

SPP Übung 3

Quang Duy Nguyen, Egemen Ulutürk, Leonard Bongard

Aufgabe 1

- a) 5 Wäscheladungen sequentiell ausführen

also brauchen wir pro Wäsche die zu waschen ist 3 Stunden. Wenn wir also 5 Wäschen haben, brauchen wir $3 \cdot 5 = 15$ Stunden.

- b) Wir verteilen die Ladungen an die Wäscher. Jeder Wäscher holt sich eine Ladung der Wäsche und wäscht diese. Wenn er fertig ist, dann schaut er ob es noch weitere Wäschen gibt und wenn ja nimmt er sich eine weitere Wäsche und wäscht diese auch. Wenn es keine Wäsche mehr gibt ist er fertig. [□]

Das Schema der Arbeitsteilung ist „data decomposition“.

Dauer der Wäsche nach Anzahl der Freunde:

1 weiterer Freund: Muss jeder min 2mal und einer auch 3mal. Also ist die min Laufzeit: $3 \cdot 3 = 9$

ab 2-3 Freunden: Muss jeder min 1mal und manche auch 2mal. Also ist die min Laufzeit: $2 \cdot 3 = 6$

ab 4 Freunden: Muss solange noch wäschen übrig sind 1 mal waschen. Also ist die min Laufzeit: $1 \cdot 3 = 3$

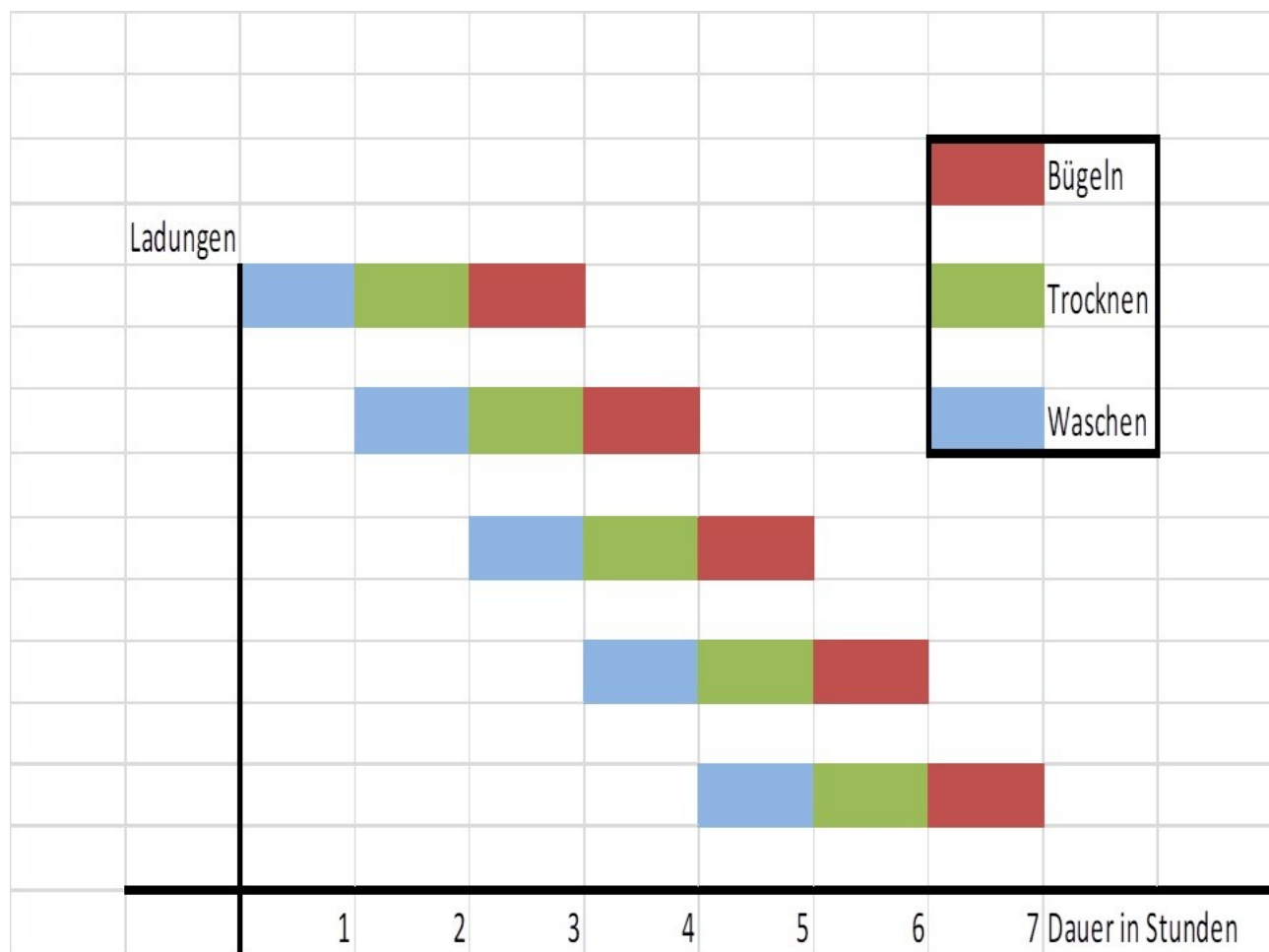
$$\lceil ((Ladungen / Personen) \cdot 3) \rceil = \text{Dauer}$$

