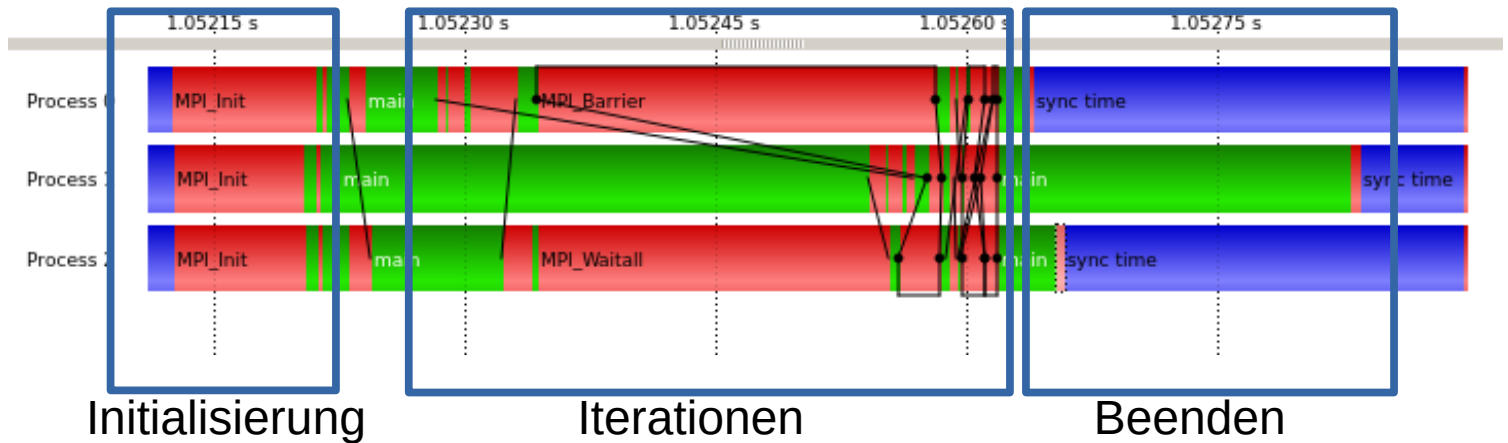
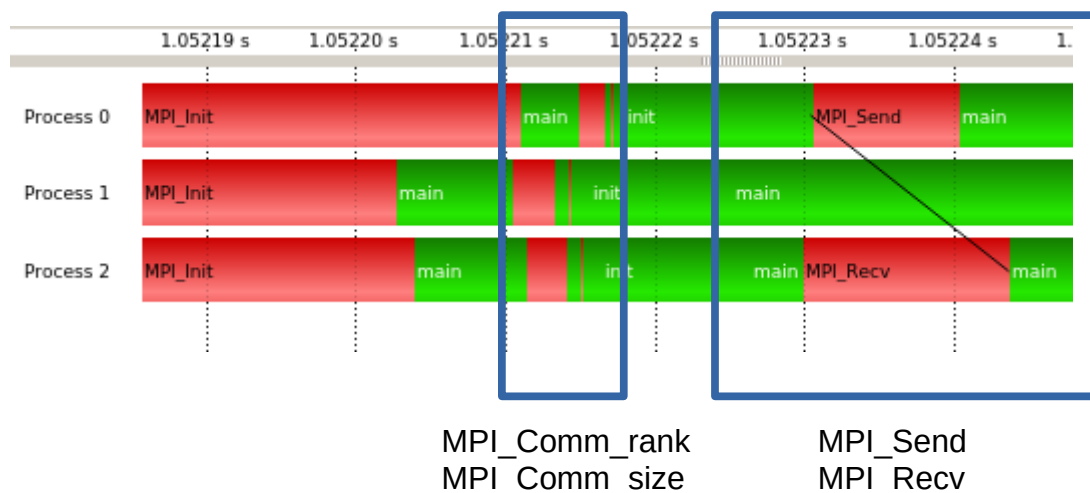


