

# Monitoring du noyau Linux

## sur une architecture NUMA

Kevin Gallardo  
Eric Lombardet  
Pierre-Yves Péneau

Université Pierre et Marie Curie

12 Mai 2014

# Introduction

- problématique:  
architectures NUMA, placement mémoire, performances

# Introduction

- problématique:  
architectures NUMA, placement mémoire, performances
- objectif:  
évaluation d'activité, mesures d'évènements, étude  
comportementale

# Architecture NUMA

## Présentation

### Objectifs

- accélérer les temps de traitement
- répondre aux besoins d'applications spécifiques

# Architecture NUMA

## Présentation

### Objectifs

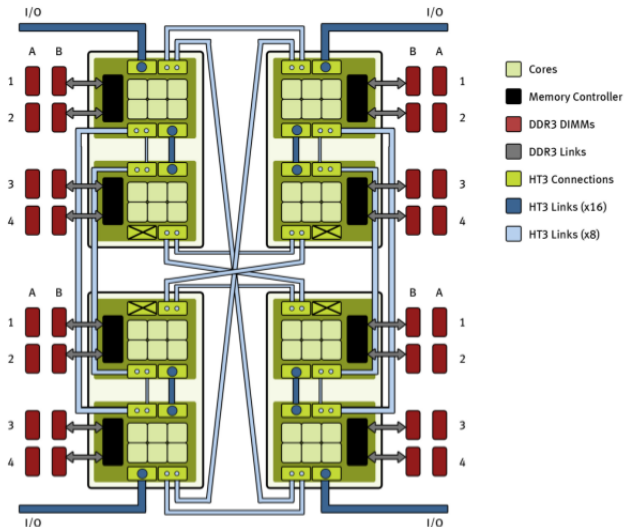
- accélérer les temps de traitement
- répondre aux besoins d'applications spécifiques

### Moyens mis en œuvre

- découpe en noeuds
- placement des contrôleurs d'E/S
- liens d'interconnexions
- mise en place d'une topologie

# Architecture NUMA

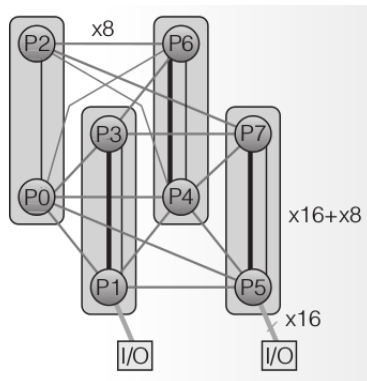
## Vue d'ensemble



# Architecture NUMA

## Enjeux

- placement mémoire
- placement des threads
- activité d'entrées/sorties



# Infrastructure de tests

- utilisation mutualisée du Magny Cour → machines virtuelles
- problème: pas d'IBS avec qemu



# Infrastructure de tests

- utilisation mutualisée du Magny Cour → machines virtuelles
- problème: pas d'IBS avec qemu

## Conséquence

Travail en réel sur le noyau pour 50% du projet

# Monitoring

Qu'est-ce que c'est ?

- étude bas niveau du comportement matériel et système
- très utile pour le débogage ou l'optimisation poussée
- différentes solutions de monitoring existent

# Monitoring

## Instruction Based Sampling - Présentation

- technologie AMD
- informations plus précises car spécifique à une famille de processeur
- problème:w
  - ▶ plus difficile à mettre en place

# Monitoring

## Instruction Based Sampling - Fonctionnement

- tag aléatoirement une instruction
- suivi de l'exécution
- deux types de mesures: fetch/execution sampling

# Monitoring

## Instruction Based Sampling - Utilisation

- beaucoup d'informations remontées par IBS
- sélection des plus utiles: cache hit/miss

# Monitoring

## Instruction Based Sampling - Utilisation

- beaucoup d'informations remontées par IBS
- sélection des plus utiles: cache hit/miss

