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

- objectifs:
  évaluation d'activité, mesures d'évènements, étude comportementale

Architecture NUMA

Présentation

Objectifs

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

Architecture NUMA

Présentation

Objectifs

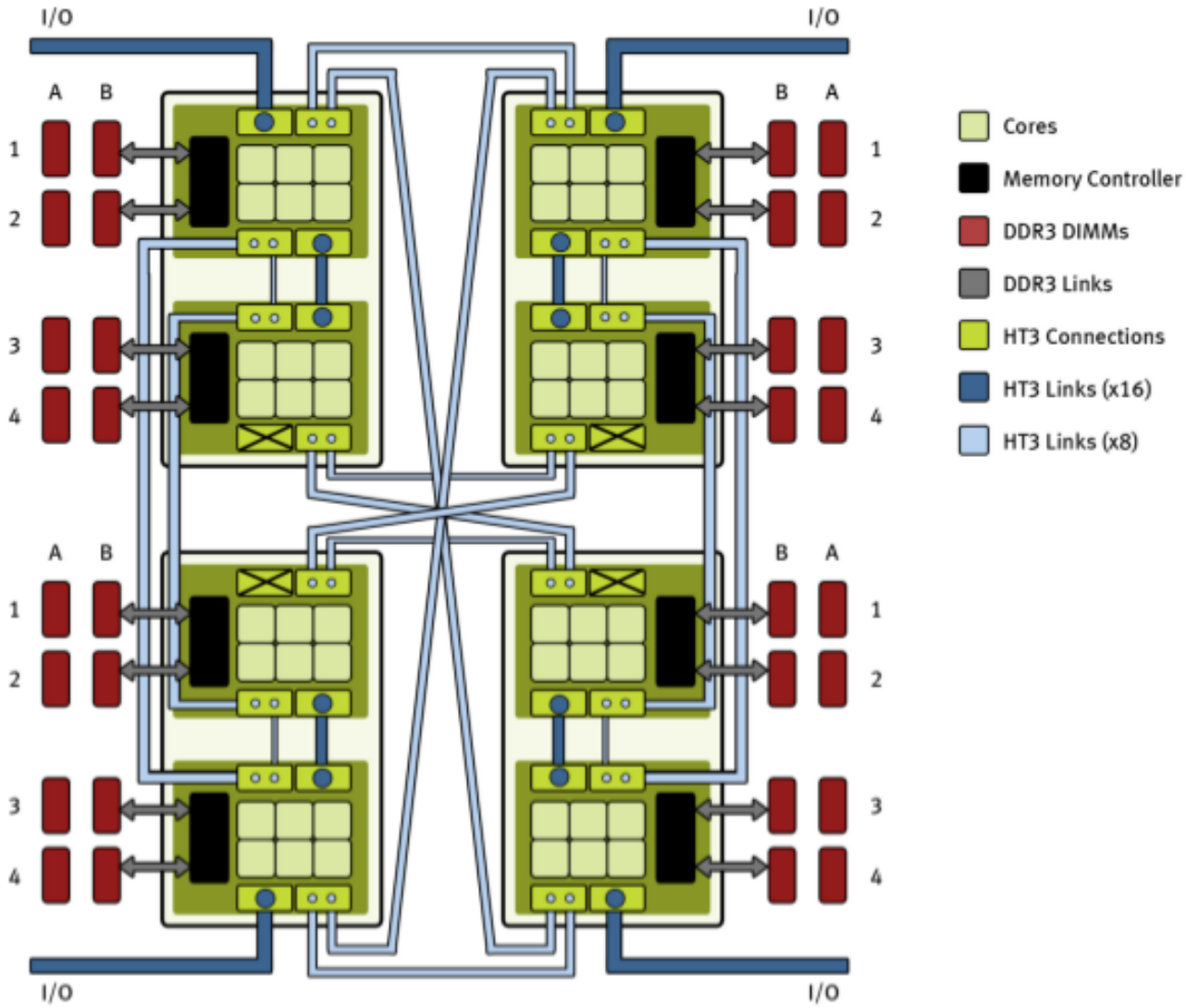
- accélerer les temps de traitement
- répondre aux besoins d'applications spécifiques

