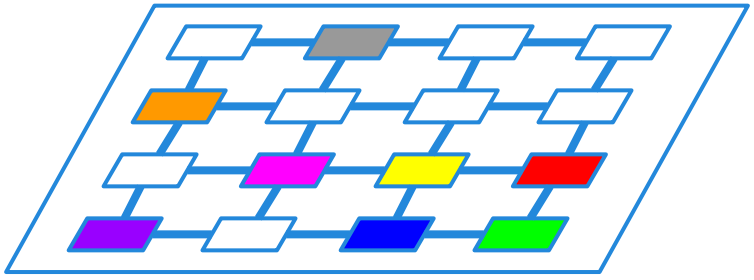


Besoin d'un système d'exploitation adapté aux many-cores



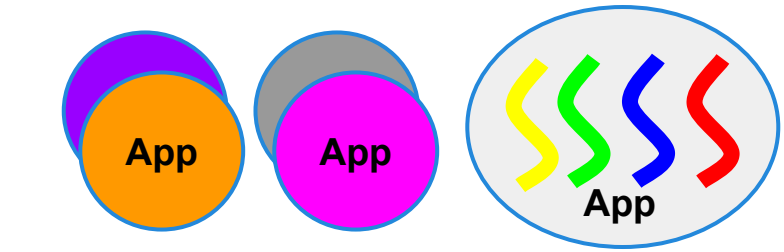
Système d'exploitation

Assurer une compatibilité avec l'existant

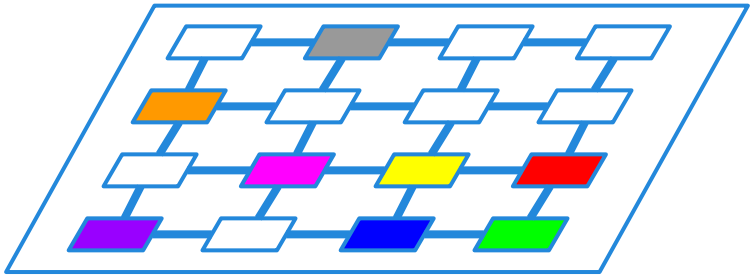


Processeur Many-Cores

Besoin d'un système d'exploitation adapté aux many-cores



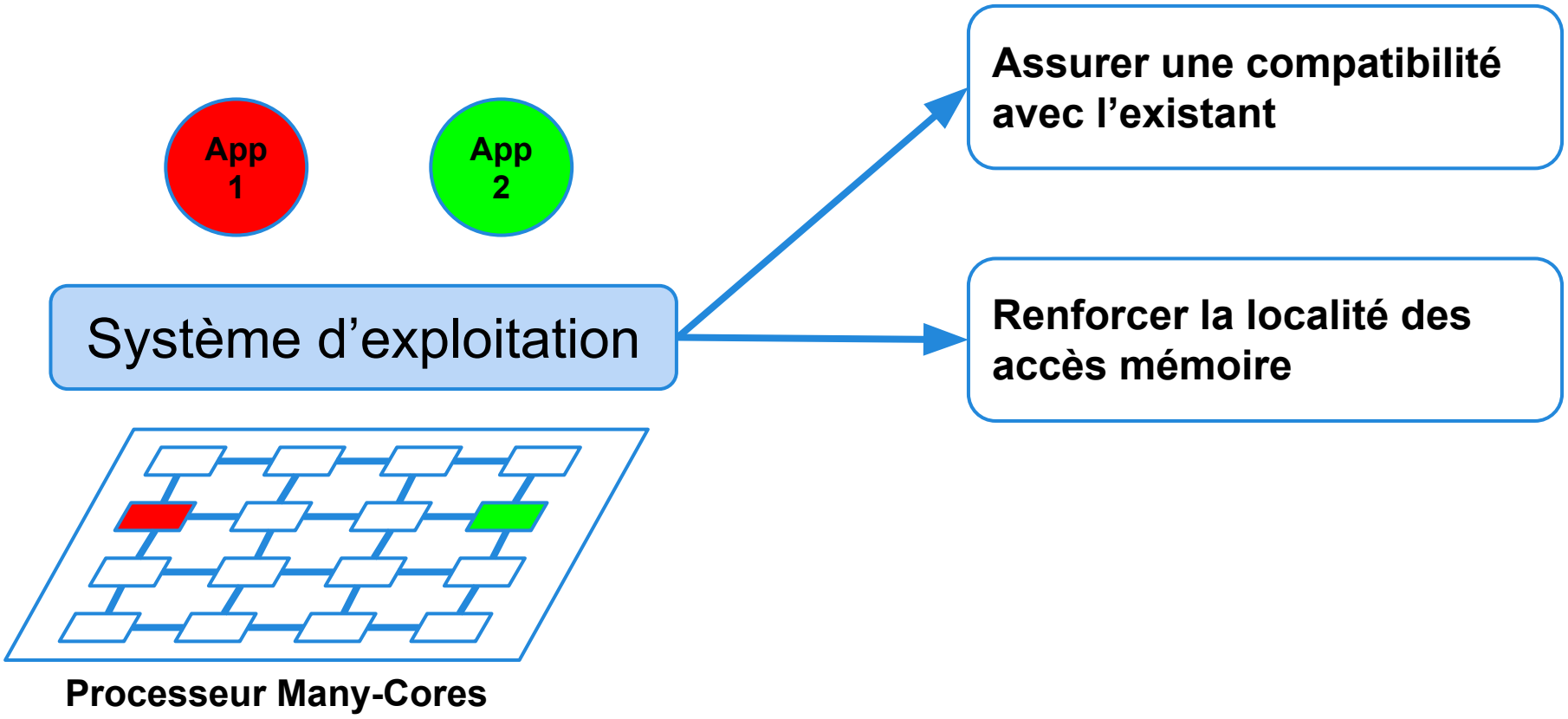
Système d'exploitation



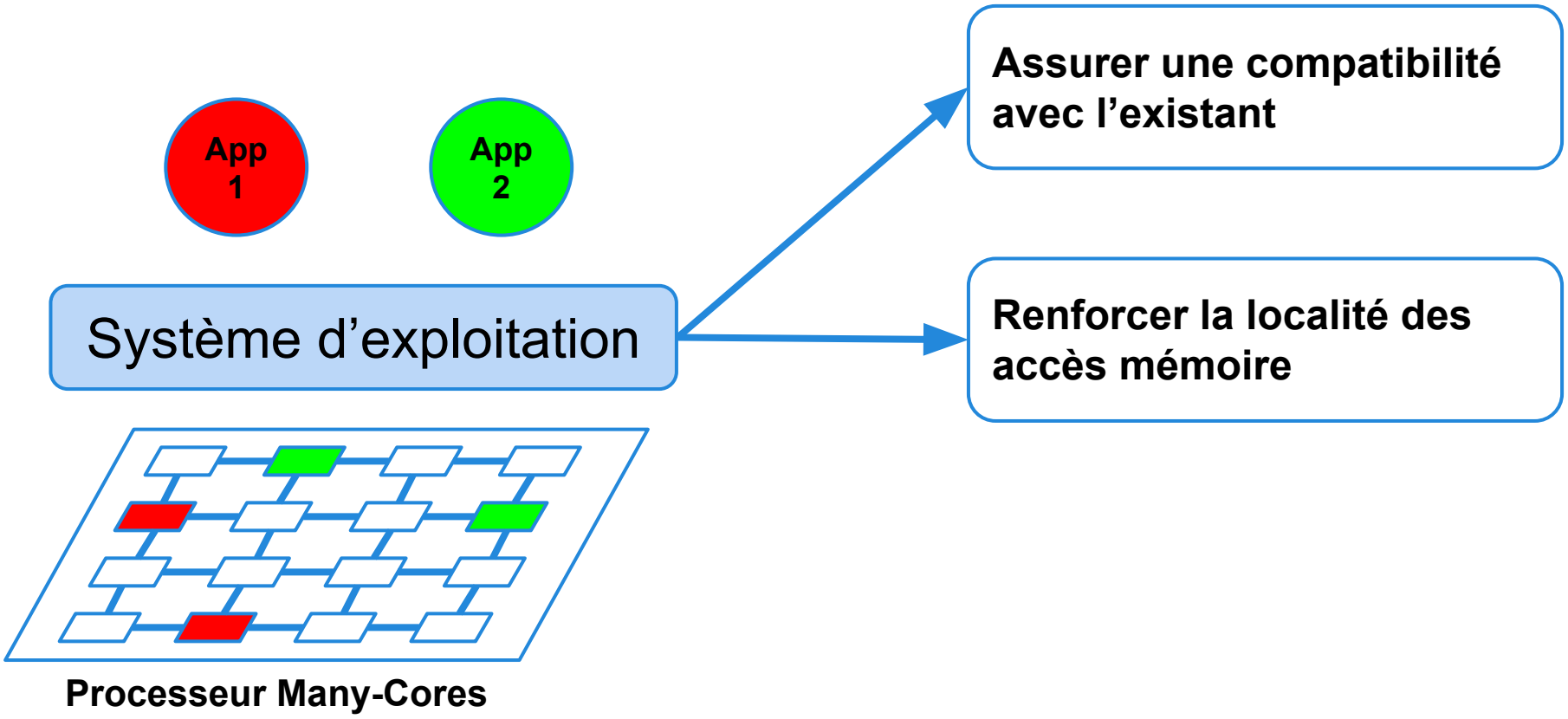
Processeur Many-Cores

- **Assurer une compatibilité avec l'existant**
- **Renforcer la localité des accès mémoire**

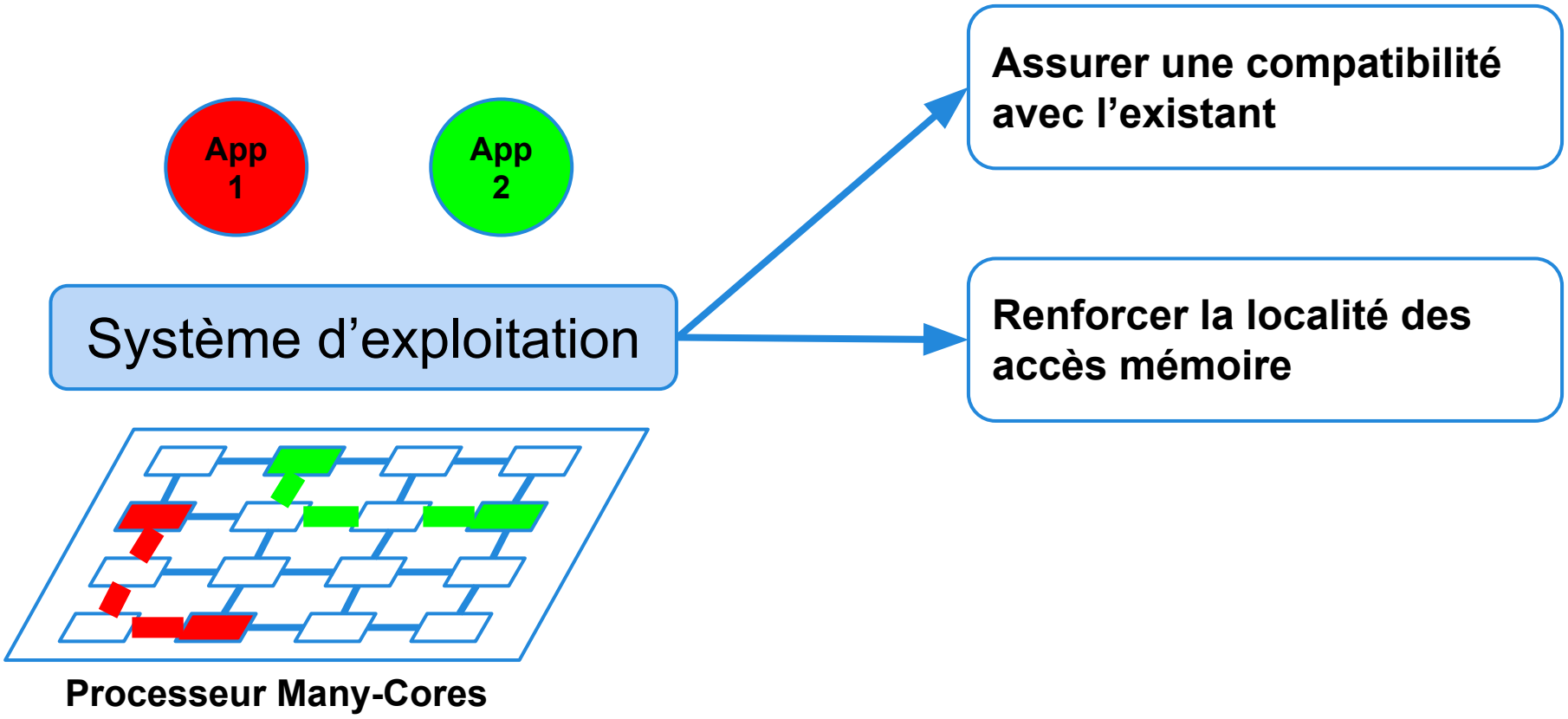
Besoin d'un système d'exploitation adapté aux many-cores



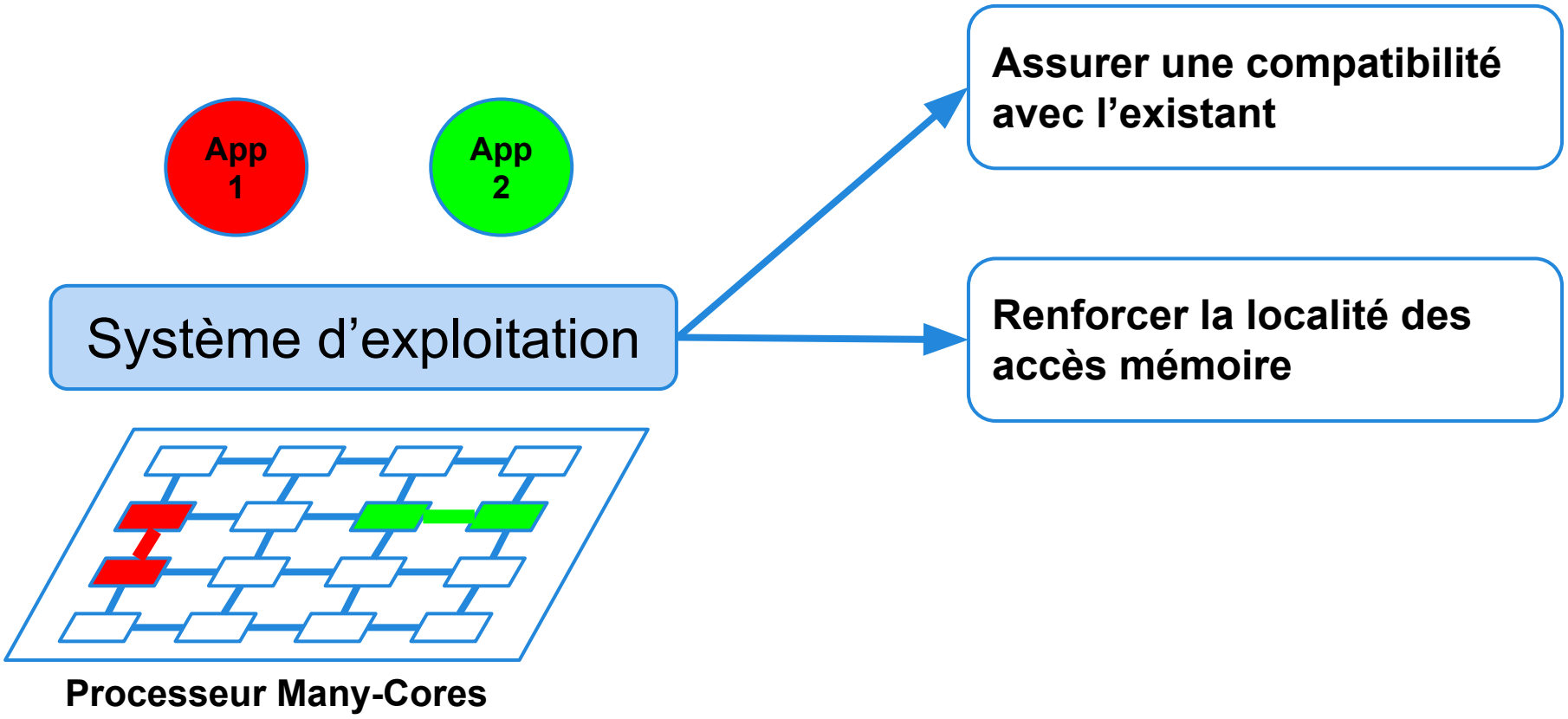
Besoin d'un système d'exploitation adapté aux many-cores



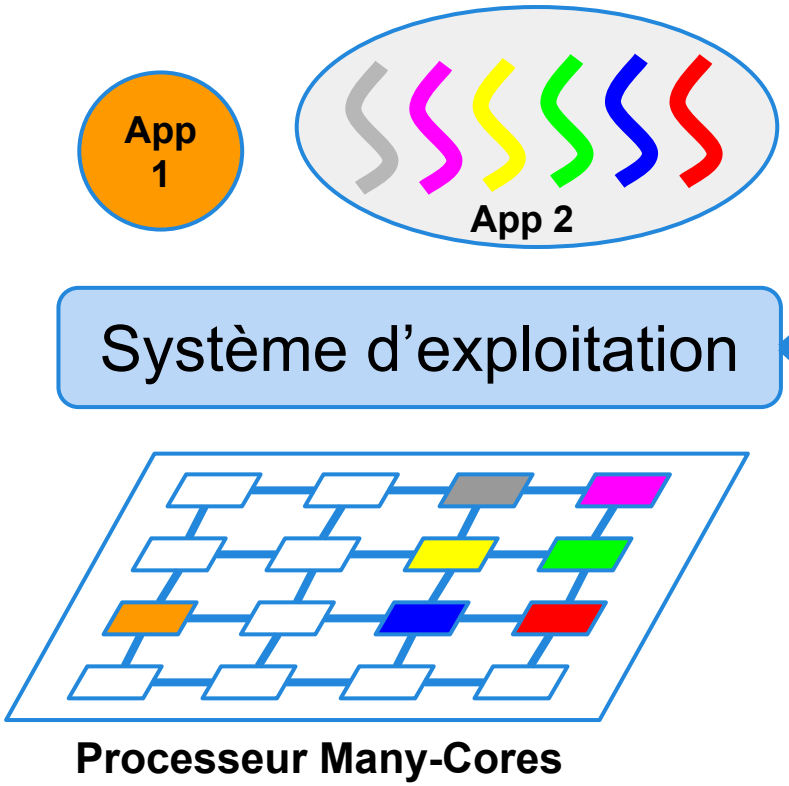
Besoin d'un système d'exploitation adapté aux many-cores



Besoin d'un système d'exploitation adapté aux many-cores

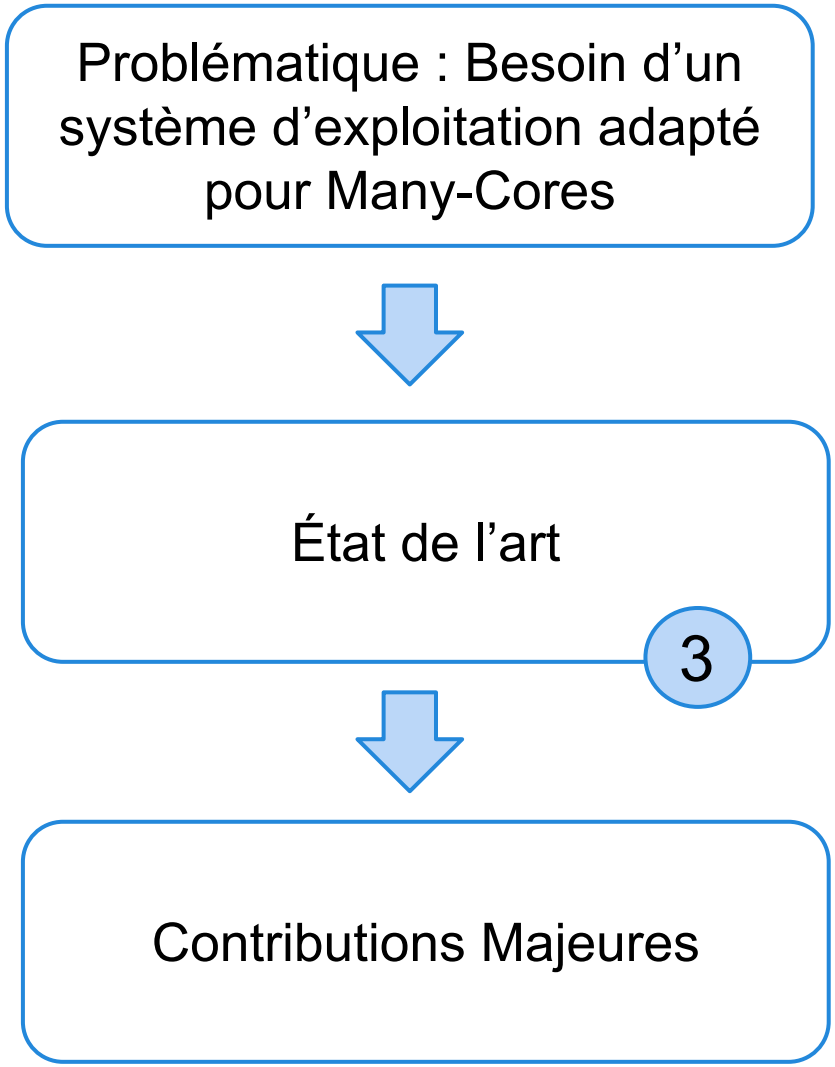


Besoin d'un système d'exploitation adapté aux many-cores



- **Assurer une compatibilité avec l'existant**
- **Renforcer la localité des accès mémoire**
- **Exécuter efficacement des applications multi-threads**