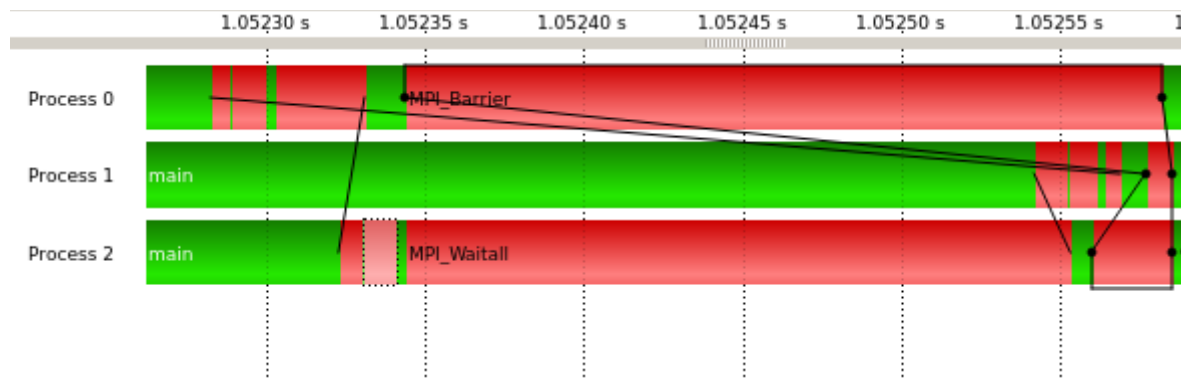
Visualisierung Circle



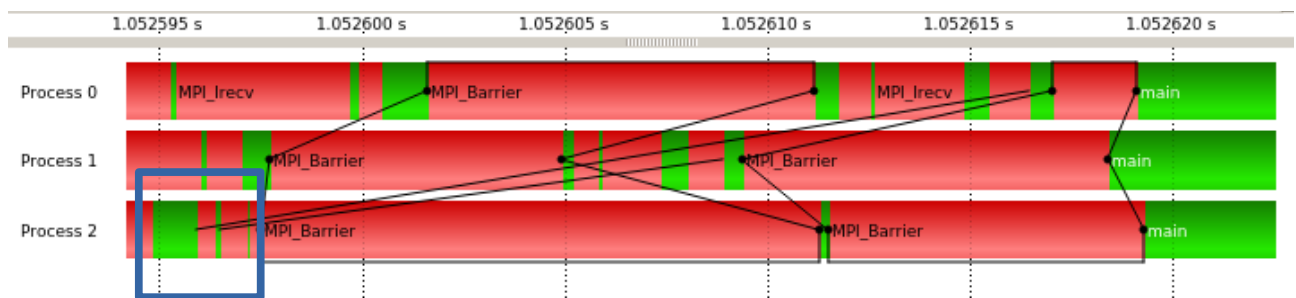
Der oberen Screenshot zeigt die drei haupt Programmphasen: Initialisierung, Iterationen und Beenden. Am anfang ist relativ langsam MPI_Init, wo das Program viel Zeit verbraucht hat. Dann kommen MPI_Comm_rank und MPI_Comm_size zwischen main und Funktion init. Anschließend wie der folgen Screenshot dargestellt ist ein Nachricht nach init von Prozess 0 zu Prozess 2 gesendet worden.(der Wert des ersten Arrayelementes von Prozess 0).



Danach senden die drei Prozesse die Nachrichten zu ihrer Nachfolgern und empfangen die Nachrichten von der Vorgänger im Zyklus. Nach jeder circle befindet MPI_Barrier sich.



Zum Schluss wenn Prozess 2 das Abbruchkriterium erfüllt, sagt er den Bescheid zu anderer Prozesse.



Zusammenfassend ist Kommunikation zwischen den Prozessen wie erwartet. Bei der Iterationen, alle Prozesse können zuerst die eigene Nachrichten senden und dann auf neue Nachrichten warten, die Folge scheint zufällig. Und MPI_Barrier funktioniert so, wenn irgendein Prozess die Aufgabe geschafft, er muss auf anderen Prozessen warten, bis alle Prozesse die gleiche Aufgabe geschafft haben, dann dürfen sie weiter laufen. Allerdings verschwendet MPI_Barrier die Zeit für schnelle geschaffte Prozesse, weil bei jeder Iteration sind die abgelaufene Zeit eines Prozesses nicht gleich.

Ein Prozess kann entweder der schnellsten oder der langsamsten sein. Das heißt wenn wir vielmal Iterationen machen, die gesamte Zeit ist die Summe aus dem langsamsten Prozess von jeder Iteration.