Moyens mis en œuvre

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

Architecture NUMA

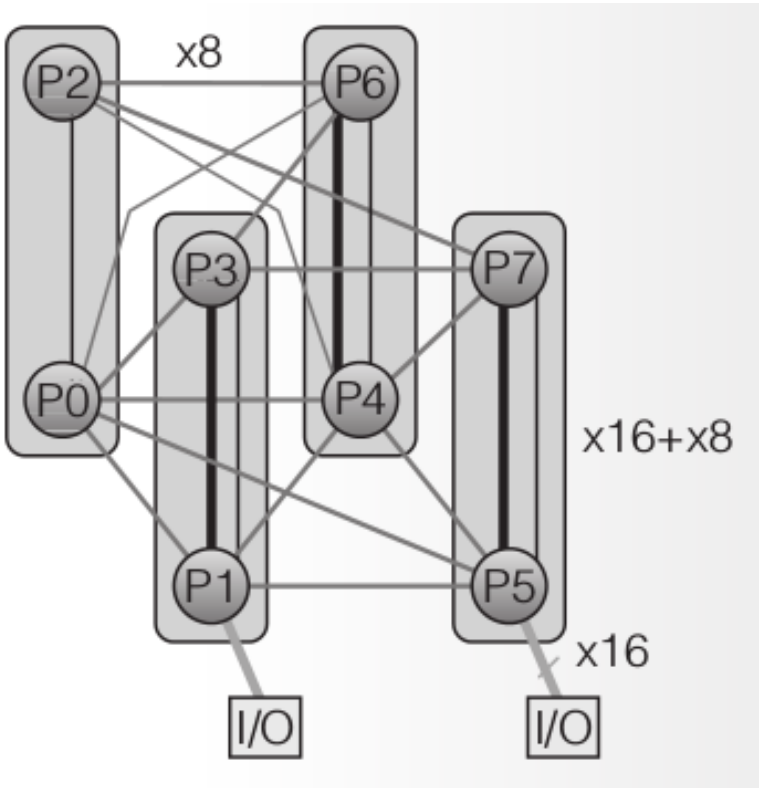
Vue d'ensemble



Architecture NUMA

Enjeux

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



Infrastructure de tests

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

Infrastructure de tests

- utilisation mutualisée du Magny Cour $\rightarrow$ 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 débugage 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:
  - 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

