Knoten 2 Prozesse 3:

Gauss-Seidel:

Figure 1: Start

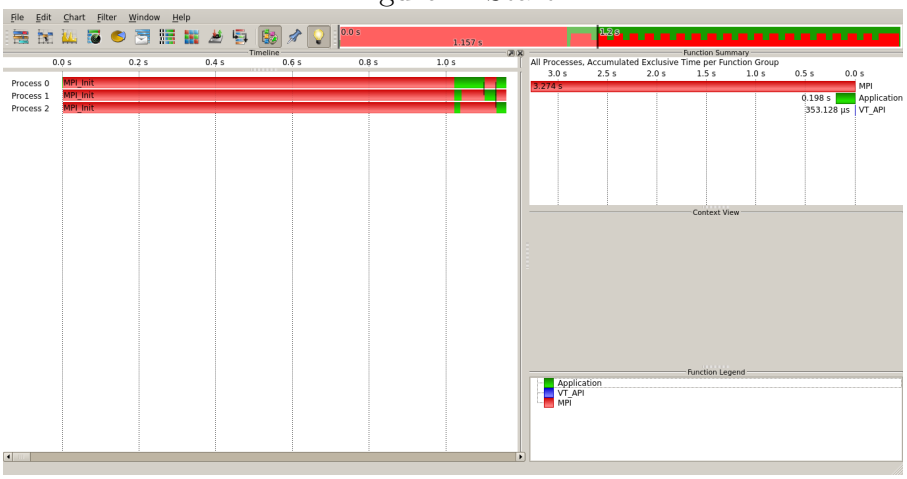


Figure 2: Iterationen

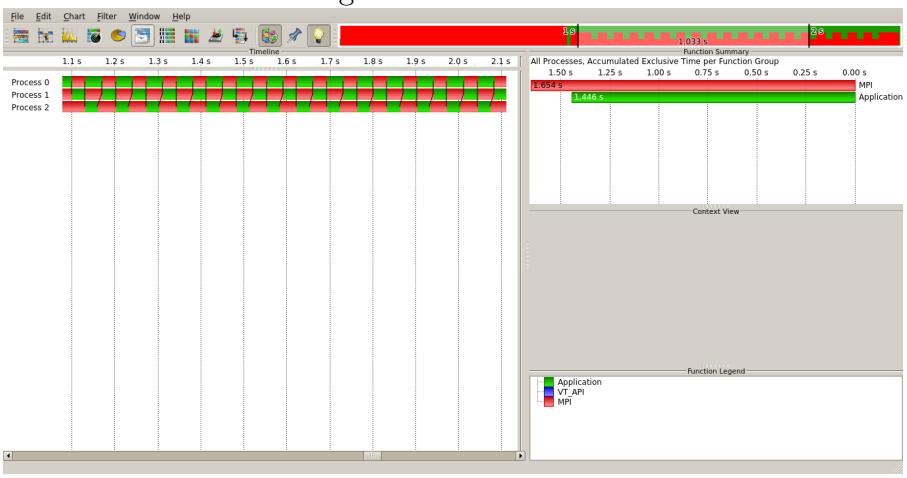


Figure 3: Einzelne Iteration

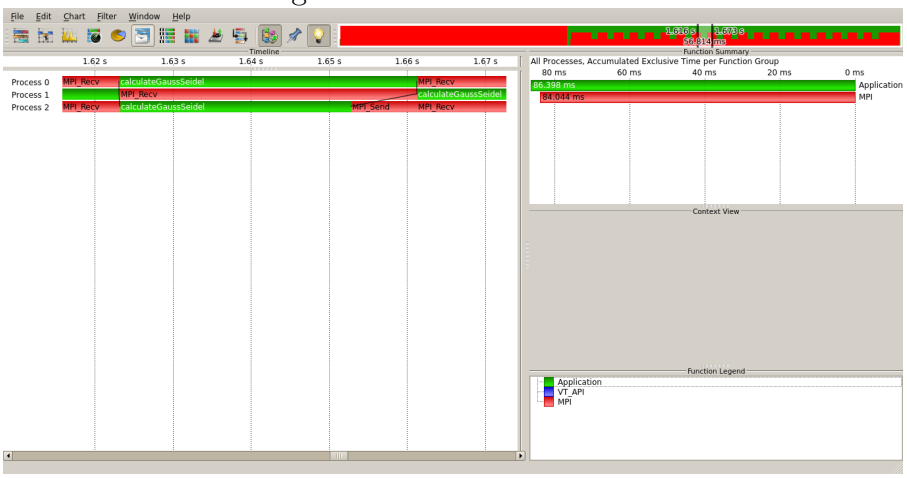


Figure 4: Synchronisation

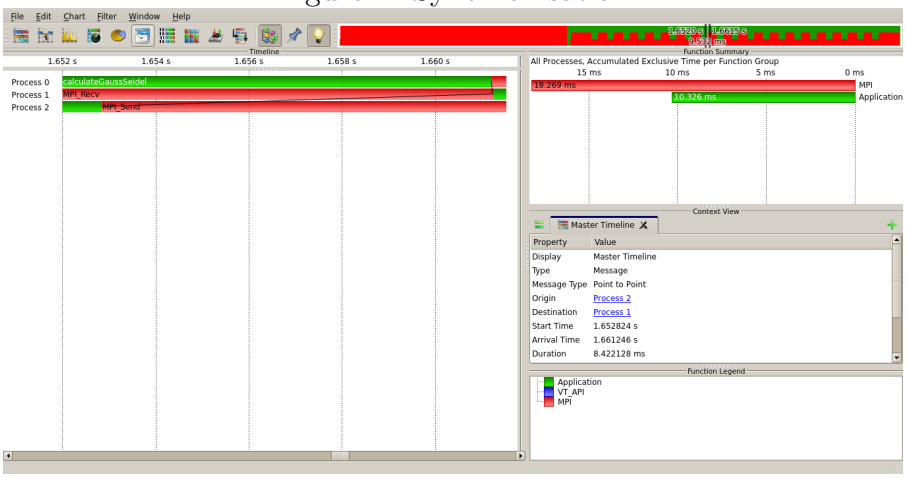
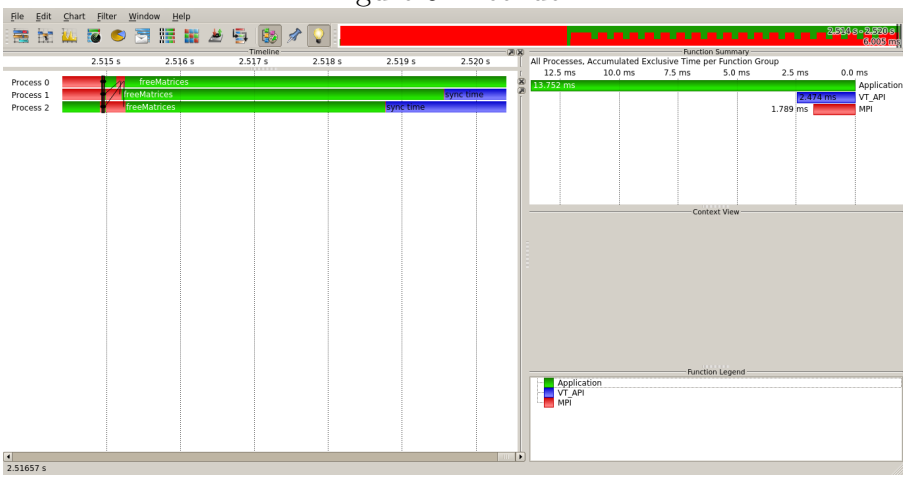


Figure 5: Beenden



Die Berechnung der Matrizen mittels des Gauss-Seidel Verfahrens läuft eigentlich wie erwartet, die einzige Anomalie, welche aber zu erwarten war, tritt bei der Synchronisation der Teilmatrizen nach einer Iteration auf. Hierbei berechnen einige Prozesse die Teilmatrix immer deutlich schneller als die Restlichen, dies folgt aber aus der Berechnung der halben Matrix, da hierbei einige Prozesse deutlich weniger Last haben können als andere.
Des Weiteren ist auch hier wie bei jedem MPI Programm die sehr lange Startzeit zu sehen, die das Programm benötigt um MPI zu initialisieren.

Jacobi:

