

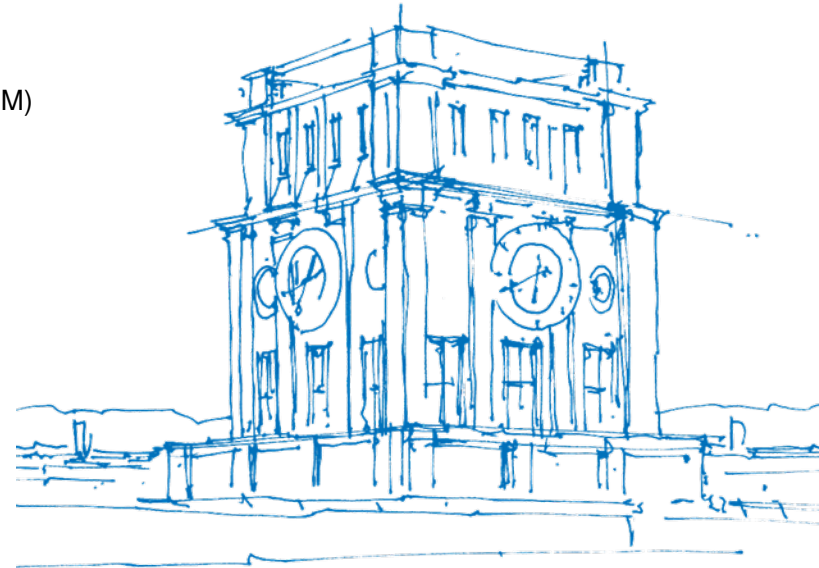
High Performance Computing für Maschinelle Intelligenz

Abschlussvortrag | Gruppe 1

Christoph, Alexander | Stümke, Daniel | Kiechle, Johannes

Fakultät für Elektro- und Informationstechnik, Technische Universität München (TUM)

