

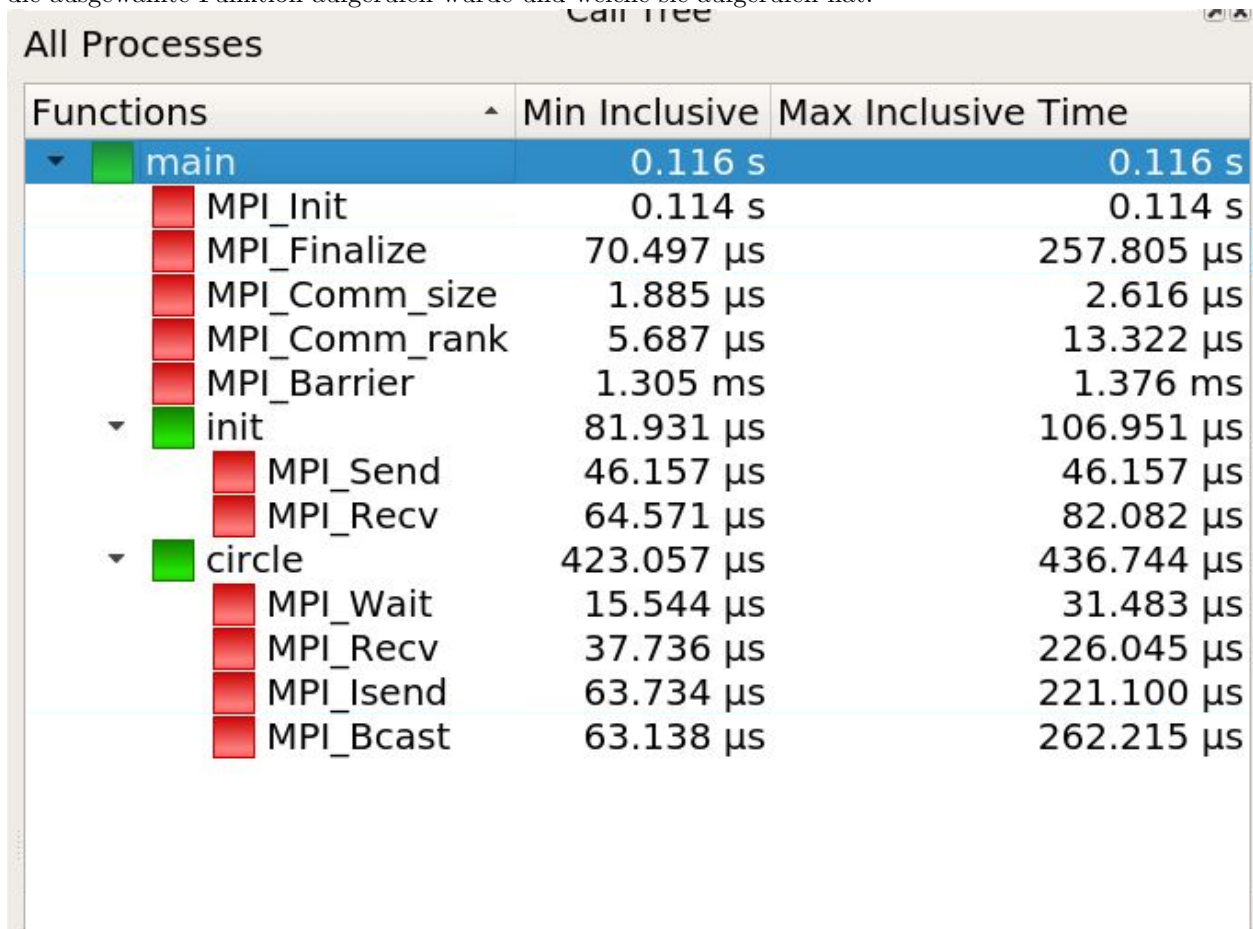
Visualisierung mit Vampir

Yannik Könneker, Maik Simke, Jonas Bögle und Flo Dreyer

November 30, 2019

Im unteren Bild kann man erkennen, welche MPI-Befehle in welcher Funktion wieviele nano bis normale Sekunden verbrauchen. Dabei ist immer eine min und eine max inclusive Time angegeben.