Plan


Problématique : Besoin d'un système d'exploitation adapté pour Many-Cores
État de l'art
3
Contributions Majeures

État de l’art

- Approche de conception traditionnelle, dite **monolithique**

- Approches alternatives :

État de l’art

- Approche de conception traditionnelle, dite **monolithique**
  - Un OS unique qui gère toute la machine

- Approches alternatives :

État de l’art

- Approche de conception traditionnelle, dite **monolithique**
  - Un OS unique qui gère toute la machine
  - Suppose que la mémoire est partagée et cohérente

- Approches alternatives :

État de l’art

- Approche de conception traditionnelle, dite **monolithique**
  - Un OS unique qui gère toute la machine
  - Suppose que la mémoire est partagée et cohérente
  - Représente la norme dans l’industrie informatique (Linux, BSD, Solaris, Windows NT/XP/2000)

- Approches alternatives :

État de l'art

- Approche de conception traditionnelle, dite **monolithique**
  - Un OS unique qui gère toute la machine
  - Suppose que la mémoire est partagée et cohérente
  - Représente la norme dans l'industrie informatique (Linux, BSD, Solaris, Windows NT/XP/2000)
  - En constante évolution .. depuis UNIX (1ère édition, **1971**)

- Approches alternatives :

État de l'art

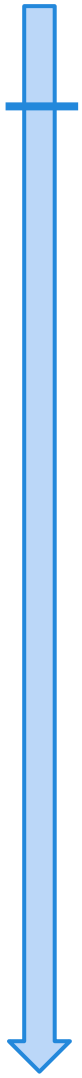
- Approche de conception traditionnelle, dite **monolithique**
  - Un OS unique qui gère toute la machine
  - Suppose que la mémoire est partagée et cohérente
  - Représente la norme dans l'industrie informatique (Linux, BSD, Solaris, Windows NT/XP/2000)
  - En constante évolution .. depuis UNIX (1ère édition, **1971**)

- Approches alternatives :

1995

Hurrican, une collection d'OS coopérants, Univ. de Toronto

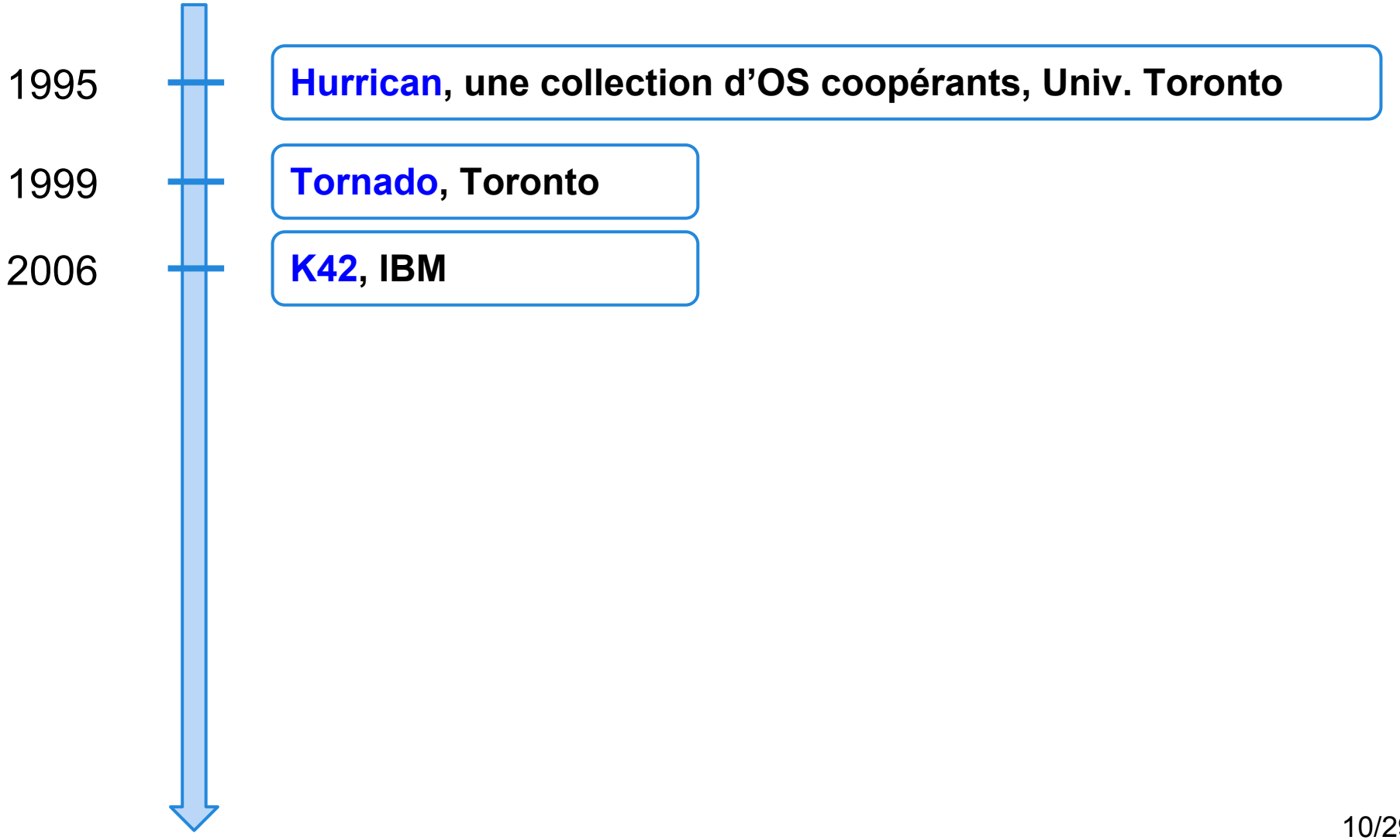
Approche de conception alternatives



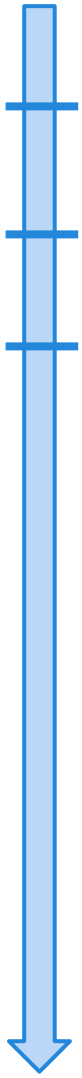
1995

Hurrican, une collection d'OS coopérants, Univ. Toronto

Approche de conception alternatives



Approche de conception alternatives



1995 — **Hurrican**, une collection d'OS coopérants, Univ. Toronto

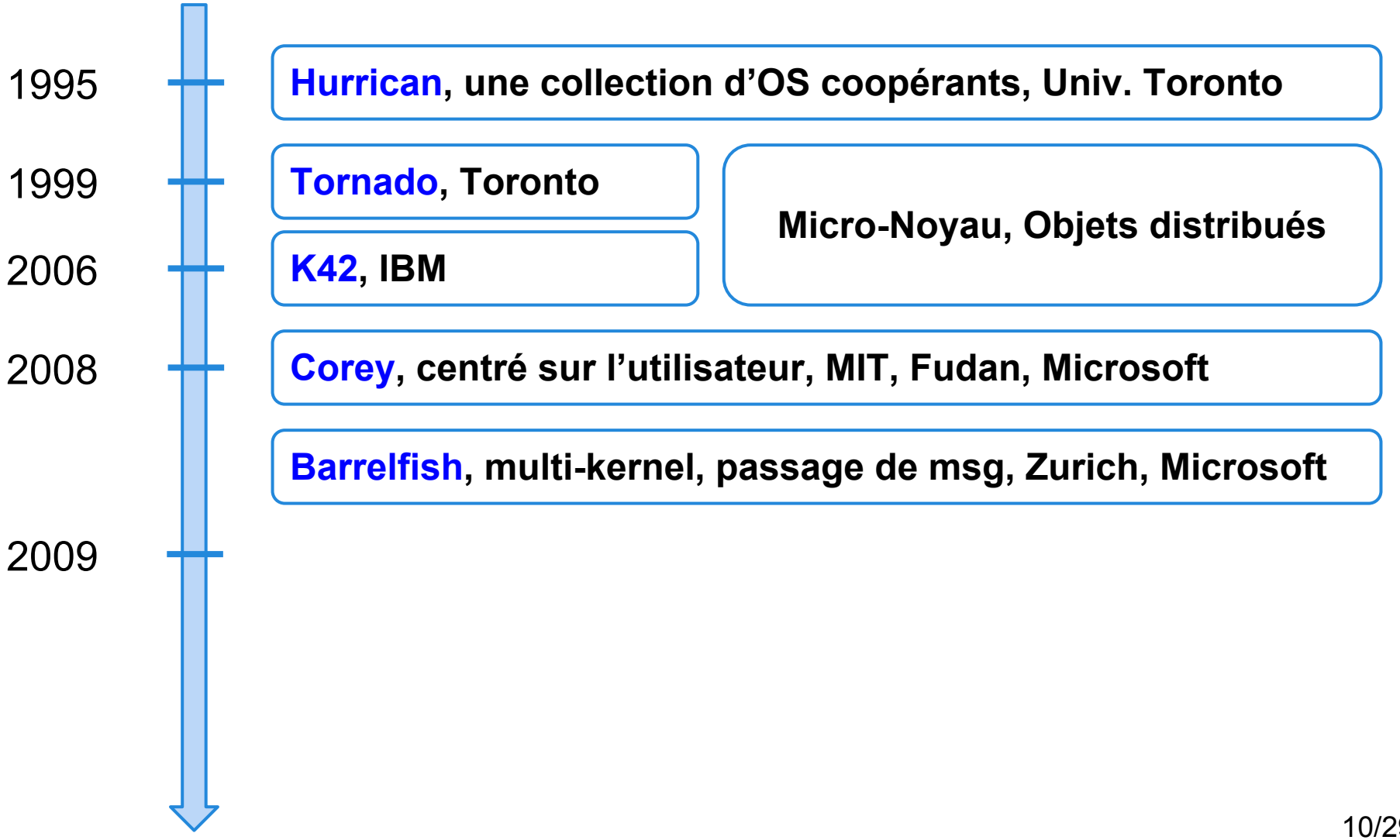
1999 — **Tornado**, Toronto

2006 — **K42**, IBM

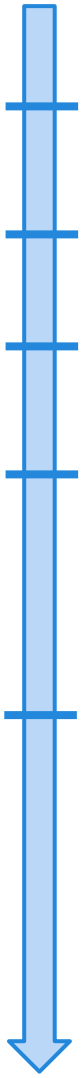
Micro-Noyau, Objets distribués

Approche de conception alternatives

Approche de conception alternatives



Approche de conception alternatives



1995 — **Hurrican**, une collection d'OS coopérants, Univ. Toronto

1999 — **Tornado**, Toronto

2006 — **K42**, IBM

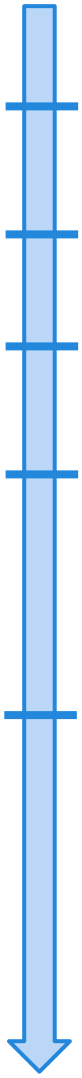
Micro-Noyau, Objets distribués

2008 — **Corey**, centré sur l'utilisateur, MIT, Fudan, Microsoft

Barrelfish, multi-kernel, passage de msg, Zurich, Microsoft

2009 — **FOS**, multi-kernel, passage de messages, MIT

Approche de conception alternatives



1995 — **Hurrican**, une collection d'OS coopérants, Univ. Toronto

1999 — **Tornado**, Toronto

2006 — **K42**, IBM

Micro-Noyau, Objets distribués

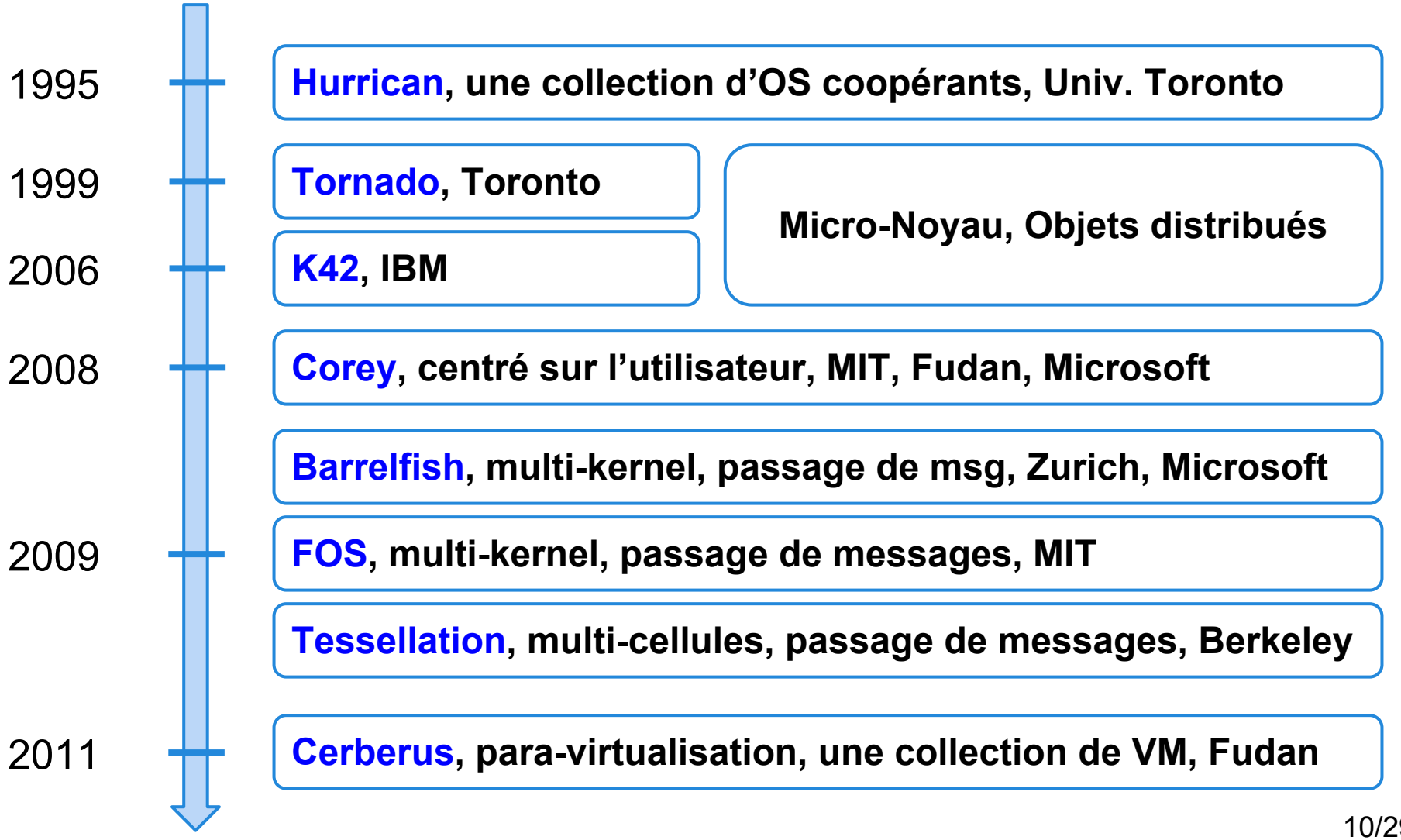
2008 — **Corey**, centré sur l'utilisateur, MIT, Fudan, Microsoft

Barrelfish, multi-kernel, passage de msg, Zurich, Microsoft

2009 — **FOS**, multi-kernel, passage de messages, MIT

Tessellation, multi-cellules, passage de messages, Berkeley

Approche de conception alternatives



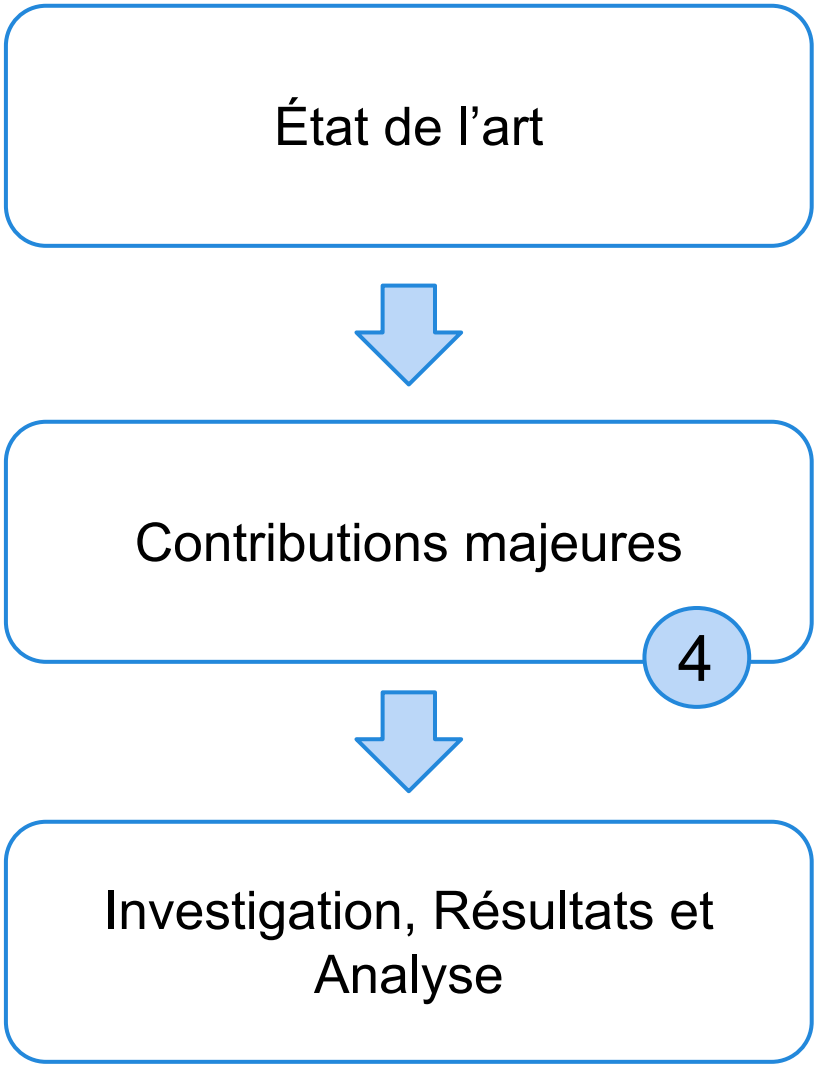
Synthèse de l'état de l'art

	Compatibilité	Localité			Scalabilité (Proc/Core)
		DATA	INST	TLB	
Linux	**Oui**	**Oui**	**Non**	**Non**	**64**
Hurrican	**Non**	**Oui**	**Non**	**Non**	**16**
Tornado/K42	**Oui**	**Oui**	**Non**	**Non**	**24**
Corey	**Non**	**Oui**	**Non**	**Non**	**16**
Barrelfish	**Non**	N/A	N/A	N/A	**16**
Cerberus	**Oui**	**Oui**	**Non**	**Non**	**16**

Synthèse de l'état de l'art

	Compatibilité	Localité			Scalabilité (Proc/Core)
		DATA	INST	TLB	
Linux	Oui	Oui	Non	Non	64
Hurrican	Non	Oui	Non	Non	16
Tornado/K42	Oui	Oui	Non	Non	24
Corey	Non	Oui	Non	Non	16
Barrelfish	Non	N/A	N/A	N/A	16
Cerberus	Oui	Oui	Non	Non	16
ALMOS	Oui	Oui	Oui	Oui	1024

Plan


État de l'art
Contributions majeures
4
Investigation, Résultats et Analyse

Les 6 contributions majeurs

ALMOS (Advanced Locality Management Operating System)

1

Les 6 contributions majeurs

ALMOS (Advanced Locality Management Operating System)

Le concept de Processus Hybride

2

Les 6 contributions majeurs

ALMOS (Advanced Locality Management Operating System)

Le concept de Processus Hybride

Le concept de Réplicas Noyau

3

Les 6 contributions majeurs

ALMOS (Advanced Locality Management Operating System)

Le concept de Processus Hybride

Le concept de Réplicas Noyau

La stratégie d'affinité mémoire automatique Auto-Next-Touch

4

Les 6 contributions majeurs

ALMOS (Advanced Locality Management Operating System)

Le concept de Processus Hybride

Le concept de Réplicas Noyau

La stratégie d'affinité mémoire automatique Auto-Next-Touch

Organisation : Ordonnanceur distribué, Services système répartis

5

Les 6 contributions majeurs

ALMOS (Advanced Locality Management Operating System)

Le concept de Processus Hybride

Le concept de Réplicas Noyau

La stratégie d'affinité mémoire automatique Auto-Next-Touch

Organisation : Ordonnanceur distribué, Services système répartis

Infrastructure distribuée DQDT : prise de décision décentralisée, multi-critères et sans verrou.