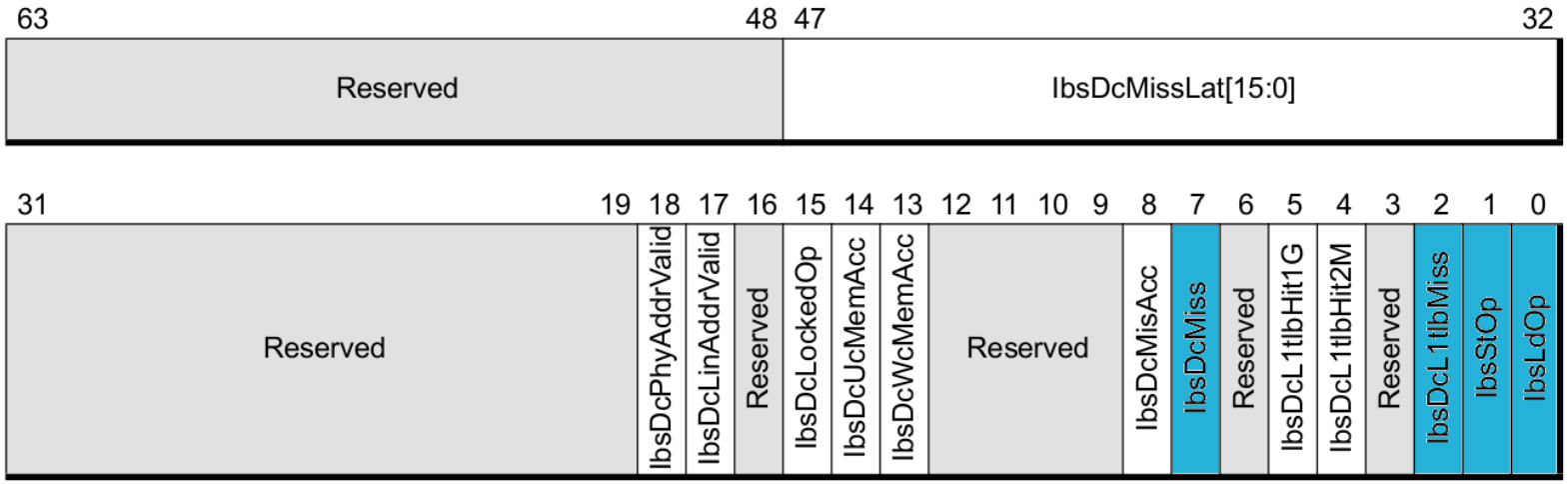


Figure : Schéma du registre MSR IbsOpData3

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 coeur
- 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
  - commnunications entre threads
  - ...