09.02.2021

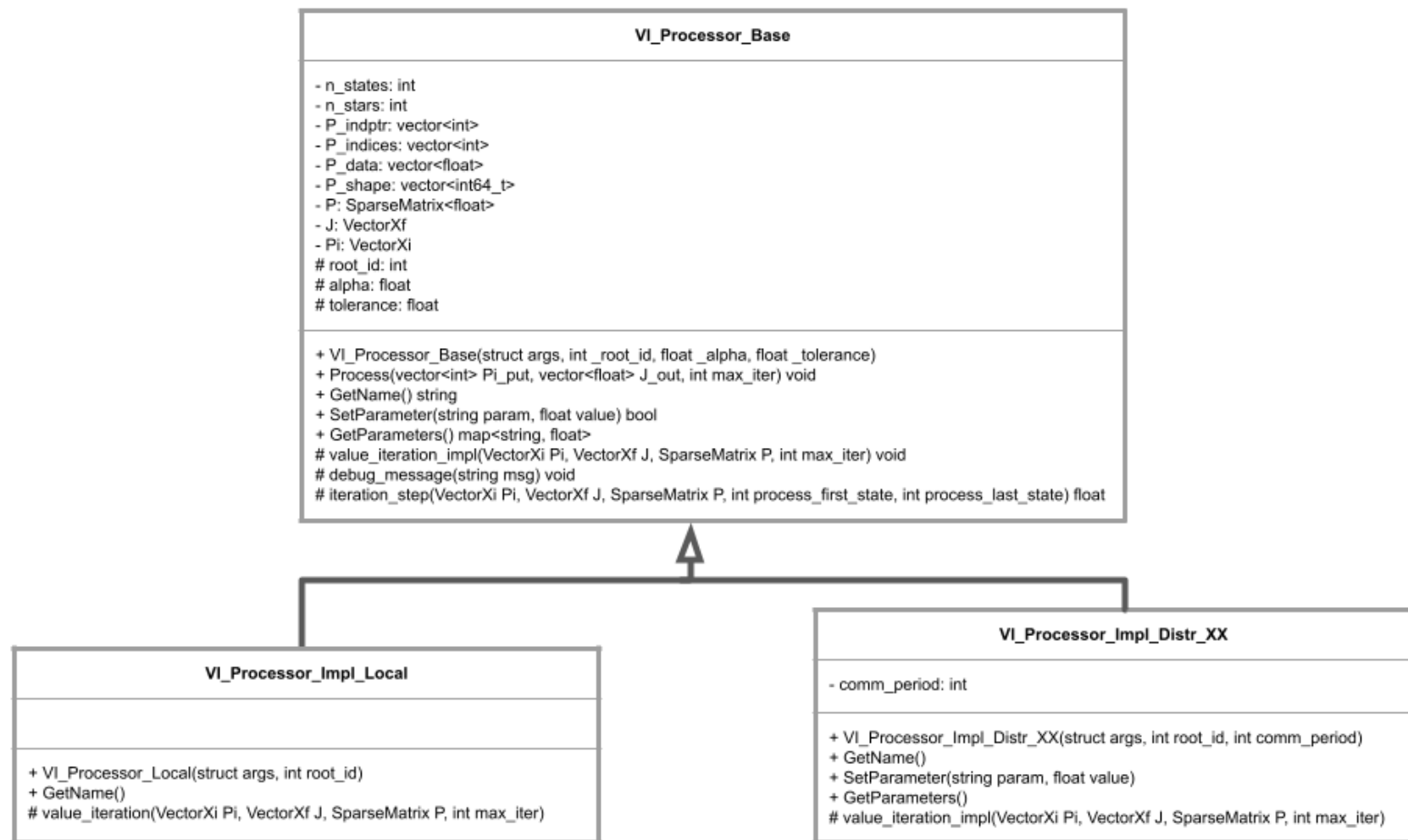


TUM Uhrenturm

Inhaltsverzeichnis

- 1 Projekt Struktur
- 2 OpenMPI Kommunikationsschemata
- 3 Ergebnisse
- 4 Fazit

Projekt Struktur



Pipeline



Makefile

- Kompilieren
- Starten der Skripts
 - run_debug
 - run_small
 - run_normal
- Schleifendurchlauf
(N = 2, 4, 6, 8)

main.cpp

Durchführung Value
Iteration für alle
Kommunikations-
schemata

benchmark_comm_period.py

Visualisierung:
Mittlere Ausführungszeit über
Kommunikationsfrequenz /
Updatehäufigkeit

(Vergleich aller Implementationen)

benchmark_np.py

Visualisierung:
Mittlere Ausführungszeit über
die verschiedenen Datensätze
bei variabler Anzahl der
Prozessoren

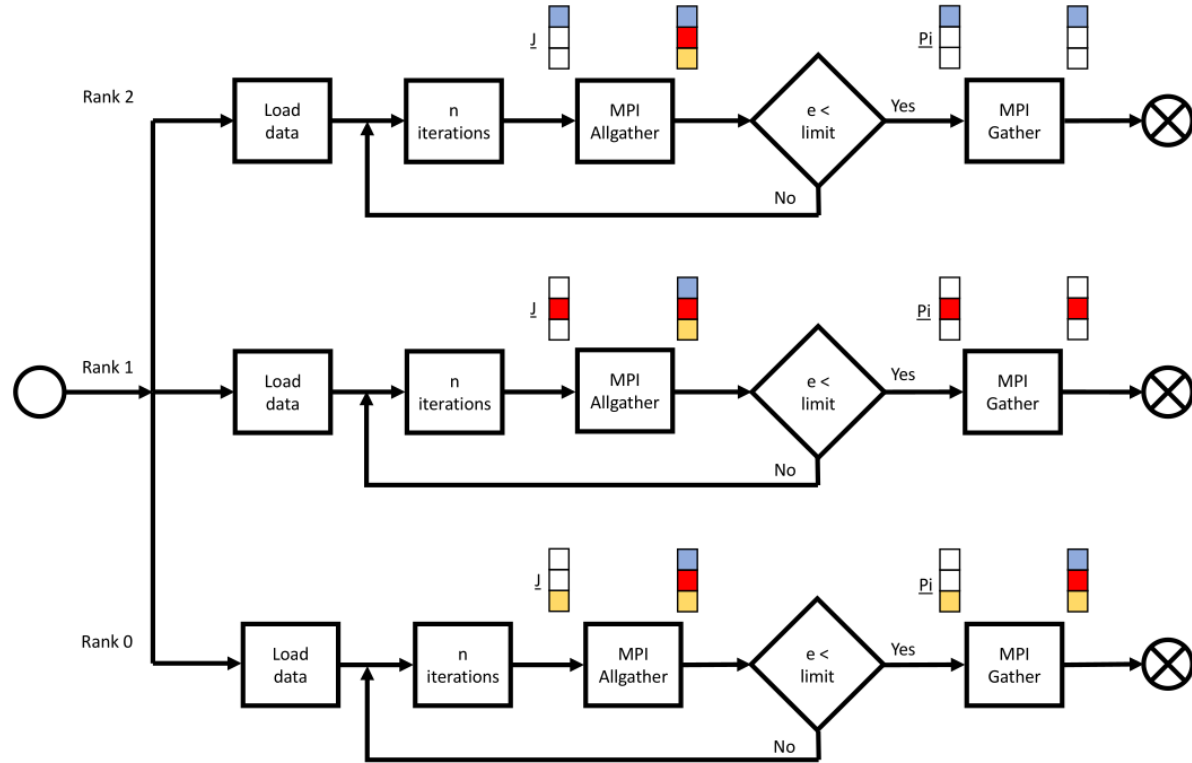
OpenMPI Kommunikationsschemata

Übersicht verwendeter OpenMPI API's

- Allgatherv + Allreduce + Gatherv
- Send + Recv + Bcast
- Sendrecv + Gatherv
- Isend + Irecv + Ibcast + Igatherv
- Igatherv + Bcast + Gatherv