Les 6 contributions majeurs

ALMOS (Advanced Locality Management Operating System)

Le concept de Processus Hybride

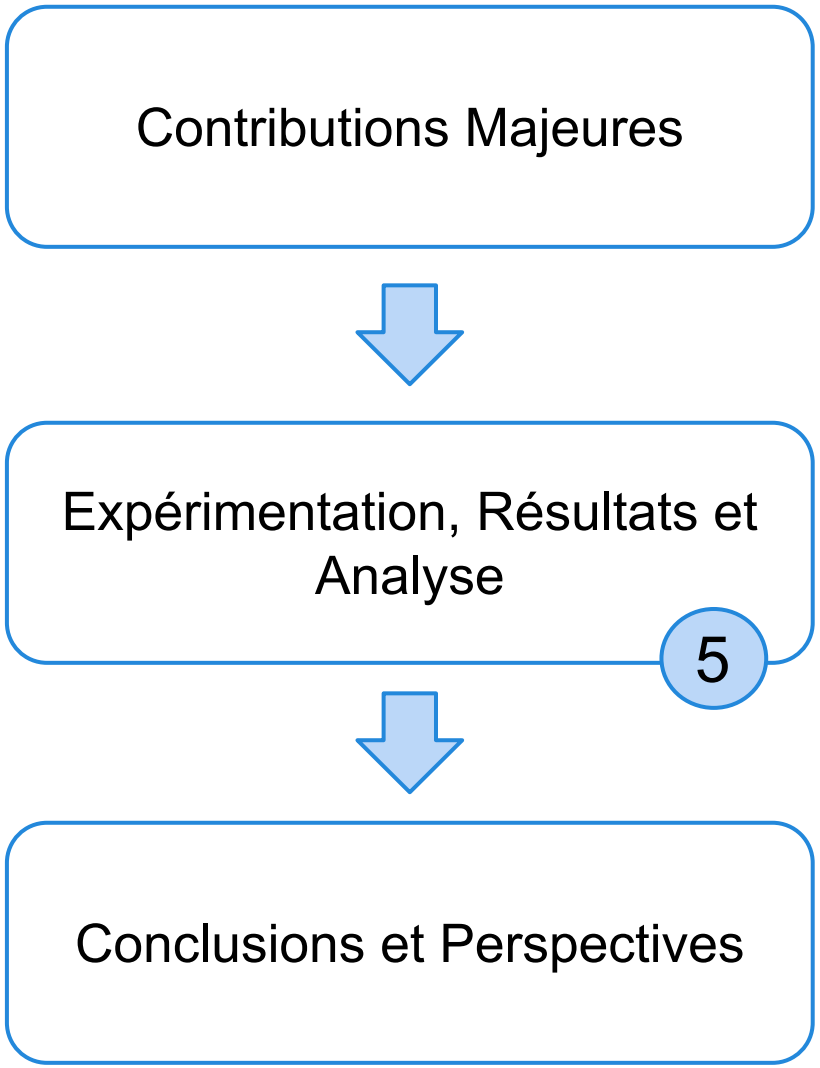
Le concept de Réplicas Noyau

La stratégie d'affinité mémoire automatique Auto-Next-Touch

Organisation : Ordonnanceur distribué, Services système répartis

Infrastructure distribuée DQDT : prise de décision décentralisée, multi-critères et sans verrou.

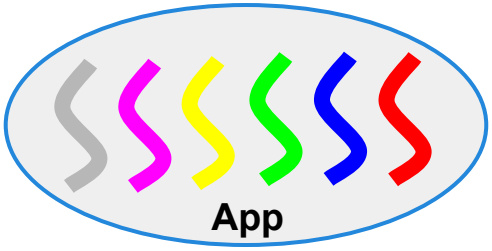
Plan


Contributions Majeures
Expérimentation, Résultats et Analyse
5
Conclusions et Perspectives

Question investiguée

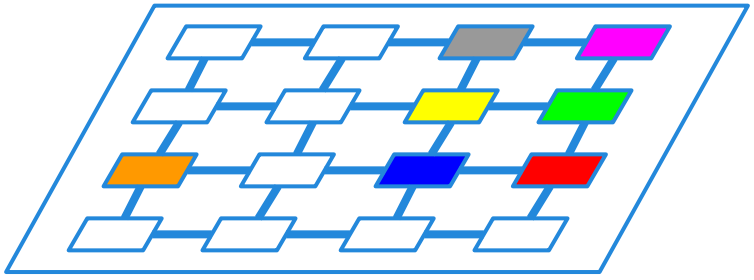
Est-ce qu'une application multi-threads **existante** passe à l'échelle ?

Dispositif d'expérimentation



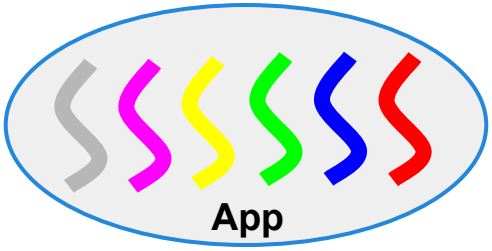
Radix-N24, FFT-M18, EP1024

Système d'exploitation

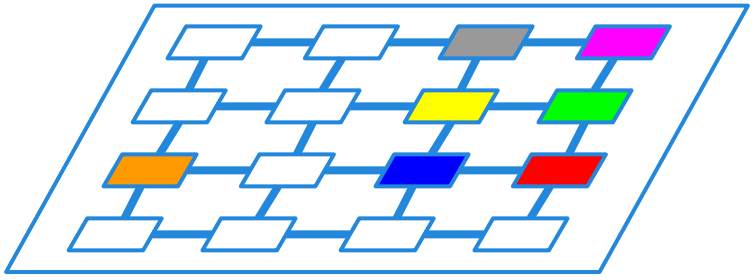


Processeur Many-Cores

Dispositif d'expérimentation



Système d'exploitation

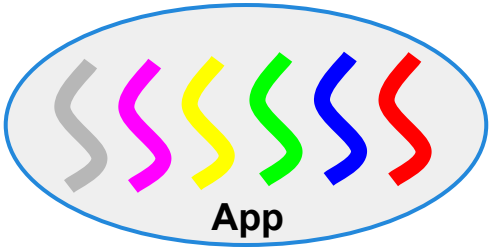


Processeur Many-Cores

Radix-N24, FFT-M18, EP1024

ALMOS, version ayant la même notion de threads que celle de Linux/Solaris/BSD

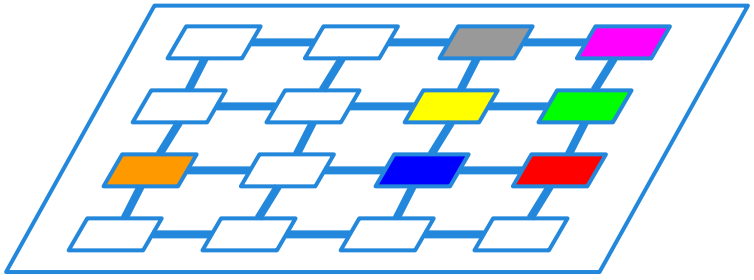
Dispositif d'expérimentation



Radix-N24, FFT-M18, EP1024

Système d'exploitation

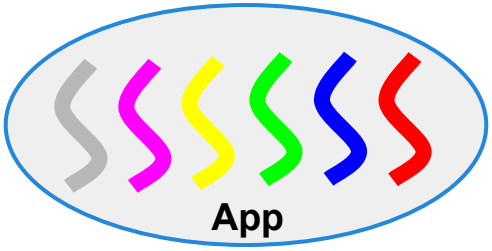
ALMOS, version ayant la même notion de threads que celle de Linux/Solaris/BSD



Processeur Many-Cores

Simulateur précis de TSAR, configuré jusqu'à 1024 cores

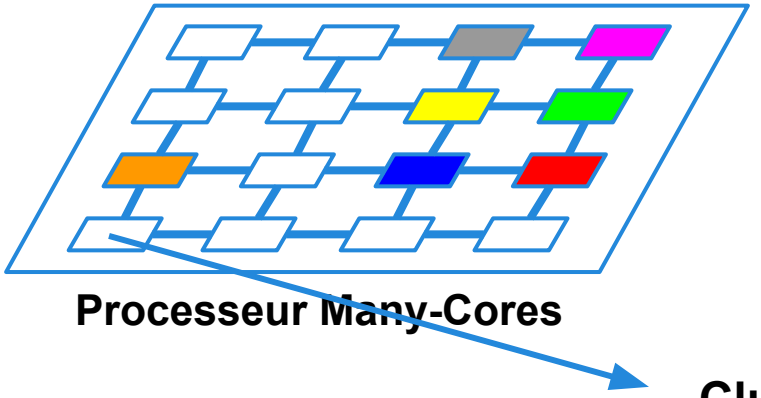
Dispositif d'expérimentation



Radix-N24, FFT-M18, EP1024

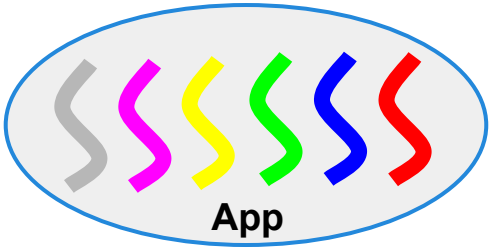
Système d'exploitation

ALMOS, version ayant la même notion de threads que celle de Linux/Solaris/BSD



Simulateur précis de TSAR, configuré jusqu'à 1024 cores

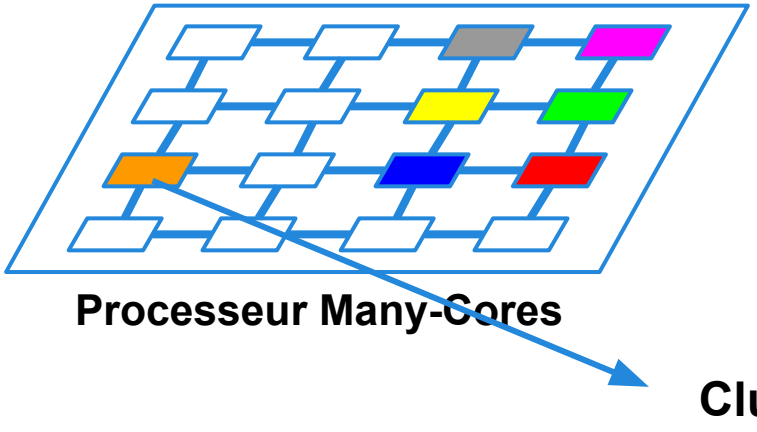
Dispositif d'expérimentation



Radix-N24, FFT-M18, EP1024

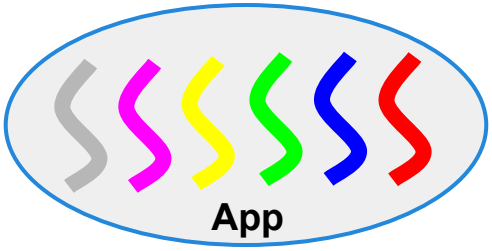
Système d'exploitation

ALMOS, version ayant la même notion de threads que celle de Linux/Solaris/BSD



Simulateur précis de TSAR, configuré jusqu'à 1024 cores

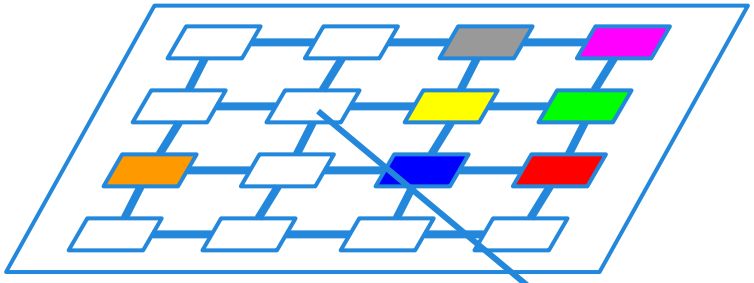
Dispositif d'expérimentation



Radix-N24, FFT-M18, EP1024

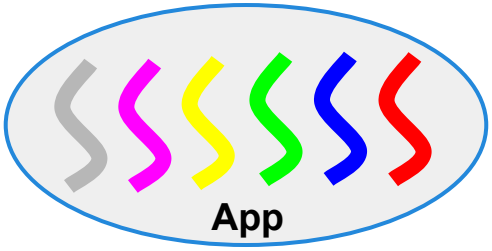
Système d'exploitation

ALMOS, version ayant la même notion de threads que celle de Linux/Solaris/BSD



Simulateur précis de TSAR, configuré jusqu'à 1024 cores

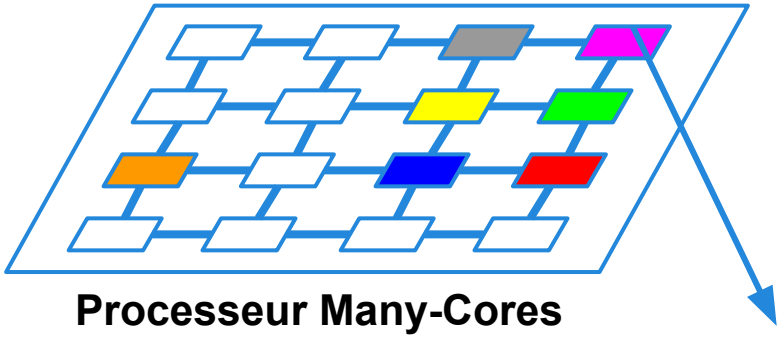
Dispositif d'expérimentation



Radix-N24, FFT-M18, EP1024

Système d'exploitation

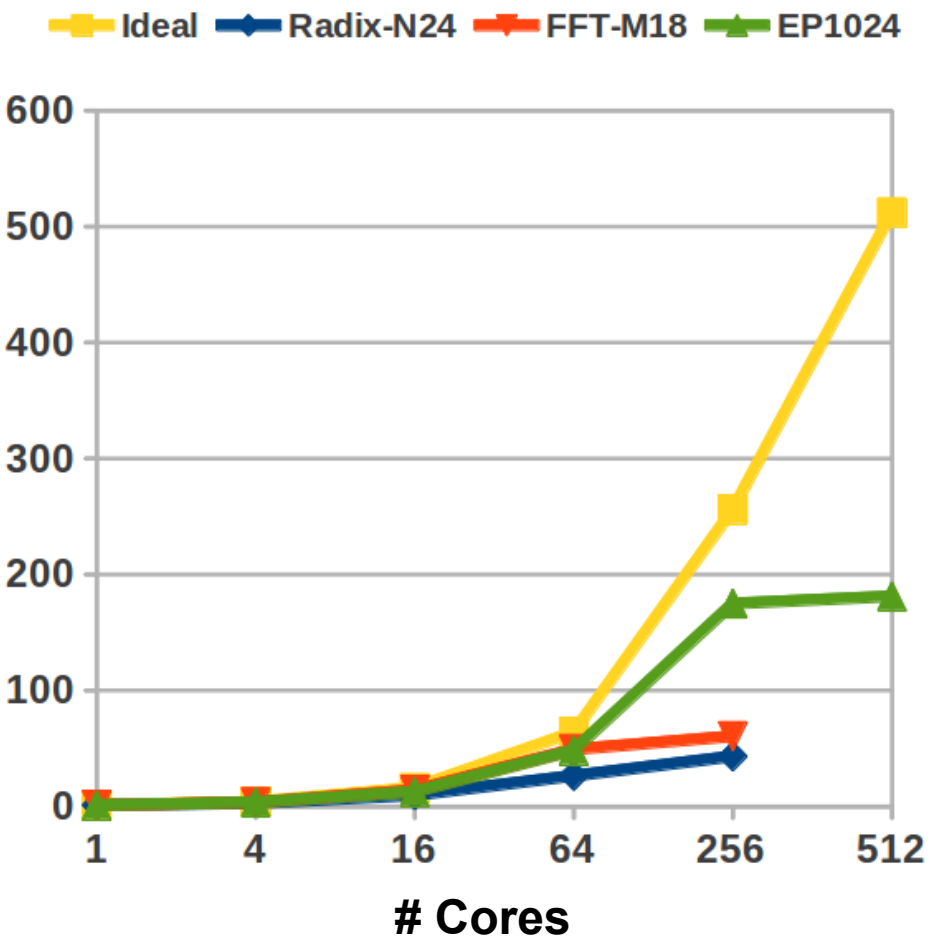
ALMOS, version ayant la même notion de threads que celle de Linux/Solaris/BSD



Simulateur précis de TSAR, configuré jusqu'à 1024 cores

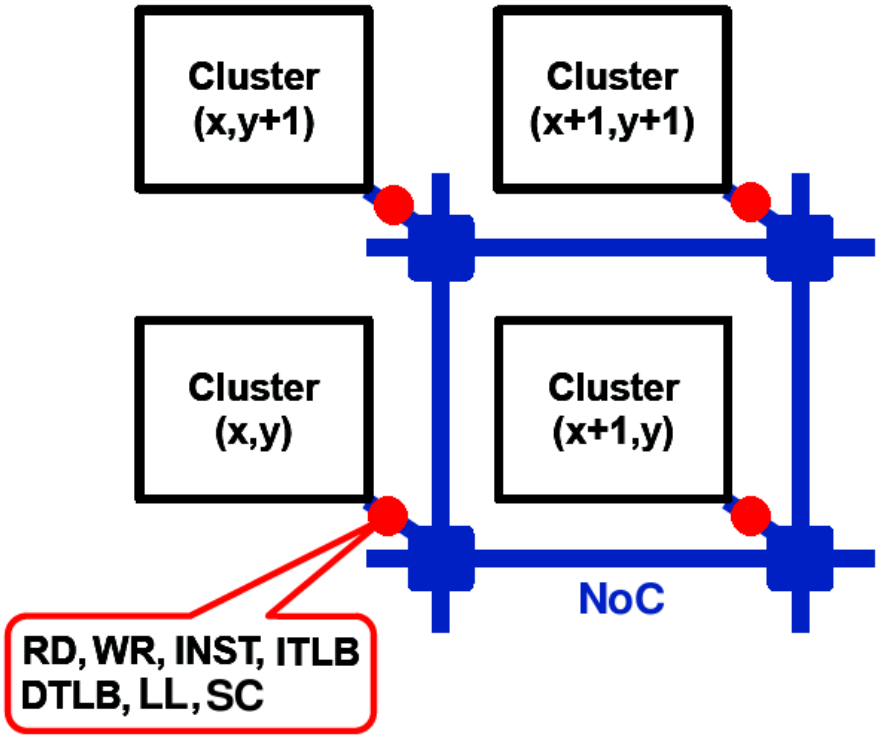
Résultats expérimentaux

Accélération (SpeedUp)

