

Simulation der MPI collective-operations im PIOsimHD

Artur Thiessen
Informatik Bachelor
Matrikelnummer: 5944906
7thiessa@informatik.uni-hamburg.de

Inhaltsverzeichnis

Einführung

Aufgabenstellung

Beispiele

Fazit

Probleme

Quellen



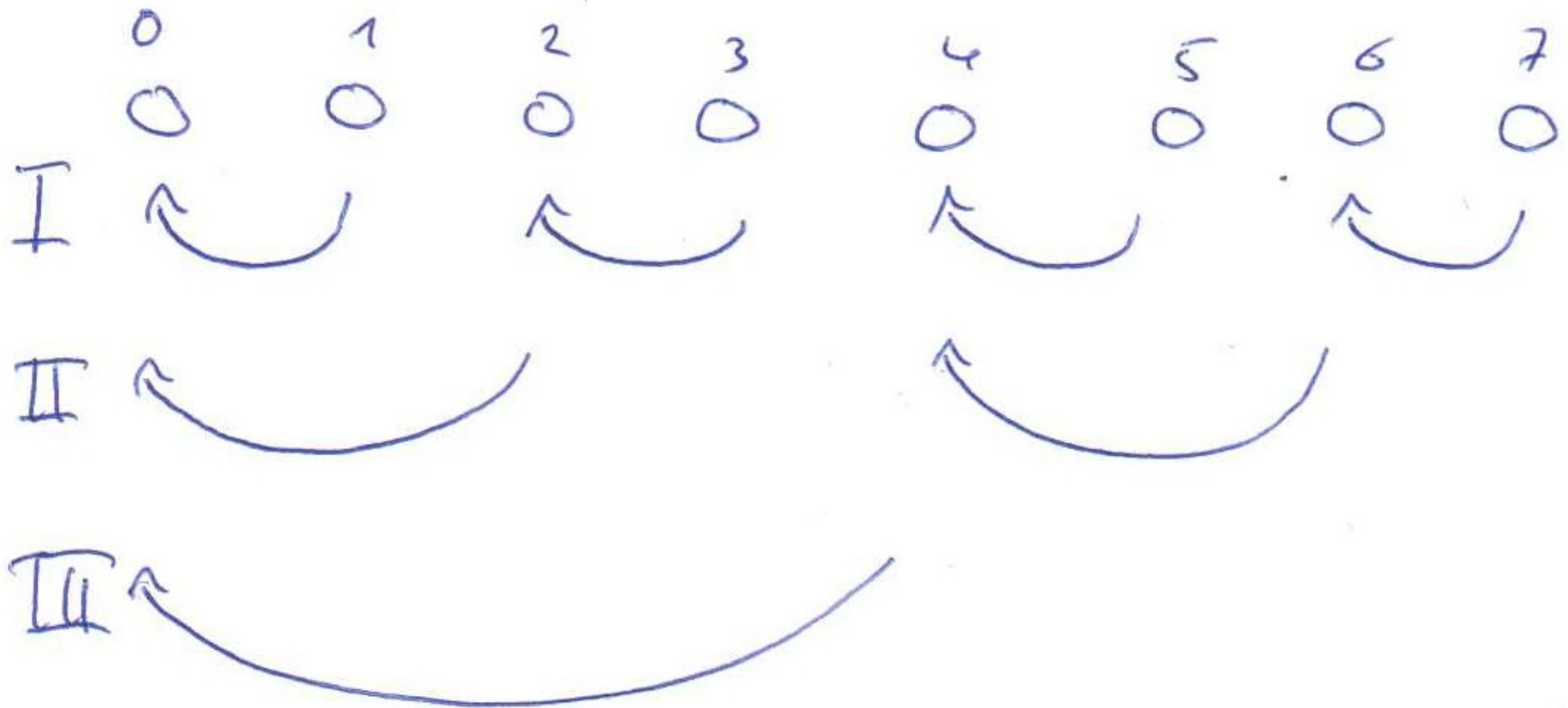
Einführung

- Alltagsherausforderungen stellen uns vor mathematische Probleme
- Diese sind oft nicht mit herkömmlichen Heim-PCs in annehmbarer zu bewältigen
- Cluster bringen genügend Rechenleistung für solche Aufgaben
- Die einzelnen unabhängigen Prozesse in einem Cluster kommunizieren z.B. über MPI