Figure 6: Start

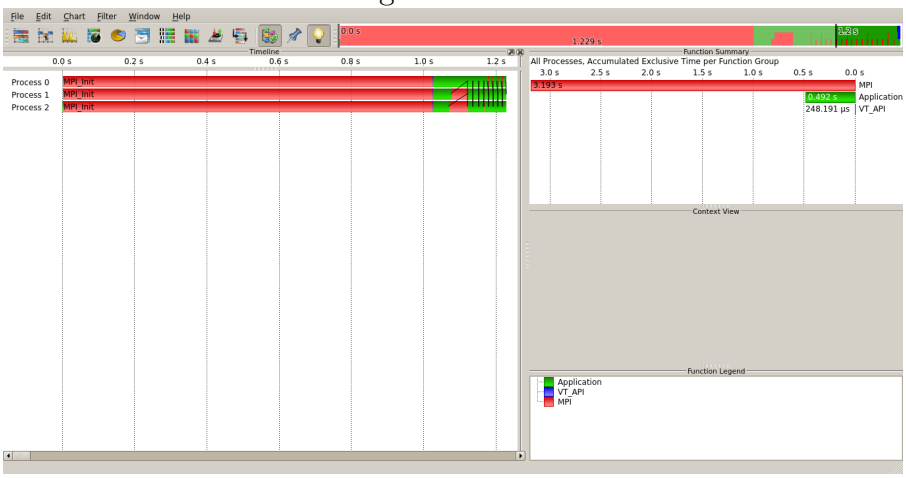


Figure 7: Alle Iterationen

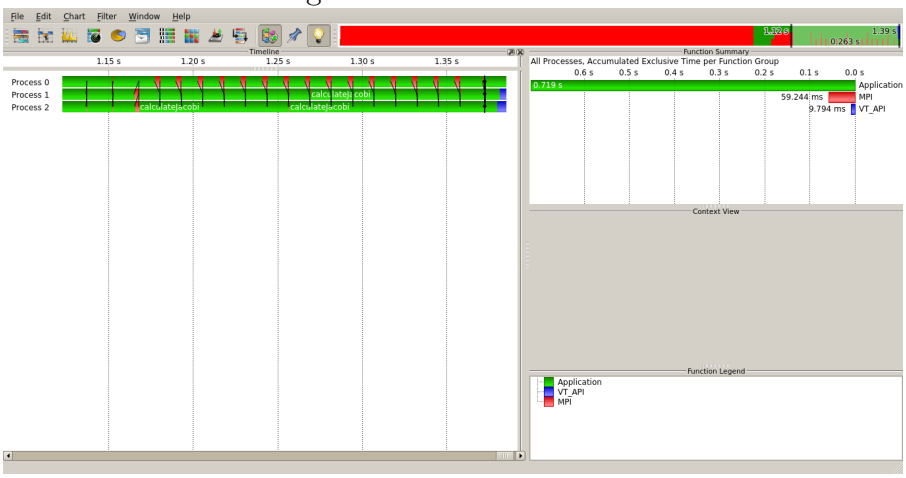


Figure 8: Eine Iteration

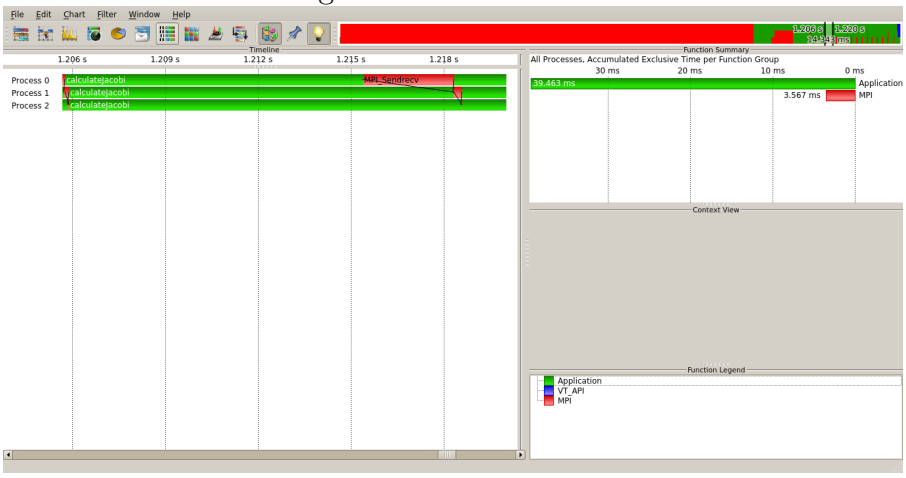


Figure 9: Synchronisation

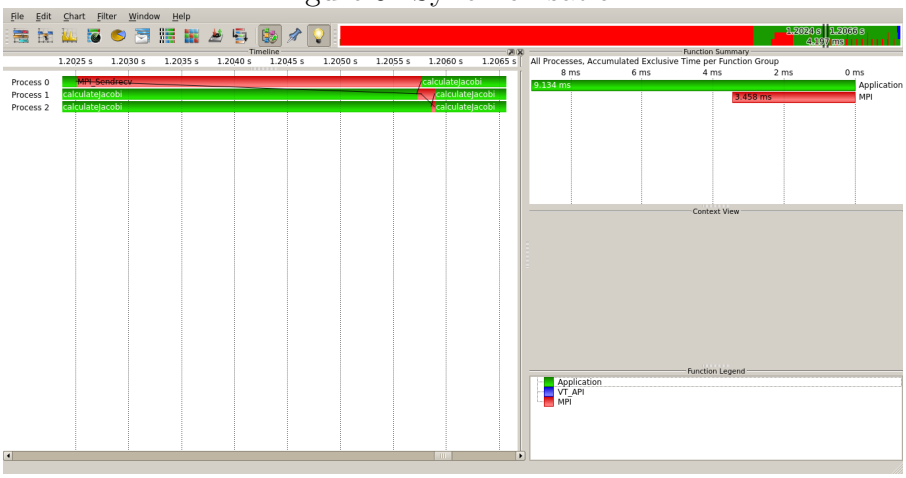
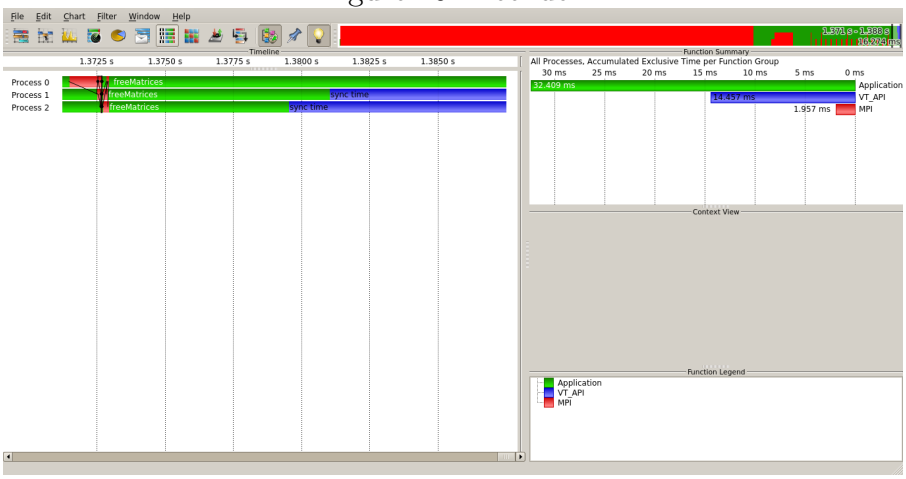


Figure 10: Beenden



Auch hier tritt der gleiche Effekt wie beim Gauss-Seidel Verfahren auf, sonst verhält sich das Programm wie erwartet.

Knoten 4 Prozesse 5:

Gauss-Seidel:

