

# Programmier-Paradigmen

Tutorium – Gruppe 2 & 8

Henning Dieterichs

Parallelprogrammierung / Java

# Aufgabe 1: Amdahlsches Gesetz

- Ein Programm mit mehreren Threads arbeitet auf einem Puffer.
- Ein Teil der Threads liest Daten aus dem Puffer (Leser), der andere Teil schreibt Daten in den Puffer hinein (Schreiber).
- Eine beliebige Anzahl lesender Threads kann gleichzeitig auf dem Puffer arbeiten.
- Wenn ein Schreiber auf den Puffer zugreift, kann kein anderer Thread (Leser oder Schreiber) auf dem Puffer aktiv sein.
- Gegeben ist ein Thread Pool, wobei jeder Leser und Schreiber durch einen Thread repräsentiert wird. Dabei sind 90% der Threads Leser und 10% Schreiber.
  - Jeder Leser benötigt 2 Sekunden jeder Schreiber 3 Sekunden zur Ausführung.
  - Das Programm ist beendet, wenn alle Threads im Pool beendet sind.
- **Unter Berücksichtigung des Amdahlschen Gesetzes, wo liegt die obere Grenze der Beschleunigung (Speedup) der oben beschriebenen Implementierung auf einem 4-Kern-Prozessor?**

# Aufgabe 2: Petri-Netze

- Implementiere *putToken* und *takeAnyToken*
- Was wäre, wenn *putToken*, *takeAnyToken* und *moveTokenTo* über *this* synchronisiert wären?
- Coffman-Bedingungen:
  - *Mutual exclusion condition* (wenn ich, dann kein anderer)
  - *No preemption condition* (wenn ich, dann ziehe ich's auch durch)
  - *Wait for condition* (ich ziehe es durch, warte aber noch auf jemand anderes)
  - *Circular wait condition* (jemand wartet auf mich, ich warte aber auch auf ihn)

# Aufgabe 3: Urlaubsreservierungssystem

- Implementiere tryTakeVacation threadsafe!
  - Mit globalem synchronized
  - Mit lokalem synchronized
  - Und zwar richtig

# Aufgabe 4: Barriere

- Implementiere eine zyklische Barriere!

# Aufgabe 5: Paralleler Primzahlentest

- Parallelisiere den sequentiellen Code mithilfe von Futures!

# Aufgabe 5: Paralleler Primzahlentest

Wie groß ist der Speedup (im Vergleich zur sequentiellen Version) für 1, 2, 4 und 8 Threads?  
Wie viele Kerne hat Ihr Testrechner?

**Beispiellösung:** Für 4 physische / 8 logische Kerne

$$T_{seq} = 221730ms$$

$$T_1 = 225987ms$$

$$T_2 = 118853ms$$

$$T_4 = 62152ms$$

$$T_8 = 50045ms$$

$$S_{seq}(1) = \frac{T_{seq}}{T_1} = 0,98$$

$$S_{seq}(2) = \frac{T_{seq}}{T_2} = 1,86$$

$$S_{seq}(4) = \frac{T_{seq}}{T_4} = 3,57$$

$$S_{seq}(8) = \frac{T_{seq}}{T_8} = 4,43$$



# Deadlock / Race-Condition Spiel

- <https://deadlockempire.github.io/>