# Aufgabe 07

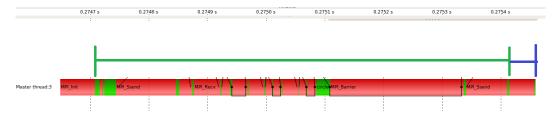
# Can Nayci, Leonhard Rattmann, Emil Sharaf 09.12.2023

## 1 Circle

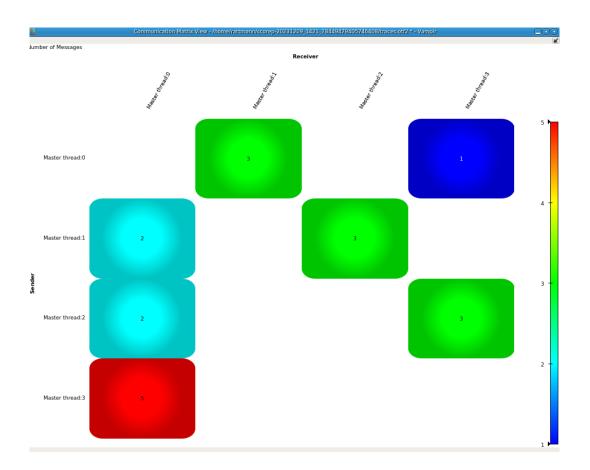
$$\begin{split} N: \text{Anzahl der Elemente des Arrays} \\ nprocs: \text{Anzahl der Prozesse} \\ start_p: \text{erstes Element für Prozess } p \\ end_p: \text{letztes Element für Prozess } p \\ start_0 = 1 \\ end_p = p \cdot (int) \frac{N}{nprocs} + a \\ a = \begin{cases} \text{wenn } ((N \bmod nprocs) < p) : 0 \\ \text{sonst } : 1 \end{cases} \\ start_p = end_{p-1} \end{split}$$

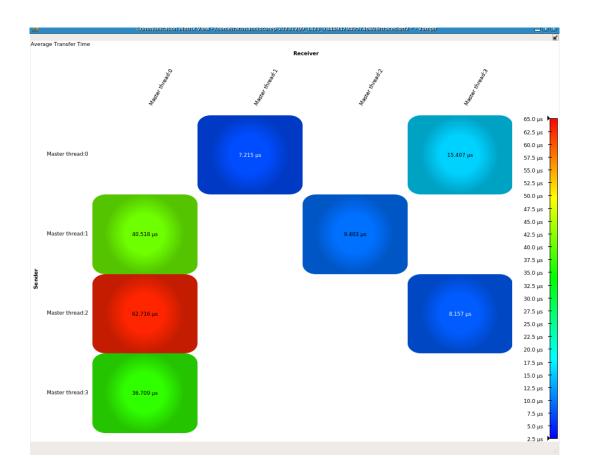
# 2 Visualisierung

1 Grafische Darstellung der Kommunikation:



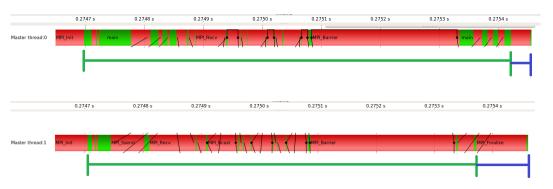
2 Communication Matrix View:

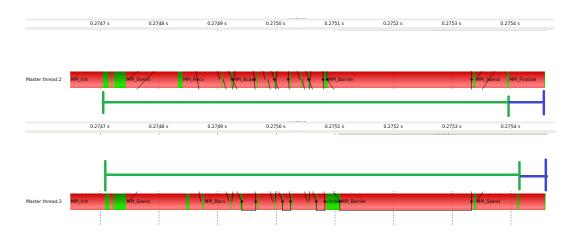




#### 3 Programmphasen:

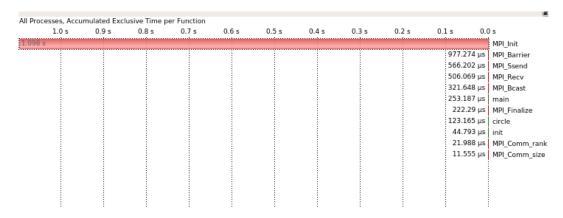
Die Phase bis 0,2747s stellt die Initialisierung dar. Die Iterationen werden in den nachfolgenden Screenshots als "grüne Phase" und das Beenden als "blaue Phase" dargestellt:





#### 4 MPI-Init-Phase:

Die MPI-Init-Phase dauerte  $\sim 1,098$ s und nimmt somit ungefähr 99,7% der Programmlaufzeit in Anspruch:

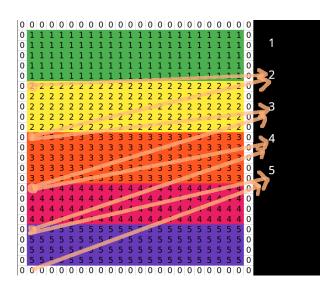


## 3 Prallelisierung mit MPI

```
int n = Anzahl der Reihen:
int nprocs = Prozessanzahl;
n = (getInterlines() * 8) + 9 - 1;
int startIndex , endIndex;
nur Prozess 0:
   int s_startIndex , s_endIndex;
   int lpp, lpp_rest;
   lpp = (int) (n - 1) / nprocs;
   lpp_rest = (n - 1) \% nprocs;
   startIndex = 1 (0er-Reihe wird ignoriert);
   endIndex = lpp + (lpp_rest < 0)? 1 : 0);
   lpp_rest ---;
   s_startIndex = endIndex;
   s_{endIndex} = 0;
    fuer alle nprocs exkl. 0:
       MPI_Ssend(*s_startIndex an Prozess i);
       s_{endIndex} = startIndex + lpp;
       if (lpp_rest < 0)
           endIndex++;
           lpp_rest ---;
       MPI_Ssend(*s_endIndex an Prozess i);
       s_startIndex = s_endIndex;
       // \Rightarrow startIndex \ ist \ inklusive, \ endIndex \ exklusive.
   }
}
alle anderen Prozesse:
   MPI_Recv(*startIndex);
   MPI_Recv(*endIndex);
}
```

z.B. 64 Interlines, 4 Prozesse:

$$n = 64*8+9-1 = 520$$
 
$$lpt = \frac{519}{4} = 129$$
 
$$lpt\_rest = 3$$
 für Prozess 0: 
$$startIndex = 1$$
 
$$endIndex = 130+1$$
 für Prozess 3: 
$$startIndex = 262$$
 
$$endIndex = 391+1$$
 für Prozess m: 
$$startIndex = 1 + m \cdot lpt + lpt\_rest$$
 nicht  $(m-1)$ , weil Prozesse ab 0 nummeriert sind 
$$endIndex = 521$$



### Kommunikation

1. Alle Prozesse außer 0 (bzw. 1) brauchen ihre Start- und endIndizes. 2. Jeder Prozess braucht in jeder Iteration den oben und unten benachbarten Teil der Matrix.