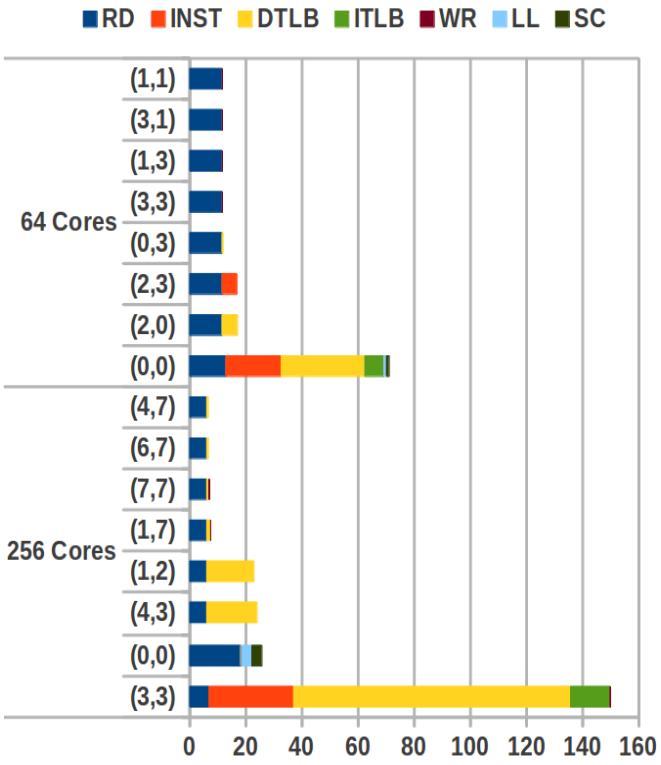


Dispositif d'instrumentation

Objectif : Analyser la distribution et la quantité du trafic distant

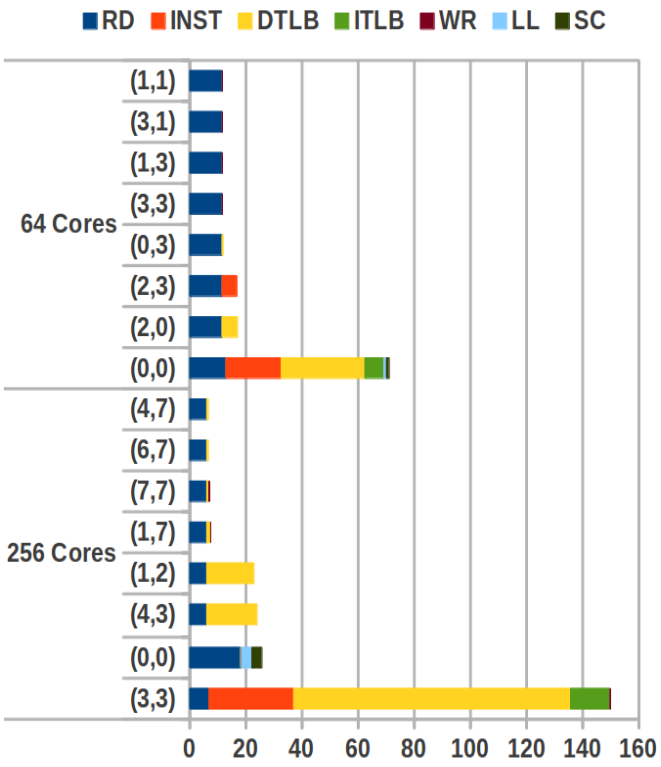


Analyse du trafic distant

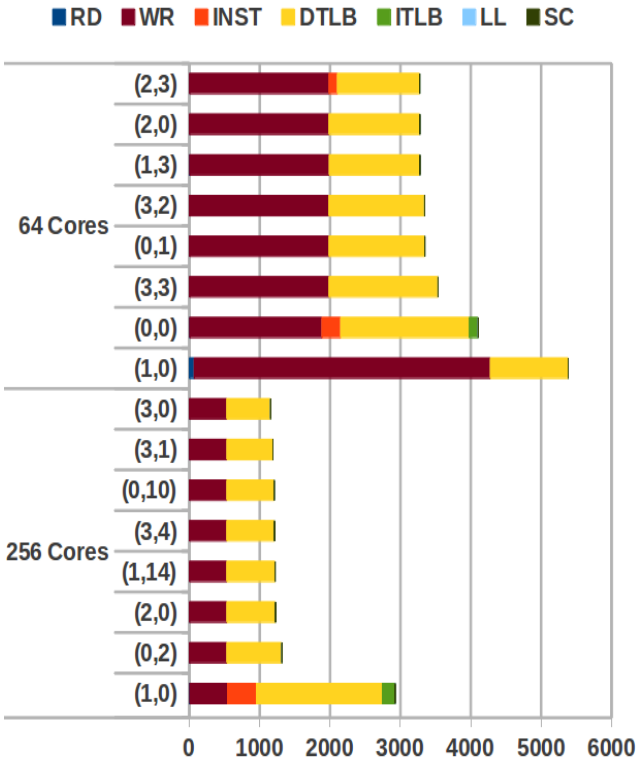


FFT-M18 Radix-N24 EP1024

Analyse du trafic distant



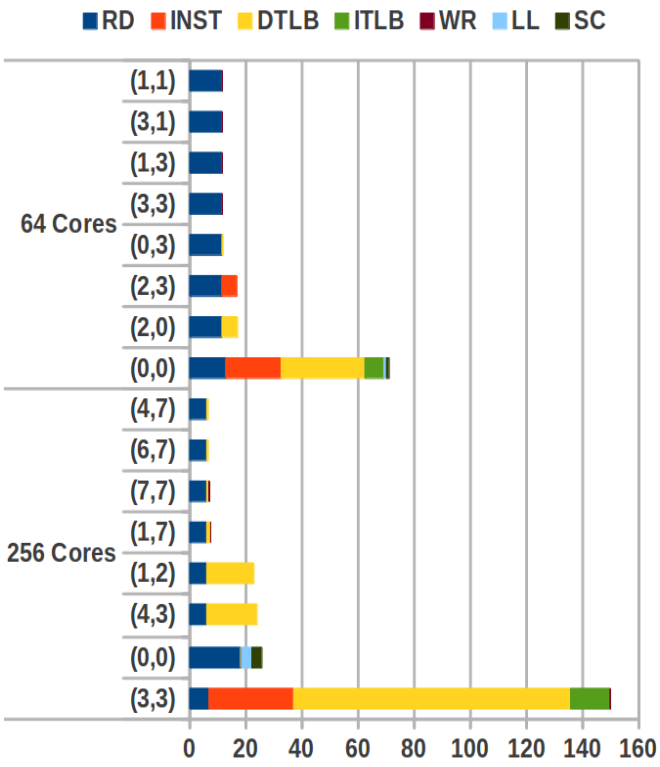
FFT-M18



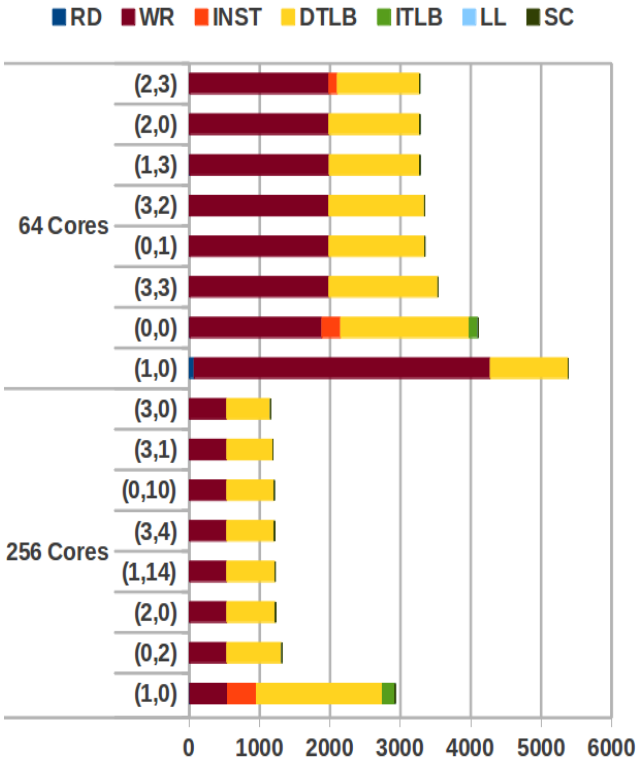
Radix-N24

EP1024

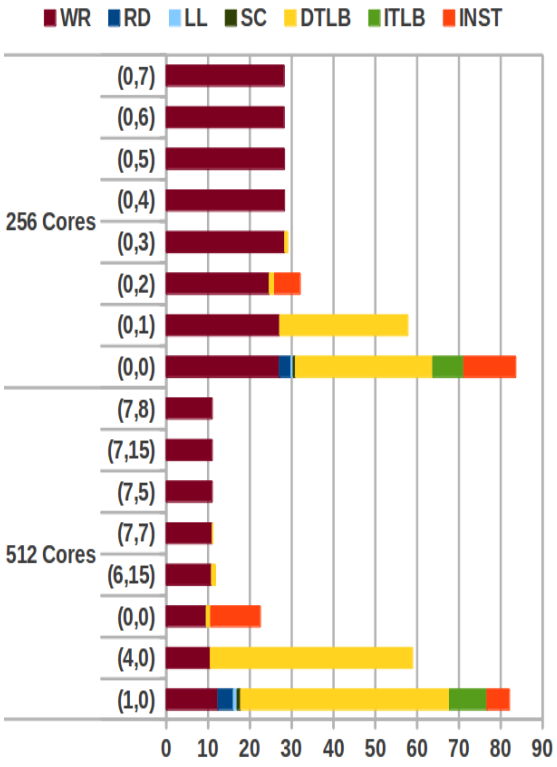
Analyse du trafic distant



FFT-M18

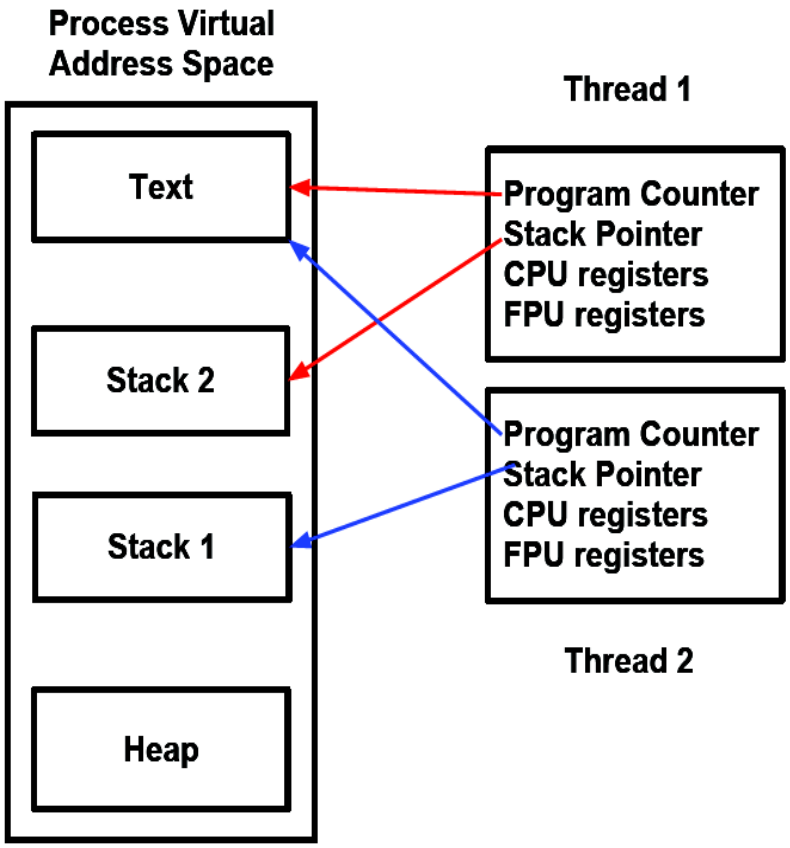


Radix-N24

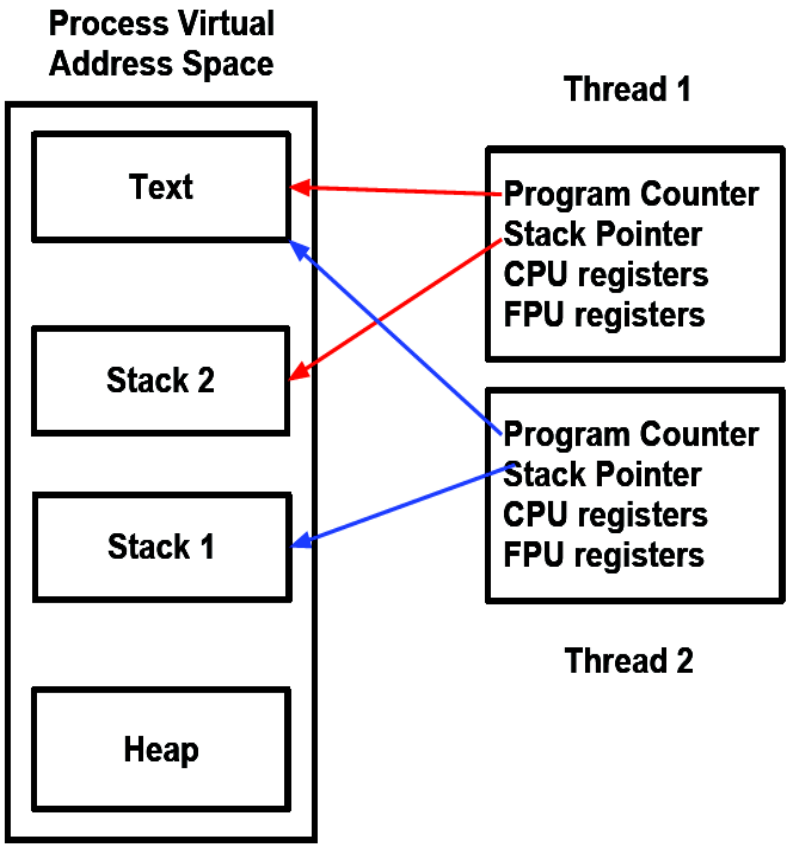


EP1024

Problème : la notion actuelle de threads

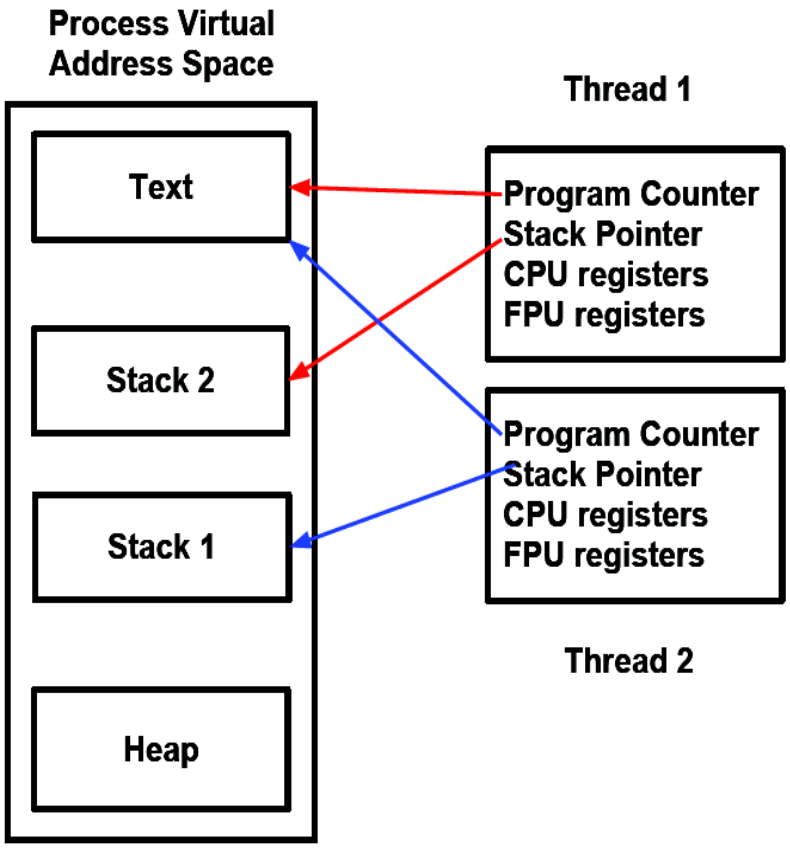


Problème : la notion actuelle de threads



L'espace d'adressage virtuel est partagé ⇔ Table de pages est référencée par tous les cores (miss TLB)

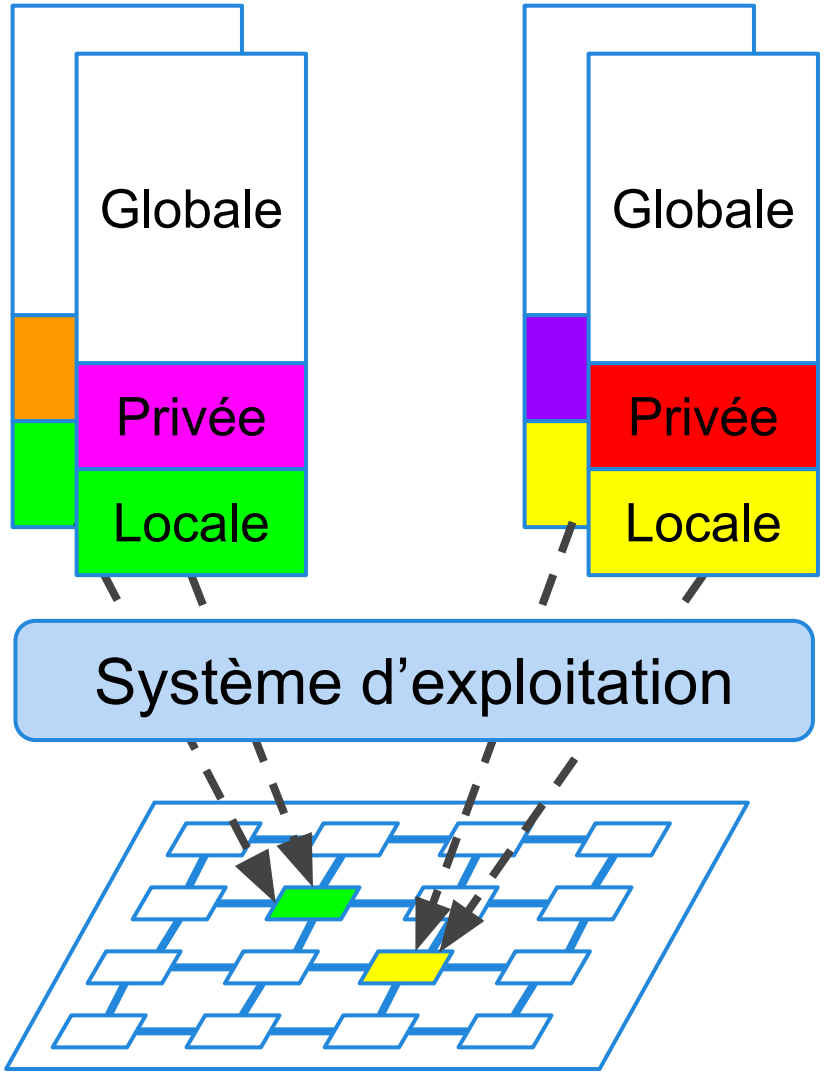
Problème : la notion actuelle de threads



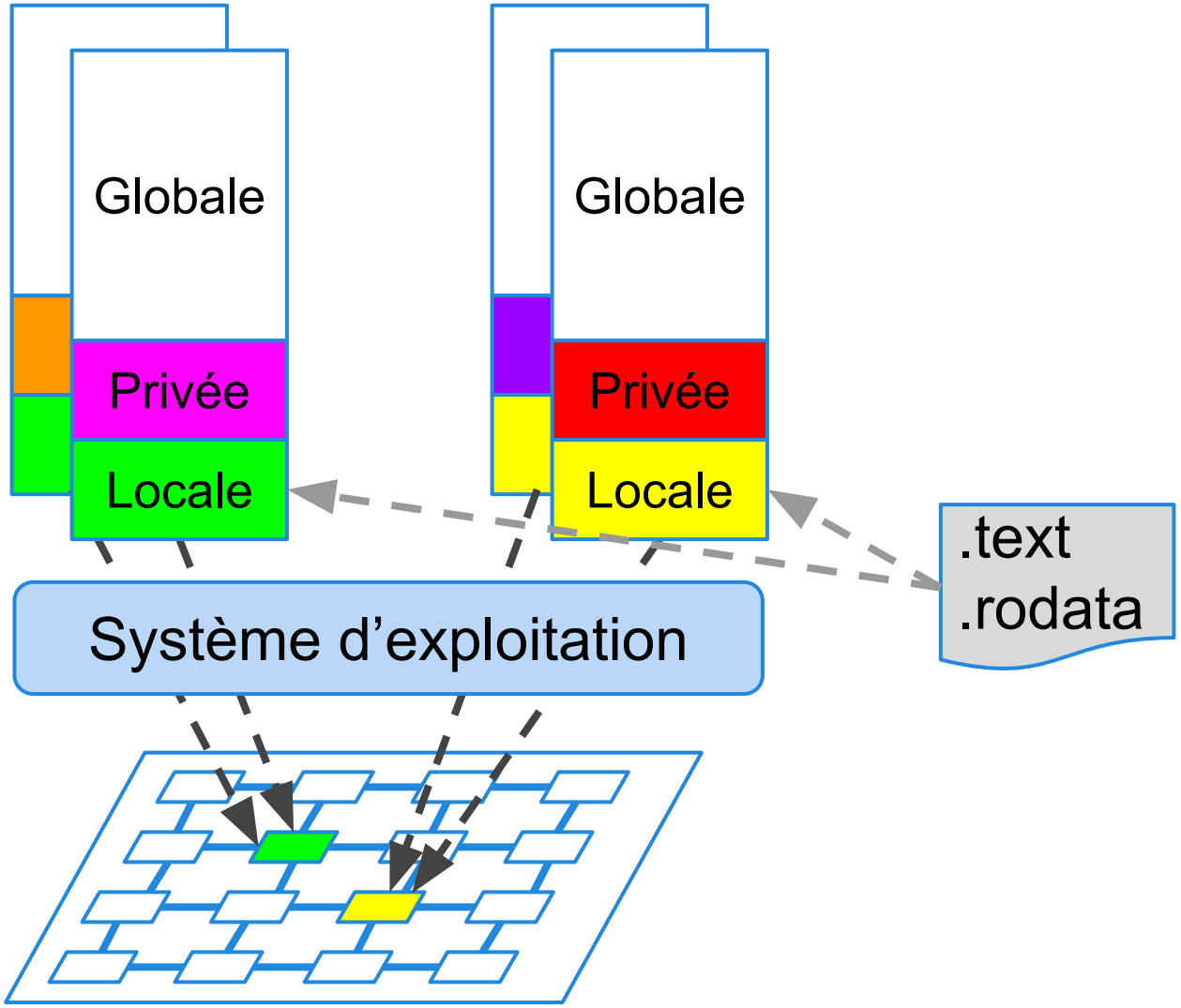
L'espace d'adressage virtuel est partagé ⇔ Table de pages est référencée par tous les cores (miss TLB)

Une seule copie des instructions ⇔ Les pages instructions sont référencées par tous les cores (miss Instructions)