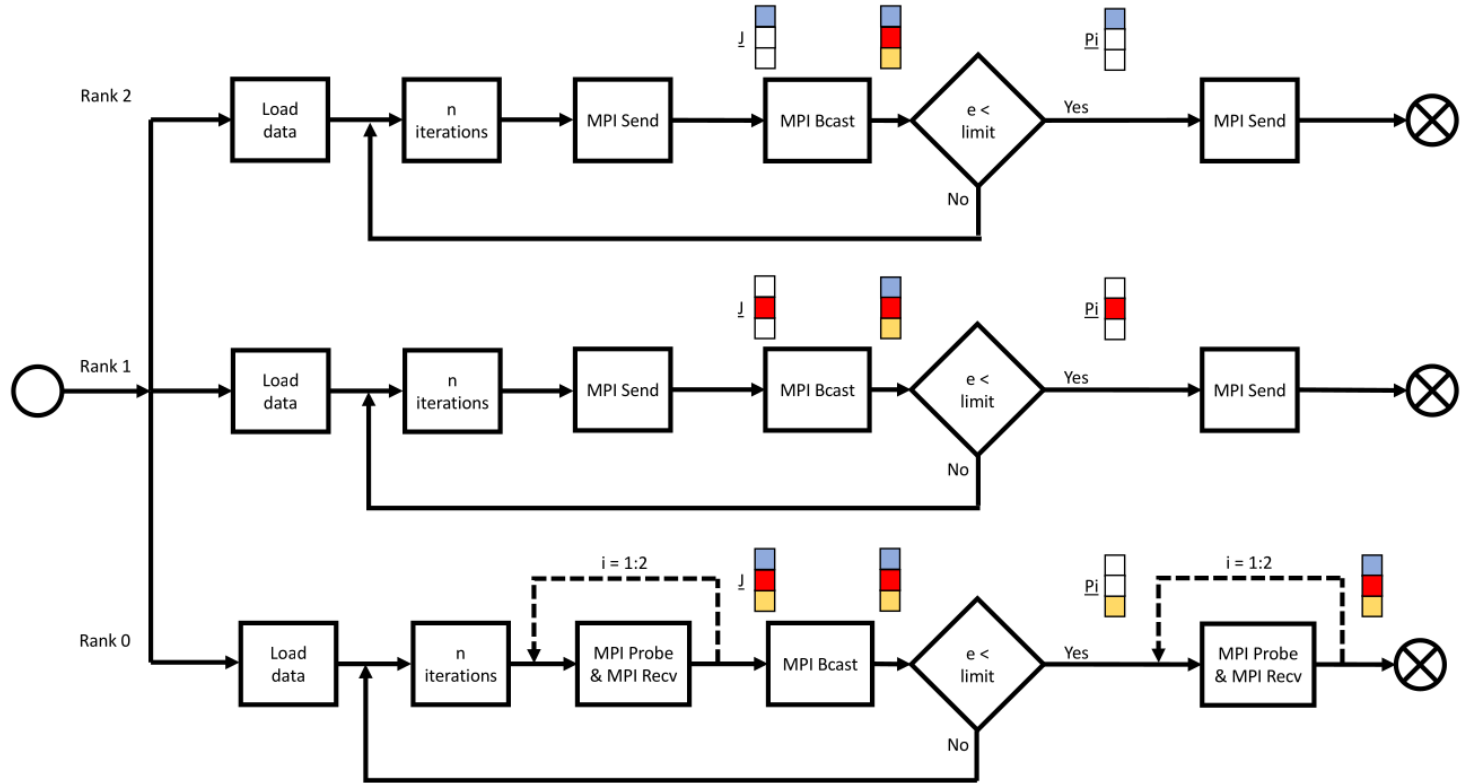
Schema 1

- Allgatherv
- Allreduce
- Gatherv



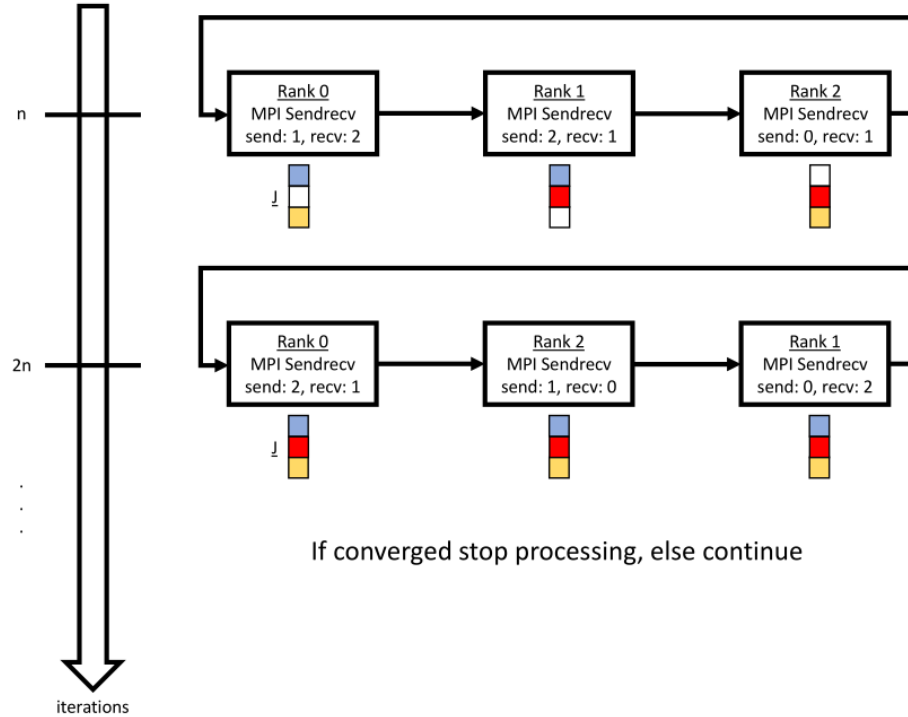