Figure 11: Start

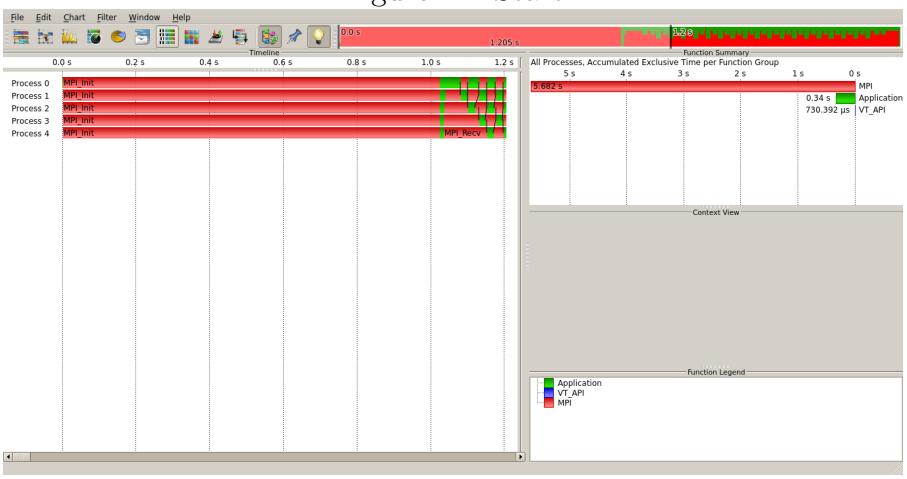


Figure 12: Alle Iterationen

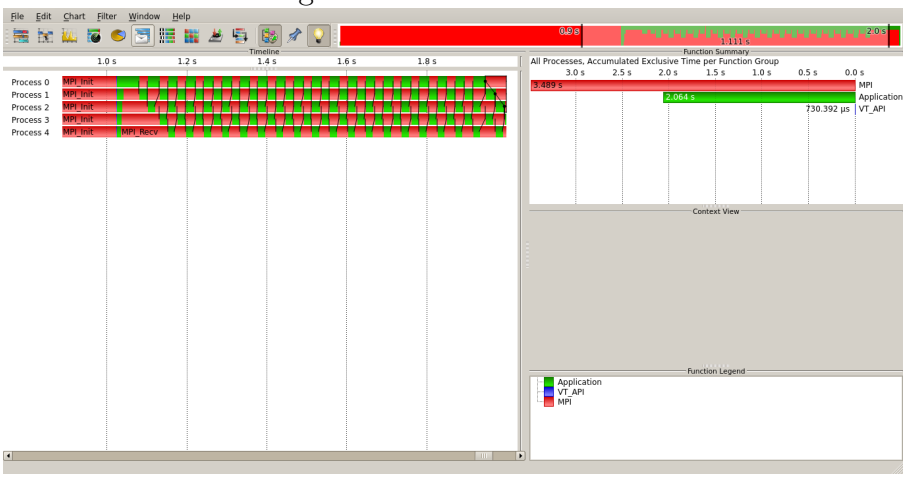


Figure 13: Eine Iteration

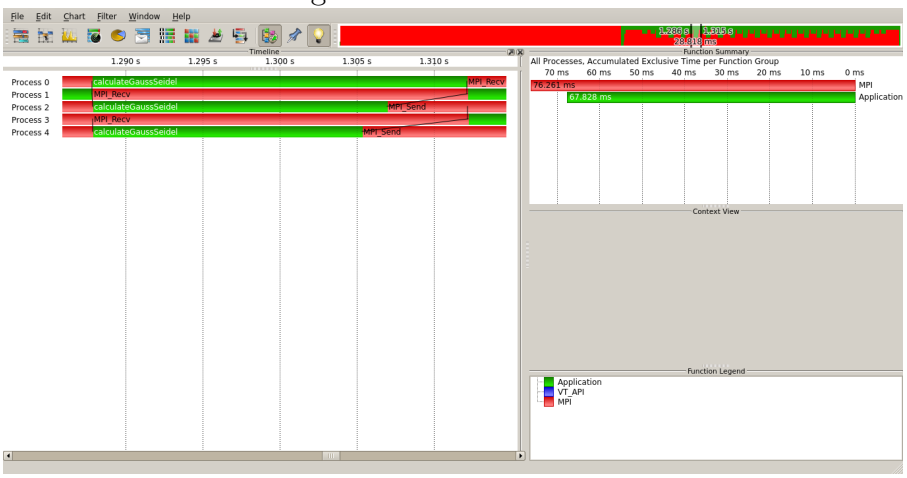


Figure 14: Synchronisation

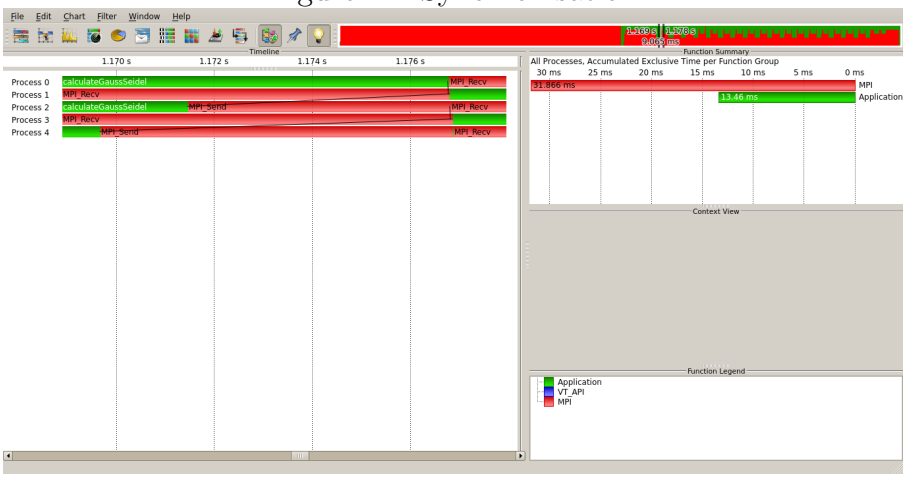
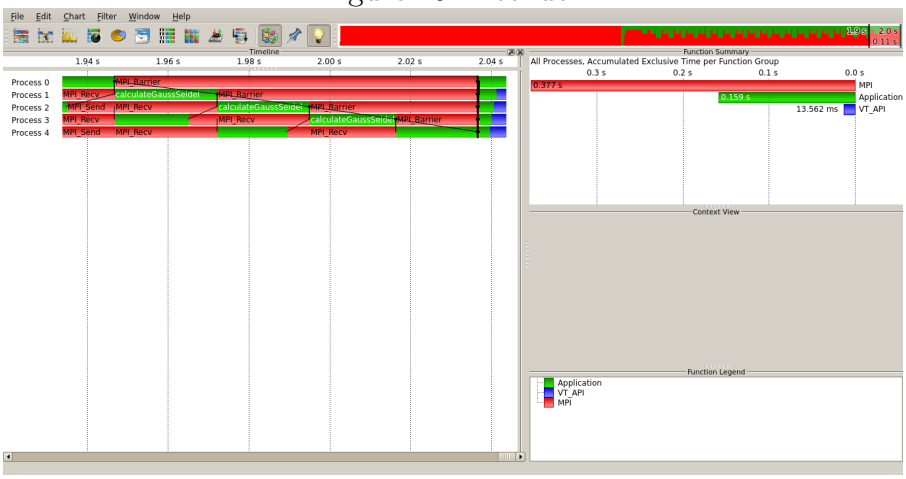


Figure 15: Beenden



Dies entspricht im wesentlichen wieder der Erwartungshaltung, dass sich die Prozesse in der Iteration abwechseln und sonst sauber starten sowie korrekt beenden. Auch hier tritt aber die Auffälligkeit auf, dass einige Prozesse immer schneller rechnen als andere, dies lässt sich aber mit der bereits beschrieben Ursache erklären.