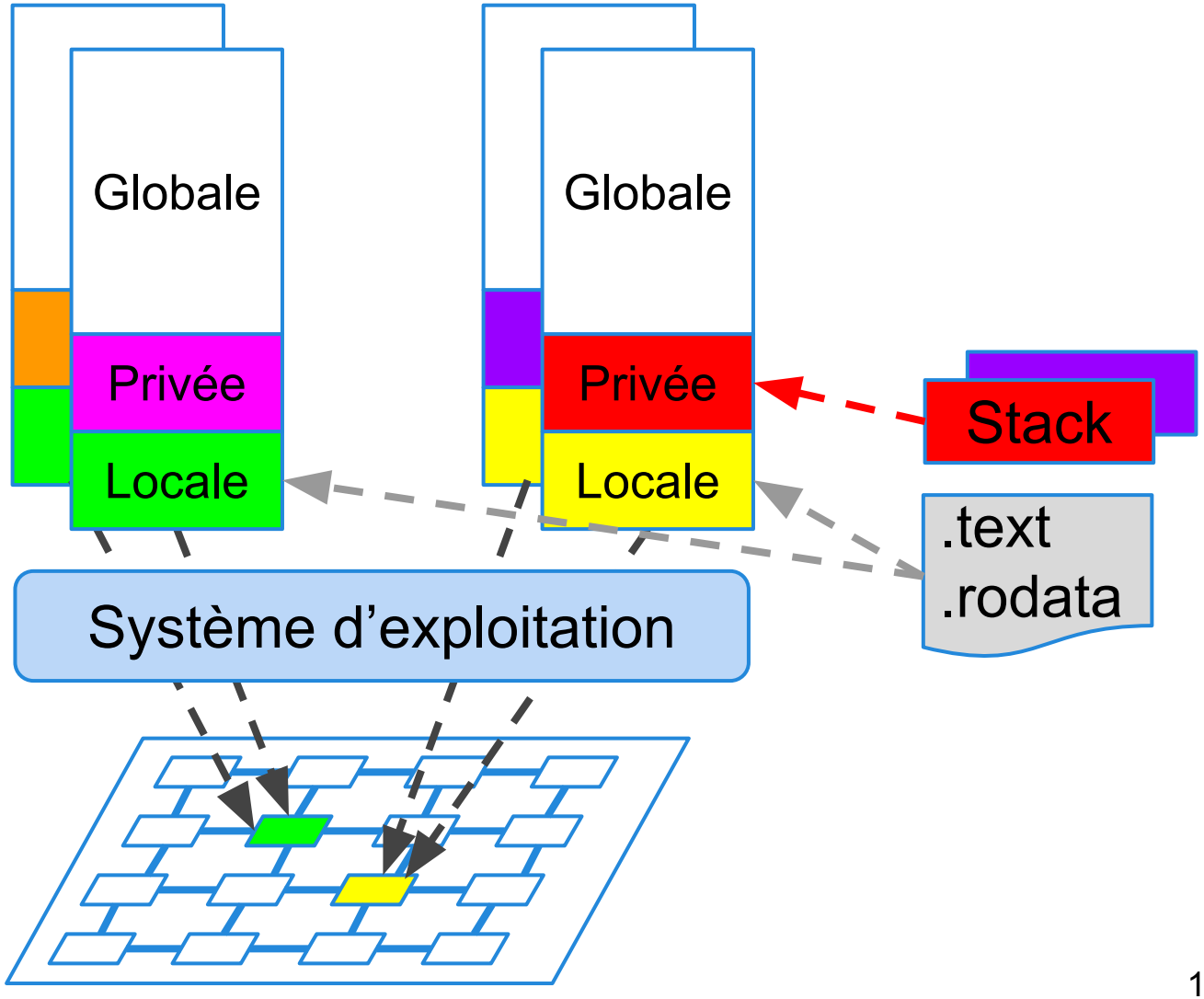
Solution : Processus Hybrides



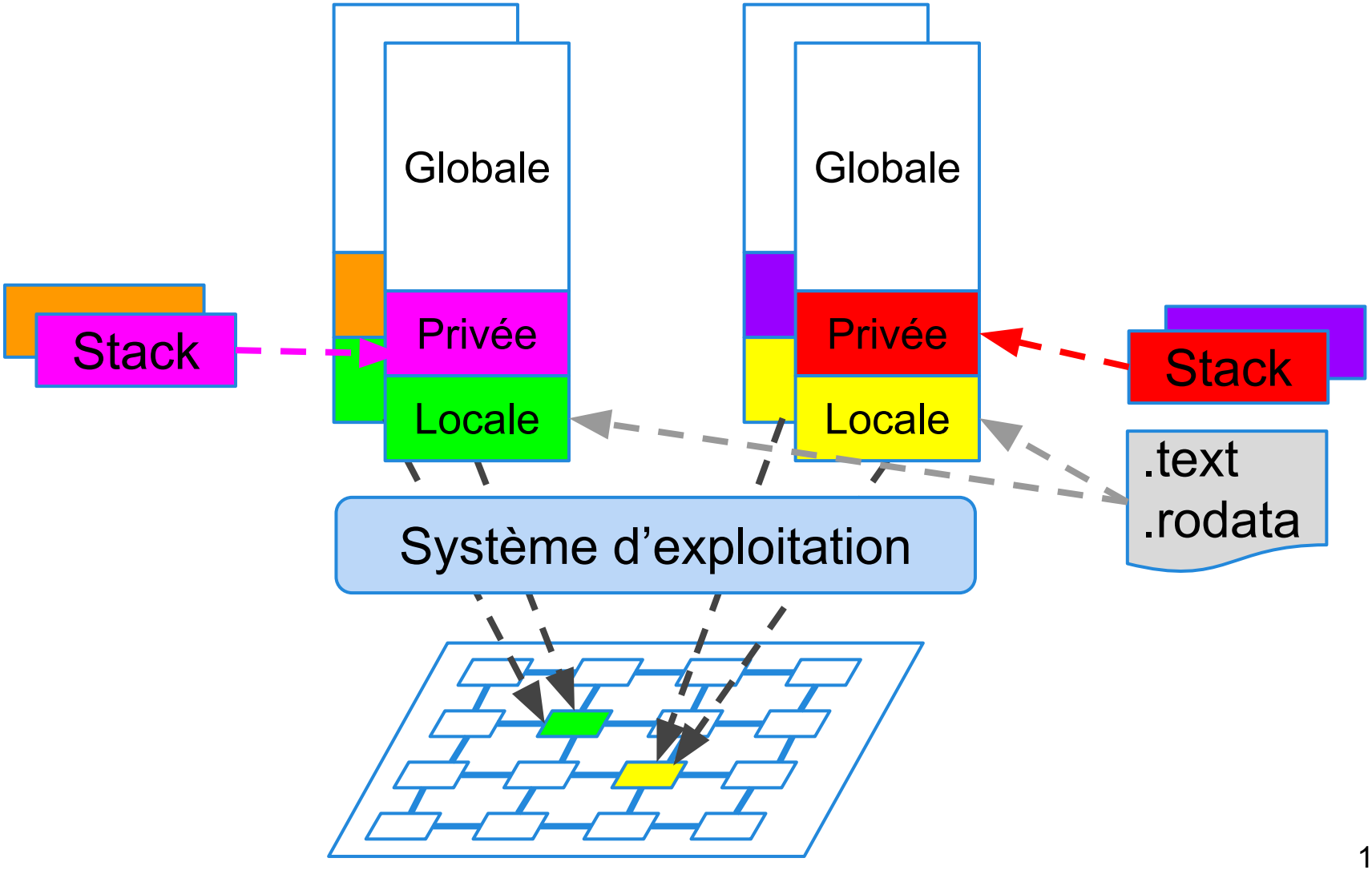
Solution : Processus Hybrides



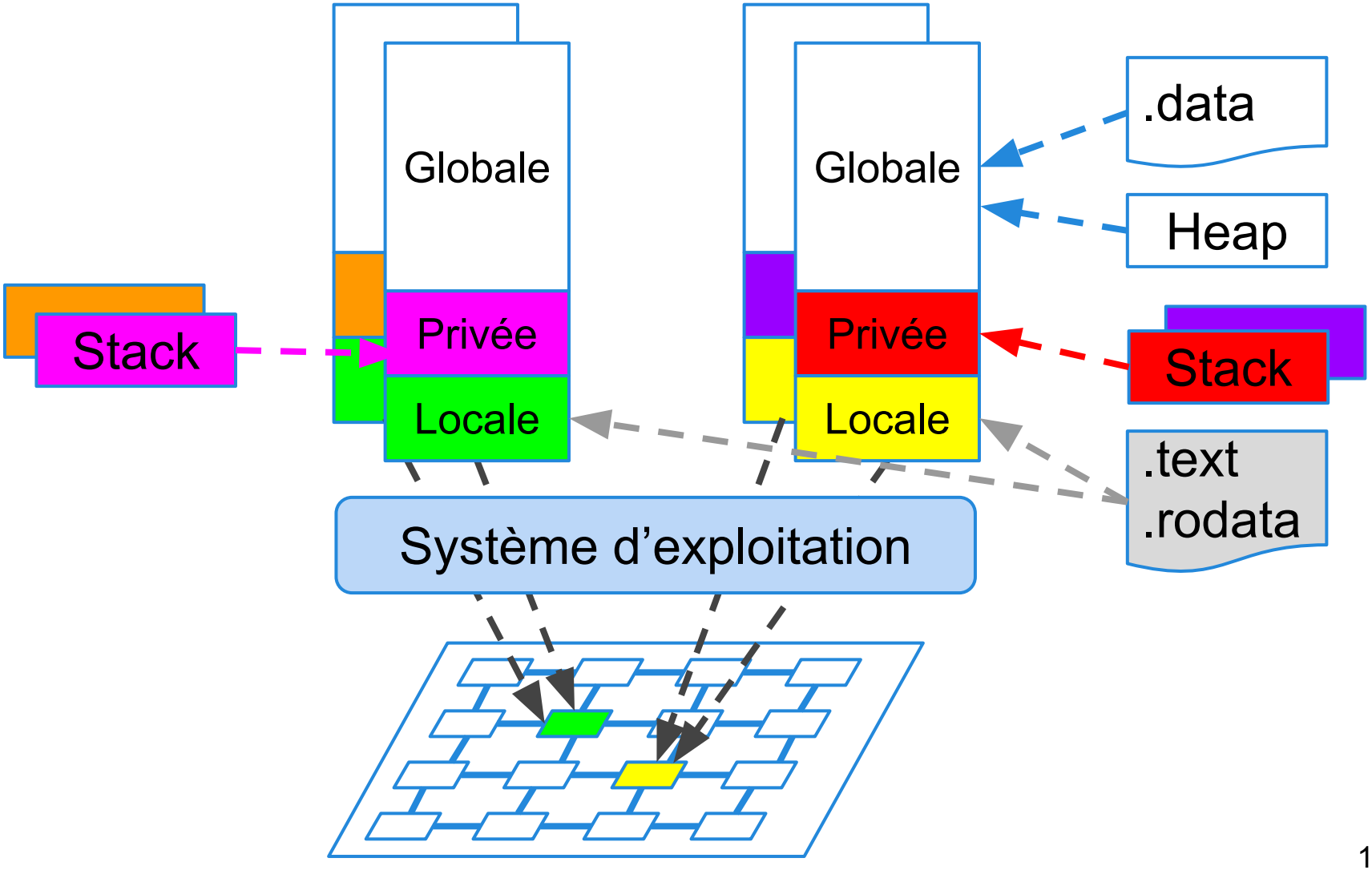
Solution : Processus Hybrides



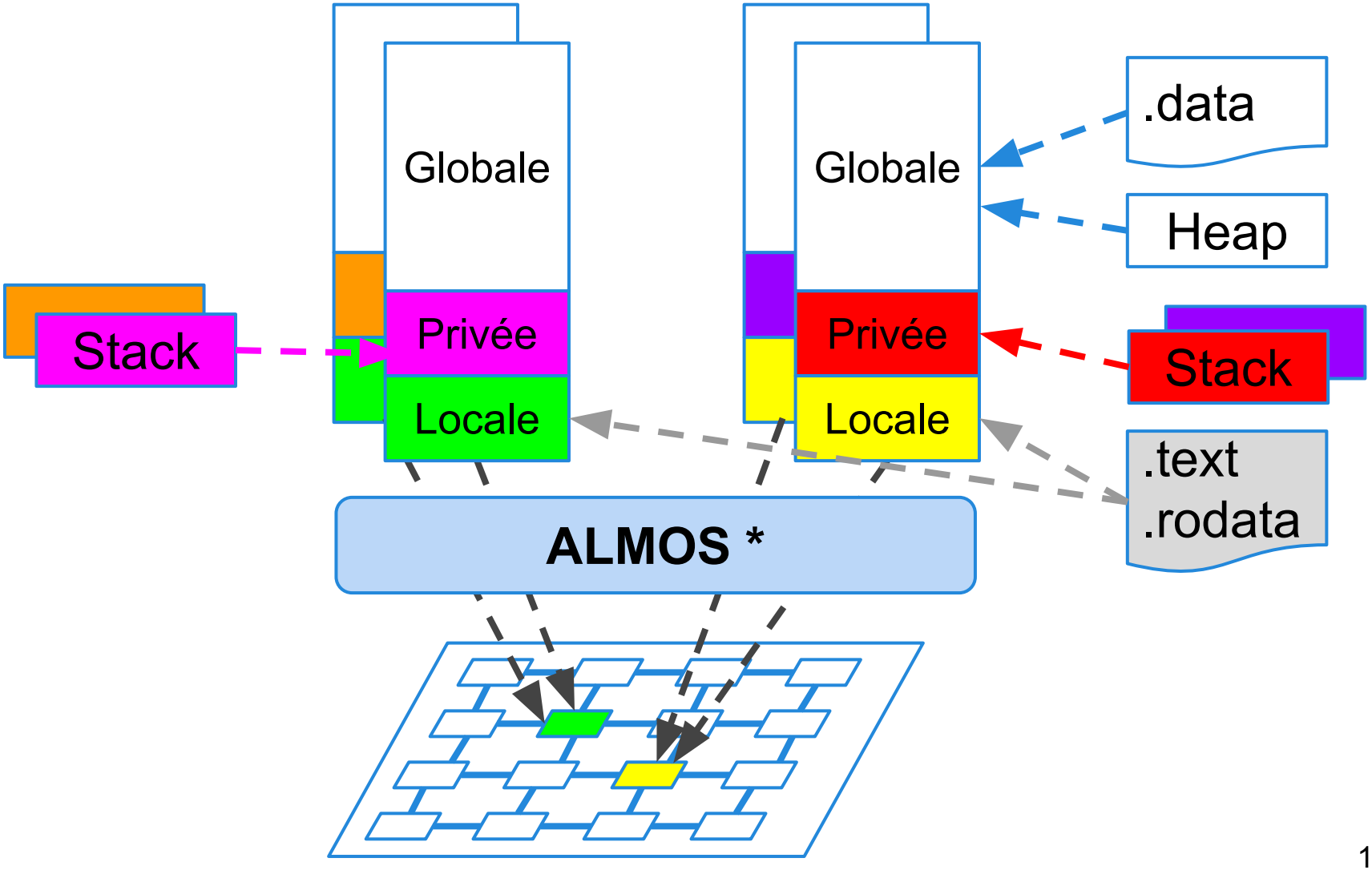
Solution : Processus Hybrides



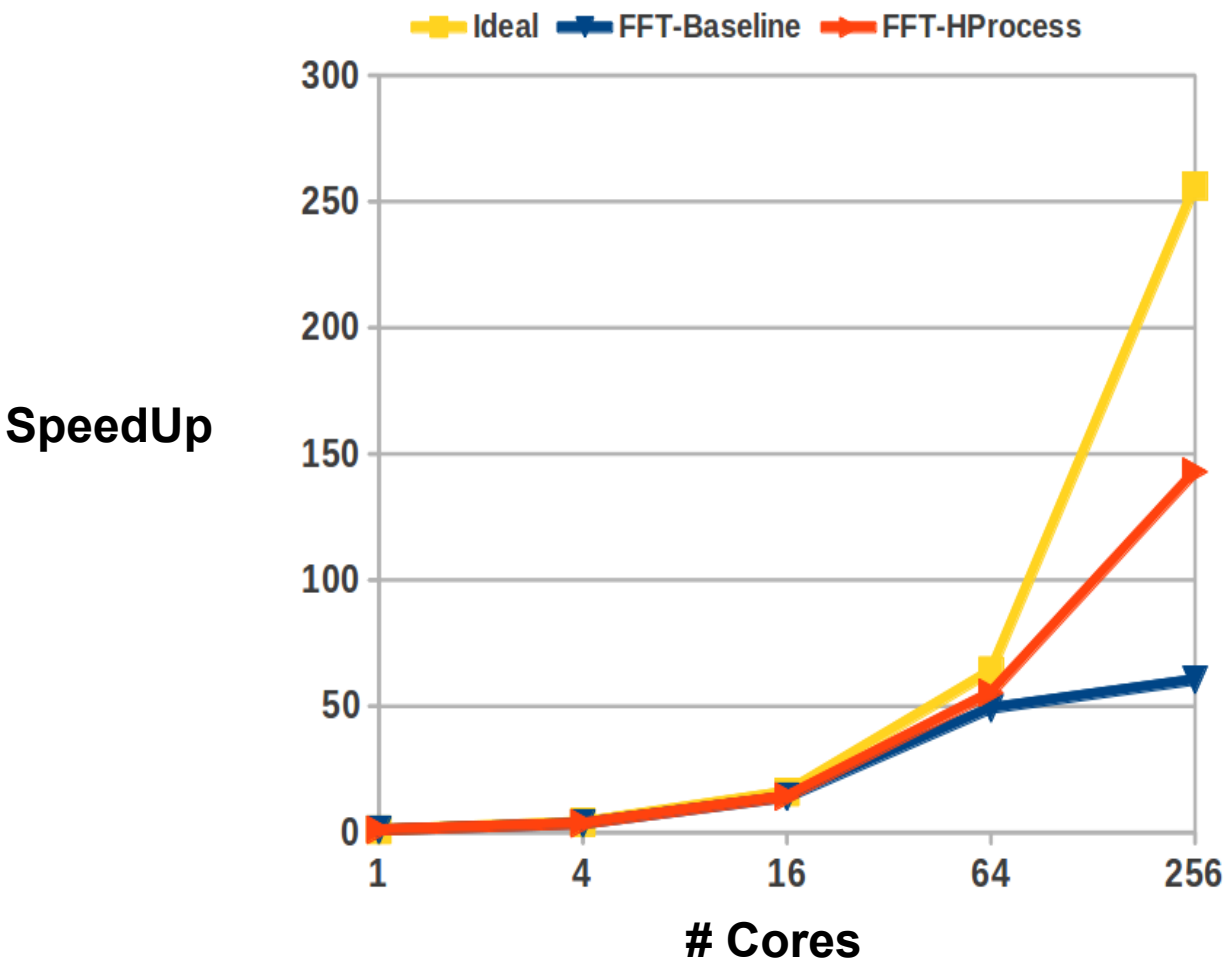
Solution : Processus Hybrides



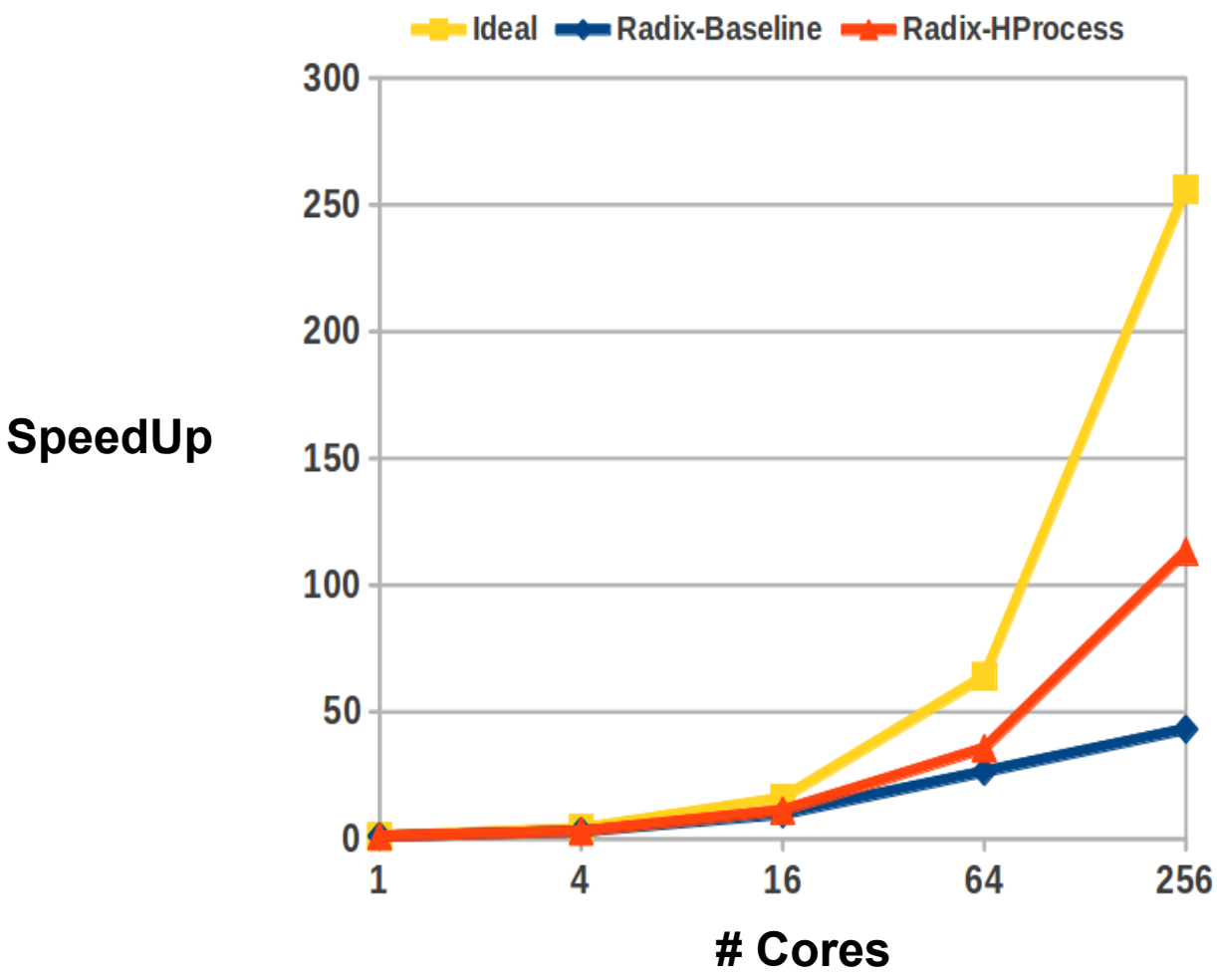
Solution : Processus Hybrides



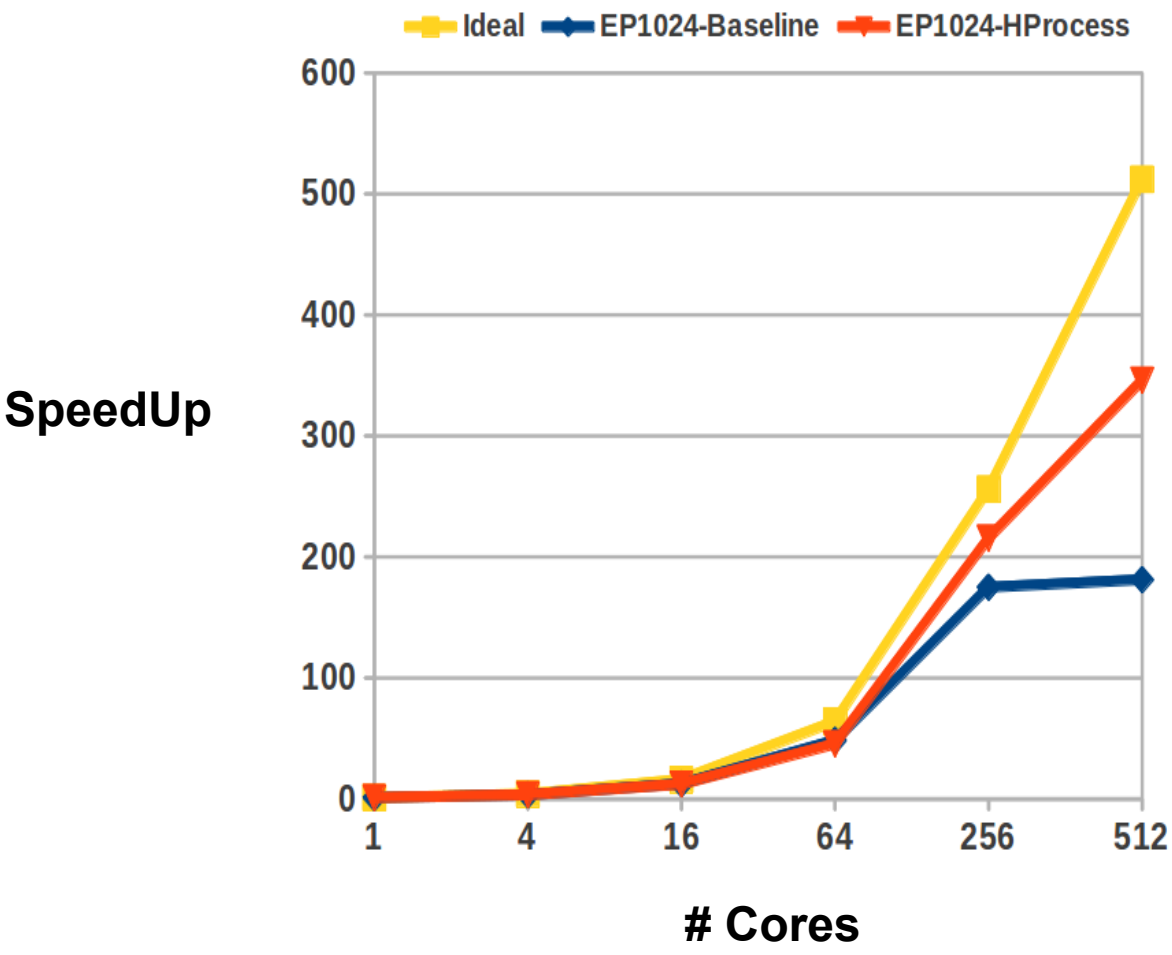
Résultats : FFT-M18



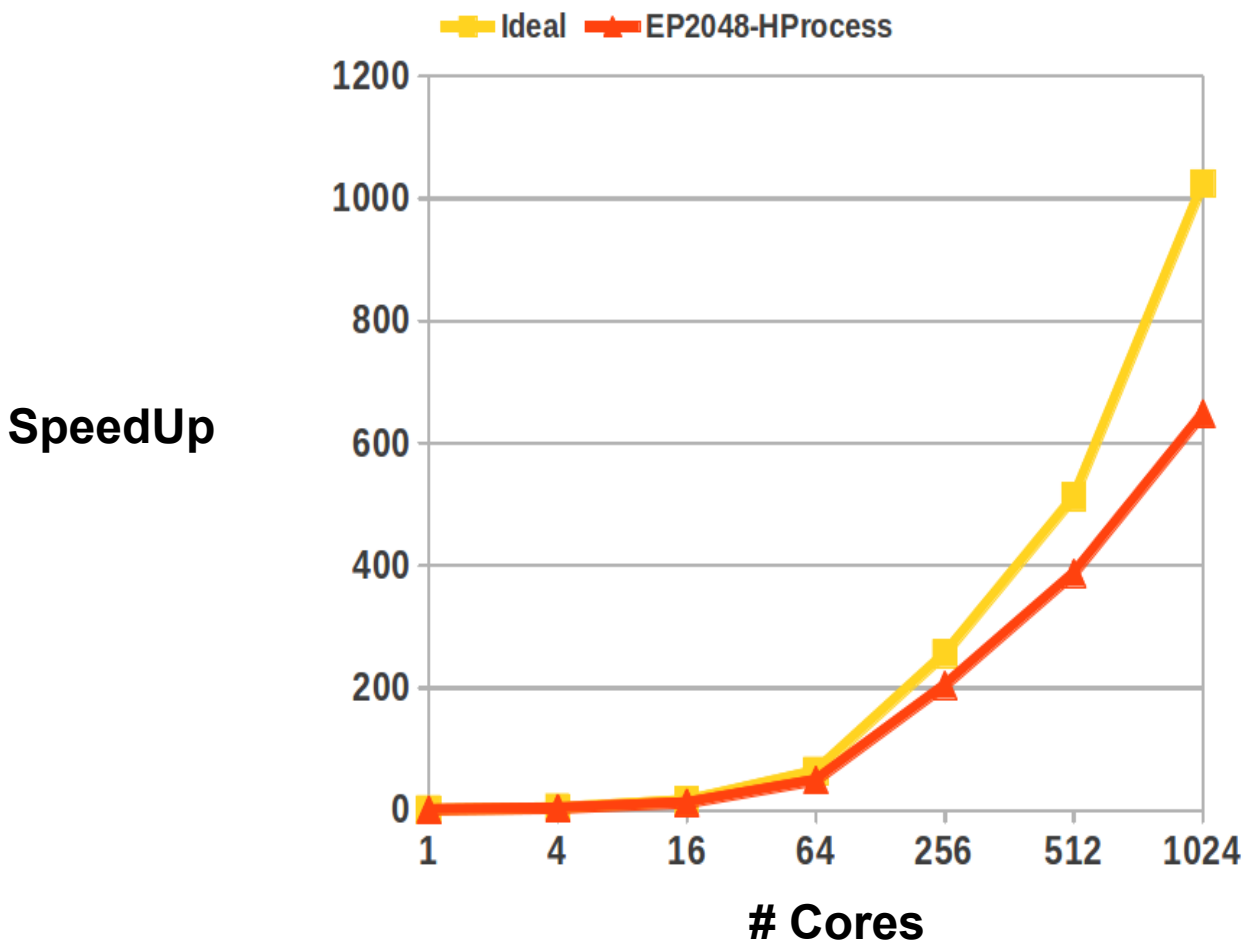
Résultats : Radix-N24



Résultats : EP1024



Résultats : EP2048



Avantages et Inconvénients

Modèle existant de threads	Processus Hybrides
- Ne passe pas à l'échelle	+ Passe à l'échelle
- Génère un trafic distant important	+ Renforce la localité par thread
- Perte de localité après migration	+ Le noyau est capable de restaurer la localité d'un thread après migration
- Aucune protection inter-threads	+ Protection pour chaque thread
- Défaut de pages en concurrence	+ Défaut de pages à coût minimal (partie privée)
+ Défaut de pages dans le heap/text	- Défaut de pages dans le heap/text

Plan

Expérimentation, Résultats et Analyse

Conclusions et Perspectives

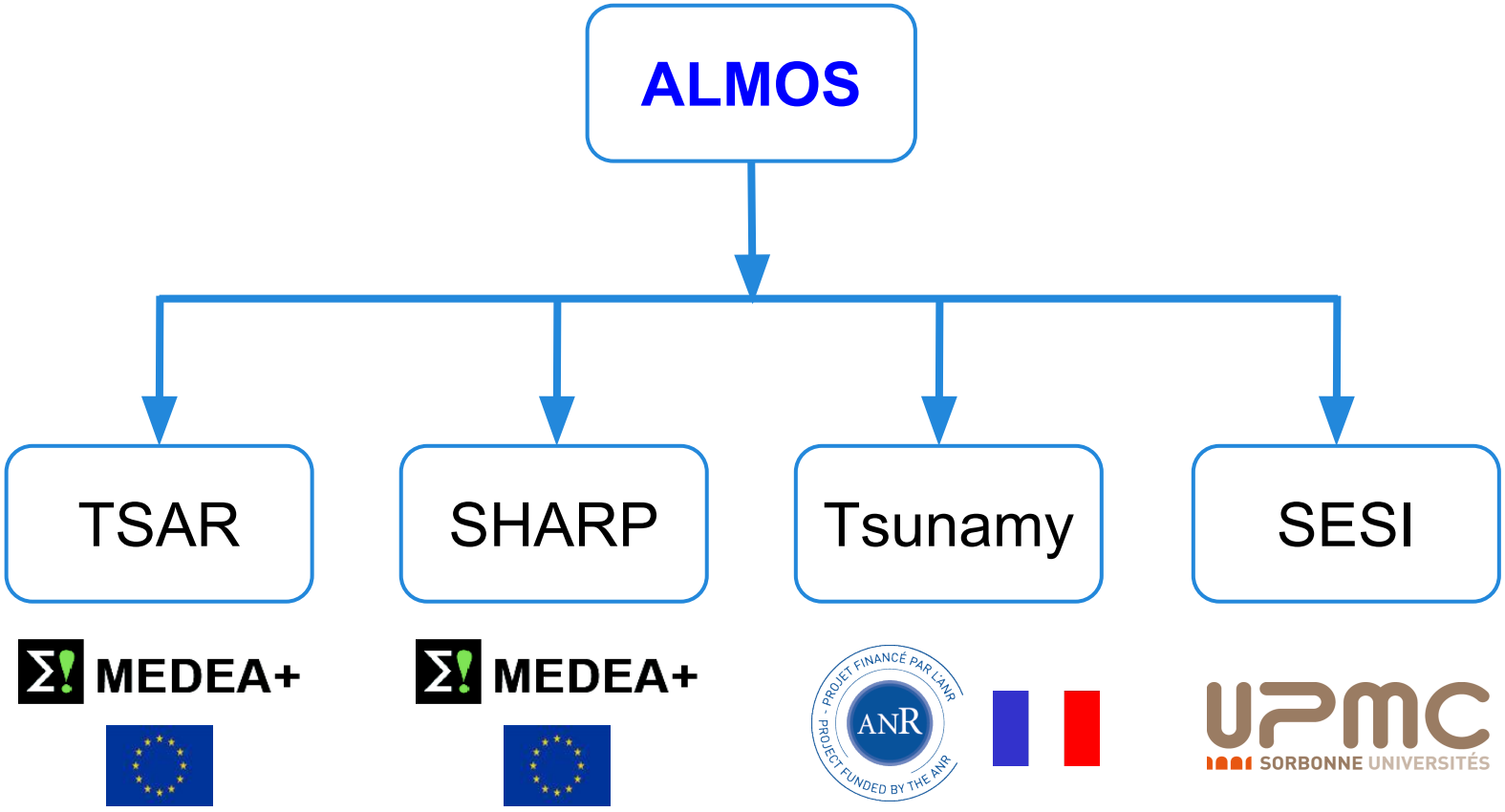
Conclusions

- Les processeurs many-cores à mémoire partagée cohérente assurée par le matériel sont une réalité et seront bientôt la norme.