Schema 2

- Send
- Recv
- Bcast



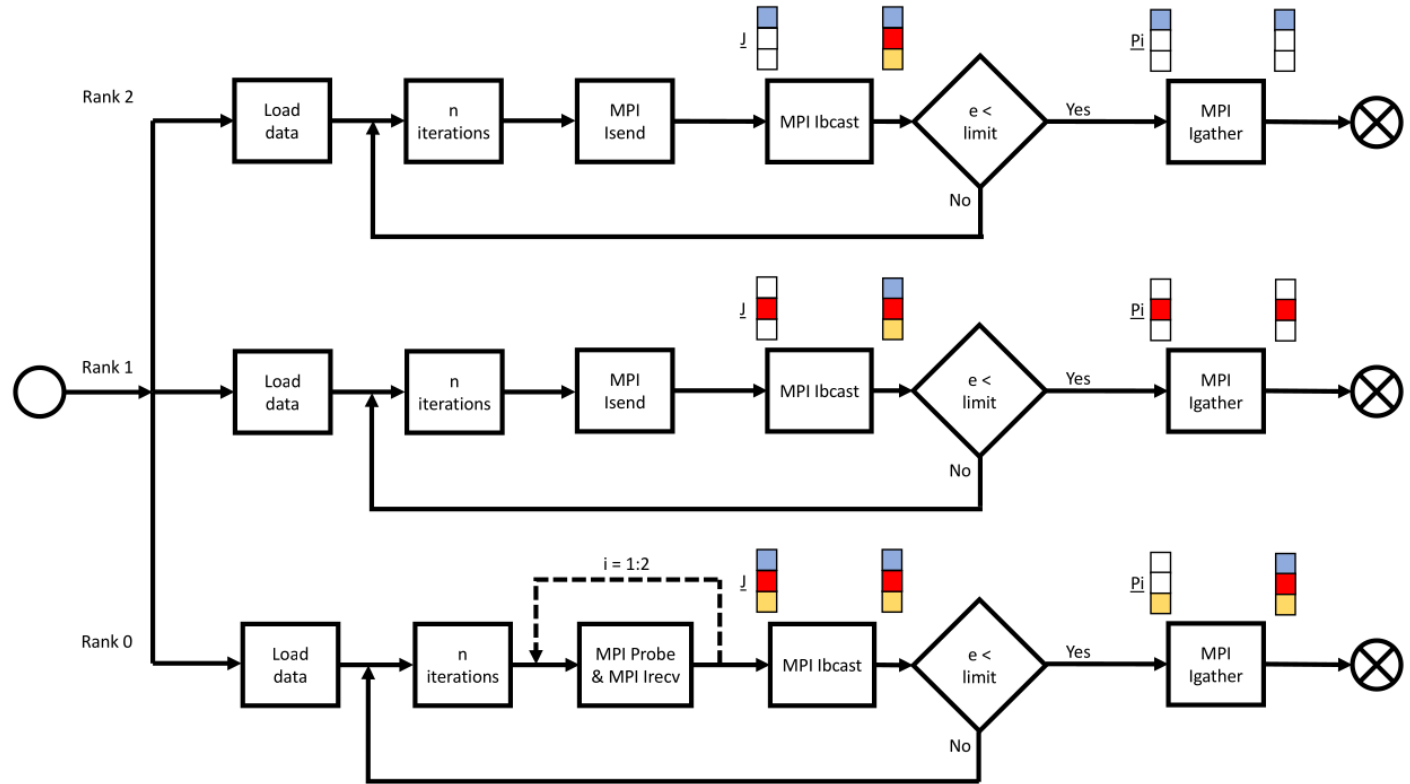
Schema 3

- Sendrecv