Mesures sur le noyau Linux

Méthodes de tri envisagées

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

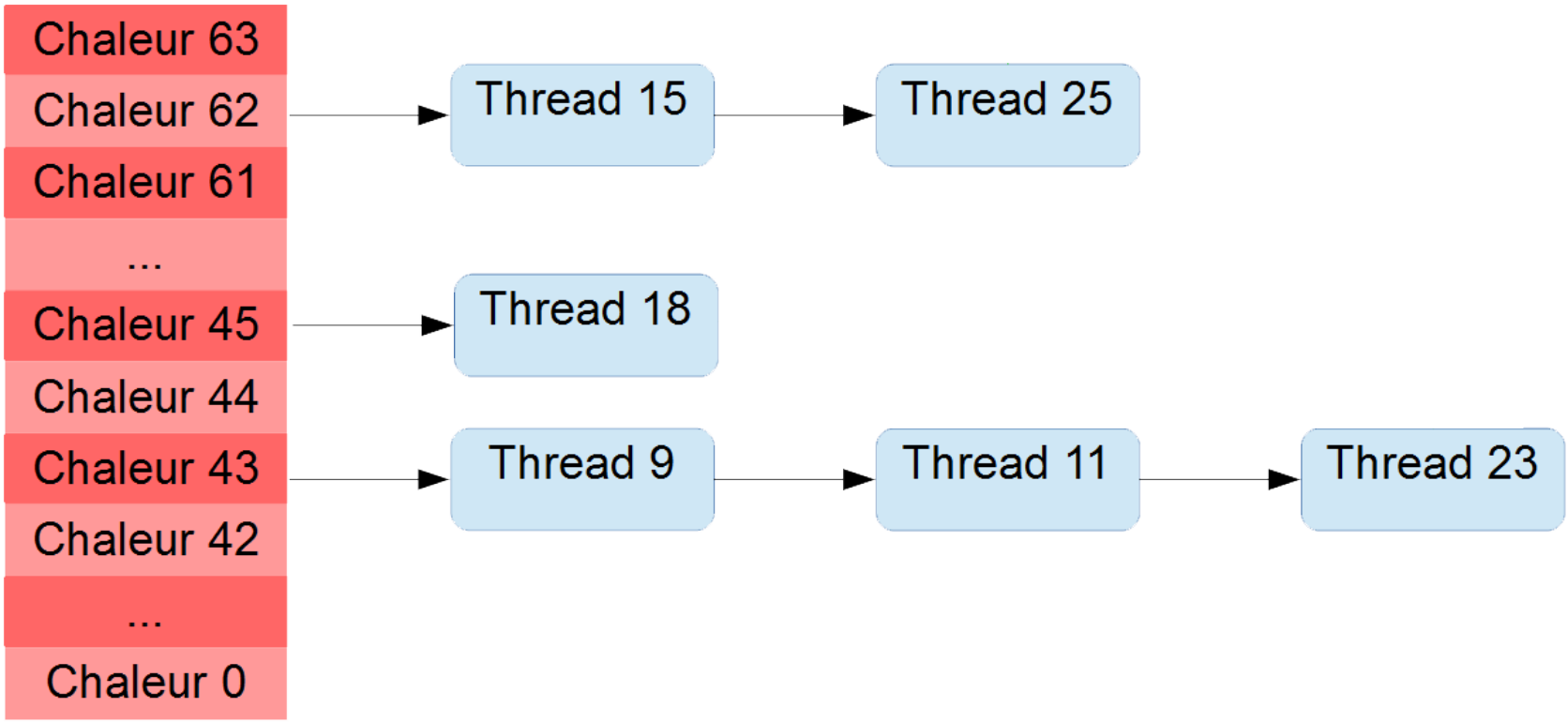
Conclusion

Solution abandonnée

Mesures sur le noyau Linux

Méthodes de tri envisagées

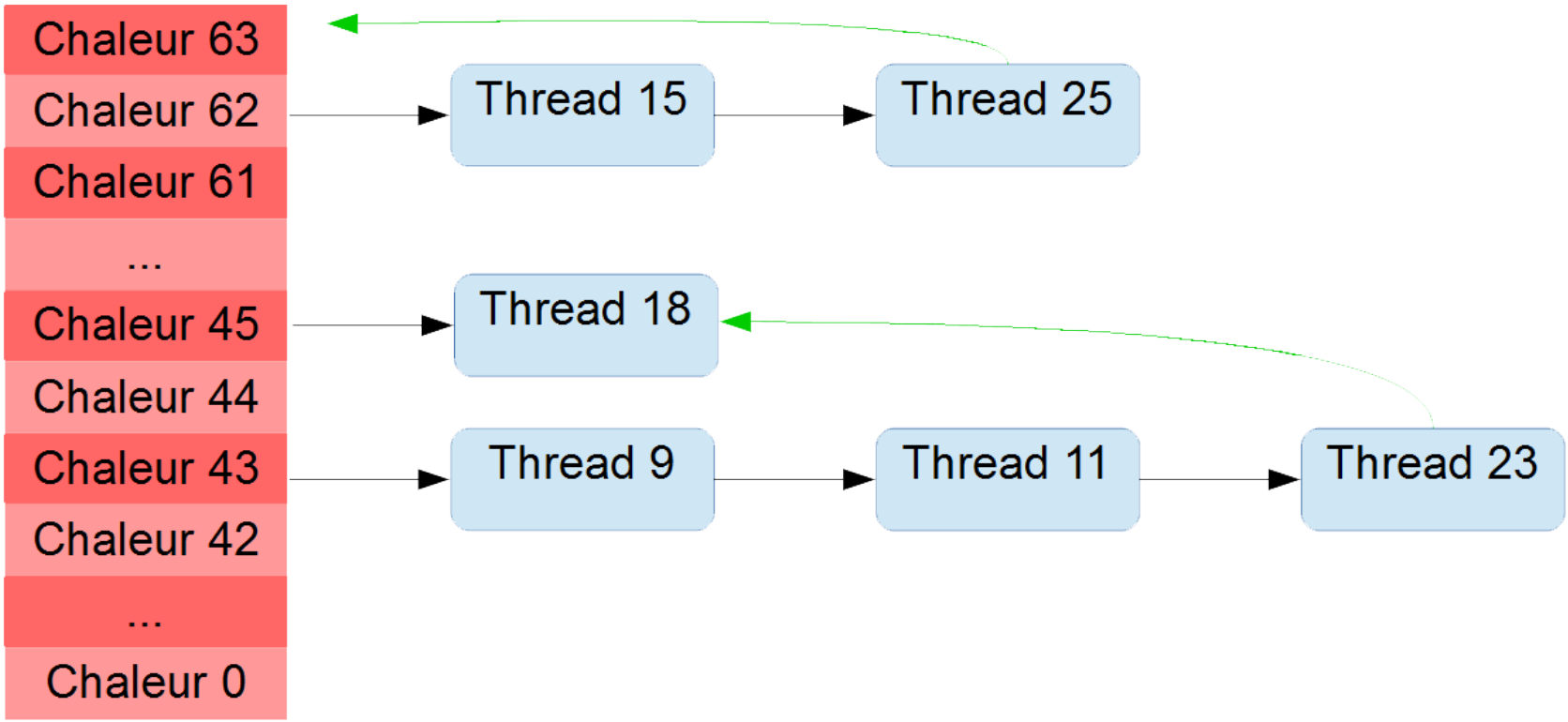
Utilisation d'un tableau de chaleur



Mesures sur le noyau Linux

Méthodes de tri envisagées

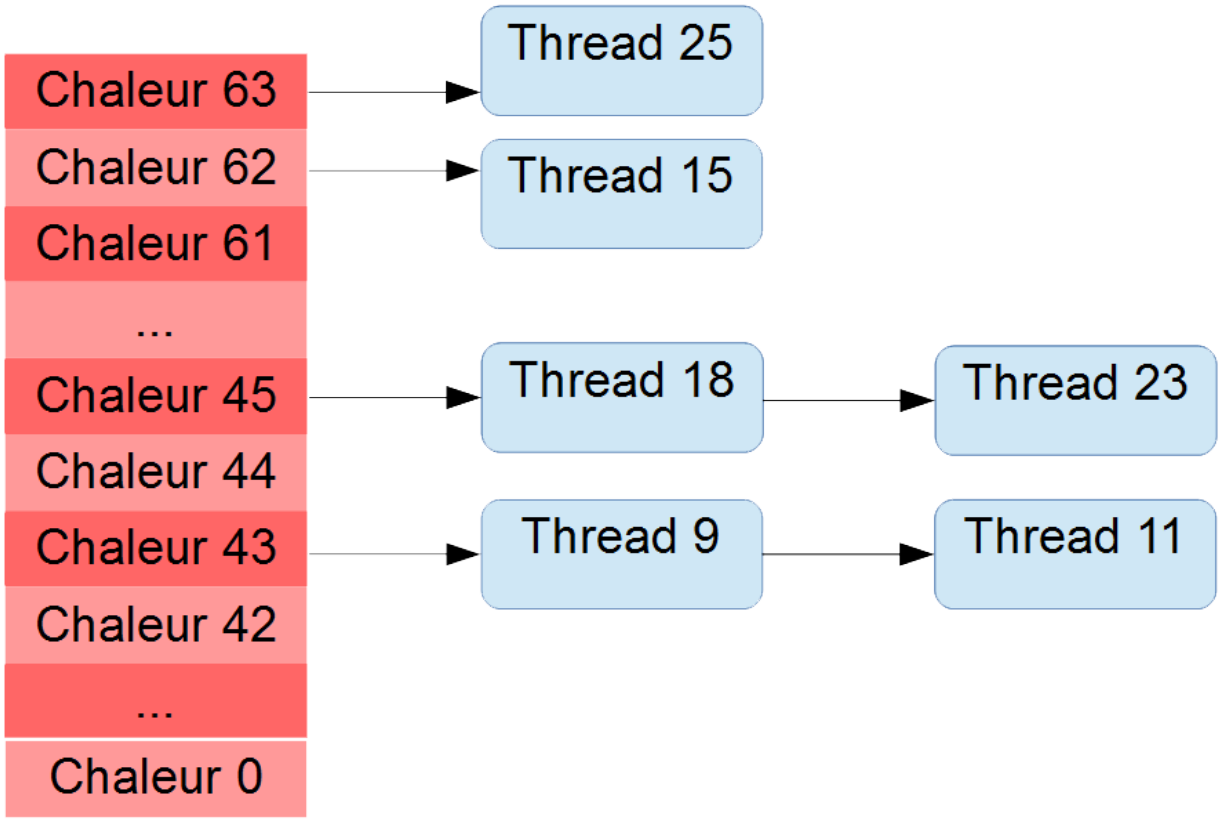
Utilisation d'un tableau de chaleur



Mesures sur le noyau Linux

Méthodes de tri envisagées