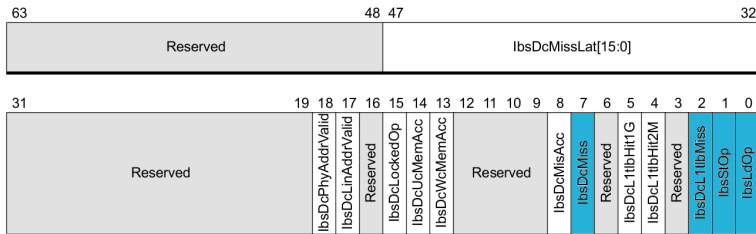


Figure: Schéma du registre MSR IbsOpData3

# Monitoring

## Instruction Based Sampling - Défauts

- overhead: traitement coûteux des mesures
- pas de vision d'ensemble

# Monitoring

## Mise en place

- configuration de l'APIC
  - ▶ informer l'APIC de la présence d'interruptions IBS
  - ▶ à faire pour chaque coeur
- enregistrement d'un handler NMI
  - ▶ appelé à chaque interruption IBS
  - ▶ récolte les informations dans les registres MSR



# Monitoring

## Mise en place

- configuration de l'APIC
  - ▶ informer l'APIC de la présence d'interruptions IBS
  - ▶ à faire pour chaque cœur
- enregistrement d'un handler NMI
  - ▶ appelé à chaque interruption IBS
  - ▶ récolte les informations dans les registres MSR

### Attention

le handler doit être enregistré une et une seule fois au niveau du système

# Mesures sur le noyau Linux

## Chaleur d'un thread

- un compteur représente l'activité d'un thread
- différents critères d'activité:
  - ▶ état: (in)actif
  - ▶ taux d'utilisation mémoire
  - ▶ nombre d'entrées/sorties
  - ▶ communications entre threads
  - ▶ ...

# Mesures sur le noyau Linux

## Méthodes de tri envisagées

- nécessité d'une structure dédiée
- utilisation d'un tableau ou d'une liste chaînée
  - ▶ insertion de nouveaux threads
  - ▶ difficulté à trouver les threads morts
  - ▶ tri peu performant

# Mesures sur le noyau Linux

## Méthodes de tri envisagées

- nécessité d'une structure dédiée
- utilisation d'un tableau ou d'une liste chaînée
  - ▶ insertion de nouveaux threads
  - ▶ difficulté à trouver les threads morts
  - ▶ tri peu performant

