

# Vol de tâches à la TBB (présentation simplifiée)

28 novembre ou 5 décembre 2017

Soit l'extrait de code suivant, qu'on suppose exécuté avec deux (2) *threads*

```
const int n = 128;

int a[n];

parallel_for(
    blocked_range<size_t>(0, n, 16),
    [&]( blocked_range<size_t> r ) {
        for( int i = r.begin(); i < r.end(); i++ ) {
            traiter( a[i] );
        }
    } );
```

Les diapositives suivantes décrivent l'état des *deques* (sacs) de tâches des *threads*  $T_0$  et  $T_1$

$T_0$ : tâche en cours de traitement (ou nil)

[ ]

↑

Tête

$T_1$ : tâche en cours de traitement (ou nil)

[ ]

↑

Tête

# État initial

$\mathcal{T}_0: [0, 128)$

[ ]

↑

Tête

$\mathcal{T}_1: \text{nil}$

[ ]

↑

Tête

# $T_0$ crée deux tâches en décomposant sa tâche

$T_0$ : nil

[ [0, 64) [64, 128) ]

↑

Tête

$T_1$ : nil

[ ]

↑

Tête

# $T_0$ obtient une tâche

$T_0$ : [0, 64)

[ [64, 128) ]

↑

Tête

$T_1$ : nil

[ ]

↑

Tête

# $T_0$ décompose sa tâche

$T_0$ : nil

[ [0, 32) [32, 64) [64, 128) ]

↑

Tête

$T_1$ : nil

[ ]

↑

Tête

$T_0$  obtient une tâche

$T_0: [0, 32)$

[ [32, 64) [64, 128) ]

↑

Tête

$T_1: \text{nil}$

[ ]

↑

Tête

# $T_0$ décompose sa tâche

$T_0$ : nil

[ [0, 16) [16, 32) [32, 64) [64, 128) ]

↑

Tête

$T_1$ : nil

[ ]

↑

Tête

# $T_1$ vole une tâche

$T_0$ : nil

[ [0, 16) [16, 32) [32, 64) ]

↑

Tête

$T_1$ : [64, 128)

[ ]

↑

Tête

# $T_1$ décompose sa tâche

$T_0$ : nil

[ [0, 16) [16, 32) [32, 64) ]

↑

Tête

$T_1$ : nil

[ [64, 96) [96, 128) ]

↑

Tête

$T_0$  obtient une tâche

$T_0$ : [0, 16)

[ [16, 32) [32, 64) ]

↑

Tête

$T_1$ : nil

[ [64, 96) [96, 128) ]

↑

Tête

# $T_1$ obtient une tâche

$T_0$ : [0, 16)

[ [16, 32) [32, 64) ]

↑

Tête

$T_1$ : [64, 96)

[ [96, 128) ]

↑

Tête

# $T_0$ termine sa tâche

$T_0$ : nil

[ [16, 32) [32, 64) ]

↑

Tête

$T_1$ : [64, 96)

[ [96, 128) ]

↑

Tête

$T_0$  obtient une tâche

$T_0$ : [16, 32)

[ [32, 64) ]

↑

Tête

$T_1$ : [64, 96)

[ [96, 128) ]

↑

Tête

# $T_0$ termine sa tâche

$T_0$ : nil

[ [32, 64) ]

↑

Tête

$T_1$ : [64, 96)

[ [96, 128) ]

↑

Tête

# $T_1$ décompose sa tâche

$T_0$ : nil

[ [32, 64) ]

↑

Tête

$T_1$ : nil

[ [96, 112) [112, 128) ]

↑

Tête

# Arbre d'activation résultant des traitements de tâches des diapositives précédentes