- La localité des accès mémoire est une question primordiale pour le passage à l'échelle.
- Elle concerne non seulement l'accès aux données, mais également l'accès aux instructions et aux tables de pages.
- Notre thèse apporte des réponses à ce problème : le concept de Processus Hybride, le concept de Réplicas Noyau, la stratégie Auto-Next-Touch et l'infrastructure de prise de décision DQDT.
- ALMOS assure la compatibilité, renforce la localité et permet de passer à l'échelle plusieurs applications multi-tâches, contrairement aux solutions existantes.

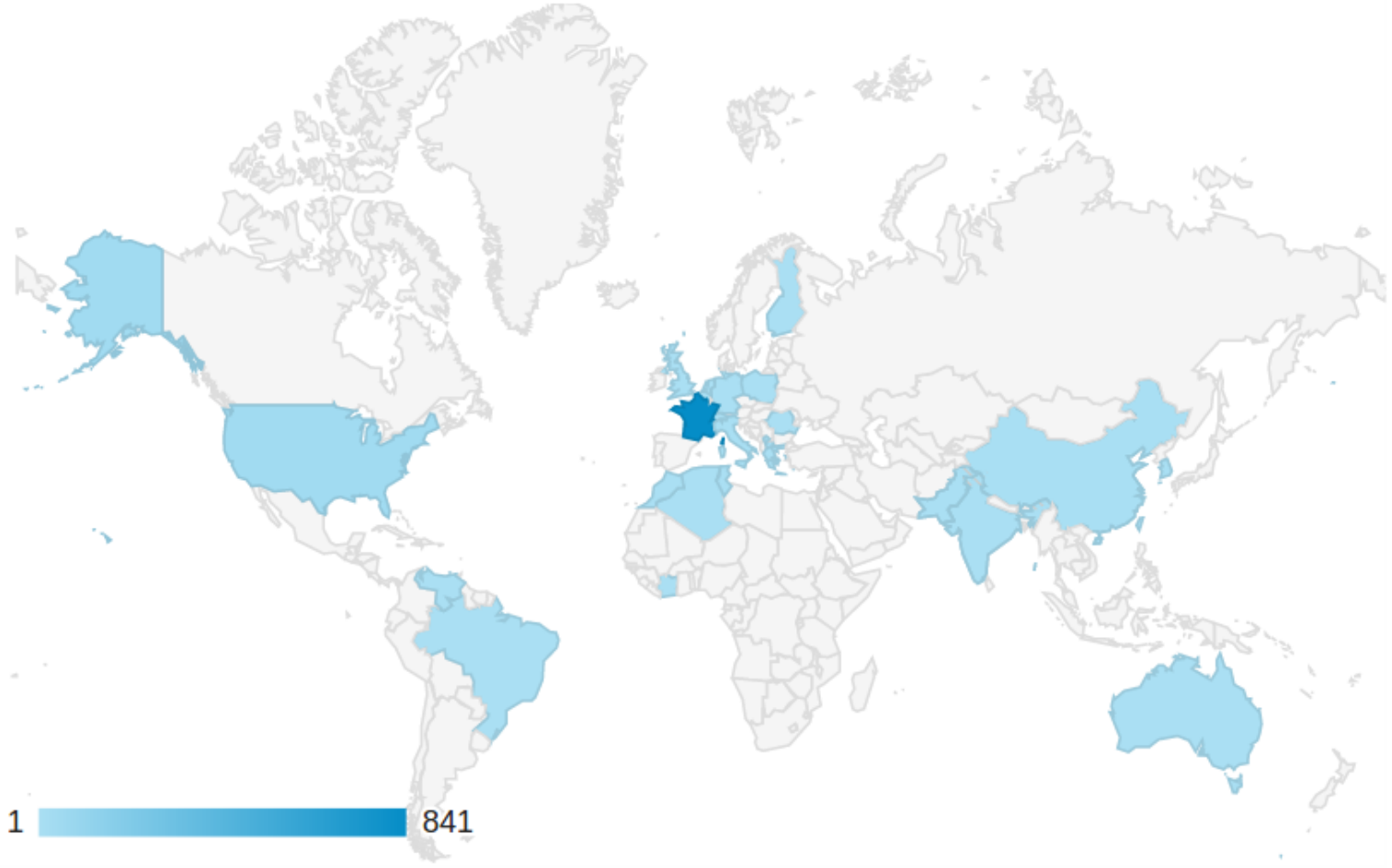
Perspectives

- Étudier expérimentalement la migration d'un H-Processus et de ses objets mémoire
- Tirer un meilleur profit du concept de Processus Hybrides
  - Investiguer le modèle de programmation PGAS
  - Étendre le contrôle du noyau sur le protocole de cohérence
- Étudier davantage les stratégies de prise de décision associées à la DQDT
- Porter ALMOS sur un autre processeur many-core, p.ex : Tilera
- Étudier la scalabilité des sous-systèmes d'E/S (p.ex : fichiers)

ALMOS : Brique technologique incontournable au LIP6



ALMOS suscite de l'intérêt au niveau mondial



Google Analytics : après 14 mois, accès depuis 24 pays

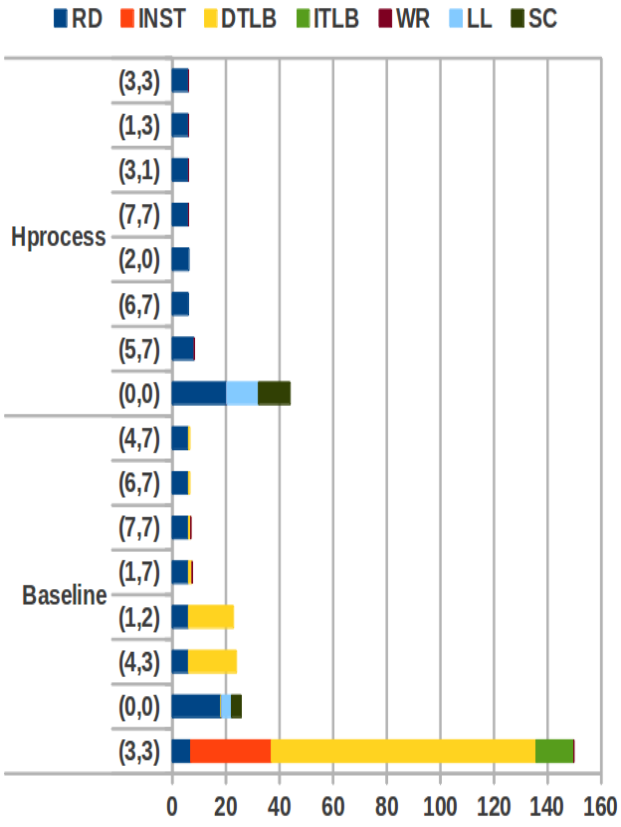
www.almos.fr

www.almos.fr

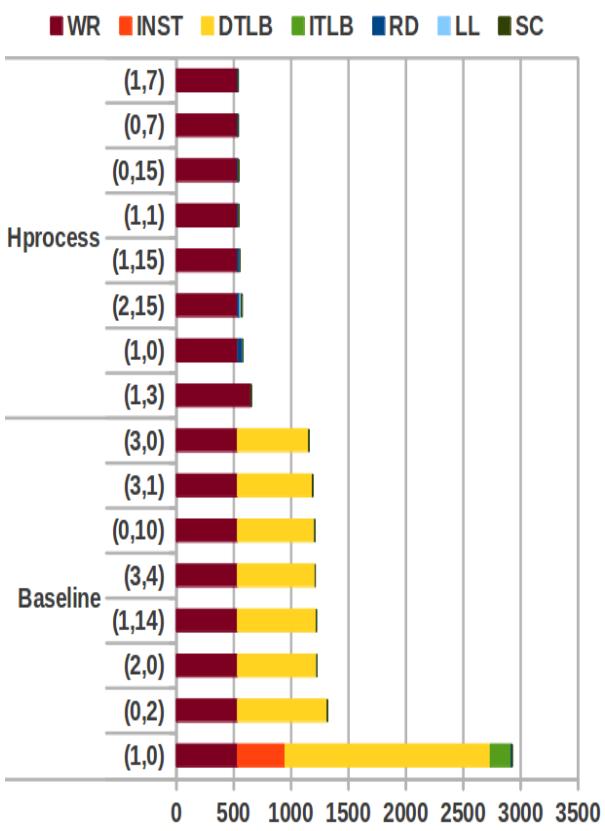
www.almos.fr

Backup Slides

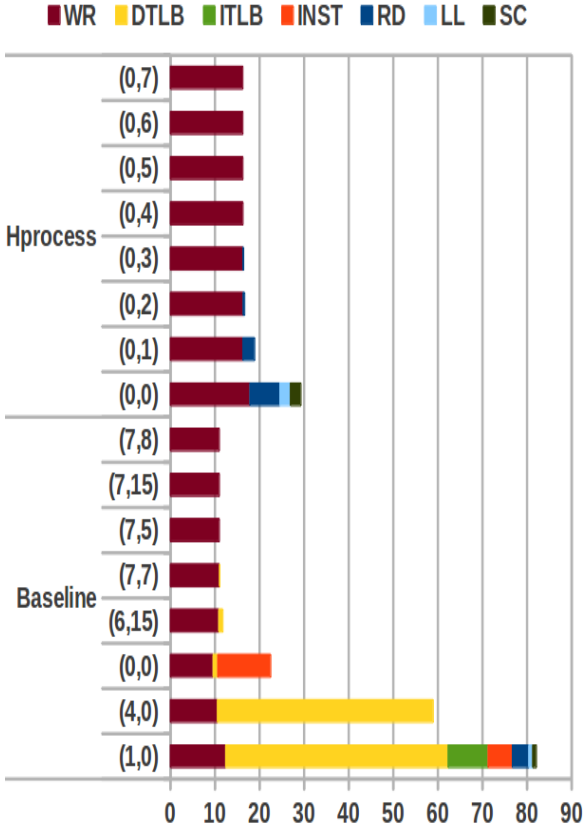
H-Process : Distribution du trafic distant