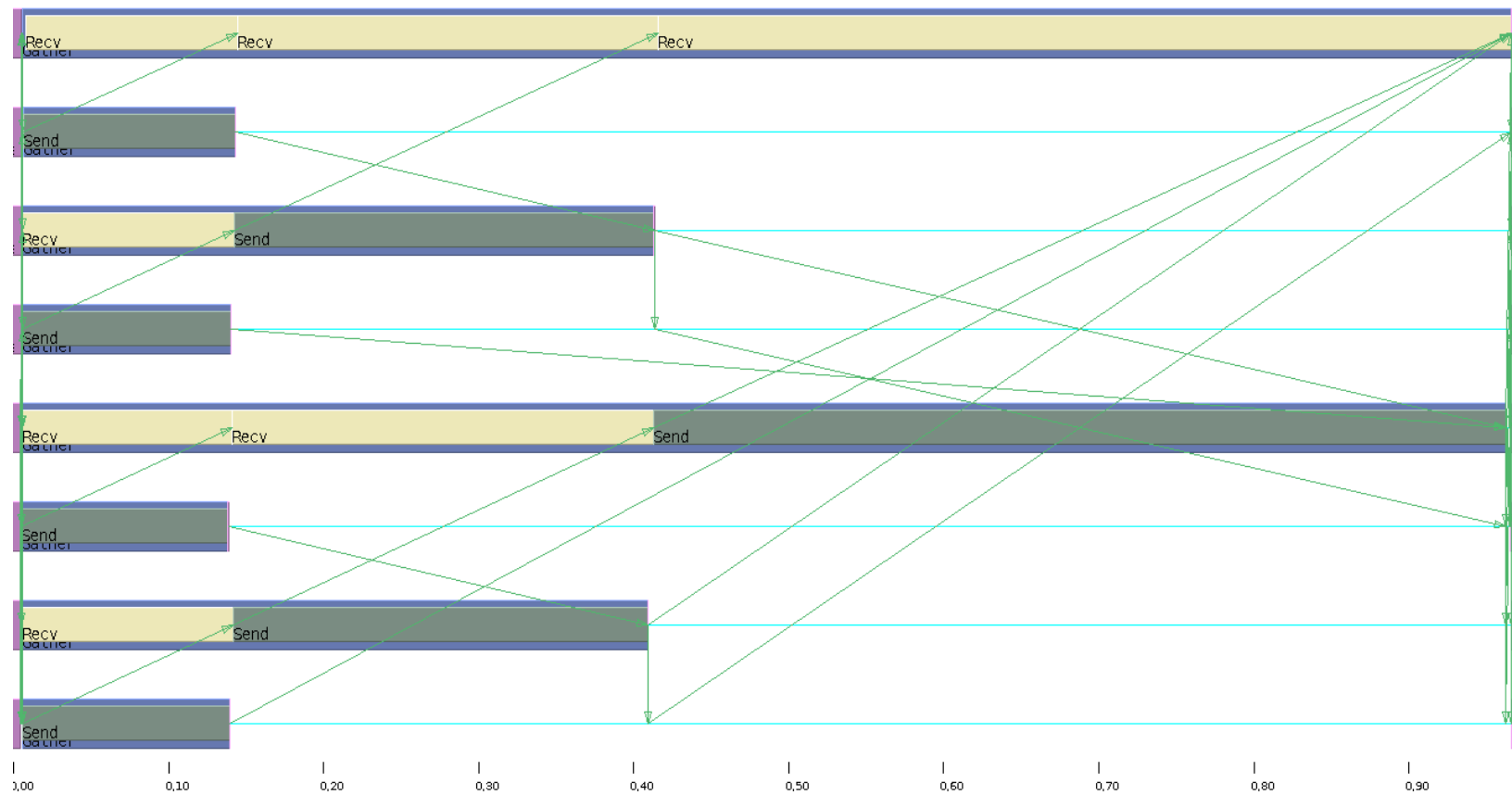
Aufgabenstellung

- Einarbeitung in den PIOsimHD Simulator
- Implementierung der MPI collective operations
- Vergleich der Laufzeiten von Simulator und Cluster
- Grafische Gegenüberstellung der Laufzeiten
- Bonus: Entwicklung von noch schnelleren Algorithmen