Mehr Freunde als Anzahl Freunde ändert nichts an der Dauer die wir zum waschen brauchen.

- c) parallisiert alleine arbeiten.

Wir fangen an eine wäsche zu waschen. Nachdem die erste Wäsche gewaschen ist, kommt sie in den Trockner und gleichzeitig können wir mit der nächsten Wäsche anfangen. Nach dem Trockner fertig ist, können wir anfangen zu bügeln, die nächste Wäsche in den Trockner werfen und noch eine Wäsche anfangen, so kann nacheinander parallisiert werden.

Das Schema der Arbeitsteilung: „pipelining pattern“



Die Arbeit dauert min $\underline{3+1+1+1+1 = 7 \text{ Stunden}}$. Für jede Wäsche kommt eine Stunde hinzu.

Aufgabe 2

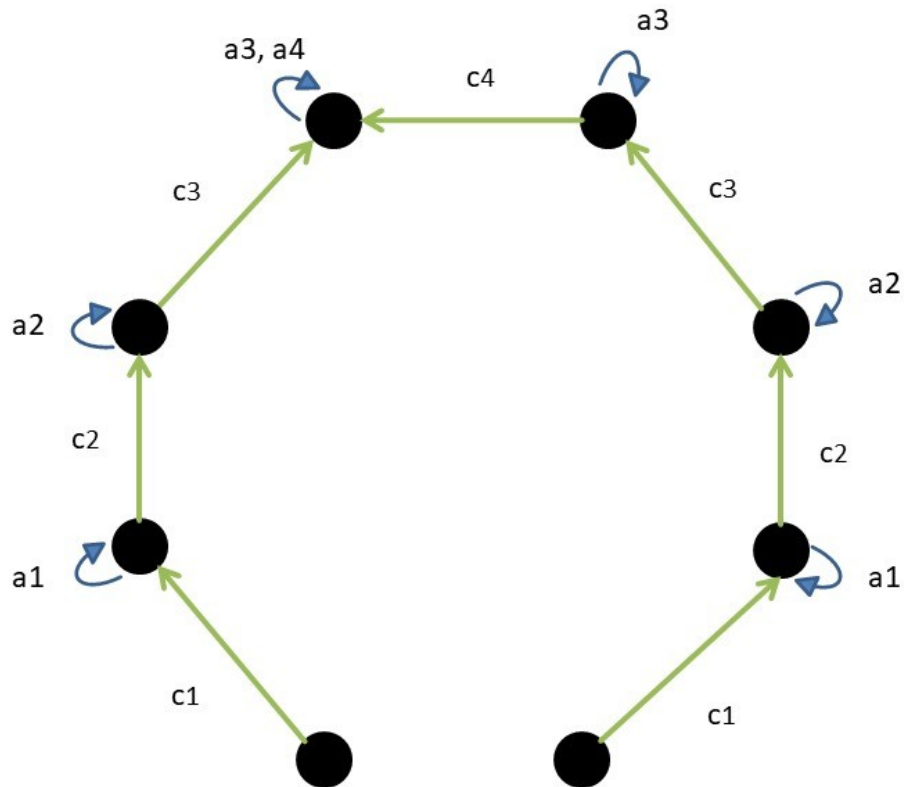
- a) Alle stehen im Kreis: $t_c(\text{Komm mit nachbarn}) + t_a + t_c(\text{jeder zweite Komm mit anderen Nachbarn}) + t_a + t_c(\text{gib Ergebnis an Nachbarn weiter}) + t_c(\text{gib Ergebnis an Nachbarn weiter}) + t_a$ (Rechne beide Seiten des Kreises zusammen).