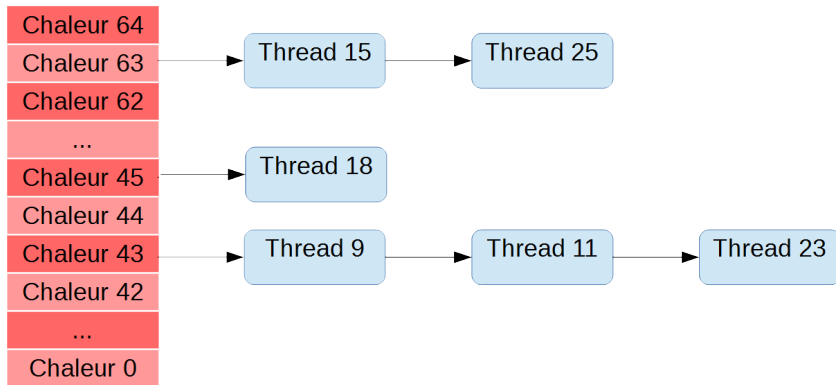
## Conclusion

Solution abandonnée

# Mesures sur le noyau Linux

## Méthodes de tri envisagées

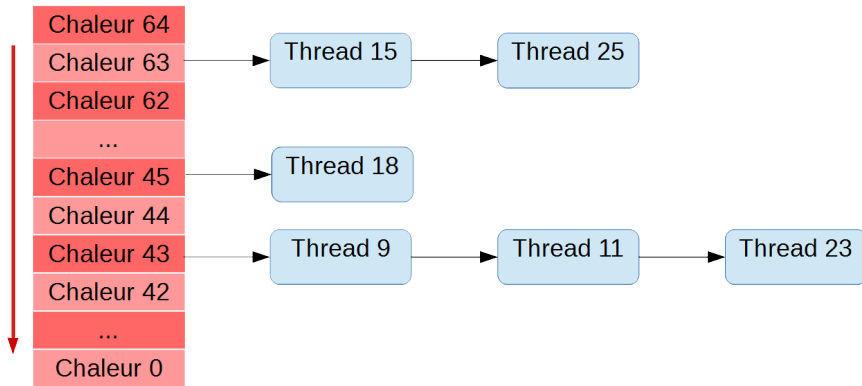
### Utilisation d'un tableau de chaleur



# Mesures sur le noyau Linux

## Méthodes de tri envisagées

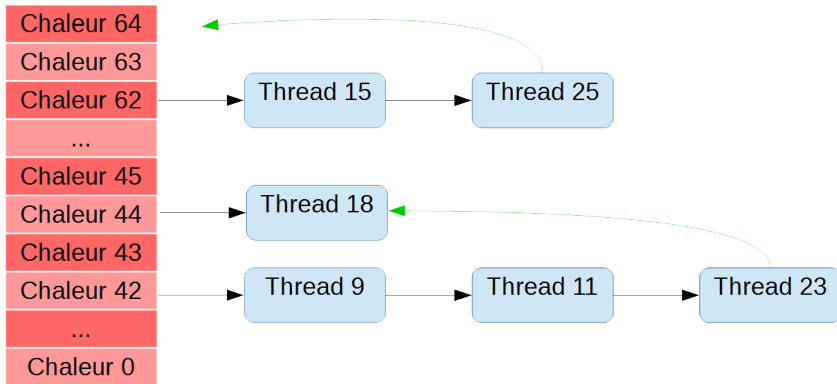
### Utilisation d'un tableau de chaleur



# Mesures sur le noyau Linux

## Méthodes de tri envisagées

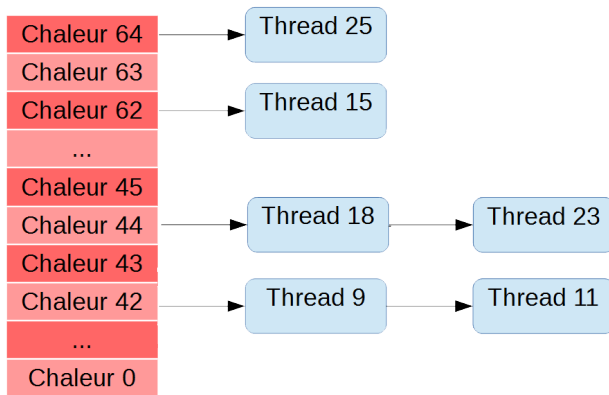
### Utilisation d'un tableau de chaleur



# Mesures sur le noyau Linux

## Méthodes de tri envisagées

### Utilisation d'un tableau de chaleur





# Mesures sur le noyau Linux

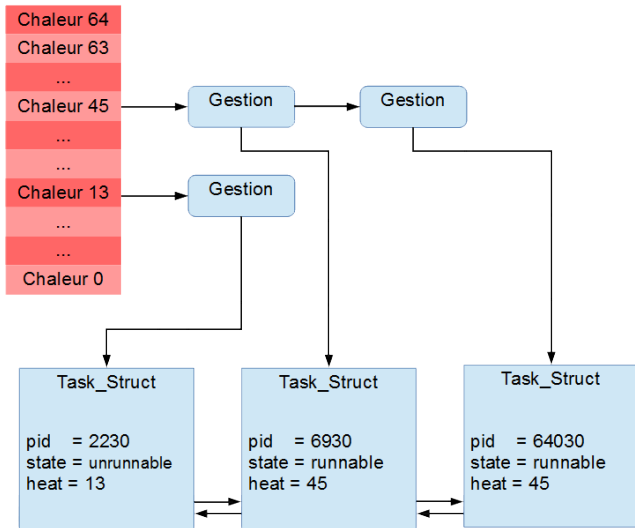
## Solution retenue

- ajout du compteur dans la task\_struct
- on conserve le tableau de chaleur précédent
- structure Gestion pour les listes

```
Struct Gestion  
  
task_struct* proc  
Gestion* next
```

# Mesures sur le noyau Linux

## Solution retenue



# Mesures sur le noyau Linux

## Réalisation

Algorithme:

- 1 stopper IBS
- 2 vider le tableau de chaleur
- 3 parcourir la `task_struct`
  - a si `RUNNING` → incrémentation du compteur de chaleur
  - b sinon décrémentation
- 4 générer le tableau de chaleur
- 5 lancer les mesures sur les threads chauds

# Mesures sur le noyau Linux

Réalisation

## Optimisation

- Utilisation d'un facteur d'incrément et de décrémentation dynamique

a

# Mesures sur le noyau Linux

Réalisation

## Optimisation

- Utilisation d'un facteur d'incrément et de décrémentation dynamique

a

## Problèmes

- pas d'IBS avec qemu

# Conclusion

## Réalisation

20/20 ou chinois tuer et manger ta famille !



*"Tu veux un bol de riz ?"*