

## Projekt „Wave“, für den 15.06. und 22.06.2018

### Problem:

Gegeben ist eine Reihe von Matrizen  $A_1, \dots, A_n$  deren Produkt  $A_1 \times \dots \times A_n$  berechnet werden soll. Die Größe der Matrizen ist durch die Liste  $l_1, \dots, l_{n+1}$  gegeben, wobei Matrix  $A_i$  eine  $l_i \times l_{i+1}$  Matrix ist. Welche Klammerung des zu berechnenden Produkts führt zu den geringsten Berechnungskosten unter der Verwendung eines Standard Matrix-Multiplikation Algorithmus?

Es gibt exponentiell viele Möglichkeiten den gesuchten Ausdruck zu klammern, was eine vollständige Suche für eine größere Anzahl an Matrizen unmöglich macht. Glücklicherweise kann das Problem mittels dynamischer Programmierung in polynomialer Zeit optimal gelöst werden.

### Algorithmus:

Die minimalen Kosten  $c_{ij}$  zur Berechnung eines Teil-Ausdrucks  $A_i \times \dots \times A_j$  ergeben sich über die Formel

$$c_{ij} = \begin{cases} 0 & i \leq j \\ \min_{i \leq k < j} (c_{ik} + c_{k+1,j} + l_i l_{k+1} l_{j+1}) & \text{otherwise} \end{cases}$$

Daher ist für jeden Teil-Ausdruck  $A_i \times \dots \times A_j$  jenes  $k$  zu finden welches die Kosten für die Berechnung

$$(A_i \times \dots \times A_k) \times (A_{k+1} \times \dots \times A_j)$$

minimiert.

### Ressourcen:

- **week\_12/dynamic\_programming** in github repo  
enthält eine sequentielle Berechnung der Minimalen Kosten

### Aufgaben (1. Woche):

1. Skizzieren sie die Matrix C für N=5 und illustrieren sie die Daten-Abhängigkeiten für die Berechnung der Werte C[2,2], C[1,3] und C[0,4]. Kann ein generelles Muster abgeleitet werden?
2. Parallelisieren Sie die gegebene sequenzielle Implementierung mittels OpenMP. Speichern Sie diese Variante als „dynamic\_programming\_omp.c“ ab.
3. Implementieren Sie eine getilte / geblockte Variante mittels OpenMP. Stellen Sie sicher das die Block-Größe über einen (compile-time) Parameter veränderlich bleibt. Speichern Sie diese Variante als „dynamic\_programming\_blocked\_omp.c“ ab.

### *Abgabe:*

1. Per email an [herbert.jordan@uibk.ac.at](mailto:herbert.jordan@uibk.ac.at), 1 Abgabe pro **Gruppe**  
Betreff: “[PS703106] [UE11] GR\_XX - NAME1 NAME2 NAME3”  
**PDF Zusammenfassung, keine Binaries** (make clean)!  
Deadline: 15.06.2018, **11:00 (!)**;

### *Aufgaben (2. Woche):*

1. Implementieren Sie eine OpenCL Version welche den Wave-Front Parallelismus der Berechnung ausnützt.
2. Evaluieren Sie ob durch die Verwendung einer Out-of-Order Queue auf Ihrem System eine höhere Performance erreicht werden kann.

### *Vorgehensweise:*

1. Basieren Sie ihre Lösung auf der geblockten OpenMP variante (Musterlösung wird nach der ersten Woche bereitgestellt).
2. Lassen Sie jeden Block von einem Kernel Aufruf bestehend aus einer einzelnen Work-Group abarbeiten.
3. Synchronisieren Sie mittels Event-basierten Abhängigkeiten

### *Abgabe:*

2. Per email an [herbert.jordan@uibk.ac.at](mailto:herbert.jordan@uibk.ac.at), 1 Abgabe pro **Gruppe**  
Betreff: “[PS703106] [UE12] GR\_XX - NAME1 NAME2 NAME3”  
**PDF Zusammenfassung, keine Binaries** (make clean)!  
Deadline: 22.06.2018, **11:00 (!)**;