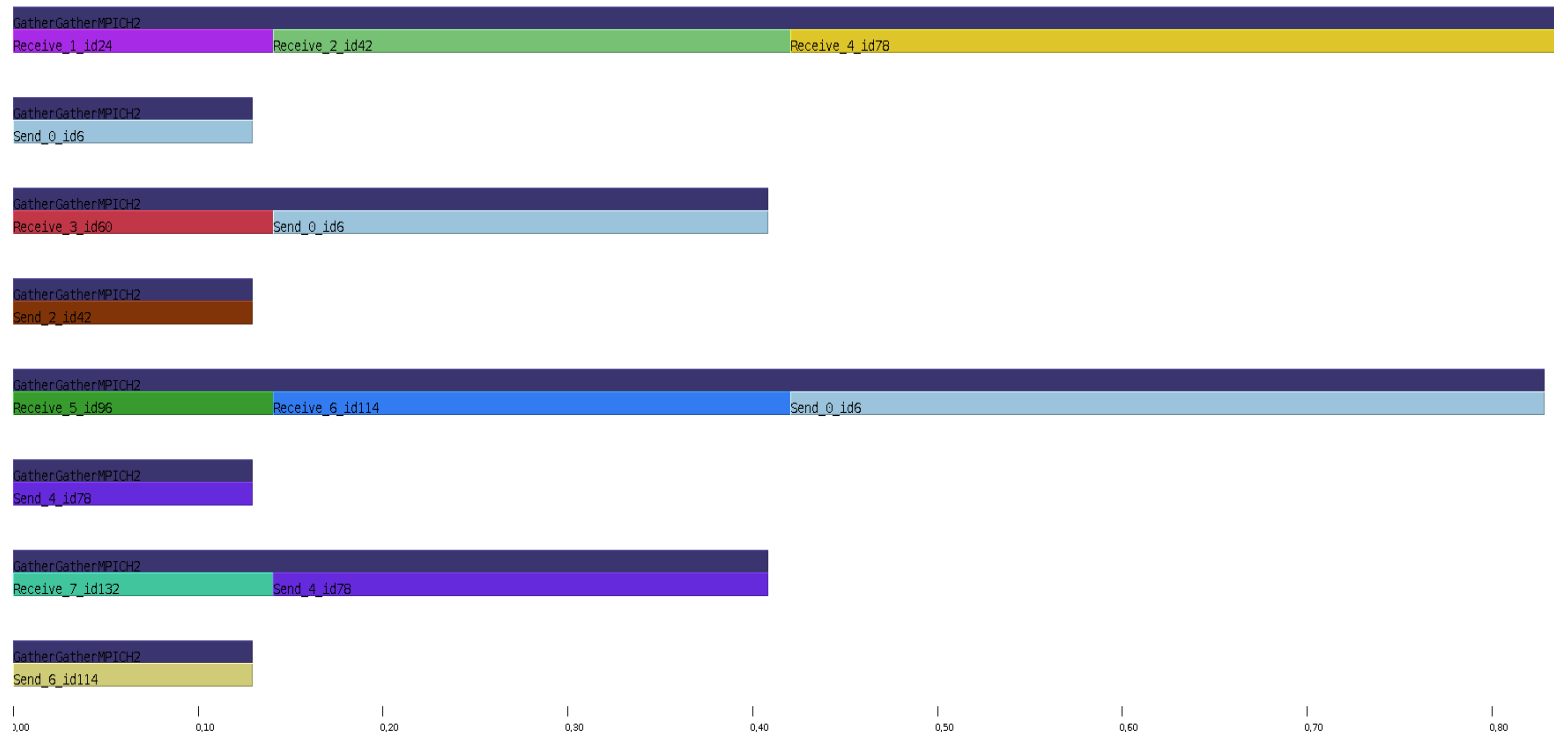
Beispiel – MPI_Gather (Skizze)



