

Serie 6

Aufgabe 6.1 (OpenMP Locks)

In der Datei `locks.c` sollen Daten aus Dateien eingelesen werden und parallel in einen binären Suchbaum eingefügt werden.

Sorgen Sie unter Verwendung von Locks dafür, dass das Einfügen in verschiedenen Teilbäumen tatsächlich parallel und korrekt ablaufen kann.

Zum Testen stehen Dateien mit zufälligen Zahlen bereit, die dem Programm als Parameter übergeben werden können.

Aufgabe 6.2 (OpenMP Tasks + Abhängigkeiten)

Wir haben in der Vorlesung gesehen, wie man eine Matrix $A \in \mathbb{R}^{n \times n}$ in das Produkt einer unteren Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ und einer oberen Dreiecksmatrix $R \in \mathbb{R}^{n \times n}$ zerlegen kann. Dies kann man nun dazu benutzen, um lineare Gleichungssysteme der Form

$$AX = B$$

zu lösen, wobei $X, B \in \mathbb{R}^{n \times n}$ ebenfalls Matrizen sind.

Durch Ausnutzung der Dreieckszerlegung erhält man die beiden äquivalenten Gleichungssysteme

$$LY = B \quad \wedge \quad RX = Y, \quad \text{mit } Y \in \mathbb{R}^{n \times n}.$$

Zunächst löst man das System mit der unteren Dreiecksmatrix L per *Vorwärtseinsetzen*. Anschließend wird das zweite System per *Rückwärtseinsetzen* gelöst.

Beim Vorwärtseinsetzen wird die Lösung Y spaltenweise „von oben nach unten“ berechnet, es gilt

$$y_{ij} = \frac{1}{\ell_{ii}} \left(b_{ij} - \sum_{k=1}^{i-1} \ell_{ik} \cdot y_{kj} \right), \quad \text{für alle } i, j \in [1 : n].$$

In einer geblockten Variante des Algorithmus wird der Ausdruck $1/\ell_{ii}$ zu einem Aufruf von `lsolve` mit dem Matrixblock `L[i][i]`. Beim Rückwärtseinsetzen wird in umgekehrter Weise die Lösung X spaltenweise „von unten nach oben“ berechnet, es gilt

$$x_{ij} = \frac{1}{r_{ii}} \left(y_{ij} - \sum_{k=i+1}^n r_{ik} \cdot x_{kj} \right), \quad \text{für alle } i, j \in [1 : n].$$

Der Ausdruck $1/r_{ii}$ wird in der geblockten Variante wieder durch einen Aufruf von `rsolve` mit den Block `R[i][i]` implementiert.

In dem Programm `matrixsolves.c` wurden bereits sowohl eine LR-Zerlegung der Matrix A berechnet, als auch das Lösen der beiden Gleichungssysteme in Form der Funktionen `lsolve` bzw. `rsolve` implementiert.

- (a) Vervollständigen Sie in der Datei `arithmetics.c` die Funktionen `block_ksolve` bzw. `block_rsolve`, die geblockte Varianten der zuvor genannten Funktionen darstellen sollen.
- (b) Implementieren Sie die Funktion `block_parallel_addmul`, die per OpenMP Tasks eine parallelisierte Version des Matrix-Matrix-Produkts realisieren soll.
- (c) Ferner implementieren Sie die Funktionen `block_parallel_ksolve` bzw. `block_parallel_rsolve`, die eine Parallelisierung mit OpenMP Tasks für das Lösen von Dreieckssystem darstellen soll.