- Gather



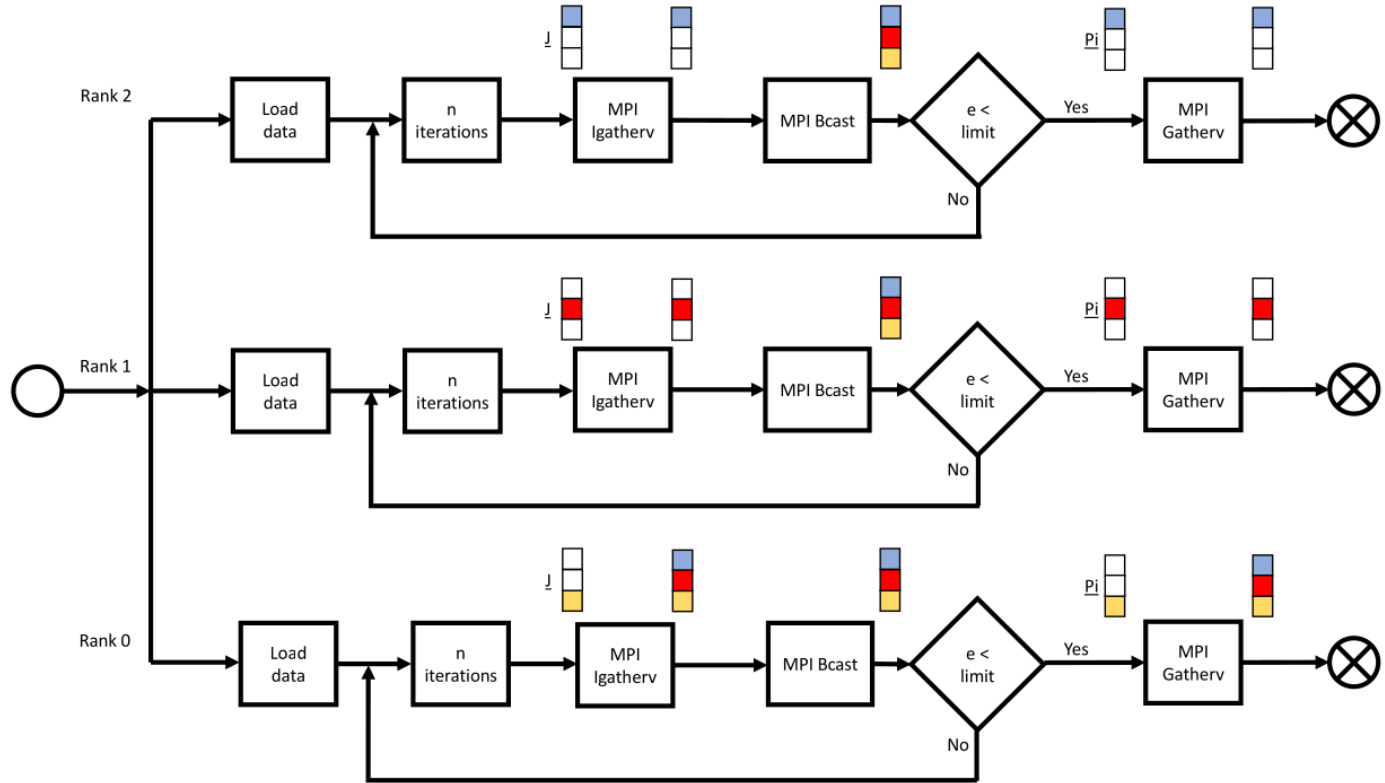
Schema 4

- Isend
- Irecv
- lbcast
- lgatherv