FFT-M18

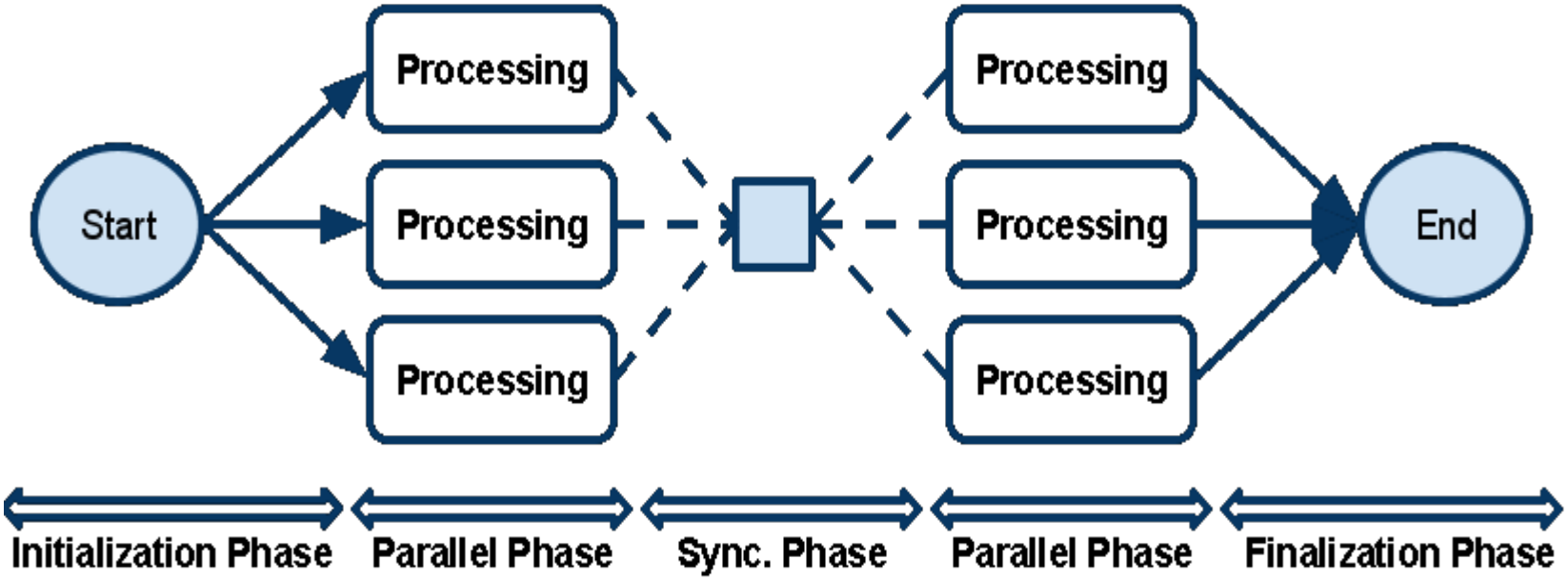


Radix-N24

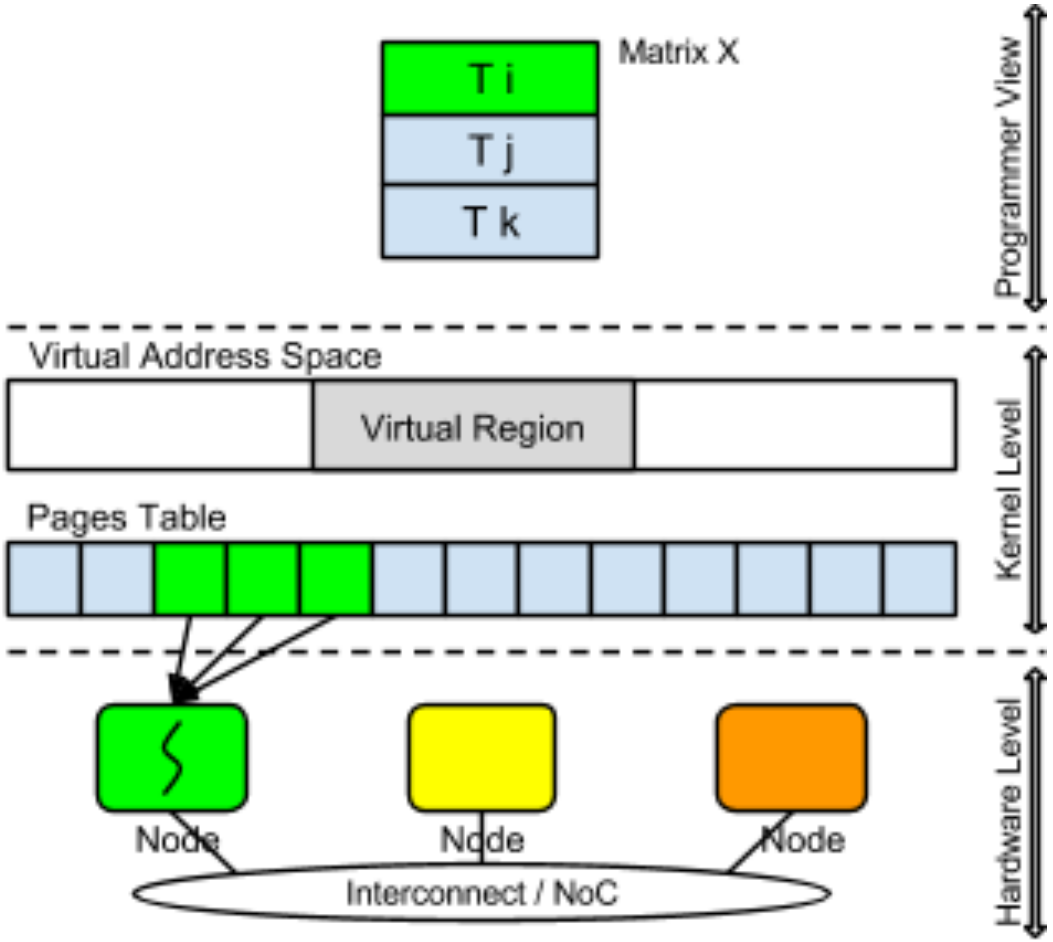


EP1024

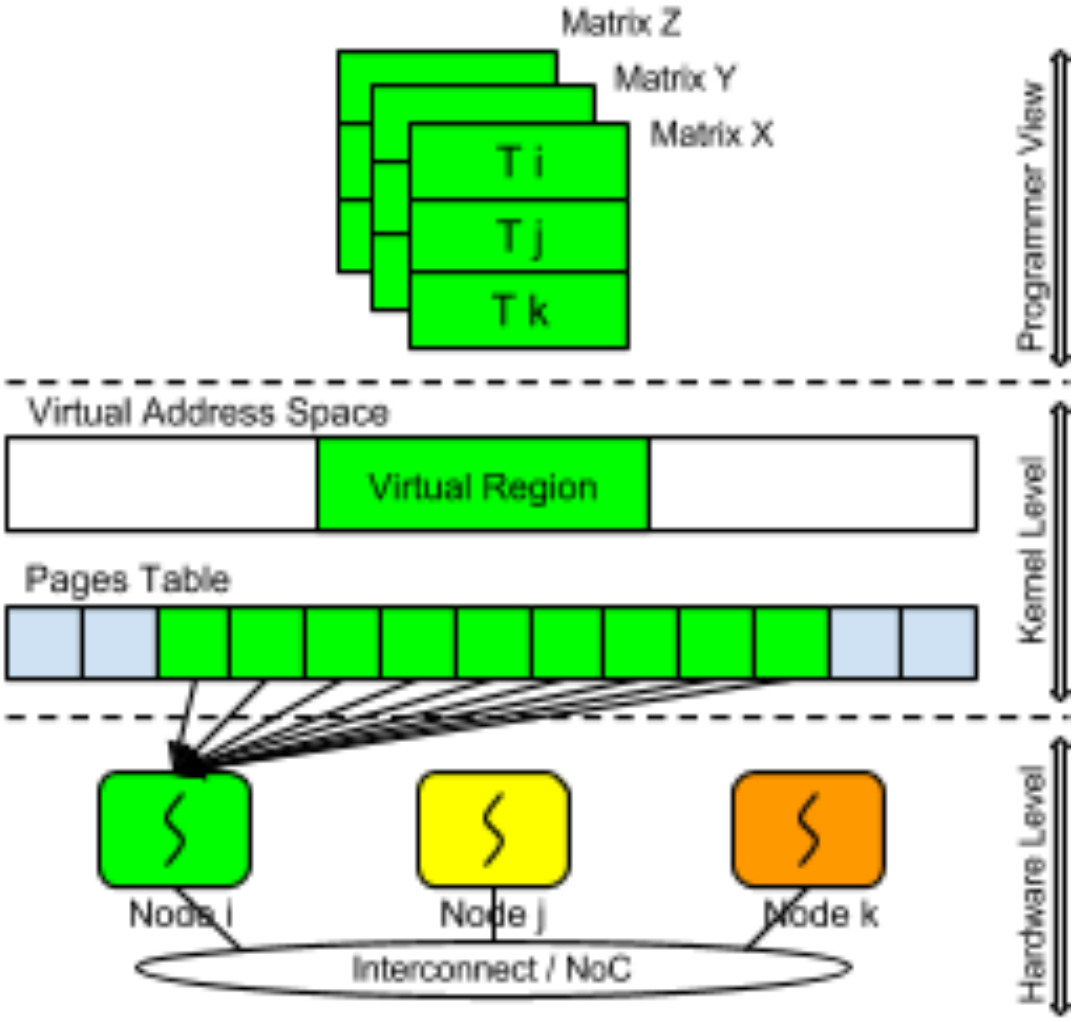
Une application parallèle typique



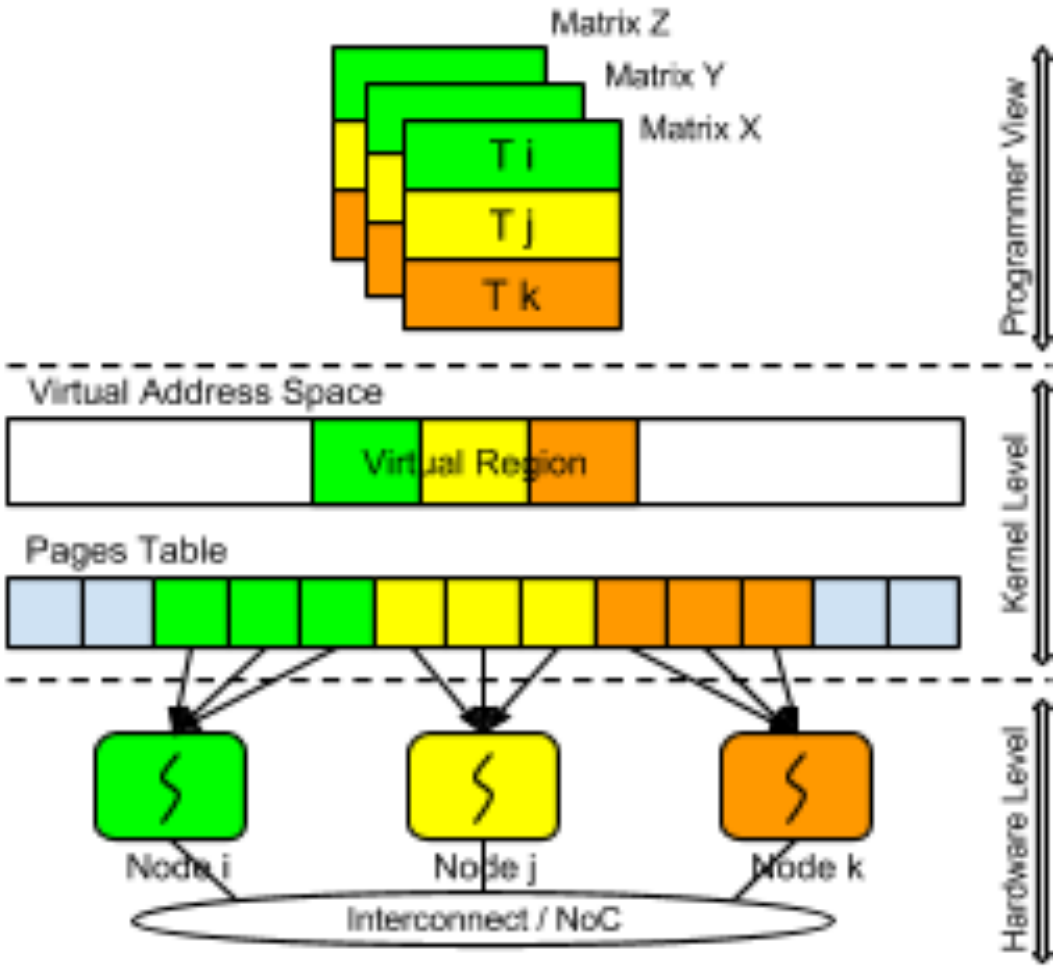
Initialization Phase



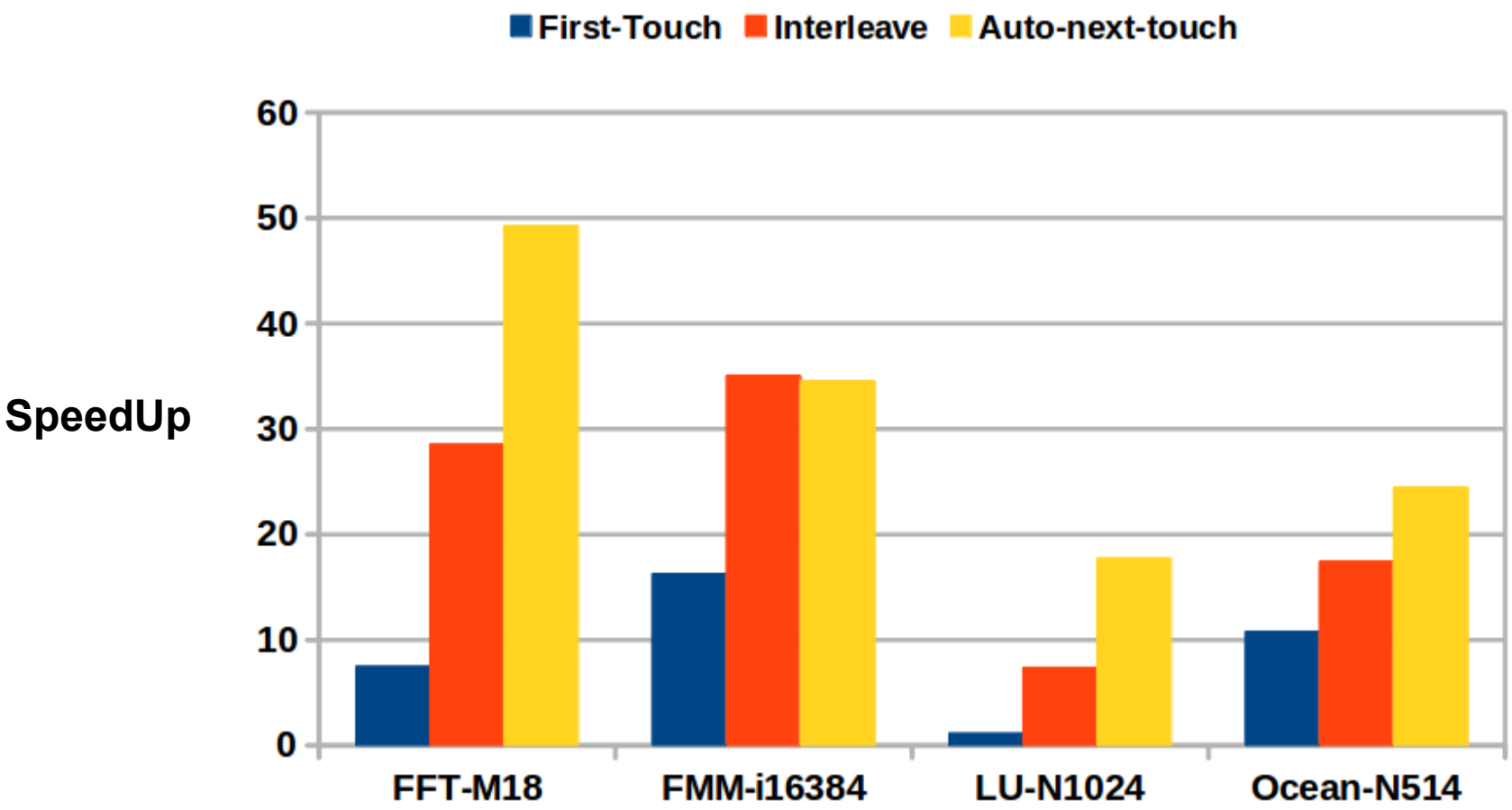
First-Touch Strategy (e.g: Linux, Solaris)