Jacobi:

Figure 16: Start

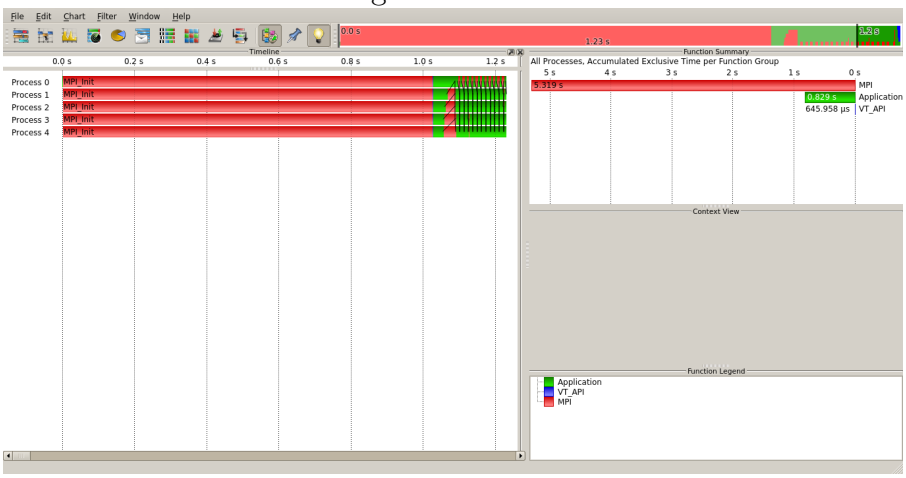


Figure 17: Alle Iterationen

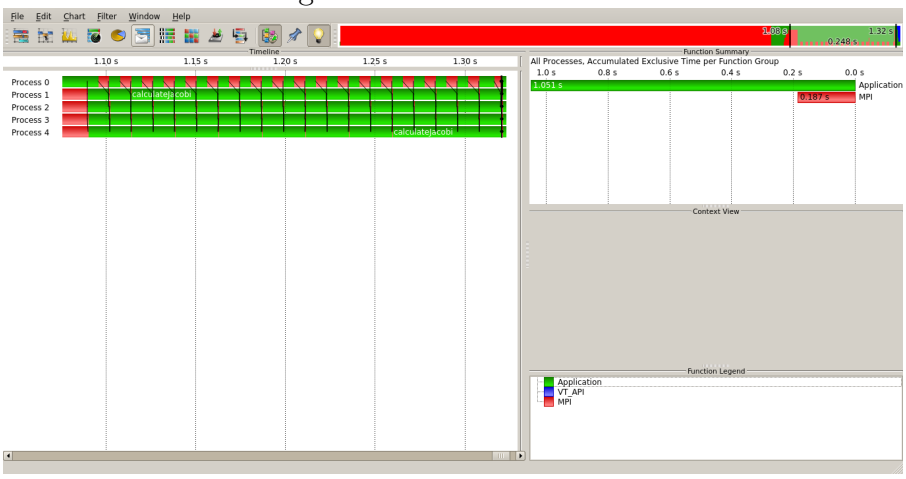


Figure 18: Eine Iteration

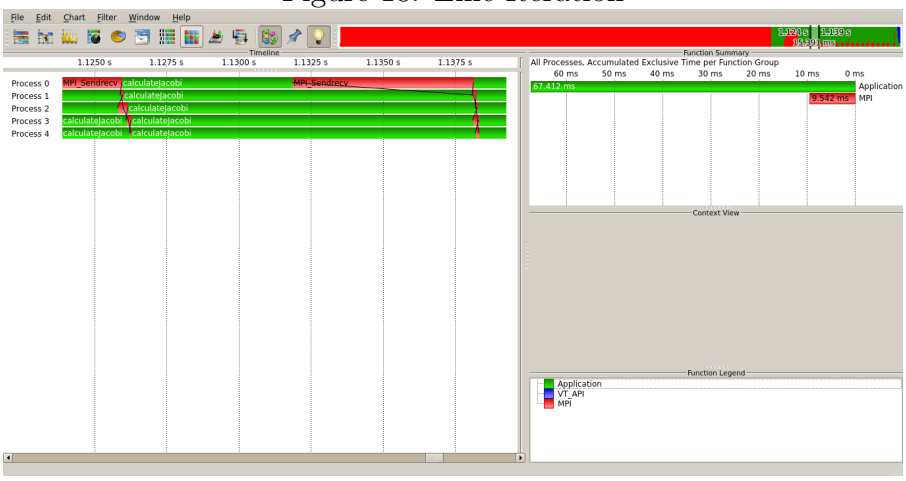


Figure 19: Synchronisation

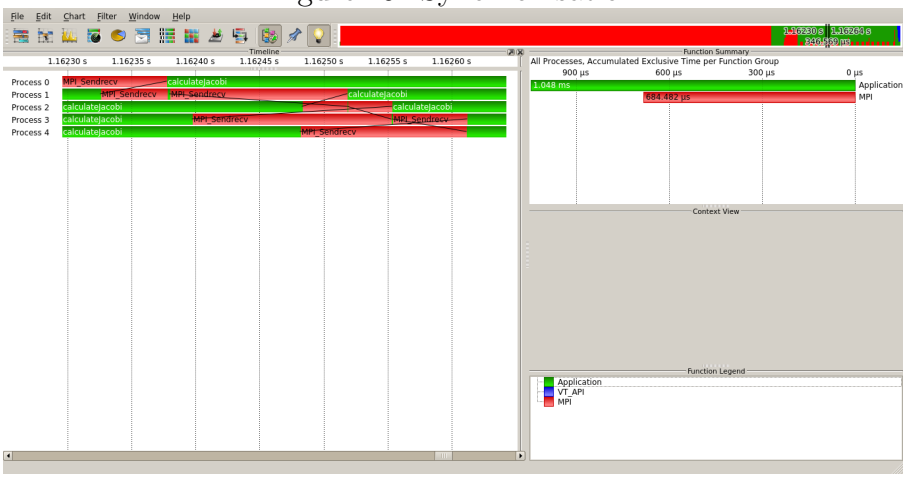
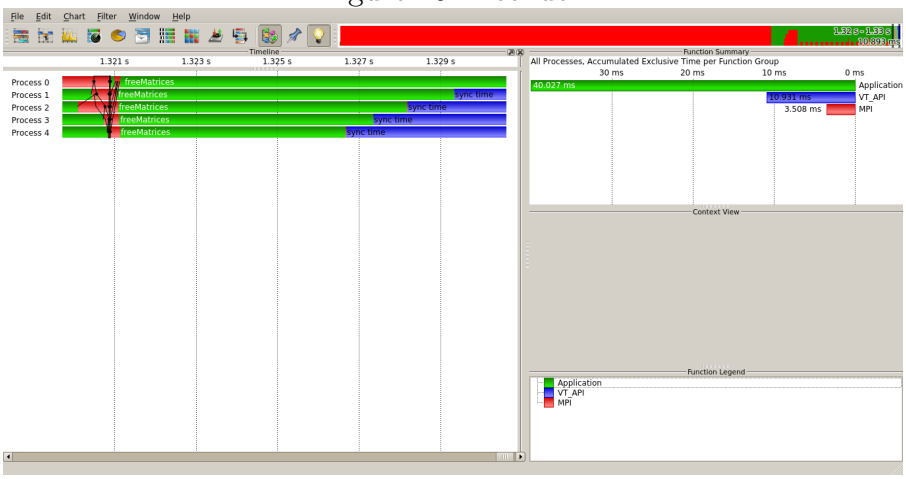