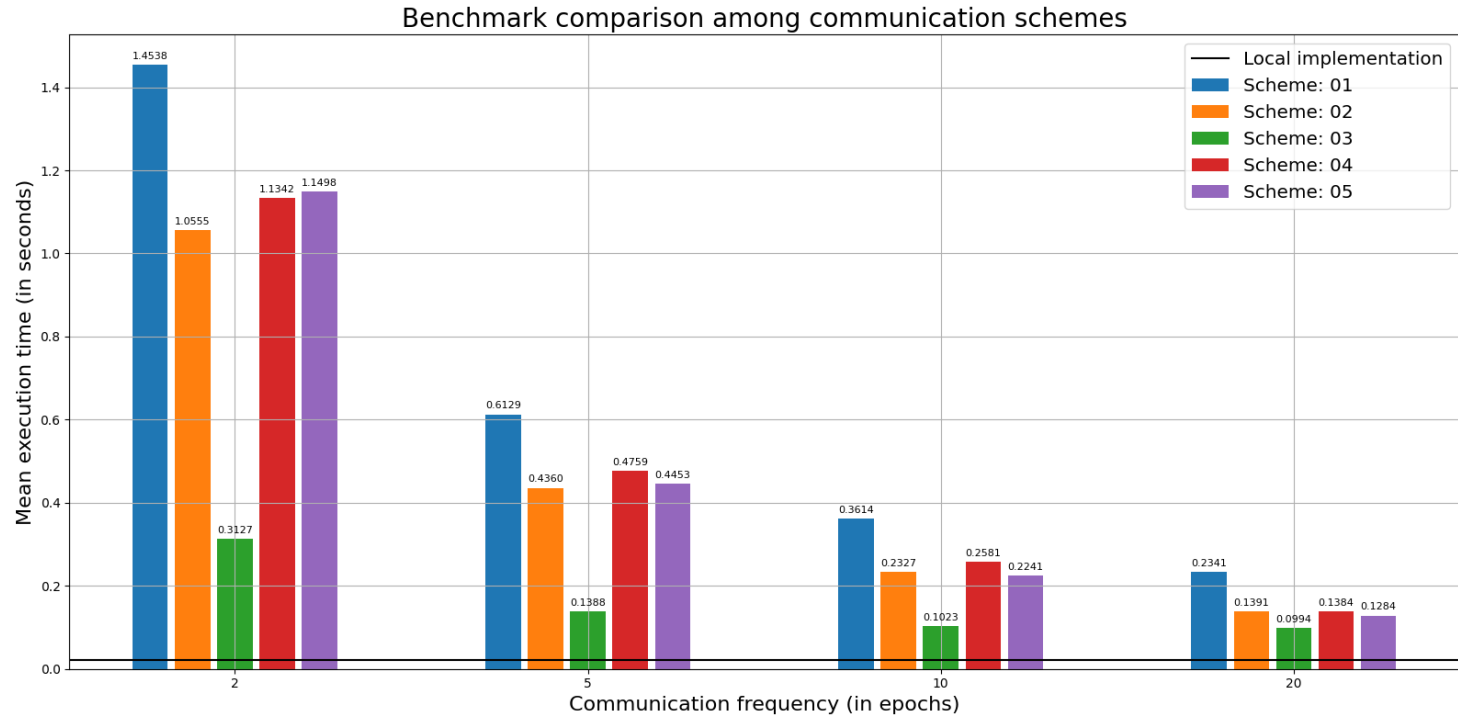
Schema 5

- lgatherv
- Bcast
- Gatherv

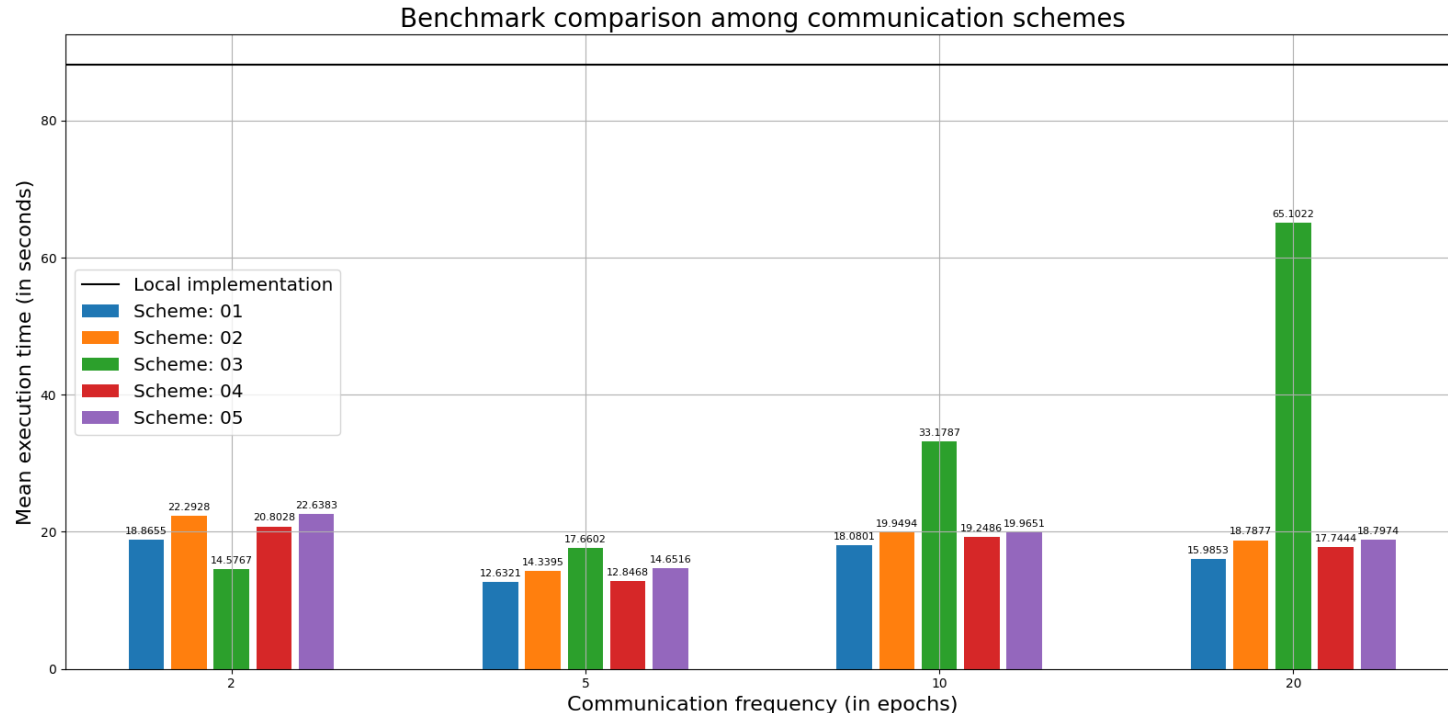


