

Programmation parallèle sans verrous

Félix-Antoine Ouellet

Université de Sherbrooke

16 octobre 2014

- 1 Motivation
- 2 Définition
- 3 Techniques
- 4 Modèle mémoire
- 5 Conclusion

Plan

- 1 Motivation
- 2 Définition
- 3 Techniques
- 4 Modèle mémoire
- 5 Conclusion

Plan

- 1 Motivation
- 2 Définition
- 3 Techniques
- 4 Modèle mémoire
- 5 Conclusion

Définitions

Il existe diverses définitions selon le niveau de garanties fourni.

- Sans attente
- Sans verrous
- Sans obstruction

Sans attente

Chaque fil d'exécution s'exécute dans un nombre fini d'étapes sans égard pour des facteurs externes.

Sans attente

Chaque fil d'exécution s'exécute dans un nombre fini d'étapes sans égard pour des facteurs externes.

Exemple:

```
void IncrementRefCounter(Object *obj)
{
    atomic_increment(obj->rc);
}
```

Sans verrous

Le système en entier va continuer de progresser malgré que certains fils d'exécution ne progressent pas.

Sans verrous

Le système en entier va continuer de progresser malgré que certains fils d'exécution ne progressent pas.

Exemple:

```
void StackPush(Stack *s, Node *n) {  
    Node* head;  
    do {  
        head = s->head;  
        n->next = head;  
    }  
    while (!CompareExchange(s->head, head, n));  
}
```

Sans obstruction

Un fil d'exécution exécuter en isolation va terminer dans un nombre fini d'étapes.

Sans obstruction

Un fil d'exécution exécuter en isolation va terminer dans un nombre fini d'étapes.

Exemple:

?

Plan

1 Motivation

2 Définition

3 **Techniques**

4 Modèle mémoire

5 Conclusion

Read-Copy-Update

Compare-And-Swap

Haut niveau

```
int CompareAndSwap(int *reg, int oldVal,  
                  int newVal) {  
    int r = *reg;  
    if (*reg == oldVal)  
        *reg = newVal;  
    return r;  
}
```

Compare-And-Swap

Pratique

```
void LockFreeQueue::push(Node *newHead) {  
    for (;;) {  
        Node *oldHead = m_Head;  
        newHead->next = oldHead;  
        if (CompareAndSwap(&m_Head, newHead,  
                           oldHead) == oldHead)  
            return;  
    }  
}
```

Compare-And-Swap

Assembleur

CMPXCHG

Problème ABA

Description

Problème ABA

Exemple - Code

```
void LockFreeQueue::push(Node *newHead) {  
    for (;;) {  
        Node *oldHead = m_Head;  
        newHead->next = oldHead;  
        if (CompareAndSwap(&m_Head, newHead,  
                           oldHead) == oldHead)  
            return;  
    }  
}
```

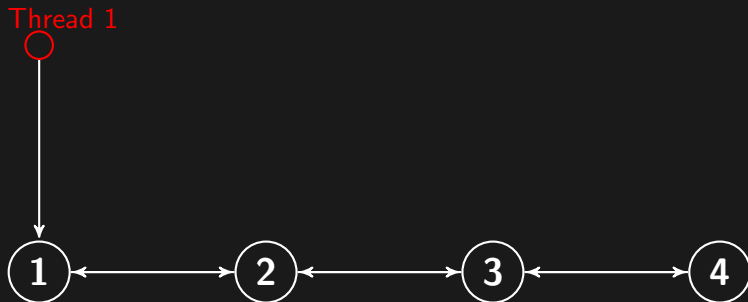
Problème ABA

Exemple - Exécution



Problème ABA

Exemple - Exécution



Problème ABA

Exemple - Exécution

Thread 1 Thread 2



Problème ABA

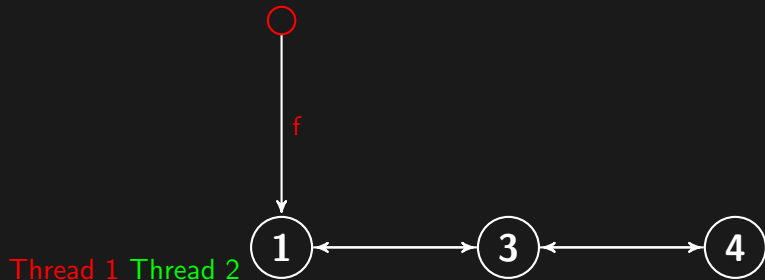
Exemple - Exécution

Thread 1 Thread 2



Problème ABA

Exemple - Exécution



Plan

- 1 Motivation
- 2 Définition
- 3 Techniques
- 4 Modèle mémoire**
- 5 Conclusion

Réordonnement mémoire

Pourquoi?

Réordonnement mémoire

Pourquoi?

- Compilateur

Réordonnement mémoire

Pourquoi?

- Compilateur
- Processeur

Consistence séquentielle

Plan

- 1 Motivation
- 2 Définition
- 3 Techniques
- 4 Modèle mémoire
- 5 Conclusion

Conclusion

- On jongle avec des lames de rasoir

Conclusion

- On jongle avec des lames de rasoir
- Il faut connaître ses outils logiciels et matériels