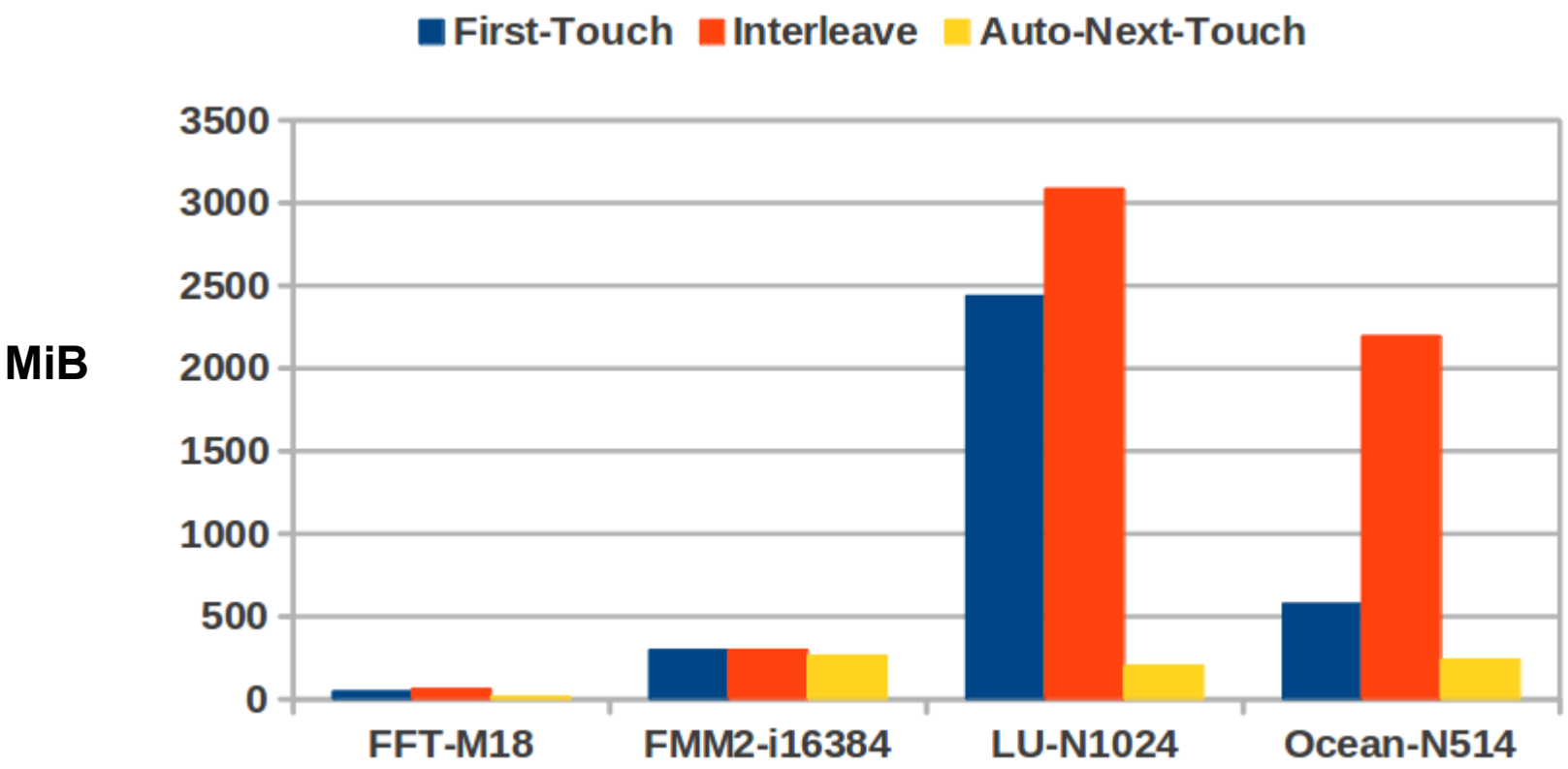
Auto-Next-Touch



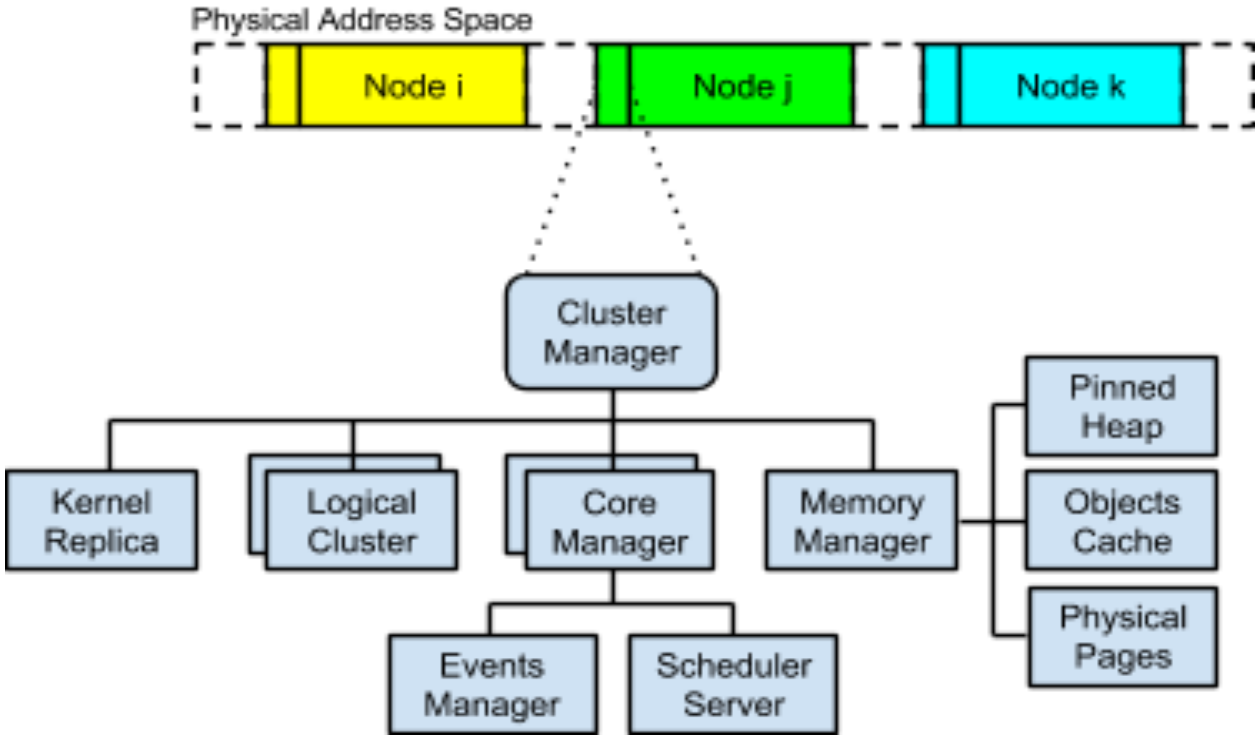
Scalability on TSAR having 64 cores



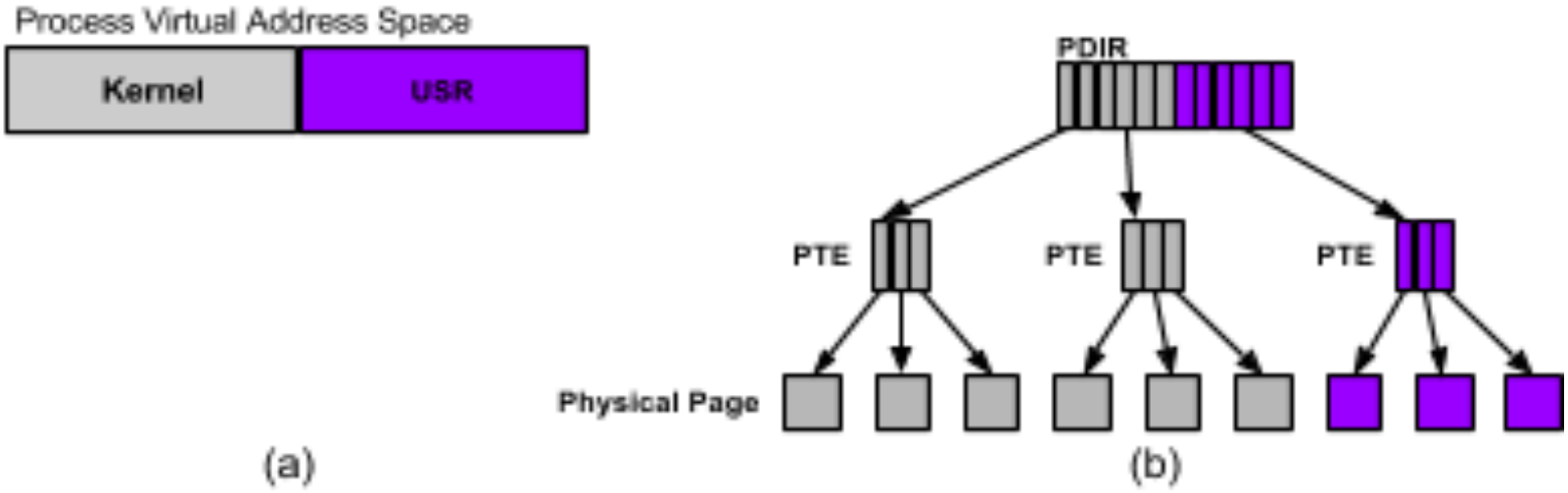
Gain in Remote Traffic on TSAR-64



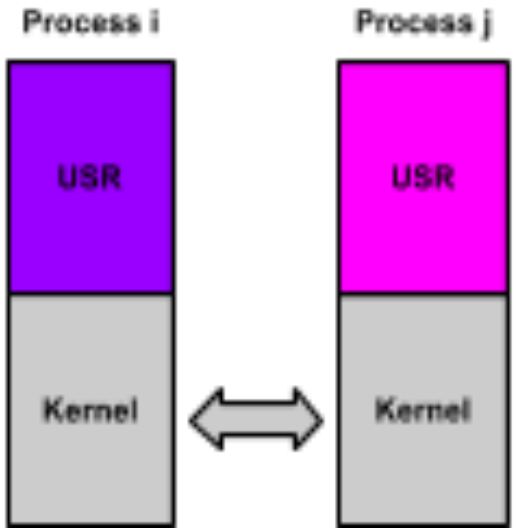
Cluster-Manager

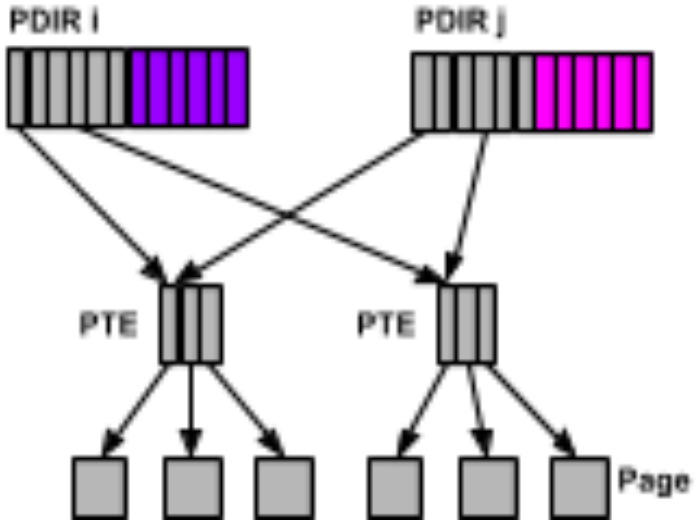


Process Vritual Address Space

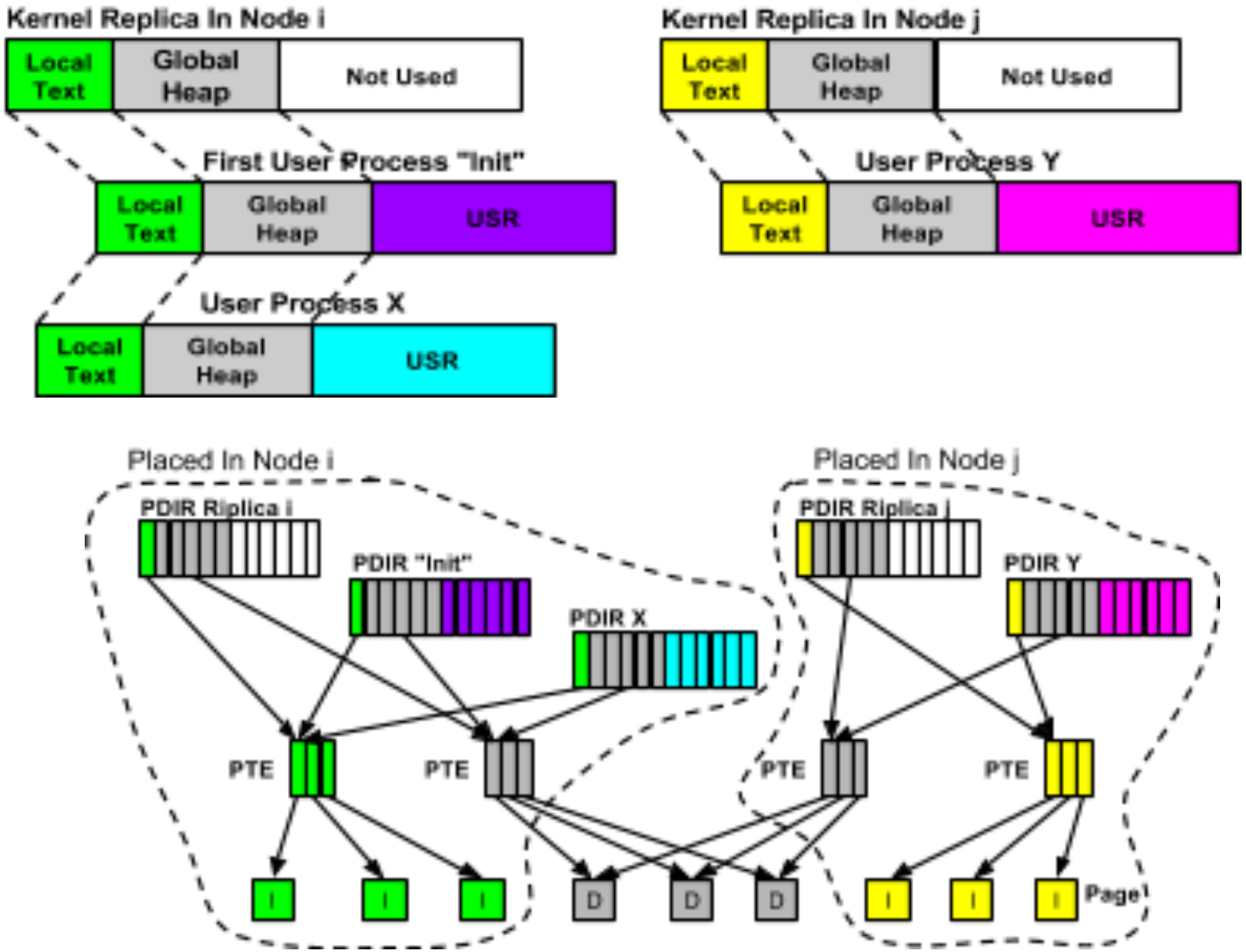


Traditionnal Kernel Mapping

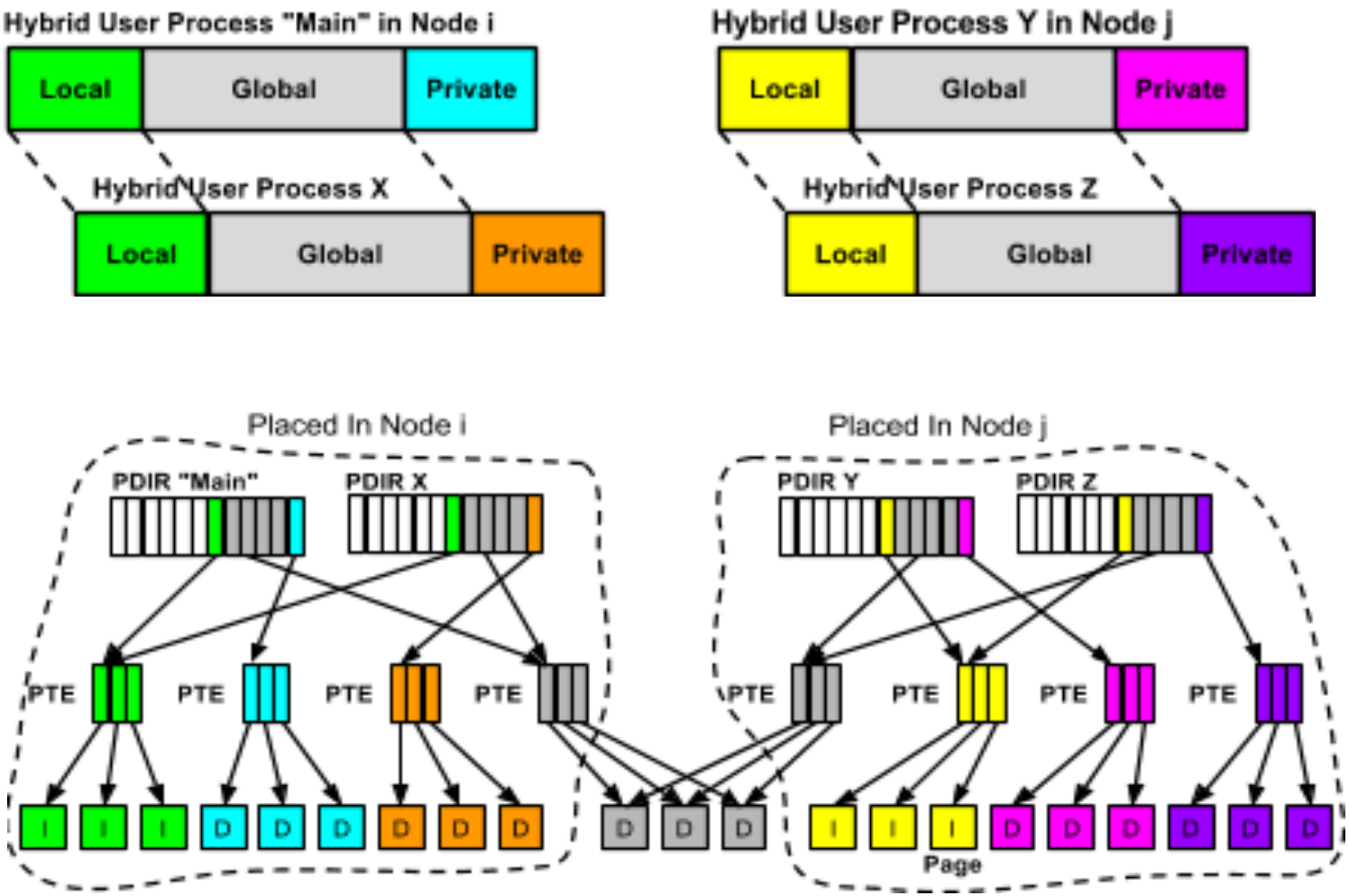




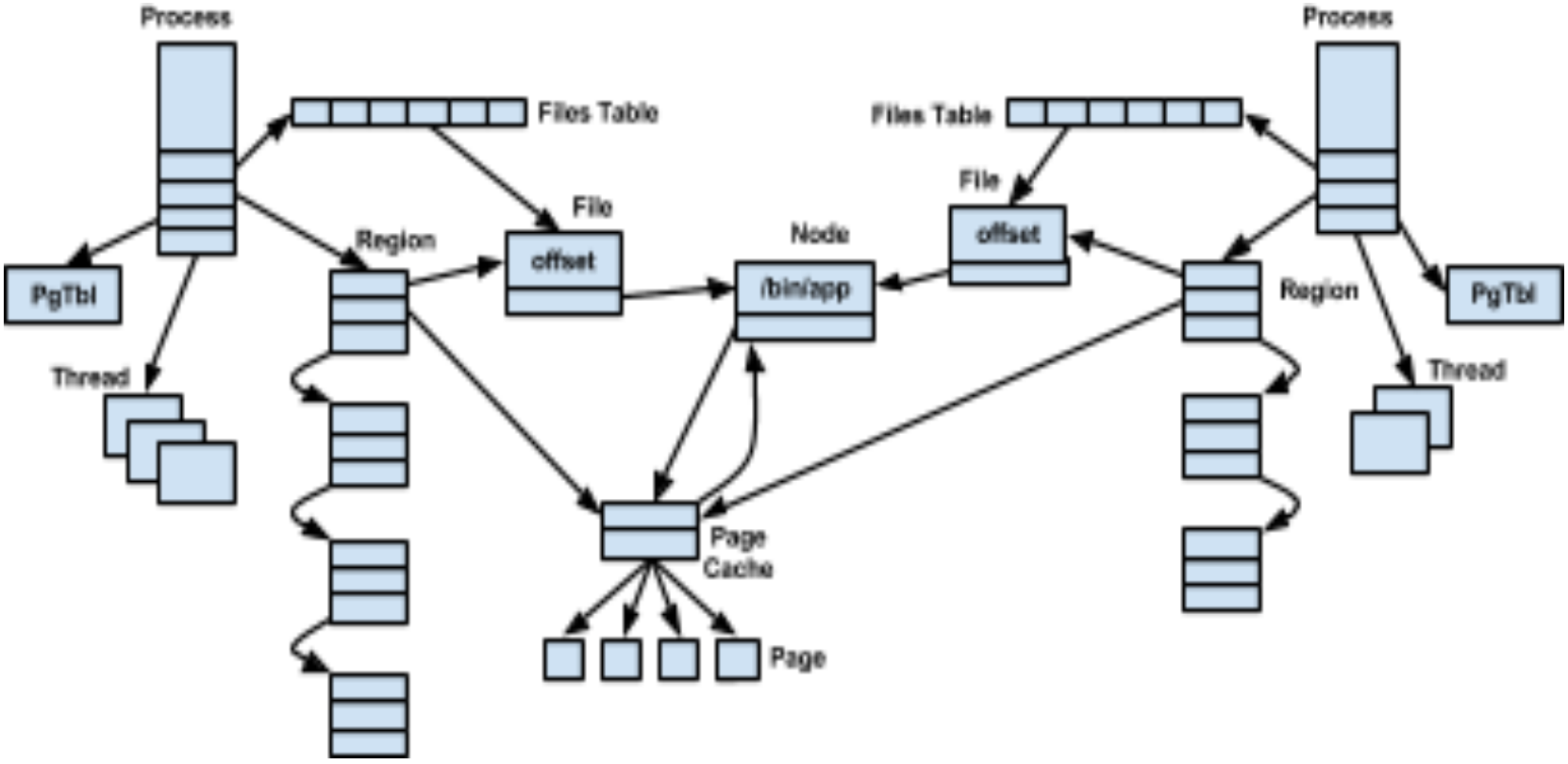
ALMOS: Kernel Replicas



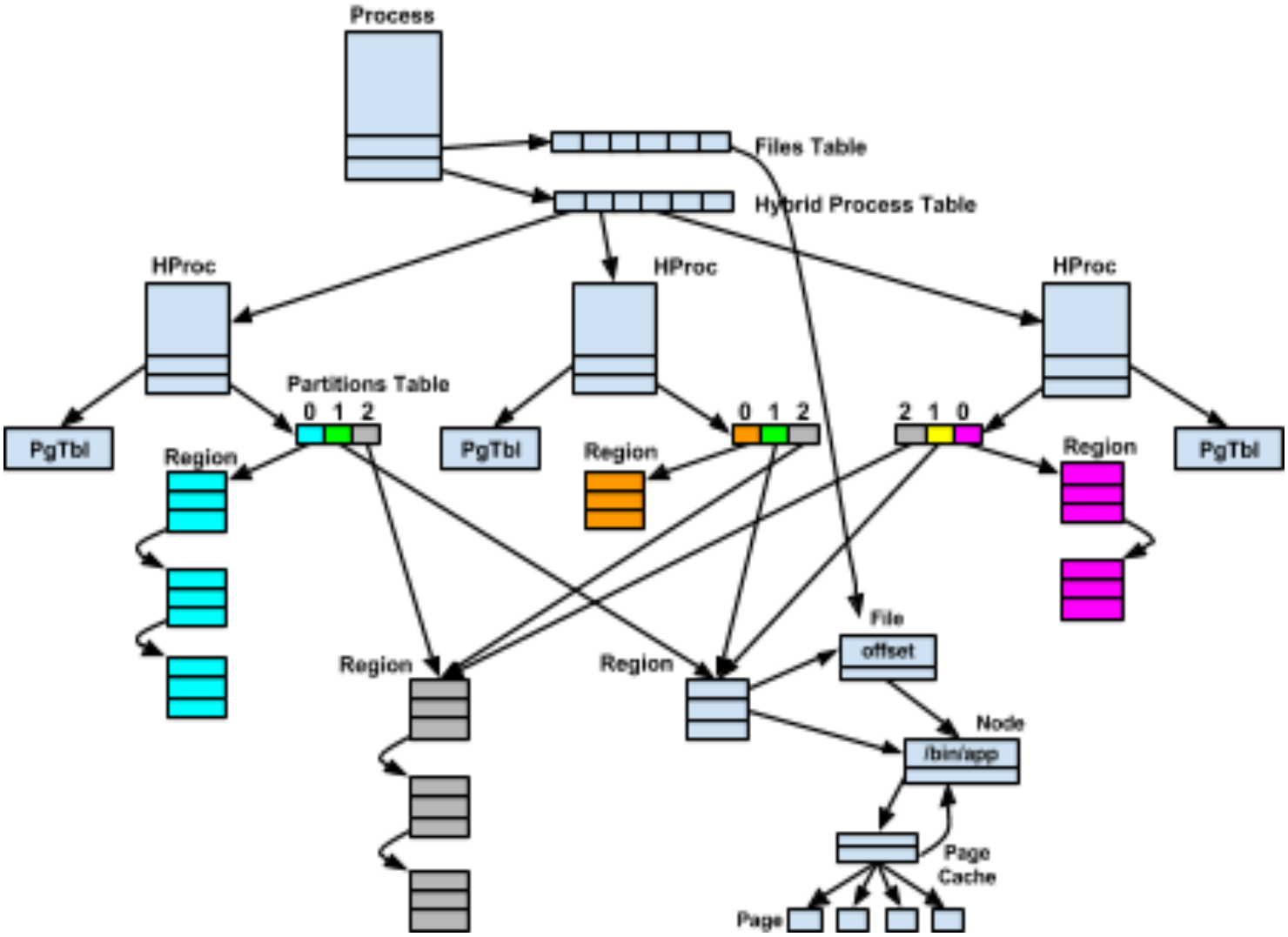
H-Process in More Details



Virtual Address Space Management in Current Kernels (Linux, Solaris, BSD)



ALMOS: H-Process Virtual Address Space Management



ALMOS/TSAR vs Linux/AMD