Figure 20: Beenden



Auch hier sind keine Auffälligkeiten zu erkennen und das Programm durchläuft die Schritte wie erwartet.

Knoten 10 Prozesse 10:

Gauss-Seidel:

Figure 21: Start

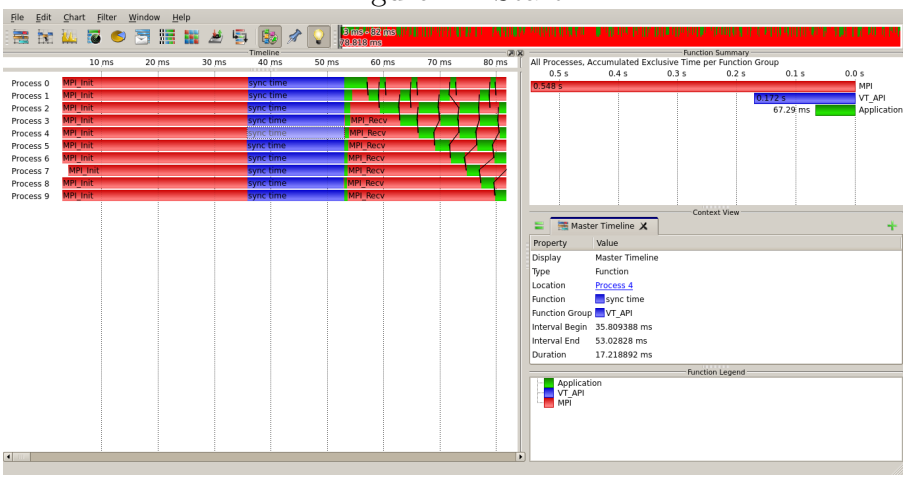


Figure 22: Alle Iterationen

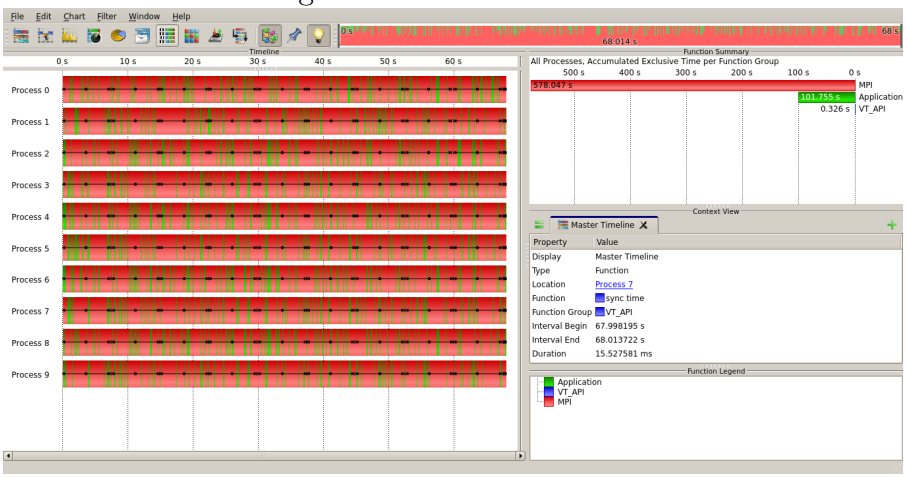


Figure 23: Eine Iteration

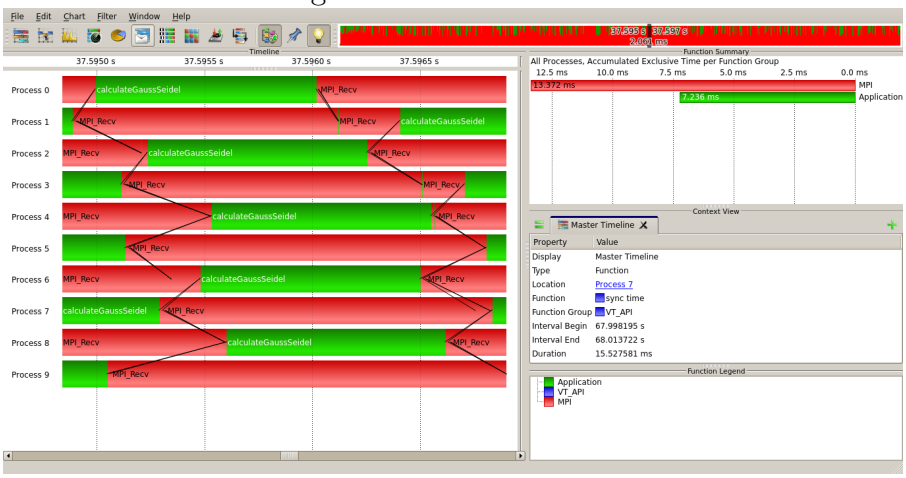


Figure 24: Synchronisation

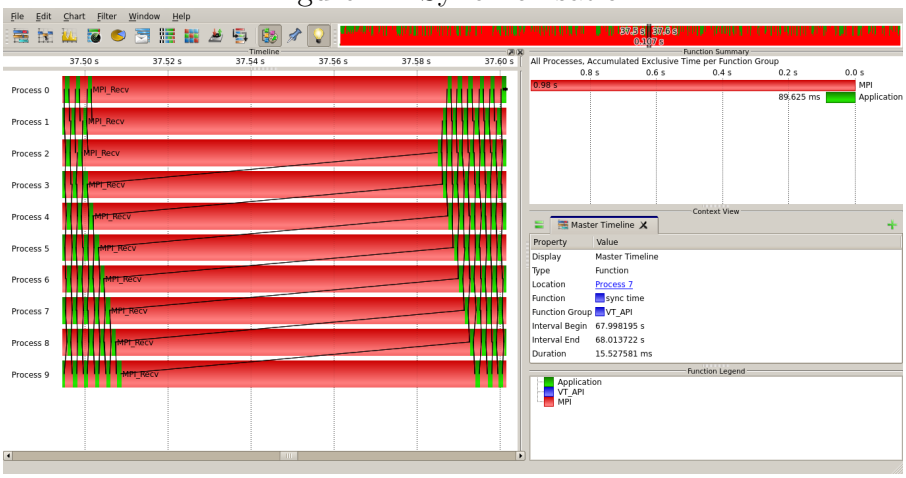
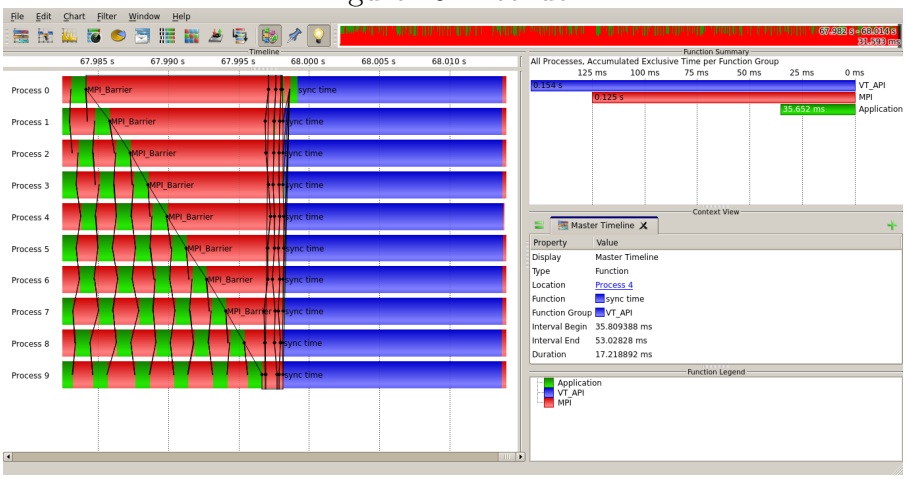


Figure 25: Beenden



Hier treten zu ersten mal Anomalien auf, die nicht zu erwarten waren, wie zum Beispiel der sehr langen Synchronisationsphase, welche in immer wieder kehrenden Abständen sehr lange benötigt, bis die Prozesse ihr Teilmatrizen ausgetauscht haben.
Leider haben wir hierfür keine rationale Erklärung bis auf die Vermutung, dass andere Prozesse oder Jobs, welche das Netzwerk benutzen dieses kurzzeitig voll auslasten, so dass keine weitere Kommunikation möglich ist.
Eventuell ist auch der Switch kurzzeitig ausgelastet, so dass ebenfalls keine Kommunikation stattfinden kann.
Eine Andere Erklärung hierfür könnte aber auch sein, dass in OpenMPI nach einer bestimmten Anzahl von Pakten welche versendet wurden Daten zwischen den Prozessen getauscht werde müssen, welche nicht aufgezeichnet wurden oder eine Art Garbage Collection durchgeführt wird.

Jacobi:

Figure 26: Start

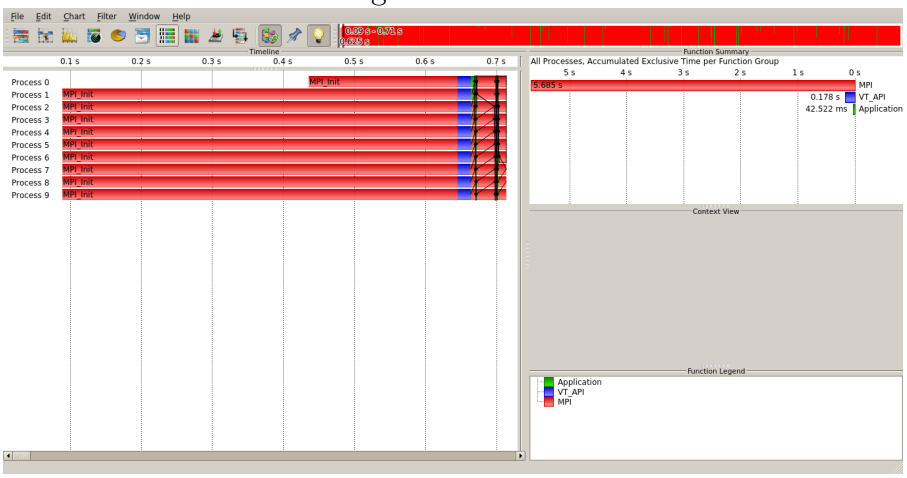


Figure 27: Alle Iterationen

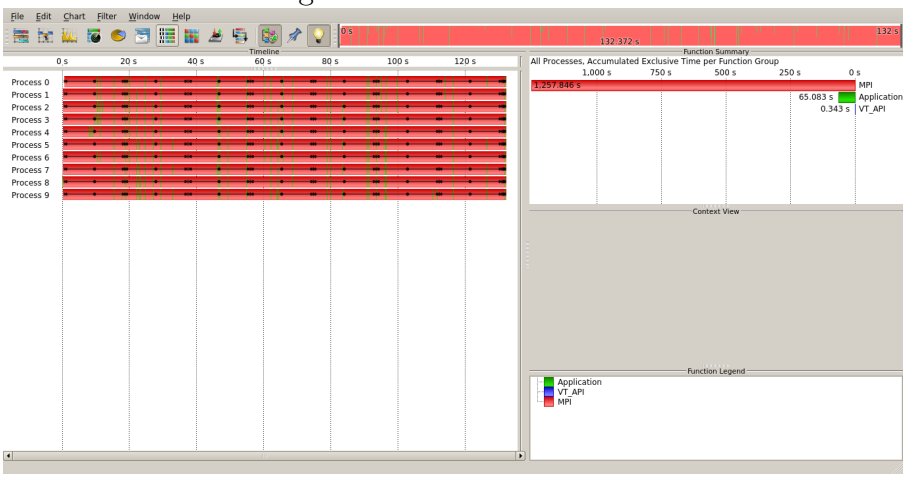


Figure 28: Eine Iteration

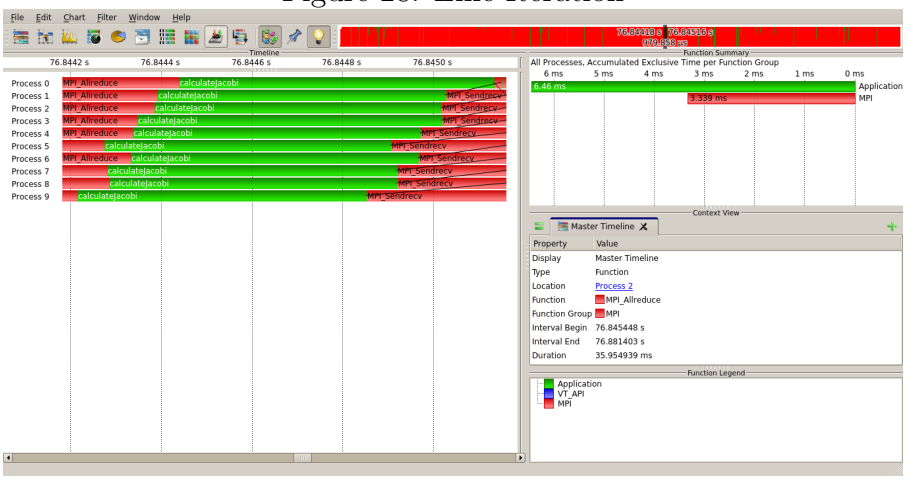


Figure 29: Synchronisation

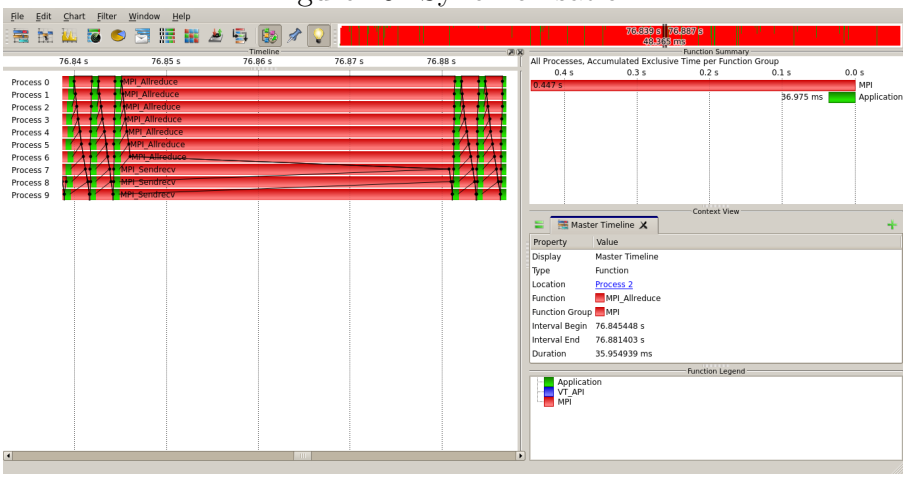
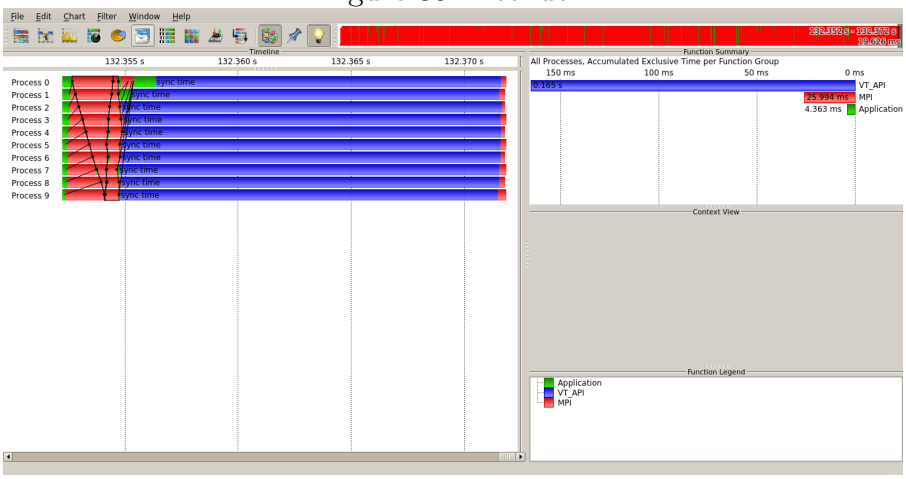


Figure 30: Beenden



Hier ist ebenfalls eine nicht erwartete Anomalie aufgetreten, da während der Synchronisation das Allreduce den Großteil der Zeit einnimmt. Dies könnte daher kommen, dass ab einer bestimmten Anzahl von Knoten die Allreduce Funktion auf einer Baumstruktur arbeitet und somit, da die Problemgröße noch nicht groß genug ist, sich diese Art der Kommunikation noch nicht lohnt. Hierbei kann es dann zu einer Verzögerung kommen, welche deutlich größer ist, als wenn die Kommunikation linear durchgeführt wird.
Alternativ könnten hier aber auch die gleichen Argumente verwendet werden wie beim Gauss-Seidel Verfahren, da hier aber keines der Argumente wirklich Plausibel ist, handelt es sich hierbei nur um Mutmaßungen.