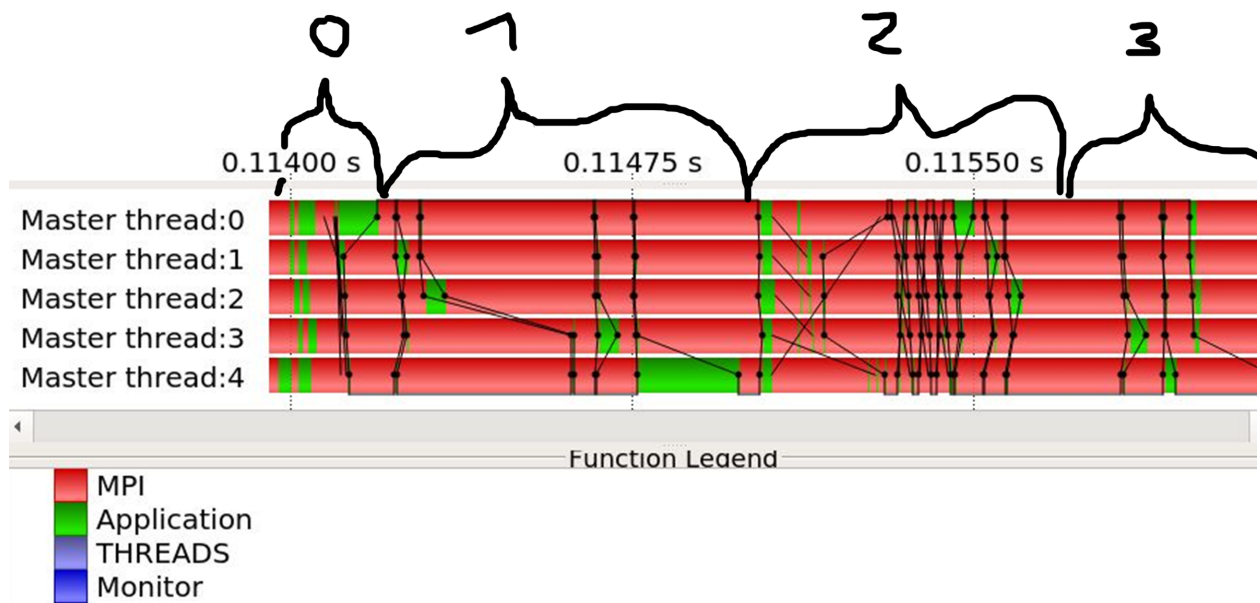
Bei anklicken einer Funktion kann man sich (nicht im Screenshot dabei) anschauen, von welchen Methoden die ausgewählte Funktion aufgerufen wurde und welche sie aufgerufen hat.



Functions	Min Inclusive	Max Inclusive Time
main	0.116 s	0.116 s
MPI_Init	0.114 s	0.114 s
MPI_Finalize	70.497 µs	257.805 µs
MPI_Comm_size	1.885 µs	2.616 µs
MPI_Comm_rank	5.687 µs	13.322 µs
MPI_Barrier	1.305 ms	1.376 ms
init	81.931 µs	106.951 µs
MPI_Send	46.157 µs	46.157 µs
MPI_Recv	64.571 µs	82.082 µs
circle	423.057 µs	436.744 µs
MPI_Wait	15.544 µs	31.483 µs
MPI_Recv	37.736 µs	226.045 µs
MPI_Isend	63.734 µs	221.100 µs
MPI_Bcast	63.138 µs	262.215 µs

Im Bild unten sieht man, durch die verschiedenen Linien dargestellt, wie die einzelnen Prozesse miteinander kommunizieren, dabei repräsentieren die Zeilen die einzelnen Prozesse und die horizontale Achse spiegelt die Zeit wieder. Die Punkte markieren einzelne MPI-Funktionsaufrufe. Von den Punkten gehen Linien aus (meist zu anderen Prozessen) welche den zeitlichen weg des Aufrufes darstellen soll, diese Linien enden in weiteren Punkten, die ebenfalls MPI-Funktionsaufrufe sind (Receive, Bcast, Barrier etc.). Dabei kann man nicht von der Schräge der Linie oder ihrem Abstand auf die Art der Funktion rückschließen.