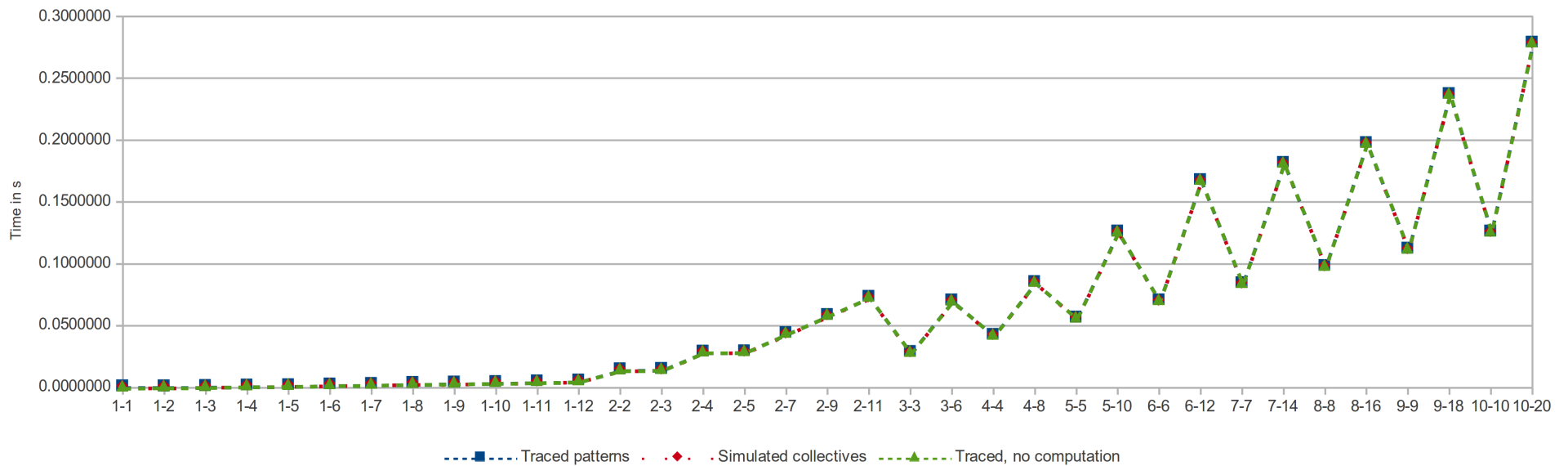
Beispiel – MPI_Gather (Cluster)



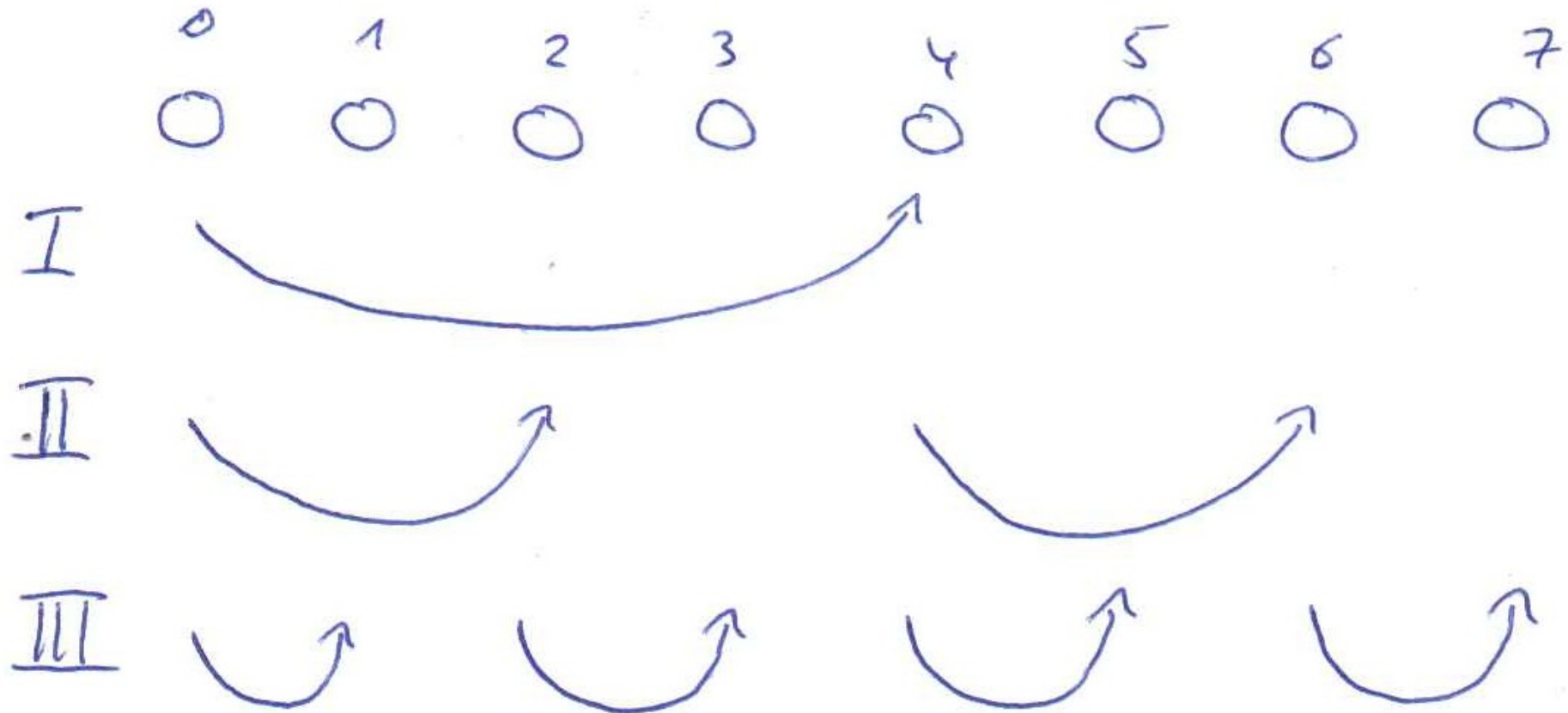
Beispiel – MPI_Gather (PIOsimHD)