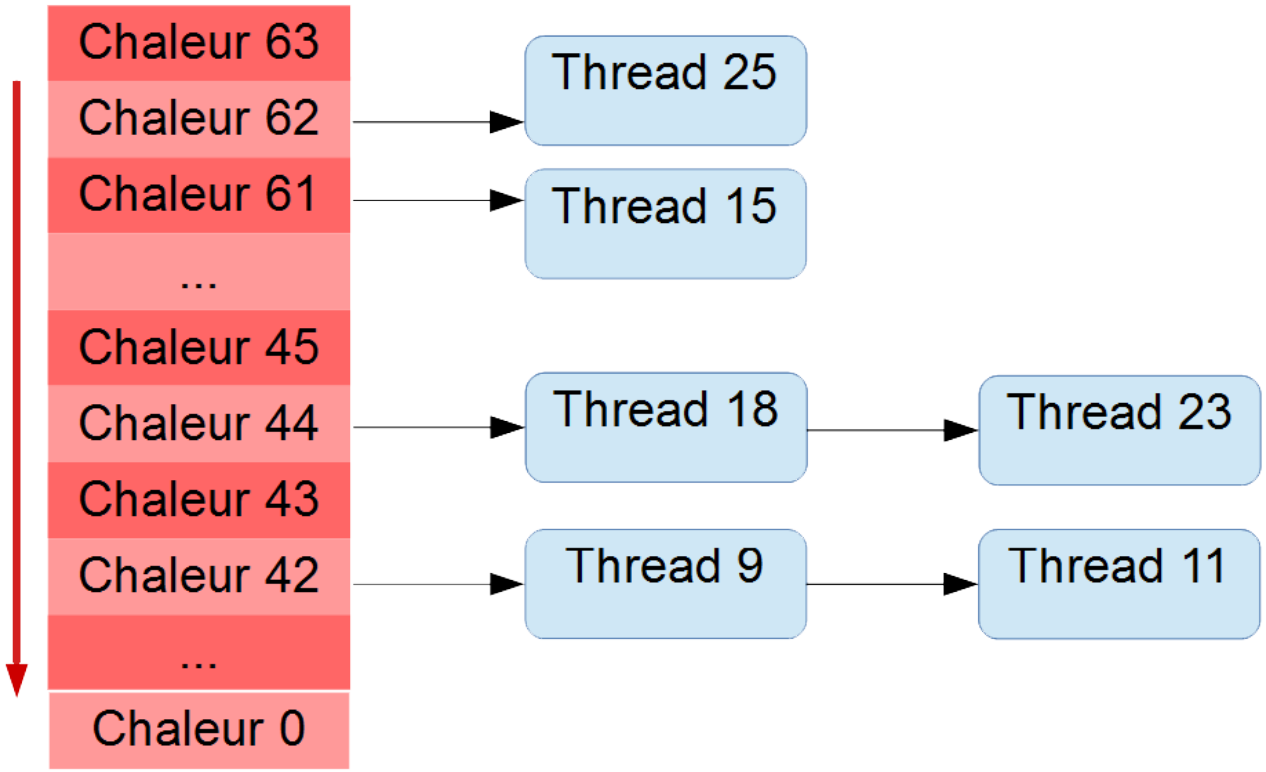
Utilisation d'un tableau de chaleur



Mesures sur le noyau Linux

Méthodes de tri envisagées

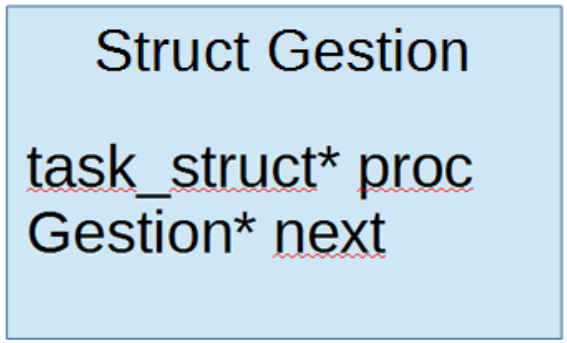
Utilisation d'un tableau de chaleur



Mesures sur le noyau Linux

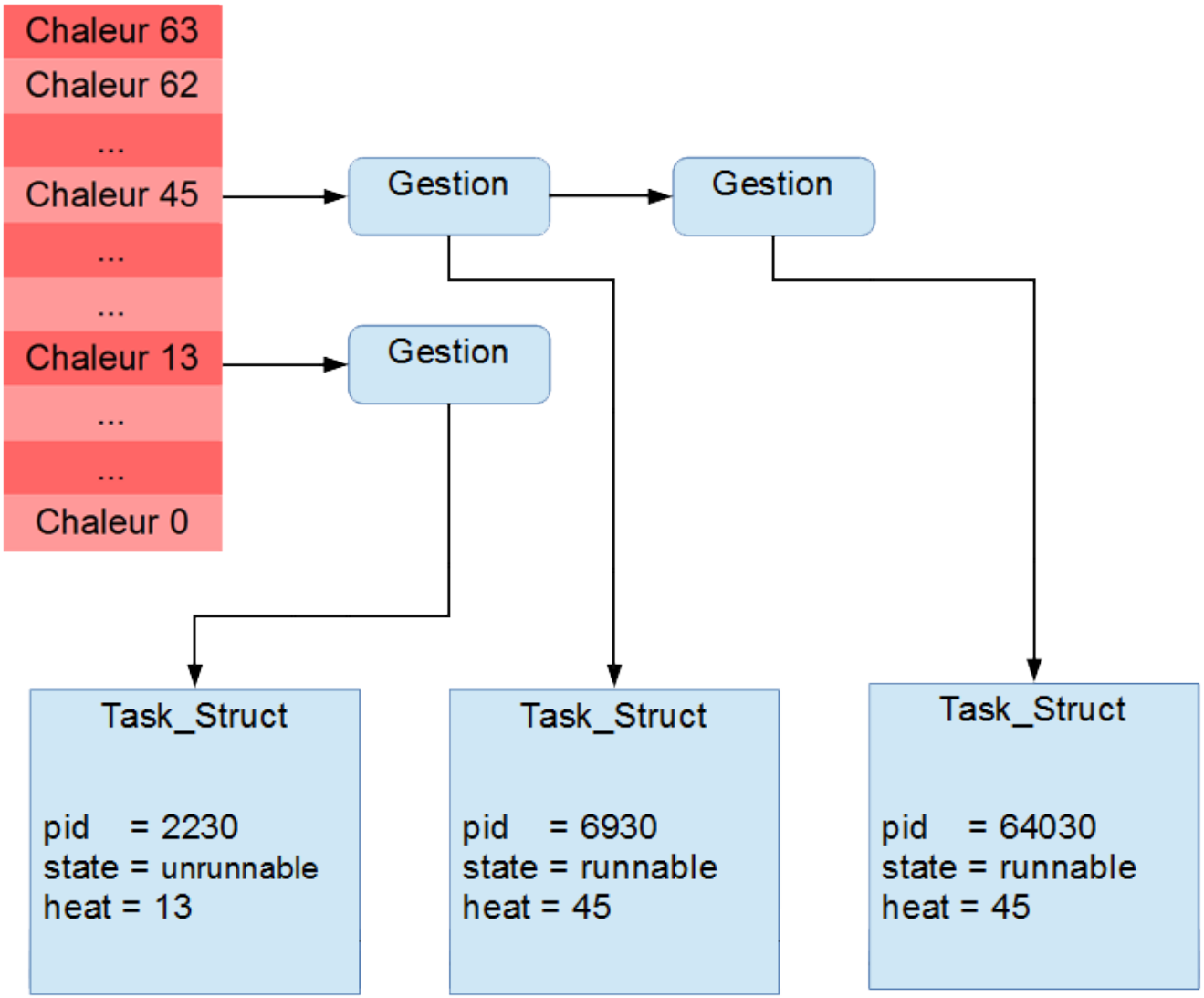
Solution retenue

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



Mesures sur le noyau Linux

Solution retenue



Mesures sur le noyau Linux

Réalisation

Algorithme:

1. parcourir la task_struct
   a. si RUNNING $\rightarrow$ incrémentation du compteur de chaleur
   b. sinon décrémentation
   2. stopper IBS
   3. vider le tableau de chaleurs
4. générer le tableau de chaleurs
5. lancer les mesures sur les threads chauds

Mesures sur le noyau Linux

Réalisation

Optimisation

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

Mesures sur le noyau Linux

Réalisation

Optimisation

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

Problèmes

- pas d'IBS avec qemu $\rightarrow$ merge impossible sur Magny Cour

Conclusion

Apports personnels

- beaucoup de connaissances acquises
- utile pour l'année prochaine
- découverte d'une nouvelle architecture prometteuse

Conclusion

Apports personnels

- beaucoup de connaissances acquises
- utile pour l'année prochaine
- découverte d'une nouvelle architecture prometteuse

Ce qu'il reste à faire

- merger les deux parties du projet sur Magny Cour
- mettre en place un traitement des données
- améliorer l'algorithme de tri d'activités