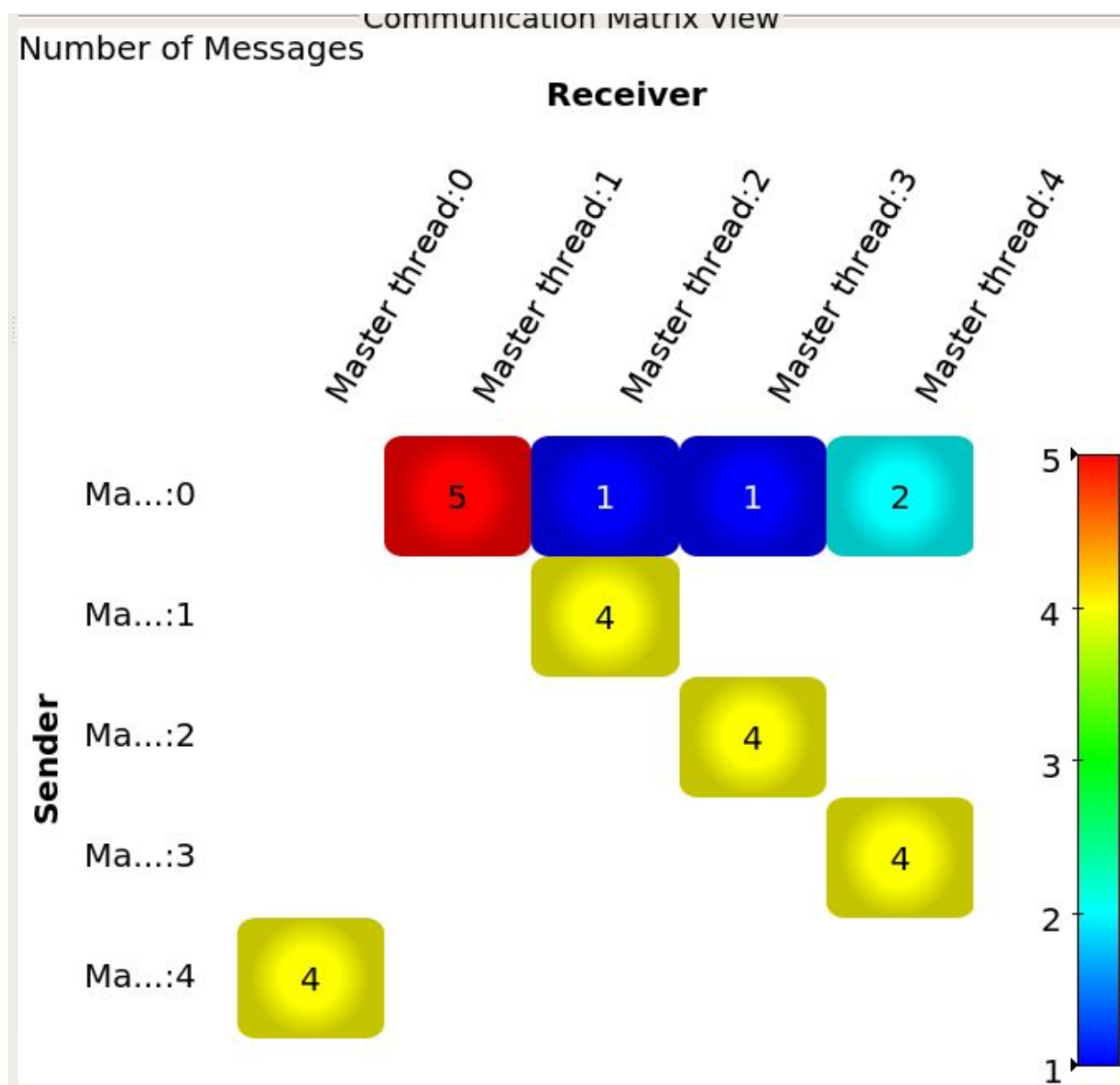
Unser Programm fungiert in 4 erkennbaren Abschnitten, die von 0 bis 3 bezeichnet wurden. Abschnitt 0 beinhaltet den initialisierungscode, bei dem die Arrays erstellt und verteilt werden und der letzte Prozess die erste Zahl des ersten Prozesses bekommt. Abschnitt 1 und 3 sind die Print-Befehle in der Main-Methode. Diese haben viele Barrieren, da wir die Arrays der Reihe nach ausgeben wollten (Hätte man sicherlich auch mit Receives etc. lösen können). Abschnitt 2 ist der eigentlich interessante Abschnitt, in dem die Circlefunktion die Arrays immer weiter durch die Prozesse schiebt bis das Abbruchkriterium erreicht ist.



Klickt man nun auf eine der Punkte oder Linien erhält man genauere Informationen zu der Art der Kommunikation, welcher Prozess die Quelle war und welcher das Ziel, Informationen über den zeitlichen Ablauf, den Kommunikator, Größe und Datenrate von der Übertragung.

pty x		Master Timeline x	
Property		Value	
Display		Master Timeline	
Type		Message	
Message Type		Point to point	
Origin		Master thread:2	
Destination		Master thread:3	
Communicator		MPI_COMM_WORLD	
Tag		0	
Start Time		0.115066 s	
Arrival Time		0.115144 s	
Duration		78.105071 μ s	
Size		20 B	
Data Rate		250.063788 KiB/s	

Im letzten Bild ist visualisiert, welche Prozesse welchen anderen Prozessen wie viele Nachrichten senden. An sich ist es selbstverständlich.



Es gibt natürlich noch weitere Möglichkeiten, die Daten zu visualisieren und darzustellen, allerdings haben wir diese in unserem Fall als unnütz deklassifiziert.