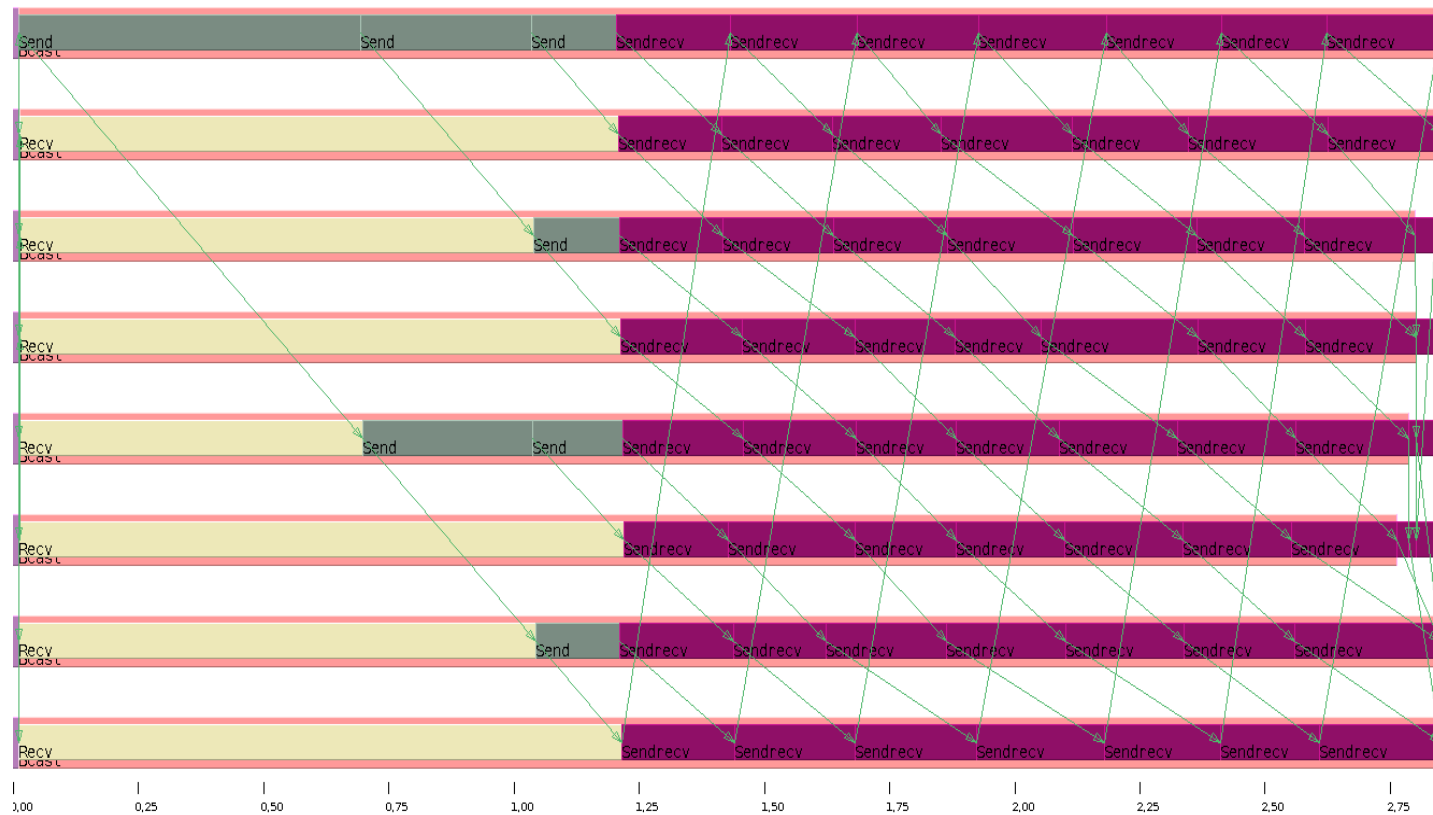
Beispiel – MPI_Gather (Plott)



