

Aufgabe 17**a)**

Die reine Kommunikationszeit in Abhängigkeit der Problemgröße N errechnet sich nach folgendem Schema...

... für einen Schritt (t_k ist die Übertragungszeit):

$$t_i(n) = \frac{n}{2^i} * t_k$$

... für die Gesamtheit aller Schritte:

$$t(N, P) = N + \sum_{i=0}^{\frac{P}{2}-1} \frac{n}{2^i} * t_k$$

b)

Die Rechenzeit in Abhängigkeit der Problemgröße N errechnet sich nach folgendem Schema...

... für einen Schritt (t_r ist Zeit für eine Multiplikation):

$$t(N) = (N \log N) * t_r$$

... für die Gesamtheit aller Rechenschritte:

$$t_{ges}(N, P) = N + \sum_{i=0}^{\frac{P}{2}-1} \frac{N}{2^i} t_k + \left(\frac{N \log N}{P} \right) t_r$$

c)

Für den SpeedUp ergibt sich:

$$S(N, P) = \frac{N \log N}{N + \sum_{i=0}^{\frac{P}{2}-1} \frac{N}{2^i} t_k + \left(\frac{N \log N}{P} \right) t_r}$$

d) Existiert eine Isoeffizienzfunktion muss gelten:

$$E(W, P) = \frac{1}{1 + \frac{T_o(W, P)}{W}}$$

und

$$\lim_{W \rightarrow \infty} E(W, P) = 1$$

und damit muss gelten, dass bei festem P der Overhead weniger als linear anwächst. D.h., es muss gelten

$$T_o(W, P)|_P = const < O(W)$$

Das ist beim parallelen Mergesort der Fall, da für jeden zusätzlichen Knoten (bzw. jedes Paar, es kommen ja immer 2 Knoten pro Schritt hinzu) der Overhead abnimmt.

Die gewünschten Testergebnisse traten leider nicht ein, da auf den Poolrechnern das Programm immer mit einem Fehler abbricht (MPI_RECV : Message truncated). Nachforschungen haben ergeben, dass es wohl an einem Bug in der MPI-Implementierung liegt, da es auf dem lokalen Rechner läuft (jedoch trat auch hier nicht das gewünschte Laufzeitverhalten ein).