Ergebnisse

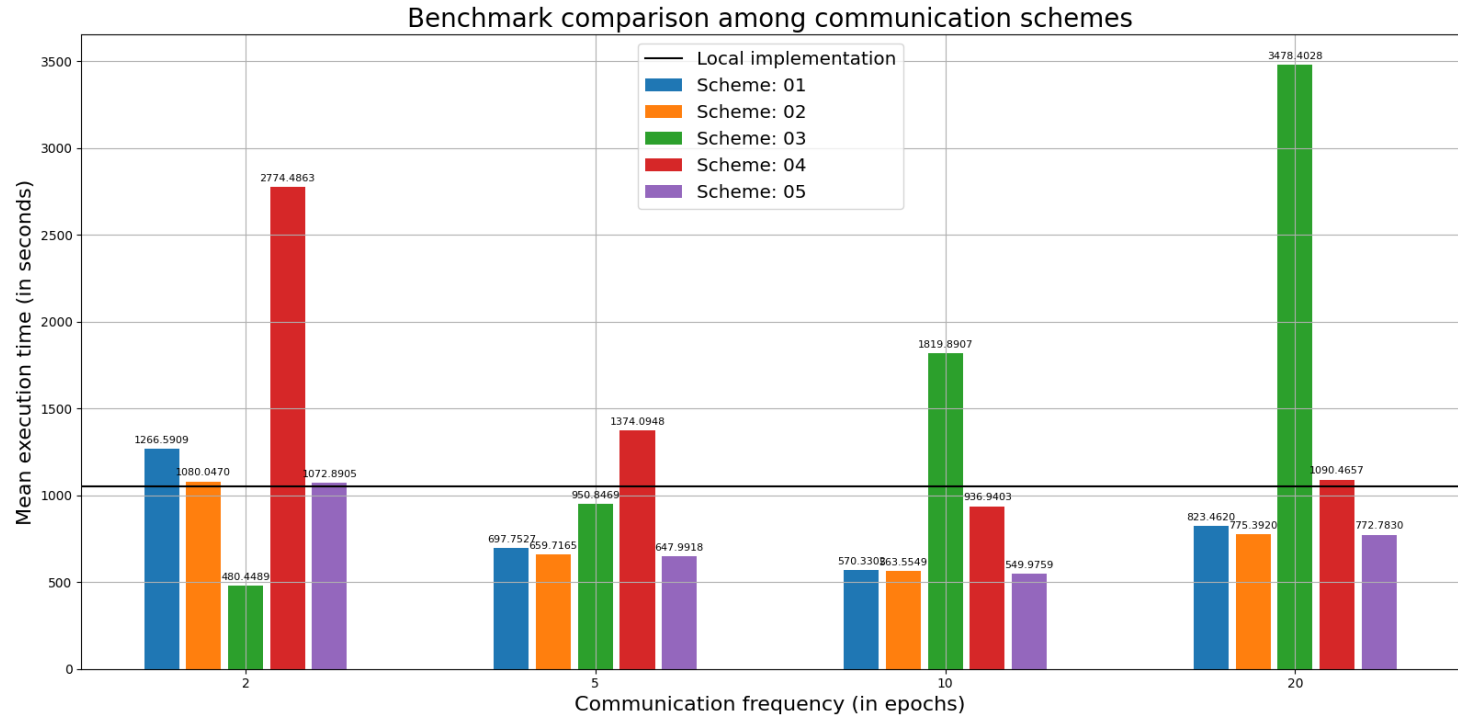
Ausführungszeit (kleiner Datensatz) - 8 Prozessoren, 2 Slots