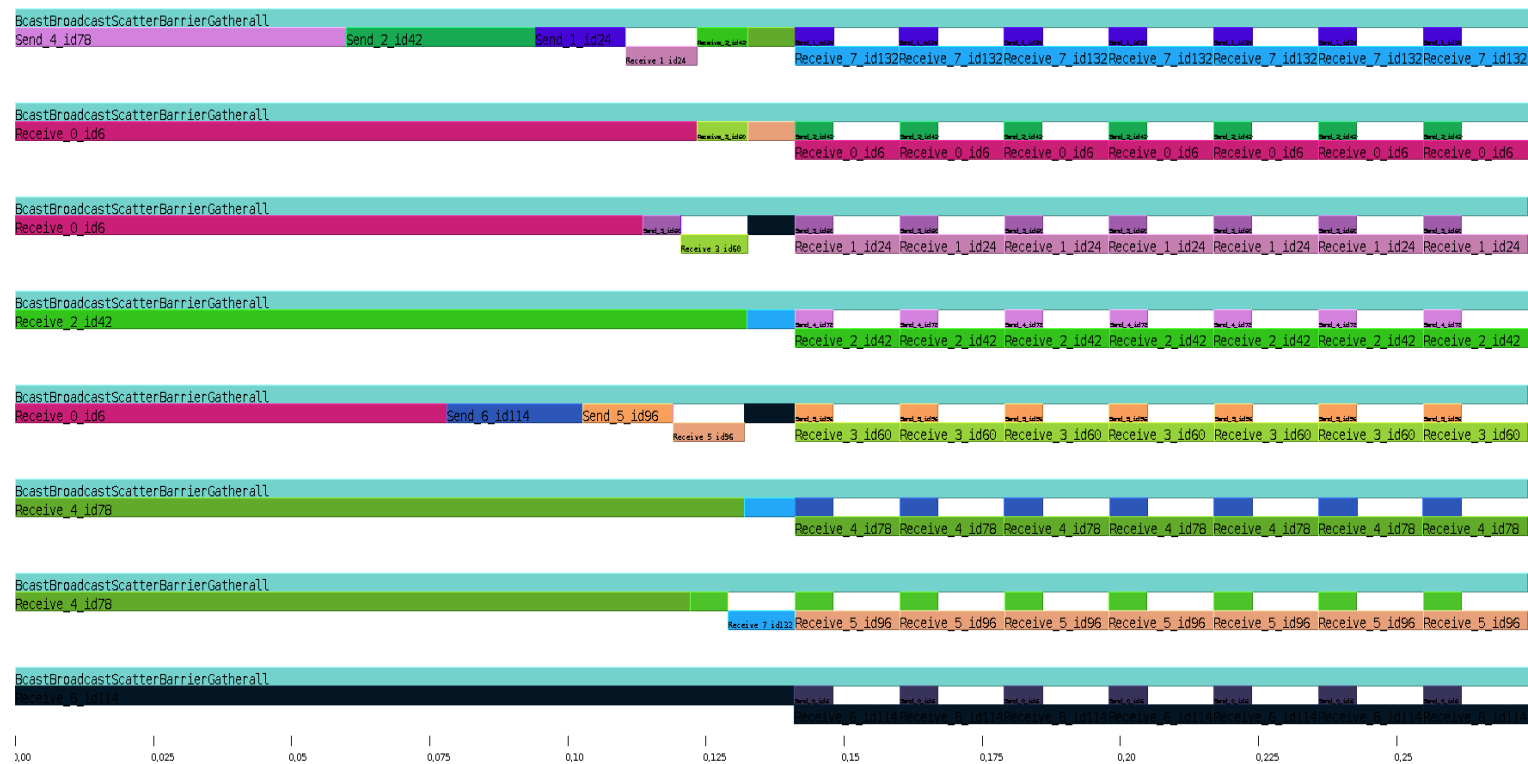
Beispiel – MPI_Bcast (Skizze)