$$\text{Also Zeit} = 4 \cdot t_c + 4 \cdot t_a$$

- b) Alle stehen in einem 4 x 2 Gitter: $t_c(\text{Komm mit Nachbarn in der Zeile des Gitters(wo 4 Leute stehen)}) + t_a + t_c(\text{Komm mit Nachbar in der 4er Zeile}) + t_a + t_c(\text{Komm mit den Leuten aus der anderen 4er zeile}) + t_a$

$$\text{Iso Zeit} = 3 \cdot t_c + 3 \cdot t_a$$

Abbildungen: Aufgabe 2a) Kreis



Ausführungsreihenfolge :

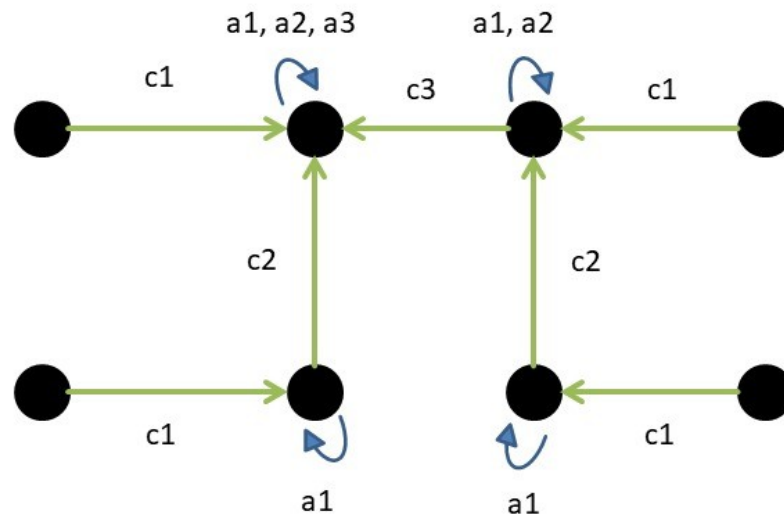
c1 -> a1 -> c2 -> a2 -> c3 -> a3 -> c4 -> a4

Legende: Blaue Pfeile: Summiert die Zahl, die in der letzten Iteration übertragen wurde mit der welcher im Knotenpunkt gespeichert ist.

Grüne Pfeile: Überträgt die Zahl aus dem Startknoten des Pfeiles zu dem Knoten auf welcher der Pfeil zeigt.

Also Zeit = $4 \cdot t_c + 4 \cdot t_a$

4x2 Matrix



Ausführungsreihenfolge :

$c1 \rightarrow a1 \rightarrow c2 \rightarrow a2 \rightarrow c3 \rightarrow a3$

Legende: Blaue Pfeile: Summiert die Zahl, die in der letzten Iteration übertragen wurde mit der welcher im Knotenpunkt gespeichert ist.

Grüne Pfeile: Überträgt die Zahl aus dem Startknoten des Pfeiles zu dem Knoten auf welcher der Pfeil zeigt.

Also Zeit = $3 \cdot t_c + 3 \cdot t_a$

Aufgabe 3

- a) $1 / (0,4 + (0,6 / 1,5)) = 1,25$, also eine 1,25 fache Verbesserung über die gesamte Programmlaufzeit. Also nur noch 80% der Laufzeit.

$1/(0,85+(0,15/8)) = 1,151$, also eine 1,151 fache Verbesserung über die gesamte Programmlaufzeit. Also nur noch 86,9% der Laufzeit.

Also ist es effizienter die gesamte Gleitkommaoperation um das 1,5 fache zu beschleunigen.

- b) $1 / (0,1 + (0,9 / 16)) = 6,4$, also maximal 6,4 facher speedup mit 16 Prozessoren, wenn 90% der Ausführungszeit perfekt parallelisiert werden.
- c) Die Formel: $1 / ((1 - \text{parallelisierbarer_Anteil}) + (\text{parallelisierbarer_Anteil} / \text{Anzahl_Prozessoren})) = \text{speed_up_faktor}$

$$\begin{array}{lcl} 1 / ((1 - p) + (p / 16)) = 10 & | & * ((1 - p) + (p / 16)) \\ 1 = 10 - 10p + (10p / 16) & | & -10 \\ -9 = -10p + 10p / 16 & | & *16 \\ -144 = -150p & | & :-150 \\ p = 0,96 \end{array}$$

das heißt, bei 16 Prozessoren muss 96% der Anwendung parallelisierbar sein, damit man einen Speedup von 10 erreichen kann.