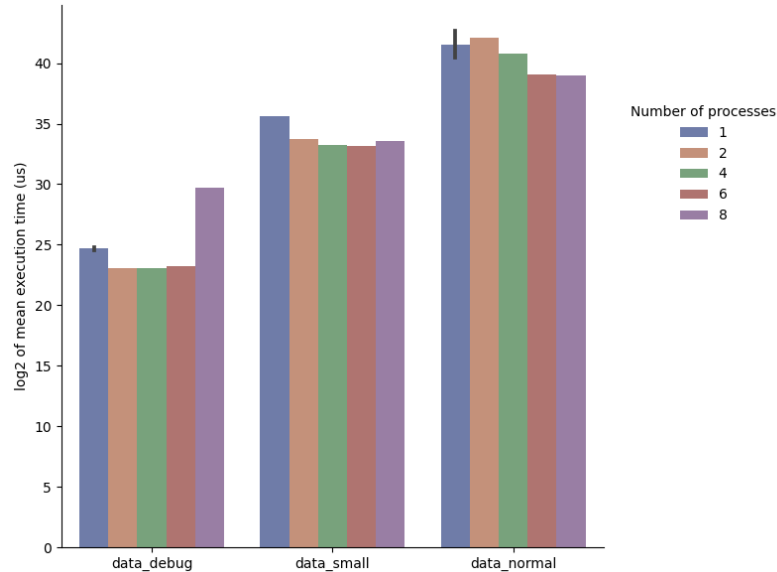
Ausführungszeit (mittlerer Datensatz) - 8 Prozessoren, 2 Slots



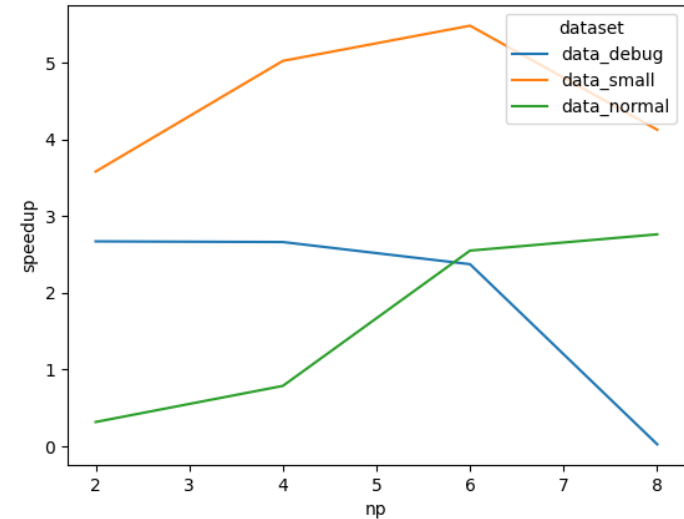
Ausführungszeit (großer Datensatz) - 8 Prozessoren, 2 Slots



Gesamtübersicht der Ergebnisse



Ausführungszeiten bei verschiedenen Datensätzen und variierender Anzahl von Prozessoren (Slots = 2)



Relative Verbesserung der Ausführungszeit in Abhängigkeit der Anzahl von Prozessoren (Slots = 2)

Fazit

Fazit

- Einsatz von OpenMPI bei größeren Datensätzen vorteilhaft
- Synchronisation ist beim verteilten Rechnen keine triviale Aufgabe (Stichpunkt: Konvergenzerkennung des VI-Algorithmus)
- Zusätzliche Verwendung von OpenMP brachte bei unseren Versuchen keine Geschwindigkeitsvorteile
- Schwankungen festgestellt, wenn andere Nutzer auf Eikon Rechnern Berechnungen durchführen
- Für das einlesen von Python Pickle-Dateien existieren momentan keine guten C++ Bibliotheken

Dataset size	# of processors	comm period
↓ <i>low</i>	↓ <i>low</i>	↗ <i>medium/high</i>
↓ <i>low</i>	↑ <i>high</i>	↑ <i>high</i>
↑ <i>high</i>	↓ <i>low</i>	↓ <i>low</i>
↑ <i>high</i>	↑ <i>high</i>	↘ <i>medium/low</i>

Table: Unsere Parameterempfehlung für die Kommunikationsperioden

Vielen Dank für eure Aufmerksamkeit!