

Übung 6: Task Parallel Library

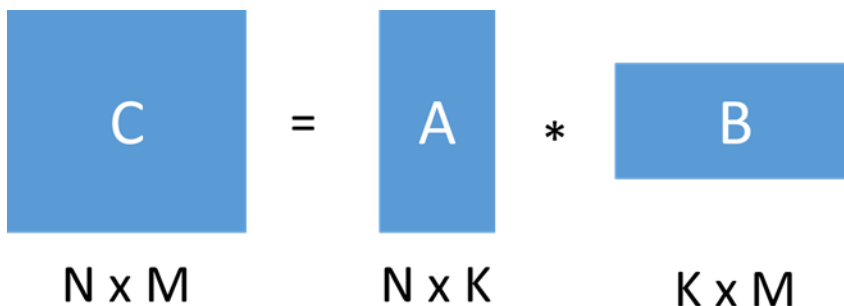
Zielsetzung:

- Erfahrung mit der .NET Task Parallel Library sammeln.
- Verschiedene Abstraktionen der TPL verstehen und einsetzen.
- Mit Taskparallelität und Datenparallelität Algorithmen beschleunigen.
- TPL Programmen debuggen und analysieren.

Aufgabe 1: Parallel For Loop

In der Vorlage (Visual Studio Solution) finden Sie eine sequentielle Matrix Multiplikation. Beschleunigen Sie den Algorithmus mittels TPL Datenparallelität. Sie können dazu eine parallele for-Schleife benutzen.

Bei der Matrix Multiplikation $C = A * B$ werden die Matrizen als zweidimensionale Arrays repräsentiert: A ist hat die Dimensionen $N \times K$, B hat $K \times M$, woraus die Matrix C mit Dimension $N \times M$ berechnet wird.



```
long[,] Multiply(long[,] matrixA, long[,] matrixB) {  
    var dimN = matrixA.GetLength(0);  
    var dimK = matrixA.GetLength(1);  
    var dimM = matrixB.GetLength(1);  
    var matrixC = new long[dimN, dimM];  
    for (var i = 0; i < dimN; i++) {  
        for (var j = 0; j < dimM; j++) {  
            matrixC[i, j] = 0;  
            for (var k = 0; k < dimK; k++) {  
                matrixC[i, j] += matrixA[i, k] * matrixB[k, j];  
            }  
        }  
    }  
    return matrixC;  
}
```

Messen Sie anschliessend die Performance. Welchen Speedup erreichen Sie?

Hinweise:

- Analysieren, welche Schleifen parallelisiert werden dürfen.
- Meist reicht es, die möglichst äussere Schleife zu parallelisieren.

Aufgabe 2: Parallel Quicksort in .NET

Parallelisieren Sie den sequentiellen Quicksort aus der Vorlage mit dem gleichen Ansatz wie in der letzten Java-Übungsserie. Sie können hierzu zwei Ansätze benutzen:

- a) Explizite(s) Task Start und Wait
- b) `Parallel.Invoke()`

Erreichen Sie den gleich guten Speedup wie in Java?

Aufgabe 3: Thread Pool Deadlock

Analysieren Sie den Deadlock im TPL Programm der Vorlage. Hierzu können Sie den Visual Studio Debugger mit folgenden Ansichten benutzen:

- Threads (Menü Debug->Windows->Threads): Zu jedem Thread können Sie den Call Stack auswählen und inspizieren.
- Tasks in TPL Thread Pool (Menü Debug->Windows->(Parallel) Tasks): Hier sehen Sie den Zustand jedes Tasks, seinen Call Stack und Zuordnung zum Worker Thread (falls aktiv).

Was ist der Grund des Deadlocks? Wie lässt er sich beseitigen?

Aufgabe 4: Asynchrone Downloads

Die Vorlage führt eine Sequenz von Web-Seiten Downloads in .NET durch. Machen Sie Downloads asynchron, d.h. alle Downloads können sie als TPL Tasks nebenläufig lancieren. Am Ende aller Downloads soll die Gesamtzeit ausgegeben werden. Sie können dafür einen Multi-Continuation wie folgt definieren:

```
Task.WhenAll(task1, task2).ContinueWith(successor);
```

