1 Thread-Beispiele

1.1 Erzeugen von Threads in Java

Schreiben Sie ein Programm in Java, welches acht Threads erstellt und startet. Jeder Thread soll dabei „Hello World from Thread i“ mit i als Index des Threads im Array ausgeben. Beobachten und interpretieren Sie das Ergebnis.

Lösung: `https://gist.github.com/lukaswerner/c1a53b8b03fe4d93e1e9dfcfc50c9cb5`

1.2 Erzeugen von Threads in Java (Ohne Synchronisierung von Strings)

Erweitern Sie das Programm aus 1.1 so, dass die Ausgabe des Strings „Hello World from Thread i“ nicht synchronisiert erfolgt. Schreiben Sie dafür eine Methode, welche einen String Zeichen für Zeichen ausgibt, rufen Sie diese Methode anstelle von System.out.println() auf.

Lösung: `https://gist.github.com/lukaswerner/61a25246aca3c9fc3fa7c57d86f42af5`

1.3 Race Condition Beispiel

Schreiben Sie ein Programm in Java, welches ≥ 10000 Threads startet. Jeder Thread soll dabei eine statische Variable (z.B. z) hochzählen mit dem $++$-Operator. Geben Sie am Ende diese statische Variable aus und beobachten, sowie interpretieren das Ergebnis.

Lösung: `https://gist.github.com/lukaswerner/32e344a4c1febc69724d08a60c56064c`

1.4 Race Condition Lösung

Erweitern Sie das Programm aus 1.3 so, dass die Race Condition nicht stattfindet. Implementieren Sie hierfür eine Methode, die gesperrt wird, sobald ein Thread sie ausführt (Stichwort: *synchronized*). Diese Methode soll dann $z{+}{+}$ ausführen.

Lösung: `https://gist.github.com/lukaswerner/12825318f45e55ec2f3f502a0a49584a`

2 Amdahls Gesetz

a) Finden Sie heraus, welcher Anteil der Zähler-Aufgabe mit $k = 10000$ Threads parallelisierbar ist.

Lösung

$a :=$ Zeit bei sequentiellem Ablauf
$b :=$ Zeit bei parallelem Ablauf

$$
\begin{aligned}
& b = a\left(1 - p + \frac{p}{n}\right) \\
\Leftrightarrow \quad & \frac{b}{a} = 1 - p + \frac{p}{n} \\
\Leftrightarrow \quad & \frac{b}{a} - 1 = \frac{p}{n} - p = p\left(\frac{1}{n} - 1\right) \\
\Leftrightarrow \quad & \frac{\frac{b}{a} - 1}{\frac{1}{n} - 1} = p \\
\Leftrightarrow \quad & p = \frac{n\left(\frac{b}{a} - 1\right)}{n\left(\frac{1}{n} - 1\right)} = \frac{n\left(\frac{b}{a} - 1\right)}{1 - n} = \frac{na\left(\frac{b}{a} - 1\right)}{a(1-n)} = \frac{n(b-a)}{a(1-n)}
\end{aligned}
$$

Testwerte:
$a = 0,0000588089s$
$b = 0,6387376938s$

$$
p = \frac{n(b-a)}{a(1-n)} = \frac{10000\,(0,6387376938s - 0,0000588089s)}{0,0000588089s\,(1 - 10000)} = -10861,3281... \approx -1086132,8\%
$$

Mit Parallelisierung erheblich langsamer!

b) Welchen Anteil erwarten Sie für $k = 20000$ Threads?
Zurück gestellt...

c) Wie weit kann man die Bearbeitung durch Parallelisierung beschleunigen, wenn man beliebig viele Prozessoren zur Verfügung hat?

$$
\lim_{n\to\infty} \frac{1}{1 - p + \frac{p}{n}} = \frac{1}{1-p} = \frac{1}{1 - \frac{n(b-a)}{a(1-n)}} = ?
$$

3 Verschränkung (1)

Thread p habe m Schritte, Thread q habe n Schritte auszuführen.

a) Geben Sie eine rekursive Definition für die Anzahl $\operatorname{anz}(m, n)$ der möglichen verschränkten Abläufe von p und q an.

Lösung

$$\text{anz}(m,n) = \begin{cases} 1 & m = 0 \lor n = 0 \\ \text{anz}(m-1,n) + anz(m,n-1) & \text{sonst} \end{cases}$$

b) Finden Sie einen geschlossenen Ausdruck für $\text{anz}(m,n)$.

Lösung

Wegbeschreibung ist ein Bitvektor mit m Nullen und n Einsen, wobei hier gilt:

$$\begin{gathered} 0 \mathrel{\widehat{=}} \text{Schritt nach rechts} \\ 1 \mathrel{\widehat{=}} \text{Schritt nach unten} \end{gathered}$$

Die Länge des Bitvektors ist $n+m$. Isomorph zum Bitvektor mit m Nullen und n Einsen sind n Teilmengen einer $n+m$ Menge.

Beispiel:
Bitvektor: 011010, $(m+n)$-Menge sei $\{1, ..., m+n\}$, dargestellte Teilmente ist $\{2,3,5\}$

Satz: Die Anzahl der k Teilmengen einer n-Menge ist $\binom{n}{k}$.

Behauptung: $\text{anz}(m,n) = \binom{m+n}{n}$

Beweis.

IA Sei $n = 0 \Rightarrow \text{anz}(m,0) = 1 = \begin{pmatrix} m \\ 0 \end{pmatrix}$

IV Für ein festes $k = m+n$ gilt: $\text{anz}(m,n) = \begin{pmatrix} m+n \\ n \end{pmatrix}$.

IS Annahme: $m+n = k+1$

$$\text{anz}(m,n) = \text{anz}(m-1,n) + \text{anz}(m,n-1) \overset{\text{IV}}{=} \begin{pmatrix} m-1+n \\ n \end{pmatrix} + \begin{pmatrix} m+n-1 \\ n-1 \end{pmatrix}$$

$$= \begin{pmatrix} k \\ n \end{pmatrix} + \begin{pmatrix} k \\ n-1 \end{pmatrix} \overset{\text{def.}}{=} \begin{pmatrix} k+1 \\ n \end{pmatrix}$$

$\square$

c) Schätzen Sie die Größenordnung von $\text{anz}(m,n)$.
Beim nächsten mal!