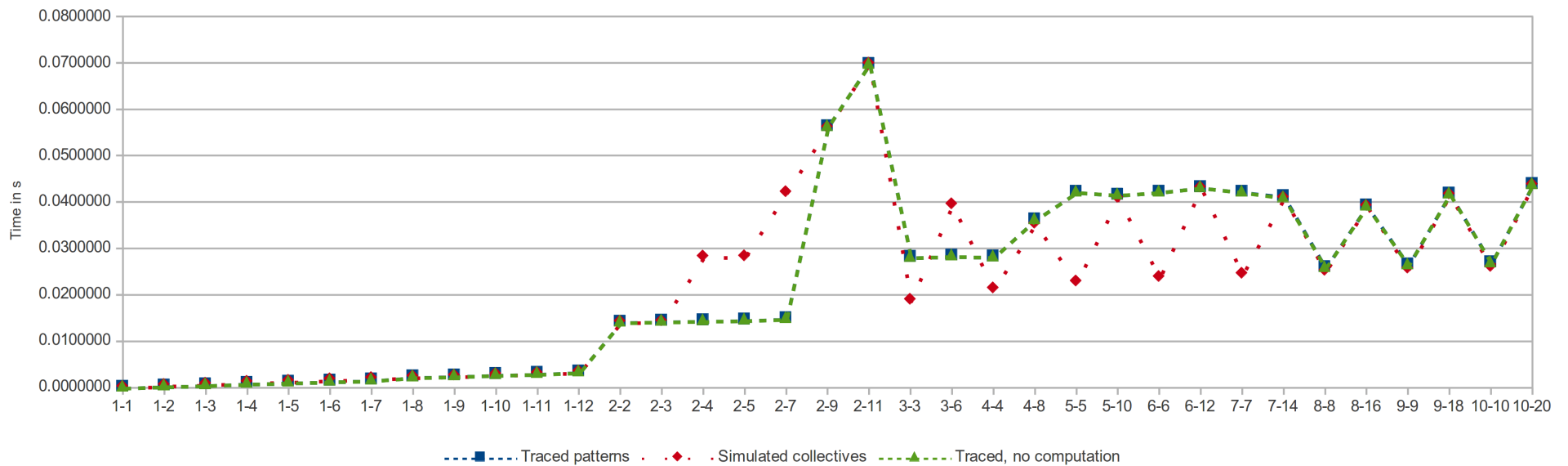
Beispiel – MPI_Bcast (Cluster)



Beispiel – MPI_Bcast (PIOsimHD)



Beispiel – MPI_Bcast (Plott)



Fazit

- Der PIOsimHD Simulator ist ein gutes Werkzeug zum Analysieren von Kommunikationslaufzeiten, ohne einen echten Cluster nutzen zu müssen
- Vieles muss noch optimiert werden z.B. Hardwareparameter
- Im nächsten Schritt könnten nun bessere/schnellere Algorithmen entwickelt werden

Probleme / Schwierigkeiten

- Einarbeitung in den PIOsimHD Simulator
- Arbeiten auf einem sich ständig ändernden System
- Extrahieren der Kernalgorithmen aus den originalen MPICH2 Implementierungen

Quellen & Tools

- MPI Forum (<http://www.mpi-forum.org/>)
- MPICH2 Implementation
- Julian Kunkel

- PIOsimHD & Sunshot
- Git
- Eclipse & Java SDK